Faculty of Science
Department of Mathematics

Course: Numerical Analysis

For

Engineering and Environmental Geology Program

Math 305: Numerical Analysis –

2 Credit (Lecture 2h/w) Prerequisite:Math 212

Content:

Errors and Computer Computations-Solution of nonlinear equations-Direct and iterative methods for linear system of equations-Interpolation-Approximation with polynomials-Numerical Differentiation- Numerical Integration

الأخطاء وحساب الحاسب- حل المعادلات غير الخطية- الطرق المباشرة وغير المباشرة لحل أنظمة المعادلات الخطية- الإستكمال- التقريب باستخدام كثيرات الحدود- التفاضل والتكامل العددي.

References: K. E. Atkinson. An Introduction to Numerical Analysis, John Wiley,1989.

R.L. Burden and J. D. Faires. Numerical analysis, Brook Cole, 3rd Ed., 2002

1- Introduction

Focus of attention in many modern and practical fields is to obtain numerical results that can be extrapolated easily and used directly. The acceleration of scientific development and the growth of industries and reliance on scientific applications has increased the need to develop sports sections that deal with such cases. The most important of these branches is numerical analysis and mathematical programming.

The numerical analysis is the science which aims to derive, describe and analyze the ways to obtain approximate solutions to mathematical problems That suffer from difficulty in solving using analytical methods.

Often there are four reasons to use numerical analysis, namely:

1- When the problem is difficult to solve with analytical methods such as:

Algebraic equations of the fifth degree and above, such as

$$x^6 - 3x^4 - 5x^3 + x + 1 = 0$$

Non-linear algebraic equations contain some functions, such as

$$xe^x - \cos x + 1 = 0$$

Or finite Integrations of functions is difficult to evaluate

$$\int_0^5 e^{x^2} dx$$

2-When the problem is given as a table of schedule resulted from the experience of certain, in this case we have only numbers, It is difficult, as an example, to find an approximate value of the second derivative at x = 0.2, for the function described in the following table

x	0.1	0.2	0.3	0.4	0.5
y	0.0001	0.0016	0.0081	0.0256	0.0625

3-When the problem can be solved by analytical methods, but the result of the solution has some problems in calculating the numerical value, for example, partial differential equation

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2}$$

Has the analytic solution

$$U(x,t) = \frac{8}{\pi^2} \sum_{n=1}^{\infty} \frac{1}{n^2} \left(\sin \frac{1}{2} n\pi \right) (\sin n\pi x) \exp\left(-n^2 \pi^2 t\right)$$

Where $-2 \le x \le 2, t \ge 0$. This solution is difficult to accurately calculate its value at a specified value of x, t .

4- Some analytical methods look like to be applicable in all situations. An example of this type: the solution of algebraic linear system of equations

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

$$
\begin{aligned}
&2x_1 + x_2 + 5x_3 = 11\\
&x_1 - x_2 + 3x_3 = -3\\
&4x_1 + 2x_2 + 6x_3 = 41
\end{aligned}
$$

using determinants or a reverse way. The accuracy of such methods weaken when the system is large.

The numerical analysis strongly requires a good knowledge of one of mathematical programming technique, which helps student to understand such topics of numerical methods and access to numerical solutions of the issues addressed by these methods and plot graphs illustrate the nature of these solutions and show the efficiency of the used numerical methods.

Observers of the technological development in the field of computing and the software will find that many of the software used for scientific programming has been developed in the last two decades. The newer versions of Visual Basic language Appeared as well as C and Fortran.

Newer versions, and more efficient mathematical programming software also appeared such as Matlab, Mathematica, Mathcad and Maple. These software is added to the existence of other languages such as Pascal, Java and the language of PHP.

On the other hand, the Foundations of education in the Arab world gives great importance to the education of students at the level of pre-university basics of dealing with the computer, the Internet, and spreadsheet Excel. Excel is chosen as one of the programming methods to deal with most of what poses in this book.

We add MATLAB to that as a language specialist in mathematical programming through which student may work exercises in the chapters of the book, after work Introduction adequate for this language.

Chapter 1

Errors and Computer Computations

1. Decimal Floating-Point Numbers

Floating point notation

It is similar to what is called scientific notation in high school algebra. For a nonzero number *x*, we can write it in the form

$$x = \sigma \cdot \bar{x} \cdot 10^{e} \qquad (1)$$

with e is an integer, $\bar{x} = (a_1.a_2a_3 \dots a_n)_{10}$, $1 \leq \bar{x} < 10$ and $\sigma = \pm 1$. Thus

124.62 = (1.2462) 10^2 , with $\sigma = +1$, $e = 2$, $\bar{x} = 1.2462$.

The general form for the floating point notation for decimal numbers is

$$x = \sigma \cdot \bar{x} \cdot 10^{e} = \sigma (a_1.a_2a_3 \dots a_n)_{10} 10^{e}$$

with $a_1 \neq 0$, so that there are *n* decimal digits in the significand $\bar{x}$.

On a decimal computer or calculator, we store *x* by instead storing σ,$\bar{x}$ and e .

We must restrict the number of digits in $\bar{x}$ and the size of the exponent e.

For example, on a Nokia 6610 Mobile calculator, the number of digits kept in $\bar{x}$ is 8, and the exponent is restricted to $-99 \leq e \leq 99$.

Binary Floating-Point Numbers

With MATLAB, we can define the corresponding binary numbers for integers from zero to ten. Here is the program:

Code	Results	
i = (0:10); j=dec2base(i,2)	j = 0000 0001 0010 0011 0100	0101 0110 0111 1000 1001 1010

We now do something similar with the binary representation of a number x. Write

$$x = \sigma \cdot \bar{x} \cdot 2^{e} \quad (2)$$

with $1 \leq \bar{x} < (10)_2$
and e an integer. For example,

$$x = 11011.0111$$

then $\sigma=+1$, $e=4=(100)_2$ and $\bar{x}=(1.10110111)_2$
so

$$x = 1.10110111 \cdot 2^{4}$$

The number x is stored in the computer by storing the σ, $\bar{x}$, and e. On all computers, there are restrictions on the number of digits in $\bar{x}$ and the size of e.

Floating Point Numbers Representation

When a number x outside a computer or calculator is converted into a machine number, we denote it by $fl(x)$. On an HP-calculator,

$$fl\left(\frac{1}{3}\right) = (3.333333333)_{10}10^{-1}$$

The decimal fraction of infinite length will not fit in the registers of the calculator, but the latter 10-digit number will fit.

On a binary computer, we use a similar notation. We shall concentrate on a particular form of computer floating point number, that called the ***IEEE*** floating point standard, reffered to ***Instite of Electrical and Electronics Engineers*** .

In ***single precision***, we write such a number as

$$fl(x) = \sigma(1.a_2a_3 \ldots a_{24})_2 \cdot 2^e$$

The significand $\bar{x} = (1.a_2 \ldots a_{24})_2$ immediately satisfies

$$1 \le \bar{x} < (10)_2 = 2_{10}.$$

The number x will be stored in the computer as follows: We store σ as a single bit, the significand $\bar{x}$ as 24 bits, and the exponent 8 bits , i.e.

$$-(1111110)_2 \le e < (1111111)_2$$

In actuality, the limits are

$$-(126)_{10} \le e < (127)_{10}$$

In Double precision arithmetic, we have:

$$-(1022)_{10} \le e < (1023)_{10}$$

$$fl(x) = \sigma(1.a_2a_3 \ldots a_{53})_2 \cdot 2^e$$

The Machine Epsilon

It is a widely used measure of the accuracy possible in representing numbers in the machine.

It equal the difference between 1 and the smallest number representable in the machine arithmetic that is greater than 1

Let y be the smallest number representable in the machine arithmetic that is greater than 1.

The machine epsilon is $\eta = y - 1$.

The number 1 has the simple floating point representation

$$1 = (1.0...0)_2 \, 2^0$$

What is the smallest number that is greater than 1? It is

$$1 + 2^{-23} = (1.0...01)_2 \, 2^0 > 1$$

and the machine epsilon in IEEE single precision floating point format is

$$\eta = 2^{-23} = 1.19 \text{x} 10^{-7}$$

This mean that the number $1 + 2^{-24}$ can not be represented exactly in this format

In Double precision arithmetic we have

$$\eta = 2^{-52} = 2.22 \text{x} 10^{-16}$$

2. Rounding And Chopping

Let us first consider these concepts with decimal arithmetic. We write a computer floating point number z as

$$z = \sigma \cdot \bar{x} \cdot 10^e = \sigma (a_1 . a_2 a_3 \dots a_n)_{10} 10^e$$

with $a_1 \neq 0$, so that there are n decimal digits in the significand $\bar{x} = (a_1 . a_2 a_3 \dots a_n)_{10}$.

Given a general number

$$x = \sigma \cdot \bar{x} \cdot 10^e = \sigma (a_1 . a_2 a_3 \dots a_n \dots)_{10} 10^e$$

, $a_1 \neq 0$

we must shorten it to fit within the computer. This is done by either ***chopping*** or ***rounding.***

The floating point ***chopped*** version of x is given by

$$fl(x) = \sigma (a_1 . a_2 a_3 \dots a_n)_{10} 10^e$$

where we assume that e fits within the bounds required by the computer.

For the ***rounded*** version, we must decide whether to round up or round down. A simplified formula is

$$fl(x) = \begin{cases} \sigma(a_1.a_2a_3\dots a_n)_{10}10^e, & a_{n+1} < 5 \\ \sigma[(a_1.a_2a_3\dots a_n)_{10} + (0.00\dots1)_{10}]10^e, & a_{n+1} \geq 5 \end{cases}$$

The term $(0.00\dots1)_{10}$ denotes $10^{-(n-1)} = 10^{-n+1}$ giving the ordinary sense of rounding with which you are familiar.

Chopping/Rounding In Binary

Let $x = \sigma(1.a_2a_3\dots a_n\dots)_2 2^e$
with all $a_i, i = 2,3,\dots,n$ equal to 0 or 1. Then for a chopped floating point representation, we have

$$fl(x) = \sigma(a_1.a_2a_3\dots a_n)_2 2^e$$

For a rounded floating point representation, we have

$$fl(x) = \begin{cases} \sigma(a_1.a_2a_3\dots a_n)_2 2^e, & a_{n+1} = 0 \\ \sigma[(a_1.a_2a_3\dots a_n)_2 + (0.00\dots1)_2]2^e, & a_{n+1} = 1 \end{cases}$$

Example 1:

[Q32] Let $x > 0$ has been represented using a positive binary floating-point representation with n bits of precision in the significand. Assume that ***chopping*** is used in going from a number x outside the computer to its floating-point approximation $fl(x)$, inside the computer

(a) Show that $0 \leq x - fl(x) \leq 2^{e-n+1}$

(b) Show that $x \geq 2^e$ and use (a) to show

$$\frac{x - fl(x)}{x} \leq 2^{-n+1}$$

Answer:

(a)the value of x is

$x = \sigma(1.a_2a_3 \ldots a_n \ldots)_2 2^e$ (1)

Let n is the number of digits available in the computer precision. The floating point reresentation for x is

$$fl(x) = \sigma(1.a_2a_3 \ldots a_n)_2 2^e \qquad (2)$$

using chopping in going from a number x outside the computer to its binary floating-point approximation $fl(x)$, inside the computer then

$$fl(x) \le x$$

and hence

$$0 \le x - fl(x)$$

$$= \sigma\,(1.a_2 \ldots a_n a_{n+1} \ldots\ldots)_2\ 2^e - \sigma\,(1.a_2 \ldots a_n)_2\ 2^e$$

$$= (0.\,a_{n+1}a_{n+2}a_{n+3} \ldots\ldots)_2 \bullet 2^{e-(n-1)}$$

$$\le 2^{e-(n-1)}$$

Since $0.\,a_{n+1}a_{n+2}a_{n+3} \le 1$

(b) We have from (a)

$$x - fl(x) \le 2^{e-n+1}$$

(3)

Since $x \ge 2^e$ and by division

$$\frac{x - fl(x)}{x} \le \frac{2^{e-n+1}}{2^e} = 2^{-n+1}$$

(4)

2- Error Measurements

Basic Definitions

Let x_T denote the true value of some number, usually unknown in practice; and let x_A denote an approximation of x_T.

$E(X_A) = X_T - X_A$,

$AE(X_A) = |X_T - X_A|$

$Rel(X_A) = |X_T - X_A| / X_T$

Example(1):

$x_T = \pi = 3.14159265$, $x_A = 22/7 = 3.1428571$

Then

$E(X_A) = \pi - 22/7 = 3.14159265 - 3.1428571 = -0.00126$

$Rel(X_A) = = -0.00126/3.14159265 = -0.000402$

Example(2):

Four students take four distances to be measured .
Their results are as follows

	1	2	3	4
F	100	20	200	400
P	104	19	194	390

Where *f* is the exact value and *P* is the measured value
Which of them is the most accurate?

Answer:

The absolute errors are

1	2	3	4
4	1	6	10

So the 2^{nd} is the best

Taking the relative error

1	2	3	4
0.04	0.05	0.03	0.025

Thus the 4^{th} is the best

Indeed the relative error is the best error measurement , since it takes into account the size of the exact solulion

Sources Of Error

This is a very rough categorization of the sources of error in the calculation of the solution of a mathematical model for some physical situation

A) Basic Errors

Modelling Error:

As an example, if a projectile of mass m is travelling thru the earth's atmosphere, then a popular description of its motion is given by

$$m\frac{d^2\mathbf{r}(t)}{dt^2} = -mg\mathbf{k} - b\frac{d\mathbf{r}}{dt}$$

with b ≥ 0. In this, *r(t)* is the vector position of the projectile; and the final term in the equation represents frictionاحتكاك. If there is an error in this a model of a physical situation, then the numerical solution of this equation is not going to improve the results

programming errors*:*

that often means **programming errors**. They are often embedded in very large codes which may mask their effect. Some simple rules:

(i) Break programs into small testable subprograms.
(ii) Run test cases for which you know the outcome.
(iii) When running the full code, look at the output, checking whether the output is reasonable or not.

Observational Error:

The radius of an electron is given by

$$C = (2.997925 + \varepsilon) \bullet 10^{10}\, cm/\sec,$$
$$|\varepsilon| \le 0.000003$$

This error cannot be removed, and it must affect the accuracy of any computation in which it is used. We need to be aware of these effects and to so arrange the computation as to minimize the effects.

Rounding/chopping Error:

This is the main source of many problems, especially problems in solving systems of linear equations. We later look at the effects of such errors.

Approximation Error:

This is also called “discretization error” and “truncation error”; and it is the main source of error with which we deal in this course. Such errors generally occur when we replace a computationally unsolvable problem with a nearby problem that is more tractable computationally.

For example, To evaluate

$$I = \int_0^1 e^{-x^2} dx$$

We usw the Taylor polynomial approximation

$$e^{-x^2} \cong 1 - x^2 + \frac{x^4}{2!} - \frac{x^6}{3!} + \frac{x^8}{4!}$$

Which contains an “approximation error”. Thus the numerical integration

$$I = \int_0^1 \left[1 - x^2 + \frac{x^4}{2!} - \frac{x^6}{3!} + \frac{x^8}{4!}\right] dx$$

Can easily obtained, but it contains an approximation error

B) Consequent Errors

Loss Of Significance Errors

This can be considered a source of error or a consequence of the finiteness of machine arithmetic. It occurs when we substract an approximately equall two numbers.

We begin with some illustrations.

Example Define

$$f(x) = x\left[\sqrt{x+1} - \sqrt{x}\right]$$

and consider evaluating it with an increasing positive values of x on a 3-digit decimal software which uses rounded arithmetic. The values of *f (x)*:

x	$\sqrt{x+1}$	$\sqrt{x}$	$\sqrt{x+1}-\sqrt{x}$	$x\left[\sqrt{x+1}-\sqrt{x}\right]$
10	3.32	3.162	0.158	1.58
100	10.0	10.0	0.00	0.00

For 4-digit decimal, we have:

x	$\sqrt{x+1}$	$\sqrt{x}$	$\sqrt{x+1}-\sqrt{x}$	$x\left[\sqrt{x+1}-\sqrt{x}\right]$
10	3.317	3.1623	0.1547	1.547
100	10.05	10.00	0.05	5.0
1000	31.64	31.62	0.02	2.0
10000	100	100	0	0

We notice that the error is still small until the 16-digit decimal is reached, then the error become large.

We can write

$$f(x) = x\left[\sqrt{x+1} - \sqrt{x}\right]$$

$$= x\left[\sqrt{x+1} - \sqrt{x}\right]\frac{\sqrt{x+1}+\sqrt{x}}{\sqrt{x+1}+\sqrt{x}} = \frac{x}{\sqrt{x+1}+\sqrt{x}}$$

Then the function

$$f(x) = \frac{x}{\sqrt{x+1}+\sqrt{x}} \qquad (2)$$

is used with no loss of significance errors.

Comparing (1) and (2), the previous two tables becomes

x	$x\left[\sqrt{x+1}-\sqrt{x}\right]$	$\frac{x}{\left[\sqrt{x+1}+\sqrt{x}\right]}$	Error
10	1.58	1.45	0.13
100	0.00	5.00	5.00

x	$x\left[\sqrt{x+1}-\sqrt{x}\right]$	$\frac{x}{\left[\sqrt{x+1}+\sqrt{x}\right]}$	Error
10	1.547	1.543	0.004
100	5.0	4.988	0.012
1000	2.0	15.81	13.81
10000	0	50	50

Using Excel, we have

x	$x\left[\sqrt{x+1}-\sqrt{x}\right]$	$\frac{x}{\left[\sqrt{x+1}+\sqrt{x}\right]}$	Error
10	1.543471302	1.543471302	1.4386E-15
100	4.987562112	4.987562112	3.2862E-14
10000	49.99875006	49.99875006	1.0651E-11
1E+8	5000.000056	4999.999988	6.8088E-05
1E+9	15811.39077	15811.3883	0.00247045
1E+10	49999.94417	50000	0.05583153
1E+11	158115.2901	158113.883	1.40709731
1E+12	500003.8072	500000	3.80724681
1E+13	1578591.764	1581138.83	2547.06611
1E+14	5029141.903	5000000	29141.9029
1E+15	0	15811388.3	15811388.3

NOISE IN FUNCTION EVALUATION

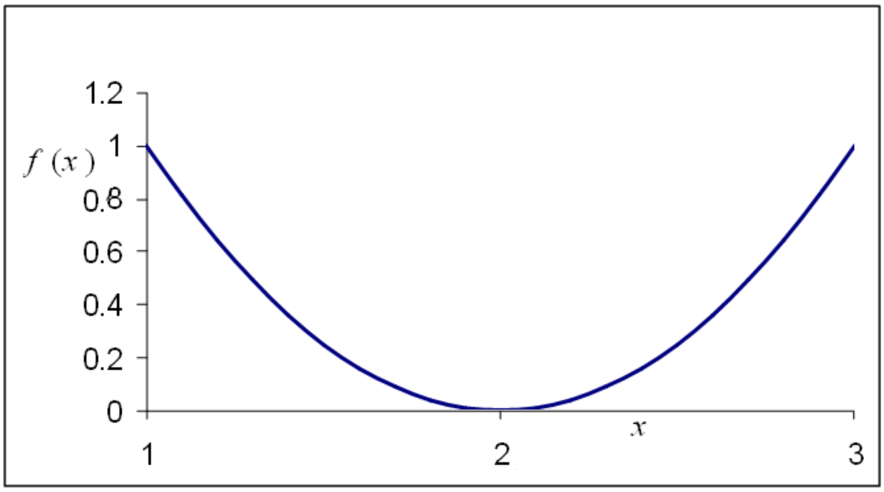
Whenever a function $f\ (x)$ is evaluated, there are arithmetic operations carried out which involve rounding or chopping errors. This means that what the computer eventually returns as an answer contains noise. This noise is generally “random” and small. But it can affect the accuracy of other calculations which depend on $f\ (x)$.

For example, we illustrate the evaluation of

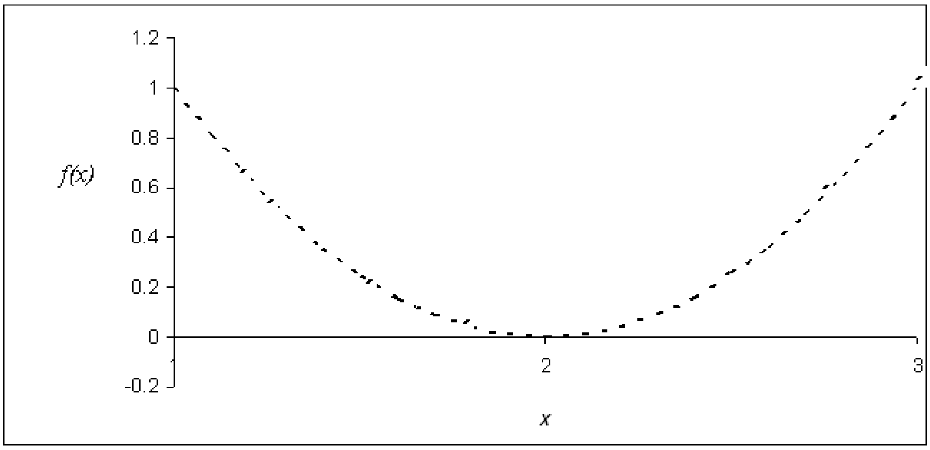
$$f\ (x) = x^2 - 4x + 4$$

which is simply $(x\text{-}1)^3$ and has only a single root at x=1. We use MATLAB with its IEEE double precision arithmetic and standard rounding.

The following Figure contains the graph of the computed values of $f(x)$ for $0 \leq x \leq 2$, Note that the graph of $f(x)$ does not appear to be taken from a continuous curve, but rather, it is a narrow "fuzzy band" of seemingly random values. This is true of all parts of the computed curve of $f(x)$, but it becomes evident only when you look at the curve very closely.



Real plot of $f(x)$



Numerical plot of $f(x)$

Underflow And Overflow Errors

Underflow Errors

If we use numbers that are too small for the floating-point format will lead to underflow errors.
The variable x which agrees underflow is set to be zero from the machine. The programer and the user needs to be aware of such errors. This kind of error does not often stop the programm runnig.

Programming Exercise 1:
Use your favorite program to generate an underflow error on your computer, write a program to repeatedly divide by 10^{10} of a number x < 1 and print the result Eventually. you will exceed your machine's exponent limit for floating-point numbers.

x=10e+19; for i=1 : 4 y=1/(x^i)^i^2 end	y =1.0000e-40 y =1.0000e-160 y =0 y=0

Overflow Errors

If we use numbers that are too large for the floating-point format will lead to overflow errors. These are generally fatal errors on most computers. With the IEEE floating-point format, overflow errors can be carried along as having a value of $\pm\infty$ or NaN, depending on the context. Usually an overflow error is an indication of a more significant problem or error in the program and the user needs to be aware of such errors.

This kind of error often stops the programm runnig and give an error massage.

Programming Exercise

Use your favorite program to generate an overflow error on your computer, write a program to repeatedly square a number x > 1 and print the result Eventually. you will exceed your machine's exponent limit for floating-point numbers.

x=10e9; for i=1 : 4 y=(x^i)^i^2 end	y =1.0000e+20 y =1.0000e+80 y =1.0000e+180 y= Inf

[Q33] Define Loss of significant error, show how you can avoid it in the following cases(x close to zero)

[1] $f(x) = log(x+1) - log(x)$, x is large.

[2] $f(x) = \sqrt[3]{1+x} - 1$

[3] $f(x) = \dfrac{\sqrt{4+x} - 2}{x}$

[Q34] Use Tylor expansion to avoid Loss of significant error in the following cases(x close to zero)

[1]

(a) $f(x) = \dfrac{1-e^{-x}}{x}$ (b) $f(x) = \dfrac{\log(1-x) + xe^{x/2}}{x^3}$

(c) $f(x) = \dfrac{x - \sin(x)}{x^3}$ (d) $f(x) = \dfrac{x - \sin(x)}{\tan(x)}$

3. Error Propagation

4.1 Propagation In Arithmetic Operations

Example

[Q35] Let x_T , y_T and ω denotes the true values for two numbers and the arithmetic operation. Let also x_A , y_A , w^* are the machine values and arithmetic operation.

Show that the process of rounding or chopping introduce a relatively small error into the copmuted value of $x_A\, w^*\, y_A$.

Answer

Let ω denote arithmetic operation such as +, −, *, or /. Let x_T , y_T be the true values and x_A , y_A are the machine values including rounding or chopping error.

The term $E = x_T\, \omega\, y_T - x_A\, \omega\, y_A$
is called ***the propagated error***;

Now, let ω^* denote the same arithmetic operation as it is actually carried out in the computer, including rounding or chopping error.

We want to obtain $x_T\, \omega y_T$, but we actually obtain $x_A\ \omega^*\ y_A$. The error in this operation is given by

$$x_T\, \omega\, y_T - x_A\, \omega^*\, y_A = [x_T\, \omega\, y_T - x_A\, \omega\, y_A] + [x_A\, \omega\, y_A - x_A\, \omega^*\, y_A]$$

The final term in the error introduced by the inexactness of the machine arithmetic. We may call it the machine error. The first term is the propagated error.

the machine error is often known and bounded. This implies that the resulting error depends on the the propagated error. When using IEEE arithmetic operation, we have

$$x_A\ \omega^*\ y_A = fl(x_A\ \omega\ y_A) \qquad (1)$$

This means that the quantity $x_A \omega y_A$ is computed exactly and is then rounded or chopped to fit the answer into the floating point representation of the machine.

The formula (1) implies

$$x_A\ \omega^*\ y_A = x_A\ \omega\ y_A\ (1+\varepsilon) \qquad (2)$$

with some limits given for ε. Manipulating (2), we have

$$\mathrm{Rel}(x_A\ \omega^*\ y_A) = -\varepsilon$$

Thus the process of rounding or chopping introduce a relatively small error in the copmuted value.

We now examine ***<u>the propagated error</u>*** for particular cases.

Propagation In Summation and Subtraction

For ω equal to − or +, we have

$[x_T \pm y_T] - [x_A \pm y_A] = [x_T - x_A] \pm [y_T - y_A]$

Thus the error in a sum is the sum of the errors in the original values, and similarly for subtraction.

Example

[Q36] Let xA = 1.36 and yA = 5.431 be correctly rounded from x_T and y_T, to the number of digits shown. Then give a bound for the propagated error in the summation and division processes.

Answer:

$x_A = 1.36$ and $y_A = 5.431$

$|x_A - x_T| \le 0.005, \quad |y_A - y_T| \le 0.0005,$

or. equivalently,

$$1.355 \le x_T < 1.365, \quad 5.4305 \le y_T < 5.4315 \qquad (1)$$

$x_A = 1.36$ and $y_A = 5.431$

For the operation of addition

$$x_A + y_A = 6.791 \qquad (2)$$

For the true value, use (1) to obtain the bounding interval

$$1.355 + 5.4305 \le x_T + y_T \le 1.365 + 5.4315$$

$$6.7865 \le x_T + y_T \le 6.7965 \qquad (3)$$

To obtain a bound for the propagated error, subtract (2) from (3) to get

$$-0.0045 \le (x_T + y_T) - (x_A + y_A) \le 0.0055$$

With division, $$\frac{x_A}{y_A} = \frac{1.36}{5.431} = 0.25041 \qquad (4)$$

Also, For the true value, use (1) to obtain the bounding interval

$$1.355 \le x_T < 1.365, \quad 5.4305 \le y_T < 5.4315 \qquad (1)$$

$$\frac{1.355}{5.4305} \le \frac{x_T}{y_T} \le \frac{1.365}{5.4315}$$

Dividing the fractions and rounding to seven digits. we obtain

$$0.24951 \le \frac{x_T}{y_T} \le 0.25131 \qquad (5)$$

To obtain a bound for the propagated error, subtract (4) from (5) to get

$$-0.0001 \le \frac{x_T}{y_T} - 0.252256 \le 0.0001$$

This technique of obtaining an interval that is guaranteed to contain the true answer is called interval arithmetic.

It is a useful technique, and it has been implemented on computers. both using software and hardware. But for extended calculations, interval arithmetic must be implemented with a great deal of care or else it will lead to predicted error bounds that are far in excess of the true error.

Propagation In Multiplication

Example

Lemma:

[Q37] Small relative errors in the values of x_A and y_A leads to a small relative error in the product $x_A\, y_A$.

Proof:

Consider first ω = *. Then for the relative error in

$$x_A * y_A = x_A y_A \ ,$$

$$\text{Rel}\,(x_A y_A) = \frac{x_T y_T - x_A y_A}{x_T y_T}$$

Write

$$x_T = x_A + \xi, \qquad y_T = y_A + \eta$$

Then

$$
\begin{aligned}
\text{Rel}(x_A y_A) &= \frac{x_T y_T - x_A y_A}{x_T y_T} \\
&= \frac{x_T y_T - (x_T - \xi)(y_T - \eta)}{x_T y_T} \\
&= \frac{x_T \eta + y_T \xi - \xi\eta}{x_T y_T} \\
&= \frac{\xi}{x_T} + \frac{\eta}{y_T} - \frac{\xi}{x_T} \cdot \frac{\eta}{y_T} \\
&= \text{Rel}(x_A) + \text{Rel}(y_A) - \text{Rel}(x_A) \cdot \text{Rel}(y_A)
\end{aligned} \tag{4}
$$

Since we usually have

$$|\text{Rel}(x_A)|, \; |\text{Rel}(y_A)| \ll 1$$

the relation (4) says

$$\text{Rel}(x_A y_A) \approx \text{Rel}(x_A) + \text{Rel}(y_A)$$

Thus small relative errors in the values of x_A and y_A leads to a small relative error in the product $x_A \; y_A$.

Also, note that there is some cancellation if these relative errors are of opposite sign.

Propagation In Division

There is a similar result for division:

$$\text{Rel}\left(\frac{x_A}{y_A}\right) \approx \text{Rel}(x_A) - \text{Rel}(y_A)$$

Provided

$$|\text{Rel}(y_A)| \ll 1$$

Propagation In Function Evaluation

Suppose we evaluate a function $f(x)$ in the machine. Then the result is generally not $f(x)$, but rather an approximate of it which we denote by $\tilde{f}(x \;\;)$. Now suppose that we have a number $x_T \cong x_A$.

We want to calculate $f(x_T)$, but instead we evaluate $\tilde{f}(x_A)$.

The error in this computed quantity is

$$f(x_T) - \tilde{f}(x_A) = [f(x_T) - f(x_A)] + [f(x_A) - \tilde{f}(x_A)]$$

The quantity $f(x_T) - f(x_A)$ is called ***the propagated error***; and it is the error that results from using perfect arithmetic in the evaluation of the function.

If the function $f(x)$ is differentiable, then we can use the "mean-value theorem" to write

$$f(x_T) - f(x_A) \approx f'(c)(x_T - x_A)$$

for some c between x_T and x_A.

Since usually x_T and x_A are close together, we can say c is close to either of them, and

$$f(x_T) - f(x_A) \approx f'(x_T)(x_T - x_A) \qquad (1)$$

Example

[Q38] Consider evaluation $f(x) = a^x$, where a is a positive real number, approximate the propagated error. Then evaluate the condition number for the computation and show how it affects in the accuracy.

==============

Answer

Define $f(x) = a^x$

Then (1) yields

$$a^{x_T} - a^{x_A} \cong (\log a) a^{x_T}(x_T - x_A)$$

divide by a^{x_T}

$$\frac{a^{x_T} - a^{x_A}}{a^{x_T}} \cong (\log a)(x_T - x_A)$$

Multiply on x_T up and down in the RHS

$$Rel(a^{x_A}) = Rel(f(x_A)) \cong x_T(\log a)\frac{x_T - x_A}{x_T}$$

$$= x_T(\log a)\, Rel(x_A) = K \cdot Rel(x_A)$$

with $K = x_T(\log a)$.

Note that if K = 10^4 and Rel(x_A) = 10^{-7}, then

$$Rel\left(b^{x_A}\right) = 10^{-3}$$

This is a large decrease in accuracy; and it is independent of how we actually calculate a^x .

Then number K is called a ***condition number*** for the computation.

4. Stable and Unstable Computations

In this section we introduce another idea that occurs repeatedly in numerical analysis:
the different between numerical processes that are stable and those that are not. Closely related are the concepts of well-conditioned problems and badly-conditioned problems.

Numerical Instability

We say that a numerical process is ***unstable*** if small errors made at one stage of the process are magnified in subsequent stages and decrease the accuracy of the overall calculation.

Example

[Q39] Consider the sequence of real numbers defined inductively by

$x_0 = 1,\ x_1 = ⅓,$ ....(1.a)

$x_{n+1} = \frac{13}{3} x_n - \frac{4}{3} x_{n-1},\ n \geq 1$ .... (1.b)

Which has the solution

$$x_n = \left(\frac{1}{3}\right)^n \quad \text{.. (2)}$$

check the validity of the solution. Then discuss the stability of the solution.

Answer

we can check the validity of (2) as a solution of (1) via Induction as follows:

Equation (2) is obviously true for $n = 0$ and 1. If its validity is granted for $n=k$, then substituting from (2) in (1) we obtain

$$x_{k+1} = \tfrac{13}{3}x_k - \tfrac{4}{3}x_{k-1} = \tfrac{13}{3}\left(\tfrac{1}{3}\right)^k - \tfrac{4}{3}\left(\tfrac{1}{3}\right)^{k-1}$$
$$= \left(\tfrac{1}{3}\right)^{k-1}\left[\tfrac{13}{9} - \tfrac{4}{3}\right]$$
$$= \left(\tfrac{1}{3}\right)^{k+1}$$

The solution of the recurrence formula

$$x_0=1,\ x_1=⅓,$$
$$x_{n+1}=\frac{13}{3}x_n-\frac{4}{3}x_{n-1},\ n\geq 1$$

is unstabe for two resons:

1-Any error present in x_n is multiplied by 13/3 in computing x_{n+1}.

Hence it propagates more than four times in each iteration

2- the solution formula $x_n=\left(\frac{1}{3}\right)^n$ becomes small and continues decreasing as n increases. So the relative error compare to the increasing error become big.

Excercises

[Q40] Consider the identity

$$\int_0^x \sin(xt)\,dt = \frac{1-\cos x^2}{x}$$

Explain the difficulty in using the right-hand fraction to evaluate this expression when x is close to zero. Give a way to avoid this problem.

[Q41] Consider the sequence of real numbers defined inductively by

$$x_{n+1}=100.01x_n-x_{n-1},\ n\geq 1 \qquad (1)$$

Which has the general solution

$$x_n=A\,(100)^n+B\left(\frac{1}{100}\right)^n \qquad (2)$$

Find the solution that satisfies the conditions:

$$x_0=1,\ x_1=0.01, \qquad (3)$$

and then discuss the stability of the solution.

Chapter 2
Solution of linear system equations
Principle of Linear System

Systems of linear equations arise in a large number of areas, both directly in modeling physical situations and indirectly in the numerical solution of other mathematical models.

These applications occur areas of the physical, biological, and social sciences. In addition, linear systems are involved in the following: optimization theory; solving systems of nonlinear equations; the approximation of functions; the numerical solution of boundary value problems for ordinary differential equations, partial differential equations, and integral equations; statistical inference; and numerous other problems.

Because of importance of linear systems, much research has been devoted to their numerical solution. Excellent algorithms have been developed for the most common types of problems for linear systems, and some of these are defined, analyzed, and illustrated in this chapter.

The most common type of problem is to solve a square linear system

$$AX = B$$

of moderate order, with coefficients that are mostly nonzero. Such linear systems, of any order, are called ***dense***. For such systems, the coefficient matrix A must generally be stored in the main memory of the computer in order to efficiently solve the linear system, and thus memory storage limitations in most computers will limit the order of the system.

With the rapid decrease in the cost of computer memory, quite large linear systems can be accommodated on some machines, but it is expected that for most smaller machines, the practical upper limits on the order will be of size 100 to 500. Most algorithms for solving such dense systems are based on *Gaussian elimination,* which is defined in Section 4.1. It is a direct method in the theoretical sense that if rounding errors are ignored, then the exact answer is found in a finite number of steps.

A second important type of problem is to solve $Ax = b$ when: A is square, sparse, and of large order. A ***sparse*** matrix is one in which most coefficients are zero. Such systems arise in a variety of ways, but we restrict our development to those for which there is a simple, known pattern for the nonzero coefficients.

These systems arise commonly in the numerical solution of partial differential equations. Iteration methods are the preferred method of solution, and these
are introduced in Section 8.6 through Section 8.9.

In this chapter, direct techniques are considered to solve the linear system

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots\ldots\ldots\ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots\ldots\ldots\ldots + a_{2n}x_n &= b_2 \\ &. \\ &. \\ &. \\ a_{n1}x_1 + a_{n2}x_2 + \ldots\ldots\ldots\ldots + a_{nn}x_n &= b_n \end{aligned} \qquad (1)$$

Which can be written in matrix form as

$$AX = B$$

Where

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots\ldots\ldots\ldots & a_{1n} \\ a_{21} & a_{22} & \ldots\ldots\ldots\ldots & a_{2n} \\ \ldots & & & \\ \ldots & & & \\ a_{n1} & a_{n2} & \ldots\ldots\ldots\ldots & a_{nn} \end{bmatrix}, \quad X = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ x_n \end{bmatrix}, B = \begin{bmatrix} b_1 \\ b_2 \\ . \\ . \\ b_n \end{bmatrix}$$

As an Example:

$$2x_1 + x_2 + 5x_3 = 11$$
$$x_1 - x_2 + 3x_3 = -3$$
$$4x_1 + 2x_2 + 6x_3 = 41$$

Can be written as:

$$\begin{bmatrix} 2 & 1 & 5 \\ 1 & -1 & 3 \\ 4 & 2 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 11 \\ -3 \\ 41 \end{bmatrix}$$

Methods for solving this problem are classified into two categories:

1-Direct method such as Gaussian elimination.

2- Indirect or Iteration methods

1 Gaussian Elimination

This is the formal name given to the method of solving systems of linear equations by successively eliminating unknowns and reducing to systems of lower order. It is the method most people learn in high school algebra or in an undergraduate linear algebra course (in which it is often associated with producing the row-echelon form of a matrix). A precise definition is given of Gaussian elimination, which is necessary when implementing it on a computer and when analyzing the effects of rounding errors that occur when computing with it.

provided $a_{11} \neq 0$, perform $E_j = E_j - \frac{a_{j1}}{a_{11}} E_1$ for $j = 2, 3, \ldots n$. This step will eliminate the Coeficient of x_1 in each of these rows. The resulting system is

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 && E_1 \\
\bar{a}_{22}x_2 + \cdots + \bar{a}_{2n}x_n &= \bar{b}_2 && E_2 \\
\bar{a}_{32}x_2 + \bar{a}_{33}x_3 + \cdots + \bar{a}_{3n}x_n &= \bar{b}_3 && E_3 \\
&\vdots \\
\bar{a}_{n2}x_2 + \bar{a}_{n3}x_3 + \cdots + \bar{a}_{nn}x_n &= \bar{b}_n && E_n
\end{aligned}
$$

we change the symbol of Coef. from $a_{..}$ to $\bar{a}_{..}$ and b to $\bar{b}$. Since we expect that entries in rows 2, 3, ..., n will be changed.

For ease of notation we again denote the entry in the ith row and jth column by a_{ij}.

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \qquad E_1$$

$$a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \qquad E_2$$

$$a_{32}x_2 + a_{33}x_3 + \cdots + a_{3n}x_n = b_3 \qquad E_3$$

$$\vdots$$

$$a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n = b_n \qquad E_n$$

Provided that $a_{ii} \neq 0$, perform

$$E_j = E_j - \frac{a_{ji}}{a_{ii}} E_i \quad , \; j = i+1, i+2, \ldots, n$$

This sequential step will eliminate x_i in each row below the ith one.

The resulting system is

Provided that $a_{ii} \neq 0$, perform

$$E_j = E_j - \frac{a_{ji}}{a_{ii}} E_i \quad , \; j = i+1, i+2, \ldots, n$$

This sequential step will eliminate x_i in each row below the ith one.

The resulting system is

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \quad E_1$$

$$a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \quad E_2$$

$$a_{33}x_3 + \cdots + a_{3n}x_n = b_3 \quad E_3$$

$$a_{nn}x_n = b_n \quad E_n$$

$$a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \quad E_2$$

$$a_{33}x_3 + \cdots + a_{3n}x_n = b_3 \quad E_3$$

$$a_{nn}x_n = b_n \quad E_n$$

This system is called upper triangular system.

Eq.(4) represent triangular system. This system can be solved by backward substitution.

(II) The upper triangular system can be solved by backward subsitution, that is,

Solving Equation E_n, we obtain

$a_{nn} x_n = b_n : E_n \Longrightarrow x_n = \frac{b_n}{a_{nn}}$

Subsitute with x_n in E_{n-1}, we obtain:

$$a_{n-1,n-1} x_{n-1} + a_{n-1,n} x_n = b_{n-1}$$

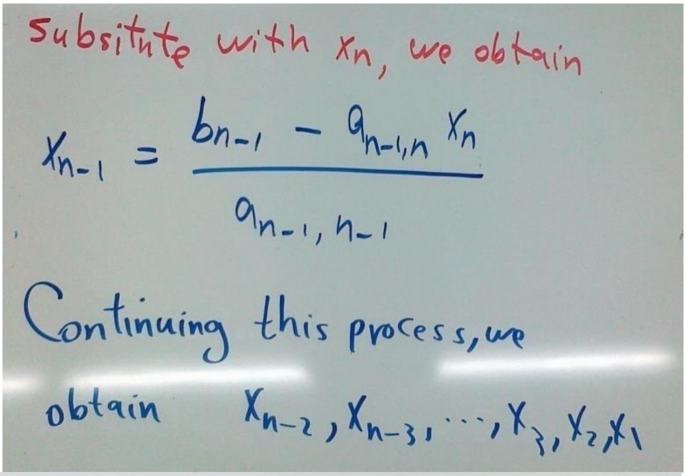
$$x_{n-1} = \frac{b_{n-1} - a_{n-1,n} x_n}{a_{n-1,n-1}}$$

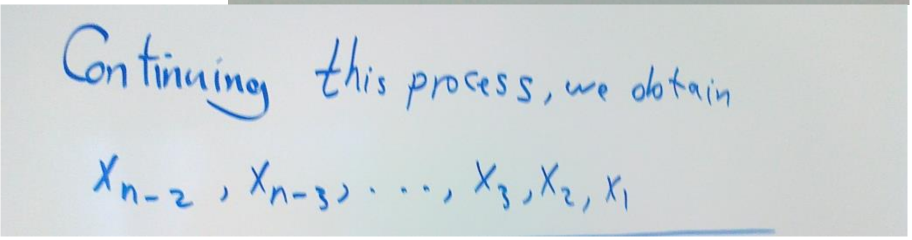
Solving Eq. (E_n) of Eq. (4)

$$x_n = \frac{b_n}{a_{nn}}$$

Now, Eq. E_{n-1} is

$$a_{n-1,n-1} x_{n-1} + a_{n-1,n} x_n = b_{n-1}$$





<u>Direct methods</u>□
Gaussian elimination method

Provided $a_{11} \neq 0$ perform $E_j = E_j - \frac{a_j}{a_{11}} E_1$, for j=2,3,…n. this step will eliminate the coefficient of x_1 in each of these rows . the resulting system is

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \qquad E_1$$
$$\bar{a}_{22}x_2 + \cdots + \bar{a}_{2n}x_n = \bar{b}_2 \qquad E_2$$
$$\bar{a}_{32}x_2 + \bar{a}_{33}x_3 + .. + \bar{a}_{3n}x_n = \bar{b}_3 \qquad E_3$$

.

.

.

$\bar{a}_{n2}x_2 + \bar{a}_{n3}x_3 + \ldots + \bar{a}_{nn}x_n = \bar{b}_n \qquad E_n$

We change the symbol of coef. from $a \quad to\ \bar{a}$ and $b \quad to\ \bar{b}$ Since we expect that entries in row 2,3,…n will be changed.

For ease of notation we again denote the entry in the $\underline{i}^{th}$row and $\underline{j}^{th}$column by a_{ij} .

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \quad E_1$$
$$\bar{a}_{22}x_2 + \cdots + \bar{a}_{2n}x_n = \bar{b}_2 \qquad E_2\square$$
$$\bar{a}_{32}x_2 + \bar{a}_{33}x_3 + .. + \bar{a}_{3n}x_n = \bar{b}_3 \qquad E_3\square$$
$$\bar{a}_{n2}x_2 + \bar{a}_{n3}x_3 + \ldots + \bar{a}_{nn}x_n = \bar{b}_n \qquad E_n$$

Provided that $a_{ii} \neq 0$ perform
$$E_j = E_j - \frac{a_j}{a_{ii}} E_j \quad ,j = i + 1 \ ,i + 2 \ldots n$$
This sequential stop will eliminate x_i in each row below the i^{th}one .
The resulting system is

$$a_{11}x_1 + a_{12}x_2 + \cdots \qquad + a_{1n}x_n = b_1 \quad E_1\square$$
$$a_{22}x_2 + \cdots \qquad + \bar{a}_{2n}x_n = b_2 \quad E_2$$
$$a_{33}x_3 + .. + a_{3n}x_n = b_3 \quad E_3$$
$$.$$
$$.$$
$$.$$
$$a_{nn}x_n = b_n \quad E_n$$

This system is called upper triangular system.

Eq.(4) represent triangular system . this system can be solved by backward substitution .

he upper triangular system can be solved by backward substitution , that is, solving Equation E_n, we obtain

$a_{nn}\, x_n = b_n$: $E_n \rightarrow \quad x_n = \frac{b_n}{a_{nn}}$□

substitution with $x_n$in E_{n-1}, we obtain :□

$a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n = b_{n-1}$□

$x_{n-1} = \dfrac{b_{n-1} - a_{n-1,n}x_n}{a_{n-1,n-1}}$□

solving Eq.(E_n)of Eq. (4)

$$x_n = \frac{b_n}{a_{nn}}$$

Now , Eq. $E_{n-1}$is

$$a_{n-1}x_{n-1} + a_{n-1,n}x_n = b_{n-1}$$

Substitute with x_n,we obtain

$$\boldsymbol{x_{n-1} = \frac{b_{n-1} - a_{n-1,n}\, x_n}{a_{n-1,n-1}}}$$□

Continuing this process , we obtain

$\boldsymbol{x_{n-2}, x_{n-3}, \ldots, x_3, x_2, x_1}$

Example : use Gaussian method to solve

$x_1 + 2x_2 + x_3 = 0 \quad : E_1$
$2x_1 + 2x_2 + 3x_3 = 3 \quad : E_2$
$-x_1 + 3x_2 = 2 \quad : E_3$

Answer

I]: Eliminating coefficients of some x_i to obtain upper triangular system :

(a) provided $a_{11} \neq 0$, perfrom $E_j = E_j - \frac{a_{j1}}{a_{11}} E_1$

j=2,3,…n

$a_{11} = 1, j = 2, a_{21} = 2$

$$E_2 = E_2 - \frac{a_{21}}{a_{11}} E_1 = E_2 - \frac{2}{1} E_1$$

The system become

$x_1 + 2x_2 + x_3 = 0 \quad : E_1$
$-2x_2 + x_3 = 3 \quad : E_2$
$-x_2 - x_3 = 2 \quad : E_3$

j=3 , $a_{31} = -1$

$$E_3 = E_3 - \frac{a_{31}}{a_{11}} E_1 = E_3 - \frac{(-1)}{1} E_1 = E_3 + E_1$$

$$E_3 \leftarrow E_3 + E_1$$

E_3: $\quad - x_1 - 3x_2 \quad = 2$
E_1: $\quad x_1 + 2x_2 + x_3 = 0$

New E_3: $\quad 0 - x_2 + x_3 = 2$

The system become

$$E_1: x_1 + 2x_2 + x_3 = 0$$
$$E_2: \quad -2x_2 + x_3 = 3$$
$$E_3: \quad -x_2 + x_3 = 2$$

up to now , we eliminate the coefficient of x_1 from E_2 andE_3. In the next step we shall eliminate x_2 from E_3 to complete the shape of upper triangular system:

(b) provided $a_{22} \neq 0$, perfrom $E_j = E_j - \frac{a_{j2}}{a_{22}} E_2$
, j=3,4,…n

$$\boldsymbol{a_{22}} = -2, j = 3, \boldsymbol{a_{32} = -1}$$

$$\boldsymbol{E_3} = \boldsymbol{E_3 - \frac{a_{32}}{a_{22}} E_2} = \boldsymbol{E_3 - \frac{(-1)}{(-2)} E_2} = \boldsymbol{E_3 - \frac{1}{2} E_2}$$

$$E_3 \leftarrow E_3 - \frac{1}{2} E_2$$

$$E_3: \quad -x_2 + x_3 = 2 = \frac{4}{2}$$

$$-\frac{1}{2} E_2: \quad -x_2 - \frac{1}{2} x_3 = \frac{-3}{2}$$

New E_3: $\quad 0 + \frac{1}{2} x_3 = \frac{1}{2}$

The system become

$$E_1: x_1 + 2x_2 + x_3 = 0$$
$$E_2: \quad -2x_2 + x_3 = 3$$
$$E_3: \quad \frac{1}{2}x_3 = \frac{1}{2}$$

This is the required upper triangular system

the upper triangular system can be solved by backward substitution :

$$E_3: \frac{1}{2}x_3 = \frac{1}{2} \rightarrow x_3 = 1$$
$$E_2: -2x_2 + x_3 = 3 \rightarrow -2x_2 + 1 = 3$$
$$\rightarrow -2x_2 = 2$$
$$\rightarrow x_2 = -1$$
$$E_1: x_1 + 2x_2 + x_3 = 0 \rightarrow x_1 + 2(-1) + 1 = 0$$
$$\rightarrow x_1 = 1$$

Hence the solution of the given system is

$$x_1 = 1, x_2 = -1, x_3 = 1$$

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Iterative methods

4 -2 الطرق التتابعية

إذا كانت المصفوفة A هشة أي تحتوي على عدد كبير من العناصر الصفرية فإنه يفضل استخدام الطرق التتابعية. و تعتمد هذه الطرق على البدء من تخمين ابتدائي للحل X_0 و التعويض في صيغة تتابعية تعتمد على نظام المعادلات موضع الحل للوصول إلى تتابعات من التقريبات $X_1, X_2, \ldots$ تتقارب نحو الحل الصحيح X. و سوف نبدأ بدراسة الطريقة اليعقوبية:

4 -2 -1 الطريقة اليعقوبية Jacobean method

لنأخذ في الاعتبار نظام المعادلات

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \qquad E_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \qquad E_2 \qquad (1) \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \qquad E_3 \end{aligned}$$

تعتمد الطريقة اليعقوبية Jacobean method على فكرة إيجاد x_k في المعادلة رقم E_k بدلالة باقي المجاهيل لتصبح المعادلات (1) كالتالي

$$\begin{aligned} x_1 &= \tfrac{1}{a_{11}}\left[b_1 - a_{12}x_2 - a_{13}x_3\right] \\ x_2 &= \tfrac{1}{a_{22}}\left[b_2 - a_{21}x_1 - a_{23}x_3\right] \qquad (2) \\ x_3 &= \tfrac{1}{a_{33}}\left[b_3 - a_{31}x_1 - a_{32}x_2\right] \end{aligned}$$

ثم باختيار $X^{(0)} = \left[x_1^{(0)}, x_2^{(0)}, x_3^{(0)}\right]^T$ كتقريب (تخمين) ابتدائي للحل.

ثم كتابة العلاقات التتابعية اعتمادا على المعادلات (2) كالتالي

$$
\begin{aligned}
x_1^{(k+1)} &= \tfrac{1}{a_{11}}\left[b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)}\right] \\
x_2^{(k+1)} &= \tfrac{1}{a_{22}}\left[b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)}\right] \qquad (3) \\
x_3^{(k+1)} &= \tfrac{1}{a_{33}}\left[b_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)}\right]
\end{aligned}
$$

حيث $k = 0,1,\ldots,3$.

هذه الطريقة تسمى الطريقة اليعقوبية Jacobean method أو طريقة الاستبدالات المتماثلة

مثال 1:

[Q3] Use Jacobean method to solve

استخدم الطريقة اليعقوبية لحل النظام التالي المعادلات الخطية

$$9x_1 + x_2 + x_3 = 10$$

$$2x_1 + 10x_2 + 3x_3 = 19$$

$$3x_1 + 4x_2 + 11x_3 = 0$$

Starting from

و ذلك لتتابعين متتالين مبتدءاً من التخمين الابتدائي $X^{(0)} = [0,0,0]^T$

Evaluate the error if you know that the exact is

و احسب الخطأ الأقصى في كل تتابع إذا علمت أن $X = [1,2,-1]^T$

الحل:

باستخدام المعادلات (3) ووضع $k = 0$ نجد أن

$$x_1^{(1)} = \frac{1}{a_{11}}\left[b_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)}\right]$$
$$x_2^{(1)} = \frac{1}{a_{22}}\left[b_2 - a_{21}x_1^{(0)} - a_{23}x_3^{(0)}\right]$$
$$x_3^{(1)} = \frac{1}{a_{33}}\left[b_3 - a_{31}x_1^{(0)} - a_{32}x_2^{(0)}\right]$$

ولكن من نظام المعادلات المعطى لدينا

$$a_{11} = 9,\ b_1 = 10,\ x_2^{(0)} = x_3^{(0)} = 0$$

$$9x_1 + x_2 + x_3 = 10$$
$$2x_1 + 10x_2 + 3x_3 = 19$$
$$3x_1 + 4x_2 + 11x_3 = 0$$

$$\boldsymbol{x_1 = \frac{1}{9}[10 - x_2 - x_3]}$$
$$\boldsymbol{x_2 = \frac{1}{10}[19 - 2x_1 - 3x_3]}$$
$$\boldsymbol{x_3 = \frac{1}{11}[0 - 3x_1 - 4x_2]}$$

وبالتالي يكون

$$x_1^{(1)} = \frac{1}{9}\left[10 - 0 - 0\right] = \frac{10}{9} = 1.111$$
$$x_2^{(1)} = \frac{1}{10}\left[19 - 0 - 0\right] = \frac{19}{10} = 1.9$$
$$x_3^{(1)} = \frac{1}{11}\left[0 - 0 - 0\right] = 0$$

و باستخدام الحل الصحيح المعطى

$$e_1 = x_1 - x_1^{(1)} = 1 - 1.111 = -0.111$$

$$e_2 = x_2 - x_2^{(1)} = 2 - 1.9 = 0.1$$

$$e_3 = x_3 - x_3^{(1)} = -1 - 0 = -1$$

و يكون الخطأ الأقصى

$$E^{(1)} = \max_{1 \le i \le 3} |e_i| = \max_{1 \le i \le 3} |0.111,\ 0.1,\ 1.0|$$

التتابع الثاني بالمثل يكون بوضع $k = 1$ نجد أن

$$\boldsymbol{x_1 = \frac{1}{9}[10 - x_2 - x_3]}$$

$$\boldsymbol{x_2 = \frac{1}{10}[19 - 2x_1 - 3x_3]}$$

$$\boldsymbol{x_3 = \frac{1}{11}[0 - 3x_1 - 4x_2]}$$

$$x_1^{(2)} = \frac{1}{a_{11}}\left[b_1 - a_{12}x_2^{(1)} - a_{13}x_3^{(1)}\right]$$

$$= \frac{1}{9}\left[10 - 1.9 - 0\right] = 0.9$$

$$x_2^{(2)} = \frac{1}{a_{22}}\left[b_2 - a_{21}x_1^{(1)} - a_{23}x_3^{(1)}\right]$$

$$= \frac{1}{10}\left[19 - 2*1.111 - 3*0\right]$$

$$= 1.6778$$

$$x_3^{(2)} = \frac{1}{a_{33}}\left[b_3 - a_{31}x_1^{(1)} - a_{32}x_2^{(1)}\right]$$

$$= \frac{1}{11}\left[0 - 3*1.111 - 4*1.9\right]$$

$$= -0.99$$

و باستخدام الحل الصحيح المعطى

$$e_1 = x_1 - x_1^{(1)} = 1 - 0.9 = 0.1$$

$$e_2 = x_2 - x_2^{(1)} = 2 - 1.6778 = 0.322$$

$$e_3 = x_3 - x_3^{(1)} = -1 + 0.99 = -0.01$$

و يكون الخطأ الأقصى

$$E^{(1)} = \max_{1 \le i \le 3} |e_i| = \max_{1 \le i \le 3} |0.1,\ 0.322,\ -0.01| = 0.322$$

وتكون النتائج كالتالي

k	x1	x2	x3
1.0000	1.1111	1.9000	0
2.0000	0.9000	1.6778	-0.9939
3.0000	1.0351	2.0182	-0.8556

4 -2 -2 طريقة جاوس – سيدال:

Gauss-Seidel method□

لنأخذ في الإعتبار نظام المعادلات

$$\left.\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \qquad E1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \qquad E2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \qquad E3 \end{aligned}\right) \qquad (1)$$

تستعين طريقة جاوس سيدل بنفس العلاقات التتابعية للطريقة اليعقوبية والتي هي

على الصورة

$$\left.\begin{aligned} x_1^{(k+1)} &= \frac{1}{a_{11}}\left[b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)}\right] \\ x_2^{(k+1)} &= \frac{1}{a_{22}}\left[b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)}\right] \\ x_3^{(k+1)} &= \frac{1}{a_{33}}\left[b_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)}\right] \end{aligned}\right) \qquad (2)$$

وتعتمد الفكرة الجديدة على استخدام كل المعلومات التي حصلنا عليها قبل حساب كل مركبة. فمثلاً؛ عند حساب $x_2^{(k+1)}$ نجد أن لدينا تقريباً جديداً لـ x_1 هو $x_1^{(k+1)}$ قد تم حسابه في الخطوة السابقة مباشرة وهو تقريب أحدث من $x_1^{(k)}$. ونفس الشيء عند حساب $x_3^{(k+1)}$ لدينا تقريبان حديثان $x_1^{(k+1)},x_2^{(k+1)}$ يمكن استخدامهما لتصبح العلاقات التتابعية الجديدة:

$$\left.\begin{aligned} x_1^{(k+1)} &= \frac{1}{a_{11}}\left[b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)}\right] \\ x_2^{(k+1)} &= \frac{1}{a_{22}}\left[b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)}\right] \\ x_3^{(k+1)} &= \frac{1}{a_{33}}\left[b_3 - a_{31}x_1^{(k+1)} - a_{32}x_2^{(k+1)}\right] \end{aligned}\right) \qquad (3)$$

و بشكل عام فإن الصيغة التتابعية لطريقة جاوس – سيدال هي

$$x_i^{(m+1)} = \frac{1}{a_{ii}}\left\{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(m+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(m)}\right\} \qquad (9)$$

لكل $i=1,2,....,n$ و n عدد المعادلات.

مثال(2):

استخدم طريقة جاوس -سيدل لتتابعين متتاليين لحل نظام المعادلات

[Q4] Use Gauss-Seidel method to solve □

$$9x_1+x_2+x_3=10,\ \ 2x_1+10x_2+3x_3=19,\ \ 3x_1+4x_2+11x_3=0$$

Starting from $X^{(0)}=[0,0,0]^T$. Do 2 iterations. Evaluate the error if you know that the exact is $X=[1,2,-1]^T$

احسب الخطأ الأقصى في كل تتابع إذا علمت أن الحل الصحيح هو

$X=[1,2,-1]^T$

الحل:

باستخدام المعادلات (3) وبوضع $k=0$ **نجد أن**

$$x_1^{(1)}=\frac{1}{a_{11}}\left[b_1-a_{12}x_2^{(0)}-a_{13}x_3^{(0)}\right]$$

حيث أنه من نظام المعادلات لدينا

$$a_{11}=9,\ b_1=10,\ x_2^{(0)}=x_3^{(0)}=0$$

إذن

$$x_1^{(1)}=\frac{1}{9}[10-0-0]=\frac{10}{9}=1.111$$

وكذلك فإن

$$x_2^{(1)} = \frac{1}{a_{22}}\left[b_2 - a_{21}x_1^{(1)} - a_{23}x_3^{(0)}\right]$$

$$x_2^{(1)} = \frac{1}{10}\left[19 - 2(1.11) - 3(-1)\right] = 1.6778$$

وكذلك

$$x_3^{(1)} = \frac{1}{a_{33}}\left[b_3 - a_{31}x_1^{(1)} - a_{32}x_2^{(1)}\right]$$

$$a_{33} = 11,\ b_3 = 0,\ a_{31} = 3,\ a_{32} = 4$$

$$x_3^{(1)} = \frac{1}{11}\left[0 - 3(1.111) - 4(1.6778)\right] = -0.9131$$

ولحساب الخطأ الأقصى لدينا

$$e_1 = x_1 - x_1^{(1)} = 1 - 1.111 = -0.111$$

$$e_2 = x_2 - x_2^{(1)} = 2 - 1.6778 = 0.322$$

$$e_3 = x_3 - x_3^{(1)} = -1 + 0.9131 = -0.0869$$

ويكون الخطأ الأقصى

$$E^{(1)} = \max_{1 \le i \le 3} |e_i| = 0.322$$

التتابع الثاني $k = 1$ فتصبح

$$x_1^{(2)} = \frac{1}{a_{11}}\left[b_1 - a_{12}x_2^{(1)} - a_{13}x_3^{(1)}\right]$$
$$x_2^{(2)} = \frac{1}{a_{22}}\left[b_2 - a_{21}x_1^{(2)} - a_{23}x_3^{(1)}\right]$$
$$x_3^{(2)} = \frac{1}{a_{33}}\left[b_3 - a_{31}x_1^{(2)} - a_{32}x_2^{(2)}\right]$$

إذن

$$x_1^{(2)} = \frac{1}{9}\left[10 - 1(1.778) - 1(-0.9131)\right] = 1.0262$$

$$x_2^{(2)} = \frac{1}{10}\left[19 - 2(1.0262) - 3(-0.9131)\right] = 1.9687$$

$$x_3^{(2)} = \frac{1}{11}\left[0 - 3(1.0262) - 4(1.9687)\right] = -0.9958$$

ولحساب الخطأ الأقصى يكون

$$e_1 = x_1 - x_1^{(2)} = 1 - 1.0262 = 0.0738$$
$$e_2 = x_2 - x_2^{(2)} = 2 - 1.9687 = 0.0313$$
$$e_3 = x_3 - x_3^{(2)} = -1 + 0.9958 = -0.0042$$

ويكون الخطأ الأقصى

$$E^{(2)} = \max_{i \le i \le 3} |e_i| = 0.0313$$

وباستمرار تتابعات الحل في المثال السابق نحصل على الجدول التالي

وتكون النتائج كالتالي

```
kmax =

   10

    k          x1         x2        x3

   1.0000    1.1111    1.6778   -0.9131

   2.0000    1.0262    1.9687   -0.9958

   3.0000    1.0030    1.9981   -1.0001

   4.0000    1.0002    2.0000   -1.0001

   5.0000    1.0000    2.0000   -1.0000

   6.0000    1.0000    2.0000   -1.0000

   7.0000    1.0000    2.0000   -1.0000
```

4 -2 -3 دراسة الخطأ في الطريقة اليعقوبية التكرارية:

[Q5] Consider the system $AX = B$, of order n such that $a_{ii} \neq 0$ for $i = 1,2,\ldots,n$. Study the error of Jacobean method for solving this system. Then obtain number of iterations k required to obtain accuracy ε.

إذا كان $AX = b$ نظام خطي من رتبة n بحيث $a_{ii} \neq 0$ لكل $i = 1,2,\ldots\ldots,n$. فإنه يمكن كتابة هذا النظام على الصورة

$$
\begin{aligned}
x_1 &= \tfrac{1}{a_{11}}\left[b_1 - a_{12}x_2 - a_{13}x_3\right] \\
x_2 &= \tfrac{1}{a_{22}}\left[b_2 - a_{21}x_1 - a_{23}x_3\right] \\
x_3 &= \tfrac{1}{a_{33}}\left[b_3 - a_{31}x_1 - a_{32}x_2\right]
\end{aligned} \qquad (1)
$$

أو اختصارا

$$
x_i = \frac{1}{a_{ii}}\left\{ b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij}\, x_j \right\} \quad ; i = 1, 2, \ldots\ldots, n \qquad (2)
$$

بفرض أن $x_i^{(0)}$ لكل $i = 1,2,\ldots\ldots,n$ تقريب أولى للحل , فإن عدد من التقريبات

الأفضل يمكن الحصول عليه من الصيغة التكرارية

$$
\begin{aligned}
x_1^{(k+1)} &= \tfrac{1}{a_{11}}\left[b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)}\right] \\
x_2^{(k+1)} &= \tfrac{1}{a_{22}}\left[b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)}\right] \\
x_3^{(k+1)} &= \tfrac{1}{a_{33}}\left[b_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)}\right]
\end{aligned} \qquad (3)
$$

أو اختصارا

$$
x_i^{(k+1)} = \frac{1}{a_{ii}}\left\{ b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij}\, x_j^{(k)} \right\} \qquad (4)
$$

لكل $k \geq 0$, $i = 1, 2, \ldots\ldots, n$

الآن لدراسة الخطأ في استخدام طريقة جاكوبى التكرارية . اجعل $X^{(m)}$ هو التقريب الناتج في الخطوة m بالصيغة (4) للحل التام X للنظام (1) وكذلك فإن $e^{(m)} = X - X^{(m)}$ لكل $m \geq 0$ تعطى الخطأ في هذا التقريب. وباستخدام (4) فإن

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left\{b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij}\, x_j^{(k)}\right\}$$

$$x_i^{(m+1)} = \frac{b_i}{a_{ii}} - \frac{1}{a_{ii}} \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij}\, x_j^{(m)} \qquad = \frac{b_i}{a_{ii}} - \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{a_{ij}}{a_{ii}} x_j^{(m)}$$

Replace *m* by *m-1*

$$x_i^{(m)} = \frac{b_i}{a_{ii}} - \frac{1}{a_{ii}} \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij}\, x_j^{(m-1)} \qquad = \frac{b_i}{a_{ii}} - \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{a_{ij}}{a_{ii}} x_j^{(m-1)}$$

By subtraction

$$x_i^{(m+1)} - x_i^{(m)} = -\sum_{\substack{j=1 \\ j \neq i}} \frac{a_{ij}}{a_{ii}} \left(x_j^{(m)} - x_j^{(m-1)} \right) \; ; \; i = 1, 2, ..., n \quad ; \; m \geq 1$$

(5)

So

$$\left| x_i^{(m+1)} - x_i^{(m)} \right| \le \left(\sum_{\substack{j=1 \\ j\ne i}} \left| \frac{a_{ij}}{a_{ii}} \right| \right) \max_{1\le j\le n} \left| x_j^{(m)} - x_j^{(m-1)} \right|$$

$$= \sum_{\substack{j=1 \\ j\ne i}} \left| \frac{a_{ij}}{a_{ii}} \right| \left\| x^{(m)} - x^{(m-1)} \right\|_\infty$$

Where $\left\| x^{(m)} - x^{(m-1)} \right\|_\infty = \max_{1\le j\le n} \left| x_j^{(m)} - x_j^{(m-1)} \right|$ (6)

Now, If we Put: $\lambda := \max_{1\le i\le n} \sum_{\substack{j=1 \\ j\ne i}} \left| \frac{a_{ij}}{a_{ii}} \right|$, we have

$$\left| x_i^{(m+1)} - x_i^{(m)} \right| \le \lambda \left\| x^{(m)} - x^{(m-1)} \right\|_\infty \qquad \forall \ i = 1,2,.....,n$$

وحيث أن الطرف الأيمن لا يعتمد على i فإن

$$\max_{1\le i\le n} \left| x_i^{(m+1)} - x_i^{(m)} \right| \le \lambda \left\| x^{(m)} - x^{(m-1)} \right\|_\infty$$

Using (6),

$$\left\| x_i^{(m+1)} - x_i^{(m)} \right\|_\infty \le \lambda \left\| x^{(m)} - x^{(m-1)} \right\|_\infty$$

الشرط $\lambda < 1$ يكافئ أن تحقق معاملات النظام (1) الشرط

$$\sum_{\substack{j=1 \\ j\ne 1}}^{n} \left| a_{ij} \right| < a_{ii} \qquad ; \ i = 1,2,.....,n \tag{7}$$

المصفوفة $A = \left(a_{ij}\right)$ التي يتحقق لها الشرط (7) تسمى مهيمنة القطر .

وإذا $X^{(m)}$, $X^{(m+1)}$ تقريبيان للحل x فإن

$$\left\|X^{(m+1)} - X^{(m)}\right\|_{\infty} \le \lambda \left\|X^{(m)} - X^{(m-1)}\right\|_{\infty}$$

$$\le \lambda\lambda \left\|X^{(m-1)} - X^{(m-2)}\right\|_{\infty}$$

$$\cdots$$

$$\le \lambda^{m} \left\|X^{(1)} - X^{(0)}\right\|_{\infty} \qquad (8)$$

من هذه المعادلة إذا $e^{m} = X^{(m)} - X^{(m-1)}$ هو الخطأ عن $1 \le m$ وإذا $1 > \lambda$ فإن

$\left\|e^{m}\right\|_{\infty} \to 0$ عندما $m \to \infty$

=================

الآن لحساب عدد التكرارات k التي تلزم للحصول على تقريب بدقة ε للحل

التام X . اجعل $X^{(m)}, X^{(k)}$ تقريبان بحيث $m > k$ فنجد

$$\left\|X^{(m)} - X^{(k)}\right\|_{\infty} \le \left\|X^{(m)} - X^{(m-1)}\right\|_{\infty} + \ldots..$$

$$+ \left\|X^{(k+1)} - X^{(k)}\right\|_{\infty}$$

وباستخدام (8) نحصل على

$$\left\| X^{(m)} - X^{(k)} \right\|_\infty \le \lambda^{m-1} \left\| X^{(1)} - X^{(0)} \right\|_\infty + \ldots\ldots + \lambda^k \left\| X^{(1)} - X^{(0)} \right\|_\infty$$

$$= \left(\lambda^{m-1} + \ldots\ldots + \lambda^k \right) \left\| X^{(1)} - X^{(0)} \right\|_\infty$$

$$= \left(\lambda^{m-1} + \ldots\ldots + \lambda^k \right) \left\| X^{(1)} - X^{(0)} \right\|_\infty$$

وباستخدام مجموع المتسلسلة الهندسية

$$\left\| X^{(m)} - X^{(k)} \right\|_\infty \le \frac{\lambda^k}{1-\lambda} \left\| X^{(1)} - X^{(0)} \right\|_\infty$$

وحيث أن $lim \quad X^{(m)} = X$ فإن

$$\left\| X - X^{(k)} \right\|_\infty \le \frac{\lambda^k}{1-\lambda} \left\| X^{(1)} - X^{(0)} \right\|_\infty \quad ; k \ge 0 \qquad (7)$$

وعليه فإنه للحصول على تقريب $X^{(k)}$ بدقة ε للحل التام X فإنه يلزم عمل التكرار عدد k مرة بحيث

$$\frac{\lambda^k}{1-\lambda} \left\| X^{(1)} - X^{(0)} \right\|_\infty \le \varepsilon$$

أي أن

$$k \geq \left\{ \log \left\| X^{(1)} - X^{(0)} \right\|_{\infty} - \log \varepsilon - \log(1-\lambda) \right\} \Big/ \operatorname{Log}(1/\lambda) \quad (8)$$

^^^

مثال(3):

للنظام الخطى

$$\begin{aligned} 2x_1 - x_2 \qquad &= 1 \\ -2x_1 + 5x_2 + 2x_3 &= 1 \\ 4x_1 \qquad + 6x_3 &= 8 \end{aligned}$$

احسب عدد التكرارات اللازمة للحصول على حل تقريبي بدقة 10^{-3} بفرض أن

$$X^{(0)} = (1, 0.2, 2/3)^T$$

الحل : - لهذا النظام فإن

$$\lambda := \underset{1 \leq i \leq 3}{Max} \sum_{j=1}^{3} \left| \frac{a_{ij}}{a_{ii}} \right|$$

i=1: (a12/a11+a13/a11)=(1/2+0)=1/2 i=2: (a21/a22 +a23/a22)=(2/5+2/5)=4/5 i=3: (a31/a33+a32/a33)=(4/6+0/6)=4/6

$$\lambda := Max \left\{ \frac{1}{2}, \frac{4}{5}, \frac{4}{6} \right\} = \frac{4}{5}$$

$$x^{(1)} = \begin{bmatrix} 1/2 \\ 1/5 \\ 8/6 \end{bmatrix} + \begin{bmatrix} 0 & 1/2 & 0 \\ 2/5 & 0 & -2/5 \\ -4/6 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0.2 \\ 2/3 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.33333 \\ 0.66667 \end{bmatrix}$$

$$\therefore \left\| x^{(1)} - x^{(0)} \right\|_{\infty} = \underset{1 \le i \le 3}{Max} \left| x_i^{(1)} - x_i^{(0)} \right|$$

$$= Max \left(0.4 \ , \ 0.13333 \ , \ 0.3334 \times 10^{-2} \right)$$

$$= 0.4$$

وإذا عدد التكرارات المطلوبة هو k فإن

$$\frac{(4/5)^k}{1 - 4/5} * 0.4 \le 10^{-3} \quad \Rightarrow \quad (0.8)^k \le 5 * 10^{-4}$$

ومنها $k \ge 34.06$

$\therefore$ عدد التكرارات المطلوبة هو $k = 35$.

Chapter 3

Solution of nonlinear equations

Principle of Root Finding

We shall study in this chapter numerical method that can solve such problems

The general form of non linear equation is

$$f(x) = 0, \qquad (1)$$

with f a given nonlinear function.

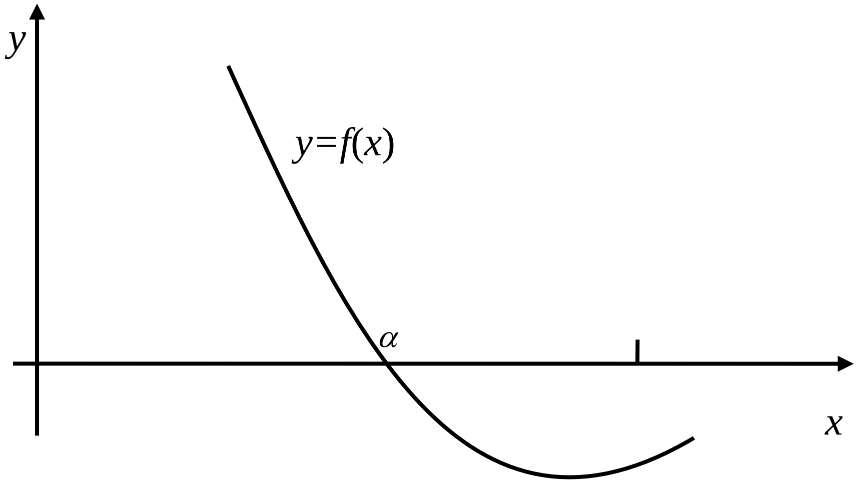
As examples, the nonlinear equations

$$x^6 - 3x^4 - 5x^3 + x + 1 = 0, \qquad xe^x - \cos x + 1 = 0$$

Have no analytical solution.

Here, we denote such **roots** or **zeroes** by the Greek letter α. Root finding problems occur in many contexts. Sometimes they are a direct formulation of some physical situation; but more often, they are an intermediate step in solving a much larger problem.

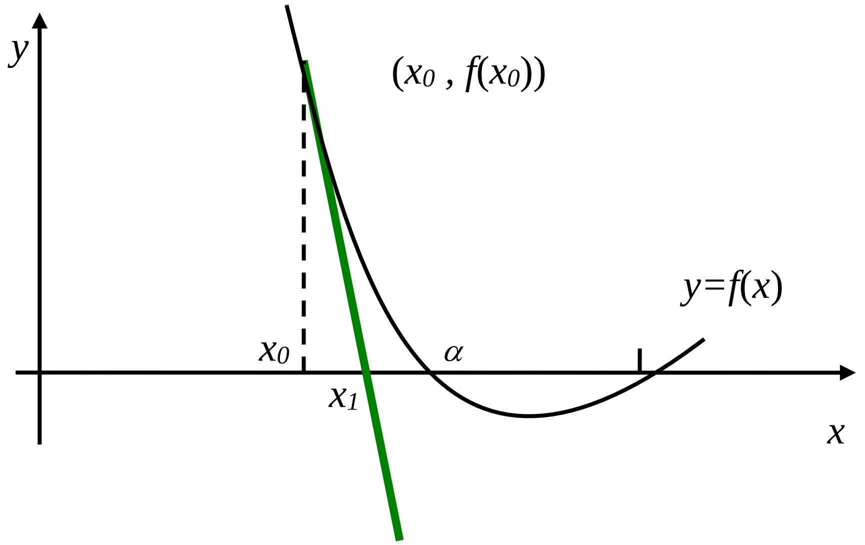
Graphically, any function $y = f(x)$ satisfies (1) must pass x axis



Finding the root α of a given function $f(x)$ corresponds to obtaining the point x at which the graph of $y = f(x)$ intersects the x-axis. One of the principles of numerical analysis is the following.

Since we cannot solve the given problem, then solve a "***nearby problem***".

The *nearby problem* is to find where a straight line intersects the x-axis. Thus we seek to replace $f(x) = 0$ by that of solving $p(x) = 0$ for some linear polynomial $p(x)$ that approximates $f(x)$ in the vicinity of the root α.



Given an estimate of α, say $\alpha \approx x_0$, approximate $f(x)$ by its linear Taylor polynomial at $(x_0,\ f(x_0))$:

$$p(x) = f(x_0) + (x - x_0) f'(x_0)$$

If x_0 is very close to α, then the root of $p(x)$ should be close to α. Denote this approximating root by x_1; repeat the process to further improve our estimate of α.

Denoting the root of $p(x) = 0$ by x_1, we solve for x_1 in

$$f(x_0) + (x - x_0) f'(x_0) = 0$$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

The general Newton's method for solving $f(x) = 0$ is derived exactly as above. The result is a sequence of numbers x_0, x_1 x_2…., defined by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},\ \mathrm{n} = 0, 1, 2, \ldots$$

Again, we want to know whether these numbers converge to the desired root α; and we would also like to know something about the speed of convergence (which says something about how many such iterates must actually be computed).

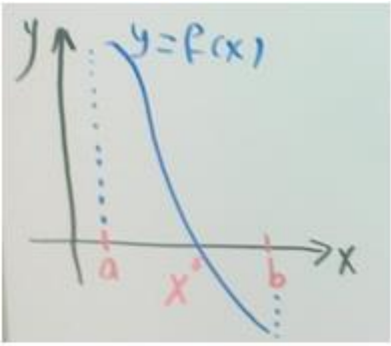
1- Closed Domain Method

Two of the simplest methods for finding the roots of a nonlinear equation are:
1. Interval halving (bisection method)
2. False position method

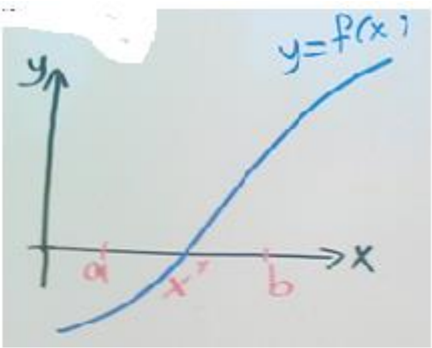
We start solving (1) , If we have a domain x=a and x=b that contains the root (solution) that is α $[a, b]$

The Bisection Method

Consider $y=f(x)$ is given
Let $x=a$ and $x=b$ satisfy $f(a)f(b) < 0$.
Then the interval $[a, b]$contains the root.
The main idea of this method is interval halving



(A) F(a)+, F(b)-



(B) F(a)-, F(b)+

Assume that $f(x)$ is continuous on a given interval $[a, b]$. We conclude the interval (a,b) must satisfies that the sign of $f(x)$ must be changed between a, b.
The condition

$$f(a)f(b) < 0 \qquad (2)$$

ensure that we have a root in $[a, b]$.

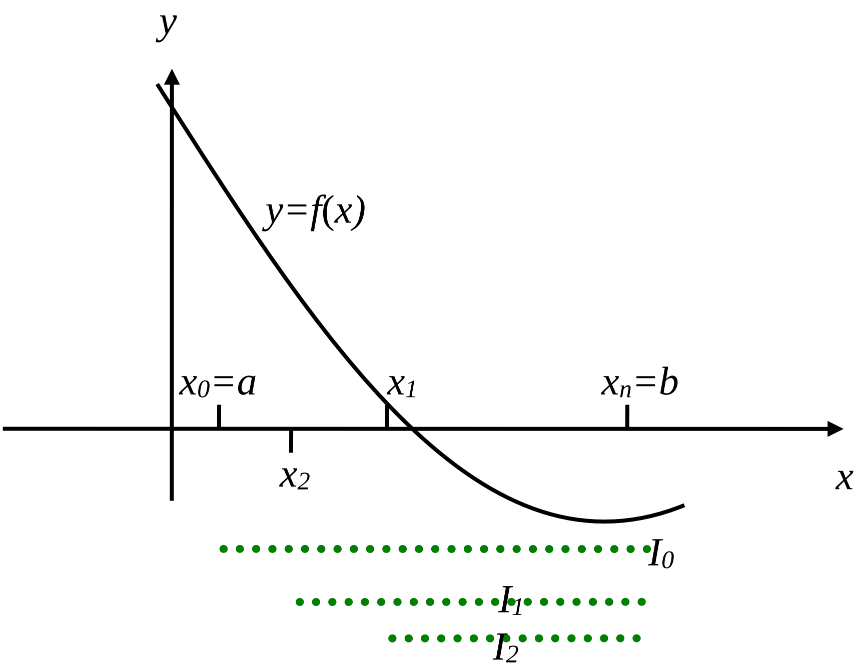


Fig. 3.1 Bisection method.

Using the intermediat value Theorem, the function $f(x)$ must have at least one root in $[a, b]$. Usually [a, b] is chosen to contain only one root α, but the following algorithm for the bisection method will always converge to some root α in $[a, b]$.

Bisection Algorithm

Input: function $f(x)$, An interval $[a, b]$, small number ε

$f(x) = x^3 - 2x - 5$, *a=0, b=3 and* ε*=0.00001*

Output: a value c satisfies $|f(c)| \leq \varepsilon$ or $|b - c| \leq \varepsilon$

1. Define *take c= (a+* b)/2, *evaluate fc=f(c)*
2. If $|f(c)| \leq \varepsilon$ or $|b - c| \leq \varepsilon$ then accept root = *c*, and exit.
3. If $f(c)\, f(b) < 0$, then a= c; otherwise, b =c.
4. Return to step 1.

The interval $[a, b]$ is halved in size for every pass through the algorithm. Because of step 3, $[a, b]$ will always contain a root of f(x). Since a root a is in $[a, b]$, it must lie within either $[a, c]$or $[c, b]$; and consequently

$$|c - \alpha| \leq b - c \,, \qquad |c - \alpha| \leq c - a$$

This is justification for the test in step 2. On completion of the algorithm, c will be an approximation to the root with

$$|c - \alpha| \leq \varepsilon$$

Example

Use bisection method to find the root α of

$$f(x) = x^3 - 2x - 5$$

in[0,3]. to within 0.00001

Answer:

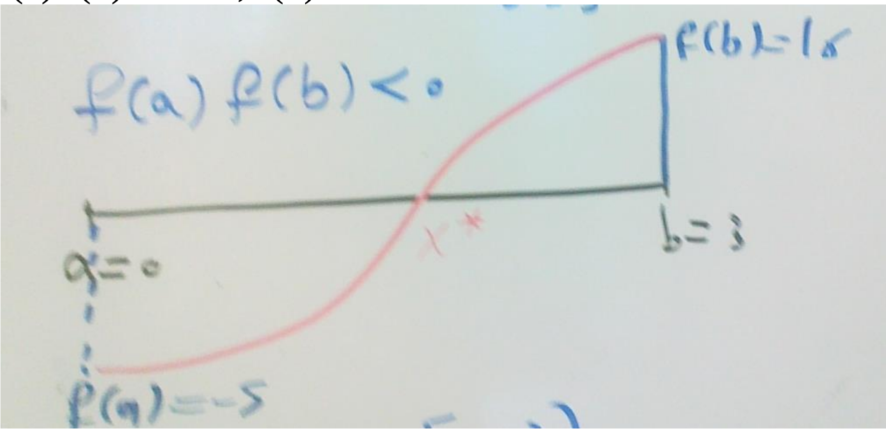
a=0 b=3

f(a) = f(o) =$o^3 - 2(o) - 5 = -5$

$f(b) = f(3) = 3^3 - 2(3) - 5 = 16$

=27 – 6 - 5=*16*

f(a) f(b) < 0 , f(b)=16



α=0 , b=3

f(a)=-5

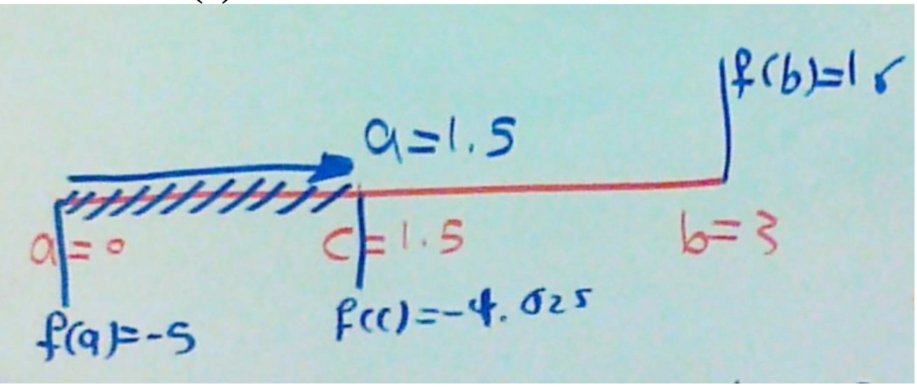
So x*∈ [0,3]

c=$\frac{0+3}{2} = 1.5$

f(c)=$(1.5)^3$-2(1.5)-5=-4.62

=========>

a=0 $\Rightarrow$ a=1.5 , f(b)=16
f(a)=-5, c=1.5 b=3
f(c)=-4.625



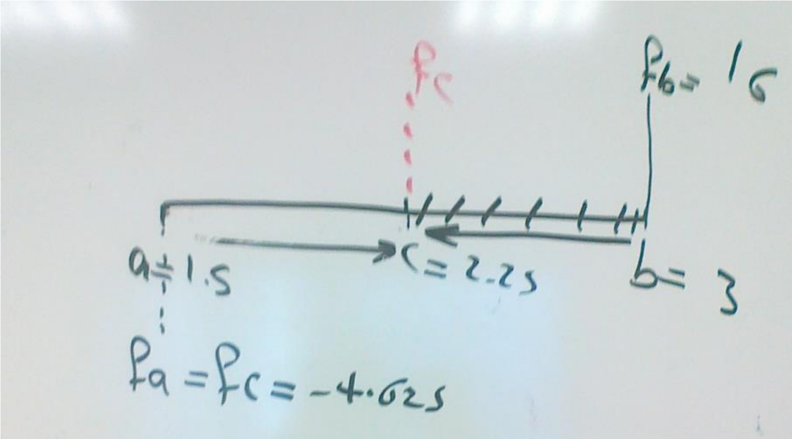
|fc|=4.625 is not $\leq \varepsilon$,
|b-c|=|3-1.5|=1.5 is not $\leq \varepsilon$
we must do one more iteration starting from step 1 .

$c = \frac{a+b}{2} = \frac{1.5+3}{2} = 2.25$

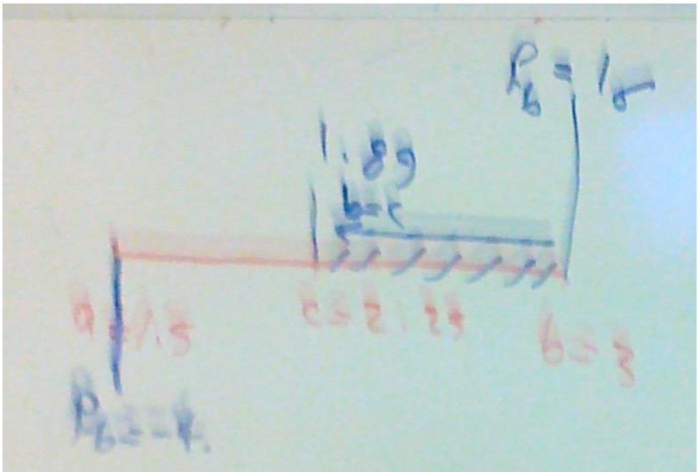
fc=f(c) =f(1.5)=$(1.5)^3 - 2(1.5) - 5 = 1.89$

Since fa fc < 0,then b=c =2.25

fb=fc=1.89



fc , fb=16
a=1.5 , c=2.25 , b=3
fa=fc=-4.625



|fc|=1.89 isnot $<\varepsilon$
|b-c|=|3=2.25| isnot $<\varepsilon$

We must do more one iteration starting from step2

Programming home work

Write program for bisection method to find the root of $f(x)=0$

In the interval $[a, b]$ with accuracy c. Check the program with the example

$f(x) = x^3 - 2x - 5$,   [0,3]   c=0.00001

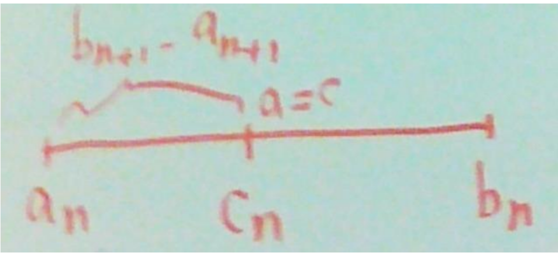
Matlab program will be as follows

Clear
f =@(x) x^3-2*x-5;
%---------------------
a = 0; fa = f(a);
b = 3; fb = f(b);
eps=0.00001;
x=a;fx=fa;
w='x fx'
 	
while abs(fx)>eps
x = (a+b)/2;
fx=f(x);
w=[x fx]
if sign(fx) == sign(fa)
a = x; fa = fx;
else
b = x; fb = fx;
end
end
x
fx
	w=
	x fx
	1.5 -4.6250
	2.25 1.8906
	1.8750 -2.1582
	
	….. …….
	
	2.0946 0.000
	
	x
	=2.0946
	
	fx
	= -8.8818e-016

Convergence of bisection method

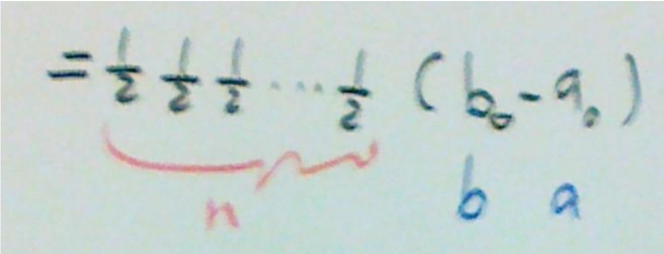
Let a_n, b_n ,c_n defines the values of a ,b , c at iteration no. n.
then we can write

$$b_{n+1}-a_{n+1}=\frac{1}{2}(b_n-a_n) \qquad ,n\geq 1$$



so

$$=\frac{1}{2}\left[\frac{1}{2}(b_{n-1}-a_{n-1})\right]$$



Continuing this, until we reach the initial data, we obtain

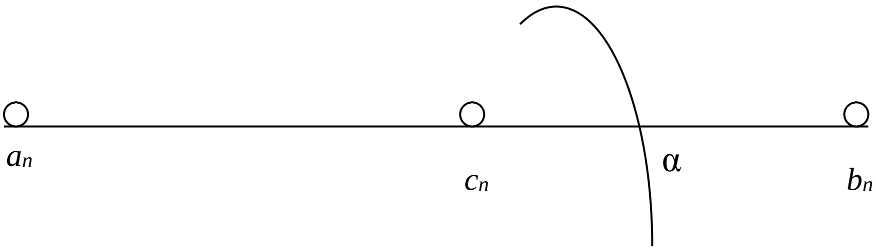
$$=\frac{1}{2^n}(b-a) \qquad ; n\geq 1$$

so

$$b_n-a_n=\frac{1}{2^{n-1}}(b-a) \qquad (*)$$

The root lies at [a, c] or [c, b]

وحيث أن الجذر α يقع إما في منتصف النطاق الحالي $[a_n, c_n]$ أو $[c_n, b_n]$ فيمكننا كتابة



let α be the exact root

$$|\alpha - c| \le \frac{1}{2}(b_n - a_n)$$

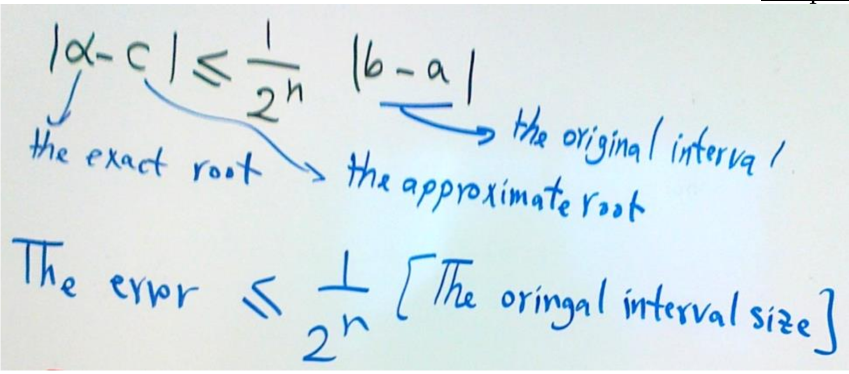
Using (*)

$$\le \frac{1}{2}\,\frac{1}{2^{n-1}}(b-a)$$

$$= \frac{1}{2^n}(b-a)$$

إذن

$$|\alpha - c_n| \le \frac{1}{2^n}(b-a)$$



This means that c_n converges as to α as $n \to \infty$.

==

To evaluate number of iteration required to obtain a certain accuracy ε.

$$|\alpha - c_n| \le \in \Rightarrow \frac{1}{2^{n-1}}(b-a) \le \varepsilon$$

$$2^n \ge \frac{b-a}{\varepsilon} \Rightarrow n \log 2 \ge \log \frac{(b-a)}{\varepsilon}$$

$$n \ge \frac{\log\left[\frac{(b-a)}{\varepsilon}\right]}{\log 2}$$

Example

Find number n of iterations required to obtain an error to within 10^{-3}

If we use bisection method to find the root of $f(x) = x^6 - x - 1$ on [0,2].

Answer

The formula for the error in bisection method is

$$|\alpha - c_n| \leq \frac{1}{2^n}(b-a)$$

Since we have *b*=2 *a*=0

$$|\text{Error}| \leq \frac{1}{2^n}[2-0] = \frac{2}{2^n} = \frac{1}{2^{n-1}} \leq 0.001$$

$1/2^{n-1} \leq 0.001$

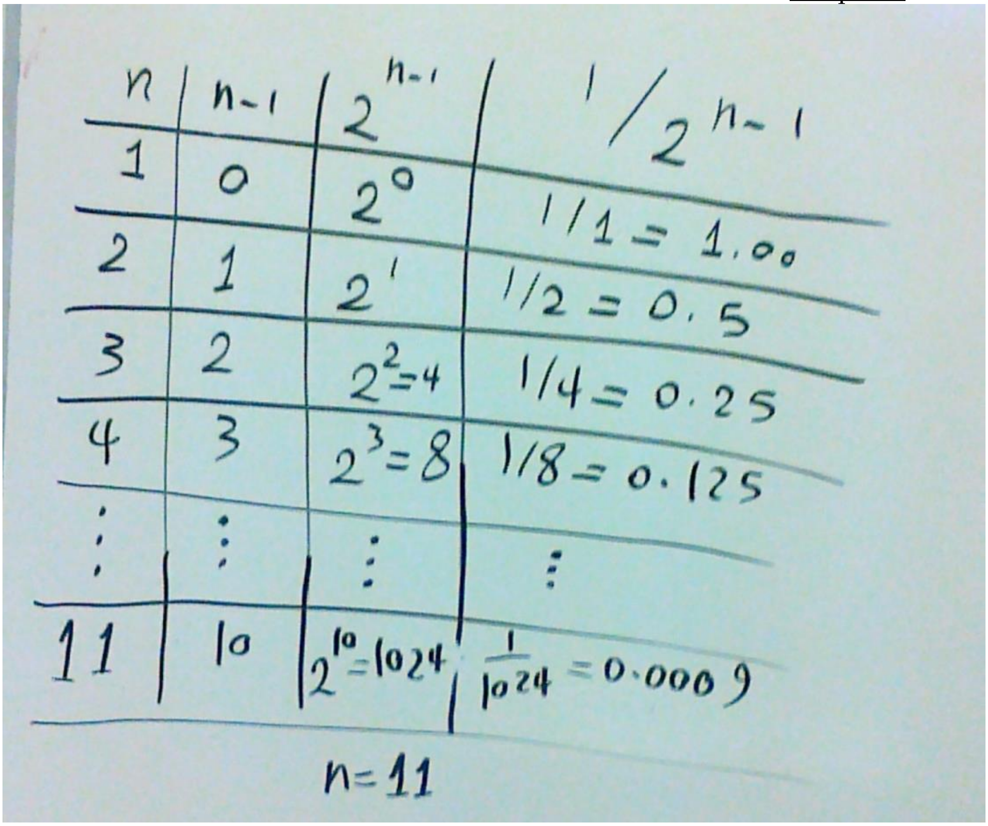
$1/2^{n-1} \leq 1/1000$

$2^{n-1} \leq 1000$

We know that $2^{10} = 1024$

n-1 = 10

n=11



n	$n-1$	2^{n-1}	$1/2^{n-1}$
1	0	2^0	$1/1 = 1.00$
2	1	2^1	$1/2 = 0.5$
3	2	$2^2 = 4$	$1/4 = 0.25$
4	3	$2^3 = 8$	$1/8 = 0.125$
⋮	⋮	⋮	⋮
11	10	$2^{10} = 1024$	$\frac{1}{1024} = 0.0009$

$n = 11$

False Position Method

The bisection method brackets a root in the interval $[a, b]$approximates the root as the midpoint of the interval. In the *false position,* the nonlinear function $f(x)$ is assumed to be a linear function $g(x)$ in the interval $[a, b]$, and the root of the linear function $g(x)$, $x = c$, is taken as the next approximation of the root of the nonlinear function $f(x)$.

The process is illustrated graphically in Figure 2. This method is also called the *linear interpolation method*. The root of the linear function $g(x)$, that is, $x = c$, is not the root of the nonlinear function $f(x)$. It is a false position, which gives the method its name. We now have two intervals, (a, c) and (c, b). As in the interval-halving (bisection) method, the interval containing the root of the nonlinear function $f(x)$ is retained. so the root remains bracketed.
The equation of the linear function $g(x)$ is

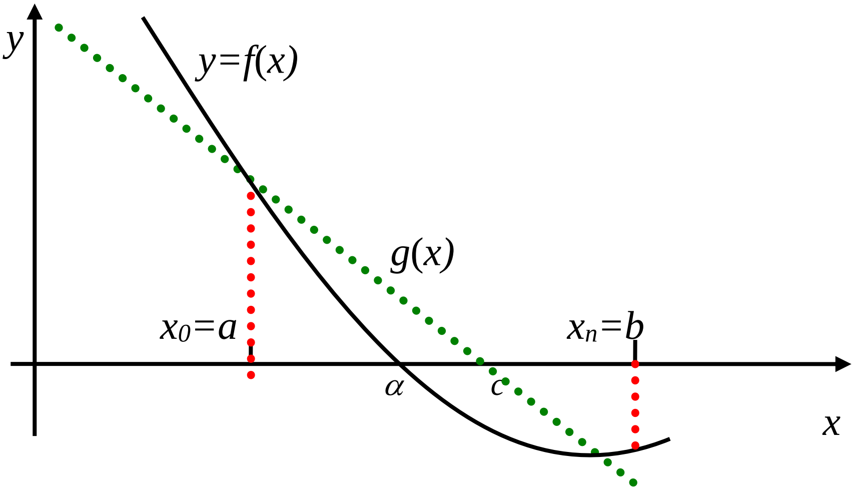
$$\frac{f(c)-f(b)}{c-b} = g'(x) \tag{1}$$

where $f(c) = 0$, and the slope of the linear function $g'(x)$ is given

$$g'(x) = \frac{f(b)-f(a)}{b-a} \tag{2}$$

Solving Eq. (1) for the value of c which gives $f(c) = 0$ yields

$$c = b - \frac{f(b)}{g'(x)} \tag{3}$$



Note that $f(a)$ and a could have been used in Eqs. (1) and (2) instead of $f(b)$.

Equation (3) is applied repetitively until either one or both of the following two convergence criteria are satisfied:

$$(i)|b-a| \leq \epsilon, \qquad (ii)|f(c)| \leq \epsilon$$

Example 1

Use False Position Method to find the root of

$f(x) = x^3 - 2x - 5$

In the interval $[0,3]$

Answer: the iterations are:

a=0	b=3

fa =-5 fb = 6 x =0.7143 fx = -6.0641
a=0.7143 b=3 x =1.3425 fx = -5.2654 a =1.3425 b =3
x = 1.7529 fx = -3.1197 a =1.7529 b = 3
x =1.9564 fx = -1.4248 a =1.9564 b =3

clear
f =@(x) x^3-2*x-5;
%----------------------
a = 0; fa = f(a)
b = 3; fb = f(b)
eps=0.1;
%----------------------
for i=1:10
gdx=(fb-fa)/(b-a);
x = b-fb/gdx;
fx=f(x);
 	
if sign(fx) == sign(fa)
a = x; fa = fx;
	ans =
	x f
	
	w =
	0.7143 -6.0641
	1.3425 -5.2654
	1.7529 -3.1197
	1.9564 -1.4248
	2.0417 -0.5723
	2.0748 -0.2179
	2.0872 -0.0813
	2.0919 -0.0301
	2.0936 -0.0111
	2.0942 -0.0041

```
else
b = x; fb = fx;
end
xx(i)=x
ff(i)=fx
end
'      x     f'
w=[xx' ff']
```

تمارين 3 -1

(1) استخدم برنامج تنصيف الفترة على أن يتم التكرار لخمسة تتابعات و ذلك لحساب جذر للمعادلة $f(x)=\tan(\pi-x)-x=0$ داخل النطاق $[1.6,3]$.

(2) استخدم برنامج تنصيف الفترة تنصيف الفترة لحل المعادلات غير الخطية في إيجاد جذر للمعادلة $f(x)=x^6-x-1=0$ داخل النطاق $[1,2]$.

(3) في طريقة تنصيف الفترة لإيجاد جذر للدالة $f(x)$ إذا كان طول الفترة الأصلية المحتمل وجود الجذر بها هو 1 ، فأوجد عدد تتابعات لتنفيذ الخوارزمية اللازم للوصول إلى دقة 0.001

(4) استخدم تنصيف الفترة لإيجاد جذر للدالة $f(x)=\sqrt{x}-\cos x=0$ في الفترة $[0,1]$ حتى دقة 10^{-3}.

(5) لنأخذ في الاعتبار الدالة $x^2=5x-3$. أوجد نطاق $[a,b]$ ذي يحتوي على جذر لهذه المعادلة.

(6) ارسم الدالتين $y = x$ و $y = 2\sin x$ في نفس الشكل. حدد نقطة التقاطع. خذ فترة تحتوي نقطة التقاطع ثم استخدم تنصيف الفترة لإيجاد جذر للدالة $f(x)=x - 2\sin x = 0$ حتى دقة 10^{-4}.

(7) صمم برنامج ماتلاب لحل التمارين (5)و(6) باستخدام تنصيف الفترة مرة و باستخدام الموضع الزائف مرة أخرى

(8) استخدم برنامج الموضع الزائف على أن يتم التكرار لخمسة تتابعات و ذلك لحساب جذر للمعادلة $f(x) = \tan(\pi - x) - x = 0$ داخل النطاق $[1.6,3]$.

(9) استخدم برنامج الموضع الزائف تنصيف الفترة لحل المعادلات غير الخطية في إيجاد جذر للمعادلة $f(x) = x^6 - x - 1 = 0$ داخل النطاق $[1, 2]$.

2- Open Domain Method

This class of methods need only one point near the solution . we name it initial guess or estimate , we denote by x_0 .

A sequence of iterated approximation is obtained by the formula

$$x_{n+1} = g'(x_n)$$

For some known function g depends on the used method .

The interval halving (bisection) method and the false position method presented in Section 3.1 converge slowly. More efficient methods for finding the roots of a nonlinear equation are desirable. Four such methods are presented in this section:

1. Fixed-point iteration
2. Newton's method
3. The secant method
4. Aitken method

These methods are called ***open domain methods*** since they are not required to keep the root bracketed in a closed domain during the refinement process. Fixed-point iteration is not a reliable method and is not recommended for use. It is included simply for completeness since it is a well-known method. Muller's method is similar to the secant method. However, it is slightly more complicated, so the secant method is generally preferred. Newton's method and the secant method are two of the most efficient methods for refining the roots of a nonlinear equation

Newton's Method

Consider solving $f(x) = 0$ with initial estimate x_0 is given near the root α . The iterates of Newton's method are generated by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \ \mathrm{n} = 0, 1, 2, \dots$$

For a general equation $f(x) = 0$, we assume we are given an initial estimate x_0 of the root α.
The first few formulae are:

نريد أن نجد الجذر α لدالة معطاة $f(x)$. أي أننا نريد أن نجد نقطة x عندها يتقاطع منحنى الدالة $y = f(x)$ مع محور x.

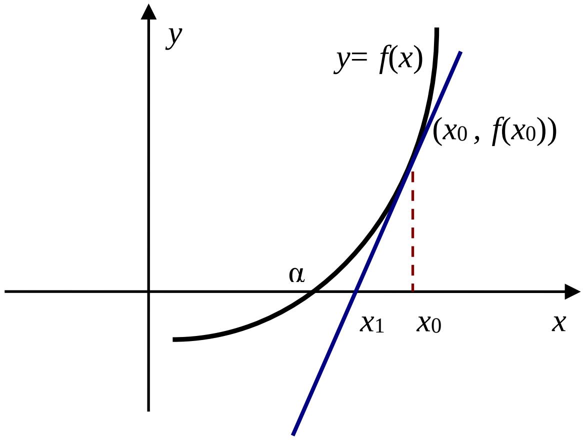
وبما أننا لا يمكننا أن نجد حلا لهذه المسألة، فسوف نحل "مسألة تقريبية". والمسألة التقريبية هي أن نجد خطاً مستقيماً (كتقريب أو أبسط حالة لمنحنى الدالة) يتقاطع مع المحور السيني. أي أننا سوف نستبدل $f(x) = 0$ بحل $p(x) = 0$ لمعادلة الخط المستقيم الذي يصل بين النقطة $(x, f(x))$ وهي أي نقطة عليه ونقطة التماس $(x_0, f(x_0))$

إذن معادلته:

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

أو الصورة:

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



وحيث إن x_1 نقطة تقاطع هذا الخط مع محور x أي أن $f(x_1)=0$

$$0=f(x_0)+(x_1-x_0)f'(x_0)$$

أو الصورة:

$$x_1=x_0-\frac{f(x_0)}{f'(x_0)}$$

وبما أنه من المتوقع أن تكون x_1 جذراً محسناً عن x_0 كتقدير للجذر α،

فإنه يمكن تكرار هذه الفكرة مع اعتبار x_1 هي لتخمين الابتدائي،

وهذا يؤدي إلى التقريب الجديد

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

بتكرار هذه العملية سوف نحصل على متتابعة من الأعداد

$$x_1,x_2,x_3,....$$

والتي نأمل أن تتقارب إلى الجذر المطلوب α. هذه الأعداد تسمى التقريبات المتتالية للجذر.

Example 3.2:

Use Newton formula to obtain the root α of

$f(x) = x^2 - 4x + 4$

Starting from $x_0=0$

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

$f'(x) = 2x - 4$

$X_0 = 0$, $f(x_0) = f(0) = 4$,

$f'(x_0) = f'(0) = -4$

from the formula

$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ n = 0 ,1, 2, …………

n = 0 $\Longrightarrow$ $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 0 - \frac{4}{-4} = 1$, $f(x_1) = 1$,

$f'(x_1) = -2$

n = 1 $\Longrightarrow$ $x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 1.5$,

$f(x_2) = 0.25$

n = 2 $\Longrightarrow$ $x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} = 1.75$,

$f(x_3) = 0.0625$

Iteration no	x_i	$\lvert f(x_i) \rvert$	$\lvert x_{i+1} - x_i \rvert$
0	0	4	0

1	1	1	1
2	1.5	0.25	0.5
3	1.75	0.0625	0.25

Newton's Alghoithm :

Input : function $f(x)$, its derivative $f'(x)$ initial estimate x_0 , small positive value ϵ ,largest number of iteration N .

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

$f'(x)= 2x - 4$, $x_0 = 0$

$\epsilon = 1 \times 10^{-5}$, N = 10

Ouput : x_n that satisfies $|f(x_n)| \leq \epsilon$

Loop : from n = 0 to N

$f_n = f(x_n)$, $f_{dn} = f'(x_n)$

$x_{np1} = x_n - \frac{fn}{fdn}$

$f_{np1} = f(x_{np1})$

if $\left| f(x_{np1}) \right| \leq \epsilon$, then

output x_{np1} as the root of $f(x)$

stop

otherwise continue loop

Programming Exercise

Use your favorite program to find numerical solution for the following non linear equations using Newton's method with the given estimate :

1- $f(x) = x^2 - 4x + 4$,x_0=0

2- $f(x) = x^3 - 2x - 5$, x_0=1.5

3- $f(x) = x^6 - x - 1 = 0$, x_0=1.5

Use your favorite program that uses Newton method for

$f(x) = x^6 - x - 1 = 0$

We use an initial guess of x_0=0.0 and 1.0

<pre>f=@(x)(x^6)-x-1;
fd=@(x)6*(x^5)-1;
x=1.0;
n=0;
while n<6
   f1=f(x);
   fd1=fd(x);
   x=x-f1/fd1;
   a=[n x f1]
   n=n+1;
end</pre>
	<pre>a =
	      0 1.2000 -1.0000
	a =
	   1.0000 1.1436 0.7860
	a =
	   2.0000 1.1349 0.0930
	a =
	   3.0000 1.1347 0.0019
	a =
	   4.0000 1.1347 0.0000
	a =
	   5.0000 1.1347 0.0000</pre>

n	x_n	$f(x_n)$	$x_n - x_{n-1}$	$\alpha - x_{n-1}$
0	1.5	8.89E + 1		
1	1.30049088	2.54E + 1	−2.00E − 1	−3.65E − 1
2	1.18148042	5.38E − 1	−1.19E − 1	−1.66E − 1
3	1.13945559	4.92E − 2	−4.20E − 2	−4.68E − 2
4	1.13477763	5.50E − 4	−4.68E − 3	−4.73E − 3
5	1.13472415	7.11E − 8	−5.35E − 5	−5.35E − 5
6	1.13472414	1.55E − 15	−6.91E − 9	−6.91E − 9

As seen from the output, the convergence is very rapid. The iterate x_6 is accurate to the machine precision of around 16 decimal digits. This is the typical results seen with Newton's method for most problems, but not all.

Example 3.3:

consider the nonlinear equation

$$f(x) \equiv b - 1/x = 0$$

(a) Use approach of Newton's method to evaluate a recurrence formula that approximate the solution.

(b) Express the relative error in the n stage of using the recurrence formula in terms of the error in the initial stage.

(c) Evaluate the interval of convergence

Answer:

(a) We consider a number b > 0, and the equation

$$f(x) \equiv b - 1/x = 0 \qquad (1)$$

The solution is, of course, $\alpha = 1/b$.

Let x_0 be an estimate of the root $\alpha = 1/b$.
We have

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

Using Newton iteration , we have

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \qquad n = 0,1,......$$

it become

$$x_{n+1} = x_n - \frac{b - \dfrac{1}{x_n}}{\dfrac{1}{x_n^2}}$$

بالضرب بسط و مقاما في x_n^2

or simply

$$x_{n+1} = x_n(2 - bx_n), \qquad n = 0,1,2,... \qquad (2)$$

for solving (1)

We use a method of analysis which works for only this example, and later we use another approach to the general Newton's method.

(b) Write

$$x_{n+1} = x_n(1 + 1 - bx_n) = x_n(1 + r_n)$$

Where $r_n = 1 - bx_n$

Note that the error and relative error in x_n are given by

$$e_n = \frac{1}{b} - x_n = \frac{r_n}{b}, \quad rel\left(x_n\right) = \frac{e_n}{\alpha} = \frac{r_n}{b} \cdot b = r_n,$$

Thus r_n is the relative error and we have x_n converges to α if and only if r_n tends to zero. We find a recursion formula for r_n, recalling that $r_n = 1 - bx_n$ for all n. Then

$$\begin{aligned} r_{n+1} &= 1 - bx_{n+1} \\ &= 1 - bx_n\left(1 + r_n\right) \\ &= 1 - \left(1 - r_n\right)\left(1 + r_n\right) \\ &= 1 - \left(1 - r_n^2\right) = r_n^2 \end{aligned}$$

Thus

$$r_{n+1} = r_n^2$$

for every integer $n \geq 0$. Thus

$$r_1 = r_0^2, \quad r_2 = r_1^2 = r_0^4, \quad r_3 = r_2^2 = r_0^8$$

By induction, we obtain

$$r_n = r_0^{2^n}, \qquad n = 0, 1, 2, ... \tag{3}$$

(c) We can use this to analyze the convergence of

$$x_{n+1} = x_n\left(1 + r_n\right), \qquad r_n = 1 - bx_n$$

In particular, we have $r_n \to 0$ if and only if

$$\left|r_0\right| < 1$$

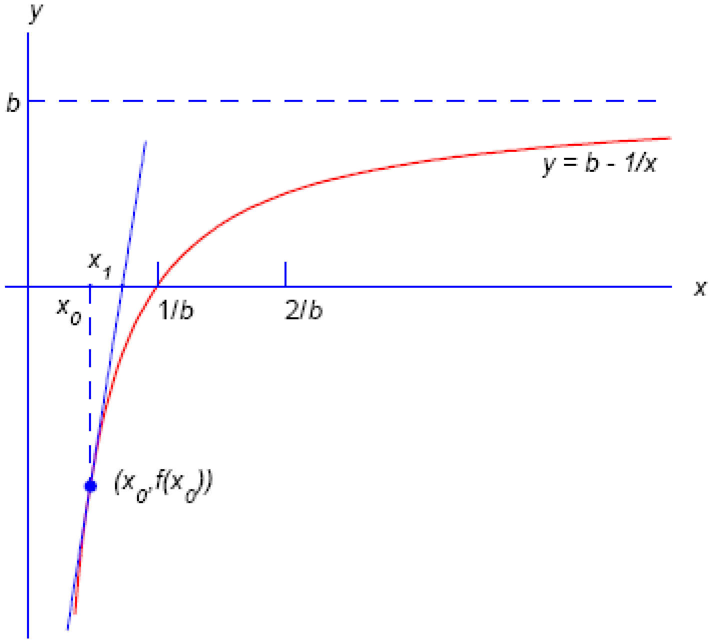
This is equivalent to saying

$$-1 < 1 - bx_0 < 1$$

$$0 < x_0 < \frac{2}{b}$$

Which is called the '***interval of convergence***'.

A look at a graph of $f(x) \equiv b - \dfrac{1}{x}$ will show the reason for this condition. If x_0 is chosen greater than $\dfrac{2}{b}$, then x_1 will be negative, which is unacceptable.



To see why, consider the relative errors in the above. Assume the initial guess x_0 has been so chosen that $r_0 = .1$. Then

$$r_n = r_0^{2^n}, \qquad n = 0, 1, 2, 3, \ldots$$

$$r_1 = 10^{-2}, \quad r_2 = 10^{-4}, \quad r_3 = 10^{-8}, \quad r_4 = 10^{-16}$$

Thus very few iterates need be computed.

ADVANTAGES & DISADVANTAGES

Advantages:
1. It is rapidly convergent in most cases.
2. It is simple in its formulation, and therefore relatively easy to apply and program.
3. It is intuitive in its construction. This means it is easier to understand its behaviour, when it is likely to behave well and when it may behave poorly.

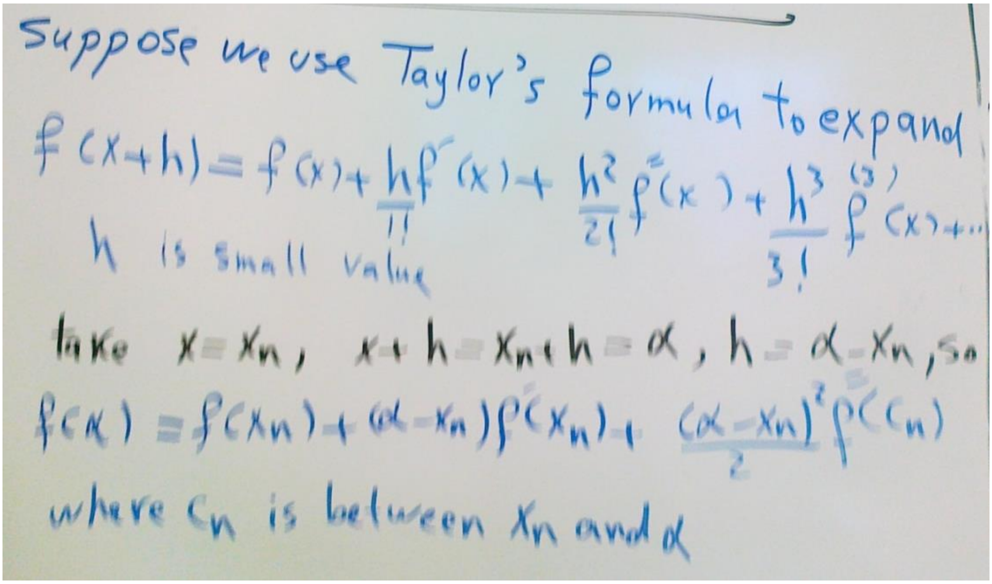
Disadvantages:
1. It may not converge.
2. It is likely to have difficulty if $f'(\alpha) = 0$. This condition means the x-axis is tangent to the graph of y = $f(x)$ at x = α.
3. It needs to know both $f(x)$ and $f'(x)$. Contrast this with the bisection method which requires only $f(x)$.

AN ERROR FORMULA

Example

Derive an error formula for Newton method to solve a nonlinear equation then prove that Newton method is sensitive for the initial estimate. find the interval of initial estimate for Newton method to be convergent.

Answer:



Suppose we use Taylor's formula to expand f (α) beside x = x_n. Then we have

$$f(\alpha) = f(x_n) + (\alpha - x_n) f'(x_n) + \frac{1}{2}(\alpha - x_n)^2 f''(c_n)$$

for some c_n between α and x_n. Note that $f(\alpha) = 0$. Then divide both sides of this equation by $f'(x_n) \neq 0$, yielding

$$0 = \frac{f(x_n)}{f'(x_n)} + \alpha - x_n + (\alpha - x_n)^2 \frac{f''(c_n)}{2f'(x_n)}$$

Note that

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

and thus

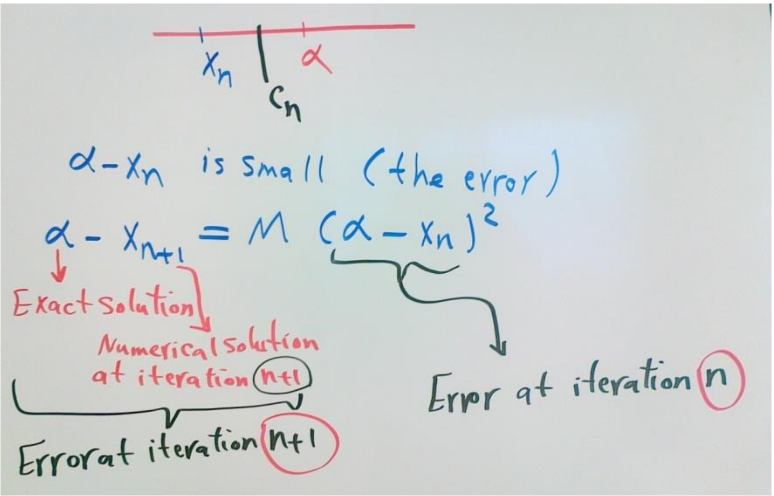
$$\alpha - x_{n+1} = \frac{-f''(c_n)}{2f'(x_n)}(\alpha - x_n)^2$$

$$\alpha - x_{n+1} = M(\alpha - x_n)^2, \qquad (3.1)$$

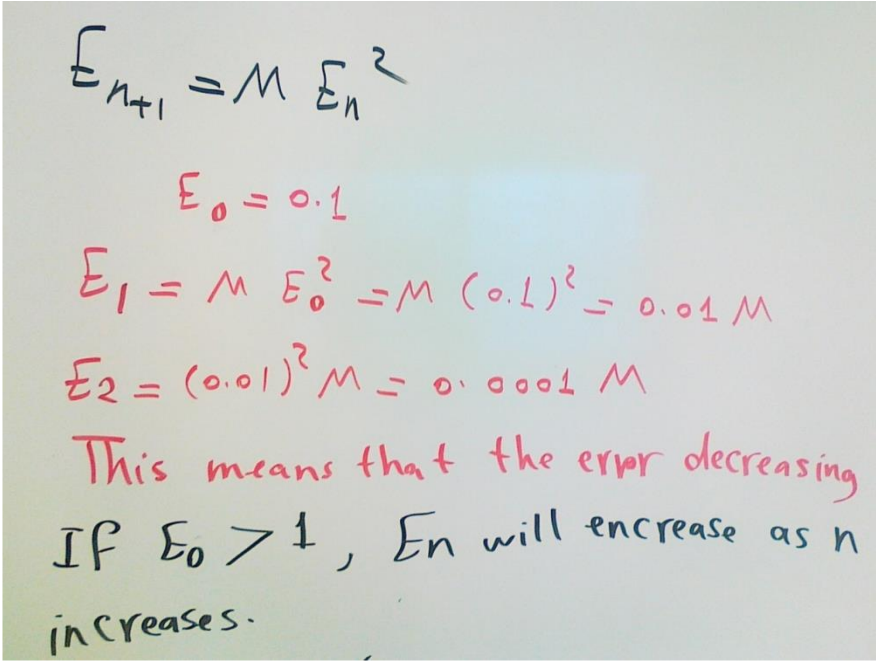
where, since x_n close to α, and therefore c_n also close to α, we have

$$M = \frac{-f''(c_n)}{2f'(x_n)} \cong \frac{-f''(\alpha)}{2f'(\alpha)}$$

Thus Newton's method is quadratically convergent, provided $f'(\alpha) \neq 0$ and $f(x)$ is twice differentiable beside the root α.



The error in iteration n+1 is proportial to the square error in iteration n



We can also use this to explore the ‘interval of convergence’ of Newton’s method. Write the above as

$$\alpha - x_{n+1} \approx M\left(\alpha - x_n\right)^2, \; M = \frac{-f''(\alpha)}{2f'(\alpha)}, \quad (3.1b)$$

Multiply both sides by M to get

$$M(\alpha - x_{n+1}) = M^2(\alpha - x_n)^2 = (M(\alpha - x_n))^2$$

$$M(\alpha - x_{n+1}) = (M(\alpha - x_{n-1}))^{2^2}$$

$$= \cdots = \cdots = \cdots = (M(\alpha - x_0))^{2^n}, n \geq 1$$

Then we want the quantity $\left[M(\alpha - x_0)\right]^{2^n}$ to decrease; and this suggests choosing x_0 so that

$$\left|M(\alpha - x_0)\right| < 1 \;\Rightarrow\; \left|\alpha - x_0\right| < \frac{1}{M} = \left|\frac{2f'(\alpha)}{f''(\alpha)}\right| \qquad (3)$$

If |M| is very large, then we may need to have a very good initial guess in order to have the iterates x_0 converge to α.

Example

Estimate the error in Newton method for obtaining the root of

$$f(x) = x^6 - x - 1 = 0$$

Answer

$$(\alpha - x_{n+1}) = (\alpha - x_n)^2\left[\frac{-f''(c_n)}{2f'(x_n)}\right]$$

$$\frac{-f''(c_n)}{2f'(x_n)} \cong \frac{-f''(x_n)}{2f'(x_n)} = \frac{-30x_n^4}{2(6x_n^5 - 1)} = \frac{-15x_n^4}{6x_n^5 - 1}\Big|_{x_n = 1} \cong \frac{-15}{6-1} = -3$$

So the error in step n+1 equal the square of the error in step n multiply in 3.

Example

obtain an iterative formula using Newton method to find $\sqrt{a}$ for real number a. Use the resulting iterative formula to find $\sqrt{5}$ to three digits of accuracy.

Consider using Newton method to find a root of $\sqrt{a}$.

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

find the interval D that containes x_0 such that Newton iteration is convergent.

Answer

نفرض أن $x = \sqrt{a}$ و بالتالي $x^2 = a$ أي $f(x) = x^2 - a = 0$

و باستخدام قانون نيوتن

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

$$x_{i+1} = x_i - \frac{x_i^2 - a}{2x_i} = \frac{1}{2}\left(x_i + \frac{a}{x_i}\right)$$

و عندما $a = 5$ فإن

$$x_{i+1} = \frac{1}{2}\left(x_i + \frac{5}{x_i}\right)$$

فإذا أخذنا $x_0 = 2$ فإن التتابعات المتتالية للوصول للجذر تكون كما في الجدول التالي

n	x_n	$f(x_n)$	x_n-x_{n-1}
0	2	-1.000000000	0.250000000
1	2.250000000	0.062500000	-0.013888889
2	2.236111111	0.000192901	-0.000043133
3	2.236067978	0.000000002	0.000000000
4	2.236067977	0.000000000	

جدول 3 -1: حساب تتابعات طريقة نيوتن لإيجاد الجذر التربيعي لـ 2

THE SECANT METHOD

The Secant Method:

When the derivative function, $f'(x)$, is unavailable or costly to evaluate, an alternative to Newton's method is required. The preferred alternative is the *secant method.*

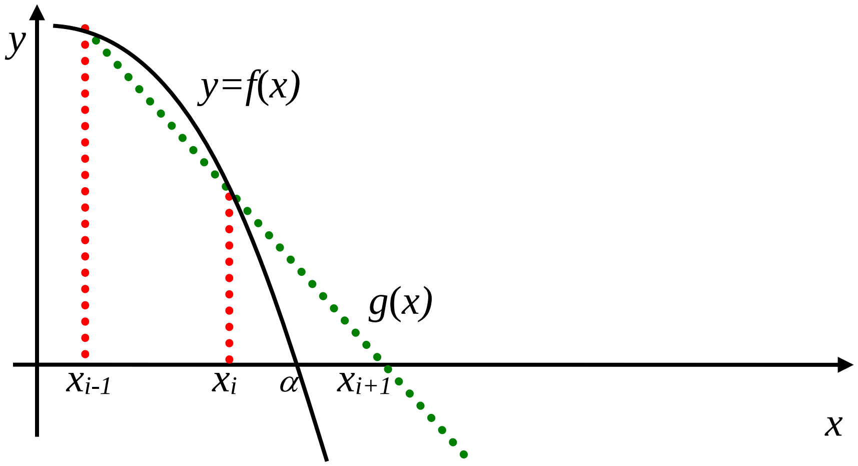
The secant method is illustrated graphically in Figure 3.4. The nonlinear function $f(x)$ is approximated locally by the linear function $g(x)$, which is the secant to $f(x)$, and the root of $g(x)$ is taken as an improved approximation to the root of the nonlinear function $f(x)$.

A secant to a curve is the straight line which passes through two points on the curve.

The procedure is applied repetitively to convergence. Two initial approximations x_0, x_1, which are not required to bracket the root, are required to initiate the secant method.

The slope of the secant passing through two points, x_{i-1}, x_i , is given by

$$g'(x_i) = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \qquad (1)$$



The equation of the secant line is given by

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

where $f(x_{i+1}) = 0$. Solving this Eq. for x_{i+1} yields

$$x_{i+1} = x_i - \frac{f(x_i)}{g'(x_i)} \quad (2)$$

Equations(1)-(2) are applied repetitively until either one or both of the following two convergence criteria are satisfied:

$$(i)|x_{i+1} - x_i| \le \epsilon, \qquad (ii)|f(x_i)| \le \epsilon$$

Example

We solve the equation

$$f(x) = x^6 - x - 1 = 0$$

which was used previously as an example for both the bisection and Newton methods. The quantity $x_n - x_{n-1}$ is used as an estimate of $\alpha - x_{n-1}$. The iterate x_9 equals α rounded to nine significant digits. As with Newton's method for this equation, the initial iterates do not converge rapidly. But as the iterates become closer to α, the speed of convergence increases.

Programming Exercise

Use your Matlab program using Secant method for example to obtain the following results:

clear
x0=2.0;
x1=1.0;
f0=(x0^6)-x0-1;
f1=(x1^6)-x1-1;
n=0;
	n x0 f0
	a=
	0 2 61
	a=
	1 1 -1

a='n x0 f0'
a=[n x0 f0]
n=1;
a=[n x1 f1]
while n<5
  f0=(x0^6)-x0-1;
  f1=(x1^6)-x1-1;
  x=x1-f1*(x1-x0)/(f1-f0);
  n=n+1;
  x0=x1;
  x1=x;
  f=(x^6)-x-1;
  a=[n x f]
end
	a=
	2.0000 1.0161 -0.9154
	a=
	3.0000 1.1906 0.6575
	a=
	4.0000 1.1177 -0.1685
	a=
	5.0000 1.1325 -0.0224

n	x_n	$f(x_n)$	$x_n - x_{n-1}$	$\alpha - x_{n-1}$
0	2.0	61.0		
1	1.0	-1.0	-1.0	
2	1.01612903	$-9.15E-1$	$1.61E-2$	$1.35E-1$
3	1.19057777	$6.57E-1$	$1.74E-1$	$1.19E-1$
4	1.11765583	$-1.68E-1$	$-7.29E-2$	$-5.59E-2$
5	1.13253155	$-2.24E-2$	$1.49E-2$	$1.71E-2$
6	1.13481681	$9.54E-4$	$2.29E-3$	$2.19E-3$
7	1.13472365	$-5.07E-6$	$-9.32E-5$	$-9.27E-5$
8	1.13472414	$-1.13E-9$	$4.92E-7$	$4.92E-7$

It is clear from the numerical results that the secant method requires more iterates than the Newton method. But note that the secant method does not require a knowledge of $f'(x)$, whereas Newton's method requires both $f(x)$ and $f'(x)$.

Comparison of Newton and Secant Methods

From the foregoing discussion, Newton's method converges more rapidly than the secant method. Thus, Newton's method should require fewer iterations to attain a given error tolerance.

The Newton method

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, n = 0,1,2,......$$

requires one function and one derivative evaluations per iteration, that of $f(x_n)$ and $f'(x_n)$. The secant method

$$x_{n+1} = x_n - f(x_n) \div \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}, \qquad n = 1,2,3...$$

requires just two function evaluations per iteration.

The derivative evaluation is more complicated than function evaluation. Indeed the numerical approximation for derivative (5) requires two function evaluations.

For this reason, the secant method is often faster in time, even though more iterates are needed with it than with Newton's method to attain a similar accuracy.

Advantages & Disadvantages

Advantages of secant method:

1. It converges at faster than a linear rate, so that it is more rapidly convergent than the bisection method.

2. It does not require use of the derivative of the function, something that is not available in a number of applications.

3. It requires only one function evaluation per iteration, as compared with Newton's method which requires two.

Disadvantages of secant method:

1. It may not converge.
2. There is no guaranteed error bound for the computed iterates.
3. It is likely to have difficulty if $f'(\alpha)$. This means the x-axis is tangent to the graph of y = $f(x)$ at x = α.
4. Newton's method generalizes more easily to new methods for solving simultaneous systems of nonlinear equations.

Fixed Point Iteration

One of the most frequently recurring ideas in numerical calculations is iteration or successive approximation. Taken generally, iteration means the repetition of a pattern of action or process.

To illustrate a more specific use of the idea of iteration, we consider the problem of solving a (usually) nonlinear equation of the form

$$x = g(x) \tag{1}$$

where g is assumed to be a differentiable function whose value can be computed for any given value of a real variable x within a certain interval. Using the method of iteration, one starts with an initial approximation x_0, and computes the sequence

$$x_1 = g(x_0),\ x_2 = g(x_1),\ x_3 = g(x_2), \ldots. \tag{2}$$

Each computation of the type $x_{i+1} = g(x_i)$ is called a fixed point iteration.

As i □grows, we would like the numbers x_i to be better and better estimates of the desired root.

If the sequence $\{x_n\}$ converges to a limiting value α then we have

$$\alpha = \lim_{n\to\infty} x_{n+1} = \lim_{n\to\infty} g(x_n) = g(\alpha)$$

so $x = \alpha$ satisfies the equation $x = g(x)$. One can then stop the iterations when the desired accuracy has been attained.

Example

مثال1:

استخدم طريقة النقطة الثابتة لحساب جذر المعادلة الغير خطية

$f(x) = x^2 - 5 = 0$

من خلال الصيغة التتابعية

$$x_{i+1} = 1 + x_i - \tfrac{1}{5} x_i^2$$

بالقرب من $x_0 = 2.5$.

الحل:

clear
' i x '
x=2.5;
i=0;
while i<5
x=1+x-(x^2)/5;
i=i+1;
a=[i x]
End
	i x
	a=
	1.0000 2.2500
	a=
	2.0000 2.2375
	a=
	3.0000 2.2362
	a=
	4.0000 2.2361
	a=
	5.0000 2.2361

The Newton method and the secant method are examples of one-point and two-point iteration methods respectively. In this section, we give a more general introduction to iteration methods, presenting a general theory for one-point iteration formulas.

Example

Consider solving the equation

$$f(x) = x^2 - 5 = 0 \qquad (1)$$

for the root $\alpha = \sqrt{5} = 2.2361$. Test which of these four iteration methods to solve this equation converges?

1. $x_{n+1} = 5 + x_n - x_n^2$ 2. $x_{n+1} = 5 / x_n$

3. $x_{n+1} = 1 + x_n - \tfrac{1}{5} x_n^2$ 4. $x_{n+1} = \tfrac{1}{2}\left(x_n + 5 / x_n\right)$

^^

All four iterations have the property that if the sequence $\{x_n |\ n \geq 0\}$ has a limit α, then is a root of (1).

For each equation, check this as follows: Replace x_n and x_{n+1} by α, and then show that this implies $\alpha = \pm\sqrt{5}$. In the following Exercise, you shall evaluate the iterates x_n for these four iteration methods.

Programming Exercise :

Write Matlab program evaluate the iteraion values for (1) making use of the above four methods statrting from x_0=2.5 to obtain the following table:

```
x2=2.5;
x3=2.5;
x4=2.5;
n=0;
while n<5
x1=5+x1-x1^2;
x2=5/x2;
x3=1+x3-0.2*x3^2;
x4=0.5*(x4+5/x4);
a=[n  x1  x2  x3  x4]
```

```
>> fixedpoint1

a =

         0    1.2500    2.0000    2.2500    2.2500


a =

    1.0000    4.6875    2.5000    2.2375    2.2361


a =

    2.0000  -12.2852    2.0000    2.2362    2.2361


a =

    3.0000 -158.2102    2.5000    2.2361    2.2361
```

n=n+1; End	

n	x_n : 1	x_n : 2	x_n : 3	x_n : 4
0	2.5	2.5	2.5	2.5
1	1.25	2.0	2.25	2.25
2	4.6875	2.5	2.2375	2.2361
3	−12.2852	2.0	2.2362	2.2361

To explain these numerical results. we present a general theory for one point iteration formula.

As another example, note that the Newton method

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

is also a fixed point iteration, for the equation

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

In general, we are interested in solving equations

$$x = g(x)$$

by means of fixed point iteration:

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

If the iterates x_n converge to a point α. then

$$\lim_{n\to\infty} x_{n+1} = \lim_{n\to\infty} g(x_n)$$

$$\alpha = g(\alpha)$$

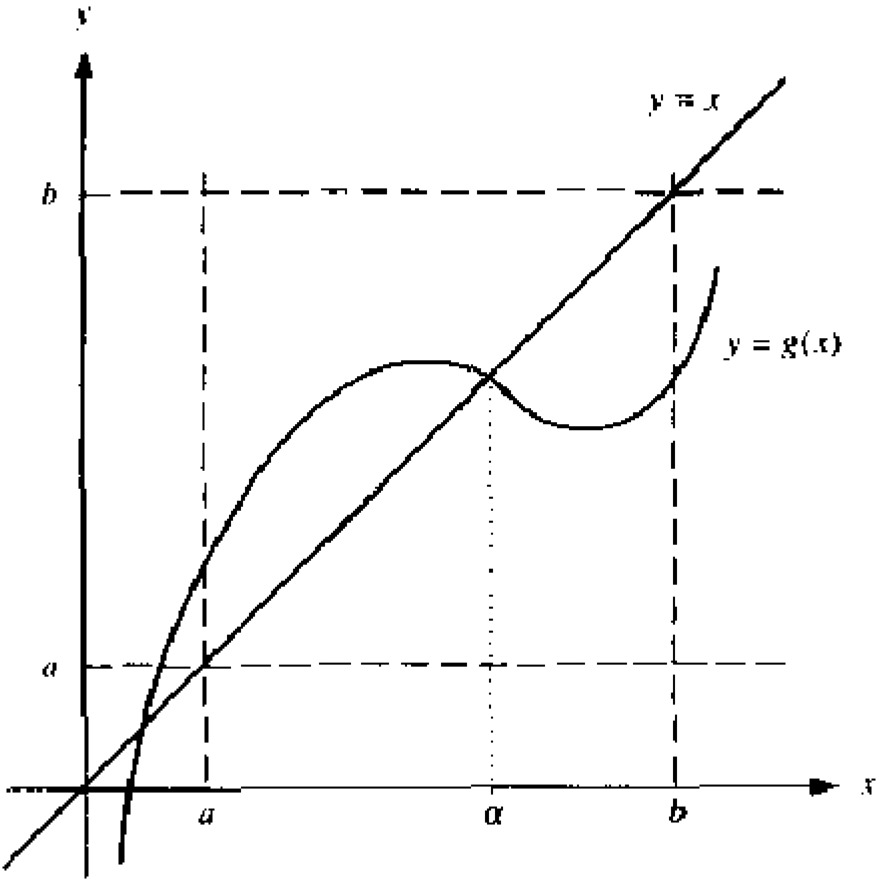
Existence Theorem

We begin by asking whether the equation $x = g(x)$ has a solution. For this to occur, the graphs of $y = x$ and $y = g(x)$ must intersect. The lemmas and theorems in the section give conditions under which we are guaranteed there is a fixed point α.

Lemma: Let $g(x)$ be a continuous function on the interval [a, b], and suppose it satisfies the property

$$a \leq x \leq b \quad \Rightarrow \quad a \leq g(x) \leq b \tag{1}$$

Then the equation $x = g(x)$ has at least one solution α in the interval [a, b]. See the graphs for examples.



Theorem(1): Assume g(*x*) and g'(*x*) exist and are continuous on the interval [a, b]; and further, assume

$$a \le x \le b \quad \Rightarrow \quad a \le g(x) \le b$$

$$\lambda \equiv \max_{a \le x \le b} \left| g'(x) \right| < 1 \tag{2}$$

Then:

S1. The equation $x = g(x)$ has a unique solution α in [a, b].

S2. For any initial guess x_0 in [a, b], the iteration

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

will converge to α.

S3. $$|\alpha - x_n| \le \frac{\lambda^n}{1-\lambda} |x_1 - x_0|, \quad n \ge 0$$

S4.
Thus for x_n close to α, $\lim_{n\to\infty} \frac{\alpha - x_{n+1}}{\alpha - x_n} = g'(\alpha)$

$$\alpha - x_{n+1} \approx g'(\alpha)(\alpha - x_n)$$

Proof. Note first that the hypotheses on g allow us to use the previous Lemma to assert the existence of at least one solution to x = g(x). In addition. using the mean value theorem, we have that for any two points w and z in [a, b]

$$g(w) - g(z) = g'(c)(w - z)$$

for some c between w and z. By using (2), we obtain

$$\begin{aligned} |g(w) - g(z)| &= |g'(c)|\,|w - z| \\ &\le \lambda\,|w - z| \qquad a \le w, z \le b \end{aligned}$$

(3)

S1. Suppose there are two solutions, denoted by $\alpha = g(\alpha)$ and $\beta = g(\beta)$. By subtracting these, we find that
$\alpha - \beta = g(\alpha) - g(\beta)$
Take absolute values and use (3):

$$|\alpha - \beta| \le \lambda |\alpha - \beta|$$
$$(1 - \lambda) |\alpha - \beta| \le 0$$

Since $\lambda < 1$, we must have $\alpha = \beta$; and thus, the equation x=g(x) has only one solution in the interval [a, b].

S2. From the assumption (1), it can be shown that for any initial guess x_0 in [a, b], the iterates x_n will all remain in [a, b].

For example, if $a \le x_0 \le b$, then (1) implies $a \le g(x_0) \le b$. Since x_1= g(x_0),this shows x_1 is in [a,b]. Repeat the argument to show that x_2 = g(x_1) is in [a, b], and continue the argument inductively.
To show that the iterates converge, subtract x_{n+1} = g(x_n) from $\alpha = g(\alpha)$, obtaining

$$\alpha - x_{n+1} = g'(c_n)(\alpha - x_n) \qquad (4)$$

for some C_n between α and x_n. Using the assumption (2), we get

$$|\alpha - x_{n+1}| \le \lambda |\alpha - x_n|, \qquad n \ge 0$$

Inductively. we can then show that

$$|\alpha - x_n| \le \lambda^n |\alpha - x_0|, \qquad n \ge 0 \qquad (5)$$

Since $\lambda < 1$, the right side of (5) goes to zero as $n \to \infty$, and this then shows that $x_n \to \alpha$ as $n \to \infty$.

S3. Use (3) with n = 1 to obtain

$$|\alpha - x_0| = |\alpha - x_1 + x_1 - x_0|$$

Using triangle inequilty

$$\begin{aligned} |\alpha - x_0| &\le |\alpha - x_1| + |x_1 - x_0| \\ &\le \lambda |\alpha - x_0| + |x_1 - x_0| \\ (1-\lambda)|\alpha - x_0| &\le |x_1 - x_0| \\ |\alpha - x_0| &\le \frac{1}{1-\lambda}|x_1 - x_0| \end{aligned}$$

combining this to (5) we obtain the required.

S4. Use (4) to write

$$\alpha - x_{n+1} = g'(c_n)(\alpha - x_n)$$

$$\lim_{n\to\infty} \frac{\alpha - x_{n+1}}{\alpha - x_n} = \lim_{n\to\infty} g'(c_n)$$

Each c_n is between α and x_n, and $x_n \to \alpha$, by S2, Thus, $c_n \to \alpha$. Combine this with the continuity of the function g'(x) to obtain

$$\lim_{n\to\infty} g'(c_n) = g'(\alpha)$$

thus proving the required

Corollary:

Assume that g(x) and g'(x) are continuous for some interval c < x < d, with the fixed point α contained in this interval. Moreover, assume that

$$|g'(\alpha)| < 1 \qquad (7)$$

Then, there is an interval [a, b] around α for which the hypotheses, and hence also the conclusions, of Theorem(1) are true.

If the contrary, $|g'(\alpha)| < 1$, then the iteration method $x_{n+1} = g(x_n)$ will not converge to α,

{When $|g'(\alpha)| = 1$, no conclusion can be drawn; and even if convergence occurs, the method would be far too slow for the iteration method to be practical}

Answer of Example 3.6:

Using the result, $|g'(\alpha)| < 1$ as a condition for convergence, we can examine the iteration methods in example 1, namely:

1. $x_{n+1} = 5 + x_n - x_n^2$
2. $x_{n+1} = 5 / x_n$
3. $x_{n+1} = 1 + x_n - \frac{1}{5} x_n^2$
4. $x_{n+1} = \frac{1}{2}\left(x_n + 5 / x_n\right)$

1. $g(x) = 5 + x - x^2, \quad g'(x) = 1 - 2x, \quad g'(\alpha) = 1 - 2\sqrt{5} < -1$

Thus, the iteration 1 will not converge to $\sqrt{5}$.

2. $g(x) = 5/x, \quad g'(x) = -5/x^2, \quad g'(\alpha) \doteq -1$.

We cannot conclude that the iteration. converges or diverges. But from Table 1, it is clear that the iterates will not converge to α.

3. $g(x) = 1 + x - \frac{1}{5}x^2, \quad g'(x) = 1 - \frac{2}{5}x, \quad g'(\alpha) = 1 - \frac{2}{5}\sqrt{5} \doteq 0.106$

From the corollary, the iteration will converge. And from(5),

$$|\alpha - x_{n+1}| \approx 0.106\,|\alpha - x_n|$$

when x_n is close to α. The errors decrease by approximately a factor of 0.1 with each iteration.

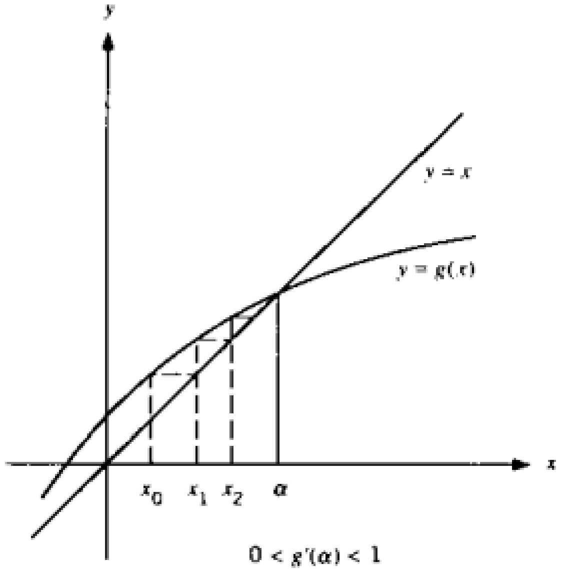
4. $g(x) = \frac{1}{2}(x + 5/x), \quad g'(x) = \frac{1}{2}(1 - 5/x^2), \quad g'(\alpha) = 0$

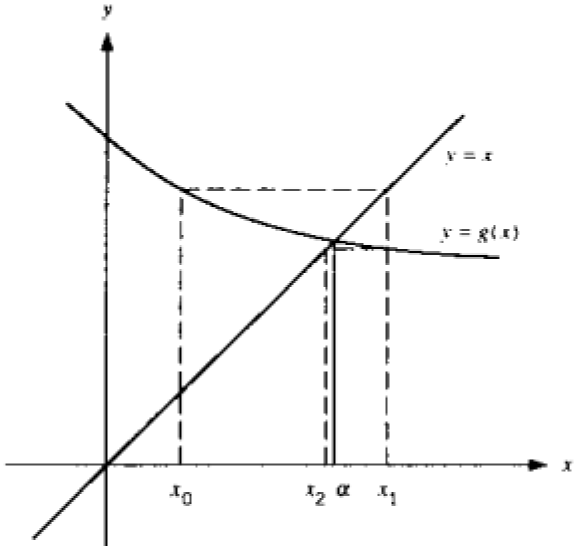
Thus. the condition for convergence is easily satisfied. Note that this is Newton's method for computing $\sqrt{5}$.

Remark:

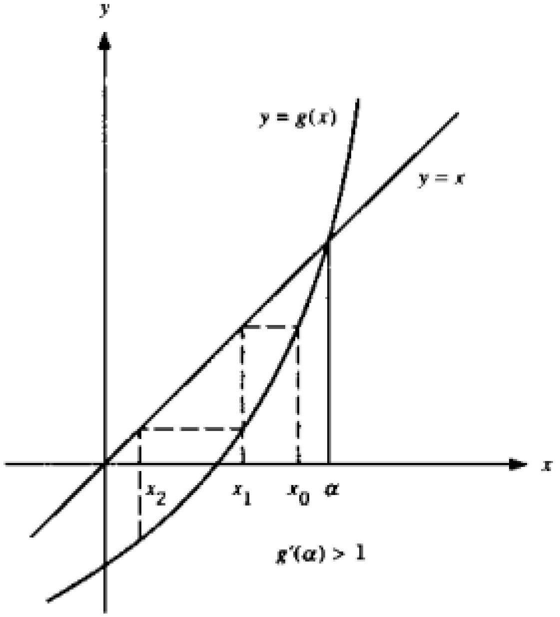
It is often difficult to know how to convert a rootfinding problem $f(x) = 0$ into a fixed point problem $x = g(x)$ that leads to a convergent method.

The possible behavior of the fixed point iterates x_n is shown graphically in Figure 1, for various sizes of g'(α). To see the convergence, consider the case of $x_1 = g(x_o)$, the height of the graph of $y = g(x)$ at x_o . We bring the number x_1 back to the x-axis by using the line $y = x$ and the height $y = x_1$. We continue this with each iterate, obtaining a stairstep behavior when $g'(\alpha)>0$. When $g'(\alpha)<0$. the iterates oscillate around the fixed point α, as can be seen in Figure 1.

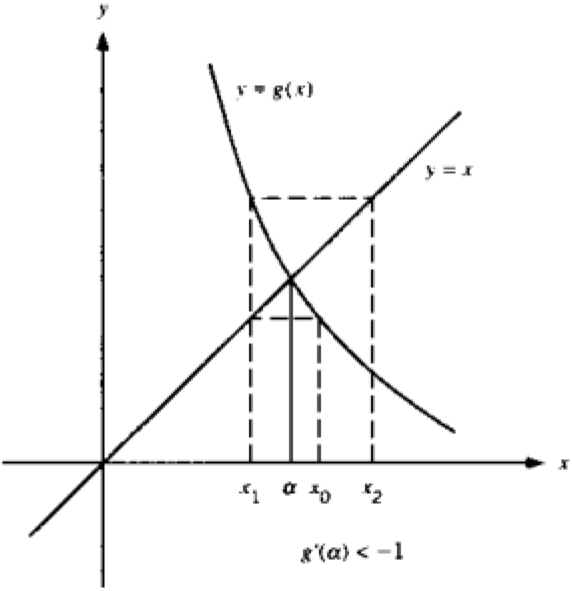




$-1 < g'(\alpha) < 0$



$g'(\alpha) > 1$



Example

In Table 1, we give results from the iteration 3

$$x_{n+1} = 1 + x_n - \frac{1}{5}x_n^2$$

along with more information on the convergence of the iterates. The errors are given, along with the ratios

$$r = g'(\alpha) = \lim_{n\to\infty}\frac{\alpha - x_{n+1}}{\alpha - x_n}$$

Empirically, the values of r converge to g'(α)= 0.105573, which agrees with(S4)(see the following table).

Table 1 The Iteration $x_{n+1} = 1 + x_n - \frac{1}{5}x_n^2$

n	x_n	$\alpha - x_n$	r_n
0	2.5	-2.64E - 1	
1	2.25	-1.39E - 2	0.0528
2	2.2375	-1.43E - 3	0.1028
3	2.23621875	-1.51E - 4	0.1053
4	2.23608389	-1.59E - 5	0.1055
5	2.23606966	-1.68E - 6	0.1056
6	2.23606815	-1.77E - 7	0.1056
7	2.23606800	-1.87E - 8	0.1056

HIGHER ORDER METHODS

The convergence formula $|g'(\alpha)| < 1$ gives less information in the case $g'(\alpha) = 0$,

although the convergence is clearly quite good.

To improve on the results in Existence Theorem, consider the Taylor expansion of $g(x_n)$ about α, assuming that $g(x)$ is twice continuously differentiable:

$$g(x_n) = g(\alpha) + (x_n - \alpha)g'(\alpha) + \frac{1}{2}(x_n - \alpha)^2 g''(c_n) \quad (8)$$

with c_n between x_n and α. Using $g(x_n) = x_{n+1}$ and $g(\alpha) = \alpha$. Also assume $g'(\alpha) = 0$.
Then

$$\begin{aligned} x_{n+1} &= \alpha + \tfrac{1}{2} g''(c_n)\,(x_n - \alpha)^2 \\ \alpha - x_{n+1} &= -\tfrac{1}{2} g''(c_n)\,(x_n - \alpha)^2 \end{aligned}$$

Thus if $g'(\alpha)= 0$, the fixed point iteration is quadratically convergent or better. More ever

$$\lim_{n\to\infty} \frac{\alpha - x_{n+1}}{(\alpha - x_n)^2} = -\frac{1}{2}\, g''(\alpha)$$

If $g''(\alpha) \neq 0$, then this formula shows. that the iteration $x_{n+1} = g(x_n)$is of order 2 or is quadrarically convergent.

If also $g''(\alpha)= 0$, and perhaps also some higher-order derivatives are zero at α, then expand the Taylor series through higher-order terms in (8), until the final error term contains a derivative of g that is nonzero at α. This leads to methods with an order of convergence greater than 2.

Aitken Error Estimation and Extrapolation

Example

Write with proof the Aitken's extrapolation formula with an algorithm to program it, derive from it Aitken's error estimation formula

Answer

Recall the result

$$\lim_{n\to\infty} \frac{\alpha - x_n}{\alpha - x_{n-1}} = g'(\alpha)$$

for the iteration

$$x_n = g(x_{n-1}), \quad n = 1, 2, \ldots$$

Thus

$$\alpha - x_n \approx \lambda(\alpha - x_{n-1}) \quad (1)$$

with $\lambda = g'(\alpha)$ and $|\lambda| < 1$.

If we were to know λ, then we could solve (1) for α:

$$\alpha \approx \frac{x_n - \lambda x_{n-1}}{1-\lambda}$$

Usually, we write this as a modification of the currently computed iterate x_n:

$$\begin{aligned} \alpha &\approx \frac{x_n - \lambda x_{n-1}}{1-\lambda} \\ &= \frac{x_n - \lambda x_n}{1-\lambda} + \frac{\lambda x_n - \lambda x_{n-1}}{1-\lambda} \\ &= x_n + \frac{\lambda}{1-\lambda}[x_n - x_{n-1}] \end{aligned} \quad (2)$$

The formula

$$x_n + \frac{\lambda}{1-\lambda}[x_n - x_{n-1}]$$

is said to be an extrapolation of the numbers x_{n-1} and x_n.

Now for estimating λ, from (1)

we have

$$\lambda \approx \frac{\alpha - x_n}{\alpha - x_{n-1}} \qquad (3)$$

Unfortunately this also involves the unknown root α which we seek; and we must find some other way of estimating λ.

To calculate λ consider the ratio

$$\lambda_n = \frac{x_n - x_{n-1}}{x_{n-1} - x_{n-2}} \qquad (4)$$

To see this is approximately λ as x_n approaches α, write

$$\frac{x_n - x_{n-1}}{x_{n-1} - x_{n-2}} = \frac{g(x_{n-1}) - g(x_{n-2})}{x_{n-1} - x_{n-2}} = g'(c_n)$$

with c_n between x_{n-1} and x_{n-2}. As the iterates approach α, the number c_n must also approach α. Thus λ_n approaches λ as $x_n \to \alpha$.

We combine these results(2)-(4) to obtain the estimation

$$\widehat{x}_n = x_n + \frac{\lambda_n}{1 - \lambda_n}\left[x_n - x_{n-1}\right], \qquad (5)$$

$$\lambda_n = \frac{x_n - x_{n-1}}{x_{n-1} - x_{n-2}}$$

We call $\widehat{x}_n$ the *Aitken extrapolate* of $\{x_{n-2}, x_{n-1}, x_n\}$; and $\alpha \approx \widehat{x}_n$.

We can also rewrite (5) as

$$\alpha - x_n \approx \widehat{x}_n - x_n = \frac{\lambda_n}{1 - \lambda_n}\left[x_n - x_{n-1}\right]$$

This is called ***Aitken's error estimation formula***.

The accuracy of these procedures is tied directly to the accuracy of the formulas(from (11))

$$\alpha - x_n \approx \lambda(\alpha - x_{n-1}),$$
$$\alpha - x_{n-1} \approx \lambda(\alpha - x_{n-2})$$

If this is accurate, then so are the above extrapolation and error estimation formulas.

Example

Consider the iteration

$$x_{n+1} = 6.28 + \sin(x_n), \qquad n = 0, 1, 2, \ldots$$

for solving

$$x=6.28+\sin x$$

So, $g(x)=6.28+\sin x$, Iterates are shown on the accompanying sheet, including calculations of λ_n, the error estimate

$$\alpha - x_n \approx \hat{x}_n - x_n = \frac{\lambda_n}{1-\lambda_n}[x_n - x_{n-1}]$$

The latter is called "Estimate" in the table. In this instance,

$$\lambda_n = \frac{x_n - x_{n-1}}{x_{n-1} - x_{n-2}}$$

$$\lambda = g'(\alpha) \doteq .9644$$

and therefore the convergence is very slow. This is apparent in the table.

Programming Exercise

Use Matlab program to calculates Aitken's error estimation for example 3.8 to obtain the following table. Detect if the iterations converge rapidly or slowly.

n	x_n	$x_n - x_{n-1}$	λ_n	$\alpha - x_n$	Estimate
0	6.0000000			1.55E − 2	
1	6.0005845	5.845E − 4		1.49E − 2	
2	6.0011458	5.613E − 4	.9603	1.44E − 2	1.36E − 2
3	6.0016848	5.390E − 4	.9604	1.38E − 2	1.31E − 2
4	6.0022026	5.178E − 4	.9606	1.33E − 2	1.26E − 2
5	6.0027001	4.974E − 4	.9607	1.28E − 2	1.22E − 2
6	6.0031780	4.780E − 4	.9609	1.23E − 2	1.17E − 2
7	6.0036374	4.593E − 4	.9610	1.18E − 2	1.13E − 2

Aitken's Algorithm

Step 1: Select x_0
Step 2: Calculate

$$x_1 = g(x_0), \quad x_2 = g(x_1)$$

Step3: Calculate

$$\lambda_2 = \frac{x_2 - x_1}{x_1 - x_0}$$

$$x_3 = x_2 + \frac{\lambda_2}{1 - \lambda_2}[x_2 - x_1],$$

Step 4: Calculate

$$x_4 = g(x_3), \quad x_5 = g(x_4)$$

and calculate x_6 as the extrapolate of $\{x_3, x_4, x_5\}$.

$$\lambda_5 = \frac{x_5 - x_4}{x_4 - x_3}, \quad x_6 = x_5 + \frac{\lambda_5}{1-\lambda_5}[x_5 - x_4]$$

Continue this procedure.

Of course in practice we will have some kind of error test to stop this procedure when believe we have sufficient accuracy.

General Comments

Aitken extrapolation can greatly accelerate the convergence of a linearly convergent iteration

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

This shows the power of understanding the behaviour of the error in a numerical process. From that understanding, we can often improve the accuracy, thru extrapolation or some other procedure.

This is a justification for using mathematical analyses to understand numerical methods. We will see this repeated at later points in the course, and it holds with many different types of problems and numerical methods for their solution.

Example 3

Consider solving

$$f(x) = x^6 - x - 1 = 0$$

for its positive root α. An initial guess x_0 can be generated from a graph of $y = f(x)$. The iteration is given by

$$x_{n+1} = x_n - \frac{x_n^6 - x_n - 1}{6x_n^5 - 1}, \qquad n \geq 0$$

We use an initial guess of $x_0 = 1.5$. We shall take $|x_{n+1} - x_n|$ as the numerical error.

Excercises

تمارين (2-3)

(1) أوجد صيغة تتابعيه باستخدام طريقة نيوتن لإيجاد قيمة كل من:

$$(i)\ \sqrt[n]{a}\,,\quad (ii)\ \frac{1}{\sqrt{a}}$$

ومن ثم أوجد قيمة كل منها عندما $a=5,\ n=2$ وذلك حتى دقة ثلاثة مواضع عشرية؟

(2) أعد تنفيذ برنامج ماتلاب(و اكسل) لطريقة نيوتن لمثال (1) مع تغيير نقطة البداية إلى $x_0=1.5,\ 5.0,\ 100$. ماذا تلاحظ؟

(3) استخدم طريقة نيوتن لحساب جذر المعادلة غير الخطية $x=\cos x$؟

(4) من خلال دراسة تقدير الخطأ لطريقة نيوتن أوجد معدل التقارب عند حساب جذر المعادلة غير الخطية $f(x)=x^6-x-1$، ومن ثم حدد الخطأ المتوقع بعد 4 تتابعات. قارن ذلك مع النتائج العددية لمثال (1).

(5) استخدم اكسل لبرمجة طريقة نيوتن لحساب جذر المعادلة غير الخطية

$$f(x)=x^6-x-1$$

وذلك بدءا من $x_0=1.5$. وقارن النتائج مع مثال (1).

(6) لكي نطبق طريقة نيوتن لإيجاد أول جذر موجب للمعادلة:

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

أوجد أكبر فترة $[a,b]$ التي تتقارب فيها طريقة نيوتن لحل هذه المسألة؟

تسهيل: ارسم الدالة المعطاة لتعرف أول جذر موجب، ثم حدد الفترة التي لا يدخل مع هذا الجذر أي جذور أخرى؟

(7) استخدم طريقة القاطع لحساب جذر المعادلة غير الخطية $x = \cos x$ ؟

(8) استخدم طريقتي نيوتن والقاطع لحساب جذر المعادلة غير الخطية $e^x = \sin(\frac{\pi x}{3})$ ؟

(9) استخدم طريقة نيوتن لحساب جذر المعادلة غير الخطية $f(x) = x^2 - 4x + 4$ بدءا من x_0= ؟

(1) Find a recurrence relation for Newton's method to find values of

$$(i)\ \sqrt[n]{a},\quad (ii)\ \frac{1}{\sqrt{a}}$$

Then find the value of each of them when a=5, n=2 to within 3 digits of accuracy.

(i) نضع $x = \sqrt[N]{a} \Leftarrow x^N = a \Leftarrow f(x) = x^N - a$

$$x_{i+1} = x_i - \frac{x_i^N - a}{N x_i^{N-1}} = \frac{1}{N}\left(x_i N - \frac{x_i^N - a}{x_i^{N-1}}\right)$$

$$= \frac{1}{N}\left(x_i N - x_i + \frac{a}{x_i^{N-1}}\right)$$

فإذا أخذنا $a = 25$ ، $N = 3$ فإن الجدول التالي سوف يكون هو الحل أيضاً

n	x_n	$f(x_n)$	$x_n - x_{n-1}$
0	2	-17.000000000	1.416666667
1	3.416666667	14.884837963	-0.425028092

2	2.991638575	1.774870159	-0.066103900
3	2.925534675	0.038929063	-0.001516150
4	2.924018525	0.000020171	

(2) rerun your program for Newton's method for Example 1 with interchange of the initial point to x_0=1.5, 5.0, 100. What is your remarks.

(3) Use Newton's method to find root for the nonlinear equation $x = \cos x$

(4) من خلال دراسة تقدير الخطأ لطريقة نيوتن أوجد معدل التقارب عند حساب جذر المعادلة الغير خطية $f(x) = x^6 - x - 1$ ، و من ثم حدد الخطأ المتوقع بعد 4 تتابعات. قارن ذلك مع النتائج العددية لمثال (1).

(5) استخدم إكسل لبرمجة طريقة نيوتن لحساب جذر المعادلة الغير خطية

$f(x) = x^6 - x - 1$

و ذلك بدءا من $x_0 = 1.5$. قارن النتائج مع مثال (1).

(6) لكي نطبق طريقة نيوتن لإيجاد أول جذر موجب للمعادلة

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

أوجد أكبر فترة $[a,b]$ التي تتقارب فيها طريقة نيوتن لحل هذه المسألة.

تسهيل:إرسم الدالة المعطاة لتعرف أول جذر موجب ثم حدد الفترة التي لا يدخل مع هذا الجذر أية جذور أخرى

(7) استخدم طريقة القاطع لحساب جذر المعادلة الغير خطية $x = \cos x$

(8) استخدم طريقتي نيوتن و القاطع لحساب جذر المعادلة الغير خطية $e^x = \sin(\frac{\pi x}{3})$.

[1](a) Derive an error formula for Newton method to solve a nonlinear equation then prove that Newton method is sensitive for the initial guess.
Answer : In the text

[1] (b) consider computing $\sqrt{a}$ using Newton's method. Find the interval D that must contains x_0 such that Newton iteration is convergent
Answer : In the text

*[1] (c) consider computing $\sqrt[m]{a}$ using Newton's method. Find the interval D that must contains x_0 such that Newton iteration is convergent

**[2] (a) use Newton's method to find a recurrence relation for the computation of $\ln(u)$, with $u > 0$.

**[2] (b) Derive an error formula for Newton method then derive a formula for the relative error in the n+1 stage in terms of the previous stage for problem [2] (a).

**[2] (c) find the interval D that must contains X_0 such that Newton iteration is convergent in evaluating $\ln(u)$, $u = e^2$.

Exercise

[1] (a) On most computers. the computation of $\sqrt{a}$ is based on Newton's method. Set up the Newton iteration for solving $x^2-a=0$, and show that it can be written in the form

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right), \qquad n \geq 0$$

(b) Write without proof the error formula for Newton method then derive a formula for the relative error in the n+1 stage in terms of the previous stage.

(c) find the interval D that must contains x_0 such that Newton iteration is convergent

(d) For x_0 near $\sqrt{a}$, the last formula becomes

$$\text{Rel}(x_{n+1}) \approx -\frac{1}{2}[\text{Rel}(x_n)]^2, \qquad n \geq 0$$

Assuming Rel(x0) = 0.1, use this formula to estimate the relative error in x_1, x_2, x_3, and x_4.

Answer:

a) $x = \sqrt{a}$

let $f(x) = x^2 - a = 0$ so $f'(x) = 2x$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \ \text{n} = 0, 1, 2, \ldots$$

$$= x_n - \frac{x_n^2 - a}{2x_n} = \frac{1}{2}\left(x_n + \frac{a}{x_n} \right)$$

b) The error formula for Newton method

$$\alpha - x_{n+1} \approx -\frac{f''(\alpha)}{2f'(\alpha)}(\alpha - x_n)^2 \qquad (1)$$

For x_n close to α, and therefore c_n also close to α,

We can also use this to explore the 'interval of convergence' of Newton's method. Write the above as

$$M = -\frac{f''(\alpha)}{2f'(\alpha)} \qquad \alpha - x_{n+1} \approx M(\alpha - x_n)^2, \qquad (2)$$

Multiply both sides by M to get

$$M(\alpha - x_{n+1}) \approx M(\alpha - x_n)^2$$

$$M(\alpha - x_{n+1}) \approx \left([M(\alpha - x_n)]^2\right)^2,\ n \geq 0$$

$$= [M(\alpha - x_0)]^{2^n}$$

Then we want the quantity $[M(\alpha - x_0)]^{2^n}$ to decrease; and this suggests choosing x_0 so that

$$|M(\alpha - x_0)| < 1 \rightarrow |\alpha - x_0| < \frac{1}{M} = \left|\frac{2f'(\alpha)}{f''(\alpha)}\right| \qquad (3)$$

If |M| is very large, then we may need to have a very good initial guess in order to have the iterates x_0 converge to α.

using (1) ; $\alpha - x_{n+1} = -\dfrac{f''(x_n)}{2f'(x_n)}(\alpha - x_n)^2$

evaluating $M = -\dfrac{f''(c_n)}{2f'(x_n)} = -\dfrac{2}{4x_n} = -\dfrac{1}{2x_n}$ (4)

with $\alpha = \sqrt{a}$; we have

$$\sqrt{a} - x_{n+1} = -\frac{1}{2x_n}(\sqrt{a} - x_n)^2 \qquad (5)$$

for relative error, we have

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

$$= -\frac{\sqrt{a}}{2x_n}\left(\frac{\sqrt{a} - x_n}{\sqrt{a}}\right)^2$$

i.e. $Re(x_{n+1}) = -\dfrac{\sqrt{a}}{2x_n}(Re(x_n))^2$

(c)

using (3) $|\alpha - x_0| < \dfrac{1}{|M|}$

using (4) $= 2|x_n|$

with $x_n \to \alpha$ $\simeq 2\alpha = 2\sqrt{a}$

Hence the inial guess must be in the interval D for Newton method to be convergent:

$$|\alpha - x_0| < 2\sqrt{a}$$

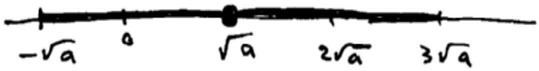
$$-2\sqrt{a} < \alpha - x_0 < 2\sqrt{a}$$

$\alpha = \sqrt{a}$

$$-2\sqrt{a} - \sqrt{a} < -x_0 < 2\sqrt{a} - \sqrt{a}$$

$$-3\sqrt{a} < -x_0 < \sqrt{a}$$

$$-\sqrt{a} < x_0 < 3\sqrt{a}$$



*[2] (d) consider computing a solution for the nonlinear equation

$$f(x) \equiv b - 1/x = 0$$

using Newton's method. Find the interval D that must contains x_0 such that Newton iteration is convergent

[3] state a convergence **Corollary** for the fixed point method. Then apply it to check the convergence of the following iteration formula:

1. $x_{n+1} = 5 + x_n - x_n^2$
2. $x_{n+1} = 5/x_n$
3. $x_{n+1} = 1 + x_n - \frac{1}{5}x_n^2$
4. $x_{n+1} = \frac{1}{2}\left(x_n + 5/x_n\right)$

[4] write with proof the Aitken's extrapolation formula with an algorithm to program it, derive from it Aitken's error estimation formula

Chapter 4

Interpolation

4.1 Principle of Linear Interpolation

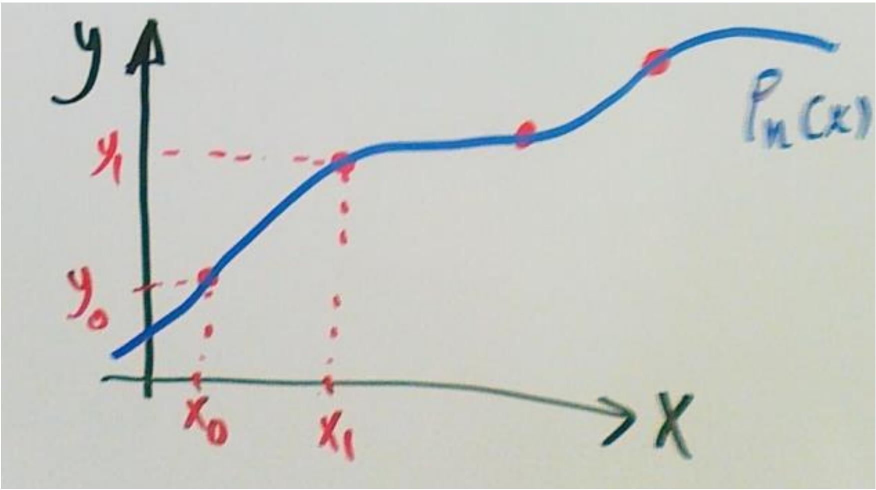
In many applications, the function may be given as a table. For example:

x	0.0	0.2	0.4	0.6	0.8	1.0
$f(x)$	0.0	0.04	0.16	0.36	0.64	1.0

It is required from the given values of the function { $f(x_i)$,i=1,2,3,4 } to obtain a value of $f(x)$ for certain value of x , as an example f (3.55) . or the value of x for certain value of $f(x)$, as an example ; what is x when $f(x)$= 0.11.

This problem is called an interpolation problem .

To solve it , we define polynomial $p_n(x)$ of degree n to be an approximation of $f(x)$. The polynomial curve must pass on the given points



1-Lagrange Interpolation

a- Linear Interpolation

Consider (x_o, y_o), (x_1, y_1), then the line pass through them is

$$y \cong p_1(x) = \frac{x - x_1}{x_0 - x_1} y_0 + \frac{x - x_o}{x_1 - x_0} y_1$$

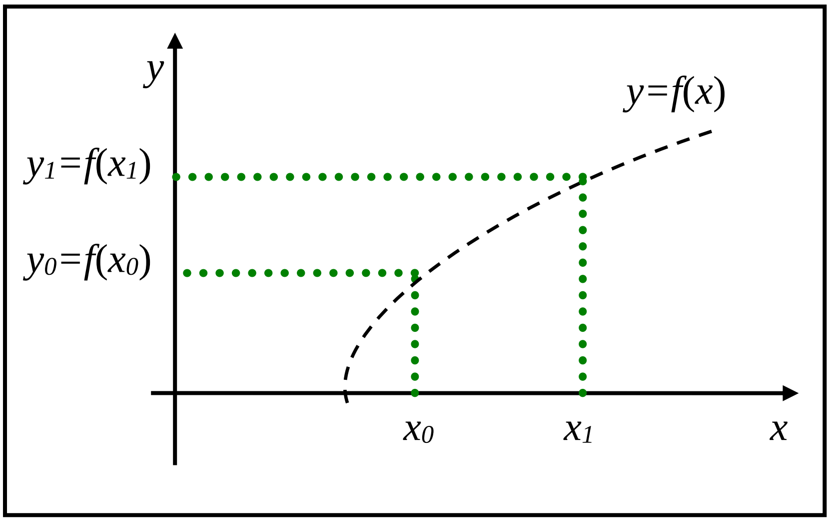
We define

$$L_0(x)=\frac{x-x_1}{x_0-x_1}, \qquad L_1(x)=\frac{x-x_o}{x_1-x_0} \qquad (1)$$

Which has the properties

$$L_0(x_0)=1,\ L_0(x_1)=0, \qquad L_1(x_0)=0,\ L_1(x_1)=1 \qquad (2)$$

These function $L_j(x)=0,\ j=0,1,$ are called Lagrange functions or multiplies.



So

$$p_1(x)=L_0(x)y_0+L_1(x)y_1 \qquad (3)$$

And then

$$p_1(x_0)=L_0(x_0)y_0+L_1(x_0)y_1=1(y_0)+0(y_1)=y_0$$
$$p_1(x_1)=L_0(x_1)y_0+L_1(x_1)y_1=0(y_0)+1(y_1)=y_1$$

Or

$$p_1(x_i)=y_i,\ i=0,1.$$

Example

Find the linear polynomial that passes through the three points (1, 1), (4, 2). Then find the value of the function at x=3

Answer: Let $(x_o,y_o)=(4,2),\ (x_1,y_1)=(1,1)$ then

$$L_0(x)=\frac{x-x_1}{x_0-x_1}=\frac{x-1}{4-1}=\tfrac{1}{3}(x-1)$$

$$L_1(x)=\frac{x-x_o}{x_1-x_0}=\frac{x-4}{1-4}=-\tfrac{1}{3}(x-4)$$

Since $f(x_0)=y_0=2,\ f(x_1)=y_1=1$ then

$$p_1(x)=L_0(x)f(x_0)+L_1(x)f(x_1)$$
$$=\tfrac{1}{3}(x-1)(2)+(-\tfrac{1}{3})(x-4)(1)$$
$$=\tfrac{2}{3}(x-1)-\tfrac{1}{3}(x-4)\ =\tfrac{2}{3}x-\tfrac{2}{3}-\tfrac{1}{3}x+\tfrac{4}{3}$$
$$=\tfrac{1}{3}x+\tfrac{2}{3}$$

at x=3

$p_1(3) = \frac{1}{3}(3) + \frac{2}{3} = \frac{5}{3}$

b- quadratic Interpolation

Let $f(x)$ is defined at three points:

$$(x_o, y_o), (x_1, y_1), (x_2, y_2) \ ,$$

Then

$$p_2(x) = L_0(x)y_0 + L_1(x)y_1 + L_2(x)y_2 \qquad (4)$$

where

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)},$$

$$L_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}$$

$$L_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} \qquad (5)$$

with

$$L_i(x_i) = 1, \quad L_i(x_j) = 0, \ i \neq j, \quad i, j = 0,1,2 \qquad (6)$$

Example

Find the polynomial of degree 2 that passes through the three points (2, 7), (1, -1) and (0,-1). Then find the value of the function at *x*=3

Answer: Let

i	0	1	2
x_i	0	1	2
y_i	0	-1	7

$$\therefore L_0(x) = \frac{(x-1)(x-2)}{(0-1)(0-2)} = \frac{1}{2}(x-1)(x-2)$$

$$L_1(x) = \frac{(x-0)(x-2)}{(1-0)(1-2)} = -\frac{x}{1}(x-2) = -x(x-2)$$

$$L_2(x) = \frac{(x-0)(x-1)}{(2-0)(2-1)} = \frac{x}{2}(x-1)$$

Then

$$P_2(x) = L_0(x)y_0 + L_1(x)y_1 + L_2(x)y_2 = \frac{-1}{2}(x-1)(x-2) + x(x-2) + \frac{7}{2}x(x-1)$$

$$f(3) \cong P_2(x) = \frac{-1}{2}(3-1)(3-2) + 3(3-2) + \frac{7}{2}(3)(3-1) = 23$$

c- Higher degree interpolation(Lagrange interpolation)

We consider the following data is given.

x	x_0	x_1	x_2	$\cdots$	x_n
y	y_0	y_1	y_2	$\cdots$	y_n

Then Lagrange polynomial interpolation is

$$f(x) \simeq P_n(x) = L_0(x)y_0 + L_1(x)y_1 + \cdots + L_n(x)y_n$$

$$p_n(x) = \sum_{j=0}^{n} L_j(x)y_i \qquad (1)$$

Where

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

$$= \prod_{\substack{k=0 \\ k\neq i}}^{n} \frac{(x-x_k)}{(x_i-x_k)}$$

We have

$$L_i(x_j) = \delta_{ij} = \begin{cases} 1 & ,i = j \\ 0 & ,i \neq j \end{cases}$$

Home work :
Use the following data obtain a suitable interpolation polynomial and then find the value of the function at $= 5$.

i	0	1	2	3	4
x	**1**	**2**	**3**	**4**	**6**
y	**1**	**8**	**27**	**64**	**216**

Answer,
we have n=4 so

$$L_0(x) = \frac{(x-x_1)(x-x_2)(x-x_3)(x-x_4)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)(x_0-x_4)}$$

$$= \frac{(x-2)(x-3)(x-4)(x-6)}{(1-2)(1-3)(1-4)(1-6)}$$

$$= \frac{-1}{30}(x-2)(x-3)(x-4)(x-6)$$

$$L_1(x) = \frac{(x-x_0)(x-x_2)(x-x_3)(x-x_4)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)(x_1-x_4)} = \frac{-1}{8}(x-1)(x-3)(x-4)(x-6)$$

$$L_2(x) = \frac{(x-x_0)(x-x_1)(x-x_3)(x-x_4)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)(x_2-x_4)} = \frac{1}{6}(x-1)(x-2)(x-4)(x-6)$$

$$L_3(x) = \frac{(x-x_0)(x-x_1)(x-x_2)(x-x_4)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)(x_3-x_4)} = \frac{-1}{12}(x-1)(x-2)(x-3)(x-6)$$

$$L_4(x) = \frac{(x-x_0)(x-x_1)(x-x_2)(x-x_3)}{(x_4-x_0)(x_4-x_1)(x_4-x_2)(x_4-x_3)} = \frac{1}{120}(x-1)(x-2)(x-3)(x-4)$$

$$\begin{aligned} P_4(x) = L_0(x)y_0 + L_1(x)y_1 + L_2(x)y_2 + L_3(x)y_3 + L_4(x)y_4 &= \frac{-1}{30}(x-2)(x-3)(x-4)(x-6)(1) \\ &+ \frac{-1}{8}(x-1)(x-3)(x-4)(x-6)(8) \\ &+ \frac{1}{6}(x-1)(x-2)(x-4)(x-6)(27) \\ &+ \frac{-1}{12}(x-1)(x-2)(x-3)(x-6)(64) \\ &+ \frac{1}{120}(x-1)(x-2)(x-3)(x-4)(216) \end{aligned}$$

$$\begin{aligned} P_4(5) = \frac{-1}{30}&(5-2)(5-3)(5-4)(5-6) \\ &-(5-1)(5-3)(5-4)(5-6) \\ &+\frac{27}{6}(5-1)(5-2)(5-4)(5-6) \\ &-\frac{64}{12}(5-1)(5-2)(5-3)(5-6) \\ &+\frac{216}{120}(5-1)(5-2)(5-3)(5-4) = 125 \end{aligned}$$

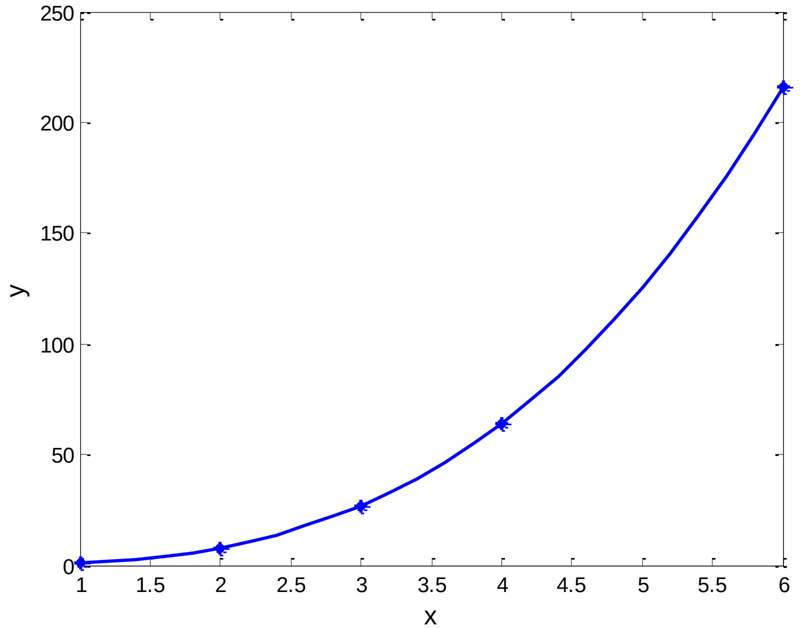
The Matlab program is as follows

clear x = [1 2 3 4 6]; y = [1 8 27 64 216];	v =

Interpolation

l = lag_intrp(x,y);
xx = [1: 0.2 : 6]; yy = polyval(l,xx);
clf, plot(xx,yy,'b', x,y,'*','LineWidth',2)
ylabel('y','FontSize',12)
xlabel('x','FontSize',12)
xx0=5;
yy0 = polyval(l,xx0);
v=[xx0 yy0]
	5.0000 125.0000

```
function [l,L] = lag_intrp(x,y)
N = length(x)-1;
l = 0;
for m = 1:N + 1
P = 1;
for k = 1:N + 1
if k ~= m, P = conv(P,[1 -x(k)])/(x(m)-x(k)); end
end
L(m,:) = P;
l = l + y(m)*P;
end
```



The function w=conv(u,v) used in this program is the product of multiplication(element by element) for two matrices u,v, with dimensions

m = length(u) and n = length(v)

Then we have

w(n) = u(1)*v(n)+u(2)*v(n-1)+ ... +u(n)*v(1)

Theory of interpolation

and we seek a polynomial $P(x)$ of lowest possible degree for which

$$p_n(x_i) = y_i,\ i = 0,1,\ldots,n$$

Such a polynomial is said to **interpolate** the data.

Does such a polynomial exist, and if so, what is its degree? Is it unique? What is a formula for producing $P(x)$ from the given data?
By writing

$$p(x) = a_0 + a_1x + \cdots + a_mx^m \quad (1)$$

for a general polynomial of degree m, we see there are $m + 1$ independent parameters $a_0, a_1, \cdots, a_m$. Since (1) imposes $n + 1$ conditions on $P(x)$, it is reasonable to first consider the case when $m = n$. Then we want to find $a_0, a_1, \cdots, a_n$ such that

$$a_0 + a_1x_0 + a_2x_0^2 + \cdots + a_nx_0^n = y_0$$

$$\vdots$$

$$a_0 + a_1x_n + a_2x_n^2 + \cdots + a_nx_n^n = y_n \quad (2)$$

This is a system of $n + 1$ linear equations in $n + 1$ unknowns, and solving it is completely equivalent to solving the polynomial interpolation problem. In vector and matrix notation, the system is

$$Xa=Y \quad (3)$$

with

$$X = [x_i^j] \qquad i, j = 0, 1, \ldots, n$$

$$a = [a_0, a_1, \ldots, a_n]^T \qquad y = [y_0, \ldots, y_n]^T$$

The matrix X is called a *Vandermonde matrix.*

Here is the theorem that governs this polynomials.

Theorem 3.1

Given $n + 1$ distinct points $x_0, \cdots, x_n$ and $n + 1$ ordinates $y_0, \cdots, y_n$, there is a polynomial $p(x)$ of degree $\leq n$ that interpolates y_i at x_i i = 0, 1, ... , n. This polynomial p(x) is unique among the set of all polynomials of degree at most n.

Given $n+1$ distinct points $x_0, \ldots, x_n$ and $n+1$ ordinates $y_0, \ldots, y_n$, there is a polynomial $p(x)$ of degree $\leq n$ that interpolates y_i at x_i, $i = 0, 1, \ldots, n$. This polynomial $p(x)$ is unique among the set of all polynomials of degree at most n.

Proof

It can be shown that for the matrix X in (3),

$$\det(X) = \prod_{0 \leq j < i \leq n} (x_i - x_j) \tag{4}$$

This shows that det(X) ≠ 0, since the points x_i are distinct. Thus X is nonsingular and the system $Xa = y$ has a unique solution a. This proves the existence and uniqueness of an interpolating polynomial of degree $\leq n$.

To prove uniqueness, suppose $q(x)$ is another polynomial of degree : n that satisfies (1). Define

$$r(x) = p(x) - q(x)$$

Then degree $r(x) \leq n$, and

$$r(x_i) = p(x_i) - q(x_i) = y_i - y_i = 0, \quad i=0,1, \ldots ,n$$

Since $r(x)$ has $n + 1$ zeros, we must have r(x) = 0. This proves $p(x) = q(x)$, completing the proof. ♦

==================================

If a function $f(x)$ is given, then we can form an approximation to it using the interpolating polynomial

$$P_n(x) = \sum_{i=0}^{n} L_i(x) y_i \tag{7}$$

This interpolates $f(x)$ at $x_0, \ldots, x_n$. For example, we later consider $f(x) = \log_{10} x$ with linear interpolation. The basic result used in analyzing the error of interpolation is the following theorem. As a notation, £{a, *b,* c, ... } denotes the smallest interval containing all of the real numbers *a, b, c, ...* .

This interpolates $f(x)$ at $x_0, \ldots, x_n$. For example, we later consider $f(x) = \log_{10} x$ with linear interpolation. The basic result used in analyzing the error of interpolation is the following theorem. As a notation, $\mathscr{H}\{a, b, c, \ldots\}$ denotes the smallest interval containing all of the real numbers $a, b, c, \ldots$.

***Theorem** 2*

Let $x_0, x_1, \ldots, x_n$ be distinct real numbers, and let f be a given real valued function with $n + 1$ continuous derivatives on the interval I_t = £{t, $x_0, \ldots, x_n$}, with t some given real number.

Let $x_0, x_1, \ldots, x_n$ be distinct real numbers, and let f be a given real valued function with $n+1$ continuous derivatives on the interval $I_t = \mathscr{H}\{t, x_0, \ldots, x_n\}$, with t some given real number.

Then. there exists $\xi \in I_t$, with

$$f(t) - \sum_{j=0}^{n} f(x_j) l_j(t) = \frac{(t - x_0) \cdots (t - x_n)}{(n+1)!} f^{(n+1)}(\xi) \tag{8}$$

The error of Lagrange interpolation is

$$R(x) = f(x) - P_n(x) = \frac{f^{(n+1)}(c)}{(n+1)!} \prod_{k=0}^{n} (x - x_k)$$

where c is somewhere between $x_0, x_1, \ldots, x_n$ and x

Proof:

Note that the result is trivially true if *t* is any node point, since then both sides of (8) are zero. Assume *t* does not equal any node point. Then define

$$E(x) = f(x) - p_n(x) \qquad p_n(x) = \sum_{j=0}^{n} f(x_j) l_j(x)$$

$$G(x) = E(x) - \frac{\Psi(x)}{\Psi(t)} E(t) \qquad \text{for all} \quad x \in I_t$$

(9)

With

$$\Psi(x) = (x - x_0) \cdots (x - x_n)$$

The function G(x) is n + 1 times continuously differentiable on the interval I, as are E(x) and Ψ (x). Also,

The function $G(x)$ is $n + 1$ times continuously differentiable on the interval I_t, as are $E(x)$ and $\Psi(x)$. Also,

$$G(x_i) = E(x_i) - \frac{\Psi(x_i)}{\Psi(t)} E(t) = 0 \qquad i = 0, 1, \ldots, n$$

$$G(t) = E(t) - E(t) = 0$$

Thus G has $n + 2$ distinct zeros in I_t. Using the mean value theorem, G' has $n + 1$ distinct zeros. Inductively, $G^{(j)}(x)$ has $n + 2 - j$ zeros in I_t, for $j = 0, 1, \ldots, n + 1$. Let ξ be a zero of $G^{(n+1)}(x)$,

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

Since

$$E^{(n+1)}(x) = f^{(n+1)}(x)$$

$$\Psi^{(n+1)}(x) = (n + 1)!$$

we obtain

$$G^{(n+1)}(x) = f^{(n+1)}(x) - \frac{(n+1)!}{\Psi(t)} E(t)$$

Substituting $x = \xi$ and solving for $E(t)$,

$$E(t) = \frac{\Psi(t)}{(n+1)!} \cdot f^{(n+1)}(\xi)$$

the desired result.

This may seem a "tricky" derivation, but it is a commonly used technique for obtaining some error formulas.

Example For $n = 1$, using x in place *of t*,

$$f(x) - \frac{(x_1 - x)f(x_0) + (x - x_0)f(x_1)}{x_1 - x_0} = \frac{(x - x_0)(x - x_1)}{2} f''(\xi_x) \qquad (10)$$

for some $\xi_x \in \pounds\{x_0, x_1, x\}$. The subscript x on ~x shows explicitly that ξ depends on x; usually we omit the subscript, for convenience.

We now apply the $n = 1$ case to the common high school technique of linear interpolation in a logarithm table. Let

$$f(x) = \log_{10} x$$

Then $f''(x) = -\log_{10} e/x^2$, $\log_{10} e \doteq 0.434$. In a table, we generally would have $x_0 < x < x_1$. Then

$$E(x) = \frac{(x - x_0)(x_1 - x)}{2} \cdot \frac{\log_{10} e}{\xi^2} \qquad x_0 \le \xi \le x_1$$

This gives the upper and lower bounds

$$\frac{\log_{10} e}{x_1^2} \cdot \frac{(x - x_0)(x_1 - x)}{2} \le E(x) \le \frac{\log_{10} e}{x_0^2} \cdot \frac{(x - x_0)(x_1 - x)}{2}$$

This shows that the error function E(x) looks very much like a quadratic polynomial, especially if the distance h = x1 - x 0 is reasonably small. For a uniform bound on [x 0 , x1]

$$\underset{x_0 \le x \le x_1}{\text{Max}} (x_1 - x)(x - x_0) = \frac{h^2}{4}$$

$$|\log_{10} x - p_1(x)| \le \frac{h^2}{8} \frac{.434}{x_0^2} = \frac{.0542 h^2}{x_0^2} \le .0542 h^2 \qquad (11)$$

for x 0 ≥1, as is usual in a logarithm table. Note that the interpolation error in a standard table is much less for x near 10 than near 1. Also, the maximum error is near the midpoint of [x 0 , x1].

For a four-place table, h = .01,

$$|\log_{10} x - p_1(x)| \le 5.42 \times 10^{-6} \qquad 1 \le x_0 < x_1 \le 10$$

Since the entries in the table are given to four digits (e.g., log10 2 = .3010), this result is sufficiently accurate. Why do we need a more accurate five-place table if the preceding is so accurate? Because we have neglected to include the effects of the rounding errors present in the table entries. For example, with log 10 2= .3010.

$$|\log_{10} 2 - .3010| \leq .00005$$

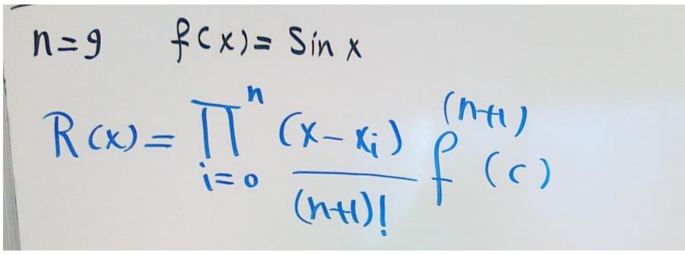
and this will dominate the interpolation error if x0 or x1 = 2.

Example 3

If the function $f(x) = \sin(\pi x)$ is approximated by a polynomial of degree 9 that interpolates $f(x)$ at ten points in the interval [0,1], estimate the error on this interval

Solution

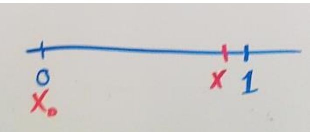
To answer this question, we use R(x) in the preceding theorem. Obviously,



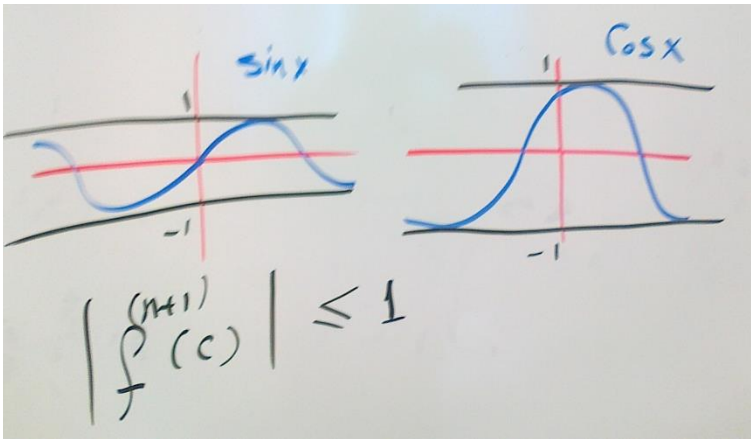
Estimate (not evaluate or calculate)

So it is enough to define a bound for R(x) ;

$$|x - x_i| \leq 1 \qquad i = 1, \ldots \ldots, n$$



So $\prod_{i=0}^{9} |x - x_i| \leqq 1$



and $|f^{(10)}(\zeta_x)| \leq 1$.

So, for all *x* in [0,1],

$$|\sin x - p(x)| \leqq \frac{1}{10!} < 2.8 \times 10^{-7}$$

مثال (4):

(أ) احسب الخطأ المتوقع عند استكمال للدالة $f(x) = \sin(\pi x)$ عند النقطة $x = \frac{1}{3}$ باستخدام كثيرة حدود لاجرانج باختيار النقاط. $x_0 = \frac{1}{3}, x_1 = \frac{1}{6}, x_2 = \frac{1}{2}$

(ب) احسب خطأ الاستكمال وذلك عند استكمال للدالة $\boldsymbol{f(x) = sin(\pi x)}$ عند النقطة $\boldsymbol{x = \frac{1}{3}}$ باستخدام كثيرة حدود لاجرانج باختيار النقاط $\boldsymbol{x_0 = \frac{1}{3}, x_1 = \frac{1}{6}, x_2 = \frac{1}{2}}$.

الحل :

أ) بالتعويض في (3)

$$R_2(x) = \frac{(x-x_0)(x-x_1)(x-x_2)}{(2+1)!} f^{(2+1)}(\xi)$$

$$= \frac{(x-0)(x-\frac{1}{2})(x-\frac{1}{6})}{3!} f^{(3)}(\xi)$$

وحيث إنَّ:

$$f^{(1)}(x) = \pi\cos(\pi x),\ f^{(2)}(x) = -\pi^2\sin(\pi x),\ f^{(3)}(x) = -\pi^3\cos(\pi x),$$

$$R_2(x) = \frac{-x(x-\frac{1}{2})(x-\frac{1}{6})}{6}\pi^3\cos(\pi\xi)$$

ولكن $|\cos(\pi\xi)| \leq 1$ و أن $x = \frac{1}{3}$ وبذلك يكون

$$\left|R_2(\tfrac{1}{3})\right| \leq \left|\frac{-\frac{1}{3}(\frac{1}{3}-\frac{1}{2})(\frac{1}{3}-\frac{1}{6})}{6}\right|\pi^3 = 0.04791$$

ب) باستخدام (1) -(2) نجد أن:

$$y_0 = \sin 0 = 0, y_1 = \sin\tfrac{\pi}{6} = \tfrac{1}{2},\ \ y_2 = \sin\tfrac{\pi}{2} = 1$$

$$p_2(x) = L_0(x)y_0 + L_1(x)y_1 + L_2(x)y_2$$

$$L_0(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{(x-\frac{1}{6})(x-\frac{1}{2})}{\left(-\frac{1}{6}\right)\left(-\frac{1}{2}\right)}$$

$$L_1(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = \frac{x(x-\frac{1}{2})}{\frac{1}{6}\left(\frac{1}{6}-\frac{1}{2}\right)}$$

$$L_2(x)=\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}=\frac{x(x-\frac{1}{6})}{\frac{1}{2}\left(\frac{1}{2}-\frac{1}{6}\right)}$$

$$p_2(x)=L_0(x)y_0+L_1(x)y_1+L_2(x)y_2$$

$$=\frac{(x-\frac{1}{6})(x-\frac{1}{2})}{\left(-\frac{1}{6}\right)\left(-\frac{1}{2}\right)}*0+\frac{x(x-\frac{1}{2})}{\frac{1}{6}\left(\frac{1}{6}-\frac{1}{2}\right)}*\frac{1}{2}+\frac{x(x-\frac{1}{6})}{\frac{1}{2}\left(\frac{1}{2}-\frac{1}{6}\right)}*1$$

$$P_2(x)=-9x(x-\tfrac{1}{2})+6x(x-\tfrac{1}{6})$$

$$\left|f(\tfrac{1}{3})-P_2(\tfrac{1}{3})\right|=\left|\sin(\tfrac{\pi}{3})-P_2(\tfrac{1}{3})\right|$$

$$=0.866234-0.83333=0.032903$$

Rounding error analysis for linear interpolation

Let

$$f(x_0)=f_0+\epsilon_0 \qquad f(x_1)=f_1+\epsilon_1$$

with f_0 and f_1 the table entries and ϵ_0, ϵ_1 the rounding errors. We will assume

$$|\epsilon_0|, |\epsilon_1| \le \epsilon$$

for a known ϵ. In the case of the four-place logarithm table, $\epsilon = .00005$.

We want to bound

$$\mathscr{E}(x)=f(x)-\frac{(x_1-x)f_0+(x-x_0)f_1}{x_1-x_0} \qquad x_0 \le x \le x_1$$

(12)

Using $f_i=f(x_i)-\epsilon_i$,

$$\mathscr{E}(x)=f(x)-\frac{(x_1-x)f(x_0)+(x-x_0)f(x_1)}{x_1-x_0}+\frac{(x_1-x)\epsilon_0+(x-x_0)\epsilon_1}{x_1-x_0}$$

$$\equiv E(x)+R(x)$$

(13)

$$E(x)=\frac{(x-x_0)(x-x_1)}{2}f''(\xi) \qquad \xi \in [x_0, x_1]$$

The error $\mathscr{E}(x)$ is the sum of the theoretical interpolation error $E(x)$ and the function $R(x)$, which depends on ϵ_0, ϵ_1. Since $R(x)$ is a straight line, its maximum on $[x_0, x_1]$ is attained at an endpoint,

$$\max_{x_0 \le x \le x_1} |R(x)| = \operatorname{Max}\{|\epsilon_0|, |\epsilon_1|\} \le \epsilon$$

(14)

With $x_1 = x_0 + h$, $x_0 \le x \le x_1$,

$$|\mathscr{E}(x)| \le \frac{h^2}{8} \max_{x_0 \le t \le x_1} |f''(t)| + \operatorname{Max}\{|\epsilon_0|, |\epsilon_1|\} \qquad (15)$$

Example

For the earlier logarithm example using a four-place table,

$$|\mathscr{E}(x)| \le 5.42 \times 10^{-6} + 5 \times 10^{-5} \doteq 5.5 \times 10^{-5}$$

For a five-place table, $h = .001$, $\epsilon = .000005$, and

$$|\mathscr{E}(x)| \le 5.42 \times 10^{-8} + 5 \times 10^{-6} \doteq 5.05 \times 10^{-6} \qquad x_0 \le x \le x_1$$

The rounding error is the only significant error in using linear interpolation in a five-place logarithm table. In fact, it would seem worthwhile to increase the five-place table to a six-place table, without changing the mesh size *h*. Then we would have a maximum error for $\mathscr{E}(x)$ of 5.5×10^{-7}, without any significant increase in computation. These arguments on rounding error generalize to higher degree polynomial interpolation, although the result on Max |R(x)| is slightly more complicated .

None of the results of this section take into account new rounding errors that occur in the evaluation of $p_n(x)$. These are minimized by results given in the next section.

Convergence of Interpolating Polynomials Theorem

If f is *a continuous function on [a, b], then there* is *a system of prescribed system of nodes*

$$a \leqq x_0^{(n)} < x_1^{(n)} < \cdots < x_n^{(n)} \leqq b \qquad (n \geqq 0)$$

such that the polynomials p_n is an interpolation to f at these nodes satisfy $\lim_{n\to\infty} \|f - p_n\|_\infty = 0.$

The Weierstrass Approximation Theorem

If f is *continuous on* [*a*,*b*] *and if* ε > 0, *then there* is *a polynomial p satisfying* $|f(x) - p(x)| \leqq \varepsilon$ *on the interval*[*a*,*b*].

PROBLEM SET

1- Determine whether the algorithm

$x = a_n b_n$
for i = 1,2,. . ., n do
$x = (x + a_{n-i})b,$
end

computes

$$x = \sum_{i=0}^{n} a_i \prod_{j=0}^{i} b_j$$

2- What is the final value of v in the algorithm shown?

$v = C_{i-1}$
for j = i, i +1, . . , n **do**
$v = vx + C_j$
end

What is the number of additions and subtractions involved in this algorithm?

3- Write an efficient algorithm for evaluating

$$u = \sum_{i=1}^{n} \prod_{j=1}^{i} d_j$$

تمارين (1-5)

(1) الجدول التالي لقيم الدالة $y = f(x)$. احسب $f(323.5)$

x	321.0	322.8	324.2	325.0
y	2.50651	2.50893	2.51081	2.51188

(2) إذا أعطيت قيم الدالة عند 4 نقاط كما في الجدول التالي

x	$f(x)$
3.35	0.298507
3.40	0.294118
3.50	0.285714
3.60	0.277778

(أ) استخدم صيغة لاجرانج الاستكمالية الخطية لحساب $y = f(3.44)$

(ب) استخدم صيغة لاجرانج الاستكمالية التربيعية لحساب $y = f(3.44)$

(ج) استخدم صيغة لاجرانج الاستكمالية التكعيبية لحساب $y = f(3.44)$

وإذا علمت أن الدالة المستكملة هي $y = f(x) = 1/x$ فاحسب الخطأ المطلق لهذا التقريب.

(3) كون للدالة $y = \sin \pi x$ كثيرة حدود لاجرانج باختيار

$x_o = 0 \qquad , x_1 = \frac{1}{6} \qquad , x_1 = \frac{1}{2}$

(4) أوجد صيغه لاجرانج للاستكمال من الدرجة الثانية من المعلومات الآتية :

x	-2	0	1
$f(x)$	10.75	-1.65	1.45

ثم أوجد $f\ (0.5)$.

(5) أوجد قيمة تقريبية لـ $y = e^{0.376}$ باستخدام صيغة لاجرانج إذا علمت أن:

x	y
0.3	1.35
0.32	1.377
0.34	1.405
0.36	1.433
0.38	1.462
0.4	1.492

4.2 Finite differences interpolation

Let $y = f(x)$ is a given function. We denote $\Delta x = h$ for the change in the independent variable. Then the formula

$$\Delta y = \Delta f(x) = f(x+\Delta x) - f(x) \qquad (1)$$

is called ***finite differences*** from the first order. For higher order we have

$$\Delta^n y = \Delta(\Delta^{n-1} y), \quad n = 2,3, \dots$$

As an example:

$$\Delta^2 y = \Delta\{f(x+\Delta x) - f(x)\}$$
$$= \{f(x+2\Delta x) - f(x+\Delta x)\} - \{f(x+\Delta x) - f(x)\}$$
$$= f(x+2\Delta x) - 2f(x+\Delta x) + f(x)$$

from (1) we have:

$$f(x+\Delta x) = f(x) + \Delta f(x)$$
$$f(x+\Delta x) = (1+\Delta) f(x) \qquad (2)$$

Appling (2) n times we obtain

$$f(x+n\Delta x) = (1+\Delta)^n f(x) \qquad (3)$$

using binomial theorem

$$f(x+n\Delta x) = \sum_{m=0}^{n} C_m^n \Delta^m f(x) \qquad (4)$$

where

$$C_m^n = \binom{n}{m} = \frac{n!}{m!(n-m)!}$$

Equation (4) express the iterative values of $f(x)$ in terms of its finite differences.

Example:

Obtain all finite differences for the function $p(x) = x^3$ with $\Delta x = 1$.

Answer:

$\Delta p(x) = (x+1)^3 - x^3 = 3x^2 + 3x + 1$,

$\Delta^2 p(x) = \left[3(x+1)^2 + 3(x+1) + 1\right]$-$[3x^2 + 3x + 1]$

$= 6x + 6$

$\Delta^3 p(x) = [6(x+1) + 6] - [6x + 6]$=6

$\Delta^n p(x) = 0, n > 3$

<u>Finite Difference Table</u>

It is convenient to put the finite differences from different orders in table. We have two types of tables:

(I) Horizontal table for finite differences

(II) Diagonal table for finite differences

The Diagonal table is as follows

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
x_0	y_o				
		Δy_o			
x_1	y_1				
		Δy_1	$\Delta^2 y_o$		
x_2	y_2		$\Delta^2 y_1$	$\Delta^3 y_o$	
		Δy_2	$\Delta^2 y_2$	$\Delta^3 y_1$	$\Delta^4 y_o$
x_3	y_3				
x_4	y_4	Δy_3			

And the Horizontal table is as follows

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
x_o	y_o	Δy_o	$\Delta^2 y_o$	...	....
x_1	y_1	Δy_1	$\Delta^2 y_1$	...	...
x_2	y_2	Δy_2	$\Delta^2 y_2$	...	...
...	...	...	...	...	....

Generalized powers

The n degree of Generalized power for number x is defined by multiplication of n factors, the first is x and the next is (x-h) where h is constant.

$$x^{[n]} = x(x-h)(x-2h)\ldots[x-(n-1)h]$$

$$x^{[0]} = 1 \quad , \quad x^{[1]} = x, \qquad x^{[2]} = x(x-h),$$

$$x^{[n]} = x^n \quad at \quad h = 0$$

Let $\Delta x = h$, then we can evaluate the finite differences for $x^{[n]}$ as follows

$$\begin{aligned}\Delta x^{[n]} &= (x+h)^{[n]} - x^{[n]} \\ &= (x+h)x((x-h)\ldots[x-(n-2)h) - x(x-h)\ldots[x-(n-1)h] \\ &= x(x-h)\ldots[x-(n-2)h]nh \\ &= nhx^{[n-1]}\end{aligned}$$

Thus,

$$\Delta x^{[n]} = nhx^{[n-1]} \qquad \textbf{(*)}$$

The second order finite difference is

$$\begin{aligned}\Delta^2 x^{[n]} = \Delta(\Delta x^{[n]}) &= \Delta(nhx^{[n-1]}) \\ &= nh(n-1)hx^{[n-2]}\end{aligned}$$

Or,

$$\begin{aligned}\Delta^2 x^{[n]} = \Delta(\Delta x^{[n]}) &= \Delta(nhx^{[n-1]}) \\ &= n(n-1)h^2 x^{[n-2]}\end{aligned}$$

Similarly,

$$\Delta^k x^{[n]} = n(n-1)(n-2)\ldots.[n-(k-1)]h^k x^{[n-k]}, k = 1,2,\ldots.,n$$

$$\Delta^k x^{[n]} = 0 \quad \forall \quad k > n$$

Newton First Formula for interpolation

Let the values $y_i = f(x_i)$ be given for $x_i = x_o + ih$, $i = 0,1,\ldots,n$ and h a given step.

We require a polynomial $p_n(x)$ of most n degree that interpolates a function f and satisfies

$$p_n(x_i) = y_i, \quad i = 0,1,\ldots,n \qquad (1)$$

Taking differences, we have

$$\Delta^m p_n(x_o) = \Delta^m y_o, \quad m = 0,1,\ldots,n$$

For Newton method we put the interpolated polynomial on the form

$$p_n(x) = a_o + a_1(x - x_o) + a_2(x - x_o)(x - x_1) + \ldots\ldots + a_n(x - x_o)(x - x_1)\ldots.(x - x_{n-1}) \qquad (2)$$

Using Generalized power

$$p_n(x) = a_o + a_1(x - x_o)^{[1]} + a_2(x - x_o)^{[2]} + + \ldots..+ a_n(x - x_o)^{[n]} \qquad (3)$$

To obtain the coefficient a_i, we set $x = x_0$ in (3),

$$p_n(x_o) = y_o = a_o$$

Taking first order differences

$$\Delta x^{[n]} = nhx^{[n-1]}$$

$$\Delta p_n(x) = a_1 h + 2a_2 h(x - x_o)^{[1]} + \ldots..+ na_n h(x - x_o)^{[n-1]} \qquad (4)$$

Set $x = x_0$ in (4)

$$\Delta p_n(x_o) = \Delta y_o = a_1 h$$

$$\therefore a_1 = \frac{\Delta y_o}{1!h}$$

Taking second order differences and Setting $x = x_o$

$$\Delta^2 p_n(x_o) = \Delta^2 y_o = 2!h^2 a_2$$

$$\therefore a_2 = \frac{\Delta^2 y_o}{2!h^2}$$

Continue in this iteration, we obtain:

$$a_i = \frac{\Delta^i y_o}{i!h^i} \qquad ,i = 0,1,2,\ldots,n$$

With $\Delta^o y = y$ ، $0! = 1$

Substituting with a_i in (3) we obtain

$$p_n(x) = y_o + \frac{\Delta y_o}{1!h}(x - x_o)^{[1]} + \frac{\Delta^2 y_o}{2!h^2}(x - x_o)^{[2]} + \ldots\ldots + \frac{\Delta^n y_o}{n!h^n}(x - x_o)^{[n]} \qquad (5)$$

Putting $q = \dfrac{x - x_o}{h}$ we obtain

$$\frac{(x - x_o)^{[i]}}{h^i} = \frac{(x - x_o)}{h} \cdot \frac{(x - x_o - h)}{h} \ldots\ldots \frac{(x - x_o - (i-1)h))}{h}$$

$$= q(q-1)\ldots(q - i + 1), \quad i = 0,1,2,\ldots.,n$$

Substituting in (4) we obtain

$$P_n(x) = y_0 + q\Delta y_0 + \frac{q(q-1)}{2!}\Delta^2 y_0 + \ldots + \frac{q(q-1)\ldots(q-n+1)}{n!}\Delta^n y_0 \qquad (6)$$

which is the Newton First Formula for interpolation.

Example

Find Newton interpolation polynomial defined by the data:

x	0	1	2	3	4	5
y	5.2	8.0	10.4	12.4	14.0	15.2

Answer:

We write first the difference table:

x	y	Δ	Δ^2	Δ^3
0	5.2			
		2.8		
1	8.0		-0.4	
		2.4		0
2	10.4		-0.4	
		2.0		0
3	12.4		-0.4	
		1.6		
4	14.0		-0.4	
		1.2		
5	15.2			

We have $x_o = 0$, $h = 1$

So using (4)

$x_o = 0$, $h = 1$

$y = 5.2 + 2.8x - \frac{0.4}{2}x(x-1)$

Or

$$y = 5.2 + 3x - 0.2x^2$$

تمارين (2-5)

(1) احسب كثيرة الحدود $y = x^3 - 8x - 4x^2 + 1$ عند قيم x الصحيحة بين $x = 1$, $x = 4$ واستنتج جدول الفروق.

(2) باعتبار الخطوة $h = 0.05$ أوجد في الفترة المغلقة $[3.5, 3.7]$ كثيرة حدود نيوتن الاستكمالية للدالة $y = e^x$ المعطاة بالجدول

x	3.50	3.55	3.60	3.65	3.70
y	33.115	34.813	36.598	38.475	40.447

(3) لدينا جدول قيم الدالة $y = \log(x)$

x	1000	1010	1020	1030	1040	1050
y	3.000000	3.004321	3.008600	3.012837	3.017033	3.021189

أوجد قيم $\log(1044)$

(4) بمعرفة جدول قيم الدالة $y = \sin x$ في الحدود من $x = 15^o$ إلى $x = 55^o$ بالخطوة $h = 5$ التالية، أوجد $Sin14^o$

(5) كون الفروق الأمامية للدالة $f(x) = \sqrt{x}$ للبيانات بالجدول التالي، ثم أوجد كثيرات حدود نيوتن للفروق الأمامية لتقريب $\sqrt{2.15}$.

x	2.0	2.1	2.2	2.3	2.4
$f(x)$	1.414214	1.449138	1.483240	1.516575	1.549193

(6) كوّن الفروق الخلفية للدالة $f(x) = \sqrt{x}$ ، ثم أوجد كثيرات حدود نيوتن للفروق الخلفية ، واستخدم كل منها لإيجاد تقريب للقيم $\sqrt{2.15}$ ، $\sqrt{2.35}$.

(7) استخدم تعريف القوى المعممة لاستنتاج صيغة نيوتن الخلفية.

(8) أوجد كثيرة حدود نيوتن الخلفية لاستكمال الدالة المعطاة بالجدول التالي:

x	4	6	8	10
y	1	3	8	20

4.3 Divided Differences

The Lagrange form of the interpolation polynomial can be used for interpolation to a function given in tabular form. But there are other forms that are much more convenient such as Divided Differences which will be developed in this section. With the Lagrange form, it is inconvenient to pass from one interpolation polynomial to another of degree one greater. Such a comparison of different degree interpolation polynomials is a useful technique in deciding what degree polynomial to use. The formulas developed in this section are for nonevenly spaced grid points {xi}. We would like to write

$$p_n(x) = p_{n-1}(x) + C(x), \qquad (*)$$

$C(x)$ *is a corection term*

Then, in general, $C(x)$ is a polynomial of degree n, since usually degree $(p_{n-1}) = n-1$ and degree $(p_n) = n$. Also we have

$$C(x_i) = p_n(x_i) - p_{n-1}(x_i) = f(x_i) - f(x_i) = 0 \qquad i = 0, \ldots, n-1$$

Thus

$$C(x) = a_n(x - x_0)\cdots(x - x_{n-1})$$

Since $p_n(x_n) = f(x_n)$, we have from (*)

$$a_n = \frac{f(x_n) - p_{n-1}(x_n)}{(x_n - x_0)\cdots(x_n - x_{n-1})}$$

For reasons derived below, this coefficient a_n is called the *nth-order Newton divided difference* of f, and it is denoted by

$$a_n \equiv f[x_0, x_1, \ldots, x_n]$$

Thus our interpolation formula becomes

$$p_n(x) = p_{n-1}(x) + (x - x_0)\cdots(x - x_{n-1}) f[x_0, \ldots, x_n] \qquad (3.2.2)$$

To obtain more information on a_n, we return to the Lagrange formula (3.1.7) for $p_n(x)$. Write

$$\Psi_n(x) = (x - x_0)\cdots(x - x_n) \qquad (3.2.3)$$

Then

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

and if x is not a node point

$$p_n(x) = \sum_{j=0}^{n} \frac{\Psi_n(x)}{(x - x_j)\Psi_n'(x_j)} \cdot f(x_j) \qquad (3.2.4)$$

Since a_n is the coefficient of x^n in $p_n(x)$, we use the Lagrange formula to obtain the coefficient of x^n. By looking at each nth-degree term in the formula (3.2.4), we obtain

$$f[x_0, x_1, \ldots, x_n] = \sum_{j=0}^{n} \frac{f(x_j)}{\Psi_n'(x_j)} \qquad (3.2.5)$$

From this formula, we obtain an important property of the divided difference. Let $(i_0, i_1, \ldots, i_n)$ be some permutation of $(0, 1, \ldots, n)$. Then easily

$$\sum_{j=0}^{n} \frac{f(x_j)}{\Psi_n'(x_j)} = \sum_{j=0}^{n} \frac{f(x_{i_j})}{\Psi_n'(x_{i_j})}$$

since the second sum is merely a rearrangement of the first one. But then

$$f[x_0, x_1, \ldots, x_n] = f\left[x_{i_0}, x_{i_1}, \ldots, x_{i_n}\right] \qquad (3.2.6)$$

for any permutation $(i_0, \ldots, i_n)$ of $(0, 1, \ldots, n)$.

Another useful formula for computing $f[x_0, \ldots, x_n]$ is

$$f[x_0, x_1, \ldots, x_n] = \frac{f[x_1, \ldots, x_n] - f[x_0, \ldots, x_{n-1}]}{x_n - x_0} \qquad (3.2.7)$$

which also explains the name of *divided difference*. This result can be proved from (3.2.5) or from the following alternative formula for $p_n(x)$:

$$p_n(x) = \frac{(x_n - x)p_{n-1}^{(0,n-1)}(x) + (x - x_0)p_{n-1}^{(1,n)}(x)}{x_n - x_0} \qquad (3.2.8)$$

with $p_{n-1}^{(0,n-1)}(x)$ the polynomial of degree $\le n-1$ interpolating $f(x)$ at $\{x_0, \ldots, x_{n-1}\}$ and $p_{n-1}^{(1,n)}(x)$ the polynomial interpolating $f(x)$ at $\{x_1, \ldots, x_n\}$. The proofs of (3.2.7) and (3.2.8) appear in Problem 13.

Returning to the formula (3.2.2), we have the formulas

Divided Differences interpolation

Consider $f(x)$ is given at $x_0, x_1, \ldots \ldots x_n$

Such that $\Delta x_i = x_{i+1} - x_i \neq 0, i = 0,1,2 \ldots$ are not equal. Then divided difference formula from first order is

$$[x_i, x_{i+1}] = \frac{y_{i+1}-y_i}{x_{i+1}-x_i} \text{ ; i= 0,1,2,...........}$$

$$i = 0 \rightarrow [x_0, x_1] = \frac{y_1 - y_0}{x_1 - x_0} \quad ,$$

$$i = 1 \rightarrow [x_1, x_2] = \frac{y_2 - y_1}{x_2 - x_1}, \ldots \ldots$$

Format for constructing divided differences of $f(x)$

x	$f(x)$	$[x_i, x_{i+1}]$	$[x_i, x_{i+1}, x_{i+2}]$
x_0	y_0	$[x_0, x_1]$	$[x_0, x_1, x_2]$
x_1	y_1	$[x_1, x_2]$	$[x_1, x_2, x_3]$
x_2	y_2	$[x_2, x_3]$	$[x_2, x_3, x_4]$
x_3	y_3	$[x_3, x_4]$	.....
x_4	y_4	$[x_4, x_5]$	......
...	...		.....

Example:

Obtain the divided difference for the following given function

x	**0**	**0.5**	**1.0**	**2.5**	**3.0**	**5.0**
y	**-5**	**-6.125**	**-9**	**-20.63**	**-23**	**-5**

Answer

$$x_0 = 0, \quad x_1 = 0.5, \quad x_2 = 1.0, \quad x_3 = 2.5, \quad x_4 = 3, \quad x_5 = 5$$

$$[x_0, x_1] = \frac{y_1 - y_0}{x_1 - x_0} = \frac{-6.125 + 5}{0.5 - 0} = -2.25$$

$$[x_1, x_2] = \frac{y_2 - y_1}{x_2 - x_1} = \frac{-9 + 6.125}{1.5 - 0.5} = -5.75$$

.

.

.

i	x	$f(x)$	$[x_i, x_{i+1}]$
0	0	-5	$[x_0, x_1]$=-2.25
1	0.5	-6.125	$[x_1, x_2]$=-5.75
2	1.0	-9	$[x_2, x_3] = -7.75$
3	2.5	-20.63	$[x_3, x_4] = -4.75$
4	3.0	-23	$[x_4, x_5] = 9$
5	5.0	-5	

================================

Divided difference is similar to differentiation

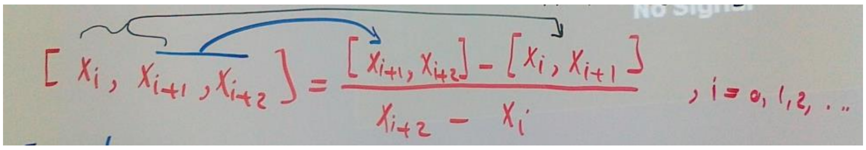
$$[x_i, x_{i+1}] = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = \frac{f(x_i + \Delta x_i) - f(x_i)}{\Delta x_i}$$

$$f' = lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

================================

The second order divided differences is

$$[x_i, x_{i+1}, x_{i+2}] = \frac{[x_{i+1}, x_{i+2}] - [x_i, x_{i+1}]}{x_{i+2} - x_i} \qquad ,i=0,1,2,……….$$



Examples:

$$[x_0, x_1, x_2] = \frac{[x_1, x_2] - [x_0, x_1]}{x_2 - x_0} = \frac{-5.75 + 2.25}{1.0 - 0} = -3.5$$

$$[x_1, x_2, x_3] = \frac{[x_2, x_3] - [x_1, x_2]}{x_3 - x_1} = \frac{-7.75 + 5.75}{2.5 - 0.5} = -1$$

.

.

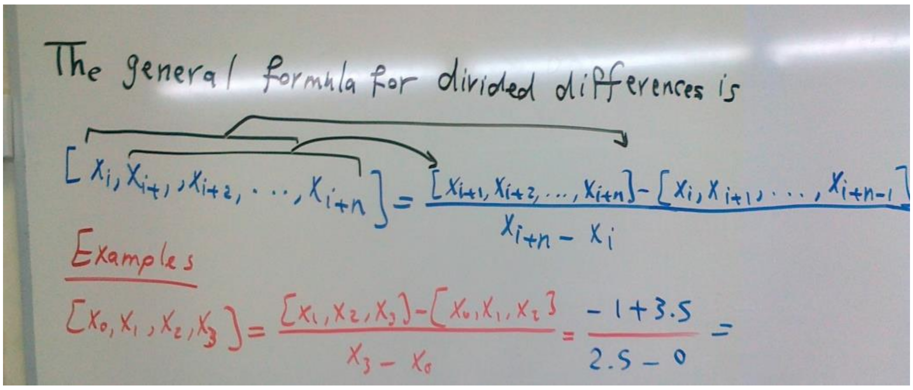
.

i	x	$f(x)$	$[x_i, x_{i+1}]$	$[x_i, x_{i+1}, x_{i+2}]$
0	0	-5	$[x_0, x_1]$=-2.25	$[x_0, x_1, x_2] = -3.5$
1	0.5	-6.125	$[x_1, x_2]$=-5.75	$[x_1, x_2, x_3] = -1$
2	1.0	-9	$[x_2, x_3] = -7.75$	$[x_2, x_3, x_4] = 1.5$
3	2.5	-20.63	$[x_3, x_4] = -4.75$	$[x_3, x_4, x_5] = 5.5$

4	3.0	-23	$[x_4, x_5] = 9$	
5	5.0	-5		

The general formula for divided differences is

$$[x_i, x_{i+1}, x_{i+2}, \dots \dots x_{i+n}] = \frac{[x_{i+1}, x_{i+2}, \dots x_{i+n}] - [x_i, x_{i+1}, x_{i+2}, \dots \dots x_{i+n-1}]}{x_{i+n} - x_i}$$



Examples:

$$[x_0, x_1, x_2, x_3] = \frac{[x_1, x_2, x_3] - [x_0, x_1, x_2]}{x_3 - x_0} = \frac{-1 + 3.5}{2.5 - 0} = 1$$

.
.
.

i	x	$f(x)$	$[x_i, x_{i+1}]$	$[x_i, x_{i+1}, x_{i+2}]$	$[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$
0	0	-5	$[x_0, x_1]$=-2.25	$[x_0, x_1, x_2] = -3.5$	$[x_0, x_1, x_2, x_3] = 1$
1	0.5	-6.125	$[x_1, x_2]$=-5.75	$[x_1, x_2, x_3] = -1$	$[x_1, x_2, x_3, x_4] = 1$
2	1.0	-9	$[x_2, x_3] = -7.75$	$[x_2, x_3, x_4] = 1.5$	$[x_2, x_3, x_4, x_5] = 1$
3	2.5	-20.63	$[x_3, x_4] = -4.75$	$[x_3, x_4, x_5] = 5.5$	
4	3.0	-23	$[x_4, x_5] = 9$		
5	5.0	-5			

Newton's polynomial for divided difference interpolation formula is

$p_0(x) = \mathrm{f}(x_0) = y_0$

$p_1(x) = \mathrm{f}(x_0) + (x - x_0)[x_0, x_1]$

$p_2(x) = \mathrm{f}(x_0) + (x - x_0)[x_0, x_1] + (x - x_0)(x - x_1)[x_0, x_1, x]$

$p_n(x) = \mathrm{f}(x_0) + (x - x_0)[x_0, x_1] + (x - x_0)(x - x_1)[x_0, x_1, x_2] + ..$

$+ (x - x_0)(x - x_1) \dots (x - x_{n-1})[x_0, x_1, \dots \dots x_n]$

This is called *Newton's divided difference formula for the interpolation polynomial.*

It is much better for computation than the Lagrange formula (although there are variants of the Lagrange formula that are more efficient than the Lagrange formula).

To construct the divided differences, use the format shown in Table 3.1. Each numerator of a difference is obtained by differencing the two adjacent entries in the column to the left of the column you are constructing.

Example

Obtain the Newton's divided difference polynomial for interpolating the given function tabulated below. Then find f(2.0).

x	**0**	**0.5**	**1.0**	**2.5**	**3.0**	**5.0**
y	**-5**	**-6.125**	**-9**	**-20.63**	**-23**	**-5**

$f(x_0)$=-5 , $[x_0, x_1] = -2.25$, $[x_0, x_1, x_2] = -3.5$, $[x_0, x_1, x_2, x_3] = 1$

$$p_n(x) = f(x_0) + (x - x_0)[x_0, x_1] + (x - x_0)(x - x_1)[x_0, x_1, x_2] + ..$$
$$+ (x - x_0)(x - x_1) \ldots (x - x_{n-1})[x_0, x_1, \ldots \ldots x_n]$$

$$p_3(x) = f(x_0) + (x - x_0)[x_0, x_1] + (x - x_0)(x - x_1)[x_0 x_1, x_2] +$$
$$+ (x - x_0)(x - x_1)(x - x_2)[x_0, x_1, x_2, x_3] + 0 + 0$$

we stop since divided differences of higher order is zero

=-5+(x-0)[-2.25]+(x-0)(x-0.5) [-3. 5]+(x-0)(x-0. 5)(x-1)[1]

$f(2) \cong p_n(2) = -5 + 2[-2.25] + 2(2 - 0.5)[-3.5] + 2(2 - 0.5)(2 - 1)(1)$

=-17

A simple algorithm can be given for constructing the differences

$$f(x_0), f[x_0, x_1], f[x_0, x_1, x_2], \ldots, f[x_0, x_1, \ldots, x_n]$$

which are necessary for evaluating the Newton form (3.2.9).

Algorithm for Divided Differences

An algorithm for computing a divided difference table can be very efficient and is recommended as the best means for producing an interpolating polynomial.

The data of interpolation are

$$x_i \, , y_i = f(x), \qquad i = 0,1,2, \ldots n$$

Set $d_{0,i} = f(x_i)$ zero order divided difference

$$d_{0,0} = f(x_0)$$

$[x_i, x_{i+1}] = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$ $\qquad d_{1,0} = [x_0, x_1]$

1[st] order divided difference $d_{1,i} = \frac{d_{0,i+1} - d_{0,i}}{x_{i+1} - x_i}$

$$[x_i, x_{i+1}, x_{i+2}] = \frac{[x_{i+1}, x_{i+2}] - [x_i, x_{i+1}]}{x_{i+2} - x_i}$$

$$d_{2,0} = [x_0, x_1, x_2]$$

2[nd] order divided difference $d_{2,i} = \frac{d_{1,i+1} - d_{1,i}}{x_{i+2} - x_i}$

$$d_{3,i} = \frac{d_{2,i+1} - d_{2,i}}{x_{i+3} - x_i}$$

$$d_{k,i} = \frac{d_{k-1,i+1} - d_{k-1,i}}{x_{i+k} - x_i} \quad , \text{ k=0,1,2,…n, i=0,1,…n}$$

Algorithm for divided differences

Input : $x_i, f(x_i), i = 0,1, \dots n$

Output: divided differences of order k, k=0,1,…n

for i=0,1,…n-k, $d_{i,k}$

Step1 : set $d_{0,i}$=$f(x_i)$,i=0,1,…n

Step2 : for k= 1,2,…n

and i=0,1,…n-k do step3

step3 : set $d_{k,i} = \left(d_{k-1,i+1} - d_{k-1,i}\right)/(x_{i+k} - x_i)$

End

Algorithm for Newton interpolation polynomial

$$p_n(x) = \text{f}(x_0) + (x - x_0)[x_0, x_1] + (x - x_0)(x - x_1)[x_0, x_1, x_2,]$$
$$+ (x - x_0)(x - x_1)(x - x_2)[x_0, x_1, x_2, x_3]$$
$$+ \cdots + (x - x_0)(x - x_1) \dots (x - x_{n-1})[x_0, x_1, \dots \dots x_n]$$
$$= d_{0,0} + (x - x_0)d_{1,0} + (x - x_0)(x - x_1)d_{2,0} + (x - x_0)(x - x_1)(x - x_2)d_{3,0} +$$
$$+ \cdots + (x - x_0)(x - x_1) \dots (x - x_{n-1})d_{n,0}$$
$$= \sum_{\text{n}=0}^{\text{n}} \left[\prod_{\text{i}=0}^{\text{k}-1} (x - x_i)\right] d_{k,0}$$

Algorithm for Newton interpolation polynomial

Input : x_i , $d_{k,o}$, i=0,1,…n , k=0,1,…n , value x=t

Output $f(t) \cong p_n(t)$

Step1 : p=0, PI=1

Step2: for k=0 to n

Step3 : for i=0 to n-1

PI=pI*(t-x_i)

Step4 : p=p + pI *$d_{k,i}$

End

Example For $f(x) = \sqrt{x}$, we give in Table 3.3 the values of $p_n(x)$ for various values of x and n. The highest degree polynomial $p_4(x)$ uses function values at the grid points $x_0 = 2.0$ through $x_4 = 2.4$. The necessary divided differences are given in the last example, Table 3.2.

Exercise

Consider the following data for $f(x) = \sqrt{x}$ Obtain Newton's polynomial that interpolate $f(x)$. Then find $f(2.05)$. Evaluate the numerical error.

x	**2.0**	**2.1**	**2.2**	**2.3**	**2.4**
y	**2.414214**	**2.449138**	**2.483240**	**2.516575**	**2.549193**

An error formula divided differences

$f(x) - p_n(x) = (x - x_1)(x - x_1)\dots(x, x_n)[x_0, x_1, \dots x_n, x]$ (1)

Comparing (1) , with the error of Lagrange formula

$$f(x) - p_n(x) = \frac{(x-x_0)(x-x_1)\dots(x-x_n)}{(n+1)\;\;!} f^{(n+1}(c) \quad (2)$$

With c is somewhere between $x_0, x_1, \dots x_n$ *and* x

So we can approximate

$$[x_0, x_1, \dots x_n, x] \cong \frac{f^{(n+1}(c)}{(n+1)\;!} \qquad (3)$$

When a value of x falls outside $\mathscr{H}\{x_0, x_1, \dots, x_n\}$, we often say that $p_n(x)$ extrapolates $f(x)$. In the last example, note the greater inaccuracy of the extrapolated value $p_3(2.45)$ as compared with $p_3(2.05)$ and $p_3(2.15)$. In this text, however, the word *interpolation* always includes the possibility that x falls outside the interval $\mathscr{H}\{x_0, \dots, x_n\}$.

Often we know the value of the function $f(x)$, and we want to compute the corresponding value of x. This is called *inverse interpolation*. It is commonly known to users of logarithm tables as computing the antilog of a number. To compute x, we treat it as the dependent variable and $y = f(x)$ as the independent variable. Given table values (x_i, y_i), $i = 0, \dots, n$, we produce a polynomial $p_n(y)$ that interpolates x_i at y_i, $i = 0, \dots, n$. In effect, we are interpolating the inverse function $g(y) \equiv f^{-1}(y)$; in the error formula of Theorem 3.2, with $x = f^{-1}(y)$,

$$x - p_n(y) = \frac{(y - y_0)\cdots(y - y_n)}{(n+1)!} g^{(n+1)}(\zeta) \qquad (3.2.10)$$

for some $\zeta \in \mathscr{H}\{y, y_0, y_1, \dots, y_n\}$. If they are needed, the derivatives of $g(y)$ can be computed by differentiating the composite formula

$$g(f(x)) = x$$

for example,

$$g'(y) = \frac{1}{f'(x)} \qquad \text{for} \quad y = f(x)$$

Remark

The divided differences symbol

$$[x_0, x_1, \dots x_n] = f[x_0, x_1, \dots x_n, x]$$

As written in some books

Example From Table 3.2, of divided differences for $f(x) = \sqrt{x}$,

$$f[2.0, 2.1, \ldots, 2.4] = -.002084$$

Since $f^{(4)}(x) = -15/(16x^3\sqrt{x})$, it is easy to show that

$$\frac{f^{(4)}(2.3103)}{4!} \doteq -.002084$$

so $\xi \doteq 2.31$ in (3.2.12) for this case.

Example 1

Compute a divided difference table for these function values:

x	3	1	5	6
$f(x)$	1	−3	2	4

Table (1)

Solution

We arrange the given table vertically and compute divided differences by use of Formula (11), arriving at

$x_0 = 3 \qquad x_1 = 1 \qquad x_2 = 5 \qquad x_3 = 6$

$$f[x_0] = f(x_0) = 1$$

$$f[x_1] = f(x_1) = -3$$

$$f[x_2] = f(x_2) = 2$$

$$f[x_3] = f(x_3) = 4$$

$$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0} = \frac{-3-1}{1-3} = \frac{-4}{-2} = 2$$

$$f[x_1, x_2] = \frac{f[x_2] - f[x_1]}{x_2 - x_1} = \frac{2-(-3)}{5-1} = \frac{5}{4}$$

$$f[x_2, x_3] = \frac{f[x_3] - f[x_2]}{x_3 - x_2} = \frac{4-2}{6-5} = \frac{2}{1} = 2$$

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{\frac{5}{4} - 2}{5-3} = \frac{\frac{5-8}{4}}{2} = \frac{-3}{8}$$

$$f[x_1, x_2, x_3] = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1} = \frac{2 - \frac{5}{4}}{6-1} = \frac{\frac{8-5}{4}}{5} = \frac{3}{20}$$

$$f[x_0, x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0} = \frac{\frac{3}{20} - (-\frac{3}{8})}{6-3}$$

$$= \frac{6+15}{40(3)} = \frac{21}{40(3)} = \frac{7}{40}$$

3	1	2	−3/8	7/40
1	−3	5/4	3/20	
5	2	2		
6	4			

$$P(x) = f[x_0] + f[x_0,x_1](x-x_0) + f[x_0,x_1,x_2](x-x_0)(x-x_1)$$
$$+ f[x_0,x_1,x_2,x_3](x-x_0)(x-x_1)(x-x_2) + \cdots$$

$$p(x) = 1 + 2(x - 3) - \tfrac{3}{8}(x - 3)(x - 1) + \tfrac{7}{40}(x - 3)(x - 1)(x - 5)$$

Theorem 3.3 (Hermite–Gennochi) Let $x_0, x_1, \ldots, x_n$ be distinct, and let $f(x)$ be n times continuously differentiable on the interval $\mathscr{H}\{x_0, x_1, \ldots, x_n\}$. Then

$$f[x_0, x_1, \ldots, x_n] = \int \cdots \int_{\tau_n} f^{(n)}(t_0x_0 + \cdots + t_nx_n)\, dt_1 \ldots dt_n \qquad (3.2.13)$$

in which

$$\tau_n = \left\{(t_1, t_2, \ldots, t_n) \mid \text{ all } \quad t_i \geq 0, \sum_1^n t_i \leq 1\right\} \qquad (3.2.14)$$

$$t_0 = 1 - \sum_1^n t_i$$

Note that $t_0 \geq 0$ and $\sum_0^n t_i = 1$.

Proof We show that (3.2.13) is true for $n = 1$ and 2, and these two cases should suggest the general induction proof.

1. $n = 1$. Then $\tau_1 = [0, 1]$.

$$\int_0^1 f'(t_0x_0 + t_1x_1)\, dt_1 = \int_0^1 f'(x_0 + t_1(x_1 - x_0))\, dt_1$$

$$= \frac{1}{x_1 - x_0} f(x_0 + t_1(x_1 - x_0))\Big|_{t_1=0}^{t_1=1}$$

$$= \frac{f(x_1) - f(x_0)}{x_1 - x_0} = f[x_0, x_1]$$

2. $n = 2$. Then τ_2 is the triangle with vertices (0,0), (0,1), and (1,0), shown in Figure 3.1.

$$\int_{\tau_n}\int f''(t_0x_0 + t_1x_1 + t_2x_2)\, dt_1\, dt_2$$

$$= \int_0^1\int_0^{1-t_1} f''(x_0 + t_1(x_1 - x_0) + t_2(x_2 - x_0))\, dt_2\, dt_1$$

$$= \int_0^1 \frac{1}{x_2 - x_0}\left[f'(x_0 + t_1(x_1 - x_0) + t_2(x_2 - x_0))\right]_{t_2=0}^{t_2=1-t_1} dt_1$$

$$= \frac{1}{x_2 - x_0}\left[\int_0^1 f'(x_2 + t_1(x_1 - x_2))\, dt_1 - \int_0^1 f'(x_0 + t_1(x_1 - x_0))\, dt_1\right]$$

$$= \frac{1}{x_2 - x_0}\{f[x_1, x_2] - f[x_0, x_1]\} = f[x_0, x_1, x_2]$$

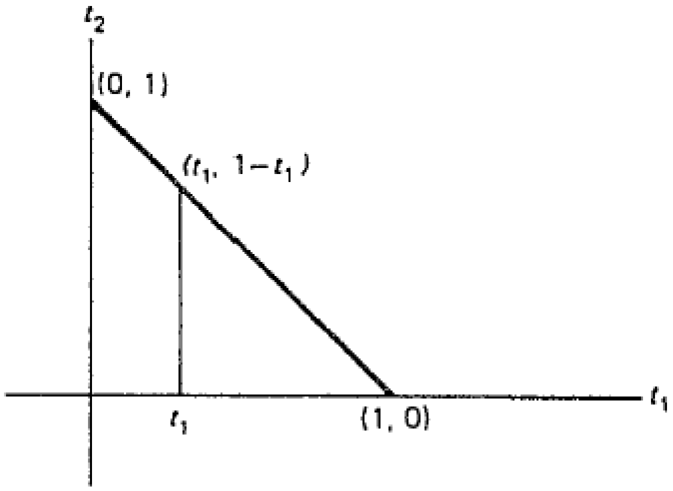


Figure 3.1 Region τ_2.

Do the general case by a similar procedure. Integrate once and reduce to one lower dimension. Then invoke the induction hypothesis and use (3.2.7) to complete the proof.

We can now look at $f[x_0, x_1, \ldots, x_n]$ using (3.2.13). Doing so, we see that if $f(x)$ is n times continuously differentiable on $\mathscr{H}\{x_0, \ldots, x_n\}$, then $f[x_0, \ldots, x_n]$ is a continuous function of the n variables $x_0, x_1, \ldots, x_n$, regardless of whether they are distinct or not. For example, if we let all points coalesce to x_0, then for the nth-order divided difference,

$$f[x_0,\ldots,x_0] = \int\!\cdots\!\int_{\tau_n} f^{(n)}(x_0)\,dt_1\ldots dt_n$$

$$= f^{(n)}(x_0)\cdot \mathrm{Vol}\,(\tau_n).$$

From Problem 15, $\mathrm{Vol}(\tau_n) = 1/n!$, and thus

$$f[x_0,\ldots,x_0] = \frac{f^{(n)}(x_0)}{n!} \tag{3.2.15}$$

This could have been predicted directly from (3.2.12). But if only some of the nodes coalesce, we must use (3.2.13) to justify the existence of $f[x_0,\ldots,x_n]$.

In applications to numerical integration, we need to know whether

$$\frac{d}{dx}f[x_0,\ldots,x_n,x] \tag{3.2.16}$$

exists. If f is $n+2$ times continuously differentiable, then we can apply Theorem 3.3. By applying theorems on differentiating an integral with respect to a parameter in the integrand, we can conclude the existence of (3.2.16). More directly,

$$\frac{d}{dx}f[x_0,\ldots,x_n,x] = \underset{h\to 0}{\mathrm{Limit}}\,\frac{f[x_0,\ldots,x_n,x+h]-f[x_0,\ldots,x_n,x]}{h}$$

$$= \underset{h\to 0}{\mathrm{Limit}}\,\frac{f[x_0,\ldots,x_n,x+h]-f[x,x_0,\ldots,x_n]}{h}$$

$$= \underset{h\to 0}{\mathrm{Limit}}\,f[x,x_0,\ldots,x_n,x+h]$$

$$= f[x,x_0,x_1,\ldots,x_n,x]$$

$$\frac{d}{dx}f[x_0,x_1,\ldots,x_n,x] = f[x_0,x_1,\ldots,x_n,x,x] \tag{3.2.17}$$

The existence and continuity of the right-hand side is guaranteed using (3.2.13).

There is a rich theory involving polynomial interpolation and divided differences, but we conclude at this point with one final straightforward result. If $f(x)$ is a polynomial of degree m, then

$$f[x_0,\ldots,x_n,x] = \begin{cases} \text{polynomial of degree } m-n-1 & n < m-1 \\ a_m & n = m-1 \\ 0 & n > m-1 \end{cases} \tag{3.2.18}$$

where $f(x) = a_m x^m +$ lower degree terms. For the proof, see Problem 14.

Problems

1. Recall the Vandermonde matrix X given in (3.1.3), and define

$$V_n(x) = \det \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ \vdots & & & & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & & x_{n-1}^n \\ 1 & x & x^2 & \cdots & x^n \end{bmatrix}$$

(a) Show that $V_n(x)$ is a polynomial of degree n, and that its roots are $x_0, \ldots, x_{n-1}$. Obtain the formula

$$V_n(x) = (x - x_0) \cdots (x - x_{n-1}) V_{n-1}(x_{n-1})$$

Hint: Expand the last row of $V_n(x)$ by minors to show that $V_n(x)$ is a polynomial of degree n and to find the coefficient of the term x^n.

(b) Show

$$\det(X) \equiv V_n(x_n) = \prod_{0 \le j < i \le n} (x_i - x_j)$$

2. For the basis functions $l_{j,n}(x)$ given in (3.1.5), prove that for any $n \geq$

$$\sum_{j=0}^{n} l_{j,n}(x) = 1 \qquad \text{for all} \quad x$$

3. Recall the Lagrange functions $l_0(x), \ldots, l_n(x)$, defined in (3.1.5) and then rewritten in a slightly different form in (3.2.4), using

$$\Psi_n(x) = (x - x_0) \cdots (x - x_n)$$

Let $w_j = [\Psi_n'(x_j)]^{-1}$. Show that the polynomial $p_n(x)$ interpolating $f(x)$ can be written as

$$p_n(x) = \frac{\sum_{j=0}^{n} [w_j f(x_j)]/(x - x_j)}{\sum_{j=0}^{n} w_j/(x - x_j)}$$

provided x is not a node point. This is called the *barycentric* representation of $p_n(x)$, giving it as a weighted sum of the values $\{f(x_0), \ldots, f(x_n)\}$. For a discussion of the use of this representation, see Henrici (1982, p. 237).

11. Let $x_0, \ldots, x_n$ be distinct real points, and consider the following interpolation problem. Choose a function

$$P_n(x) = \sum_{j=0}^{n} c_j e^{jx}$$

such that

$$P_n(x_i) = y_i \qquad i = 0, 1, \ldots, n$$

with the $\{y_i\}$ given data. Show there is a unique choice of $c_0, \ldots, c_n$. *Hint:* The problem can be reduced to that of ordinary polynomial interpolation.

12. Consider finding a rational function $p(x) = (a + bx)/(1 + cx)$ that satisfies

$$p(x_i) = y_i \qquad i = 1, 2, 3$$

with x_1, x_2, x_3 distinct. Does such a function $p(x)$ exist, or are additional conditions needed to ensure existence and uniqueness of $p(x)$? For a general theory of rational interpolation, see Stoer and Bulirsch (1980, p. 58).

3.3 Finite Differences and Table-Oriented Interpolation Formulas

In the preceding section, we discussed the problem of *interpolating* a function by a polynomial. We return to that problem now. Let f be a function whose values are known or computable at a set of points *(nodes)* $x_o, x_1, \ldots, x_n$. We assume in this section that these points are *distinct*, but they need not be *equally spaced* on the real line.

We know that there exists a unique polynomial p of degree at most n that interpolates f at the $n + 1$ nodes:

$$p(x_i) = f(x_i) \qquad (0 \leqq i \leqq n) \tag{1}$$

Of course, the polynomial p can be constructed as a linear combination of the basic polynomials $1, x, x^2, \ldots, x^n$, *namely,*

$$p(x) = \sum_{k=0}^{n} c_k x^k .$$

As discussed in the previous section, this basis is not recommended, and we prefer to use a basis appropriate to the Newton form of the interpolating polynomial:

$$\begin{aligned}
q_0(x) &= 1 \\
q_1(x) &= (x - x_0) \\
q_2(x) &= (x - x_0)(x - x_1) \\
q_3(x) &= (x - x_0)(x - x_1)(x - x_2) \\
&\vdots \\
q_n(x) &= (x - x_0)(x - x_1)(x - x_2)\cdots(x - x_{n-1})
\end{aligned}$$

These lead to the Newton form

$$p(x) = \sum_{j=0}^{n} c_j q_j(x)$$

The interpolation conditions (1) give rise to a linear system of equations for the determination of the unknown coefficients c_j:

$$\sum_{j=0}^{n} c_j q_j(x_i) = f(x_i) \qquad (0 \leqq i \leqq n) \tag{2}$$

In this system of equations, the coefficient matrix is an $(n + 1)$ x $(n + 1)$ matrix A whose elements are

$$a_{ij} = q_j(x_i) \qquad (0 \leqq i, j \leqq n) \tag{3}$$

The matrix A is lower triangular because

$$\begin{aligned} q_j(x) &= \prod_{k=0}^{j-1}(x - x_k) \\ q_j(x_i) &= \prod_{k=0}^{j-1}(x_i - x_k) = 0 \qquad \text{if } \ i \leqq j - 1 \end{aligned} \tag{4}$$

For example, consider the case of three nodes with

$$\begin{aligned} p_2(x) &= c_0 q_0(x) + c_1 q_1(x) + c_2 q_2(x) \\ &= c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) \end{aligned}$$

Setting $x = x_0$, $x = x_1$, and $x = x_2$, we have a lower triangular system

$$P_2(x) = C_0 + C_1(x - x_0) + C_2(x - x_0)(x - x_1)$$

$x = x_0 \Rightarrow P_2(x_0) = f(x_0)$ and

$$C_0 + 0 + 0 = f(x_0) \longrightarrow \boxed{1}$$

$x = x_1 \Rightarrow P_2(x_1) = f(x_1)$

$$C_0 + C_1(x_1 - x_0) + 0 = f(x_1) \rightarrow \boxed{2}$$

$x = x_2 \Rightarrow P_2(x_2) = f(x_2)$

$$C_0 + C_1(x_2 - x_0) + C_2(x_2 - x_0)(x_2 - x_1) = f(x_2) \quad \boxed{3}$$

والتي يمكن كتابتها على الصورة المناسبة التالية

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & (x_1 - x_0) & 0 \\ 1 & (x_2 - x_0) & (x_2 - x_0)(x_2 - x_1) \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \end{bmatrix} \quad (5)$$

In solving Equation (2) for $c_0, c_1, \ldots, c_n$, we can start at the top and work down, computing the coefficients c_j in the order given by their subscripts. In this process, see that c_0 depends only on $f(x_o)$, that c_1 depends on $f(x_o)$ and $f(x_1)$, and so Thus, c_n depends on f at $x_o, x_1, \ldots, x_n$. We define

$$c_n = f[x_0, x_1, \ldots, x_n] \quad (6)$$

to be the coefficient of q_n when $\sum_{k=0}^{n} c_k q_k$ interpolates f at $x_o, x_1, \ldots, x_n$. Since

$$q_n(x) = (x - x_0)(x - x_1)\cdots(x - x_{n-1}) = x^n + \text{ lower-order terms}$$

we can also say that $f[x_o, x_1, \ldots, x_n]$ is the coefficient of x_n in the polynomial of degree at most n that interpolates f at $x_o, x_1, \ldots, x_n$.
In all of the preceding description, n can take on arbitrary values. The expressions $f[x_o, x_1, \ldots, x_n]$ are called **divided differences** of f.

Explicit formulae for the first few divided differences will now be given. First, $f[x_o]$ is the coefficient of x^0 in the polynomial of degree 0 interpolating f at x_o. Thus, we must have

$$f[x_0] = f(x_0) \quad (7)$$

The quantity $f[x_o, x_1]$ is the coefficient of x in the polynomial of degree at most 1 interpolating f at x_o and x_1. Since that polynomial is

$$p(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) \quad (8)$$

we see that the coefficient of $q_1(x)$ is

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad (9)$$

This gives a hint as to why the term *divided difference* was adopted. A divided difference table of the following form can be displayed

$$\begin{array}{lll} x_0 & f(x_0) & f[x_0, x_1] \\ x_1 & f(x_1) & \end{array}$$

and the interpolation polynomial is easily formed from

$$p(x) = f(x_0) + f[x_0, x_1](x - x_0)$$

Formulae (7) and (9) can also be obtained by solving for c_0 and c_1 in System (5), because $c_0 = f[x_o]$ and $c_1 = f[x_o, x]$, in accordance with Equation (6). Equation (1) allows us to write the Newton interpolating polynomial in the form

$$P(x) = f[x_0] + f[x_0,x_1](x-x_0) + f[x_0,x_1,x_2](x-x_0)(x-x_1) + f[x_0,x_1,x_2,x_3](x-x_0)(x-x_1)(x-x_2) + \cdots$$

Higher-Order Divided Differences

THEOREM1 *Divided differences satisfy the equation*

$$f[x_0, x_1, \ldots, x_n] = \frac{f[x_1, x_2, \ldots, x_n] - f[x_0, x_1, \ldots, x_{n-1}]}{x_n - x_0} \qquad (11)$$

The preceding theorem gives us these particular formulae:

$$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0}$$

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$

etc.

In these formulae, x_o, x_1, . . . can be interpreted independent variables. Because of that, we also have equations such as

$$f[x_i, x_{i+1}, \ldots, x_{i+j}] = \frac{f[x_{i+1}, x_{i+2}, \ldots, x_{i+j}] - f[x_i, x_{i+1}, \ldots, x_{i+j-1}]}{x_{i+j} - x_i} \qquad (13)$$

Here $f[x_i]$, $f[x_i, x_{i+1}]$, $f[x_i, x_{i+1}, x_{i+2}]$, $f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$, *etc.* are differences of order 0, 1, 2, 3, *etc.*, respectively.

If a table of function values (x_i, $f(x_i)$) is given, we can construct from it a table of divided differences. This is customarily laid out in the following form, where differences of orders 0, 1, 2, 3 are shown in each successive column:

x_0	$f[x_0]$	$f[x_0, x_1]$	$f[x_0, x_1, x_2]$	$f[x_0, x_1, x_2, x_3]$
x_1	$f[x_1]$	$f[x_1, x_2]$	$f[x_1, x_2, x_3]$	
x_2	$f[x_2]$	$f[x_2, x_3]$		
x_3	$f[x_3]$			

The information to the left of the vertical line is given, and the quantities on the right are to be computed. Formula (11) is used to do this. The recursive nature of Formula (11) dictates the triangular form of the divided difference table. For example, the data given do not allow us to compute $f[x_3, x_4]$, $f[x_2, x_3, x_4]$, *etc.*

By comparing Equations (10) and (11), we see that the coefficients required in the Newton interpolating polynomial occupy the top row in the divided difference table.

Algorithm for Divided Differences

An algorithm for computing a divided difference table can be very efficient and is recommended as the best means for producing an interpolating polynomial.

Let us change the notation so that our divided difference table has the entries as shown here:

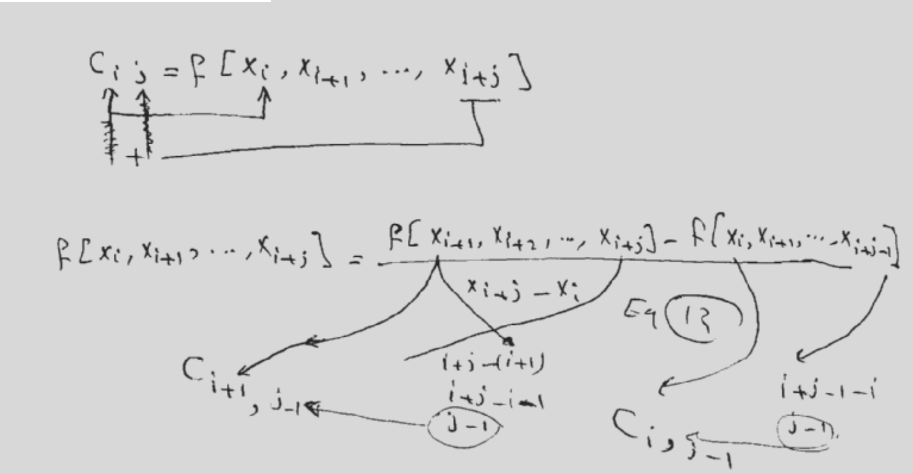
x_0	c_{00}	c_{01}	c_{02}	c_{03}	$\cdots$	$c_{0,n-1}$	$c_{0,n}$
x_1	c_{10}	c_{11}	c_{12}	c_{13}	$\cdots$	$c_{1,n-1}$	
x_2	c_{20}	c_{21}	c_{22}	c_{23}	$\iddots$		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\iddots$			
$\vdots$	$\vdots$	$\vdots$	$\iddots$				
x_{n-1}	$c_{n-1,0}$	$c_{n-1,1}$					
x_n	c_{n0}						

The vertical line separates the data (on the left) from the entries to be computed. It is clear that we have set

$$c_{ij} = f[x_i, x_{i+1}, \ldots, x_{i+j}]$$

An algorithm is obtained from a direct translation of Equation (13), and goes as follows:

```
for j = 1, 2, ..., n do
    for i = 0, 1, ..., n − j do
        c_ij ← (c_{i+1,j−1} − c_{i,j−1})/(x_{i+j} − x_i)
    end
end
```



In this algorithm, the numbers c_{i0} (which are input)

$c_{i0} = f[x_i] = f(x_i)$ $\qquad$ $c_{i0} = P(x_i)$

are the values of the function *f* at the points x_i . They are also the values that the interpolating polynomial will have at those points.

The interpolating polynomial, of course, is

$$p(x) = c_{00} + c_{01}(x-x_0) + c_{02}(x-x_0)(x-x_1) + c_{03}(x-x_0)(x-x_1)(x-x_2) + \cdots + c_{0n}(x-x_0)(x-x_1)\cdots(x-x_{n-1})$$

write an algorithm to compute the Newton divided differences polynomial when n values of f(x) are known at n different nodes.

Programming Exercise

Use your favorite program that Compute a divided difference table for these function values:

x	3	1	5	6
$f(x)$	1	-3	2	4

==

Divided Difference Properties

Divided Difference *symmetry* THEOREM

The divided difference is *a symmetric function of its arguments. Thus, if* $(z_o, z_1, \ldots, z_n)$ is *a permutation of* $(x_o, x_1, \ldots, x_n)$ *then*

$$f[z_0, z_1, \ldots, z_n] = f[x_0, x_1, \ldots, x_n] \qquad (15)$$

Divided Difference Error **THEOREM**

Let p be the polynomial of degree at most n that interpolates a function f at a set of n + 1 *distinct nodes,* $x_o, x_1, \ldots, x_n$. *If t* is *a point different from the nodes, then*

$$f(t) - p(t) = f[x_0, x_1, \ldots, x_n, t]\prod_{j=0}^{n}(t - x_j) \qquad (16)$$

Proof

First, let q be the polynomial of degree at most $n + 1$ that interpolates f at the nodes $x_o, x_1, \ldots, x_n, t$. We know that q is obtained from p by adding one term. In fact,

$$q(x) = p(x) + f[x_0, x_1, \ldots, x_n, t] \prod_{j=0}^{n} (x - x_j)$$

Since $q(t) = f(t)$, we obtain at once (by letting $x = t$)

$$f(t) = p(t) + f[x_0, x_1, \ldots, x_n, t] \prod_{j=0}^{n} (t - x_j)$$

Divided Difference estimation **THEOREM**

If f is n times continuously differentiable on [a, b] and if $x_o, x_1, \ldots, x_n$ *are distinct points in* [*a*, b], *then there exists a point* ξ *in (a, b) such that*

$$f[x_0, x_1, \ldots, x_n] = \frac{1}{n!} f^{(n)}(\xi) \qquad (17)$$

Proof

First, let p be the polynomial of degree at most $n - 1$ that interpolates f at *the nodes xo, x1, . . . , xn-1. By Error In Polynomial Interpolation Theorem of the previous section* , there exists a point ξ in (a, b) such that

$$f(x_n) - p(x_n) = \frac{1}{n!} f^{(n)}(\xi) \prod_{j=0}^{n-1} (x_n - x_j) \qquad (18)$$

By Theorem of Error in this section,

$$f(x_n) - p(x_n) = f[x_0, x_1, \ldots, x_n] \prod_{j=0}^{n-1} (x_n - x_j) \qquad (19)$$

By comparing Equations (18) and (19), we deduce Equation (17).

3. Hermite Interpolation

The term **Hermite interpolation** refers to the interpolation of a function and some of its derivatives at a set of nodes. (A more precise definition will be given later.) When a distinction is being made between this type of interpolation and the simpler type (in which no derivatives are interpolated) the latter is often termed *Lagrange interpolation.*

Basic Concepts

An instructive and useful example of Hermite interpolation is the following. We require a polynomial of least degree that interpolates a function f and its *derivative* f' at two distinct points, say x_o and x_1. The polynomial sought will satisfy these four conditions:

$$p(x_i) = f(x_i) \qquad p'(x_i) = f'(x_i) \qquad (i = 0, 1)$$

Since there are four conditions, it seems reasonable to look for a solution in $\prod_3$, the linear space of all polynomials of degree at most 3. An element of $\prod_3$ has four coefficients at our disposal. Rather than writing $p(x)$ in terms of 1, x, x^2, x^3,

$$p(x) = a + bx + cx^2 + dx^3$$

however, let us write

$$p(x) = a + b(x - x_0) + c(x - x_0)^2 + d(x - x_0)^2(x - x_1)$$

since this will simplify the work. This leads to

$$p'(x) = b + 2c(x - x_0) + 2d(x - x_0)(x - x_1) + d(x - x_0)^2$$

The four conditions on p can now be written in the form

$$\begin{aligned} f(x_0) &= a \\ f'(x_0) &= b \\ f(x_1) &= a + bh + ch^2 \qquad (h = x_1 - x_0) \\ f'(x_1) &= b + 2ch + dh^2 \end{aligned}$$

Obviously, a and b are obtained at once. Then c can be determined from the third equation, in which the terms involving a and b should be transferred to the left side. Finally, d can be determined from the fourth equation. Hence, the problem is solvable-no matter what the values of $f(x_i)$ and $f'(x_i)$ may be.

In general, if values of a function f and some of its derivatives are to be interpolated by a polynomial, we shall encounter some difficulties because the linear systems of equations (from which we expect to compute the coefficients in the polynomial) may be *singular*. A simple example will illustrate this.

Example 1

Find a polynomial p that assumes these values:

$$p(0) = 0,\ p(1) = 1,\ p'(\tfrac{1}{2}) = 2.$$

Solution

Since there are three conditions, we try a quadratic,

$$p(x) = a + bx + cx^2$$

The condition $p(0) = 0$ leads to $a = 0$. The other two conditions lead to

$$\begin{aligned} 1 = p(1) &= b + c \\ 2 = p'(\tfrac{1}{2}) &= b + c \end{aligned}$$

Thus, no quadratic solves our problem. Notice that the coefficient matrix is singular.

If we now try a cubic polynomial for the same problem, we discover that there exists a solution but it is not unique.

$$p(x) = a + bx + cx^2 + dx^3$$

We notice that $a = 0$ as before. The remaining conditions are

$$1 = b + c + d$$
$$2 = b + c + \tfrac{3}{4}d$$

The solution of this system is $d = -4$ and $b + c = 5$.

The *general* problem of this type obviously has some intriguing difficulties associated with it.

A large class of interpolation problems having unique solutions will now be discussed. The problems in this restricted class are the ones generally known as *Hermite* interpolation.

In a Hermite problem, it is assumed that whenever a derivative $p^{(j)}(x_i)$ is to be prescribed توصف (at a node x_i), then $p^{(j-1)}(x_i)$, $p^{(j-2)}(x_i), \ldots, p'(x_i)$, and $p(x_i)$ will also be prescribed.

We choose our notation so that at node x_i, k_i interpolatory conditions are prescribed. Notice that k_i may vary with i. Let the nodes $x_o, x_1, \ldots, x_n$, and suppose that at node x_i these interpolation conditions are given:

$$p^{(j)}(x_i) = c_{ij} \qquad (0 \leqq j \leqq k_i - 1,\ 0 \leqq i \leqq n) \tag{1}$$

The total number of conditions on p is denoted by m + 1, and therefore

$$m + 1 = k_0 + k_1 + \cdots + k_n \tag{2}$$

THEOREM 1 *There exists a unique polynomial p in* $\prod_m$ *fulfilling the Hermite interpolation conditions in Equation* (1).

Example 2

What happens in Hermite interpolation when there is only one node?

Solution

In this case, we require a polynomial p of degree k, say, for which

$$p^{(j)}(x_0) = c_{0j} \qquad (0 \leqq j \leqq k) \qquad (*)$$

The solution is the Taylor polynomial

$$P(x) = P(0) + (x - x_0)\, p^{(1)}(0) + \frac{(x-x_0)^2}{2!}\, p^{(2)}(0) + \cdots + \frac{(x-x_0)^k}{k!}\, p^{(k)}(0) + \cdots$$

using the condition (*)

$$p(x) = c_{00} + c_{01}(x - x_0) + \frac{c_{02}}{2!}(x - x_0)^2 + \cdots + \frac{c_{0k}}{k!}(x - x_0)^k$$

Newton Divided Difference Method for Hermite problem

Now let us explain how the Newton divided difference method can be extended to solve Hermite interpolation problems. We begin with a simple case in which a quadratic polynomial *p* is sought taking prescribed values:

$$p(x_0) = c_{00} \qquad p'(x_0) = c_{01} \qquad p(x_1) = c_{10} \tag{3}$$

We write the divided difference table in this form:

$$\begin{array}{llll} x_0 & c_{00} & c_{01} & ? \\ x_0 & c_{00} & ? & \\ x_1 & c_{10} & & \end{array}$$

The question marks stand for entries that are not yet computed. Observe that x_o appears twice in the argument column since two conditions are being imposed on *p* at x_o. Note further that the prescribed value of $p'(x_o)$ has been placed in the column of first-order divided differences. This is in accordance with the equation

$$\lim_{x \to x_0} f[x_0, x] = \lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0} = f'(x_0)$$

This equation justifies our defining

$$f[x_0, x_0] \equiv f'(x_0)$$

The remaining entries in the divided difference table can be computed in the usual way. The difficulty to be expected when the nodes are repeated occurs only at the entry c_{10}, and the value of c_{10} has been supplied by the *data* rather than being computed. The entries denoted by question marks are then computed in the usual way:

$$p[x_0, x_1] = \frac{p(x_1) - p(x_0)}{x_1 - x_0} = \frac{c_{10} - c_{00}}{x_1 - x_0} \tag{4}$$

and

$$p[x_0, x_0, x_1] = \frac{p[x_0, x_1] - p[x_0, x_0]}{x_1 - x_0} = \frac{c_{10} - c_{00}}{(x_1 - x_0)^2} - \frac{c_{01}}{x_1 - x_0} \tag{5}$$

The interpolating polynomial is written in the usual way:

$$p(x) = p(x_0) + p[x_0, x_0](x - x_0) + p[x_0, x_0, x_1](x - x_0)^2 \qquad (6)$$

Returning to the example used at the beginning of this section, in which we require a polynomial that interpolates a function f and its *derivative* f' at two distinct points, say x_o and x_1. The polynomial sought will satisfy these four conditions:

$$p(x_i) = f(x_i) \qquad p'(x_i) = f'(x_i) \qquad (i = 0, 1)$$

We can obtain the interpolation polynomial

$$p(x) = f(x_0) + f'(x_0)(x - x_0) + f[x_0, x_0, x_1](x - x_0)^2 + f[x_0, x_0, x_1, x_1](x - x_0)^2(x - x_1)$$

directly from the following divided difference table

x_0	$f(x_0)$	$f'(x_0)$	$f[x_0, x_0, x_1]$	$f[x_0, x_0, x_1, x_1]$
x_0	$f(x_0)$	$f[x_0, x_1]$	$f[x_0, x_1, x_1]$	
x_1	$f(x_1)$	$f'(x_1)$		
x_1	$f(x_1)$			

The divided differences in this table can be defined in accordance with Theorem of estimation in Section 4.2. That theorem asserts the existence of a point ξ such that

$$f[x_0, x_1, \ldots, x_k] = \frac{1}{k!} f^{(k)}(\xi)$$

Here it must be assumed that $f^{(k)}$ exists and is continuous in the smallest interval containing the nodes $x_o, x_1, \ldots, x_k$. The point ξ will lie in the same interval. If the length of that interval shrinks to zero, we obtain in the limit

$$f[x_0, x_0, \ldots, x_0] = \frac{1}{k!} f^{(k)}(x_0) \qquad (7)$$

Notice that when $k \geq 2$, we must be careful to include the factor $1/k!$.

Example 3

Use the extended Newton divided difference algorithm to determine a polynomial that takes these values:

$$p(1) = 2 \qquad p'(1) = 3 \qquad p(2) = 6 \qquad p'(2) = 7 \qquad p''(2) = 8$$

Solution

extended Newton divided difference table for this problem is of the form

x_0	$f(x_0)$	$f'(x_0)$	$f[x_0, x_0, x_1]$	$f[x_0, x_0, x_1, x_1]$	$f[x_0, x_0, x_1, x_1, x_1]$
x_0	$f(x_0)$	$f[x_0, x_1]$	$f[x_0, x_1, x_1]$	$f[x_0, x_1, x_1, x_1]$	
x_1	$f(x_1)$	$f'(x_1)$	$f''(x_1)/2$		
x_1	$f(x_1)$				
x_1	$f(x_1)$				

Here x_o *and* x_1 appears twice and three times in the argument column since two and three conditions are being imposed on p at x_o *and* x_1 respectively.

We put the data in the divided difference array as follows, using "?" to signify that quantities are to be computed.

1	2	3	?	?	?
1	2	?	?	?	
2	6	7	4		
2	6				
2	6				

Notice that in row 3, a second difference of 4 is inserted in accordance with Formula (7) in the case $k = 2$.

$$f'(x_0) = p'(x_0) = 3 \qquad f'(x_1) = p'(x_1) = 7$$

$$\frac{f''(x_1)}{2} = \frac{p''(x_1)}{2} = \frac{8}{2} = 4$$

The remaining entries in the divided difference table can be computed in the usual way as follows:

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{6-2}{2-1} = 4$$

$$f[x_0, x_0, x_1] = \frac{f[x_0, x_1] - f[x_0, x_0]}{x_1 - x_0} = \frac{4 - \frac{1}{1!} f'(x_0)}{2-1}$$

$$= \frac{4-3}{2-1} = 1$$

$$f[x_0, x_1, x_1] = \frac{f[x_1, x_1] - f[x_0, x_1]}{x_1 - x_0} = \frac{\frac{1}{1!} f'(x_1) - 4}{2-1}$$

$$= \frac{7-4}{1} = 3$$

$$f[x_0, x_0, x_1, x_1] = \frac{f[x_0, x_1, x_1] - f[x_0, x_0, x_1]}{x_1 - x_0} = \frac{3-1}{2-1} = 2$$

$$f[x_0, x_1, x_1, x_1] = \frac{f[x_1, x_1, x_1] - f[x_0, x_1, x_1]}{x_1 - x_0} = \frac{\frac{1}{2!} f''(x_1) - 3}{2-1}$$

$$= \frac{\frac{8}{2} - 3}{2-1} = \frac{4-3}{1} = 1$$

$$f[x_0, x_0, x_1, x_1, x_1] = \frac{f[x_0, x_1, x_1, x_1] - f[x_0, x_0, x_1, x_1]}{x_1 - x_0} = \frac{1-2}{2-1} = -1$$

The final result is:

1	2	3	1	2	−1
1	2	4	3	1	
2	6	7	4		
2	6				
2	6				

When the array is completed, the numbers in the top row (excluding the node) are the coefficients in the interpolating polynomial:

$$p(x) = 2 + 3(x-1) + (x-1)^2 + 2(x-1)^2(x-2) - (x-1)^2(x-2)^2$$

An error formula for this type of Hermite interpolation is given in the next theorem.

THEOREM of Hermite interpolation error formula

Let $x_0, x_1, \ldots, x_n$ be distinct nodes in [a,b] and let $f \in C^{2n+2}[a, b]$. If p is the polynomial of degree at most 2n + 1 such that

$$p(x_i) = f(x_i) \qquad p'(x_i) = f'(x_i) \qquad (0 \leqq i \leqq n)$$

then to each x in [a, b] there corresponds a point ξ in (a, b) such that

$$f(x) - p(x) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \prod_{i=0}^{n} (x - x_i)^2$$

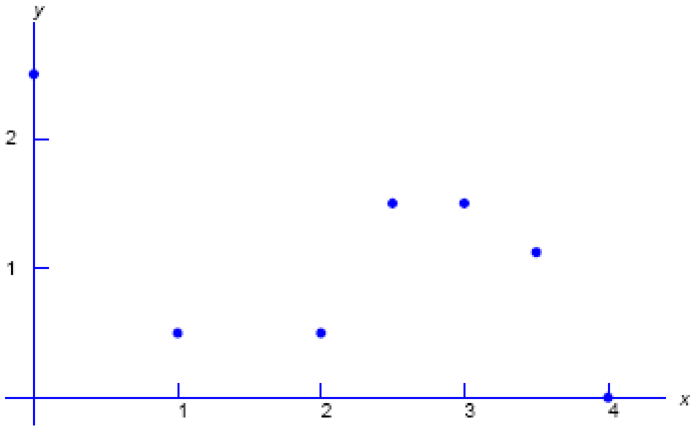
5- PIECEWISE POLYNOMIAL INTERPOLATION

Recall the preceding three section, we study some interpolation methods. There method may fail in some cases such as when the interpolated function *f* (*x*) is oscillatory.

To obtain interpolants that are better behaved, we look at other forms of interpolating functions. Consider the data

x	0	1	2	2.5	3	3.5	4
y	2.5	0.5	0.5	1.5	1.5	1.125	0

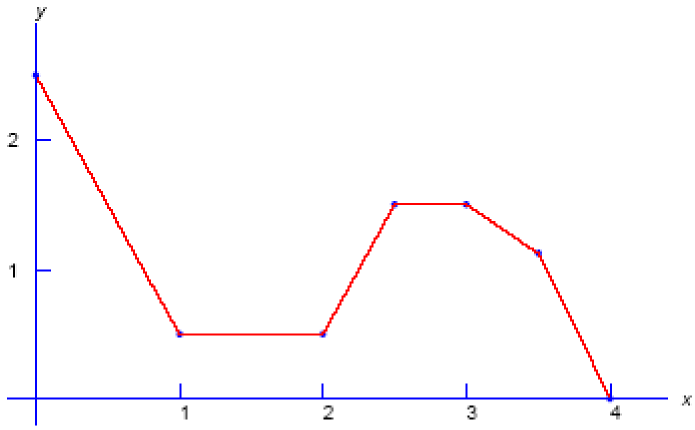
What are methods of interpolating this data, other than using a degree 6 polynomial. Shown in the text are the graphs of the degree 6 polynomial interpolant, along with those of piecewise linear and a piecewise quadratic interpolating functions.
Since we only have the data to consider, we would generally want to use an interpolant that had somewhat the shape of that of the piecewise linear interpolant.



The data points

Piecewise Polynomial Functions

Consider being given a set of data points $(x_1, y_1), \ldots,(x_n, y_n)$, with $x_1 < x_2 < \cdots < x_n$ Then the simplest way to connect the points (x_j, y_j) is by straight line segments. This is called a piecewise linear interpolant of the data $\{(x_j, y_j)\}$.



Piecewise linear interpolation

This graph has "corners", and often we expect the interpolant to have a smooth graph.

To obtain a somewhat smoother graph, consider using piecewise quadratic interpolation. Begin by constructing the quadratic polynomial that interpolates

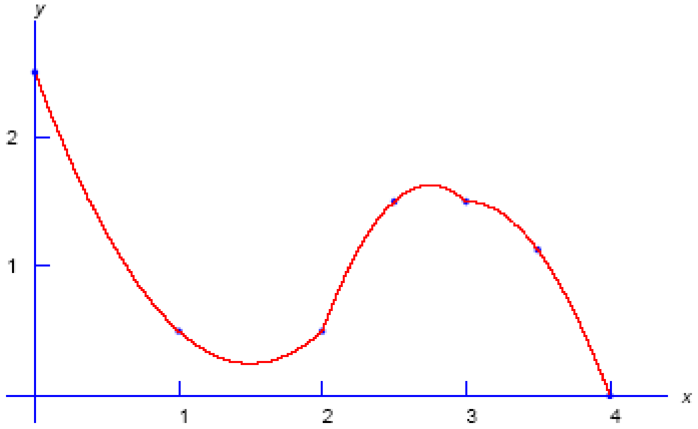
$$\{(x_1, y_1), (x_2, y_2), (x_3, y_3)\}$$

Then construct the quadratic polynomial that interpolates

$$\{(x_3, y_3), (x_4, y_4), (x_5, y_5)\}$$

Continue this process of constructing quadratic interpolants on the subintervals

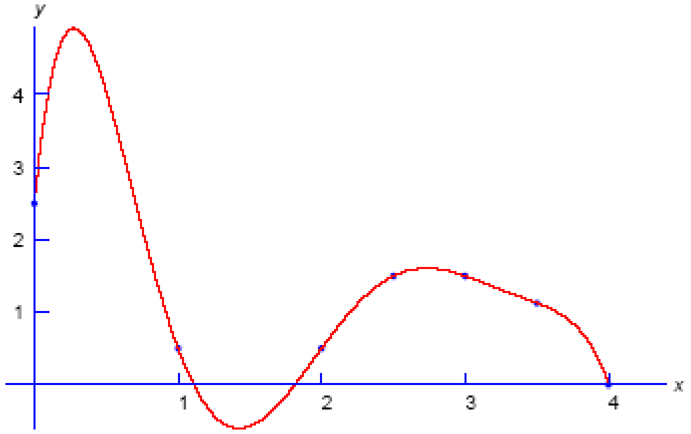
$[x_1, x_3], [x_3, x_5], [x_5, x_7], \ldots$



Piecewise quadratic interpolation

If the number of subintervals is even (and therefore n is odd), then this process comes out fine, with the last interval being $[x_{n-2}, x_n]$. This was illustrated on the graph for the preceding data. If, however, n is even, then the approximation on the last interval must be handled by some modification of this procedure. Suggest such!

With piecewise quadratic interpolants, however, there are "corners" on the graph of the interpolating function. With our preceding example, they are at x_3 and x_5. we can avoid this by enlarging the polynomial power for interpolation.



Polynomial Interpolation

Smooth Non-Oscillatory Interpolation

Let data points (x_1, y_1), ..., (x_n, y_n) be given, as let $x_1 < x_2 < \cdots < x_n$

Consider finding functions s(x) for which the following properties hold:

(1) $s(x_i) = y_i$, i= 1, ..., n

(2) $s(x)$, $s'(x)$, $s''(x)$ are continuous on $[x_1, x_n]$.

Then among such functions s(x) satisfying these properties, find the one which minimizes the integral

$$\int_{x_1}^{x_n} \left|s''(x)\right|^2 \, dx$$

The idea of minimizing the integral is to obtain an interpolating function for which the first derivative does not change rapidly. It turns out there is a unique solution to this problem, and it is called a natural cubic spline function.

SPLINE FUNCTIONS

Let a set of node points $\{x_i\}$ be given, satisfying

$$a \le x_1 < x_2 < \cdots < x_n \le b$$

for some numbers a and b. Often we use [a, b] =[x_1, x_n]. A cubic spline function s(x) on [a, b] with "breakpoints" or "knots" $\{xi\}$ has the following properties:

1. On each of the intervals
   [a, x_1], [x_1, x_2], ..., [x_{n-1}, b]
   s(x) is a polynomial of degree ≤ 3.
2. s(x), s'(x), s"(x) are continuous on [a, b].

In the case that we have given data points (x_1, y_1),...,(x_n, y_n), we say s(x) is a cubic interpolating spline function for this data if

3. s(x_i) = y_i, i = 1, ..., n.

EXAMPLE

Define

$$(x-\alpha)_+^3 = \begin{cases} (x-\alpha)^3, & x \ge \alpha \\ 0, & x \le \alpha \end{cases}$$

This is a cubic spline function on (−∞, ∞) with the single breakpoint $x_1 = \alpha$.
Combinations of these form more complicated cubic spline functions. For example,

$$s(x) = 3\,(x-1)_+^3 - 2\,(x-3)_+^3$$

is a cubic spline function on (−∞, ∞) with the breakpoints x_1=1, x_2=3.
Define

$$s(x) = p_3(x) + \sum_{j=1}^{n} a_j \left(x - x_j\right)_+^3$$

with p_3(x) some cubic polynomial. Then s(x) is a cubic spline function on (−∞, ∞) with breakpoints {x_1, ..., x_n}.
Return to the earlier problem of choosing an interpolating function s(x) to minimize the integral

$$\int_{x_1}^{x_n} \left| s''(x) \right|^2 \, dx$$

There is a unique solution to problem. The solution s(x) is a cubic interpolating spline function, and moreover, it satisfies

$$s''(x_1) = s''(x_n) = 0$$

Spline functions satisfying these boundary conditions are called “natural” cubic spline functions, and the solution to our minimization problem is a “natural cubic interpolatory spline function”. We will show a method to construct this function from the interpolation data.

Motivation for these boundary conditions can be given by looking at the physics of bending thin beams of flexible materials to pass thru the given data. To the left of x_1 and to the right of x_n, the beam is straight and therefore the second derivatives are zero at the transition points x_1 and x_n.

Construction Of The Interpolating Spline Function

To make the presentation more specific, suppose we have data

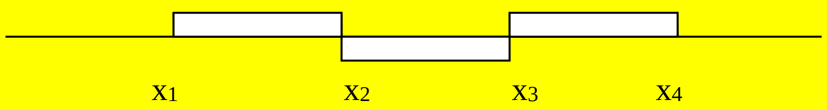
$$(x_1, y_1) , (x_2, y_2) , (x_3, y_3) , (x_4, y_4)$$

With

$$x_1 < x_2 < x_3 < x_4.$$

Then on each of the intervals

$$[x_1, x_2] , [x_2, x_3] , [x_3, x_4]$$



s(x) is a cubic polynomial. Taking the first interval, s(x) is a cubic polynomial and s"(x) is a linear polynomial. Let

$$M_i = s''(x_i), \; i = 1, 2, 3, 4$$

Then on $[x_1, x_2]$,

$$s''(x) = \frac{(x_2 - x)\, s''(x_1)}{x_2 - x_1} + \frac{(x - x_1)\, s''(x_2)}{x_2 - x_1}$$

$$s''(x_1) = 1 \; s''(x_1) + 0$$

$$s''(x_2) = 0 \; s''(x_1) + 1 \; s''(x_2)$$

أي أننا يمكن أن نكتب

$$s''(x) = \frac{(x_2 - x)\, s''(x_1) + (x - x_1)\, s''(x_2)}{x_2 - x_1}$$

وبالتالي نكتب

$$s''(x) = \frac{(x_2 - x)\,M_1 + (x - x_1)\,M_2}{x_2 - x_1}, \quad x_1 \le x \le x_2$$

We can find s(x) by integrating twice:

$$s(x) = \frac{(x_2 - x)^3 M_1 + (x - x_1)^3 M_2}{6\,(x_2 - x_1)} + c_1 x + c_2$$

We determine the constants of integration by using

$s(x_1) = y_1$, $s(x_2) = y_2$ (*)

Then

صفر

$$y_1 = \frac{(x_2 - x_1)^3 M_1 + (x_1 - x_1)^3 M_2}{6\,(x_2 - x_1)} + c_1 x_1 + c_2$$

صفر

$$y_2 = \frac{(x_2 - x_2)^3 M_1 + (x_2 - x_1)^3 M_2}{6\,(x_2 - x_1)} + c_1 x_2 + c_2$$

solving these two eqs., we have

c_1 and c_2

$$s(x) = \frac{(x_2 - x)^3 M_1 + (x - x_1)^3 M_2}{6\,(x_2 - x_1)} + \frac{(x_2 - x)\,y_1 + (x - x_1)\,y_2}{x_2 - x_1} - \frac{x_2 - x_1}{6}\left[(x_2 - x)\,M_1 + (x - x_1)\,M_2\right]$$

for $x_1 \le x \le x_2$.

We can repeat this on the intervals $[x_2, x_3]$ and $[x_3, x_4]$, obtaining similar formulas.
For $x_2 \le x \le x_3$,

$$s(x) = \frac{(x_3 - x)^3 M_2 + (x - x_2)^3 M_3}{6(x_3 - x_2)} + \frac{(x_3 - x) y_2 + (x - x_2) y_3}{x_3 - x_2} - \frac{x_3 - x_2}{6}\left[(x_3 - x) M_2 + (x - x_2) M_3\right]$$

For $x_3 \le x \le x_4$,

$$s(x) = \frac{(x_4 - x)^3 M_3 + (x - x_3)^3 M_4}{6(x_4 - x_3)} + \frac{(x_4 - x) y_3 + (x - x_3) y_4}{x_4 - x_3} - \frac{x_4 - x_3}{6}\left[(x_4 - x) M_3 + (x - x_3) M_4\right]$$

We still do not know the values of the second derivatives $\{M_1, M_2, M_3, M_4\}$. The above formulas guarantee that s(x) and s''(x) are continuous for $x_1 \le x \le x_4$. For example, the formula on $[x_1, x_2]$ yields

$$s(x_2) = y_2,\ s''(x_2) = M_2$$

The formula on $[x_2, x_3]$ also yields

$$s(x_2) = y_2,\ s''(x_2) = M_2$$

All that is lacking is to make s'(x) continuous at x_2 and x_3. Thus we require

$$s'(x_2 + 0) = s'(x_2 - 0)$$
$$s'(x_3 + 0) = s'(x_3 - 0)$$

This means

$$\lim_{x \to x_2^-} s'(x) = \lim_{x \to x_2^+} s'(x) \qquad (**)$$

and similarly for x_3.

To simplify the presentation somewhat, I assume in the following that our node points are equally spaced:

$$x_2 = x_1 + h,\ x_3 = x_1 + 2h,\ x_4 = x_1 + 3h$$

Then our earlier formulas simplify to

$$s(x) = \frac{(x_2 - x)^3 M_1 + (x - x_1)^3 M_2}{6h} + \frac{(x_2 - x)\, y_1 + (x - x_1)\, y_2}{h} - \frac{h}{6}\left[(x_2 - x)\, M_1 + (x - x_1)\, M_2\right]$$

for $x_1 \le x \le x_2$, with similar formulas on $[x_2, x_3]$ and $[x_3, x_4]$.
Without going thru all of the algebra, the conditions (**) leads to the following pair of equations.

$$\frac{h}{6}M_1 + \frac{2h}{3}M_2 + \frac{h}{6}M_3 = \frac{y_3 - y_2}{h} - \frac{y_2 - y_1}{h}$$

$$\frac{h}{6}M_2 + \frac{2h}{3}M_3 + \frac{h}{6}M_4 = \frac{y_4 - y_3}{h} - \frac{y_3 - y_2}{h}$$

This gives us two equations in four unknowns. The earlier boundary conditions on s''(*x*) gives us immediately

$$M_1 = M_4 = 0$$

$s''(x_4) = s''(x_1) = 0$ لأن المادة خارج النظام = صفر

Then we can solve the linear system for M_2 and M_3.

[Q] Construct cubic spline interpolation function that interpolates the data (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4),
with the conditions $f''(x_1) = f''(x_4) = 0$.

EXAMPLE

Consider the interpolation data points

x	1	2	3	4
y	1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$

In this case, h = 1, and linear system becomes

$$\begin{aligned} \frac{2}{3}M_2 + \frac{1}{6}M_3 &= y_3 - 2y_2 + y_1 = \frac{1}{3} \\ \frac{1}{6}M_2 + \frac{2}{3}M_3 &= y_4 - 2y_3 + y_2 = \frac{1}{12} \end{aligned}$$

This has the solution

$M_2 = ½$, $M_3 = 0$

This leads to the spline function formula on each subinterval.
On [1, 2],

$$\begin{aligned} s(x) &= \frac{(x_2 - x)^3 M_1 + (x - x_1)^3 M_2}{6h} \\ &\quad + \frac{(x_2 - x)\, y_1 + (x - x_1)\, y_2}{h} \\ &\quad - \frac{h}{6}\left[(x_2 - x)\, M_1 + (x - x_1)\, M_2\right] \\ &= \frac{(2-x)^3 \cdot 0 + (x-1)^3 \left(\frac{1}{2}\right)}{6} + \frac{(2-x)\cdot 1 + (x-1)\left(\frac{1}{2}\right)}{1} \\ &\quad - \frac{1}{6}\left[(2-x)\cdot 0 + (x-1)\left(\frac{1}{2}\right)\right] \\ &= \tfrac{1}{12}(x-1)^3 - \tfrac{7}{12}(x-1) + 1 \end{aligned}$$

Similarly, for $2 \le x \le 3$,

$$s(x) = \frac{-1}{12}(x-2)^3 + \frac{1}{4}(x-2)^2 - \frac{1}{3}(x-1) + \frac{1}{2}$$

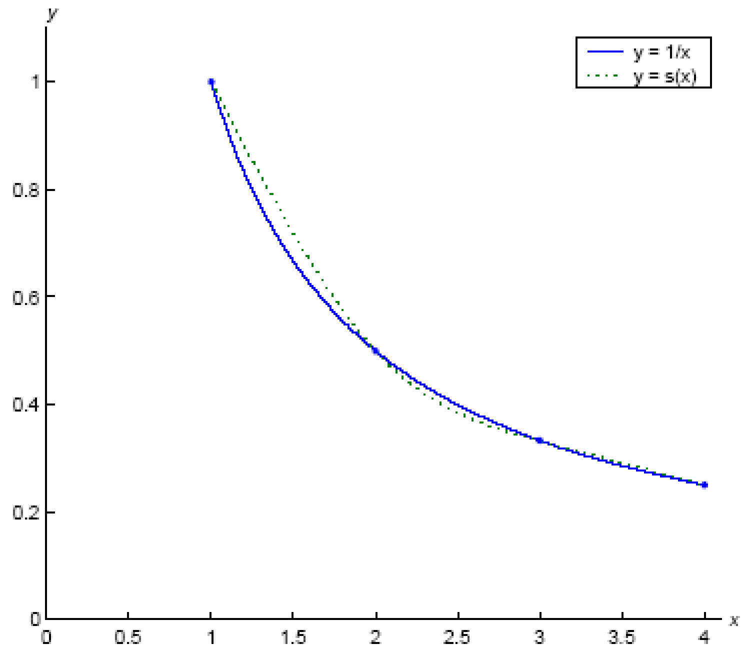
and for $3 \le x \le 4$,

$$s(x) = \frac{-1}{12}(x-4) + \frac{1}{4}$$

so

$$s(x) = \begin{cases} \frac{1}{12}x^3 - \frac{1}{4}x^2 - \frac{1}{3}x + \frac{3}{2}, & 1 \le x \le 2, \\ -\frac{1}{12}x^3 + \frac{3}{4}x^2 - \frac{7}{3}x + \frac{17}{6}, & 2 \le x \le 3, \\ -\frac{1}{12}x + \frac{7}{12}, & 3 \le x \le 4 \end{cases}$$

x	1	2	3	4
y	1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$



Graph of example of natural cubic spline interpolation

THE GENERAL PROBLEM

Consider the spline interpolation problem with n nodes

$$(x_1, y_1),\ (x_2, y_2),\ \ldots,\ (x_n, y_n)$$

and assume the node points {xi} are evenly spaced,

$$x_j = x_1 + (j-1)h, \quad j = 1, \ldots, n$$

We have that the interpolating spline s(x) on $x_j \le x \le x_{j+1}$ is given by

$$s(x) = \frac{(x_{j+1}-x)^3 M_j + (x-x_j)^3 M_{j+1}}{6h} + \frac{(x_{j+1}-x)\, y_j + (x-x_j)\, y_{j+1}}{h} - \frac{h}{6}\left[(x_{j+1}-x) M_j + (x-x_j) M_{j+1}\right]$$

for j = 1, ..., n − 1.

To enforce continuity of s'(x) at the interior node points $x_2, \ldots, x_{n-1}$, the second derivatives{Mj}must satisfy the linear equations

$$\frac{h}{6} M_{j-1} + \frac{2h}{3} M_j + \frac{h}{6} M_{j+1} = \frac{y_{j-1} - 2y_j + y_{j+1}}{h}$$

for j = 2, ..., n − 1. Writing them out,

$$\begin{aligned}
\frac{h}{6}M_1 + \frac{2h}{3}M_2 + \frac{h}{6}M_3 &= \frac{y_1 - 2y_2 + y_3}{h} \\
\frac{h}{6}M_2 + \frac{2h}{3}M_3 + \frac{h}{6}M_4 &= \frac{y_2 - 2y_3 + y_4}{h} \\
&\vdots \\
\frac{h}{6}M_{n-2} + \frac{2h}{3}M_{n-1} + \frac{h}{6}M_n &= \frac{y_{n-2} - 2y_{n-1} + y_n}{h}
\end{aligned}$$

This is a system of n −2 equations in the n unknowns $\{M_1, ..., M_n\}$. Two more conditions must be imposed on s(x) in order to have the number of equations equal the number of unknowns, namely n. With the added boundary conditions, this form of linear system can be solved very efficiently.

ERROR IN CUBIC SPLINE INTERPOLATION

Let an interval [a, b] be given, and then define

$$h = \frac{b-a}{n-1}, \qquad x_j = a + (j-1)h, \quad j = 1, ..., n$$

Suppose we want to approximate a given function *f* (*x*) on the interval [a,b] using cubic spline interpolation. Define
$y_i = f(x_i)$, j = 1, ..., n

Let $s_n(x)$ denote the cubic spline interpolating this data. Then it can be shown that for a suitable constant c,

$$E_n \equiv \max_{a \le x \le b} |f(x) - s_n(x)| \le ch^4$$

where c depends on $f''(a)$, $f''(b)$, and $\max_{a \le x \le b} \left|f^{(4)}(x)\right|$.
The corresponding bound for natural cubic spline interpolation contains only a term of h_2 rather than h_4; it does not converge to zero as rapidly.

Chapter 5

Numerical Differentiation

Principle of Linear Numerical Differentiation

There are two major reasons for considering numerically approximations of the differentiation process.

1. Approximation of derivatives in ordinary differential equations and partial differential equations. This is done in order to reduce the differential equation to a form that can be solved more easily than the original differential equation.

2. Forming the derivative of a function $f(x)$ which is known only as empirical data $\{(x_i, y_i) \mid i = 1, . . . , m\}$. The data generally is known only approximately, so that $y_i \cong f(x_i)$, i = 1, . . . , m.

<u>4.1 differentiation for the first order derivative</u>

<u>4.1.1 differentiation depends on two points</u>

<u>Example1:</u>
Use Taylor's expansion to obtain two different formulae for finite difference approximation of $f'(x)$ that depend on two points.
Answer:
Consider Taylor's expansion for function $f(x)$ at a point x=a, so

$$f(x) = f(a) + (x-a)f^{(1)}(a) + \frac{1}{2}(x-a)^2 f^{(2)}(a)$$
$$+ \cdots + \frac{1}{n!}(x-a)^n f^{(n)}(a) \qquad (1)$$

Take $a = x_i$ and $x = x_i + h$, so $x - a = x_i + h - x_i = h$, Eq(1) become

$$f(x_i + h) = f(x_i) + hf'(x_i) + \tfrac{1}{2!}h^2 f''(x_i) + \tfrac{1}{3!}h^3 f^{(3)}(x_i) + \tfrac{1}{4!}h^4 f^{(4)}(x_i) + \ldots$$

(2)

$$f(x_i + h) - f(x_i) = hf'(x_i) + \tfrac{1}{2!}h^2 f''(x_i) + \tfrac{1}{3!}h^3 f^{(3)}(x_i) + \tfrac{1}{4!}h^4 f^{(4)}(x_i) + \ldots$$

By canceling terms of degree higher than 2, we obtain

$$f(x_i + h) - f(x_i) = hf'(x_i) + \tfrac{1}{2!}h^2 f''(c), \quad x_i \le c \le x_i + h$$

Dividing by h

$$f'(x_i) = \frac{f(x_i+h) - f(x_i)}{2h} - \frac{h}{2} f''(\xi), \quad x_i \le c \le x_i + h \ldots\ldots(3)$$

Formula (3) is called a **forward difference** formula for approximating f '(x).

We can write:

$$\boldsymbol{f'(x_i) = D_F + E_F}$$

Where

$$\boldsymbol{D_F} = \frac{f(x_i+h) - f(x_i)}{2h}, \quad \boldsymbol{E_F} = -\frac{h}{2} f''(\xi)$$

$\boldsymbol{D_F}$ called a **forward difference** formula for approximating $f'(x\ \ )$ and $\boldsymbol{E_F}$ is the error.

Take $a = x_i$ and $x = x_i - h$, so $x - a = x_i - h - x_i = -h$, Eq(1) become

$$f(x_i - h) = f(x_i) - hf'(x_i) + \tfrac{1}{2!}h^2 f''(x_i) - \tfrac{1}{3!}h^3 f^{(3)}(x_i) + \tfrac{1}{4!}h^4 f^{(4)}(x_i) + \ldots$$

(4)

$f(x_i - h) - f(x_i) = -hf'(x_i) + \frac{1}{2!}h^2 f''(c), \; x_i - h \le c \le x_i$

Dividing by h

$$f'(x_i) = \frac{f(x_i) - f(x_i - h)}{2h} + \frac{h}{2} f''(\xi), \quad x_i - h \le c \le x_i \text{.....(3)}$$

Formula (5) is called a **backward difference** formula for approximating $f'(x\quad)$.

forward and backward difference are called two points methods

Example2:

Use forward difference formula for approximating $f(x) = x^3$ at $x = 2$. Take $h = 0.01$. Compute the numerical and estimated errors.

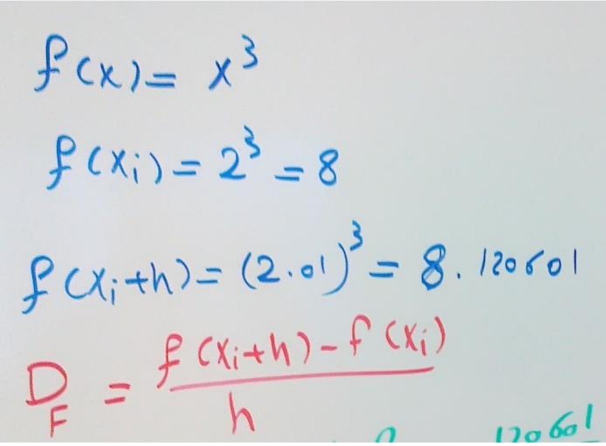
Answer:

$x_i = 2.0 \qquad x_i + h = 2.01$

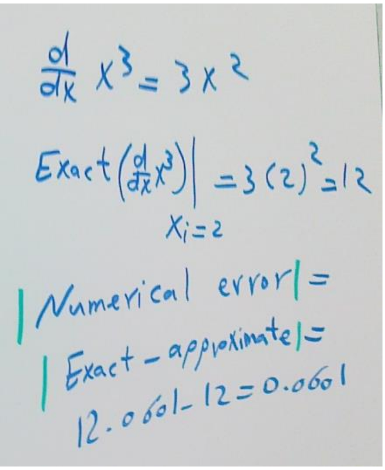
$f'(x_i) = F + E_F$

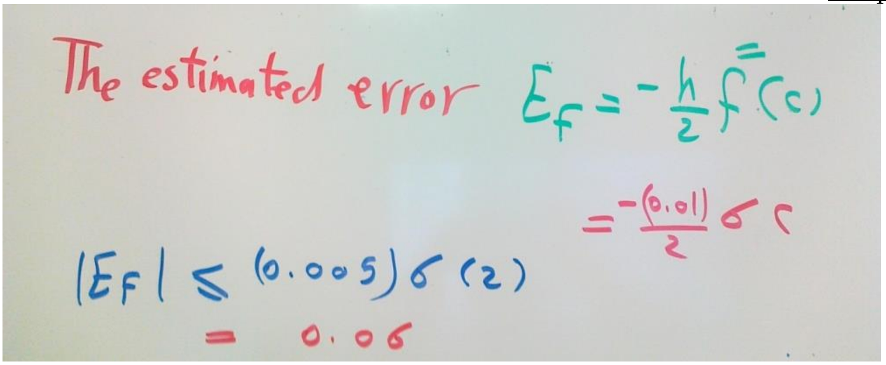
$F = \frac{f(x_i + h) - f(x_i)}{h}, \;\; E_F = -\frac{1}{2}hf''(c), \;\; x_i \le c \le x_i + h$

Numerical Differentiation



$$D_F = \frac{f\left(x_1+h\right)-f\left(x_1\right)}{h} = \frac{8.120601-8.0}{0.01} = 12.0601$$





^^

This is the matlab program for numerical differentiation by forward difference formula.

x=1.0;
h = 0.0001;
f = @(x) x.^3;
fp=f(x+h);
f0=f(x);
st='forward FD Differentiation is'
DIFf = (fp-f0)/h
	forward FD Differentiation is
	DIFf =3.0003

^^

Example3:

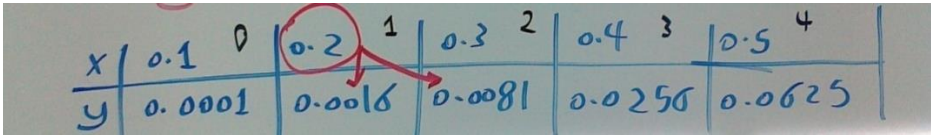
Consider the function given by the following data

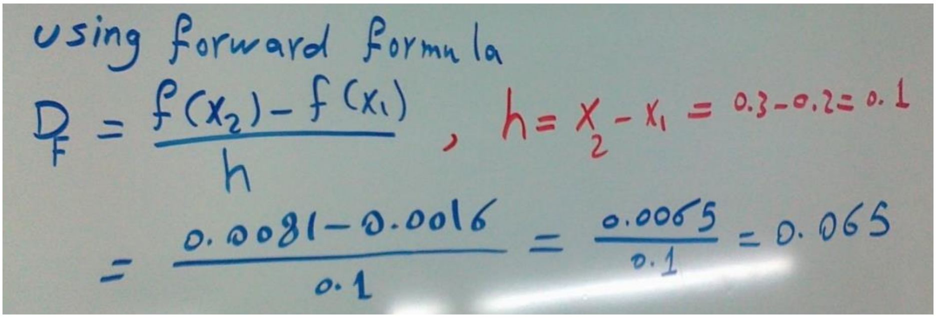
x	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
y	**0.0001**	**0.0016**	**0.0081**	**0.0256**	**0.0625**

Use forward and backward difference formula for approximating $f'(x)$ at $x = 0.2$. Compute the numerical and estimated errors if $f(x) = x^4$.

Answer:

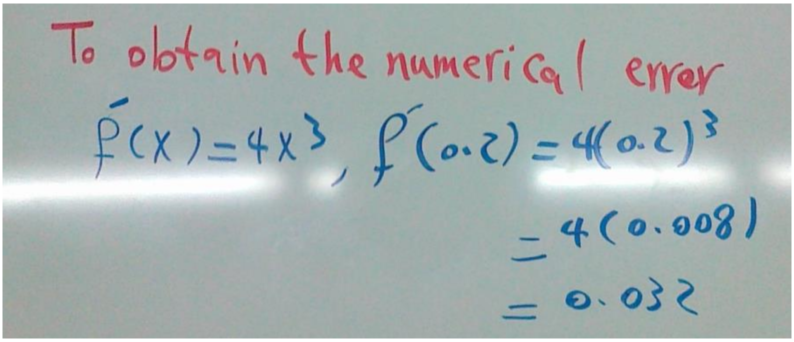
Numerical Differentiation





$$f'(x_1) = \frac{f(x_1+h) - f(x_1)}{h} = \frac{0.0081 - 0.0016}{0.1} = \frac{0.0065}{0.1} = 0.065$$

$f_e'(0.2) = 0.032 \quad |Error| = |0.065 - 0.032| = 0.033$



$$f'(x_1) = \frac{f(x_1) - f(x_1-h)}{h} = \frac{0.0016 - 0.0001}{0.1} = 0.015$$

$$|Error| = |0.032 - 0.015| = 0.017$$

To estimate the error

$$E = -\frac{h}{2} f''(c) \qquad 0.1 \le c \le 0.2$$

$$f(x) = x^4, \quad f'(x) = 4x^3, \quad f'' = 12x^2$$

$$|f''(c)| = 12|c|^2 \qquad 0.2 \le c \le 0.3$$

The largest value for this function is at $c = 0.3$

$$|f''(c)| \le 12(0.3)^2 = (0.09)12$$

$$= 1.08$$

So

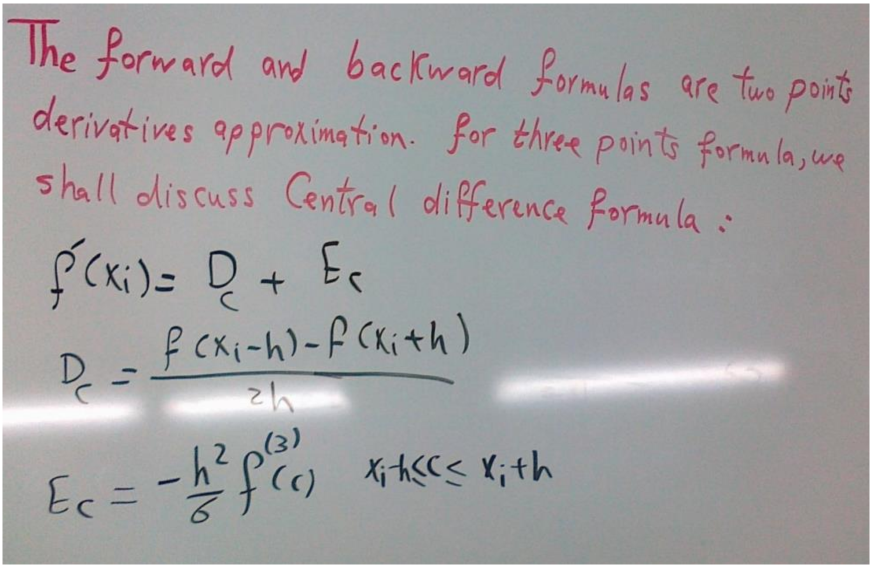
$$|E| = \frac{h}{2}|f''(c)|$$

$$\le \frac{0.1}{2}(1.08) = 0.058$$

$$|Error| = \frac{h}{2}|f''(c)| = \frac{0.1}{2}\left(12c^2\right) = 0.6c^2$$

Select $c = x_1 = 0.2$

$$= 0.6(0.2)^2 = 0.6(0.04) = 0.024$$



4.1.2 differentiation depends on three points

Example1:

Use Taylor's expansion to obtain different formulae for finite difference approximation of $f'(x_i)$ that depend on three points.

Answer:

Consider Taylor's expansion for function $f(x)$ at a point $x = a$, so

$$f(x) = f(a) + (x-a)f^{(1)}(a) + \frac{1}{2}(x-a)^2 f^{(2)}(a) + \cdots + \frac{1}{n!}(x-a)^n f^{(n)}(a) \qquad (1)$$

Take $a = x_i$ and $x = x_i + h$, so $x - a = x_i + h - x_i = h$, Eq(1) become

$$f(x_i + h) = f(x_i) + hf'(x_i) + \tfrac{1}{2!}h^2 f''(x_i) + \tfrac{1}{3!}h^3 f^{(3)}(x_i) + \tfrac{1}{4!}h^4 f^{(4)}(x_i) + \ldots$$

(2)

Take $a = x_i$ and $x = x_i - h$, so $x - a = x_i - h - x_i = -h$, Eq(1) become

$$f(x_i - h) = f(x_i) - hf'(x_i) + \tfrac{1}{2!}h^2 f''(x_i) - \tfrac{1}{3!}h^3 f^{(3)}(x_i) + \tfrac{1}{4!}h^4 f^{(4)}(x_i) + \ldots$$

(3)

We use Eq.(2)-(3) to obtain

$$f(x_i + h) - f(x_i - h) = f(x_i) + hf'(x_i) + \tfrac{1}{2!}h^2 f''(x_i) + \tfrac{1}{3!}h^3 f^{(3)}(x_i) + \tfrac{1}{4!}h^4 f^{(4)}(x_i) + \ldots - \left[f(x_i) - hf'(x_i) + \tfrac{1}{2!}h^2 f''(x_i) - \tfrac{1}{3!}h^3 f^{(3)}(x_i) + \tfrac{1}{4!}h^4 f^{(4)}(x_i) + \ldots\right]$$

So

$$f(x_i + h) - f(x_i - h) = 2hf'(x_i) + \tfrac{2}{3!}h^3 f^{(3)}(x_i) + \tfrac{2}{6!}h^6 f^{(6)}(x_i) + \ldots$$

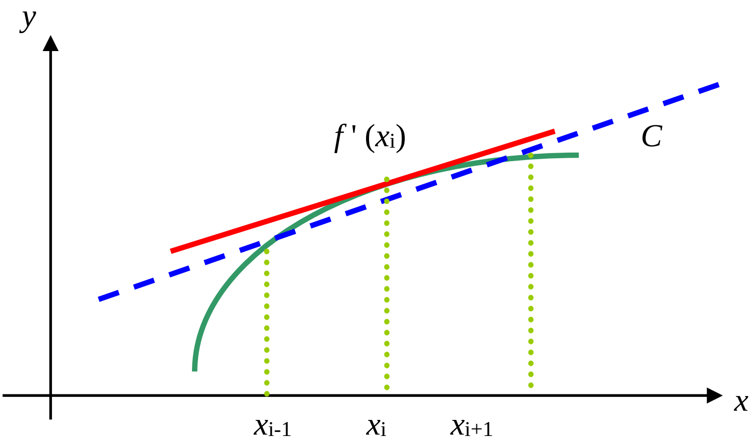
$$f(x_i + h) - f(x_i - h) = 2hf'(x_i) + \tfrac{2}{6}h^3 f^{(3)}(c), \; x_i - h \le c \le x_i + h$$

Dividing by *2h*

$$f'(x_i) = \frac{f(x_i+h) - f(x_i-h)}{2h} - \frac{h^2}{6}f'''(\xi), \qquad x_i - h \le c \le x_i + h$$

(4)

This is the central method



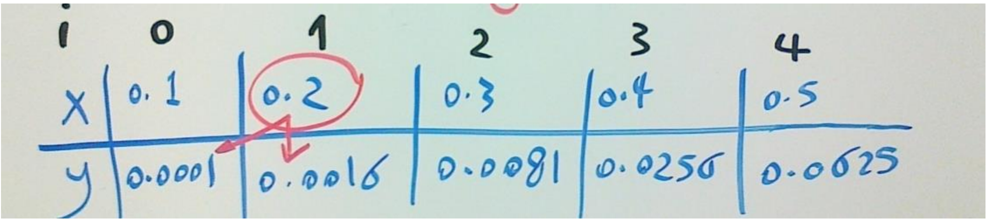
Example3:

Consider the function given by the following data

x	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
y	**0.0001**	**0.0016**	**0.0081**	**0.0256**	**0.0625**

Use backward and central difference formula for approximating $f'(x)$ at $x = 0.2$. Compute the numerical and estimated errors for each if $f(x) = x^4$. Which of them is the best.

Answer:



Answer: $h = x_2 - x_1 = 0.2 - 0.1 = 0.1$

backward formula

$$D_B = \frac{f(x_1) - f(x_0)}{h} = \frac{0.0016 - 0.0001}{0.1}$$

$$= 0.015$$

The exact solution for the first derivative is

$$f(x) = x^4 \quad , f'(x) = 4x^3$$

$$f'(0.2) = 4(0.2)^3 = 0.032$$

Numerical error is

$$E = |D_B - f'(0.2)|$$

$$= |0.032 - 0.015| = 0.017$$

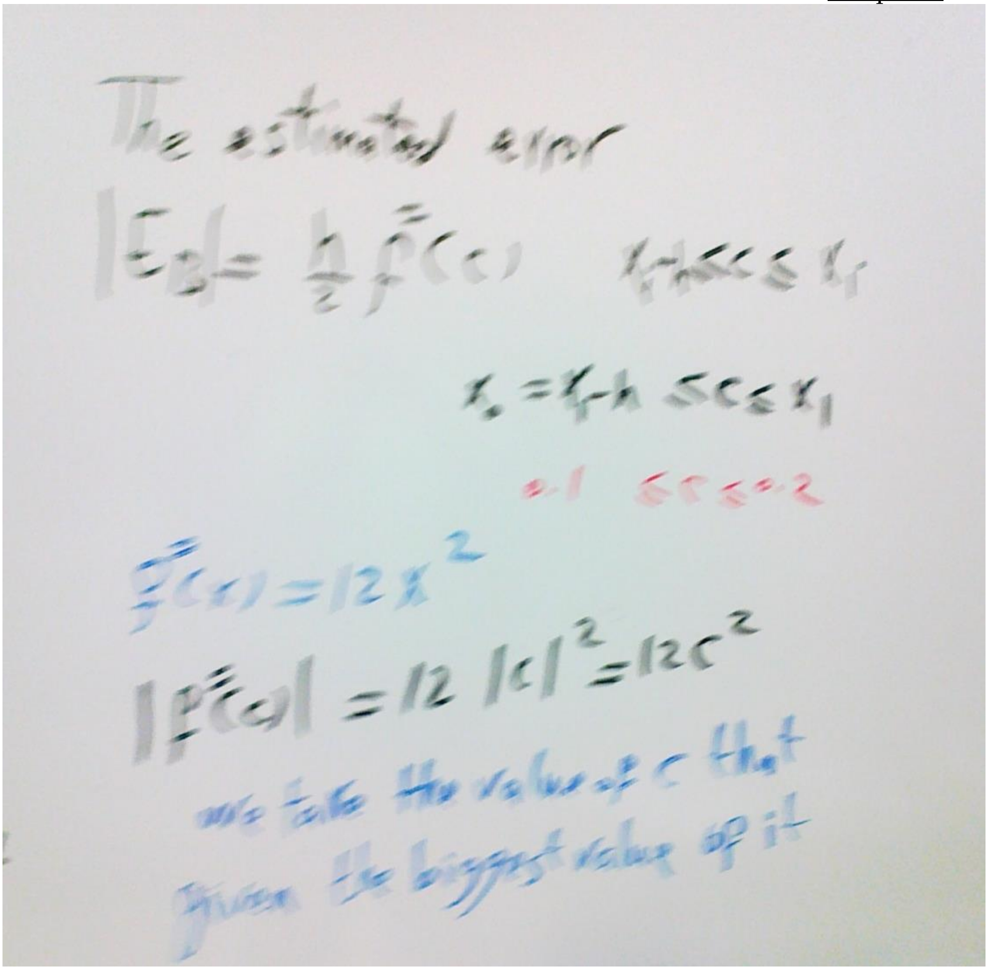
$$x_1 = 0.2, f(x_1 + h) = 0.0081, \ f(x_1 - h) = 0.0001, \ h = 0.1$$

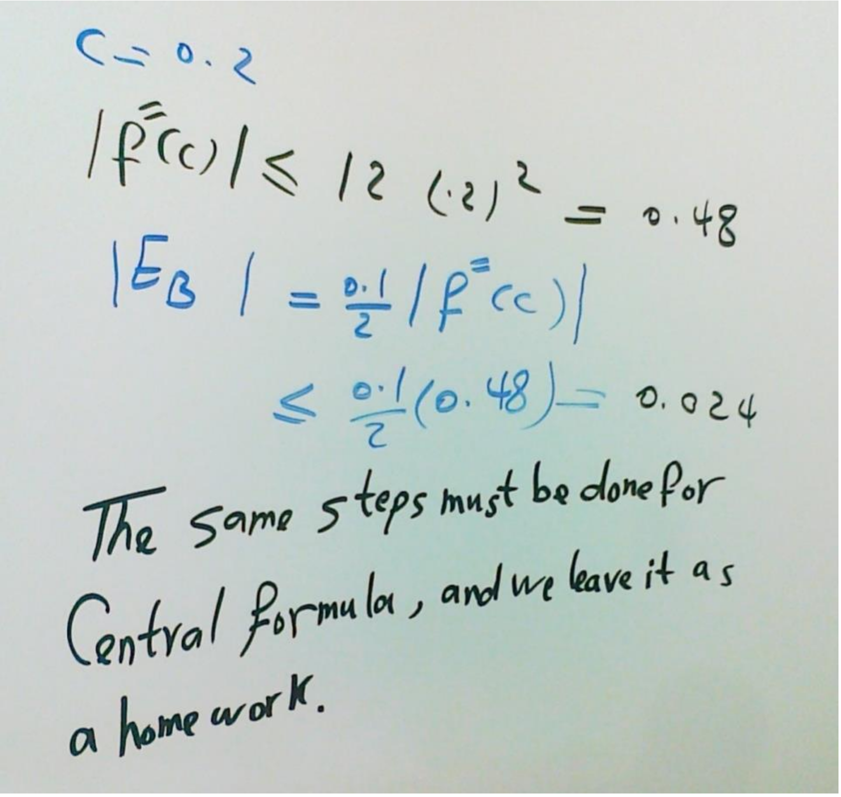
$$f'(x_1) \cong \frac{f(x_1 + h) - f(x_1 - h)}{2h} = \frac{0.0081 - 0.0.01}{2(0.1)} = \frac{0.0080}{0.2} = 0.04$$

$$f(x) = x^4 \Rightarrow f'(x) = 4x^3, \ f_e'(0.2) = 4(0.2)^3 = 0.032$$

$$|Error| = \left|f_e'(x_1) - f_a'(x_1)\right| = |0.04 - 0.032| = 0.08$$

$$f''(x) = 12x^2, \ f^{(3)}(x) = 24x$$


The estimated error
we take the value of c that
given the biggest value of it



$$\left|R\right| = \frac{h^2}{6}\left|f^{(3)}(c)\right| = \frac{(0.1)^2}{6}(24c) = 0.01*24c \le 0.24\,(0.1) = 0.024$$

Differentiation Using Interpolation

Use Lagrange interpolation formula to derive an approximation for the first derivative of $f(x)$ at a point belong to a set of three equally spaced nodes.

Let $P_n(x)$ be the degree n polynomial that interpolates $f(x)$ at n + 1 node points $x_0, x_1, \ldots, x_n$. To calculate f '(x) at some point x = t, use

$$f'(t) \approx P_n'(t) \qquad (6)$$

Many different formulas can be obtained by varying n and by varying the placement of the nodes $x_0, \ldots, x_n$.

Example. Take n = 2, and use equally spaced nodes
x_0, $x_1 = x_0 + h$, $x_2 = x_1 + h$.
Then

$$\begin{aligned} P_2(x) &= f(x_0)L_0(x) + f(x_1)L_1(x) + f(x_2)L_2(x) \\ P_2'(x) &= f(x_0)L_0'(x) + f(x_1)L_1'(x) + f(x_2)L_2'(x) \end{aligned}$$

with

$$L_0(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}$$
$$L_1(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}$$
$$L_2(x) = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

Forming the derivatives of these Lagrange basis functions and evaluating them at $x = x_1$

$$f'(x_1) \approx P_2'(x_1) = \frac{f(x_1+h) - f(x_1-h)}{2h} \equiv D_h f(x_1) \qquad (7)$$

For the error,

$$f'(x_1) - \frac{f(x_1+h) - f(x_1-h)}{2h} = -\frac{h^2}{6} f'''(c_2) \quad (8)$$

with x1 − h ≤ c2 ≤ x1 + h.

Undetermined Coefficients

Use Undetermined Coefficients to Derive an approximation for f ''(x). estimate the error.

Derive an approximation for f ''(x) at x = t. Write

$$f''(t) \approx D_h^{(2)} f(t) \equiv Af(t+h) + Bf(t) + Cf(t-h)$$

(1)

with A, B, and C are unspecified constants. Use Taylor polynomial approximations

$$f(t-h) \approx f(t) - hf'(t) + \frac{h^2}{2}f''(t) - \frac{h^3}{6}f'''(t) + \frac{h^4}{24}f^{(4)}(t) \qquad (2)$$

$$f(t+h) \approx f(t) + hf'(t) + \frac{h^2}{2}f''(t) + \frac{h^3}{6}f'''(t) + \frac{h^4}{24}f^{(4)}(t)$$

Substitute into (1) and rearrange:

$$D_h^{(2)} f(t) \approx (A+B+C)f(t)$$
$$+h(A-C)f'(t) + \frac{h^2}{2}(A+C)f''(t) \qquad (3)$$
$$+\frac{h^3}{6}(A-C)f'''(t) + \frac{h^4}{24}(A+C)f^{(4)}(t)$$

To have

$$D_h^{(2)} f(t) \approx f''(t) \qquad (4)$$

for arbitrary functions f (x), require

$$A+B+C=0: \quad \text{coefficient of } f(t)$$
$$h(A-C)=0: \quad \text{coefficient of } f'(t)$$
$$\frac{h^2}{2}(A+C)=1: \quad \text{coefficient of } f''(t)$$

Solution:

$$A=C=\frac{1}{h^2}, \qquad B=-\frac{2}{h^2} \qquad (5)$$

This determines

$$D_h^{(2)} f(t) = \frac{f(t+h)-2f(t)+f(t-h)}{h^2}$$

(6)

For the error, substitute (2) into (3):

$$D_h^{(2)} f(t) \approx f''(t) + \frac{h^2}{12} f^{(4)}(t)$$

Thus

$$f''(t) - \frac{f(t+h) - 2f(t) + f(t-h)}{h^2} \approx \frac{-h^2}{12} f^{(4)}(t)$$

(6)

Example. Let f (x) = cos(x), t = $\frac{1}{6}\pi$; use (14) to calculate f "(t) = − cos ($\frac{1}{6}\pi$)

h	$D_h^{(2)} f$	*Error*	*Ratio*
0.5	−0.84813289	−1.789E − 2	
0.25	−0.86152424	−4.501E − 3	3.97
0.125	−0.86489835	−1.127E − 3	3.99
0.0625	−0.86574353	−2.819E − 4	4.00
0.03125	−0.86595493	−7.048E − 5	4.00

5.2 differentiation for the Second order derivative

Example4:

Use Taylor's expansion to obtain different formulae for finite difference approximation of f "(x).

Answer:

Consider Taylor's expansion for function f(x) at a point x=a, so

$$f(x) = f(a) + (x-a)f^{(1)}(a) + \frac{1}{2}(x-a)^2 f^{(2)}(a) + \cdots + \frac{1}{n!}(x-a)^n f^{(n)}(a) \qquad (1)$$

Take $a = x_i$ and $x = x_i + h$, so $x - a = x_i + h - x_i = h$, Eq(1) become

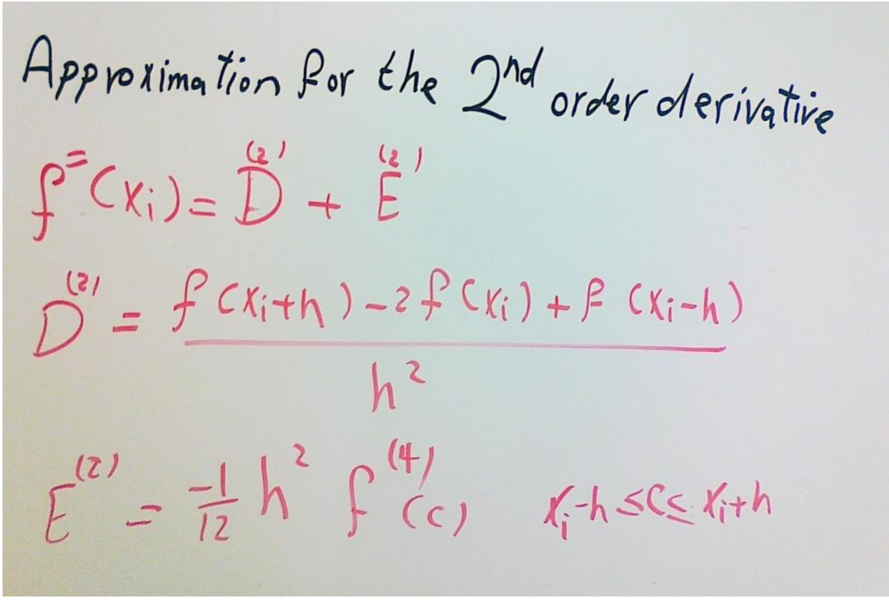
$$f(x_i + h) = f(x_i) + hf'(x_i) + \tfrac{1}{2!}h^2 f''(x_i) + \tfrac{1}{3!}h^3 f^{(3)}(x_i) + \tfrac{1}{4!}h^4 f^{(4)}(x_i) + \ldots. \quad \textbf{(2)}$$

Take $a = x_i$ and $x = x_i - h$, so $x - a = x_i - h - x_i = -h$, Eq(1) become

$$f(x_i - h) = f(x_i) - hf'(x_i) + \tfrac{1}{2!}h^2 f''(x_i) - \tfrac{1}{3!}h^3 f^{(3)}(x_i) + \tfrac{1}{4!}h^4 f^{(4)}(x_i) + \ldots. \quad \textbf{(3)}$$

We use Eq.(2)-(3) to obtain

$$f''(x_i) = \frac{f(x_i+h) - 2f(x_i) + f(x_i-h)}{h^2} - \frac{h^2}{12} f^{(4)}(c), x_i - h \le c \le x_i + h \ldots..(4)$$



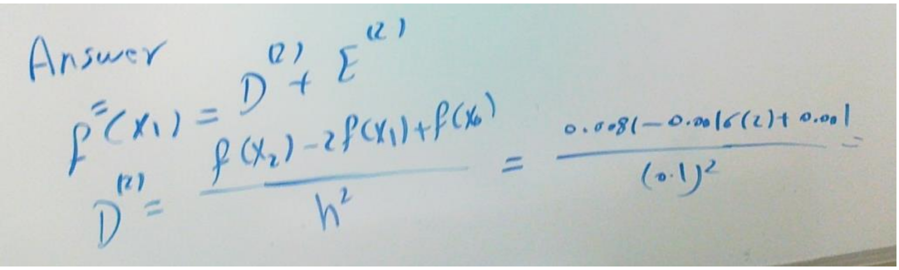
Example5:

Consider the function given by the following data

x	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
y	**0.0001**	**0.0016**	**0.0081**	**0.0256**	**0.0625**

Use difference formula for approximating $f''(x)$ at $x = 0.2$. Compute the numerical and estimated errors for each if $f(x) = x^4$.

Answer:



$$f''(x_i) = \frac{f(x_i+h) - 2f(x_i) + f(x_i-h)}{h^2} - \frac{h^2}{12} f^{(4)}(c), x_i - h \le c \le x_i + h$$

$$f''(0.2) = \frac{0.0081 - 2(0.0016) + 0.0001}{(0.1)^2} = \frac{0.0082 - 0.0032}{0.01} = \frac{0.005}{0.01} = 0.5$$

$$f''(x) = 12x^2 \text{ , } f_e''(0.2) = 12(0.2)^2 = 12(0.04) = 0.48$$

$$|Error| = |f_e'(x_1) - f_a'(x_1)| = |0.48 - 0.5| = 0.02$$

$$|R| = \frac{h^2}{12} f^{(4)}(c) = \frac{(0.1)^2}{12}(24) = 0.01 * 2 = 0.02$$

Programming home work

Write program for approximating

the first derivative with Central method
and the second order derivative.
test the program using

a) $f(x) = x^4$ b) $f(x) = e^x$

evaluate the numerical error in
each if we take

$h = 0.1,\ 0.01,\ 0.001$

Chapter 6

Numerical Integration

0-Introduction

The problem of evaluating definite integrals arises both in mathematics and beyond, in many areas of science and engineering. At some point in our mathematical education we all learned to calculate simple integrals such as

$\int_0^1 x^2 dx$ or $\int_0^{\pi} \cos x \, dx$

using a table of integrals, we obtain the values of them to be [3 $^{-1}$] and 0 respectively; but

Most of integration such as

$$\int_0^1 xe^{x^3} dx \quad , \quad \int_0^1 xe^{x^4} dx \quad , \quad \int_0^1 xe^{x^5} dx$$

Have no analytical solution.

Moreover , if the function is given as a table

x	**0.0**	**0.2**	**0.4**	**0.6**	**0.8**	**1.0**
$f(x)$	**0.0**	**0.04**	**0.16**	**0.36**	**0.64**	**1.0**

and $\int_0^1 f(x)\mathrm{d}x$ is required, and also in this case on analytic solution can be done.
In these cases , we turn to numerical integration

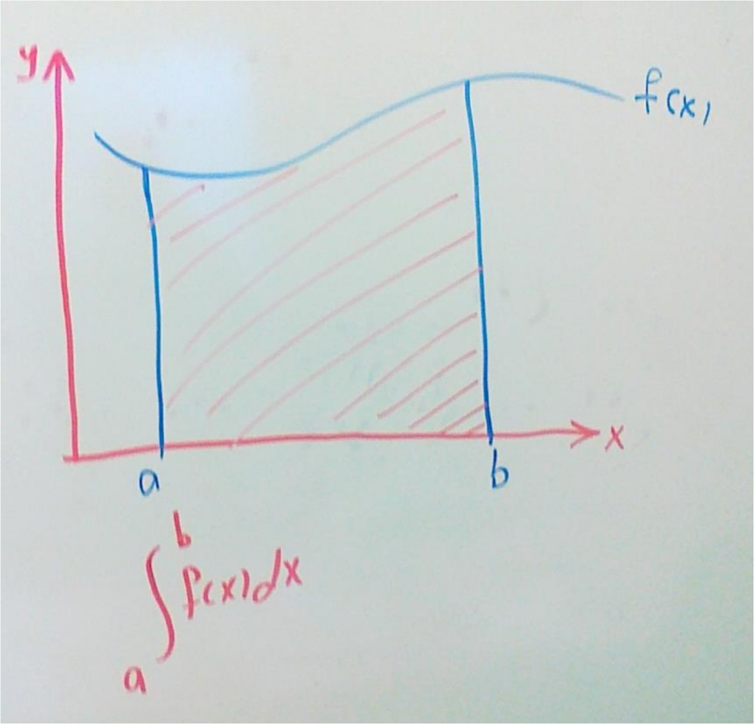
a continuous real-valued function $f(x)$ defined on a closed interval $[a, b]$ of the real line such that the definite integral

$$I = \int_a^b f(x)dx \qquad (1)$$

is very hard to reduce to an entry in the table of integrals by means of the usual tricks of variable substitution and integration by parts

The purpose of this chapter, is to answer this question. Specifically, we shall address the problem of evaluating approximately, by applying the results of polynomial interpolation to derive formulae for numerical integration (also called numerical quadrature rules).

6.1 Rules for single integrations



The physical meaning of $\int_a^b f(x)dx$ is the area between the lines x=a , x=b and between x axis and the curve $y = f(x)$

We draw a line between (a, f (a)) and (b, f (b)) and then take the area between this line and x as is as an approximation for the required integration

The Basic Trapezoidal Rule

The idea of Numerical integration depends on constructing Taylor polynomial approximations $p_n(x)$ as an interpolation for $f(x)$. So we have:

$$I = \int_a^b f(x)dx = \int_a^b p_n(x)dx \qquad (1)$$

The interval $[a, b]$ is divided into n subinterval making use of the points $x_0, x_1, x_2, ..., x_n$ in the given interval. These points are equidistance h. So we have

$$x_0 = a, x_i = x_0 + ih,\ i = 1, ..., n-1,\ x_n = b\ , \qquad (2)$$

The integration is evaluated using the function values at these points.

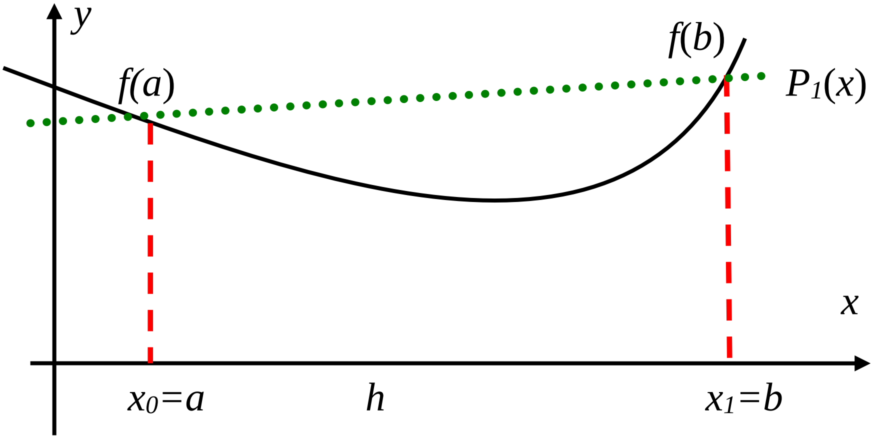
Linear Numerical Integration

In this case, we use polynomial approximations $p_1(x)$ of degree one as an interpolation for $f(x)$. The resulting rule is called the trapezoidal rule.

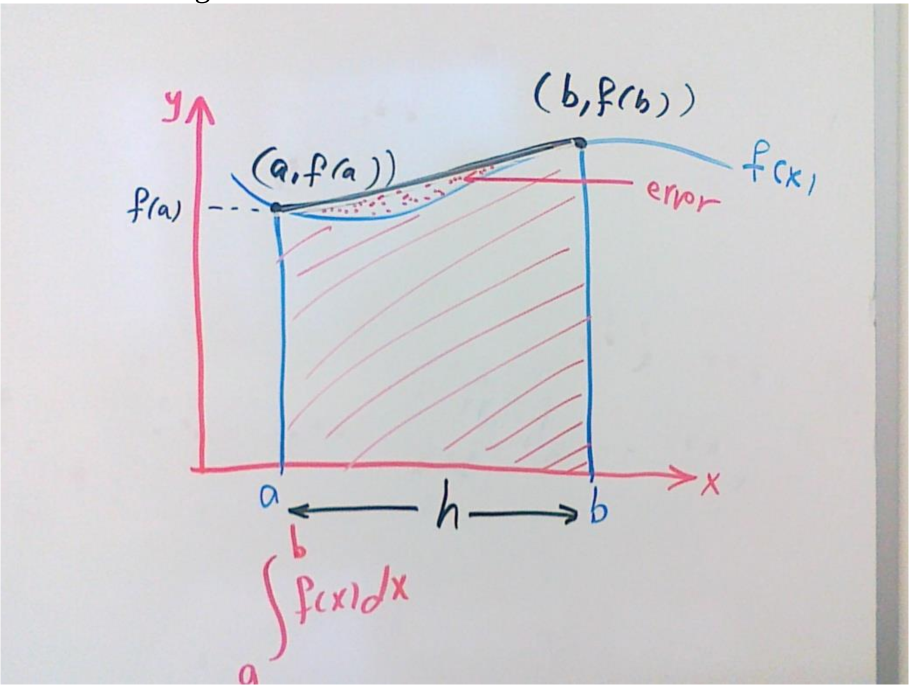
$$p_1(x) = \alpha + \beta x$$

The unknown constants α and β can be evaluated using the function values at the points:
$(a, f(a)), (b, f(b))$ (How?)

In the other hand, we approximate the area under the curve of $f(x)$ by the area under the segment pass through $(a, f(a)), (b, f(b))$.



Illustrating $I \approx T_1(f)$



The area of the trapezoidal is half of average base (the two bases are $f(a), f(b)$)multiplied on the altitude b-a.

Take h=b-a. Then the area of trapezoidal is

$T_1 = \frac{f(a)+f(b)}{2} h = \frac{h}{2}[f(a) + f(b)]$

Then the basic trapezoidal rule for $I = \int_a^b f(x)dx$ is

$$T_1 = \frac{f(a)+f(b)}{2} h = \frac{h}{2}[f(a) + f(b)] \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

Example:

Approximate $\int_0^1 \frac{dx}{1+x^2}$ using basic trapezoidal rule , then evaluate the numerical error.

Answer:

$$f(x) = \frac{1}{1+x^2}\ , a = 0\ , \qquad b = 1$$

f(a)=f(0)=1 f(b)=f(1)= $\frac{1}{2}$

h=b-a=1-0=1

$$T_1 = \frac{h}{2}[f(a) + f(b)]$$

$$T_1 = \frac{h}{2}[f(a) + f(b)] = \frac{1}{2}\left[1 + \frac{1}{2}\right] = 0.75$$

To obtain the numerical error we must have the exact solution first :

$$\frac{d}{dx}\tan^{-1} x = \frac{1}{1+x^2} \rightarrow \int \frac{dx}{1+x^2} = \tan^{-1} x + c$$

$$\sin 0 = 0 \qquad \cos 0 = 1$$

$$\tan 0 = \frac{\sin 0}{\cos 0} = 0$$

$$0 = \tan^{-1} \text{c}$$

$$\sin 45^0 = \sqrt[1]{2} = \cos 45^0$$

$$\tan 45^0 = \frac{\sin 45^0}{\cos 45^0} = 1$$

$$\frac{\pi}{4} = 45^0 = \tan^{-1} 1$$

$$\pi = 180^0 = \frac{22}{7}$$

$$I = \int_0^1 \frac{dx}{1+x^2} = \tan^{-1}(1) = \tfrac{\pi}{4} = 0.785714286$$

numerical error then is

$$E_1 = |I - T_1| = 0.785714286 - 0.75 = 0.035714286 = 3.5\times10^{-2}$$

Obtaining Greater Accuracy

To improve our estimate of the integral

$$I = \int_a^b f(x)\,dx$$

One direction is to increase the degree of the approximation, moving next to a quadratic interpolating polynomial for *f* (*x*). We first look at an alternative. Instead of using the trapezoidal rule on the original interval [a, b], apply it to integrals of *f* (*x*) over smaller subintervals. For example:

$$\begin{aligned} I &= \int_a^c f(x)\,dx + \int_c^b f(x)\,dx, \qquad c = \tfrac{b+a}{2} \\ &\approx \tfrac{c-a}{2}[f(a)+f(c)] + \tfrac{b-c}{2}[f(c)+f(b)] \\ &= \tfrac{h}{2}[f(a)+2f(c)+f(b)] \equiv T_2(f), \qquad h = \tfrac{b-a}{2} \end{aligned}$$

Example 2

divide the interval [0,1] into two halves then evaluate the integral of $\frac{1}{1+x^2}$ in each. Find an approximation for $\int_0^1 \frac{dx}{1+x^2}$. Evaluate the numerical error

Answer: since

$$\int_0^1 \frac{dx}{1+x^2} = \int_0^{\frac{1}{2}} \frac{dx}{1+x^2} + \int_{\frac{1}{2}}^1 \frac{dx}{1+x^2}$$

Then $h = \frac{1}{2}$ so applying

$$T_1 = \frac{h}{2}[f(a) + f(b)]$$

in each subinterval

$$T_1 = \frac{(\frac{1}{2})}{2}\left[f\ (0)+f\ \left(\tfrac{1}{2}\right)\right]+\frac{(\frac{1}{2})}{2}\left[f\ \left(\tfrac{1}{2}\right)+f\ (1)\right]$$

$$= \frac{(\frac{1}{2})}{2}\left[1+\frac{1}{1+\frac{1}{4}}\right]+\frac{(\frac{1}{2})}{2}\left[\frac{1}{1+\frac{1}{4}}+\frac{1}{2}\right]$$

$$= \frac{31}{40} = 0.775$$

$$I = \int_0^1 \frac{dx}{1+x^2} = \tan^{-1}(1) = \frac{\pi}{4} = 0.785714286$$

$$E = |I - T_1| = 0.785714286 - 0.775 = 0.010714286 = 1.1\times 10^{-2}$$

$$E_2 = 1.1 \times 10^{-2}$$

If we compare the error

$$E_1 > E_2$$

We conclude from the result of example 2 that dividing the interval and evaluating T_1 in each sub interaval will decrease the error

Working with this conclusion , we shall divide $[a, b]$ into n equally subintervals using the points

The Composite Trapezoidal Rule

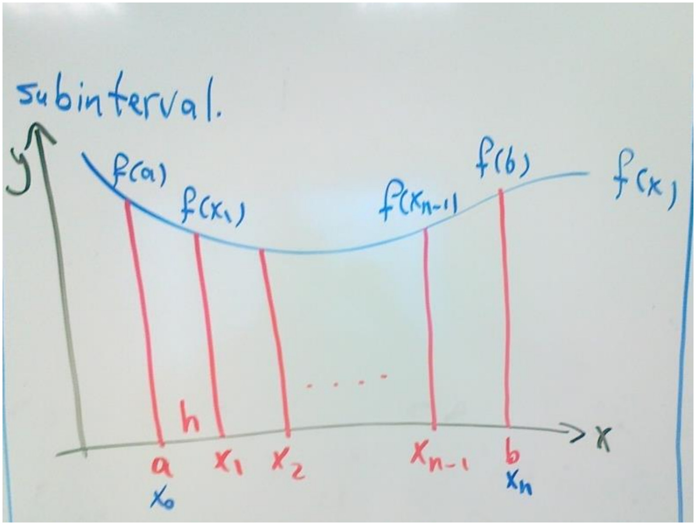
Derive composite trapezoidal rule for integration starting from linear interpolant

Consider approximating $\int_a^b f(x)dx$. We divide $[a, b]$to n subinterval with equally spaced points

$$x_i = a + ih \quad , h = \frac{b-a}{n} \; , i = 0,1, \dots , n$$

And then evaluate T_1 in each

$$T_1 = \frac{h}{2}[f(a) + f(b)]$$



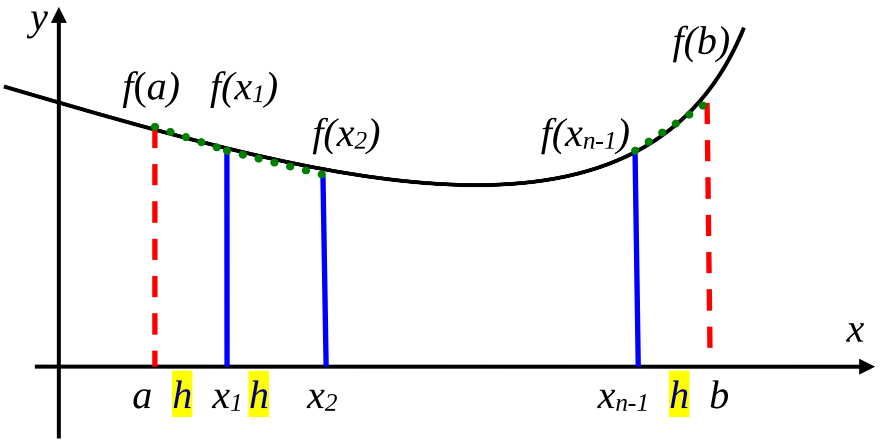
We can continue as above by dividing $[a, b]$ into even smaller subintervals and applying

$$\int_{\alpha}^{\beta} f(x)dx = \frac{h}{2}[f(\alpha) + f(\beta)]$$

on each of the smaller subintervals. Begin by introducing a positive integer n≥ 1,

Then

$$T_n = \int_{a}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \cdots + \int_{x_{n-1}}^{b} f(x)dx$$



Use $[\alpha, \beta] = [x_0, x_1], [x_1, x_2], ..., [x_{n-1}, x_n]$, for each of which the subinterval has length h.   Then applying

The numerical integration is approximated by summation of all trapezoidal resulting from passing lines between the points:

$(a, f(a)),\ (x_1, f(x_1)),\ (x_2, f(x_2)), ..., (x_{n-1}, f(x_{n-1})), (b, f(b))$

We have:

$$I = \int_a^b f(x)dx\ , T_1 = \frac{h}{2}[f(a) + f(b)], h = b - a$$

$$T_n = \int_a^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \cdots + \int_{x_{n-1}}^{b} f(x)dx$$

$$= \frac{h}{2}[f(a) + f(x_1)] + \frac{h}{2}[f(x_1) + f(x_2)] + \cdots + \frac{h}{2}[f(x_{n-1}) + f(b)]$$

$$= \frac{h}{2}\{f(a) + f(x_1) + f(x_1) + f(x_2) + \cdots + f(x_{n-1}) + f(b)\}$$

$$= \frac{h}{2}\{f(a) + 2f(x_1) + \cdots + 2f(x_{n-1}) + f(b)\}$$

$$= \frac{h}{2}\{f(a) + 2\sum_{i=1}^{n-1} F(x_i) + f(b)\} = T_n(f)$$

Thus

$$I = \int_a^b f(x)dx = T_n + R_T$$

Where

$$T_n = \frac{h}{2}\left[f(a) + f(b) + 2\sum_{i=1}^{n-1} f(x_i)\right]$$

and

$$R_T = -\frac{(b-a)h^2}{12} f''(c), a \le c \le b \qquad (4)$$

T_n is called composite trapezoidal rule

Example 3 :

Approximate $\int_0^1 f(x)dx$ using composite trapezoidal rule for the function $f(x)$ is given by the table

	a=x_0	x_1	x_2	x_3	x_4	$x_5 = x_n$ b
x	0	0.2	0.4	0.6	0.8	1
$f(x)$	0	0.04	0.16	0.36	0.64	1

n=5 $\frac{b-a}{n} = \frac{1}{n} = \frac{1}{5} = 0.2$

$$I = \int_a^b f(x)dx = T_n + R_T$$

Where

$$T_n = \frac{h}{2}\left[f(a) + f(b) + 2\sum_{i=1}^{n-1} f(x_i)\right]$$

$h = x_{i+1} - x_i = 0.2$

So

$T_5 = \frac{0.2}{2}[0 + 1 + 2(0.04 + 0.16 + 0.36 + 0.64)] = 0.34$

مثال (3):

استخدم قاعدة شبه المنحرف للتكامل في حساب $\int_0^1 f(x)dx$

للدالة $f(x)$ المعرفة بالجدول التالي:

x	0.0	0.2	0.4	0.6	0.8	1.0
$f(x)$	0.0	0.04	0.16	0.36	0.64	1.0

الحل:

في هذه المسألة لدينا n هي عدد النقاط، مخصوماً منها واحد،

وبالتالي:

n=5 $h = \frac{b-a}{n} = 0.2, h = x_{i+1} - x_i = 0.2$

وقانون شبه المنحرف في هذه المسألة يتعين من :

$T_n(f) = \frac{h}{2}\{f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b)\}$

وبالتالي:

$$T_5 = \frac{0.2}{2}[0 + 1 + 2(0.04 + 0.16 + 0.36 + 0.64)] = 0.34$$

Error formula for Trapezoidal rule

$$R_T = I - T_n = \frac{-(a-b)h^2}{12} f''(c), a \le c \le b$$

Example : Estimate the error for using Trapezoidal rule to obtain $\int_0^1 x^2 dx$, with n=5

Answer:

a=0 b=1 h=$\frac{a-b}{n} = \frac{1}{5} = 0.2$

$f(x) = x^2 \quad f'(x) = 2x \quad f''(x) = 2$

$|R_T| = \frac{1(0.2)^2}{12}$ (2) =0.00667

Consider approximating $\int_0^1 x^3 dx$ using composite trapezoidal rule. Estimate the error, with n=10

Answer:

The error formula is

$$R_T = \frac{-(b-a)h^2}{12} f''(c), a \leq c \leq b$$

$$f(x) = x^3 , f' = 3x^2 , f'' = 6x$$

$$a = 0 , b = 1 , h = \frac{b-a}{\pi} = \frac{1}{10} - 0.1$$

$$\mathrm{R_T} = \frac{-(1-0)(0.1)^2}{12} 6(\mathrm{c}), 0 \leq \mathrm{c} \leq 1$$

$$= \frac{-0.01}{2} c = -0.005c$$

$$|R_T| = 0.005c$$

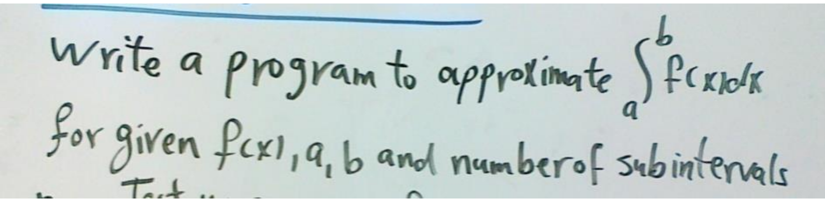
Since this value increases when c increases . we shall take the greatest value in [0.1] as an estimation

$|R_T| \leq 0.005(1)$

$|R_T| \leq 0.005$

Programming Exercise

Write



given $f(x)$,a ,b and number of subintervals n . test your program for

$f(x) = x^2$, $a = 0$, $b = n$, take n = 5,10,100,1000 and complete the table:

n	error
5	
10	
100	
100	

$$T_n = \frac{h}{2}\left[f(a) + f(b) + 2\sum_{i=1}^{n-1} f(x_i)\right], h = \frac{b-a}{n}$$

Algorithm for Trapezoidal rule

Input : f(x) as a function , $I = \int_a^b f(x)dx$

a,b,n

Output : T_n that proximates $\int_a^b f(x)dx$, the error

Sept1 : h=(b-a)/n

Step2 : T=f(a)+f(b)

Step3:for i=1 to n-1 do step4

Step 4 : $T = T + 2 * f(x_i)$

Step5 : T_n=T*h/2

Step6 : $E = |T_n - I|$

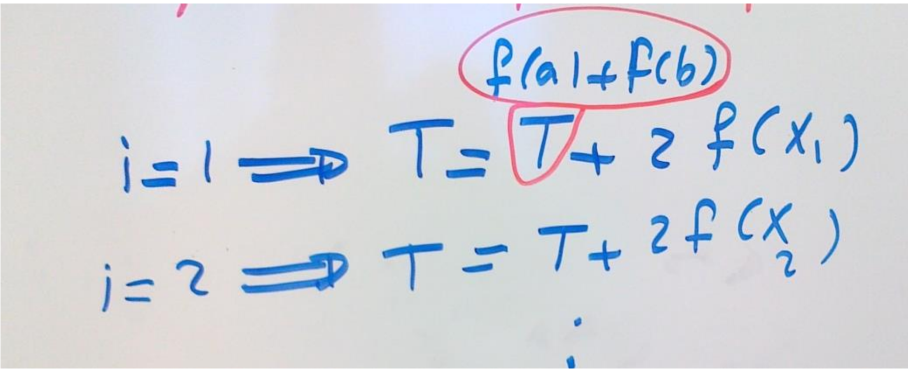
============

T =f(a)+f(b)

$i = 1 \rightarrow T = T + 2f(x_1)$

$i = 2 \rightarrow T = T + 2f(x_2)$

================



وإذا قمنا بعمل برنامج ماتلاب، وكتبنا فيه النص المقابل، والذي برمجنا فيه تكامل الدالة $f(x)=x^3$ باستخدام طريقة شبه المنحرف؛ فإن ناتج تشغيل البرنامج عند N=10 هو :

a=0.0;
b=1.0;
N=10;
h = (b - a)/N;
for i=1:N
x(i) = a +i*h;
end
f = @(x) x.^3;
fa=f(a);
fb=f(b);
sumf=0.0;
for i=1:N-1
sumf=sumf+f(x(i));
end
st='trapezoidal integraion is'
INTf = 0.5*h*(fa+fb+2*sumf)
	st=
	trapezoidal integraion is
	INTf
	= 0.2525
	
	------------------------------------ 	
	بينما ناتج تشغيل البرنامج عند N=100 هو
	------------------------------------ 	
	st=
	trapezoidal integraion is
	INTf=
	0.2500

مثال(4) : في المثال السابق ؛ إذا علم أن $f(x)=x^2$ فاحسب الخطأ في التقريب والخطأ المتوقع.

الحل: حيث أن $f(x)=x^2$ فإن :

$$I=\int_0^1 x^2 dx=\frac{1}{3}x^3\Big|_0^1=\frac{1}{3}(1)^3-0=\frac{1}{3}=0.3333$$

إذن الخطأ في التقريب.

$$|E| = |I - T_5| = |0.3333 - 0.0.34| = 0.00667$$

صيغة الخطأ المتوقع في قاعدة شبه المنحرف تتعين من

$$R_T = -(b-a)\frac{h^2}{12}f''(c)\ ,\ \ a \le c \le b$$

ولكن

$$f(x) = x^2 \qquad f' = 2x \qquad f'' = 2$$

$$|R| = |b-a|\frac{h^2}{12}|f''(c)| = (1-0)\frac{(0.2)^2}{12}(2)$$

$$= 1\frac{(0.04)}{12}(2) = \frac{0.02}{3} = 0.00667 = 6.6 \times 10^{-3}$$

Basic Simpson's Rule

Derive Simpson's rule for integration via quadratic interpolant

Consider $f(x)$ is defined at three points (a, f (a)) , $(x_1, f(x_1))$ and (b, f (b))

In this case, we use polynomial approximations $p_2(x)$ of degree two as an interpolation for $f(x)$. The resulting rule is called the trapezoidal rule.

$$p_2(x) = \alpha + \beta x + \gamma x^2$$

The unknown constants α, β and γ can be evaluated using the function values at the points:

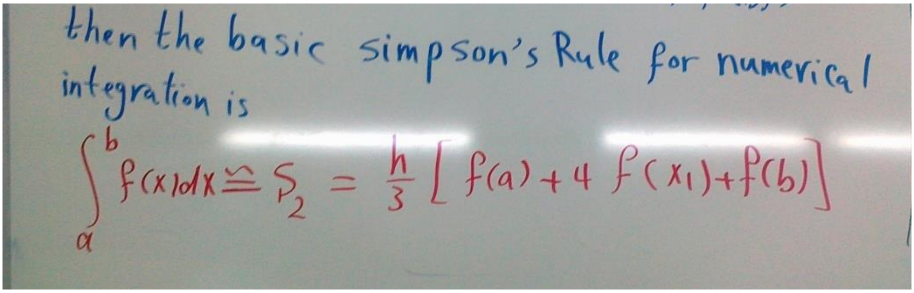
$$(a, f(a)), (x_1, f(x_1), (b, f(b))$$

In the other hand, we want to approximate $I = \int_a^b f(x)dx$ using quadratic interpolation of $f(x)$. Interpolate $f(x)$ at the points {a, c, b}, with $c = \frac{1}{2}(a+b)$. Also let $h = \frac{1}{2}(b-a)$. The quadratic interpolating polynomial is given by

$$P_2(x) = \frac{(x-c)(x-b)}{2h^2} f(a) + \frac{(x-a)(x-b)}{-h^2} f(c) + \frac{(x-a)(x-c)}{2h^2} f(b)$$

Replacing $f(x)$ by $P_2(x)$, we obtain the approximation

$$\int_a^b f(x)\,dx \approx \int_a^b P_2(x)\,dx = \frac{h}{3}[f(a) + 4f(c) + f(b)] \equiv S_2(f)$$



Then the basic Simpson's Rule for numerical integration is

$$\int_a^b f(x)dx \cong s_2 = \frac{h}{3}[f(a) + 4f(x_1) + f(b)]$$

Where $x_1 = \frac{a+b}{2}$

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

This is called Simpson's rule.

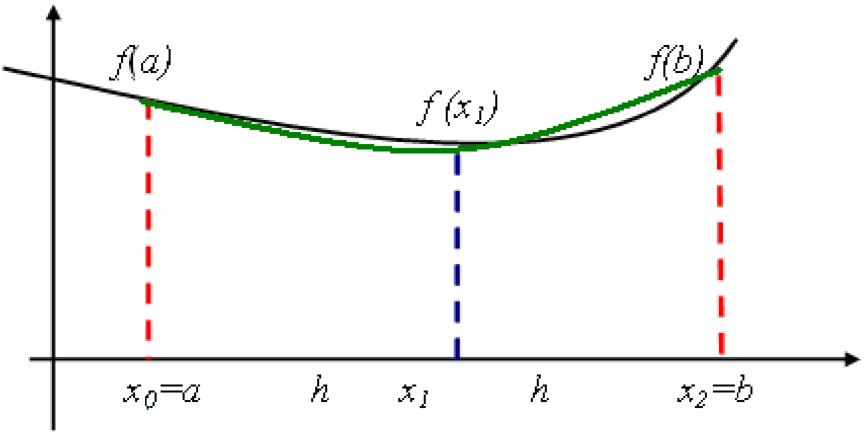


Illustration of I ≈ $S_2(f)$

شكل 6 -3:التكامل بطريقة شبه سيمبسون الأساسية

Approximate $\int_0^1 \frac{dx}{1+x^2}$ using basic Simpson's rule , then evaluate the numerical error.

$$f(a) = f(0) = \frac{1}{1+0} = 1$$

$$f(b) = f(1) = \frac{1}{1+1^2} = \frac{1}{2}$$

$$x_1 = \frac{a+b}{2} = \frac{0+1}{2} = \frac{1}{2}$$

$$f\left(\frac{1}{2}\right) = \frac{1}{1+(\frac{1}{2})^2} = \frac{4}{5}$$

$$h = \frac{b-a}{2} = \frac{1-0}{2} = \frac{1}{2}$$

The numerical value for integration is

$$S_2 = \frac{h}{3}\left[f(a) + 4f(x_1) + f(b)\right]$$

$h=(b-a)/2=1/2$

$$S_2 = \frac{h}{3}\left[f(a) + 4f\left(\tfrac{a+b}{2}\right) + f(b)\right] = \frac{\left(\frac{1}{2}\right)}{3}\left[1 + \frac{1}{1+\frac{1}{4}} + \frac{1}{2}\right]$$

$$= \tfrac{47}{60} = 0.783333$$

$$= \frac{1}{2(3)}\left[1 + 4\left(\frac{4}{5}\right) + \frac{1}{2}\right] = \frac{47}{60} = 0.78333$$

The exact solution is

$$I = \int_0^1 \frac{dx}{1+x^2} = \tan^{-1}(1) = \tfrac{\pi}{4} = 0.785714286$$

Hence the error is

$$E = |I - S_2| = |0.785714286 - 0.783333| = 0.02409 = 2.4\times10^{-2}$$

Composite Simpson’s Rule

Derive composite Simpson’s rule for integration via quadratic interpolant

We divide the interval $[a, b]$ to even number of intervals n using the points

$a = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = b$

With $h = \frac{b-a}{n}$ and $x_k = a + kh$

K=0,1,2,…,n

As with the trapezoidal rule, we can apply Simpson's rule on smaller subdivisions in order to obtain better accuracy in approximating

$$I = \int_a^b f(x)\,dx$$

Again, Simpson's rule is given by

$$\int_\alpha^\beta f(x)\,dx \approx \frac{\delta}{3}\left[f(\alpha) + 4f(\gamma) + f(\beta)\right], \quad \gamma = \frac{\alpha+\beta}{2}$$

and $\delta = \frac{1}{2}(\beta - \alpha)$.

Let n be a positive even integer, and

$$h = \frac{b-a}{n}, \qquad x_j = a + j\,h, \quad j = 0, 1, \ldots, n$$

Then write

$$\begin{aligned} I &= \int_{x_0}^{x_n} f(x)\,dx \\ &= \int_{x_0}^{x_2} f(x)\,dx + \int_{x_2}^{x_4} f(x)\,dx + \cdots + \int_{x_{n-2}}^{x_n} f(x)\,dx \end{aligned}$$

Apply

$$\int_{\alpha}^{\beta} f(x)\,dx \approx \frac{\delta}{3}\left[f(\alpha)+4f(\gamma)+f(\beta)\right], \quad \gamma=\frac{\alpha+\beta}{2}$$

to each of these subintegrals, with

$$[\alpha,\beta]=[x_0,x_2],[x_2,x_4],\ldots,[x_{n-2},x_n]$$

In all cases,(β − α)/2 = h. Then

$$\begin{aligned} I \approx & \tfrac{h}{3}\left[f(x_0)+4f(x_1)+f(x_2)\right] \\ & +\tfrac{h}{3}\left[f(x_2)+4f(x_3)+f(x_4)\right] \\ & +\cdots \\ & +\tfrac{h}{3}\left[f(x_{n-4})+4f(x_{n-3})+f(x_{n-2})\right] \\ & +\tfrac{h}{3}\left[f(x_{n-2})+4f(x_{n-1})+f(x_n)\right] \end{aligned}$$

This can be simplified to

$$\begin{aligned} \int_a^b f(x)\,dx \approx S_n(f) \equiv & \tfrac{h}{3}\left[f(x_0)+4f(x_1)\right. \\ & +2f(x_2)+4f(x_3)+2f(x_4) \\ & \left.+\cdots+2f(x_{n-2})+4f(x_{n-1})+f(x_n)\right] \end{aligned}$$

ولتحسين تقريب التكامل نقسم النطاق إلى n من الأقسام باستخدام النقاط

$$a=x_0,x_1,x_2,\ldots,x_{2k}=b, 2k=n$$

The composite Simpon Rule for integration then is

$$I=\int_a^b f(x)dx=S_n+R_S$$

Where

Odd values of the function

$$s_n = \frac{h}{3}[f(a) + 4\{f(x_1) + f(x_3) + \cdots + f(x_{n-1})\}$$

Even values of the function

$$+2[f(x_2) + f(x_4) + \cdots + f(x_{n-2})] + f(b)]$$

This is called the "***composite Simpson's rule***" or more simply, ***Simpson's rule***

Set n=2k

$$s_n = \frac{h}{3}[f(a) + 4\{f(x_1) + f(x_3) + \cdots + f(x_{2k-1})\}$$

$$+2[f(x_2) + f(x_4) + \cdots + f(x_{2k-2})] + f(b)]$$

And the error is obtained by

$$R_s = I - S_n = \frac{-(b-a)h^4}{180} f^{(4)}(c), a \le c \le b$$

Example:

(a) Approximate $\int_0^1 f(x)dx$ using composite Simpson's rule for the function f(x) is given by the table

x	**0.0**	**0.2**	**0.4**	**0.6**	**0.8**	**1.0**	**1.2**

$f(x)$	0.0	0.04	0.16	0.36	0.64	1.0	1.44

الحل:

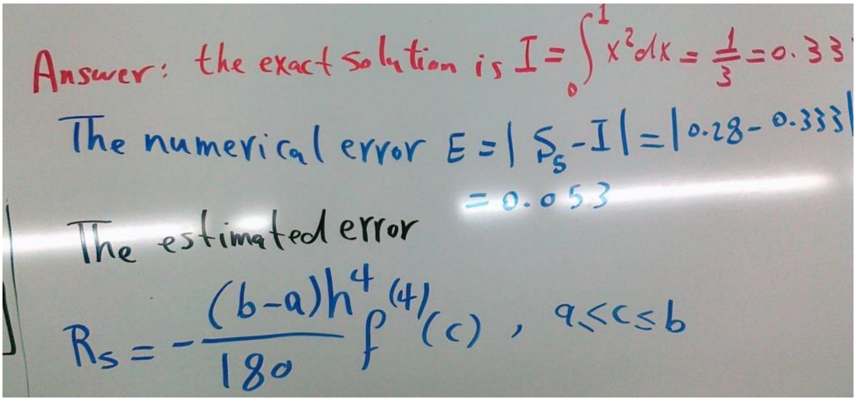
نظراً لأن طريقة سمبسون تميز القيمة الفردية عن الزوجية فسوف نحدد القيم كالتالي:

$$f(x_0) = f(a) = f(0) = 0,$$
$$f(x_6) = f(b) = f(1.2) = 1.44$$

القيم الفردية		القيم الزوجية	
k	$f(x_k)$	k	$f(x_k)$
1	0.04	2	0.16
3	0.36	4	0.64
5	1.0		
المجموع	1.40		0.80

$$S_6 = \frac{0.2}{3}[0 + 4\{1.4\} + 2\{0.80\} + 1.44]$$
$$= \frac{0.2}{3}[5.6 + 1.6 + 1.44]$$
$$= \frac{0.2}{3}[5.6 + 1.6 + 1.44] = \frac{0.2}{3}(8.64) = 0.567$$

(b) If $f(x) = x^2$ evaluate the numerical and estimated error.



الحل:

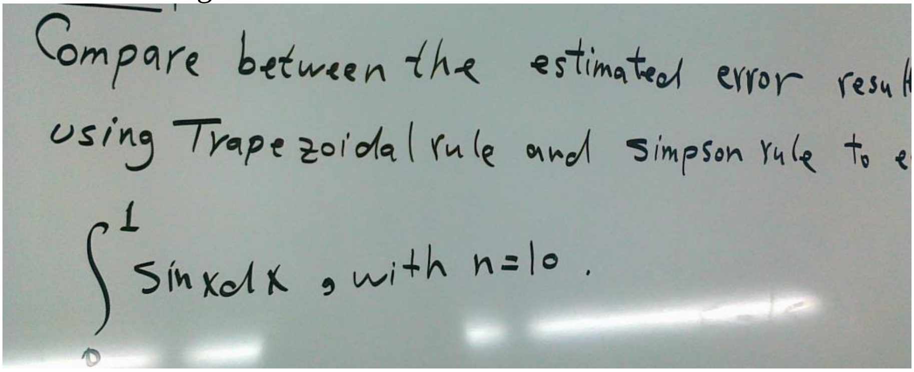
$$\int_0^{1.2} x^2 dx = \left.\frac{x^3}{3}\right|_0^{1.2} = \frac{(1.2)^3}{3} = 0.567$$

إذن الخطأ في التقريب

$$E = 0.567 - 0.567 = 0$$

$$R_s = \frac{-(b-a)h^4}{180} f^{(4)}(c)$$

وحيث أن $f^{(4)}(x) = 0$ إذن $|R| = 0$.



مثال (8):

قارن بين الخطأ المتوقع عند استخدام كلٍ من قاعدة شبه المنحرف وقاعدة سمبسون لتكامل الدالة $\int_0^{\frac{\pi}{2}} \sin(x)\, dx$ و $n = 10$

الحل:

$$f(x) = sin(x), \qquad [a,b] = \left[0, \frac{\pi}{2}\right], n = 10.$$

باستخدام قاعدة شبه المنحرف

$$R_T = \frac{-(a-b)h^2}{12} f''(c), a \le c \le b$$

$$f(x) = \sin x\, , f'(x) = \cos x\, , f''(x) = -\sin x.$$

$$|R_T| = \frac{(a-b)h^2}{12} |f''(c)|$$

$$h = \frac{b-a}{n} = \frac{\frac{\pi}{2} - 0}{n} = \frac{\pi}{2n}$$

$$|R_T| \le \frac{(0 - \frac{\pi}{2})(\frac{\pi}{2(10)})^2}{12} (1) = 0.02$$

و باستخدام قاعدة سمبسون

$$R_s = I - S_n = \frac{-(b-a)h^4}{180} f^{(4)}(c), a \le c \le b$$

$$f^{(3)} = -\cos x \qquad f^{(4)} = \sin x$$

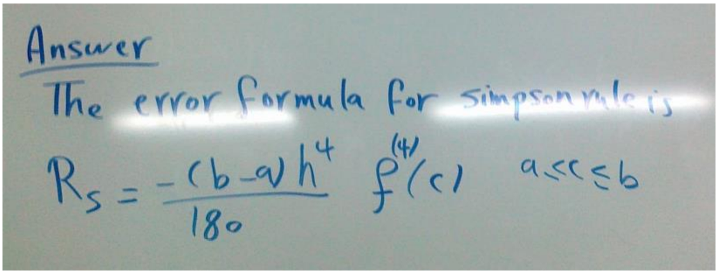
وحيث إن

$$|\sin x| < 1$$

$$|R_s| = \frac{(b-a)h^4}{180} \left| f^{(4)}(c) \right|$$

$$|R_s| \le \frac{\left(\frac{\pi}{2} - 0\right)}{180} \left(\frac{\pi}{2(10)}\right)^4 (1) = 5.3 * 10^{-6}$$

Consider using Simpson's rule to evaluate $\int_0^{\frac{\pi}{2}} \sin(x)\, dx$. Find n required to obtain an error within 10^{-5}.



الحل: صيغة الخطأ المتوقع لطريقة سمبسون تتعين من

$$\left|R\right| = \frac{(b-a)h^4}{180} f^{(4)}(c), \quad a \le c \le b$$

$$R_s = \frac{-(b-a)h^4}{180} f^{(4)}(c), a \le c \le b$$

وحيث إن الدالة المعطاة هي:

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

والتي تكون المشتقة الرابعة لها

$$\left|f^{(4)}(x)\right| \le 1$$

وحيث إن النطاق

$$[a,b] = \left[0, \frac{\pi}{2}\right]$$

إذن:

$$\left|R\right| \le \frac{\pi}{2(180)} h^4 = 0.00873h^4$$

وللحصول على دقة خمس مواضع عشرية نكتب

$$0.00873h^4 \le 0.000005$$

$$\therefore h^4 \le \frac{0.000005}{0.00873} = \frac{5}{8730} = 0.000573$$

وبأخذ الجذر التربيعي

$$h^2 \le 0.0239$$

وبأخذ الجذر التربيعي مرة أخرى

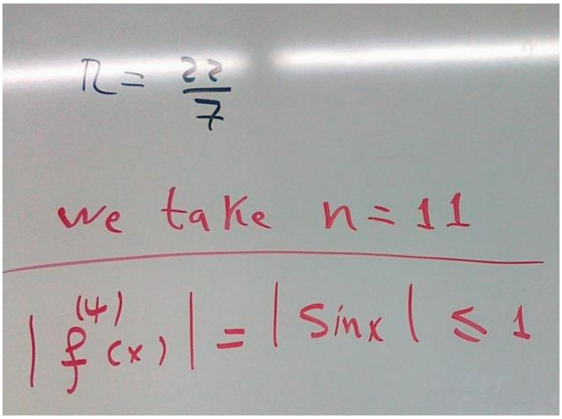
$$h \le 0.1547$$

$$\frac{b-a}{n} \le 0.1547$$

$$\frac{\pi}{2n} \le 0.1547$$

$$\frac{1}{n} \leq \frac{2(0.1547)}{\pi}$$

$$n \geq \frac{\pi}{2(0.1547)} = 10.5$$



$h^2 \leq 0.0239$

وبأخذ الجذر التربيعي مرة أخرى

$h \leq 0.1547$

$h = \frac{b-a}{n}$ => $n = \frac{b-a}{h}$ => $n \geq \frac{\pi}{2(0.15)} = \frac{22}{7(2)(0.15)} = 10.5$

وبالتالي فإن $n = 11$ سوف تعطي على الأقل دقة خمسة مواضع عشرية إذا استخدم سمبسون في التكامل العددي

Trapezoidal Error Formula Theorem

Let $f(x)$ have two continuous derivatives on the interval $a \le x \le b$. Then

$$E_n^T(f) \equiv \int_a^b f(x)\,dx - T_n(f) = -\frac{h^2(b-a)}{12} f''(c_n)$$

for some cn in the interval $[a, b]$.

The above formula says that the error decreases in a manner that is roughly proportional to h^2. Thus doubling n (and halving h) should cause the error to decrease by a factor of approximately 4. This is what we observed with a past example from the preceding section.

Example

Consider evaluating

$$I = \int_0^2 \frac{dx}{1+x^2}$$

using the trapezoidal method $T_n(f)$. How large should n be chosen in order to ensure that

$$\left|E_n^T(f)\right| \le 5 \times 10^{-6}$$

We begin by calculating the derivatives:

$$f'(x) = \frac{-2x}{\left(1+x^2\right)^2}, \qquad f''(x) = \frac{-2+6x^2}{\left(1+x^2\right)^3}$$

From a graph of $f''(x)$,

$$\max_{0 \le x \le 2} \left| f''(x) \right| = 2$$

Recall that b − a = 2. Therefore,

$$E_n^T(f) = -\frac{h^2(b-a)}{12} f''(c_n)$$

$$\left| E_n^T(f) \right| \le \frac{h^2 2}{12} \cdot 2 = \frac{h^2}{3}$$

We bound $|f''(c_n)|$ since we do not know cn, and therefore we must assume theworst possible case, that which makes the error formula largest. That is what has been done above.

$$\left| E_n^T(f) \right| \le 5 \times 10^{-6} \qquad (1)$$

To ensure this, we choose h so small that

$$\frac{h^2}{3} \le 5 \times 10^{-6}$$

This is equivalent to choosing h and n to satisfy

$$h \le .003873$$

$$n = \frac{2}{h} \ge 516.4$$

Thus n ≥ 517 will imply (1).

تمارين6-1

(1) للتكامل $I = \int_0^1 \frac{dx}{1+x^2}$ احسب T_4 و S_4 ثم احسب الخطأ العددي و الخطأ المتوقع و قارن بينهما.

(2) نفرض أن $f(0)=1,\ f(\tfrac{1}{2})=\tfrac{5}{2},\ f(1)=2,\ f(\tfrac{1}{4})=f(\tfrac{3}{4})=\alpha$

أوجد قيمة α إذا كانت صيغة شبه المنحرف تعطي $\int_0^1 f(x)dx=\tfrac{7}{4}$

(3) باستخدام البرنامج المعروض في هذا الفصل لصيغة شبه المنحرف كوّن جدول لقيم T_n حيث n=2,4,8,256 للتكاملات التالية ، و حيث أن الحل الصحيح متاح أمام كل تكامل ، أضف للجدول الخطأ العددي و الخطأ النسبي.

(a) $\int_0^{\pi} e^x \sin(x)dx=\tfrac{1}{2}\left(e^{\pi}+1\right)$ (b) $\int_0^1 x^{\frac{7}{3}}dx=\tfrac{3}{10}$

(c) $\int_0^{\pi} e^{\cos x}\sin x\,dx=e-\tfrac{1}{e}$

(d) $\int_0^3 \frac{dx}{1+(x-\pi)^2}=\tan^{-1}(3-\pi)+\tan^{-1}(\pi)$

(4) كرر التمرين (3) باستخدام برنامج طريقة سمبسون.

(5) قارن بين الخطأ المتوقع عند إستخدام كل من قاعدة شبه المنحرف و قاعدة سمبسون لتكامل الدالة

(i) $f(x)=x^7-4x^3$
(ii) $f(x)=1/(1+x)$

(6) استخدم صيغة شبه المنحرف حيث n=2,4,8,256 لحساب تقريب للمساحة تحت المنحنى $y=f(x)$ للدوال التالية ، في النطاق المعطى

(a) $\int_0^1 x^2\ln x\,dx$ (b) $\int_0^5 e^{-x^2}dx$

(c) $\int_3^9 \frac{2x}{x^2-5}dx$ (d) $\int_0^1 \sqrt{x}\,e^x dx$

(7) كرر التمرين (6) باستخدام طريقة سمبسون.

(8) إذا كان تطبيق صيغة شبه المنحرف للتكامل $\int_0^2 f(x)dx$ تعطي 4 ،

و عند تطبيق صيغة سيمبسون تعطي 2 فأوجد قيمة $f(1)$.

(9) احسب المسافة h اللازمة لحساب التكامل التالي صحيحا لأربعة

أرقام عشرية باستخدام قاعدة شبه المنحرف: $I = \int_1^2 dx / x$

6.2- Error Analysis for single numerical integration

Derive the error formula for the trapezoidal method **for numerical integrations then derive an asymptotic error estimate**

There are two stages in deriving the error:

(1) Obtain the error formula for the case of a single subinterval (n = 1);
(2) Use this to obtain the general error formula given earlier.

For the trapezoidal method with only a single subinterval, we have
From Trapezoidal Error Formula Theorem with *h* =(*b*-*a*)

$$\int_{\alpha}^{\alpha+h} f(x)\,dx - \frac{h}{2}\left[f(\alpha) + f(\alpha+h)\right] = -\frac{h^3}{12} f''(c)$$

for some c in the interval [α, α + h].

Recall that the general trapezoidal rule $T_n(f)$ was obtained by applying the simple trapezoidal rule to a subdivision of the original interval of integration. Recall defining and writing

$$h = \frac{b-a}{n}, \qquad x_j = a + j\,h, \quad j = 0, 1, \ldots, n$$

$$\begin{aligned} I &= \int_{x_0}^{x_n} f(x)\,dx \\ &= \int_{x_0}^{x_1} f(x)\,dx + \int_{x_1}^{x_2} f(x)\,dx + \cdots \\ &\qquad\qquad + \int_{x_{n-1}}^{x_n} f(x)\,dx \end{aligned}$$

$$\begin{aligned} I \approx\ & \tfrac{h}{2}\left[f(x_0) + f(x_1)\right] + \tfrac{h}{2}\left[f(x_1) + f(x_2)\right] \\ & + \cdots \\ & + \tfrac{h}{2}\left[f(x_{n-2}) + f(x_{n-1})\right] + \tfrac{h}{2}\left[f(x_{n-1}) + f(x_n)\right] \end{aligned}$$

Then the error

$$E_n^T(f) \equiv \int_a^b f(x)\,dx - T_n(f)$$

can be analyzed by adding together the errors over the subintervals $[x_0, x_1], [x_1, x_2], \ldots, [x_{n-1}, x_n]$. Recall

$$\int_{\alpha}^{\alpha+h} f(x)\,dx - \frac{h}{2}[f(\alpha) + f(\alpha+h)] = -\frac{h^3}{12}f''(c)$$

Then on $[x_{j-1}, x_j]$,

$$\int_{x_{j-1}}^{x_j} f(x)\,dx - \frac{h}{2}\left[f(x_{j-1}) + f(x_j)\right] = -\frac{h^3}{12}f''(\gamma_j)$$

with $x_{j-1} \leq \gamma_j \leq x_j$, but otherwise γj unknown.
Then combining these errors, we obtain

$$E_n^T(f) = -\frac{h^3}{12}f''(\gamma_1) - \cdots - \frac{h^3}{12}f''(\gamma_n)$$

This formula can be further simplified, and we will do so in two ways.
Rewrite this error as

$$E_n^T(f) = -\frac{h^3 n}{12}\left[\frac{f''(\gamma_1) + \cdots + f''(\gamma_n)}{n}\right]$$

Denote the quantity inside the brackets by ζn. This number satisfies

$$\min_{a \leq x \leq b} f''(x) \leq \zeta_n \leq \max_{a \leq x \leq b} f''(x)$$

Since $f''(x)$ is a continuous function (by original assumption), we have that there must be some number c_n in $[a, b]$ for which

$$f''(c_n) = \zeta_n$$

Recall also that $h_n = b - a$. Then

$$\begin{aligned} E_n^T(f) &= -\frac{h^3 n}{12}\left[\frac{f''(\gamma_1)+\cdots+f''(\gamma_n)}{n}\right] \\ &= -\frac{h^2(b-a)}{12} f''(c_n) \end{aligned}$$

This is the error formula given on the first slide.

An Error Estimate

We now obtain a way to estimate the error $E_n^T(f)$ Return to the formula

$$E_n^T(f) = -\frac{h^3}{12} f''(\gamma_1) - \cdots - \frac{h^3}{12} f''(\gamma_n)$$

and rewrite it as

$$E_n^T(f) = -\frac{h^2}{12}\left[f''(\gamma_1)h + \cdots + f''(\gamma_n)h\right]$$

The quantity

$$f''(\gamma_1)h + \cdots + f''(\gamma_n)h$$

is a Riemann sum for the integral

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

By this we mean

$$\lim_{n\to\infty}\left[f''(\gamma_1)h + \cdots + f''(\gamma_n)h\right] = \int_a^b f''(x)\,dx$$

Thus

$$f''(\gamma_1)h + \cdots + f''(\gamma_n)h \approx f'(b) - f'(a)$$

for larger values of n. Combining this with the earlier error formula we have

$$E_n^T(f) = -\frac{h^2}{12}\left[f''(\gamma_1)h + \cdots + f''(\gamma_n)h\right]$$

This is a computable estimate of the error in the numerical

$$E_n^T(f) \approx -\frac{h^2}{12}\left[f'(b) - f'(a)\right] \equiv \tilde{E}_n^T(f)$$

integration. It is called an ***asymptotic error estimate***.

Example.

Consider evaluating the integral $\int_a^b f(x)dx$ where $[a,b] = [0,\pi]$ and $f(x) = e^x \cos x$ by the trapezoidal rule. Prove that the error and the error estimate are quite close(assume that $|f''(x)| \leq 14.921$). Evaluate the corrected trapezoidal rule

Answer

Consider evaluating

$$I(f) = \int_0^\pi e^x \cos x\, dx = -\frac{e^\pi + 1}{2} \doteq -12.070346$$

In this case,

$$\begin{aligned} f'(x) &= e^x\left[\cos x - \sin x\right] \\ f''(x) &= -2e^x \sin x \\ \max_{0\le x\le\pi} \left|f''(x)\right| &= \left|f''(.75\pi)\right| = 14.921 \end{aligned}$$

Then

$$\begin{aligned} E_n^T(f) &= -\frac{h^2(b-a)}{12} f''(c_n) \\ \left| E_n^T(f) \right| &\leq \frac{h^2 \pi}{12} \cdot 14.921 = 3.906 h^2 \end{aligned}$$

Also

$$\begin{aligned} \tilde{E}_n^T(f) &= -\frac{h^2}{12} \left[f'(\pi) - f'(0) \right] \\ &= \frac{h^2}{12} \left[e^{\pi} + 1 \right] \doteq 2.012 h^2 \end{aligned}$$

For evaluating the integral I by the trapezoidal rule, we see that the error $E_n^T(f)$ and the error estimate $\tilde{E}_n^T(f)$ are quite close. Therefore

$$\begin{aligned} I(f) - T_n(f) &\approx \frac{h^2}{12} \left[e^{\pi} + 1 \right] \\ I(f) &\approx T_n(f) + \frac{h^2}{12} \left[e^{\pi} + 1 \right] \end{aligned}$$

This last formula is called the corrected trapezoidal rule. We see it gives a much smaller error for essentially the same amount of work; and it converges much more rapidly. In general,

$$I(f) - T_n(f) \approx -\frac{h^2}{12}\left[f'(b) - f'(a)\right]$$
$$I(f) \approx T_n(f) - \frac{h^2}{12}\left[f'(b) - f'(a)\right]$$

This is the corrected trapezoidal rule. It is easy to obtain from the trapezoidal rule, and in most cases, it converges more rapidly than the trapezoidal rule.

Simpson's Rule Error Formula

Recall the general Simpson's rule

$$\int_a^b f(x)\,dx \approx S_n(f) \equiv \tfrac{h}{3}\left[f(x_0) + 4f(x_1) + 2f(x_2)\right.$$
$$+4f(x_3) + 2f(x_4) + \cdots$$
$$\left.+2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)\right]$$

For its error, we have

$$E_n^S(f) \equiv \int_a^b f(x)\,dx - S_n(f) = -\frac{h^4\,(b-a)}{180} f^{(4)}(c_n)$$

for some $a \le c_n \le b$, with cn otherwise unknown. For an asymptotic error estimate,

$$\int_a^b f(x)\,dx - S_n(f) \approx \widetilde{E}_n^S(f) \equiv -\frac{h^4}{180}\left[f'''(b) - f'''(a)\right]$$

Discussion

For Simpson's error formula, both formulas assume that the integrand *f* (*x*) has four continuous derivatives on the interval [a, b]. What happens when this is not valid? We return later to this question.

Both formulas also say the error should decrease by a factor of around 16 when n is doubled. Compare these results with those for the trapezoidal
rule error formulas:.

$$E_n^T(f) \equiv \int_a^b f(x)\,dx - T_n(f) = -\frac{h^2\,(b-a)}{12} f''\,(c_n)$$

$$E_n^T(f) \approx -\frac{h^2}{12}\left[f'(b) - f'(a)\right] \equiv \widetilde{E}_n^T(f)$$

Example 6
Consider evaluating

$$I = \int_0^2 \frac{dx}{1+x^2}$$

using Simpson's rule $S_n(f)$. How large should n be chosen in order to ensure that

$$\left|E_n^S(f)\right| \le 5 \times 10^{-6}$$

Begin by noting that

$$f^{(4)}(x) \;=\; 24\frac{5x^4 - 10x^2 + 1}{\left(1+x^2\right)^5}$$

$$\max_{0\le x\le 1}\left|f^{(4)}(x)\right| \;=\; f^{(4)}(0) = 24$$

Then

$$E_n^S(f) = -\frac{h^4(b-a)}{180} f^{(4)}(c_n)$$

$$\left|E_n^S(f)\right| \le \frac{h^4 \cdot 2}{180} \cdot 24 = \frac{4h^4}{15}$$

Then $\left|E_n^S(f)\right| \le 5 \times 10^{-6}$ is true if

$$\frac{4h^4}{15} \le 5 \times 10^{-6}$$

$$h \le .0658$$

$$n \ge 30.39$$

Therefore, choosing n ≥ 32 will give the desired error bound. Compare this with the earlier trapezoidal example in which n ≥ 517 was needed.

For the asymptotic error estimate, we have

$$f'''(x) = -24x\frac{x^2-1}{\left(1+x^2\right)^4}$$

$$\tilde{E}_n^S(f) \equiv -\frac{h^4}{180}\left[f'''(2) - f'''(0)\right]$$

$$= \frac{h^4}{180} \cdot \frac{144}{625} = \frac{4}{3125}h^4$$

Integrating Sqrt(*x*)

Consider the numerical approximation of

$$\int_0^1 \mathrm{sqrt}(x)\,dx = \frac{2}{3}$$

In the following table, we give the errors when using both the trapezoidal and Simpson rules.

n	E_n^T	$Ratio$	E_n^S	$Ratio$
2	$6.311E-2$		$2.860E-2$	
4	$2.338E-2$	2.70	$1.012E-2$	2.82
8	$8.536E-3$	2.74	$3.587E-3$	2.83
16	$3.085E-3$	2.77	$1.268E-3$	2.83
32	$1.108E-3$	2.78	$4.485E-4$	2.83
64	$3.959E-4$	2.80	$1.586E-4$	2.83
128	$1.410E-4$	2.81	$5.606E-5$	2.83

The rate of convergence is slower because the function $f(x)$ =sqrt(x) is not sufficiently differentiable on [0, 1]. Both methods converge with a rate proportional to $h^{1.5}$.

ASYMPTOTIC ERROR FORMULAS

If we have a numerical integration formula,

$$\int_a^b f(x)\,dx \approx \sum_{j=0}^{n} w_j f(x_j)$$

let $E_n(f)$ denote its error,

$$E_n(f) = \int_a^b f(x)\,dx - \sum_{j=0}^{n} w_j f(x_j)$$

We say another formula $\tilde{E}_n(f)$ is an asymptotic error formula for this numerical integration if it satisfies

$$\lim_{n\to\infty} \frac{\widetilde{E}_n(f)}{E_n(f)} = 1$$

Equivalently,

$$\lim_{n\to\infty} \frac{E_n(f) - \widetilde{E}_n(f)}{E_n(f)} = 0$$

These conditions say that $\tilde{E}_n(f)$ looks increasingly like $E_n(f)$ as n increases, and thus

$$E_n(f) \approx \tilde{E}_n(f)$$

Example.

For the trapezoidal rule,

$$E_n^T(f) \approx \widetilde{E}_n^T(f) \equiv -\frac{h^2}{12}\left[f'(b) - f'(a)\right]$$

where $f(x)$ has two continuous derivatives on the interval [a, b].

Example.

For Simpson's rule,

$$E_n^S(f) \approx \widetilde{E}_n^S(f) \equiv -\frac{h^4}{180}\left[f'''(b) - f'''(a)\right]$$

where $f(x)$ has four continuous derivatives on the interval $[a, b]$.

Note that both of these formulas can be written in an equivalent form as

$$\tilde{E}_n(f) = \frac{c}{n^p}$$

for appropriate constant c and exponent p. With the trapezoidal rule, p = 2

$$p = 2, \quad c = -\frac{(b-a)^2}{12}\left[f'(b) - f'(a)\right]$$

$$\tilde{E}_n(f) = -\frac{(b-a)^2}{12(n^2)}\left[f'(b) - f'(a)\right]$$

$$h = \frac{b-a}{n} \quad \longrightarrow \quad n = \frac{b-a}{h}$$

$$\tilde{E}_n(f) = -\frac{\cancel{(b-a)^2}\, h^2}{12\,\cancel{(b-a)^2}}\left[f'(b) - f'(a)\right]$$

$$= -\frac{h^2}{12}\left[f'(b) - f'(a)\right]$$

and

$$c = -\frac{(b-a)^2}{12}\left[f'(b) - f'(a)\right]$$

and for Simpson's rule, p = 4 with a suitable c

The formula

$$\tilde{E}_n(f) = \frac{c}{n^p} \qquad (2)$$

occurs for many other numerical integration formulas that we have not yet defined or studied. In addition, if we use the trapezoidal or Simpson rules with an integrand $f(x)$ which is not sufficiently differentiable, then (2) may hold with an exponent p that is less than the ideal.

Application Of Asymptotic Error Formulas

Assume we know that an asymptotic error formula

E= $$I - I_n \approx \frac{c}{n^p}$$

is valid for some numerical integration rule denoted by I_n. Initially, assume we know the exponent p. Then imagine calculating both I_n and I_{2n}. With I_{2n}, we have

$$I - I_{2n} \approx \frac{c}{2^p n^p}$$

This leads to

$$-I_n = 2^p I - 2^p I_{2n} - I$$

$$= I(2^p - 1) - 2^p I_{2n}$$

$$I(2^p - 1) = 2^p I_{2n} - I_n$$

بإضافة وطرح I_{2n} في البسط

$$I = \frac{2^p I_{2n} - I_n}{2^p - 1} = \frac{(2^p - 1) I_{2n} + I_{2n} - I_n}{2^p - 1}$$

$$= I_{2n} + \frac{I_{2n} - I_n}{2^p - 1}$$

$$I - I_n \approx 2^p [I - I_{2n}]$$

$$I \approx \frac{2^p I_{2n} - I_n}{2^p - 1} = I_{2n} + \frac{I_{2n} - I_n}{2^p - 1}$$

The formula

$$I \approx I_{2n} + \frac{I_{2n} - I_n}{2^p - 1} \qquad (4)$$

is called Richardson's extrapolation formula.

Example.

With the trapezoidal rule and with the integrand $f(x)$ having two continuous derivatives,

$$I \approx T_{2n} + \frac{1}{3}[T_{2n} - T_n]$$

$(2^2-1=4-1=3)$

Example.

With Simpson's rule and with the integrand $f(x)$ having four continuous derivatives,

$$I \approx S_{2n} + \frac{1}{15}[S_{2n} - S_n]$$

$(2^4-1=16-1=15)$

We can also use the formula (4) to obtain error estimation formulas:

$$I - I_{2n} \approx \frac{I_{2n} - I_n}{2^p - 1} \qquad (5)$$

This is called Richardson's error estimate. For example, with the trapezoidal rule,

$$I - T_{2n} \approx \frac{1}{3}[T_{2n} - T_n]$$

6.3 Gaussian Numerical Integration

We look for numerical integration formulas

$$\int_{-1}^{1} f(x)\,dx \approx \sum_{j=1}^{n} w_j f(x_j)$$

which are to be exact for polynomials of as large degree as possible. There are no restrictions placed on the nodes $\{x_i\}$ nor the weights $\{w_i\}$ in working towards that goal. The motivation is that if it is exact for high degree polynomials, then perhaps it will be very accurate when integrating functions that are well approximated by polynomials.

There is no guarantee that such an approach will work. In fact, it turns out to be a bad idea when the node points $\{x_i\}$ are required to be evenly spaced over the interval of integration. But without this restriction on $\{x_i\}$ we are able to develop a very accurate set of quadrature formulas.

The case n = 1. We want a formula

(1)Evaluate Gaussian numerical integration formula when there is only one node.

(1b) prove that the midpoint integration rule is a spectral case of Gaussian numerical integration.

Answer:

We look for numerical integration formulas

$$w_1 f(x_1) \approx \int_{-1}^{1} f(x)\,dx$$

The weight w_1 and the node x_1 are to be so chosen that the formula is exact for polynomials of as large degree as possible.

To do this we substitute f (x) = 1 and f (x) = x. The first choice leads to

$$\begin{aligned} w_1 \cdot 1 &= \int_{-1}^{1} 1\,dx \\ w_1 &= 2 \end{aligned}$$

The choice f (x) = x leads to

$$\begin{aligned} w_1 x_1 &= \int_{-1}^{1} x\,dx = 0 \\ x_1 &= 0 \end{aligned}$$

The desired formula is

$$\int_{-1}^{1} f(x)\,dx \approx 2f(0)$$

We say it has degree of precision equal to 1. since it integrates exactly all linear polynomials. It is called the midpoint rule.

<u>The case n = 2. We want a formula</u>

(2) obtain 3 precision formula for Gaussian integration. Then use it to obtain $\int_{-1}^{1} f(x)dx$, where $f(x) = x^3$.

Answer:

$$w_1 f(x_1) + w_2 f(x_2) \approx \int_{-1}^{1} f(x)\, dx$$

The weights w_1, w_2 and the nodes x_1, x_2 are to be so chosen that the formula is exact for polynomials of as large a degree as possible. We substitute and force equality for

$$f(x) = 1, x, x^2, x^3$$

This leads to the system

$$\begin{aligned} w_1 + w_2 &= \int_{-1}^{1} 1\, dx = 2 \\ w_1 x_1 + w_2 x_2 &= \int_{-1}^{1} x\, dx = 0 \\ w_1 x_1^2 + w_2 x_2^2 &= \int_{-1}^{1} x^2\, dx = \frac{2}{3} \\ w_1 x_1^3 + w_2 x_2^3 &= \int_{-1}^{1} x^3\, dx = 0 \end{aligned}$$

The solution is given by

$$w_1 = w_2 = 1, \quad x_1 = \frac{-1}{\text{sqrt}(3)}, \quad x_2 = \frac{1}{\text{sqrt}(3)}$$

This yields the formula

$$\int_{-1}^{1} f(x)\, dx \approx f\left(\frac{-1}{\text{sqrt}(3)}\right) + f\left(\frac{1}{\text{sqrt}(3)}\right) \qquad (1)$$

We say it has degree of precision equal to 3 since it integrates exactly all polynomials of degree ≤ 3. We can verify directly that it does not integrate exactly f (x) = x^4.

$$\int_{-1}^{1} x^4 \, dx = \frac{2}{5}$$

$$f\left(\frac{-1}{\text{sqrt}(3)}\right) + f\left(\frac{1}{\text{sqrt}(3)}\right) = \frac{2}{9}$$

Thus (1) has degree of precision exactly 3.

EXAMPLE

For an Integrated function which is not in a polynomial form, we have as an example:

$$\int_{-1}^{1} \frac{dx}{3+x} = \log 2 \doteq 0.69314718$$

The formula (1) yields

$$\int_{-1}^{1} \frac{dx}{3+x} \, dx = \ln(3+x) \Big|_{-1}^{1}$$

$$= \ln(4) - \ln(2)$$

$$= 2\ln 2 - \ln 2 = \ln 2$$

$$f(x) = \frac{1}{3+x}$$

$$f\left(\frac{-1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right) = \frac{1}{3 - \frac{1}{\sqrt{3}}} + \frac{1}{3 + \frac{1}{\sqrt{3}}}$$

$$= \frac{\sqrt{3}}{3\sqrt{3} - 1} + \frac{\sqrt{3}}{3\sqrt{3} + 1} = 0.6923076$$

The General Case

We want to find the weights $\{w_i\}$ and nodes $\{x_i\}$ so as to have

$$\int_{-1}^{1} f(x)\,dx \approx \sum_{j=1}^{n} w_j f(x_j)$$

be exact for a polynomials f (x) of as large a degree as possible. As unknowns, there are n weights wi and n nodes x_i. Thus it makes sense to initially impose 2n conditions so as to obtain 2n equations for the 2n unknowns. We require the quadrature formula to be exact for the cases

لأن عدد المجاهيل 2n فإننا نحتاج لمثل هذا العدد من الشروط

$$f(x) = x^i, \qquad i = 0, 1, 2, ..., 2n - 1$$

Then we obtain the system of equations

$$w_1 x_1^i + w_2 x_2^i + \cdots + w_n x_n^i = \int_{-1}^{1} x^i\,dx$$

for i = 0, 1, 2, ..., 2n − 1. For the right sides,

$$\int_{-1}^{1} x^i \, dx = \begin{cases} \dfrac{2}{i+1}, & i = 0, 2, ..., 2n-2 \\ 0, & i = 1, 3, ..., 2n-1 \end{cases}$$

The system of equations

$$w_1 x_1^i + \cdots + w_n x_n^i = \begin{cases} \dfrac{2}{i+1}, & i = 0, 2, ..., 2n-2 \\ 0, & i = 1, 3, ..., 2n-1 \end{cases}$$

has a solution, and the solution is unique except for re-ordering the unknowns. The resulting numerical integration rule is called ***Gaussian quadrature***.

In fact, the nodes and weights are not found by solving this system. Rather, the nodes and weights have other properties which enable them to be found more easily by other methods. There are programs to produce them; and most subroutine libraries have either a program to produce them or tables of them for commonly used cases.

Symmetry Of Formula

The nodes and weights possess symmetry properties. In particular,

$$x_i = -x_{n-i}, \qquad w_i = w_{n-i}, \qquad i = 1, 2, ..., n$$

Change Of Interval Of Integration

Integrals on other finite intervals [a, b] can be converted to integrals over [−1, 1], as follows:

$$\int_a^b F(x)\,dx = \frac{b-a}{2}\int_{-1}^{1} F\left(\frac{b+a+t(b-a)}{2}\right) dt$$

based on the change of integration variables

$$x = \frac{b+a+t(b-a)}{2}, \qquad -1 \le t \le 1$$

EXAMPLE

Over the interval [0, π], use

$$x = (1+t)\,\tfrac{\pi}{2}$$

Then

$$\int_0^{\pi} F(x)\ dx = \tfrac{\pi}{2}\int_{-1}^{1} F\left((1+t)\,\tfrac{\pi}{2}\right)\ dt$$

EXAMPLE

Consider again the integrals used before :

$$\begin{aligned} I^{(1)} &= \int_0^1 e^{-x^2} dx \doteq .74682413281234 \\ I^{(2)} &= \int_0^4 \frac{dx}{1+x^2} = \arctan 4 \\ I^{(3)} &= \int_0^{2\pi} \frac{dx}{2+\cos x} = \frac{2\pi}{\mathrm{sqrt}(3)} \end{aligned}$$

n	$I - I^{(1)}$	$I - I^{(2)}$	$I - I^{(3)}$
2	$2.29E-4$	$-2.33E-2$	$8.23E-1$
3	$9.55E-6$	$-3.49E-2$	$-4.30E-1$
4	$-3.35E-7$	$-1.90E-3$	$1.77E-1$
5	$6.05E-9$	$1.70E-3$	$-8.12E-2$
6	$-7.77E-11$	$2.74E-4$	$3.55E-2$
7	$8.60E-13$	$-6.45E-5$	$-1.58E-2$
10	*	$1.27E-6$	$1.37E-3$
15	*	$7.40E-10$	$-2.33E-5$
20	*	*	$3.96E-7$

Compare these results with those before

An Error Formula

Let $f(x)$ be continuous for $a \le x \le b$; let $n \ge 1$. Then, for the Gaussian numerical integration formula

$$I \equiv \int_a^b f(x)\,dx \approx \sum_{j=1}^{n} w_j f(x_j) \equiv I_n$$

on [a, b], the error in In satisfies

$$|I(f) - I_n(f)| \le 2\,(b-a)\,\rho_{2n-1}(f) \qquad (3)$$

Here $\rho_{2n-1}(f)$ is the minimax error of degree $2n-1$ for $f(x)$ on $[a, b]$:

$$\rho_m(f) = \min_{\deg(p)\le m}\left[\max_{a\le x\le b}|f(x)-p(x)|\right], \quad m \ge 0$$

EXAMPLE

Let $f(x) = e^{-x^2}$. Then the minimax errors $\rho_m(f)$ are given in the following table.

m	$\rho_m(f)$	m	$\rho_m(f)$
1	5.30E − 2	6	7.82E − 6
2	1.79E − 2	7	4.62E − 7
3	6.63E − 4	8	9.64E − 8
4	4.63E − 4	9	8.05E − 9
5	1.62E − 5	10	9.16E − 10

Using this table, apply (3) to

$$I = \int_0^1 e^{-x^2}\, dx$$

For n = 3, (3) implies

$$|I - I_3| \leq 2\rho_5\left(e^{-x^2}\right) \doteq 3.24 \times 10^{-5}$$

The actual error is 9.55E − 6.

Weighted Gaussian Quadrature

Consider needing to evaluate integrals such as

$$\int_0^1 f(x) \log x\, dx, \qquad \int_0^1 x^{\frac{1}{3}} f(x)\, dx$$

How do we proceed? Consider numerical integration formulas

$$\int_a^b w(x) f(x)\, dx \approx \sum_{j=1}^{n} w_j f(x_j)$$

in which f (x) is considered a "nice" function (one with several continuous derivatives). The function w(x) is allowed to be singular, but must be integrable. We assume here that [a, b] is a finite interval. The function w(x) is called a “weight function”, and it is implicitly absorbed into the definition of the quadrature weights $\{w_i\}$. We again determine the nodes $\{x_i\}$ and weights $\{w_i\}$ so as to make the integration formula exact for f (x) a polynomial of as large a degree as possible. The resulting numerical integration formula

$$\int_a^b w(x)f(x)\,dx \approx \sum_{j=1}^{n} w_j f(x_j)$$

is called a Gaussian quadrature formula with weight function w(x). We determine the nodes $\{x_i\}$ and weights $\{w_i\}$ by requiring exactness in the above formula for

$$f(x) = x^i, \qquad i = 0, 1, 2, ..., 2n-1$$

To make the derivation more understandable, we consider the particular case

$$\int_0^1 x^{\frac{1}{3}} f(x)\,dx \approx \sum_{j=1}^{n} w_j f(x_j)$$

We follow the same pattern as used ear

The case n = 1. We want a formula

$$w_1 f(x_1) \approx \int_0^1 x^{\frac{1}{3}} f(x)\,dx$$

The weight w_1 and the node x_1 are to be so chosen that the formula is exact for polynomials of as large a degree as possible. Choosing f (x) = 1, we have

$$w_1 = \int_0^1 x^{\frac{1}{3}}\,dx = \frac{3}{4}$$

Choosing f (x) = x, we have

$$\begin{aligned} w_1 x_1 &= \int_0^1 x^{\frac{1}{3}} x\,dx = \frac{3}{7} \\ x_1 &= \frac{4}{7} \end{aligned}$$

Thus

$$\int_0^1 x^{\frac{1}{3}} f(x)\,dx \approx \frac{3}{4} f\left(\frac{4}{7}\right)$$

has degree of precision 1.

The case n = 2. We want a formula

$$w_1 f(x_1) + w_2 f(x_2) \approx \int_0^1 x^{\frac{1}{3}} f(x)\,dx$$

The weights w_1, w_2 and the nodes x_1, x_2 are to be so chosen that the formula is exact for polynomials of as large a degree as possible. We determine them by requiring equality for

$$f(x) = 1, x, x^2, x^3$$

This leads to the system

$$
\begin{aligned}
w_1 + w_2 &= \int_0^1 x^{\frac{1}{3}}\,dx = \tfrac{3}{4} \\
w_1x_1 + w_2x_2 &= \int_0^1 x x^{\frac{1}{3}}\,dx = \tfrac{3}{7} \\
w_1x_1^2 + w_2x_2^2 &= \int_0^1 x^2 x^{\frac{1}{3}}\,dx = \tfrac{3}{10} \\
w_1x_1^3 + w_2x_2^3 &= \int_0^1 x^3 x^{\frac{1}{3}}\,dx = \tfrac{3}{13}
\end{aligned}
$$

The solution is

$$x_1 = \tfrac{7}{13} - \tfrac{3}{65}\,\mathrm{sqrt}(35), \quad x_2 = \tfrac{7}{13} + \tfrac{3}{65}\,\mathrm{sqrt}(35)$$
$$w_1 = \tfrac{3}{8} - \tfrac{3}{392}\,\mathrm{sqrt}(35), \quad w_2 = \tfrac{3}{8} + \tfrac{3}{392}\,\mathrm{sqrt}(35)$$

Numerically,

$$x_1 = .2654117024, \quad x_2 = .8115113746$$
$$w_1 = .3297238792, \quad w_2 = .4202761208$$

The formula

$$\int_0^1 x^{\frac{1}{3}} f(x)\,dx \approx w_1 f(x_1) + w_2 f(x_2) \qquad (4)$$

has degree of precision 3.

EXAMPLE

Consider evaluating the integral

$$\int_0^1 x^{\frac{1}{3}} \cos x \, dx \qquad (5)$$

In applying (4), we take f (x) = cosx. Then

$$w_1 f(x_1) + w_2 f(x_2) = 0.6074977951$$

The true answer is

$$\int_0^1 x^{\frac{1}{3}} \cos x \, dx \doteq 0.6076257393$$

and our numerical answer is in error by E_2= .000128.

This is quite a good answer involving very little computational effort (once the formula has been determined). In contrast, the trapezoidal and Simpson rules applied to (5) would converge very slowly because the first derivative of the integrand is singular at the origin.

6.4 Numerical approximation of Singular integrals

يكون التكامل معتل singular إذا تحقق أحد الشرطين:

(أ) أن يكون للدالة المكاملة نقطة انفصال داخل نطاق التكامل مثل:

$$\int_{-3}^{3}\frac{dx}{x^2},\ \int_{1}^{2}\frac{dx}{x-1},\ \int_{0}^{\pi}\tan x\,dx$$

(ب) أن يكون نطاق التكامل غير محدود مثل:

$$\int_{1}^{+\infty}\frac{dx}{x^2},\ \int_{-\infty}^{+\infty}\frac{dx}{1+x^2},\ \int_{-\infty}^{0}e^{x}dx$$

والطرق العددية لحساب التكامل العددي تفترض أن التكامل غير معتل non singular أي يتحقق أن:

الدالة المكاملة نظامية ،أي :

1 -منتهية finite. 2 -قابلة للتفاضل. 3 - متصلة.

وذلك في نطاق التكامل .وبالتالي يمكن تمثيل الدالة بمتسلسلة تيلور في هذا النطاق.

(2) فترة التكامل منتهية. و بالتالي فإن التكامل يكون على الصورة:

$$\int_{a}^{b} f(x)dx,\quad a,b\in R$$

و يجب أولا تحويل التكامل معتل إلى تكامل غير معتل ثم تطبيق إحدى الطرق العددية لحساب قيمة تقريبية للتكامل.

و فيما يلي نذكر طرق تحويل التكاملات معتلة إلى تكاملات غير معتلة

6 -2 -1 تكامل الدوال ذات نقاط الانفصال

وفيها تكون الدالة المكاملة غير محدودة عند بعض قيم x ، وعند ذلك نقوم بإزالة الاعتلال بإحدى الطرق التالية :

1 - التعويض

مثال(1) :

استخدم التعويض في إزالة إعتلال التكامل $I=\int_0^1 \frac{\cos x}{\sqrt{x}}dx$

الحل: هذا التكامل معتل عند $x=0$ فباستخدام التعويض $x=t^2$ نجد أن

$$I=\int_0^1 \frac{\cos t^2}{t}.2tdt=2\int_0^1 \cos t^2 dt$$

وهو تكاملا غير معتل.

2 - استخدام متسلسلة تيلور

مثال(2): وضح كيف يمكن لاستخدام متسلسلة تيلور أن يزيل إعتلال التكامل $I=\int_0^1 \frac{\sin x}{x}dx$

الحل: هذا التكامل معتل عند $x=0$ فباستخدام متسلسلة تيلور نجد أن

$$\sin(x)=\sum_{k=1}^{\infty}\frac{(-1)^{k-1}x^{2k-1}}{(2k-1)!}$$

إذن

$$I=\int_0^1 \frac{\sin x}{x}dx \quad =\int_0^1 \frac{1}{x}\sum_{k=1}^{\infty}\frac{(-1)^{k-1}x^{2k-1}}{(2k-1)!}dx$$

$$=\int_0^1 \sum_{k=1}^{\infty}\frac{(-1)^{k-1}x^{2k-2}}{(2k-1)!}dx$$

وهو تكاملا غير معتل.

3 -التكامل التجزئ

مثال(3) : استخدم التكامل التجزئ في إزالة إعتلال التكامل

$$I=\int_0^1 \frac{\cos x}{\sqrt{x}}dx$$

الحل: هذا التكامل معتل عند $x=0$ فباستخدام التكامل التجزئ نجد أن

$$I=\left[\frac{x^{\frac{1}{2}}\cos x}{\frac{1}{2}}\right]_0^1-\int_0^1 2\sqrt{x}.(-\sin x)dx$$

$$=2\cos 1-2\int_0^1 2\sqrt{x}\,\sin x\,dx$$

وهو تكاملا غير معتل.

4 - إذا كانت نقطة الاعتلال داخل النطاق فإنه يلزم تقسيم التكامل إلى تكاملين بتقسيم النطاق إلى جزئين عند نقطة الاعتلال ثم التعامل مع كل نقطة على حده:

$$\int_{-\infty}^{+\infty} f(x)dx = \int_{-\infty}^{c} f(x)dx + \int_{c}^{+\infty} f(x)dx$$

$$\int_{a}^{b} f(x)dx = \int_{a}^{c} f(x)dx + \int_{c}^{b} f(x)dx \,,\, f(c) \notin \mathbb{R}$$

حيث c عدد حقيقي

6 -2 -2 التكاملات على فترة غير محدودة

وفيه تكون فترة التكامل لا نهائية

مثال(4) :

وضح كيف يمكن أن تزيل إعتلال التكامل $I = \int_0^{\infty} \frac{e^{-x}}{xe^{-2x}+1}dx$

الحل: نستخدم التعويض $x = -ln(t) \rightarrow t = e^{-x}$

$$I = \int_1^0 \frac{t}{[-ln(t)]t^2+1}\left(\frac{-1}{t}\right)dt = \int_0^1 \frac{dt}{1-t^2 ln(t)}$$

وحيث أن $\lim_{t\rightarrow 0} t\ ln(t) = 0$

فإن هذا التكامل غير معتل.

مثال(5) : حدد أسباب اعتلال التكامل التالي ثم بين كيف يمكن إزالتها

$$\int_0^{+\infty}\frac{dx}{\sqrt{x}\,(x+1)}$$

الحل: التكامل هنا معتل لسببين: أولاً لأن النطاق غير محدود, وثانياً لأن

الحد الأسفل يؤدي إلى اللانهاية. من التعريف السابق سوف نقسم التكامل

لجزئين عند $x=1$ فنحصل على

$$\int_0^{+\infty}\frac{dx}{\sqrt{x}\,(x+1)}=\int_0^{1}\frac{dx}{\sqrt{x}\,(x+1)}+\int_1^{+\infty}\frac{dx}{\sqrt{x}\,(x+1)}$$

و لحل التكامل دون اعتبار الحدود فإننا نأخذ التعويض

$$x=u^2,\ \ dx=2udu$$

فيكون

$$\int_0^{+\infty}\frac{dx}{\sqrt{x}\,(x+1)}=2\int_0^{1}\frac{du}{u^2+1}+2\int_1^{+\infty}\frac{du}{u^2+1}$$

و حيث أن الدالة العكسية للدالة المكاملة في الجزئين هي $\tan^{-1}u$ معرفة

عند $0,1\text{and}\,\infty$ ، فإنه يتضح أنه قد تم إزالة الاعتلال.

مثال(6) : أوجد عدد النقاط n اللازم استخدامها للحصول على تقريب للتكامل

$$\int_0^{1}\frac{\sin x}{x}dx$$

باستخدام قاعدة شبه المنحرف بدقة 5×10^{-6}

الحل:

حيث أن $f(x)=\dfrac{\sin x}{x}$ غير معينة عند $x=0$ فهذا التكامل معتل، و

لتحويله إلى تكامل غير معتل نستخدم متسلسلة تيلور:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + ...$$

$$f(x) = \frac{\sin x}{x} = 1 - \frac{x^2}{3!} + \frac{x^4}{5!} - \frac{x^6}{7!} + ...$$

$$f'(x) = \frac{-2x}{6} + \frac{4x^3}{5!} - \frac{6x^5}{7!} + ...$$

$$f''(x) = -\frac{1}{3} + \frac{4(3)x^2}{1\bullet2\bullet3\bullet4\bullet5} - \frac{6(5)x^4}{1\bullet2\bullet3\bullet4\bullet5\bullet6\bullet7} + ...$$

$$= -\frac{1}{3} + \frac{x^2}{10} - \frac{x^4}{168} + ...$$

وحيث أن $0<x<1$

$$|f''(x)| = \left|-\frac{1}{3} + \frac{x^2}{10} - \frac{x^4}{168} + ...\right|$$

فإنه عند $x=0$ نجد أن $|f''(0)| = \dfrac{1}{3} = 0.3333$

وعند $x=1$ نجد أن

$$|f''(1)| = \left|-\frac{1}{3} + \frac{1}{10} - \frac{1}{168} + ...\right|$$

واضح أن هذه القيمة لن تتخطى $0.5 = \frac{1}{2} >$ إذن الحد الأقصى لمقياس المشتقة من المرتبة الثانية هو

$$|f''(x)| < \frac{1}{2}$$

وبالتالي يكون تقدير الخطأ

$$|R| = \frac{(b-a)}{12} h^2 |f''(c)|$$

$$\leq \frac{h^2}{12} \bullet \frac{1}{2} = \frac{h^2}{24}$$

وحيث أن الدقة المطلوبة هي 5×10^{-6}

$$\frac{h^2}{24} \leq 5 \times 10^{-6} \Rightarrow h^2 \leq 24(5) \times 10^{-6} = 120 \times 10^{-6}$$

$$\therefore h \leq 1.095 \times 10^{-3}$$

وحيث أن $h = \frac{b-a}{n} = \frac{1}{n} \Rightarrow n > \frac{1}{h} = 91.287$

وبالتالي يمكن إختيار $n = 92$

تمارين6 -2

(1) حدد أسباب اعتلال التكامل التالي ثم بين كيف يمكن إزالتها

(i) $\int_1^4 \frac{dx}{(x-2)^{2/3}}$ **(ii)** $I = \int_1^\infty \frac{\ln x}{x^2} dx$

(iii) $\int_0^3 \frac{dx}{[x-1]^{\frac{2}{3}}}$ (iv) $\int_0^{+\infty} (1-x)e^{-x} dx$

(2) ناقش إتصال الدالة

$$f(x) = \begin{cases} x^3+1, & 0 \le x \le 0.1 \\ 1.001+0.03(x-0.1)+0.3(x-0.1)^2+2(x-0.1)^3 & 0.1 \le x \le 0.2 \\ 1.009+0.15(x-0.2)+0.9(x-0.2)^2+2(x-0.2)^3 & 0.2 \le x \le 0.3 \end{cases}$$

ثم استخدم كل من طريقتي شبه المنحرف و سمبسون لإيجاد تقريب لـ

$\int_0^{0.3} f(x) dx$ أوجد حداً للخطأ المتوقع في كل حالة.

(3) وضح كيف يمكن لاستخدام متسلسلة تيلور أن يزيل إعتلال التكامل

$$\int_0^{\pi/2} \frac{\cos x dx}{\sqrt{x}} dx$$

ثم احسب قيمته عدديا.

6.5 Multiple Integrals

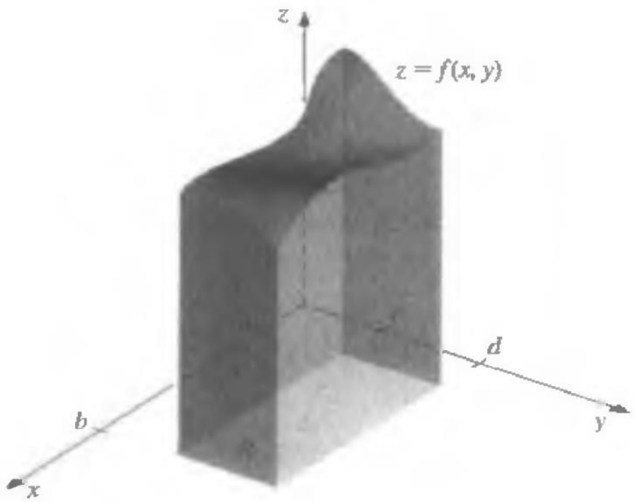
The techniques discussed in the previous sections can be modified in a straightforward manner for use in the approximation of multiple integrals. Let us first consider The double integral

$$\iint_R f(x, y)\, dA,$$

where R is a rectangular region in the plane:

$$R = \{(x, y) \mid a \le x \le b, c \le y \le d\},$$

for some constant and *b,* c, *and d* (see Figure). We will employ the Composite Simpson's rule to illustrate the approximation technique. although any other approximation formula could be used without major modifications.



Suppose that even integers n and m are chosen to determine the step sizes h = (b -a)/n and k = (d - c)/m. We first write the double integral as an iterated integral.

$$\iint\limits_R f(x,y)\,dA = \int_a^b \left(\int_c^d f(x,y)\,dy \right) dx,$$

and use the Composite Simpson's rule to approximate

$$\int_c^d f(x,y)\,dy,$$

treating x as a constant. Let $y_j = c + jk$ for each j = 0,1,.. . .m. Then

$$\int_c^d f(x,y)\,dy = \frac{k}{3}\left[f(x,y_0) + 2\sum_{j=1}^{(m/2)-1} f(x,y_{2j}) + 4\sum_{j=1}^{m/2} f(x,y_{2j-1}) + f(x,y_m) \right] - \frac{(d-c)k^4}{180}\frac{\partial^4 f(x,\mu)}{\partial y^4},$$

for some, μ in (c, d). Thus

$$\int_a^b \int_c^d f(x,y)\,dy\,dx = \frac{k}{3}\left[\int_a^b f(x,y_0)\,dx + 2\sum_{j=1}^{(m/2)-1} \int_a^b f(x,y_{2j})\,dx + 4\sum_{j=1}^{m/2} \int_a^b f(x,y_{2j-1})\,dx + \int_a^b f(x,y_m)\,dx \right] - \frac{(d-c)k^4}{180}\int_a^b \frac{\partial^4 f(x,\mu)}{\partial y^4}\,dx.$$

Composite Simpson's rule is now employed on each integral in this equation. Let
$x_i = a + ih$ for each i =0,1,2.. ,n. Then for each j = 0,1,… ,m, we have

$$\int_a^b f(x,y_j)\,dx = \frac{h}{3}\left[f(x_0,y_j) + 2\sum_{i=1}^{(n/2)-1} f(x_{2i},y_j) + 4\sum_{i=1}^{n/2} f(x_{2i-1},y_j) + f(x_n,y_j) \right] - \frac{(b-a)h^4}{180}\frac{\partial^4 f}{\partial x^4}(\xi_j,y_j),$$

for some ξ_j in (a, b). The resulting approximation has the form

$$\int_a^b \int_c^d f(x,y)\,dy\,dx \approx \frac{hk}{9}\Bigg\{\Bigg[f(x_0,y_0) + 2\sum_{i=1}^{(n/2)-1} f(x_{2i},y_0) + 4\sum_{i=1}^{n/2} f(x_{2i-1},y_0) + f(x_n,y_0)\Bigg]$$

$$+ 2\Bigg[\sum_{j=1}^{(m/2)-1} f(x_0,y_{2j}) + 2\sum_{j=1}^{(m/2)-1}\sum_{i=1}^{(n/2)-1} f(x_{2i},y_{2j}) + 4\sum_{j=1}^{(m/2)-1}\sum_{i=1}^{n/2} f(x_{2i-1},y_{2j}) + \sum_{j=1}^{(m/2)-1} f(x_n,y_{2j})\Bigg]$$

$$+ 4\Bigg[\sum_{j=1}^{m/2} f(x_0,y_{2j-1}) + 2\sum_{j=1}^{m/2}\sum_{i=1}^{(n/2)-1} f(x_{2i},y_{2j-1}) + 4\sum_{j=1}^{m/2}\sum_{i=1}^{n/2} f(x_{2i-1},y_{2j-1}) + \sum_{j=1}^{m/2} f(x_n,y_{2j-1})\Bigg]$$

$$+ \Bigg[f(x_0,y_m) + 2\sum_{i=1}^{(n/2)-1} f(x_{2i},y_m) + 4\sum_{i=1}^{n/2} f(x_{2i-1},y_m) + f(x_n,y_m)\Bigg]\Bigg\}.$$

The error term. E. is given by

$$E = \frac{-k(b-a)h^4}{540}\Bigg[\frac{\partial^4 f(\xi_0,y_0)}{\partial x^4} + 2\sum_{j=1}^{(m/2)-1}\frac{\partial^4 f(\xi_{2j},y_{2j})}{\partial x^4} + 4\sum_{j=1}^{m/2}\frac{\partial^4 f(\xi_{2j-1},y_{2j-1})}{\partial x^4} + \frac{\partial^4 f(\xi_m,y_m)}{\partial x^4}\Bigg] - \frac{(d-c)k^4}{180}\int_a^b \frac{\partial^4 f(x,\mu)}{\partial y^4}\,dx.$$

If $\partial^4 f/\partial x^4$ is continuous. the Intermediate Value Theorem and Mean Value Theorem for Integrals can be used to show that the error formula can be simplified to

$$E = \frac{-k(b-a)h^4}{540}\left[3m\frac{\partial^4 f}{\partial x^4}(\overline{\eta},\overline{\mu})\right] - \frac{(d-c)k^4}{180}\int_a^b \frac{\partial^4 f(x,\mu)}{\partial y^4}\,dx,$$

for some $(\overline{\eta},\overline{\mu})$ in R.If $\partial^4 f/\partial y^4$ is also continuous, the Weighted Mean Value Theorem for Integrals implies that

$$\int_a^b \frac{\partial^4 f(x,\mu)}{\partial y^4}\,dx = (b-a)\frac{\partial^4 f}{\partial y^4}(\hat{\eta},\hat{\mu}),$$

for some $(\hat{\eta},\hat{\mu})$ in R. Since m = (d - c)/k, the error term has the form

$$E = \frac{-k(b-a)h^4}{540}\left[3m\frac{\partial^4 f}{\partial x^4}(\overline{\eta},\overline{\mu})\right] - \frac{(d-c)(b-a)}{180}k^4\frac{\partial^4 f}{\partial y^4}(\hat{\eta},\hat{\mu})$$

or

$$E = -\frac{(d-c)(b-a)}{180}\left[h^4\frac{\partial^4 f}{\partial x^4}(\overline{\eta},\overline{\mu}) + k^4\frac{\partial^4 f}{\partial y^4}(\hat{\eta},\hat{\mu})\right],$$

for some $(\overline{\eta},\overline{\mu})$ and $(\hat{\eta},\hat{\mu})$ in $\boldsymbol{R}$.

Example:
Evaluate the error when simpson' s rule is used to approximate the double integral

$$\int_0^{\frac{1}{2}}\int_0^{\frac{1}{2}} e^{y-x}\,dy\,dx$$

With n=m=4

Answer: $d-c=\frac{1}{2}$, $b-a=\frac{1}{2}$

$$h=k=\frac{\frac{1}{2}}{(4)}=\frac{1}{8}$$

$$f(x,y)=e^{y-x} \qquad f_y=e^{y-x}=f_{yy}=f_{yyy}=f_{yyyy}$$

$$f_x=-e^{y-x} \qquad f_{xx}=+$$

$$f_{xxx}=- \qquad f_{xxxx}=e^{y-x}$$

$$\frac{\partial^4 f(\bar{x},\bar{y})}{\partial x^4}=e^{\bar{y}-\bar{x}}=\frac{\partial^4 f(\bar{x},\bar{y})}{\partial y^4}$$

$$\bar{x},\bar{y}\in[0,\tfrac{1}{2}]$$

* Use Trapezoidal rule to Find

$$\int_a^b\int_c^d f(x,y)\,dy\,dx.$$

Solution :-

Let $R = [a,b] \times [c,d]$ be a Rectangle

i.e $R = \{(x,y) : a \leq x \leq b , c \leq y \leq d\}$

we divide $[a,b]$ and $[c,d]$ into even n and m respectively.

The mesh points will be $x_0, x_1, \ldots, x_n$ and $y_0, y_1, \ldots, y_m$ with step size

$$h = \frac{b-a}{n}, \qquad k = \frac{d-c}{m} \ldots$$

we first use Trapezoidal rule to approximate

$$\int_c^d f(x,y)\,dy$$

with treating x as a constant.

we have $y \in [c,d]$, $k = \frac{d-c}{m}$

$y_j = c + jk$, $j = 0, 1, \ldots, m$.

$$\therefore \int_c^d f(x,y)\,dy = \frac{k}{2}\left[f(x,y_0) + f(x,y_m) + 2\sum_{j=1}^{m-1} f(x,y_j)\right] - \frac{(d-c)}{12} k^2 \frac{\partial^2 f(x,\bar{y})}{\partial y^2}$$

$$\bar{y} \in [c,d]$$

$$\therefore \int_a^b \left[\int_c^d f(x,y)\,dy\right] dx = \frac{k}{2}\left[\int_a^b f(x,y_0)\,dx + \int_a^b f(x,y_m)\,dx + 2\sum_{j=1}^{m-1}\int_a^b f(x,y_j)\,dx\right]$$

$$- \frac{(d-c)}{12} k^2 \int_a^b \frac{\partial^2}{\partial y^2} f(x,\bar{y})\,dx$$

Now, we employ the Trapezoidal rule in each integral where $x \in [a,b]$

$$h = \frac{b-a}{n} \quad , \quad x_i = a + ih \quad , \quad i = 0, 1, \ldots, n$$

$$\therefore \int_a^b f(x, y_j)\, dx = \frac{h}{2}\Big[f(x_0, y_j) + f(x_n, y_j) + 2\sum_{i=1}^{n-1} f(x_i, y_j)\Big] - \frac{(b-a)}{12} h^2 \frac{\partial^2}{\partial x^2} f(\bar{x}_j, y_j)$$

where $a \le \bar{x}_j \le b \qquad j = 0, 1, \ldots, m.$

$$\therefore \int_a^b \int_c^d f(x,y)\, dy\, dx = \frac{Kh}{4}\Big[\Big\{ f(x_0, y_0) + f(x_n, y_0) + 2\sum_{i=1}^{n-1} f(x_i, y_0)\Big\} + \Big\{ f(x_0, y_m) + f(x_n, y_m) + 2\sum_{i=1}^{n-1} f(x_i, y_m)\Big\} + 2\sum_{j=1}^{m-1}\Big\{ f(x_0, y_j) + f(x_n, y_j) + 2\sum_{i=1}^{n-1} f(x_i, y_j)$$

$$+ E$$

where
$$E = -\frac{K(b-a)h^2}{24}\Big[\frac{\partial^2 f}{\partial x^2}(\bar{x}_0, y_0) + \frac{\partial^2 f}{\partial x^2}(\bar{x}_m, y_m) + 2\sum_{j=1}^{m-1}\frac{\partial^2 f}{\partial x^2}(\bar{x}_j, y_j)\Big]$$

$$-\frac{(d-c)}{12} K^2 \int_a^b \frac{\partial^2 f}{\partial y^2}(x, \bar{y})\, dx.$$

rewrite E as following.

$$E = \frac{-K(b-a)h^2}{24}\left[2m\left(\frac{\frac{\partial^2 f}{\partial x^2}(\bar{x}_0,\bar{y}_0)+\frac{\partial^2 f}{\partial x^2}(\bar{x}_m,y_m)+2\sum_{j=1}^{m-1}\frac{\partial^2 f}{\partial x^2}(\bar{x}_j,y_j)}{2m}\right)\right]$$

$\longleftarrow \varepsilon_{n,m} \longrightarrow$

$$-\frac{(d-c)}{12}K^2\int_a^b \frac{\partial^2 f}{\partial y^2}(x,\bar{y})\,dx.$$

if $\frac{\partial^2 f}{\partial x^2}$ and $\frac{\partial^2 f}{\partial y^2}$ are continuous then

by intermediate value theorem and Mean value theorem $\exists\ (\bar{x},\bar{y}) \in R$ such that

$$\int_a^b \frac{\partial^2 f}{\partial y^2}(x,\bar{y})\,dx = (b-a)\frac{\partial^2 f}{\partial y^2}(\bar{x},\bar{y}).$$

and since $\varepsilon_{n,m}$ satisfies that

$$\min \frac{\partial^2 f}{\partial x^2}(x,y) \leq \varepsilon_{n,m} \leq \max \frac{\partial^2 f}{\partial x^2}(x,y)$$

where $(x,y) \in [a,b] \times [c,d]$

So,

$$\varepsilon_{n,m} = \frac{\partial^2 f}{\partial x^2}(\bar{x},\bar{y})$$

So,

$$E = \frac{-K(b-a)h^2}{24}\left(2m\frac{\partial^2 f}{\partial x^2}(\bar{x},\bar{y})\right)$$

$$-\frac{(d-c)}{12}K^2(b-a)\frac{\partial^2 f}{\partial y^2}(\bar{x},\bar{y})$$

but $m = \frac{d-c}{k}$

$$\therefore \quad E = -\frac{(b-a)(d-c)h^2}{12}\frac{\partial^2 f}{\partial x^2}(\bar{x},\bar{y})$$

$$-\frac{(b-a)(d-c)}{12}k^2\frac{\partial^2 f}{\partial y^2}(\bar{x},\bar{y})$$

$$E = -\frac{(b-a)(d-c)}{12}\left[h^2\frac{\partial^2 f}{\partial x^2}(\bar{x},\bar{y}) + k^2\frac{\partial^2 f}{\partial y^2}(\bar{x},\bar{y})\right]$$

* Finding The Error in Simpson's rule for Multiple Integrals.

we have

$$E = \frac{-k(b-a)h^4}{540}\left[\frac{\partial^4 f}{\partial x^4}(\bar{x}_0, y_0) + \frac{\partial^4 f}{\partial x^4}(\bar{x}_m, y_m)\right.$$

$$\left. + 2\sum_{j=1}^{\frac{m}{2}-1}\frac{\partial^4 f}{\partial x^4}(\bar{x}_{2j}, y_{2j}) + 4\sum_{j=1}^{\frac{m}{2}}\frac{\partial^4 f}{\partial x^4}(\bar{x}_{2j-1}, y_{2j-1})\right]$$

$$-\frac{(d-c)}{180}k^4\int_a^b \frac{\partial^4}{\partial y^4} f(x,\bar{y})\,dx$$

where $\bar{x}_j \in [a,b]$ $\quad j=0,1,\ldots,m$

if $\frac{\partial^4 f}{\partial x^4}$ and $\frac{\partial^4 f}{\partial y^4}$ are continuous then

by using intermediat value theorem and Mean value theorem $\exists\ (\bar{x},\bar{y}) \in R$

$R = [a,b] \times [c,d]$.

we have

$$\int_a^b \frac{\partial^4 f}{\partial y^4}(x, \bar{y})\, dx = (b-a)\frac{\partial^4 f}{\partial y^4}(\bar{x}, \bar{y})$$

and if we write

$$\varepsilon_{n,m} = \frac{\frac{\partial^4 f}{\partial x^4}(\bar{x}_0, y_0) + \frac{\partial^4 f}{\partial x^4}(\bar{x}_m, y_m) + 2\sum_{j=1}^{\frac{m}{2}-1}\frac{\partial^4 f}{\partial x^4}(\bar{x}_{2j}, y_{2j}) + 4\sum_{j=1}^{\frac{m}{2}}\frac{\partial^4 f}{\partial x^4}(x_{2j-1}, y_{2j-1})}{3m}$$

then $\varepsilon_{n,m}$ satisfies

$$\min \frac{\partial^4}{\partial x^4} f(x,y) \leq \varepsilon_{n,m} \leq \max \frac{\partial^4 f}{\partial x^4}(x,y)$$

where $(x,y) \in R$

There fore $\varepsilon_{n,m} = \frac{\partial^4 f}{\partial x^4}(\bar{x}, \bar{y})$

So,

$$E = \frac{-k(b-a)h^4}{540}\left[3m \frac{\partial^4 f}{\partial x^4}(\bar{x}, \bar{y})\right]$$

$$- \frac{(d-c)}{180} k^4 (b-a) \frac{\partial^4 f}{\partial y^4}(\bar{x}, \bar{y})$$

but $m = \frac{d-c}{k}$

$$\therefore \quad E = \frac{-(b-a)(d-c)}{180}\left[h^4 \frac{\partial^4 f}{\partial x^4}(\bar{x}, \bar{y}) + k^4 \frac{\partial^4 f}{\partial y^4}(\bar{x}, \bar{y})\right].$$

where $(\bar{x}, \bar{y}) \in R = [a,b] \times [c,d]$.

we choose $\bar{x}, \bar{y}$ such that $e^{\bar{y}-\bar{x}}$ has its largest value at $[0, \frac{1}{2}]$

which is $e^{0.5}$

$$|E| \leq \frac{0.5(0.5)}{180}\left[\left(\tfrac{1}{8}\right)^4 e^{0.5} + \left(\tfrac{1}{8}\right)^4 e^{0.5}\right]$$

$$= \ldots$$

Homework

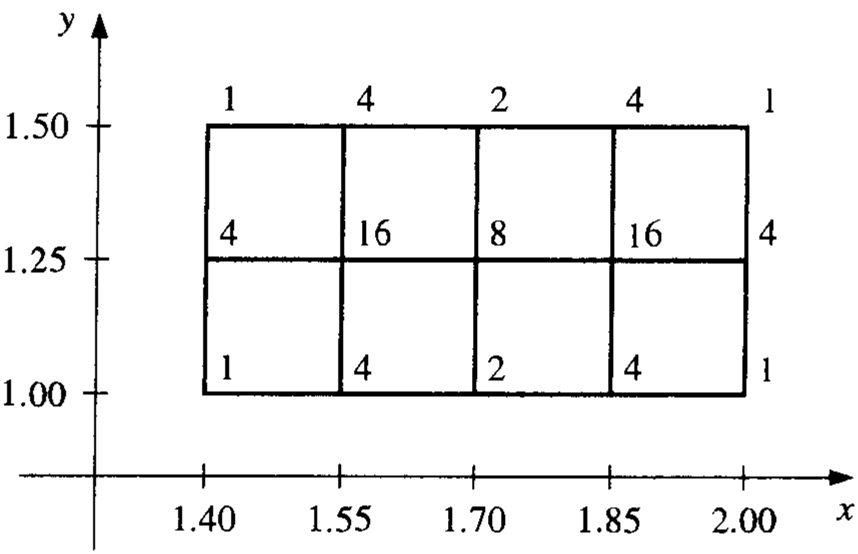
use Trapezoidal rule to find

$$\int_a^b \int_c^d f(x,y)\,dy\,dx$$

EXAM PLE 1 The Composite Simpson's rule applied to approximate

$$\int_{1.4}^{2.0} \int_{1.0}^{1.5} \ln(x + 2y)\,dy\,dx,$$

with n = 4 and m = 2 uses the step sizes h = 0.15 and k = 0.25. The region of integration R is shown in Figure together with the nodes (x_i, y_j) far i = 0, 1,2,3,4 and j = 0, 1,2, and the coefficients $w_{i,j}$ of $f(x_i,y_i)$ = In(x_i + 2y_i) in the sum.



The approximation is

$$\int_{1.4}^{2.0}\int_{1.0}^{1.5} \ln(x+2y)\,dy\,dx \approx \frac{(0.15)(0.25)}{9}\sum_{i=0}^{4}\sum_{j=0}^{2} w_{i,j}\ln(x_i+2y_j)$$

$$= 0.4295524387.$$

Since

$$\frac{\partial^4 f}{\partial x^4}(x,y) = \frac{-6}{(x+2y)^4} \quad \text{and} \quad \frac{\partial^4 f}{\partial y^4}(x,y) = \frac{-96}{(x+2y)^4},$$

and the maximum value of $\frac{1}{(x+2y)^4}$ on R occurs at $(1.4, 1.0)$,

the error is bounded by

$$|E| \le \frac{(0.5)(0.6)}{180}\left[(0.15)^4 \max_{(x,y)\text{in}R}\frac{6}{(x+2y)^4} + (0.25)^4 \max_{(x,y)\text{in}R}\frac{96}{(x+2y)^4}\right]$$

$$\le 4.72 \times 10^{-6}.$$

The actual value of the integral to 10 decimal places is

$$\int_{1.4}^{2.0}\int_{1.0}^{1.5} \ln(x+2y)\,dy\,dx = 0.4295545265,$$

so the approximation is accurate to within **2.1×10^{-6}**.

The same techniques can be applied for the approximation of triple integrals. as well as higher integrals for functions of more than three variables. The number of functional evaluations required for the approximation is the product of the number required when the method is applied to each variable.

To reduce the number of functional evaluations. more efficient methods such as Gaussian quadrature, Romberg integration, or Adaptive quadrature can be incorporated in place of Simpson's formula. The following example illustrates the use of Gaussian quadrature for the integral considered in Example I.

EXAM PLE 2

Consider the double integral given in Example 1. Before employing a Gaussian quadrature technique to approximate this integral, we must transform the region of integration

$$R = \{(x, y) \mid 1.4 \le x \le 2.0, \quad 1.0 \le y \le 1.5\}$$

into

$$\hat{R} = \{(u, v) \mid -1 \le u \le 1, \quad -1 \le v \le 1\}.$$

The linear transformations that accomplish this are

$$u = \frac{1}{2.0 - 1.4}(2x - 1.4 - 2.0), \quad \text{and} \quad v = \frac{1}{1.5 - 1.0}(2y - 1.0 - 1.5).$$

Employing this change of variables gives an integral on which Gaussian quadrature can be applied:

$$\int_{1.4}^{2.0} \int_{1.0}^{1.5} \ln(x + 2y)\, dy\, dx = 0.075 \int_{-1}^{1} \int_{-1}^{1} \ln(0.3u + 0.5v + 4.2)\, dv\, du.$$

The Gaussian quadrature formula for n = 3 in both u and v requires that we use the nodes

$$u_1 = v_1 = r_{3,2} = 0, \quad u_0 = v_0 = r_{3,1} = -0.7745966692,$$
$$\text{and} \quad u_2 = v_2 = r_{3,3} = 0.7745966692.$$

The associated weights are found in Table 4.3 (Section 4.5) to be C3.2 = 0.88 and c3.1 = c3.3 = 0.55,so

$$\int_{1.4}^{2.0} \int_{1.0}^{1.5} \ln(x + 2y)\, dy\, dx \approx 0.075 \sum_{i=1}^{3} \sum_{j=1}^{3} c_{3,i} c_{3,j} \ln(0.3 r_{3,i} + 0.5 r_{3,j} + 4.2)$$
$$= 0.4295545313.$$

Even though this result requires only 9 functional evaluations compared to 15 for
the Composite Simpson's rule considered in Example I, the result is accurate to within 4.8 X 10^{-9}, compared to an accuracy of only 2X10^{-6}forSimpson's rule.

The use of approximation methods for double integrals is not limited to integrals with rectangular regions of integration. The techniques previously discussed can be modified to approximate double integrals with variable inner limits-that is. integrals of the form

$$\int_a^b \int_{c(x)}^{d(x)} f(x, y)\, dy\, dx.$$

For this type of integral we begin as before by applying Simpson's Composite rule to integrate with respect to both variables. The step size for the variable x is h = (b - a)/2, but the step size k(x) for y varies with x (see Figure 4.16 on page 152):

$$k(x) = \frac{d(x) - c(x)}{2}.$$

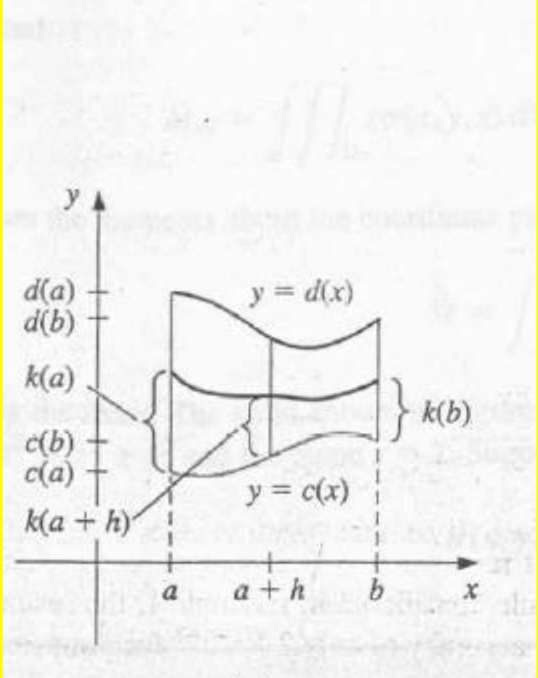
Consequently,

$$\int_a^b \int_{c(x)}^{d(x)} f(x, y)\, dy\, dx \approx \int_a^b \frac{k(x)}{3}[f(x, c(x)) + 4f(x, c(x) + k(x)) + f(x, d(x))]\, dx$$

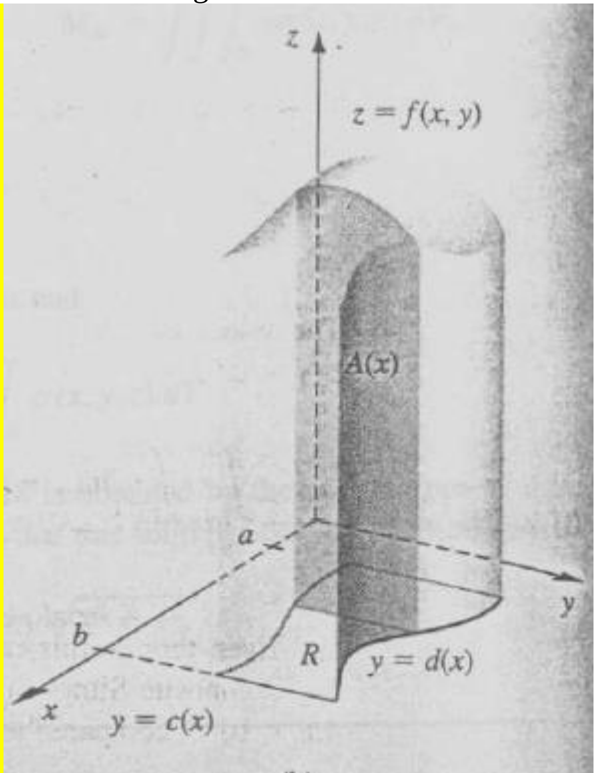
$$\approx \frac{h}{3}\left\{ \frac{k(a)}{3}[f(a, c(a)) + 4f(a, c(a) + k(a)) + f(a, d(a))] \right.$$

$$+ \frac{4k(a+h)}{3}[f(a+h, c(a+h)) + 4f(a+h, c(a+h) + k(a+h)) + f(a+h, d(a+h))]$$

$$\left. + \frac{k(b)}{3}[f(b, c(b)) + 4f(b, c(b) + k(b)) + f(b, d(b))] \right\}.$$



(a)



(b)

The program DINTGL44 applies the Composite Simpson's rule to a double integral in this form and is also appropriate. *of* course. when *c(x)* == *c* and *d(x)* == *d*.
To apply Gaussian quadrarure to the double integral first requires transforming, for
each *x* in *[a,b1*, the interval *[c(x),d(x)]* to [-1, t] and then applying Gaussian quadrature.   This results in the formula

$$\int_a^b \int_{c(x)}^{d(x)} f(x,y)\,dy\,dx \approx \int_a^b \frac{d(x)-c(x)}{2} \sum_{j=1}^{n} c_{n,j} f\left(x, \frac{(d(x)-c(x))r_{n,j} + d(x) + c(x)}{2}\right) dx,$$

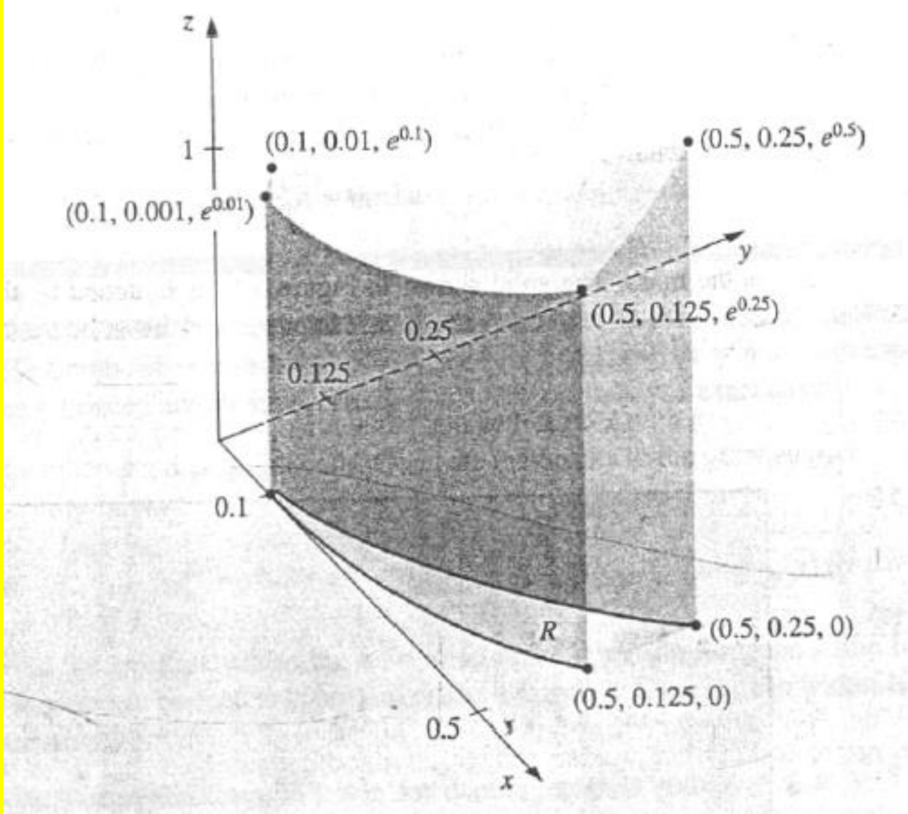
where, as before, the roots $r_{n,j}$ and coefficients $c_{n,j}$ come from Table 4.6. Now the interval [a,b] is transformed to [-1,1], and Gaussian quadrature is applied to approximate the integral on the right side of this equation. The program DGQINT45 uses this technique

EXAMPLE

Applying Simpson's double integral program DINTGL44 with n = m = 10 to

$$\int_{0.1}^{0.5} \int_{x^3}^{x^2} e^{y/x}\,dy\,dx$$

requires 121 evaluations *of* the function *f(x,y)* = $e^{y/x}$ and produces the approximation 0.0333054, accurate to nearly 7 decimal places, to the volume *of* the solid shown in Figure 4.17. Applying the Gaussian quadrature program DGQINT45 with n = m. = .5 requires only 25 function evaluations and gives the approximation, 0.3330556611. which IS accurate . . . . to 11 decimal places.



EXERCISE SET 4.7

1. Use Composite Simpson's rule for double integrals with $n = m = 4$ to approximate the following double integrals. Compare the results to the exact answer

a. $\int_{2.1}^{2.5}\int_{1.2}^{1.4} xy^2\,dy\,dx$

b. $\int_{0}^{0.5}\int_{0}^{0.5} e^{y-x}\,dy\,dx$

c. $\int_{2}^{2.2}\int_{x}^{2x} (x^2+y^3)\,dy\,dx$

d. $\int_{1}^{1.5}\int_{0}^{x} (x^2+\sqrt{y})\,dy\,dx$

2- Find the smallest values for n = m so that Composite Simpson's can be used to approximate the integrals in Exercise 1 to within 10^{-6} of the actual value.

3. Use Algorithm 4.4 with (i) $n = 4$, $m = 8$, (ii) $n = 8$, $m = 4$, and (iii) $n = m = 6$ to approximate the following double integrals, and compare the results to the exact answers.

a. $\int_0^{\pi/4}\int_{\sin x}^{\cos x} (2y\sin x + \cos^2 x)\,dy\,dx$ **b.** $\int_1^e\int_1^x \ln xy\,dy\,dx$

c. $\int_0^1\int_x^{2x} (x^2 + y^3)\,dy\,dx$ **d.** $\int_0^1\int_x^{2x} (y^2 + x^3)\,dy\,dx$

e. $\int_0^{\pi}\int_0^x \cos x\,dy\,dx$ **f.** $\int_0^{\pi}\int_0^x \cos y\,dy\,dx$

g. $\int_0^{\pi/4}\int_0^{\sin x} \frac{1}{\sqrt{1-y^2}}\,dy\,dx$ **h.** $\int_{-\pi}^{3\pi/2}\int_0^{2\pi} (y\sin x + x\cos y)\,dy\,dx$

3. Composite Simpson's rule first with $n = 4$ and $m = 8$, then with $n = 8$ and $m = 4$, and finally with $n = m = 6$ gives the following.

a. 0.5119875, 0.5118533, 0.5118722

b. 1.718857, 1.718220, 1.718385

c. 1.001953, 1.000122, 1.000386

d. 0.7838542, 0.7833659, 0.7834362

e. −1.985611, −1.999182, −1.997353

f. 2.004596, 2.000879, 2.000980

g. 0.3084277, 0.3084562, 0.3084323

h. −22.61612, −19.85408, −20.14117

[6] Find the smallest values for n = m so that Composite Simpson's rule for double integrals can be used to approximate the integral

to within 10-6 of the actual value.

Answer

Since n=m, so h=k,

h=(b-a)/n=(1/2n)

n=4