Second Revised Edition - 2008

Numerical Techniques

J. S. Chitode

Technical Publications Pune

Numerical Techniques

J. S. Chitode
M. E. Electronics (Digital Systems)
M.I.S.T.E., M.I.E.T.E., M.B.E.S.I.
Professor and Head
Department of Electronics Engineering
Bharati Vidyapeeth Deemed University
College of Engineering
Pune

Price Rs. 295/-

Visit us at : **www.vtubooks.com**

Technical Publications Pune

1-

1- numerical methods

Numerical Techniques

ISBN 978 - 81 - 8431 - 018 - 4

Published by :

Technical Publications Pune®

#1, Amit Residency, 412, Shaniwar Peth, Pune - 411 030, India

Printer :

Alert DTPrinters
Sr.no. 10/3,Sinhagad Road,
Pune - 411 041

Preface

First edition of this book is highly appreciated by the students as well as teachers. Feedback from many of them encouraged us to bring out this revised edition. This new edition includes large number of solved examples from university question papers. Some of the topics are revised in detail. We are sure that this new edition will also be received by students and teachers as well.

Preface to first edition

The basic aim of this subject is to learn numerical techniques,develop logical thinking and programming ability of the students. Therefore use of C language and MATLAB is recommended to implement the algorithms of Numerical Techniques. The subject of Numerical Techniques is very much important for engineering students. Many times it is not possible to obtain the solution to the problem easily using analytical methods. Then numerical techniques are applied and solution is obtained. The logic of the numerical techniques is implemented on Computer. Because of the fast computers today,the solutions are obtained fast. Thus the main part of this procedure is to transfer the numerical algorithm into computer program, with minimum errors. The Program should take less time for execution, it should give output with minimum error and it should be user friendly. This all needs skilled programming.

Contents and Organization

The aim of this book is to develop your skills of programming as well as learn numerical methods. The programs which are given in this book are supported by detailed explanation of the logic. The results of the programs are also presented. Some numerical problems are taken to test the programs. The numerical techniques in this book are organized as follows:

1. Explanation of the numerical technique.
2. One or two illustrative problems.
3. Algorithm and flowchart of the method.
4. C-program of the method.
5. Explanation of the logic.
6. Results of the program with some numerical problem.
7. More solved examples.

8. Engineering applications
9. MATLAB programs
10. Exercise (unsolved examples) and computer exercise at the end of every chapter.

Minor changes may be there. Some fundamentals of C programming are covered in first chapter. This chapter also gives basic principles of numerical techniques etc.

The second chapter explains the computer representation of numbers and errors. It gives various types of errors, various ways to represent numbers etc. The third chapter explains various methods to find a root of simple algebraic or transcendental equations.

The fourth chapter presents techniques to solve the systems of linear equations.The fifth chapter is based on interpolation and polynomial approximation. Various methods of interpolation based on finite and other differences are discussed in this chapter.

Sixth chapter presents numerical differentiation techniques. Seventh chapter presents numerical integration techniques such as trapezoidal and simpson's rules. Eighth chapter presents the solutions to ordinary differential equations. And last chapter discusses optimization. It presents constrained and unconstrained optimization methods.

Every chapter is followed by engineering applications mainly to electrical and electronics. MATLAB programs are also presented along with results at the end of every chapter.

Large number of solved examples are given for every method. Typing and other errors are removed to the maximum possible extent. Any suggestions, criticism, comments regarding the organization of the book, errors, software programs etc will be highly appreciated.

Acknowledgements

First of all I thank all the students and teachers who highly appreciated earlier edition of this book. I also thank Shri. Avinash Wani, Shri. Ravindra Wani and staff of Technical Publications to bring this revised edition up to you.

Author

J.S. Chitode

Table of Contents

List of Programs

The following programs are available on **www.vtubooks.com** for free download. You will get 'zip' file from website. You will have to extract the programs from 'zip' file using 'winzip' facility available in Windows. The source (`.cpp`) and executable (`.exe`) files are provided in different folders. The programs have `.cpp` extension. But they are basically 'C' programs. The MATLAB programs are with (`.m`) extension in separate folder.

1
Introduction to Numerical Techniques & C-Programming

1.1 General Principles of Numerical Techniques

Definition

Numerical techniques are the techniques by which mathematical problems are formulated and they can be solved with arithmetic operations. Those techniques are basically **numerical methods**. Those arithmetic operations are implemented with the help of programming languages on digital computers. Because of the fast and powerful computers are available today, numerical methods are implemented easily.

The subject of numerical techniques mainly deals with numerical methods and required programming logic development. Nowadays, because of the fast development of computers, they are being used in each and every field. Most of the fields needs computations. Those computations are performed on computers with the help of numerical methods. Most of the mathematical operations like integration, differentiation, interpolation, curve fitting, solutions of linear and nonlinear equations etc. are converted to simple arithmetic operations like addition, subtraction, multiplication and division etc. with the help of numerical methods. Then the programs can be easily written to implement these arithmetic operations on

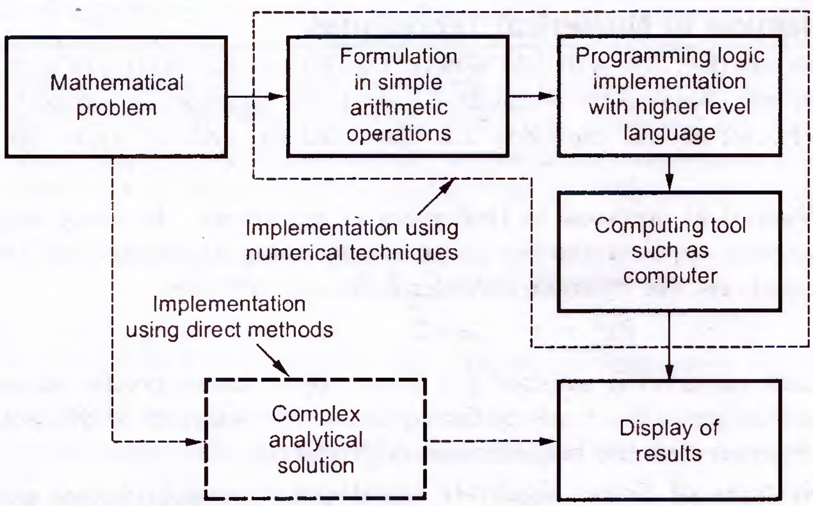


Fig. 1.1.1 Illustration of solution of mathematical problem using numerical techniques

computers. Very much advanced programming tools like C/C ++, VC ++ etc. are available today. They even support the standard library of mathematical functions. Fig. 1.1.1 illustrates the basic stages which shows how a mathematical problem is solved using numerical techniques.

Those basic stages are as follows :

1) Mathematical problems are formulated in simple arithmetic operations. This formulation is called numerical implementation of the problem.
2) A programming logic is then developed for this numerical implementation. The programming is usually done with some higher level languages.
3) The programs are then executed on the computing tools such as mini or micro computers.
4) The results are displayed on the screen or printed on the paper.

Now let's see the principles of numerical techniques.

1) Numerical Techniques are powerful problems solving tools. Large number of complexities are made simple with the help of numerical techniques.
2) Many problems can not be directly solved, since the direct solution does not exists. That needs the problem to redefined. Such problems can be solved with the help of numerical techniques.
3) Numerical techniques are implemented on computers. Therefore learning of computer, different platforms, exploiting the features of computers can be effectively done by numerical techniques.
4) Numerical Techniques need solid mathematical foundation as well as knowledge of computers and languages. Thus the combined knowledge of mathematics and software tools is extremely useful in many applications.

1.2 Introduction to Numerical Techniques

In this section we will see where the numerical methods can be used for problem solving. Numerical methods support the solution of almost every type of problem. The numerical methods are classified depending upon the type of the problem.

1) **Numerical methods to find roots of equations :** In many engineering and science applications we come across some algebraic and transcendental equations. For example consider following equation.

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

Such equation is satisfied [i.e. $f(x) = 0$] at some specific values of x. These values are called roots of the equation. The solution of this equation can be obtained with the help of numerical methods.

2) **Systems of linear algebraic equations :** Some problems generate linear equations with many variables. Those equations are normally given as,

$$2x + 3y = 7$$
$$5x + 8y = 18$$

The values of x and y in those equations can be obtained with the help of numerical methods. Direct methods become complex for more number of variables. Numerical methods are the best solution for large systems. Such equations are generated in the analysis of electronic circuits, large structures etc.

3) **Interpolation and curve fitting :** In most of the applications a digital data is available. This data may be taken at regular time intervals. A curve which passes through all data samples can be generated. Numerical methods give some techniques to approximate curves for such data. With the help of interpolation it is possible to find data values between the given data samples.

4) **Numerical integration :** Integration is used to find area under the curve. Numerical integration techniques provide best tools for finding the areas under complex curves. In addition to this, the numerical integration techniques are used wherever integration is required.

5) **Ordinary differential equations :** Ordinary differential equations are of great importance in engineering field. Most of the physical laws are formulated with the help of differential equations. Electromagnetic waves and their laws are very well described with the help of differential equations. Numerical methods provide some techniques to solve those equations.

1.3 Common Ideas and Concepts in Numerical Techniques

1.3.1 Mathematical Modeling

The numerical methods or techniques are basically the numerical algorithms. These numerical algorithms are used to find roots of equations, interpolation, numerical integration, solution of differential equations etc. as introduced in previous section. To implement these numerical algorithms in practice we need to define the problem or application. This is called *mathematical modeling* of the problem. Mathematical modeling means to describe by mathematical equations. For example, current in the resistance is proportional to applied voltage. This can be mathematically written as,

$$I = \frac{V}{R}$$

Here $\frac{1}{R}$ is the constant of proportionality. The above equation is nothing but the mathematical model which relates current and voltage in the resistance. This equation can be implemented numerically or directly to calculate the current.

1.3.2 Recursion and Numerical Methods

Once the problem is described by mathematical equations, next step is to select the proper numerical method. The numerical method solves the problem. These numerical methods can be implemented on computers, micro processors etc in hardware or software. Almost all the numerical methods use recursive principle. Recursion means to calculate next solution from existing solution. More and more recursions (also called as iterations) give more correct solution. Note that the solutions obtained by numerical methods are not exact. The existing solution is used to calculate next solution, which is more accurate.

1.3.3 Numerical Instability

Recursive principles are used in numerical methods. Since numerical solutions are not exact, there is always 'error' between numerical solution and exact solution. This 'error' goes on reducing in successive recursions (iterations). There is a chance that 'error' may increase instead of reducing. This error may increase when the numerical method doesnot suit to the problem being solved. Hence some other numerical method must be used. This concept is called as 'convergence' of the numerical method. If the numerical method is 'convergent' then 'error' reduces in successive 'recursions'. If the error increases then proper solution is never obtained. This is called as *numerical instability* or numerical instability. This concept is explained with the help of examples in next chapter.

1.3.4 Computing Tools

Computing tools are the required machinery or softwares to implement numerical methods. Those tools are mainly computers, softwares, logic development tools like flowcharts and algorithms etc.

1.3.4.1 Computers

Fast processors are available today. Therefore the computing speed is very large. Numerical methods are implemented on computers. There are different types of computers. They are microcomputers, minicomputers and mainframe computers. Minicomputers are the personnel computers which are in use by maximum people. The operating speed, RAM, address and data bus width, type of the instruction set are most important factors for computers. These factors are mainly related to processor used in the computers. Most of the computers which we use today are IBM computers. In starting PC's (Personnel Computers) were available with 8088 as a processor.

Then came the computers with 80286 and 80386 processors. They had 32 address lines and 32 data lines. They have other advanced features. Presently Pentium processors are available. Those computers use 80586 (called Pentium) as a processor. Pentiums have 64 data lines and 32 address lines. They operate on the speeds upto 1GHz. They support some special functions like multiprocessing and multitasking.

The computing speed of Pentium is very high. Advanced operating systems like windows are supported by these computers. Because of these features, the software development is relatively easier.

The type of the computer being used by the numerical method plays an important role. Almost all the type of numerical techniques demand fast computers.

Operating system, memory, multiprocessing environment, networking etc. are less important factors compared to speed and word length (i.e. data bus width) of the computer.

1.3.4.2 Softwares

Numerical algorithms can be implemented with the help of any higher level languages such as BASIC, FORTRAN, PASCAL, C/C++, VC++, Visual Basic etc. Since most of the numerical techniques demand structured implementation, PASCAL and C/C++ can be used. Basically C/C++, PASCAL and other higher level languages are structured languages. C/C++ and its window based versions are most widely used languages today. It supports structured programming. A structured programming is a set of rules that prescribes good style habits for the programmer.

Other standard softwares such as MATLAB are available to run the numerical methods. In such softwares there is no need to actually implement the algorithm but more stress is given on how to effectively use it. In this book we will implement some basic numerical methods. These programs can be composed to build more flexible and advanced programs.

We will see algorithms and flowcharts in more details in the next sections.

1.3.4.3 Algorithms

An algorithm is the sequence of logical steps required to perform a specific task. A good algorithm greatly reduces programming efforts. A good algorithm has following characteristics

1) Each step of the algorithm must be deterministic.
2) The algorithm should be independent of the programming language.
3) The algorithm should not use any codewords. That is all the statements in algorithm should be understood by any person.
4) The algorithm must be general such that it can be used to implement program in any software tool.
5) The algorithm must end after a finite number of steps. An algorithm should not be open ended.
6) With the same algorithm, if two programmers are working independently; they should obtain the same results.

Let's consider an example of adding up of two numbers. The algorithm for this operation is given below.

Algorithm to add two numbers

Step 1 : Start the calculation.

Step 2 : Input the value for variable A

Step 3 : Input the value for variable B

Step 4 : Add the two variables, i.e. calculate $C = A + B$

Step 5 : Display the value of result, C on the screen.

Step 6 : End the calculation

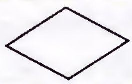
Observe that, with this algorithm anybody will get the same answer. Each and every step is well defined meaningful statement.

1.3.4.4 Flowcharts

Flowchart is the graphical or visual representation of an algorithm. Flowchart uses standard symbols to represent the type of the statement. Well organized flowchart gives complete understanding of the program.

The following table gives symbols of the flowchart

Table 1.3.1 : Symbols of a flowchart

Sr.No.	Symbol	Name	Meaning / function
1		Terminal	Represents begining or end of the program
2		Input/Output	It represents input or output of data. The input can be from keyboard, harddisk, floppy, internet or whatever. Output can be on printer, display screen, floppy etc,
3		Process	It represents data processing, calculations involved in the program.
4		Decision	Represents a comparison, question or decision at the end of loop or after some calculation.
5		Subroutine function	It represents subroutine or function. It can represent a complete program also as a subroutine.
6		On page connector	Represents break in the path of a flowchart that is used to continue flow from one point on a page to another.
7		OFF page connector	This is used to show continuity of a flow of programe from one page to other page. Normally a letter is written inside the symbol to distinguish between different off page connectors.
8	↓ ↑	Flow lines	They represent path of execution of the program.

Based on these symbols lets draw a flowchart for the algorithm to add two numbers. In the last subsection we described on algorithm to add two numbers. Fig. 1.3.1 shows a flowchart to add two numbers.

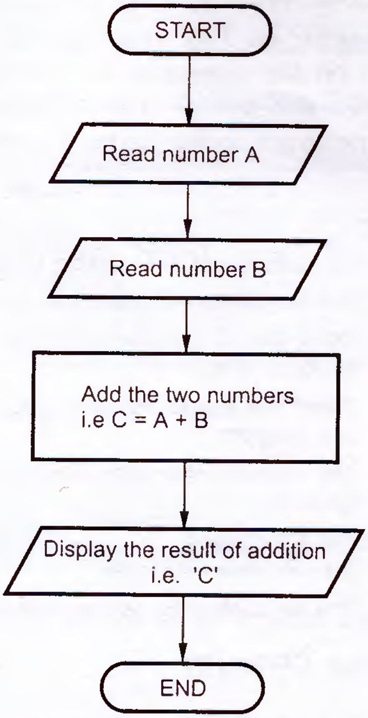


Fig. 1.3.1 Flowchart to add two number A & B

This is very simple flowchart. As the complexity of the program increases, the blocks can be merged into each other to make flowchart simple.

The above complete flowchart can be represented by a subroutine block.

Fig. 1.3.2 shows this subroutine block.

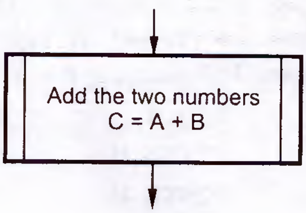


Fig. 1.3.2 Flowchart of Fig. 1.3.1 shown as a subroutine block

But the flowchart shown in Fig. 1.3.2 is incomplete since it has no start and end. But always it is a part of some other complex flowchart.

* 1.4 Introduction to 'C' Programming

A C-program is a sequence of statements that performs a specific task. Those statements uses some keywords. These keywords carry some meaning. A turbo C editor is used to type those statements. This editor is called **tc**. This **tc** supports editing the program and making its EXE file. An EXE file means, it is executable program. That program runs on the computer and perform the specific task. The 'C' program is first written into the text format using its keywords. Then an EXE program is prepared from this text program using turbo C compiler. **tc** performs all these operations.

1.4.1 Information about 'tc'

A 'tc' package consists of minimum following things :

1	**tc**.exe	⇒	This is main **tc** editor and compiler.
2	'Header' files	⇒	These files contain declarations of the standard programs (functions) available in '**tc**'.
3	'Library' files	⇒	These files contain standard programs (function) those are used by user program.
4	tchelp.tch	⇒	This contains information about how to use standard functions and keywords.
5	'bgi' files	⇒	These files contain necessay information of drivers required for graphics mode of display.
6	tcconfig.tc	⇒	This file contain the setting of configuration of '**tc**'

1.4.1.1 Check Out 'tc' on Your Computer

Locating 'tc' Package

Just now we mentioned a package of 'tc'. This package can exist on C, D, or E drive of your computer. This can be checked with the help of 'find' utility under windows.

As shown in Fig. 1.4.1 (See Fig. 1.4.1 on next page) observe that 'tc' exists in c:\tc directory of this computer.

Observing various utilities in 'tc' package

Go in the 'explore' menu and click on 'tc'. You will get the cntents of 'tc' directory. This is shown in Fig. 1.4.2. Observe that there is `tc.exe` which is executable program. The 'bgi' files are present in 'bgi' subdirectory. The header files are present in 'include' subdirectory. Following is the list of important header files.

ALLOC.H,	ASSERT.H,	BCD.H,	BIOS.H
COMPLEX.H,	CONIO.H,	CTYPE.H,	DIR.H
DOS.H,	ERRNO.H,	FCNTL.H,	FLOAT.H
FSTREAM.H,	GRAPHIC.H	GRAPHICS.H	IO.H
LIMITS.H,	MATH.H	PROCESS.H	STDIO.H
STDLIB.H,	STREAM.H	STRING.H	TIME.H

* Introduction to `C' programming given in this section is totally introductory. It covers few concepts of `C' which will be required to you again and again in this book. However for thorough understanding of C programming please refer "Programming with C", by Byron S Gottfried, Schaum series, Tata McGRaw Hill edition, New Delhi.

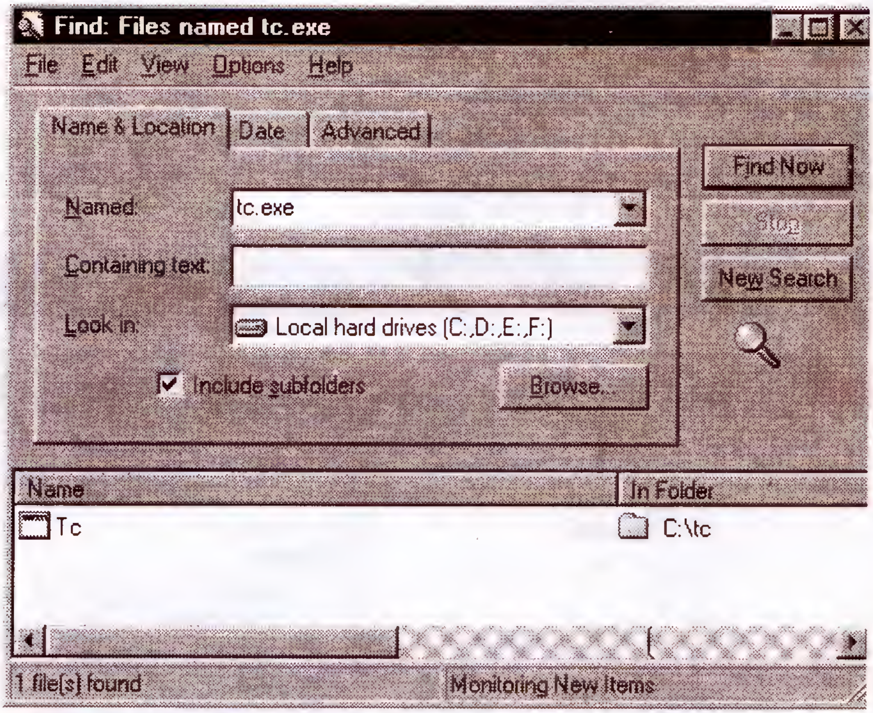


Fig. 1.4.1 Locating 'tc' in your computer

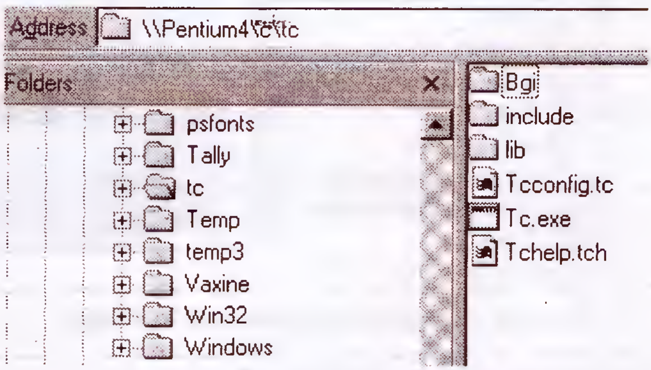


Fig. 1.4.2 Contents of 'tc' package

The library files are present in 'lib' subdirectory. Following is the list of imporatnt library files.

C0T.OBJ	COL.OBJ	COC.OBJ	COM.OBJ
C0H.OBJ	INIT.OBJ	MATHS.LIB	CS.LIB
CL.LIB	EMU.LIB	GRAPHICS.LIB	FP87.LIB
MATHC.LIB	MATHM.LIB	CC.LIB	CH.LIB
CM.LIB	MATHL.LIB	C0S.OBJ	MATHH.LIB

You can also see other files such as tchelp.tch (help file) and tcconfig.tc (configuration) file.

Creating a shortcut of 'tc' program

You can create a shortcut of tc.exe for easy operation. Go to tc.exe file and right click the mouse. Select 'create shortcut' menu. This will create the shortcut to 'tc'. It will appear at the end of all files in 'tc' directory.

Then drag this shortcut on desktop. Fig. 1.4.3 (a) shows the create shortcut procedure and Fig. 1.4.3 (b) shows the shortcut created.

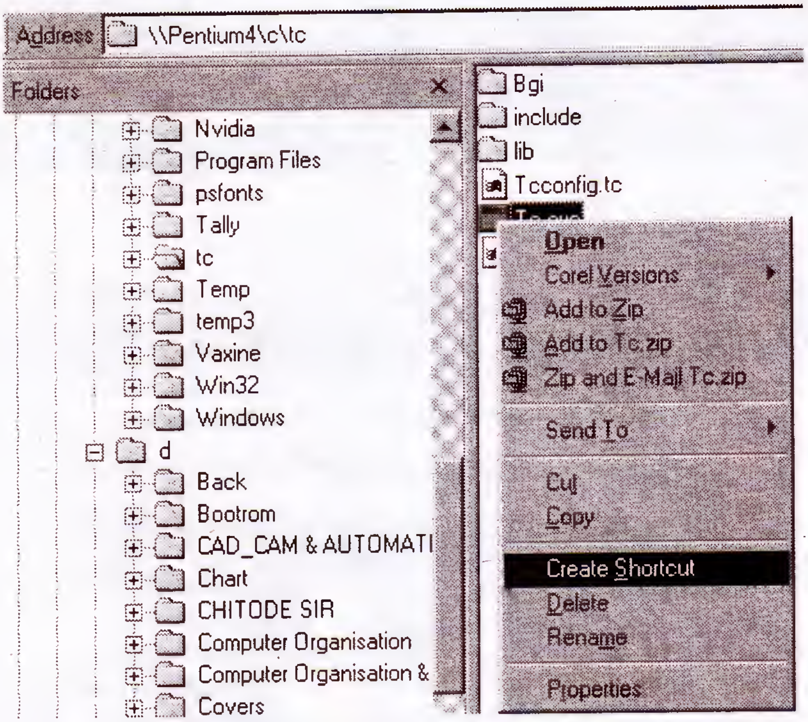


Fig. 1.4.3 (a) Creating shortcut to tc.exe

Fig. 1.4.3 (b) Shortcut of tc.exe

Now 'tc' will run if you just double click the 'shortcut to tc' icon on desktop.

1.4.1.2 How to Use 'tc' Editor?

Double click the shortcut to tc.exe icon.

Then a window will appear on the screen. This is the window of **'tc'** editor. **Press Alt and F keys simultaneously** (Normally written as Alt + F). The display will appear as shown in Fig 1.4.4.

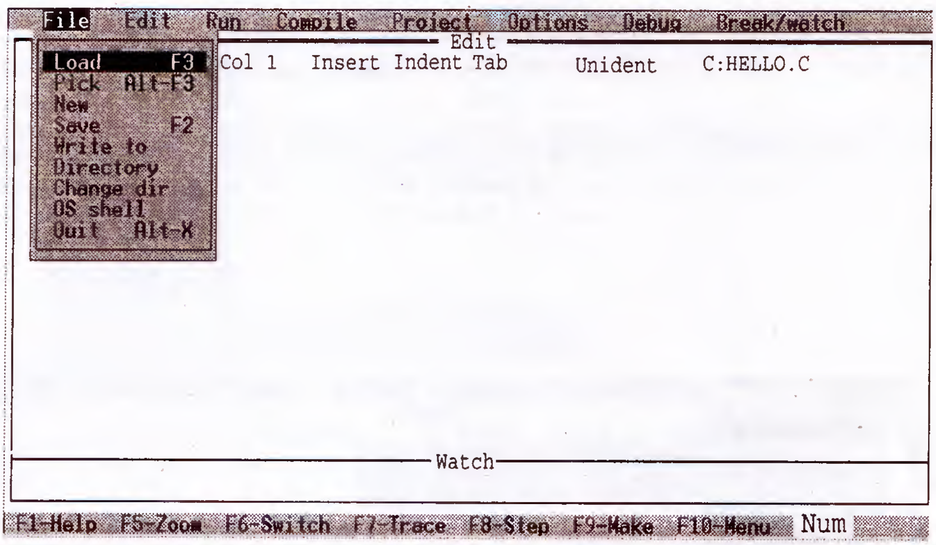


Fig. 1.4.4 Window of 'tc' editor, showing contents of `File` menu

Thus on the top of the screen there are other menus along with `File` menu. They are `Edit`, `Run`, `Compile`, `Project`, `Options`, `debug` and `Break/watch`. At the bottom of the screen, the tasks performed by various function keys are displayed.

1 `F1-Help` ⇒ F1 key is used for help on some function or keyword.

2 `F5-Zoom` ⇒ F5 key is used to enlarge the display window.

3 `F6-Switch` ⇒ F6 key is used to switch the cursor in editing window or message/watch window. Observe that message/watch window appears of the bottom of the screen. This window is used to display errors and warning. Message/watch window disappears when you zoom (i.e. when you press F5.)

4 `F7-Trace` ⇒ F7 key is used to execute program in single stepping mode. When you press F7, **'tc'** executes only one statement. This is useful for debugging.

5 `F8-Step` ⇒ F8 key is also used for single stepping.

6 `F9-Make` ⇒ F9 key is used to make EXE file of the `C' file.

7 `F10-Menu` ⇒ F10 key selects menu bar at the top of the screen. When you press F10, the first menu `File` will be highlighted.

Now press F10 (or Alt+F) key and select menu bar at the top of the screen. Press 'enter' when `File` menu is highlighted. A second window appears on the screen. This window shows submenus under `File`. This is shown in Fig. 1.4.4.

Let's see the function of those submenus-

1 `Load F3` ⇒ With the help of ↑ ↓ arrow keys highlight (or select) this menu. As shown in Fig. 1.4.4 `Load F3` is high lighted (i.e. selected). Then press `enter' key. Then other window appears which asks `Load File Name`. In this window type the name of the file which you want to create or which you want to load. For example, let the file `hello.c` is to be created. This is shown in Fig. 1.4.5.

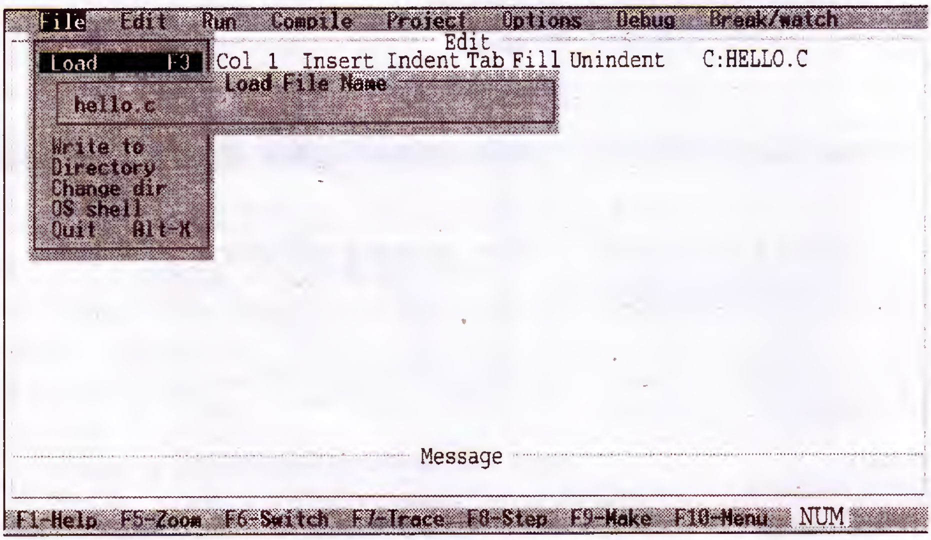


Fig. 1.4.5 Submenus of `File` indicating function of `Load File Name`

[**Note :** Always use `C' as an extension to the `tc' files]

When you press `enter'. The file `hello.c` is loaded. This name of file appears in top right corner of the screen. You can just press F3 to activate this menu.

2 `Pick Alt-F3` ⇒ Out of some selected files this menu is used to pickup a particular file.

3 `New` ⇒ When new file is to be created, this menu is used.

4 `Save F2` ⇒ This is used to save the file. F2 key can be used to save the file. Thus when you press F2, current file is stored on disk.

Press F2 to save your .c file

5 `Directory` ⇒ With this menu you can load the program in a particular directory. Select this menu with ↑ ↓ arrow keys. Press `enter' key. It asks for file name. You have to enter the directory and complete path of the file you want to load. Fig. 1.4.6 show this.

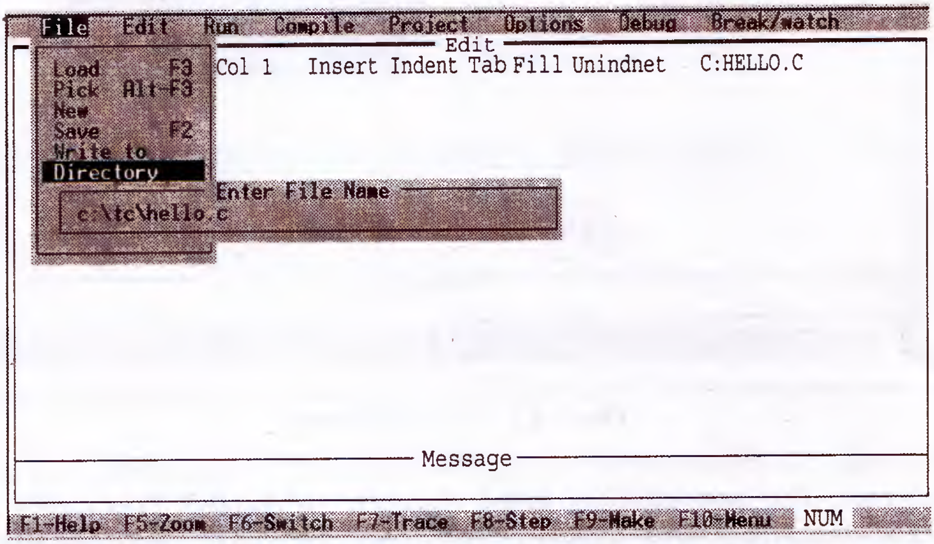


Fig. 1.4.6 Use of `Directory` submenu to load a file

6 `OS shell` ⇒ See Fig 1.4.5. This menu is second last in `File` menu window. With the help of ↑ ↓ arrow keys select (highlight) this menu and press `enter' key. The program comes to the DOS PROMPT. You can do whatever operations you want on DOS PROMPT. Remember that **'tc'** remains active. If you want to go back to 'tc' type `exit` and press 'enter' key. The 'tc' window again appears on the screen.

7 `Quit Alt-X` ⇒ If you want to stop working on **'tc'** and close it, then select this option and press `enter' key. Other wise you can press Alt+X (Alt and X key simultaneously) to quit from **'tc'**.

What is '`options`' menu?

Before going to `Run' and `Compile' menus, we will see what is the function of 'options' menu.

Press Alt+O (or press F10 and with the help of ← / → arrow keys) to select a menu called 'options'. When you press `enter' key, you will get submenus window. If you want to check any option, select (i.e. highlight) that option and press `enter' key Fig. 1.4.7 shows the submenus of 'compiler'.

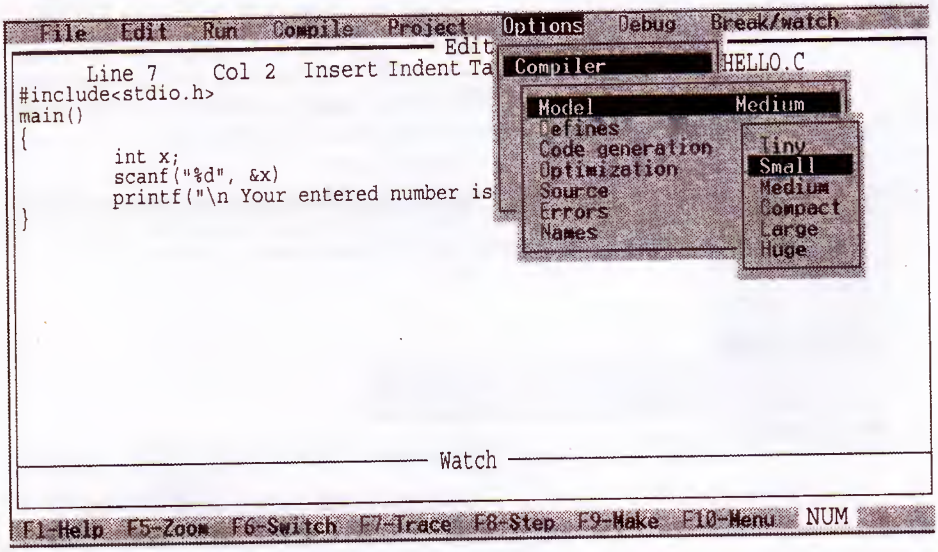


Fig. 1.4.7 'Options' **menu**

As shown in figure above, a compiler submenu is highlighted, in that model is highlighted. The next submenu called small is highlighted. This means that the "present setting in options for compiler model is small" you can select other compiler models like tiny, medium compact etc. The selection of this model depends upon size of the program. For numerical techniques 'small' model is sufficient. Press 'Esc' to come out of these submenus. Again select options menu (by pressing Alt + o or F10 and ← / → arrow keys). Press `enter' key. It displays the submenus under options. The first submenu called compiler we discussed just now. The fourth submenu is `directories'. Select this submenu with the help of ↑↓ arrow keys and press 'enter' key. It displays the window of 'directories' which are being used by `tc' Fig. 1.4.8 shows the display of this.

Please refer Fig. 1.4.8 on next page.

Let's see what is the meaning of each suboption.

1 Include directories : This option is for the path of header (.h files) files. Whenever the program is compiled, it needs header files. Therefore you should give correct path of directories where header files are stored. Observe that in the figure this path is, C:\TC\INCLUDE This means header files are present in `C:\TC\INCLUDE' directory.

2 Library directories : Here you should give the directory path where library files are stored. Observe that this path in the figure is,

C:\TC\LIB

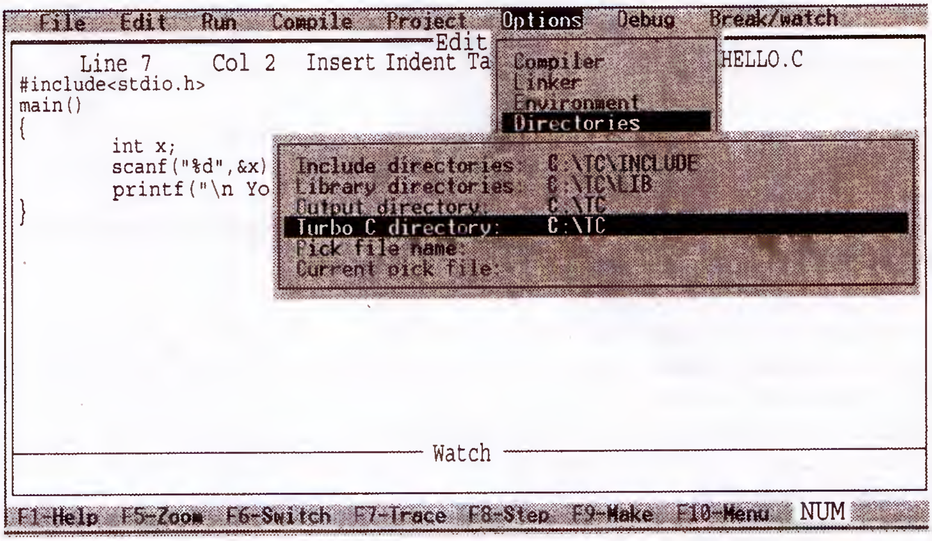


Fig. 1.4.8 Settings of the `directories` in `options` menu

This means the library files are present in C:\TC\LIB directory.

3 `Output directory` : This is the directory path where you are working. If you are working in the same directory of **'tc'** you need not give this path.

4 `Turbo C directory` : This is the path of directory in which `tc' is present. In the figure, this setting is,

`C:\TC`

This means **'tc'** is stored in C:\TC directory.

After the installation (or loading) of **'tc'** is complete on your computer, you should first set these `directory options`.

If you load **'tc'** on **'D'** drive, then these options will be as follows,

`Include directories`	:	`D:\TC\INCLUDE`
`Library directories`	:	`D:\TC\LIB`
`Output directory`	:	Here if you are working in the same directory of **'tc'** you need not give any option.

The other menus are related to running and compiling the program.

1.4.1.3 How to Compile and Run the Program?

We will first type a simple program and lets see how to make its EXE file (compile) and run the program.

Program : To enter the number in the computer and to print the same on screen.

The `C' source code (i.e. program) is given below.

```
# include<stdio.h>                This includes stdio.h to the program
main()
{
        int x;
        scanf ("%d", &x);This statement gets number from keyboard.
```

```
        printf("\n your entered number is = % d", x);
}
```
This statement prints the number on the screen.

First read the complete program given above. Here we are using two standard functions

`scanf()` ⇒ This function gets data from keyboard.

`printf()` ⇒ This function prints data on the screen.

We are using one variable -

`x` ⇒ This variable is assigned a value inputted from key board.

The first statement in the program is,

```
# include<stdio.h>
```

This statement 'includes' the header file 'stdio.h' in the program. We know that header files contain declarations of the standard functions. we are using printf and scanf standard functions in our program. 'C' program requires that those functions should be declared first in the program. stdio.h contains, the declarations of printf and scanf. stdio.h stands for `standard input and output' functions.

The next statement is,

```
main()
{
        ______
        ______
        ______  Program statements.
        ______
}
```

In every program you should write this function called main(). The '{' and `}' brackets indicate the program area of function main(). You should write your program within those brackets only. The next statement is,

```
int x;
```

This statement is the declaration of variable 'x'. It declares that variable x will be used as integer number. Every executable 'C' statement ends with `;' (i.e. semicolon).

The next two printf and scanf statements are in their formats. We will see their details later on. Fig. 1.4.9 shows this program written in 'tc' editor.

Please refer Fig. 1.4.9 on next page.

To type this program in the computer go to File menu by pressing 'Alt+F' key or 'F10' key and press 'enter'. Select load menu and press 'enter'. The display of Fig. 1.4.5 will appear. In that, write the file name as hello.c and press 'enter'. Now you have created hello.c file. Observe that we have given 'c' as an extension to the file. Then type the program given. Your program will look like that shown in Fig. 1.4.9 above. Then **press F2 key to save your program.**

Now your program is stored on hard disk.

To compile the program

Now press 'F10' key and '← / →' arrows to select compile menu. Actually you can select any of the menus on the top of the screen also by pressing **Alt and (first letter of the name of the menu)**. For example you can select compile menu by pressing `Alt and C' keys simultaneously (normally written as Alt+C). When compile

```
 File  Edit  Run  Compile  Project  Options  Debug  Break/watch
                              Edit
     Line 8     Col 1   Insert Indent Tab Fill Unindent   C:HELLO.C
#include<stdio.h>
main()
{
        int x;
        scanf("%d",&x)
        printf("\n Your enered number is = %d",x);
}

                              Watch

F1-Help  F5-Zoom  F6-Switch  F7-Trace  F8-Step  F9-Make  F10-Menu  NUM
```

Fig. 1.4.9 The 'C' program

is selected, press 'enter'. Then the window of submenus appears as shown in Fig. 1.4.10.

Please refer Fig. 1.4.10 on next page.

With the help of '↑ / ↓' arrow keys select '`Build all`' submenu and press 'enter'. This compiles and links the program.

What you should do if an error occurs ?

When you say `Build all` and press 'enter', immediately a window appears displaying compiling status. Fig. 1.4.11 shows how this window looks. (See Fig. 1.4.11 on next page.).

Observe that the message shows,

```
Lines compiled:219
     Warnings:0
       Errors:1
```

Here there are only 7 lines in our `hello.c` program. But we have included `stdio.h` file at the start of the program. Hence total lines of `hello.c` and `stdio.h` are displayed as 219. Observe that the message window shows one error. Unless we remove this error we can not proceed further. Then press any key. Then the display appears as shown in Fig. 1.4.12.

Fig. 1.4.12 shows that an error message is displayed in the message window at the bottom of the screen. The corresponding statement containing error is highlighted in the edit window. The error message is also highlighted. If there are more than one error, then you can use ↑ ↓ keys to see the error message and corresponding errornous statement.

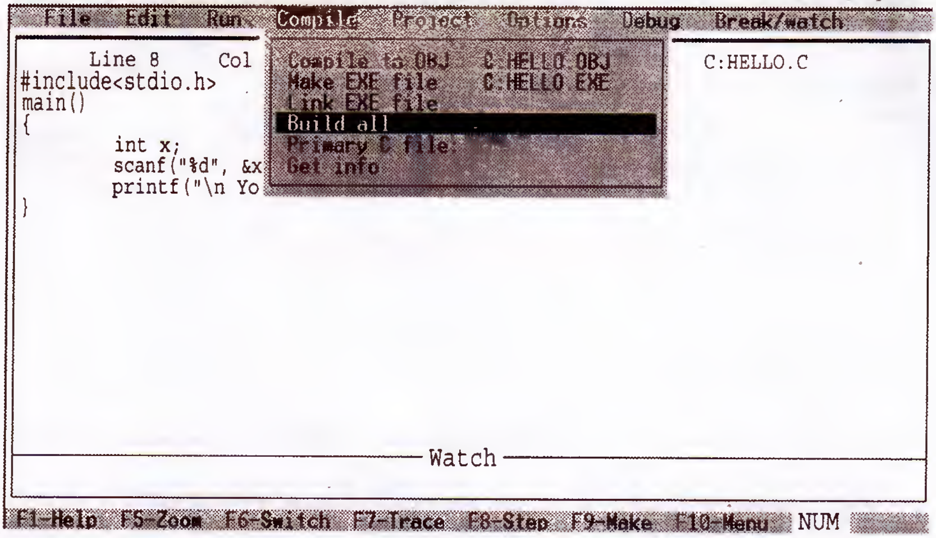


Fig. 1.4.10 `Compile` **submenus**

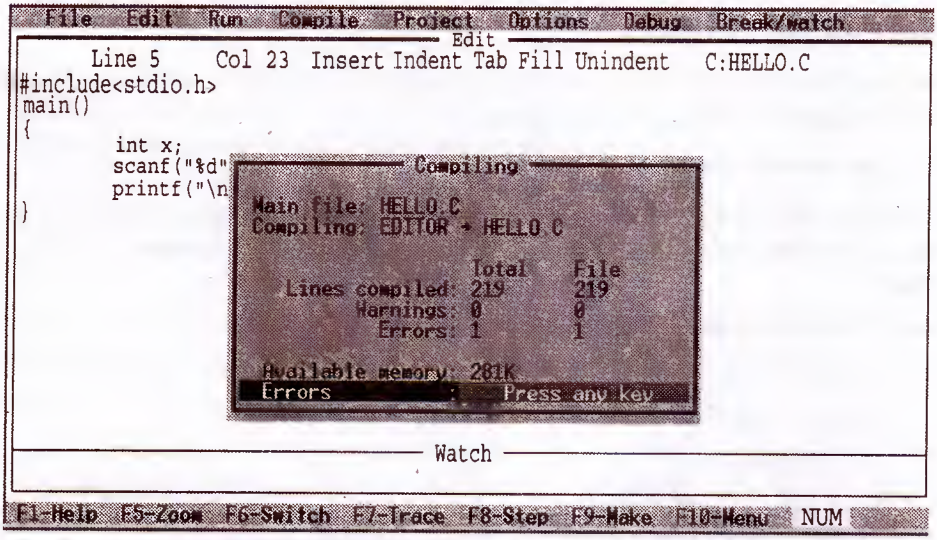


Fig. 1.4.11 Display of messages during compiling the program

This error can be present in the statement being highlighted or statements before and after this. **Observe that the semicolon (;) is missing after the `scanf` statement.**

This `scanf` statement is just before the highlighted `printf` statement. The compiler treates both of these statements as a single statement since semicolon (;) is missing. It gives error message on the second statement (`printf` statement) since there cannot be two functions in one statement. Therefore it gives an error **that some where semicolon (;) is missing near line 6.** Press 'F6' to switch into the edit window to correct the error. Type semicolon (;) at the end of `scanf` statement and press 'F2' to save this corrected program.

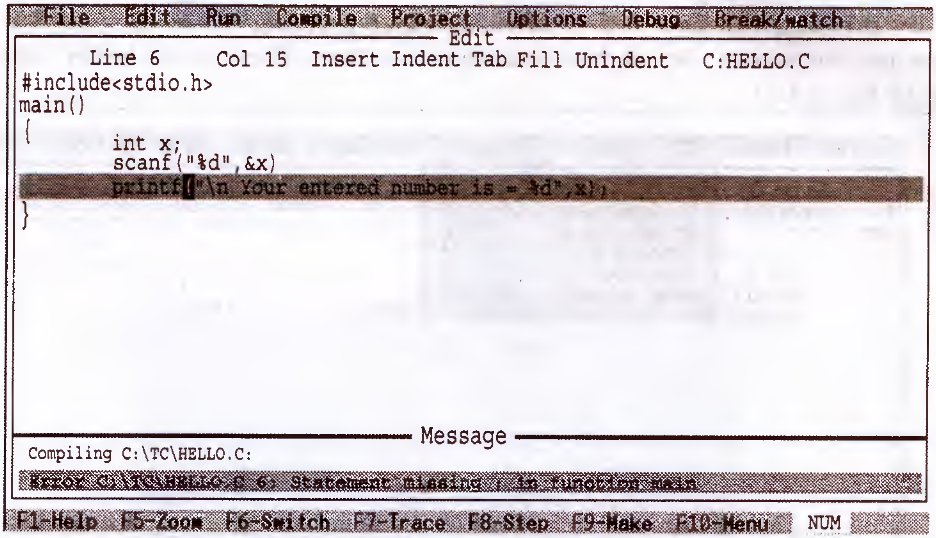


Fig. 1.4.12 A program with an error. Error message is displayed in message window

Again go to 'compile' menu and select 'Build all' and press 'enter' to compile the program. Now there are no errors and program is compiled and it makes EXE file also. The linking status window appears on the screen. Fig. 1.4.13 shows this.

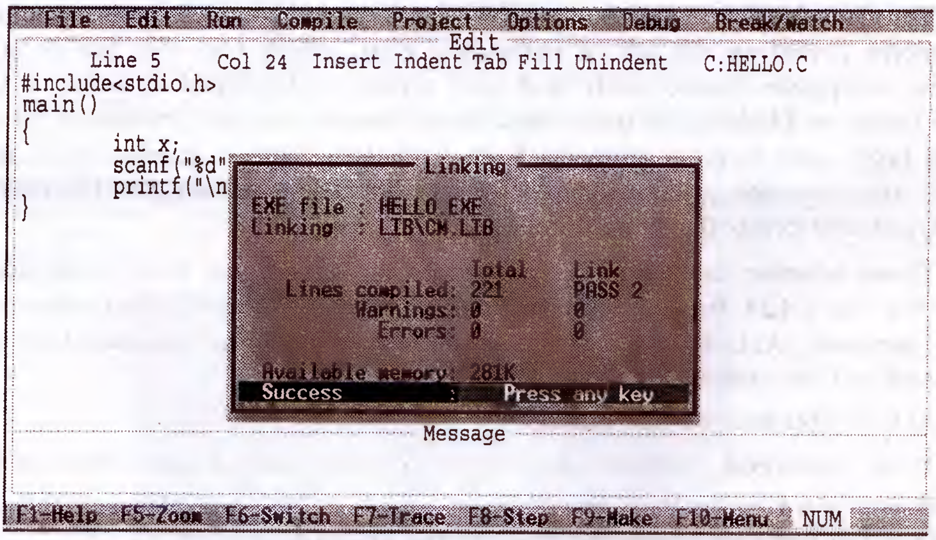


Fig. 1.4.13 Display of linking status window

Observe that linking means conversion of compiled object file to an executable code by combining all the necessary programs.

How to Run the program ?

Press Alt & R keys simultaneously *to select* `Run` *menu (you can also select this* menu by pressing F10 and ← / → arrows). After pressing 'enter' a submenu window appears on the screen. Select first submenu i.e. `Run`. Then press 'enter' key. This is shown in Fig. 1.4.14.

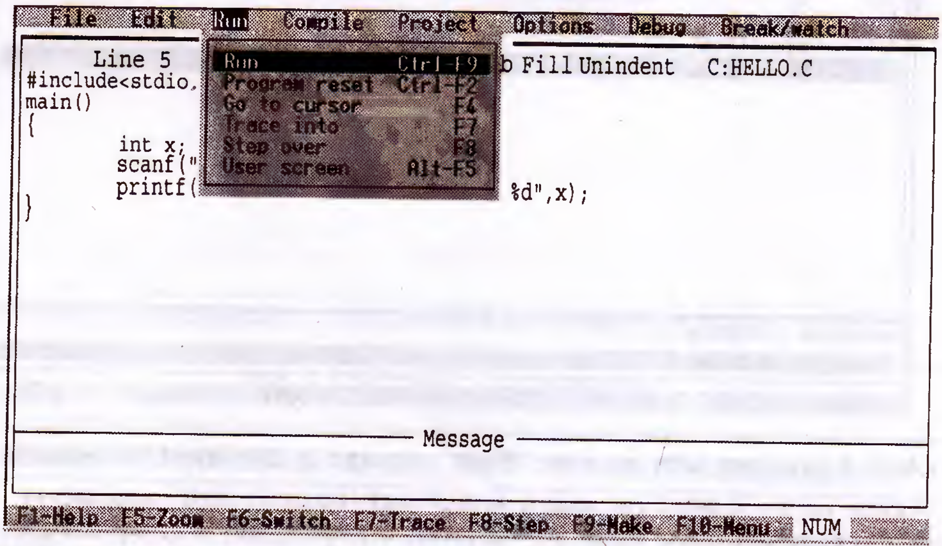


Fig. 1.4.14 Running the program using `Run` **menu. It is activated using** `Ctrl-F9`

You can also 'run' the program by pressing `ctrl` and `F9` keys simultaneously. This menu performs the job of compiling also. When you run the program 'tc' window disappears immediately and user screen is displayed in front of you. The cursor keeps on blinking on the screen. Since there is a `scanf` statement, it waits for you to type some number (integer) from keyboard. Type a number '450' and press 'enter'. After this, the program comes back to 'tc'. The program gets the number you have typed and prints the same number on the screen.

To see whether the number is printed on the screen, you have to see user screen again. See Fig. 1.4.14. In this figure in the `Run` submenus, there is last submenu called '`User screen Alt-F5`'. Just press 'Alt and F5' keys simultaneously. User screen is displayed and the statement displayed by the program is,

`450` ← You entered this number

`Your entered number is = 450` ← Computer program displayed this

If you want to go back to 'tc', press any key. You can also run this program from DOS PROMPT by typing the name of the program i.e.,

`C:\TC>hello` and press 'enter' key

Then type some 'integer' number and again press 'enter'. The program displays your typed number.

1.4.1.4 How to Use 'Help' in tc?

Go back to your `hello.c` program in 'tc'. Let's say you want more information on `printf` statement. Then adjust cursor below `printf` word as shown below.

```
Printf ("\n your entered number is =%d",x);
```

Then press ctrl and F1 keys simultaneously. The help window for `printf` appears on the screen. This is shown in Fig. 1.4.15. Press `Esc' to cancel help.

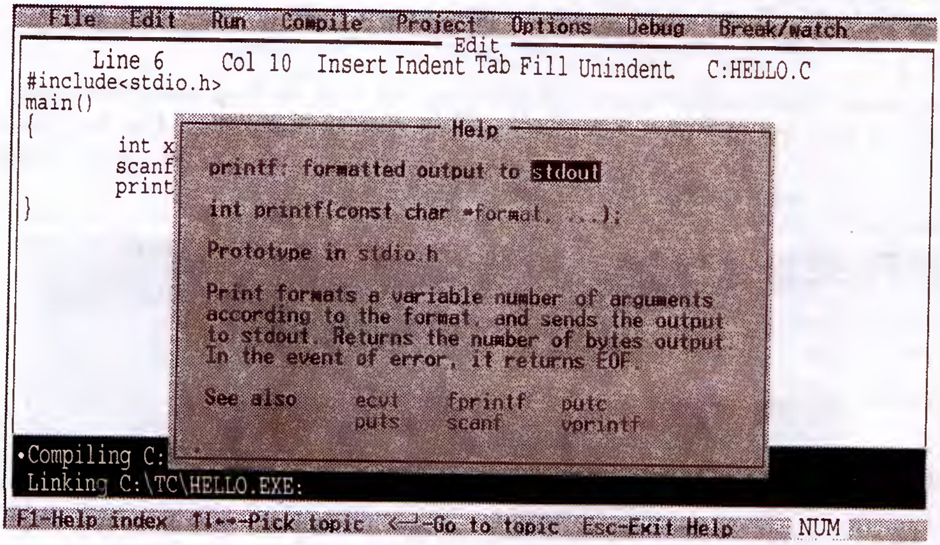


Fig. 1.4.15 Using 'help' in 'tc'

If you want to see what are the functions of various keys just press `F1' key. A help window appears on the screen displaying various commands. Fig. 1.4.16 shows this.

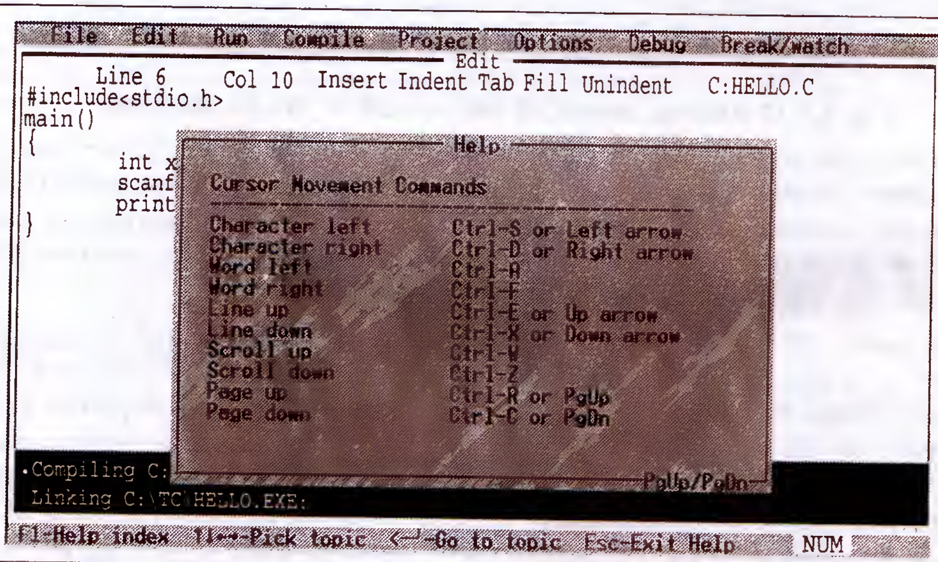


Fig. 1.4.16 Various commands displayed by help

Observe that in the bottom right corner of the window there `PgUp/PgDn` written. This means you can use Page Up and Page Down keys to see more information.

1.4.1.5 Single Stepping and Debugging the Program

You can execute one statement at a time in your program. This is called single stepping. Single stepping is very much useful to trace out errors.

Go back to your `hello.c` program in 'tc' Let's say you want to monitor the value of variable 'x'. Then adjust cursor below x, as shown.

```
int x;
```

Then select the `Break/watch` by pressing Alt and 'b' key simultaneously. (You can also select this menu by pressing F10 and ← / → arrow keys). The first submenu `Add watch ctrl-F7` is highlighted. When you press `enter', `Add watch` window is displayed to the left. This is shown in Fig. 1.4.17.

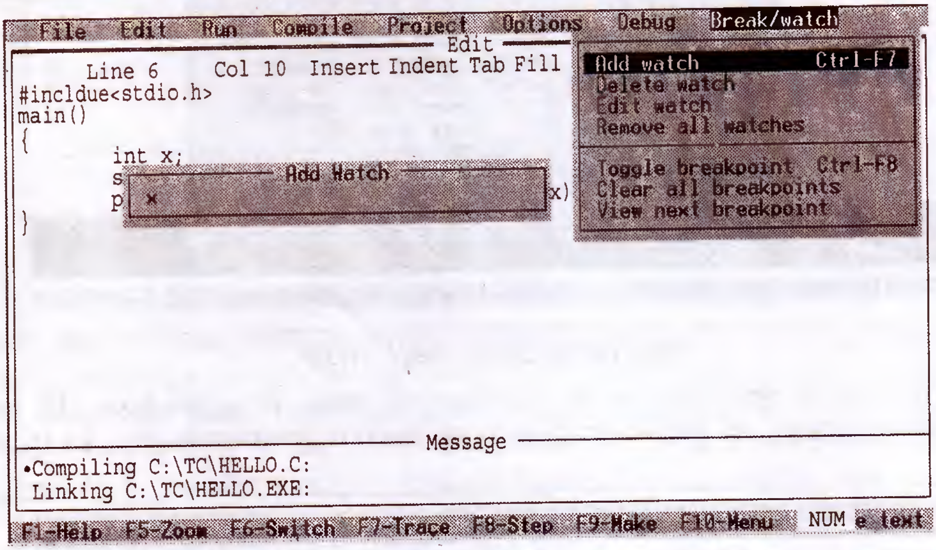


Fig. 1.4.17 Adding `watch' of the variable in single stepping

You can type any other variable name than x in this `Add watch` window. After pressing 'enter', a watch on variable 'x' is added. Fig. 1.4.18 shows what happens when you add a watch on 'x'. Observe in Fig. 1.4.18 that a '`watch`' window is created at the bottom of the display screen. And 'x' is being displayed in this window. This is shown in the Fig. 1.4.18. (Refer Fig. 1.4.18 on next page.).

Now press 'F7' for single stepping, the statement being executed is highlighted. Now `main()` is highlighted. Again press F7, `scanf` statement is highlighted. When you press F7, program executes `scanf` statement and user screen is displayed in front of you. Here you have to enter some integer numer, since `scanf` statement asks for integer number input. Let's enter a number '450' and press 'enter. Then 'tc' screen is again displayed back and next statement. i.e. `printf` is highlighted. The value of '450' is assigned to variable 'x' by `scanf` statement. Observe that '`x:450`' in the `watch`

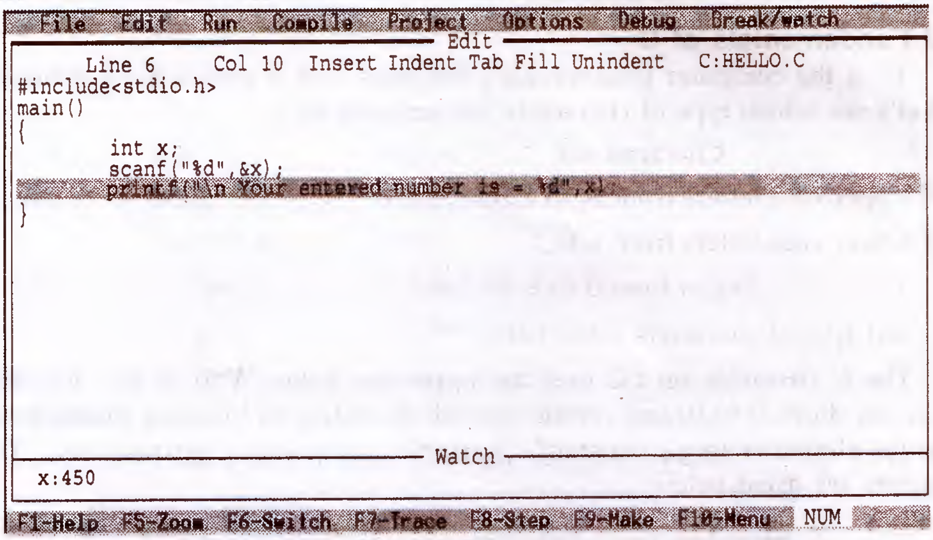


Fig. 1.4.18 Value of `x' being monitored in single stepping

window at the bottom of the screen. This is shown in Fig. 1.4.15. This is how you can see whether you are working correctly or not.

When you press 'F7' next, the program executes printf statement. Press Alt and F5 keys simultaneously to see user scrren. On the screen,

```
Your entered number is = 450
```

must be displayed. Press any key to go back to 'tc'. Thus you can debug your program by using single stepping.

Important points to be remembered when writing a program in `tc'

* *Use F3 key to load the program file in 'tc'. That file may be old or new.*
* *Always use 'c' extension to 'tc' files.*
* *Use F2 key to save your 'c' programs.*
* *Use Ctrl + F9 to run your program.*
* *Use F6 key to switch from message window to edit window and vice versa.*
* *Use F10 key to select menus at the top of the screen.*
* *If you want help on some function, adjust cursor below the name of that function and press Ctrl + F1 keys.*
* *Use 'Esc' to cancel the various windows of menus.*
* *If you want to go to DOS PROMPT from 'tc', use* **os shell** *in the* **file** *menu. For this, press Alt + F (or F10 key), go to* **os shell** *using ↑ ↓ arrow keys and 'enter' A DOS PROMPT appears on the screen. If you want to go back to 'tc', type '***exit***' and press 'enter' key.*
* *If you want to quit from 'tc', press 'Alt + X' keys.*

1.4.2 Fundamentals of C

'C' is the computer programming language and it uses a fixed defined character set. Let's see which type of characters are accepted by C.

Character set

Upper case letters from A to Z

Lower case letters from a to z

Digits from 0 to 9.

and special characters listed below,

The C character set : C uses the uppercase letters A to Z, the lowercase letters a to z, the digits 0 to 9, and certain special characters as building blocks to form basic program elements (e.g., constants, variables, operators, expressions). The special characters are listed below.

!	*	+	\	"	<
*	(	=	|	{	>
%	)	~	;	}	/
^	–	[	:	,	?
&	–	]	'	.	(blank)

Most versions of the language also allow certain other characters, such as @ and $, to be included within strings and comments.

C uses certain combinations of these characters, such as \b, \n and \t, to represent special conditions such as backspace, newline and horizontal tab, respectively. These character combinations are known as escape sequences. We will discuss escape sequences. For now we simply mention that each escape sequence represents a single character, even though it is written as two or more characters.

Any characters other than above will not be recognized by C.

Identifiers :

These are the names given by the programmer to various program elements. During programing the programmer has to give name to variables, arrays, functions etc. to idenify them from each other. These are just like the names of the people. For example *Ravi* is the identifier for the person whose name is `Ravi'.

Key words :

Some words have standard meaning in C. Those words are called keywords. For example do, `while`, `if`, `int` etc. have standard meanings. They should not be used as identifiers. Those key words are given below.

```
auto        extern      sizeof
break       float       static
```

```
case        for         struct
char        goto        switch
const       if          typedef
continue    int         union
default     long        unsigned
do          register    void
double      return      volatile
else        short       while
enum        signed
```

Data types :

'C' supports 4 basic types of data. They are as shown below.

Data type	Description
`int`	This is integer quantity. It can be a long integer or short integer.
`char`	Single character type data.
`float`	This is a floating point number. This number contains exponent or decimal point.
`double`	Double precision floating point number. More significant digits are used after decimal point. The exponent is also large.

Variables :

A variable represents a data item. Almost all the variables are identifiers. Different names are given to the variables to identify them. For example consider the following statement,

```
int  i,  j,  k;
float  x;
```

Here `i, j` and `k` are `integer' type variables. `x` is the variable of `float` type.

Arrays :

An array is a collection of variables. For example consider,

```
float  x[10];
```

This statement declares and array 'x' of ten data elements. Those data elements are floating type variables. Those data elements of an array are represented as,

`x[0]` = First data element

`x[1]` = Second data element

`x[2]` = Third data element

`x[3]` = Fourth data element

```
x[4]  =    Fifth data element
x[5]  =    Sixth data element
x[6]  =    Seventh data element
x[7]  =    Eighth data element
x[8]  =    Nineth data element
x[9]  =    Tenth data element
```

This array is called one dimensional array, since it represents a string of data elements.

Declarations :

Variables and functions are associated with some data type. The statements doing this are called declarations. For example consider the following example

```
int  i,  j,  k;
char  c;
```

Those statements are declaration statements. Variables `i`, `j`, and `k` are 'declared' as `integer` type data. Variable `c` is declared as `character` type data.

Expressions : Expression is a statement which involves arithmetic or logical operation. For example consider,

```
k  =  i  +  j;  ←  This expression represents addition of variables i&j
x  <=  y;       ←  This is logical expression and it means x ≤ y.
i++;            ←  This expression increments value of i by one.
                   (i.e.  i  =  i  +  1)
```

Statements :

```
k  =  i  +  j;          ←  This statement adds two variables.
printf  ("%d"  ,  k);   ←  This statement prints variable 'k' on the screen.
```

1.4.3 Operators and Expressions

Now let's see operators and expressions in 'C' in little more details.

Arithmetic operators :

Following are the arithmetic operators in C.

Operator	Function
+	Addition
–	Subtraction
*	Multiplication
/	Division
%	Remainder after integer division (Modulus operation)

Unary Operators : Those operators operate upon a single variable. Consider for example the following unary operators,

`-i;` ← This is unary minus operator. It makes value of variable 'i' negative.

`++i;` ← This is pre increment operator. It is equivalent to `i = i + 1`

`i++;` ← This is post increment operator.

Let's see the difference between preincrement and post increment.

Consider,

```
i = 10
printf ("%d" , ++i);
```

Since this statement uses preincrement operator, the value of 'i' is first incremented and then printed on the screen. Thus printed value will be 11. Thus the above two statements are equivalent to,

```
    i = 10;                                          i = 10;
printf ("%d" , ++i);    This is equivalent to →      i = i + 1;
                                                 printf ("%d" , i);
```

Now consider,

```
i = 10;
printf ("%d" , i++);
```

Since this statement uses post increment operator, the value of 'i' is first printed on the screen and then it is incremented. Thus printed value will be 10. Thus the above two statements are equivalent to

```
    i = 10;                                          i = 10;
printf ("%d" , i++);    This is equivalent to →  printf ("%d" , i);
                                                     i = i + 1;
```

`i--;` ← This is unary post decrement operator.
This equivalent to `i = i-1`

`--i;` ← This is predecrement operator

Both the post and predecrement operators have the same difference as that of post and preincrement operators.

Relational and logical operators :

The following are relational operators in 'C'

Operator	Meaning
<	less than
>	greater than

Operator	Meaning
<=	less than or equal to
>=	greater than or equal to
= =	equal to (this is equality operator)
! =	Not equal to (this is equality operator)

Lets consider the examples of these operators and their meaning.

Let,

```
i  =  1;
j  =  2;
k  =  3;
```

then,

`i < j` ← This is the true statement

`k > j` ← True

`(i + j) ≥ k` ← True

k = = 3 ← True

j ! = 2 ← False

Assignment operator :

These operators assign some value or expression to the variable and they can also perform some arithmetic operation at the same time.

Consider the following example

`i = j;` ← This assigns the value of `j` to i, if i = 2 & j = 50 ; then this statement makes i = 50 and j = 50

`i += j;` $\xrightarrow{\text{This is equivalent to}}$ `i = i + j;`

`i -= j;` $\xrightarrow{\text{This is equivalent to}}$ `i = i - j;`

`i * =j;` $\xrightarrow{\text{This is equivalent to}}$ `i = i * j;`

`i /= j` $\xrightarrow{\text{This is equivalent to}}$ `i = i/j;`

1.4.4 Data Input and Output

We have seen that `printf` and `scanf` functions are used to take data and print data. These two functions are important input output functions in `C'.

The `Scanf` function :

`scanf` is the formatted input function. This statement is written as,

```
scanf ( control string, arg 1, arg 2 ,  .... argn );
```

Here control string contains the formatting information.

`arg 1, arg2, ....` `argn` represent individual data items.

Consider for example, the following statements,

```
int  i;       ← Variable 'i' is integer
float  x;     ← Variable 'x' is floating type
scanf  ("%d   %f"  ,  &i  ,  &x);
```

In the above statement, the part in double quotes, i.e. `"%d  %f"` is the control string and `i` and `x` are arguments.

Here `%d` means the first input argument is `integer` type variable.

`%f` means the second input argument is `floating` type variable.

And, `&i` means the first value entered from keyboard will be assigned to variable `'i'`.

`&x` means the second value entered from keyboard will be assigned to variable `'x'`.

Here observe that each variable is preceded by & (ampersand) sign. It is essential in `scanf` statement.

Each input variable must be preceded by & (ampersand) sign in scanf statement.

You can input any number of variables at a time by `scanf` statement. This statement can be used to input arrays also.

The `printf` function :

This is formatted output function. This statement is written,

```
printf  (control  string,  arg1,  arg2,  ...  argn);
```

Here `control string` contains the formatting information of variables to be printed on the screen and `arg1, arg2, ... argn` are the variables to be printed.

Consider for example the following statement –

```
i  =  10;          ← 'i' is integer type variable
x  =  150.89;      ← 'x' is floating type variable.
printf (" %d  %f ", i, x);
```

In the above statement `"%d  %f"` is the control string and `i` & `x` are variables whose values are to be printed. When this statement is executed, the displayed values will be,

`10   150.89` ← displayed values of i & x on the screen

Here `%d` means the first argument (variable) is `integer` type

`% f` means the second argument (variable) is `floating` type.

Observe the biggest difference in `printf` and `scanf` that **there is no & (ampersand) sign before the variables in `printf` statements.**

The characters used in the control strings of `scanf` and `printf` statements have almost similar meanings. Some of those characters and their meanings are given below for both `printf` and `scanf` functions.

Character in control string	Meaning
%C	This is 'character' type data.
%d	This is a decimal integer type data.
%ld	This is long integer type data.
%f	This is floating type data.
%lf	This is double precision floating type data.
%e	This is floating type data but displayed with exponent.
%o	Data type is octal integer.
%s	This is a string.
%x	Data type is hex integer.

You can use `printf` statement in various ways.

Here are some of the examples –

```
i=10;
x=158.29;
printf ("following are the values of i & x");
printf ("\n i=%d", i);
printf ("\n x=%lf", x);
```

When those statements are executed the outputs will look like as shown below.

Following are the values of i & x

```
i=10;
x=158.29
```

Here we have printed just a message of characters by the first `printf` statement. The second and third `printf` statements print values of `i` & `x` respectively.

Observe that message is printed on first line and `i` and `x` are also printed on separate lines. This is not because of the three separate `printf` statements. But printing on the 'new line' is done by `'\n'` in the control string. Observe that `'\n'` is not printed on the screen, but it is used like `%d` & `%lf` in the control string.

'\n' is used in printf statement to display data on the next line.

Thus all the three `printf` statements can be combined to a single statement as shown below –

```
printf("following are the values of i&x\ni=%d\nx=%lf", i,x);
```

Thus here wherever there is \n, the display will go to next line.

Consider the next example,

```
x=4.472839126;  ← 'x' is a floating point number
printf ("x=%5.3lf", x);
```

After executing the above statement the display on the screen will be,

```
x=4.473
```

Thus only first three significant digits after decimal point are printed. The third digit is rounded to 3. In the actual value of x there are many digits after decimal point.

`%5.3lf` → This decides how the number is to be printed.

Five digits before decimal and three digits after decimal point will be printed on the screen.

Consider,

```
printf("x= %4.15lf", x);
```

This statement will print 4 digits before decimal point and 15 digits after decimal point.

Consider the following statements,

```
i=10;
j=25;
printf("i+j = %d \n i*j=%d", i+j, i*j);
```

After executing the above statements the display will be

```
i+j=25
i*j=150
```

In the above `printf` statement, the first argument is `i+j` and second argument is `i*j`. Thus `printf` statement accepts 'expressions' also as an arguments.

`printf` and `scanf` statements are defined in `stdio.h`. Hence always `stdio.h` should be included in the program.

1.4.5 Control Statements

The control statements are mainly used for looping operations. They normally check some condition and keep on executing some part of the program until the condition is not satisfied.

While statement :

The format of this statement is,

```
while (expression) statement
```

Let's consider the example program to illustrate this statement.

```
# include<stdio.h>
main()
{
    int digit;
    digit = 0;
    while (digit <= 9)
    {
        printf ("\n %d", digit);
        ++digit;
    }
}
```

This is the complete program to display the digits from 0 to 9.

The first statement is,

```
# include<stdio.h>
```

This statement includes `stdio.h` in the program. This is necessary since we are using `printf` in the program. Then the program is written in function `main()` observe that the complete program is enclosed by { & } brackets at the start and end.

The next statement is,

```
int digit;
```

This is variable declaration statement. It declares the variable name `digit` as an `integer` type data.

```
digit=0;
```

This statement is an assignment statement. It assigns a value of 'zero' to `digit`. Next there is `while` control loop to display the digits from 0 to 9. Let's see this loop in detail.

```
While   (digit  <=  9)
                           expression
{
    printf  ("\n &d",   digit);
    ++digit;                           statement
}
```

Here `(digit <= 9)` is the expression of `while` loop and two statements inside `'{'&'}'` brackets is the statement of `while` loop. Thus,

While (expression)

{

statements

}

When the program first enters `while` loop, the value of `digit=0`. Since zero is less than 9, `while` loop executes statements. The `printf` statement then displays value of `digit` as zero on the screen. Since '`\n`' exists in the control string of `printf`, the value of `digit` is always printed on next line. The next statement in `while` loop is,

`++digit;`

This increments the value of `digit=1`. Again this value is less than '9', hence `while` loop executes its statements. Thus on the screen the output will be,

0

1

2

3

4

5

6

7

8

9

After printing '9' on the screen, `++digit` statement increments value of `digit`. This makes `digit=10`. Since `digit` is greater than 9, the `expression` (i.e. `digit <= 9`) of `while` loop is not satisfied and program comes out of the loop.

Write this program in your computer. After running this program see whether you get the same result.

do-while statement :

This control loop is written as,

```
do
{
    statement
} while (expression)
```

Using do-while loop we can write the program of displaying numbers from 0 to 9 as follows,

```
#include<stdio.h>   ← Include stdio.h since we are using input/output
                                                        statements.
main ()
{
    int digit;
    digit=0;
    do
    {
        printf ("\n %d", digit);
        ++digit;
    } while (digit <=  9);
}
```

The function performed by this program is exactly similar to the previous one. Type and compile this program on your computer and see the results.

The for statement :

This is the most widely used control statement in C. This control statement is written as,

```
for (expression 1; expression 2; expression 3)
{
    statements
}
```

Here	expression 1	is normally initialization statement.
	expression 2	is normally a condition which is checked during every pass of the loop.
	expression 3	This modifies the variables being checked in expression 2

Using for loop the program to print digits from 0 to 9 can be written as follows –

```
# include<stdio.h>
main ( )
{
    int digit;
    for (digit=0; digit <=9; ++digit)
    {
        printf ("\n %d", digit);
    }
}
```

Here `digit=0;` is the initialization statement
`digit<=9;` is the condition checked by `for` loop
`++digit` This increments `digit` in every cycle.

Thus `for` loop combines the other two statements also. Initially `digit=0` by the `for` loop. Simultaneously the condition `digit <= 9` is checked. Then value of `digit` is printed on the screen. After this value of `digit` is incremented by the `++digit` expression. The program remains in loop till value of `digit` is less than or equal to 9.

In the `for` loop observe that there is semicolon (;) after `expression 1` & `expression 2`, but there is no semicolon (;) after `expression 3`.

`if-else` statement :

This is a very simple statement. The format of this control statement is,

```
if (expression) statement 1
else statement2
```

Here if an `expression` is true, then it executes `statement 1`. If an `expression` is false, then it executes `statement 2`.

Consider the following parts of the program,

```
i=4;j=10;
if (i > j) k=i*j;
else k=i+j;
```

Here since `i` is not greater than `j`, the `else` part is executed & `k=14`. If there are more than one statement, then `'{'` & `'}'` brackets must be used.

`Continue` statement :

This statement is used to bypass remainder part of the loop. Consider the following program. This program we have discussed to print digits from 0 to 9. Let's say we want to print digits only upto 4.

```
# include<stdio.h>
main ( )
{
    int digit;
    for (digit=0; digit<=9; ++digit)
    {
        if (digit>4) continue;
        printf ("\n %d", digit);
    }
}
```

There is following statement in the above loop,

```
if (digit > 4) continue;
```

before the printf statement. If value of digit becomes higher than 4, then printf statement is bypassed. Thus this program will print the number which are less than or equal to 4.

1.4.6 Functions

Function are basically small programs which are given a particular task. Let's say we want to calculate a factorial of a number repeatedly. Then the same code repeats in the program. This can be avoided by writing a function for factorial and that function is called wherever required. Consider the complete program to calculate the factorial of numbers from 1 to 5 and display it on the screen.

```
# include<stdio.h>
main ( ) ← This is main function
{
    int fact (int i); /*declaration of the function */
    int i, k;
    clrscr( ); ← This is standard clear screen function to clear the screen.
    for (i=1; i<=5; i++)
    {
      k=fact (i);← This statement calls the function fact. It passes value of
                                                        i to function.
      printf ("\n The factorial of %d is =%d", j, k);
         This statement prints the number and its factorial on the screen.
    }
} ← 'main' function ends here
```

```
int fact (int i)  ← This is independent function to calculate factorial
{
    int n prod;      This statement is comment, it is not executed.
    prod=1; /* Make initial value of factorial to one */
    for (n=1; n<=i; n++)
    {
       prod; = prod*n;/* calculation of factorial */
    }
    return (prod);
}
```

Observe that the first statement in main ( ) function is declaration of the function fact which calculates factorial of a number.

```
int          fact(int i);/* declaration of the function */
 ↑                ↑
data type       data type
returned by function   accepted by function
```

This means the function accepts 'integer' type data and returns 'integer' type data. If program is using some function, it must be declared in that program. On the same line there is,

`/* declaration of the function */`

This is the comment about the statement.

Any thing which is between `/* ........ */` is considered as comment and is not executed by C.

In the `for` loop the first statement is,

`k = fact (i);`

This statement calls function `fact` in the `main` program to calculate value of factorial. The function `fact` calculates the factorial of 'i' and assigns it to variable 'k'. The next `printf` statement prints the number and its factorial on the screen.

When function `fact` is called by the statement

`k = Fact (i);`

In the `main` program, the execution goes to function `fact`. After the above statement, program goes to first statement of `fact` i.e.,

`int fact (int i)` ← value of 'i' from main program is assigned here

Then the remaining statements of function calculate factorial. The last statement is,

`return (prod);`

This means function 'returns' the value of `prod` to the `main` program. Here `prod` contains the factorial of `i`. The execution again goes back to

```
k = fact (i);
```

statement in `main` program and `k` is assigned the factorial of `i`. In other words,

$$k = \text{prod} \qquad \text{and} \qquad \text{prod} = i!$$

Here 'prod' is calculated by function `fact`.

compile and run this program on your computer.

If the function accepts a floating point number x and returns double precision number, then declaration of such function will be,

```
double 'function name' (float x);
```

Here 'function name' should be given.

1.4.7 Two Dimensional Arrays

Two dimensional arrays are used widely in numerical techniques. Consider the matrix,

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 4.5 & 2 & 3.1 \\ 8 & 1 & 6.8 \\ 2 & 9 & 7.2 \end{bmatrix}$$

To process such matrices effectively in 'C', two dimensional arrays are used. The two dimensional arrays can be declared as,

```
Float a[3][3];
```

This is the declaration of two dimensional array of 3 × 3 size. The data type of an array is `float`. We can assign the elements of matrix A to an array a. The counting of elements in an array starts from zero. Thus first element in array a is `a[0][0]`. Thus we can assign,

$$\left.\begin{array}{l} \texttt{a[0] 0]} = a_{11} \\ \texttt{a[0][1]} = a_{12} \\ \texttt{a[0][2]} = a_{13} \end{array}\right\} \text{row 1}$$

$$\left.\begin{array}{l} \texttt{a[1][0]} = a_{21} \\ \texttt{a[1][1]} = a_{22} \\ \texttt{a[1][2]} = a_{33} \end{array}\right\} \text{row 2}$$

$$\left.\begin{array}{l} \texttt{a[2] 0]} = a_{31} \\ \texttt{a[2][1]} = a_{32} \\ \texttt{a[2][2]} = a_{33} \end{array}\right\} \text{row 3}$$

You can use an array `a[4][4]` of 4 × 4 elements also to store this matrix A. i.e.,

$$\left.\begin{array}{l} \texttt{a[0][0]} = \\ \texttt{a[0][1]} = \\ \texttt{a[0][2]} = \\ \texttt{a[0][3]} = \\ \texttt{a[1][0]} = \end{array}\right\} \text{Nothing is stored here}$$

$$\left.\begin{array}{l} \texttt{a[1][1]} = a_{11} \\ \texttt{a[1][2]} = a_{12} \\ \texttt{a[1][3]} = a_{13} \end{array}\right\} \text{row 1}$$

$\texttt{a[2][0]}$ = Nothing is stored here

$$\left.\begin{array}{l} \texttt{a[2][1]} = a_{21} \\ \texttt{a[2][2]} = a_{22} \\ \texttt{a[2][3]} = a_{23} \end{array}\right\} \text{row 2}$$

$\texttt{a[3][0]}$ = Nothing is stored here

$$\left.\begin{array}{l} \texttt{a[3][1]} = a_{31} \\ \texttt{a[3][2]} = a_{32} \\ \texttt{a[3][3]} = a_{33} \end{array}\right\} \text{row 3}$$

Observe that there are total 16 elements in 4×4 array. Thus we can use higher size arrays for lower size matrices. Those arrays can be effectively processed by `for` control statements.

❑❑❑

2
Errors in Numerical Computation

2.1 Introduction

2.1.1 Why to Study Errors ?

At the very beginning of this subject we are discussing errors. The term *error* can be broadly defined as the difference between true value and approximate value. In the previous chapter we have seen that numerical techniques are used to solve engineering problems. Numerical techniques lead to approximate solutions. Hence there is possibility of *error* in the solution obtained by numerical techniques. The basic idea of any numerical technique is to reduce the error and obtain more and more accurate solution. Hence we have to define various types of errors. We have to discuss the propagation of errors in the numerical techniques.

For most of the applications, error tolerance is specified. Hence the numerical technique should limit the error within the specified tolerance. If error tolerance is exceeded, then severe penalty is to be paid or failure of the system takes place.

2.1.2 What We Will Discuss About Errors ?

In this chapter we will define errors and their sources. Numerical methods are implemented on computers. Computers can handle the variables with limited number of digits. For example, $\frac{10}{6} = 1.666666666....$ If the computer can handle only four digits, than $\frac{10}{6} = 1.667$ by rounding. Thus error is introduced in the calculation. In this chapter we will discuss how numbers are represented in computers and their calculations. Error propagation and numerical in stability is also discussed.

2.1.3 Significant Digits

Definition : Significant digits of a number are those that can be used with confidence.

Types : 1) ***Total significant digits*** : These are the number of digits that are present in the number. For ex. 2.314 has four significant digits.

2) ***Significant digits after decimal point*** : These are the number of significant digits after decimal point. For Ex. 2.314 has three significant digits afeter decimal point.

Let us consider following values of π

(Table 2.1.1. see on next page.)

Table 2.1.1 Different significant digits in π

Sr No.	Value of π	Comments
1.	$\pi = 3.14159265358979323....$	This is exact value. It contains infinite number of significant digits.
2.	$\pi = 3.1415926536$ This digit is rounded. Nine significant digits are correct after decimal point. Hence value of π is said to be correct upto 9 significant digits or 9 decimal places.	This is approximate value. It has 10 significant digits after decimal point or total 11 significant digits. Last digit is rounded. Hence it is in error.
3.	$\pi = 3.141593$ This digit is rounded. Correct digits	There are 6 significant digits after decimal point or total 7 significant digits. Value of π is correct upto 5 decimal places.

From the above table, it is clear that significant digits are the number of digits that are used to represent a number *reliably.*

As more significant digits are used, the number is more accurately represented. Hence more significant digits reduce the error in number representation.

Let us clear some doubts and confusion

Consider various numbers given in following table.

Table 2.1.2 : Trailing and leading zeros in numbers

Sr.No.	Numbers	Comments
1.	0.00002853 $= 0.2853 \times 10^{-4}$ 0.0002853 $= 0.2853 \times 10^{-3}$ 0.002853 $= 0.2853 \times 10^{-2}$ 0.02853 $= 0.2853 \times 10^{-1}$ 0.2853	All these numbers have only four significant digits, since zeros are used only to locate the decimal point.
2.	6.31×10^3 ← Three significant digits 6.310×10^3 ← Four significant digits 6.3100×10^3 ← Five significant digits	Trailing zeros are used to represent the significant digits of a number.

From the table 2.1.2, it is clear that zeros after decimal point are used to locate the decimal point. They doesnot represent significant digits. But trailing zeros are used to represent the number of significant digits.

2.1.4 Accuracy and Precision of Numbers

Accuracy is defined as the closeness of calculated value to the exact value. Following table illustrates accuracy.

Table 2.1.3 Illustration of accuracy

Sr. No.	Exact or true value	Approximate values	Comments
1.	4.2138	4.1182	Less accurate
		4.2146	More accurate
		3.9392	Least accurate

The above table shows that, most accurate value means lowest error. Thus high accuracy means low error.

Precision means repeatativeness of the values. High precision means, individual values of the same variable repeat. These values may or may not be accurate. Following table illustrates this concept.

Table 2.1.4 Accuracy and precision of numbers

Sr. No.	Exact or true value	Approximate values	Comments
1	4.2138	4.1182 4.1183 4.1181 4.1182	These values are highly precise but less accurate.
2.	4.2138	4.2137 4.2139 4.2137 4.2136	These values are highly precise and highly accurate

The inaccuracy and imprecision are reflected in error definitions. The solutions obtained by numerical techniques must be accurate and precise as far as possible.

Exercise

1. Explain the term significant digits. What is its importance ?
2. Explain accuracy and precision of numbers.

2.2 Floating Point Numbers

Computers handle finite number of significant digits of a number. These number of digits depend upon data bus width of the processor. Floating point format is used in computers to represent numbers. Integers as well as fractions are represented in floating point format. The fractional part of a number is expressed as mantissa or significant, and integer part is called exponent i.e.,

$$\text{Floating point number} = mb^e \qquad \text{... (2.2.1)}$$

Here m is the mantissa,

b is the base of the number system

(i.e. binary, hex or decimal)

e is the exponent.

For example 0.3875×10^2 is the floating point number. It has mantissa of 0.3875 and exponent of 2. The base of the number system is 10, i.e. decimal number system. Let us consider another example,

$$X = 24.389 \times 10^2$$

This number can be represented in floating point format as,

$$X = \underset{\text{Mantissa}}{24.389} \times \underset{\text{Exponent}}{10^2}$$

This number can be stored in memory as,

Floating point representation of number X in memory = 24.389 E 2

Here E stands for exponent.

2.2.1 Normalized Floating Point Numbers

For the normalized floating point number the mantissa is greater than or equal to 0.1 and less than 1. i.e.,

For normalized floating point number,

$$0.1 \leq \text{Mantissa} < 1 \qquad \text{... (2.2.2)}$$

The exponent is arranged accordingly. Again consider the above number,

$$X = 24.389 \times 10^2$$

When this number is normalized, then we have to shift decimal point to left by two digits and adjust the power of ten. i.e.,

$$X \text{ (normalized)} = 0.24389 \times 10^4 \qquad \text{... (2.2.3)}$$

Consider another number given as,

$$Y = 0.00078634,$$

Then normalized floating point representation of this number will be,

$$Y \text{ (normalized)} = 0.78634 \times 10^{-3} \qquad \text{... (2.2.4)}$$

For this number, Mantissa = 0.78634 and Exponent = –3

∴ Floating point representation of Y = 0.78634 E–3

Similarly for 'X' given by equation 2.2.3

$$X = 0.24389 \times 10^4$$

Mantissa of X = 0.24389 and Exponent of X = 4

∴ Floating point representation of X = 0.24389 E4

If we use 6 digits to store floating point number in the memory, then the number will be stored as follows.

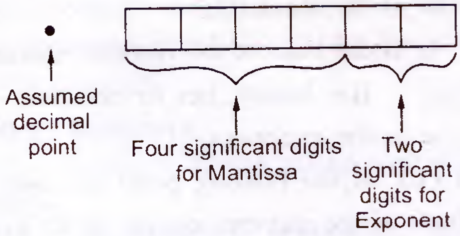
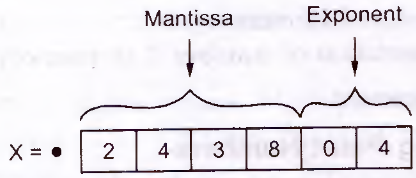


Fig. 2.2.1 Storing of normalized floating point number in memory

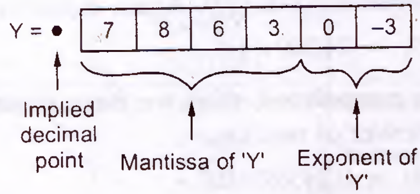
Thus number `X' will be stored according to Fig 2.2.1 as,

$$X = 0.24389 \text{ E4}$$



Here we have truncated the mantissa of X to four significant digits to accommodate into four digits similarly,

$$Y = 0.78634 \text{ E–3}$$



With this type of floating point representation of numbers, six digits can store maximum and minimum numbers of:

$$\text{Maximum number} = 0.9999 \times 10^{99}$$

$$\text{or} \quad \text{Minimum number} = 0.1000 \times 10^{-99}$$

Thus very much large range is covered because of normalized floating point representation in computers. At the same time, the number of digits in memory or on data bus required to represent such numbers are less.

2.2.2 Arithmetic Operations with Floating Point Numbers

1. Addition and Subtraction :

To add or subtract floating point numbers we follow following steps :

i) Normalize the given numbers.

ii) Make their exponents equal by adjusting decimal point.

iii) Perform addition or subtraction of mantissa of the two numbers.

iv) Normalize the result.

Note that the exponents are not added or subtracted.

Ex. 2.2.1 : *Add the following floating point numbers.*

a) 0.3879 E7 and 0.813 E7

b) 723.813 E14 and 89.73 E12

c) 100.312 E25 and 81.813 E27

Sol. : a) Here the given two floating point numbers are normalized. Therefore we can add them directly.

```
      0.3879  E7
    + 0.813   E7
      1 1 1
   ----------------
      1.2009  E7    Exponent of the result is same as
                    that of numbers
   Only mantissa are
        added
```

Here result should be normalized.

$\therefore$ Ans = 0.12009 E8

(b) 723.813 E14 and 89.73 E12

Here first normalize the given numbers.

i.e. 723.813 E14 = 0.723813 E17 and 89.73 E12 = 0.8973 E14

Then we make exponents equal. We will make exponent of 2^{nd} number to 17.

i.e. 0.8973 E14 = 0.0008973 E17

Now adding the two numbers,

```
    0.723813   E17
  + 0.0008973  E17
        1 1 1
  ----------------
    0.7247103  E17
```

The result in normalized form

(c) 100.312 E25 and 81.813 E27

First normalize the two numbers

100.312 E25 = 0.100312 E28 and 81.813 E27 0.81813 E29

Here we will make exponent of first number to 29. i.e.,

0.100312 E28 = 0.0100312 E29

Now adding the two numbers,

$$\begin{array}{r l} & 0.0100312 \ E29 \\ + & 0.818133 \quad E29 \\ \hline & 0.8281642 \ E29 \end{array}$$

The result is normalized, hence it is the answer.

Ex. 2.2.2 *Subtract following floating point numbers*

a) 0.4189628 E5 and 0.23818 E5 b) 583.1863 E20 and 78.1671 E19

Sol. : a) 0.4189628 E5 and 0.23818 E5 Both the numbers are normalized therefore we can subtract directly. i.e.,

$$\begin{array}{r l} & 0.4189628 \quad E5 \\ - & 0.23818 \qquad E5 \\ \hline & \underbrace{0.187828}_{\text{Only mantissa are subtracted}} \quad E5 \end{array}$$

Exponent of the result is same as that of numbers

The result of subtraction is thus,

Result = 0.1807828 E5. It is normalized.

b) 583.1863 E20 and 78.1671 E19

First we will normalize the numbers.

583.1863 E20=0.5831863 E23 and 78.1671 E19 = 0.781671 E 21

Here we will make exponent of second number to 23.

i.e. 0.781671 E21 = 0.00781671 E23.

Now subtracting the numbers.

$$\begin{array}{r l} & 0.58318630 \quad E23 \\ - & 0.00781671 \quad E23 \\ \hline & 0.57536959 \quad E23 \end{array}$$

The result is in the normalized form.

2. Multiplication of the floating point numbers :

To multiply the two floating point numbers following steps should be followed.

i) Multiply the mantissa of two numbers

ii) Add the exponents and

iii) Normalize the result

Ex. 2.2.3 : *Multiply the following two numbers 3.1897 E10 and 0.18631 E11.*

Sol. : Here multiply the mantissa and add the exponents.

$$\begin{array}{r r l} & & 3.1897 \quad E10 \\ & \times & 0.18631 \quad E11 \\ \hline \text{Multiply mantissa} \rightarrow & & 0.594273 \quad E21 \leftarrow \text{Add exponents} \end{array}$$

The result is in normalized form.

3. Division of the floating point numbers :

To divide two floating point numbers, we perform following steps -

i) Divide mantissa of the two numbers

ii) Exponent of the result = Exponent of numerator – Exponent of denominator.

iii) Normalize the result.

Ex. 2.2.4 : *Evaluate*

$$\frac{41.9875 \;\; E16}{378.1619 \;\; E11}$$

Sol. : Here we will divide mantissa and subtract exponents.

$$\therefore \quad \text{Result} = \frac{41.9875}{378.1619} E(16-11)$$

$$= 0.1110304\ E(5)$$

$$= 0.1110304\ E5$$

The result is normalized.

Ex. 2.2.5 *Addition of small and large number.*

Add the small number 0.0010 to large number 4000, using a computer with 4-digit mantissa and 1-digit exponent.

Sol : Let us first normalize the numbers to 4-digit mantissa.

$$4000 \Rightarrow 0.4000\ E4$$

$$0.0010 \Rightarrow 0.1000\ E\text{-}2$$

Since we have to perform addition, the exponents of two numbers must be made equal. In such case we make all exponents equal to the highest exponent. Here we have to make exponent of 2^{nd} number equal to 4. i.e.,

$$0.1000\ E\text{-}2 \Rightarrow 0.0000001\ E4$$

Now let us add the two numbers. i.e.,

$$\begin{array}{r} 0.4000 \quad\quad E4 \\ \underline{+\ 0.0000001\ E4} \\ 0.4000001\ E4 \end{array}$$

The result must have 4-digit mantissa. Hence rounding to 4 significant digits, above result becomes :

$$0.4000001 \quad E4 \Rightarrow 0.4000\ E4\ !$$

This is equal to the first number only. Thus as if addition is not performed. This result shows that error can be introduced when large number and small number are added.

Ex. 2.2.6 *Suppose an estimate of sine function upto 10 terms is to be calculated. The summation is done in two ways, from the first term to the tenth term or alternatively from tenth term to the first term. Which yields a more accurate estimate and why ? What type of error is involved ?* **May-2000, 8 Marks**

Dec-1996, 6 Marks

Sol. : The Taylor series expansion of sin x is given as,

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots\ldots + \frac{(-1)^{n+1} x^{2n-1}}{(2n-1)!}$$

First 10 terms of this expansion are calculated for x = 1 radian. Note that these terms are calculated by a computer program with double precision. Table 2.1.5 lists these 10 terms in the expansion of sin x.

Table 2.1.5 First 10 terms in the expansion of sin x for x = radians

S.N.	Value of the term	Normalized floating point representation
1.	1	0.1000000 E1
2.	– 0.1666667	– 0.1666667 E0
3.	0.8333333×10^{-2}	0.8333333 E-2
4.	$-0.1984126 \times 10^{-3}$	– 0.1984126 E-3
5.	0.2755732×10^{-5}	0.2755732 E-5
6.	$-0.2505211 \times 10^{-7}$	– 0.2505211 E-7
7.	0.1605904×10^{-9}	0.1605904 E-9
8.	$-0.7647164 \times 10^{-12}$	– 0.7647164 E-12
9.	$0.2811457 \times 10^{-14}$	0.2811457 E-14
10.	$-0.8220636 \times 10^{-17}$	– 0.8220636 E-17

Table 2.1.6 Addition from 1st term towards 10th term

Terms being added	Exponents made equal for addition	Results (4-digit mantissa)	Comments
0.1000000E1 ⇒	0.1000000E1		
−0.1666667E0 ⇒	−0.0166667E1		
(First two terms of sin x)	0.0833333E1 ⇒	0.8333 E0	
0.8333 E0 ⇒	0.8333 E0		
0.8333333E−2 ⇒	+0.0083333E0		
(3rd term of sin x)	0.8416333E0 ⇒	0.8416 E0	
0.8416 E0 ⇒	0.8416 E0		
−0.1984126E−3 ⇒	−0.0001984E0		
(4th term of sin x)	0.8414016E0 ⇒	0.8414 E0	These two results are equal. Since large number is added with small number.
0.8414 E0 ⇒	0.8414 E0		
−0.2755732E−5 ⇒	+0.0000027E0		
(5th term of sin x)	0.8414027E0 ⇒	0.8414 E0	

Table 2.1.6 shows that there will be no change in the result after the addition of 5th and higher terms. Note that this problem arises because of adding large number with small number. Truncation or rounding of the result takes place to 4-digit mantissa. This introduces error in the result. This type of error is also discussed n example 2.2.4.

Reverse order addition :

Table 2.1.7 illustrates the addition of terms in reverse order. As shown in the table, 10th and 9th terms are added first. Then the result of these terms is added to 8th term and so on.

Table 2.1.7 Addition from 10^{th} term towards 1^{st} term

Terms being added	Exponents made equal for addition	Results (4-digit mantissa)	Comments
-0.8220636 E$-18 \Rightarrow$	-0.0008221 E-14		No repeatition of results since terms of comparable magnitudes are added $\Downarrow$
0.2811457 E$-14 \Rightarrow$	0.2811457 E-14		
(9^{th} & 10^{th} terms of sinx)	0.2803236 E$-14 \Rightarrow$	0.2803 E-14	
0.2803 E$-14 \Rightarrow$	0.002803 E-12		
-0.7647164 E$-12 \Rightarrow$	-0.7647164 E-12		
(8^{th} term of sin x)	-0.7619134 E$-12 \Rightarrow$	-0.7619 E-12	
-0.7619 E$-12 \Rightarrow$	-0.0007619 E-9		
0.1605904 E$-9 \Rightarrow$	0.1605904 E-9		
(7^{th} term of sin x)	0.1529714 E$-9 \Rightarrow$	0.1530 E-9	
0.1530 E$-9 \Rightarrow$	0.001530 E-7		
-0.2505211 E$-7 \Rightarrow$	-0.2505211 E-7		
(6^{th} term of sin x)	-0.2489911 E$-7 \Rightarrow$	-0.2490 E-7	

As shown in the above table, the result of addition of 9^{th} and 10^{th} term is comparable to 8^{th} term. Hence there is less effect on the result. Hence errors are reduced.

Conclusion

1. Reverse order addition yields more accurate estimate of sin x. This is because, the accumulated sum is comparable to new term being added. But in forward order addition the large term is added to small term and results in more error.
2. In this operation, truncation. as well as rounding errors are involved.

Ex. 2.2.7 *Subtractive cancellation. Determine the roots of the quadratic equation :*

$$x^2 - 400x + 1 = 0$$

Use four significant digits for mantissa.

Sol. : The generalized quadratic equation is written as,

$$ax^2 + bx + c = 0$$

Comparing with the given quadratic,

$$a = 1$$
$$b = -400$$
$$c = 1$$

Let us convert these values to normalized floating point representation i.e.,

$$a = 0.1\ E1$$
$$b = -0.4\ E3$$
$$c = 0.1\ E1$$

The roots of the quadratic equation are given by,

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

Putting values in above equation,

$$x = \frac{0.4E3 \pm \sqrt{(-0.4E3)^2 - 4(0.1E1)(0.1E1)}}{2(0.1\ E1)}$$

$$= \frac{0.4\,E3 \pm \sqrt{0.16\ E6 - 0.4\ E1}}{0.2\ E1}$$

The exponents of terms under square root must be made equal so as to subtract them. Hence making exponent of 2^{nd} term to 6, then we have,

$$x = \frac{0.4\,E3 \pm \sqrt{0.16\ E6 - 0.000004\ E6}}{0.2\ E1}$$

$$= \frac{0.4\,E3 \pm \sqrt{0.160004\ E6}}{0.2\ E1}$$

Since 4-significant digits are used for mantissa, the term under square root will be 0.160004 E6 ≈ 0.1600 E6. Hence above equation becomes,

$$x = \frac{0.4\,E3 \pm \sqrt{0.1600\ E6}}{0.2\ E1}$$

$$= \frac{0.4\,E3 \pm 0.4\,E3}{0.2\,E1}$$

$$= 0.4\ E3 \text{ and } 0\ !$$

This shows that one root is zero. Actually the two roots are :

$$x = \frac{400 \pm \sqrt{(400)^2 - 4(1)(1)}}{2(1)}$$

$$= 399.9975 \text{ and } 0.0025$$

Thus error is introduced in the first root but second root becomes zero. This happens because of subtractive cancellation. Here $b^2 >> 4ac$. Very large number and small number are subtracted. Due to 4-digit mantissa, b $\approx \sqrt{b^2 - 4ac}$. Hence error is introduced in the result. The small root is given by

$$x = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Let us rearrange as follows :

$$x = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \times \frac{-b + \sqrt{b^2 - 4ac}}{-b + \sqrt{b^2 - 4ac}}$$

$$= \frac{4ac}{2a(-b + \sqrt{b^2 - 4ac})}$$

$$= \frac{2c}{-b + \sqrt{b^2 - 4ac}}$$

Putting values in above equation,

$$x = \frac{2(0.1\,E1)}{0.4\,E3 + \sqrt{(-0.4\,E3)^2 - 4(0.1\,E1)\,(0.1\,E1)}}$$

$$= \frac{0.2\,E1}{0.4\,E3 + \sqrt{0.16\,E6 - 0.4\,E1}}$$

The exponents of terms under square root must be made equal,

$$x = \frac{0.2\,E1}{0.4\,E3 + \sqrt{0.16\,E6 - 0.000004\,E6}}$$

$$= \frac{0.2\,E1}{0.4\,E3 + \sqrt{0.160004\,E6}}$$

Making 4-digit mantissa of term under square root. i.e. 0.160004 E6 $\approx$ 0.1600 E6. Then above equation becomes

$$x = \frac{0.2\,E1}{0.4\,E3 + \sqrt{0.1600\,E6}}$$

$$= \frac{0.2\,E1}{0.4\,E3 + 0.4\,E3}$$

$$= 0.0025$$

Thus the small root is recovered. Note that now the terms in the denominator doesnot cancel each other. Thus subtractive cancellation can be avoided.

Ex. 2.2.8 *Calculate the middle value of a = 4.568 and b = 6.762 using four digit arithmetic.*

Sol. : Let us first normalize the number s and represent them with 4-digit mantissa. i.e.

$$a = 4.568 \Rightarrow 0.4568\ E1$$

$$\text{and } b = 6.762 \Rightarrow 0.6762\ E1$$

The middle value is given as,

$$\text{Middle value} = \frac{a+b}{2}$$

$$= \frac{0.4568E1 + 0.6762E1}{2}$$

Since the exponents of the two numbers in numerator are equal, they can be added directly i.e.,

$$\text{Middle value} = \frac{1.133E1}{2}$$

$$= \frac{0.1133E2}{2}$$

$$= 0.05665\ E2$$

Since we are using 4-digit mantissa, 0.05665 ≈ 0.0567 after rounding to 4-significant digits. Hence,

$$\text{Middle value} = 0.0567\ E2 = 5.67$$

There is alternate formula for calculating middle value i.e,

$$\text{Middle value} = a + \frac{b-a}{2}$$

Putting values in above expression,

$$\text{Middle value} = 0.4568E1 + \frac{0.6762E1 - 0.4568E1}{2}$$

$$= 0.4568\ E1 + 0.1097\ E1$$

$$= 0.5665\ E1 = 5.665$$

And the exact value is,

$$\text{Middle value} = \frac{a+b}{2}$$

$$= \frac{4.568 + 6.762}{2}$$

$$= 5.665$$

Comment

Thus 2^{nd} method of calculation gives more accurate result. In the first method, we perform $\frac{a+b}{2}$. The addition (a+b) exceeds 4-digit mantissa and rounding is necessary. Hence error is introduced. But in second case all calculations (b – a) are performed within 4-digit mantissa and no truncation or rounding is required.

Exercise

1. Explain floating point representation of numbers. What is the importance of normalized floating point representation.
2. Explain how various arithmetic operations are performed on normalized floating point numbers.

Unsolved Problem

1. Express the following numbers to normalized floating point representation. (Use 4 significant digits for mantissa and two digits for exponent).

   i) 23.873 **[Ans. : 0.2387 E01]**

   ii) 4875.676×10^3 **[Ans. : 0.4875 E07]**

   iii) $0.000003125 \times 10^{-3}$ **[Ans. 0.3125 E–8]**

   iv) $0.00010000631 \times 10^2$ **[Ans. : 0.1000 E05]**

University Question

1. Suppose an estimate of the sine function upto 10 terms is to be calculated. The summation is done in two ways, from the first term to the tenth term or alternatively from the tenth term to the first term. Which yields a more accurate estimate and why ? What is the type of error involved ? **[Dec - 96, May-2000]**

2.3 Types of Errors

Numerical techniques use approximate values of variables. This gives rise to two main types of errors :

i) Truncation errors

ii) Rounding errors

2.3.1 Truncation Errors

Truncation errors are generated when only required significant digits are considered and remaining are discarded. For example, consider the value of $\frac{1}{6}$. True value of $\frac{1}{6}$ = 0.166666666 ..... If our computer represents only 6 significant digits after decimal point, then above value will be,

Approximate value of $\frac{1}{6}$ = 0.166666 (6 digits). Here note that only 6 digits after the decimal point are considered and remaining are truncated or chopped. Hence error will be,

$$\begin{aligned} \text{Error } (\varepsilon_t) &= \text{True value} - \text{Approximate value} \\ &= 0.1666666666 \ldots - 0.166666 \\ &= 0.000000666 \ldots \end{aligned} \qquad \ldots (2.3.1)$$

Here ε_t means true error. Note that the truncation error obtained above also has infinite significant digits.

Truncation errors also occur in the series expansion. For example e^x, cos x, sin x etc terms are evaluated in computers by series expansions. These series contain infinite number of terms. But computers can evaluate only limited number of terms in series. Hence truncation of series is done. This introduces error in the value being evaluated.

2.3.2 Rounding Errors

Rounding errors take place because of rounding the last significant digit to nearest value. For example consider

$\frac{1}{6} = 0.1666666666$ ...... If our computer represents only 6 significant digits after the decimal point, then 6^{th} digit must be rounded. Then above value becomes, $\frac{1}{6} = 0.166667$. Here note that the last digit is '7' since it is rounded. Hence error will be,

$$\text{Error } (\varepsilon_t) = 0.16666666 \ldots\ldots - 0.166667$$
$$= 0.0000003333 \ldots\ldots$$

Following table illustrates the results of truncation (chopping) and rounding :

Table 2.3.1 Errors in truncation (chopping) and rounding

Method	True or actual value	Approximate value	Absolute error $\lvert\varepsilon_t\rvert$
Truncation (chopping)	5^{th} decimal > 5 $\frac{1}{6} = 0.166666666$....	0.166666	0.000000666....
Rounding	This digit is made 7, since 5^{th} decimal place is greater than 5	0.166667	0.000000333....

The above results show that rounding operation results in less absolute error compared to truncation.

2.3.3 Other Types of Errors

There are some other errors, which indirectly affect the results off numerical techniques.

(i) *Inherent errors* : It is an error which is present in the statement of the problem before its solution. Thus inherent error is out of control.

Inherent error can only be reduced by defining the problem correctly.

(ii) *Manual errors* : These errors are generated due to man made mistakes, such as inaccurate data entry in the computer, improper choice of numerical method, incorrect decisions during iterations etc. These errors can be eliminated to some extent.

(iii) *Machine error or Machine epsilon* : The chopping and rounding errors we discussed just now are called machine errors. Computers have finite wordlength. Hence they can store finite number of digits in mantissa and exponent. In rare cases exponent overflow occurs, but mantissa digits always overflow. Then these digits are rounded or chopped to fit into computer wordlength. The maximum amount of error in such operation is called *machine epsilon*. It can be computed as,

$$\text{Machine epsilon} = b^{1-t} \quad \text{... (2.3.2)}$$

Here b is the base of number system and

t is number of significant digits in mantissa.

Ex. 2.3.1 *Explain the term significant figures. Round the following numbers to two decimal places.*
(i) 24.5431 (ii) 7.4679 (iii) 102.6554
What would be the effect of truncating the above numbers ? **Dec-1998, 8 Marks**

Sol. : The term significant digits is explained in detail in sec 2.1.3. Table 2.3.2 shows the truncation and rounding of the given numbers.

Table 2.3.2 Truncation and rounding of numbers

Sr No	Given number	Rounding	Truncation
(i)	24.5431	3^{rd} digit is less than 5. Hence 2^{nd} digit will remain as it is. i.e. ⇒ 24.54	24.54
(ii)	7.4679	3^{rd} digit is greater than 5. Hence 2^{nd} digit will be rounded to higher value. i.e. ⇒ 7.47	7.46
(iii)	102.6554	3^{rd} digit is equal to 5. Hence 2^{nd} digit will be rounded to higher value. i.e. ⇒ 102.66	102.65

In the above table observe that the numbers are rounded to two decimal places. Hence we have to check digit at 3^{rd} decimal place. If 3^{rd} decimal digit is greater than or equal to 5, then 2^{nd} decimal digit is increased by one. If 3^{rd} decimal digit is less than 5, than 2^{nd} decimal digit is left as it is i.e., in general for rounding we can write,

If n^{th} *decimal* ≥ 5, *then* $(n-1)^{th}$ *decimal* + 1

If n^{th} *decimal* < 5, *then* $(n-1)^{th}$ *decimal is unchanged.*

For truncation, as shown in Table 2.1.5, two digits after decimal point are considered and remaining digits are simply discarded.

Ex. 2.3.2 *a) Explain the term significant digits. Round the following numbers to two decimal places :*
(i) 48.2141 (ii) 2.375 (iii) 81.255
b) Roundoff the following numbers to four significant figures :
i) 38.46235 (ii) 0.70029 (iii) 0.0022218 **May-1997, 4 Marks**

Sol. : a) To round the numbers to two decimal places

i) $48.2141 \approx 48.21$

ii) $2.375 \approx 2.38$

iii) $81.255 \approx 81.26$

b) To roundoff the numbers to four significant figures

i) $38.46235 \approx 38.46$ if total four significant digits are used.

≈ 38.4624 if four significant digits after decimal point.

$$\left.\begin{array}{ll} \text{ii)} & 0.70029 \approx 0.7003 \\ \text{iii)} & 0.0022218 \approx 0.2222 \end{array}\right\}$$ Note that zeros before / after decimal point are not counted as significant digits.

Exercise

1. What is machine epsilon? How machine epsilon and rounding errors are related.
2. How roundoff error varies according to chopping and rounding operations.
3. How finite word length of computer affects errors ?

University Questions

1. Explain with suitable examples different types of errors encountered in numerical computations. **[Dec - 95, May-98, May-99, May - 2001]**
2. Explain the term significant figures. Round the following numbers to two decimal places-
   i) 48.2141 ii) 2.375 iii) 81.255
   Roundoff the following numbers to four significant figures
   i) 38.46235 ii) 0.70029 iii) 0.0022218. **[May - 97]**
3. Explain the term significant figures. Round the following numbers to two decimal places.
   i) 24.5431 ii) 7.4679 iii) 102.6564
   What would be the effect of truncating the above numbers ? **[Dec - 98]**
4. Explain the different types of errors with example. **[Dec - 2002]**
5. Define Inherent, Truncation and Round-off error and give an example for each. **[May - 2003]**
6. Write short note on Different types of Errors. **[Dec - 2003]**
7. Explain different types of errors that occur in numerical calculations. **[May - 2004]**

2.4 Analysis and Estimation of Errors

2.4.1 Absolute Error and Relative Error

We know that error can be calculated as,

$$\text{Error} = \text{True or actual value} - \text{approximate value} \quad \text{... (2.4.1)}$$

The above error is also called true error (ε_t). Always the magnitude of the error is considered. Then it is called *absolute error* i.e.

$$\text{Absolute error } (\varepsilon_a) = |\text{ True or actual value} - \text{Approximate value }| \quad \text{... (2.4.2)}$$

$$\text{i.e., } \quad \varepsilon_a = |\text{ Error }| = |\varepsilon_t|$$

The absolute error depends upon magnitudes of actual and approximate values. Hence only absolute error doesnot provide complete information. Hence it is normalized with respect to actual value. Then it is called *relative error* i.e.,

$$\text{Relative error } (\varepsilon_r) = \frac{|\text{True (actual) value} - \text{Approximate value}|}{|\text{True value}|} \quad \text{... (2.4.3)}$$

$$= \frac{\text{Absolute error}}{|\text{True value}|} \quad \text{... (2.4.4)}$$

The relative error is also expressed in percentage. i.e.

$$\text{Percentage relative error } (\varepsilon_r) = \frac{\text{Absolute error}}{|\text{True value}|} \times 100 \quad \text{... (2.4.5)}$$

Ex. 2.4.1 *Calculate the absolute and relative errors in the following cases and comment on the result.*

i) True value $= 1 \times 10^{-6}$, *Approximate value* $= 0.5 \times 10^{-6}$

ii) True value $= 1 \times 10^{6}$, *Approximate value* $= 0.99 \times 10^{6}$

Sol. : i)

$$\text{Absolute error} = |\text{ True value} - \text{Approximate value }|$$

$$= 1 \times 10^{-6} - 0.5 \times 10^{-6} = 0.5 \times 10^{-6}$$

Hence relative error will be,

$$\text{Relative error} = \frac{\text{Absolute error}}{|\text{True value}|}$$

$$= \frac{0.5 \times 10^{-6}}{1 \times 10^{-6}} = 0.5$$

ii)

$$\text{Absolute error} = |\text{ True value} - \text{approximate value }|$$

$$= 1 \times 10^{6} - 0.99 \times 10^{6} = 10000$$

Hence relative error will be,

$$\text{Relative error} = \frac{\text{Absolute error}}{|\text{True value}|}$$

$$= \frac{10000}{1 \times 10^{6}} = 0.01$$

The results of this example are summarized in the following table.

Table 2.4.1 Relative and absolute errors

Example	Absolute error	Relative error	Comments
(i)	0.5×10^{-6}	0.5	Relative error is more
(ii)	10000	0.01	Relative error is less

Important comment

The above results shows that :

i) It is not possible to compare errors in two computations with the help of absolute error.

ii) Absolute of second example seems to be quite large, but relative error is about $\left(\frac{1}{50}\right)^{th}$ of the first example. Thus relative error is more useful.

Ex. 2.4.2 *Find the error E_x and relative error R_x and also determine the number of significant digits in the approximation of the following :*
i) Let $x = 2.71828182$ and $\tilde{x} = 2.7182$
ii) Let $y = 98350$ and $\tilde{y} = 98000$
iii) Let $z = 0.000068$ and $\tilde{z} = 0.00006$

Dec-2000, 8 Marks; Dec-1997, 6 Marks; May-2003, 6 Marks

Sol. : i) $x = 2.71828182$ and $\tilde{x} = 2.7182$

$$\text{Error } E_x = x - \tilde{x} = 2.71828182 - 2.7182$$
$$= 0.00008182$$
$$\text{Relative error} = \frac{\text{Absolute error}}{\text{True value}} = \frac{0.00008182}{2.71828182}$$
$$= 3.00925 \times 10^{-5}$$

The significant digits are those which the actual value and approximate value agrees i.e.,

$$\begin{aligned} x &= \boxed{2.7182}8182 \\ \tilde{x} &= \underbrace{\boxed{2.7182}}_{\text{x and } \tilde{x} \text{ agree to these 5 digits}} \end{aligned}$$

As shown above, x and $\tilde{x}$ agree to five digits. Hence 5 significant digits are present in the approximation.

ii) $y = 98350$ and $\tilde{y} = 98000$

$$\text{Error } E_y = y - \tilde{y} = 98350 - 98000 = 350$$
$$\text{Relative error} = \frac{\text{Absolute error}}{\text{True value}} = \frac{350}{98350}$$
$$= 0.0035587$$

The two values are :

$$y = \boxed{9\ 8}\ 3\ 5\ 0$$
$$\tilde{y} = \boxed{9\ 8}\ 0\ 0\ 0$$

As shown above, y and $\tilde{y}$ agree to two digits. Hence two significant digits are present in the approximation.

iii) $z = 0.000068$ and $\tilde{z} = 0.00006$

$$\text{Error} \quad E_z = z - \tilde{z} = 0.000068 - 0.00006$$
$$= 8 \times 10^{-6}$$

$$\text{Relative error} = \frac{\text{Absolute error}}{\text{True value}} = \frac{8 \times 10^{-6}}{0.000068}$$
$$= 0.117647$$

The two values are

$$z = 0\ .\ 0\ 0\ 0\ 0\ \boxed{6}\ 8$$
$$\tilde{z} = 0\ .\ 0\ 0\ 0\ 0\ \boxed{6}$$

We know that, zeros after decimal point are not considered as significant digits. Hence z and $\tilde{z}$ agree only to one digit. Therefore one significant digit is present in the approximation.

Ex. 2.4.3 *Suppose that you have a task of measuring the lengths of a bridge and a rivet and come up with 9999 and 9 cm respectively. If the true values are 10,000 and 10 cm respectively, compute.*
i) the error
ii) the percentage relative error in each case

May-1998, 6 Marks; May-1996, 6 Marks; Dec.-2004, 5 Marks

Sol. : Given data

True value of bridge = 10,000 cm
Approximate value of bridge = 9999 cm
True value of rivet = 10 cm
Approximate value of rivet = 9 cm

i) **To obtain error (ε)**

True error in bridge will be,

$$\varepsilon_t\,(\text{bridge}) = \text{True value} - \text{Approximate value}$$
$$= 10000 - 9999 = 1 \text{ cm}$$

Similarly true error in rivet will be,

$$\varepsilon_t\,(\text{rivet}) = \text{True value} - \text{approximate value}$$
$$= 10 - 9 = 1 \text{ cm}$$

ii) **To obtain relative error (ε_r)**

Percentage relative error for bridge is,

$$\varepsilon_r \text{ (bridge)} = \frac{|\text{ True value} - \text{Approximate value }|}{|\text{ True value }|} \times 100$$

$$= \frac{1}{10000} \times 100 = 0.01 \%$$

And percentage relative error for rivet is,

$$\varepsilon_r \text{ (rivet)} = \frac{|\text{ True value} - \text{Approximate value }|}{|\text{ True value }|} \times 100$$

$$= \frac{1}{10} \times 100 = 10 \%$$

Comment

The above results shows that absolute error in both the measurements is same. But relative error in the measurement of rivet is much larger than bridge.

Ex. 2.4.4 *A civil engineer has measured the height of 10 floor building as 2950m and the working height of each beam as 35 cm while the true values are 2945 cm and 30 cm respectively. Compare the absoulate and relative error.* **Dec.-2003, 3 Marks**

Sol. :

Height : Absolute error = $|2945 - 2950| = 5$

Relative error = $\frac{5}{2945} = 0.0016977$

Beam : Absolute error = $|30 - 35| = 5$

Relative error = $\frac{5}{30} = 0.16666666$

Thus even through absolute error is same, relative erros are different.

How to calculate error when true value is not known ?

In most of the situations, true or actual value is not known. When solutions are obtained by numerical techniques, then actual (true) solutions are not known. In such situations, relative error is calculated as,

$$\text{Relative error } (\varepsilon_r) = \frac{\text{approximate error}}{\text{approximate value}} \quad \text{... (2.4.6)}$$

Here approximate error cannot be obtained by equation 2.4.1. The numerical techniques work iteratively to obtain the solution. Then approximate error can be calculated as,

$$\text{Approximate error} = \text{Current approximation} - \text{Previous approximation.} \quad \text{... (2.4.7)}$$

And the relative error is then calculated as,

$$\text{Relative error } (\varepsilon_r) = \frac{\text{Current approximation} - \text{Previous approximation}}{\text{Current approximation}} \quad \text{... (2.4.8)}$$

Here note that current approximation is more close to the true (actual) value. Note that, absolute values must be taken in the above equation.

Ex. 2.4.5 *Obtain the value of expression $(1+x)^2$ by two methods using four digit floating point arithmetic for $x = 0.5129$. Calculate the relative errors in two methods.*

Sol. : Here $(1+x)^2$ can be evaluated by two methods as follows :

$$(1+x)^2 = (1+x)\,(1+x)$$

and

$$(1+x)^2 = 1+2x+x^2$$

We have to calculate error in these two methods.

(i) To calculate exact value of $(1+x)^2$

$$\text{Exact value} = (1+x)^2 = (1+0.5129)^2$$
$$= 2.2888664$$
$$= 0.22888664 \times 10^1$$

(ii) To calculate value and relative error in $(1+x)^2 = (1+x)\,(1+x)$

Putting the value of x,

$$(1+x)^2 = (1+0.5129)\;(1+0.5129)$$
$$= (1.5129)\;(1.5129)$$

The number 1.5129 should be represented by four digit floating point arithmetic. It will be 0.1512 E1. Hence the expression becomes,

$$(1+x)^2 = (0.1512\,E1)\;(0.1512\,E1)$$

$= 0.0228614\ \ E2$ (floating point multiplication)

$= 0.0228\ \ E2$ (four digits mantissa)

$= 0.2280\ \ E1$ (Normalized representation)

$= 0.2280 \times 10^1$

Now relative error will be,

$$\text{Relative error} = \frac{0.22888664 \times 10^1 - 0.2280 \times 10^1}{0.22888664 \times 10^1}$$
$$= 3.8735339 \times 10^{-3}$$

(iii) To calculate value and relative error in $(1+x)^2 = 1+2x+x^2$

Putting the value of x,

$$(1+x)^2 = 1+(2\times 0.5129)+(0.5129\times 0.5129)$$

The numbers in above equation are first converted to 4 digit floating point arithmetic. i.e.,

$$1 \Rightarrow 0.1000 \quad E\,1$$

$$2 \Rightarrow 0.2000 \quad E\,1$$

$$0.5129 \Rightarrow 0.5129 \quad E\,0$$

Hence the expression can be written as,

$$(1+x)^2 = (0.1000\,E1) + (0.2000\,E1 \times 0.5129\,E\,0) + (0.5129\,E\,0 \times 0.5129\,E\,0)$$

$$= (0.1000\,E\,1) + (0.1025\,E\,1) + (0.2630\,E\,0)$$

To perform this addition, we have to equalize exponents i.e.,

$$(1+x)^2 = (0.1000\,E\,1) + (0.1025\,E\,1) + (0.0263\,E\,1)$$

$$= 0.2288 \quad E\,1$$

$$= 0.2288 \times 10^1$$

Now relative error will be,

$$\text{Relative error} = \frac{0.22888664 \times 10^1 - 0.2288 \times 10^1}{0.22888664 \times 10^1}$$

$$= 3.783533 \times 10^{-4}$$

(iv) **Results**

Method of evaluation	Relative error
$(1+x)(1+x)$	3.8735339×10^{-3}
$1+2x+x^2$	3.783533×10^{-4}

Thus the relative error in second method is less.

Ex. 2.4.6 *If X = 0.4000 is correct to 4 significant digits, find the relative error.*

Sol. : Here 'X' is correct to 4 significant digits. This means there will be error in the 5^{th} digit. The maximum value of this error will be,

$$\text{Error } (\varepsilon) = 0.\underbrace{0000}_{\substack{\text{Four zeros for} \\ \text{4 correct significant} \\ \text{digits after decimal} \\ \text{point}}}\underbrace{5}_{\substack{\text{Maximum value} \\ \text{of the } 6^{th} \text{ digit.}}}$$

Note that error cannot be 0.00009, since it will be rounded and then error becomes 0.0001. Note that this error represents only three correct significant digits. Thus maximum absolute error is,

$$\varepsilon_a = |\varepsilon| = 0.00005$$

$$\text{Hence, Relative error } (\varepsilon_r) = \frac{\text{Absolute error}}{\text{Actual value}}$$

$$= \frac{0.00005}{0.4000}$$

$$= 0.000125$$

Thus the relative error is 0.000125, if the number is correct upto 4 significant digits.

2.4.2 Absolute Error in Algebraic Manipulations

Now let us consider the treatment of absolute error in algebraic operations such as addition, subtraction, multiplication and division.

1. **Absolute error in summation and subtraction :**

let us define the absolute error as ε_a. If two numbers are added or subtracted, then the magnitude of absolute error in the result is equal to sum of magnitudes of individual absolute errors. Let $\varepsilon_{a1}, \varepsilon_{a2}, \varepsilon_{a3}, \ldots\ldots\varepsilon_{an}$ are absolute errors in 'n' numbers. Then absolute error of addition or subtraction of these numbers is given as,

$$\varepsilon_a = \varepsilon_{a1} + \varepsilon_{a2} + \varepsilon_{a3} + \ldots\ldots\ldots + \varepsilon_{an} \qquad \ldots (2.4.9)$$

2. **Absolute error in product :**

Let there be two numbers with actual values 'a' and 'b'. Let the absolute error in 'a' is ε_{a_1} and that in 'b' is ε_{a_2}.

$$\therefore \text{Approximate value of 'a'} = a + \varepsilon_{a1}$$
$$\text{and approximate value of 'a'} = b + \varepsilon_{a2}$$

$$\therefore \quad \begin{pmatrix}\text{Error in the}\\ \text{product}\end{pmatrix} = \left[\begin{pmatrix}\text{Approximate value}\\ \text{of 'a'}\end{pmatrix} \times \begin{pmatrix}\text{Approximate value}\\ \text{of 'b'}\end{pmatrix}\right] - \left[\begin{pmatrix}\text{Actual value}\\ \text{of 'a'}\end{pmatrix} \times \begin{pmatrix}\text{Actual value}\\ \text{of 'b'}\end{pmatrix}\right]$$

$$\begin{aligned}\varepsilon_a &= [(a+\varepsilon_{a1})(b+\varepsilon_{a2})] - (ab)\\ &= ab + a\varepsilon_{a2} + b\varepsilon_{a1} + \varepsilon_{a1}\varepsilon_{a2} - ab\\ &= a\varepsilon_{a2} + b\varepsilon_{a1} + \varepsilon_{a1}\varepsilon_{a2}\end{aligned}$$

Since $\varepsilon_{a1}\, e_{a2}$ is very small product, it can be neglected. Hence absolute error in the product is given as,

$$\varepsilon_a = a\varepsilon_{a2} + b\varepsilon_{a1} \qquad \ldots (2.4.10)$$

3. **Absolute error in the division**

Here we take the division of the two numbers, i.e.

$$\therefore \quad \begin{pmatrix}\text{Error in the}\\ \text{division}\end{pmatrix} = \left[\frac{\begin{matrix}\text{Approximate value}\\ \text{of 'a'}\end{matrix}}{\begin{matrix}\text{Approximate value}\\ \text{of 'b'}\end{matrix}}\right] - \left[\frac{\begin{matrix}\text{Actual value}\\ \text{of 'a'}\end{matrix}}{\begin{matrix}\text{Actual value}\\ \text{of 'b'}\end{matrix}}\right]$$

$$\text{i.e.} \quad \varepsilon_a = \frac{a+\varepsilon_{a1}}{b+\varepsilon_{a2}} - \frac{a}{b}$$

$$= \frac{b\varepsilon_{a1} - a\varepsilon_{a2}}{b^2\left(1+\frac{\varepsilon_{a2}}{b}\right)}$$

Since $\frac{\varepsilon_{a2}}{b}$ is very small number compared to 1, it can be neglected. Hence,

$$\varepsilon_a = \frac{b\varepsilon_{a1} - a\varepsilon_{a2}}{b^2} \quad \text{... (2.4.11)}$$

Ex. 2.4.7 *Consider the expression for $x^2 - y^2$ in a computer program. Assume that the expression is computed for values of $x = a$ and $y = b$ with relative error e_a and e_b in a and b respectively. Expression can be evaluated in two forms :*
i) $f_1 = a \times a - b \times b$
ii) $f_2 = (a + b)(a - b)$
Express the error in f_1 and f_2 in terms of e_a and e_b.

May-2000, 8 Marks; Dec-1998, 8 Marks; Dec-1995, 8 Marks

Sol. : i) Given data :

$$e_a = \text{relative error in a}$$

$$e_b = \text{relative error in b}$$

ii) To obtain absolute errors in a and b

Let the absolute error in 'a' be ε_{a1}. We know that relative error in a is,

$$e_a = \frac{\text{Absolute error in a}}{\text{True value of a}} = \frac{\varepsilon_{a1}}{a}$$

$$\therefore \quad \varepsilon_{a1} = a\, e_a \quad \text{... (2.4.12)}$$

Similarly let the absolute error in 'b' be ε_{a2}. Then relative error in b can be given as,

$$e_b = \frac{\text{Absolute error in b}}{\text{True value of b}}$$

$$= \frac{\varepsilon_{a2}}{b}$$

$$\therefore \quad \varepsilon_{a2} = b\, e_b \quad \text{... (2.4.13)}$$

iii) To obtain error in $f_1 = a \times a - b \times b$

To obtain error in the product $(a \times a)$

We know that absolute error in 'a' is ε_{a1}. Hence absolute error in the product $(a \times a)$ can be obtained using equation 2.4.10 i.e.

$$\text{Absolute error in } (a \times a) = a\varepsilon_{a1} + a\,\varepsilon_{a1}$$

$$= 2a\varepsilon_{a1}$$

Putting the value of ε_{a1} from equation 2.4.12,

$$\text{Absolute error in } (a \times a) = 2a \cdot a\, e_a = 2a^2 e_a \quad \text{... (2.4.14)}$$

To obtain error in the product $(b \times b)$

We know that absolute error in 'b' is ε_{a2}. Hence absolute error in the product $(b \times b)$ can be obtained using equation 2.4.10. i.e.

$$\text{Absolute error in } (b \times b) = b\varepsilon_{a2} + b\varepsilon_{a2}$$

$$= 2b\varepsilon_{a2}$$

Putting the value of ε_{a2} from equation 2.4.13,

$$\text{Absolute error in } (b \times b) = 2b \cdot b\, e_b = 2b^2 e_b \quad \text{... (2.4.15)}$$

To obtain error in f_1

$f_1 = a \times a - b \times b$ is the subtraction of $(a \times a)$ and $(b \times b)$. Equation 2.4.9 shows that absolute error in the subtraction operation is obtained by adding absolute errors of individual terms. i.e.,

$$\text{Error in } f_1 = \text{error in } (a \times a) + \text{error in } (b \times b)$$

Putting the values from equation 2.4.14 and equation 2.4.15,

$$\text{Error in } f_1 = 2a^2 e_a + 2b^2 e_b$$

Thus error in f_1 is expressed in terms of e_a and e_b.

iv) To obtain error in f_2 = (a + b) · (a – b)

To obtain error in the addition (a + b)

We know that absolute error in 'a' is ε_{a1} and that in 'b' is ε_{a2}. Equation 2.4.9 gives the absolute error in the addition i.e.,

$$\text{Absolute error in } (a + b) = \varepsilon_{a1} + \varepsilon_{a2}$$

Putting values of ε_{a1} and ε_{a2} from equation 2.4.12 and equation 2.4.13,

$$\text{Absolute error in } (a + b) = a\, e_a + b\, e_b \quad \text{... (2.4.16)}$$

To obtain error in the subtraction (a – b)

Equation 2.4.9 gives absolute error in subtraction also i.e.,

$$\text{Absolute error in } (a - b) = \varepsilon_{a1} + \varepsilon_{a2}$$

Putting the values of ε_{a1} and ε_{a2},

$$\text{Absolute error in } (a - b) = a\, e_a + b\, e_b \quad \text{... (2.4.17)}$$

To obtain error in f_2

$f_2 = (a+b) \cdot (a-b)$ is the product of (a + b) and (a – b). Equation 2.4.10 gives an error in the product. Using this equation we can write,

Absolute error in product (a + b) · (a – b) = (a + b) · (error in a – b) + (a – b) (error in a + b)

Putting the values from equation 2.4.16 and equation 2.4.17,

$$\text{Error in } f_2 = (a+b)(a\, e_a + b\, e_b) + (a-b)(a\, e_a + b\, e_b)$$

$$= 2a(a\, e_a + b\, e_b)$$

Thus error in f_2 is expressed in terms of e_a and e_b

Ex. 2.4.8 *Current flows through a 10 ohm resistance that is accurate within 10%. The current is measured as 2 A within ± 0.1 Amps. What are the absolute and relative errors in the computed voltage ? Neglect roundoff errors.* **Dec-1996, 6 Marks**

Sol. : The current is 2 ± 0.1 A. Hence absolute error in current is 0.1. Let,

$$a = 2 \text{ and } \varepsilon_{a1} = 0.1.$$

$$\text{Similarly } R = 10 \pm 10\ \%\ \Omega \text{ i.e. } 10 \pm 1\ \Omega.$$

Hence absolute error in R is 1. Let

$$b = 10 \text{ and } \varepsilon_{a2} = 1$$

$$\text{We know that } V = I\,R$$

$$\text{i.e, } V = a \cdot b$$

This is a product. The absolute error in the product is given by equation 2.4.10. i.e.,

$$\varepsilon_a = a\varepsilon_{a2} + b\varepsilon_{a1}$$

Putting values in above equation,

$$\varepsilon_a = 2 \times 1 + 10 \times 0.1 = 3$$

Thus the absolute error in the voltage is 3 volts.

The true values of I = 2 A and R = 10 Ω. Hence true voltage is,

$$V = I\,R$$

$$= 2 \times 10 = 20 \text{ volts.}$$

Hence relative error in voltage is,

$$\text{Relative error, } \varepsilon_r = \frac{\text{Absolute error in voltage}}{\text{True value of voltage}} = \frac{3}{20} = 0.15$$

Thus the relative error in voltage is 0.15.

Ex. 2.4.9 *Find the approximate maximum error in* 5.43×27.2

Sol. : Here we have to calculate error in product. Let

$$a = 5.43 \quad \text{and} \quad b = 27.2$$

Here a = 5.43. There are two significant digits after decimal point. Hence maximum error in 'a' will be ± 0.005. In other words, maximum absolute error in 'a' will be,

$$\varepsilon_{a1} = 0.005$$

Similarly b = 27.2. There is one significant digit after decimal point. Hence maximum error in 'b' will be ± 0.05. That is, maximum absolute error in 'b' will be,

$$\varepsilon_{a2} = 0.05$$

Equation 2.4.10 gives the error in product. i.e.,

$$\varepsilon_a = a\,\varepsilon_{a2} + b\,\varepsilon_{a1}$$

Putting the values in above equation,

$$\varepsilon_a = 5.43 \times 0.05 + 27.2 \times 0.005$$

$$= 0.4075$$

Thus absolute maximum error in product of 5.43 and 27.2 will be 0.4075.

Ex. 2.4.10 *The quotient* $\frac{25.4}{12.37}$ *gives the result 2.053. To what extent this can be relied upon?*

Sol. : Maximum absolute error in 25.4 will be 0.05. Let $a = 25.4$ hence $\varepsilon_{a1} = 0.05$. Similarly maximum absolute error in 12.37 will be 0.005. Let $b = 12.37$ hence $\varepsilon_{a2} = 0.005$.

From equation 2.4.11, the absolute error in division is given as,

$$\varepsilon_a = \frac{b\,\varepsilon_{a1} - a\,\varepsilon_{a2}}{b^2}$$

Putting the values in above equation,

$$\varepsilon_a = \frac{12.37 \times 0.05 - 25.4 \times 0.005}{(12.37)^2}$$

$$= 0.003212$$

Hence the true quotient will have the value of 2.053 ± 0.003212.

Ex. 2.4.11 *The area of cross section of rod is desired upto 0.2% error. How accurately should the diameter be measured ?* **May-2003, 6 Marks**

Sol. : The area 'A' and diameter 'd' are related as,

$$A = \pi d^2 = \pi d.\ d$$

Here the diameter is multiplied i.e. d_x d. Thus the error in calculation of diameter is reflected into error in area. Error in area is 0.2% or 0.002. In other words, this error of 0.002 is obtained due to eror in product d_xd. From equation 2.4.10 we know that error in product is given as,

$$\varepsilon_a = a\,\varepsilon_{a2} + b\,\varepsilon_{a1}$$

Here $\varepsilon_a = 0.002$, i.e. error in the product or area of the rod. And a = b = d similarly $\varepsilon_{a2} = \varepsilon_{a1} = \varepsilon_a$ i.e. error in the diameter of rod. Hence above equation will be,

$$\varepsilon_a = d\,\varepsilon_d + d\,\varepsilon_d$$

$$\varepsilon_a = 2d\,\varepsilon_d$$

$$\therefore \quad 0.002 = 2d\,\varepsilon_d$$

$$\text{or} \quad \varepsilon_d = \frac{0.002}{2d} = \frac{1}{1000\,d}$$

Thus the error in the diameter should not exceed $\frac{1}{1000\,d}$.

2.4.3 Taylor's Series for Approximation of Functions

The functions such as sin x, cos x, e^x etc. can be implemented with the help of Taylor's series. The Taylor's series is used widely in numerical methods to approximate functions. The first term in Taylor series is,

$$f(x_{i+1}) \cong f(x_i) \qquad \text{... (2.4.18)}$$

This means, the value of f at x_{i+1} is same as its value at x_i. This is called zero-order approximation. This is true when x_{i+1} and x_i are close to each other. When second term is added in equation 2.4.18, it becomes,

$$f(x_{i+1}) \cong f(x_i) + f'(x_i)(x_{i+1} - x_i) \quad \text{... (2.4.19)}$$

This is called first-order approximation. It better approximates the function compared to equation 2.4.18. This is also called straight line approximation. When third term is added in above equation it becomes,

$$f(x_{i+1}) \cong f(x_i) + f'(x_i)(x_{i+1} - x_i) + \frac{f''(x_i)}{2!}(x_{i+1} - x_i)^2 \quad \text{... (2.4.20)}$$

This is called second order approximation and it is better than equation 2.4.19. Error is reduced in this approximation. Similarly additional terms can be included in the approximation. Then it becomes complete Taylor series i.e,.

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) + \frac{f''(x_i)}{2!}(x_{i+1} - x_i)^2 + \frac{f^{(3)}(x_i)}{3!}(x_{i+1} - x_i)^3 + \ldots\ldots + \frac{f^{(n)}(x_i)}{n!}(x_{i+1} - x_i)^n + R_n \quad \text{... (2.4.21)}$$

This is the complete Taylor series. It approximates the function perfectly. The remainder term includes all the terms from (n + 1) to infinity. It is given as,

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x_{i+1} - x_i)^{n+1} \quad \text{... (2.4.22)}$$

Here $x_i \le \xi \le x_{i+1}$ We can write step size $h = x_{i+1} - x_i$, then Taylor series of equation 2.4.21 can be written as,

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f^{(3)}(x_i)}{3!}h^3 + \ldots\ldots + \frac{f^{(n)}(x_i)}{n!}h^n + R_n \quad \text{... (2.4.23)}$$

and $$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}h^{n+1} \quad \text{from equation 2.4.16} \quad \text{... (2.4.24)}$$

Truncation error in series approximation

The Taylor series approximates functions. It is infinite series. It is not always possible to consider all the terms in Taylor series. Hence, few terms are calculated and remaining are *discarded.* This is also called truncation of series. Because of this truncation operation, an error is introduced in the result and it is called *truncation error* in series approximation. For example in zero-order approximation (equation 2.4.18) only first terms of Taylor series is considered. This results in large truncation error. In first-order approximation (equation 2.4.19) first two terms of

Taylor series are considered. Hence truncation error is reduced. Similarly in second order approximation (equation 2.4.20) truncation error is reduced further.

Ex. 2.4.12 *Use zero, 1^{st}, 2^{nd}, 3^{rd} and 4^{th} order Taylor series expansions to approximate the function. :*
$f(x) = -0.1x^4 - 0.15x^3 - 0.5x^2 - 0.25x + 1.2$ *from* $x_i = 0$ *with* $h = \pm 1$. *That is predict the function value at* $x_{i+1} = 1$. *State truncation error in each case.*

May-2000 8 Marks, May-1996 10 Marks

Ans. :

$$\begin{aligned} \text{Here } x_i &= 0 \\ \text{and } x_{i+1} &= x_i + h \\ &= 0 + 1, \qquad \text{since h = 1} \\ &= 1 \end{aligned}$$

Since we want $f(x_{i+1} = 1)$ the approximation will be obtained in one step only.

i) To obtain the derivatives

We have, $$f(x) = -0.1\,x^4 - 0.15\,x^3 - 0.5\,x^2 - 0.25\,x + 1.2 \qquad \text{... (2.4.25)}$$

$$\therefore \quad f'(x) = \frac{d}{dx} f(x) = -0.4\,x^3 - 0.45\,x^2 - x - 0.25 \qquad \text{... (2.4.26)}$$

$$\therefore \quad f''(x) = \frac{d}{dx} f'(x) = -1.2\,x^2 - 0.9\,x - 1 \qquad \text{... (2.4.27)}$$

$$\therefore \quad f'''(x) = \frac{d}{dx} f''(x) = -2.4\,x - 0.9 \qquad \text{... (2.4.28)}$$

$$\therefore \quad f^{(4)}(x) = \frac{d}{dx} f'''(x) = -2.4 \qquad \text{... (2.4.29)}$$

And $$f^{(5)}(x) = f^{(6)}(x) = \text{.......}\ 0$$

We have $x_i = 0$. Hence values of above derivatives at $x = x_i = 0$ will be :

$$\begin{aligned} f(x_i) &= f(x)|_{x=x_i=0} \text{ in equation 2.4.25} \\ &= -0.1(0)^4 - 0.15(0)^3 - 0.5(0)^2 - 0.25(0) + 1.2 = 1.2 \\ f'(x_i) &= f'(x)|_{x=x_i=0} \text{ in equation 2.4.26} \\ &= -0.4(0)^3 - 0.45(0)^2 - 0 - 0.25 = -0.25 \\ f''(x_i) &= f'(x)|_{x=x_i=0} \text{ in equation 2.4.27} \\ &= -1.2(0)^2 - 0.9(0) - 1 = -1 \\ f'''(x_i) &= f'''(x)|_{x=x_i=0} \text{ in equation 2.4.28} \\ &= -2.4(0) - 0.9 = -0.9 \\ f^{(4)}(x_i) &= f^{(4)}(x)|_{x=x_i=0} \text{ in equation 2.4.29} \\ &= -2.4 \end{aligned}$$

Thus we obtained the values of $f(x)$ and its derivatives at $x_i = 0$ as

$$\left.\begin{aligned} f(x_i) &= 1.2 \\ f'(x_i) &= -0.25 \\ f''(x_i) &= -1 \\ f'''(x_i) &= -0.9 \\ f^{(4)}(x_i) &= -2.4 \end{aligned}\right\} \qquad \ldots (2.4.30)$$

ii) To obtain actual value of f(x) at $x_{i+1} = 1$

Consider $f(x) = -0.1\,x^4 - 0.15\,x^3 - 0.5\,x^2 - 0.25\,x + 1.2$

At x = 1, we get,

$$f(1) = -0.1\,(1)^4 - 0.15\,(1)^3 - 0.5\,(1)^2 - 0.25\,(1) + 1.2$$

$$= 0.2 \qquad \ldots. (2.4.31)$$

Thus actual value of $f(1)$ is 0.2.

iii) To obtain zero-order approximation

The zero-order approximation is given by equation 2.4.18 i.e.,

$f(x_{i+1}) \cong f(x_i)$ i.e. first term in Taylor series

From equation 2.4.30, $f(x_i) = 1.2$ Hence approximate value of $f(x_{i+1} = 1)$ is,

$f(1) \cong 1.2$

Hence truncation error in this approximation is,

$$\varepsilon_t \text{ (zero-order)} = \text{Actual value} - \text{Approximate value}$$
$$= 0.2 - 1.2$$
$$= -1.0$$

iv) To obtain first order approximation

First order approximation is given by equation 2.4.19 i.e.,

$$f(x_{i+1}) \cong f(x_i) + f'(x_i)(x_{i+1} + x_i)$$

$$\cong f(x_i) + f'(x_i)\,h \qquad \text{i.e. first two terms in Taylor series.}$$

$x_{i+1} - x_i = h = 1,$ Putting other values from equation 2.4.30 in above equation,

$$f(x_{i+1} = 1) \cong 1.2 - 0.25(1)$$

$$\text{i.e. } f(1) \cong 0.95$$

Hence truncation error in this approximation is,

$$\varepsilon_t\,(1^{st} \text{ order}) = \text{Actual value} - \text{Approximate value}$$
$$= 0.2 - 0.95$$
$$= -0.75$$

v) To obtain second order approximation

The second order approximation is given by equation 2.4.20. i.e.,

$$\therefore \qquad f(x_{i+1}) \cong f(x_i)+f'(x_i)(x_{i+1}-x_i)+\frac{f''(x_i)}{2!}(x_{i+1}-x_i)^2$$

This can also be obtained from Taylor series of equation 2.4.23 with n=2, i.e., first three terms i.e.,

$$f(x_{i+1}) \cong f(x_i)+f'(x_i)h+\frac{f''(x_i)}{2!}h^2$$

Here $x_{i+1}-x_i$ = h = 1. And putting other values from equation 2.4.30 in above equation,

$$f(x_{i+1}=1) \cong 1.2-0.25\,(1)-\frac{1}{2!}(1)^2$$

$$\text{i.e.} \quad f(1) \cong 0.45$$

Hence truncation error in this approximation is,

$$\varepsilon_t\,(2^{nd}\text{ order}) = \text{Actual value} - \text{Approximate value}$$
$$= 0.2 - 0.45$$
$$= -0.25$$

vi) To obtain 3rd order approximation

The third order approximation can be obtained by n = 3 in equation 2.4.23, i.e. first four terms. i.e.

$$f(x_{i+1}) = f(x_i)+f'(x_i)h+\frac{f''(x_i)}{2!}h^2+\frac{f'''(x_i)}{3!}h^3$$

Putting values from equation 2.4.30,

$$f(x_{i+1}=1) = 1.2-0.25(1)-\frac{1}{2!}(1)^2-\frac{0.9}{3!}(1)^3$$

$$\text{i.e.} \quad f(1) = 0.3$$

Hence truncation error will be,

$$\varepsilon_t\,(3^{rd}\text{ order}) = \text{Actual value} - \text{Approximate value}$$
$$= 0.2 - 0.3$$
$$= -0.1$$

vii) To obtain 4th order approximation

The 4th order approximation can be obtained by n = 4 in equation 2.4.23 i.e. first five terms. i.e.

$$f(x_{i+1}) = f(x_i)+f'(x_i)h+\frac{f''(x_i)}{2!}h^2+\frac{f'''(x_i)}{3!}h^3+\frac{f^{(4)}(x_i)}{4!}h^4$$

Putting values from equation 2.4.30,

$$f(x_{i+1}=1) = 1.2-0.25(1)-\frac{1}{2!}(1)^2-\frac{0.9}{3!}(1)^3-\frac{2.4}{4!}(1)^4$$

i.e. $f(1) = 0.2$

Hence truncation error will be,

$$\varepsilon_t \,(4^{th} \text{order}) = \text{Actual value} - \text{Approximate value}$$
$$= 0.2 - 0.2$$
$$= -0$$

Thus there is no truncation error in 4^{th} order approximation.

viii) Results and comments

Table 2.4.2 presents all the results of this example collectively.

Table 2.4.2 Results of various approximations of $f(x)$
Actual value of $f(x)$ at x = 1 is, f(1) = 0.2

Approximation	Approximate value	Truncation error
Zero order (n = 0)	1.2	– 1.0
First-order (n = 1)	0.95	– 0.75
Second-order (n = 2)	0.45	– 0.25
Third-order (n = 3)	0.3	– 0.1
Fourth-order (n = 4)	0.2	0

In the above table observe that truncation error goes on reducing as higher-order approximations are used. Here note that $f(x)$ is fourth order polynomial. Hence its 5^{th} and higher order derivatives are zero. Hence in Taylor series (equation 2.4.23), the derivatives upto 4^{th} order results in exact value. In such case truncation error is zero.

Ex. 2.4.13 *Obtain Maclaurin's expansion for e^x using Taylor series.*

Sol. : Taylor series is given by equation 2.4.23 as,

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f^{(3)}(x_i)}{3!}h^3 + \ldots + \frac{f^{(n)}(x_i)}{n!}h^n + R_n \quad \ldots (2.4.32)$$

Let $x = x_{i+1}$ and evaluate $f(x)$ at $x_i = 0$. Then,

$$f(x_i) = e^{x_i}$$

Since $x_i = 0$, $f(x_i) = e^0 = 1$

$f'(x_i) = e^{x_i} = 1$ since $x_i = 0$

Similarly, $f''(x_i) = f^{(3)}(x_i) = \ldots\ldots = f^{(n)}(x_i) = 1$

Here note that all the above derivatives of e^x are evaluated at $x_i = 0$ we know that,

$$\begin{aligned} h &= x_{i+1} - x_i \\ &= x - 0, \qquad \text{since } x_{i+1} = x \\ &= x \end{aligned}$$

Hence equation 2.4.32 becomes,

$$f(x) = 1 + 1(x) + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \ldots + \frac{1}{n!}x^n + R_n$$

Since $f(x) = e^x$, the above expansion is,

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^n}{n!} + R_n \qquad \ldots (2.4.33)$$

Here R_n is the remainder term. It is given by equation 2.4.24 i.e.,

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!} h^{n+1}$$

$f^{(n+1)}(\xi)$ is the $(n+1)^{th}$ derivative of e^x at x = ξ. Hence $f^{(n+1)}(\xi) = e^{\xi}$.

Hence above equation becomes,

$$R_n = \frac{e^{\xi}}{(n+1)!} h^{n+1}$$

Since h=x, as we have seen earlier, we can write

$$R_n = \frac{x^{n+1}}{(n+1)!} e^{\xi} \qquad \ldots (2.4.34)$$

This is the remainder term in the expansion of e^x. This expansion is also called Maclaurin's expansion. Similarly other functions like cos x, sin x, log x etc. can be approximated using Taylor series.

Ex. 2.4.14 *Calculate e^x upto first five terms and estimate the truncation error at $x = 1$.*

Sol. : The exponential series, e^x is given by equation 2.4.33. Taking first five terms of the series we get,

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

Putting $x = 1$ in above equation we get,

$$\begin{aligned} e^1 &= 1 + \frac{1}{1!} + \frac{1^2}{2!} + \frac{1^3}{3!} + \frac{1^4}{4!} \\ &= 1 + 1 + \frac{1}{2} + \frac{1}{6} + \frac{1}{24} \\ &= 2.7083333 \qquad \text{(approximate value)} \qquad \ldots (2.4.35) \end{aligned}$$

Actual value of e^x at $x = 1$ will be,

$$\text{Actual value} = 2.7182818 \quad \ldots (2.4.36)$$

Hence truncation error in approximation of exponential series will be,

$$\begin{aligned}\text{Truncation error} &= \text{Actual value} - \text{Approximate value}\\ &= 2.7182818 - 2.708333\\ &= 0.0099485 \quad \ldots (2.4.37)\end{aligned}$$

2.4.4 C Program for Estimation of Truncation Error

Now let us develop an algorithm and computer program for calculation of truncation error in series approximation.

Algorithm

In this algorithm, it is assumed that series function is given.

Step 1 : Accept the number of terms to be computed in series approximation.

Step 2 : Accept the value of variable at which the series is to be evaluated.

Step 3 : Calculate the function value using given series.

Step 4 : Calculate actual value of the function directly.

Step 5 : Find out the absolute error between actual value and computed value of the function.

Step 6 : Truncation error is the absolute error obtained in step(5). Display this error.

Step 7 : If Truncation error is above required value, then increase number of terms to be computed in series and repeat above steps.

Program :

The C program for calculation of truncation error in series approximation is given below. The program first accepts the number of terms to be calculated in series. Then it accepts the value of variable x. The `for` loop then calculates terms of series for e^x. i.e. from equation 2.4.32.

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots \text{ upto given number of terms} \quad \ldots (2.4.38)$$

The program then displays computed value of series according to equation 2.4.38 above. The next `printf` statement prints actual value of e^x. The last `printf` statement displays error. The `fact` function is written to calculate factorial of a number

```
/* Download this program from   www.vtubooks.com          */
/* File name : trn_err.cpp                                */

/*--- CALCULATION OF TRUNCATION ERROR OF SERIES APPROXIMATION ---*/

/*      EXPONENTIAL SERIES exp(x) IS USED HERE

 INPUTS :  1) The total number of terms to be computed
```

```
                  in series starting from first term.
              2)  The value of 'x' in exp(x).

   OUTPUTS  : 1)  Computed value of series approximation of exp(x).
              2)  Actual value of exp(x) function.
              3)  Absolute error between series approximation and
                  actual value of exp(x) function.                    */

/*------------------------------ PROGRAM ---------------------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>

void main()
{
     double fact ( int n);         /* DECLARATION OF  FACTORIAL FUNCTION   */

      double x,sum,z,suml;
      int n,i;
      clrscr();
      printf("\n  CALCULATION OF TRUNCATION ERROR OF SERIES APPROXIMATION");
      printf("\n\nGive number of terms "
            "in series to be computed = ");
      scanf("%d",&n);         /* NUMBER OF TERMS TO BE COMPUTED IN SERIES    */
      n = n - 1;      /* TERMS ARE COMPUTED FROM ZERO HENCE 'n-1'            */

      printf("\nGive value of x = ");
      scanf("%lf",&x);     /* VALUE OF  'x'   */

      sum = 0;   suml = 0;
      for(i = 0; i<=n; i++)
        {
   z = fact(i);
   sum = pow(x,(double)i)/z;
   suml = suml + sum;

        }
    printf("\nThe computed value of exp(x) "
          "function is = %1.15lf",suml);       /* COMPUTED VALUE   */
    printf("\n\nThe actual value of exp(x) "
          "function is = %1.15lf",exp(x));   /* ACTUAL VALUE  */
    printf("\n\nTruncation error in series "
          "approximation is = %1.15lf", exp(x)-suml);    /* ERROR */
}
/*------------------ FUNCTION PROCEDURE TO CALCULATE FACTORIAL --------------*/

double fact( int n)
{
   double facto;
   facto = 1;
   if(n == 0) return( facto = 1);
   do
   {
       facto = facto * n;
   }
   while(n- - > 1);
   return(facto);
}
/*--------------------------------- END OF PROGRAM -----------------------*/
```

Compile and run the above program. Enter the value of terms in series to be 5. Then enter the value of x = 1. Then the program displays following.

`Computed value=2.7083333` (Which is same as that we obtained in equation 2.4.35)

Actual value=2.7182818 (Which is same as that we obtained in equation 2.4.36)

Truncation error=0.0099485 (Which is same as that we obtained in equation 2.4.37)

The results displayed by computer are shown below :

```
--------------------------- Result 1 ---------------------------

  CALCULATION OF TRUNCATION ERROR OF SERIES APPROXIMATION

Give number of terms in series to be computed = 5

Give value of x = 1

The computed value of exp(x) function is = 2.708333333333333

The actual value of exp(x) function is = 2.718281828459045

Truncation error in series approximation is = 0.009948495125712
```

If we increase the number of terms in series computation, then truncation error reduces. Results below show the truncation error for 10 terms of series. Observe that truncation error now is reduced to 0.0000003028.

The results are given below.

```
--------------------------- Result 2 ---------------------------

  CALCULATION OF TRUNCATION ERROR OF SERIES APPROXIMATION

Give number of terms in series to be computed = 10

Give value of x = 1

The computed value of exp(x) function is = 2.718281525573192

The actual value of exp(x) function is = 2.718281828459045

Truncation error in series approximation is = 0.000000302885853
```

Ex. 2.4.15 *Find the value of* e^x *using the expansion,*

$$e^x = 1 + x^2 + \frac{x^2}{2\,!} + \frac{x^3}{3\,!} + \ldots\ldots$$

for $x = 0.5$ *with absolute error less than 0.005.*

Sol. : Here upper bound on the absolute error is given. We have to determine the number of terms in the series of e^x so that error will be less than 0.005. The series for e^x is given as,

$$e^x = 1 + x + \frac{x^2}{2\,!} + \frac{x^3}{3\,!} + \ldots.. + \frac{x^n}{n\,!}$$

From above equation it is clear that n^{th} term in series of e^x is,

$$n^{th} \text{ term} = \frac{x^n}{n\,!}$$

If we use 'n' terms of series to find e^x, then $(n+1)$ and higher terms are neglected. Hence magnitude of the truncated error is given by $(n+1)^{th}$ term. i.e.,

$$(n+1)^{th} \text{ term in } e^x = \frac{x^{n+1}}{(n+1)\,!}$$

Since the absolute error to be less than 0.005,

$$\frac{x^{n+1}}{(n+1)\,!} < 0.005$$

Take logarithm of both the side to simplify and solve above equation. i.e.,

$$log\left(x^{n+1}\right) - log\,(n+1)\,! < log\,(0.005)$$

$$\therefore\ (n+1)\,log\,x - log\,(n+1)\,! < -2.30103$$

The above equation can also be written as,

$$log\,(n+1)\,! - (n+1)\,log\,x > 2.30103 \qquad \text{... (2.4.39)}$$

Now we have $x = 0.5$. Let us see, which value of 'n' satisfies this equation. Let $n = 2$ in above equation. We get,

$$log\,(2+1)\,! - (2+1)\,log\,0.5 = 1.6812412$$

Thus $n = 2$ do not satisfy the condition of equation 2.4.39. Now let $n = 3$ in equation 2.4.39 we get,

$$log\,(3+1)\,! - (3+1)\,log\,0.5 = 2.5843312$$

Thus the condition of equation 2.4.39 is satisfied. Hence, with $n = 3$ in the given series for e^x we get,

$$e^x = 1 + x + \frac{x^2}{2\,!} + \frac{x^3}{3\,!} \qquad (\text{for } n = 3)$$

With $x = 0.5$ in above equation we get,

$$e^x = 1 + 0.5 + \frac{(0.5)^2}{2\,!} + \frac{(0.5)^3}{3\,!}$$

$$= 1 + 0.5 + 0.125 + 0.0208333$$

$$= 1.6458333 \qquad (\text{Approximate value})$$

Actual value of e^x for $x = 0.5$ is,

$$e^x = e^{0.5} = 1.6487213 \qquad (\text{Actual value})$$

Hence absolute error is,

$$\text{Absolute error} = |\text{Actual value} - \text{Approximate value}|$$

$$= |1.6487213 - 1.6458333|$$

$$= 0.0028879$$

Thus the absolute error is less than 0.005. Thus first four terms (i.e. $n=3$) yields,

$$e^x = 1.6458333 \qquad \text{(Approximate value)}$$

Ex. 2.4.16 *Find the value of sin x using series expansion*

$$\sin x = x - \frac{x^3}{3\,!} + \frac{x^5}{5\,!} \ldots\ldots$$

for $x = 0.2$, with absolute error less than 0.005.

Sol. : The expansion of $\sin x$ is given as,

$$\sin x = x - \frac{x^3}{3\,!} + \frac{x^5}{5\,!} - \ldots\ldots + \frac{(-1)^{n+1}\,x^{2n-1}}{(2n-1)\,!}$$

Here $\qquad n^{th}$ term $= \dfrac{(-1)^{n+1}\,x^{2n-1}}{(2n-1)\,!}$

In first 'n' terms of the series are used. then magnitude of truncation error is given by $(n+1)^{th}$ term. i.e.,

$$(n+1)^{th} \text{ term} = \frac{(-1)^{n+2}\,x^{2n+1}}{(2n+1)\,!}$$

$$\text{Hence absolute error} = \frac{x^{2n+1}}{(2n+1)\,!}$$

For this error to be less than 0.005,

$$\frac{x^{2n+1}}{(2n+1)\,!} < 0.005$$

$\therefore \quad \log x^{2n+1} - \log\,(2n+1)\,! < \log\,(0.005)$

$\therefore \quad (2n+1)\log x - \log\,(2n+1)\,! < -2.30103$

$\therefore \quad \log\,(2n+1)\,! - (2n+1)\log x > 2.30103 \qquad \ldots (2.4.40)$

We have $x = 0.2$. Now we have to see at what value of 'n' above equation is satisfied. Let $n=1$ in above equation. i.e.,

$$\log\,(2+1)\,! - (2+1)\log 0.2 = 2.8750613$$

Thus the condition of equation 2.4.40 is satisfied for $n=1$. Thus for absolute error to be less than 0.005 we take $n=1$ in expansion of $\sin x$. i.e.,

$$\sin x = x \qquad \text{with } n=1 \text{ only first term.}$$

At $\qquad x = 0.2, \qquad \sin(0.2) = 0.2 \qquad$ (Approximate value)

And $\qquad \sin(0.2) = 0.1986693 \qquad$ (Actual value)

Thus absolute error is,

$$\text{Absolute error} = |0.1986693 - 0.2|$$
$$= 0.0013306$$

Thus the absolute error is less than 0.005.

Hence $\sin x = 0.2$ for $x = 0.2$ with absolute error less than 0.005.

Ex. 2.4.17 *Find the number of terms of the exponential series such that their sum gives the value of e^x correct to five decimal places for all values of x in the range $0 \le x \le 1$.*

Dec-1999 6 Marks, May-1997 6 Marks

Sol. : The Maclaurin expansion for e^x is obtained in equation 2.4.33 and equation 2.4.34 i.e.

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots\ldots + \frac{x^n}{n!} + \frac{x^{n+1}}{(n+1)!} e^{\xi} \text{ and } 0 < \xi < x \qquad \ldots (2.4.41)$$

Here the remainder term $\frac{x^{n+1}}{(n+1)!} e^{\xi}$ is an error term. At $\xi = x$ this term will have maximum value. Hence maximum absolute error will be given by the remainder term. i.e.,

$$\text{Maximum absolute error} = \frac{x^{n+1}}{(n+1)!} e^x \quad \text{since } \xi = x$$

Hence maximum relative error will be,

$$\text{Maximum relative error} = \frac{\text{Absolute error}}{\text{Actual value}}$$

Here actual value is e^x. Putting values in above equation,

$$\text{Maximum relative error} = \frac{\frac{x^{n+1}}{(n+1)!} e^x}{e^x} = \frac{x^{n+1}}{(n+1)!} \qquad \ldots (2.4.42)$$

Since $0 \le x \le 1$, the maximum relative error will take place at x = 1. Then above equation becomes,

$$\text{Maximum relative error} = \frac{1}{(n+1)!} \qquad \ldots (2.4.43)$$

When any value is correct to five decimal places, than maximum error in that value is,

$$\text{Maximum error} = 0.\underbrace{0\,0\,0\,0\,0}_{\text{Five zeros for 5 decimal places}}5$$

This is the maximum value at this place

Here note that maximum error contains 5 zeros after decimal point so that value is correct to 5 decimal places. The sixth digit is 5. We cannot write maximum error as 0.000009 since,

$$0.000009 \cong 0.00001 \text{ after rounding last digit.}$$

Note that after rounding the error becomes 0.00001, which is correct only to four decimal places (four zeros after decimal point).

The error of equation 2.4.43 must be less than 0.000005 (five decimal places)

Hence we can write

$$\frac{1}{(n+1)!} \leq 0.000005$$

$$\text{i.e.} \quad \frac{1}{(n+1)!} \leq \frac{1}{2} \times 10^{-5}$$

Now let us evaluate the factorials of 7, 8 and 9 as follows :

$$\frac{1}{7!} = 1.984127 \times 10^{-4} > \frac{1}{2} \times 10^{-5}$$

$$\frac{1}{8!} = 2.4801587 \times 10^{-5} > \frac{1}{2} \times 10^{-5}$$

$$\frac{1}{9!} = 2.7557319 \times 10^{-6} < \frac{1}{2} \times 10^{-5}$$

Hence we should take n + 1 = 9, i.e. n = 8. In the expansion of Taylor series we will take n = 0, 1, 2, ..., 8. That is, we have to take first 9 terms of e^x, since n starts from zero.

2.4.5 A General Error Formula (Error in Function or More Than One Variable)

Here we will derive a general error formula for functions which are dependent on several other variables. Let us consider,

$$u = f(x_1, x_2, x_3, \ldots\ldots x_n)$$

Here u is the function of variables $x_1, x_2, x_3, \ldots\ldots x_n$. Then the error εu in u is given as,

$$u + \varepsilon u = f(x_1 + \varepsilon x_1, x_2 + \varepsilon x_2, \ldots\ldots x_n + \varepsilon x_n) \quad \ldots (2.4.44)$$

Here εx_1 is error in x_1, εx_2 is error in x_2 and so on. The right handside of above equation can be expanded using Taylor series for several variables. i.e.,

$$u + \varepsilon u = u + \left\{ \frac{\partial u}{\partial x_1} \varepsilon x_1 + \frac{\partial u}{\partial x_2} \varepsilon x_2 + \ldots\ldots + \frac{\partial u}{\partial x_n} \varepsilon x_n \right\}$$

$$+ \frac{1}{2} \left\{ \frac{\partial^2 u}{\partial x_1^2} (\varepsilon x_1)^2 + \ldots\ldots + \frac{\partial^2 u}{\partial x_n^2} (\varepsilon x_n)^2 \right.$$

$$\left. + 2 \frac{\partial^2 u}{\partial x_1 \partial x_2} \varepsilon x_1 \, \varepsilon x_2 + \ldots + 2 \frac{\partial^2 u}{\partial x_{n-1} \partial x_n} \varepsilon x_{n-1} \varepsilon x_n \right\} + \ldots$$

Note that the errors in x_i are very small. Hence squares and high powers of εx_i will be very small and can be neglected. Hence all the terms from second { } bracket and onward can be neglected. Then above equation becomes,

$$\varepsilon u = \frac{\partial u}{\partial x_1} \varepsilon x_1 + \frac{\partial u}{\partial x_2} \varepsilon x_2 + \ldots\ldots + \frac{\partial u}{\partial x_n} \varepsilon x_n \quad \ldots (2.4.45)$$

This is the required formula for error in u, when it is function of several variables. Then the relative error (ε_r) is given as,

$$\varepsilon_r = \frac{\varepsilon u}{u}$$

i.e. $$\varepsilon_r = \frac{\partial u}{\partial x_1}\cdot\frac{\varepsilon x_1}{u}+\frac{\partial u}{\partial x_2}\cdot\frac{\varepsilon x_2}{u}+\ldots\ldots+\frac{\partial u}{\partial x_n}\cdot\frac{\varepsilon x_n}{u} \qquad \ldots (2.4.46)$$

This is an expression for relative error in u, when it is function of several variables.

Ex. 2.4.18 *Find the maximum relative error in using the following formula :*

$$u = \frac{5xy^2}{z^3} \qquad \text{at } x = y = z = 1$$

When error in each one of therm is 0.001. Derive the formula used .

May-1999 8 Marks, May-1997 8 Marks, Dec-1997 5 Marks

Sol. : Here observe that u is the function of three variables x, y and z. Hence we must use the error formula of several variables. Let us determine the partial derivatives with respect to x, y and z. i.e.

$$\left.\begin{aligned} \frac{\partial u}{\partial x} &= \frac{5y^2}{z^3} \\ \frac{\partial u}{\partial y} &= \frac{10xy}{z^3} \\ \text{and} \quad \frac{\partial u}{\partial z} &= -\frac{15xy^2}{z^4} \end{aligned}\right\} \qquad \ldots (2.4.47)$$

Hence error in 'u' is given by equation 2.4.45 as,

$$\varepsilon u = \frac{\partial u}{\partial x_1}\varepsilon x_1+\frac{\partial u}{\partial x_2}\varepsilon x_2+\ldots\ldots+\frac{\partial u}{\partial x_n}\varepsilon x_n$$ for three variables x, y and z above equation becomes,

$$\varepsilon u = \frac{\partial u}{\partial x}\varepsilon x+\frac{\partial u}{\partial y}\varepsilon y+\frac{\partial u}{\partial z}\varepsilon z$$

We want maximum absolute error. Hence. we should take absolute values of all the terms on right hand side. i.e.,

$$\varepsilon u = \left|\frac{\partial u}{\partial x}\varepsilon x\right|+\left|\frac{\partial u}{\partial y}\varepsilon y\right|+\left|\frac{\partial u}{\partial z}\varepsilon z\right|$$

Putting expression from equation 2.4.47 in above equation,

$$\varepsilon u = \left|\frac{5y^2}{z^3}\varepsilon x\right|+\left|\frac{10xy}{z^3}\varepsilon y\right|+\left|-\frac{15xy^2}{z^4}\varepsilon z\right|$$

It is given that x=y=z=1. And errors in x, y and z are $\varepsilon x = \varepsilon y = \varepsilon z = 0.001$. Putting these values in above equation.

$$\begin{aligned} \varepsilon u &= |5(0.001)| + |10(0.001)| + |-15(0.001)| \\ &= 0.005 + 0.01 + 0.015 \\ &= 0.03 \end{aligned}$$

Now let us calculate value of 'u' i.e.

$$u = \frac{5xy^2}{z^3}$$

At $x = y = z = 1$, above equation becomes,

$$u = 5$$

The relative error in 'u' will be

$$\text{Relative error} \quad \varepsilon_r = \frac{\varepsilon u}{u}$$

$$= \frac{0.03}{5} = 0.006$$

Thus maximum relative error in 'u' will be 0.006.

Exercise

1. Explain what is error? Explain inherent, relative and absolute errors.
2. Explain how absolute error changes according to addition, subtraction, multiplication and division.
3. What is the importance of series approximation in numerical analysis ?
4. What is truncation error in series approximation.

Unsolved Problems

2. It is given that

$a = 10.00 \pm 0.05$ $\qquad$ $b = 0.0356 \pm 0.0002$

$c = 15300 \pm 100$ $\qquad$ $d = 62000 \pm 500$

Find the maximum value of error in following operations

a) a + b + c + d **[Ans. : 600.05]**

b) a + 5c − d **[Ans. 1000.05]**

c) c^3 **[Ans. : 5.766×10^{12}]**

3. Find out the truncation error in the exponential series given as,

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots\ldots + \frac{x^n}{n!}$$

For computation of first six terms in expansion and x = 2.5 **[Ans. : 0.51192104]**

University Questions

1. Consider the expression for $x^2 - y^2$ in a computer program. Assume that the expression is computed for values of x = a and y = b with relative errors e_a and e_b in a and b respectively. Evaluation of the expression can be made of the two forms :

i) f_1 : = a * a – b * b

ii) f_2 : = (a+b) (a–b)

Express the errors of f_1 and f_2 interms of e_a and e_b.

[Dec - 95, Dec-98, Dec - 99, May - 2000]

2. Suppose that you have the task of measuring the lengths of a bridge and a rivet and come up with 9999 and 9 cm respectively. If the true values are 10,000 and 10 cm respectively, compute-

   i) the error and

   ii) the percentage relative error in each case. **[May - 96, May-98]**

3. Use zero, 1st, 2nd, 3rd and 4th order Taylor's series expansions to approximate the function -

   $f(x) = -0.1x^4 - 0.15x^3 - 0.5x^2 - 0.25x + 12$

   from $x_i = 0$ with h = ± 1. i.e., predict the function value at $x_{i+1} = 1$. State the trancation error in each case. **[May - 96, May-2000]**

4. Current flows through a 10 ohm resistance that is accurate within 10 %. The current is measured as 2.0 A within ± 0.1 Amps. What are absolute and relative errors in the computed voltage ? Neglect roundoff errors. **[Dec - 96]**

5. What is meant by absolute and relative errors ?

   Find the number of terms of the exponential series such that their sum gives the value of e^x correct to five decimal places for all values of x in the range $0 \le x \le 1$.

   [May - 97, Dec-99]

6. Find the maximum relative error in using the following formula :

   $$u = \frac{5xy^2}{z^3}$$

   at x = y = z = 1 when the error in each one of them is 0.001.

   Derive the formula used **[May - 97, Dec - 97, May - 99]**

7. Find the error E_x and the relative error R_x and also determine the number of significant digits in the approximation for the following :

   i) Let x = 2.71828182 and $\bar{x}$ = 2.7182

   ii) Let y = 98350 and $\bar{y}$ = 98000

   iii) Let z = 0.000068 and $\bar{z}$ = 0.00006. **[Dec - 97, Dec - 2000]**

8. Find the absolute error, relative error and also determine the number of significant digits in the approximation of the following numbers :

   i) Let x = 2.71828182 and $\bar{x}$ = 2.7182

   ii) Let y = 98350 and d$\bar{y}$ = 98000. **[May - 2003]**

9. Find the absolute error E_A in a product of two numbers a and b and in the quetient a/b. **[Dec-2001]**

10. The area of cross section of rod is desired upto 0.2% error. How accurately should the diameter be measured ? **[May - 2003]**

11. A civil engineer has measured the height of 10 floor building as 2950 m and the working height of each beam as 35 cm while the true values are 2945 cm and 30 cm respectively. Compare the absoulate and relative error. **[Dec - 2003]**
12. You have a task of measuring the lengths of bridge and rivet and they measured 9999 cm and 9cm respectively. If the true values are 10000 cm and 10 cm respectively compute (a) the true error (b) % relative error for each case. **[Dec - 2004]**

2.5 Error Propagation

When an error is introduced in the variable, it propagates in the other variables because of computations. The amount of error propagated depends upon the type of mathematical or numerical operation performed. Consider the function,

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

Let's calculate $f(x)$ for x = 0.9. Then exact value of $f(x)$ for x = 0.9 is,

$$f(x) = \frac{1}{1-(0.9)^2} = 5.2631579$$

If this value is normalized to six digits normalized floating point number, then mantissa will contain 6 digits and exponent will be adjusted accordingly. i.e.,

$$f(x) = 5.2631579 = 0.526315 \text{ E1} \qquad \text{... (2.5.1)}$$

Let's assume that approximate value of x is,

$$\tilde{x} = 0.900005 \quad [\tilde{x} \text{ is approximate value of x}].$$

This value is normalized and contains `6' digits in mantissa. Error in the value is 0.000005, which is very small. With this value, approximate value of $f(x)$ is,

$$f(\tilde{x}) = \frac{1}{1-(0.900005)^2} = 5.2634072$$

$f(\tilde{x})$ normalized to `6' digits = 0.526340 E1 ... (2.5.2)

$$\therefore \quad \text{Error in } f(x) = |f(x) - f(\tilde{x})| = |\,0.526315\,E1 - 0.526340\,E1\,|$$

From equation 2.5.1 and equation 2.5.2

$$= 0.000025 \qquad \text{... (2.5.3)}$$

Thus error in 'x' in sixth digit has created an error in $f(x)$ in fifth digit. This is also called magnification of the error. Since error is zero upto 4^{th} digit (in mantissa of error in equation 2.5.3, the approximate value of $f(x)$ is said to be correct upto `4' digits only. Where as $f(x)$ is given for `6' digits. Thus because of error propagation we lost two significant digits.

If such an error magnification occurs in subsequent computations, then the error becomes so large that the computed result is totally redundant. Under such condition, the numerical method or computation procedure is said to be numerically unstable. To avoid this instability, the numerical process is rearranged, the calculation steps are interchanged or some other method is adopted. For the method to be numerically stable, error from previous to next computations should die out or decay to negligible value. Numerical instability may arise because of the ill conditioness of the problem itself. Then the problem should only be redefined to avoid instability.

2.5.1 Stability and Condition

Consider the function $f(x)$, which is function of x. Let us assume $\tilde{x}$ is an approximate value of x. Let $f(\tilde{x})$ be approximate value of $f(x)$. Then error in $f(x)$ will be given as,

$$\varepsilon f(\tilde{x}) = |f(x) - f(\tilde{x})|$$

Here actual value x is unknown. Hence $f(x)$ is unknown. Hence above error equation is difficult to solve directly. Taylor series can be used to compute $f(x)$ near $f(\tilde{x})$. i.e.,

$$f(x) = f(\tilde{x}) + f'(\tilde{x})(x-\tilde{x}) + \frac{f''(\tilde{x})}{2}(x-\tilde{x})^2 + \ldots.$$

Neglect the second and higher order derivatives in the above equation. Then we get,

$$f(x) = f(\tilde{x}) + f'(\tilde{x})(x-\tilde{x}) \qquad \ldots (2.5.4)$$

Relative error in $f(x)$ can be given as (using equation 2.4.6),

$$\varepsilon_r [f(x)] = \frac{f(x) - f(\tilde{x})}{f(\tilde{x})}$$

From equation 2.5.4 we can write above equation as,

$$\varepsilon_r [f(x)] = \frac{f'(\tilde{x})(x-\tilde{x})}{f(\tilde{x})} \qquad \ldots (2.5.5)$$

Relative error in x is given as,

$$\varepsilon_r (x) = \frac{x-\tilde{x}}{\tilde{x}} \qquad \ldots (2.5.6)$$

The *condition* or *condition number* of the function is defined as the ratio of relative error in $f(x)$ to relative error in x. i.e.,

$$\text{Condition number} = \frac{\varepsilon_r [f(x)]}{\varepsilon_r (x)}$$

Putting the values from equation 2.9.5 and equation 2.9.6 in above equation we get,

$$\text{Condition number} = \frac{\tilde{x} f'(\tilde{x})}{f(\tilde{x})} \qquad \ldots (2.5.7)$$

The *condition* number represents the reflection of relative error in x to that in $f(x)$. In other wrods, the *condition* represents sensitivity of $f(x)$ to uncertainty in x.

There are three values of condition number which classify the functions as follows :

(i) Condition number = 1

This means the relative error in x is same as relative error in $f(x)$. There is no change.

(ii) Condition number > 1

This means the relative error is amplified. Such functions are called *ill-conditioned* or *numerically unstable.*

(iii) Condition number < 1

This means that the relative error is attenuated. Such functions are numerically stable.

Ex. 2.5.1 *Given a value of $\tilde{x} = 2.5$ with an error of $\varepsilon\, \tilde{x} = 0.01$, estimate the resulting error in the function $f(x) = x^3$.*

Sol. : Here $f(x) = x^3$

Taking derivative of $f(x)$ we get,

$$f'(x) = 3x^2$$

The approximate value of $f'(x)$ for $\tilde{x} = 2.5$ will be,

$$f'(\tilde{x}) = 3(2.5)^2 = 18.75$$

Consider equation 2.5.4 which gives $f(x)$ as,

$$f(x) = f(\tilde{x}) + f'(\tilde{x})(x - \tilde{x})$$

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

Here $f(x) - f(\tilde{x}) = \varepsilon f(\tilde{x})$. And $x - \tilde{x} = \varepsilon\, \tilde{x}$. Hence above equation can be written as,

$$\varepsilon f(\tilde{x}) = f'(\tilde{x}) \cdot \varepsilon\, \tilde{x}$$

Putting the values of $f'(\tilde{x})$ and $\varepsilon\, \tilde{x}$,

$$\varepsilon f(\tilde{x}) = 18.75 \times 0.01$$
$$= 0.1875$$

Thus the error in $f(x)$ will be 0.1875.

Exercise

1. Explain how error propagation can lead to numerical instability.

University Questions

1. What do you understand by propagation of errors ?

   [Dec - 97, May-98, Dec-99, May-2000, May-2001]

2. Explain the term significant figures. What is the effect of truncation of numbers ? Explain using at least 3 examples. What is propagation of errors and its effect ?

   [Dec - 2000]

3. Explain propagation of errors. **[Dec - 2002, May - 2004]**

❑❑❑

3
Roots of Equations

In the science and engineering applications we always come across the equations of the form,

$$f(x) = 0 \qquad \text{... (3.1)}$$

When this equation is quadratic, then standard methods are available to find roots of such equation. That is,

$$f(x) = ax^2 + bx + c = 0$$

Then roots of this equation can be obtained using direct methods. Consider the following equation,

$$f(x) = ax^3 + bx^2 + cx + d = 0$$

Such equation cannot be easily solved using standard methods or direct formulae are not available find roots of such equation. Sin, cos etc. functions are called transcendental functions. Equations involving such functions are called transcendental equations.

$$f(x) = x^2 - 2sinx + 3 = 0$$

is a transcendental equation. There is no direct formula available to obtain roots of transcendental equations. In above discussion we considered polynomial equations and transcendental equations. The roots of those equations can be obtained with the help of recursive methods or numerical methods.

Types of methods to obtain root

There are two types of methods : bracketing methods and open methods.

Table 3.1 : Bracketing methods and open methods

Sr. no	Bracketing methods	Open methods
1	There are two initial values which bracket the root.	There are one or more initial values which doesnot bracket the root.
2	These methods are always convergent.	These methods can be divergent sometimes.
3	More number of iterations are reaquired	Less number of iterations are required.
4	Ex. Bisection method and regula falsi method	Ex. Newton-raphson method and secant method.

In this chapter we will study some numerical methods and their `C' programmes to find single or multiple roots of algebraic and transcendental equations.

3.1 Bisection Method

3.1.1 Method

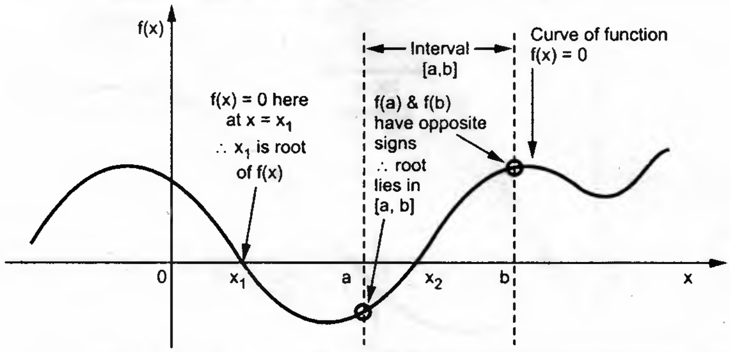


Fig. 3.1.1 Interpretation of intermediate value theorem

Consider the function

$$f(x) = 0 \qquad \text{... (3.1.1)}$$

The values of x at which $f(x)$ becomes zero are called roots of the function. Fig. 3.1.1 shows arbitrary curve of $f(x)$.

As shown in Fig. 3.1.1,

$$f(x) = 0 \qquad \text{at} \quad x = x_1$$
$$\& \; x = x_2$$

$\therefore$ x_1 and x_2 are roots of $f(x)$

x_2 lies in interval [a, b]. From the figure it is clear that,

$$f(x) = -\text{ve value when } x = a$$
$$= +\text{ve value when } x = b$$

Thus $f(a)$ and $f(b)$ have opposite signs if root lies between [a,b]. The same thing is true for x_1. This principle is applicable for continuous functions only. It is also called intermediate value theorem.

Since $f(a)$ & $f(b)$ have opposite signs, their multiplication is always less than zero. Then we can state,

$$\boxed{\textit{if } f(a) \times f(b) < 0, \textit{ then root of function lies between } [a, b]}$$

Bisection methods calculates new point `c' i.e.

$$c = \frac{a+b}{2} \qquad \text{... (3.1.2)}$$

The point `c' lies exactly at the centre of interval [a, b]. This situation is shown in Fig. 3.1.2.

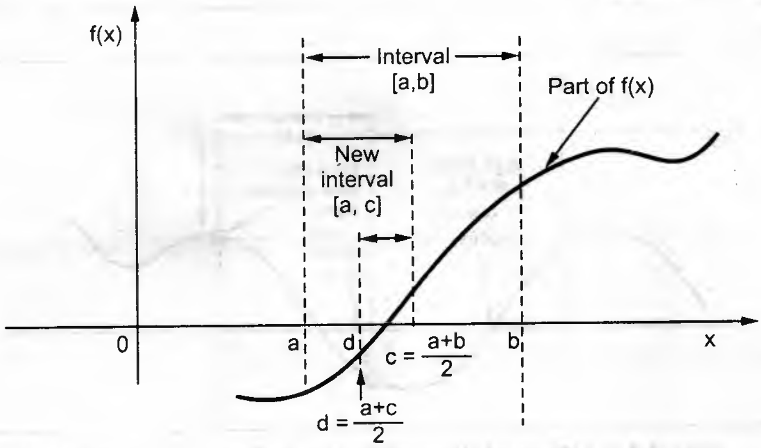


Fig. 3.1.2 Bisection method

It divides intervals in half and checks for presence of root.

From Fig 3.1.2 it is clear that $f(a)$ & $f(c)$ have opposite signs, therefore root lies in interval [a, c]. Thus new interval is [a, c]. Next point `d' is calculated by

$$d = \frac{a+c}{2} \qquad \text{(Which is similar to eq. 3.1.2)}$$

From Fig. 3.1.2 it is clear that $f(d)$ & $f(c)$ have opposite signs, therefore root lies in interval [a, c]. Therefore new interval is [d, c]. Thus the method continues to approximate and the centre point becomes very much close to actual root.

If small error 'ε' is allowed between actual value of root and approximated value of root then number of steps required are given as

$$n \geq \frac{\log(b-a) - \log \varepsilon}{\log 2} \qquad \text{... (3.1.3)}$$

where n = number of steps or iterations required to approximate root with in limit of 'ε'.

and $[a, b]$ = initial interval taken,

Proof of equation 3.1.3 is very simple and it is left as an exercise to students. Here error in the method will be zero if centre point of the interval lies exactly at the centre of interval.

Ex. 3.1.1 : *Using bisection method find the root of equation $x^3 - 1.8x^2 - 10x + 17 = 0$ that lies between the interval (1,2) at the end of 5^{th} iteration.*

Sol. : The given interval is [a, b] = [1, 2] and number of iterations are n = 5

Given that, $f(x) = x^3 - 1.8x^2 - 10x + 17$ ... (3.1.4)

Iteration No. 1

$$f(a) = f(1) = (1)^3 - 1.8(1)^2 - 10(1) + 17$$
$$= 6.2$$
$$f(b) = f(2) = (2)^3 - 1.8(2)^2 - 10(2) + 17$$
$$= -2.2$$
$$c = \frac{a+b}{2} = \frac{1+2}{2}$$
$$= 1.5$$
$$f(c) = f(1.5) = (1.5)^3 - 1.8(1.5)^2 - 10(1.5) + 17$$
$$= 1.325$$

Here $f(2) \cdot f(1.5) < 0$

or $f(b) \cdot f(c) < 0$

Hence root lies between b & c. Therefore replace \`a' by \`c' in new interval

∴ New interval becomes

$[a, b] = [1.5, 2]$ Here \`a' is replaced by \`c' i.e. 1.5.

Iteration No. 2

We know from iteration No. 1 that a = 1.5, b = 2

$$\therefore \quad c = \frac{a+b}{2} = \frac{1.5+2}{2} = 1.75$$
$$f(a) = f(1.5) = (1.5)^3 - 1.8(1.5)^2 - 10(1.5) + 17$$
$$= 1.325 \quad \text{(Also obtained in iteration No. 1)}$$
$$f(b) = f(2) = -2.2 \quad \text{(From iteration No. 1)}$$
$$f(c) = f(1.75) = (1.75)^3 - 1.8(1.75)^2 - 10(1.75) + 17$$
$$= -0.653125$$

Here $f(1.5) \cdot f(1.75) < 0$

or $f(a) \cdot f(c) < 0$

Hence root lies between a & c. Therefore replace \`b' by \`c' (b← c) in the new interval

∴ New interval becomes

$[a, b] = [1.5, 1.75]$ Here \`b' ← \`c'.

Iteration No. 3

From iteration No. 2 we have = 1.5, b = 1.75

$$\therefore \quad c = \frac{a+b}{2} = \frac{1.5+1.75}{2}$$
$$= 1.625$$

We know that,

$$f(1.5) = f(a) = 1.325$$
$$f(1.75) = f(b) = -0.653125$$

and $f(1.625) = f(c) = (1.625)^3 - 1.8(1.625)^2 - 10(1.625) + 17$

$$= 0.287891$$

Here $f(1.75) \cdot f(1.625) < 0$

or $f(b) \cdot f(c) < 0$

Hence root lies between b & c. Therefore replace `a' by `c' (a ← c) in the new interval

∴ New interval becomes

$$[a, b] = [1.625, 1.75] \qquad \because \; a \leftarrow c$$

Iteration No. 4

We have a = 1.625, b = 1.75

$$\therefore \qquad c = \frac{a+b}{2} = \frac{1.625 + 1.75}{2}$$
$$= 1.6875$$

We know that,

$$f(1.625) = f(a) = 0.287891$$
$$f(1.75) = f(b) = -0.653125$$

and $f(1.6875) = (1.6875)^3 - 1.8(1.6875)^2 - 10(1.6875) + 17$

$$= -0.195361$$

Here $f(1.625) \cdot f(1.6875) < 0$

or $f(a) \cdot f(c) < 0$

Hence root lies between a & c. Therefore replace `b' by `c' (b ← c) in next interval

∴ New interval becomes

$$[a, b] = [1.625, 1.6875]$$

Iteration No. 5

We have a = 1.625, b = 1.6875

$$\therefore \qquad c = \frac{a+b}{2} = \frac{1.625 + 1.6875}{2}$$
$$= 1.65625$$

We know that,

$$f(1.625) = f(a) = 0.287891$$
$$f(1.6875) = f(b) = -0.195361$$

$$f(1.65625) = f(c) = (1.65625)^3 - 1.8(1.65625)^2 - 10(1.65625) + 17$$
$$= 0.04317$$

Here $f(1.6875) \cdot f(1.65625) < 0$

or $f(b) \cdot f(c) < 0$

Hence root lies between b & c.

Since root at the end of 5^{th} iteration is asked, here we will stop iterations. The approximate value of root will be at the center of `b' and `c'.

Thus

$$\text{Root} = \underset{\substack{\uparrow \\ \text{Lower limit of} \\ \text{an interval}}}{a} + \underset{\substack{\uparrow \\ \text{Center of} \\ \text{an interval}}}{\left(\frac{b-a}{2}\right)} = \underset{\substack{\uparrow \\ \text{Center of an} \\ \text{interval} \\ \text{considered from} \\ \text{origin}}}{\frac{a+b}{2}}$$

$$\therefore \quad \text{Root} = \frac{1.6875 + 1.65625}{2}$$
$$= 1.671875$$

Thus,

Approximate value of root at the end of 5^{th} iteration = 1.671875.

Error in the value of function f(x) :

Here note that x = 1.671875 is not exact value of root. It is just an approximation. When we substitute exact value of root in given equation (eq. 3.1.4), it's value is zero.

$$f(x) = 0 \text{ at } x = \text{exact value of root of } f(x) \text{ and}$$

$$\left.\text{Approximate value of } f(x)\right|_{x=1.671875} = f(x)|_{x=1.671875}$$

$$f(x) = x^3 - 1.8x^2 - 10x + 17 \quad \text{from equation 3.1.4}$$

$$\therefore \quad f(x)|_{x=1.671875} = (1.671875)^3 - 1.8(1.671875)^2 - 10(1.671875) + 17$$
$$= -0.076881$$

Thus error in the value of function $f(x)$ at root

$$= \left| \, |f(x)|_{x=\text{exact root}} - |f(x)|_{x=\text{Approximate root}} \right|$$
$$= |\,0\,| - |-0.076881| = 0.076881$$

Thus,

Error in value of function = value or function at approximate root.

Error in approximated value of root :

Let's tabulate the results of this example. We will tabulate center point of every interval `c' and value of f(x) at c; i.e. f(c).

Table 3.1.1

Iteration No.	Center point of [a, b] = c	$f(x)\|_{x=c}$ = f(c)	Number of significant digits `unchanged' after decimal point.
1	1.5	1.325	—
2	1.75	– 0.653125	All digits are changed
3	1.625	0.287891	All digits are changed
4	1.6875	– 0.195361	One digit unchanged
5	1.65625	0.04317	One digit unchanged

At the end of 5^{th} iteration, the center point of interval is,

$$c = 1.671875$$

And this point is called as an approximate value of root. Thus all the values of `c' listed in Table 3.1.1 can be called approximate value of roots. And values of f(c) in the table are absolute error in the function f(x) at approximate root. In Table 3.1.1 note that,

> *As we go on increasing number of iterations, value of f(x) at approximate root [f(c)] goes on decreasing and approaches towards zero.*

In the Table also note that,

At fourth iteration c = 1.6875

At fifth iteration c = 1.65625

At the end of fifth iteration c = 1.671875
(6^{th} iteration)

One significant digit after decimal point is unchanged. Thus we can say that *approximate value of root at the end of fifth iteration is correct upto one significant digit after decimal point.* We can say this because the value of `c' is unchanged upto one digit (after decimal point) in two successive iterations. Thus second digit after decimal point represent an error.

∴ Maximum error in approximate value of root = Maximum value of second digit after decimal point

= 0.09

Thus,

> *Maximum value of error in obtained root = 0.09*

Reverse check of an error :

Now let's calculate number of iterations required to obtain error less than 0.09.

$$\therefore \qquad n \geq \frac{\log(b-a) - \log \varepsilon}{\log 2} \qquad \text{From equation 3.1.3.}$$

Here $[a, b] = [1, 2]$ Given in example

and $\varepsilon = \varepsilon_{max} = 0.09$

$\therefore \qquad n \geq 4$ iterations

Thus `4' iterations are required. Here we have performed `5' iterations. This verifies reverse check of an error in root.

3.1.2 Algorithm and Flow Chart

From the discussion of bisection method and example just solved, we will prepare an algorithm for the computer software. Various steps of an algorithm are listed below -

Assumptions :

1) The function f(x) is predefined.

2) Correct values of interval [a, b] are entered in which root lies.

Step 1 : ***Read interval [a, b] and permissible error or correct significant digits after decimal point.***

Step 2 : ***Calculate number of iterations `n' using***

$$n \geq \frac{\log(b-a) - \log \varepsilon}{\log 2}$$

Step 3 : ***Calculate center point of an interval*** $c = \dfrac{a+b}{2}$.

Step 4 : ***Calculate,***

$f(x)|_{x=a} = f(a) \quad f(x)|_{x=b} = f(b)$ ***and*** $f(x)|_{x=c} = f(c)$

Step 5 : ***If f(a).f(c) < 0, then root lies between a & c. Hence replace `b' by `c'.***

If f(b).f(c) < 0, then root lies between b & c. Hence replace `a' by `c'.

Step 6 : ***Repeat step 3 to step 5 until number of iterations are reached.***

OR

Repeat step 3 to step 5 until value of `c' repeats upto given significant digits after decimal point.

Step 7 : ***Approximate value of root is equal to centre point of latest interval.***

Step 8 : ***Display approximate value of root and stop.***

Flowchart :

Fig. 3.1.3 shows the flowchart of bisection method. The steps of an algorithm are also marked in the flowchart. (See Flow chart on next page)

3.1.3 Logic Development and C Program

A computer program in 'C' language is discussed here. The program uses a function given by equation 3.1.4 i.e.,

$$f(x) = x^3 - 1.8x^2 - 10x + 17 \qquad \dots (3.1.5)$$

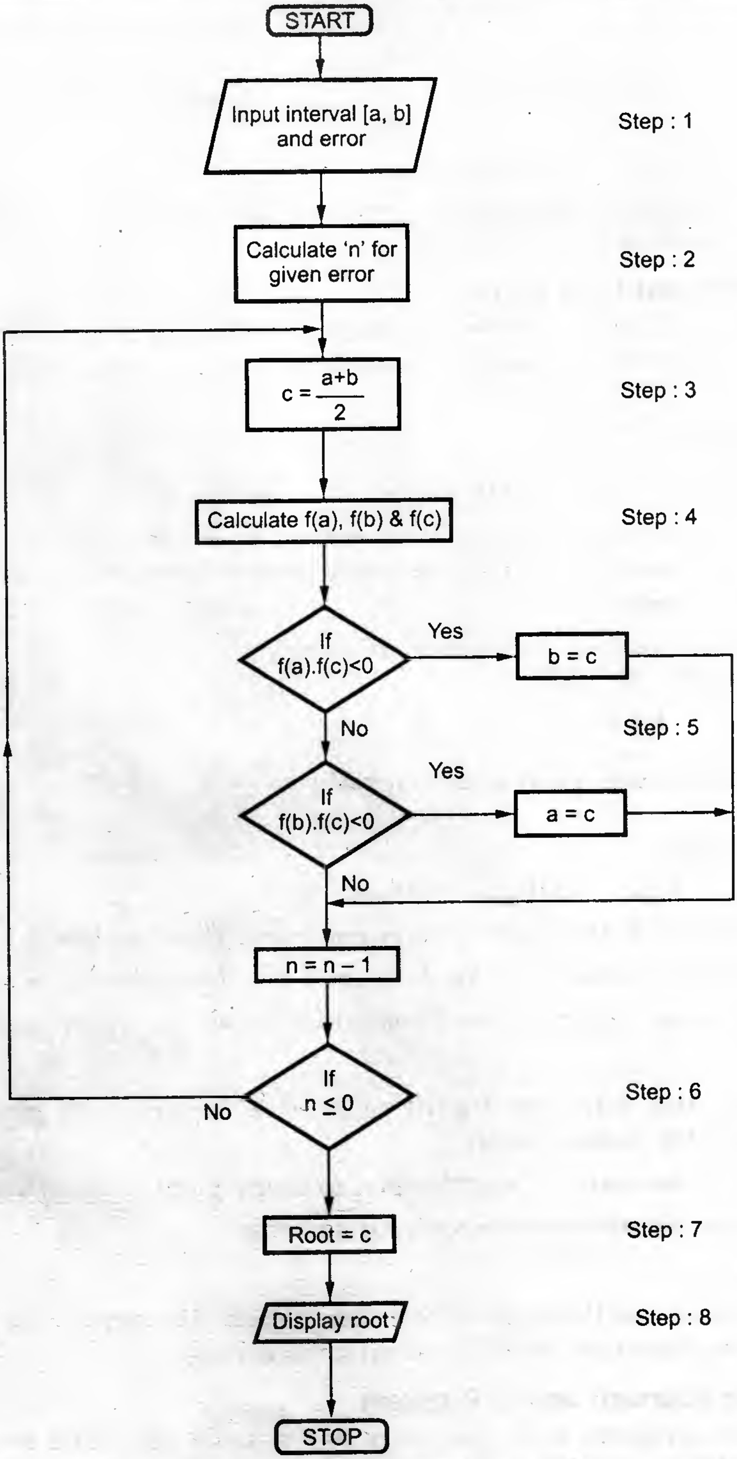


Fig. 3.1.3 Flowchart of bisection method

```
/*  Download this program from www.vtubooks.com                    */
/* file name : bisect.cpp                                           */

/*------------ BISECTION METHOD TO FIND ROOT OF AN EQUATION --------*/

/*      THE EXPRESSION FOR AN EQUATION IS DEFINED IN function fx
        YOU CAN WRITE DIFFERENT EQUATION IN function fx.
        HERE,
                    f(x) = x*x*x - 1.8*x*x - 10*x - 17

        INPUTS :  1) Initial interval [a,b] in which root is to
                      be found.
                  2) Permissible error in the root.

        OUTPUTS : 1) Number of iterations for given interval and
                      permissible error.
                  3) Value of the root in given interval.             */

/*------------------------------    PROGRAM   ----------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>

void main()
{
      double fx ( double x);      /* DECLARATION OF  FUNCTION  */
      double x,a,b,fa,fb,c,fc,err;
      int n,i;

      clrscr();
      printf("\n        BISECTION METHOD TO FIND ROOT OF AN EQUATION");

      printf("\n\n            f(x) = x*x*x - 1.8*x*x - 10*x - 17");

      printf("\n\nEnter an interval [a,b] in "
             "which root is to be found");

      printf("\na = ");
      scanf("%lf",&a);      /* INTERVAL [a,b] IS TO BE ENTERED HERE     */
      printf("b = ");
      scanf("%lf",&b);
      printf("\nEnter the value of permissible error = ");
      scanf("%lf",&err);
      n = (log10(abs(b - a)) - log10( err ) )/log10(2);   /* CALCULATION OF STEPS 'n'*/
      n = n + 1;       /* n SHOULD NOT BE A FRACTIONAL NUMBER;  ADD '1'     */
      i = 0;
      printf("\nNumber of iterations for this error bound are = %d",n);
      printf("\n\npress any key for step by step display of intervals");
      getch();
      printf("\n\n                     RESULTS OF BISECTION METHOD\n");
      while(n-- > 0)
      {
         fa = fx(a);            /*     CALCULATE f(x) AT x = a         */
         fb = fx(b);            /*     CALCULATE f(x) AT x = b         */
         c = (a + b)/2;         /*     CENTER OF THE INTERVAL          */
         fc = fx(c);            /*     CALCULATE f(x) AT x = c         */

         i++;

         printf("\n\n%d     a = %lf        b = %lf       c = %lf",i,a,b,c);
         printf("\n  f(a) = %lf     f(b) = %lf        f(c) = %lf",fa,fb,fc);

         if( (fc*fa) < 0) b = c; /* IF f(c)f(a) < 0, NEW INTERVAL IS [a,c]  */
         if( (fc*fb) < 0) a = c; /* IF f(c)f(b) < 0, NEW INTERVAL IS [b,c]  */

         printf("\n                   interval : a = %lf      b = %lf",a,b);

         getch();
      }
      x = a + (b - a)/2;     /* ROOT  = a + half interval [a,b]        */
```

```
    printf("\nThe value of root is = %20.15lf",x);        /*   ROOT      */
}
/*---------- FUNCTION PROCEDURE TO CALCULATE VALUE OF EQUATION --------*/

double fx ( double x)
{
  double f;
  f = x*x*x - 1.8*x*x - 10*x + 17;           /*      FUNCTION f(x)    */
  return(f);
}
/*----------------------- END OF PROGRAM ----------------------------*/
```

A root of *f(x)* in interval [1, 2] is calculated in example 3.1.1. Now let's see how logic is developed to find root of f(x) using bisection method. The source code of the program is shown after flowchart.

The program first asks for an interval [a, b]. Then it asks for permissible error. All these variables are defined as double floating type. The program then calculates number of iterations `n' required using equation 3.1.3 i.e.,

$$n \geq \frac{\log(b-a) - \log\varepsilon}{\log 2} \qquad \dots (3.1.6)$$

Here `n' is an integer type variable, and a, b and ε are floating point numbers. The program implements this relation as follows

$$n = (\log_{10}(\text{abs}(b - a)) - \log_{10}(\text{err}))/\log_{10}(2) \qquad \dots (3.1.7)$$

Here `abs(b - a)` takes absolute value of difference and discards sign. This is done to avoid error due to negative values. Log of negative value gives error in computer. The right handside of this equation is double float, its result will also be of the same type. That is a real number with integer and fractional part. This result is assigned to `n' which is defined as integer.

As in this example, we know tehat a = 1 and b = 2

Let's calculate `n' for error = 0.05 equation 3.1.7 becomes,

$$n = (\log_{10}(\text{abs}(2-1)) - \log_{10}(0.05))/\log_{10}(2)$$
$$= 4.3219281$$

Since `n' is an integer type variable, the above obtained value is rounded to `4' in computer.

$\therefore \quad n = 4$

This satisfies equation 3.1.6. Hence value of n is increased by `1' in next instruction in computer program.

Thus, $n = n + 1$ gives

$n = 5$ which is greater than 4.3219281

The `while` loop then keeps on computing a,b,c, f(a), f(b) and f(c) till number of iterations are reached. The program also prints iteration number, new interval [a, b] and value of root at the end of that iteration. You have to press any key to go for next iteration.

A separate function `double fx(double x)` is written to evaluate the value of f(x) of equation 3.1.5. i.e.

$$f(x) = x^3 - 1.8x^2 - 10x + 17 \quad \text{and}$$

in computer, `f = x * x * x - 1.8 * x * x - 10 * x + 17` expression calculates this value. If you want to use the same program for other function, then simply change the above expression for `f`.

Results of the program :

Compile and make exe file of this program `bisect`. Run the program. It first displays an expression you are evaluating and asks for the value of

a = Here enter 1. Then it will display,

b = Here enter 2

Then computer asks for

`Enter the value of permissible error =` Here enter 0.05.

The program then displays,

`Number of iterations for this error bound are = 5`

Then press any key to see step by step results. All these results are displayed below :

```
-------------------------- Results ---------------------------------

        BISECTION METHOD TO FIND ROOT OF AN EQUATION

             f(x) = x*x*x - 1.8*x*x - 10*x - 17

Enter an interval [a,b] in which root is to be found
a = 1.0
b = 2.0

Enter the value of permissible error = 0.05

Number of iterations for this error bound are = 5

press any key for step by step display of intervals

                    RESULTS OF BISECTION METHOD

1     a = 1.000000          b = 2.000000           c = 1.500000
   f(a) = 6.200000       f(b) = -2.200000            f(c) = 1.325000
                           interval : a = 1.500000      b = 2.000000

2     a = 1.500000          b = 2.000000           c = 1.750000
   f(a) = 1.325000       f(b) = -2.200000            f(c) = -0.653125
                           interval : a = 1.500000      b = 1.750000

3     a = 1.500000          b = 1.750000           c = 1.625000
   f(a) = 1.325000       f(b) = -0.653125            f(c) = 0.287891
                           interval : a = 1.625000      b = 1.750000

4     a = 1.625000          b = 1.750000           c = 1.687500
   f(a) = 0.287891       f(b) = -0.653125            f(c) = -0.195361
                           interval : a = 1.625000      b = 1.687500

5     a = 1.625000          b = 1.687500           c = 1.656250
   f(a) = 0.287891       f(b) = -0.195361            f(c) = 0.043170
                           interval : a = 1.656250      b = 1.687500
The value of root is =    1.671875000000000
---------------------------------------------------------------------
```

Now let's run the same program with very small permissible error bound. Let an interval [a, b] = [1, 2]. Let an error bound be 0.00005. Run the above program and enter values of [a, b] and error

```
--------------------------------------- Results ---------------------------------------------------
                    BISECTION METHOD TO FIND ROOT OF AN EQUATION

                    f(x) = x*x*x - 1.8*x*x - 10*x - 17
Enter an interval [a,b] in which root is to be found
a = 1.0
b = 2.0

Enter the value of permissible error = 0.00005
Number of iterations for this error bound are = 15

press any key for step by step display of intervals
                              RESULTS OF BISECTION METHOD

1      a = 1.000000          b = 2.000000          c = 1.500000
    f(a) = 6.200000       f(b) = -2.200000      f(c) = 1.325000
                          interval : a = 1.500000     b = 2.000000

2      a = 1.500000          b = 2.000000          c = 1.750000
    f(a) = 1.325000       f(b) = -2.200000      f(c) = -0.653125
            interval : a = 1.500000     b = 1.750000

3      a = 1.500000          b = 1.750000          c = 1.625000
    f(a) = 1.325000       f(b) = -0.653125      f(c) = 0.287891
                          interval : a = 1.625000     b = 1.750000

4      a = 1.625000          b = 1.750000          c = 1.687500
    f(a) = 0.287891       f(b) = -0.653125      f(c) = -0.195361
                          interval : a = 1.625000     b = 1.687500

5      a = 1.625000          b = 1.687500          c = 1.656250
    f(a) = 0.287891       f(b) = -0.195361      f(c) = 0.043170
                          interval : a = 1.656250     b = 1.687500

6      a = 1.656250          b = 1.687500          c = 1.671875
    f(a) = 0.043170       f(b) = -0.195361      f(c) = -0.076881
                          interval : a = 1.656250     b = 1.671875

7      a = 1.656250          b = 1.671875          c = 1.664062
    f(a) = 0.043170       f(b) = -0.076881      f(c) = -0.017050
                          interval : a = 1.656250     b = 1.664062

8      a = 1.656250          b = 1.664062          c = 1.660156
    f(a) = 0.043170       f(b) = -0.017050      f(c) = 0.013012
                          interval : a = 1.660156     b = 1.664062

9      a = 1.660156          b = 1.664062          c = 1.662109
    f(a) = 0.013012       f(b) = -0.017050      f(c) = -0.002031
                          interval : a = 1.660156     b = 1.662109
10     a = 1.660156          b = 1.662109          c = 1.661133
    f(a) = 0.013012       f(b) = -0.002031      f(c) = 0.005487
                          interval : a = 1.661133     b = 1.662109
```

```
11     a = 1.661133           b = 1.662109             c = 1.661621
    f(a) = 0.005487        f(b) = -0.002031         f(c) = 0.001727
                           interval : a = 1.661621      b = 1.662109

12     a = 1.661621           b = 1.662109             c = 1.661865
    f(a) = 0.001727        f(b) = -0.002031         f(c) = -0.000152
                           interval : a = 1.661621      b = 1.661865

13     a = 1.661621           b = 1.661865             c = 1.661743
    f(a) = 0.001727        f(b) = -0.000152         f(c) = 0.000787
                           interval : a = 1.661743      b = 1.661865

14     a = 1.661743           b = 1.661865             c = 1.661804
    f(a) = 0.000787        f(b) = -0.000152         f(c) = 0.000317
                           interval : a = 1.661804      b = 1.661865
15     a = 1.661804           b = 1.661865             c = 1.661835
    f(a) = 0.000317        f(b) = -0.000152         f(c) = 0.000083
                           interval : a = 1.661835      b = 1.661865
The value of root is =     1.661849975585938
-------------------------------------------------------------------------
```

Observe that `15' iterations are required to achieve the given error limit.

In the 7^{th} iteration c = 1.664062

In the 8^{th} iteration c = 1.660156

Since two digits after decimal point are unchanged here, the value of root i.e.`c' will be correct upto two significant digits i.e. 1.66.

Observe that,

In the 14^{th} iteration c = 1.661804

In the 15^{th} iteration c = 1.661835

Here four digits after decimal point are unchanged in `c'. Therefore we can say that the root (i.e. latest value of `c') is correct upto 4 digits after decimal point (i.e. Root upto 4 digits after decimal point = 1.6618). The value of root calculated at the end of 15^{th} iteration (16^{th} iteration) is,

Root = 1.661849975585938 (Correct upto `4' digits)

This also varifies with number of zeros present after decimal point in given error. There are four zeros and root is correct up to `4' digits after decimal point.

3.1.4 Solved Examples

Here there are some of the examples solved using bisection method. All the calculations here are performed on computer with double float. The values of variables upto 15 significant digits after decimal point are used. Whenever you will solve these examples with normal pocket calculators, some mismatch in the values given here is possible.

Ex. 3.1.2 : *Obtain the root of the following equation correct to three decimal places using bisection method. Give the steps in detail.*

$$f(x) = x^3 + x^2 + x + 7 = 0$$

Sol. : To find an initial approximation let a = –2 and b = –3.

Then, $f(a) = f(-2) = -8 + 4 - 2 + 7 = 1$ (Positive)

and $f(b) = f(-3) = -27 + 9 - 3 + 7 = -14$ (Negative)

$\therefore$ Root lies between [a, b] = [–2, –3]

Iteration No. 1 $a = -2, \quad b = -3$

$$\therefore \quad c = \frac{a+b}{2} = -2.5$$

$$f(-2) = f(a) = 1$$

$$f(-3) = f(b) = -14$$

and $f(-2.5) = f(c) = -4.875$

Here $f(a) \cdot f(c) < 0$ $\therefore$ root lies in [a, c]

Hence $b \leftarrow c$ $\therefore$ new interval [a, b] = [–2, –2.5]

Iteration No. 2 $a = -2, \quad b = -2.5$

$$\therefore \quad c = \frac{a+b}{2} = -2.25$$

$$f(-2) = f(a) = 1$$

$$f(-2.5) = f(b) = -4.875$$

and $f(-2.25) = f(c) = -1.518125$

Hence $f(a) \cdot f(c) < 0$ $\therefore$ root lies in [a, c]

Here $b \leftarrow c$ $\therefore$ new interval [a, b] = [–2, –2.25]

Iteration No. 3 $a = -2, \quad b = -2.25$

$$\therefore \quad c = \frac{a+b}{2} = -2.125$$

$$f(-2) = f(a) = 1$$

$$f(-2.25) = f(b) = -1.578125$$

and $f(-2.125) = f(c) = -0.205078$

Here $f(a) \cdot f(c) < 0$ $\therefore$ root lies in [a, c]

Hence $b \leftarrow c$ $\therefore$ new interval [a, b] = [–2, –2.125]

Iteration No. 4 $a = -2, \quad b = -2.125$

$$\therefore \quad c = \frac{a+b}{2} = -2.0625$$

$$f(-2) = f(a) = 1$$

$$f(-2.125) = f(b) = -0.205078$$

and $f(-2.0625) = f(c) = 0.417725$

Here $f(b) \cdot f(c) < 0$ $\therefore$ root lies in [b, c]

Hence $a \leftarrow c$ $\therefore$ new interval [a, b] = [–2.0625, –2.125]

Iteration No. 5 $a = -2.0625, \quad b = -2.125$

$$\therefore \quad c = \frac{a+b}{2} = -2.09375$$

$f(-2.0625) = f(a) = 0.417725$

$f(-2.125) = f(b) = -0.205078$

and $f(-2.09375) = f(c) = -0.111481$

Here $f(b) \cdot f(c) < 0$ $\therefore$ root lies in [b, c]

Hence a ← c $\therefore$ new interval [a, b] = [–2.09375, –2.125]

Iteration No. 6

$a = -2.09375, \quad b = -2.125$

$$\therefore \quad c = \frac{a+b}{2} = -2.109375$$

$f(-2.09375) = f(a) = 0.111481$

$f(-2.125) = f(b) = -0.205078$

and $f(-2.109375) = f(c) = -0.045498$

Here $f(a) \cdot f(b) < 0$ $\therefore$ root lies in [a, c]

Hence b ← c $\therefore$ new interval [a, b] = [–2.09375, –2.109375]

Iteration No. 7 $a = -2.09375, \quad b = -2.109375$

$$\therefore \quad c = \frac{a+b}{2} = -2.101562$$

$f(-2.09375) = f(a) = 0.111481$

$f(-2.109375) = f(b) = -0.045498$

and $f(-2.101562) = f(c) = -0.033315$

Here $f(a) \cdot f(c) < 0$ $\therefore$ root lies in [b, c]

Hence a ← c new interval [a, b] = [–2.101562, –2.109375]

Iteration No. 8 $a = -2.101562, b = -2.109375$

$$\therefore \quad c = \frac{a+b}{2} = -2.105469$$

[Here value of `c' repeating to two digits after decimal point i.e. –2.10]

$f(-2.101562) = f(a) = 0.033315$

$f(-2.109375) = f(b) = -0.045498$

and $f(-2.105469) = f(c) = -0.006010$

Here $f(a) \cdot f(c) < 0$ $\therefore$ root lies in [a, c]

Hence b ← c $\therefore$ new interval [a, b] = [–2.101562, –2.105469]

Iteration No. 9 $a = -2.101562, b = -2.105469$

$\therefore \quad c = \dfrac{a+b}{2} = -2.103516$

$f(-2.101562) = f(a) = 0.033315$

$f(-2.105469) = f(b) = -0.006010$

and $f(-2.103516) = f(c) = -0.013673$

Here $f(b) \cdot f(c) < 0$ $\therefore$ root lies in [b, c]

Hence $a \leftarrow c$ $\therefore$ new interval [a, b] = [–2.103516, –2.105469]

Iteration No. 10 $a = -2.103516, \quad b = -2.105469$

$\therefore \quad c = \dfrac{a+b}{2} = -2.104492$

$f(-2.103516) = f(a) = 0.013673$

$f(-2.105469) = f(b) = -0.006010$

and $f(-2.104492) = f(c) = -0.003836$

Here $f(b) \cdot f(c) < 0$ $\therefore$ root lies in [b, c]

Hence $a \leftarrow c$ $\therefore$ new interval [a, b] = [–2.104492, –2.105469]

Iteration No. 11

$a = -2.104492, \quad b = -2.1055469$

$\therefore \quad c = \dfrac{a+b}{2} = -2.104980$

Thus Value of \`c' in 10^{th} iteration = **– 2.104492**

and Value of \`c' in 11^{th} iteration = **– 2.104980**

Here we observe that values repeat upto 3 digits after decimal point. Since \`c' is the center of interval [a, b] and represents root, we can say that

Root of f(x) correct upto \`3' digits after decimal point = – 2.104980

$\therefore$ Root = – 2.104980 correct to 3 decimal places is the answer.

Ex. 3.1.3 : *Obtain a root of the following equation.*

$f(x) = x^3 - 4x - 9 = 0$

Find the root for permissible error of 0.02

Sol. : Let's first find the interval.

$f(0) = -9, \quad f(1) = -12, \quad f(2) = -9, \quad f(3) = 6$

Thus $f(2) \cdot f((3) < 0$, means root lies between 2 and 3.

Hence take interval as [a, b] = [2, 3]

Permissible error $\varepsilon = 0.02$

$\therefore$ Number of iterations are given from equation 3.1.3 as,

$$n \geq \frac{\log(b-a)-\log\varepsilon}{\log 2} \geq \frac{\log(3-2)-\log(0.02)}{\log 2}$$

$$\geq 5.64$$

Take $n = 6$

Iteration No. 1 $a = 2, \quad b = 3$

$\therefore \quad c = \dfrac{a+b}{2} = 2.5$

$f(2) = f(a) = -9$

$f(3) = f(b) = 6$

and $f(2.5) = f(c) = -3.375$

$\because \quad f(b) \,.\, f(c) < 0$ New interval will be $[a, b] = [2.5, 3]$

Iteration No. 2 $a = 2.5, \quad b = -3 \quad \therefore c = 2.75$

$f(2.5) = f(a) = -3.375$

$f(3) = f(b) = 6$

$f(2.75) = f(c) = -\,0.796875$

$\because \quad f(a) \,.\, f(c) < 0$ New interval will be $[a, b] = [2.5, 2.75]$

Iteration No. 3 $a = 2.5, \quad b = 2.75 \quad \therefore c = 2.625$

$f(2.5) = f(a) = -3.375$

$f(2.75) = f(b) = 0.796875$

$f(2.625) = f(c) = -1.412109$

$\because \quad f(b) \,.\, f(c) < 0$ New interval will be $[a, b] = [2.625, 2.75]$

Iteration No. 4 $a = 2.625, \quad b = 2.75 \quad \therefore c = 2.6875$

$f(2.625) = f(a) = -\,1.412109$

$f(2.75) = f(b) = 0.796875$

$f(2.6875) = f(c) = -\,0.339111$

$\because \quad f(b) \,.\, f(c) < 0$ New interval will be $[a, b] = [2.6875, 2.75]$

Iteration No. 5 $a = 2.6875, \quad b = 2.75 \quad \therefore c = 2.71875$

$f(2.6875) = f(a) = -\,0.339111$

$f(2.75) = f(b) = 0.796875$

$f(2.71875) = f(c) = 0.220917$

$\because \quad f(a) \,.\, f(c) < 0$ New interval will be $[a, b] = [2.6875, 2.71875]$

Iteration No. 6 $a = 2.6875, \quad b = 2.71875 \quad \therefore c = 2.703125$

$f(2.6875) = f(a) = -\,0.339111$

$f(2.71875) = f(b) = 0.220917$

$f(2.703125) = f(c) = -\,0.061077$

$\because \quad f(a) \,.\, f(c) < 0$ New interval is $[a, b] = [2.703125, 2.71875]$

$\therefore$ Root = c $= \dfrac{a+b}{2} = \dfrac{2.703125 + 2.71875}{2}$

$= 2.7109375$ (At the end of 6^{th} iteration)

Thus, Root $= 2.7109375$ at the end of 6^{th} iteration.

Ex. 3.1.4 : *Use the method of bisection to find the root of the equation,*

$f(x) = x^4 + 2x^3 - x - 1 = 0$

lying in the interval [0, 1] at the end of sixth iteration. How many iterations are required if the permissible error is $\varepsilon = 0.0005$?

Sol. : The given interval is [a, b] = [0, 1]

Iteration No. 1 a = 0, b = 1 $\therefore$ c $= \dfrac{a+b}{2} = \dfrac{0+1}{2} = 0.5$

$f(a) = f(0) = -1$

$f(b) = f(1) = 1$

$f(c) = f(0.5) = -1.1875$

Since $f(b) \cdot f(c) < 0$, root lies between 'b' and 'c'. Hence 'a' will be replaced by 'c'. Therefore new interval is [a, b] = [0.5, 1]

Iteration No. 2 a = 0.5, b = 1 $\therefore$ c $= \dfrac{0.5+1}{2} = 0.75$

$f(a) = f(0.5) = -1.1875$

$f(b) = f(1) = 1$

$f(c) = f(0.75) = -0.5898$

Since $f(b) \cdot f(c) < 0$, root lies between 'b' and 'c'. Hence 'a' will be replaced by 'c'. and new interval will be [a, b] = [0.75, 1]

Iteration No. 3 a = 0.75, b = 1 $\therefore$ c $= \dfrac{0.75+1}{2} = 0.875$

$f(a) = f(0.75) = -0.5898$

$f(b) = f(1) = 1$

$f(c) = f(0.875) = 0.0510254$

$\because$ $f(a) \cdot f(c) < 0$, root lies between 'a' and 'c'

$\therefore$ $b \leftarrow c$ and New interval [a, b] = [0.75, 0.875]

Iteration No. 4 a = 0.75, b = 0.875 $\therefore$ c $= \dfrac{0.75+0.875}{2} = 0.8125$

$f(a) = f(0.75) = -0.5898$

$f(b) = f(0.875) = 0.0510254$

$f(c) = f(0.8125) = -0.3039398$

$\because$ $f(b) \cdot f(c) < 0$, root lies between 'b' and 'c'

$\therefore$ $a \leftarrow c$ and New interval [a, b] = [0.8125, 0.875]

Iteration No. 5 $a = 0.8125, \quad b = 0.875 \quad c = \dfrac{0.8125+0.875}{2} = 0.84375$

$$f(a) = f(0.8125) = -0.3039398$$
$$f(b) = f(0.875) = 0.0510254$$
$$f(c) = f(0.84375) = -0.1355733$$

$\because$ $f(b)\cdot f(c) < 0$, root lies between 'b' and 'c'

$\therefore$ $a \leftarrow c$ and new interval [a, b] = [0.84375, 0.875]

Iteration No. 6 $a = 0.84375, \quad b = 0.875 \quad c = \dfrac{0.84375+0.875}{2} = 0.859375$

$$f(a) = f(0.84375) = -0.1355733$$
$$f(b) = f(0.875) = 0.0510254$$
$$f(c) = f(0.859375) = -0.0446147$$

$\because$ $f(b)\cdot f(c) < 0$, root lies between 'b' and 'c'

$\therefore$ $a \leftarrow c$ and New interval [a, b] = [0.859375, 0.875]

Hence root will be $\dfrac{a+b}{2}$ i.e.,

$$\text{root} = \frac{0.859375+0.875}{2} = 0.8671875$$

Thus root = 0.8671875 at the end of 6^{th} iteration.

To determine number of iterations for permissible error ε = 0.0005

Given interval is [a, b] = [0, 1]

Permissible error ε = 0.0005

$\therefore$ Number of iterations are given from equation 3.1.3 as,

$$n \geq \frac{\log(b-a) - \log \varepsilon}{\log 2} \geq \frac{\log(1-0) - \log(0.0005)}{\log 2} \geq \frac{0-(-3.30103)}{0.30103}$$

$$\geq 10.96$$

Hence n = 11 iterations are required to get the error less than permissible error.

Ex. 3.1.5 *Use bisection method to determine the root of $f(x) = e^{-x} - x = 0$.*

[May-96, 8 Marks; May-98, 8 Marks]

Sol. : To obtain an interval [a, b] in which root lies

Let $f(0) = e^0 - 0 = 1$

$$f(1) = e^{-1} - 1 = -0.6321205$$

Since $f(0)\cdot f(1) < 0$, root lies in [0, 1]

Iteration No. 1 Take a = 0, b = 1

$$\text{Hence, } c = \frac{a+b}{2} = \frac{0+1}{2} = 0.5$$

$$\therefore \quad f(c) = f(0.5) = 0.1065306$$

$$\text{And} \quad f(a) = f(0) = 1$$

$$f(b) = f(1) = -0.6321205$$

$$\text{Since } f(0.5) \cdot f(1) < 0, \text{ root lies in } [0.5, 1]$$

Iteration No. 2

$$\text{New interval : } a = 0.5, \quad b = 1$$

$$\text{Hence, } c = \frac{a+b}{2} = \frac{0.5+1}{2} = 0.75$$

$$\therefore \quad f(c) = f(0.75) = -0.2776334$$

$$\text{and} \quad f(a) = f(0.5) = 0.1065306$$

$$f(b) = f(1) = -0.6321205$$

$$\text{Since } f(0.5) \cdot f(0.75) < 0, \text{ root lies in } [0.5, 0.75]$$

Iteration No. 3

$$\text{New interval : } a = 0.5, \quad b = 0.75$$

$$\text{Hence, } c = \frac{a+b}{2} = \frac{0.5+0.75}{2} = 0.625$$

$$\therefore \quad f(c) = f(0.625) = -0.0897385$$

$$\text{and} \quad f(a) = f(0.5) = 0.1065306$$

$$f(b) = f(0.75) = -0.2776334$$

$$\text{Since } f(0.5) \cdot f(0.625) < 0, \text{ root lies in } [0.5, 0.625]$$

Iteration No. 4

$$\text{New interval : } a = 0.5, \quad b = 0.625$$

$$\text{Hence, } c = \frac{a+b}{2} = \frac{0.5+0.625}{2} = 0.5625$$

$$\therefore \quad f(c) = f(0.5625) = 0.0072828$$

$$\text{and} \quad f(a) = f(0.5) = 0.1065306$$

$$f(b) = f(0.625) = -0.0897385$$

$$\text{Since } f(0.5625) \cdot f(0.625) < 0, \text{ root lies in } [0.5625, 0.625]$$

Iteration No. 5

$$\text{New interval : } a = 0.5625, \quad b = 0.625$$

$$\text{Hence, } c = \frac{a+b}{2} = \frac{0.5625+0.625}{2} = 0.59375$$

$$\therefore \quad f(c) = f(0.59375) = -0.0414975$$

$$\text{and} \quad f(a) = f(0.5625) = 0.0072828$$

$$f(b) = f(0.625) = -0.0897385$$

Since $f(0.5625) \cdot f(0.59375) < 0$, root lies in $[0.5625, \; 0.59375]$

Iteration No. 6

New interval : $a = 0.5625, \quad b = 0.59375$

$$\text{Hence, } c = \frac{a+b}{2} = \frac{0.5625 + 0.59375}{2} = 0.578125$$

$$\therefore \quad f(c) = f(0.578125) = -0.0171758$$

$$\text{and} \quad f(a) = f(0.5625) = 0.0072828$$

$$f(b) = f(0.59375) = -0.0414975$$

Since $f(0.5625) \cdot f(0.578125) < 0$, root lies in $[0.5625, \; 0.578125]$

Iteration No. 7

New interval : $a = 0.5625, \quad b = 0.578125$

$$\text{Hence, } c = \frac{a+b}{2} = \frac{0.5625 + 0.578125}{2} = 0.5703125$$

$$\therefore \quad f(c) = f(0.5703125) = -0.0049637$$

$$\text{and} \quad f(a) = f(0.5625) = 0.0072828$$

$$f(b) = f(0.578125) = -0.0171758$$

Since $f(0.5625) \cdot f(0.5703125) < 0$, root lies in $[0.5625, \; 0.5703125]$

Iteration No. 8

New interval : $a = 0.5625, \quad b = 0.5703125$

$$\text{Hence, } c = \frac{a+b}{2} = \frac{0.5625 + 0.5703125}{2} = 0.5664062$$

$$\therefore \quad f(c) = f(0.5664062) = -0.0011552$$

$$\text{and} \quad f(a) = f(0.5625) = 0.0072828$$

$$f(b) = f(0.5703125) = -0.0049637$$

Since $f(0.5664062) \cdot f(0.5703125) < 0$, root lies in $[0.5664062, \; 0.5703125]$

Iteration No. 9

New interval : $a = 0.5664062, \quad b = 0.5703125$

$$\text{Hence, } c = \frac{a+b}{2} = \frac{0.5664062 + 0.5703125}{2} = 0.5683593$$

$$\therefore \quad f(c) = f(0.5683593) = -0.0019053$$

$$\text{and} \quad f(a) = f(0.5664062) = 0.0011552$$

$$f(b) = f(0.5703125) = -0.0049637$$

Since $f(0.5664062) \cdot f(0.5683593) < 0$, root lies in $[0.5664062, \; 0.5683593]$

Iteration No. 10

New interval : $a = 0.5664062, \quad b = 0.5683593$

$$\text{Hence, } c = \frac{a+b}{2} = \frac{0.5664062 + 0.5683593}{2} = 0.5673827$$

$$\therefore \quad f(c) = f(0.5673827) = -\,0.0003753$$

$$\text{and} \quad f(a) = f(0.5664062) = 0.0011552$$

$$f(b) = f(0.5683593) = -\,0.0019053$$

Since $f(0.5673827) \cdot f(0.5664062) < 0$, root lies in [0.5664062, 0.5673827]

Iteration No. 11

New interval : a = 0.5664062, b = 0.5673827

$$\text{Hence, } c = \frac{a+b}{2} = \frac{0.5664062 + 0.5673827}{2} = 0.5668944$$

$$\therefore \quad f(c) = f(0.5668944) = 0.0003899$$

$$\text{And} \quad f(a) = f(0.5664062) = 0.0011552$$

$$f(b) = f(0.5673827) = -\,0.0003753$$

Since $f(0.5668944) \cdot f(0.5673827) < 0$, root lies in [0.5668944, 0.5673827]

Iteration No. 12

New interval : a = 0.5668944, b = 0.5673827

$$\text{Hence, } c = \frac{a+b}{2} = \frac{0.5668944 + 0.5673827}{2}$$

$$= 0.5671386$$

$$\therefore \quad f(c) = f(0.5671386) = 7.35551 \times 10^{-6}$$

$$\text{And} \quad f(a) = f(0.5668944) = 0.0003899$$

$$f(b) = f(0.5673827) = -\,0.0003753$$

Since $f(0.5671386) \cdot f(0.5673827) < 0$, root lies in [0.5671386, 0.5673827]

Iteration No. 13

New interval : a = 0.5671386, b = 0.5673827

$$\text{Hence, } c = \frac{a+b}{2} = \frac{0.5671386 + 0.5673827}{2}$$

$$= 0.5672606$$

Here note that,

12^{th} iteration $\Rightarrow$ root = **0.5671386**

13^{th} iteration $\Rightarrow$ root = **0.5672606**

The value of root repeats upto 3 significant digits after the decimal point. Hence the root = 0.5672606 is correct upto 3 decimal places.

Ex. 3.1.6 : *Find the real root of the equation $x^3 - x - 1 = 0$ lying between 1 and 2 correct to three decimal places by using bisection method.* **[Dec - 2003, 8 Marks]**

Sol. : Following table lists the solution.

Table 3.1.2 Solution of $x^3 - x - 1 = 0$

Itration No	a	f(a)	b	f(b)	c	f(c)
1	1	– 1	2	5	1.5	0.875
2	1	– 1	1.5	0.875	1.25	– 0.296875
3	1.25	– 0.296875	1.5	0.875	1.375	0.224609
4	1.25	– 0.296875	1.375	0.224609	1.3125	– 0.051514
5	1.3125	– 0.051514	1.375	0.224609	1.34375	0.082611
6	1.3125	– 0.051514	1.34375	0.082611	1.328125	0.014576
7	1.3125	– 0.051514	1.328125	0.014576	1.320312	– 0.018711
8	1.320312	– 0.018711	1.328125	0.014576	1.324219	– 0.002128
9	1.324219	– 0.002128	1.328125	0.014576	1.326172	0.006209
10	1.324219	– 0.002128	1.326172	0.006209	1.325195	0.002037
11	1.324219	– 0.002128	1.325195	0.002037	1.324707	– 0.000047
12	1.324707	– 0.000047	1.325195	0.002037	1.324951	0.000995

Since value of C = 1.324951 is repeating upto 3 decimal places, it is the required root.

Exercise

1. Using bisection method find root of $x^3 - x - 4$ to two decimal places. How many iterations are required if permissible error is 0.02?

   [Ans. : 1.796386718, n = 6 iterations]

2. Find out root of $f(x) = x^3 - x^2 - x - 1 = 0$ correct upto '4' digits after decimal point using bisection method. **[Hint : Take [a, b] = [1, 2], Root = 1.839279]**

3. Find root of equation $x^3 - 5x + 1$ using bisection method. Take error bound of 0.01. How many iterations are required for this error bound.

   [Ans. : n = 7, Root = 2.1289062]

4. For the $f(x) = x^3 - \sin x + 1$, find root in [-2, 1] up to '4' correct digits after decimal point. **[Ans. : Root = – 1.249053]**

University Questions

1. Use bisection method to determine the root of $f(x) = e^{-x} - x$. **[May - 96, May - 98]**
2. Find the real root of the equation $x^3 - x - 1 = 0$ lying between 1 and 2 correct to three decimal places by using bisection method. **[Dec - 2003, 8 Marks]**

3.2 Secant Method

3.2.1 Method

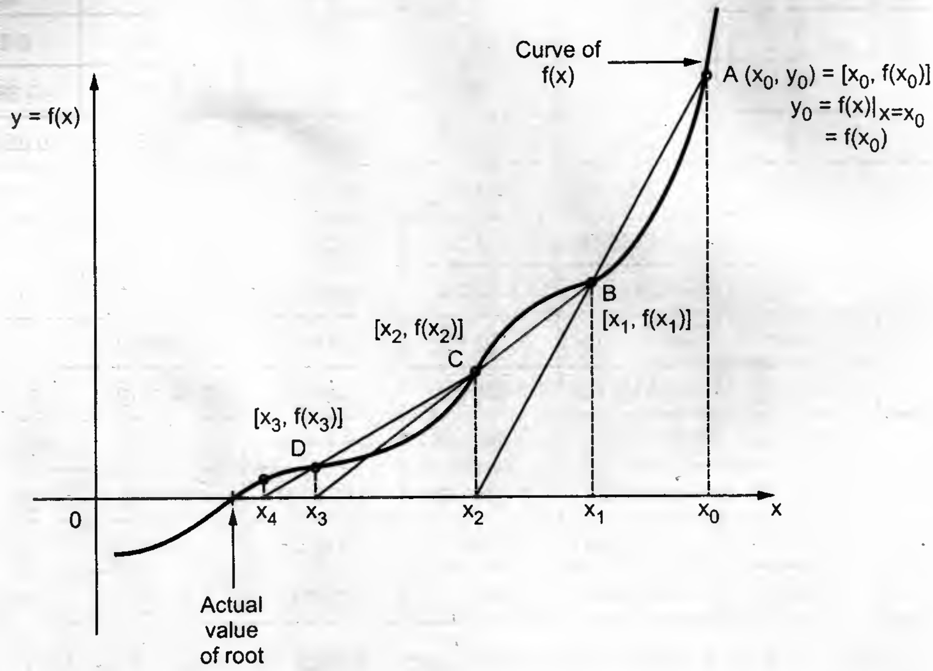


Fig. 3.2.1 Interpretation of secant method

Lets consider the curve of f(x) as shown in Fig. 3.2.1. We know that f(x) contains a polynomial equation or transcendental equation.

When we approximate f(x) by a first degree equation (maximum power of x is one in first degree equation), then it can be written as,

$$f(x) = a_0 x + a_1 \qquad \text{... (3.2.1)}$$

Here RHS is first degree equation. LHS is the polynomial or transcendental equation. Thus RHS is approximation of f(x).

When $\quad f(x) = 0 \quad \rightarrow \quad a_0 x + a_1 = 0$

Then the root of this first degree equation is given as,

$$x = -\frac{a_1}{a_0} \qquad \text{... (3.2.2)}$$

The first degree equation,

$$f(x) = a_0 x + a_1 \quad \text{takes the form of straight line equation}$$

$$y = mx + c$$

Here $\quad y = f(x), \quad m = a_0 = \text{slope}, \quad \text{and} \quad c = \text{intercept}$

Consider that curve AB of f(x) in Fig. 3.2.1 is approximated to a chord AB (first degree approximation). Then we can write first degree equation at point A for this chord using equation 3.2.1 as,

$$f(x_0) = a_0\, x_0 + a_1 \qquad \text{... (3.2.3)}$$

At point B we can write similar equation using equation 3.2.1 as,

$$f(x_1) = a_0\, x_1 + a_1 \qquad \text{... (3.2.4)}$$

Note that since both equations represent same chord (line AB), a_0 & a_1 remain same.

Let's solve equation 3.2.3 and equation 3.2.4 for a_0 & a_1. Subtracting equation 3.2.3 from equation 3.2.4 gives,

$$f(x_1) - f(x_0) = a_0\, x_1 + a_1 - a_0\, x_0 - a_1 = a_0\,(x_1 - x_0)$$

$$\therefore \quad a_0 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \qquad \text{... (3.2.5)}$$

Clearly a_0 obtained above is slope of line AB.

$$\text{i.e. slope} \quad m = a_0 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \qquad \text{... (3.2.6)}$$

Consider equation 3.2.4 i.e.,

$$f(x_1) = a_0\, x_1 + a_1$$

$$\therefore \quad a_1 = f(x_1) - a_0\, x_1$$

Putting value of a_0 from equation 3.2.5 in above equation,

$$a_1 = f(x_1) - \frac{f(x_1) - f(x_0)}{x_1 - x_0}\, x_1$$

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

$$\therefore \quad a_1 = \frac{x_1\, f(x_0) - x_0\, f(x_1)}{x_1 - x_0} \qquad \text{... (3.2.7)}$$

Chord AB also passes through x_2. Therefore we can write,

$$f(x_2) = a_0\, x_2 + a_1 \qquad \text{... (3.2.8)}$$

Observe that this equation is similar to equation 3.2.3 and equation 3.2.4. In this equation we have to substitute values of slope (a_0) and intercept (a_1) obtained earlier.

Putting values of a_0 & a_1 in equation 3.2.8.

$$f(x_2) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}\, x_2 + \frac{x_1\, f(x_0) - x_0\, f(x_1)}{x_1 - x_0}$$

Line AB crosses x axis at x_2, hence $f(x_2) = 0$ at x_2 . Therefore above equation becomes.

$$f(x_2) = 0 \quad \text{at} \quad x_2 \quad \text{for line AB}$$

$$\therefore \quad 0 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} x_2 + \frac{x_1 f(x_0) - x_0 f(x_1)}{x_1 - x_0}$$

$$\therefore \quad x_2 = -\frac{x_1 f(x_0) - x_0 f(x_1)}{f(x_1) - f(x_0)}$$

or
$$x_2 = \frac{x_0 f(x_1) - x_1 f(x_0)}{f(x_1) - f(x_0)}$$

Let's rearrange numerator of this equation

i.e.
$$x_2 = \frac{x_0 f(x_1) - x_1 f(x_0) - x_1 f(x_1) + x_1 f(x_1)}{f(x_1) - f(x_0)}$$

[Here $x_1 f(x_1)$ is subtracted and added in numerator]

$$\therefore \quad x_2 = \frac{x_1 [f(x_1) - f(x_0)] - f(x_1)(x_1 - x_0)}{f(x_1) - f(x_0)}$$

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

Thus
$$x_2 = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_1) \qquad \text{... (3.2.9)}$$

This equation gives the root of first degree equation defined by equation 3.2.8. Thus if we approximate curve AB of f(x) by a straight line, then point x_2 gives the root of f(x). That is using x_0 and x_1, we have obtained next approximation to the root at x_2.

On the same lines, using x_2 and x_1, we can obtain next approximation to the root at x_3. This can be done by approximating curve BC by straight line BC. The equation for this approximation also remains of the same form i.e.,

$$x_3 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2) \qquad \text{(from equation 3.2.9)}$$

$\therefore$ We can write recursive equation for secant method as follows :

$$\boxed{x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n)} \qquad \text{... (3.2.10)}$$

Ex. 3.2.1 : *Find root of $f(x) = x^3 - 5x - 7 = 0$ correct upto three places of decimal point.*

Sol. : Let's take initial approximations as $x_0 = 1$ and $x_1 = 2$ arbitrarily.

Iteration No. 1 From equation 3.2.10 we know that,

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n) \qquad \text{... (3.2.11)}$$

Here we will take n = 1

$$\therefore \quad x_2 = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_1) = 2 - \frac{2-1}{-9-(-11)} (-9) = 6.5$$

Iteration No. 2 With n = 2, equation 3.2.11 becomes,

$$x_3 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2) = 6.5 - \frac{6.5 - 2}{235.125 - (-9)} (235.125)$$

$$= 2.165899$$

Iteration No. 3 Here take n = 3 in equation 3.1.11. Then we have,

$$x_4 = x_3 - \frac{x_3 - x_2}{f(x_3) - f(x_2)} f(x_3)$$

$$= 2.165899 - \frac{2.165899 - 6.5}{(-7.669010) - (235.125)} (-7.669010) = 2.302798$$

Iteration No. 4 Here take n = 4 in equation 3.2.11. becomes,

$$x_5 = x_4 - \frac{x_4 - x_2}{f(x_4) - f(x_2)} f(x_4)$$

$$= 2.302798 - \frac{2.302798 - 2.165899}{(-6.302536) - (-7.669010)} (-6.302536)$$

$$= 2.934212$$

Iteration No. 5 Here take n = 5 in equation 3.2.11. becomes,

$$x_6 = x_5 - \frac{x_5 - x_4}{f(x_5) - f(x_4)} f(x_5)$$

$$= 2.934212 - \frac{2.934212 - 2.302798}{(3.591326) - (-6.302536)} (3.591326)$$

$$= 2.705018$$

Iteration No. 6 Here take n = 6 in equation 3.2.11. becomes,

$$x_7 = x_6 - \frac{x_6 - x_5}{f(x_6) - f(x_5)} f(x_6)$$

$$= 2.705018 - \frac{2.705018 - 2.934212}{(-0.732147) - (3.591326)} (-0.732147) = 2.743830$$

Iteration No. 7 Here take n = 7, Then next approximation equation for x_8 can be written using equation 3.2.11. as,

$$x_8 = x_7 - \frac{x_7 - x_6}{f(x_7) - f(x_6)} f(x_7)$$

$$= 2.743830 - \frac{2.743830 - 2.705018}{-0.061944 - (-0.732147)} (-0.061944) = 2.747417$$

Iteration No. 8 Take n = 8, Then next approximation x_9 from equation 3.2.11 will be,

$$x_9 = x_8 - \frac{x_8 - x_7}{f(x_8) - f(x_7)} f(x_8)$$

$$= 2.747417 - \frac{2.747417 - 2.743830}{0.001247 - (-0.061944)}(0.001247) = 2.747346$$

Let's stop iteration here.

Iteration No. 7 : 8^{th} approximation to root $= x_8 =$ **2.747417**

Iteration No. 8 : 9^{th} approximation to root $= x_9 =$ **2.747346**

From the above value of x_8 and x_9 we observe that values of x_8 and x_9 are similar upto `3' digits after decimal point. That means those digits will not change in further approximations now. Therefore if we call $x_9$ as the last approximation to root, then we can say that `three' digits after decimal point are correct in root.

i.e. Root = 2.747346 correct upto `3' decimal places.

Alternative and short method :

To solve this example using secant method, we select x_0 & x_1 (initial approximations) arbitrarily. But if we select x_0 & x_1 such that root lies in interval $[x_0, x_1]$, then we require less number of iterations to reach the answer.

We know that, $f(1) = (1)^3 - 5(1) - 7 = -11$

and $f(2) = (2)^3 - 5(2) - 7 = -9$

Since both $f(1)$ & $f(2)$ are negative root doesnot lie between 1 and 2.

$$f(2.5) = (2.5)^3 - 5(2.5) - 7 = -3.875$$

and $f(3) = (3)^3 - 5(3) - 7 = 5$

Since $f(2.5) . f(3) < 0$, root lies between 2.5 and 3. Take $x_0 = 2.5$ and $x_1 = 3$ as initial approximations for secant method.

Iteration No. 1 Take n = 1 in equation 3.2.11 to get x_2. Then we have,

$$x_2 = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_1) = 3 - \frac{3 - 2.5}{(5) - (-3.875)}(5)$$

$$= 2.718310$$

Iteration No. 2 Take n = 2 in equation 3.2.11 we have next approximation for root as,

$$x_3 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2)$$

$$= 2.718310 - \frac{2.718310 - 3}{(-0.505391) - (5)}(-0.505391) = 2.744169$$

Iteration No. 3 Take n = 3 in equation 3.2.11 we have next approximation for root as,

$$x_4 = x_3 - \frac{x_3 - x_2}{f(x_3) - f(x_2)} f(x_3)$$

$$= 2.744169 - \frac{2.744169 - 2.718310}{(-0.055984) - (-0.505391)} (-0.055984)$$

$$= 2.747390$$

Iteration No. 4 Take n = 4 in equation 3.2.11 then next approximation of root can be obtained as,

$$x_5 = x_4 - \frac{x_4 - x_3}{f(x_4) - f(x_3)} f(x_4)$$

$$= 2.747390 - \frac{2.747390 - 2.744169}{(0.000769) - (-0.055984)} (0.000769)$$

$= 2.747346$ (This value we had obtained in 8^{th} iteration in last method)

Now let's stop iterations here and compare x_4 & x_5 obtained in two last successive iterations.

Iteration No. 3 : 4^{th} approximation to root = x_4 = **2.747390**

Iteration No. 4 : 5^{th} approximation to root = x_5 = **2.747346**

Thus `3' digits after decimal point are repeating in two successive intervals. Hence,

Root = 2.747346 correct upto 3 decimal places.

Important Remark :

We require `8' iterations in first method and only `four' iterations in second method to obtain the same answer. Therefore in secant method, it is better to select $[x_0, x_1]$ interval such that at least one root of f(x) lies between x_0 & x_1.

3.2.2 Algorithm and Flowchart

We have discussed secant method and one illustrative example in last subsection. Now we will prepare a computer algorithm and its flowchart. For this algorithm, we have following assumptions.

Assumptions :

1) The function f(x) is predefined.
2) Correct values of initial approximations x_0 and x_1 are entered in which one root of f(x) lies.

Step 1 : *Read interval $[x_0, x_1]$ in which root of f(x) lies and read number of iterations to be performed.*

Step 2 : *Calculate*

$$f(x_0) = f(x)|_{x=x_0}$$
$$f(x_1) = f(x)|_{x=x_1}$$

Step 3 : Calculate next approximation to the root (x_2) as

$$x_2 = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_1)$$

Step 4 : $x_0 \leftarrow x_1$ *and* $x_1 \leftarrow x_2$ *for next cycle*

Step 5 : Repeat steps 2 to step 4 until last given iteration. OR if number of iterations are not given; repeat step 2 to step 4 until x_2 *repeats upto given number of significant digits after decimal point.*

Step 6 : Approximate value of the root is equal to x_2 *of last iteration.*

Step 7 : Display approximate value of root and stop.

Flowchart : Based on the algorithm above we can develop the flowchart for computer program. The flowchart is shown in Fig. 3.2.2.

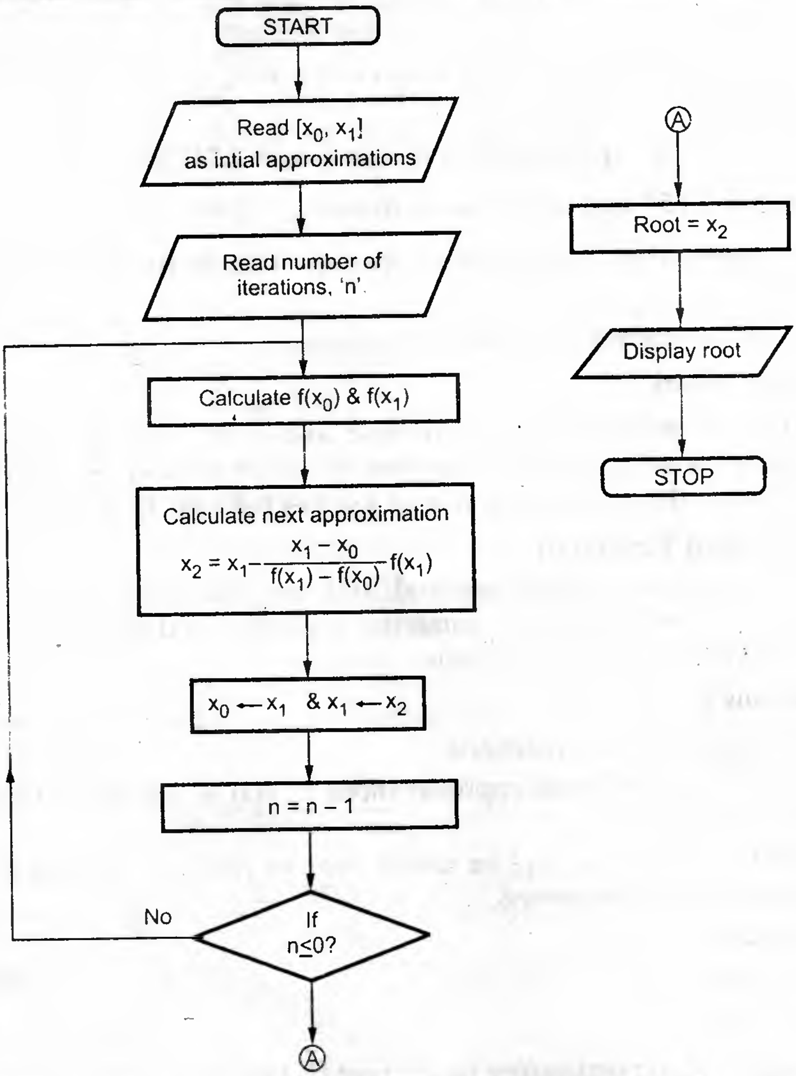


Fig. 3.2.2 Flowchart of secant method

3.2.3 Logic Development and C Program

We have solved an illustrative example of secant method in example 3.2.1. Function f(x) considered in example is $f(x) = x^3 - 5x - 7$. Here we will use the same function in our `C' program for secant method. The program first includes `stdio.h`, `math.h` and `stdlib.h` header files. Then in function `main ()`, a subroutine function `fx` is declared. This function calculates value of f(x). It takes value of x. Then different variables are declared. The approximations to the root x_0 , x_1 & x_2 are defined as double floating type numbers. See the program code listed below.

```
/* Download this program from www.vtubooks.com                              */
/* file name : secant.cpp                                                   */

/*--------------- SECANT METHOD TO FIND ROOT OF AN EQUATION  -----------*/

/*      THE EXPRESSION FOR AN EQUATION IS DEFINED IN function fx
        YOU CAN WRITE DIFFERENT EQUATION IN function fx.
        HERE,
                  f(x) = x*x*x - 5*x - 7

        INPUTS :  1) Initial interval [x0,x1] in which root is to
                     be found.
                  2) Number of iterations for given interval and
                     permissible error.

        OUTPUTS : Value of the root in given interval.              */

/*---------------Program starts here ------------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>

void main()
{
      double fx ( double x);        /* DECLARATION OF  FUNCTION  */

      double x0,x1,x2,f0,f1,f2,err;
      int n,i;

      clrscr();
      printf("\n          SECANT METHOD TO FIND ROOT OF AN EQUATION");

      printf("\n\n            f(x) = x*x*x - 5*x - 7");

      printf("\n\nEnter an interval [x0,x1] in "
             "which root is to be found");

      printf("\nx0 = ");
      scanf("%lf",&x0);       /* INTERVAL [x0,x1] IS TO BE ENTERED HERE     */
      printf("x1 = ");
      scanf("%lf",&x1);

      printf("\nEnter the number of iterations = ");
      scanf("%d",&n);
      printf("\npress any key for display of iterations...\n");
      getch();
      i = 0;

      while(n-- > 0)
      {
         f0 = fx(x0);            /*     CALCULATE f(x) AT x = x0          */
         f1 = fx(x1);            /*     CALCULATE f(x) AT x = x1          */
         x2 = x1  - ((x1 - x0)/(f1 - f0)) * f1;
                               /*     CALCULATION OF NEXT APPROXIMATION     */
         i++;
```

```
        printf("\n%d     x[%d] = %lf      x[%d] = %lf",i,i-1,x0,i,x1);
        printf("\n     f[%d] = %lf     f[%d] = %lf",i-1,f0,i,f1);
        printf("\n                               x[%d] = %lf",i+1,x2);

        x0 = x1;
        x1 = x2;

        getch();
    }

    printf("\n\nThe value of root is = %20.15lf",x2);        /*  ROOT   */
}
/*---------- FUNCTION PROCEDURE TO CALCULATE VALUE OF EQUATION --------*/

double fx ( double x)
{
  double f;
  f = x*x*x - 5*x - 7;          /*     FUNCTION f(x)   */
  return(f);
}
/*-------------------------- END OF PROGRAM -----------------------------*/
```

The program then asks for initial approximations x_0 & x_1. After these values it asks for number of iteration `n' to be performed. The `while` loop then keeps on evaluating approximations of root until number of iterations are over.

In the `while` loop,

`f0 = fx(x0)`

This statement calculates f(x) at x = x_0. Similarly `f1 = fx(x1)` statement calculates f(x) at x = x_1. In both of these statements we have used function subroutine `fx`. Then,

`x2 = x1-((x1 - x0)/(f1 - f0))*f1;`

This statement calculates next approximation x_2 to the root.

To make the while loop recursive we make `x1` as `x0` and `x2` as `x1` for next iteration and again we calculate new value of `x2`.

After getting out of `while` loop, the program prints value of root. The `printf` statement program prints value of root. The `printf` statement uses 15 significant digits after decimal point of printing.

The function procedure `fx` is listed after the program..

The statement `fx = x * x * x - 5 * x - 7`

Calculates the value of function $f(x) = x^3 - 5x - 7$ at supplied value of x.

Testing and running the program :

Compile and make `EXE' file of the above program. Run the EXE file of the program. Program will first display the name of the method and function $f(x) = x^3 - 5x - 7$. Then it displays.

```
Enter an interval [x0, x1] in which root is to be found
x0 =
```
Here enter 2.5 and press `Enter' key.

Then it displays,

`x1 =` Here enter 3 and press `Enter' key.

Then program displays,

`Enter the number of iterations =`

Here enter number of iterations you want let's say `4'.

Then you have to press any key to see values of x_2 and $f(x_2)$ in each iteration. After last iteration, the program displays value of root (i.e. x_2 of last iteration with around 15 significant digits after decimal point. Complete display of results is shown below

```
--------------------------- Results ---------------------------------

           SECANT METHOD TO FIND ROOT OF AN EQUATION

             f(x) = x*x*x - 5*x - 7

Enter an interval [x0,x1] in which root is to be found
x0 = 2.5
x1 = 3.0

Enter the number of iterations = 4

press any key for display of iterations...

1      x[0] = 2.500000       x[1] = 3.000000
       f[0] = -3.875000      f[1] = 5.000000
                                    x[2] = 2.718310
2      x[1] = 3.000000       x[2] = 2.718310
       f[1] = 5.000000      f[2] = -0.505391
                                    x[3] = 2.744169
3      x[2] = 2.718310       x[3] = 2.744169
       f[2] = -0.505391      f[3] = -0.055984
                                    x[4] = 2.747390
4      x[3] = 2.744169       x[4] = 2.747390
       f[3] = -0.055984      f[4] = 0.000769
                                    x[5] = 2.747346

The value of root is =    2.747346475533402
---------------------------------------------------------------------
```

How to use the same program for other f(x) :

Let's see how to use the program of secant method discussed here to find root of

$$f(x) = \cos - xe^x = 0$$

In the source code of the program there is a function subroutine called `fx`. It is written as `double fx (double x)` there. In this routine there is one statement.

```
f = x * x * x - 5 * x - 7
```

This statement is written for $f(x) = x^3 - 5x - 7$. Replace this statement by following statement of $f(x) = \cos x - xe^x$,

```
f = cos(x) - x * exp(x)
```

Compile and make EXE file with this statement.

Run the EXE program. Enter initial approximation as,

$$x_0 = 0 \quad \text{and} \quad x_1 = 1$$

Enter the number of iterations you want. At the 5^{th} iteration you will get,

$$\text{Root} = 0.517747$$

3.2.4 Solved Examples

Ex. 3.2.2 : *Use secant method to determine the root of following equation*

$$f(x) = cosx - xe^x = 0$$

Find the root correct upto `3' places of decimal point.

Sol. : Let's take initial approximations as $x_0 = 1$ and $x_1 = 1$.

$$f(0) = cos(0) - 0e^0 = 1$$

and $$f(1) = \cos 1 - e = -2.1779795$$

$\therefore$ root lies between [0, 1]

In the following iterations we will use following recursive equation,

$$x_{n+1} = \frac{x_n - x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n) \quad \text{(from equation 3.2.10)}$$

Iteration No. 1 $$x_2 = x_1 - \frac{x_1 - x_0}{f(x_1 - f(x_0)} f(x_1)$$

$$= 1 - \frac{1-0}{(-2.17798) - (1)} (-2.17798)$$

$$= 0.314665$$

Iteration No. 2 $$x_3 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2)$$

$$= 0.314665 - \frac{0.314665 - 1}{(0.519871) - (-2.17798)} (0.519871)$$

$$= 0.446728$$

Iteration No. 3 $$x_4 = x_3 - \frac{x_3 - x_2}{f(x_3) - f(x_2)} f(x_3)$$

$$= 0.446728 - \frac{0.446728 - 0.314665}{0.203545 - 0.519871} (0.203545)$$

$$= 0.531706$$

Iteration No. 4 $$x_5 = x_4 - \frac{x_4 - x_3}{f(x_4) - f(x_3)} f(x_4)$$

$$= 0.531706 - \frac{0.531706 - 0.446728}{(-0.042931) - (0.203545)} (-0.042931)$$

$$= 0.516904$$

Iteration No. 5 $$x_6 = x_5 - \frac{x_5 - x_4}{f(x_5) - f(x_4)} f(x_5)$$

$$= 0.516904 - \frac{0.516904 - 0.531706}{(0.002593) - (-0.042931)} (0.002593)$$

$$= 0.517747$$

Iteration No. 6 $$x_7 = x_6 - \frac{x_6 - x_5}{f(x_6) - f(x_5)} f(x_6)$$

$$= 0.517747 - \frac{0.517747 - 0.516904}{(0.00003) - (0.002593)} (0.00003)$$

$$= 0.517757$$

Here we have,

6^{th} Approximation to the root = **0.517747**

and 7^{th} Approximation to the root = **0.517757**

Thus approximate value of root repeats upto 4 significant digits after decimal point. Hence the roots obtained above are correct upto '4' decimal places.

$\therefore$ Ans = 0.517757 correct to '4' decimal points.

Ex. 3.2.3 : *Using secant method find out the square root of 25 correct upto '3' decimal places.*

Sol. : Here we know that square root of '25' is '5'. Using numerical methods also we can obtain this square root.

Let's assume that square root of 25 is 'x'. In the equation form we can write this statement as follows.

$$x = \text{square root of } 25$$

or $$x = \sqrt{25}$$

Taking square of both the sides,

$$x^2 = 25 \quad \text{or} \quad x^2 - 25 = 0$$

Thus function f(x) becomes,

$$f(x) = x^2 - 25 = 0$$

We can find root of the above equation f(x), using secant method. We have to find value of x (i.e. root of f(x)) such that f(x) = 0 is satisfied. That is nothing but square root of 25.

Let's take initial approximations as $x_0 = 3$ & $x_1 = 6$

$\therefore$ $$f(x_0) = f(3) = -16$$

and $$f(x_1) = f(6) = 11$$

$\therefore$ root lies between x_0 & x_1

Using equation 3.2.10 we can obtain approximations recursively for secant method, i.e.

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

Iteration No. 1

$$x_2 = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_1)$$

$$= 6 - \frac{6-3}{11-(-16)} \quad (6)$$

$$= 4.777778$$

Iteration No. 2

$$x_3 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2)$$

$$= 4.777778 - \frac{4.777778 - 6}{(-2.17284) - (11)} \quad (-2.17284)$$

$$= 4.979381$$

Iteration No. 3

$$x_4 = x_3 - \frac{x_3 - x_2}{f(x_3) - f(x_2)} f(x_3)$$

$$= 4.979381 - \frac{4.979381 - 4.777778}{(-0.20576) - (-2.17284)} \quad (-0.20576)$$

$$= 5.000470$$

Iteration No. 4

$$x_5 = x_4 - \frac{x_4 - x_3}{f(x_4) - f(x_3)} f(x_4)$$

$$= 5.000470 - \frac{5.000470 - 4.979381}{0.004696 - (-0.20576)} \quad (0.004696)$$

$$= 4.999999$$

Iteration No. 5

$$x_6 = x_5 - \frac{x_5 - x_4}{f(x_5) - f(x_4)} f(x_5)$$

$$= 4.999999 - \frac{4.999999 - 5.000470}{(-0.00001) - 0.004696} \quad (-0.00001)$$

$$= 5.000000$$

Iteration No. 6

$$x_7 = x_6 - \frac{x_6 - x_5}{f(x_6) - f(x_5)} f(x_6)$$

$$= 5.000000 - \frac{5.000000 - 4.999999}{0 - (-0.00001)} (0)$$

$$= 5.000000$$

Since we are getting same approximation to the root in 5^{th} & 6^{th} iteration,

Root of f(x) = 5

$\because$ Root of f(x) = square root of (25) = x

$\therefore$ x = square root of 25 = root of f(x) = 5

Ex. 3.2.4 : *Use secant method to find root of,*

$f(x) = x \log_{10}(x) - 1.9 = 0$ at the end of 3^{rd} iteration.

Sol. : Here first we have to establish initial approximations x_0 & x_1.

$$f(0) = \infty$$
$$f(1) = -1.9$$
$$f(2) = -1.29794$$
$$f(3) = -0.4686362$$
$$f(4) = 0.5082399$$

Since f(3).f(4) < 0, root lies in [3, 4]. Let $x_0 = 3$ & $x_1 = 4$

Using the recursive relation of equation 3.2.10 of secant method, we can obtain successive approximations to the root.

Equation 3.2.10 is given as,

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

Iteration No. 1 $\quad x_2 = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_1)$

$$= 4 - \frac{4-3}{0.50824 - (-0.468636)} (0.50824)$$

$$= 3.479729$$

Iteration No. 2 $\quad x_3 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2)$

$$= 3.479729 - \frac{3.479729 - 4}{(-0.015568) - (-0.50824)} (-0.015568)$$

$$= 3.495193$$

Iteration No. 3 $\quad x_4 = x_3 - \frac{x_3 - x_2}{f(x_3) - f(x_2)} f(x_3)$

$$= 3.495193 - \frac{3.495193 - 3.479729}{(-0.000464) - (-0.015568)} (-0.000464)$$

$= 3.495667$

Thus approximate value of root at the end of 3^{rd} iteration is,

$$\text{Root} = 3.495667 \qquad \text{(Ans.)}$$

Also, if we compare x_3 & x_4, we can say that the answer repeats upto 3 significant digits after decimal point.

Ex. 3.2.5 : *Using secant method find the root of equation $x^3 - 3x^2 + x + 1 = 0$.*

[Dec - 2003, 8 Marks]

Sol. : Here f(0) = 1 and f(2) = – 1. Since f(0) · f(2) < 0, root lies between [0, 2]

Following table lists the calculations

Table 3.2.1 Solution of $x^3 - 3x^2 + x + 1 = 0$

Iteration no	x_n	$f(x_n)$	x_{n+1}	$f(x_{n+1})$	x_{n+2}
1	$x_0 = 0$	1	$x_1 = 2$	– 1	$x_2 = 1$
2	$x_1 = 2$	–1	$x_2 = 1$	0	$x_3 = 1$

Here the root is repeating and f(1) = 0. Hence x = 1 is the required root.

Exercise

1. Find a positive root of equation $x^3 - 2x^2 + 3x - 4 = 0$ at the end of 5^{th} iteration using secant method. **[Ans. : 1.6506290518]**

   [Hint : Check f(x) only for positive values of `x'. Take $x_0 = 1$ & $\mathbf{x}_1 = 2$. Then use equation 3.2.10 to obtain approximations to the root]

2. Obtain the cube root of 50 using secant method. **[Ans. : 3.684028]**

   [Hint : Here cube root of 50 means we can write, $x = (50)^{\frac{1}{3}}$

   Taking 3^{rd} power on both sides, $x^3 = 50$ or $f(x) = x^3 - 50 = 0$

   Take $\mathbf{x}_0 = 2$ & $\mathbf{x}_1 = 4$ as initial approximations.]

3. Using secant method, obtain the smallest root of (whether +ve or –ve)
   $f(x) = x^3 - 6.37x^2 + 6.48x + 7.11 = 0$
   at the end of 4^{th} iteration. **[Ans. : – 0.645732]**

   [Hint : To solve such problem use following steps.

   (i) find f(0) = 7.11

   (ii) find f(–1) = – 6.74

   ∵ f(0).f(1) < 0, root lies between [0, –1]

   (iii) find f(–2) = –39.33

   (iv) find f(–3) = – 96.66

   (v) find f(–4) = – 184.73

Thus the function value is increasing on negative values of `x'. Now let's locate roots on positive side.

(i) find (0) = 7.11

(ii) find f(1) = 8.22

(iii) find f(2) = 2.59

(iv) find f(3) = –3.78

∵ f(2).f(3) < 0, root lies between [2, 3]

(v) find f(4) = – 4.89

(vi) find f(5) = 5.26

∵ f(4).f(5) < 0, root lies between [4, 5]

The order of the equation for f(x) is `3', therefore this equation will have `3' roots.

Here we have located `3' intervals [0, –1], [2, 3] & [4, 5] in which one root of f(x) lies.

Since smallest root is asked, it will be in interval [0, –1].

∴ take $x_0 = 0$ & $x_1 = -1$

If highest root is asked, them take $x_0 = 4$ & $x_1 = 5$ for interval [4, 5].

Root in [0, –1] will be negative. If lowest positive root is asked, then it will lie in interval [2, 3].

∴ Take $x_0 = 2$ and $x_1 = 3$. **[Ans. : Smallest positive root = 2.37**

Ans. : Highest root (+ve or –ve) = 4.645671]

4. Obtain the root of $e^x - 4x = 0$ correct upto `4' decimal places. [Use secant method] **[Ans. : 0.357403]**

[Hint : Here take $x_0 = 0$ and $x_1 = 1$ as an initial approximation].

University Question

1. Using secant method find the root of equation $x^3 - 3x^2 + x + 1 = 0$.

[Dec - 2003, 8 Marks]

3.3 Regula Falsi Method or Method of False Position

3.3.1 Method

In the secant method we approximate a function f(x) by a straight line or chord. The point at which this line crosses x-axis is called the approximate root of that function. If x_0 & x_1 are initial approximations to the root, then the approximated line to function f(x) passes through $[x_0, f(x_0)]$ and $[x_1, f(x_1)]$.

The point at which this line crosses x axis is called next approximation to the root. This crossing point is $[x_2, 0]$ or x_2 on x –axis. We have derived an expression for x_2 and is given by equation 3.2.9. It is reproduced here for convenience i.e.,

$$x_2 = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_1) \quad \text{... (3.3.1)}$$

From this x_2, we obtain next approximation to the root, i.e. x_3 and so on.

The recursive relation given by equation 3.2.10 is the generalized equation for getting next approximation from previous two approximations. It is reproduced here for convenience,

i.e.,
$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n) \quad \text{... (3.3.2)}$$

If the approximations are selected such that, $f(x_n) \cdot f(x_{n-1}) < 0$, then secant method is called Regula Falsi method.

Fig. 3.3.1 shows the graphical interpretation of Regula Falsi method.

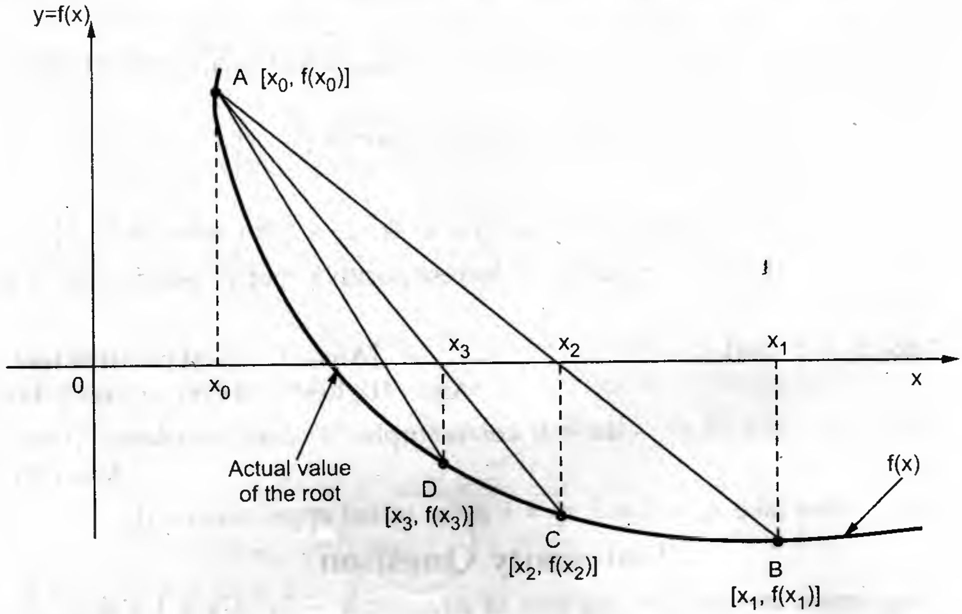


Fig. 3.3.1 Graphical interpretation of Regula Falsi method

Thus in Fig. 3.3.1 x_0 & x_1 are the initial approximations to the root and they are selected such that, $f(x_0) \cdot f(x_1) < 0$ or negative. Therefore curve AB of f(x) is approximated by a straight line AB. Therefore next approximation to the root is the point where line AB crosses x axis. This point is x_2. Here we have three approximations to the root i.e. x_0, x_1 and x_2. From Fig. 3.3.1 it is clear that,

$f(x_0) \cdot f(x_2) < 0$ or negative

Hence we select x_0 & x_2 as an initial approximations for x_3. Because of this, the method converges fast.

Now let's discuss an illustrative example of Regula Falsi method.

Ex. 3.3.1 : *Find the root of $f(x) = e^x - 4x = 0$ using Regula Falsi method correct upto `3' decimal places.*

Sol. :

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

$$f(0) = f(x)|_{x=0} = 1$$

$$f(1) = f(x)|_{x=1} = e^1 - 4 = -1.281718$$

Here since $f(0) \cdot f(1) < 0$,

Let's take $x_0 = 0$ and $x_1 = 1$ as initial approximations. Then using recursive relations given by equation 3.3.2 we can obtain next approximations to the root. Equation 3.3.2 states that next approximation is given as,

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

Iteration No. 1

$$x_2 = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_1)$$

$$= 1 - \frac{1-0}{(-1.281718) - 1} (-1.281718)$$

$$= 0.438266$$

$$f(x_2) = f(x)|_{x = x_2 = 0.438266} = e^{(0.438266)} - 4 \times (0.438266) = -0.203047$$

Here since $f(x_0) \cdot f(x_2) < 0$ root lies in the interval [0, 0.438266].

Hence we take initial approximations for second iterations as,

$$x_1 = 0 \quad \& \quad x_2 = 0.438266$$

Iteration No. 2

With initial approximations of $x_1 = 0$ and $x_2 = 0.438266$ from the previous iteration, we find next approximation `x_3' to the root as

$$x_3 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2)$$

$$= 0.438266 - \frac{0.438266 - 0}{(-0.203047) - 1} (-0.203047)$$

$$= 0.364297$$

$$f(x_3) = f(x)|_{x = x_3 = 0.364297}$$

$$= e^{(0.364297)} - 4 \times (0.364297)$$

$$= -0.017686$$

Here we have $x_1 = 0$ $\quad x_2 = 0.438266$ $\quad x_3 = 0.364297$

and $f(x_1) = 1$ $\quad f(x_2) = 0.203047$ $\quad f(x_3) = -0.017686$

Since $f(x_1) \cdot f(x_3) < 0$ root lies in the interval $[x_1, x_3]$ = [0, 0.364297]

Hence initial approximations for next iteration should be $x_2 = 0$ & $x_3 = 0.364297$

Iteration No. 3

With the initial approximations of $x_2 = 0$ and $x_3 = 0.364297$ from previous iteration, we find next approximation `x_4' to the root as,

$$x_4 = x_3 - \frac{x_3 - x_2}{f(x_3) - f(x_2)} f(x_3)$$

$$f(x_3) = f(x)|_{x = x_3 = 0.364297}$$
$$= -0.017686$$

and $$f(x_2) = f(x)|_{x = x_2 = 0} = 1$$

$\therefore$ $$x_4 = 0.364297 - \frac{0.364297 - 0}{(-0.017686) - 1} (-0.017686)$$
$$= 0.357966$$

and $$f(x_4) = f(x)|_{x = x_4 = 0.357966}$$
$$= e^{(0.357966)} - 4 \times (0.357966)$$
$$= -0.001447$$

Here we have, $x_2 = 0$ $\quad x_3 = 0.364297$ $\quad x_4 = 0.357966$

and $f(x_2) = 1$ $\quad f(x_3) = -0.017686$ $\quad f(x_4) = -0.001447$

Since $f(x_2) \cdot f(x_4) < 0$, root lies in the interval $[x_2, x_4] = [0, 0.357966]$

Hence initial approximation for the next iteration should be,

$x_3 = 0$ and $x_4 = 0.357966$

Iteration No. 4

With the initial approximations of $x_3 = 0$ and $x_4 = 0.357966$ from previous iteration, we find next approximation `x_5' to the root as,

$$x_5 = x_4 - \frac{x_4 - x_3}{f(x_4) - f(x_3)} f(x_4)$$

$$f(x_4) = f(x)|_{x = x_4 = 0.357966}$$
$$= -0.001447$$

and $$f(x_3) = f(x)|_{x = x_3 = 0} = 1$$

$\therefore$ $$x_5 = 0.357966 - \frac{0.357966 - 0}{(-0.001447) - 1} (-0.001447)$$
$$= 0.357449$$

4^{th} approximation to the root = x_4 = **0.357966**

5^{th} approximation to the root = x_5 = **0.357449**

Since three decimal digits repeat in successive approximations, the 5^{th} approximation to the root is correct upto 3 decimal places.

$\therefore$ Ans. = 0.357449 correct to 3 significant digits after decimal point.

3.3.2 Algorithm and Flowchart

From the discussion and an illustrative example in last subsection we will now prepare the algorithm and flowchart for the regula falsi method.

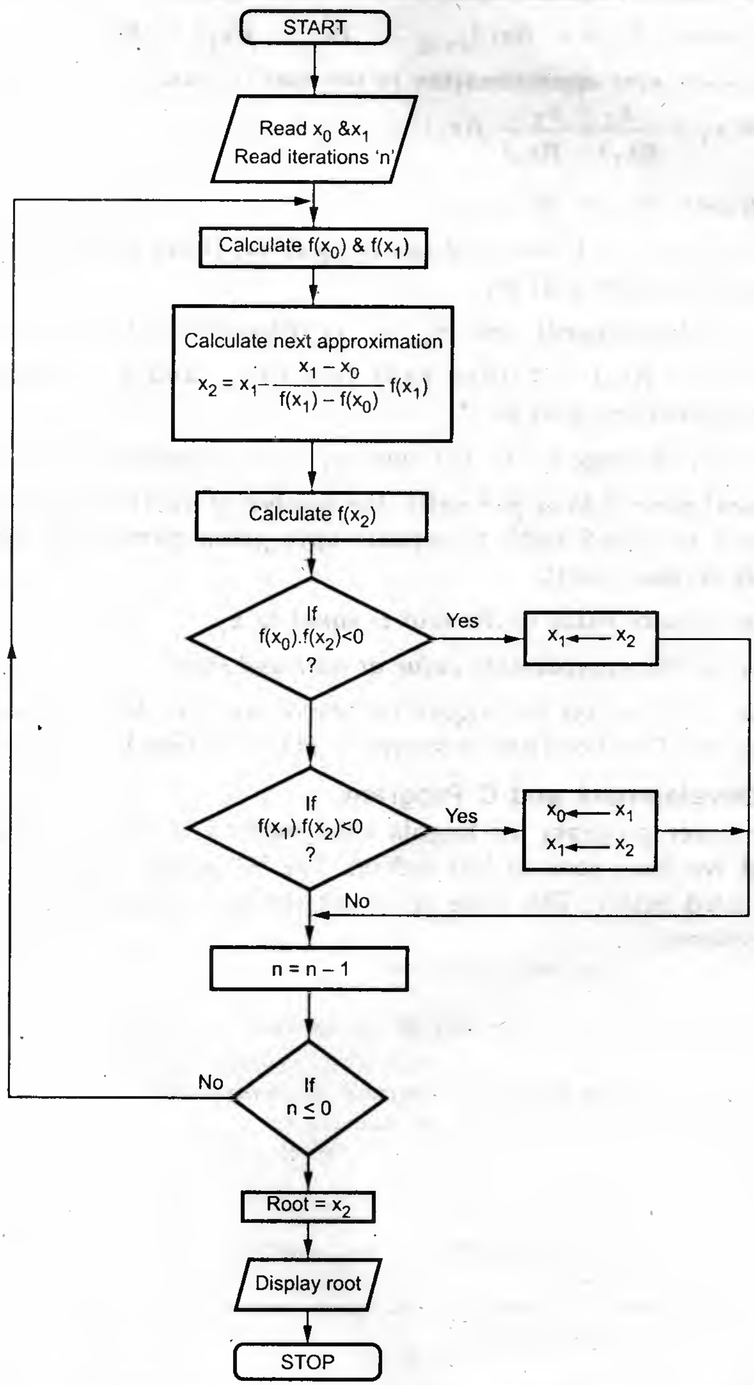


Fig. 3.3.2 Flowchart for Regula Falsi method

Assumptions : 1) The function f(x) is predefined 2) Correct values of initial approximations x_0 & x_1 are entered in which one root of f(x) lies.

Step 1 : *Read interval* $[x_0 , x_1]$ *in which root of f(x) lies and read number of iterations to be performed.*

Step 2 : *Calculate :* $f(x_0) = f(x)\,|_{x=x_0}$ & $f(x_1) = f(x)\,|_{x=x_1}$

Step 3 : *Calculate next approximation to the root* (x_2) *as,*

$$x_2 = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)}\, f(x_1)$$

Step 4 : *Calculate* $f(x_2) = f(x)|_{x=x_2}$

Step 5 : *If* $f(x_0) \cdot f(x_2) < 0$ *then root lies in* x_0 & x_2. *Hence new initial approximations will be,*

x_0 *(unchanged) and* $x_1 \leftarrow x_2$ *(Changed* x_2 *to* x_1*)*

If $f(x_1) \cdot f(x_2) < 0$, *then root lies in* x_1 *and* x_2. *Hence new initial approximations will be,*

$x_0 \leftarrow x_1$ *(Change* x_1 *to* x_0*) and* $x_1 \leftarrow x_2$ *(Change* x_2 *to* x_1*)*

Step 6 : *Repeat steps-2 to step-5 until the number of iterations are over OR Repeat step-2 to step-5 until* x_2 *repeats upto given number of significant digits after decimal point.*

Step 7 : *Approximate value of the root is equal to* x_2.

Step 8 : *Display the approximate value of root and stop.*

Flowchart : Based on the algorithm above we can develop the flowchart for computer program. This flowchart is shown in Fig. 3.3.2 (See Fig. on previous page).

3.3.3 Logic Development and C Program

The computer program for Regula Falsi method is exactly similar to that for secant method we have seen in last section. The 'C' source code of `reg_fls.cpp` program file listed below. This code is almost similar to that of secant method with minor modifications.

```
/* Download this program from www.vtubooks.com                    */
/* file name : reg_fls.cpp                                          */
/*——— REGULA FALSI METHOD TO FIND ROOT OF AN EQUATION  ———*/


/*    THE EXPRESSION FOR AN EQUATION IS DEFINED IN function fx
      YOU CAN WRITE DIFFERENT EQUATION IN function fx.
      HERE,
                f(x)  = exp(x)  - 4*x

      INPUTS :   1)  Initial interval [x0,x1] in which root is to
                     be found.
                 2)  Number of iterations for given interval and
                    permissible error.
```

```
        OUTPUTS : Value of the root in given interval.                    */

/*---------------------------------------- PROGRAM ----------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>

void main()
{
        double fx ( double x);          /* DECLARATION OF  FUNCTION   */

        double x0,x1,x2,f0,f1,f2,err;
        int n,i;

        clrscr();

        printf("\n          REGULA FALSI METHOD TO FIND ROOT OF AN EQUATION");

        printf("\n\n                    f(x) = exp(x) - 4*x");

        printf("\n\nEnter an interval [x0,x1] in "
               "which root is to be found");

        printf("\nx0 = ");
        scanf("%lf",&x0);          /* INTERVAL [x0,x1] IS TO BE ENTERED HERE     */
        printf("x1 = ");
        scanf("%lf",&x1);

        printf("\nEnter the number of iterations = ");
        scanf("%d",&n);
        printf("\npress any key for display of iterations...\n");

        getch();
        i = 0;

        while(n-- > 0)
        {
        f0 = fx(x0);              /*    CALCULATE f(x) AT x = x0          */
        f1 = fx(x1);              /*    CALCULATE f(x) AT x = x1          */
        x2 = x1 - ((x1 - x0)/(f1 - f0)) * f1;
                /*    CALCULATION OF NEXT APPROXIMATION    */
        f2 = fx(x2); i++;
        printf("\n\n%d  x[%d] = %lf                x[%d] = %lf            x[%d] = %lf\n"
               "    f[%d] = %lf                f[%d] = %lf            f[%d] = %lf\n"
           ,i,i,x1,i-1,x0,i+1,x2,i,f1,i-1,f0,i+1,f2);
        if((f0 * f2) < 0) x1 = x2;
               if((f1 * f2) < 0)
               {
```

```
            x0 = x1;
            x1 = x2;
         }
         printf("\nNew interval : x[%d] = %lf      x[%d] = %lf",i-1,x0,i+1,x1);
         getch();
     }

     printf("\n\nThe value of root is = %20.15lf",x2);     /*   ROOT    */
}
/*—— FUNCTION PROCEDURE TO CALCULATE VALUE OF EQUATION ——*/

double fx ( double x)
{
   double f;
   f = exp(x) - 4*x;             /*      FUNCTION f(x)    */
   return(f);
}
/*-------------------------------- END OF PROGRAM ----------------------------------*/
```

The program has include statements at the start. In function `main( )`, function subroutine `fx` is defined first. Then some variables are declared.

The program asks for initial approximations `x0`, `x1` and number of iterations `'n'`. The program then enters a `while` loop. In this loop it calculates next approximations `x2`. The program then calculates $f(x_2)$ by passing `x2` to subroutine `fx`. Subroutine `fx` returns the value of f(x) at x = x_2. Then the first `if` statement checks whether root lies between x_0 & x_2. The second `if` statement checks whether root lies between x_1 & x_2. Depending on the location of the root, values of `x0` and `x1` are updated for next iteration.

After coming out of the `while` loop, program displays the last obtained value of `x2` as approximate root.

The subroutine function `fx` is listed after the program. The statement,

```
f = exp(x) - 4 * x
```

in subroutine `fx` is written for,

$$f(x) = e^x - 4x$$

Thus the program evaluates the root of $e^x - 4x = 0$. If you want to use the same program for some other function, Then change this statement in function subroutine fx only.

How to Run this program :

Compile and make EXE file of this source code. Then run this executable file.

The program first displays the names of the method. Then it displays,

`Enter the interval [x0, x1] in which root is to be found`

`x0 =` Here you have to enter value of x_0. Enter `0' and press `Enter' key.

The program then displays,

x1 = Here you have to enter value of x_1. Here enter `1' and press `Enter' key.

Then the program displays,

Enter the number of iterations = Here enter the number of iterations you want (now n = 6) and press `Enter' key. Then go on pressing any key and the program displays the results of every iteration step by step.

Here is the complete display of results shown below.

```
-------------------------- Results ---------------------------------

       REGULA FALSI METHOD TO FIND ROOT OF AN EQUATION

            f(x) = exp(x) - 4*x

Enter an interval [x0,x1] in which root is to be found
x0 = 0
x1 = 1
Enter the number of iterations = 6
press any key for display of iterations...

1     x[1] = 1.000000            x[0] = 0.000000         x[2] = 0.438266
      f[1] = -1.281718           f[0] = 1.000000         f[2] = -0.203047

New interval : x[0] = 0.000000     x[2] = 0.438266

2     x[2] = 0.438266            x[1] = 0.000000         x[3] = 0.364297
      f[2] = -0.203047           f[1] = 1.000000         f[3] = -0.017686

New interval : x[1] = 0.000000     x[3] = 0.364297

3     x[3] = 0.364297            x[2] = 0.000000         x[4] = 0.357966
      f[3] = -0.017686           f[2] = 1.000000         f[4] = -0.001447

New interval : x[2] = 0.000000     x[4] = 0.357966

4     x[4] = 0.357966            x[3] = 0.000000         x[5] = 0.357449
      f[4] = -0.001447           f[3] = 1.000000         f[5] = -0.000118

New interval : x[3] = 0.000000     x[5] = 0.357449

5     x[5] = 0.357449            x[4] = 0.000000         x[6] = 0.357407
      f[5] = -0.000118           f[4] = 1.000000         f[6] = -0.000010

New interval : x[4] = 0.000000     x[6] = 0.357407

6     x[6] = 0.357407            x[5] = 0.000000         x[7] = 0.357403
      f[6] = -0.000010           f[5] = 1.000000         f[7] = -0.000001

New interval : x[5] = 0.000000     x[7] = 0.357403

The value of root is =    0.357403259182090
---------------------------------------------------------------------
```

Compare the results obtained by computer with the results obtained in illustrative example 3.3.1.

3.3.4 Solved Examples

Ex. 3.3.2 : *Find the root of* $f(x) = 2x - \log_{10} x - 7 = 0$ *which should be correct upto `3' decimal places.*

Sol. : Here, since interval is not given, we have to first establish an interval where root of f(x) lies.

Let $f(0) = 2(0) - \log_{10}(0) - 7 = \infty$

$f(1) = 2(1) - \log_{10}(1) - 7 = -5$

$f(2) = 2(2) - \log_{10}(2) - 7 = -3.301030$

$f(3) = 2(3) - \log_{10}(3) - 7 = -1.477121$

$f(4) = 2(4) - \log_{10}(4) - 7 = 0.397940$

Here f(3).f(4) < 0. Hence root lies between 3 & 4.

$\therefore$ Take $x_0 = 3$ and $x_1 = 4$ as initial approximations.

Iteration No. 1

$$x_2 = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_1)$$

$$= 4 - \frac{4-3}{0.39794 - (-1.477121)} (0.39794)$$

$$= 3.787772$$

$$f(x_2) = f(3.787772) = -0.002839$$

Here $f(x_2).f(x_1) < 0$

$\therefore$ Root lies in [4, 3.787772]

Iteration No. 2

Take $x_1 = 4$ and $x_2 = 3.787772$ as a initial approximation now

$$x_3 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2)$$

$$= 3.787772 - \frac{3.787772 - 4}{(-0.002839) - (0.39794)} (-0.002839)$$

$$= 3.789276$$

$$f(x_3) = f(3.789276) = -0.000005$$

Here $f(x_1).f(x_3) < 0$

$\therefore$ Root lies in [4, 3.789276]

Iteration No. 3

From the previous iteration, take $x_2 = 4$ and $x_3 = 3.789276$ as initial approximation.

Then

$$x_4 = x_3 - \frac{x_3 - x_2}{f(x_3) - f(x_2)} f(x_3)$$

$$= 3.789276 - \frac{3.789276 - 4}{(-0.000005) - (0.397940)} (-0.000005)$$

$$= 3.789278$$

Here we observe that,

$$x_3 = \mathbf{3.789276}$$
$$x_4 = \mathbf{3.789278}$$

Since `5' digits are repeating after decimal point, the root (x_4) is given as,

Ans. : Approximate value of root = x_4 = 3.789278

Ex. 3.3.3 : *Use false position method to determine the roots of the equation $e^{-x} - x = 0$. Two initial guess values being $x_0 = 0$ and $x_1 = 1$. Compute the root at the end of 4^{th} iteration.*

Sol. : Take $x_0 = 0$ and $x_1 = 1$

Iteration No. 1 $$x_2 = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_1)$$
$$= 1 - \frac{1-0}{(-0.632121 - 1)}(-0.632121) = 0.6127$$
$$f(x_2) = e^{-0.6127} - 0.6127 = -0.070814$$

Here $f(0) \, . \, f(0.6127) < 0$ Hence root lies in [0, 0.6127].

Iteration No. 2 $$x_3 = 0.6127 - \frac{0.6127 - 0}{f(0.6127) - f(0)} f(0.6127)$$
$$= 0.572181$$
$$f(0.572181) = -0.007888$$

$\because$ $f(0) \, . \, f(0.572181) < 0$; Hence root lies in [0, 0.572181]

Iteration No. 3 $$x_4 = 0.572181 - \frac{0.572181 - 0}{f(0.572181) - f(0)} f(0.572181)$$
$$= 0.567703$$
$$f(0.567703) = -0.000877$$

$\because$ $f(0) \, . \, f(0.567703) < 0$; root lies in [0, 0.567703]

Iteration No. 4 $$x_5 = 0.567703 - \frac{0.567703 - 0}{f(0.567703) - f(0)} f(0.567703)$$
$$= 0.567203$$

Thus the approximate value of root at 4^{th} iteration is = 0.567203.

Ex. 3.3.4 : *Determine the drag coefficient c (in kg/s) needed for a parachutist of mass m = 68.1 kg to have a velocity of 40 m/s after free falling for time t = 10 sec.*
Note : *The acceleration due to gravity is 9.8 m/s². The velocity of parachutist is given by*

$$v(t) = \frac{gm}{c}[1 - e^{-(c/m)t}]$$

Use method of false position :
*[**Hint :** Drag coefficient lies in range 12 kg/s to 16 kg/s]*

Sol. : The velocity of the parachutist is given as,

$$v(t) = \frac{gm}{c}[1 - e^{-(c/m)t}]$$

The different quantities are given as :

mass $\quad m = 68.1$ kg

time $\quad t = 10$ sec

Gravitational acceleration

$$g = 9.8 \text{ m/s}^2$$

Velocity $\quad v(t) = 40$ m/s

Putting the above values in equation for velocity, we get,

$$40 = \frac{9.8 \times 68.1}{c}[1 - e^{(c/68.1) \times 10}]$$

Simplifying the above equation,

$$c = 16.6845 - 16.6845\, e^{-0.1468428\, c}$$

The above equation should be solved for 'c'. Hence we will write above equation as,

$$f(c) = c + 16.6845\, e^{-0.1468428\, c} - 16.6845 = 0$$

Let us replace 'c' by 'x' for simplicity of notations i.e.

$$f(x) = x + 16.6845\, e^{-0.1468428\, x} - 16.6845 = 0$$

The value c lies in the range of 12 to 16 kg/s. Hence let $c_0 = x_0 = 12$ and $c_1 = x_1 = 16$ be initital two vales of c. Now let us obtain value of x(i.e. c) by iterative equation of regula falsi method. The iterations are given next.

Iteration No. 1

$$x_2 = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_1)$$

$$= 16 - \frac{16 - 12}{0.907507 - (-1.820078)}(0.907507) = 14.669142$$

and $\quad f(x_2) = f(14.669142) = -0.079748$

Here $\quad f(x_1) \cdot f(x_2) < 0 \quad$ Hence root lies in

$$[x_1, x_2] = [16, 14.669142]$$

Iteration No. 2

Take $x_2 = 14.669142$ and $x_1 = 16$ as initial approximation

$$\therefore \quad x_3 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2)$$

$$= 14.669142 - \frac{14.669142 - 16}{-0.079748 - 0.907507}(-0.079748)$$

$$= 14.776646$$

and $f(x_3) = f(14.776646) = -0.002560$

Here $f(1) \cdot f(3) < 0$ Hence root lies in

$[x_1, x_3] = [16, 14.776646]$

Iteration No. 3

Take $x_3 = 14.776646$ and $x_2 = 16$ as initial approximation

Then,
$$x_4 = x_3 - \frac{x_3 - x_2}{f(x_3) - f(x_2)} f(x_3)$$
$$= 14.776646 - \frac{14.776646 - 16}{-0.002560 - 0.907507}(-0.002560)$$
$$= 14.780088$$

and $f(x_4) = f(14.780088)$
$= -0.000081$

Here $f(x_2) \cdot f(x_4) < 0$ Hence root lies in

$[x_2, x_4] = [16, 14.780088]$

Iteration No. 4

Take $x_4 = 14.780088$ and $x_3 = 16$ as initial approximation

Then,
$$x_5 = x_4 - \frac{x_4 - x_3}{f(x_4) - f(x_3)} f(x_4)$$
$$= 14.780088 - \frac{14.780088 - 16}{-0.000081 - 0.907507}(-0.000081)$$
$$= 14.780197$$

and $f(x_5) = f(14.780197) = -0.000003$

Here $f(x_3) \cdot f(x_5) < 0$ Hence root lies in

$[x_3, x_5] = [16, 14.780197]$

Iteration No. 5

Take $x_5 = 14.780197$ and $x_4 = 16$ as initial approximation

Then,
$$x_6 = x_5 - \frac{x_5 - x_4}{f(x_5) - f(x_4)} f(x_5)$$
$$= 14.780197 - \frac{14.780197 - 16}{-0.000003 - 0.907507}(-0.000003)$$
$$= 14.780201$$

The value of 'x' is repeating upto 3rd place after decimal point. Hence Drag coefficient c = 14.780201 kg/s is correct upto 3 decimal places for given data.

Ex. 3.3.5 *Find the real root of the equation $x^3 - 2x - 5 = 0$ using regula falsi method correct upto 3 decimal places.* **[Dec.-2001 10 Marks]**

Sol. : The given function is,

$$f(x) = x^3 - 2x - 5 = 0$$

Here the interval of the root is not given. Hence let us first establish an interval.

$$\text{Let} \quad f(0) = (0)^3 - 2(0) - 5 = -5$$

$$f(1) = (1)^3 - 2(1) - 5 = -6$$

$$f(2) = (2)^3 - 2(2) - 5 = -1$$

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

Here $f(2) \cdot f(3) < 0$, hence root lies between 2 and 3. Therefore let us take $x_0 = 2$ and $x_1 = 3$ as initial approximations.

Iteration No. 1 : The next approximation in regula falsi method is given by (equation 3.3.1),

$$x_2 = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_1)$$

$$= 3 - \frac{3-2}{16-(-1)} 16$$

$$= 2.05882353$$

$$f(x_2) = f(2.05882353) = -0.39079992$$

Here $\quad f(x_2) \cdot f(x_1) < 0$. Hence root lies in [3, 2.05882353]

Iteration No. 2 : Now take $x_1 = 3$ and $x_2 = 2.05882353$ as an initial approximation. Hence,

$$x_3 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2)$$

$$= 2.05882353 - \frac{2.05882353 - 3}{-0.39079992 - 16}(-0.399079992)$$

$$= 2.08126366$$

$$f(x_3) = f(2.08126366) = -0.14720406$$

Since $\quad f(x_1) \cdot f(x_3) < 0$, root lies in the interval [3, 2.08126366]

Iteration No. 3 : Now take $x_2 = 3$ and $x_3 = 2.08126366$ as an initial approximation. Then we can write,

$$x_4 = x_3 - \frac{x_3 - x_2}{f(x_3) - f(x_2)} f(x_3)$$

$$= 2.08126366 - \frac{2.08126366 - 3}{-0.14720406 - 16}(-0.14720406)$$

$$= 2.08963921$$

$$f(x_4) = f(2.08963921) = -0.05467650$$

Since $f(x_2) \cdot f(x_4) < 0$, root lies in [3, 2.08963921]

Iteration No. 4 : Now take $x_3 = 3$ and $x_4 = 2.08963921$ as an initial approximation. Then,

$$x_5 = x_4 - \frac{x_4 - x_3}{f(x_4) - f(x_3)} f(x_4)$$

$$= 2.08963921 - \frac{2.08963921 - 3}{-0.05467650 - 16}(-0.05467650)$$

$$= 2.09273957$$

$$f(x_5) = f(2.09273957) = -0.02020287$$

Since $f(x_3) \cdot f(x_5) < 0$, root lies in the interval [3, 2.09273957]

Similarly next iterations can be calculated. Following table lists the values :

Table 3.3.1 : Results of example 3.3.5

Iteration No.	root	f(x)
5.	2.0938837	f(2.09388371) = – 0.00745051
6.	2.09430545	f(2.09430545) = – 0.00274567
7.	2.09446085	f(2.09446085) – 0.00101157
8.	2.09451809	f(2.09451809) = – 0.00037265

In the above table observe that the value of root repeats upto 3 decimal places in 7^{th} and 8^{th} iterations. Hence the root correct upto 3 decimal places is 2.09451809.

Ex. 3.3.6 *With the help of a neat flowchart, explain the regula falsi method for evaluating the root of a function of the form $f(x) = 0$. Comment on the rate of convergence. Obtain the smallest positive root for the equation, $x^3 - x - 4 = 0$*

[Dec.-98, 8 Marks ; Dec.-2000, 8 Marks]

Obtain the smallest positive root of the following equation correct to four significant digits.

$$x^3 - x - 4 = 0$$

Use (i) Regula falsi method and

(ii) Secant method **[May-97 12 Marks]**

Sol. : The regula falsi method is discussed at the beginning of this section. The rate of covergence of this method is discussed in section 3.5.

i) **To establish an interval for smallest positive root**

We know that, $f(x) = x^3 - x - 4 = 0$

Let $f(0) = (0)^3 - 0 - 4 = -4$

$f(0.1) = (0.1)^3 - 0.1 - 4 = -4.099$

$$f(0.5) = (0.5)^3 - 0.5 - 4 = -4.375$$

$$f(1) = (1)^3 - 1 - 4 = -4$$

$$f(2) = (2)^3 - 2 - 4 = 2$$

Since $f(1) \cdot f(2) < 0$, root lies in [1, 2]. Hence,

Let $x_0 = 1$ and $x_1 = 2$

ii) **Solution using regula falsi method**

Iteration No. 1 : Next approximation in regula falsi is given by equation 3.3.1 as,

$$x_2 = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_1)$$

$$= 2 - \frac{2-1}{f(2) - f(1)} f(2)$$

$$= 2 - \frac{1}{2-(-4)}(2)$$

$$= 1.666667$$

$$\therefore \quad f(x_2) = f(1.666667)$$

$$= -1.037037$$

$$\text{And } f(x_1) = f(2) = 2$$

$$f(x_0) = f(1) = -4$$

$\because \quad f(x_2) \cdot f(x_1) < 0$, root lies in [2, 1.666667]

Iteration No. 2 : Now take $x_1 = 2$ and $x_2 = 1.666667$. Then x_3 will be calculated as,

$$x_3 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2)$$

$$= 1.666667 - \frac{1.666667 - 2}{-1.037037 - 2}(-1.037037)$$

$$= 1.780488$$

$$\therefore \quad f(x_3) = f(1.780488) = -0.136098$$

$$\text{And } f(x_2) = f(1.666667) = -1.037037$$

$$f(x_1) = f(2) = 2$$

$\because \quad f(x_3) \cdot f(x_1) < 0$, root lies in [2, 1.780488].

Iteration No. 3 : Take $x_2 = 2$ and $x_3 = 1.780488$. Then x_4 will be,

$$x_4 = x_3 - \frac{x_3 - x_2}{f(x_3) - f(x_2)} f(x_3)$$

$$= 1.780488 - \frac{1.780488 - 2}{-0.136098 - 2}(-0.136098)$$

$$= 1.794474$$

$$\therefore \quad f(x_4) = f(1.794474) = -0.016025$$
$$\text{And } f(x_3) = f(1.780488) = -0.136098$$
$$f(x_2) = f(2) = 2$$

$\because \quad f(x_4)\cdot f(x_2) < 0$, root lies in [2, 1.794474].

Iteration No. 4 : Take $x_3 = 2$ and $x_4 = 1.794474$. Then x_5 will be,

$$x_5 = x_4 - \frac{x_4 - x_3}{f(x_4) - f(x_3)} f(x_4)$$
$$= 1.794474 - \frac{1.794474 - 2}{-0.016025 - 2}(-0.016025)$$
$$= 1.796107$$
$$\therefore \quad f(x_5) = f(1.796107) = -0.001862$$
$$\text{and } f(x_4) = f(1.794474) = -0.016025$$
$$f(x_3) = f(2) = 2$$

$\because \quad f(x_5)\cdot f(x_3) < 0$, root lies in [2, 1.796107].

Iteration No. 5 : Take $x_4 = 2$ and $x_5 = 1.796107$. Then x_6 will be,

$$x_6 = x_5 - \frac{x_5 - x_4}{f(x_5) - f(x_4)} f(x_5)$$
$$= 1.796107 - \frac{1.796107 - 2}{-0.001862 - 2}(-0.001862)$$
$$= 1.796297$$
$$\therefore \quad f(x_6) = f(1.796297) = -0.0002161$$
$$\text{And } f(x_5) = f(1.796107) = -0.001862$$
$$f(x_4) = f(2) = 2$$

$\because \quad f(1.796297)\cdot f(2) < 0$, root lies in [2, 1.796297].

Iteration No. 6 : Take $x_5 = 2$ and $x_6 = 1.796297$, then x_7 will be,

$$x_7 = x_6 - \frac{x_6 - x_5}{f(x_6) - f(x_5)} f(x_6)$$
$$= 1.796297 - \frac{1.796297 - 2}{-0.0002161 - 2}(-0.0002161)$$
$$= 1.796319$$
$$\therefore \quad f(x_7) = f(1.796319) = -251346 \times 10^{-5}$$
$$\text{And } f(x_6) = f(1.796297) = -0.0002161$$
$$f(x_5) = f(2) = 2$$

Since $\quad f(2)\cdot f(1.796319) < 0$, root lies in [2, 1.796319].

Iteration No. 7 : Take $x_6 = 2$ and $x_7 = 1.796319$. Then x_8 will be,

$$x_8 = x_7 - \frac{x_7 - x_6}{f(x_7) - f(x_6)} f(x_7)$$

$$= 1.796319 - \frac{1.796319 - 2}{-2.51346 \times 10^{-5} - 2}(-2.51346 \times 10^{-5})$$

$$= 1.7963216$$

Here note that,

6^{th} iteration root = **1.796319**

7^{th} iteration root = **1.7963216**

Since 4 digits after the decimal point repeat, root = 1.7963216 is correct upto 4 significant digits.

iii) **Solution using secant method**

We know that the root lies in [1, 2]. Hence let $x_0 = 1$ and $x_1 = 2$. Then next approximation in secant method is given by equation 3.2.10 as,

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

All iterations can be performed with the help of this equation.

Iteration No. 1

$$x_2 = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_1)$$

$$= 2 - \frac{2-1}{2-(-4)}(2)$$

$$= 1.666667$$

$$\therefore \quad f(x_2) = f(1.666667) = -1.037037$$

$$\text{And } f(x_1) = f(2) = 2$$

Iteration No. 2

$$x_3 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2)$$

$$= 1.666667 - \frac{1.666667 - 2}{-1.037037 - 2}(-1.037037)$$

$$= 1.780488$$

$$\therefore \quad f(x_3) = f(1.780488) = -0.136098$$

$$\text{and } f(x_2) = f(1.666667) = -1.037037$$

Iteration No. 3

$$x_4 = x_3 - \frac{x_3 - x_2}{f(x_3) - f(x_2)} f(x_3)$$

$$= 1.780488 - \frac{1.780488 - 1.666667}{-0.136098 - (-1.037037)}(-0.136098)$$

$$= 1.797682$$

$$\therefore \quad f(x_4) = f(1.797682) = -0.011815$$

$$\text{And } f(x_3) = f(1.780488) = -0.136098$$

Iteration No. 4

$$x_5 = x_4 - \frac{x_4 - x_3}{f(x_4) - f(x_3)} f(x_4)$$

$$= 1.797682 - \frac{1.797682 - 1.780488}{0.011815 - (-0.136098)}(0.011815)$$

$$= 1.796308$$

$$\therefore \quad f(x_5) = f(1.796308) = -0.000117$$

$$\text{And } f(x_4) = f(1.797682) = 0.011815$$

Iteration No. 5

$$x_6 = x_5 - \frac{x_5 - x_4}{f(x_5) - f(x_4)} f(x_5)$$

$$= 1.796308 - \frac{1.796308 - 1.797682}{-0.000117 - 0.011815}(-0.000117)$$

$$= 1.796322$$

Here note that,

4^{th} iteration root = **1.796308**

5^{th} iteration root = **1.796322**

Four digits after the decimal point repeat in 4^{th} and 5^{th} iterations.

Hence root = 1.796322 is correct upto four significant figures.

iv) **Results and comments**

Table 3.3.2 lists the results and comments.

Table 3.3.2 Solutions obtained by regula falsi and secant methods

No.	Name of the method	Number of iterations	Value of the root	Comments
1.	Regula falsi	7	1.7963216	For this example, secant method converges fast compared to regula falsi method.
2.	Secant	5	1.796322	

Ex. 3.3.7 *The current in a particular circuit is given by $I^3 - 5I - 7 = 0$. Find the value of current using regula falsi and biscection methods.* **[May-2001, 8 Marks]**

OR

The current flowing through some circuit it is given by equation $I^3 - 5I - 7 = 0$. Use any bracketing method to find the value of I correct upto three decimal places.

[Dec.-2002, 6 Marks]

Sol. : To establish an interval in which root lies.

Here the given function is.

$$f(I) = I^3 - 5I - 7 = 0$$

Let $f(0) = (0)^3 - 5(0) - 7 = -7$

$f(1) = (1)^3 - 5(1) - 7 = -11$

$f(2) = (2)^3 - 5(2) - 7 = -9$

$f(3) = (3)^3 - 5(3) - 7 = 5$

$\because$ $f(2) \cdot f(3) < 0$, root lies in [2, 3]

To obtain the current using bisection method

Iteration No. 1 : The root lies in [2, 3]. Hence

Let a = 2 and b = 3

$$\therefore \quad c = \frac{a+b}{2} = \frac{2+3}{2} = 2.5$$

$$\therefore \quad f(c) = f(2.5) = -3.875$$

And $f(a) = f(2) = -9$

$f(b) = f(3) = 5$

Since $f(2.5) \cdot f(3) < 0$, root lies in [2.5, 3]

Iteration No. 2 : The new interval is : a = 2.5, b = 3.

Hence, $$c = \frac{a+b}{2} = \frac{2.5+3}{2} = 2.75$$

$$\therefore \quad f(c) = f(2.75) = 0.046875$$

And $f(a) = f(2.5) = -3.875$

$f(b) = f(3) = 5$

Since $f(2.5) \cdot f(2.75) < 0$, root lies in [2.5, 2.75]

Iteration No. 3 : New interval is : a = 2.5, b = 2.75.

Hence, $$c = \frac{a+b}{2} = \frac{2.5+2.75}{2} = 2.625$$

$$\therefore \quad f(c) = f(2.625) = -2.0371094$$

And $f(a) = f(2.5) = -3.875$

$f(b) = f(2.75) = 0.046875$

Since $f(2.625) \cdot f(2.75) < 0$, root lies in [2.625, 2.75]

Iteration No. 4 : New interval is : a = 2.625, b = 2.75.

Hence, $$c = \frac{a+b}{2} = \frac{2.625+2.75}{2} = 2.6875$$

$$\therefore \quad f(c) = f(2.6875) = -1.0266113$$

And $f(a) = f(2.625) = -2.0371094$

$f(b) = f(2.75) = 0.046875$

Since $f(2.6875) \cdot f(2.75) < 0$, root lies in [2.6875, 2.75]

Iteration No. 5 : New interval is : a = 2.6875, b = 2.75.

Hence, $$c = \frac{a+b}{2} = \frac{2.6875 + 2.75}{2} = 2.71875$$

$\therefore$ $f(c) = f(2.71875) = -0.4978332$

And $f(a) = f(2.6875) = -1.0266113$

$f(b) = f(2.75) = 0.046875$

Since $f(2.71875) \cdot f(2.75) < 0$, root lies in [2.71875, 2.75]

Iteration No. 6 : New interval is a = 2.71875, b = 2.75.

Hence, $$c = \frac{a+b}{2} = \frac{2.71875 + 2.75}{2} = 2.734375$$

$\therefore$ $f(c) = f(2.734375) = -0.2274818$

And $f(a) = f(2.71875) = -0.4978332$

$f(b) = f(2.75) = 0.046875$

Since $f(2.734375) \cdot f(2.75) < 0$, root lies in [2.734375, 2.75]

Iteration No. 7 : New interval is : a = 2.734375, b = 2.75. Hence,

$$c = \frac{a+b}{2} = \frac{2.734375 + 2.75}{2} = 2.7421875$$

$\therefore$ $f(c) = f(2.7421875) = -0.0908055$

And $f(a) = f(2.734375) = -0.2274818$

$f(b) = f(2.75) = 0.046875$

Since $f(2.7421875) \cdot f(2.75) < 0$, root lies in [2.7421875, 2.75]

Iteration No. 8 : New interval is : a = 2.7421875, b = 2.75.

Hence, $$c = \frac{a+b}{2} = \frac{2.7421875 + 2.75}{2} = 2.7460938$$

$\therefore$ $f(c) = f(2.7460938) = -0.020912$

And $f(a) = f(2.7421875) = -0.2274818$

$f(b) = f(2.75) = 0.046875$

Since $f(2.7460938) \cdot f(2.75) < 0$, root lies in [2.7460938, 2.75]

Iteration No. 9 : New interval is : a = 2.7460938, b = 2.75.

Hence, $$c = \frac{a+b}{2} = \frac{2.7460938 + 2.75}{2} = 2.7480469$$

$\therefore$ $f(c) = f(2.7480469) = -0.012361$

And $f(a) = f(2.7460938) = -0.0220912$

$f(b) = f(2.75) = 0.046875$

Since $f(2.7480469) \cdot f(2.7460938) < 0$, root lies in [2.7460938, 2.7480469]

Iteration No. 10 : New interval is : a = 2.7460938, b = 2.7480469.

Hence, $$c = \frac{a+b}{2} = \frac{2.7460938 + 2.7480469}{2} = 2.7470704$$

$$\therefore \quad f(c) = f(2.7470704) = -0.0048724$$

And $$f(a) = f(2.7460938) = -0.0220912$$

$$f(b) = f(2.7480469) = 0.012361$$

Since $f(2.7470704) \cdot f(2.7480469) < 0$, root lies in [2.7470704, 2.7480469]

Iteration No. 11 : New interval is : a = 2.7470704, b = 2.7480469.

Hence, $$c = \frac{a+b}{2} = \frac{2.7470704 + 2.7480469}{2} = 2.7475587$$

Here note that,

10^{th} iteration $\Rightarrow$ c = **2.7470704**

11^{th} iteration $\Rightarrow$ c = **2.7475587**

Three digits repeat after the decimal point. Hence value of the root is correct to three decimal places. Hence current in the circuit is,

I = 2.7475587 A correct upto 3 decimal places.

To obtain current using regula falsi method

We know that root lies in [2, 3]. Hence let us take $x_0 = 2$ and $x_1 = 3$ as initial approximations.

Iteration No. 1 : Next root in regula falsi method is given as,

$$x_2 = x_1 - \frac{x_1 - x_o}{f(x_1) - f(x_0)} f(x_1)$$

$$= 3 - \frac{3-2}{5-(-9)} \times 5$$

$$= 2.6428571$$

$$\therefore \quad f(x_2) = f(2.6428571) = -1.754738$$

and $$f(x_1) = f(3) = 5$$

$$f(x_0) = f(2) = -9$$

Since $f(2.6428571) \cdot f(3) < 0$, root lies in [3, 2.6428571]

Iteration No. 2 : Take $x_1 = 3$ and $x_2 = 2.6428571$

Then, $$x_3 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2)$$

$$= 2.6428571 - \frac{2.6428571 - 3}{-1.754738 - 5} \times (-1.754738)$$

$$= 2.735635$$

$$\therefore \quad f(x_3) = f(2.735635) = -0.205502$$

$$\text{And } f(x_2) = f(2.6428571) = -1.754738$$
$$f(x_1) = f(3) = 5$$

Since $f(2.735635) \cdot f(3) < 0$, root lies in [3, 2.735635]

Iteration No. 3 : Take $x_2 = 3$ and $x_3 = 2.735635$

Then,
$$x_4 = x_3 - \frac{x_3 - x_2}{f(x_3) - f(x_2)} f(x_3)$$
$$= 2.735635 - \frac{2.735635 - 3}{-0.205502 - 5}(-0.205502)$$
$$= 2.746072$$
$$\therefore \; f(x_4) = f(2.746072) = -0.022478$$
$$\text{And } f(x_3) = f(2.735635) = -0.205502$$
$$f(x_2) = f(3) = 5$$

Since $f(2.746072) \cdot f(3) < 0$, root lies in [3, 2.746072].

Iteration No. 4 : Take $x_3 = 3$ and $x_4 = 2.746072$

Then,
$$x_5 = x_4 - \frac{x_4 - x_3}{f(x_4) - f(x_3)} f(x_4)$$
$$= 2.746072 - \frac{2.746072 - 3}{-0.022478 - 5}(-0.022478)$$
$$= 2.747208$$
$$\therefore \; f(x_5) = f(2.747208) = -0.002440$$
$$\text{And } f(x_4) = f(2.746072) = -0.022478$$
$$f(x_3) = f(3) = 5$$

Since $f(2.747208) \cdot f(3) < 0$, root lies in [3, 2.747208]

Iteration No. 5 : Take $x_4 = 3$ and $x_5 = 2.747208$

Then,
$$x_6 = x_5 - \frac{x_5 - x_4}{f(x_5) - f(x_4)} f(x_4)$$
$$= 2.747028 - \frac{2.747028 - 3}{-0.002440 - 5}(-0.002440)$$
$$= 2.74733154$$

Here note that,

4^{th} iteration $\Rightarrow$ root = **2.747208**

5^{th} iteration $\Rightarrow$ root = **2.74733154**

Three digits repeat after the decimal point. Hence the value of the root is correct to three decimal places.

Comparison of Bisection method and Regula falsi method

The following Table 3.3.3 shows the results.

Table 3.3.3 Results of bisection and regula falsi method.

Sr. No.	Name of the method	Value of current	Number of iterations required	Comments
1.	Bisection	2.7475587	11	Regula falsi method converges fast compared to bisection method. Hence regula falsi method requires less number of iteration.
2.	Regula falsi	2.7473315	5	

Ex. 3.3.8 *Locate the root of $f(x) = x^{10} - 1$ between $x = 0$ and 1.3 using bisection method and false position. Comment on which method is preferable.*

[May-96, 8 Marks; May-98, 8 Marks; Dec.-99, 8 Marks; May-2004, 10 Marks]

Sol. : i) **To obtain root using bisection method.**

Iteration No. 1 : The root lies in [0, 1.3]. Hence take a = 0 and b = 1.3

Then, $$c = \frac{a+b}{2} = \frac{0+1.3}{2} = 0.65$$

$$\therefore \quad f(c) = f(0.65) = (0.65)^{10} - 1 = -0.9865372$$

And $$f(a) = f(0) = (0)^{10} - 1 = -1$$

$$f(b) = f(1.3) = (1.3)^{10} - 1 = 12.785849$$

Since $f(0.65) \cdot f(1.3) < 0$, root lies in [0.65, 1.3]

Iteration No. 2 : New interval is : a = 0.65 and b = 1.3

Hence, $$c = \frac{a+b}{2} = \frac{0.65+1.3}{2}$$

$$= 0.975$$

$$\therefore \quad f(c) = f(0.975) = -0.223670$$

And $$f(a) = f(0.65) = -0.986537$$

$$f(b) = f(1.3) = 12.785849$$

Since $f(0.975) \cdot f(1.3) < 0$, root lies in [0.975, 1.3]

Iteration No. 3 : New interval is : a = 0.975 and b = 1.3

Hence, $$c = \frac{a+b}{2} = \frac{0.975+1.3}{2}$$

$$= 1.1375$$

$$\therefore \quad f(c) = f(1.1375) = 2.626720$$

And $$f(a) = f(0.975) = -0.223670$$

$$f(b) = f(1.3) = 12.785849$$

Since $f(0.975) \cdot f(1.1375) < 0$, root lies in [0.975, 1.1375]

Similarly next iterations of bisection method can be performed. Table 3.3.4 lists the results of these iterations.

Table 3.3.4 Solution using bisection method

Iteration No.	New Interval [a, b]	$c = \frac{a+b}{2}$	f (a)	f (b)	f (c)
4.	a = 0.975 b = 1.1375	1.056250	– 0.223670	2.626720	0.728491
5.	a = 0.975 b = 1.056250	1.015625	– 0.223670	0.728491	0.167707
6.	a = 0.975 b = 1.015625	0.995313	– 0.223670	0.167707	– 0.045898
7.	a = 0.995313 b = 1.015625	1.005469	– 0.045898	0.167707	0.056053
8.	a = 0.995313 b = 1.005469	1.000391	– 0.045898	0.056053	0.003913
9.	a = 0.995313 b = 1.000391	0.997852	– 0.045898	0.003913	– 0.021278
10.	a = 0.997852 b = 1.000391	0.999121	– 0.021278	0.003913	– 0.008754
11.	a = 0.999121 b = 1.000391	0.999756	–	–	–

In the above table observe that,

10^{th} iteration : C = **0.999121**

11^{th} iteration : C = **0.999756**

Three digits repeat after the decimal point. Hence value of root current to 3 decimal places is 0.999756.

ii) **To obtain root using regula falsi method.**

We know that the root lies in [0, 1.3]. Hence let us take $x_0 = 0$ and $x_1 = 1.3$ as initial approximations.

$$\text{And} \quad f(x_0) = f(0) = -1$$

$$f(x_1) = f(1.3) = 12.785849$$

Iteration No. 1 : Next root in regula falsi method is given as,

$$x_2 = x_1 - \frac{x_1}{f(x_1) - f(x_0)} f(x_1)$$

$$= 1.3 - \frac{1.3 - 0}{12.785849 - (-1)} 12.785859$$

$$= 0.0942995$$

$\therefore$ $\quad f(x_2) = f(0.0942995) = -1$

And $\quad f(x_1) = f(1.3) = 12.785849$

$\quad f(x_0) = f(0) = -1$

Since $f(0.0942995) \cdot f(1.3) < 0$, root lies in [1.3, 0.0942995]

Iteration No. 2 : Take $x_1 = 1.3$ and $x_2 = 0.0942995$

Then $$x_3 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2)$$

$$= 0.0942995 - \frac{0.0942995 - 1.3}{-1 - 12.785849}(-1)$$

$$= 0.181759$$

$\therefore$ $\quad f(x_3) = f(0.181759) = -1$

And $\quad f(x_2) = f(0.0942995) = -1$

$\quad f(x_1) = f(1.3) = 12.785849$

Since $f(0.181759) \cdot f(1.3) < 0$, root lies in [1.3, 0.181759]

Iteration No. 3 : Take $x_2 = 1.3$ and $x_3 = 0.181759$

Then $$x_4 = x_3 - \frac{x_3 - x_2}{f(x_3) - f(x_2)} f(x_3)$$

$$= 0.181759 - \frac{0.181759 - 1.3}{-1 - 12.785849}(-1)$$

$$= 0.262874$$

$\therefore$ $\quad f(x_4) = f(0.262874) = -0.999998$

And $\quad f(x_3) = f(0.181759) = -1$

$\quad f(x_2) = f(1.3) = 12.785849$

Since $f(0.262874) \cdot f(1.3) < 0$, root lies in [1.3, 0.262874]

Iteration No. 4 : Take $x_3 = 1.3$ and $x_4 = 0.262874$

Then $$x_5 = x_4 - \frac{x_4 - x_3}{f(x_4) - f(x_3)} f(x_4)$$

$$= 0.262874 - \frac{0.262874 - 1.3}{-0.999998 - 1}(-0.999998)$$

$$= 0.338105$$

$\therefore$ $\quad f(x_5) = f(0.338105) = -0.999980$

And $\quad f(x_4) = f(0.262874) = -0.999998$

$$f(x_3) = f(1.3) = 12.785849$$

Since $f(0.338105) \cdot f(1.3) < 0$, root lies in [1.3, – 0.338105]

Further iterations are given in the table 3.3.5.

Table 3.3.5 : Solution using regula falsi method

Iteration No.	New interval x_{n-1}, x_n	Next root x_{n+1}	$f(x_{n-1})$	$f(x_n)$	$f(x_{n+1})$
5.	$x_4 = 1.3$ $x_5 = 0.338105$	$x_6 = 0.407878$	12.785849	– 0.999980	– 0.999873
6.	$x_5 = 1.3$ $x_6 = 0.407878$	$x_7 = 0.472583$	12.785849	– 0.999873	– 0.999444
7.	$x_6 = 1.3$ $x_7 = 0.472583$	$x_8 = 0.532572$	12.785849	– 0.999444	– 0.998164
8.	$x_7 = 1.3$ $x_8 = 0.532572$	$x_9 = 0.588145$	12.785849	– 0.998164	– 0.995047
9.	$x_8 = 1.3$ $x_9 = 0.588145$	$x_{10} = 0.639544$	12.784849	– 0.995047	– 0.988553
10.	$x_9 = 1.3$ $x_{10} = 0.639544$	$x_{11} = 0.686943$	12.875849	– 0.988553	– 0.976600
11.	$x_{10} = 1.3$ $x_{11} = 0.686943$	$x_{12} = 0.730446$	12.785849	– 0.976600	– 0.956760
: :	: :	: :	: :	: :	: :
35.	$x_{34} = 1.3$ $x_{35} = 0.998817$	$x_{36} = 0.999094$	12.785849	– 0.011766	– 0.009023
36.	$x_{35} = 1.3$ $x_{36} = 0.999094$	$x_{37} = 0.999306$	--	--	--

Here note that,

35^{th} iteration $\Rightarrow$ root = **0.999094**

36^{th} iteration $\Rightarrow$ root = **0.999306**

Three significant digits after decimal point repeat in 36^{th} iteration. Hence the root = 0.999306 is correct upto 3 decimal places.

iii) **Comment on the results**

Here note that bisection method needs 11 iterations whereas regula falsi method needs 36 iterations for the same accuracy in the root. Hence regula falsi method converges slowly for this example. To overcome this problem, initial approximations should be selected very close to the root.

Important

This example is solved with the help of computer program. In exam students can calculate upto 11 iterations of regula falsi method. In 11^{th} iteration $x_{12} = 0.730446$, which is far away from the root. Hence we can say that regula falsi method needs more number of iterations compared to bisection method. And hence, regula falsi converges slowly for this example. But note that convergence of regula falsi is better in general case.

Ex. 3.3.9 *Evaluate the root for the equation*

$$f(x) = \cos x - xe^{x} = 0$$

Using (i) bisection method (ii) False position method. **[Dec.-96 10 Marks]**

Sol. : i) To establish an interval in which root lies.

We know that $f(x) = \cos x - xe^{x} = 0$

$$\text{Let} \quad f(0) = \cos(0) - 0.e^{0} = 1$$

$$f(0.5) = \cos(0.5) - 0.5\,e^{0.5} = 0.0532219$$

$$f(0.7) = \cos(0.7) - 0.7\,e^{0.7} = -0.6447847$$

Since $f(0.5) \cdot f(0.7) < 0$, root lies in [0.5, 0.7]

ii) **To obtain root using bisection method**

Iteration No. 1 : The root lies in [0.5, 0.7]. Hence take a = 0.5 and b = 0.7. Then,

$$c = \frac{a+b}{2} = \frac{0.5+0.7}{2} = 0.6$$

$$\therefore \quad f(c) = f(0.6) = -0.267936$$

$$\text{And} \quad f(a) = f(0.5) = 0.05322219$$

$$f(b) = f(0.7) = -0.6447847$$

Since $f(0.5) \cdot f(0.6) < 0$, root lies in [0.5, 0.6]

Iteration No. 2 : New interval is : a = 0.5, b = 0.6.

$$\text{Hence} \quad c = \frac{a+b}{2} = \frac{0.5+0.6}{2} = 0.55$$

$$\therefore \quad f(c) = f(0.55) = -0.100765$$

$$\text{And} \quad f(a) = f(0.5) = 0.05322219$$

$$f(b) = f(0.6) = -0.267936$$

Since $f(0.5) \cdot f(0.55) < 0$, root lies in [0.5, 0.55]

Iteration No. 3 : New interval is : a = 0.5, b = 0.55

Hence $c = \frac{a+b}{2} = \frac{0.5+0.55}{2} = 0.525$

$\therefore$ $f(c) = f(0.525) = -0.022167$

And $f(a) = f(0.5) = 0.05322219$

$f(b) = f(0.55) = -0.100765$

Since $f(0.5) \cdot f(0.525) < 0$, root lies in [0.5, 0.525]

Iteration No. 4 : New interval is : a = 0.5, b = 0.525.

Hence $c = \frac{a+b}{2} = \frac{0.5+0.525}{2} = 0.5125$

$\therefore$ $f(c) = f(0.5125) = 0.015923$

And $f(a) = f(0.5) = 0.05322219$

$f(b) = f(0.525) = -0.022167$

Since $f(0.5125) \cdot f(0.525) < 0$, root lies in [0.5125, 0.525]

Similarly remaining iterations can be performed. Following table lists the calculations of these iterations.

Table 3.3.6 : Solution using bisection method

Iteration No.	New Interval [a, b]	$c = \frac{a+b}{2}$	f(a)	f(b)	f(c)
5.	a = 0.5125 b = 0.525	0.518750	0.015923	– 0.022167	– 0.003022
6.	a = 0.5125 b = 0.518750	0.515625	0.015923	– 0.003022	0.006475
7.	a = 0.515625 b = 0.518750	0.517188	0.006475	– 0.003022	0.001733
8.	a = 0.517188 b = 0.518750	0.517969	--	--	--

From the above table note that,

7^{th} iteration $\Rightarrow$ root = **0.517188**

8^{th} iteration $\Rightarrow$ root = **0.517969**

The root repeats upto 3 significant digits after decimal point.

Hence root = 0.517969 is correct upto 3 decimal places.

iii) **To obtain root using false position method**

We know that root lies in [0.5, 0.7]. Hence let us take $x_0 = 0.5$ and $x_1 = 0.7$ as initial approximations. And

$$f(x_0) = f(0.5) = 0.053222$$

$$f(x_1) = f(0.7) = -0.644785$$

Iteration No. 1 : Next root in regula falsi method is given as,

$$x_2 = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_1)$$

$$= 0.7 - \frac{0.7 - 0.5}{-0.644785 - 0.053222}(-0.644785)$$

$$= 0.515250$$

$\therefore \quad f(x_2) = f(0.515250) = 0.007613$

And $\quad f(x_1) = f(0.7) = -0.644785$

$$f(x_0) = f(0.5) = 0.053222$$

Since $f(0.7) \cdot f(0.515250) < 0$, root lies in $[0.7, 0.515250]$

Iteration No. 2 : Take $x_1 = 0.7$ and $x_2 = 0.515250$

Then $\quad x_3 = x_2 - \dfrac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2)$

$$= 0.515250 - \frac{0.515250 - 0.7}{0.007613 - (-0.644785)}(0.007613)$$

$$= 0.517405$$

$\therefore \quad f(x_3) = f(0.517405) = 0.001070$

And $\quad f(x_2) = f(0.515250) = 0.007613$

$$f(x_1) = f(0.7) = -0.644785$$

Since $f(0.7) \cdot f(0.517405) < 0$, root lies in $[0.7, 0.517405]$

Iteration No. 3 : Take $x_2 = 0.7$ and $x_3 = 0.517405$

Then $\quad x_4 = x_3 - \dfrac{x_3 - x_2}{f(x_3) - f(x_2)} f(x_3)$

$$= 0.517405 - \frac{0.517405 - 0.7}{0.001070 - (-0.644785)}(0.001070)$$

$$= 0.517708$$

Here note that,

2^{nd} iteration $\Rightarrow$ root = **0.517405**

3^{rd} iteration $\Rightarrow$ root = **0.517708**

The root repeats upto 3 digits after decimal point. Hence root = 0.517708 is correct upto 3 decimal places.

iv) **Results and comments**

Table 3.3.7 shows the results and comments of this example.

Table 3.3.7 : Results of example 3.3.9

Sr.No.	Name of the method	Value of the root	Number of iterations required	Comments
1.	Bisection	0.517969	8	It is clear that regula falsi method converges fast compared to bisection method.
2.	False position	0.517708	3	

Exercise

1. Use Regula Falsi method to determine the root of equation $\cos x - xe^x = 0$

[Ans. : 0.515201099]

[**Hint** : Here use initial approximation as $x_0 = 0$ and $x_1 = 1$ and take 5 iterations]

2. Find the value of root of $x^3 - 3x + 4$ correct upto `3' significant digits after decimal point. Use initial approximation as $x_0 = -2$ and $x_1 = -3$.

[Ans. : -2.195708]

[**Hint** : Here the answer is obtained in 7^{th} iteration. Value of approximate root repeats upto `3' decimal places in 6^{th} and 7^{th} iteration.]

3. Find the negative root of the equation $x^4 - 1.8x^3 + 6x^2 + 18x - 13.9 = 0$ by Regula Falsi method. How many iterations are required so that your answer repeats upto two decimal places. **[Ans. : 1.91995 repeats upto two decimal places in 3^{rd} iteration]**

[**Hint** : Since –ve root is asked, try negative value of x to establish initial approximation

f(0) = –13.9

f(–1) = –23.1

f(–2) = 4.5 $\quad \because \; f(-1) \cdot f(-2) < 0$; root lies between –1 and –2

$\therefore$ Take $x_0 = -1$ and $x_1 = -2$ as initial approximations].

4. Find the value of angle `θ' when amplitude of sine wave becomes half of its peak positive value. **[Ans. : 0.5236014 radians in 3^{rd} iteration]**

[Hint : The value of sine wave lies in the range $-1 < \sin\theta < +1$ Half of the peak positive value means $\frac{1}{2}$].

$\therefore$ we can write, $\quad \sin\theta = \frac{1}{2} \quad$ or $\quad \sin\theta - \frac{1}{2} = 0$

Thus $f(\theta) = \sin\theta - \frac{1}{2}$ for our Regula Falsi method and we have to obtain value of `θ'.

Let's try,

$f(0) = -0.5$

$f(0.2) = -0.301331$

$f(0.4) = -0.110582$

$f(0.6) = 0.064642$

$\because$ $f(0.4) \cdot f(0.6) < 0$, root lies in 0.4 and 0.6 radians.

$\therefore$ Take $x_0 = 0.4$ and $x_1 = 0.6$ as initial approximation].

University Questions

1. With the help of a neat flowchart, explain the Regula Falsi method for evaluating the root of a function of the form $f(x) = 0$. Comment on the rate of convergence.

[Dec - 95, Dec - 98, May - 99, May - 2000, Dec - 2000]

2. Locate the root of $f(x) = x^{10} - 1$ between x = 0 and 1.3.

   Using - (i) bisection and (ii) false position. Comment on which method is preferable for the above equation. **[May - 96, May - 98, Dec - 99]**

3. Evaluate the root for the equation $f(x) = \cos x - xe^x = 0$.

   Using : i) bisection method and ii) false position method. **[Dec - 96]**

4. Obtain the smallest positive root of the following equation correct to four significant figures :

   $x^3 - x - 4 = 0$

   use i) Secant and ii) Regula falsi method. **[May - 97, Dec - 98]**

5. The current in a particular circuit is given by $I^3 - 5I - 7 = 0$. Find the value of the current using Regula Falsi and Bisection methods. **[May - 2001]**

6. Find the real root of the equation $f(x) = x^3 - 2x - 5 = 0$ using Regula Falsi method correct upto 3 decimal place. Also write a C program for the above. **[Dec - 2001]**

7. The current flowing through some circuit it is given by equation $I^3 - 5I - 7 = 0$. Use any bracketing method to find the value of I correct upto three decimal places.

[Dec.-2002, 6 Marks]

8. Using geometrical interpretation derive the equation for Regula-Falsi method.

[May - 2003, 6 Marks]

9. Solve $f(x) = x^{10} - 1$ between x = 0 and x = 13 using bisection and false position methods. Compare your results. **[May - 2004, 10 Marks]**

3.4 Newton Raphson Method

3.4.1 Method

Let x_0 be an approximate root of f(x) = 0. Let actual value of root x_1 is given as,

$$x_1 = x_0 + h \qquad ...(3.4.1)$$

Since x_1 is correct root we can write

$$f(x)|_{x=x_1} = f(x_1)$$
$$= 0$$

$$\therefore \quad f(x_1) = f(x_0 + h)$$
$$= 0$$

$f(x_0 + h)$ can be expanded using Taylor's series as,

$$f(x_1) = f(x_0) + hf'(x_0) + \frac{h^2}{2!} f''(x_0) + ... = 0 \qquad ... (3.4.2)$$

Since value of `h' is very very small, we can neglect all the terms above 2^{nd} term in equation 3.4.2. Thus second and higher order derivatives are neglected in equation 3.4.2. Therefore equation 3.4.2 becomes.

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

$$\therefore \quad h = -\frac{f(x_0)}{f'(x_0)} \qquad ... (3.4.3)$$

Putting this value of `h' in equation 3.4.1 we get,

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \qquad ... (3.4.4)$$

Thus x_1 is the next approximation and obtained from x_0. Similarly we can obtain x_2 from x_1 as,

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

Thus a generalized recursive relation can be written as,

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \frac{f(x_n)}{f'(x_n)} \qquad ... (3.4.5)$$

This equation is called Newton Raphson formula.

Graphical interpretation of Newton Raphson Method :

Let's say that initial approximation to the root of f(x) = 0 is x_0. And the curve f(x) is approximated by a straight line which is tangent to the curve at point A$[x_0, f(x_0)]$ as shown in Fig. 3.4.1.

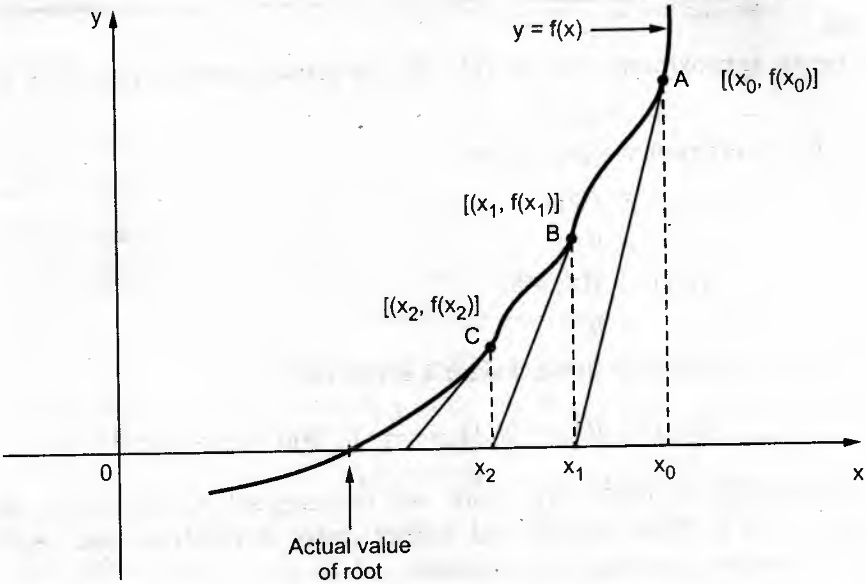


Fig. 3.4.1 Graphical interpretation of Newton Raphson method

$\therefore$ We can write equation of this tangent line at point A as,

$$y - y_0 = \text{(slope at point A)} [x - x_0] \qquad \text{... (3.4.6)}$$

Here we know that,

$$y_0 = f(x)\big|_{x = x_0} = f(x_0)$$

$$\text{Slope of line at point A} = \text{first derivative of f(x) at } x = x_0$$

$$= f'(x_0)$$

Putting these values in the equation 3.4.6 we obtain,

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

This tangent line crosses x - axis when y = 0.

And value of $x = x_1$ at this intersection. Hence,

$$0 - f(x_0) = f'(x_0) [x_1 - x_0]$$

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

or

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

Which is similar to equation 3.4.4.

Thus Newton Raphson method approximates the curve as a tangent and next approximation is the point where tangent intersects x-axis.

Observe in Fig 3.4.1 that new tangent is drawn at point B. This point is obtained on the curve at $x = x_1$ and $y = f(x_1)$. The procedure is then repeated.

Few important things to be followed using Newton Raphson method :

1) For this method, $f'(x_0)$ should not be equal to zero, otherwise the method fails.
   If $f'(x_0) = 0$, then change initial approximation x_0.
2) For better convergence of the method select x_0 such that, $f(x_0) \cdot f''(x_0)$ product will be positive.

Ex. 3.4.1 : *Find the smallest positive root of* $x^3 - 5x + 3 = 0$ *using Newton Raphson method using 4 iterations.*

Sol. : We have,

$$f(x) = x^3 - 5x + 3 = 0$$

$$f'(x) = \frac{d}{dx}[f(x)] = 3x^2 - 5$$

and

$$f''(x) = \frac{d}{dx}[f'(x)] = 6x$$

First let's establish an interval in which the smallest positive root of f(x) lies

$$f(0) = f(x)|_{x=0} = 3$$

$$f(1) = f(x)|_{x=1} = -1$$

$\because$ $f(0) \cdot f(1) < 0$, root lies between 0 and 1. From these two values we have to select one value for initial approximation. Either value can be selected, but for fast and better convergence we should select proper value of x_0.

We know that we should select a value of x_0 such that

$$f(x_0) \cdot f''(x_0) > 0 \qquad \text{... (3.4.7)}$$

Here, $f''(0) = f''(x)|_{x=0} = 6 \times 0 = 0$

and $f''(1) = f''(x)|_{x=1} = 6 \times 1 = 6$

Thus, $f(0) \cdot f''(0) = 3 \times 0 = 0$

and $f(1) \cdot f''(1) = -1 \times 6 = -6$

Thus both 0 and 1 does not satisfy the condition of equation 3.4.7. Hence let's try some other value in interval [0, 1]

$$f(0.3) = f(x)|_{x=0.3} = 1.527$$

$\because$ $f(0.3) \cdot f(1) = 1.527 \times (-1) < 0,$ root lies in [0.3, 1]

$$f''(0.3) = f''(x)|_{x=0.3} = 6 \times 0.3 = 1.8$$

$\because$ $f(0.3) \cdot f''(0.3) = 1.527 \times 1.8 > 0$, 0.3 is proper value of initial approximation

Hence take $x_0 = 0.3$

Now in successive iterations use recursive formula of Newton Raphson method given by equation 3.4.5 i.e.,

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

Iteration No. 1 $\quad x_1 = x_0 - \dfrac{f(x_0)}{f'(x_0)}$

$$= 0.3 - \frac{f(0.3)}{f'(0.3)} = 0.3 - \frac{1.527}{(-4.73)}$$

$$= 0.622833$$

Iteration No. 2 $\quad x_2 = x_1 - \dfrac{f(x_1)}{f'(x_1)} = 0.622833 - \dfrac{0.127445}{(-3.836237)}$

$$= 0.656054$$

Iteration No. 3 $\quad x_3 = x_2 - \dfrac{f(x_2)}{f'(x_2)}$

$$= 0.656054 - \frac{0.002099}{(-3.708778)} = 0.656620$$

Iteration No. 4 $\quad x_4 = x_3 - \dfrac{f(x_3)}{f'(x_3)}$

$$= 0.656620 - \frac{0.000001}{(-3.706549)} = 0.65662043$$

Thus we have, $\quad x_3 = 0.656620$

and $\quad x_4 = 0.65662043$

The value of root repeats upto 6 digits after decimal point, hence it is correct upto 6 decimal places. We will get the same value of approximate root using $x_0 = 0$ and $x_0 = 1$ also. Students are given the exercise to carry out calculations for Newton Raphson method for $x_0 = 0$ and $x_0 = 1$. Table 3.4.1 shows the results obtained in every iteration for $x_0 = 0$, $x_0 = 0.3$ and $x_0 = 1$.

Table 3.4.1

Iter. No.	$x_0 = 0$	$x_0 = 0.3$	$x_0 = 1$
1	0.6	0.622833	0.5
2	0.655102	0.656054	0.647059
3	0.656619	0.656620	0.656573
4	0.656620	0.656620	0.656620
5	0.656620	0.656620	0.656620

Thus from table we observe that if we take initial approximation as $x_0 = 0$ or $x_0 = 1$ are need one iteration extra to repeat approximate value of root upto 6 decimal digits.

3.4.2 Algorithm and Flowchart

Based on the discussion of Newton Raphson method and an illustrative example we develop an algorithm for Newton Raphson method.

Assumptions : 1) Function $f(x)$ and its derivative $f'(x)$ is predefined. 2) Correct value of initial approximation is entered which satisfies equation 3.4.7.

Step 1 : *Read the initial approximation* x_0 *and number of iterations `n' required.*

Step 2 : *Calculate :*

$f(x_0) = f(x)\,|_{x=x_0}$ *and*

$f'(x_0) = f'(x)\,|_{x=x_0}$

Step 3 : *Next approximation can be obtained as,*

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

Step 4 : *For the next iteration,* $x_0 \leftarrow x_1$

Step 5 : *Repeat step 2 to step 4 until the given number of iterations. OR Repeat step 2 to step 4 until value of root repeats upto given number of significant digits after decimal point.*

Step 6 : *Approximate value of the root is* x_1 *obtained in last iteration.*

Step 7 : Display the approximate value of root and stop.

Flowchart :

Based on the above, algorithm we can develop the flowchart for computer program. This flow chart is shown in Fig. 3.4.2. (See Fig. on next page).

3.4.3 Logic Development and C Program

A source code in `C' is listed below for Newton Raphson method. The standard header files `stdio.h`, `math.h` and `stdlib.h` are included at the start of program. Then function `main ( )` starts.

```
double    fx(double x)
```

This statement declares function `fx`, which calculates the value of function `f(x)` at given value of x. Then the next statement,

```
double    f_x(double x)
```

declares the function `f_x`, which calculates the value of $f'(x)$ [derivative of f(x)] at given value of x.

Then the program asks for value of initial approximation x_0. Then it asks for number of iterations required. The second `scanf` statement does this job. The `while` loop then keeps on calculating the values of roots in every iteration until all iterations are over.

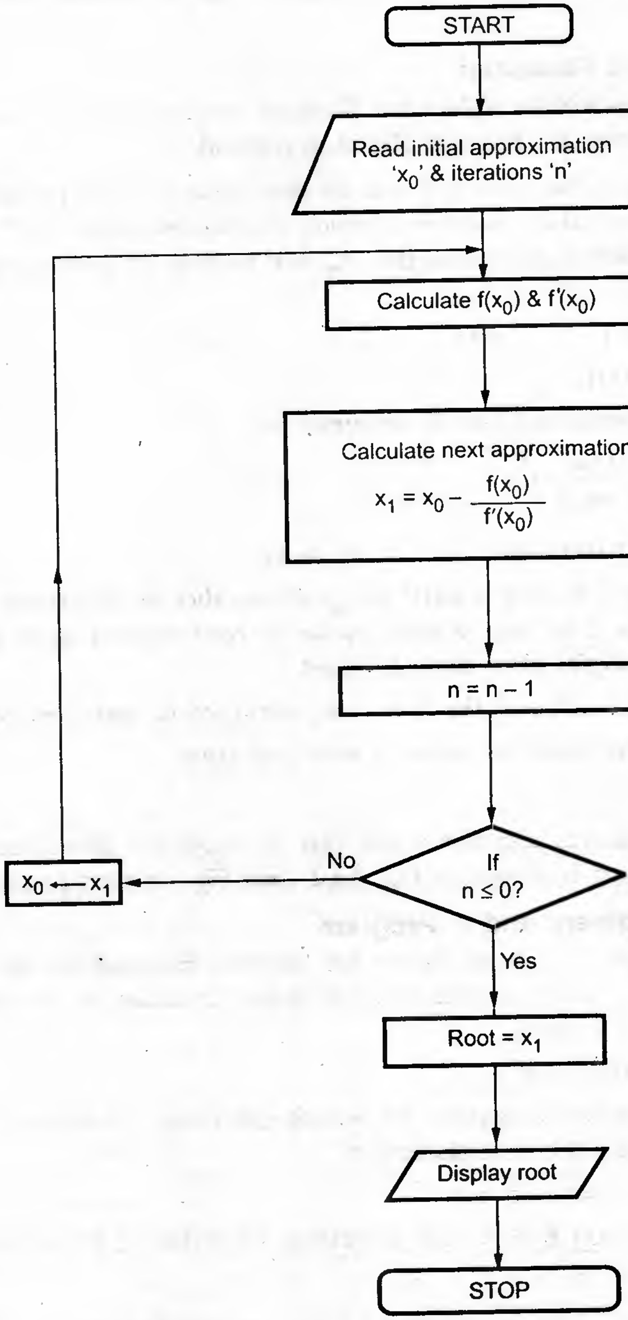


Fig. 3.4.2 Flow chart for computer program of Newton Raphson method

The first statement in the `while` loop i.e.,

```
f0 = fx(x0);
```

Calculates $f(x_0) = f(x)|_{x=x_0}$

The second statement,

```
f_0 = f_x(x0);
```

Calculates $f'(x_0) = f'(x)|_{x=x_0}$

The 3^{rd} statement in `while` loop

```
x1 = x0 - (f0/f_0);
```

Calculates next approximation to the root i.e.

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

The next `printf` statement prints the results of iteration.

For next iteration $x0 = x1$

Statement makes the program recursive.

Finally the last `printf` statement prints the value of root on the screen.

```
/* Download this program from www.vtubooks.com                 */
/* file name : nwtn_rap.cpp                                            */
/*----------- NEWTON RAPHSON METHOD TO FIND ROOT OF AN EQUATION  -------*/

/*      THE EXPRESSION FOR AN EQUATION IS DEFINED IN function fx
        YOU CAN WRITE DIFFERENT EQUATION IN function fx.
        HERE,
                  f(x) = x*x*x - 5*x + 3

        INPUTS :  1) Initial approximation x0 to the root.
                  2) Number of iterations.
        OUTPUTS : Value of the root.                                 */

/*--------------------------------   PROGRAM  ----------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>

void main()
{
      double fx ( double x);        /* DECLARATION OF  FUNCTION  */
      double f_x ( double x);       /* CALCULATION OF DERIVATIVE */

      double x0,x1,f_0,f0;
      int n,i;

      clrscr();

      printf("\n      NEWTON RAPHSON METHOD TO FIND ROOT OF AN EQUATION");

      printf("\n\n            f(x) = x*x*x - 5*x + 3");

      printf("\n\nEnter the value of initial "
             "approximation  x0 = ");
      scanf("%lf",&x0);
                    /* INITIAL APPROXIMATION x0 IS TO BE ENTERED HERE    */

      printf("\nEnter the number of iterations = ");
      scanf("%d",&n);
```

```
    printf("\npress any key for display of iterations...\n");

    getch();
    i = 0;

    while(n-- > 0)
    {
        f0 = fx(x0);            /*     CALCULATE f(x) AT x = x0         */
        f_0 = f_x(x0);      /*   CALCULATE f'(x) AT x = x1   */
        x1  = x0 - (f0/f_0);
                          /*     CALCULATION OF NEXT APPROXIMATION      */
        i++;
        printf("\n%d      x%d = %lf                          \n"
              "\n      f%d = %lf          f_%d = %lf          x%d = %lf\n"
              ,i,i-1,x0,i-1,f0,i-1,f_0,i,x1);

        x0 = x1;
        getch();
    }

    printf("\n\nThe value of root is = %20.15lf",x1);      /*   ROOT    */
}
/*---------- FUNCTION PROCEDURE TO CALCULATE VALUE OF EQUATION --------*/

double fx ( double x)
{
  double f;
  f = x*x*x - 5*x + 3;            /*      FUNCTION f(x)    */
  return(f);
}

/*---------- FUNCTION PROCEDURE TO CALCULATE f'(x0) -------------------*/

double f_x ( double x)
{
  double f_dash;
  f_dash = 3*x*x - 5; /*  DERIVATIVE OF f(x) i.e. f'(x)   */
  return(f_dash);
}
/*--------------------- End of program --------------------------------*/
```

After the `main` function two subroutine functions are listed. The first subroutine `fx` calculates f(x) at given value of x. The statement,

`f = x*x*x-5*x+3;`

is the implementation of $f(x) = x^3 - 5x + 3 = 0$. The function `f_x` calculates value of $f'(x)$ [derivative of $f(x)$] at given value of x. The statement,

`f = 3*x*x - 5;`

is the implementation of

$$f'(x) = \frac{d}{dx}[f(x)] = 3x^2 - 5$$

How to run this program :

From this source code, make EXE file. For this, compile the program on Turbo `C' compiler. It prepares executable (EXE) file. Run this executable program. The program first displays the name of the method and equation being solved.

The program then displays,

`Enter the value of initial approximation x0 =`

Here enter the value of initial approximation $x_0 = 0.3$ as we have discussed in example 3.4.1 [Note here that we are using $f(x) = x^3 - 5x + 3 = 0$ in our program, which is solved in example 3.4.1. Thus we can compare the results of this program and example 3.4.1].

Press `Enter' key after entering the value of x_0. The program then displays.

```
Enter the number of iterations =
```

Here enter the number of iterations you want to perform. Enter `4' and press `Enter' key. The program then displays the results of first iteration. The value of x1 is displayed on the screen.

The display of all the results is shown below.

```
----------------------- Results -------------------------------------

      NEWTON RAPHSON METHOD TO FIND ROOT OF AN EQUATION

             f(x) = x*x*x - 5*x + 3

Enter the value of initial approximation  x0 = 0.3

Enter the number of iterations = 4

press any key for display of iterations...

1      x0 = 0.300000

       f0 = 1.527000         f_0 = -4.730000         x1 = 0.622833

2      x1 = 0.622833

       f1 = 0.127445         f_1 = -3.836237         x2 = 0.656054

3      x2 = 0.656054

       f2 = 0.002099         f_2 = -3.708778         x3 = 0.656620

4      x3 = 0.656620

       f3 = 0.000001         f_3 = -3.706549         x4 = 0.656620

The value of root is =    0.656620431047095
----------------------------------------------------------------------
```

How to use the same program to find root of $f(x) = x - e^{-x} = 0$:

The same program can be used to solve any other equation with minor modifications.

Here, $f(x) = x - e^{-x}$

and $f'(x) = \frac{d}{dx}[f(x)] = 1 + e^{-x}$

$f''(x) = -e^{-x}$

$f(0.5) = -0.1065306$

and $f(1) = 0.6321205$ $\therefore$ root lies in [0.5, 1]

$f''(0.5) = -1.64872$

$\because$ $f(0.5)\, f''(0.5) > 0$, take initial approximation as $x_0 = 0.5$. In function

fx replace

`f=x*x*x - 5*x + 3` statement by

`f = x - exp(- x)` to implement $f(x) = x - e^{-x}$

And in function f_x replace

`f = 1 + exp(- x)` to implement $f'(x) = 1 + e^{-x}$

Compile this program again and make new EXE file. Run this program and enter $x_0 = 0.5$ and number of iterations as `4'. At the end of 4^{th} iteration, the program displays,

```
The value of root is = 0.56714329040978.
```

3.4.4 Newton Raphson Method for Multiple Roots

Without going into much detailed mathematics, we can state that if `z' is the multiple root of $f(x)$, then `z' is also root of $f'(x)$.

For example consider,

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

From above equation, it is clear that $f(x)$ has two roots at x = 2.

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

This equation has single root at x = 2. Thus x = 2 is double root of f(x) and single root of $f'(x)$. This statement can be generalized as,

If `z' is root of f(x) = 0 with multiplicity `P', then `z' is also root of $f'(x) = 0$ with multiplicity (P – 1). `z' is also root of $f''(x) = 0$ with multiplicity (P – 2) and so on.

From the graphical representation of Fig. 3.4.1 we verified the relation,

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \qquad \text{... (3.4.8)}$$

Here x_0 is initial approximation to the root and x_1 is next approximation to the root. $f'(x_0)$ is the slope of f(x) at approximate value of root. If there are `P' multiple roots at point x_0,

Then $\frac{1}{P} f'(x_0)$ is the slope of f(x) at $[x_0, f(x_0)]$

$\therefore$ Then we can write equation 3.4.8 as,

$$x_1 = x_0 - P\frac{f(x_0)}{f'(x_0)} \qquad \text{... (3.4.9)}$$

This equation can be generalized as,

$$x_{n+1} = x_n - P\frac{f(x_n)}{f'(x_n)} \qquad \text{... (3.4.10)}$$

$\because$ x_{n+1} (Approximate value of actual root `z') is also a root of $f'(x)$ with multiplicity (P – 1) as stated above, we can write equation 3.4.10 as,

$$x_{n+1} = x_n - (P-1)\frac{f'(x_n)}{f''(x_n)} \qquad \text{... (3.4.11)}$$

The above relation is obtained by treating $f'(x)$ as $f(x)$.

Thus using either equation [equation 3.4.10 or equation 3.4.11] we obtain same value of root.

Ex. 3.4.2 : *Find the double root of equation, $f(x) = x^3 - x^2 - x + 1 = 0$ using Newton Raphson method. Use $x_0 = 0.8$ as initial approximation.*

Sol. : We can use equation 3.4.10 with P = 2, since f(x) has double root.

$$f'(x) = 3x^2 - 2x - 1$$

Iteration No. 1 $\quad x_1 = x_0 - P\frac{f(x_0)}{f'(x_0)} = 0.8 - 2\frac{f(0.8)}{f'(0.8)} = 1.011765$

Iteration No. 2 $\quad x_2 = x_1 - P\frac{f(x_1)}{f'(x_1)}$

$$= 1.011765 - 2\frac{0.000278}{0.47474} = 1.000034$$

Thus the multiple root at second iteration is Root = 1.000034 (approximate).

Now let's solve the same problem using equation 3.4.11.

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

We have, $\quad f'(x) = 3x^2 - 2x - 1$

and $\quad f''(x) = 6x - 2$

$$x_0 = 0.8$$

Iteration No. 1 $\quad x_1 = x_0 - (P-1)\frac{f'(x_0)}{f''(x_0)}$

$$x_1 = 0.8 - (2-1)\frac{f'(0.8)}{f''(0.8)} = 1.042857$$

Iteration No. 2 $\quad x_2 = x_1 - (2-1)\frac{f'(x_1)}{f''(x_1)}$

$$= 1.042857 - \frac{0.176939}{4.257143} = 1.001294$$

Iteration No. 3
$$x_3 = x_2 - (2-1)\frac{f'(x_2)}{f''(x_2)}$$
$$= 1.001294 - \frac{0.005182}{4.007766}$$
$$= 1.000001$$

Thus we obtain the same value of multiple root

$$\text{Root} = 1.000001$$

3.4.5 Modified Newton Raphson Method

In modified Newton Raphson method, the derivative of f(x) is executed only at initial approximation.

Then the recursive relation of equation 3.4.5 becomes,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_0)} \quad \text{... (3.4.12)}$$

Note that $f'(x)$ is evaluated at $x = x_0$ for all values of n. This method is called modified Newton Raphson method. The advantage of this method is that it is not required to calculate $f'(x)$ again and again.

Fig 3.4.3 shows graphical representation of modified Newton Raphson method.

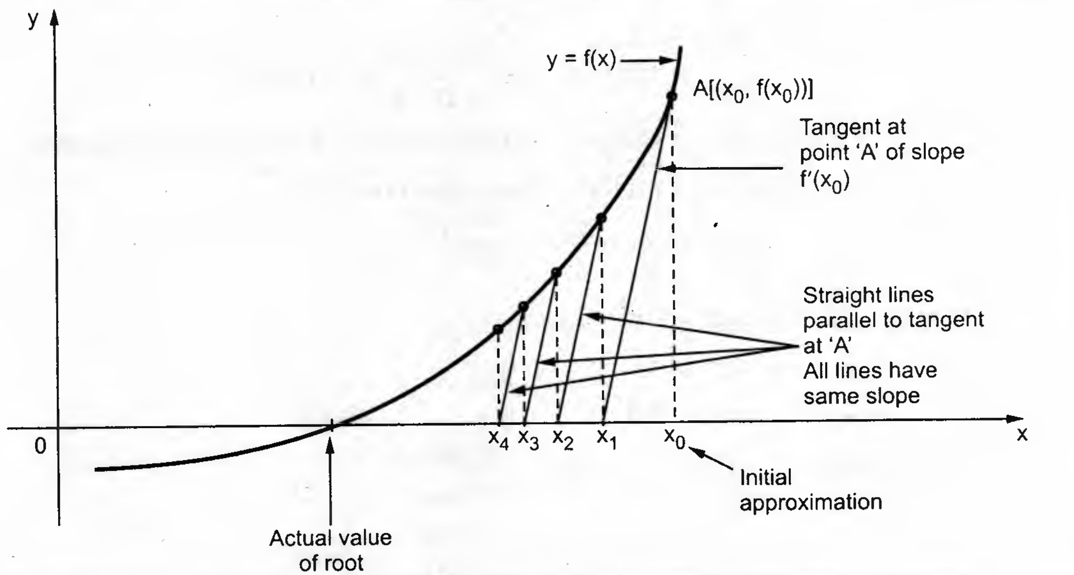


Fig. 3.4.3 Graphical interpretation of modified Newton Raphson method

Ex. 3.4.3 : *Find the root of $x^3 - 20x + 20 = 0$ lying between 1 and 1.1 correct up to 4 places of decimal using modified Newton Raphson method.*

Sol. : We have,
$$f(x) = x^3 - 20x + 20 = 0$$
$$f'(x) = 3x^2 - 20$$

and $f''(x) = 6x$

$$f(1) = 1 \qquad f'(1) = -17 \qquad f''(1) = 6$$

$$f(1.1) = -0.669 \qquad f'(1.1) = -16.37 \qquad f''(1.1) = 6.6$$

$\because \quad f(1)\cdot f''(1) > 0,$ select $x_0 = 1$

Iteration No. 1

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

$$= 1 - \frac{1}{-17}$$

$$= 1.058824$$

Iteration No. 2

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_0)}$$

$$= 1.058824 - \frac{0.010584}{-17}$$

$$= 1.059446$$

Iteration No. 3

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_0)}$$

$$= 1.059446 - \frac{0.000227}{-17}$$

$$= 1.059460$$

Thus value of x_3 repeats upto '4' significant digits after decimal point. Hence value of approximate root = 1.059460.

3.4.6 Solved Examples

Ex. 3.4.4 : *Using Newton Raphson method solve the equation $f(x) = x - e^{-x} = 0$, correct upto 5 significant digits after decimal point.*

Sol. : For this example just now on the last page we have established that x_0 should be taken as 0.5.

Iteration No. 1

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

$$= 0.5 - \frac{-0.106531}{1.606531}$$

$$= 0.566311$$

Iteration No. 2

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

$$= 0.566311 - \frac{-0.001305}{-1.1567616}$$

$$= 0.56$$

Iteration No. 3 $x_3 = x_2 - \dfrac{f(x_2)}{f'(x_2)}$

$$= 0.567143 - \frac{-0.0000002686}{1.567143}$$

$$= 0.567143$$

Here observe that, $x_2 = 0.567143$

and $x_3 = 0.567143$

Since both the values repeat upto 6 decimal places,

Approximate value of root is = 0.567143 correct upto 6 digits after decimal point.

Ex. 3.4.5 : *Find the smallest positive root of equation, tan x = x, Correct to `3' decimal places.*

Sol. : The given equation is, tanx = x

For simplicity let's write this equation as,

$$\frac{\sin x}{\cos x} = x \qquad \text{since} \quad \tan x = \frac{\sin x}{\cos x}$$

$$\therefore \quad \sin x = x \cos x$$

or $$f(x) = \sin x - x \cos x = 0$$

$$f'(x) = \frac{d}{dx}[f(x)] = \cos x - \cos x + x \sin x$$

$$f''(x) = x \sin x = \frac{d}{dx}[f'(x)] = x \cos x + \sin x$$

To establish initial approximation, we should try for only positive values of x since positive root is asked. Since required root is smallest positive, we should try from `0'. [All the values of x are in radians].

$$f(0) = 0$$

$$f(\pi/2) = 1$$

$$f(\pi) = \pi$$

$$f(3\pi/2) = -1$$

$\therefore$ root lies between π and $\dfrac{3\pi}{2}$

$$f''(\pi) = -3.141593$$

and $$f''(3\pi/2) = -1$$

$\because \quad f(3\pi/2) \cdot f''(3\pi/2) > 0,$

Take initial approximation as $x_0 = \dfrac{3\pi}{2}$

Iteration No. 1 $x_1 = x_0 - \dfrac{f(x_0)}{f'(x_0)}$

$$= \frac{3\pi}{2} - \frac{f(^{3\pi}/_2)}{f'(^{3\pi}/_2)}$$

$$= 4.712389 - \frac{(-1)}{-4.712389} \qquad (\because\ ^{3\pi}/_2 = 4.712389)$$

$$= 4.500182$$

Iteration No. 2 $\quad x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$

$$= 4.500182 - \frac{(-0.029751)}{(-4.399237)}$$

$$= 4.493420$$

Iteration No. 3 $\quad x_3 = x_2 - \frac{f(x_2)}{f'(x_2)}$

$$= 4.493420 - \frac{(-0.000044)}{(-4.386125)}$$

$$= 4.493409$$

Here we see that,

$$x_2 = 4.493420$$

and $\quad x_3 = 4.493409$

Both the values repeat upto 4^{th} decimal places. Hence,

Value of root = x_3 = 4.493409 correct upto 4 decimal places.

Ex. 3.4.6 : *Obtain the root of*

$f(x) = x \log_{10}(x) - 1.2 = 0$

using Newton Raphson method at the end of 4^{th} iteration.

Sol. : The given function is,

$$f(x) = x \log_{10}(x) - 1.2$$

We can write $\log_{10}(x)$ as

$$\log_{10}(x) = \frac{ln\ x}{ln\ 10}$$

$$\therefore \qquad f(x) = x\,\frac{lnx}{ln\,10} - 1.2$$

$$f'(x) = \frac{d}{dx}[f(x)] = \frac{1}{ln\,10}\left\{\frac{d}{dx}[x\,ln\,x]\right\} - 0$$

$$= \frac{1}{ln\,10}\left\{x \cdot \frac{1}{x} + (1)\cdot ln\,x\right\} = \frac{1 + ln\,x}{ln\,10}$$

$$f''(x) = \frac{d}{dx}[f'(x)] = \frac{1}{ln\,10}\left\{0 + \frac{1}{x}\right\} = \frac{1}{x\,ln\,10}$$

Let's establish an interval in which root lies.

$$f(0) = \infty$$

$$f(1) = -1.2 \qquad f'(1) = 0.434294 \qquad f''(1) = 0.434294$$

$$f(2) = -0.59794 \qquad f'(2) = 0.735324 \qquad f''(2) = 0.217147$$

$$f(3) = 0.231364 \qquad f'(3) = 0.911416 \qquad f''(3) = 0.144765$$

From the above two trials,

$\because$ $\quad f(2) \cdot f(3) < 0, \qquad$ root lies in [2, 3].

and $\quad f(3) \cdot f''(3) > 0, \qquad$ take initial approximation as $x_0 = 3$

Iteration No. 1 $\quad x_1 = x_0 - \dfrac{f(x_0)}{f'(x_0)} = 3 - \dfrac{f(3)}{f'(3)}$

$$= 3 - \frac{0.231364}{0.911416}$$

$$= 2.746149$$

Iteration No. 2 $\quad x_2 = x_1 - \dfrac{f(x_1)}{f'(x_1)}$

$$= 2.746149 - \frac{0.004802}{0.873019}$$

$$= 2.740649$$

Iteration No. 3 $\quad x_3 = x_2 - \dfrac{f(x_2)}{f'(x_2)}$

$$= 2.740649 - \frac{0.000002}{0.872148}$$

$$= 2.740646$$

The answer is repeating here upto 6 digits after decimal point. Therefore,

Value of root $= 2.740646$ correct upto 6 decimal digits

Ex. 3.4.7 : *Obtain one root of the equation $8x^3 - 6x - 1 = 0$ correct to 4 decimal places considering the initial value as 0.95 using Newton Raphson method.*

Sol. : $\quad f(x) = 8x^3 - 6x - 1 = 0$

$$f'(x) = 24x^2 - 6$$

Iteration No. 1 $\quad x_1 = x_0 - \dfrac{f(x_0)}{f'(x_0)}$

Here $x_0 = 0.95$ given

$\therefore \quad x_1 = 0.95 - \dfrac{0.159}{15.66}$

$$= 0.9398$$

Iteration No. 2
$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$
$$= 0.9398 - \frac{0.002342}{15.199486}$$
$$= 0.939693$$

Iteration No. 3
$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)}$$
$$= 0.939693 - \frac{0.000001}{15.192535}$$
$$= 0.939693$$

Since x_2 & x_3 repeat upto 6 decimal digits,

Value of root = x_3 = 0.939693 correct upto 6 decimal digits.

Ex. 3.4.8 : *Find the real root of the equation* $x^3 + 2x - 5 = 0$ *by applying Newton-Raphson method at the end of fifth iteration.*

Sol. : We have
$$f(x) = x^3 + 2x - 5 = 0$$
$$f'(x) = \frac{d}{dx}[f(x)] = 3x^2 + 2$$
and
$$f''(x) = \frac{d}{dx}[f'(x)] = 6x$$

$f(0) = 0 + 0 - 5 = -5$

$f(1) = 1 + 2 - 5 = -2$

$f(2) = 2^3 + 2\times 2 - 5 = 7$

Since $f(1)\cdot f(2) < 0$, root lies between 1 and 2. For better convergence we should select initial value of root x_0 such that,
$$f(x_0)\cdot f''(x_0) > 0$$

Here
$$f''(1) = f''(x)\big|_{x=1} = 6\times 1 = 6$$
$$f''(2) = f''(x)\big|_{x=2} = 6\times 2 = 12$$

Here we will select $x_0 = 2$, since,
$$f(2)\cdot f''(2) = 7\times 12 > 0$$

Now let us perform successive iterations using recursive formula of Newton Raphson Method i.e.,
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Iteration No. 1
$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$
$$= 2 - \frac{f(2)}{f'(2)} = 2 - \frac{7}{14} = 1.5$$

Iteration No. 2 $\quad x_2 = x_1 - \dfrac{f(x_1)}{f'(x_1)} = 1.5 - \dfrac{1.375}{8.75} = 1.342857$

Iteration No. 3 $\quad x_3 = x_2 - \dfrac{f(x_2)}{f'(x_2)} = 1.342857 - \dfrac{0.107242}{7.409796}$

$= 1.328384$

Iteration No. 4 $\quad x_4 = x_3 - \dfrac{f(x_2)}{f'(x_2)} = 1.328384 - \dfrac{0.000841}{7.293813}$

$= 1.328269$

Iteration No. 5 $\quad x_5 = x_4 - \dfrac{f(x_4)}{f'(x_4)} = 1.328269 - \dfrac{0}{7.292895}$

$= 1.328269$

Here observe that root is repeating upto 6 digits after decimal point. Hence the root at the end of 5th iteration is correct to 6 decimal places. It is x = 1.328269.

Ex. 3.4.9 *Apply Newton Raphson method to find a root of the equation $x^4 - x - 10 = 0$ correct upto four decimal places.* **[Dec.-95, 6 Marks, May-2000, 8 Marks]**

Sol. : To obtain initial approximation of the root

We have, $\quad f(x) = x^4 - x - 10 = 0$

$$f'(x) = \frac{d}{dx} f(x) = 4x^3 - 1$$

$$\therefore \quad f''(x) = \frac{d}{dx} f(x) = 12x$$

Let $\quad f(0) = (0)^4 - 0 - 10 = -10$

$f(1) = (1)^4 - 1 - 10 = -10$

$f(0) = (2)^4 - 2 - 10 = 4$

Since $f(1) \cdot f(2) < 0$, root lies between 1 and 2. For better convergence we should select x_0 such that $f(x_0) \cdot f''(x_0) > 0$.

Here $\quad f''(1) = 12 \times 1 = 12$

$f''(2) = 12 \times 2 = 24$

Here note that, $f(2) \cdot f''(2) = 4 \times 24 > 0$. Hence let us select initial approximation to the root as $x_0 = 2$. The recursive formula of Newton Raphson method is,

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

Iteration No. 1 : $\quad x_1 = x_0 - \dfrac{f(x_0)}{f'(x_0)} = 2 - \dfrac{f(2)}{f'(2)}$

$$= 2 - \frac{4}{4(2)^3 - 1}$$

$$= 1.870968$$

Iteration No. 2 : $x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$

$$= 1.870968 - \frac{0.382675}{25.197442}$$

$$= 1.855781$$

Iteration No. 3 : $x_3 = x_2 - \frac{f(x_2)}{f'(x_2)}$

$$= 1.855781 - \frac{0.004818}{24.564656}$$

$$= 1.855585$$

Iteration No. 4 : $x_4 = x_3 - \frac{f(x_3)}{f'(x_3)}$

$$= 1.855585 - \frac{0.000001}{24.556551}$$

$$= 1.855585$$

Here note that,

3^{rd} iteration $\Rightarrow$ root = 1.855585

4^{th} iteration $\Rightarrow$ root = 1.855585

Six digits repeat after the decimal point. Hence the root = 1.855585 is correct upto 6 decimal places.

Ex. 3.4.10 *Use Newton raphson method to obtain a root to three decimal places of the following equation :*

$$x \sin x + \cos x = 0$$

Derive the formula used.

[May-97, 10 Marks
Dec-98, 8 Marks
Dec-99, 8 Marks]

Sol. : Newton raphson formula is derived in section 3.4.1.

To obtain an initial approximation to the root

We know that $f(x) = x \sin x + \cos x$

$$\therefore \quad f'(x) = \frac{d}{dx} f(x) = x \cos x + \sin x - \sin x = x \cos x$$

$$f''(x) = \frac{d}{dx} f'(x) = x(-\sin x) + \cos x = \cos x - x \sin x$$

Let $f(0) = 0 \cdot \sin(0) + \cos(0) = 1$

$$f\left(\frac{\pi}{2}\right) = \frac{\pi}{2}\cdot\sin\frac{\pi}{2}+\cos\frac{\pi}{2} = \frac{\pi}{2} = 1.5707963$$

$$f(\pi) = \pi\cdot\sin(\pi)+\cos(\pi) = -1$$

Since $f\left(\frac{\pi}{2}\right)\cdot f(\pi) < 0$, root lies between $\frac{\pi}{2}$ and π

$$f''\left(\frac{\pi}{2}\right) = \cos\left(\frac{\pi}{2}\right)-\frac{\pi}{2}\cdot\sin\left(\frac{\pi}{2}\right) = -\frac{\pi}{2}$$

And $\quad f''(\pi) = \cos(\pi)-\pi\cdot\sin(\pi) = -1$

Here $\quad f(\pi)\cdot f''(\pi) = (-1)\cdot(-1) = 1 > 0.$

Hence, take initial approximation as $x_0 = \pi = 3.1415927$. The recursive formula of Newton Raphson method is given as,

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

Iteration No. 1 :

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

$$= x_0 - \frac{x_0\sin x_0+\cos x_0}{x_0\cos x_0}$$

$$= \pi - \frac{\pi\sin(\pi)+\cos(\pi)}{\pi\cos(\pi)}$$

$$= 2.823283$$

Iteration No. 2 :

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

$$= 2.823283 - \frac{(-0.066186)}{(-2.681457)}$$

$$= 2.798600$$

Iteration No. 3 :

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)}$$

$$= 2.798600 - \frac{(-0.000564)}{(-2.635588)}$$

$$= 2.798386$$

Iteration No. 4 :

$$x_4 = x_3 - \frac{f(x_3)}{f'(x_3)}$$

$$= 2.798386 - \frac{1.2053\times10^{-7}}{-2.635185}$$

$$= 2.798386$$

Here note that,

$$3^{rd} \text{ iteration } \Rightarrow 2.798386$$
$$4^{th} \text{ iteration } \Rightarrow 2.798386$$

Six digits repeat after the decimal point. Hence root = 2.798386 is correct upto 6 decimal places.

Ex. 3.4.11 *Use Newton Raphson method to obtain root upto 4 decimal places for the following equation.*

$$\sin(x) = 1 - x$$

[Dec-97 8 Marks]

Sol. : To obtain initial approximation to the root

The given function is, $\sin(x) = 1 - x$

i.e. $$f(x) = \sin x + x - 1 = 0$$

$\therefore$ $$f'(x) = \cos x + 1$$
$$f''(x) = -\sin x$$

Let $$f(0) = \sin(0) + 0 - 1 = -1$$
$$f\left(\frac{\pi}{2}\right) = \sin\left(\frac{\pi}{2}\right) + \frac{\pi}{2} - 1 = \frac{\pi}{2}$$

Since $f(0) \cdot f\left(\frac{\pi}{2}\right) < 0$, root lies between 0 and $\frac{\pi}{2}$

Now $$f''(0) = -\sin(0) = 0$$
$$f''\left(\frac{\pi}{2}\right) = -\sin\left(\frac{\pi}{2}\right) = -1$$

Here $f(0) \cdot f''(0) = -1 \times 0 = 0$, hence take $x_0 = 0$

To obtain the root

The recursive formula for root in Newton rephson method is given as,

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

Iteration No. 1 : $$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$
$$= x_0 - \frac{\sin(x_0) + x_0 - 1}{\cos(x_0) + 1} = 0 - \frac{\sin(0) + 0 - 1}{\cos(0) + 1} = 0.5$$

Iteration No. 2 : $$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 0.5 - \frac{-0.020574}{1.877583} = 0.510958$$

Iteration No. 3 : $$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)}$$
$$= 0.510958 - \frac{-0.000029}{1.872276} = 0.510973$$

Here note that,

2^{nd} iteration $\Rightarrow$ root = **0.510958**

3^{rd} iteration $\Rightarrow$ root = **0.510973**

Four digits after the decimal point repeat, hence root = 0.510973 is correct upto 4 decimal places.

Ex. 3.4.12 *Explain the shortcoming of Newton Raphson method with the help of an example given below. Determine the positive root of $f(x) = x^{10} - 1$ using Newton raphson method and an initial guess of x = 0.5* **[May-96, 10 Marks, May-99, 8 Marks]**

Sol. : To obtain the root

The initial approximation is given as $x_0 = 0.5$.

The function is, $f(x) = x^{10} - 1$

Hence $$f'(x) = \frac{d}{dx} f(x) = 10x^9$$

The recursive formula of Newton raphson method is given as,

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

Iteration No. 1 :

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = x_0 - \frac{(x_0)^{10} - 1}{10\, x_0^{\,9}}$$

$$= 0.5 - \frac{(0.5)^{10} - 1}{10(0.5)^9}$$

$$= 0.5 - \frac{-0.999023}{0.019531} = 51.65$$

Iteration No. 2 :

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 51.65 - \frac{1.3511490 \times 10^{17}}{2.615971 \times 10^{16}} = 46.485$$

Iteration No. 3 :

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} = 46.485 - \frac{4.711165 \times 10^{16}}{1.013480 \times 10^{16}} = 41.8365$$

Iteration No. 4 :

$$x_4 = x_3 - \frac{f(x_3)}{f'(x_3)}$$

$$= 41.8365 - \frac{1.642682 \times 10^{16}}{3.926433 \times 10^{15}}$$

$$= 37.65285$$

Iteration No. 5 :

$$x_5 = x_4 - \frac{f(x_4)}{f'(x_4)}$$

$$= 37.65285 - \frac{5.727677 \times 10^{15}}{1.521180 \times 10^{15}}$$

$$= 33.887565$$

Iteration No. 6 : $$x_6 = x_5 - \frac{f(x_5)}{f'(x_5)}$$

$$= 33.887565 - \frac{1.997118 \times 10^{15}}{5.89336401 \times 10^{14}}$$

$$= 30.498809$$

Next iterations are given in the following table.

Table 3.4.2 Solution to $x^{10} - 1$ by Newton Raphson method.

Iteration No. (x+1)	x_{n+1}	x_n	$f(x_n)$	$f'(x_n)$
7.	x_7 = 27.448928	x_6 = 30.498809	6.963518×10^{14}	2.283209×10^{14}
8.	x_8 = 24.704035	x_7 = 27.448928	2.428029×10^{14}	8.845623×10^{13}
⋮	⋮	⋮	⋮	⋮
38	x_{38} = 1.083350	x_{37} = 1.178355	4.161316	43.801037
39	x_{39} = 1.023665	x_{38} = 1.083350	1.226829	20.555034
40	x_{40} = 1.002316	x_{39} = 1.023665	0.263505	12.342962
41	x_{41} = 1.000024	x_{40} = 1.002316	0.023403	10.21384
42	x_{42} = 1.0000001	x_{41} = 1.000024	0.000239	10.002154

From the above table, observe that 42 iterations are required to reach to the root which is correct upto 3 decimal places.

Shortcoming of Newton Raphson method

In the example just discussed, large number of iterations are required. This happens because the derivative $f'(x)$ is very very small. That is near zero slope in the function. Because of this, the method may diverge. Another shortcoming is that two functions $f(x)$ and $f'(x)$ is to be calculated in every step. But this method converges fast compared to other methods. A careful selection of initial approximation can avoid the possibility of divergence.

Ex. 3.4.13 *Obtain an initial approximation to a root of the equation :*

$$f(x) = \cos x - xe^x = 0$$ **[May-97 4 Marks]**

Sol. : The given function is,

$$f(x) = \cos x - xe^x$$

$$\therefore \qquad f'(x) = \frac{d}{dx}f(x) = -\sin x - [x e^x + e^x] = -\sin x - xe^x - e^x$$

And
$$f''(x) = \frac{d}{dx}f'(x) = -\cos x - [x e^x + e^x] - e^x$$
$$= -\cos x - xe^x - 2e^x$$

Let us establish an interval in which the root lies. Hence,

Let $\qquad f(0) = \cos(0) - 0.e^0 = 1$

and $\qquad f(1) = \cos(1) - 1.e^1 = -2.1779795$

Since $f(0) \cdot f(1) < 0$, root lies between 0 and 1.

Now $\qquad f''(0) = -\cos(0) - 0.e^0 - 2e^0 = -3$

and $\qquad f''(1) = -\cos(1) - 1.e^1 - 2e^1 = -8.6951478$

Here note that $f(1) \cdot f''(1) = (-2.1779795) \cdot (-8.6951478) > 0$, hence initial approximation can be taken as $x_0 = 1$.

Ex. 3.4.14 *Use NR method to find the root of*

$$f(x) = -0.4x^2 + 2.2x + 4.7$$ **[Dec - 2004, 10 Marks]**

Sol. : Here $\qquad f(7) = 0.5$

and $\qquad f(8) = -3.3$

Since $f(7) \cdot f(8) < 0$, hence root lies between 7 and 8. Now let us determine initial approximation x_0.

$$f'(x) = -0.8x + 2.2$$
$$f''(x) = -0.8$$

Since $f(8) \cdot f''(8) > 0$, we will take $x_0 = 8$.

Following table lists the calculations.

Table 3.4.3 Solution of ex. 3.4.14

Iteration no	x_{n+1}	x_n	$f(x_n)$	$f'(x_n)$
1	$x_1 = 7.214286$	$x_0 = 8$	$f(8) = -3.3$	$f'(8) = -4.2$
2	$x_2 = 7.145143$	$x_1 = 7.214286$	-0.246939	-3.571429
3	$x_3 = 7.144599$	$x_2 = 7.145143$	-0.001912	-3.516144
4	$x_4 = 7.144599$	$x_3 = 7.144599$	0	-3.515679

Thus the root is $x_4 = 7.144599$, which is correct upto six decimal places.

Ex. 3.4.15 *Evaluate $\sqrt{12}$ to three decimal places using (i) Newton-Raphson method, (ii) Bisection method.* **[May - 2003, 10 Marks]**

Sol. : Here $\qquad x = \sqrt{12}$ or $x - \sqrt{12} = 0$

i.e. $\qquad x^2 - 12 = 0$

Hence $f(x) = x^2 - 12$

Let $f(3) = -3$ and $f(4) = 4$

Since $f(3) \cdot f(4) < 0$, root lies between [3, 4].

(i) Solution using N-R method

Here $f(x) = x^2 - 12$

and $f'(x) = 2x$

$\therefore$ $f''(x) = 2$

Since $f(4) \cdot f''(4) > 0$, will take $x_0 = 4$. Following table lists the calculations.

Table 3.4.4 Solution of $x^2 - 12 = 0$.

Iteration no	x_{n+1}	x_n	$f(x_n)$	$f'(x_n)$
1	$x_1 = 3.5$	$x_0 = 4$	$f(4) = 4$	8
2	$x_2 = 3.464286$	$x_1 = 3.5$	0.25	7
3	$x_3 = 3.464102$	$x_2 = 3.464286$	0.001276	6.928571

Here the root is repeating upto three decimal places. Hence $x_3 = 3.464102$ is the required root.

ii) Solution using bisection method

Following table lists the calculations starting with a = 3 and b = 4

Table 3.4.5 Solution of $x^2 - 12 = 0$

Iteration No	a	f(a)	b	f(b)	c	f(c)
1	3	– 3	4	4	3.5	0.25
2	3	– 3	3.5	0.25	3.25	– 1.4375
3	3.25	– 1.4375	3.5	0.25	3.375	– 0.609375
4	3.375	– 0.609375	3.5	0.25	3.4375	– 0.183594
5	3.4375	– 0.183594	3.5	0.25	3.46875	0.032227
6	3.4375	– 0.183594	3.46875	0.032227	3.453125	– 0.075928
7	3.453125	– 0.075928	3.46875	0.032227	3.460938	– 0.021912
8	3.460938	– 0.021912	3.46875	0.032227	3.464844	0.005142
9	3.460938	– 0.021912	3.464844	0.005142	3.462891	– 0.008389
10	3.462891	– 0.008389	3.464844	0.005142	3.463867	– 0.001624
11	3.463867	– 0.001624	3.464844	0.005142	3.464355	0.001754
12	3.463867	– 0.001624	3.464355	0.001754	3.464111	0.000067

Thus the value of 'c' is repeating upto 3 decimal places. Hence root is, C = 3.464111.

Conclusion : NR method required only 3 iterations, but bisection method required 12 iterations.

Exercise

1. Using Newton Raphson method, find a root of the following equation.

   $x^3 - 3x^2 - 5.5x + 9.5 = 0$ The initial guess may be assumed as zero.

   [Ans : 1.23686 correct to 5^{th} decimal place]

   [**Hint :** Consider $x_0 = 0$

   $$f(x) = x^3 - 3x^2 - 5.5x + 9.5$$

   $$f'(x) = 3x^2 - 6x - 5.5$$

   Then calculate approximate root in successive iterations

2. Use Newton Raphson method to find the root of equation $x^3 - 1.8x^2 - 10x + 17 = 0$ that lies in the interval [1, 2]. Use 5 iterations. **[Ans. : 1.661845 in 5^{th} iteration]**

   [**Hint :**

   $$f(x) = x^3 - 1.8x^2 - 10x + 17$$

   $$f'(x) = 3x^2 - 3.6x - 10$$

   and $$f''(x) = 6x - 3.6$$

   $$f(1) = 1 - 1.8 - 10 + 17 = 6.2$$

   $$f''(1) = 6 - 3.6 = 2.4$$

   $\because \quad f(1) \cdot f''(1) > 0, \quad$ select $x_0 = 1$]

3. Obtain a root of $x^3 - 4x - 9 = 0$ using Newton Raphson method.

   [Ans. : 2.706528 in 4^{th} iteration]

   [**Hint :**

   $$f(x) = x^3 - 4x - 9$$

   $$f(0) = -9$$

   $$f(1) = -12$$

   $$f(2) = -9$$

   $$f(3) = 6$$

   $\because \quad f(2) \; f(3) < 0, \quad$ root lies in 2 & 3

   $$f'(x) = 3x^2 - 4$$

   and $$f''(x) = 6x$$

   $f(2) = -9$ and $f''(2) = 12$

   $f(3) = 6$ and $f''(3) = 18$

   $\because \quad f(3) \cdot f''(3) > 0, \quad$ select $x_0 = 3$]

University Questions

1. Apply Newton Raphson's method to find a root of the equation
   $x^4 - x - 10 = 0$ correct upto four decimal places. **[Dec - 95, May - 2000]**
2. Derive the equation for Newton-Raphson method from Taylor's series. Explain using graph the working of this method. **[May - 96, May - 2001]**
3. Explain the shortcoming of Newton-Raphson method with the help of an example given below. Determine the positive root of $f(x) = ((x)^{10} - 1)$ using the Newton-Raphson method and an initial guess of x = 0.5. **[May - 96, May - 99]**
4. Develop a C program to evaluate the root of a function of the form $f(x) = 0$ using the Newton-Raphson method. **[Dec - 96]**
5. Obtain an initial approximation to a root of the equation :
   $f(x) = \cos x - xe^x = 0$ **[May - 97]**
6. Use Newton-Raphson method to obtain a root to three decimal places of the following equation :
   $$x \sin x + \cos x = 0$$
   Derive the formula used. **[May - 97, Dec - 98, Dec - 99]**
7. Use Newton-Raphson method to obtain the root upto 4 decimal places for the following equation.
   $\sin(x) = 1 - x.$ **[Dec - 97]**
8. Develop a C program to evaluate the root (correct to 5 decimal places) using any iteration method for the following :
   $e^{-x} = 10x.$ **[Dec - 97, Dec - 2000]**
9. Derive the formula for Newton-Raphson's method analytically and verify graphically. **[Dec - 2001]**
10. Derive Newton-Raphson's method graphically and verify it analytically. Write a program in C/C + + for the same. **[Dec - 2002]**
11. Name and explain the bracketing method and open method to solve transcedental equation. Use NR method to find the root of f(x) = $-0.4x^2 + 2.2x + 4.7$. **[Dec - 2004, 10 Marks]**

3.5 Convergence of Methods

Every methods need different number of iterations to reach to the actual value of root. Rate of convergence gives the measure of how fast a particular method goes close to actual value of root.

Rate of convergence is defined as follows.

The rate of convergence p is the largest positive real number for which there exists a finite constant C $\neq$ 0 such that

$$|\varepsilon_{k+1}| \leq C\,|\varepsilon_k|^p$$

where $\varepsilon_k = x_k - z$ (z is actual value of root)

$= $ error in k^{th} iteration

$\varepsilon_{k+1} = $ error in $(k+1)^{th}$ iteration

C = Asymptotic error constant and it depends upon derivatives of $f(x)$ at x = z

3.5.1 Rate of Convergence of Secant Method

By secant method, the next approximation to the root is given by equation 3.2.10 as,

$$x_{n+1} = \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}\, f(x_n) \qquad \text{... (3.5.1)}$$

Here

$$x_n = z + \varepsilon_n$$

$$x_{n+1} = z + \varepsilon_{n+1}$$

$$x_{n-1} = z + \varepsilon_{n-1}$$

Putting those values in equation 3.5.1.

$$z + \varepsilon_{n+1} = z + \varepsilon_n - \frac{z + \varepsilon_n - z - \varepsilon_{n-1}}{f(z+\varepsilon_n) - f(z - \varepsilon_{n-1})}\, f(z+\varepsilon_n)$$

$$\therefore \qquad \varepsilon_{n+1} = \varepsilon_n - \frac{(\varepsilon_n - \varepsilon_{n-1}) f(z+\varepsilon_n)}{f(z+\varepsilon_n) - f(z - \varepsilon_{n-1})} \qquad \text{... (3.5.2)}$$

Now let's expand $f(z+\varepsilon_n)$ using Taylor's series about z.

$$f(z+\varepsilon_n) = f(z) + \varepsilon_n f'(z) + \frac{\varepsilon_n^2}{2!} f''(z) + \ldots\ldots..$$

$$f(z+\varepsilon_{n-1}) = f(z) + \varepsilon_{n-1} f'(z) + \frac{\varepsilon_{n-1}^2}{2!} f''(z) + \ldots\ldots.$$

$$f(z+\varepsilon_n) - f(z+\varepsilon_{n-1}) = \varepsilon_n f'(z) - \varepsilon_{n-1} f'(z) + \frac{\varepsilon_n^2}{2!} f'(z) - \frac{\varepsilon_{n-1}^2}{2!} f''(z) + \ldots\ldots.$$

$$= (\varepsilon_n - \varepsilon_{n-1}) f'(z) + \left(\frac{\varepsilon_n^2}{2} - \frac{\varepsilon_{n-1}^2}{2}\right) f''(z) + \ldots\ldots \qquad \text{.... (3.5.3)}$$

and

$$(\varepsilon_n - \varepsilon_{n-1}) f(z+\varepsilon_n) = (\varepsilon_n - \varepsilon_{n-1}) f(z) + (\varepsilon_n - \varepsilon_{n-1})\varepsilon_n f'(z) + (\varepsilon_n - \varepsilon_{n-1}) \frac{\varepsilon_n^2}{2!} f''(z) + \ldots\ldots$$

since $f(z) = 0$ at value of root z,

$$(\varepsilon_n - \varepsilon_{n-1}) f(z+\varepsilon_n) = (\varepsilon_n - \varepsilon_{n-1})\, \varepsilon_n f'(z) + (\varepsilon_n - \varepsilon_{n-1}) \frac{\varepsilon_n^2}{2!} f''(z) + \ldots. \qquad \text{... (3.5.4)}$$

Take the ratio,

$$\frac{(\varepsilon_n - \varepsilon_{n-1})\ f(z+\varepsilon_n)}{f(z+\varepsilon_n) - f(z+\varepsilon_{n-1})} = \frac{(\varepsilon_n - \varepsilon_{n-1})\varepsilon_n\ f'(z) + (\varepsilon_n - \varepsilon_{n-1})\ \frac{\varepsilon_n^2}{2!} f''(z) + \ldots}{(\varepsilon_n - \varepsilon_{n-1}) f'(z) + \left(\frac{\varepsilon_n^2}{2} - \frac{\varepsilon_{n-1}^2}{2}\right) f''(z) + \ldots.}$$

(from equation 3.5.3 and equation 3.5.4)

$$= \frac{(\varepsilon_n - \varepsilon_{n-1})\left[\varepsilon_n\ f'(z) + \frac{\varepsilon_n^2}{2!} f''(z) + \ldots\right]}{(\varepsilon_n - \varepsilon_{n-1})\left[f'(z) + \left(\frac{\varepsilon_n + \varepsilon_{n-1}}{2}\right) f''(z) + \ldots.\right]} = \frac{\varepsilon_n\ f'(z) + \frac{\varepsilon_n^2}{2!} f''(z) + \ldots}{f'(z) + \left(\frac{\varepsilon_n + \varepsilon_{n-1}}{2}\right) f''(z) + \ldots.}$$

$$= \frac{f'(z)\left[\varepsilon_n + \frac{\varepsilon_n^2}{2}\frac{f''(z)}{f'(z)} + \ldots\right]}{f'(z)\left[1 + \left(\frac{\varepsilon_n + \varepsilon_{n-1}}{2}\right)\frac{f''(z)}{f'(z)} + \ldots.\right]}$$

$$= \frac{\varepsilon_n + \frac{\varepsilon_n^2}{2}\frac{f''(z)}{f'(z)} + \ldots}{1 + \frac{\varepsilon_n + \varepsilon_{n-1}}{2}\frac{f''(z)}{f'(z)} + \ldots.} \qquad \ldots (3.5.5)$$

Let's perform this division

$$\varepsilon_n - \frac{\varepsilon_n \varepsilon_{n-1}}{2}\frac{f''(z)}{f'(z)} + \ldots.$$

$$1 + \frac{\varepsilon_n + \varepsilon_{n-1}}{2}\frac{f''(z)}{f'(z)} + \ldots. \quad \Big) \quad \varepsilon_n + \frac{\varepsilon_n^2}{2}\frac{f''(z)}{f'(z)} + \ldots.$$

$$-\varepsilon_n + \frac{\varepsilon_n^2}{2}\frac{f''(z)}{f'(z)} + \frac{\varepsilon_n \varepsilon_{n-1}}{2}\frac{f''(z)}{f'(z)} + \ldots.$$

$$0 \qquad 0 \qquad -\frac{\varepsilon_n \varepsilon_{n-1}}{2}\frac{f''(z)}{f'(z)} + \ldots.$$

$$-\frac{\varepsilon_n \varepsilon_{n-1}}{2}\frac{f''(z)}{f'(z)} - \frac{\varepsilon_n \varepsilon_{n-1}}{2}\cdot\frac{\varepsilon_n \varepsilon_{n-1}}{2}\frac{f''(z)^2}{f'(z)^2} + \ldots\ldots..$$

0 0 + Remainder terms of higher order

Therefore we can write equation 3.5.5 as,

$$\frac{(\varepsilon_n - \varepsilon_{n-1})\, f(z+\varepsilon_n)}{f(z+\varepsilon_n) - f(z-\varepsilon_{n-1})} = \varepsilon_n - \frac{\varepsilon_n\, \varepsilon_{n-1}}{2}\, \frac{f''(z)}{f'(z)} + \ldots \qquad \text{..... (3.5.6)}$$

(Remainder of higher terms is neglected)

Putting the value obtained in equation 3.5.6 in equation 3.5.2,

$$\varepsilon_{n+1} = \varepsilon_n - \varepsilon_n + \frac{\varepsilon_n \varepsilon_{n-1}}{2}\, \frac{f''(z)}{f'(z)} + \ldots$$

$$= \frac{\varepsilon_n \varepsilon_{n-1}}{2}\, \frac{f''(z)}{f'(z)} \qquad \text{... (3.5.7)}$$

(Here we have neglected higher order derivatives)

Let's define here value of `c' as,

$$c = \frac{1}{2}\, \frac{f''(z)}{f'(z)} \qquad \text{... (3.5.8)}$$

∴ equation 3.5.7 can be written as,

$$\varepsilon_{n+1} = c\, \varepsilon_n \varepsilon_{n+1} \qquad \text{... (3.5.9)}$$

This equation is called error equation.

The convergence relation we have defined as,

$$|\varepsilon_{n+1}| \leq c\, |\varepsilon_n|^p \qquad \text{... (3.5.10)}$$

Let's write this relation as,

$$\varepsilon_{n+1} = A\, \varepsilon_n^p$$

(Here we have to define values of A and p) ... (3.5.11)

We can also write this equation as,

$$\varepsilon_n = A\, \varepsilon_{n-1}^p$$

$$\therefore \quad \varepsilon_{n-1}^p = \frac{\varepsilon_n}{A}$$

$$\therefore \quad \varepsilon_{n-1} = \left(\frac{\varepsilon_n}{A}\right)^{\frac{1}{p}} \qquad \text{... (3.5.12)}$$

Putting the values of ε_{n+1} and ε_{n-1} from equation 3.5.11 and 3.5.12 in equation 3.5.9 we get,

$$A\varepsilon_n^p = c\varepsilon_n \left(\frac{\varepsilon_n}{A}\right)^{\frac{1}{p}}$$

$$A\varepsilon_n^p = c\varepsilon_n (\varepsilon_n)^{\frac{1}{p}} \cdot \left(\frac{1}{A}\right)^{\frac{1}{p}}$$

$$\therefore \qquad A\varepsilon_n^p = c(\varepsilon_n)^{1+\frac{1}{p}} \cdot \left(\frac{1}{A}\right)^{\frac{1}{p}}$$

Since LHS = RHS of above equations, the powers of ε_n can be compared. Then we get

$$p = 1+\frac{1}{p} \qquad \text{or} \qquad p^2 - p - 1 = 0$$

Solving this equation for p we get,

$$p = \frac{1}{2}(1 \pm \sqrt{5})$$

$$p = 1.618 \text{ by taking higher value of p}$$

Thus we can write convergence equation 3.5.10 as,

$$\varepsilon_{n+1} \le \frac{1}{2}\frac{f''(z)}{f'(z)}|\varepsilon_n|^{1.618} \qquad \text{... (3.5.13)}$$

This is the convergence equation for secant method, and Rate of convergence, p = 1.618.

3.5.2 Rate of Convergence of Newton Raphson Method

From equation 3.4.5, we know that,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \qquad \text{... (3.5.14)}$$

This equation gives next approximation to the root by newton raphson method.

Let `z' be the actual value of root.

Then we can write

$$x_n = z + \varepsilon_n$$

$$x_{n+1} = z + \varepsilon_{n+1}$$

Putting those values in equation 3.5.14 we have,

$$z + \varepsilon_{n+1} = z + \varepsilon_n - \frac{f(z+\varepsilon_n)}{f'(z+\varepsilon_n)}$$

OR

$$\varepsilon_{n+1} = \varepsilon_n - \frac{f(z+\varepsilon_n)}{f'(z+\varepsilon_n)} \qquad \text{... (3.5.15)}$$

Let's expand $f(z+\varepsilon_n)$ and $f'(z+\varepsilon_n)$ using Taylor's series around `z'

$$f(z+\varepsilon_n) = f(z) + \varepsilon_n f'(z) + \frac{\varepsilon_n^2}{2!}f''(z) + ...$$

$$f'(z+\varepsilon_n) = f'(z) + \varepsilon_n f''(z) + \frac{\varepsilon_n^2}{2!}f'''(z) + ...$$

From above two expansions we can write,

$$\frac{f(z+\varepsilon_n)}{f'(z+\varepsilon_n)} = \frac{f(z)+\varepsilon_n f'(z)+\frac{\varepsilon_n^2}{2!}f''(z)+\ldots\ldots}{f(z)+\varepsilon_n f''(z)+\frac{\varepsilon_n^2}{2!}f'''(z)+\ldots\ldots}$$

At actual value of root, $f(z) = 0$

$$\therefore \quad \frac{f(z+\varepsilon_n)}{f'(z+\varepsilon_n)} = \frac{\varepsilon_n f'(z)+\frac{\varepsilon_n^2}{2!}f''(z)+\ldots\ldots}{f'(z)+\varepsilon_n f''(z)+\ldots\ldots}$$

Neglecting 3^{rd} and higher derivatives

$$\therefore \quad \frac{f(z+\varepsilon_n)}{f'(z+\varepsilon_n)} = \frac{f'(z)\left[\varepsilon_n+\frac{\varepsilon_n^2}{2}\frac{f''(z)}{f'(z)}+\ldots\ldots\right]}{f'(z)\left[1+\varepsilon_n\frac{f''(z)}{f'(z)}+\ldots\ldots\right]}$$

$$= \frac{\varepsilon_n+\frac{\varepsilon_n^2}{2}\frac{f''(z)}{f'(z)}+\ldots\ldots}{1+\varepsilon_n\frac{f''(z)}{f'(z)}+\ldots\ldots} = \frac{N}{D} \qquad \ldots (3.5.16)$$

Now let's perform this division as follows.

$$\begin{array}{r|l}
 & \varepsilon_n-\frac{\varepsilon_n^2}{2}\frac{f''(z)}{f'(z)}+\ldots\ldots \\
\hline
1+\varepsilon_n\frac{f''(z)}{f'(z)}+\ldots\ldots & \varepsilon_n+\frac{\varepsilon_n^2}{2}\frac{f''(z)}{f'(z)}+\ldots\ldots \\
 & -\varepsilon_n+\frac{\varepsilon_n^2}{2}\frac{f''(z)}{f'(z)}+\ldots\ldots \\
\cline{2-2}
 & 0-\frac{\varepsilon_n^2}{2}\frac{f''(z)}{f'(z)}+\ldots\ldots \\
 & -\frac{\varepsilon_n^2}{2}\frac{f''(z)}{f'(z)}-\frac{\varepsilon_n^3}{2}\frac{f''(z)^2}{f'(z)^2}+\ldots\ldots \\
\cline{2-2}
 & 0 \qquad + \text{Higher order derivatives remainders}
\end{array}$$

From this division result we can write equation 3.5.16 as,

$$\frac{f(z+\varepsilon_n)}{f'(z+\varepsilon_n)} = \varepsilon_n-\frac{\varepsilon_n^2}{2}\frac{f''(z)}{f'(z)}+\ldots\ldots$$

Putting this result in equation 3.5.16 we get,

$$\varepsilon_{n+1} = \varepsilon_n - \varepsilon_n + \frac{\varepsilon_n^2}{2} \frac{f''(z)}{f'(z)} + \ldots\ldots$$

Neglecting the higher order derivatives we get,

$$\varepsilon_{n+1} = \frac{\varepsilon_n^2}{2} \frac{f''(z)}{f'(z)} \qquad \ldots (3.5.17)$$

In this equation let $c = \frac{1}{2} \frac{f''(z)}{f'(z)}$

$$\therefore \qquad \varepsilon_{n+1} = c\varepsilon_n^2 \qquad \ldots (3.5.18)$$

From equation 3.5.10 we know that

$$|\varepsilon_{n+1}| \le c|\varepsilon_n|^p$$

Comparing this equation with equation 3.5.18, we obtain, p = 2

Thus Newton Raphson method has second order convergence. It is also called quadratic convergence.

3.5.3 Rate of Convergence of Regula Falsi Method

The iteration equation of regula falsi method is given as,

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n) \qquad \text{(from equation 3.3.2)}$$

Normally in regula falsi method x_n and x_{n-1} are selected such that,

$$f(x_n) \cdot f(x_{n-1}) < 0$$

and next approximation x_{n+1} as well as final value root lies in $[x_n, x_{n-1}]$. Depending upon the selection of x_n and x_{n-1}, the value of x_{n+1} varies. x_n and x_{n-1} are updated in every iteration.

We know from equation 3.3.2 that,

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n) \qquad \ldots (3.5.19)$$

Here root lies in $[x_n, x_{n-1}]$

Let x_{n-1} = a = fixed value

And root always lies in $[x_n, a]$.

Then equation 3.6.19 becomes,

$$x_{n+1} = x_n - \frac{x_n - a}{f(x_n) - f(a)} f(x_n) \qquad \ldots (3.5.20)$$

Let `z' be the actual value of root then we can write

$$x_{n+1} = z + \varepsilon_{n+1}$$

$$x_n = z + \varepsilon_n$$

Putting those values in equation 3.5.20,

$$z+\varepsilon_{n+1} = z+\varepsilon_n - \frac{z+\varepsilon_n - a}{f(z+\varepsilon_n)-f(a)} f(z+\varepsilon_n)$$

$$\therefore \quad \varepsilon_{n+1} = \varepsilon_n - \frac{(z+\varepsilon_n - a) f(z+\varepsilon_n)}{f(z+\varepsilon_n)-f(a)}$$

Expanding $f(z+\varepsilon_n)$ around `z' using Taylor's series,

$$\varepsilon_{n+1} = \varepsilon_n - \frac{(z+\varepsilon_n - a)\left[f(z)+\varepsilon_n f'(z)+\frac{\varepsilon_n^2}{2!}f''(z)+\ldots\ldots\right]}{\left[f(z)+\varepsilon_n f'(z)+\frac{\varepsilon_n^2}{2!}f''(z)+\ldots\ldots\right]-f(a)}$$

After simplification and neglecting higher order derivatives, the above equation reduces to,

$$\varepsilon_{n+1} = \frac{f(a)+(z-a)f'(z)}{f(a)}\varepsilon_n \qquad \ldots (3.5.21)$$

Let

$$c = \frac{f(a)+(z-a)f'(z)}{f(a)}$$

Then equation 3.5.21 becomes,

$$\varepsilon_{n+1} = c\,\varepsilon_n \qquad \ldots (3.5.22)$$

From equation 3.5.10 the convergence equation is given as,

$$|\varepsilon_{n+1}| \le c|\varepsilon_n|^p$$

On comparison of this equation with equation 3.5.22, we observe that the rate of convergence `p' is given as, p = 1.

Thus the convergence rate of Ragula Falsi method is linear.

University Questions

1. Prove that the Newton-Raphson process has a quadratic convergence. **[May - 97]**
2. Evaluate $\sqrt{12}$ to three decimal places using (i) Newton-Raphson method, (ii) Bisection method. **[May - 2003, 10 Marks]**

3.6 Comparison or Choice of Iterative Method

So far now we have seen various types of numerical methods to find the roots of a transcendental and algebraic equations. Those methods can be compared or selected depending upon the following aspects :

1. The rate of convergence of the method.
2. Computational complexity of the method.
3. Dependence of the method on initial approximation.

Of all methods, Newton-Raphson method converges fast compared to other methods. This can be easily detected by solving the same example by different methods. But if $f'(x)$ becomes very very small in Newton Raphson method, then there is the possibility that the method may diverge. More computation efforts are required in Newton Raphson method, Since we have to calculate $f(x)$ and $f'(x)$in every step.

The Regula Falsi or False position method have definite convergence. The method has first order convergence rate hence it requires more number of iterations. That is the method has slow convergence.

The bisection method is simplest but has slow convergence. The computational effort required is comparatively less.

The computation procedure of Regula Falsi and secant method is similar. But in secant method, x_n and x_{n-1} need not enclose the root. The convergence rate of secant method is better than bisection and successive approximation.

The successive approximation or iterative method is simplest of all methods. It has first order convergence, but the method has no guaranteed convergence.

Table 3.6.1 shows the comparison of all the methods.

Table 3.6.1 : Comparison of methods

Sr. No.	Name of the method	Iterative formula of the method	Rate of convergence	Evaluation of functions per iteration	Minimum statements of `c' required for logic implementation	Reliability of convergence
1.	Bisection method	$c = \frac{a+b}{2}$ Root lies in [a, b] `c' is next approximation.	Gain of one bit per iteration	1	1	Guaranteed convergence
2.	Secant method	$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n)$	1.618	1	1	No guarantee of convergence if not near root
3.	Regula Falsi or False position method	$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n)$ x_n & x_{n-1} enclose the root	1	1	1	Guaranteed convergence
4.	Newton Raphson method	$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$	2	2	2	Sensitive to starting value. Convergence is fast if x_0 is close to root.
5.	Iterative method	$x_{n+1} = \phi(x_n)$	1	1	1	Easy to program but no guarantee of convergence

The table also gives logical implementation complexity of various methods. Following steps must be followed to obtain the value of root :

1. Plot the function $f(x)$ and find the approximate location of the root.
2. Find two initial values which enclose the root.
3. Start iterations using bisection method.
4. When the root is very close to actual value, perform next iterations using Newton Raphson method.

3.7 Engineering Applications

The methods discussed in this chapter can be used to solve problems in electrical engineering. The equations can be solved for given parameters. Following example illustrates this application.

Ex. 3.7.1 *Fig. 3.7.1 shows an electric circuit. The switch 's' was closed to position '1' for a long time.*

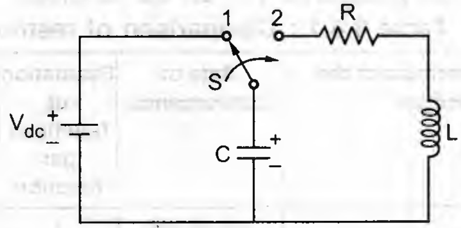


Fig. 3.7.1 An electric circuit

The switch is then closed to position '2'. Then the charge 'Q' in the capacitor can be expressed as,

$$Q(t) = Q_0 e^{-t\frac{R}{2L}} \cos\left[\sqrt{\frac{1}{LC}-\left(\frac{R}{2L}\right)^2}\cdot t\right] \qquad \text{... (3.7.1)}$$

And at $t = 0$, $Q_0 = V_{dc} \cdot C$. Determine the value of resistance which dissipates 1 percent charge of the capacitor of its original value in 50 milliseconds. Take $L = 5H$ and $C = 10^{-4}$ F.

Sol. : i) Formulation of the problem

Consider equation 3.7.1, i.e.,

$$Q(t) = Q_0 e^{-t\frac{R}{2L}} \cos\left[\sqrt{\frac{1}{LC}-\left(\frac{R}{2L}\right)^2}\cdot t\right]$$

The above equation can be written as,

$$e^{-t\frac{R}{2L}} \cos\left[\sqrt{\frac{1}{LC}-\left(\frac{R}{2L}\right)^2}\cdot t\right] - \frac{Q(t)}{Q_0} = 0$$

Here Q_0 is the original (initial) charge on the capacitor. And we have to determine 'R' such that Q(t) becomes 1 percent of Q_0.

Hence $$\frac{Q(t)}{Q_0} = 1\ \% = 0.01$$

Putting this value in above equation,

$$e^{-t\frac{R}{2L}} \cos\left[\sqrt{\frac{1}{LC}-\left(\frac{R}{2L}\right)^2}\ .\ t\right] - 0.01 = 0$$

L = 5H, C = $10^{-4}F$ and t = 50 milliseconds (given). Then above equation becomes,

$$e^{-0.05\frac{R}{2\times 5}} \cdot \cos\left[\sqrt{\frac{1}{5\times 10^{-4}}-\left(\frac{R}{2\times 5}\right)^2}\ .\ 0.05\right] - 0.01 = 0$$

$$\therefore \quad e^{-0.005R} \cos(0.05\sqrt{2000-0.01R^2}) - 0.01 = 0 \qquad \text{... (3.7.2)}$$

Solution of the above equation gives the value of resistance 'R' which will dissipate the required charge of the capacitor in given time. Clearly, the solution of above equation is complicated. This equation can be solved using numerical methods such as bisection, secant or regula falsi.

iii) **To establish the interval in which value of 'R' lies**

Equation 3.7.2 can be called $f(R)$, i.e.

$$\therefore \quad f(R) = e^{-0.005R} \cos(0.05\sqrt{2000-0.01R^2}) - 0.01 \qquad \text{... (3.7.3)}$$

Let $$f(0) = e^0 \cos(0.05\sqrt{2000}) - 0.01 = -\ 0.6272728$$

In equation 3.7.3 note that 2000 – 0.01 R^2 must be greater than zero, otherwise R will be imaginary.

i.e. $$2000 - 0.01R^2 \geq 0$$

Hence $$2000 \geq 0.01\ R^2$$

$$\therefore \quad R \leq 447.21$$

Hence Let $$f(400) = e^{-\ 0.005\times 400} \cos(0.05\sqrt{2000 - 0.01(400)^2}) - 0.01$$
$$= 0.0631219$$

Since $f(0)\cdot f(400) < 0$, value of R lies between 0 and 400 Ω.

iv) **To obtain 'R' using bisection method**

Iteration No. 1 : Since R lies between 0 and 400 Ω. Let a = 0 and b = 400.

Then $$c = \frac{a+b}{2} = \frac{0+400}{2} = 200$$

$$\therefore \quad f(c) = f(200) = e^{-0.005\times 200} \cos(0.05\sqrt{2000-0.01(200)^2}) - 0.01$$

$$= -\,0.1630918$$

And $f(a) = f(0) = -\,0.6272728$

$f(b) = f(400) = 0.0631219$

Since $f(200) \cdot f(400) < 0$, R lies in [200, 400]

Iteration No. 2 : New interval : a = 200, b = 400.

Then $c = \dfrac{a+b}{2} = \dfrac{200+400}{2} = 300$

$\therefore$ $f(c) = f(300) = -\,0.029503$

and $f(a) = f(200) = -\,0.1630918$

$f(b) = f(400) = 0.0631219$

Since $f(300) \cdot f(400) < 0$, R lies in [300, 400]

Iteration No. 3 : New interval : a = 300, b = 400.

Hence $c = \dfrac{a+b}{2} = \dfrac{300+400}{2} = 350$

$\therefore$ $f(c) = f(350) = -\,0.020915$

and $f(a) = f(300) = -\,0.029503$

$f(b) = f(400) = 0.0631219$

Since $f(300) \cdot f(350) < 0$, R lies in [300, 350]

Iteration No. 4 : New interval : a = 300, b = 350.

Then $c = \dfrac{a+b}{2} = \dfrac{300+350}{2}$

$= 325$

$\therefore$ $f(c) = f(325) = -\,0.003155$

and $f(a) = f(300) = -\,0.029503$

$f(b) = f(350) = 0.020915$

Since $f(325) \cdot f(350) < 0$, R lies in [325, 350]

Table 3.7.1 lists the next iterations of this method.

Table 3.7.1 : Solution using bisection method

Iteration No.	Interval [a, b]	$c = \frac{a+b}{2}$	F(a)	F(b)	F(c)
5.	a = 325 b = 350	337.5	– 0.003155	0.020915	0.009150
6.	a = 325 b = 337.5	331.25	– 0.003155	0.009150	0.003067
7.	a = 325 b = 331.25	328.125	– 0.003155	0.003067	– 0.000026

8.	a = 328.125 b = 331.25	329.6875	– 0.000026	0.003067	0.001525
9.	a = 328.125 b = 329.6875	328.90625	– 0.000026	0.001525	0.000750
10.	a = 328.125 b = 328.906	328.5156	– 0.000026	0.000750	0.000362
⋮	⋮	⋮	⋮	⋮	⋮
19.	a = 328.15094 b = 328.152466	328.151703			
20.	a = 328.15094 b = 328.151703	328.151321			

From the above table observe that R = 328.151321 which is correct upto three decimal places. Actually R = 328 Ω is sufficient answer in this example. Hence we can stop after 10^{th} iterations also.

3.8 MATLAB to Obtain Root of the Polynomial

We have seen how C language can be effectively used to implement numerical techniques. Now we will consider a MATLAB program to calculate roots of the polynomial consider a polynomial,

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + ...+ a_1 x + a_0$$

The roots to this polynomial can be obtained by a simple MATLAB program as given below :

```
%   Download this program from www.vtubooks.com
%   file name : polyRoots
%   Roots of the polynomial. Matlab Version 6
%   This program calculates the roots of the polynomial
%   Inputs : Coefficients of the polynomial of the form,
%            y = an * x^n + an-1 * x^n-1 + . . . + a1 * x + a0
%        The above coefficients should be entered as follows:
%            a = [an an-1 an-2 . . . a1 a0]
%   Outputs: Display of roots of the polynomial

%-----------------------------------------------------------------------

a = input('Enter the coefficients of polynomial = ');
%                       Coefficients of the polynomial are accepted here
r = roots(a);           % Roots are calculated here
disp('The roots are as follows');
disp(r);                % Display of the roots

%-------------- End of the program ----------------------------------------
```

In the above program the `input` statement accepts the coefficients $a_n, a_{n-1}, ...a_1, a_0$. The statement `r = roots(a);` calculates the roots of the polynomial. Then the last statement displays the roots.

To test this program

Let us consider the polynomial of example 3.1.1 i.e.

$$f(x) = x^3 - 1.8x^2 - 10x + 17 = 0$$

Run this MATLAB program and enter the coefficients as,

```
[ 1 - 1.8 -10  17 ]
```

Then the program displays the roots. The complete display of results of this program is given below :

```
---------------------- Results -------------------------------------------
>> polyRoots
Enter the coefficients of polynomial = [1 -1.8 -10 17]
The roots are as follows
   -3.1300
    3.2682
    1.6618
--------------------------------------------------------------------------
```

From the above results observe that one of the root is 1.6618. This root is also calculated by bisection method in example 3.1.1 and it is 1.671875.

Computer Exercise

We have discussed programs for every method. Perform the following assignments and write your own `C' programs.

1. *Modify the program given here to find the root of function $f(x) = x^3 - 4x - 9 = 0$ using bisection method.* **[Ans. : 2.710375 in 6^{th} iteration]**

2. *Write the `C' program to find the root of function $f(x) = \cos x - xe^x = 0$ using secant method.*

3. *Write the `C' program to find the value of angle `θ' when amplitude of sine wave becomes half of its peak positive value.*

   [Hint : Here define f(θ) as, $f(\theta) = \sin\theta - \frac{1}{2} = 0$]

4. *Write the `C' program to find the value of root using modified newton raphson method. Assume suitable function.*

5. *Write the `C' program to find multiple roots of the function. Use the following function.* $f(x) = x^3 - x^2 - x + 1 = 0$

   This function has double root near 0.8.

❑❑❑

4
Systems of Linear Algebraic Equations

In chapter 3 we studied some methods to obtain a root of transcendental and algebraic equations. In this chapter we will study some methods to obtain solution of system of linear equations. In engineering and science applications we always come across these type of equations. Normally a system of linear equations with 'n' unknowns is represented as,

$$\begin{aligned}
a_{11}\, x_1 + a_{12}\, x_2 + \ldots + a_{1n}\, x_n &= b_1 \\
a_{21}\, x_1 + a_{22}\, x_2 + \ldots + a_{2n}\, x_n &= b_2 \\
a_{31}\, x_1 + a_{32}\, x_2 + \ldots + a_{3n}\, x_n &= b_3 \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots &\;\ldots \\
a_{n1}\, x_1 + a_{n2}\, x_2 + \ldots + a_{nn}\, x_n &= b_n \qquad \ldots (4.1)
\end{aligned}$$

Here $x_1, x_2, x_3, \ldots x_n$ are the variables whose values are to be found.

In this chapter we will study two major types of methods to obtain solution of system of linear equations. They are –

1) **Direct Methods :** These methods use elimination of variable. They transform the system of equations to triangular form. The important direct methods are Cramer rule, Gauss elimination method, Gauss jordon elimination method, Triangularization method, Cholesky method etc.

2) **Iterative Method :** These methods use principles of successive approximation. The iterations are repeated till the required accuracy is obtained. Jacobi's iteration method, Gauss seidel iteration method, etc. are important iterative methods.

4.1 Systems with Small Number of Equations

The number of equations of unknowns upto three can be solved directly. No computer methods are required for such small number of equations. Cramer's rule, graphical method, etc methods are used for small number of equations.

4.1.1 Graphical Method

In this method, the two equations are plotted on cartesian co-ordinates. One axis corresponds to x_1 and other corresponds to x_2. The two equations represent the two straight lines. The solution of the equations is the point where two lines cross each other.

Ex. 4.1.1 *Obtain the solution of the simultaneous equations graphically by plotting graphs for*

$$3x_1 + 2x_2 = 10$$
$$-x_1 + 2x_2 = 2$$

[May-96 8 Marks, May-98 6 Marks, May-2000 8 Marks]

Sol. : The given equations can be written as,

$$2x_2 = -3x_1 + 10$$

and $$2x_2 = x_1 + 2$$

That is, $$\left.\begin{aligned} x_2 &= -\frac{3}{2}x_1 + 5 \\ \text{and} \quad x_2 &= \frac{1}{2}x_1 + 1 \end{aligned}\right\} \quad \ldots (4.1.1)$$

Thus the above two equations represent two straight lines with slopes of $-\frac{3}{2}$ and $\frac{1}{2}$. Following table lists the values of x_1 and x_2 for above equations.

Table 4.1.1 Values of x_1 and x_2 as per equation 4.1.1

x_1	$x_2 = -\frac{3}{2}x_1 + 5$
1	3.5
2	2
3	0.5
4	− 1
5	− 2.5
6	− 4
7	− 5.5

x_1	$x_2 = -\frac{3}{2}x_1 + 5$
1	1.5
2	2
3	2.5
4	3
5	3.5
6	4
7	4.5

The values of x_1 and x_2 calculated in above table are plotted in Fig. 4.4.1. (See Fig. 4.1.1 on next page).

In the Fig. 4.1.1, observe that the two lines intersect each other at $x_1 = 2$, $x_2 = 2$. Hence the solution of the system of equations is,

$$x_1 = 2 \quad x_2 = 2$$

4.1.2 Cramer's Rule

Let's explain this rule with three variables. Then the system of equations for 3 unknown becomes,

$$\begin{aligned} a_{11}\, x_1 + a_{12}\, x_2 + a_{13}\, x_3 &= b_1 \\ a_{21}\, x_1 + a_{22}\, x_2 + a_{23}\, x_3 &= b_2 \\ a_{31}\, x_1 + a_{32}\, x_2 + a_{33}\, x_3 &= b_3 \end{aligned} \quad \ldots (4.1.2)$$

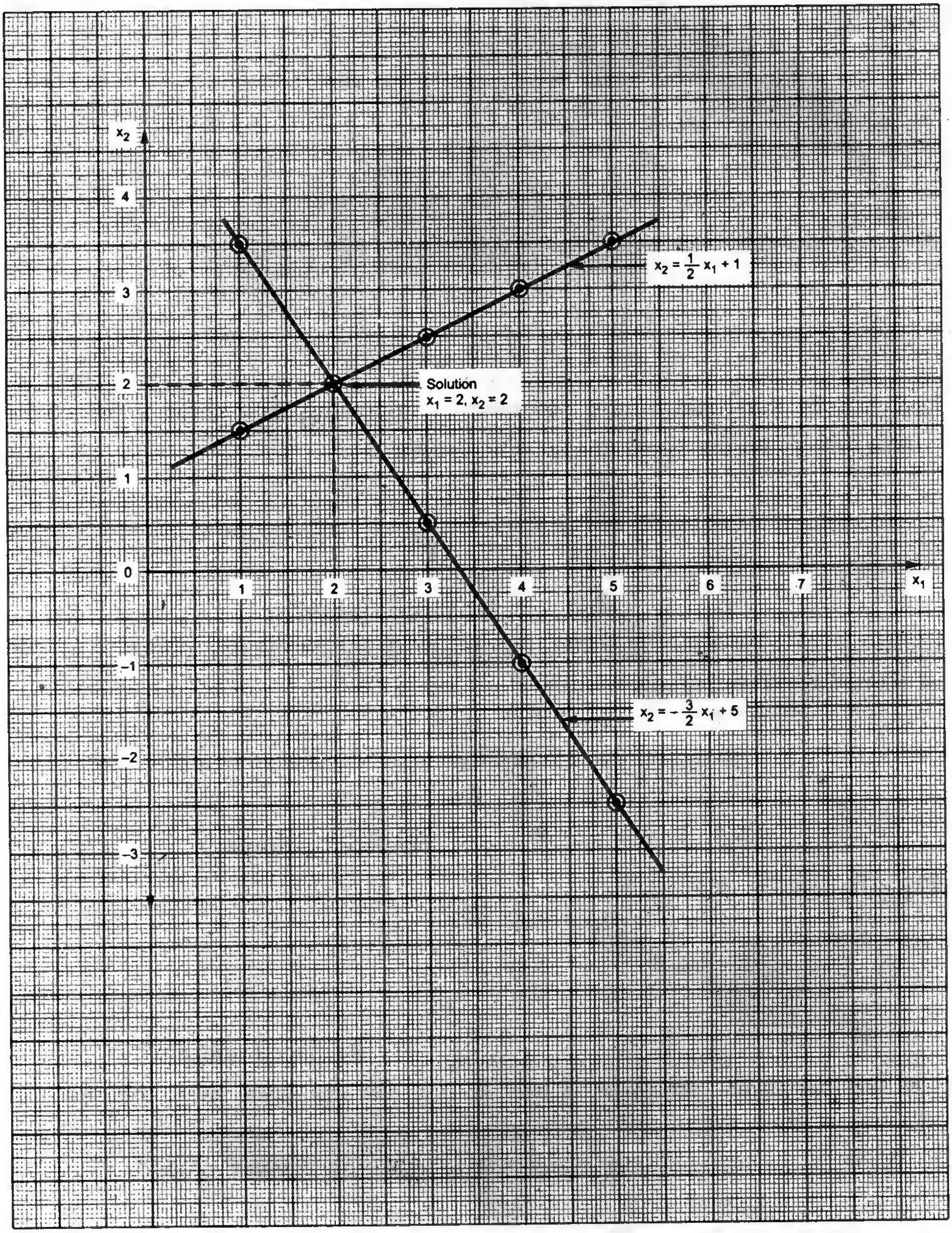


Fig. 4.1.1 Graphical method to solve linear equations

Here we have to find values of x_1, x_2 and x_3. By Cramer's rule, these values are obtained as,

$$x_1 = \frac{\Delta x_1}{\Delta} \quad \text{... (4.1.3)}$$

$$x_2 = \frac{\Delta x_2}{\Delta} \quad \text{... (4.1.4)}$$

& $$x_3 = \frac{\Delta x_3}{\Delta} \quad \text{... (4.1.5)}$$

Here Δ is the determinant of coefficients i.e.,

$$\Delta = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \quad \text{... (4.1.6)}$$

And Δx_1 is the determinant obtained by replacing first column (or coefficient column of x_1) in `Δ' by values of b_1, b_2 and b_3 i.e.,

$$\Delta x_1 = \begin{vmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & b_{23} \\ b_3 & a_{32} & a_{33} \end{vmatrix} \quad \text{... (4.1.7)}$$

Replaced column

Similarly Δx_2 is the determinant obtained by replacing coefficient column x_2 in `Δ' by values of b_1, b_2, b_3 etc. i.e.,

$$\Delta x_2 = \begin{vmatrix} a_{11} & b_1 & a_{13} \\ a_{21} & b_2 & b_{23} \\ a_{31} & a_3 & a_{33} \end{vmatrix} \quad \text{... (4.1.8)}$$

Replaced column

Similarly, Δx_3 is written as,

$$\Delta x_3 = \begin{vmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ a_{31} & a_{32} & b_3 \end{vmatrix} \quad \text{... (4.1.9)}$$

Replaced column

Here the necessary condition is that $\Delta \neq 0$. On the same lines this rule can be extended to more than three unknowns.

Ex 4.1.2 *Solve the following system of linear equations using Cramer's rule.*

$$\begin{aligned} x_1 + x_2 + x_3 &= 6 \\ 3x_1 + 3x_2 + 4x_3 &= 20 \\ 2x_1 + x_2 + 3x_3 &= 13 \end{aligned}$$

Sol. : Let's find the value of determinant of coefficients using equation 4.1.6,

$$\Delta = \begin{vmatrix} 1 & 1 & 1 \\ 3 & 3 & 4 \\ 2 & 1 & 3 \end{vmatrix} \quad \text{... (4.1.10)}$$

$$= 1\begin{vmatrix} 3 & 4 \\ 1 & 3 \end{vmatrix} - 1\begin{vmatrix} 3 & 4 \\ 2 & 3 \end{vmatrix} + 1\begin{vmatrix} 3 & 3 \\ 2 & 1 \end{vmatrix}$$

$$= (9-4)-(9-8)+(3-6)$$

$$= 1$$

Let's calculate Δx_1 from equation 4.1.7,

$$\Delta x_1 = \begin{vmatrix} 6 & 1 & 1 \\ 20 & 3 & 4 \\ 13 & 1 & 3 \end{vmatrix} = 6(9-4)-1(60-52)+1(20-39)$$

$$= 3$$

From equation 4.1.3 the value of x_1 is given as,

$$x_1 = \frac{\Delta x_1}{\Delta} = \frac{3}{1}$$

$\therefore$ $x_1 = 3$

Let's calculate Δx_2 from equation 4.1.8 i.e.,

$$\Delta x_2 = \begin{vmatrix} 1 & 6 & 1 \\ 3 & 20 & 4 \\ 2 & 13 & 3 \end{vmatrix} = 1(60-52)-6(9-8)+1(39-40)$$

$$= 1$$

From equation 4.1.4 value of x_2 is given as,

$$x_2 = \frac{\Delta x_2}{\Delta} = \frac{1}{1}$$

$\therefore$ $x_2 = 1$

Let's calculate Δx_3 from equation 4.1.9 i.e.

$$\Delta x_3 = \begin{vmatrix} 1 & 1 & 6 \\ 3 & 3 & 20 \\ 2 & 1 & 13 \end{vmatrix}$$

$$= 1(39-20)-1(39-40)+6(3-6)$$

$$= 2$$

$\therefore$ We can obtain value of x_3 from equation 4.1.5 as,

$$x_3 = \frac{\Delta x_3}{\Delta}$$

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

$\therefore$ $x_3 = 2$

Thus we have,

$x_1 = 3$, $x_2 = 1$ and $x_3 = 2$

4.1.3 Algorithm and Computer Program Logic

Let's see the algorithm for Cramer's rule.

Algorithm :

Step 1 : *Read the total number of variables 'n' in the system of equations.*

Step 2 : *Read the values of coefficients equation–wise in the two dimensional matrix.*

i.e.

Read a_{ij} *for* $i = 1$ *to* n

& $j = 1$ *to* n

& read b_{ji} *for* $i = 1$ *to* n

Step 3 : *Calculate determinant of the coefficient matrix i.e.*

$$\Delta = |a_{ij}|$$

Step 4 : *Calculate determinant* Δx_i *by replacing* i^{th} *column by* b_j. *Here* b_j *is the column of n elements.*

Step 5 : *Calculate value of variable as,*

$$x_i = \frac{\Delta x_i}{\Delta}$$

Step 6 : *Repeat step 4 and step 5 for* $i = 1, 2, \ldots n$.

Step 7 : *Display the values of variables obtained in step 4. i.e. Display,*

x_i *for* $i = 1, 2, \ldots n$.

Step 8 : *Stop.*

Exercise

1. Solve the following system of equations using Cramer's rule.

$$x_1 + 2x_2 - x_3 = 2$$
$$3x_1 + 6x_2 + x_3 = 1$$
$$3x_1 + 3x_2 + 2x_3 = 3$$

$$\left[\textbf{Ans. :}\ x_1 = \frac{35}{12},\ x_2 = -\frac{13}{12} \text{ and } x_3 = -\frac{15}{12}\right]$$

2. Solve the following system of equations using Cramer's rule.

$$3x + y + 2z = 3$$
$$2x - 3y - z = -3$$
$$x + 2y + z = 4$$

[Ans. : x = 1, y = 2 and z = – 1]

3. Use Cramer's rule to solve

$$0.3x_1 + 0.52x_2 + x_3 = -0.01$$
$$0.5x_1 + x_2 + 1.9x_3 = 0.67$$
$$0.1x_1 + 0.3x_2 + 0.5x_3 = -0.44$$

[May-96, 9 Marks; Dec-96, 9 Marks; May-98, 8 Marks]

Ans. : $x_1 = -14.9$, $x_2 = -29.5$ and $x_3 = 19.8$

University Questions

1. Obtain the solution of the simultaneous equations graphically by plotting graphs for -

[May - 96, May - 98, May - 2000]

$$3x_1 + 2x_2 = 10$$
$$-x_1 + 2x_2 = 2$$

2. Use Cramer's rule to solve -

$$0.3x_1 + 0.52x_2 + x_3 = -0.01$$
$$0.5x_1 + x_2 + 1.9x_3 = 0.67$$
$$0.1x_1 + 0.3x_2 + 0.5x_3 = -0.44$$

[May - 96, May - 98, Dec - 99]

4.2 Matrix Inversion Method

Consider the system of equation of 3 unknowns,

$$\begin{aligned} a_{11}\, x_1 + a_{12}\, x_2 + a_{13}\, x_3 &= b_1 \\ a_{21}\, x_1 + a_{22}\, x_2 + a_{23}\, x_3 &= b_2 \\ a_{31}\, x_1 + a_{32}\, x_2 + a_{33}\, x_3 &= b_3 \end{aligned} \quad \text{... (4.2.1)}$$

Let's denote these equations in the matrix form,

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad \text{... (4.2.2)}$$

Here let,

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \; X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \; \& \; B = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad \text{... (4.2.3)}$$

Then equation 4.2.2 becomes,

$$AX = B \quad \text{... (4.2.4)}$$

Multiply both the sides of above matrices by A^{-1} (Here A^{-1} is inverse of matrix A),

$$A^{-1} AX = A^{-1} B$$

$$\therefore \quad X = A^{-1} B \qquad (\because AA^{-1} = I \text{ and } IX = X) \quad \text{... (4.2.5)}$$

Thus the values of x_1, x_2 and x_3 can be easily obtained by solving equation 4.2.5. This method can be further extended on the same lines for more number of unknowns.

How to obtain Inverse of a Matrix?

The inverse of matrix is given as,

$$A^{-1} = \frac{1}{|A|} \, Adj\ (A) \quad \text{... (4.2.6)}$$

Here $|A|$ is determinant of matrix A and Adj (A) is adjoint of matrix A.

Now we will see how to obtain adjoint of matrix.

Consider matrix A given by equation 4.2.3,

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \qquad \text{... (4.2.7)}$$

Let's define cofactor A_{ij} as,

$$A_{ij} = (-1)^{i+j} M_{ij} \qquad \text{Here } M_{ij} \text{ is major of } a_{ij} \qquad \text{... (4.2.8)}$$

Using this relation we will first define cofactors for matrix A of equation 4.2.7 as follows,

$$A_{11} = (-1)^{1+1} M_{11} \qquad \text{... (4.2.9)}$$

Here, M_{11} = Minor of a_{11} in matrix A

$$= \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix}$$

Similarly, $A_{12} = (-1)^{1+2} M_{12}$ ... (4.2.10)

& M_{12} = Minor of a_{12} in matrix A

$$= \begin{bmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{bmatrix}$$

$$A_{13} = (-1)^{1+3} M_{13} \qquad \text{... (4.2.11)}$$

& M_{13} = Minor of a_{13} in matrix A

$$= \begin{bmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}$$

Similarly,

$$A_{21} = (-1)^{2+1} M_{21} \;\&\; M_{21} = \begin{bmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{bmatrix} \qquad \text{... (4.2.12)}$$

$$A_{22} = (-1)^{2+2} M_{22} \;\&\; M_{22} = \begin{bmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{bmatrix} \qquad \text{... (4.2.13)}$$

$$A_{23} = (-1)^{2+3} M_{23} \;\&\; M_{23} = \begin{bmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{bmatrix} \qquad \text{... (4.2.14)}$$

Similarly we can obtain other cofactors.

The matrix of cofactors of A will be,

$$\text{Matrix of cofactors} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \qquad \text{... (4.2.15)}$$

The adjoint matrix 'Adj (A)' is equal to the transpose of matrix of cofactors given above.

$$Adj(A) = \text{transpose of matrix of cofactors} = \begin{bmatrix} A_{11} & A_{21} & A_{31} \\ A_{12} & A_{22} & A_{32} \\ A_{13} & A_{23} & A_{33} \end{bmatrix} \qquad \text{... (4.2.16)}$$

By putting the value of Adj (A) in equation 4.2.6 we will get A^{-1}. The same procedure discussed above can be extended to find inverse of higher size matrices.

Ex. 4.2.1 *Find the inverse of the matrix, given below.*

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

Sol. : Determinant of this matrix is,

$$|A| = 5(0-1)+2(0-4)+4(-2-4) = -37$$

Now lets find cofactors with the help of equation 4.2.9 to equation 4.2.14 given just now

$$A_{11} = \begin{vmatrix} 1 & 1 \\ 1 & 0 \end{vmatrix} = 0-1 = -1$$

$$A_{12} = -\begin{vmatrix} -2 & 1 \\ 4 & 0 \end{vmatrix} = -(0-4) = 4$$

$$A_{13} = \begin{vmatrix} -2 & 1 \\ 4 & 1 \end{vmatrix} = (-2-4) = -6$$

$$A_{21} = -\begin{vmatrix} -2 & 4 \\ 1 & 0 \end{vmatrix} = -(0-4) = 4$$

$$A_{22} = \begin{vmatrix} 5 & 4 \\ 4 & 0 \end{vmatrix} = (0-16) = -16$$

$$A_{23} = -\begin{vmatrix} 5 & -2 \\ 4 & 1 \end{vmatrix} = -(5+8) = -13$$

$$A_{31} = \begin{vmatrix} -2 & 4 \\ 1 & 1 \end{vmatrix} = (-2-4) = -6$$

$$A_{32} = -\begin{vmatrix} 5 & 4 \\ -2 & 1 \end{vmatrix} = -(5+8) = -13$$

$$A_{33} = \begin{vmatrix} 5 & -2 \\ -2 & 1 \end{vmatrix} = (5-4) = 1$$

Thus the matrix of cofactors is,

$$\text{Matrix of cofactors} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = \begin{bmatrix} -1 & 4 & -6 \\ 4 & -16 & -13 \\ -6 & -13 & 1 \end{bmatrix}$$

Adjoint of A is equal to the transpose of the cofactors matrix i.e.,

$$Adj(A) = \begin{bmatrix} A_{11} & A_{21} & A_{31} \\ A_{12} & A_{22} & A_{32} \\ A_{13} & A_{23} & A_{33} \end{bmatrix} = \begin{bmatrix} -1 & 4 & -6 \\ 4 & -16 & -13 \\ -6 & -13 & 1 \end{bmatrix}$$

From equation 4.2.6 A^{-1} is given as,

$$A^{-1} = \frac{1}{|A|} Adj(A) = -\frac{1}{37}\begin{bmatrix} -1 & 4 & -6 \\ 4 & -16 & -13 \\ -6 & -13 & 1 \end{bmatrix}$$

This is the required inverse.

Ex. 4.2.2 *Solve the following system of equations using matrix inversion method.*

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

Sol. : Let's express these equations in the matrix format of equation 4.2.2. i.e.,

$$\begin{bmatrix} 3 & 1 & 2 \\ 2 & -3 & -1 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 \\ -3 \\ 4 \end{bmatrix} \quad \text{... (4.2.17)}$$

Here according to equation 4.2.3 we have,

$$A = \begin{bmatrix} 3 & 1 & 2 \\ 2 & -3 & -1 \\ 1 & 2 & 1 \end{bmatrix}, X = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, B = \begin{bmatrix} 3 \\ -3 \\ 4 \end{bmatrix}$$

From equation 4.2.5 the solution of these equations is given as,

$$X = A^{-1} B$$

Let's first find inverse of matrix A.

Determinant of matrix A is given as,

$$\begin{aligned} |A| &= 3(-3+2) - 1(2+1) + 2(4+3) \\ &= 8 \end{aligned}$$

Let's find cofactors of matrix A.

$$A_{11} = \begin{vmatrix} -3 & -1 \\ 2 & 1 \end{vmatrix} = (-3+2) = -1$$

$$A_{12} = -\begin{vmatrix} 2 & -1 \\ 1 & 1 \end{vmatrix} = -(2+1) = -3$$

$$A_{13} = \begin{vmatrix} 2 & -3 \\ 1 & 2 \end{vmatrix} = (4+3) = 7$$

$$A_{21} = -\begin{vmatrix} 1 & 2 \\ 2 & 1 \end{vmatrix} = -(1-4) = 3$$

$$A_{22} = \begin{vmatrix} 3 & 2 \\ 1 & 1 \end{vmatrix} = (3-2) = 1$$

$$A_{23} = -\begin{vmatrix} 3 & 1 \\ 1 & 2 \end{vmatrix} = -(6-1) = -5$$

$$A_{31} = \begin{vmatrix} 1 & 2 \\ -3 & -1 \end{vmatrix} = (-1+6) = 5$$

$$A_{32} = -\begin{vmatrix} 3 & 2 \\ 2 & -1 \end{vmatrix} = -(-3-4) = 7$$

$$A_{33} = \begin{vmatrix} 3 & 1 \\ 2 & -3 \end{vmatrix} = (-9-2) = -11$$

$$\therefore \text{ Cofactor matrix of 'A'} = \begin{bmatrix} -1 & -3 & 7 \\ 3 & 1 & -5 \\ 5 & 7 & -11 \end{bmatrix}$$

$\therefore$ Adjoint of matrix A = Transpose of cofactor matrix of A.

$$\therefore \quad Adj\ (A) = \begin{bmatrix} -1 & 3 & 5 \\ -3 & 1 & 7 \\ 7 & -5 & -11 \end{bmatrix}$$

$$A^{-1} = \frac{1}{|A|}\ Adj\,(A) \qquad \text{from equation 4.2.6}$$

$$\therefore \quad A^{-1} = \frac{1}{8}\begin{bmatrix} -1 & 3 & 5 \\ -3 & 1 & 7 \\ 7 & -5 & -11 \end{bmatrix}$$

From equation 4.2.5 the solution of equations is given as,

$$X = A^{-1}B = \frac{1}{8}\begin{bmatrix} -1 & 3 & 5 \\ -3 & 1 & 7 \\ 7 & -5 & -11 \end{bmatrix}\begin{bmatrix} 3 \\ -3 \\ 4 \end{bmatrix}$$

$$= \frac{1}{8}\begin{bmatrix} (-3-9+20) \\ (-9-3+28) \\ (21+15-44) \end{bmatrix} = \frac{1}{8}\begin{bmatrix} 8 \\ 16 \\ -8 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}$$

Thus we have, $X = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}$

$\therefore$ We have $\quad x = 1, \quad y = 2 \quad \& \quad z = -1$

Exercise

1. Solve the following system of equations using matrix inversion method.

$$x_1 + 2x_2 - x_3 = 2$$
$$3x_1 + 6x_2 + x_3 = 1$$
$$3x_1 + 3x_2 + 2x_3 = 3$$

$$\left[\textbf{Ans.}: x_1 = \frac{35}{12},\ x_2 = -\frac{13}{12} \text{ and } x_3 = -\frac{15}{12}\right]$$

2. Compute the inverse of the matrix.

$$\begin{bmatrix} 3 & 2 & 4 \\ 2 & 1 & 1 \\ 1 & 3 & 5 \end{bmatrix}$$

and use the result to solve the system of equations :

$$3x + 2y + 4z = 7$$
$$2x + y + z = 7$$
$$x + 3y + 5z = 2$$

$$\left[\textbf{Ans.}: x = \frac{9}{4},\ y = -\frac{9}{8} \text{ and } z = \frac{5}{8}\right]$$

4.3 Substitution Methods

The methods we studied in the last two sections become complex when number of variables are more than three. Two substitution methods are discussed here. They are backward substitution and forward substitution methods. Normally these methods are used along with other methods to reduce the burden of calculations.

4.3.1 Backward Substitution Method

Consider the following system of equations involving four unknowns,

$$\begin{aligned} a_{11}\,x_1 + a_{12}\,x_2 + a_{13}\,x_3 + a_{14}\,x_4 &= b_1 \\ 0 + a_{22}\,x_2 + a_{23}\,x_3 + a_{24}\,x_4 &= b_2 \\ 0 + 0 + a_{33}\,x_3 + a_{34}\,x_4 &= b_3 \\ 0 + 0 + 0 + a_{44}\,x_4 &= b_4 \end{aligned} \qquad \text{... (4.3.1)}$$

Let's form the matrix of these equations.

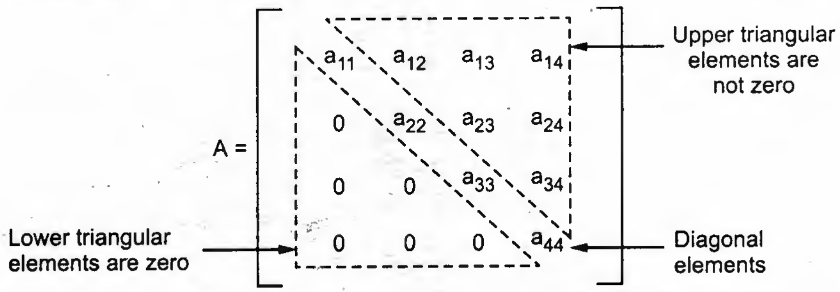


Fig. 4.3.1

In the matrix A, we observe that all the elements below diagonal are zero. This is called 'upper triangular form' of equations. To solve these equations we start from the last equation.

i.e., $a_{44}\,x_4 = b_4$ (Last equation in equation 4.3.1)

$$\therefore \quad x_4 = \frac{b_4}{a_{44}} \qquad \text{... (4.3.2)}$$

From equation 4.3.1 we know that the second last equation is,

$$a_{33}\,x_3 + a_{34}\,x_4 = b_3$$

Putting value of x_4 in this equation obtained above,

$$a_{33}\,x_3 + a_{34}\cdot\frac{b_4}{a_{44}} = b_3$$

$$\therefore \quad a_{33}\,x_3 = b_3 - a_{34}\cdot\frac{b_4}{a_{44}}$$

$$\therefore \quad x_3 = \frac{1}{a_{33}}\left(b_3 - a_{34}\cdot\frac{b_4}{a_{44}}\right) \qquad \text{... (4.3.3)}$$

Similarly we can substitute the values of x_3 and x_4 in second equation to obtain value of x_2. The values of x_2, x_3 and x_4 are substituted in first equation and value of x_1 is calculated.

Since we start from the last equation, this method is called backward substitution.

4.3.1.1 Algorithm for Backward Substitution

Step 1 : *Read number of variables 'n' in the system of equations.*

Step 2 : *Read the upper triangular system of equations*

i.e. Read $\mathbf{a}_{ij}$

For $i = 1$ *to* n

& $j = 1$ *to* n

(only for $j \geq i$*)*

& Read $\mathbf{b}_i$ *for* $i = 1$ *to* n

Step 3 : *Calculate,*

$$\mathbf{x}_n = \frac{\mathbf{b}_n}{a_{nn}} \quad \textit{i.e. last variable}$$

Step 4 : *Calculate*

$\mathbf{x}_i$ *for* $i = n - 1$ *to* 1

By putting values of x_j*;* $j = i + 1$ *to* n *in respective equation.*

Step 5 : *Display the values of variables* $\mathbf{x}_i$ *for* $i = 1$ *to* n *on the screen.*

Step 6 : *Stop.*

4.3.2 Forward Substitution Method

Now lets consider an alternate form of equations having four unknowns –

$$\begin{aligned}
a_{11}\,x_1 + 0 + 0 + 0 &= b_1 \\
a_{21}\,x_1 + a_{22}\,x_2 + 0 + 0 &= b_2 \\
a_{31}\,x_1 + a_{32}\,x_2 + a_{33}\,x_3 + 0 &= b_3 \\
a_{41}\,x_1 + a_{42}\,x_2 + a_{43}\,x_3 + a_{44}\,x_4 &= b_4
\end{aligned} \qquad \text{... (4.3.4)}$$

Let's form the matrix of these equations,

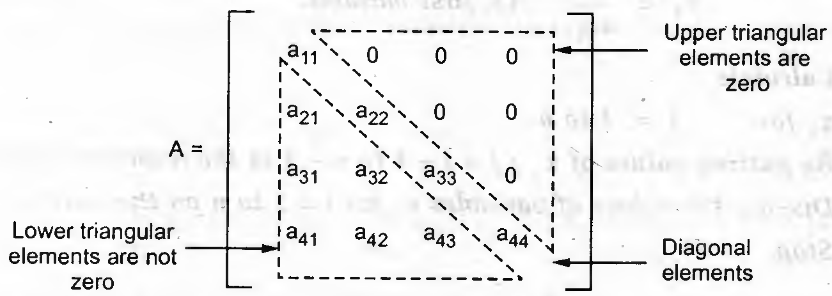


Fig. 4.3.2

Here we see that all the elements above diagonal are zero. This system of equations is called 'lower triangular form' of equations. To solve these type of equations, we start from first equation in 4.3.4 i.e.,

$$a_{11} x_1 = b_1$$

$$\therefore \quad x_1 = \frac{b_1}{a_{11}} \qquad \text{... (4.3.5)}$$

The second equation in 4.3.4 is,

$$a_{21} x_1 + a_{22} x_2 = b_2$$

Substitute the value of x_1 in this equation,

$$a_{21} \cdot \frac{b_1}{a_{11}} + a_{22} x_2 = b_2$$

$$\therefore \quad x_2 = \frac{1}{a_{22}} \left(b_2 - a_{21} \cdot \frac{b_1}{a_{11}} \right) \qquad \text{... (4.3.6)}$$

Similarly we can obtain the value of x_3 from third equation by putting value of x_1 and x_2.

Since we start from the first equation, this method is called forward substitution method. In rare cases we get the equations of upper triangular or lower triangular form. The backward and forward substitution methods are used with other methods to solve system of equations.

4.3.2.1 Algorithm for Forward Substitution

Step 1 : Read the number of variables 'n' in the system of equations.

Step 2 : Read the lower triangular system of equations.

i.e. Read $\mathbf{a}_{ij}$

for $i = 1$ *to* n

& $j = 1$ *to* n

(only for $\mathbf{i} \geq \mathbf{j}$*)*

& Read $\mathbf{b}_i$ *for* $i = 1$ *to* n

Step 3 : Calculate,

$$\mathbf{x}_1 = \frac{\mathbf{b}_1}{a_{11}} \quad \textit{i.e. first variable.}$$

Step 4 : Calculate

$\mathbf{x}_i$ *for* $i = 2$ *to* n

By putting values of $\mathbf{x}_j$ *;* $j = i - 1$ *to* $n - 1$ *in the respective equations.*

Step 5 : Display the values of variables $\mathbf{x}_i$ *for* $i = 1$ *to* n *on the screen.*

Step 6 : Stop.

4.4 Gauss Elimination Method

4.4.1 Technique

In gauss elimination method, the given system of linear equations is converted to upper triangular system by elimination of variables. The upper triangular system is then solved by backward substitution.

Consider the following system of equations having three unknowns.

$$a_{11}\, x_1 + a_{12}\, x_2 + a_{13}\, x_3 = b_1 \qquad \text{... [4.4.1. (a)]}$$

$$a_{21}\, x_1 + a_{22}\, x_2 + a_{23}\, x_3 = b_2 \qquad \text{... [4.4.1 (b)]}$$

$$a_{31}\, x_1 + a_{32}\, x_2 + a_{33}\, x_3 = b_3 \qquad \text{... [4.4.1 (c)]}$$

First stage of elimination :

Divide equation 4.4.1 (a) by a_{11} we get,

$$x_1 + \frac{a_{12}}{a_{11}}\, x_2 + \frac{a_{13}}{a_{11}}\, x_3 = \frac{b_1}{a_{11}} \qquad \text{... (4.4.2)}$$

Here a_{11} should not be zero.

Multiply equation 4.4.2 by a_{21} we get,

$$a_{21}\, x_1 + \frac{a_{12}}{a_{11}}\, a_{21}\, x_2 + \frac{a_{13}}{a_{11}}\, a_{21}\, x_3 = \frac{b_1}{a_{11}}\, a_{21}$$

Subtract this equation from equation 4.4.1 (b), i.e. second equation, we get,

$$0 + \left(a_{22} - \frac{a_{12}}{a_{11}}\, a_{21}\right) x_2 + \left(a_{23} - \frac{a_{13}}{a_{11}}\, a_{21}\right) x_3 = \left(b_2 - \frac{b_1}{a_{11}}\, a_{21}\right) \qquad \text{... (4.4.3)}$$

Similarly multiply equation 4.4.2 by coefficient of x_1 of the third equation [i.e. equation 4.4.1 (c)],

$$a_{31}\, x_1 + \frac{a_{12}}{a_{11}}\, a_{31}\, x_2 + \frac{a_{13}}{a_{11}}\, a_{31}\, x_3 = \frac{b_1}{a_{11}}\, a_{31} \qquad \text{... [4.4.4 (a)]}$$

Subtract this equation from third equation in the system of equations [i.e. equation 4.4.1(c)],

$$0 + \left(a_{32} - \frac{a_{12}}{a_{11}}\, a_{31}\right) x_2 + \left(a_{33} - \frac{a_{13}}{a_{11}}\, a_{31}\right) x_3 = \left(b_3 - \frac{b_1}{a_{11}}\, a_{31}\right) \qquad \text{... [4.4.4 (b)]}$$

We obtained two equations in this procedure i.e. equation 4.4.3 and equation 4.4.4, they are written below –

$$\left(a_{22} - \frac{a_{12}}{a_{11}}\, a_{21}\right) x_2 + \left(a_{23} - \frac{a_{13}}{a_{11}}\, a_{21}\right) x_3 = \left(b_2 - \frac{b_1}{a_{11}}\, a_{21}\right) \qquad \text{... [4.4.5 (a)]}$$

&

$$\left(a_{32} - \frac{a_{12}}{a_{11}}\, a_{31}\right) x_2 + \left(a_{33} - \frac{a_{13}}{a_{11}}\, a_{31}\right) x_3 = \left(b_3 - \frac{b_1}{a_{11}}\, a_{31}\right) \qquad \text{... [4.4.5 (b)]}$$

These two equations have two unknowns, x_2 and x_3.

Thus we eliminated x_1 from the equations.

Second stage of elimination :

Let's write equation 4.4.5 (a) and equation 4.4.5 (b) obtained above in shorthand notation as,

$$a'_{22}\, x_2 + a'_{23}\, x_3 = b'_2 \qquad \text{... [4.4.6 (a)]}$$

$$\& \quad a'_{32}\, x_2 + a'_{33}\, x_3 = b'_3 \qquad \text{... [4.4.6 (b)]}$$

Here, $a'_{22} = a_{22} - \dfrac{a_{12}}{a_{11}} a_{21}$, $a'_{23} = a_{23} - \dfrac{a_{13}}{a_{11}} a_{21}$, $b'_2 = b_2 - \dfrac{b_1}{a_{11}} a_{21}$

& $a'_{32} = a_{32} - \dfrac{a_{12}}{a_{11}} a_{31}$, $a'_{33} = a_{33} - \dfrac{a_{13}}{a_{11}} a_{31}$, $b'_3 = b_3 - \dfrac{b_1}{a_{11}} a_{31}$

Divide equation 4.4.6 (a) by a'_{22} i.e. coefficient of x_2 we get,

$$x_2 + \frac{a'_{23}}{a'_{22}} x_3 = \frac{b'_2}{a'_{22}} \qquad \text{... (4.4.7)}$$

Here a'_{22} should not be zero.

Multiply the above equation by coefficient of x_2 of next equation [i.e. by a'_{32} of equation 4.4.6 (b)] we get,

$$a'_{32}\, x_2 + \frac{a'_{23}}{a'_{22}} a'_{32}\, x_3 = \frac{b'_2}{a'_{22}} a'_{32}$$

Subtract this equation from equation 4.4.6 (b) we get,

$$0 + \left(a'_{33} - \frac{a'_{23}}{a'_{22}} a'_{32} \right) x_3 = \left(b'_3 - \frac{b'_2}{a'_{22}} a'_{32} \right)$$

We can write the above equation as,

$$a''_{33}\, x_3 = b''_3 \qquad \text{... (4.4.8)}$$

where $a''_{33} = a'_{33} - \dfrac{a'_{23}}{a'_{22}} a'_{32}$ & $b''_3 = b'_3 - \dfrac{b'_2}{a'_{22}} a'_{32}$

The above equation contains only one unknown i.e. x_3. Thus we eliminated x_2 from this equation. Let's take first equation from systems of equation 4.4.1, equation 4.4.6 and equation 4.4.8 i.e.,

equation 4.4.1 (a) is ⇒ $a_{11}\, x_1 + a_{12}\, x_2 + a_{13}\, x_3 = b_1$ ... [4.4.9 (a)]

equation 4.4.6 (a) is ⇒ $a'_{22}\, x_2 + a'_{23}\, x_3 = b'_2$ ... [4.4.9 (b)]

equation 4.4.8 ⇒ $a''_{33}\, x_3 = b''_3$ ... [4.4.9 (c)]

Thus we obtained upper triangular form of equations. These equations can be solved using backward substitution. i.e., From last [i.e. equation 4.4.9 (c)] equation, $a''_{33}\, x_3 = b''_3$

$$\therefore \quad x_3 = \frac{b''_3}{a''_{33}}$$

By putting this value of x_3 in equation 4.4.9 (b) we can obtain value of x_2 and so on.

The most important condition is that

$$\boxed{a_{11} \neq 0, \qquad a'_{22} \neq 0, \qquad a''_{33} \neq 0 \ldots}$$

4.4.2 Solved Examples

Ex.4.4.1 *Solve the following system of equations using Gauss elimination method –*

$$\left.\begin{aligned} 2x_1 + x_2 + x_3 &= 10 \\ 3x_1 + 2x_2 + 3x_3 &= 18 \\ x_1 + 4x_2 + 9x_3 &= 16 \end{aligned}\right\} \quad \ldots (4.4.10)$$

[Dec - 2001, 8 Marks]

Sol. : First stage of elimination :

First we will eliminate x_1 from the above system of equations. Divide first equation in equation 4.4.10 by 2 then we get,

$$x_1 + \frac{1}{2}x_2 + \frac{1}{2}x_3 = 5 \quad \ldots (4.4.11)$$

Multiply this equation by '3' and subtract from second equation of equation 4.4.10 then we get,

$$0 + \left(2 - \frac{3}{2}\right)x_2 + \left(3 - \frac{3}{2}\right)x_3 = 18 - 15$$

$$\therefore \quad \frac{1}{2}x_2 + \frac{3}{2}x_3 = 3 \quad \ldots (4.4.12)$$

Since coefficient of x_1 of third equation in equation 4.4.10 is one, subtract equation 4.4.11 directly from it.

$$\therefore \quad 0 + \left(4 - \frac{1}{2}\right)x_2 + \left(9 - \frac{1}{2}\right)x_3 = 16 - 5$$

$$\therefore \quad \frac{7}{2}x_2 + \frac{17}{2}x_3 = 11 \quad \ldots (4.4.13)$$

Second stage of elimination :

In the second stage of elimination we have two equations i.e.,

$$\left.\begin{aligned} &\text{From equation 4.4.12} \Rightarrow & \frac{1}{2}x_2 + \frac{3}{2}x_3 &= 3 \\ &\text{\& From equation 4.4.13} \Rightarrow & \frac{7}{2}x_2 + \frac{17}{2}x_3 &= 11 \end{aligned}\right\} \quad \ldots (4.4.14)$$

Now we have to eliminate x_2 from the above system of equations.

Divide first equation of equation 4.4.14 by 1/2 i.e. multiply by 2 we get,

$$x_2 + 3x_3 = 6$$

Multiply this equation by coefficient of x_2 in the second equation i.e. by 7/2 and subtract from second equation.

$$0 + \left(\frac{17}{2} - \frac{21}{2}\right)x_3 = 11 - 21$$

$$\therefore \quad -2x_3 = -10 \qquad \text{... (4.4.15)}$$

Backward substitution :

Let's form the following system of equations by taking first equation from equation 4.4.10, first equation from equation 4.4.14 and equation 4.4.15, i.e.,

$$\left.\begin{aligned} 2x_1 + x_2 + x_3 &= 10 \\ \frac{1}{2}x_2 + \frac{3}{2}x_3 &= 3 \\ -2x_3 &= -10 \end{aligned}\right\} \qquad \text{... (4.4.16)}$$

This is the system of equations in upper triangular form. By using backward substitution we can solve these equation.

The last equation is $\quad -2x_3 = -10$

$$\therefore \quad x_3 = 5$$

The second equation in equation 4.4.16 is,

$$\frac{1}{2}x_2 + \frac{3}{2}x_3 = 3$$

Putting values of $x_3 = 5$ in above equation,

$$\frac{1}{2}x_2 + \frac{3}{2} \times 5 = 3$$

$$\therefore \quad x_2 = -9$$

The first equation is,

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

Putting values of x_2 and x_3 in the above equation,

$$2x_1 - 9 + 5 = 10$$

$$\therefore \quad x_1 = 7$$

Thus we obtained $x_1 = 7$, $x_2 = -9$ and $x_3 = 5$ using Gauss elimination method.

Ex. 4.4.2 *With the help of Gauss elimination method find the solution of following system of linear equations.*

$$\left.\begin{aligned} x_1 + \frac{1}{2}x_2 + \frac{1}{3}x_3 &= 1 \\ \frac{1}{2}x_1 + \frac{1}{3}x_2 + \frac{1}{4}x_3 &= 0 \\ \frac{1}{3}x_1 + \frac{1}{4}x_2 + \frac{1}{5}x_3 &= 0 \end{aligned}\right\} \qquad \dots (4.4.17)$$

Sol. : First stage of elimination :

Multiply first equation by $\frac{1}{2}$ and subtract from second equation then we get,

$$\frac{1}{12}x_2 + \frac{1}{12}x_3 = -\frac{1}{2}$$

Then multiply first equation by $\frac{1}{3}$ and subtract from third equation then we get,

$$\frac{1}{12}x_2 + \frac{4}{45}x_3 = -\frac{1}{3}$$

Thus we obtained two equations as,

$$\frac{1}{12}x_2 + \frac{1}{12}x_3 = -\frac{1}{2}$$

&

$$\frac{1}{12}x_2 + \frac{4}{45}x_3 = -\frac{1}{3} \qquad \dots (4.4.18)$$

In these equations variable x_1 is eliminated.

Second stage of elimination :

Since coefficients of x_2 in equation 4.4.18 are equal, subtract first equation from second equation. Then we get,

$$\frac{1}{180}x_3 = \frac{1}{6} \qquad \dots (4.4.19)$$

Backward substitution :

Taking the first equations in equation 4.4.17, equation 4.4.18 and equation 4.4.19 we get,

$$\begin{aligned} x_1 + \frac{1}{2}x_2 + \frac{1}{3}x_3 &= 1 \\ \frac{1}{12}x_2 + \frac{1}{12}x_3 &= -\frac{1}{2} \\ \frac{1}{180}x_3 &= \frac{1}{6} \end{aligned} \qquad \dots (4.4.20)$$

This is the system of equation in upper triangular form and can be solved using backward substitution.

From last equation in above equations we obtain,

$$\frac{1}{180} x_3 = \frac{1}{6}$$

$$\therefore \quad x_3 = 30$$

Putting this value in second equation we obtain

$$x_2 = -36$$

and putting x_2 and x_3 in first equation we obtain $x_1 = 9$.

Thus from gauss elimination method we obtained

$x_1 = 9, \quad x_2 = -36 \quad$ and $\quad x_3 = 30$

Ex. 4.4.3 *Solve the following system of equations using Gauss elimination method.*

$$2x + y + 4z = 16$$

$$3x + 2y + z = 10$$

$$x + 3y + 3z = 16$$

[Dec - 2003, 8 Marks]

Sol. : The given system of equation can be solved using Gauss elimination method as discussed in earlier examples. The solution is;

x = 1, y = 2 and z = 3

Exercise

1. Solve the following system of equation using Gauss elimination method.

$$2x + y - 0.1z + t = 2.7$$

$$0.4x + 0.5y + 4z - 8.5t = 21.9$$

$$0.3x - y + z + 5.2t = -3.9$$

$$x + 0.2y + 2.5z - t = 9.9$$

[Ans. : x = 1, y = 2, z = 3 and t = –1]

2. Solve the following equations using Gauss elimination method.

$$2x + y + 4z + 7t = 1$$

$$-4x + y - 6z - 13t = -1$$

$$4 + 5y + 7z + 7t = 4$$

$$-2x + 5y - 4z - 8t = -5$$

[Ans. : x = 2, y = – 1, z = 3 and t = –2]

4.4.3 Pitfalls of Elimination Methods

Gauss elimination technique is powerful in solving many systems of equations. But there are some pitfalls in this technique.

4.4.3.1 Division by Zero

Consider the system of equations given below :

$$\begin{aligned} 3x_2 - x_3 &= 8 \\ x_1 - 3x_2 + 4x_3 &= 5 \\ -x_1 + 2x_2 + 2x_3 &= -4 \end{aligned}$$

Here observe that $a_{11} = 0$ (first equation). Hence divide by zero will occur and gauss elimination method cannot be applied to these equations. Hence let us exchange the first and the third equation i.e.,

$$\begin{aligned} -x_1 + 2x_2 + 2x_3 &= -4 \\ x_1 - 3x_2 + 4x_3 &= 5 \\ 3x_2 - x_3 &= 8 \end{aligned}$$

Now $a_{11} = -1$ in the above system of equations and gauss elimination method can be applied here. Sometimes a_{11} is very very small. Then similar problems may occur. This can be avoided by partial pivoting. This technique is discussed next.

4.4.3.2 Round-off Errors

We know that computers handle finite number of significant digits. Hence numbers are always rounded-off during calculations. This introduces errors and they are propagated in the results.

4.4.3.3 ILL Conditioned Systems

When there is a very small change in the coefficients of the system, it may produce large changes in the solution. Such systems are said to be ill conditioned systems.

Consider for example,

$$\begin{aligned} x + 5y &= 17 \\ 1.5x + 7.501y &= 25.503 \end{aligned}$$

The solution of these equations can be easily calculated. i.e. x = 2 and y = 3.

Let there be a very small change in the second equation such that,

$$\begin{aligned} x + 5y &= 17 \\ 1.5x + 7.501y &= 25.5 \quad \text{This value is changed by 0.003.} \end{aligned}$$

Then solution of these equations will be,

$$x = 17 \quad \text{and} \quad y = 0 \text{ !}$$

Thus with a very small change in coefficient the solution is changed from (2, 3) to (17 , 0). These equations are then called ill conditioned.

As discussed above, a very small change in the number makes the system ill conditioned. Such small changes in numbers occur because of rounding errors.

4.4.3.4 Singular Systems

Let us consider the system of 'n' equations having 'n' number of unknowns. If two equations are identical, then there will be (n - 1) equations with 'n' number of unknowns. Such equations cannot be solved. These equations are called singular systems.

4.4.4 Techniques for Improving Solutions

Now let us discuss some techniques to overcome the pitfalls of gauss elimination method.

4.4.4.1 Use of More Significant Digits

Ill conditioned systems can be effectively handled with more significant digits. Using more significant digits make the solutions accurate. But memory requirement and computational complexity increases due to more significant digits.

4.4.4.2 Scaling

Scaling of the coefficients is performed in order to standardize the size of the determinant. When coefficients of some of the equations are very large compared to others, then scaling is done. Such scaling also reduces round-off errors. And hence error in the solution is reduced.

4.4.4.3 Partial Pivoting in Gauss Elimination Method

In the basic Gauss elimination method we divide the equations by a_{11}, a'_{22}, a''_{33} and so on to eliminate variables in successive stages.

These elements (i.e. a_{11}, a'_{22}, a''_{33}... etc) are called **"pivots"**. If any pivot becomes zero, then we rearrange equations. If any pivot becomes very very small, then division by this pivot is equivalent to multiplication by a very large number. This may create errors in the calculations. Equations are rearranged to avoid these problems. This is called **"Pivoting"**.

Partial Pivoting :

"The first element of the first equation is normally the pivot in that particular elimination stage. The equations are rearranged so that this pivot is always largest among all the elements in the first column." This is called partial pivoting. It is called "partial" because pivoting is performed in every elimination stage.

The strategy involved partial pivoting is to divide equations by a large element. Let's see the concept with the help of an example.

Ex. 4.4.4 *Solve the following system of equations using partial pivoting (Gauss elimination method).*

$$\left.\begin{aligned} x_1 + x_2 + x_3 &= 6 \\ 3x_1 + 3x_2 + 4x_3 &= 20 \\ 2x_1 + x_2 + 3x_3 &= 13 \end{aligned}\right\} \quad \dots (4.4.21)$$

Sol. : First elimination stage

Partial Pivoting : First we will check for largest element among first column (coefficients of x_1). Second equation has '3' which is largest. Therefore interchanging first and second equation we get new system of equations as follows :

$$\left.\begin{aligned} 3x_1 + 3x_2 + 4x_3 &= 20 \\ x_1 + x_2 + x_3 &= 6 \\ 2x_1 + x_2 + 3x_3 &= 13 \end{aligned}\right\} \qquad \text{... (4.4.22)}$$

Here pivoting element (i.e. '3') is now largest among elements in first column.

Divide first equation by '3' (i.e. pivot), we get,

$$x_1 + x_2 + \frac{4}{3}x_3 = \frac{20}{3} \qquad \text{... (4.4.23)}$$

Subtract this equation from second equation, then we get,

$$0 + 0 + \left(1 - \frac{4}{3}\right)x_3 = \left(6 - \frac{20}{3}\right)$$

$$\therefore \qquad -\frac{1}{3}x_3 = -\frac{2}{3}$$

i.e. $$x_3 = 2 \qquad \text{... (4.4.24)}$$

Multiply equation 4.4.23 by 2 and subtract from third equation of equation 4.4.22. i.e.,

$$0 + (1-2)\,x_2 + \left(3 - \frac{8}{3}\right)x_3 = 13 - \frac{40}{3}$$

i.e. $$-x_2 + \frac{1}{3}x_3 = -\frac{1}{3} \qquad \text{... (4.4.25)}$$

Thus we obtained two new equations in this stage as,

From equation 4.4.24 $\Rightarrow$ $\quad 0\,x_2 + x_3 = 2$ $\quad$ (by writing '0' as coefficient of x_2)

From equation 4.4.25 $\Rightarrow$ $\quad -x_2 + \frac{1}{3}x_3 = -\frac{1}{3}$ $\qquad$ ... (4.4.26)

In these equations first variable, i.e. x_1 is eliminated.

Second elimination stage :

In this stage we will eliminate variable x_2.

Partial pivoting : In this stage the new equations are as given by equation 4.4.26. Here the larger element in the first column is – 1 and it is in second equation. [Note : Larger element means we see magnitude]. Therefore we exchange first and second equations in system of equation 4.4.26, i.e.,

$$-x_2 + \frac{1}{3}x_3 = -\frac{1}{3}$$

$$0\,x_2 + x_3 = 2 \qquad \text{... (4.4.27)}$$

Divide first equation by – 1, then we get,

$$x_2 - \frac{1}{3} x_3 = \frac{1}{3} \qquad \text{... (4.4.28)}$$

Multiply this equation by zero and subtract from second equation. [Note : Normally we multiply equation 4.4.28 by coefficient of x_2 in second equation and then subtract from it to eliminate x_2 in second equation].

$\therefore$ We get, $\quad (0-0)\, x_2 + (1-0)\, x_3 = 2 - 0$

$$\therefore \qquad x_3 = 2 \qquad \text{... (4.4.29)}$$

[* Observe that the above equation can be directly obtained from equation 4.4.27. But we have done all the above procedure to illustrate the method].

By taking the first equations in equation 4.4.22, equation 4.4.27 & equation 4.4.29,

$$\begin{aligned} 3x_1 + 3x_2 + 4x_3 &= 20 \\ -x_2 + \frac{1}{3} x_3 &= -\frac{1}{3} \\ x_3 &= 2 \end{aligned} \qquad \text{... (4.4.30)}$$

Backward substitution :

The system of equations obtained above in equation 4.4.30 is in the upper triangular form. These equations can be solved using backward substitution.

From last equation we obtain,

$$x_3 = 2$$

Putting this value in second equation we obtain,

$$-x_2 + \frac{1}{3} \times 2 = -\frac{1}{3}$$

$$\therefore \qquad x_2 = 1$$

Putting for x_3 and x_2 in first equation of equation 4.4.30,

$$3x_1 + 3 \times 1 + 4 \times 2 = 20$$

$$\therefore \qquad x_1 = 3$$

Thus the values of variables are,

$x_1 = 3, \quad x_2 = 1 \quad \& \quad x_3 = 2$

Ex. 4.4.5 : *Highlight the importance of sensitivity by comparing solution of two systems of equations given below :*

System-I

$$\begin{aligned} x_1 + 2x_2 &= 10 \\ 1.1 x_1 + 2x_2 &= 10.4 \end{aligned}$$

System-II

$$\begin{aligned} x_1 + 2x_2 &= 10 \\ 1.05 x_1 + 2x_2 &= 10.4 \end{aligned}$$

[May-96, 10 Marks]

Sol. : System-I

The given equations are,

$$\begin{aligned} x_1 + 2x_2 &= 10 \\ 1.1 x_1 + 2x_2 &= 10.4 \end{aligned}$$

Multiply first equation by 1.1 and subtract from the second,

$$-0.2 x_2 = -0.6$$

$$\therefore \quad x_2 = 3$$

Putting this value in first equation and solving we get,

$$x_1 + 2(3) = 10$$

$$\therefore \quad x_1 = 4$$

Thus we obtained, $\boxed{x_1 = 4, \quad x_2 = 3}$

System-II : The given equations are :

$$x_1 + 2x_2 = 10$$
$$1.05x_1 + 2x_2 = 10.4$$

Multiply first equation by 1.05 and subtract from second,

$$-0.1x_2 = -0.1$$

$$\therefore \quad x_2 = 1$$

Putting this value in the first equation,

$$x_1 + 2(1) = 10$$

$$\therefore \quad x_1 = 8$$

Thus we obtained, $\boxed{x_1 = 8, \quad x_2 = 1}$

The equations in system–I and II are same except a_{21}. In system-I, $a_{21} = 1.1$. In system-II, it is changed slightly i.e. = $a_{21} = 1.05$. But this makes a large change in the solutions of two systems. Here note that the two equations are almost similar. Hence solutions of such equations are highly sensitive to coefficient values.

Exercise

1. Solve the following system of equations using partial pivoting.

$$x_1 - 2x_2 + 3x_3 + 9x_4 = 5$$
$$3x_1 + 10x_2 + 4x_3 + 2x_4 = 7$$
$$11x_1 + 5x_2 + 9x_3 + 2x_4 = 13$$
$$2x_1 + 3x_2 + 7x_3 + 6x_4 = 11$$

[Hint : In the first stage of elimination, interchange first and third equations and so on. **[May-2001]**

[Ans. : $x_1 = -0.078431$, $x_2 = 0.122549$, $x_3 = 1.448039$ and $x_4 = 0.108824$]

2. Solve the following system of equations using partial pivoting.

$$10x_1 - x_2 + 2x_3 = 4$$
$$x_1 + 10x_2 - x_3 = 3$$
$$2x_1 + 3x_2 + 20x_3 = 7$$

[Ans. : $x_1 = 0.375$, $x_2 = 0.289$ and $x_3 = 0.269$]

4.4.5 C Program, Algorithm and Flowchart for Gauss Elimination Method

Now we will see the programming logic of gauss elimination method. First we will see algorithm.

Algorithm :

Step 1 : *Read the number of variables 'n' in the system of equations. (n = number of variables = number of equations).*

Step 2 : *Read the coefficients of equations i.e.*

Read a_{ij} *for* $i = 1$ *to* n

& $j = 1$ *to* n

& *Read* b_i *for* $i = 1$ *to* n

Step 3 : (i) *Eliminate variable* x_1 *from all the equations and form new set of* $(n - 1)$ *equations.*

(ii) *Eliminate variable* x_2 *from this newly formed set of equations and create new set of* $(n - 2)$ *equations.*

(iii) *Eliminate variables* $x_3, x_4 \ldots\ldots$ *and so on until there is only one equation generated with single variable. This requires* $(n - 1)$ *elimination steps.*

Step 4 : *Use backward substitution starting from last equation obtained and find out values of variables* $x_1, x_2, x_3 \ldots\ldots x_n$.

Step 5 : *Display the values of calculated variables, i.e. Display* x_i, $i = 1, 2, 3, \ldots n$.

Step 6 : *Stop.*

Flowchart :

Based on the above algorithm, we will prepare the flowchart for gauss elimination method as shown in Fig. 4.4.1. This flowchart shows only important blocks. The detailed logic to generate upper triangular matrix and backward substitution will be discussed in the program. (See Fig. 4.4.1 on next page)

Computer Program :

The 'C' program to implement gauss elimination method is shown below.

```
/* Download this program from  www.vtubooks.com                          */
/* File name : g_elimn.cpp                                                */

/*-------- GAUSS ELIMINATION METHOD TO SOLVE LINEAR EQUATIONS ---------*/

/*      THE PROGRAM SOLVES THE SYSTEM OF LINEAR EQUATIONS USING

        GAUSS ELIMINATION METHOD.

        INPUTS :  1) Number of variables in the equation.
                  2) Coefficient's of linear equations.
        OUTPUT  : Calculated values of x1,x2,x3,...,xn etc.             */

/*-------------------------------    PROGRAM  --------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>

void main()
{
      double a[10][10],x[10],ratio,coefsum;
                /* ARRAY OF a[n][n] STORING COEFFICIENTS OF EQUATIONS */
      int i,j,n,k;

      clrscr();

      printf("\n     GAUSS ELIMINATION METHOD TO SOLVE LINEAR EQUATIONS");

      printf("\n\n              The form of equations is as follows\n\n"
```

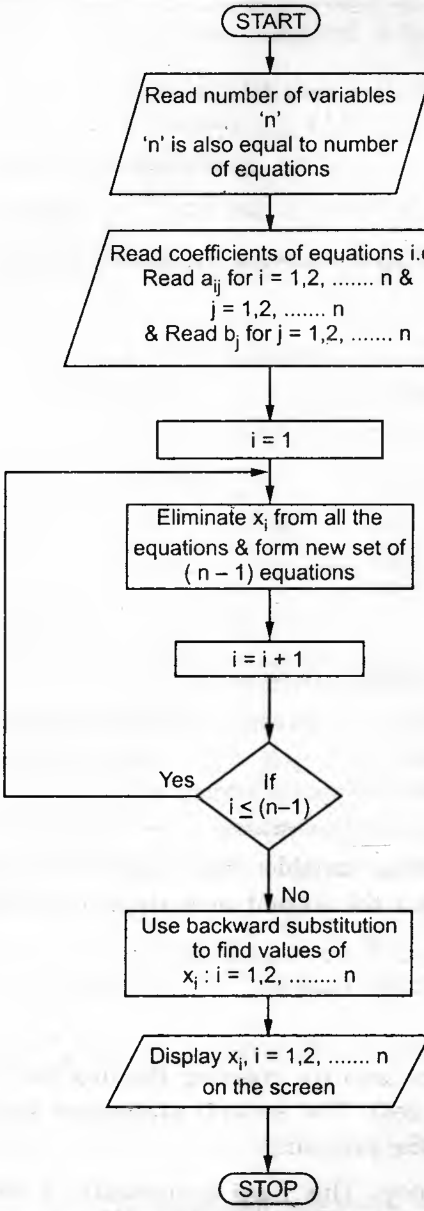


Fig. 4.4.1 Flowchart for Gauss elimination method

```
        "                   a11x1 + a12x2 + ... + a1nxn = b1\n"
        "                   a21x1 + a22x2 + ... + a2nxn = b2\n"
        "                   a31x1 + a32x2 + ... + a3nxn = b3\n"
        "                   ..............................\n"
        "                   an1x1 + an2x2 + ... + annxn = bn\n"
        "\n\nHere a11,a22,a33,a44,.....etc.  should not be zero\n");

printf("\n\nEnter the number of variables (max 10) = ");
                /* ENTER THE NUMBER OF VARIABLES IN THE EQUATION    */
scanf("%d",&n);

for(i = 1; i <= n; i++)
```

```
{           /*  LOOP TO GET COEFFICIENTS a11,a12...,ann & so on  */

    for(j = 1; j <= n; j++)
    {
        printf("a%d%d = ",i,j);  scanf("%lf",&a[i][j]);
    }
    printf("b%d = ",i);  scanf("%lf",&a[i][j]);
    x[i] = 0;
}

for(k = 1; k <= n-1; k++)
{
      /*  LOOP TO GENERATE UPPER TRIANGULAR SYSTEM  */

    for(i = k+1; i <= n; i++)
    {
        ratio = a[i][k]/a[k][k];

        for(j = k+1; j <= n+1; j++)
        {
            a[i][j] = a[i][j] - ratio * a[k][j];
        }
    }
    for(i = k+1; i <= n; i++)  a[i][k] = 0;
}
x[n] = a[n][n+1]/a[n][n];

for(i = n-1; i >= 1; i--)
{
                                /* LOOP FOR BACKWARD SUBSTITUTION  */
     coefsum = 0;
     for(j = i+1; j <= n; j++)    coefsum = coefsum + a[i][j] * x[j];
     x[i] = (a[i][n+1] - coefsum)/a[i][i];
}
printf("\n\nThe values of variables in the given equations are "
       "as follows....\n");
for(i = 1; i <= n; i++)      printf("\n   x%d = %lf   ",i,x[i]);
      /* LOOP TO PRINT VALUES OF x1,x2,...xn etc  */
}
/*-------------------------------- END OF PROGRAM ----------------------*/
```

The program first declares variables then it prints the names of the method. The third printf statement displays the format of system of equations. The next statement is,

```
printf  ("\n\n enter the number of variables  (max  10)=");
scanf  ("%d",  &n);
```

Here the first statement asks for entering the number of simultaneous equations (i.e. also number of variables). The second statement gets this number and it is assigned to variable `'n'` in the program.

Next there is a `for` loop. This loop is reproduced below for explanation. The given system of equations is represented in matrix form.

```
for  (i=1;  i<=n;  i++)          ← This loop is for rows of matrix aij
{
    for  (j=1;  j<=n;  j++)      ← This loop is for columns of matrix aij
                                   i.e. for equations.
    {
     printf ("a%d %d=", i, j,); scanf("%lf, &a[i][j]);
```

These two state-ments get a_{ij} element in the array a[i][j].

```
}
printf ("b%d=", i); scanf ("%lf",&a[i][j]);
```

These two statements get $b_1, b_2, b_3 \ldots b_n$ in the $(n+1)^{th}$ column of array 'a'.

`x[i]=0;` ← This statements clears array x [ ] for next computations.

```
}
```

The format of the matrix is shown below.

$$
\texttt{a[i][j]} = \begin{array}{c}
\begin{bmatrix}
j=1 & j=2 & j=3 & \ldots\ldots & j=n & j=n+1 \\
\downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\
a_{11} & a_{12} & a_{13} & \ldots\ldots & a_{1n} & b_1 \\
a_{21} & a_{22} & a_{23} & \ldots\ldots & a_{2n} & b_2 \\
a_{31} & a_{32} & a_{33} & \ldots\ldots & a_{3n} & b_3 \\
\vdots & \vdots & \vdots & & \vdots & \vdots \\
\vdots & \vdots & \vdots & & \vdots & \vdots \\
a_{n1} & a_{n2} & a_{n3} & \ldots\ldots & a_{nn} & b_n
\end{bmatrix}
\end{array}
\begin{array}{l}
\\ \\ \leftarrow i=1 \\ \leftarrow i=2 \\ \leftarrow i=3 \\ \\ \leftarrow i=4 \\ \\
\end{array}
$$

Array `a[i][j]` which is used for storing a_{ij} and b_j in the program, its structure is shown in the matrix form as above. It is clear that outer `for` loop is used for equations and inner `for` loop is used for coefficients of equations. The values of b_1, b_2, b_3, etc. are stored in $(n+1)^{th}$ column of the matrix 'a'. The next `for` loop eliminates variables $x_1, x_2, \ldots. x_{n-1}$ in successive stages and generates an upper triangular matrix. This loop is reproduced below for convenience.

`for (k=1; k<=n-1; k++)` ← This loop is used for stages of elimination.

```
{
```

`for(i=k+1;i<=n;i++)` ← This loop acts as counter for number of variables.

```
{
```

`ratio=a[i][k]/a[k][k];` ← In the first stage of elimination (when k=1) this statement generates $\frac{a_{12}}{a_{11}}, \frac{a_{13}}{a_{11}}, \ldots \& \frac{b_1}{a_{11}}$. In the second stage it generates $\frac{a'_{23}}{a'_{22}}, \frac{a'_{24}}{a'_{22}}, \ldots \& \frac{b'_2}{a'_{22}}$ & so on.

`for (j=k+1;j<=n+1;j++)` ← This loop subtracts equations.

```
{
```

```
        a[i][j]=a[i][j]-ratio*a[k][j];
```

In the first stage of elimination
this statement generates

$$\left(a_{22}-\frac{a_{12}}{a_{11}}a_{21}\right),\left(a_{23}-\frac{a_{13}}{a_{11}}a_{21}\right)..$$

In the second stage it generates,

$$\left(a'_{33}-\frac{a'_{23}}{a'_{22}}a'_{32}\right)....$$

```
      }
   }
   for  (i=k+1;  i<=n;  i++)  a[i][k]=0;
```

This statement makes eliminated variables, coefficients zero.
That is in first stage, $a_{21} = a_{31} = \ldots a_{n1} = 0$.
In the second stage, $a'_{32} = a'_{42} = \ldots a'_{n2} = 0$.

```
}
```

Let's see how the loop works. The systematic flow of program is shown in the flow diagram below.

Flow Diagram :

$\boxed{\text{First Loop } k = 1 \text{ to } n-1}$

k = 1 ⇒ $\boxed{\text{Second Loop } i = k + 1 \text{ to } n}$

i = 2, k = 1

```
ratio = a[i][k]/a[k][k];
      =  a[2][1]/a[1][1]
```

$= \dfrac{a_{21}}{a_{11}}$ ⇒ $\boxed{\text{Third loop } j=k+1 \text{ to } n+1}$

k=1, i=2, j=2

```
a[i][j]=a[i][j]
-ratio*a[k][j];
```

This statement calculates
$a'_{22}, a'_{23}, a'_{24} \ldots b'_2$

$\boxed{\text{Second loop } i = k+1 \text{ to } n}$ ⇐ The program comes out of this loop when j > n + 1

i++ gives i = 3 & k = 1

ratio = a[i][k]/a[k][k]

= a[3][1]/a[1][1]

$= \frac{a_{31}}{a_{11}} \Rightarrow$ Third loop j = k + 1 to n + 1

This statement calculates $a'_{32}, a'_{33}, a'_{34} \ldots b'_3$ & so on.

Second loop i = k+1 to n $\Leftarrow$

i++ gives i = 4 & k = 1

:

:

Program comes out of this loop

First loop k = 1 to n − 1 $\Leftarrow$ when i > n (Here x_1 is eliminated from all equations)

k++ gives k = 2

$\Rightarrow$ Second loop i = k+1 to n

i = 3, k = 2

ratio = a[i][k]/a[k][k]

= a[3][2]/a[2][2]

$= \frac{a'_{32}}{a'_{22}} \Rightarrow$ Third loop j = k + 1 to n + 1

k=2, i=3, j=3,

```
a[i][j]=a[i][j]
-ratio*a[k][j];
```

This statement calculates $a''_{33}, a''_{34}, \ldots b''_3$ & so on.

The program then comes out of the loop when $j > n+1$

Second loop i = k+1 to n $\Leftarrow$

i++ gives i = 4 & k=2

ratio = $\frac{a'_{42}}{a'_{22}}$ ⇒ Third loop
j = k + 1 to n + 1

k=2, i=4, j=3

Calculation of

a''_{43}, a''_{44} & so on.

Second loop
i = k+1 to n ⇐

i++ gives i = 5 & k=2

:

:

Program comes out of this loop

First loop
k = 1 to n – 1 ⇐ when i > n (Here x_2 is eliminated
from all equations)

k++ gives k = 3

⇒ ... repeat till (n – 1) variables are eliminated

After coming out of this `for` loop, the next statement in the program is,

```
x[n] = a[n][n+1]/a[n][n];
```

This statement calculates x_n from the last equation. This is first execution in backward substitution. We know that upto last equation (n–1) variables are eliminated.

Here, `x[n]` = x_n (last variable in last equation)

`a[n][n+1]` = b_n : constant term on R.H.S. of last equation

& `a[n][n]` = a_{nn} : coefficient of x_n

Next, there is a `for` loop which performs remaining backward substitutions to find other variables. This loop is reproduced below for explanation.

```
for (i=n-1; i>=1; i--) ←
{
```

This is the loop for variables starting from second last variable i.e. $x_{n-1}, x_{n-2} \dots x_2, x_1$.

```
    coefsum=0;
  for (j=i+1,j<=n; j++)    coefsum=coefsum+a[i][j]*x[j];
```

Let's say that n = 3 and x_3 and x_2 are calculated.

Then this loop calculates coefsum = $a_{12}\,x_2 + a_{13}\,x_3$.

```
  x[i]=(a[i][n+1] - coefsum)/a[i][i];
```

This statement calculates value of variable.

```
}
```

This loop goes from second last equation to first equation. It performs backward substitution. Consider the first equation in the system of three equations.

$$a_{11}\, x_1 + a_{12}\, x_2 + a_{13}\, x_3 = b_1$$

Let x_2 and x_3 are calculated by this `for` loop.

Then the statement,

```
for  (j=i+1;  j<=n;j++)  coefsum=coefsum+a[i][j]*x[j];
```

This is actually a `for` loop with one statement. When i = 1 and n = 3 it calculates,

`coefsum` $= a_{12}\, x_2 + a_{13}\, x_3$ in the above stated condition.

The next statement is,

```
x[i]  =  (a[i][n+1]  -  coefsum)/a[i][i];
```

When i = 1 and n = 3 as above it calculates,

$$x_1 = \frac{\texttt{a[1][4]} - \texttt{coefsum}}{\texttt{a[1][1]}} = \frac{b_1 - (a_{12}\, x_2 + a_{13}\, x_3)}{a_{11}}$$

$\because$ We have `a[1][4]` $= b_1$

Thus we obtain solution to the system of equations.

How to Run this program?

Compile the 'C' program given here and make its EXE file.

We will use the system of equations of example 4.4.1 for the program. This system of equations is given below for convenience,

$$2x_1 + x_2 + x_3 = 10$$
$$3x_1 + 2x_2 + 3x_3 = 18$$
$$x_1 + 4x_2 + 9x_3 = 16$$

Here n = 3,

$a_{11} = 2,$	$a_{12} = 1,$	$a_{13} = 1,$	$b_1 = 10.$
$a_{21} = 3,$	$a_{22} = 2,$	$a_{23} = 3,$	$b_2 = 18$
& $a_{31} = 1,$	$a_{32} = 4,$	$a_{33} = 9,$ &	$b_3 = 16.$

After running the program it displays the name of the method and displays the format of system of equations. i.e.,

The form of equations is as follows :

```
a11 x1+ a12 x2+ . . . .+ a1n xn = b1
a21 x1+ a22 x2+ . . . .+ a2n xn = b2
a31 x1+ a32 x2+ . . . .+ a3n xn =  b3
.. .. .. .. .. .. .. .. .. ..

an1 x1+ an2 x2+ . . . .+ ann xn = bn
```

This defines all the notations of coefficients used by the program.
Then the program displays,
`Enter the number of variables (max 10) =`
Since we have n = 3, here enter number of variables and '3' press enter key.
Then it displays,

a11 =	Here enter `2' and press `enter' key
a12 =	1 ↵
a13 =	1 ↵
b1 =	10 ↵
a21 =	:
:	:
:	: Similarly enter other values as given above
b3 =	16 ↵ etc.

Then the program displays values of x_1, x_2 and x_3. Here is the complete display of the result.

```
------------------------------- Results ----------------------------

        GAUSS ELIMINATION METHOD TO SOLVE LINEAR EQUATIONS

             The form of equations is as follows

                 a11x1 + a12x2 + ... + a1nxn = b1
                 a21x1 + a22x2 + ... + a2nxn = b2
                 a31x1 + a32x2 + ... + a3nxn = b3
                 ................................
                 an1x1 + an2x2 + ... + annxn = bn

Here a11,a22,a33,a44,.....etc.  should not be zero

Enter the number of variables (max 10) = 3
a11 = 2
a12 = 1
a13 = 1
b1 = 10
a21 = 3
a22 = 2
a23 = 3
b2 = 18
a31 = 1
a32 = 4
a33 = 9
b3 = 16

The values of variables in the given equations are as follows....

   x1 = 7.000000
   x2 = -9.000000
   x3 = 5.000000
-------------------------------------------------------------------
```

Observe that results obtained by computer are same as that we had obtained in example 4.4.1.

University Questions

1. Currents in a circuit are given by the following equations :

$$
\begin{aligned}
28I_1 - 3I_2 &= 10;\\
-3I_1 + 38I_2 - 10I_3 - 5I_5 &= 0;\\
-10I_2 + 25I_2 - 15I_4 &= 0;\\
-15I_3 + 45I_4 &= 0;\\
-5I_2 + 30I_5 &= 0.
\end{aligned}
$$

   Estimate the currents using Gauss elimination method. **[Dec - 95, May - 99]**

2. Highlight the importance of sensitivity by comparing the solution of the two system of equations given below :

   System I
$$
\begin{aligned}
x_1 + 2x_2 &= 10\\
1.1x_1 + 2x_2 &= 10.4
\end{aligned}
$$
   System II
$$
\begin{aligned}
x_1 + 2x_2 &= 10\\
1.05x_1 + 2x_2 &= 10.4
\end{aligned}
$$
   [May - 96]

3. Write a detailed note on : Pivoting methods employed in triangularisation of matrices. **[Dec - 97, May - 2000]**

4. Write C procedure to find the inverse of a matrix using Gauss Elimination method and write a program to solve a system of n simultaneous equations using this procedure. **[May - 2001]**

5. Solve the following system

$$
\begin{aligned}
2x + y + z &= 10\\
3x + 2y + 3z &= 18\\
x + 4y + 9z &= 16
\end{aligned}
$$

   using Gauss Elimination method **[Dec - 2001]**

6. Write C propram for the above method and explain all the steps clearly. **[Dec - 2001]**

7. Write a program in C/C + + for Gauss Elimination with partial pivoting. **[Dec - 2002]**

8. Write the algorithm for Gauss Elimination method. **[May-2003]**

9. Using Gauss Elimination method solve the following :

$$
\begin{aligned}
3x_1 - 0.1x_2 - 0.2x_3 &= 7.85\\
0.1x_1 - 7x_2 - 0.3x_3 &= -19.3\\
0.3x_1 - 0.2x_2 + 10x_3 &= 71.4
\end{aligned}
$$

   [May - 2003]

10. Solve the following system of equations using Gauss Elimination method. **[Dec - 2003]**

$$
\begin{aligned}
2x + y + 4z &= 16\\
3x + 2y + z &= 10\\
x + 3y + 3z &= 16
\end{aligned}
$$

4.5 Gauss Jordan Elimination Method

Consider the system of equations having four unknowns,

$$\left.\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 &= b_2 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 &= b_3 \\
a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 &= b_4
\end{aligned}\right\} \quad \text{... (4.5.1)}$$

Let's represent this system in

$$AX = B \qquad \text{Format of matrices.}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \quad \text{... (4.5.2)}$$

The coefficient matrix 'A' is reduced to a diagonal matrix rather than triangular matrix. Thus in this method all the elements in matrix 'A' are eliminated except diagonal elements. The solution is then directly obtained. The equation 4.5.2 is thus transformed as follows

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} \quad \text{... (4.5.3)}$$

Equation 4.5.3 can be written as,

$$IX = Z \quad \text{... (4.5.4)}$$

Here I is unit matrix having only diagonal elements equal to one and all other elements are zero. Z is the matrix formed due to row transformations.

By the property of unit matrix we know that

$$IX = X$$

$\therefore$ Equation 4.5.4 becomes,

$$X = Z \quad \text{... (4.5.5)}$$

i.e.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix}$$

$\therefore$ We obtain,

$x_1 = z_1, \quad x_2 = z_2, \quad x_3 = z_3 \;\&\; x_4 = z_4$

Procedure to perform matrix A to unit matrix.

Here first form the augmented matrix by adding the column of matrix B to the matrix A. This augmented matrix is denoted by [A | B] i.e.,

$$[A|B] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ a_{21} & a_{22} & a_{23} & a_{24} & b_2 \\ a_{31} & a_{32} & a_{33} & a_{34} & b_3 \\ a_{41} & a_{42} & a_{43} & a_{44} & b_4 \end{array}\right] \quad \text{... (4.5.6)}$$

↑

Column of 'B' matrix is added to A.

Perform elementary row operations on this matrix such that elements of A matrix forms unit matrix I. This is shown below.

$$[I|Z] = \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & z_1 \\ 0 & 1 & 0 & 0 & z_2 \\ 0 & 0 & 1 & 0 & z_3 \\ 0 & 0 & 0 & 1 & z_4 \end{array}\right] \quad \text{... (4.5.7)}$$

The last column in above matrix is then the solution of the equations.

Thus gauss jordan method gives,

$$[A \mid B] \xrightarrow{\text{Gauss Jordan}} [I \mid Z] \quad \text{... (4.5.8)}$$

This method we have explained for 4 variables. But the same method can be extended to any number of variables.

4.5.1 Solved Examples

Ex.4.5.1 *Solve the following system of equations using gauss jordan method.*

$$\begin{aligned} 2x_1 + x_2 + 2x_3 + x_4 &= 6 \\ 6x_1 - 6x_2 + 6x_3 + 12x_4 &= 36 \\ 4x_1 + 3x_2 + 3x_3 - 3x_4 &= -1 \\ 2x_1 + 2x_2 - x_3 + x_4 &= 10 \end{aligned}$$

Sol. : Let's represent this system in matrix form

$$AX = B$$

i.e.

$$\begin{bmatrix} 2 & 1 & 2 & 1 \\ 6 & -6 & 6 & 12 \\ 4 & 3 & 3 & -3 \\ 2 & 2 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 6 \\ 36 \\ -1 \\ 10 \end{bmatrix}$$

Let's form the augmented matrix by adding column of 'B' to 'A' i.e.

$$[A|B] = \left[\begin{array}{cccc|c} 2 & 1 & 2 & 1 & 6 \\ 6 & -6 & 6 & 12 & 36 \\ 4 & 3 & 3 & -3 & -1 \\ 2 & 2 & -1 & 1 & 10 \end{array}\right]$$

By performing $\frac{1}{2}R_1$ we get,

$$\sim \left[\begin{array}{cccc|c} 1 & ½ & 1 & ½ & 3 \\ 6 & -6 & 6 & 12 & 36 \\ 4 & 3 & 3 & -3 & -1 \\ 2 & 2 & -1 & 1 & 10 \end{array}\right]$$

By $R_2 - 6R_1$, $R_3 - 4R_1$, and $R_4 - 2R_1$ we get,

$$\sim \left[\begin{array}{cccc|c} 1 & ½ & 1 & ½ & 3 \\ 0 & -9 & 0 & 9 & 18 \\ 0 & 1 & -1 & -5 & -13 \\ 0 & 1 & -3 & 0 & 4 \end{array}\right]$$

By $-\frac{1}{9}R_2$ we get,

$$\sim \left[\begin{array}{cccc|c} 1 & ½ & 1 & ½ & 3 \\ 0 & 1 & 0 & -1 & -2 \\ 0 & 1 & -1 & -5 & -13 \\ 0 & 1 & -3 & 0 & 4 \end{array}\right]$$

By $R_1 - \frac{1}{2}R_2$ and $R_4 - R_3$ we get,

$$\sim \left[\begin{array}{cccc|c} 1 & 0 & 1 & 1 & 4 \\ 0 & 1 & 0 & -1 & -2 \\ 0 & 1 & -1 & -5 & -13 \\ 0 & 0 & -2 & 5 & 17 \end{array}\right]$$

By $R_3 - R_2$ we get,

$$\sim \left[\begin{array}{cccc|c} 1 & 0 & 1 & 1 & 4 \\ 0 & 1 & 0 & -1 & -2 \\ 0 & 0 & -1 & -4 & -11 \\ 0 & 0 & -2 & 5 & 17 \end{array}\right]$$

By $-R_3$ we get,

$$\sim \left[\begin{array}{cccc|c} 1 & 0 & 1 & 1 & 4 \\ 0 & 1 & 0 & -1 & -2 \\ 0 & 0 & 1 & 4 & 11 \\ 0 & 0 & -2 & 5 & 17 \end{array}\right]$$

By $R_1 - R_3$ and $R_4 + 2R_3$ we get,

$$\sim \left[\begin{array}{cccc|c} 1 & 0 & 0 & -3 & -7 \\ 0 & 1 & 0 & -1 & -2 \\ 0 & 0 & 1 & 4 & 11 \\ 0 & 0 & 0 & 13 & 39 \end{array}\right]$$

By $\frac{1}{13} R_4$ we get,

$$\sim \left[\begin{array}{cccc|c} 1 & 0 & 0 & -3 & -7 \\ 0 & 1 & 0 & -1 & -2 \\ 0 & 0 & 1 & 4 & 11 \\ 0 & 0 & 0 & 1 & 3 \end{array}\right]$$

$R_1 + 3R_4$, $R_2 + R_4$, $R_3 - 4R_4$ gives,

$$\sim \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 3 \end{array}\right] = [\, I \mid Z \,]$$

Here $$Z = \begin{bmatrix} 2 \\ 1 \\ -1 \\ 3 \end{bmatrix}$$

From equation 4.5.5 we know that

$$X = Z$$

i.e.,

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = Z = \begin{bmatrix} 2 \\ 1 \\ -1 \\ 3 \end{bmatrix}$$

$\therefore\ x_1 = 2,\ \ x_2 = 1,\ \ x_3 = -1,$ and $x_4 = 3$

Ex. 4.5.2 *Find the values of x_1, x_2 and x_3 by Gauss Jordan elimination method.*

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

Sol. : We know that the system of equations can be represented in matrix form as,

$$AX = B$$

Here,

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 4 & 3 & -1 \\ 3 & 5 & 3 \end{bmatrix} \quad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \text{and } B = \begin{bmatrix} 1 \\ 6 \\ 4 \end{bmatrix}$$

Let us form the augmented matrix [A/B] by adding column of 'B' to 'A'. i.e.,

$$[A|B] = \left[\begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 4 & 3 & -1 & 6 \\ 3 & 5 & 3 & 4 \end{array}\right]$$

By $R_2 - 4R_1$ and $R_3 - 3R_1$ we get,

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

By $-R_2$,

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

By $R_3 - 2R_2$ and $R_1 - R_2$ we get,

$$\sim \left[\begin{array}{ccc|c} 1 & 0 & -4 & 3 \\ 0 & 1 & 5 & -2 \\ 0 & 0 & -10 & 5 \end{array}\right]$$

By, $-\dfrac{1}{10}R_3$,

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

By $R_1 + 4R_3$ and $R_2 - 5R_3$ we get,

$$\sim \left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & -\frac{1}{2} \end{array}\right] = [\,I|Z\,]$$

Here $$Z = \begin{bmatrix} 1 \\ \frac{1}{2} \\ -\frac{1}{2} \end{bmatrix}$$

From equation 4.5.5 we know that,

$$X = Z$$

i.e. $$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{1}{2} \\ -\frac{1}{2} \end{bmatrix}$$

Thus, $x_1 = 1$, $x_2 = \frac{1}{2}$ and $x_3 = -\frac{1}{2}$

Ex. 4.5.3 *Find the inverse of the matrix*

$$\begin{bmatrix} 1 & 1 & 1 \\ 4 & 3 & -1 \\ 3 & 5 & 3 \end{bmatrix}$$

Using Gauss-Jordan method and hence solve the system.

$$\begin{aligned} x_1 + x_2 + x_3 &= 4 \\ 4x_1 + 3x_2 - x_3 &= 12 \\ 3x_1 + 5x_2 + 3x_3 &= 15 \end{aligned}$$

Sol. : Here observe that the given matrix represents the coefficient matrix of the system of equations. i.e.,

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

Let us form an augmented matrix [A | I] as follows :

$$[\,A \mid I\,] = \left[\begin{array}{ccc|ccc} 1 & 1 & 1 & 1 & 0 & 0 \\ 4 & 3 & -1 & 0 & 1 & 0 \\ 3 & 5 & 3 & 0 & 0 & 1 \end{array}\right]$$

Let us perform row manipulations on this matrix such that it will be converted to $[I \mid A^{-1}]$ form.

Let $R_2 - 4R_1$ and $R_3 - 3R_1$ then we get,

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

Let $-R_2$

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

Let $R_1 - R_2$ and $R_3 - 2R_2$,

$$\sim \left[\begin{array}{ccc|ccc} 1 & 0 & -4 & -3 & 1 & 0 \\ 0 & 1 & 5 & 4 & -1 & 0 \\ 0 & 0 & -10 & -11 & 2 & 1 \end{array}\right].$$

By $-\dfrac{1}{10}R_3$,

$$\sim \left[\begin{array}{ccc|ccc} 1 & 0 & -4 & -3 & 1 & 0 \\ 0 & 1 & 5 & 4 & -1 & 0 \\ 0 & 0 & 1 & 11/10 & -1/5 & -1/10 \end{array}\right]$$

By $R_1 + 4R_3$ and $R_2 - 5R_3$,

$$\sim \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & \frac{7}{5} & \frac{1}{5} & -\frac{2}{5} \\ 0 & 1 & 0 & -\frac{3}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 1 & \frac{11}{10} & -\frac{1}{5} & -\frac{1}{10} \end{array}\right]$$

We converted the matrix to $[I \mid A^{-1}]$ form. Hence inverse matrix is,

$$A^{-1} = \begin{bmatrix} 7/5 & 1/5 & -2/5 \\ -3/2 & 0 & 1/2 \\ 11/10 & -1/5 & -1/10 \end{bmatrix}$$

We know that the given system of equation is represented in matrix form as,

$$AX = B$$

$$\therefore \quad X = A^{-1}B$$

Putting for A^{-1} and B we get,

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7/5 & 1/5 & -2/5 \\ -3/2 & 0 & 1/2 \\ 11/10 & -1/5 & -1/10 \end{bmatrix} \begin{bmatrix} 4 \\ 12 \\ 15 \end{bmatrix}$$

$$\begin{bmatrix} \frac{28}{5} + \frac{12}{5} - \frac{30}{5} \\ -\frac{12}{2} + 0 + \frac{15}{2} \\ \frac{44}{10} - \frac{12}{5} - \frac{15}{10} \end{bmatrix} = \begin{bmatrix} 2 \\ 3/2 \\ 1/2 \end{bmatrix}$$

Thus we obtained x_1, x_2 and x_3 as, $x_1 = 2$, $x_2 = \dfrac{3}{2}$ and $x_3 = \dfrac{1}{2}$

Exercise

1. *Solve the following system of equations using gauss jordan method.*

$$x + y + z = 4$$
$$4x + 3y - z = 12$$
$$3x + 5y + 3z = 15$$

[Ans. : x = 2, y = 1.5 and z = 0.5]

2. *Solve by using gauss jordon method,*

$$10x_1 - 7x_2 + 3x_3 + 5x_4 = 6$$
$$-6x_1 + 8x_2 - x_3 - 4x_4 = 5$$
$$3x_1 + x_2 + 4x_3 + 11x_4 = 2$$
$$5x_1 - 9x_2 - 2x_3 + 4x_4 = 7$$

[Ans. : $x_1 = 5$, $x_2 = 4$, $x_3 = -7$, and $x_4 = 1$]

4.5.2 Algorithm of Gauss Jordan Method

Algorithm :

Step 1 : Read the number of variables 'n' in the system of equations.

Step 2 : Read the coefficients of equations i.e.,

Read a_{ij} *for* $i = 1$ *to* n

& $j = 1$ *to* n

& Read b_i *for* $i = 1$ *to* n

Step 3 : Form the augmented matrix [A | B] by adding extra column of 'B' to matrix A.

Step 4 : Convert the augmented matrix to new matrix [I | Z] by elementary row transformations such that $[I]_{n \times n}$ and $[Z]_{n \times 1}$.

Step 5 : $X = Z$ *i.e.,*

$x_i = z_i$ *for* $i = 1$ *to* n

Step 6 : Display values of variables x_i *;* $i = 1$ *to* n *on the screen.*

Step 7 : Stop.

University Questions

1. Use Gauss -Jordan Technique to solve the following system of equations.

$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85$$
$$0.1x_1 - 7x_2 - 0.3x_3 = -19.3$$
$$0.3x_1 - 0.2x_2 + 10x_3 = 71.4$$

[May - 2003]

2. Use Gauss Jordan method to compute the inverse of matrix

$$[A] = \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0.1 & 7 & -0.3 \\ 0.3 & -0.2 & 10 \end{bmatrix}$$

[May - 96]

4.6 Triangularization or LU Decomposition Method or Method of Factorization

Consider the system of equations involving three unknowns i.e.,

$$\begin{aligned} a_{11} x_1 + a_{12} x_2 + a_{13} x_3 &= b_1 \\ a_{21} x_1 + a_{22} x_2 + a_{23} x_3 &= b_2 \\ a_{31} x_1 + a_{32} x_2 + a_{33} x_3 &= b_3 \end{aligned} \quad \text{... (4.6.1)}$$

Represent these equations in the matrix form

$$AX = B \quad \text{... (4.6.2)}$$

i.e.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad \text{... (4.6.3)}$$

Let $A = LU$ ... (4.6.4)

where,

$$L = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \;\&\; U = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \quad \text{... (4.6.5)}$$

Putting value of $A = LU$ in equation 4.6.2,

$$LUX = B \quad \text{...(4.6.6)}$$

Let $UX = Z$ ... (4.6.7)

then equation 4.6.6 becomes, $LZ = B$ ... (4.6.8)

This equation is solved for Z. Then putting the value of Z in equation 4.6.7 we can obtain X.

4.6.1 Solved Examples

Ex.4.6.1 *Solve the following system of equations using triangularization (LU decomposition) method :*

$$\begin{aligned} 2x + 3y + z &= 9 \\ x + 2y + 3z &= 6 \\ 3x + y + 2z &= 8 \end{aligned}$$

Sol. : By expressing the system in matrix form we have,

$$\begin{bmatrix} 2 & 3 & 1 \\ 1 & 2 & 3 \\ 3 & 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 9 \\ 6 \\ 8 \end{bmatrix} \quad \text{... (4.6.9)}$$

Here,

$$A = \begin{bmatrix} 2 & 3 & 1 \\ 1 & 2 & 3 \\ 3 & 1 & 2 \end{bmatrix}, \quad X = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \text{ and } B = \begin{bmatrix} 9 \\ 6 \\ 8 \end{bmatrix} \qquad \ldots (4.6.10)$$

We have

$$LU = A \qquad \text{From equation 4.6.4}$$

Putting matrices for L & U from equation 4.6.5 and matrix for A from equation 4.6.10 above,

$$\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 \\ 1 & 2 & 3 \\ 3 & 1 & 2 \end{bmatrix}$$

$\therefore$ We get,

$$\boxed{u_{11} = 2}, \quad \boxed{u_{12} = 3,} \quad \boxed{u_{13} = 1}$$

and $\quad l_{21}\, u_{11} = 1 \quad \Rightarrow \quad \boxed{l_{21} = \frac{1}{2}}$

$$l_{31}\, u_{11} = 3 \quad \Rightarrow \quad \boxed{l_{31} = \frac{3}{2}}$$

and $\quad l_{21}\, u_{12} + u_{22} = 2$

Putting values of u_{21} and u_{12} we get,

$$\frac{1}{2} \times 3 + u_{22} = 2 \quad \Rightarrow \quad \boxed{u_{22} = \frac{1}{2}}$$

$$l_{21}\, u_{13} + u_{23} = 3$$

Putting values of l_{21} and u_{13} we get,

$$\frac{1}{2} \times 1 + u_{23} = 3 \quad \Rightarrow \quad \boxed{u_{23} = \frac{5}{2}}$$

Also,

$l_{31}\, u_{12} + l_{32}\, u_{22} = 1$ and $l_{31}\, u_{13} + l_{32}\, u_{23} + u_{33} = 2$

Putting values,

$$\boxed{l_{32} = -7} \quad \text{and} \quad \boxed{u_{33} = 18}$$

Thus we have,

$$L = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{3}{2} & -7 & 1 \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} 2 & 3 & 1 \\ 0 & \frac{1}{2} & \frac{5}{2} \\ 0 & 0 & 18 \end{bmatrix} \qquad \dots (4.6.11)$$

Let Z be the column matrix such that,

$$Z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

Then equation 4.6.8 gives,

$$LZ = B$$

Putting matrices for L, Z and B in the above equation,

$$\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{3}{2} & -7 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 9 \\ 6 \\ 8 \end{bmatrix}$$

Here, $\boxed{Z_1 = 9}$

$$\frac{1}{2} Z_1 + Z_2 = 6 \qquad \text{Putting for } Z_1\text{, we get } \boxed{Z_2 = \frac{3}{2}}$$

$$\& \quad \frac{3}{2} Z_1 - 7 Z_2 + Z_3 = 8$$

Putting for Z_1 and Z_2, we get, $\boxed{Z_3 = 5}$

[Observe that this solution of z_1, z_2 and z_3 we obtained by forward substitution].

Thus we have Z matrix as,

$$Z = \begin{bmatrix} 9 \\ \frac{3}{2} \\ 5 \end{bmatrix}$$

From equation 4.6.7 we know that

$$UX = Z$$

Putting matrices for U, X and Z in the above equation,

$$\begin{bmatrix} 2 & 3 & 1 \\ 0 & \frac{1}{2} & \frac{5}{2} \\ 0 & 0 & 18 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 9 \\ \frac{3}{2} \\ 5 \end{bmatrix}$$

From this we have,

$$18z = 5 \quad \Rightarrow \quad z = \frac{5}{18}$$

$$\frac{1}{2}y + \frac{5}{2}z = \frac{3}{2} \quad \Rightarrow \quad y = \frac{29}{18}$$

$$\& \qquad 2x + 3y + z = 9 \quad \Rightarrow \quad x = \frac{35}{18}$$

[Here observe that we have obtained values of x, y and z by backward substitution].

Ex.4.6.2 *Solve the following system of equations by using triangularization (LU decomposition) method.*

$$7x + 2y - 5z = -18$$
$$x + 5y - 3z = -40$$
$$2x - y - 9z = -26$$

Sol. : Represent this system in matrix form,

i.e. $$AX = B$$

Here, $$A = \begin{bmatrix} 7 & 2 & -5 \\ 1 & 5 & -3 \\ 2 & -1 & -9 \end{bmatrix}, \quad X = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \text{ and } B = \begin{bmatrix} -18 \\ -40 \\ -26 \end{bmatrix}$$

We know that,

$$LU = A$$

Putting values in above matrices,

$$\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} = \begin{bmatrix} 7 & 2 & -5 \\ 1 & 5 & -3 \\ 2 & -1 & -9 \end{bmatrix}$$

Solving for L and U matrices in last example, we get these matrices as,

$$L = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1/7 & 1 & 0 \\ 2/7 & -1/3 & 1 \end{bmatrix}$$

$$\& \qquad U = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} = \begin{bmatrix} 7 & 2 & -5 \\ 0 & 33/7 & -16/7 \\ 0 & 0 & -8.33 \end{bmatrix}$$

Let, $$LZ = B$$

$$Z \text{ is the column matrix } = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

Putting values of L and B,

$$\begin{bmatrix} 1 & 0 & 0 \\ 1/7 & 1 & 0 \\ 2/7 & -1/3 & 1 \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} -18 \\ -40 \\ -26 \end{bmatrix}$$

From this we have,

$$z_1 = -18$$

$$\frac{1}{7}z_1 + z_2 = -40 \quad \Rightarrow \quad z_2 = -37.428$$

& $$\frac{2}{7}z_1 - \frac{1}{3}z_2 + z_3 = -26 \quad \Rightarrow \quad z_3 = -33.33$$

Thus we have, $$Z = \begin{bmatrix} -18 \\ -37.428 \\ -33.33 \end{bmatrix}$$

We know that,

$$AX = B$$

$$LUX = B \qquad \because \; A = LU$$

$$LZ = B \qquad \because \; UX = Z$$

$$\therefore \quad UX = Z$$

Putting the values of matrices U, X and Z in above equation we get,

$$\begin{bmatrix} 7 & 2 & -5 \\ 0 & 33/7 & -16/7 \\ 0 & 0 & -8.33 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -18 \\ -33.428 \\ -33.33 \end{bmatrix}$$

From this we have,

$$7x + 2y - 5z = -18$$

$$\frac{33}{7}y - \frac{16}{7}z = -37.428$$

& $$-8.33z = -33.33$$

Solving the system of equations by backward substitution we get,

$x = 2, \quad y = -6 \quad \& \quad z = 4$

4.6.2 Algorithm for LU Decomposition Method

Based on the illustrative problems and method we will now prepare an algorithm for this method.

Algorithm :

Step 1 : *Read the number of variables 'n' in the system of equations.*

Step 2 : *Read the coefficients of equations i.e.*

Read a_{ij} *for* $i = 1$ *to* n

& $j = 1$ to n

& Read b_j for $j = 1$ to n

Step 3 : Prepare two dimensional array for a_{ij} i.e.

$$A = [a_{ij}] \quad i = 1 \text{ to } n \ \& \ j = 1 \text{ to } n$$

Prepare one dimensional array for b_i.

i.e. $B = [b_i]$ $j = 1$ to n

Step 4 : Decompose A into two matrices L & U of same order i.e.

$$A = LU$$

Here for 'n' elements, $[L]_{n \times n}$ and $[U]_{n \times n}$

Step 5 : Take matrix $[Z]_{n \times 1}$: Column matrix of n elements.

Step 6 : Solve,

$$LZ = B \quad \text{for } Z$$

Step 7 : Solve, $UX = B$ for X

Step 8 : The elements of column matrix X are $x_1, x_2 \ldots. x_n$.

Step 9 : Display values of variables x_i ; $i = 1, 2, \ldots n$ and then stop.

4.6.3 Matrix Inverse (Using Numerical Technique)

Now let us study a numerical technique to obtain inverse of the matrix. Inverse of a matrix is required in large number of applications. LU decomposition can be used to compute inverse of a matrix numerically. In the LU decomposition method if we solve X for B = $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, then we get first column of A^{-1}. i.e.,

Let B = $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, solve for X, then X = 1^{st} column of A^{-1}.

Let B = $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$, solve for X, then X = 2^{nd} column of A^{-1}.

Let B = $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$, solve for X, then X = 3^{rd} column of A^{-1}.

Next example illustrates this technique.

Ex. 4.6.3 : *Compute the inverse of the following matrix using numerical method :*

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

And use the above result to solve the system of equations :

$$3x + 2y + 4z = 7$$
$$2x + y + z = 7$$
$$x + 3y + 5z = 2$$

[May-97, 8 Marks; Dec-98, 8 Marks; Dec-2001, 8 Marks]

Sol. : i) **To obtain L and U decomposition**

We know that

$$L\,U = A$$

and,

$$L = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}, \quad A = \begin{bmatrix} 3 & 2 & 4 \\ 2 & 1 & 1 \\ 1 & 3 & 5 \end{bmatrix}$$

Hence the equation LU = A becomes,

$$\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} = \begin{bmatrix} 3 & 2 & 4 \\ 2 & 1 & 1 \\ 1 & 3 & 5 \end{bmatrix}$$

Now let us solve the above equation for L and U components. i.e.,

$$\boxed{u_{11} = 3} \qquad \boxed{u_{12} = 2} \qquad \boxed{u_{13} = 4}$$

and $\quad l_{21}\, u_{11} = 2$

$\therefore \quad l_{21} \cdot 3 = 2 \quad \Rightarrow \quad \boxed{l_{21} = \frac{2}{3}}$

and $\quad l_{31} \cdot u_{11} = 1$

$\therefore \quad l_{31} \cdot 3 = 1 \quad \Rightarrow \quad \boxed{l_{31} = \frac{1}{3}}$

Similarly, $\quad l_{21} \cdot u_{12} + u_{22} = 1$

$\frac{2}{3} \cdot 2 + u_{22} = 1 \quad \Rightarrow \quad \boxed{u_{22} = -\frac{1}{3}}$

and, $\quad l_{21} \cdot u_{13} + u_{23} = 1$

$\frac{2}{3} \cdot 4 + u_{23} = 1 \quad \Rightarrow \quad \boxed{u_{23} = -\frac{5}{3}}$

Also, $\quad l_{31} \cdot u_{12} + l_{32} \cdot u_{22} = 3$

$$\frac{1}{3}\cdot 2+l_{32}\cdot\left(-\frac{1}{3}\right) = 3 \quad\Rightarrow\quad \boxed{l_{32}=-7}$$

and, $$l_{31}\cdot u_{13}+l_{32}\cdot u_{23}+u_{33} = 5$$

$$\frac{1}{3}\cdot 4-7\cdot\left(-\frac{5}{3}\right)+u_{33} = 5 \quad\Rightarrow\quad \boxed{u_{33}=-8}$$

Thus we obtained L and U components as :

$$L = \begin{bmatrix} 1 & 0 & 0 \\ \frac{2}{3} & 1 & 0 \\ \frac{1}{3} & -7 & 1 \end{bmatrix} \quad U = \begin{bmatrix} 3 & 2 & 4 \\ 0 & -\frac{1}{3} & -\frac{5}{3} \\ 0 & 0 & -8 \end{bmatrix} \qquad \text{... (4.6.12)}$$

(ii) **To obtain first column of A^{-1}**

To obtain first column of A^{-1}, take $B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

Then evaluate, $LZ = B$

$$\begin{bmatrix} 1 & 0 & 0 \\ \frac{2}{3} & 1 & 0 \\ \frac{1}{3} & -7 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Solve the above equation for z_1, z_2 and z_3.

Hence $z_1 = 1$

and $$\frac{2}{3}z_1+z_2 = 0 \quad\Rightarrow\quad z_2=-\frac{2}{3}$$

and $$\frac{1}{3}z_1-7z_2+z_3 = 0$$

$$\therefore \quad \frac{1}{3}(1)-7\left(-\frac{2}{3}\right)+z_3 = 0 \quad\Rightarrow\quad z_3=-5$$

Thus we obtained, $$\boxed{z_1=1,\quad z_2=-\frac{2}{3}\quad \text{and}\quad z_3=-5}$$

Now $UX = Z$ will give first column of A^{-1}.

Hence we can write,

$$\begin{bmatrix} 3 & 2 & 4 \\ 0 & -\frac{1}{3} & -\frac{5}{3} \\ 0 & 0 & -8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -\frac{2}{3} \\ -5 \end{bmatrix}$$

Solve above equation for x_1, x_2 and x_3

Hence $$-8x_3 = -5 \quad\Rightarrow\quad x_3=\frac{5}{8}$$

and $$-\frac{1}{3}x_2 - \frac{5}{3}x_3 = -\frac{2}{3}$$

$$\therefore \quad -\frac{1}{3}x_2 - \frac{5}{3}\left(\frac{5}{8}\right) = -\frac{2}{3} \quad \Rightarrow \quad x_2 = -\frac{9}{8}$$

and $$3x_1 + 2x_2 + 4x_3 = 1$$

$$\therefore \quad 3x_1 + 2\left(-\frac{9}{8}\right) + 4\left(\frac{5}{8}\right) = 1 \quad \Rightarrow \quad x_1 = \frac{1}{4}$$

Thus we obtained first column of A^{-1} as,

$$\text{First column of } A^{-1} = \begin{bmatrix} \frac{1}{4} \\ -\frac{9}{8} \\ \frac{5}{8} \end{bmatrix} \qquad \text{... (4.6.13)}$$

iii) **To obtain second column of A^{-1}**

To obtain second column of A^{-1}, take B = $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$

Then evaluate LZ = B i.e.

$$\begin{bmatrix} 1 & 0 & 0 \\ \frac{2}{3} & 1 & 0 \\ \frac{1}{3} & -7 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Solve the above equation for z_1, z_2 and z_3

Hence $$z_1 = 0$$

and $$\frac{2}{3}z_1 + z_2 = 1 \quad \Rightarrow \quad z_2 = 1$$

and $$\frac{1}{3}z_1 - 7z_2 + z_3 = 0 \quad \Rightarrow \quad z_3 = 7$$

Thus we obtained, $\boxed{z_1 = 0, \quad z_2 = 1 \quad \text{and} \quad z_3 = 7}$

Now UX = Z will give second column of A^{-1}.

Hence we can write

$$\begin{bmatrix} 3 & 2 & 4 \\ 0 & -\frac{1}{3} & -\frac{5}{3} \\ 0 & 0 & -8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 7 \end{bmatrix}$$

Solve above equation for x_1, x_2 and x_3.

Hence $\quad -8x_3 = 7 \quad \Rightarrow \quad x_3 = -\frac{7}{8}$

and $\quad -\frac{1}{3}x_2 - \frac{5}{3}x_3 = 1$

$\therefore \quad -\frac{1}{3}x_2 - \frac{5}{3}\left(-\frac{7}{8}\right) = 1 \quad \Rightarrow \quad x_2 = \frac{11}{8}$

and $\quad 3x_1 + 2x_2 + 4x_3 = 0$

$\therefore \; 3x_1 + 2\left(\frac{11}{8}\right) + 4\left(-\frac{7}{8}\right) = 0 \quad \Rightarrow \quad x_1 = \frac{1}{4}$

Thus we obtained second column of A^{-1} as,

$$\text{Second column of } A^{-1} = \begin{bmatrix} \frac{1}{4} \\ \frac{11}{8} \\ -\frac{7}{8} \end{bmatrix} \qquad \text{... (4.6.14)}$$

iv) **To obtain third column of A^{-1}**

To obtain third column of A^{-1}, take B = $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

Then evaluate L Z = B. i.e.

$$\begin{bmatrix} 1 & 0 & 0 \\ 2/3 & 1 & 0 \\ 1/3 & -7 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Solve the above equation for z_1, z_2 and z_3

Hence $\quad z_1 = 0$

and $\quad \frac{2}{3}z_1 + z_2 = 0 \quad \Rightarrow \quad z_2 = 0$

and $\quad \frac{1}{3}z_1 - 7z_2 + z_3 = 1 \quad \Rightarrow \quad z_3 = 1$

Thus we obtained, $\boxed{z_1 = 0, \quad z_2 = 0 \quad \text{and} \quad z_3 = 1}$

Now U X = Z will give third column of A^{-1}.

Hence we can write,

$$\begin{bmatrix} 3 & 2 & 4 \\ 0 & -\frac{1}{3} & -\frac{5}{3} \\ 0 & 0 & -8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Solve above equation for x_1, x_2 and x_3

$$\text{Hence} \qquad -8\,x_3 = 1 \quad \Rightarrow \quad x_3 = -\frac{1}{8}$$

$$\text{and} \qquad -\frac{1}{3}x_2 - \frac{5}{3}x_3 = 0$$

$$\therefore \qquad -\frac{1}{3}x_2 - \frac{5}{3}\left(-\frac{1}{8}\right) = 0 \quad \Rightarrow \quad x_2 = \frac{5}{8}$$

$$\text{and} \qquad 3x_1 + 2x_2 + 4x_3 = 0$$

$$\therefore \quad 3x_1 + 2\left(\frac{5}{8}\right) + 4\left(-\frac{1}{8}\right) = 0 \quad \Rightarrow \quad x_1 = -\frac{1}{4}$$

Thus we obtained third column of A^{-1} as,

$$\text{Third column of } A^{-1} = \begin{bmatrix} -\frac{1}{4} \\ \frac{5}{8} \\ -\frac{1}{8} \end{bmatrix}$$

Thus the matrix A^{-1} can be written from equation 4.6.13, equation 4.6.14 and above equation as,

$$A^{-1} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} \\ -\frac{9}{8} & \frac{11}{8} & \frac{5}{8} \\ \frac{5}{8} & -\frac{7}{8} & -\frac{1}{8} \end{bmatrix} \qquad \ldots (4.6.15)$$

v) **To solve the given system of equations**

The given equations are

$$3x + 2y + 4z = 7$$
$$2x + y + z = 7$$
$$x + 3y + 5z = 2$$

$$\text{Here,} \qquad A = \begin{bmatrix} 3 & 2 & 4 \\ 2 & 1 & 1 \\ 1 & 3 & 5 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 7 \\ 7 \\ 2 \end{bmatrix}$$

In matrix form, the equations can be written as,

$$AX = B$$

$$\therefore \qquad X = A^{-1}B$$

Putting the value of A^{-1} from equation 4.6.15 and B,

$$X = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} \\ -\frac{9}{8} & \frac{11}{8} & \frac{5}{8} \\ \frac{5}{8} & -\frac{7}{8} & -\frac{1}{8} \end{bmatrix} \begin{bmatrix} 7 \\ 7 \\ 2 \end{bmatrix}$$

$$X = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{7}{4} + \frac{7}{4} - \frac{2}{4} \\ -\frac{63}{8} + \frac{77}{8} + \frac{10}{8} \\ \frac{35}{8} - \frac{49}{8} - \frac{2}{8} \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ -2 \end{bmatrix}$$

Thus the solution is,

x = 3, y = 3 and z = –2

Ex. 4.6.4 *Determine the matrix inverse for the matrix*

$$A = \begin{bmatrix} 12 & -7 & 3 \\ 1 & 7 & -4 \\ 4 & -4 & 9 \end{bmatrix}$$

Using LU decomposition method. Using the inverse solve the following equations :

$$12x_1 - 7x_2 + 3x_3 = 8$$

$$x_1 + 7x_2 - 4x_3 = -51$$

$$4x_1 - 4x_2 + 9x_3 = 62$$

[Dec - 2004, 15 Marks]

Sol. : We have discussed the procedure to solve such example in detail in previous example. The LU decomposition is given as,

$$\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} = \begin{bmatrix} 12 & -7 & 3 \\ 1 & 7 & -4 \\ 4 & -4 & 9 \end{bmatrix}$$

Solving above equation we get following L and U matrices :

$$\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{12} & 1 & 0 \\ \frac{1}{3} & -\frac{20}{91} & 1 \end{bmatrix}$$

and $$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} = \begin{bmatrix} 12 & -7 & 3 \\ 0 & \frac{91}{12} & -\frac{17}{4} \\ 0 & 0 & \frac{643}{91} \end{bmatrix}$$

Now LZ = B where $B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ gives Z.

i.e. $$\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{12} & 1 & 0 \\ \frac{1}{3} & -\frac{20}{91} & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \therefore Z = \begin{bmatrix} 1 \\ -0.0833 \\ -0.3516 \end{bmatrix}$$

And UX = Z will give 'X' which is first column of A^{-1}. i.e.,

$$\begin{bmatrix} 12 & -7 & 3 \\ 0 & \frac{91}{12} & -\frac{17}{4} \\ 0 & 0 & \frac{643}{91} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -0.0833 \\ -0.3516 \end{bmatrix}$$

First column of $$A^{-1} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.0731 \\ -0.0389 \\ -0.0498 \end{bmatrix}$$

Similarly 2^{nd} and 3^{rd} columns of A^{-1} can be obtained. The complete A^{-1} matrix is,

$$A^{-1} = \begin{bmatrix} 0.0731 & 0.0793 & 0.0109 \\ -0.0389 & 0.1493 & 0.0793 \\ -0.0498 & 0.0311 & 0.1415 \end{bmatrix}$$

To obtain solution of equations

From the system of equations we have,

$$A = \begin{bmatrix} 12 & -7 & 3 \\ 1 & 7 & -4 \\ 4 & -4 & 9 \end{bmatrix} \text{ and } B = \begin{bmatrix} 8 \\ -51 \\ 62 \end{bmatrix}$$

$$\therefore \qquad AX = B$$

or
$$X = A^{-1} B$$
$$= \begin{bmatrix} 0.0731 & 0.0793 & 0.0109 \\ -0.0389 & 0.1493 & 0.0793 \\ -0.0498 & 0.0311 & 0.1415 \end{bmatrix} \begin{bmatrix} 8 \\ -51 \\ 62 \end{bmatrix}$$
$$= \begin{bmatrix} -2.7854 \\ -3 \\ 6.79 \end{bmatrix}$$

Thus, $x_1 = -2.7854$

$x_2 = -3$

and $x_3 = 6.79$

Exercise

1. Solve the following using triangularization (LU decomposition) method.

$$\begin{bmatrix} 8 & 4 & 2 & 0 \\ 4 & 10 & 5 & 4 \\ 2 & 5 & 6.5 & 4 \\ 0 & 4 & 4 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 24 \\ 32 \\ 26 \\ 21 \end{bmatrix}$$

[Hint : Here A is 4 × 4 matrix, Hence L & U will also be 4 × 4 size as shown below.

$$L\,U = A$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix} = \begin{bmatrix} 8 & 4 & 2 & 0 \\ 4 & 10 & 5 & 4 \\ 2 & 5 & 6.5 & 4 \\ 0 & 4 & 4 & 9 \end{bmatrix}$$

& matrix Z will be,

$$Z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix}$$

[Ans. : $x_1 = 2$, $x_2 = 1$, $x_3 = 2$ and $x_4 = 1$]

2. Using LU decomposition method solve the following system of equations.

$$2x_1 - 5x_2 + x_3 = 12$$
$$-x_1 + 3x_2 - x_3 = -8$$
$$3x_1 - 4x_2 + 2x_3 = 16$$

[Ans. : $x_1 = 2$, $x_2 = -1$ and $x_3 = 3$]

3. Solve the following equations by triangularization method.

$$2x + 2y + 3z = 4$$
$$4x - 2y + z = 9$$
$$x + 5y + 4z = 3$$

[Ans. : x = −2, y = 1.5 and z = 3.5]

4. Explain any numerical method to evaluate the inverse of a matrix. Using this, evaluate the inverse of

$$\begin{bmatrix} 0.7 & -5.4 & 1.0 \\ 3.5 & 2.2 & 0.8 \\ 1.0 & -1.5 & 4.3 \end{bmatrix}$$

[Dec.-95 8 Marks, Dec.-97 8 Marks, May-98 8 Marks]

Hint : Use LU decomposition.

Ans. : $A^{-1} = \begin{bmatrix} 0.1385 & 0.2822 & -0.0847 \\ -0.1852 & 0.0261 & 0.0382 \\ -0.0968 & -0.0565 & 0.2656 \end{bmatrix}$

University Questions

1. Write a short note on LU decomposition technique.

[Dec - 95, Dec - 98, May - 99, Dec - 2000]

2. Explain any numerical method to evaluate the inverse of a matrix. Using this, evaluate the inverse of

$$\begin{bmatrix} 0.7 & -5.4 & 1.0 \\ 3.5 & 2.2 & 0.8 \\ 1.0 & -1.5 & 4.3 \end{bmatrix}$$

[Dec - 95, Dec - 97, May - 98]

3. Compute the inverse of the matrix -

$$\begin{bmatrix} 3 & 2 & 4 \\ 2 & 1 & 1 \\ 1 & 3 & 5 \end{bmatrix}$$

And use the result to solve the system of equations :

$$3x + 2y + 4z = 7$$
$$2x + y + z = 7$$
$$x + 3y + 5z = 2$$

[May - 97, Dec - 98, Dec - 2001]

4. Find A^{-1} using LU Decomposition method and hence solve -

$$7x_1 + 2x_2 - 3x_3 = -12$$
$$2x_1 + 5x_2 - 3x_3 = -20$$
$$x_1 - x_2 - 6x_3 = -26$$

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

[Dec - 2002]

5. Solve by LU Decomposition method

$$x + y + z = 9$$
$$2x - 3y + 4z = 13$$
$$3x + 4y + 5z = 40$$

[Dec - 2003]

6. Determine the matrix inverse for the

$$\text{Matrix} \begin{bmatrix} 12 & -7 & 3 \\ 1 & 7 & -4 \\ 4 & -4 & 9 \end{bmatrix}$$

Using LU decomposition method. Using the inverse solve the following equations :

$$12x_1 - 7x_2 + 3x_3 = 8$$
$$x_1 + 7x_2 - 4x_3 = -51$$
$$4x_1 - 4x_2 + 9x_3 = 62$$

[Dec - 2004, 15 Marks]

4.7 Gauss Seidel Iterative Method

Consider the system of equations having four unknowns for explanation

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 = b_1$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 = b_2$$
$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 = b_3$$
$$a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 = b_4$$

We can write the above equations as follows :

$$\left.\begin{aligned} x_1 &= \frac{1}{a_{11}}[b_1 - a_{12}x_2 - a_{13}x_3 - a_{14}x_4] \\ x_2 &= \frac{1}{a_{22}}[b_2 - a_{21}x_1 - a_{23}x_3 - a_{24}x_4] \\ x_3 &= \frac{1}{a_{33}}[b_3 - a_{31}x_1 - a_{32}x_2 - a_{34}x_4] \\ x_4 &= \frac{1}{a_{44}}[b_4 - a_{41}x_1 - a_{42}x_2 - a_{43}x_3] \end{aligned}\right\} \quad \dots (4.7.1)$$

Thus the variables are expressed in terms of other variables.

Let $x_1^{(k)}$ = Value of x_1 in k^{th} iteration

$x_2^{(k)}$ = Value of x_2 in k^{th} iteration

$x_3^{(k)}$ = Value of x_3 in k^{th} iteration

$x_4^{(k)}$ = Value of x_4 in k^{th} iteration

And Let $x_1^{(k+1)}$, $x_2^{(k+1)}$, $x_3^{(k+1)}$ and $x_4^{(k+1)}$ be the values of those variables in $[(k+1)^{th}]$ iteration. Then equation 4.7.1 gives values of variables in next iteration. We can then write equation 4.7.1 as follows :

$$\left.\begin{aligned}
x_1^{(k+1)} &= \frac{1}{a_{11}}\left[b_1 - a_{12}\,x_2^{(k)} - a_{13}\,x_3^{(k)} - a_{14}\,x_4^{(k)}\right] \\
x_2^{(k+1)} &= \frac{1}{a_{22}}\left[b_2 - a_{21}\,x_1^{(k)} - a_{23}\,x_3^{(k)} - a_{24}\,x_4^{(k)}\right] \\
x_3^{(k+1)} &= \frac{1}{a_{33}}\left[b_3 - a_{31}\,x_1^{(k)} - a_{32}\,x_2^{(k)} - a_{34}\,x_4^{(k)}\right] \\
x_4^{(k+1)} &= \frac{1}{a_{44}}\left[b_4 - a_{41}\,x_1^{(k)} - a_{42}\,x_2^{(k)} - a_{43}\,x_3^{(k)}\right]
\end{aligned}\right\}$$

Observe here that to calculate values of variables of next iteration we use values of variables in previous iteration. Actually when we calculate $x_2^{(k+1)}$ the value of $x_1^{(k+1)}$ can be used. Similarly to calculate $x_4^{(k+1)}$ the value of x_1 , x_2 and x_3 can be used from present $[(k+1)^{th}]$ iteration only. Because of this, the convergence is fast and less number of iterations are required. This is called ***Gauss Seidel*** iterative method. According to this method, the above iterative equations will be changed as follows :

$$\left.\begin{aligned}
x_1^{(k+1)} &= \frac{1}{a_{11}}\left[b_1 - a_{12}\,x_2^{(k)} - a_{13}\,x_3^{(k)} - a_{14}\,x_4^{(k)}\right] \\
x_2^{(k+1)} &= \frac{1}{a_{22}}\left[b_2 - a_{21}\,x_1^{(k+1)} - a_{23}\,x_3^{(k)} - a_{24}\,x_4^{(k)}\right] \\
x_3^{(k+1)} &= \frac{1}{a_{33}}\left[b_3 - a_{31}\,x_1^{(k+1)} - a_{32}\,x_2^{(k+1)} - a_{34}\,x_4^{(k)}\right] \\
x_4^{(k+1)} &= \frac{1}{a_{44}}\left[b_4 - a_{41}\,x_1^{(k+1)} - a_{42}\,x_2^{(k+1)} - a_{43}\,x_4^{(k+1)}\right]
\end{aligned}\right\} \quad \ldots (4.7.2)$$

Before proceeding for solution always check that all the diagonal elements are dominant. i.e.,

a_{11} , a_{22} , a_{33} .... a_{nn} are as far as possible higher than the other elements of their respective columns.

The procedure for Gauss Seidel iterative method can be extended similarly to higher number of variables.

4.7.1 Solved Examples

Ex. 4.7.1 *Solve the following system of equations using gauss seidel iterative method.*

$$\begin{aligned}
10x - 2y - z - t &= 3 \\
-2x + 10y - z - t &= 15 \\
-x - y + 10z - 2t &= 27 \\
-x - y - 2z + 10t &= -9
\end{aligned}$$

Sol. : Since the diagonal elements are higher than other elements of their respective columns, we can use the system of equations without rearranging.

Let's write the equations in the format of equation 4.7.2.

$$\left.\begin{aligned} x^{(k+1)} &= \frac{1}{10}\left[3 + 2y^{(k)} + z^{(k)} + t^{(k)}\right] \\ y^{(k+1)} &= \frac{1}{10}\left[15 + 2x^{(k+1)} + z^{(k)} + t^{(k)}\right] \\ z^{(k+1)} &= \frac{1}{10}\left[27 + x^{(k+1)} + y^{(k+1)} + 2t^{(k)}\right] \\ t^{(k+1)} &= \frac{1}{10}\left[-9 + x^{(k+1)} + y^{(k+1)} + 2z^{(k+1)}\right] \end{aligned}\right\} \quad \ldots (4.7.3)$$

Iteration 1 :

Let k = 0 and take initial approximation as,

$$x^{(0)} = y^{(0)} = z^{(0)} = t^{(0)} = 0$$

$$x^{(1)} = \frac{1}{10}[3 + 0 + 0 + 0] = 0.3$$

$$y^{(1)} = \frac{1}{10}\left[15 + 2x^{(1)} + 0 + 0\right] = \frac{1}{10}[15 + (2 \times 0.3)]$$

$$= 1.56$$

$$z^{(1)} = \frac{1}{10}\left[27 + x^{(1)} + y^{(1)} + 0\right] = \frac{1}{10}[27 + 0.3 + 1.56] = 2.886$$

$$t^{(1)} = \frac{1}{10}\left[-9 + x^{(1)} + y^{(1)} + 2z^{(1)}\right] = \frac{1}{10}[-9 + 0.3 + 1.56 + (2 \times 2.886)]$$

$$= -0.1368$$

Iteration No. 2 :

Let k = 1 in equation 4.7.3 and substitute respective values.

$$x^{(2)} = \frac{1}{2}\left[3 + 2y^{(1)} + z^{(1)} + t^{(1)}\right] = \frac{1}{10}[3 + (2 \times 1.56) + 2.886 - 0.1368]$$

$$= 0.8869$$

$$y^{(2)} = \frac{1}{10}\left[15 + 2x^{(2)} + z^{(1)} + t^{(1)}\right]$$

$$= \frac{1}{10}[15 + (2 \times 0.8869) + 2.886 - 0.1368] = 1.9523$$

$$z^{(2)} = \frac{1}{10}\left[27 + x^{(2)} + y^{(2)} + 2t^{(1)}\right]$$

$$= \frac{1}{10}[27 + 0.8869 + 1.9523 - (2 \times 0.1368)] = 2.9556$$

$$t^{(2)} = \frac{1}{10}\left[-9 + x^{(2)} + y^{(2)} + 2z^{(2)}\right]$$

$$= \frac{1}{10}[-9 + 0.8869 + 1.9523 + (2 \times 2.9566)] = 0.0248$$

The results of next iterations are tabulated as shown below.

Iteration No.	k	x	y	z	t
3	k = 2	0.983641	1.989908	2.992402	– 0.004165
4	k = 3	0.996805	1.998185	2.998666	– 0.000768
5	k = 4	0.999427	1.999675	2.999757	– 0.000138
6	k = 5	0.999897	1.999941	2.999956	– 0.000025

Thus the exact solution will be

$x = 1, \quad y = 2, \quad z = 3, \quad \text{and} \quad t = 0$

Ex.4.7.2 *Solve the following system of equations using gauss seidel iterative method. The answer should be correct to two significant digits.*

$$9x_1 + 2x_2 + 4x_3 = 20$$
$$2x_1 - 4x_2 + 10x_3 = -15$$
$$x_1 + 10x_2 + 4x_3 = 6$$

Sol. : We will first rearrange the equations by exchanging second and third equations. i.e.,

$$9x_1 + 2x_2 + 4x_3 = 20$$
$$x_1 + 10x_2 + 4x_3 = 6$$
$$2x_1 - 4x_2 + 10x_3 = -15$$

In this system of equations we observe that the diagonal elements are dominant. That is they are as far as possible higher than other elements in their respective columns.

Let's express these equations in the form of equation 4.7.2. i.e.,

$$\left.\begin{aligned} x_1^{(k+1)} &= \frac{1}{9}\left[20 - 2x_2^{(k)} - 4x_3^{(k)}\right] \\ x_2^{(k+1)} &= \frac{1}{10}\left[6 - x_1^{(k+1)} - 4x_3^{(k)}\right] \\ x_3^{(k+1)} &= \frac{1}{10}\left[-15 - 2x_1^{(k+1)} + 4x_2^{(k+1)}\right] \end{aligned}\right\} \quad \text{...(4.7.4)}$$

Iteration 1 :

Put k = 0 and let $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$

Putting these values in the system of iterative equations of equation 4.7.4 we get,

$$x_1^{(1)} = \frac{1}{9}\left[20 - 2x_2^{(0)} - 4x_3^{(0)}\right] = \frac{1}{9}[20 - 0 - 0] = 2.22222$$

$$x_2^{(1)} = \frac{1}{10}\left[6 - x_1^{(1)} - 4x_3^{(0)}\right] = \frac{1}{10}[6 - 2.2222 - 0] = 0.377778$$

$$x_3^{(1)} = \frac{1}{10}\left[-15 - 2x_1^{(1)} + 4x_2^{(1)}\right]$$

$$= \frac{1}{10}[-15 - (2 \times 2.2222) + (4 \times 0.37778)] = -1.793332$$

Iteration No. 2

Let k = 1 in equation 4.7.4 and putting values obtained in iteration 1.

We get, $x_1^{(2)} = \frac{1}{9}\left[20 - 2x_2^{(1)} - 4x_3^{(1)}\right]$

$= \frac{1}{9}[20 - (2 \times 0.37778) + (4 \times 1.793332)] = 2.935308$

$x_2^{(2)} = \frac{1}{10}\left[6 - x_1^{(2)} - 4x_3^{(1)}\right] = \frac{1}{10}[6 - 2.935308 + (4 \times 1.793332)]$

$= 1.023802$

$x_3^{(2)} = \frac{1}{10}\left[-15 - 2x_1^{(2)} + 4x_2^{(2)}\right]$

$= \frac{1}{10}[-15 - (2 \times 2.935308) + (4 \times 1.023802)] = -1.677541$

The result of next iterations are tabulated as shown in the table below.

Iteration No.	k	x_1	x_2	x_3
3	k = 2	2.740284	0.996988	– 1.649262
4	k = 3	2.733675	0.986337	– 1.65220
5	k = 4	2.737347	0.987145	– 1.652611

Since the values of variables are repeating upto two digits after decimal points, these are the required values.

Ex. 4.7.3 : *Find the solution to 3 decimal places of the system :*

$$7x_1 + 52x_2 + 13x_3 = 104$$
$$83x_1 + 11x_2 - 4x_3 = 95$$
$$3x_1 + 8x_2 + 29x_3 = 71$$

Using gauss seidel method **[Dec.-2001 8 Marks]**

Sol. : Let us rearrange the given system of equations as follows :

$$83x_1 + 11x_2 - 4x_3 = 95$$
$$7x_1 + 52x_2 + 13x_3 = 104$$
$$3x_1 + 8x_2 + 29x_3 = 71$$

Here note that the diagonal elements are dominant. Let us express these equations in the form of equation 4.7.2 i.e.,

$$\left.\begin{aligned} x_1^{(k+1)} &= \frac{1}{83}[95 - 11x_2^{(k)} + 4x_3^{(k)}] \\ x_2^{(k+1)} &= \frac{1}{52}[104 - 7x_1^{(k+1)} - 13x_3^{(k)}] \\ x_3^{(k+1)} &= \frac{1}{29}[71 - 3x_1^{(k+1)} - 8x_3^{(k+1)}] \end{aligned}\right\} \quad \text{... (4.7.5)}$$

Iteration No. 1 :

Put k = 0 and let $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$.

Putting these values in the system of iterative equations of equation 4.7.5, we get,

$$x_1^{(1)} = \frac{1}{83}[95 - 11x_2^{(0)} + 4x_3^{(0)}] = \frac{1}{83}[95 - 0 + 0]$$

$$= 1.144578$$

$$x_2^{(1)} = \frac{1}{52}[104 - 7x_1^{(1)} - 13x_3^{(0)}] = \frac{1}{52}[104 - 7(1.144578) - 0]$$

$$= 1.845922$$

$$x_3^{(1)} = \frac{1}{29}[71 - 3x_1^{(1)} - 8x_2^{(1)}] = \frac{1}{29}[71 - 3(1.144578) - 8(1.845922)]$$

$$= 1.820651$$

Iteration No. 2 :

Let k = 1 in equation 4.7.5, then we get,

$$x_1^{(2)} = \frac{1}{83}[95 - 11x_2^{(1)} + 4x_3^{(1)}]$$

$$= \frac{1}{83}[95 - 11(1.845922) + 4(1.820651)] = 0.987680$$

$$x_2^{(2)} = \frac{1}{52}[104 - 7x_1^{(2)} - 13x_3^{(1)}]$$

$$= \frac{1}{52}[104 - 7(0.987680) - 13(1.820651)] = 1.411880$$

$$x_3^{(2)} = \frac{1}{29}[71 - 3x_1^{(2)} - 8x_2^{(2)}]$$

$$= \frac{1}{29}[71 - 3(0.987680) - 8(1.411880)] = 1.956618$$

The next iterations can be performed similarly. The results are tabulated below :

Iteration No.	k	x_1	x_2	x_3
3	k = 2	1.051756	1.369263	1.961746
4	k = 3	**1.057**652	**1.367**187	**1.961**708
5	k = 4	**1.057**925	**1.367**160	**1.961**688

As shown in above table, the values of x_1 x_2 and x_3 repeat upto 3 significant degits after decimal point. Hence the values obtained in last iteration are correct to 3 decimal places.

$$x_1 = 1.057925$$

$$x_2 = 1.367160$$

$$x_3 = 1.961688$$

Ex. 4.7.4 : *Use gauss-seidel method to obtain the solution of the system*

$$\begin{aligned} 3x_1 - 0.1x_2 - 0.2x_3 &= 7.85 \\ 0.1x_1 + 7x_2 - 0.3x_3 &= -19.3 \\ 0.3x_1 - 0.2x_2 + 10x_3 &= 71.4 \end{aligned}$$

[May-96 10 Marks, May-98 10 Marks, May-2000 8 Marks, Dec-2002 8 Marks]

Sol. : The given system of equations can be written as per equation 4.7.2 i.e.,

$$\left.\begin{aligned} x_1^{(k+1)} &= \frac{1}{3}[7.85 + 0.1\, x_2^{(k)} + 0.2\, x_3^{(k)}] \\ x_2^{(k+1)} &= \frac{1}{7}[-19.3 - 0.1\, x_1^{(k+1)} + 0.3\, x_3^{(k)}] \\ x_3^{(k+1)} &= \frac{1}{10}[71.4 - 0.3\, x_1^{(k+1)} + 0.2\, x_2^{(k+1)}] \end{aligned}\right\} \quad \ldots (4.7.6)$$

Iteration No. 1 :

Let k = 0 and $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$ in equation 4.7.6. Then we have,

$$\begin{aligned} x_1^{(1)} &= \frac{1}{3}[7.85 + 0.1x_2^{(0)} + 0.2x_3^{(0)}] \\ &= \frac{1}{3}[7.85 + 0.1(0) + 0.2(0)] = 2.616667 \\ x_2^{(1)} &= \frac{1}{7}[-19.3 - 0.1x_1^{(1)} + 0.3x_3^{(0)}] \\ &= \frac{1}{7}[-19.3 - 0.1(2.616667) + 0.3(0)] = -2.794524 \\ x_3^{(1)} &= \frac{1}{10}[71.4 - 0.3x_1^{(1)} + 0.2x_2^{(1)}] \\ &= \frac{1}{10}[71.4 - 0.3\,(2.616667 + 0.2\,(-2.794524)] = 7.005610 \end{aligned}$$

Iteration No. 2 :

Let k = 1 in equation 4.7.6 and putting relevant values,

$$\begin{aligned} x_1^{(2)} &= \frac{1}{3}[7.85 + 0.1x_2^{(1)} + 0.2x_3^{(1)}] \\ &= \frac{1}{3}[7.85 + 0.1\,(-2.794524) + 0.2\,(7.005610)] = 2.990557 \\ x_2^{(2)} &= \frac{1}{7}[-19.3 - 0.1x_1^{(2)} + 0.3x_3^{(1)}] \\ &= \frac{1}{7}[-19.3 + 0.1\,(2.990557) + 0.3\,(7.005610)] = -2.499625 \\ x_3^{(2)} &= \frac{1}{10}[71.4 - 0.3x_1^{(2)} + 0.2x_2^{(2)}] \end{aligned}$$

$$= \frac{1}{10}[71.4 - 0.3\,(2.990557) + 0.2\,(-2.499625)] = 7.000291$$

Next iterations can be performed similarly. Following table lists the results of next iterations :

Iteration No.	k	x_1	x_2	x_3
3	2	3.000032	−2.499988	6.999999
4	3	3.000000	−2.500000	7.000000

Thus the above values are correct upto 6 decimal places.

Ex. 4.7.5 *Solve the following system of equations sing Gauss-Seidel method.*

$$6x + 15y + 2z = 72$$

$$x + y + 54z = 110$$

$$27x + 6y - z = 85$$

[May-2003, 8 Marks]

Sol. : Let us rearrange the equations as follows :

$$27x + 6y - z = 85$$

$$6x + 15y + 2z = 72$$

$$x + y + 54z = 110$$

The above system of equations can be written as follows :

$$x^{(k+1)} = \frac{1}{27}[85 - 6y^{(k)} + z^{(k)}]$$

$$y^{(k+1)} = \frac{1}{15}[72 - 6x^{(k+1)} - 2z^{(k)}]$$

$$z^{(k+1)} = \frac{1}{54}[110 - x^{(k+1)} - y^{(k+1)}]$$

Following table lists the results of each iteration.

Iteration no	k	x	y	z
1	0	3.148148	3.540741	1.913169
2	1	2.432175	3.572041	1.925848
3	2	2.425689	3.572945	1.925951
4	3	2.425492	3.573010	1.925954
5	4	2.425478	3.573015	1.925954

The above values are repeating upto four decimal places. Hence they are correct upto 4 significant digits after decimal point.

Exercise

1. Solve the following system of equations using gauss siedel method.

$$17a + 65b - 13c + 50d = 84$$
$$12a + 16b + 37c + 18d = 25$$
$$56a + 23b + 11c - 19d = 36$$
$$3a - 5b + 47c + 10d = 18$$

[Hint : Rearrange this system of equation as,

$$56a + 23b + 11c - 19d = 36$$
$$17a + 65b - 13c + 50d = 84$$
$$12a + 16b + 37c + 18d = 25$$
$$3a - 5b + 47c + 10d = 18$$

[Ans. : a = 4.834, b = – 4.697, c = –1.641 and d = 5.717]

2. Solve the following system of equations using gauss seidel method.

$$4x + y + 2z = 4$$
$$3x + y + z = 7$$
$$x + y + 3z = 3$$

[Ans. : x = – 0.5, y = 11 and z = – 2.5]

3. Solve the following equations by gauss seidel method :

$$x_1 + 10x_2 + 4x_3 = 6$$
$$2x_1 - 4x_2 + 10x_3 = -15$$
$$9x_1 + 2x_2 + 4x_3 = 20$$

[May- 97 8 Marks, May-99 8 Marks, May-2001 8 Marks]

Hint : Arrange the above equations as follow :

$$9x_1 + 2x_2 + 4x_3 = 20$$
$$x_1 + 10x_2 + 4x_3 = 6$$
$$2x_1 - 4x_2 + 10x_3 = -15$$

Ans. : In 8^{th} iteration, solution is correct upto 4 decimal places. i.e.,

$x_1 = 2.737287$, $x_2 = 0.987288$ and $x_3 = -1.652542$

4. Given the linear equations :

$$4x + y + z = 5$$
$$x + 6y + 2z = 19$$
$$-x - 2y + 5z = 10$$

Obtain the values of x, y and z for three successive iterations using gauss – seidel method with an initial guess of x = y = z = 0

[Dec.-97 6 Marks, Dec.-98 8 Marks, Dec.-2000 10 Marks]

Ans. :

Iteration No.	x	y	z
1	1.25	2.958333	3.433333
2	– 0.347917	2.080208	2.762500
3	0.039323	2.239280	2.903576

5. Develop the gauss-seidel iterative scheme for the solution of the system :

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

Iterate upto maximum of 10 times or upto an accuracy of 0.0001 starting with initial solution vector $x_0 = 0$. **[Dec-96 8 Marks, Dec.-99 8 Marks]**

Ans. :

Iteration No.	x_1	x_2	x_3
1	0.3	0.42	0.582
2	0.6264	0.72516	0.797736
3	0.822127	0.868172	0.903116
4	0.914709	0.936818	0.953562
5	0.959122	0.969717	0.977742
6	0.980407	0.985486	0.989332
7	0.990609	0.993043	0.994887
8	0.995499	0.996666	0.997549
9	0.997843	0.998402	0.998825
10	0.998966	0.999234	0.999437

4.7.2 C Program and Algorithm

Let's now prepare the algorithm for gauss seidel method based on our discussion and illustrative examples.

Algorithm :

Step 1 : *Read number of variables 'n' in the system of equations.*

Step 2 : *Read the coefficients of equations i.e.,*

Read a_{ij} *for* $i = 1$ *to* n

& $j = 1$ *to* n

& *Read* b_i *for* $i = 1$ *to* n

Step 3 : *Rearrange the equations so that method convergence problem will not arise. (i.e. arrange array equations such that diagonal elements are dominant).*

Step 4 : *Represent the system of equations in their iterative form as,*

$$x_i = \frac{1}{a_{ii}} \left[b_i - \left(a_{j1} x_1 + a_{j2} x_2 + a_{j3} x_3 + \ldots + a_{jn} x_n \right) \right]$$

& $i = 1$ *to* n

$j = 1$ *to* n

& $i \neq j$

(Here iteration number is not marked on variables since they use values obtained in current iterations also).

Step 5 : *Take initial values of* x_i ; $i = 1$ *to* n *to be zero.*

Step 6 : *Execute iterative equation of step 4 to obtain* x_i , $i = 1$ *to* n. *And use latest values of* $x_1, x_2, x_3 \ldots x_n$.

Step 7 : *Repeat step 6 until required number of iterations or required accuracy is achieved in the value of variables.*

Step 8 : *Display the values of variables* x_i ; $i = 1$ *to* n *on the screen and stop.*

Flowchart :

The simplified flowchart of this method is shown in Fig. 4.7.1.

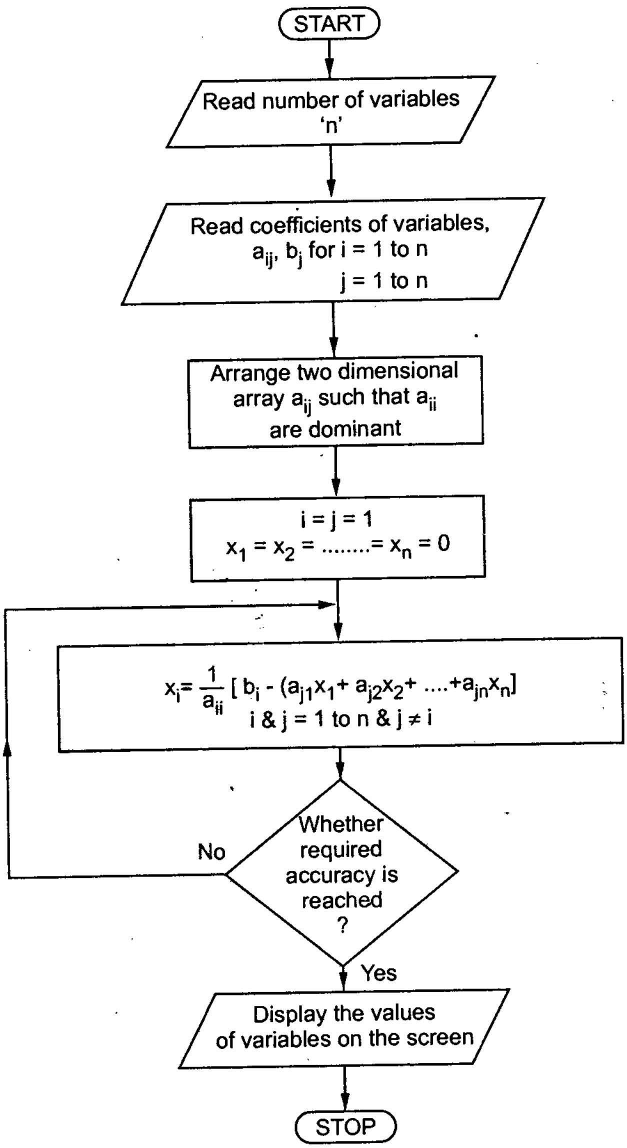


Fig. 4.7.1 Flowchart for gauss seidel iterative method

Computer program :

The source code of the 'C' program is shown below.

```
/* Download this program from  www.vtubooks.com                           */
/* File name : g_siedel.cpp                                                */

/*------- GAUSS SIEDEL ITERATION METHOD TO SOLVE LINEAR EQUATIONS -----*/

/*      THE PROGRAM SOLVES THE SYSTEM OF LINEAR EQUATIONS USING

        GAUSS SIEDEL ITERATION METHOD.
```

```
        INPUTS :  1) Number of variables in the equation.

                  2) Coefficient's of linear equations.

        OUTPUTS : Results of every iteration till 'q' is pressed.          */

/*------------------------------   PROGRAM  -------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>

void main()

      double a[10][10],x[10],y[10];
            /* ARRAY OF a[n][n] STORING COEFFICIENTS OF EQUATIONS */
      int i,j,n;
      char ch;

      clrscr();

      printf("\n  GAUSS SIEDEL ITERATION METHOD TO SOLVE LINEAR EQUATIONS");

      printf("\n\n              The form of equations is as follows\n\n"
             "                  a11x1 + a12x2 + ... + a1nxn = b1\n"
             "                  a21x1 + a22x2 + ... + a2nxn = b2\n"
             "                  a31x1 + a32x2 + ... + a3nxn = b3\n"
             "                  ..............................\n"
             "                  an1x1 + an2x2 + ... + annxn = bn\n");


      printf("\n\nEnter the number of variables (max 10) = ");
                      /* ENTER THE NUMBER OF VARIABLES IN THE EQUATION   */
      scanf("%d",&n);

      for(i = 1; i <= n; i++)
      {             /*  LOOP TO GET COEFFICIENTS a11,a12...,ann & so on  */

          for(j = 1; j <= n; j++)
          {
              printf("a%d%d = ",i,j);  scanf("%lf",&a[i][j]);
          }
          printf("b%d = ",i);  scanf("%lf",&a[i][j]);
          x[i] = y[i] = 0;
      }
      printf("\n\nThe results are as follows....\n\n"
            "press 'enter' key to continue iterations &"
            " press 'q' to stop iterations....\n\n");
      while(ch != 'q')
      {
          for(i = 1; i <= n; i++)
          {           /*  LOOP TO CALCULATE VALUES OF x1,x2,...,xn etc  */

              for(j = 1; j <= n; j++)
              {
                  if(i == j) continue;
                  x[i] = x[i] - a[i][j]*y[j];
              }
              x[i] = x[i] + a[i][j];
              x[i] = x[i]/a[i][i];
              y[i] = x[i];      /*  TAKE VALUES FROM CURRENT ITERATIONS */
          }
          for(i = 1; i <= n; i++)
          {                    /* LOOP TO PRINT VALUES OF x1,x2,...xn etc  */
              y[i] = x[i];
```

```
            printf("x%d = %lf  ",i,x[i]);
            x[i] = 0;
        }
        ch = getch();  printf("\n\n");
    }
}
/*------------------------------- END OF PROGRAM ----------------------*/
```

Before going through this program read the explanation of the program given for gauss elimination method.

As usual, the program first displays the name of the method and then it displays the format of equation being used by the program. The fourth `printf` statement in the program is,

```
printf ("\n\n Enter the number of variables (max 10)=");
scanf ("%d", & n);
```

The first statement asks for value of 'n'. And the second statement gets value of 'n' in the computer. Next there is a `for` loop to get the coefficients a_{11}, a_{12}, a_{13} .... a_{nn} and $b_1, b_2 \ldots b_n$ in the arrays.

Observe that all the statements in this program upto this `for` loop are taken as it is from the program of gauss elimination method.

Next, there is a `while` loop to calculate values of variables and print them on the screen simultaneously. The `while` statement is,

`While (ch!='q')` ← `While` loop is continued till `q' is not pressed.

```
{
    for (   )      First loop
    {
        ---  This loop calculates values of variables, x1 ,x2 ,... xn according
        ---  to equation 4.7.2
    for (   )
    {               Second loop
        ---  This loop displays values of variables on the screen.
    }
    ch=getch(); printf("\n");← This statement takes display of cursor
                                   on next line.
}   This statement is used to sense whether `q' is pressed for stopping iterations.
```

In this loop the first conditional statement is,

`While (ch!='q')`

The meaning of this statement is that keep on executing the statements inside this `while` loop till the key pressed **is not equal to** 'q'. The last statement in this `while` loop is,

```
ch = getch( );
```

Here program waits for key to be pressed from the key board. If you press any other key other than 'q', the program remains in `while` loop and displays the results of next iteration.

Consider the first `for` loop in the `while` loop.

It is reproduced below for explanations.

```
for (i=1; i<=n; i++)        ← This loop is for equations.
{
    for(j=1; j<=n; j++)  ← This loop is for variable.
    {
        if(i==j)continue;  ← if i = j quit this loop.
        x[i]=x[i]-a[i][j]*y[j];  ← This statement calculates
                                   x[i]= -a_i1 x_1 - a_i2 x_2 - ... - a_in x_n & i≠j
    }
    x[i]=x[i]+a[i][j];      ← This statement adds b_i to x[i]
    x[i]=x[i]/a[i][i];      ← This statement gives x_i = x[i]/a_ii
    y[i]=x[i];              ← This statement uses lastest values
}
```

Thus in the above loop a particular equation is iterated as follows –

Consider $$x_1 = \frac{1}{a_{11}}\left[b_1 - a_{12}\,x_2 - a_{13}\,x_3 - a_{14}\,x_4\right]$$

This is iterative equation for 4 variables and gives value of x_1. The statement inside the inner `for` loop,

```
x[i] = x[i] - a[i][j]*y[j];
```

Here i = 1 and `x[i]` $\rightarrow$ x_1

When j goes from 1 to n, `x[i]` will be,

$$\texttt{x[i]} = -a_{12}\,x_2 - a_{13}\,x_3 - a_{14}\,x_4$$

The next statement is,

```
x[i] = x[i] + a[i][j];
```

Here i = 1 and j = n + 1 = 4 + 1 = 5 (after execution of inner `for` loop) and we know that `a[i][5]` = b_1 i.e. in $(n+1)^{th}$ column of matrix a_{ij} where values of b_i are stored. Therefore this statement adds b_1 to the `x[1]`

i.e. $$x[1] = -a_{12}\,x_2 - a_{13}\,x_3 - a_{14}\,x_4 + b_1$$

The next statement is,

```
x[i] = x[i]/a[i][i]
```

$\therefore$ i = 1 This statement gives,

$$x[1] = \frac{-a_{12}\,x_2 - a_{13}\,x_3 - a_{14}\,x_4 + b_1}{a_{11}}$$

$$= x_1 \text{ from the given equation.}$$

The next statement is,

```
y[i] = x[i]; /*TAKE VALUES FROM CURRENT
                ITERATIONS*/
```

Here `x[i]` contains values of current iterations and `y[i]` contains values of previous iteration. The above statement appends `y[i]` after every calculation of x_i and thus it uses latest values.

Next, there is a `for` loop to print values of `x[i]` in every iteration. This loop prints first 'n' values of `x[i]`. That is, it prints,

x[1] = x_1, x[2] = x_2, x[3] = x_3, ... x[n] = x_n

How to Run this program?

Compile the source code given here and make its EXE file. We will use following system of equations to test the program –

$$13x_1 + 5x_2 - 3x_3 + x_4 = 18$$
$$2x_1 + 12x_2 + x_3 - 4x_4 = 13$$
$$3x_1 - 4x_2 + 10x_3 + x_4 = 29$$
$$2x_1 + x_2 - 3x_3 + 9x_4 = 31$$

Thus for these equations,

n = 4 and

$a_{11} = 13,$	$a_{12} = 5,$	$a_{13} = -3,$	$a_{14} = 1$	and	$b_1 = 18$
$a_{21} = 2,$	$a_{22} = 12,$	$a_{23} = 1,$	$a_{24} = -4$	and	$b_2 = 13$
$a_{31} = 3,$	$a_{32} = -4,$	$a_{33} = 10,$	$a_{34} = 1$	and	$b_3 = 29$
$a_{41} = 2,$	$a_{42} = 1,$	$a_{43} = -3,$	$a_{44} = 9$	and	$b_4 = 31$

Run the program on your computer. It first displays the name of the method, then it displays,

`Enter the number of variables (max 10) =` Here enter '4' and press 'enter' key.

Then it displays

`a11 =` Here enter `13' and press `enter' key

`a12 =` 5 ↵

`a13 =` −3 ↵

`a14 =` 1 ↵

```
b1   = 18↵
a21  =
 :              : ← Similarly enter other coefficients.
b4   = 31↵
```

The program then displays the results in every iteration. Press any key to continue and press 'q' to stop.

All the results are shown below combinely.

```
------------------------------ Results ---------------------------

  GAUSS SIEDEL ITERATION METHOD TO SOLVE LINEAR EQUATIONS
             The form of equations is as follows

                  a11x1 + a12x2 + ... + a1nxn = b1
                  a21x1 + a22x2 + ... + a2nxn = b2
                  a31x1 + a32x2 + ... + a3nxn = b3
                  ................................
                  an1x1 + an2x2 + ... + annxn = bn

Enter the number of variables (max 10) = 4
a11 = 13
a12 = 5
a13 = -3
a14 = 1
b1 = 18
a21 = 2
a22 = 12
a23 = 1
a24 = -4
b2 = 13
a31 = 3
a32 = -4
a33 = 10
a34 = 1
b3 = 29
a41 = 2
a42 = 1
a43 = -3
a44 = 9
b4 = 31

The results are as follows....

press 'enter' key to continue iterations & press 'q' to stop iterations....

x1 = 1.384615  x2 = 0.852564  x3 = 2.825641  x4 = 3.983903

x1 = 1.402323  x2 = 1.942110  x3 = 2.857757  x4 = 3.869613

x1 = 0.999470  x2 = 1.968480  x3 = 3.000590  x4 = 4.003817

x1 = 1.011966  x2 = 1.999229  x3 = 2.995720  x4 = 3.996000

x1 = 0.999617  x2 = 1.999087  x3 = 3.000150  x4 = 4.000237

x1 = 1.000367  x2 = 2.000005  x3 = 2.999868  x4 = 3.999874

x1 = 0.999977  x2 = 1.999973  x3 = 3.000009  x4 = 4.000011

x1 = 1.000012  x2 = 2.000001  x3 = 2.999996  x4 = 3.999996

x1 = 0.999999  x2 = 1.999999  x3 = 3.000000  x4 = 4.000000

x1 = 1.000000  x2 = 2.000000  x3 = 3.000000  x4 = 4.000000
-------------------------------------------------------------------
```

The same system of equations we have used for jacobi's method. Compare the two results obtained by computer.

4.7.3 Convergence of Gauss-Seidel Method

The sufficient conditions for the convergence of two nonlinear equations u(x, y) and v(x, y) are,

$$\left|\frac{\partial u}{\partial x}\right| + \left|\frac{\partial v}{\partial x}\right| < 1 \qquad \text{... (4.7.7(a))}$$

and $$\left|\frac{\partial u}{\partial y}\right| + \left|\frac{\partial v}{\partial y}\right| < 1 \qquad \text{... (4.7.7(b))}$$

This criteria is also applicable to linear equations. For two simultaneous linear equations, the gauss-seidel algorithm is given as,

$$u(x_1, x_2) = \frac{1}{a_{11}}[b_1 - a_{12}x_2] \qquad \text{... (4.7.8(a))}$$

and $$v(x_1, x_2) = \frac{1}{a_{22}}[b_2 - a_{21}x_1] \qquad \text{... (4.7.8(b))}$$

Hence, the partial derivatives of above equations are,

$$\frac{\partial u}{\partial x_1} = 0, \quad \frac{\partial u}{\partial x_2} = -\frac{a_{12}}{a_{11}} \quad \text{from equation 4.7.8(a)}$$

and $$\frac{\partial v}{\partial x_2} = 0, \quad \frac{\partial v}{\partial x_1} = -\frac{a_{21}}{a_{22}} \quad \text{from equation 4.7.8(b)}$$

Putting these partial derivatives in equation 4.7.7 we get,

$$\left|\frac{a_{21}}{a_{22}}\right| < 1$$

and $$\left|\frac{a_{12}}{a_{11}}\right| < 1$$

Above equations can also be written as,

$$|a_{22}| > |a_{21}|$$

and $$|a_{11}| > |a_{12}|$$

That is the diagonal elements must be greater than the off diagonal elements in each row. The above criteria can be extended for multiple equations, and it will be,

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \qquad \text{... (4.7.9)}$$

This is the condition for convergence of the gauss seidel method.

University Questions

1. Use the Gauss Seidel method to obtain the solution of the system -

$$\begin{aligned} 3x_1 - 0.1x_2 - 0.2x_3 &= 7.85 \\ 0.1x_1 + 7x_2 - 0.3x_3 &= -19.3 \\ 0.3x_1 - 0.2x_2 + 10x_3 &= 71.4 \end{aligned}$$

[May - 96, May - 98, May - 2000]

2. Develop the Gauss-Siedel iterative scheme for the solution of the system :

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

Iterate upto a maximum of 10 times or upto an accuracy of 0.0001 starting with the initial solution vector $x_0 = 0$. **[Dec - 96, Dec - 99]**

3. Solve the following equation by Gauss-Seidel procedure. The answer should be correct to three significant digits.

$$\begin{aligned} x_1 + 10x_2 + 4x_3 &= 6 \\ 2x_1 - 4x_2 + 10x_3 &= -15 \\ 9x_1 + 2x_2 + 4x_3 &= 20 \end{aligned}$$

[May - 97, May - 99, May - 2001]

4. What is the condition for the convergence of Gauss Siedel method ? Explain. **[Dec - 97, Dec - 2000]**

5. Given the linear equations :

$$\begin{aligned} 4x + y + z &= 5 \\ x + 6y + 2z &= 19 \\ -x - 2y + 5z &= 10 \end{aligned}$$

Obtain the values of x, y and z for three successive iterations using Gauss Siedel method with an initial guess of x = y = z = 0. **[Dec - 97, Dec - 98, Dec - 2000]**

6. Find the solution to 3 decimal places of the system :

$$\begin{aligned} 7x + 52y + 13z &= 104 \\ 83x + 11y - 4z &= 95 \\ 3x + 8y + 29z &= 71 \end{aligned}$$

using Gauss-Siedal method. **[Dec - 2001]**

7. Use the Gauss Seidel method to obtain the solution of the system -

$$\begin{aligned} 0.1x_1 + 7x_2 - 0.3x_3 &= -19.3 \\ 3x_1 - 0.1x_2 - 0.2x_3 &= 7.85 \\ 0.3x_1 - 0.2x_2 + 10x_3 &= 71.4 \end{aligned}$$

[Dec - 2002]

8. Solve the following system of equations using Gauss-Seidel method.

$$\begin{aligned} 6x + 15y + 2z &= 72 \\ x + y + 54z &= 110 \\ 2727x + 6y - z &= 85 \end{aligned}$$

[May - 2003]

9. Write a program in C/C ++ to solve a system of 'n' linear equations. **[May - 2004]**

4.8 Error Analysis

In the method to solve the system of linear equations, the calculations are not performed exactly. Normally during computation the numbers are rounded or chopped. This produces round off errors in computation. Therefore the result produced by methods are different slightly from the exact solution.

Let exact solution matrix be denoted by X and corresponding approximate solution matrix be denoted by $\hat{X}$. Then we can write the two equations as,

$$A X = B \qquad \text{... (4.8.1)}$$

and

$$(A + \delta A)\, \hat{X} = (B + \delta B) \qquad \text{... (4.8.2)}$$

Here δA is change in A and δB is change in B due to round off errors during computation.

From equation 4.8.1 X is given as,

$$X = A^{-1} B \qquad \text{... (4.8.3)}$$

and from equation 4.8.2 $\hat{X}$ is given as,

$$\hat{X} = (A + \delta A)^{-1}\ (B + \delta B) \qquad \text{... (4.8.4)}$$

The error matrix (also called as error vector) is given as,

$$\text{Error matrix } \varepsilon = \hat{X} - X$$

Putting values of $\hat{X}$ and X we get,

$$\begin{aligned} \varepsilon &= (A + \delta A)^{-1} (B + \delta B) - A^{-1} B \\ &= \left[(A + \delta A)^{-1} - A^{-1}\right] B + (A + \delta A)^{-1}\, \delta B \end{aligned} \qquad \text{... (4.8.5)}$$

In the vector and matrix forms we can write,

$$\frac{\|\hat{X} - X\|}{\|X\|} \leq \frac{K(A)}{\left(1 - \|A^{-1}\,\delta A\|\right)} \left\{\frac{\|\delta B\|}{\|B\|} + \frac{\|\delta A\|}{\|A\|}\right\}$$

Here $K(A) = \|A^{-1}\|\ \|A\|$ is the condition number of the matrix A.

L.H.S. of equation gives overall relative error in X.

$\dfrac{\|\delta B\|}{\|B\|}$ is overall relative error in B and $\dfrac{\|\delta B\|}{\|A\|}$ is overall relative error in A.

4.9 Comparison and Choice of Methods

The following points are considered for selection of a method for application.

1. For large systems of equations iterative methods are faster than direct methods.
2. For small system it is preferable to use direct methods.
3. Gauss seidel is twice as fast as jacobi's method.

4. Triangularization method and cholesky methods are preferred for computer softwares.
5. It is necessary to get an information if the system is ill conditioned.
6. If proper method is not selected it may also induce instability.

Let's see the comparison of the methods.

The following table shows comparison of direct and iterative method.

Sr.No.	Direct methods	Iterative methods
1.	These methods eliminate variables and transfer set of equations to triangular form.	These methods use successive approximations to get the final answer.
2.	The answers of these methods are exact.	The answers of these methods are approximate.
3.	The examples are Gauss elimination, Backward substitution, Gauss jordan etc.	The examples are Gauss seidel, Jacobi's method etc.
4.	Algorithm of implementation are complex.	Algorithm of implementations are simple.
5.	Less steps to get the answer.	More steps to get the answer.

4.10 Engineering Applications

In this chapter we studied various methods to solve the system of linear equations. Now we will apply these methods to solve the circuit analysis problems.

Ex. 4.10.1 *Fig. 4.10.1 shows an electrical circuit. Write the loop equations for this circuit and obtain the loop currents using gauss seidel iterative method.*

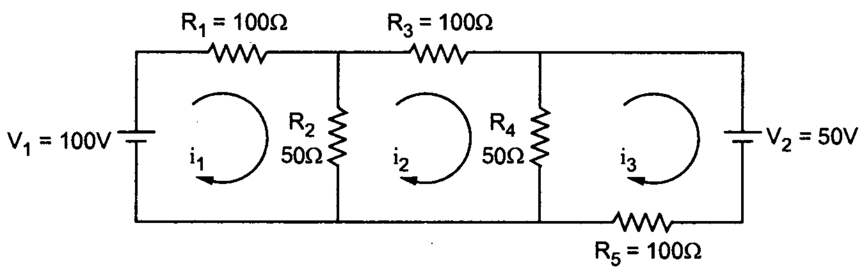


Fig. 4.10.1 An electrical circuit

Sol. : i) To obtain loop equations :

Observe that there are three loops in the given circuit. Hence we can write three loop equations as follows :

$$\text{Left loop : } R_1\, i_1 + R_2(i_1 - i_2) - V_1 = 0$$

$$\text{Middle loop : } R_2\,(i_2 - i_1) + R_3 i_2 + R_4(i_2 - i_3) = 0$$

$$\text{Right loop : } R_4\,(i_3 - i_2) + R_5 i_3 + V_2 = 0$$

Rearranging the above equations we get,

$$(R_1 + R_2)i_1 - R_2 i_2 = V_1$$

$$-R_2\, i_1 + (R_2 + R_3 + R_4)i_2 - R_4\, i_3 = 0$$

$$-R_4\ i_2 + (R_4 + R_5)\ i_3 = -V_2$$

Putting the values of resistances and voltages in above equations from Fig. 4.10.1,

$$\begin{aligned} (100+50)\ i_1 - 50\ i_2 &= 100 \\ -50i_1 + (50+100+50)i_2 - 50i_3 &= 0 \\ -50i_2 + (50+100)i_3 &= -50 \end{aligned}$$

Simplifying the above equations,

$$\left.\begin{aligned} 150i_1 \quad - 50i_2 &= 100 \\ -50i_1 + 200i_2 \quad -50i_3 &= 0 \\ -50i_2 \quad +150i_3 &= -50 \end{aligned}\right\} \qquad \text{... (4.10.1)}$$

Observe that there are three equations and three unknowns. These are simultaneous linear equations. The loop currents i_1, i_2 and i_3 can be obtained using any method discussed in this chapter.

ii) To solve the loop equations using gauss seidel method

The system of equation 4.10.1 can be written as per equation 4.7.2 i.e.,

$$\left.\begin{aligned} i_1^{(k+1)} &= \frac{1}{150}[100 + 50i_2^{(k)}] \\ i_2^{(k+1)} &= \frac{1}{200}[50\ i_1^{(k+1)} + 50i_3^{(k)}] \\ i_3^{(k+1)} &= \frac{1}{150}[-50 + 50i_2^{(k+1)}] \end{aligned}\right\} \qquad \text{.... (4.10.2)}$$

The above equations can be solved using gauss seidel iterative.

Iteration No. 1 :

Let k = 0 in equation 4.10.2 and $i_1^{(0)} = i_2^{(0)} = i_3^{(0)} = 0$. then we get,

$$i_1^{(1)} = \frac{1}{150}[100 + 50i_2^{(0)}] = \frac{1}{150}[100 + 50(0)] = 0.666667$$

$$i_2^{(1)} = \frac{1}{200}[50\ i_1^{(1)} + 50i_3^{(0)}]$$

$$= \frac{1}{200}\ [50\ (0.666667) + 50(0)] = 0.166667$$

$$i_3^{(1)} = \frac{1}{150}[-50 + 50i_2^{(1)}]$$

$$= \frac{1}{150}\ [-\ 50 + 50\ (0.166667)] = -\ 0.277778$$

Iteration No. 2 :

Let k = 1 in equation 4.10.2 and substitute respective values,

$$i_1^{(2)} = \frac{1}{150}[100 + 50i_2^{(1)}] = \frac{1}{150}\ [100 + 50\ (0.666667)] = -\ 0.722222$$

$$i_2^{(2)} = \frac{1}{200}[50\, i_1^{(2)} + 50 i_3^{(1)}]$$

$$= \frac{1}{200}[5\,0\ (0.722222) + 50\,(-0.277778)] = 0.111111$$

$$i_3^{(2)} = \frac{1}{150}[-50 + 50 i_2^{(2)}]$$

$$= \frac{1}{150}[-50 + 50(0.111111)] = -\ 0.296296$$

Similarly next iterations can be performed. Following table lists the details of next iterations :

Table 4.10.1 Results of gauss seidel method

Iteration No.	k	i_1	i_2	i_3
3	2	0.703704	0.101852	– 0.299383
4	3	0.700617	0.100309	– 0.299897
5	4	0.700103	0.100051	– 0.299983
6	5	0.700017	0.100009	– 0.299997
7	6	0.700003	0.100001	– 0.300000

Thus at the end of 7^{th} iteration, the current values in the given electric circuit are,

$$i_1 = 0.700003$$
$$i_2 = 0.100001$$
$$i_3 = -\ 0.300000$$

Ex 4.10.2 *Use Gauss Elimination method to find i_1, i_2 and i_3 for the following circuit (All resistances in ohms).*

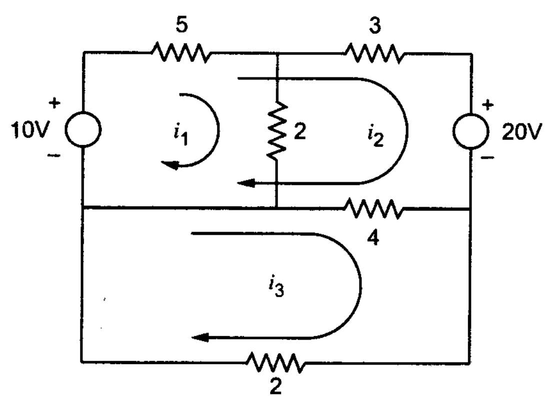


Fig. 4.10.2 Circuit of ex.4.10.2

[Dec-2004, 10 Marks]

Sol. : (i) To obtain loop equations

Applying KVL to the loops in above circuit,

$$5(i_1+i_2)+2i_1 = 10$$
$$5(i_1+i_2)+3i_2+4(i_2-i_3) = 10-20$$
$$4(i_3-i_2)+2i_3 = 0$$

Simplifying the above equations we get,

$$7i_1+5i_2 = 10$$
$$5i_1+12i_2-4i_3 = -10$$
$$-4i_2+6i_3 = 0$$

(ii) To solve equations

Above system of equations can be solved using any method discussed earlier. The currents obtained are,

$$i_1 = 3.55$$
$$i_2 = -2.97$$
$$i_3 = -1.98$$

Ex 4.10.3 *Use Gauss elimination method to find i_1, i_2, i_3. What are the pit falls of naive Gauss elimination method (All resistors in ohms).* **[Dec-2002, 10 Marks]**

Sol. : (i) To obtain loop equations

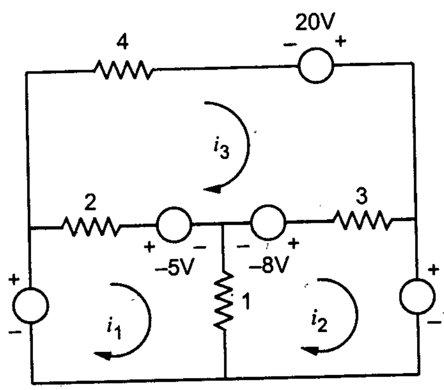


Fig. 4.10.3 Circuit of ex 4.10.3

Applying KVL to the three loops we get,

$$2(i_1-i_3)-5+1(i_1-i_2) = 0$$
$$1(i_2-i_1)-(-8)+3(i_2-i_3) = 0$$
$$4i_3-20+3(i_3-i_2)-8-(-5)+2(i_3-i_1) = 0$$

Simplifying and rearranging above equations,

$$3i_1 - i_2 - 2i_3 = 5$$
$$-i_1 + 4i_2 - 3i_3 = -8$$
$$-2i_1 - 3i_2 + 9i_3 = 23$$

(ii) Solution of equations

Above equations can be solved using Gauss elimination method or any other numerical method. The solution is as follows :

$$i_1 = 6.0909$$
$$i_2 = 3.2727$$
$$i_3 = 5$$

Exercise

1. Currents in the circuit are given by the following equations :

$$28i_1 - 3i_2 = 10$$
$$-3i_1 + 38i_2 - 10i_3 - 5i_5 = 0$$
$$-10i_2 + 25i_3 - 15i_4 = 0$$
$$-15i_3 + 45i_4 = 0$$
$$-5i_2 + 30i_5 = 0$$

Estimate the currents using gauss elimination method.

[Dec.-95 8 Marks, May-99 10 Marks]

Ans. : $i_1 = 0.360748$, $i_2 = 0.033645$, $i_3 = 0.016822$, $i_4 = 0.005607$, $i_5 = 0.005607$

University Questions

1. Use Gauss elimination method to find i_1, i_2, i_3. What are the pit falls of naive Gauss elimination method (All resistances in ohms). **[Dec-2002]**

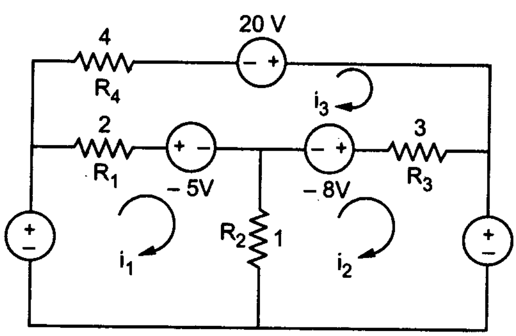


Fig. 4.10.4

2. Use Gauss Elimination to find i_1, i_2, i_3 for the following circuit (All resistance in ohms).

[Dec-2004]

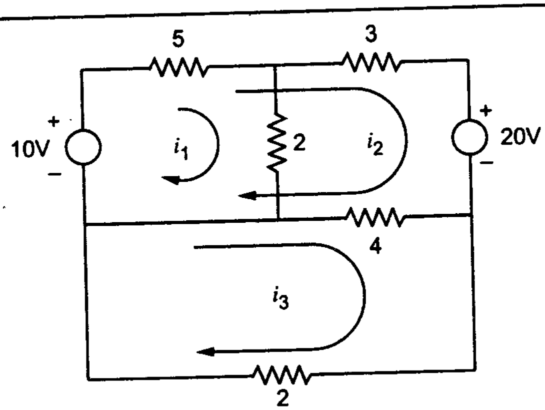


Fig. 4.10.5

4.11 MATLAB to Solve Linear Equations

Now let us see how MATLAB can be used to handle linear algebraic equations. A MATLAB program is shown below to solve the system of linear equations.

```
%   Download this program from www.vtubooks.com
%   file name : gaussElimn.m

%   Solution of the linear algebraic equations. Matlab Version 6
%   This program solves the system of linear algebraic equations
%   Inputs : Coefficients of the system of equations. For 3 variables
%                  the equations are given as follows:

%                      a11x1 + a12x2 + a13x3 = b1
%                      a21x1 + a22x2 + a23x3 = b2
%                      a31x1 + a32x2 + a33x3 = b3

%                  The above coefficients should be entered as follows:

%                      A = [a11 a12 a13; a21 a22 a23; a31 a32 a33]
%                      B = [b1; b2; b3]

%   Output: Dsiplay of solution of the equations

%-----------------------------------------------------------------------
clc                                               % clear the screen
disp('      Solution of the linear algebraic equations'); disp(' ');
disp('The equations are in the form  AX = B');

A = input('Enter the matrix A = ')            %  Matrix A is entered here
B = input('Enter the matrix B = ')            %  Matrix B is entered here

X = A\B;                        % This obtains solution using gauss elimination

disp('The solution is as follows:'); disp(' ');
disp(X);                                      % Display of the solution

%--------------- End of the program -----------------------------------
```

As shown in the above program,

```
A = input ('Enter the matrix A = ')
```

This statement accepts the matrix A, i.e. coefficients of equations, similarly, the next statement,

```
B = input ('Enter the matrix B = ')
```

This statement accepts the B matrix, i.e. right handside of the equations. The next statement of the program is,

```
X = A\B ;
```

This statement solves the equations using gauss elimination. Note that the system of equations can also be solved using $X = A^{-1}B$. This can be written in MATLAB as,

```
x = inv(A)*B;
```

The next statements in the program display the solution of the equations.

To test the program

Let us consider the linear equations given in example 4.6.2. i.e.,

$$\begin{aligned} 7x + 2y - 5z &= -18 \\ x + 5y - 3z &= -40 \\ 2x - y - 9z &= -26 \end{aligned}$$

Here, $A = \begin{bmatrix} 7 & 2 & -5 \\ 1 & 5 & -3 \\ 2 & -1 & -9 \end{bmatrix}$ and $B = \begin{bmatrix} -18 \\ -40 \\ -26 \end{bmatrix}$

The MATLAB accepts above matrices as follows :

```
A = [7 2 -5; 1 5 -3; 2 -1 -9]
B = [-18;  -40;  -26;]
```

The complete display of MATLAB program is given below :

```
%---------------------- Results ------------------------------------------
      Solution of the linear algebraic equations

The equations are in the form  AX = B
Enter the matrix A = [7 2 -5; 1 5 -3; 2 -1 -9]

A =

     7     2    -5
     1     5    -3
     2    -1    -9

Enter the matrix B = [-18; -40; -26]

B =

   -18
   -40
   -26

The solution is as follows:

    2.0000
   -6.0000
    4.0000
%-------------------------------------------------------------------------
```

As shown in the above results, the solution of the equations is,

x = 2, y = – 6 and z = 4

Note that the same answer is obtained in examp!^ 4.6.2.

MATLAB program for LU decomposition

Now let us consider a MATLAB program for LU decomposition of linear equations. The program is given below :

```
%   Download this program from www.vtubooks.com
%   file name : LUdecomp.m

%   LU decomposition of a system linear algebraic equations.
%                                                Matlab Version 6
%   This program obtains the LU decomposition of the system
%                                        of linear algebraic equations
%   Inputs : Coefficients of the system of equations. For 3 variables
%            the equations are given as follows:

%                a11x1 + a12x2 + a13x3 = b1
%                a21x1 + a22x2 + a23x3 = b2
%                a31x1 + a32x2 + a33x3 = b3


%            The above coefficients should be entered as follows:

%                A = [a11 a12 a13; a21 a22 a23; a31 a32 a33]
%                B = [b1; b2; b3]

%   Output: Dsiplay of L and U matrices

%-------------------------------------------------------------------------
clc                                         % clear the screen
disp('  LU decomposition of the linear algebraic equations'); disp(' ');
disp('The equations are in the form  AX = B');

A = input('Enter the matrix A = ')          %  Matrix A is entered here
B = input('Enter the matrix B = ')          %  Matrix B is entered here

[L,U] = lu(A);                            % This obtains L and U matrices

disp('The LU components are as follows:'); disp(' ');
disp('L = '); disp(L);                      % Display of the L matrix
disp('U = '); disp(U);                      % Display of the U matrix
%-------------- End of the program ------------------------------------
```

In the above program, the two `input` statements get the matrices `A` and `B`. Then the next statement is,

`[L,U] = lu(A);`

This statement generates the L and U matrices such that LU = A. Further, the program displays L and U matrices.

To test the program

let us consider again the system of equations given in example 4.6.2. i.e.,

$$\begin{aligned} 7x+2y-5z &= -18 \\ x+5y-3z &= -40 \\ 2x-y-9z &= -26 \end{aligned}$$

Here, $$A = \begin{bmatrix} 7 & 2 & -5 \\ 1 & 5 & -3 \\ 2 & -1 & -9 \end{bmatrix} \text{ and } B = \begin{bmatrix} -18 \\ -40 \\ -26 \end{bmatrix}$$

The above matrices are given as input to the program. Then the L and U components generated by the MATLAB program are as given below :

```
%--------------------- Results ----------------------------------------
  LU decomposition of the linear algebraic equations

The equations are in the form  AX = B
Enter the matrix A = [7 2 -5; 1 5 -3; 2 -1 -9]

A =

     7     2    -5
     1     5    -3
     2    -1    -9

Enter the matrix B = [-18; -40; -26]

B =

   -18
   -40
   -26

The LU components are as follows:

L =
    1.0000         0         0
    0.1429    1.0000         0
    0.2857   -0.3333    1.0000

U =
    7.0000    2.0000   -5.0000
         0    4.7143   -2.2857
         0         0   -8.3333
%------------------------------------------------------------------------
```

Observe that the L and U matrices given in above results are same as those obtained in example 4.6.2 i.e.,

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 1/7 & 1 & 0 \\ 2/7 & -1/3 & 1 \end{bmatrix}$$

and
$$U = \begin{bmatrix} 7 & 2 & -5 \\ 0 & 33/7 & -16/7 \\ 0 & 0 & -8.33 \end{bmatrix}$$

Here note that the diagonal elements of matrix A must be dominant. This is the main requirement of the MATLAB program.

Computer Exercise

1. Develop the program for LU decomposition method.
2. Develop the program for matrix inversion method.
3. Develop the computer program for gauss jordan method.
4. Make your programs interactive so that they can have selection of a method and other facilities.

❑❑❑

5
Curve Fitting

5.1 Introduction

Interpolation means to find a function $f(x)$ which passes through a given set of points. Consider the data, x and $f(x)$ given below in the table.

x	$y = f(x)$
1	15
2	28
3	42
4	60
5	81

By looking at this table we cannot directly find the value of $y = f(x)$ at $x = 2.5$.

But if we know the polynomial which passes through all these points (1, 15), (2, 28), (3, 42), (4, 60) and (5, 81); then we can easily find the value of y at $x = 2.5$. Approximate curve passing through all these points can be obtained graphically (See Fig. 5.1.1)

The approximate curve of $f(x)$ obtained by graphical interpolation through all the points is shown in Fig. 5.1.1. As shown by dotted lines, the approximate value of y at $x = 2.5$ is 36.

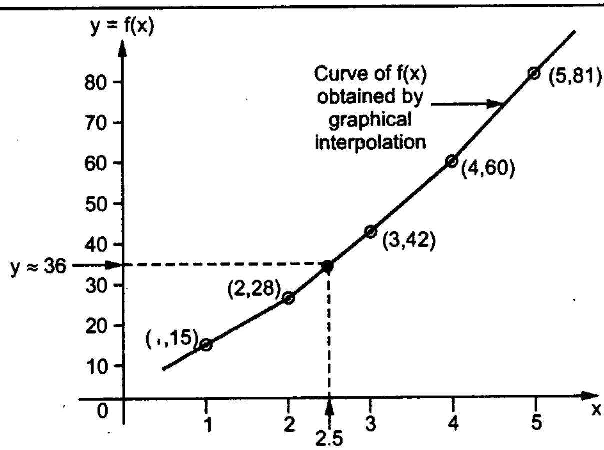


Fig. 5.1.1 Interpolation

Finite differences are used in interpolation techniques. Therefore before going to the methods of interpolation, we will first see the theory of finite differences.

5.2 Finite Difference Operators

Depending upon the type of difference between the two values of y, finite difference operators are defined.

5.2.1 Types of Finite Difference Operators

1. Shift operator (E) :

This operator is defined as,

$$E f(x) = f(x+h) \quad \text{... (5.2.1)}$$

The operator E^{-1} is defined as,

$$E^{-1} f(x) = f(x-h) \quad \text{... (5.2.1 (a))}$$

Thus because of operator E or E^{-1}, the value of function $f(x)$ shifts at $f(x+h)$ or $f(x\ f(x-h)$ respectively.

Here 'h' is the interval by which values of x are separated i.e.

$x_2 - x_1 = x_3 - x_2 = \ldots = h.$

2. Forward difference operator (Δ) :

When forward difference operator (Δ) operates on the function $f(x)$, then it gives the difference defined by following expression,

$$\Delta f(x) = f(x+h) - f(x) \quad \text{... (5.2.2)}$$

3. Backward difference operator (∇) :

When backward difference operator (∇) operates on $f(x)$, then it gives the difference defined by following expressions,

$$\nabla f(x) = f(x) - f(x-h) \quad \text{... (5.2.3)}$$

4. Central difference operator (δ) :

When central difference operator (δ) operates on $f(x)$, then the central difference expression is given as,

$$\delta f(x) = f\left(x+\frac{h}{2}\right) - f\left(x-\frac{h}{2}\right) \quad \text{... (5.2.4)}$$

5. Averaging operator (μ) :

When this operator operates on function $f(x)$, the expression is given as,

$$\mu f(x) = \frac{f\left(x+\frac{h}{2}\right) + f\left(x-\frac{h}{2}\right)}{2} \quad \text{... (5.2.5)}$$

5.2.2 Relationship Between Operators

In this section we will see the relationship between various types of operators.

1. We know that

$$\Delta f(x) = f(x+h) - f(x) \qquad \text{From equation 5.2.2}$$
$$= Ef(x) - f(x) \qquad (\because Ef(x) = f(x+h) \text{ from equation 5.2.1})$$
$$= (E-1)f(x)$$

Comparing the operators of $f(x)$ on both sides,

$$\boxed{\Delta = E-1} \qquad \text{... (5.2.6)}$$

or

$$\boxed{E = 1+\Delta} \qquad \text{... (5.2.7)}$$

2. From equation 5.2.3 we know that,

$$\nabla f(x) = f(x) - f(x-h)$$
$$= f(x) - E^{-1} f(x) \qquad [\because E^{-1} f(x) = f(x-h)]$$

Comparing the operators of $f(x)$ on both sides we get,

$$\nabla = 1 - E^{-1} \qquad \text{... (5.2.8)}$$

or

$$E^{-1} = 1 - \nabla \qquad \text{... (5.2.9)}$$

3. From equation 5.2.1 we know that,

$$Ef(x) = f(x+h) \qquad \text{... (5.2.10)}$$

In this expression replace x by $x+h$,

$$Ef(x+h) = f(x+h+h)$$

i.e.

$$Ef(x+h) = f(x+2h)$$

From equation 5.2.10 we know that $f(x+h) = Ef(x)$. Putting this value of $f(x+h)$ in above equation, we get

$$E \cdot Ef(x) = f(x+2h)$$
$$\therefore \quad E^2 f(x) = f(x+2h) \qquad \text{... (5.2.11)}$$

Again replace x by $x+h$ in above equation,

$$E^2 f(x+h) = f(x+h+2h)$$

i.e.

$$E^2 Ef(x) = f(x+3h) \qquad [\because Ef(x+h) = Ef(x)]$$
$$\therefore \quad E^3 f(x) = f(x+3h) \qquad \text{... (5.2.12)}$$

The generalized form of equation 5.2.11 and equation 5.2.12 is,

$$\boxed{E^n f(x) = f(x+nh)} \qquad \text{... (5.2.13)}$$

Similarly,

$$E^{-1} f(x) = f(x-h)$$
$$\therefore \quad E^{-2} f(x) = f(x-2h)$$

$\therefore$ Generalizing this equation,

$$E^{-n} f(x) = f(x-nh) \qquad \text{... (5.2.14)}$$

4. From equation 5.2.13 we know that,

$$E^{n} \cdot f(x) = f(x+nh) \qquad \dots (5.2.15)$$

From definition of central difference operator given by equation 5.2.4 we know that,

$$\delta f(x) = f\left(x+\frac{h}{2}\right) - f\left(x-\frac{h}{2}\right)$$

i.e.

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

$$= E^{1/2} f(x) - E^{-1/2} f(x) \quad \text{(From equation 5.2.15)}$$

$$= \left(E^{1/2} - E^{-1/2}\right) f(x)$$

Comparing the operators of f(x) on both sides,

$$\boxed{\delta = E^{1/2} - E^{-1/2}} \qquad \dots (5.2.16)$$

5. From the definition of averaging operator μ given by equation 5.2.5 we know that,

$$\mu f(x) = \frac{f\left(x+\frac{h}{2}\right) + f\left(x-\frac{h}{2}\right)}{2}$$

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

$$= \frac{1}{2}\left[E^{1/2} f(x) + E^{-1/2} f(x)\right] \quad \text{(From equation 5.2.15)}$$

$$= \frac{1}{2}\left[E^{1/2} + E^{-1/2}\right] f(x)$$

Comparing the operators of f (x) on both sides, we get,

$$\boxed{\mu = \frac{1}{2}\left[E^{1/2} + E^{-1/2}\right]} \qquad \dots (5.2.17)$$

6. Multiply equation 5.2.16 and equation 5.2.17 we get,

$$\mu\delta = \frac{1}{2}\left[E^{1/2} + E^{-1/2}\right]\left[E^{1/2} - E^{-1/2}\right]$$

$$= \frac{1}{2}\left[E - E^{-1}\right] \qquad \dots (5.2.18)$$

From equation 5.2.9,

$$E^{-1} = 1 - \nabla$$

& From equation 5.2.7,

$$E = 1 + \Delta$$

Putting those values of E & E^{-1} in equation 5.2.18 above we get,

$$\mu\delta = \frac{1}{2}[(1+\Delta)-(1-\nabla)]$$

$$= \frac{1}{2}(\Delta+\nabla)$$

$$\boxed{\mu\delta = \frac{1}{2}(\Delta+\nabla)} \quad \text{... (5.2.19)}$$

7. From equation 5.2.16 we know that

$$\delta = E^{1/2} - E^{-1/2}$$

$$= E^{1/2}\left(1-E^{-1}\right)$$

$$= E^{1/2}\ \nabla \qquad (\because E^{-1} = 1-\nabla \text{ from equation 5.2.9})$$

Also,

$$\delta = E^{1/2} - E^{-1/2}$$

$$= E^{-1/2}\ (E-1) \qquad \text{By taking } E^{-1/2} \text{ common.}$$

$$= E^{-1/2}\ \Delta \quad (\because \Delta = E-1 \text{ from equation 5.2.6})$$

Thus

$$\delta = E^{1/2}\nabla = E^{-1/2}\Delta \quad \text{...(5.2.20)}$$

8. By definition of shift operator from equation 5.2.1 we know that,

$$E f(x) = f(x+h)$$

In this expression replace x by $x-h$, then we have,

$$E f(x-h) = f(x-h+h)$$

i.e.

$$E f(x-h) = f(x)$$

From equation 5.2.1(a) we know that $E^{-1} f(x) = f(x-h)$. Putting for $f(x-h)$ in above equation,

$$E\ E^{-1} f(x) = f(x)$$

Comparing the operators of $f(x)$ on both the sides we get

$$\boxed{EE^{-1} = 1} \quad \text{...(5.2.21)}$$

9. From equation 5.2.20, we know that,

$$\delta = E^{1/2}\nabla$$

&

$$\delta = E^{-1/2}\ \Delta$$

Multiplying the above two equations,

$$\delta^2 = \left(E^{1/2}\nabla\right) \times \left(E^{-1/2}\Delta\right)$$

$$\delta^2 = \nabla\Delta$$

Thus

$$\boxed{\delta^2 = \nabla\ \Delta} \quad \text{... (5.2.22)}$$

10. From equation 5.2.7 we know that,

$$E = 1+\Delta$$

& from equation 5.2.9 we know that

$$E^{-1} = 1-\nabla$$

Multiplying the above two equations

$$E\,E^{-1} = (1+\Delta)(1-\nabla)$$

$\therefore$ $$E\,E^{-1} = 1 \qquad \text{By equation 5.2.21, we get}$$

$$(1+\Delta)(1-\nabla) = 1$$

Thus $$(1+\Delta)(1-\nabla) = 1 \qquad \text{... (5.2.23)}$$

11. From definition of forward differance operator of equation 5.2.2 we know that,

$$\Delta f(x) = f(x+h)-f(x)$$

i.e. $$f(x) = \Delta^{-1}[f(x+h)-f(x)]$$

$\because$ $$f(x+h) = E f(x)$$

From equation 5.2.1, we can write the above equation as,

$$f(x) = \Delta^{-1}[E f(x)-f(x)]$$

$$= \Delta^{-1}(E-1)f(x)$$

Comparing the operators of $f(x)$ on both sides,

$$\Delta^{-1}(E-1) = 1$$

$$\Delta^{-1}\Delta = 1 \qquad (\because E-1 = \Delta \text{ from equation 5.2.6})$$

Thus $$\boxed{\Delta^{-1}\Delta = 1} \qquad \text{... (5.2.24)}$$

On the same lines we can prove,

$$\nabla^{-1}\nabla = 1 \qquad \text{... (5.2.25)}$$

[Hint : To prove above relation, start from equation 5.2.3, i.e.

$$\nabla f(x) = f(x)-f(x-h)$$

& follow the same procedure as above]

12. From equation 5.2.17 we know that,

$$\mu = \frac{1}{2}\left[E^{1/2}+E^{-1/2}\right]$$

$\therefore$ $$\mu^2 = \mu\times\mu = \frac{1}{2}\left[E^{1/2}+E^{-1/2}\right]\times\frac{1}{2}\left[E^{1/2}+E^{-1/2}\right]$$

$$= \frac{1}{4}\left[E+2E^{1/2}E^{-1/2}+E^{-1}\right]$$

$$= \frac{1}{4}\left[E+2+E^{-1}\right]$$

We can write this equation as,

$$\mu^2 = \frac{1}{4}\left[E+E^{-1}+2\right]$$

$$= \frac{1}{4}\left[E+E^{-1}\right]+\frac{1}{2} \qquad \text{... (5.2.26)}$$

From equation 5.2.16 we know that,

$$\delta = E^{1/2} - E^{-1/2}$$

$$\therefore \quad \delta^2 = \delta\times\delta = \left(E^{1/2}-E^{-1/2}\right)\times\left(E^{1/2}-E^{-1/2}\right)$$

$$= E-2E^{1/2}E^{-1/2}+E^{-1}$$

$$= E-2+E^{-1}$$

Thus $\qquad \delta^2 = E+E^{-1}-2 \qquad$ ... (5.2.27)

or $\qquad E+E^{-1} = \delta^2+2$

Putting this value of $E+E^{-1}$ in equation 5.2.26,

$$\mu^2 = \frac{1}{4}\left(\delta^2+2\right)+\frac{1}{2}$$

$$= \frac{\delta^2}{4}+\frac{1}{2}+\frac{1}{2}$$

$$= \frac{\delta^2}{4}+1$$

Thus $\qquad \mu^2 = 1+\frac{\delta^2}{4} \qquad$ ... (5.2.28)

13. By definition of shift operator we know that,

$$E f(x) = f(x+h)$$

Expanding $f(x+h)$ by taylors series around x,

$$E f(x) = f(x+h)$$

$$= f(x)+h f'(x)+\frac{h^2}{2!}f''(x)+\frac{h^3}{3!}f'''(x)+\ldots.$$

$$= f(x)+h D f(x)+\frac{h^2}{2!}D^2 f(x)+\frac{h^3}{3!}D^3 f(x)+\ldots.$$

$$= \left[1+hD+\frac{h^2D^2}{2!}+\frac{h^3D^3}{3!}+\ldots..\right]f(x)$$

$$= e^{hD} f(x)$$

Comparing the operators of $f(x)$ on both the sides we get,

$$\boxed{E = e^{hD}} \quad \text{... (5.2.29)}$$

14. From equation 5.2.17 we know that,

$$\mu = \frac{1}{2}\left[E^{1/2} + E^{-1/2}\right]$$

$$= \frac{1}{2}\left[\left(e^{hD}\right)^{1/2} + \left(e^{hD}\right)^{-1/2}\right] \quad \text{(From equation 5.2.29)}$$

$$= \frac{e^{hD/2} + e^{-hD/2}}{2}$$

$$= cosh\left(\frac{hD}{2}\right) \quad \left(\because \frac{e^{\theta} + e^{-\theta}}{2} = cosh\,\theta\right)$$

Thus

$$\mu = cosh\left(\frac{hD}{2}\right) \quad \text{... (5.2.30)}$$

Table 5.2.1 summarizes the interrelationships among various operators.

Table 5.2.1 Relationship between the operators

	E	Δ	∇	δ
E	E	$\Delta + 1$	$(1-\nabla)^{-1}$	$1+\frac{1}{2}\delta^2 + \delta\sqrt{(1+\frac{1}{4}\delta^2)}$
Δ	$E-1$	Δ	$(1-\nabla)^{-1}-1$	$\frac{1}{2}\delta^2 + \delta\sqrt{(1+\frac{1}{4}\delta^2)}$
∇	$1-E^{-1}$	$1-(1+\Delta)^{-1}$	∇	$-\frac{1}{2}\delta^2 + \delta\sqrt{(1+\frac{1}{4}\delta^2)}$
δ	$E^{1/2}-E^{-1/2}$	$\Delta(1+\Delta)^{-1/2}$	$\nabla(1-\nabla)^{-1/2}$	δ
μ	$\frac{1}{2}(E^{1/2}+E^{-1/2})$	$(1+\frac{1}{2}\Delta)(1+\Delta)^{1/2}$	$(1-\frac{1}{2}\nabla)(1-\nabla)^{-1/2}$	$\sqrt{(1+\frac{1}{4}\delta^2)}$

Ex. 5.2.1 *Show that* $\delta^2 = \Delta\nabla = \nabla\Delta$

Sol. : In equation 5.2.20 it is proved that,

$$\delta = E^{1/2}\nabla = E^{-1/2}\Delta$$

i.e.

$$\delta = E^{1/2}\nabla \quad \text{and}$$

$$\delta = E^{-1/2}\Delta$$

Multiplying the above two equations,

$$\delta^2 = (E^{1/2}\nabla)\times(E^{-1/2}\Delta) \quad \text{... (5.2.31)}$$

$$\therefore \quad \delta^2 = \Delta\nabla$$

Since product is associative, we can write equation 5.2.31 as,

$$\delta^2 = (E^{1/2}\Delta)\times(E^{-1/2}\nabla)$$

$$\therefore \qquad \delta^2 = \Delta\nabla$$

Thus $$\delta^2 = \nabla\Delta = \Delta\nabla$$

The above relation shows that the product of forward and backward difference operators is associative.

Ex. 5.2.2 *Determine $\Delta f(x)$ for $f(x) = x^3 - 2x + 5$ with $h = 1$.*

Sol. : We know that

$$\Delta f(x) = f(x+h) - f(x) \qquad \text{... (5.2.32)}$$

$$\therefore \qquad f(x+h) = (x+h)^3 - 2(x+h) + 5$$

$$= x^3 + 3x^2h + 3h^2x + h^3 - 2x - 2h + 5$$

Putting h = 1 in above equation,

$$f(x+h) = x^3 + 3x^2 + 3x + 1 - 2x - 2 + 5$$

$$= x^3 + 3x^2 + x + 4$$

Putting for $f(x+h)$ & $f(x)$ in equation 5.2.32 we get,

$$\Delta f(x) = x^3 + 3x^2 + x + 4 - (x^3 - 2x + 5)$$

$$\therefore \qquad \Delta f(x) = 3x^2 + 3x - 1$$

Ex. 5.2.3 *Show that*

$$\mu = \frac{2+\Delta}{2\sqrt{1+\Delta}} = \frac{2-\nabla}{2\sqrt{1-\nabla}} \qquad \text{... (5.2.33)}$$

[Dec - 96, 99, 98, Dec - 2001, May - 2001, Dec - 2003, 4 Marks]

Sol. : From equation 5.2.17 we know that,

$$\mu = \frac{1}{2}\left[E^{1/2} + E^{-1/2}\right]$$

$$= \frac{1}{2}\left[E^{1/2} + \frac{1}{E^{1/2}}\right]$$

$$= \frac{1}{2}\left[\frac{E+1}{E^{1/2}}\right]$$

From equation 5.2.7 we have $E = 1 + \Delta$.

Hence above equation becomes,

$$\mu = \frac{1}{2}\left[\frac{1+\Delta+1}{(1+\Delta)^{1/2}}\right]$$

$$= \frac{2+\Delta}{2\sqrt{1+\Delta}}$$

Consider equation 5.2.17 again. i.e.,

$$\mu = \frac{1}{2}\left[E^{1/2} + E^{-1/2}\right]$$

$$= \frac{1}{2}\left[\frac{1}{E^{-1/2}} + E^{-1/2}\right]$$

$$= \frac{1}{2}\left[\frac{1+E^{-1}}{E^{-1/2}}\right]$$

$$= \frac{1}{2}\left[\frac{1+E^{-1}}{\sqrt{E^{-1}}}\right]$$

From equation 5.2.9 we have $E^{-1} = 1-\nabla$. Hence above equation becomes,

$$\mu = \frac{1}{2}\left[\frac{1+1-\nabla}{\sqrt{1-\nabla}}\right]$$

$$= \frac{2-\nabla}{2\sqrt{1-\nabla}}$$

Thus both the relations are proved.

Ex. 5.2.4 *Show that*

i) $E = 1+\Delta$

ii) $\Delta = \nabla\ (1-\nabla)^{-1}$ *and*

iii) $1+\Delta = (E-1)\ \nabla^{-1}$ **[Dec - 2000, May - 97, 99, Dec - 97]**

Sol. : i) $E = 1+\Delta$ is proved in equation 5.2.7.

ii) Consider RHS : $\nabla\ (1-\nabla)^{-1}$

From equation 5.2.8 we know that $\nabla = 1-E^{-1}$ and $1-\nabla = E^{-1}$. Then above equation becomes,

$$\nabla\ (1-\nabla)^{-1} = \left(1-E^{-1}\right)\left(E^{-1}\right)^{-1}$$

$$= \left(1-E^{-1}\right)E$$

$$= E-1 = \Delta \quad \text{from equation 5.2.6}$$

iii) Consider RHS : $(E-1)\ \nabla^{-1}$

From equation 5.2.8 we know that $\nabla = 1-E^{-1}$.

Hence above equation will be,

$$(E-1)\ \nabla^{-1} = (E-1)\left(1-E^{-1}\right)^{-1}$$

$$= (E-1)\ \frac{1}{\left(1-E^{-1}\right)}$$

$$= (E-1)\frac{E}{(E-1)}$$

$$= E = 1+\Delta \quad \text{from equation 5.2.7}$$

Exercise

Prove the following relations :

1. $\nabla - \Delta = -\Delta\nabla$
2. $\Delta + \nabla = \frac{\Delta}{\nabla} - \frac{\nabla}{\Delta}$
3. $\Delta = \nabla(1-\nabla)^{-1}$
4. $(1+\Delta)(1-\nabla) = 1$
5. $\Delta = \frac{1}{2}\delta^2 + \delta\sqrt{1+\frac{\delta^2}{4}}$
6. $\mu\delta = \frac{1}{2}\Delta E^{-1} + \frac{1}{2}\Delta$
7. $E^{1/2} = \mu + \frac{1}{2}\delta$ & $E^{-1/2} = \mu - \frac{1}{2}\delta$
8. $\sqrt{1+\delta^2\mu^2} = 1+\frac{1}{2}\delta^2$

University Questions

1. Define the forward difference operator Δ, backward difference operator ∇ and the central difference operator δ. Establish the relationship between them and show that :

$$\mu = \frac{2+\Delta}{2\sqrt{(1+\Delta)}} = \frac{2-\nabla}{2\sqrt{(1-\nabla)}}$$

[Dec - 96, Dec - 98, Dec - 99, May - 2001, Dec - 2001]

2. Define the operators Δ, ∇, δ and E, show that

   i) $E = 1+\Delta$ ii) $\Delta = \nabla(1-\nabla)^{-1}$ and

   iii) $1+\Delta = (E-1)\nabla^{-1}$ **[May - 97, Dec - 97, May - 99, Dec - 2000, May - 2001]**

3. Define $\Delta, \nabla, \mu, \delta$ and show that

$$\mu = \sqrt{1+\frac{1}{4}\delta^2}$$ **[Dec - 2001]**

4. Prove the following :

   i) $$\mu = \frac{2+\Delta}{2\sqrt{1+\Delta}}$$

   ii) $$\mu\delta = \frac{1}{2}(\Delta+\nabla)$$ **[Dec - 2003]**

5.3 Finite and Other Differences

In the last section we defined various types of difference operators. Here we will discuss those differences in detail.

5.3.1 Forward Differences

From equation 5.2.2 we know that,

$$\Delta f(x) = f(x+h) - f(x) \qquad \text{... (5.3.1)}$$

Here Δ is forward difference operator and $\Delta f(x)$ is called forward difference.

If $x_0, x_1, x_2 \ldots . x_n$ are the values of x, which are separated by 'h' from each other; then we can write,

$$\begin{aligned} x_1 &= x_0 + h \\ x_2 &= x_1 + h = x_0 + h + h = x_0 + 2h \\ x_3 &= x_2 + h = x_0 + 2h + h = x_0 + 3h \\ &\vdots \\ &\vdots \\ x_n &= x_{n-1} + h \quad \text{\& so on} \end{aligned} \qquad \text{... (5.3.2)}$$

Let's write x as x_0 in equation 5.3.1, then we have,

$$\Delta f(x_0) = f(x_0 + h) - f(x_0) \qquad \text{... (5.3.3)}$$

We know that $x_0 + h = x_1$, putting this value in equation above.

$$\Delta f(x_0) = f(x_1) - f(x_0) \qquad \text{... (5.3.4)}$$

Normally, we write,

$$\begin{aligned} y_0 &= f(x_0) \\ y_1 &= f(x_1) \\ y_2 &= f(x_2) \\ &\vdots \\ &\vdots \\ y_n &= f(x_n) \qquad \text{as a short hand notation .} \end{aligned} \qquad \text{... (5.3.5)}$$

With this shorthand notation we can write equation 5.3.4, as

$$\Delta y_0 = y_1 - y_0 \qquad \text{... (5.3.6)}$$

Now replace x by x_1 in equation 5.3.1, we have,

$$\Delta f(x_1) = f(x_1 + h) - f(x_1)$$

We know that $x_2 = x_1 + h$, hence above equation becomes,

$$\Delta f(x_1) = f(x_2) - f(x_1)$$

With shorthand notation above equation becomes,

$$\Delta y_1 = y_2 - y_1 \qquad \text{... (5.3.7)}$$

Similarly we can write other forward differences from equation 5.3.6 and equation 5.3.7 as,

$$\Delta y_2 = y_3 - y_2 \quad \text{\& so on}$$

$$\text{or} \quad \Delta y_n = y_{n+1} - y_n \qquad \text{... (5.3.8)}$$

The forward differences Δy_0, Δy_1, Δy_2 ... etc. defined above are called first order differences.

The second order forward difference $\Delta^2 y_0$ is defined as,

$$\Delta^2 y_0 = \Delta [\Delta y_0]$$

$$= \Delta [y_1 - y_0] \qquad \text{(From equation 5.3.6)}$$

i.e. $\quad \Delta^2 y_0 = \Delta y_1 - \Delta y_0 \qquad$ (Since Δ is linear operator)

Similarly, $\quad \Delta^2 y_1 = \Delta [\Delta y_1]$

$$= \Delta [y_2 - y_1] \qquad \text{(From equation 5.3.7)}$$

i.e. $\quad \Delta^2 y_1 = \Delta y_2 - \Delta y_1$

Similarly, $\quad \Delta^2 y_2 = \Delta y_3 - \Delta y_2$

Here $\Delta^2 y_0$, $\Delta^2 y_1$, $\Delta^2 y_2$, ... etc. are called second order forward differences.

Similarly higher order forward differences are defined as,

$$\Delta^3 y_0 = \Delta^2 y_1 - \Delta^2 y_0 \quad \text{\& so on}$$

$$\Delta^4 y_0 = \Delta^3 y_1 - \Delta^3 y_0, \ \text{... \& so on.}$$

All these differences are shown in Table 5.3.1. This table is called forward difference table. Observe in the table that the forward difference of two values is written at the center of those values. It is written only for convenience.

Table 5.3.1 : Forward difference table

x	y = f (x)	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
x_0	y_0				
		$\Delta y_0 = y_1 - y_0$			
x_1	y_1		$\Delta^2 y_0 = \Delta y_1 - \Delta y_0$		
		$\Delta y_1 = y_2 - y_1$		$\Delta^3 y_0 = \Delta^2 y_1 - \Delta^2 y_0$	
x_2	y_2		$\Delta^2 y_1 = \Delta y_2 - \Delta y_1$		$\Delta^4 y_0 = \Delta^3 y_1 - \Delta^3 y_0$
		$\Delta y_2 = y_3 - y_2$		$\Delta^3 y_1 = \Delta^2 y_2 - \Delta^2 y_1$	
x_3	y_3		$\Delta^2 y_2 = \Delta y_3 - \Delta y_2$		
		$\Delta y_3 = y_4 - y_3$			
x_5	y_4				

Ex.5.3.1 *Calculate forward differences and prepare forward difference table for the following data.*

x	1	11	21	31	41	51	61
y	19.96	39.65	58.81	77.21	94.61	114.67	125.31

Sol. : Using the relations given in Table 5.3.1 we can prepare the forward difference table for this data. Table 5.3.2 shows forward differences of this example.

In the vertical column, where we write the difference value is immaterial. In this table we have written the next order forward difference of two values at the center of these two values in next column. For example we have written $\Delta^2 y_0$ at the center of two value Δy_0 and Δy_1. Observe in the table that the last data element is (x_6, y_6) [actually 7 data points], hence the highest order difference obtained is $\Delta^6 y_0$. Thus $\Delta^7 y$ is not possible here.

Table 5.3.2 Forward difference table of example 5.3.1 on next page.

Ex. 5.3.2 ***Differences of a polynomial***

Show that n^{th} difference of the polynomial of degree n is constant.

[Dec - 2001, 3 marks]

Sol. : Let $f(x)$ be the polynomial of degree n.

i.e.,

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

Then $f(x+h) - f(x)$ will be given as,

$$f(x+h) - f(x) = a_n \left[(x+h)^n - x^n\right] + a_{n-1} \left[(x+h)^{n-1} - x^{n-1}\right]$$
$$+ a_{n-2} \left[(x+h)^{n-2} - x^{n-2}\right] + \dots$$
$$= a_n\, nh\, x^{n-1} + a'_{n-1}\, x^{n-2} + \dots\dots + a'_0$$

Here $a'_{n-1}, a'_{n-2}, \dots\dots, a'_0$ are the new coefficients.

We know that $f(x+h) - f(x) = \Delta f(x)$. Hence,

$$\Delta f(x) = a_n\, nh\, x^{n-1} + a'_{n-1}\, x^{n-2} + \dots + a'_0$$

The above equation is the first order difference of the polynomial of degree n. Note that the degree of above equation is $(n-1)$. Similarly the second difference $\Delta^2 f(x)$ will be of degree $(n-2)$. And the coefficient of x^{n-2} will be $a_n\, n\,(n-1)\, h^2$. Hence the n^{th} difference will be $a_n\, n\,(n-1)\,(n-2) \dots\dots (2)\,(1)\, h^n = a_n\, n\,!\, h^n$. Thus, the n^{th} difference is constant.

Table 5.3.2 Forward differnece table of example 5.3.1

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$	$\Delta^6 y$
$x_0 = 1$	$y_0 = 19.96$						
		$\Delta y_0 = y_1 - y_0$ $= 39.65 - 19.96$ $= 19.69$					
$x_1 = 11$	$y_1 = 39.65$		$\Delta^2 y_0 = \Delta y_1 - \Delta y_0$ $= -0.53$				
		$\Delta y_1 = y_2 - y_1$ $= 58.81 - 39.65$ $= 19.16$		$\Delta^3 y_0 = \Delta^2 y_1 - \Delta^2 y_0$ $= -0.23$			
$x_2 = 21$	$y_2 = 58.81$		$\Delta^2 y_1 = \Delta y_2 - \Delta y_1$ $= -0.76$		$\Delta^4 y_0 = -0.01$		
		$\Delta y_2 = y_3 - y_2$ $= 77.21 - 58.81$ $= 18.40$		$\Delta^3 y_1 = \Delta^2 y_2 - \Delta^2 y_1$ $= -0.24$		$\Delta^5 y_0 = 3.91$	
$x_3 = 31$	$y_3 = 77.21$		$\Delta^2 y_2 = \Delta y_3 - \Delta y_2$ $= -1$		$\Delta^4 y_1 = 3.90$		$\Delta^6 y_0 = -23.55$
		$\Delta y_3 = y_4 - y_3$ $= 94.61 - 77.21$ $= 17.40$		$\Delta^3 y_2 = \Delta^2 y_3 - \Delta^2 y_2$ $= 3.66$		$\Delta^5 y_1 = -19.64$	
$x_4 = 41$	$y_4 = 94.61$		$\Delta^2 y_3 = \Delta y_4 - \Delta y_3$ $= 2.66$		$\Delta^4 y_2 = -15.74$		
		$\Delta y_4 = y_5 - y_4$ $= 114.67 - 94.61$ $= 20.06$		$\Delta^3 y_3 = \Delta^2 y_4 - \Delta^2 y_3$ $= -12.08$			
$x_5 = 51$	$y_5 = 114.67$		$\Delta^2 y_4 = \Delta y_5 - \Delta y_4$ $= -9.42$				
		$\Delta y_5 = y_6 - y_5$ $= 125.31 - 114.67$ $= 10.64$					
$x_6 = 61$	$y_6 = 125.31$						

5.3.1.1 Algorithm and Flow Chart to Generate Forward Differences

Based on the discussion so far we will now discuss an algorithm to generate the forward difference table.

Algorithm :

Step 1 : *Enter the maximum number of data points.*

Step 2 : *Read all the data points (i.e. x and y)*

Step 3 : *Calculate*

$$\Delta y_0 = y_1 - y_0$$
$$\Delta y_1 = y_2 - y_1$$
$$\Delta y_2 = y_3 - y_2$$
$$\vdots$$

Then calculate,

$$\Delta^2 y_0 = \Delta y_1 - \Delta y_0$$
$$\Delta^2 y_1 = \Delta y_2 - \Delta y_1$$
$$\Delta^2 y_2 = \Delta y_3 - \Delta y_2 \quad \text{\& so on}$$

On the same lines calculate, $\Delta^3 y_0$, $\Delta^3 y_1$, $\Delta^3 y_2$, ... *& so on.*

Calculate forward differences successively till there is only one forward difference.

Step 4 : *Display the forward difference table*

Step 5 : *Stop.*

Here we will assume the complete forward difference table as two dimensional array. Thus we will define the table in rows and columns. Then the particular row and column gives the value of a forward difference. Table 5.3.3 shows these rows and columns and corresponding forward differences. (Table 5.3.3 on next page.)

As shown in table, for values of '*x*' an array x [5] is used. Therefore x_0 , x_1 , x_2 ... etc. are stored in this one dimensional array. For the values of y and forward differences a two dimensional array *y* [5] [5] is used. Thus this two dimensional array forms a table of 'i' rows and '*j*' columns. In this table $i = j = 5$. Thus array *x* [5] and array *y* [5] [5] combinely give a forward difference table. Value of *x*, i.e. x_0 , x_1 , x_2 , ... etc. are stored in array *x* [5], where as corresponding values of *y*, i.e. y_1 , y_2, etc. are stored in first column (j = 0) of array *y* [5][5].

Table 5.3.3 Two dimensional array representation of a forward difference table for computer program

x[5] = One Dimension Array

i = 0 x[0] = x_0
i = 1 x[1] = x_1
i = 2 x[2] = x_2
i = 3 x[3] = x_3
i = 4 x[4] = x_4

y = [5] [5] = Two Dimensional Array

i = 0, j = 0 y[0] [0] = y_0	i = 0, j = 1 y[0] [1] = Δy_0 $= y_1 - y_0$ = y[1] [0] – y[0] [0]	i = 0, j = 2 y[0] [2] = $\Delta^2 y_0$ $= \Delta y_1 - y_0$ = y[1] [1] – y[0] [1]	i = 0, j = 3 y[0] [3] = $\Delta^3 y_0$ $= \Delta^2 y_1 - \Delta^2 y_0$ = y[1] [2] – y[0] [2]	i = 0, j = 4 y[0] [4] = $\Delta^4 y_0$ $= \Delta^3 y_1 - \Delta^3 y_0$ = y[1] [3] – y[0] [3]
i = 1, j = 0 y[1] [0] = y_1	i = 1, j = 1 y[1] [1] = Δy_1 $= y_2 - y_1$ = y[2] [0] – y[1] [0]	i = 1, j = 2 y[1] [2] = $\Delta^2 y_1$ $= \Delta y_2 - y_1$ = y[2] [1] – y[1] [1]	i = 1, j = 3 y[1][3] = $\Delta^3 y_1$ $= \Delta^2 y_2 - \Delta^2 y_1$ = y[2][2] – y[1][2]	i = 1, j = 4 No difference
i = 2, j = 0 y[2] [0] = y_2	i = 2, j = 1 y[2] [1] = Δy_2 $= y_3 - y_2$ = y[3] [0] – y[2] [0]	i = 2, j = 2 y[2] [2] = $\Delta^2 y_2$ $= \Delta y_3 - y_2$ = y[3] [1] – y[2] [1]	i = 2, j = 3 No difference	i = 3, j = 4 No difference
i = 3, j = 0 y[3] [0] = y_3	i = 3, j = 1 y[3] [1] = Δy_3 $= y_4 - y_3$ = y[4] [0] – y[3] [0]	i = 3, j = 2 This place will be blank since no difference is calculated here. ∴ y[3] [2] = 0	i = 3, j = 3 No difference	i = 2, j = 4 No difference
i = 4, j = 0 y[4] [0] = y_4	i = 4, j = 1 This place in the array will be blank ∴ y[4] [1] = 0 i.e. No difference.	i = 4, j = 2 No difference ∴ y[4] [2] = 0	i = 4, j = 3 No difference	i = 4, j = 4 No difference

The first forward difference Δy_0 is calculated in the array as,

$$\begin{aligned} \Delta y_0 &= y_1 - y_0 \\ &= y[1][0] - y[0][0] \end{aligned}$$

This difference is stored in first row ($i = 0$) and second column ($j = 1$) i.e.,

$$y[0][1] = \Delta y_0$$

Similarly, $$y[1][1] = \Delta y_1$$

$$y[2][1] = \Delta y_2 \qquad \text{\& so on}$$

The place $y[4][1]$ in the array is empty, since there is no difference to calculate. In the third column ($j = 2$) last two places; i.e. $y[3][2]$ and $y[4][2]$ are empty similarly. And,

$$\begin{aligned} y[0][2] &= \Delta^2 y_0 = \Delta y_1 - \Delta y_0 \\ &= y[1][1] - y[0][1] \end{aligned}$$

Similarly, $$y[1][2] = \Delta^2 y^1 \qquad \text{\& so on.}$$

In the fourth column ($j = 3$) first two places, i.e. $y[0][3] = \Delta^3 y_0$, $y[1][3] = \Delta^3 y_1$ are occupied. And remaining places $y[2][3]$, $y[3][3]$ and $y[4][3]$ are empty.

In the last column ($j = 4$) only first place is occupied. i.e.

$$y[0][4] = \Delta^4 y_0$$

and remaining places are empty.

This is the logic for preparing forward difference table. We have explained this logic for n = 5 elements.

However, it can be further expanded for any other elements. Observe that

For 'n' elements forward difference table,

$$x[n] = \text{One dimensional array of 'n' elements}$$

& $$y[n][n] = \text{Two dimensional array of n * n elements.}$$

Fig. 5.3.1 shows the computer program flowchart for this logic on next page.

5.3.1.2 C Program & Logic Development

Now we will see how to write a 'C' program to implement the logic we discussed in the last sub section.

The program first prints the name of the method and format of the function.

The fourth `printf` statement asks for number of data points 'n'. Here we have defined x and y arrays of (`y[20][20] & x[20]`) 20 elements. Hence maximum '20' elements can be processed by this program. If you want to process more than 20 data points, then increase the size of `x[ ] & y[ ][ ]` arrays.

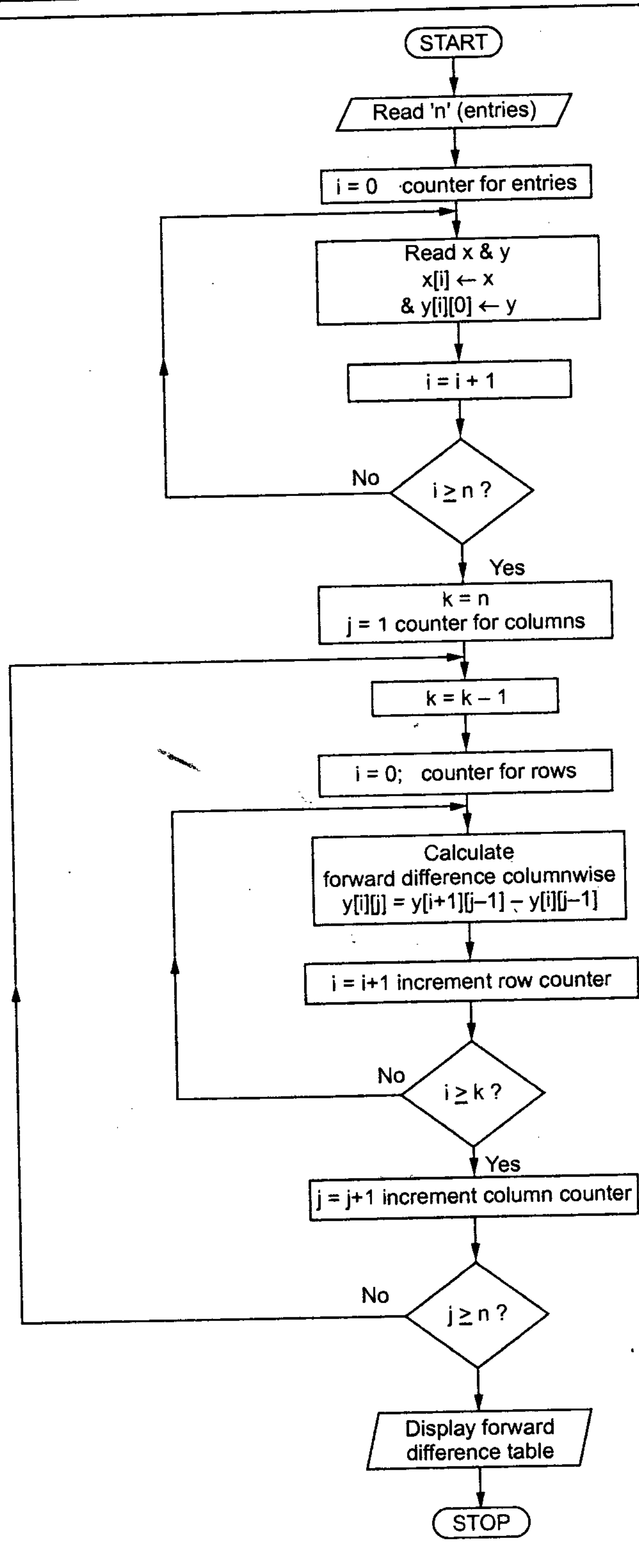


Fig. 5.3.1 Flowchart to generate forward difference table

The next for loop is used to get values of x & y. Observe that all the values of y are entered in first column (y[i][0] i.e. j = 0). This is done by second scanf statement in the for loop. The next for loop calculates the forward differences. The forward differences are calculated columnwise.

Two for loops are required to calculate forward differences.

The first for loop starts with second column (j=1). Number of forward differences goes on decreasing, as we move towards higher order differences. The counter 'k' takes care of this. It is decremented (k = k - 1) whenever j is incremented in for loop.

Observe in Table 5.3.3 that when j = 1, k = 4 i.e. four forward differences of first order. When j = 2, there are three second order differences. Hence k should be '3' for this column.

The next for loop calculates 'k' number of forward differences in the same column. The statement,

```
y[i][j] = y[i+1][j-1]  -  y[i][j-1]
```

Calculates the forward difference. Observe that in this second for loop 'j' remains same and only 'i' is incremented upto 'k'.After calculating all forward differences, the outer for loop then goes to next column. It then decrements value of 'k' and calculates next higher forward order differences.

```
/* Download this program from  www.vtubooks.com                    */
/* File name : for_diff.cpp                                         */

/*-------------- FORWARD DIFFERENCES GENERATION FOR INTERPOLATION -----*/

/*       THE PROGRAM GENERATES A FORWARD DIFFERENCES TABLE FROM GIVEN

         DATA.  THE VALUES OF x AND CORRESPONDING y = f(x) ARE TO BE

         ENTERED IN THE ARRAY FORMAT.

         INPUTS :  1) Number of entries of the data.
                   2) Values of 'x' & corresponding y = f(x).
         OUTPUTS : forward difference table array.                 */

/*------------------------------    PROGRAM  ---------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>

void main()
```

```
{
    double y[20][20],x[20];
                /* ARRAY OF y[n][n] ELEMENTS FOR FORWARD DIFFERENCE TABLE */
    int i,j,k,n;

    clrscr();
    printf("\n      FORWARD DIFFERENCES GENERATION FOR INTERPOLATION");

    printf("\n\n  The form of equation is y = f(x)\n");

    printf("\n\nEnter the number of entries (max 20) = ");
                          /* ENTER THE NUMBER OF ENTRIES IN THE TABLE   */
    scanf("%d",&n);

    for(i = 0; i < n; i++)
    {                     /*  LOOP TO GET x AND y = f(x) IN THE TABLE   */

        printf("x%d = ",i);  scanf("%lf",&x[i]);
        printf("             y%d = ",i);  scanf("%lf",&y[i][0]);
    }
    k = n;
    for(j = 1; j < n; j++)
    {           /*  LOOP TO CALCULATE FORWARD DIFFERENCES IN THE TABLE  */
        k = k - 1;
        for(i = 0; i < k; i++)
        {
            y[i][j] = y[i+1][j-1] - y[i][j-1];
        }
    }

    k = n;    clrscr();
    printf("\n The forward difference table is as follows . . .\n");
    printf("\n\tx\ty\tDy\tD2y\tD3y\tD4y\tD5y\tD6y\n");
    for(i = 0; i < n; i++)
    {           /*  LOOP TO PRINT FORWARD DIFFERENCES IN THE TABLE  */

        printf("\nx%d = %4.2lf",i,x[i]);

        for(j = 0; j < k; j++)
        {
          printf("\t%4.2lf ",y[i][j]);
        }
        k = k - 1;
        printf("\n");
    }
}
/*--------------------  End of program-----------------------------------*/
```

The program then prints those forward differences in array y [ ] [ ]. Here also to print rows and columns of two dimensional array 'y', we need two `for` loops. The outer `for` loop takes care of rows (`i`) and inner `for` loop takes care of columns (`j`).

How to Run this program?

Compile the source code of 'C' program given and make EXE file. Run this EXE file on your computer. Let's use the data elements given in example 5.3.1. There are total '7' data points. After running the program displays name of the program and then it displays,

`Enter the number of entries (max 20) =`

Here enter '7' and press 'enter' key.

The program then asks for actual values of data points i.e.

x0 = Enter '1' and press 'Enter' key

y0 = Enter '19.96' & press 'Enter' key

$x1$ = Enter '11' and press 'Enter' key

$y1$ = Enter '39.65' & press 'Enter' key

$x2$ = 21 ↵ $y2$ = 58.81 ↵

$x3$ = 31 ↵ $y3$ = 77.21 ↵

$x4$ = 41 ↵ $y4$ = 94.61 ↵

$x5$ = 51 ↵ $y5$ = 114.67 ↵

$x6$ = 61 ↵ $y6$ = 125.31 ↵

Here symbol `↵' means press 'enter' key after entering a number before the symbol.

The program then displays a forward difference table.

Here is the display of complete results.

```
------------------------ Results --------------------------------

      FORWARD.DIFFERENCES GENERATION FOR INTERPOLATION

   The form of equation is y = f(x)


 Enter the number of entries (max 20) = 7
 x0 = 1
            y0 = 19.96
 x1 = 11
            y1 = 39.65
 x2 = 21
            y2 = 58.81
 x3 = 31
            y3 = 77.21
 x4 = 41
            y4 = 94.61
 x5 = 51
            y5 = 114.67
 x6 = 61
            y6 = 125.31

  The forward difference table is as follows . . .

         x         y        Dy       D2y      D3y      D4y      D5y      D6y

 x0 = 1.00        19.96    19.69    -0.53    -0.23    -0.01    3.91     -23.55

 x1 = 11.00       39.65    19.16    -0.76    -0.24    3.90     -19.64

 x2 = 21.00       58.81    18.40    -1.00    3.66     -15.74

 x3 = 31.00       77.21    17.40    2.66     -12.08

 x4 = 41.00       94.61    20.06    -9.42

 x5 = 51.00       114.67   10.64

 x6 = 61.00       125.31
----------------------------------------------------------------
```

The first column is x, it displays the values of $x_0, x_1, x_2, \ldots$ etc.

The second column is y. It displays values of $y_0, y_1, y_2 \ldots$ etc.

The third column is Dy. Here 'D' stands for Δy. The first entry in this column is,

$$\Delta y_0 = 19.69$$

Second entry is $\Delta y_1 = 19.16$

The last entry is $\Delta y_5 = 10.64$

Here note that this column will contain '6' first order differences. Hence last place in this column is empty.

The second column is `D2y`. Here `D2y` = $\Delta^2 y$. This is column lists second order differences. The first entry in this column is,

$$\Delta^2 y_0 = -0.53$$

Similarly second entry is $\Delta^2 y_1 = -0.76.$

The last entry is $\Delta^2 y_4 = -9.42$

Note here that there will be '5' second order differences. Hence last two entries in this column will be empty.

The next column is `D3y` = $\Delta^3 y$. `This column lists,`

$$\Delta^3 y_0 = -0.23$$

$$\Delta^3 y_1 = -0.24 \quad \text{\& so on.}$$

Similarly other differences are displayed.

The table displayed here lists exactly the similar values to that we have obtained in Table 5.3.2.

Observe that the displayed difference table is similar to Table 5.3.3.

5.3.2 Backward Differences

From equation 5.2.3 we know that,

$$\nabla f(x) = f(x) - f(x-h) \qquad \text{... (5.3.9)}$$

Let's replace x by x_1 in the above equation.

Then we have,

$$\nabla (f(x_1)) = f(x_1) - f(x_1 - h) \qquad \text{... (5.3.10)}$$

We know that $y = f(x)$

$\therefore$ $y_1 = f(x_1)$

& $x_1 - x_0 = h$ $\quad\because$ x_0, x_1, x_2 etc. are separated by 'h'.

or $x_0 = x_1 - h$

$\therefore$ Equation 5.3.10 becomes,

$$\nabla y_1 = y_1 - y_0 \qquad \text{... (5.3.11)}$$

Here ∇y_1 is called first order backward difference.

Similarly, we can write,

$$\nabla y_2 = y_2 - y_1$$

$$\nabla y_3 = y_3 - y_2 \quad \text{\& so on}$$

Thus $\nabla y_n = y_n - y_{n-1}$ ... (5.3.12)

This is generalized first order backward difference relation.

Now let's consider second order backward difference.

$$\nabla^2 y_2 = \nabla[\nabla y_2]$$

$$= \nabla[y_2 - y_1] \qquad \text{(From equation 5.3.12)}$$

i.e. $$\nabla^2 y_2 = \nabla y_2 - \nabla y_1 \qquad \dots (5.3.13)$$

Similarly, $$\nabla^2 y_3 = \nabla y_3 - \nabla y_2$$

$$\nabla^2 y_4 = \nabla y_4 - \nabla y_3 \quad \text{\& so on.}$$

On the same line we can calculate $\nabla^3 y_1, \nabla^3 y_2, \dots\dots \nabla^4 y_1, \nabla^4 y_2 \dots$ etc.

Here observe that we cannot calculate ∇y_0.

From equation 5.3.12,

$$\nabla y_n = y_n - y_{n-1}$$

When n = 0, $$\nabla y_0 = y_0 - y_{-1}$$

Since y_{-1} is not available, we cannot calculate ∇y_0.

But we can calculate ∇y_n since y_{n-1} is always available. Hence calculation of backward differences is started from bottom side of the data points. Table 5.3.4 illustrates backward differences for five (n = 5) data points.

Table 5.3.4 : Backward difference table

x	y	∇ y	∇^2 y	∇^3 y	∇^4 y
x_0	y_0				
		$\nabla y_1 = y_1 - y_0$			
x_1	y_1		$\nabla^2 y_2 = \nabla y_2 - \nabla y_1$		
		$\nabla y_2 = y_2 - y_1$		$\nabla^3 y_3 = \nabla^2 y_3 - \nabla^2 y_2$	
x_2	y_2		$\nabla^2 y_3 = \nabla y_3 - \nabla y_2$		$\nabla^4 y_4 = \nabla^3 y_4 - \nabla^3 y_3$
		$\nabla y_3 = y_3 - y_2$		$\nabla^3 y_4 = \nabla^2 y_4 - \nabla^2 y_3$	
x_3	y_3		$\nabla^2 y_4 = \nabla y_4 - \nabla y_3$		
		$\nabla y_4 = y_4 - y_3$			
x_4	y_4				

Observe Table 5.3.4 carefully. In this table, the last data point is (x_4, y_4). We get all backward differences at y_4, i.e.

$\nabla y_4, \nabla^2 y_4, \nabla^3 y_4$ & $\nabla^4 y_4$.

Compare this table with forward difference table we prepared given by Table 5.3.1. In this forward difference table we obtain all forward differences at y_0, i.e.

$\Delta y_0, \Delta^2 y_0, \Delta^3 y_0$ & $\Delta^4 y_0$.

These are the major differences between forward and backward differences.

Ex.5.3.2 *Obtain the backward difference table for following set of data points.*

x	1	2	3	4	5
y	2.38	3.65	5.85	9.95	14.85

Sol. : Using relations given in Table 5.3.4 we can generate backward differences shown by Table 5.3.5. (See Table 5.3.5 on next page)

5.3.2.1 Relation Between Forward and Backward Differences

If we want to take forward difference, then,

$$\Delta y_0 = y_1 - y_0 \quad \text{(By definition)}$$

Here we know that,

$$\nabla y_1 = y_1 - y_0$$

Thus $\quad \Delta y_0 = \nabla y_1 \quad$ From above two relations.

i.e. $$\Delta y_{n-1} = \nabla y_n \quad \text{... (5.3.14)}$$

This is true for all forward and backward differences.

5.3.2.2 Algorithm and Flowchart to Generate Backward Differences

Based on the discussion in the last subsection, we will now develop an algorithm to generate backward differences.

Algorithm :

Step 1 : *Read maximum number of data points.*

Step 2 : *Read all the data points (i.e. x & y).*

Step 3 : *Calculate*

$\nabla y_1 = y_1 - y_0$

$\nabla y_2 = y_2 - y_1$

$\nabla y_3 = y_3 - y_2$

& so on

Calculate second order differences as,

$\nabla^2 y_2 = \nabla y_2 - \nabla y_1$

$\nabla^2 y_3 = \nabla y_3 - \nabla y_2$

$\nabla^2 y_4 = \nabla y_4 - \nabla y_3$ *& so on*

Similarly calculate

$\nabla^3 y_3, \nabla^3 y_4, \ldots \nabla^4 y_4, \nabla^4 y_5$ *& so on.*

Step 4 : *Display the backward difference table.*

Step 5 : *Stop.*

Table 5.3.5 Backward difference table for example 5.3.2

x	y	∇y	$\nabla^2 y$	$\nabla^3 y$	$\nabla^4 y$
$x_0 = 1$	$y_0 = 2.38$				
		$\nabla y_1 = y_1 - y_0$ $= 3.65 - 2.32 = 1.27$			
$x_1 = 2$	$y_1 = 3.65$		$\nabla^2 y_2 = \nabla y_2 - \nabla y_1$ $= 2.2 - 1.27 = 0.93$		
		$\nabla y_2 = y_2 - y_1$ $= 5.85 - 3.65 = 2.2$		$\nabla^3 y_3 = \nabla^2 y_3 - \nabla^2 y_2$ $= 1.9 - 0.93 = 0.97$	
$x_2 = 3$	$y_2 = 5.85$		$\nabla^2 y_3 = \nabla y_3 - \nabla y_2$ $= 4.1 - 2.2 = 1.9$		$\nabla^4 y_4 = \nabla^3 y_4 - \nabla^3 y_3$ $= -1.1 - 0.97 = -2.07$
		$\nabla y_3 = y_3 - y_2$ $= 9.95 - 5.85 = 4.1$		$\nabla^3 y_4 = \nabla^2 y_4 - \nabla^2 y_3$ $= 0.8 - 1.9 = -1.1$	
$x_3 = 4$	$y_3 = 9.95$		$\nabla^2 y_4 = \nabla y_4 - \nabla y_3$ $= 4.9 - 4.1 = 0.8$		
		$\nabla y_4 = y_4 - y_3$ $= 14.85 - 9.95 = 4.9$			
$x_4 = 5$	$y_4 = 14.85$				

Flowchart to generate forward difference table :

The similar type of logic, which we have used for generating forward differences will be used here. To store values of x, one dimensional array x [ ] is used. Similarly to store values of y and backward differences a two dimensional array y [ ] [ ] is used.

Table 5.3.6 shows how backward differences are assigned to the array elements. This table is explained for 5 data points.

The calculation of backward difference is started from bottom of first column (j=1) in the table.

Thus, $y[4][1] = \nabla y_4$

$= y_4 - y_3$

i.e. $y[4][1] = y[4][0][3][0]$

Similarly, $y[3][1] = \nabla y_3 = y_3 - y_2$

$= y[3][0] - y[2][0]$

The last difference (first order) is ∇y_1 &

$y[1][1] = \nabla y_1 = y_1 - y_0$

$= y[1][0] - y[0][0]$

As we have seen, ∇y_0 cannot be calculated.

Since, $\nabla y_0 = y_0 - y_{-1}$

& we do not have element y_{-1}. Therefore the place $y[0][1]$ in the array will be empty. Again to calculate second order backward difference we have to start from bottom of next column (j = 2). Thus $\nabla^2 y_4 = y_4 - y_3$ is stored in $y[4][2]$ place of the array. Similarly other backward differences are stored in the two dimensional array.

In the last column (j = 4) there is only one backward difference and it is stored in $y[4][4]$ i.e.,

$$y[4][4] = \nabla^4 y_4$$

And other places are empty in this column. Thus a two dimensional array forms a backward difference table.

This logic is used to prepare a flowchart and 'C' program in the next subsection.

The simplified flowchart is shown in Fig. 5.3.2 to generate backward differences. Observe the difference between the flowchart of Fig. 5.3.1 for forward differences and this flowchart. The flowchart of Fig. 5.3.2 does not give all the steps required for programming in detail, where as the flowchart of Fig. 5.3.1 gives each and every step in detail. Such detailed steps in the flowchart are not required always. Also they make the flowchart complex. Hence simplified flowchart of Fig. 5.3.2 is more flexible and easy to understand.

Table 5.3.6 Two dimensional array representation of a backward difference table for computer program

x[5] = One Dimension Array

y = [5] [5] = Two Dimensional Array

i = 0 x[0] = x_0	i = 0, j = 0 y[0] [0] = y_0	i = 0, j = 1 This place will be empty (No difference)	i = 0, j = 2 No difference	i = 0, j = 3 No difference	i = 0, j = 4 No difference
i = 1 x[1] = x_1	i = 1, j = 0 y[1] [0] = y_1	i = 1, j = 1 y[1] [1] = ∇y_1 $= y_1 - y_0$ = y[1] [0] – y[0] [0]	i = 1, j = 2 No difference	i = 1, j = 3 No difference	i = 1, j = 4 No difference
i = 2 x[2] = x_2	i = 2, j = 0 y[2] [0] = y_2	i = 2, j = 1 y[2] [1] = ∇y_2 $= y_2 - y_1$ = y[2] [0] – y[1] [0]	i = 2, j = 2 y[2] [2] = $\nabla^2 y_2$ $= \nabla y_2 - \nabla y_1$	i = 2, j = 3 No difference	i = 2, j = 4 No difference
i = 3 x[3] = x_3	i = 3, j = 0 y[3] [0] = y_3	i = 3, j = 1 y[3] [1] = ∇y_3 $= y_3 - y_2$ = y[3] [0] – y[2] [0]	i = 3, j = 2 y[3] [2] = $\nabla^2 y_3$ $= \nabla y_3 - \nabla y_2$	i = 3, j = 3 y[3] [3] = $\nabla^3 y_3$ $= \nabla^2 y_3 - \nabla^2 y_2$	i = 3, j = 4 No difference
i = 4 x[4] = x_4	i = 4, j = 0 y[4] [0] = y_4	i = 4, j = 1 y[4] [1] = ∇y_4 $= y_4 - y_3$ = y[4] [0] – y[3] [0]	i = 4, j = 2 y[4] [2] = $\nabla^2 y_4$ $= \nabla y_4 - \nabla y_3$	i = 4, j = 3 y[4] [3] = $\nabla^3 y_4$ $= \nabla^2 y_4 - \nabla^2 y_3$	i = 4, j = 4 y[4] [4] = $\nabla^4 y_4$ $= \nabla^3 y_4 - \nabla^3 y_3$

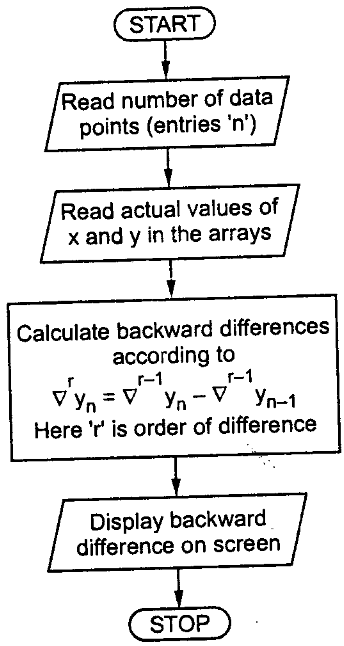


Fig. 5.3.2 Simplified flowchart to generate backward differences

5.3.2.3 C Program and Logic Development

The 'C' program based on the logic discussed in last subsection is shown below.

```
/* Download this program from  www.vtubooks.com                    */
/* File name : bak_diff.cpp                                          */

/*-------------- BACKWARD DIFFERENCES GENERATION FOR INTERPOLATION -----*/

/*      THE PROGRAM GENERATES A BACKWARD DIFFERENCES TABLE FROM GIVEN

        DATA.  THE VALUES OF x AND CORRESPONDING y = f(x) ARE TO BE

        ENTERED IN THE ARRAY FORMAT.

        INPUTS :  1) Number of entries of the data.

                  2) Values of 'x' & corresponding y = f(x).
                                                                    */
        OUTPUTS : Backward difference table array.

/*------------------------------   PROGRAM  ---------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>

void main()
{
      double y[20][20],x[20];
                   /* ARRAY OF N*N ELEMENTS FOR BACKWARD DIFFERENCE TABLE  */
      int i,j,k,n;

      clrscr();
      printf("\n      BACKWARD DIFFERENCES GENERATION FOR INTERPOLATION");
```

```
    printf("\n\n  The form of equation is y = f(x)\n");

    printf("\n\nEnter the number of entries (max 20) = ");
    scanf("%d",&n);          /* ENTER NUMBER OF ENTRIES IN THE TABLE  */

    for(i = 0; i < n; i++)
    {                           /* LOOP TO GET x AND y = f(x) IN THE TABLE */

        printf("x%d = ",i);  scanf("%lf",&x[i]);
        printf("            y%d = ",i);  scanf("%lf",&y[i][0]);
    }
    k = 0;
    for(j = 1; j < n; j++)
    {                           /*  LOOP TO CALCULATE BACKWARD DIFFERENCES  */
        k++;
        for(i = n-1; i >= k; i--)
        {
           y[i][j] = y[i][j-1] - y[i-1][j-1];
        }
    }

    k = n ;    clrscr();
    printf("\n The backward difference table is as follows . . .\n");
    printf("\n\tx\ty\tDy\tD2y\tD3y\tD4y\tD5y\tD6y\n");
    for(i = 0; i < n; i++)
    {                              /*  LOOP TO PRINT BACKWARD DIFFERENCES  */

      printf("\nx%d = %4.2lf",i,x[i]);

      for(j = 0; j < i+1; j++)
      {
        printf("\t%4.2lf ",y[i][j]);
      }
      printf("\n");
    }
}
/*------------------------------End of program---------------------------*/
```

In the function `main`, the first statement is,

`double  y[20][20],  x[20];`

This defines `y` as two dimensional array of 20 rows and 20 columns and `x` as one dimensional array of 20 elements.

The fourth `printf` and succeeding `scanf` statement asks for number of entries or data points for which backward difference is to be calculated.

Then there is a `for` loop to get values of x and y in the arrays.

`x[i]` $\Leftarrow$ values of x &

`y[i][0]` $\Leftarrow$ values of y

Here observe that all values of y are stored in first column (j = 0) of array `y`.

Next, there are two `for` loops to calculate backward differences. They are reproduced below

```
For  (j=1;  j<n;  j++)   [This is columns loop]
{    k++;                    This counter takes care of number of backward
                             differences in every column
     for  (i=n-1;  i>=k;  i--)   [This is rows loop]
     {
          y [i][j]  =  y  [i][j-1]  -  y  [i-1][j-1];
                             This statement calculates backward differences.
     }
}
```

Here first `for` loop is for columns and it starts for first column (`j=1`). The second `for` loop is for rows and its starts always from last row (`i=n-1`); since total rows are n and they are numbered from i = 0 to i = n – 1).

Thus second `for` loop calculates first order backward differences when `j=1`. Since number of backward differences in every column goes on decreasing as order of difference increases, a counter 'k' is used to take care of this. As we move from column 1 to column 2 (`j=2`), k is incremented so that number of backward differences are reduced by one in this column.

Lastly in the program again there are two `for` loops to print backward difference table. It prints the table rowwise.

How to Run this program?

Compile and make EXE file of the 'C' program just discussed. Run this EXE program. The program, after running, first displays the name of the method on the screen.

To calculate the backward differences, we will use the data of example 5.3.2 here. This table is reproduced here for convenience.

x	1	2	3	4	5
y	2.38	3.65	5.85	9.95	14.85

Thus there are '5' entries (data points) in this table.

After running the program it displays,

`Enter the number of entries (max 20) =`

Here enter '5' and press 'enter' keys.

The program then displays,

$x0$ `= Here enter '1' and press 'enter' key.`

$y0$ = Here enter `2.38' & press `enter' key

$x1$ `= Here enter `2' and press `enter' key.`

$y1$ = Here enter `3.65' & press `enter' key

$x1$ = 3 ↵ $y2$ = 5.85 ↵

$x3$ = 4 ↵ $y3$ = 9.95 ↵

$x4$ = 3 ↵ $y4$ = 14.85 ↵

Here symbol `↵' means to press 'enter' key after typing the number before the symbol.

The program then displays the forward difference table. In the table the first line is x y Dy D2y D3y ....

Here $Dy = \nabla y$

$D^2y = \nabla^2 y$ & so on

The first entry under column Dy (or ∇y) is 1.27 (at i = 1),

$\therefore$ $\nabla y_1 = 1.27$

Similarly $\nabla y_2 = 2.20$ & so on.

The first place in this column Dy is empty, means there is no backward difference or ∇y_0 cannot be calculated.

Here is the display of all the results given below.

```
------------------------------- Results -----------------------------

     BACKWARD DIFFERENCES GENERATION FOR INTERPOLATION

  The form of equation is y = f(x)


Enter the number of entries (max 20) = 5
x0 = 1
           y0 = 2.38
x1 = 2
           y1 = 3.65
x2 = 3
           y2 = 5.85
x3 = 4
           y3 = 9.95
x4 = 5
           y4 = 14.85

 The backward difference table is as follows . . .

           x         y        Dy       D2y       D3y       D4y       D5y       D6y

x0 = 1.00          2.38

x1 = 2.00          3.65      1.27

x2 = 3.00          5.85      2.20      0.93

x3 = 4.00          9.95      4.10      1.90      0.97

x4 = 5.00          14.85     4.90      0.80     -1.10     -2.07
----------------------------------------------------------------------
```

Observe that the format of displayed table is similar to Table 5.3.6 we discussed in last subsection. All these backward differences we have calculated in example 5.3.2 and they are listed in Table 5.3.5. Thus all the values of backward differences displayed on the screen are similar to those listed in Table 5.3.5.

You can use this program to calculate backward differences for maximum 20 data points. If you want to increase the data points, then increase the size of arrays x and y in the program.

5.3.3 Divided Differences

In the forward and backward differences we have seen that the values of x are evenly spaced. that is,

$$x_1 - x_0 = x_2 - x_1 = x_3 - x_2 = \ldots x_n - x_{n-1} = h$$

Thus interval 'h' between any two neighbouring values of x is always constant.

But it is also possible that these values of x may not be evenly spaced. The forward and backward differences are defined only for evenly spaced values of x.

If the values of 'x' are not evenly spaced, then 'divided differences' are used.

> *Divided differences are defined for any data irrespective of the spacing between the values of x.*

Let $(x_0, y_0), (x_1, y_1), (x_2, y_2) \ldots$ etc. be the given data points.

The first order divided difference over $(x_0\ x_1)$ is given as,

$$y(x_0\ x_1) \text{ or simply } [x_0\ x_1] = \frac{y_1 - y_0}{x_1 - x_0} \qquad \ldots (5.3.15)$$

Since the ratio of differences is taken, it is independent of spacing between values of x.

$$y(x_0\ x_1) = y(x_1\ x_0)$$

$\therefore$ We can write equation 5.3.15, as

$$y(x_1, x_0) = \frac{y_1 - y_0}{x_1 - x_0} \qquad \ldots (5.3.16)$$

Similarly we can define other differences of the first order as,

$$y(x_2\ x_1) = \frac{y_2 - y_1}{x_2 - x_1} \qquad \ldots (5.3.17)$$

$$y(x_3\ x_2) = \frac{y_3 - y_2}{x_3 - x_2} \qquad \ldots (5.3.18)$$

$$y(x_4\ x_3) = \frac{y_4 - y_3}{x_4 - x_3} \qquad \ldots (5.3.19)$$

& so on.

$\therefore$ The generalized relation for first order divided differences will be,

$$y(x_n\ x_{n-1}) = \frac{y_n - y_{n-1}}{x_n - x_{n-1}} \qquad \ldots (5.3.20)$$

Second order divided differences :

The second order divided difference over $(x_2\ x_1\ x_0)$ is defined as,

$$y(x_2\ x_1\ x_0) = \frac{y(x_2\ x_1) - y(x_1\ x_0)}{x_2 - x_0} \qquad \ldots (5.3.21)$$

Similarly we can define other second order divided differences as,

$$y(x_3\ x_2\ x_1) = \frac{y(x_3\ x_2) - y(x_2\ x_1)}{x_3 - x_1} \qquad \text{... (5.3.22)}$$

$$y(x_4\ x_3\ x_2) = \frac{y(x_4\ x_3) - y(x_3\ x_2)}{x_4 - x_2} \qquad \text{... (5.3.23)}$$

$\therefore$ The generalized relation for second order divided differences will be,

$$y(x_n\ x_{n-1}\ x_{n-2}) = \frac{y(x_n\ x_{n-1}) - y(x_{n-1}\ x_{n-2})}{x_n - x_{n-2}} \qquad \text{... (5.3.24)}$$

Third order divided differences :

The third order divided difference over $(x_3\ x_2\ x_1\ x_0)$ is defined as,

$$y(x_3\ x_2\ x_1\ x_0) = \frac{y(x_3\ x_2\ x_1) - y(x_2\ x_1\ x_0)}{x_3 - x_0} \qquad \text{... (5.3.25)}$$

Similarly we can define other third order divided differences as,

$$y(x_4\ x_3\ x_2\ x_1) = \frac{y(x_4\ x_3\ x_2) - y(x_3\ x_2\ x_1)}{x_4 - x_1} \qquad \text{... (5.3.26)}$$

$\therefore$ The generalized relation for third order divided differences will be,

$$y(x_n\ x_{n-1}\ x_{n-2}\ x_{n-3}) = \frac{y(x_n\ x_{n-1}\ x_{n-2}) - y(x_{n-1}\ x_{n-2}\ x_{n-3})}{x_n - x_{n-3}} \qquad \text{... (5.3.27)}$$

From equation 5.3.20, equation 5.3.24 and equation 5.3.27 the generalized relation for n^{th} order divided difference will be,

$$y(x_n\ x_{n-1}\ x_{n-2} \cdots x_0) = \frac{y(x_n\ x_{n-1} \cdots x_1) - y(x_{n-1}\ x_{n-2} \cdots x_0)}{x_n - x_0} \qquad \text{... (5.3.28)}$$

Table 5.3.7 shows these divided differences for five data points.

(Table 5.3.7 on next page.)

Ex.5.3.3 *Obtain the divided difference table for the following data points.*

x	2	4	9	10
y	4	56	711	980

Sol. : First order divided differences

The divided difference over $x_0\ x_1$ is given by equation 5.3.16 as,

$$y(x_1\ x_0) = y(4, 2) = \frac{y_1 - y_0}{x_1 - x_0}$$

i.e. $$y(4, 2) = \frac{56-4}{4-2} = 26$$

Similarly we can calculate other divided differences as follows.

Table 5.3.7 Divided differences table representation

x	y	$y(x_n x_{n-1})$	$y(x_n x_{n-1} x_{n-2})$	$y(x_n x_{n-1} x_{n-2} x_{n-3})$	$y(x_n x_{n-1} x_{n-2} \ldots\ldots\ldots\ldots x_{n-4})$
x_0	y_0				
		$y(x_1 x_0) = \frac{y_1 - y_0}{x_1 - x_0}$			
x_1	y_1		$y(x_2 x_1 x_0) = \frac{y(x_2 x_1) - y(x_1 x_0)}{x_2 - x_0}$		
		$y(x_2 x_1) = \frac{y_2 - y_1}{x_2 - x_1}$		$y(x_3 x_2 x_1 x_0) = \frac{y(x_3 x_2 x_1) - y(x_2 x_1 x_0)}{x_3 - x_0}$	
x_2	y_2		$y(x_3 x_2 x_1) = \frac{y(x_3 x_2) - y(x_2 x_1)}{x_3 - x_1}$		$y(x_4 x_3 x_2 x_1 x_0) = \frac{y(x_4 x_3 x_2 x_1) - y(x_3 x_2 x_1 x_0)}{x_4 - x_0}$
		$y(x_3 x_2) = \frac{y_3 - y_2}{x_3 - x_2}$		$y(x_4 x_3 x_2 x_1) = \frac{y(x_4 x_3 x_2) - y(x_3 x_2 x_1)}{x_4 - x_1}$	
x_3	y_3		$y(x_4 x_3 x_2) = \frac{y(x_4 x_3) - y(x_3 x_2)}{x_4 - x_2}$		
		$y(x_4 x_3) = \frac{y_4 - y_3}{x_4 - x_3}$			
x_4	y_4				

$$y(x_2\ x_1) = y(9,4) = \frac{y_2 - y_1}{x_2 - x_1}$$

i.e. $$y(9,4) = \frac{711-56}{9-4}$$

$$= 131$$

$$y(x_3\ x_2) = y(10,9) = \frac{y_3 - y_2}{x_3 - x_2}$$

i.e. $$y(10,9) = \frac{980-711}{10-9}$$

$$= 269$$

Second order divided differences :

The second order divided difference over $(x_0\ x_1\ x_2)$ is given by equation 5.3.21 as,

$$y(x_2\ x_1\ x_0) = \frac{y(x_2\ x_1) - y(x_1\ x_0)}{x_2 - x_0}$$

i.e. $$y(9,4,2) = \frac{y(9,4) - y(4,2)}{9-2} = \frac{131-26}{9-2}$$

$$= 15$$

Similarly, $$y(x_3\ x_1\ x_2) = \frac{y(x_3\ x_1) - y(x_1\ x_2)}{x_3 - x_2}$$

i.e. $$y(10,9,4) = \frac{y(10,9) - y(9,4)}{10-4}$$

$$= \frac{269-131}{10-4}$$

$$= 23$$

Third order divided differences :

The third order divided difference over $(x_0\ x_1\ x_2\ x_3)$ is given as,

$$y(x_3\ x_2\ x_1\ x_0) = \frac{y(x_3\ x_1\ x_2) - y(x_2\ x_1\ x_0)}{x_3 - x_0}$$

i.e. $$y(10,9,4,2) = \frac{y(10,9,4) - y(9,4,2)}{10-2}$$

$$= \frac{23-15}{10-2}$$

$$= 1$$

Table 5.3.8 shows the divided difference table with those calculated values.

Table 5.3.8 : Divided difference table for example 5.3.3

x	y	y (x_n x_{n-1})	y (x_n x_{n-1} x_{n-2})	y (x_n x_{n-1} x_{n-2} x_{n-3})
$x_0 = 2$	$y_0 = 4$			
		$y(x_1\ x_0) = 26$		
$x_1 = 4$	$y_1 = 56$		$y(x_2\ x_1\ x_0) = 15$	
		$y(x_2\ x_1) = 131$		$y(x_3\ x_2\ x_1\ x_0) = 1$
$x_2 = 9$	$y_2 = 711$		$y(x_3\ x_2\ x_1) = 23$	
		$y(x_3\ x_2) = 269$		
$x_3 = 10$	$y_3 = 980$			

5.3.3.1 Algorithm and Flowchart to Generate Divided Differences

Based on the discussion of divided differences we will now prepare an algorithm for it.

Algorithm :

Step 1 : Read the number of entries (data points) of which divided difference is to be calculated.

Step 2 : Read values of x & y.

Step 3 : Calculate the divided differences using,

$$y(x_n\ x_{n-1}\ x_{n-2}\ \ldots x_0) = \frac{y(x_n\ x_{n-1}\ \ldots x_n) - y(x_{n-1}\ x_{n-2}\ \ldots x_0)}{x_n - x_0}$$

Step 4 : Display the divided difference table

Step 5 : Stop

Flowchart to generate divided differences :

First we will discuss the logic to generate a divided difference table.

We will use one dimensional array x [ ] to store $x_0, x_1, x_2, x_3, \ldots x_n$ (i.e. values of x). A two dimensional array y [ ] [ ] is used to store values of y and divided differences. Table 5.3.9 shows how these differences are assigned to array elements. 'i' represents rows of the table and 'j' represents columns of the table. The values of 'y' are stored in first column (j = 0) of the two dimensional array y [ ] [ ].

Table 5.3.9 (See on next page)

In the second column of the array (i.e. j = 1) first order divided differences are stored.

i.e. $$y[0][1] = y(x_1\ x_0) = \frac{y_1 - y_0}{x_1 - x_0} = \frac{y[1][0] - y[0][0]}{x[1] - x[0]}$$

Similarly, $$y[1][1] = y(x_2\ x_1)$$

Table 5.3.9 Two dimensional array representation of a divided difference table y[4] [4] = Two dimensional array

y = [4] [4] = Two Dimensional Array

One Dimensional Array x[4]				
i = 0 x[0] = x_0	$i = 0, j = 0$ $y[0][0] = y_0$	$i = 0, j = 1$ $y[0][1] = y(x_1 x_0)$ $= \frac{y_1 - y_0}{x_1 - x_0}$ $= \frac{y[1][0] - y[0][0]}{x[1] - x[0]}$	$i = 0, j = 2$ $y[0][2] = y(x_2 x_1 x_0)$ $= \frac{y(x_2 x_1) - y(x_1 x_0)}{x_2 - x_0}$ $= \frac{y[1][1] - y[0][1]}{x[2] - x[0]}$	$i = 0, j = 3$ $y[0][3] = y(x_3 x_2 x_1 x_0)$ $= \frac{y(x_3 x_2 x_1) - y(x_2 x_1 x_0)}{x_3 - x_0}$ $= \frac{y[1][2] - y[0][2]}{x[3] - x[0]}$
i = 1 x[1] = x_1	$i = 1, j = 0$ $y[1][0] = y_1$	$i = 1, j = 1$ $y[1][1] = y(x_2 x_1)$ $= \frac{y_2 - y_1}{x_2 - x_1}$ $= \frac{y[2][0] - y[1][0]}{x[2] - x[1]}$	$i = 1, j = 2$ $y[1][2] = y(x_3 x_2 x_1)$ $= \frac{y(x_3 x_2) - y(x_2 x_1)}{x_3 - x_1}$ $= \frac{y[2][1] - y[1][1]}{x[3] - x[1]}$	$i = 1, j = 3$ No difference
i = 2 x[2] = x_2	$i = 2, j = 0$ $y[2][0] = y_2$	$i = 2, j = 1$ $y[2][1] = y(x_3 x_2)$ $= \frac{y_3 - y_2}{x_3 - x_2}$ $= \frac{y[3][0] - y[2][0]}{x[3] - x[2]}$	$i = 2, j = 2$ No difference	$i = 2, j = 3$ No difference
i = 3 x[3] = x_3	$i = 3, j = 0$ $y[3][0] = y_3$	$i = 3, j = 1$ No difference is possible here	$i = 3, j = 2$ No difference	$i = 3, j = 3$ No difference

and $\qquad y[2][1] = y(x_3\ x_2)$

The last place in this column, i.e. $y[3][1]$ will be empty since no divided difference is calculated there. This is because, we know that number of differences goes on reducing as order of difference increases. In the third column (j = 2) second order divided differences are stored. Observe that two divided differences are stored at $y[0][2]$ and $y[1][2]$ and remaining two places are empty.

If there are more data points, then array size will be increased accordingly.

For example if there are 10 data points, then x [10] and y [10] [10].

This logic is used to prepare a flowchart and 'C' program in next subsection. A simplified flowchart is shown in Fig. 5.3.3.

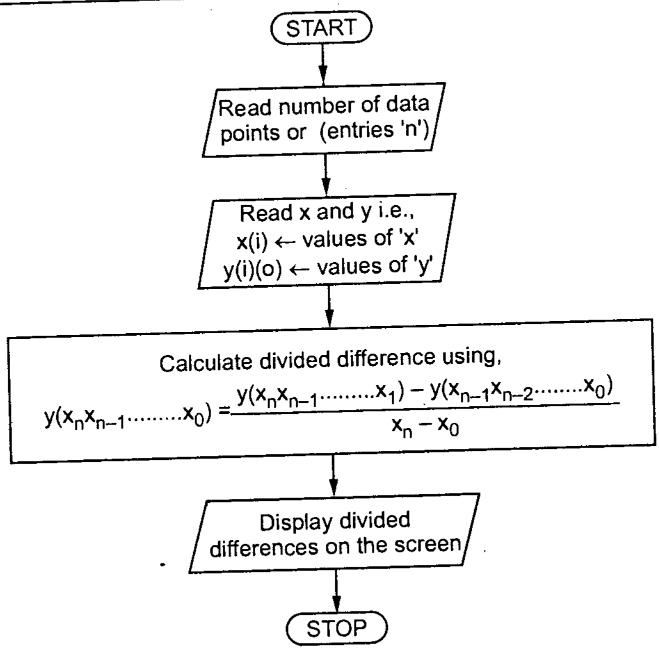


Fig. 5.3.3 Simplified flowchart of divided differences table

5.3.3.2 C Program and Logic Development

In the last subsection, we discussed a logic to prepare a divided difference table. A 'C' program based on this logic is shown below.

```
/* Download this program from  www.vtubooks.com                        */
/* File name : div_diff.cpp                                             */

/*-------------- DIVIDED DIFFERENCES GENERATION FOR INTERPOLATION -----*/

/*        THE PROGRAM GENERATES A DIVIDED DIFFERENCES TABLE FROM GIVEN

          DATA.  THE VALUES OF x AND CORRESPONDING y = f(x) ARE TO BE

          ENTERED IN THE ARRAY FORMAT.

          INPUTS :  1) Number of entries of the data.
                    2) Values of 'x' & corresponding y = f(x).
```

```
        OUTPUTS : divided difference table array.                    */

/*----------------------------- PROGRAM --------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>

void main()
{
      double y[20][20],x[20];
               /* ARRAY OF y[n][n] ELEMENTS FOR DIVIDED DIFFERENCE TABLE */
      int i,j,k,n;

      clrscr();
      printf("\n      DIVIDED DIFFERENCES GENERATION FOR INTERPOLATION");

      printf("\n\n  The form of equation is y = f(x)\n");

      printf("\n\nEnter the number of entries (max 20) = ");
                              /* ENTER THE NUMBER OF ENTRIES IN THE TABLE   */
      scanf("%d",&n);

      for(i = 0; i < n; i++)
      {                         /*  LOOP TO GET x AND y = f(x) IN THE TABLE   */

          printf("x%d = ",i);  scanf("%lf",&x[i]);
          printf("           y%d = ",i);  scanf("%lf",&y[i][0]);
      }
      k = n;
      for(j = 1; j < n; j++)
      {          /*  LOOP TO CALCULATE DIVIDED DIFFERENCES IN THE TABLE  */
          k = k - 1;
          for(i = 0; i < k; i++)
          {
             y[i][j] = (y[i+1][j-1] - y[i][j-1])/(x[i+j]-x[i]);
          }
      }

      k = n;    clrscr();
      printf("\n The divided difference table is as follows . . .\n");
      printf("\n\tx\ty\tDy\tD2y\tD3y\tD4y\tD5y\tD6y\n");
      for(i = 0; i < n; i++)
      {        /*  LOOP TO PRINT DIVIDED DIFFERENCES IN THE TABLE  */

        printf("\nx%d = %4.2lf",i,x[i]);

        for(j = 0; j < k-i; j++)
        {
          printf("\t%4.2lf ",y[i][j]);
        }
        printf("\n");
      }
}
/*------------------------------End of program ------------------------*/
```

The first statement in function `main` is

`double y[20][20], x [20];`

This defines a two dimensional array 'y' of $20 * 20$ elements and one dimensional array 'x' of 20 elements. If you want to find divided difference of more than 20 elements, then you will have to increase the size of arrays. The fourth `printf` and `scanf` statements combinely ask for number of entries or data points of which divided difference is to be found. Next there is a `for` loop. This loop gets values of x & y in the arrays. Values of y are stored in first (j = 0) column.

Next there are two `for` loops to calculate divided differences. These loops are reproduced below for convenience.

```
for  (j=1;  j  <  n;  j++)   ← This loop is used for columns.
{
     k  =  k-1;      ←   This counter takes care of number of divided
          differences to be calculated in each column.
     for  (i=0;  i<k;  i++)  ← This loop is used for rows.
          It repeats `k' times to calculate `k' differences.
     {
          y[i][j]=(y[i+1][j-1]-y[i][j-1])/(x[i+j]-x[i]);
     }              This statement calculates divided difference.
}
```

The function of each statement in the program is indicated above.

Next two `for` loops print the divided difference table. The outer `for` loop is used for printing rows and internal `for` loop is used for printing columns. Here the display of divided differences table looks like that of forward difference.

How to Run this program?

Compile the 'C' program given here and make its EXE file. Here we will use the data of example 5.3.3 to obtain the results from computer program. The data points given in example 5.3.3 are reproduced here for convenience.

x	$x_0 = 2$	$x_1 = 4$	$x_2 = 9$	$x_3 = 10$
y	$y_0 = 4$	$y_1 = 56$	$y_2 = 711$	$y_3 = 980$

Run the EXE file on computer. The program first displays names of the method. Then it displays,

`Enter the number of entries (max 20) =` Here enter the number of data points, i.e. 4 & press 'enter' key.

Then the program displays,

x0 = Here enter `2' and press `enter' key.

y0 = Here enter `4' and press `enter' key

x1 = enter `4' & ↵

y1 = enter `56' & ↵

x2 = enter `9' & ↵

y2 = enter `711' & ↵

x3 = enter `10' & ↵

y3 = enter `980' & ↵

Here symbol '↵' means press ' enter ' key of computer. The program then displays the divided difference table. Here is the display of all the results combinely,

```
---------------------------------- Results -------------------------

      DIVIDED DIFFERENCES GENERATION FOR INTERPOLATION

   The form of equation is y = f(x)

 Enter the number of entries (max 20) = 4
 x0 = 2
            y0 = 4
 x1 = 4
            y1 = 56
 x2 = 9
            y2 = 711
 x3 = 10
            y3 = 980

 The divided difference table is as follows . . .

        x        y       Dy       D2y      D3y      D4y      D5y      D6y

 x0 = 2.00       4.00    26.00    15.00    1.00

 x1 = 4.00       56.00   131.00   23.00

 x2 = 9.00       711.00  269.00

 x3 = 10.00      980.00
--------------------------------------------------------------------
```

The values under third column in the display of table (i.e. Dy) are first order divided differences. The first entry in this column is '26'.

$$\therefore \quad Dy1 = y(x_1\ x_0) = 26$$

Similarly

$$Dy2 = y(x_2\ x_1) = 131$$

$$Dy3 = y(x_3\ x_2) = 269$$

The entries in next column (D2y) are second order differences, i.e.,

$$D2y1 = y(x_2\ x_1\ x_0) = 15$$

$$D2y2 = y(x_3\ x_2\ x_1) = 23$$

There is only one entry under D3y and it is,

$$y(x_3\ x_2\ x_1\ x_0) = 1$$

Observe that the table displayed on the screen is similar to that we obtained in example 5.3.3, i.e. Table 5.3.8.

5.3.3.3 Relation Between Divided Differences and Forward Differences

We know from equation 5.3.16 that,

$$y(x_1\ x_0) = \frac{y_1 - y_0}{x_1 - x_0}$$

From equation 5.3.6 we know that,

$$\Delta y_0 = y_1 - y_0$$

Putting the value of $y_1 - y_0$ in the equation above,

$$y(x_1\ x_0) = \frac{\Delta y_0}{x_1 - x_0}$$

It $x_0, x_1, \ldots x_n$ are equally spaced, then $x_1 - x_0 = h$. Therefore above equation becomes,

$$y(x_1\ x_0) = \frac{\Delta y_0}{h} \quad \ldots (5.3.29)$$

Similarly, $$y(x_2\ x_1) = \frac{y_2 - y_1}{x_2 - x_1}$$

$$= \frac{\Delta y_1}{h} \quad \ldots (5.3.30)$$

From equation 5.3.21 we know that

$$y(x_2\ x_1\ x_0) = \frac{y(x_2\ x_1) - y(x_1\ x_0)}{x_2 - x_0}$$

Putting values of $y(x_2\ x_1)$ and $y(x_1\ x_0)$ from equation 5.3.29 and equation 5.3.30 in the above equation,

$$y(x_2\ x_1\ x_0) = \frac{\dfrac{\Delta y_1}{h} - \dfrac{\Delta y_0}{h}}{2h} \qquad \because\ x_2 - x_0 = 2h$$

$$= \frac{\Delta y_1 - \Delta y_0}{2h^2}$$

$\because \quad \Delta y_1 - \Delta y_0 = \Delta^2 y_0$, The above equation will be,

$$y(x_2\ x_1\ x_0) = \frac{\Delta^2 y_0}{2h^2} \quad \ldots (5.3.31)$$

Similarly, $y(x_3\ x_2\ x_1) = \dfrac{\Delta^2 y_1}{2h^2}$ can be easily obtained. ... (5.3.32)

Now $$y(x_3\ x_2\ x_1\ x_0) = \frac{y(x_3\ x_2\ x_1) - y(x_2\ x_1\ x_0)}{x_3 - x_0}$$

Putting the values of $y(x_3\ x_2\ x_1)$ and $y(x_2\ x_1\ x_0)$ from equation 5.3.31 and equation 5.3.32 we obtain,

$$y(x_3\ x_2\ x_1\ x_0) = \frac{\dfrac{\Delta^2 y_1}{2h^2} - \dfrac{\Delta^2 y_0}{3h^2}}{3h} \qquad \because\ x_2 - x_0 = 2h$$

$$= \frac{\Delta^2 y_1 - \Delta^2 y_0}{6h^3}$$

Since $\Delta^2 y_1 - \Delta^2 y_0 = \Delta^3 y_0$ from forward differences, the above equation will be,

$$y(x_3\ x_2\ x_1\ x_0) = \frac{\Delta^3 y_0}{6h^3}$$

or $$\boxed{y(x_3\ x_2\ x_1\ x_0) = \frac{\Delta^3 y_0}{3!\,h^3}} \quad \text{since } 3! = 6 \qquad \text{... (5.3.33)}$$

This equation can be extended to 'n' values similarly. i.e.

$$y(x_n\ x_{n-1}\ x_{n-2} \cdots x_0) = \frac{\Delta^n y_0}{n!\,h^n} \qquad \text{... (5.3.34)}$$

This is the relation between forward and divided differences.

5.3.4 Central Differences

We have seen forward differences in section 5.3.1 Table 5.3.1 lists those forward differences. This table is reproduced here for convenience.

Table 5.3.10 Forward difference table

Sr.No.	x	y	Δ y	Δ^2 y	Δ^3 y	Δ^4 y
1	x_0	y_0				
			Δy_0			
2	x_1	y_1		$\Delta^2 y_0$		
			Δy_1		$\Delta^3 y_0$	
3	x_2	y_2		$\Delta^2 y_1$		$\Delta^4 y_0$
			Δy_2		$\Delta^3 y_1$	
4	x_3	y_3		$\Delta^2 y_2$		
			Δy_3			
5	x_4	y_4				

In this table we have called out starting data point as (x_0, y_0). If we call the same points as (x_{-2}, y_{-2}) and change the next data points accordingly then the forward differences table will look like as shown in Table 5.3.11 below.

This type of (x_0, y_0) shift we have done to introduce central difference.

From equation 5.2.16 we know that,

$$\delta = E^{1/2} - E^{-1/2} \qquad \text{... (5.3.35)}$$

Multiply both sides of this equation by $E^{1/2}$

$$\therefore \quad \delta E^{1/2} = E - 1$$

Table 5.3.11 : Forward difference table with (x_0, y_0) shifted

Sr. No.	x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
1	x_{-2}	y_{-2}				
			$\Delta y_{-2} = y_{-1} - y_{-2}$			
2	x_{-1}	y_{-1}		$\Delta^2 y_{-2} = \Delta y_{-1} - \Delta y_{-2}$		
			$\Delta y_{-1} = y_0 - y_{-1}$		$\Delta^3 y_{-2} = \Delta^2 y_{-1} - \Delta^2 y_{-2}$	
3	x_0	y_0		$\Delta^2 y_{-1} = \Delta y_0 - \Delta y_{-1}$		$\Delta^4 y_{-2} = \Delta^3 y_{-1} - \Delta^3 y_{-2}$
			$\Delta y_0 = y_1 - y_0$		$\Delta^3 y_{-1} = \Delta^2 y_0 - \Delta^2 y_{-1}$	
4	x_1	y_1		$\Delta^2 y_0 = \Delta y_1 - \Delta y_0$		
			$\Delta y_1 = y_2 - y_1$			
5	x_2	y_2				

Thus $$\boxed{\delta E^{1/2} = \Delta} \quad \text{... (5.3.36)}$$

Now let's operate above equation on y_{-2}

i.e. $$\delta E^{1/2} y_{-2} = \Delta y_{-2} \quad \text{... (5.3.37)}$$

From equation 5.2.13 we know that,

$$E^n f(x) = f(x + nh)$$

Replace x by x_r in the above equation,

$$E^n f(x_r) = f(x_r + nh) \quad \text{... (5.3.38)}$$

We know that $$y = f(x)$$

$$\therefore \quad y_r = f(x_r)$$

With the above shorthand notation equation 5.3.38 becomes,

$$E^n y_r = y_{r+nh} \quad \text{... (5.3.39)}$$

Let h = 1 then above equation becomes,

$$\boxed{E^n y_r = y_{r+n}} \quad \text{... (5.3.40)}$$

We can apply this equation to LHS of equation 5.3.37 with

$$n = \frac{1}{2} \quad \& \quad r = -2$$

i.e. $$\delta E^{1/2} y_{-2} = \Delta y_{-2}$$

$$= \delta y_{-2+\frac{1}{2}} \qquad \text{(From equation 5.3.40)}$$

$$= \delta y_{-3/2}$$

or $$\boxed{\Delta y_{-2} = \delta y_{-3/2}} \quad \text{... (5.3.41)}$$

From equation 5.3.36 we know that,

$$\Delta = \delta E^{1/2} \quad \text{... (5.3.42)}$$

Now operate this equation on y_{-1}

$$\therefore \quad \Delta y_{-1} = \delta E^{1/2} y_{-1}$$

$$= \delta y_{-1+\frac{1}{2}} \quad \text{(From equation 5.3.40 with } r = -1 \text{ \& } n = \frac{1}{2}\text{)}$$

i.e. $$\boxed{\Delta y_{-1} = \delta y_{-1/2}} \quad \text{... (5.3.43)}$$

Let's operate equation 5.3.42 on y_0. i.e.,

$$\Delta y_0 = \delta E^{1/2} y_0$$

$$= \delta y_{0+\frac{1}{2}} \quad \text{(From equation 5.3.40 with } r = 0 \text{ \& } n = \frac{1}{2}\text{)}$$

i.e. $$\boxed{\Delta y_0 = \delta y_{1/2}} \quad \text{... (5.3.44)}$$

Thus from equation 5.3.41, equation 5.3.43 and equation 5.3.44 we have

$$\left.\begin{aligned} \Delta y_{-2} &= \delta y_{-3/2} \\ \Delta y_{-1} &= \delta y_{-1/2} \\ \Delta y_0 &= \delta y_{1/2} \end{aligned}\right\} \quad \text{... (5.3.45)}$$

Similarly we can obtain,

$$\Delta y_1 = \delta y_{3/2}$$

Let's recall equation 5.3.35 again i.e.,

$$\delta = E^{1/2} - E^{-1/2}$$

$$\delta^2 = \delta \times \delta = \left(E^{1/2} - E^{-1/2}\right)\left(E^{1/2} - E^{-1/2}\right)$$

$$= E - 2 + E^{-1}$$

Multiply both sides of above equation by E.

i.e. $$\delta^2 E = \left(E - 2 + E^{-1}\right)E = E^2 - 2E + 1$$

or $$\delta^2 E = (E-1)^2 \qquad \because (E-1)^2 = E^2 - 2E + 1$$

$$\therefore \quad \delta^2 E = \Delta^2 \qquad \text{(Since } E - 1 = \Delta \text{ from equation 5.2.6)}$$

or $$\boxed{\Delta^2 = \delta^2 E} \quad \text{... (5.3.46)}$$

Let's operate this equation on y_{-2}

$$\therefore \quad \Delta^2 y_{-2} = \delta^2 E y_{-2}$$

$$= \delta^2 y_{-2+1} \quad \text{(From equation 5.3.40 with n = 1 \& r = –2)}$$

i.e. $$\boxed{\Delta^2 y_{-2} = \delta^2 y_{-1}} \qquad \text{... (5.3.47)}$$

Let's operate equation 5.3.36 on y_{-1}

i.e. $$\Delta^2 y_{-1} = \delta^2 E y_{-1}$$

$$= \delta^2 y_{-1+1} \quad \text{(From equation 5.3.40 with n = 1 \& r = –1)}$$

i.e. $$\boxed{\Delta^2 y_{-1} = \delta^2 y_0} \qquad \text{... (5.3.48)}$$

Thus we have,

$$\left.\begin{array}{l} \Delta^2 y_{-2} = \delta^2 y_{-1} \\ \Delta^2 y_{-1} = \delta^2 y_0 \end{array}\right\} \qquad \text{... (5.3.49)}$$

Similarly we can obtain,

$$\Delta^2 y_0 = \delta^2 y_1$$

We know equation 5.3.42 that

$$\Delta = \delta E^{1/2}$$

& from equation 5.3.46 we know that

$$\Delta^2 = \delta^2 E$$

Multiply the above two equations, we get,

$$\Delta \times \Delta^2 = \delta E^{1/2} \times \delta^2 E$$

$$\therefore \quad \boxed{\Delta^3 = \delta^3 E^{3/2}} \qquad \text{... (5.3.50)}$$

Let's operate this equation on y_{-2},

$$\Delta^3 y_{-2} = \delta^3 E^{3/2} y_{-2}$$

$$= \delta^3 y_{-2+\frac{3}{2}} \quad \text{(From equation 5.3.40 with } n = \frac{3}{2} \text{ \& } r = -2)$$

i.e. $$\boxed{\Delta^3 y_{-2} = \delta^3 y_{-\frac{1}{2}}} \qquad \text{... (5.3.51)}$$

Let's operate equation 5.3.50 on y_{-1}

$$\Delta^3 y_{-1} = \delta^3 E^{3/2} y_{-1}$$

$$= \delta^3 y_{-1+\frac{3}{2}} \quad \text{(From equation 5.3.40 with } n = \frac{3}{2} \text{ \& } r = -1)$$

i.e. $$\boxed{\Delta^3 y_{-1} = \delta^3 y_{1/2}} \qquad \text{... (5.3.52)}$$

Thus we have

$$\left.\begin{aligned} \Delta^3 y_{-2} &= \delta^3 y_{-\frac{1}{2}} \\ \Delta^3 y_{-1} &= \delta^3 y_{\frac{1}{2}} \end{aligned}\right\} \quad \text{... (5.3.53)}$$

Similarly we can obtain,

$$\Delta^4 y_{-2} = \delta^4 y_0$$

Using equation 5.3.45, equation 5.3.49 and equation 5.3.53, we can transform forward differences of table 5.3.11 as follows

Table 5.3.12 : Central differences table (See table 5.3.11 also)

x	y	$\Delta y = \delta y$	$\Delta^2 y = \delta^2 y$	$\Delta^3 y = \delta^3 y$	$\Delta^4 y = \delta^4 y$
x_{-2}	y_{-2}				
		$\Delta y_{-2} = \delta y_{-3/2}$			
x_{-1}	y_{-1}		$\Delta^2 y_{-2} = \delta^2 y_{-1}$		
		$\Delta y_{-1} = \delta y_{-1/2}$		$\Delta^3 y_{-2} = \delta^3 y_{-1/2}$	
x_0	y_0		$\Delta^2 y_{-1} = \delta^2 y_0$		$\Delta^4 y_{-2} = \delta^4 y_0$
		$\Delta y_0 = \delta y_{1/2}$		$\Delta^3 y_{-1} = \delta^3 y_{1/2}$	
x_1	y_1		$\Delta^2 y_0 = \delta^2 y_1$		
		$\Delta y_1 = \delta y_{1/2}$			
x_2	y_2				

As shown in Table 5.3.12,

$$\delta y_{-3/2}, \delta y_{-1/2} \ldots \delta^2 y_{-1}, \delta^2 y_0, \delta^3 y_{-1/2}, \delta^3 y_{1/2}, \delta^4 y_0 \text{ .... etc.}$$

are called 'central differences' and Table 5.3.12 is called central difference table.

Ex.5.3.4 *Generate a central difference table for the following data –*

x	$x_{-2} = 2$	$x_{-1} = 6$	$x_0 = 10$	$x_1 = 14$	$x_2 = 18$
y	$y_{-2} = 21.857$	$y_{-1} = 21.025$	$y_0 = 20.132$	$y_1 = 19.145$	$y_2 = 18.057$

Sol. : From Table 5.3.12 we know that,

$$\Delta y_{-2} = \delta y_{-3/2} = y_{-1} - y_{-2}$$

$$\therefore \quad \delta y_{-3/2} = 21.025 - 21.857 = -0.832$$

$$\Delta y_{-1} = \delta y_{-1/2} = y_0 - y_{-1}$$

$$\therefore \quad \delta\, y_{-1/2} = 20.132 - 21.025 = -\ 0.893$$

$$\Delta\, y_0 = \delta\, y_{1/2} = y_1 - y_0$$

$$\therefore \quad \delta^2\, y_{1/2} = 19.145 - 20.132 = -\ 0.987$$

Similarly $\Delta\, y_1 = \delta\, y_{3/2} = y_2 - y_1 = 18.057 - 19.145 = -\ 1.088$

Important Note : Basically central difference table shown above is a forward difference table. Only different notations are used for forward differences and those notations are central differences.

On the same lines we have calculated remaining values in Table 5.3.13.

Table 5.3.13 : Central differences table of example 5.3.4

x	y	$\delta y = \Delta y$	$\delta^2 y = \Delta^2 y$	$\delta^3 y = \Delta^3 y$	$\delta^4 y = \Delta^4 y$
$x_{-2} = 2$	$y_{-2} = 21.857$				
		$\delta\, y_{-3/2} = \Delta\, y_{-2}$ $= y_{-1} - y_{-2}$ $= -\ 0.832$			
$x_{-1} = 6$	$y_{-1} = 21.025$		$\delta^2\, y_{-1} = \Delta^2\, y_{-2}$ $= \Delta\, y_{-1} - \Delta\, y_{-2}$ $= -\ 0.061$		
		$\delta\, y_{-1/2} = \Delta\, y_{-1}$ $= y_0 - y_{-1}$ $= -\ 0.893$		$\delta^3\, y_{-1/2} = \Delta^3\, y_{-2}$ $= \Delta^2\, y_{-1} - \Delta^2\, y_{-2}$ $= -\ 0.033$	
$x_0 = 10$	$y_0 = 20.132$		$\delta^2\, y_0 = \Delta^2\, y_{-1}$ $= \Delta\, y_0 - \Delta\, y_{-1}$ $= -\ 0.094$		$\delta^4\, y_0 = \Delta^4\, y_{-2}$ $= \Delta^3\, y_{-1} - \Delta^3\, y_{-2}$ $= -\ 0.026$
		$\delta\, y_{1/2} = \Delta\, y_0$ $= y_{-1} - y_0$ $= -\ 0.987$		$\delta^3\, y_{1/2} = \Delta^3\, y_{-1}$ $= \Delta^2\, y_0 - \Delta^2\, y_{-1}$ $= -\ 0.007$	
$x_1 = 14$	$y_1 = 19.145$		$\delta^2\, y_1 = \Delta^2\, y_0$ $= \Delta\, y_1 - \Delta\, y_0$ $= -\ 0.101$		
		$\delta\, y_{3/2} = \Delta\, y_1$ $= y_2 - y_1$ $= -\ 1.088$			
$x_2 = 18$	$y_2 = 18.057$				

5.3.4.1 Important Points About Central Differences

1) The first order central differences are,

$$\ldots\ldots \delta y_{-5/2}, \delta y_{-3/2}, \delta y_{-1/2}, \delta y_{1/2}, \delta y_{3/2}, \delta y_{5/2}, \ldots\ldots$$

their equivalent forward differences are

$$\ldots\ldots \Delta y_{-3}, \Delta y_{-2}, \Delta y_{-1}, \Delta y_0, \Delta y_1, \Delta y_2 \ldots\ldots$$

> *Thus central differences are basically forward differences with different notations.*

2) These differences are called central differences because they use central difference operator `δ' for their representation.

3) Every central difference table is a forward difference table compulsory.

4) Shifting of (x_0, y_0) data point in the table is based on the method of interpolation used. Therefore the forward/central difference table is modified accordingly.

5) Central differences are generated only for convenient interpolation method.

5.3.5 Detection of Errors by Difference Tables

The difference tables can be used to detect errors in the given data. Let us consider the following data :

$y_0 = 0,\ y_1 = 0,\ y_2 = 0,\ y_3 = \varepsilon,\ y_4 = 0,\ y_5 = 0$ and $y_6 = 0$

Here 'ε' is an error. Thus error is present in y_3. In rest of the values, there is no error. Now let us prepare a forward differences table of this data. Table 5.3.14 shows this table.

Table 5.3.14 : Difference table for one term in error

y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$	$\Delta^6 y$
$y_0 = 0$						
	0					
$y_1 = 0$		0				
	0		ε			
$y_2 = 0$		ε		-4ε		
	ε		-3ε		10ε	
$y_3 = \varepsilon$		-2ε		6ε		-20ε
	$-\varepsilon$		3ε		-10ε	
$y_4 = 0$		ε		-4ε		
	0		$-\varepsilon$			
$y_5 = 0$		0				
	0					
$y_6 = 0$						

Following points can be noted from the Table 5.3.14 :

(i) The effect of error goes on increasing with the order of differences. Observe that error in Δy is ε and $-\varepsilon$. Whereas error in $\Delta^2 y$ is ε, -2ε and ε.

(ii) Δy has errors of ε and $-\varepsilon$. Similarly $\Delta^2 y$ has errors of $\varepsilon, -2\varepsilon$ and ε. And $\Delta^3 y$ has errors of $\varepsilon, -3\varepsilon, 3\varepsilon$ and $-\varepsilon$. This shows that errors in any column have binomial coefficients. These coefficients have alternating sign.

(iii) The algebraic sum of errors in any difference column is zero. For example sum of errors in $\Delta^2 y$ is : $\varepsilon - 2\varepsilon + \varepsilon = 0$.

(iv) As indicated in the table, the maximum error occurs directly opposite the entry which is in error. For example, maximum value of $\Delta^2 y$ is -2ε. This is opposite to $y_3 = \varepsilon$, which is in error. In $\Delta^3 y$, such entries are -3ε and 3ε. These entries are shown by lines in table 5.3.14. The error can then be easily calculated from the difference table.

Ex. 5.3.5 *Locate and correct the error in the following table of values.*

[May - 97, 8 marks, Dec - 99, 9 marks]

x	2.5	3.0	3.5	4.0	4.5	5.0	5.5
y	4.32	4.83	5.27	5.47	6.26	6.79	7.23

Sol. : Let us prepare the forward difference table for above data.

Table 5.3.15 : Difference table for example 5.3.5

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$	$\Delta^6 y$
$x_0 = 2.5$	$y_0 = 4.32$						
		0.51					
$x_1 = 3.0$	$y_1 = 4.83$		−0.07				
		0.44		−0.17			
$x_2 = 3.5$	$y_2 = 5.27$		−0.24		1.0		
		0.2		0.83			
$x_3 = 4.0$	$y_3 = 5.47$		0.59		−1.68	Maximum error opposite to y_3	
		0.79		−0.85			
$x_4 = 4.5$	$y_4 = 6.26$		−0.26		1.02		
		0.53		0.17			
$x_5 = 5.0$	$y_5 = 6.79$		−0.09				
		0.44					
$x_6 = 5.5$	$y_6 = 7.23$			Alternating signs of differences			

In the Table 5.3.15, observe that the third order differences have alternating signs. Similarly 4^{th} order differences also have alternating signs. In the 4^{th} order differences -1.68 is the maximum value. It is opposite to y_3. This is indicated by dotted lines in the table. Hence there is an error in y_3. Now compare the 4^{th} order $(\Delta^4 y)$ differences of table 5.3.14 and above table. We get,

$$6\varepsilon = -1.68$$

$$\therefore \quad \varepsilon = -0.28$$

Thus the error in $y_3 = 5.47$ is -0.28.

Hence the true value is,

$$\text{True value of} \quad y_3 = y_3 - \varepsilon$$
$$= 5.47 - (-0.28)$$
$$= 5.75$$

Exercise

1. A polynomial $f(x)$ of lowest degree is tabulated below :

x	−3	−2	−1	0	1	2	3	4	5	6	7
$f(x)$	−30	−12	0	3	4	5	10	19	38	69	125

Locate and correct errors in values of $f(x)$.

[May - 2001, 8 marks, Dec - 98, 8 marks, Dec - 96, 8 marks]

Hint : From the above data, the 5^{th} order difference $\Delta^5 y$ will be

$-15, \ 7, \ -6, \ 10, \ -10, \ 15$

Observe that values alternate in sign. First and last differences have maximum amplitudes. Hence there are multiple errors in the given data. These errors can be located by splitting data and repeating the procedure. Let us select following part of the data.

x	−1	0	1	2	3	4	5	6
$f(x)$	0	3	4	5	10	19	38	69

The difference table for this data is as follows :

Please refer Table on next page.

In the above table, observe that $\Delta^4 y$ has alternating signs and its maximum value (i.e. 6) is opposite to $x=3$, $y=10$. Hence $y=10$ contains error. From table 5.3.14 we know that maximum value of $\Delta^4 y$ is 6ε.

Hence $6\varepsilon = 6$, or $\varepsilon = 1$.

Ans. : At $x=3$, $y=10-1=9$. Similarly other errors can be located.

x	y = f(x)	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$
− 1	0					
		3				
0	3		− 2			
		1		2		
1	4		0		2	
		1		4		− 6
2	5		4		− 4	
		5		0		10
3 ←	10		4		6	
		9		6		− 10
4	19		10		− 4	
		19		2		
5	38		12			
		31				
6	69					

5.4 Interpolation Techniques Based on Finite Differences

We have seen finite differences in last section in detail. There are some interpolation techniques which use those finite differences. These techniques are

i) Newton's forward differences interpolation method.

ii) Newton's backward differences interpolation method.

iii) Stirling's interpolation method based on central differences.

Let's discuss those techniques in detail.

5.4.1 Newton's Forward Differences Interpolation Method

(or Gregory–Newton's forward interpolation method)

For the evenly spaced values of x we know that,

$$x_1 = x_0 + h$$
$$x_2 = x_2 + h = x_0 + 2h$$
$$x_3 = x_2 + h = x_0 + 3h$$
$$x_4 = x_4 + h = x_0 + 4h \quad \text{\& so on.}$$

Let's say that we are required to find a value of y at $x_r = x_0 + rh$.

i.e. $$y_r = f(x_0 + rh) \qquad (\text{Here } y_r = f(x_r) \,\&\, x_r = x_0 + rh) \qquad \ldots (5.4.1)$$

The value of y_r is not given, and we are required to calculate it.

From equation 5.2.13, we know that

$$E^n f(x) = f(x+nh)$$

Replace 'n' by 'r' in above equation. Then we have,

$$f(x+rh) = E^r f(x) \qquad \text{... (5.4.2)}$$

Applying this equation to equation 5.4.1, we have,

$$y_r = E^r f(x_0)$$

From equation 5.2.7 we know that $E=1+\Delta$ and $f(x_0)=y_0$, then the above equation becomes,

$$y_r = (1+\Delta)^r y_0 \qquad \text{... (5.4.3)}$$

Lets expand $(1+\Delta)^r$ using binomial series,

$$y_r = \left[1+r\,\Delta+\frac{r(r-1)}{2!}\Delta^2+\frac{r(r-1)(r-2)}{3!}\Delta^3+\dots\right]y_0$$

or

$$y_r = y_0+r\,\Delta y_0+\frac{r(r-1)}{2!}\Delta^2 y_0+\frac{r(r-1)(r-2)}{3!}\Delta^3 y_0+\dots \qquad \text{... (5.4.4)}$$

This is the formula for Newton's forward differences interpolation. This interpolation is used when x_r is close to first value x_0 in the given data.

5.4.1.1 Solved Examples

Ex. 5.4.1 *Construct the difference table from the following data.*

x	50	51	52	53	54
y = f (x)	39.1961	39.7981	40.3942	40.9843	41.5687

Obtain f (50.5) using Newton's forward difference formula.

Sol. : The forward difference table is shown below for the given data.

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
50	39.1961				
		0.602			
51	39.7981		− 0.0059		
		0.5961		− 0.0001	
52	40.3942		−0.0060		0.0004
		0.5901		0.0003	
53	40.9843		− 0.0057		
		0.5844			
54	41.5687				

We have $x_r = x_0 + rh$ (From equation 5.4.1)

$$\therefore \quad r = \frac{x_r - x_0}{h} \quad \ldots (5.4.5)$$

We have to find f (50.5)

i.e. $y_r = f(x_r)$

$= f(50.5) = ?$

Therefore $x_r = 50.5$

h = Spacing between values of x

$= 51 - 50$

$= 52 - 51$

$= \ldots.\ 54 - 53$

$= 1$

We have $x_0 = 50$, putting values of x_0, h and x_r in equation 5.4.5, we get value of 'r' as,

$$r = \frac{50.5 - 50}{1}$$

$\therefore \quad r = 0.5$

Also we have $\Delta y_0 = 0.602$, $\Delta^2 y_0 = -0.0059$, $\Delta^3 y_0 = -0.0001$ and

$\Delta^4 y_0 = 0.0004$ from forward difference table.

Putting all these values in Newton's forward interpolation formula given by equation 5.4.4,

$$y_r = f(x_r) = y_0 + r\,\Delta y_0 + \frac{r(r-1)}{2!}\Delta^2 y_0 + \frac{r(r-1)(r-2)}{3!}\Delta^3 y_0 + \ldots.$$

$$= 39.1961 + (0.5 \times 0.602) + \frac{0.5(0.5-1)}{2}(-0.0059)$$

$$+ \frac{0.5\,(0.5-1)\,(0.5-2)}{3 \times 2 \times 1}\,(-0.0001)$$

$$+ \frac{0.5\,(0.5-1)\,(0.5-2)\,(0.5-3)}{4 \times 3 \times 2 \times 1}\,(0.0004)$$

$= 39.4978$

Thus $f(50.5) = 39.4978$

Ex.5.4.2 *For the following data calculate forward differences and obtain the forward difference polynomial. Interpolate this polynomial at x = 0.25.*

x	0.1	0.2	0.3	0.4	0.5
y = f(x)	1.40	1.56	1.76	2.00	2.28

Sol. : The forward difference table is shown below.

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
0.1	1.40				
		0.16			
0.2	1.56		0.94		
		0.20		0	
0.3	1.76		0.04		0
		0.24		0	
0.4	2.00		0.04		
		0.28			
0.5	2.28				

Here we have, $x_0 = 0.1, \quad y_0 = 1.40$

$$h = x_1 - x_0 = 0.2 - 0.1 = 0.1$$

From equation 5.4.5 we know that,

$$r = \frac{x_r - x_0}{h}$$

$$\therefore \quad r = \frac{x_r - 0.1}{0.1} \quad \ldots (5.4.6)$$

From equation 5.4.4, Newtons forward differences interpolation formula is given as

$$y_r = y_0 + r\,\Delta y_0 + \frac{r(r-1)}{2!}\Delta^2 y_0 + \frac{r(r-1)(r-2)}{3!}\Delta^3 y_0 + \ldots..$$

Putting the value of 'r' from equation 5.4.6 in above equation,

$$y_r = 1.4 + \left(\frac{x_r - 0.1}{0.1}\right) \times 0.16 + \frac{\left(\frac{x_r - 0.1}{0.1}\right)\left(\frac{x_r - 0.1}{0.1} - 1\right)}{2!} \times 0.04$$

$$= 1.4 + 1.6(x_r - 0.1) + \frac{(x_r - 0.1)(x_r - 0.2)}{0.02} \times 0.04$$

$$= 1.4 + 1.6\,x_r - 0.16 + 2x_r^2 - 0.6\,x_r + 0.04$$

$$= 2x_r^2 + x_r + 1.28$$

If we replace x_r by x in above equation then,

$$y = 2x^2 + x + 1.28$$

or $$f(x) = 2x^2 + x + 1.28 \text{ [since y = f (x)]} \quad \ldots (5.4.7)$$

This is the required expression for interpolating polynomial. The equation for f(x) obtained is the equation of the curve which passes through all the given data points.

To interpolate this polynomial at x = 0.25, substitute this value of x in equation 5.4.7, i.e.,

$$f(0.25) = 2\times(0.25)^2 + 0.25 + 1.28$$

$$= 1.655$$

Ex.5.4.3 *The data points are given below,*

x	45	46	50	55	60	65
y	2.8710	Y	2.4040	2.0830	1.8620	1.7120

In the above table, obtain value of 'Y'.

Sol. : Here we observe that all values of 'x' except 46 are equally spaced. Therefore we can use newton's forward difference interpolation to find value of Y at x = 46. Lets prepare forward differences table as shown below.

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
45	2.8710				
		− 0.467			
50	2.4040		0.146		
		− 0.321		− 0.046	
55	2.0830		− 0.100		0.017
		− 0.221		− 0.029	
60	1.8620		− 0.071		
		− 0.150			
65	1.7120				

Here Let $x_0 = 45, \quad y_0 = 2.8710$

& $h = x_1 - x_0 = 50 - 45 = 5$

& $x_r = 46$

$\therefore \quad r = \dfrac{x_r - x_0}{h}$ (From equation 5.4.5)

$\therefore \quad r = \dfrac{46 - 45}{5} = 0.2$

Thus $Y = f(x_r = 46)$

or $y_r = Y$

From equation 5.4.4 y_r is given as,

$$y_r = y_0 + r\,\Delta y_0 + \frac{r(r-1)}{2!}\Delta^2 y_0 + \frac{r(r-1)(r-2)}{3!}\Delta^3 y_0 + \ldots$$

$$= 2.871 + 0.2(-0.467) + \frac{0.2(0.2-1)}{2!}(0.146)$$

$$+ \frac{0.2(0.2-1)(0.2-2)}{3!}(-0.046)$$

$$+ \frac{0.2(0.2-1)(0.2-2)(0.2-3)}{4!}(0.017)$$

$$= 2.763141$$

Ex.5.4.4 *The following table gives the temperatures recorded in Mumbai from first February to first July in 1987.*

Feb	March	April	May	June	July
30.3°C	32.1°C	37.2°C	39.8°C	35.3°C	29.8°C

Find out the approximate value of temperature on 15th Feb, 1987.

Sol. : Lets give numbers to the months as shown below in the table. Call months as 'x' and temperatures as 'y'. x_0 = Feb = 1, x_1 = March = 2, ... x_5 = July = 6. Let's prepare forward differences table as shown below.

x = Month	y = Temp.	Δ y	Δ^2 y	Δ^3 y	Δ^4 y	Δ^5 y
x_0 = Feb = 1	30.3					
		1.8				
x_1 = Mar = 2	32.1		3.3			
		5.1		– 5.8		
x_2 = Apr = 3	37.2		– 2.5		1.2	
		2.6		– 4.6		9.5
x_3 = May = 4	39.8		– 7.1		10.7	
		– 4.5		6.1		
x_4 = June = 5	35.3		– 1.0			
		– 5.5				
x_5 = July = 6	29.8					

The temperature on 15th February is required. This month is given the number.

$$x_0 = 1$$

15th February means half month after x_0. Thus we can write,

$$x_r = x_0 + 05 = 1.5$$

Therefore we have to interpolate 'y' at this value of x_r.

$$h = x_1 - x_0 = 2 - 1 = 1$$

$$r = \frac{x_r - x_0}{h} = \frac{1.5 - 1}{1} = 05$$

From equation 5.4.4 value of y_r is given as,

$$y_r = y_0 + r\,\Delta y_0 + \frac{r(r-1)}{2!}\Delta^2 y_0 + \frac{r(r-1)(r-2)}{3!}\Delta^3 y_0 + \ldots..$$

$$= 30.3 + 0.5 \times 1.8 + \frac{0.5\,(0.5-1)}{2!} \times 3.3$$

$$+ \frac{0.5\,(0.5-1)\,(0.5-2)}{3!} \times (-5.8)$$

$$+ \frac{0.5\,(0.5-1)\,(0.5-2)\,(0.5-3)}{4!} \times (1.2)$$

$$+ \frac{0.5\,(0.5-1)\,(0.5-2)\,(0.5-3)\,(0.5-4)}{5!} \times 9.5$$

$$= 30.637891$$

Thus approximate value of temperature on 15th February will be 30.637891° C. In short we interpolated y_r at $x_1 = 1.5$.

Thus given any type of data; you have to convert that data in suitable interpolation method.

In this example we obtained interpolated value of temperature, but it can be rainfall also. Other similar examples of percentage of marks, population of cities, attendance of students etc.

5.4.1.2 Algorithm and C Program

Based on the Newton's gregery forward difference interpolation we discussed in the last section, now we will prepare an algorithm and logic for computer program.

Algorithm :

Step 1 : Read number of total data points and values of those data points. i.e.

x & $y = f(x)$.

Step 2 : Read value of $x = x_r$ at which y is to be interpolated.

Step 3 : Calculate forward differences.

$$\Delta y_0 = y_1 - y_0$$
$$\Delta^2 y_0 = \Delta y_1 - \Delta y_0$$
$$\Delta^3 y_0 = \Delta^2 y_1 - \Delta^2 y_0 \qquad \text{\& so on.}$$

Step 4 : Calculate

$$h = x_1 - x_0$$

$$\& \; r = \frac{x_r - x_0}{h}$$

Step 5 : Calculate

$$y_r = y_0 + r\,\Delta y_0 + \frac{r(r-1)}{2!}\Delta^2 y_0 + \frac{r(r-1)(r-2)}{3!}\Delta^3 y_0 + \ldots..$$

Step 6 : The interpolated value of y at $x = x_r$ is equal to y_0

i.e. $f(x_r) = y_r$

Step 7 : Display x_r and $y_r = f(x_r)$ on the screen and stop.

Flow chart :

Now based on the algorithm above, we will prepare simplified flowchart for this method. We have already prepared the flowchart for forward differences. We will show here the flowchart to calculate forward differences as a subroutine. Fig. 5.4.1 shows the flowchart.

Computer program :

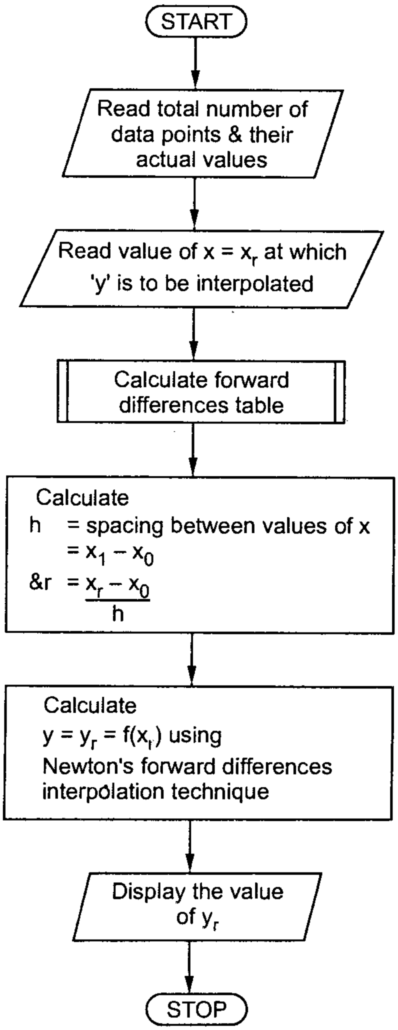


Fig. 5.4.1 Flow chart of Newton forward differences interpolation method

The source code of the 'C' program for newton's forward differences interpolation method is shown below.

```
/* Download this program from  www.vtubooks.com                        */
/* File name : nwtn_for.cpp                                            */

/*------ NEWTON'S FORWARD DIFFERENCES INTERPOLATION  METHOD -----------*/

/*       THE PROGRAM GENERATES A FORWARD DIFFERENCES TABLE FROM GIVEN

         DATA, & CALCULATES THE VALUE OF f(x) AT GIVEN VALUE OF x.

         INPUTS :   1) Number of entries of the data.
                    2) Values of 'x' & corresponding y = f(x).
                    3) Value of 'xr' at which y = f(x) to be calculated.
         OUTPUTS : Interpolated value f(x) at x = xr.                   */

/*------------------------------    PROGRAM  --------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>

void main()
{
      double y[20][20],x[20],xr,yr,h,r,sum,fy,facto;
      int i,j,k,n,m,t;

      clrscr();
      printf("\n    NEWTON'S FORWARD DIFFERENCES INTERPOLATION TECHNIQUE");

      printf("\n\nEnter the number of entries (max 20) = ");
                              /* ENTER THE NUMBER OF ENTRIES IN THE TABLE   */
      scanf("%d",&n);

      for(i = 0; i < n; i++)
      {                      /*  LOOP TO GET x AND y = f(x)  IN THE TABLE    */

          printf("x%d = ",i);  scanf("%lf",&x[i]);
          printf("            y%d = ",i);  scanf("%lf",&y[i][0]);
      }

      printf("\nEnter the value of xr at which  y = f(x) is to be "
              "interpolated, xr  = ");
      scanf("%lf",&xr);

      h =  x[1] - x[0];
      r = (xr - x[0])/h;         /* CALCULATE VALUE OF 'r'  */
      printf("\nThe value of h = %lf    and  value of r = %lf\n",h,r);

      k = n;
      for(j = 1; j < n; j++)
      {         /*  LOOP TO CALCULATE FORWARD DIFFERENCES IN THE TABLE  */
          k = k - 1;
          for(i = 0; i < k; i++)
          {
             y[i][j] = y[i+1][j-1] - y[i][j-1];
          }
      }

      sum = 0;
      for(t = 1; t < n; t++)
      {   /* LOOP FOR NEWTON'S  FORWARD DIFFERENCE INTERPOLATION FORMULA */
         fy = 1;
         facto = 1;
         for(m = 0; m < t; m++)
         {
            fy = fy * (r - m);
            facto = facto * (m + 1);
         }
         fy = fy * (y[0][t]/facto);
         sum = sum + fy;
      }
```

```
    yr = sum + y[0][0];

    printf("\nThe value of y = f(x) at xr = %lf is yr = %lf", xr,yr);

}
/*--------------------------- END OF PROGRAM ---------------------------*/
```

As can be seen from the source code, the initial part of the program is similar to that we have seen to general forward differences table. The third `printf` and `scanf` statement asks for number of entries (data points). i.e.,

```
Enter the number of entries (max 20) =
```

Here your data points should not be more than 20. If the data points are greater than 20, then increase the size of array y and x.

Next there is a `for` loop to get values of x and y in the arrays `x` and `y`.

The next statement is

```
Printf ("\n Enter the value of xr at which y=f(x) is to be
interpolated, xr= ");
scanf ("% lf", & xr);
```

These two statements get the value of x_r at which value of 'y' is to be calculated.

The next statement calculates n and r. Those values of 'h' and 'r' are then printed on the screen by next `printf` statement.

Next, there is a `for` loop and one more `for` loop in it. This loop calculates forward differences. This loop we have taken from the program of forward differences discussed in section 5.3.1.2.

Next, there is `for` loop, which implements newton's forward differences interpolation formula. This loop is reproduced here for explanation.

```
for (t=1; t<n; t++)  ← loop for terms in expansion of yr.
{
        fy = 1;
        facto = 1;  ← calculates factorial in every term.
        for (m=0; m<t; m++)  ← loop for calculation of every term
        {
                fy = fy* (r-m);  ← calculation of r(r–1), r(r–1) (r–2)etc
                products.
                facto = facto * (m+1);  ← calculation of factorial
        }
        fy = fy*(y[0][t]/facto);  ← calculation of value of each term.
        sum = sum + fy;  ← Addition of every calculated term.
}
```

In the outer `for` loop when `t=1` then it calculates second term i.e. $r\,\Delta\,y_0$, if t = 2, it calculates $\frac{r\,(r-1)}{2!}\,\Delta^2\,y_0$ and so on.

In the inner for `loop` the first statement is,

fy = fy*(r – m);

This calculates r (r – 1), when t = 2, r (r – 1) (r – 2) when t = 3 and so on. The next statement calculates factorials. The next statement is out of the inner `for` loop, i.e.

fy = fy*(y[0][t]/facto);

This calculates values of terms. `y[0][t]` gives $\Delta\,y_0$ if t =1, $\Delta^2\,y_0$ if t = 2, $\Delta^3\,y_0$ if t = 3 and so on. The last statement adds all the terms.

After the `for` loop the next statement is,

yr = sum+y[0][0]

This statement adds sum obtained in `for` loop to y_0. Here `y[0][0]` = y_0. Thus `yr` gives value of y_r.

The last statement in the program is `printf` statement and it prints values of x_r and y_r.

How to Run this Program?

Compile and make EXE file of the source code of the program. We will use the data points of example 5.4.1 to illustrate this program. The data points given in example 5.4.1 are reproduced here with their x and y notations for convenience.

x	$x_0 = 50$	$x_1 = 51$	$x_2 = 52$	$x_3 = 53$	$x_4 = 54$
y = f (x)	$y_0 = 39.1961$	$y_1 = 39.7981$	$y_2 = 40.3942$	$y_3 = 40.9843$	$y_4 = 41.5687$

'y' is to be calculated at $x_r = 50.5$

After running this program, it displays names of the method and then it displays,

```
Enter the number of entries (max 20) =
```

Here enter `5' and press `enter' key

Then it displays,

x0 = Here enter 50 and press `enter' key

y0 = Here enter 39.1961 and press `enter' key

```
x1 = 51 ↵
        y1 = 39.7981 ↵
x2 = 52 ↵
        y2 = 40.3942 ↵
x3 = 53 ↵
        y3 = 40.9843 ↵
x4 = 54 ↵
        y4 = 41.5687 ↵
```

Then the program displays,

```
    Enter the value of xr at which y = f(x) is to be interpolated,
xr = Here enter 50.5 and press 'enter' key.
```

The program displays value of yr as,

```
yr =   39.497816
```

which is similar to that we obtained in example 5.4.1.

Here is the display of complete results.

```
---------------------------- Results ----------------------------

        NEWTON'S FORWARD DIFFERENCES INTERPOLATION TECHNIQUE

Enter the number of entries (max 20) = 5
x0 = 50
            y0 = 39.1961
x1 = 51
            y1 = 39.7981
x2 = 52
            y2 = 40.3942
x3 = 53
            y3 = 40.9843
x4 = 54
            y4 = 41.5687

Enter the value of xr at which  y = f(x) is to be interpolated, xr = 50.5

The value of h = 1.000000    and  value of r = 0.500000

The value of y = f(x) at xr = 50.500000 is yr = 39.497816
-----------------------------------------------------------------
```

*This program you can use for *any data, there is no need to change the program.*

5.4.2 Newton's Backward Differences Interpolation Method

(or Gregory – Newton's backward interpolation method)

This interpolation is used when x_r is close to last value x_n in the given data.

From equation 5.4.2 in the last section we know that,

$$f(x+rh) = E^r f(x)$$

In this equation replace x by x_n.

i.e. $$f(x_n+rh) = E^r f(x_n) \quad \text{... (5.4.8)}$$

Here we can write $f(x_n+rh) = y_r$ as a short hand notation. i.e.,

$$y_r = E^r f(x_n)$$

Similarly $f(x_n) = y_n$ can be a short hand notation. i.e.,

$$y_r = E^r y_n \quad \text{... (5.4.9)}$$

Let's rearrange power of E as $(E^{-1})^{-r}$ for convenience. i.e.,

$$y_r = (E^{-1})^{-r} y_n$$

$$= (1-\nabla)^{-r} y_n \qquad \because \; E^{-1} = 1-\nabla \text{ from equation 5.2.9.}$$

Expanding $(1-\nabla)^{-r}$ binomially we get,

* Here any data means the data which are suitable to be used for newtons forward differences interpolation.

$$y_r = \left[1 + r\nabla + \frac{r(r+1)}{2!}\nabla^2 + \frac{r(r+1)(r+2)}{3!}\nabla^3 + \ldots\right] y_n$$

or

$$\boxed{y_r = y_n + r\nabla y_n + \frac{r(r+1)}{2!}\nabla^2 y_n + \frac{r(r+1)(r+2)}{3!}\nabla^3 y_n + \ldots} \quad \ldots (5.4.10)$$

5.4.2.1 Solved Examples

Ex.5.4.5 *From the given data. Find value of y at x = 4.5.*

x	1	2	3	4	5
y	2.38	3.65	5.85	9.95	14.85

Sol. : Here we have to find y at x = 4.5.

i.e. To find y_r at $x_r = 4.5$.

This value can be obtained by using both newton's forward as well as newton's backward interpolation techniques. But interpolation will be more accurate when use newtons backward difference interpolation. This is because x_r is close to the last value i.e. x = 5 (& $x_r = 4.5$).

Therefore let's prepare backward difference table.

x	y	∇y	$\nabla^2 y$	$\nabla^3 y$	$\nabla^4 y$
$x_0 = 1$	$y_0 = 2.38$				
		1.27			
$x_1 = 2$	$y_1 = 3.65$		0.93		
		2.20		0.97	
$x_2 = 3$	$y_2 = 5.85$		1.90		– 2.07
		4.10		– 1.10	
$x_3 = 4$	$y_3 = 9.95$		0.80		
		4.90			
$x_4 = 5$	$y_4 = 14.85$				

Here $\quad y_4 = 14.85, \quad \nabla y_4 = 4.90, \quad \nabla^2 y_4 = 0.80,$

$$\nabla^3 y_4 = -1.10, \quad \nabla^4 y_4 = -2.07$$

We know that,

$$x_r = x_0 + rh \qquad \text{From equation 5.4.1 in last subsection}$$

Since we are using backward differences we should express x_r in terms of x_n (last value) Replace x_0 by x_n in above equation.

i.e. $$x_r = x_n + rh \qquad \text{... (5.4.11)}$$

Thus we have to obtain $f(x_r)$, which is equal to y_r.

i.e. $$y_r = f(x_r) = f(x_n + rh)$$

or $$y_r = f(x_n + rh) = E^r f(x_n) \qquad \text{(From equation 5.4.8)}$$

& $$y_r = y_n + r\nabla y_n + \frac{r(r+1)}{2!}\nabla^2 y_n + \frac{r(r+1)(r+2)}{3!}\nabla^3 y_n + \ldots..$$

(From equation 5.4.10)

Thus we use newton's backward difference formula. To use this formula, we have to find r. From equation 5.4.11 we can easily obtain value of r as,

$$r = \frac{x_r - x_n}{h} \qquad \text{... (5.4.12)}$$

$\nabla y_n, \nabla^2 y_n, \nabla^3 y_n, \ldots$, etc are last differences in the backward difference table.

In the present example,

$$x_n = x_4 = 5 \qquad \text{(last value of x)}$$

& $$x_r = 4.5 \qquad \text{at which interpolation is required.}$$

$$\therefore \quad r = \frac{4.5 - 5}{1} \qquad \because \quad h = x_1 - x_0 = 1$$

$$\therefore \quad r = -0.5$$

& $$\nabla y_n = \nabla y_4 = 4.90, \; \nabla^2 y_n = \nabla^2 y_4 = 0.80$$

$$\nabla^3 y_n = \nabla^3 y_4 = -1.10, \; \nabla^4 y_n = \nabla^4 y_4 = -2.07$$

& $$y_n = y_4 = 14.85$$

Putting these values in expression of y_r above,

$$y_r = y_4 + r\nabla y_4 + \frac{r(r+1)}{2!}\nabla^2 y_4 + \frac{r(r+1)(r+2)}{3!}\nabla^3 y_4 + \ldots..$$

(Since $y_n = y_4$ in this example)

$$\therefore \quad y_r = 14.85 + (-0.5)(4.90) + \frac{(-0.5)(-0.5+1)}{2!}(0.80)$$

$$+ \frac{(-0.5)(-0.5+1)(-0.5+2)}{3!}(-1.10)$$

$$+ \frac{(-0.5)(-0.5+1)(-0.5+2)(-0.5+3)}{4!}(-2.07)$$

$$= 12.449609$$

Thus at x = 4.5 value of y = 12.449609.

Ex.5.4.6 *Use the data of example 5.4.2 and obtain the backward differences polynomial passing through all the points.*

Sol. : The data of example 5.4.2 is reproduced here for convenience.

x	0.1	0.2	0.3	0.4	0.5
y = f (x)	1.40	1.56	1.76	2.00	2.28

Now we will prepare the backward differences table from this data as shown below.

x	y	∇ y	∇^2 y	∇^3 y	∇^4 y
0.1	1.40				
		0.16			
0.2	1.56		0.04		
		0.20		0.00	
0.3	1.76		0.04		0.0
		0.24		0.00	
0.4	2.00		0.04		
		0.28			
0.5	2.28				

Here we have last value of x as 0.5 and last value of y as 2.28.

i.e. $x_n = 0.5$ & $y_n = 2.28$

$$h = x_1 - x_0 = 0.2 - 0.1 = 0.1$$

$\therefore$ From equation 5.4.12,

$$r = \frac{x_r - x_n}{h}$$

$$\therefore \quad r = \frac{x_r - 0.5}{0.1} \quad \text{Here value of } x_r \text{ is not given.}$$

From table we have,

$$\nabla y_4 = 0.28, \qquad \nabla^2 y_4 = 0.04, \qquad \nabla^3 y_4 = \nabla^4 y_4 = 0.0$$

And $y_n = y_4 =$ last value.

From equation 5.4.10, y_r at x_r is given as,

$$y_r = y_n + r\,\nabla y_n + \frac{r(r+1)}{2!}\nabla^2 y_n + \frac{r(r+1)(r+2)}{3!}\nabla^3 y_n + \ldots..$$

Since $y_n = y_4$ in this example,

$$y_r = y_4 + r\,\nabla y_4 + \frac{r(r+1)}{2!}\nabla^2 y_4 + \frac{r(r+1)(r+2)}{3!}\nabla^3 y_4 + \ldots..$$

$$= 2.28 + \left(\frac{x_r - 0.5}{0.1}\right) \times 0.28 + \frac{\left(\frac{x_r - 0.5}{0.1}\right)\left(\frac{x_r - 0.5}{0.1} + 1\right)}{2!} \times 0.04$$

$$= 2.28 + (x_r - 0.5) \times 2.8 + (x_r - 0.5)(x_r - 0.4) \times 2$$

$$= 2x_r^2 + x_r + 1.28$$

Here $$y_r = f(x_r) = 2x_r^2 + x_r + 1.28$$

replace x_r by x in above equation, we obtain,

$$f(x) = 2x^2 + x + 1.28$$

This polynomial is exactly similar to that we obtained in example 5.4.2.

Ex. 5.4.7 *Construct difference table from the following data :*

x	50	51	52	53	54
f (x)	39.1961	39.7981	40.3942	40.9843	41.5687

Obtain f(50.5) using Newton's forward difference formula and f(53.4) using Newton's backward difference formula correct to 4 decimal places.

Sol. : The first part of this example is solved in example 5.4.1. The difference table is also given in this example. The relationship between forward and backward differences is given by equation 5.3.14 The difference table is reproduced below. The table indicates forward as well as backward differences which are required for calculations.

x	y	Δy or ∇y	$\Delta^2 y$ or $\nabla^2 y$	$\Delta^3 y$ or $\nabla^3 y$	$\Delta^4 y$ or $\nabla^4 y$
$x_0 = 50$	$y_0 = 39.1961$				
		$\Delta y_0 = 0.602$			
51	39.7981		$\Delta^2 y_0 = -0.0059$		
		0.5961		$\Delta^3 y_0 = -0.0001$	
52	40.3942		-0.0060		$\Delta^4 y_0 = \Delta^4 y_4 = 0.0004$
		0.5901		$\nabla^3 y_4 = -0.0003$	
53	40.9843		$\nabla^2 y_4 = -0.0057$		
		$\nabla y_4 = 0.5844$			
$x_4 = 54$	$y_4 = 41.5687$				

f(50.5) is obtained using Newtons forward differences interpolation method in example 5.4.1 i.e.,

$$f(50.5) = 39.4978$$

To obtain f(53.4) using newton's backward differences formula :

Here we have $x_r = 53.4$ and $x_n = x_4 = 54$. Then 'r' is given by equation 5.4.12 as,

$$r = \frac{x_r - x_n}{h}$$

Putting the values

$$r = \frac{53.4 - 54}{1} = -0.6 \text{ Since } h = x_1 - x_0 = 51 - 50 = 1$$

Newton's backward differences interpolation formula is given by equation 5.4.10 as,

$$y_r = y_n + r\nabla y_n + \frac{r(r+1)}{2!}\nabla^2 y_n + \frac{r(r+1)(r+2)}{3!}\nabla^3 y_n + \ldots\ldots$$

For $n = 4$ above equation becomes,

$$y_r = y_4 + r\nabla y_4 + \frac{r(r+1)}{2!}\nabla^2 y_4 + \frac{r(r+1)(r+2)}{3!}\nabla^3 y_4 + .\frac{r(r+1)(r+2)(r+3)}{r!}\nabla^4 y_4$$

$r = -0.6$ and from difference table we have,

$y_4 = 41.5687$, $\nabla y_4 = 0.5844$, $\nabla^2 y_4 = -0.0057$, $\nabla^3 y_4 = -0.0003$,

$\nabla^4 y_4 = 0.0004$

$$\therefore\ y_r = 41.5687 - 0.6(0.5844) - \frac{0.6(-0.6+1)}{2!}(-0.0057)$$

$$-\frac{0.6(-0.6+1)(-0.6+2)}{3!}(-0.0003) - \frac{0.6(-0.6+1)(-0.6+2)(-0.6+3)}{4!}(0.0004)$$

$$= 41.218747$$

Thus f(5,3,4) = 41.218747 is the required value.

Ex. 5.4.8 *Compute sin (0.157) and sin (0.215) from the following data.*

x	0.15	0.17	0.19	0.21	0.23
sin x	0.14944	0.16918	0.18886	0.20846	0.22798

[May - 2001, 8 marks, May - 99, 8 marks, Dec - 97, 8 marks]

Sol. : Here x is evenly spaced. We have to evaluate sin (0.157), which is neat begining of the data. Hence we can use newton's forward difference interpolation. Similarly sin (0.215) is near end of the data, hence we can use newton's backward differences interpolation. The difference table for forward and backward differences is shown below.

x	$y = \sin(x)$	Δy or ∇y	$\Delta^2 y$ or $\nabla^2 y$	$\Delta^3 y$ or $\nabla^3 y$	$\Delta^4 y$ or $\nabla^4 y$
$x_0 = 0.15$	$y_0 = 0.14944$				
		$\Delta y_0 = 0.01974$			
$x_1 = 0.17$	$y_1 = 0.16918$		$\Delta^2 y_0 = -0.00006$		
		0.01968		$\Delta^3 y_0 = -0.00002$	
$x_2 = 0.19$	$y_2 = 0.18886$		-0.00008		$\Delta^4 y_0 = \nabla^4 y_4 = 0.00002$
		0.0196		$\nabla^3 y_4 = 0$	
$x_3 = 0.21$	$y_3 = 0.20846$		$\nabla^2 y_4 = -0.00008$		
		$\nabla y_4 = 0.01952$			
$x_4 = 0.23$	$y_4 = 0.22798$				

To obtain sin (0.157)

This can be obtained using forward differences interpolation. Newton's forward differences interpolation formula is given by equation 5.4.4 as,

$$y_r = y_0 + r\,\Delta y_0 + \frac{r(r-1)}{2!}\Delta^2 y_0 + \frac{r(r-1)(r-2)}{3!}\Delta^3 y_0 + \ldots..$$

Here $x_r = 0.157$ and $h = x_1 - x_0 = 0.17 - 0.15 = 0.02$

And, $$x_r = x_0 + r\,h$$

$$\therefore \quad r = \frac{x_r - x_0}{h}$$

$$= \frac{0.157 - 0.15}{0.02} = 0.35$$

Putting the values in newton's forward differences interpolation formula,

$$y_r = 0.14944 + 0.35\,(0.01974) + \frac{0.35\,(0.35-1)}{2!}(-\,0.00006)$$

$$+ \frac{0.35\,(0.35-1)\,(0.35-2)}{3!}(-\,0.00002)$$

$$+ \frac{0.35(0.35-1)\,(0.35-2)\,(0.35-3)}{4!}(0.00002)$$

$$= 0.156354$$

Thus sin (0.157) = 0.156354. The exact value is 0.1563558.

To obtain sin (0.215)

This can be obtained from backward differences interpolation. We have $x_r = 0.215$ and $x_n = x_4 = 0.23$. Then r is given as (equation 5.4.12),

$$r = \frac{x_r - x_n}{h}$$

$$= \frac{0.215 - 0.23}{0.02} = -\ 0.75$$

Newton's backward differences interpolation formula is given as (equation 5.4.10),

$$y_r = y_n + r \nabla y_n + \frac{r(r+1)}{2!} \nabla^2 y_n + \frac{r(r+1)(r+2)}{3!} \nabla^3 y_n + \ldots\ldots$$

Putting the values in above formula we get,

$$\begin{aligned} y_r &= 0.22798 + (-0.75)(0.01952) \\ &+ \frac{(-0.75)(-0.75+1)}{2!}(-0.00008) \\ &+ \frac{(-0.75)(-0.75+1)(-0.75+2)}{3!}(0) \\ &+ \frac{(-0.75)(-0.75+1)(-0.75+2)(-0.75+3)}{4!}(0.00002) \\ &= 0.213347 \end{aligned}$$

Thus sin (0.215) = 0.213347. The exact value is 0.2133474.

Ex. 5.4.9 *From the following table, find the number of students who*

i) Obtained less than 45 marks *ii) Obtained more than 65 marks*

[Dec - 2000, 8 marks, Dec - 97, 8 marks; Dec - 2003, 8 Marks]

Marks	30 – 40	40 – 50	50 – 60	60 – 70	70 – 80
Number of students	30	41	52	36	31

Sol. : Observe the given table carefully.

i) 30 students obtained marks less than 40.

ii) $30 + 41 = 71$ students obtained marks less than 50.

iii) $30 + 41 + 52 = 123$ students obtained marks less than 60.

iv) $30 + 41 + 52 + 36 = 159$ students obtained marks less than 70.

v) $30 + 41 + 52 + 36 + 31 = 190$ students obtained marks less than 80.

The above information can be tabulated as follows :

Marks less than x	40	50	60	70	80
Number of students y	30	71	123	159	190

We have to obtain the number of students who secured marks less than 45. This can be obtained with the help of Newton's forward differences interpolation, since it is near beginning of the data.

Next is, we want number of students who obtained marks more than 65. Since this is near end of the data, we will use Newton's backward differences interpolation. The forward/backward differences table is shown below.

x	y	Δy or ∇y	$\Delta^2 y$ or $\nabla^2 y$	$\Delta^3 y$ or $\nabla^2 y$	$\Delta^4 y$ or $\nabla^4 y$
$x_0 = 40$	$y_0 = 30$				
		$\Delta y_0 = 41$			
$x_1 = 50$	$y_1 = 71$		$\Delta^2 y_0 = 11$		
		52		$\Delta^3 y_0 = -27$	
$x_2 = 60$	$y_2 = 123$		-16		$\Delta^4 y_0 = \nabla^4 y_4 = 38$
		36		$\nabla^3 y_4 = 11$	
$x_3 = 70$	$y_3 = 159$		$\nabla^2 y_4 = -5$		
		$\nabla y_4 = 31$			
$x_4 = 80$	$y_4 = 190$				

i) To obtain number of students who secured less than 45 marks.

From equation 5.4.1, $x_r = x_0 + r\,h$

Here $x_r = 45$, $x_0 = 40$ and $h = x_1 - x_0 = 50 - 40 = 10$

$$\therefore \quad r = \frac{x_r - x_0}{h}$$

$$= \frac{45-40}{10} = 0.5$$

Newton's forward difference interpolation formula is given as (equation 5.4.4),

$$y_r = y_0 + r\,\Delta y_0 + \frac{r\,(r-1)}{2!}\,\Delta^2 y_0 + \frac{r\,(r-1)\,(r-2)}{3!}\,\Delta^3 y_0 + \ldots\ldots$$

Putting values in above equation,

$$y_r = 30 + 0.5\,(41) + \frac{0.5(0.5-1)}{2\,!}\,(11)$$

$$+ \frac{0.5\,(0.5-1)\,(0.5-2)}{3\ !}\,(-27)$$

$$+ \frac{0.5\,(0.5-1)\,(0.5-2)\,(0.5-3)}{4\ !}\,(38)$$

$$= 45.953125 \approx 46$$

Thus 46 students obtained marks less than 45.

ii) To obtain number of students who secured more than 65 marks :

Here we will first obtain the number of students who secured less than 65 marks using backward differences interpolation.

Hence we have $x_r = 65$

And from equation 5.4.12, $r = \frac{x_r - x_n}{h}$

$x_n = x_4 = 80$, hence above equation becomes,

$$r = \frac{65-80}{10} = -1.5$$

Newton's backward difference interpolation formula is given as (equation 5.4.10),

$$y_r = y_n + r\,\nabla y_n + \frac{r(r+1)}{2!}\nabla^2 y_n + \frac{r(r+1)(r+2)}{3!}\nabla^3 y_n + \ldots\ldots$$

Putting values in above equation from difference table,

$$y_r = 190 + (-1.5)\,31 + \frac{(-1.5)(-1.5+1)}{2!}(-5) + \frac{(-1.5)(-1.5+1)(-1.5+2)}{3!}(11) + \frac{(-1.5)(-1.5+1)(-1.5+2)(-1.5+3)}{4!}(38)$$

$$= 143.203125 \approx 143.$$

Thus 143 students obtained marks less than 65.

There are total 190 students. Hence $190 - 143 = 47$ students obtained marks greater than 65.

5.4.2.2 Algorithm and C Program

Now we will prepare an algorithm for newton gregory backward differences interpolation.

Algorithm :

Step 1 : *Read number of total data points and value of these data points i.e. x & $y = f(x)$.*

Step 2 : *Read the value of $x = x_r$ at which y is to be interpolated.*

Step 3 : *Calculate backward differences,*

$$\nabla y_n = y_n - y_{n-1}$$
$$\nabla^2 y_n = \nabla y_n - \nabla y_{n-1}$$
$$\nabla^3 y_n = \nabla^2 y_n - \nabla y_{n-1} \qquad \text{\& so on,}$$

Step 4 : *Calculate*

$$h = x_1 - x_0 \;\;\&\;\; r = \frac{x_r - x_n}{h}$$

Step 5 : *Calculate*

$$y_r = y_n + r\nabla y_n + \frac{r(r+1)}{2!}\nabla^2 y_n + \frac{r(r+1)(r+2)}{3!}\nabla^3 y_n + \ldots$$

Step 6 : *The interpolated value of y at $x = x_r$ is equal to y_r obtained in step 5.*
i.e. $f(x_r) = y_r$

Step 7 : *Display x_r & $y_r = f(x_r)$ on the screen and stop.*

Flowchart :

The algorithm presented above is almost similar to that of Newton's forward differences interpolation. Only the calculations of backward differences, 'r' and interpolation formula are modified. This flowchart is shown in Fig. 5.4.2. In the flowchart observe that we have shown one subroutine block to calculate backward differences table.

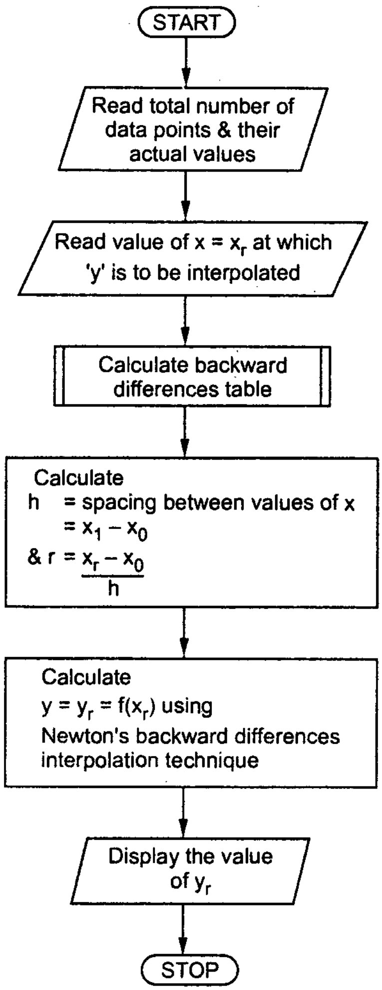


Fig. 5.4.2 Flowchart of Newton's backward differences interpolation method

Computer program :

The source code of the 'C' program for newton's backward differences interpolation method is shown below.

Initial part of the program is similar to that of the Newton's forward differences interpolation.

```
/* Download this program from  www.vtubooks.com                              */
/* File name : nwtn_bak.cpp                                                  */

/*------ NEWTON'S BACKWARD DIFFERENCES INTERPOLATION  METHOD -----------*/

/*       THE PROGRAM GENERATES A BACKWARD DIFFERENCES TABLE FROM GIVEN

         DATA, & CALCULATES THE VALUE OF f(x) AT GIVEN VALUE OF x.

         INPUTS :  1) Number of entries of the data.
                   2) Values of 'x' & corresponding y = f(x).
                   3) Value of 'xr' at which y = f(x) to be calculated.

         OUTPUTS : Interpolated value f(x) at x = xr.                      */

/*------------------------------    PROGRAM  --------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>

void main()
{
      double y[20][20],x[20],xr,yr,h,r,sum,fy,facto;
      int i,j,k,n,m,t;

      clrscr();
      printf("\n    NEWTON'S BACKWARD DIFFERENCES INTERPOLATION TECHNIQUE");

      printf("\n\nEnter the number of entries (max 20) = ");
                               /* ENTER THE NUMBER OF ENTRIES IN THE TABLE   */
      scanf("%d",&n);

      for(i = 0; i < n; i++)
      {                        /*  LOOP TO GET x AND y = f(x)  IN THE TABLE   */

         printf("x%d = ",i);  scanf("%lf",&x[i]);
         printf("             y%d = ",i);  scanf("%lf",&y[i][0]);
      }

      printf("\nEnter the value of xr at which  y = f(x)\nis to be "
             "interpolated, xr  = ");
      scanf("%lf",&xr);

      h =  x[1] - x[0];
      r = (xr - x[n-1])/h;          /* CALCULATE VALUE OF 'r'  */
      printf("\nThe value of h = %lf    and  value of r = %lf\n",h,r);

      k = 0;
      for(j = 1; j < n; j++)
      {                        /*  LOOP TO CALCULATE BACKWARD DIFFERENCES  */
         k++;
         for(i = n-1; i >= k; i--)
         {
            y[i][j] = y[i][j-1] - y[i-1][j-1];
         }
      }
      sum = 0;
      for(t = 1; t < n; t++)
      {  /* LOOP FOR NEWTON'S  BACKWARD DIFFERENCE INTERPOLATION FORMULA */
         fy = 1;
         facto = 1;
         for(m = 0; m < t; m++)
         {
            fy = fy * (r + m);
            facto = facto * (m + 1);
         }
```

```
      fy = fy * (y[n-1][t]/facto);
      sum = sum + fy;
    }

    yr = sum + y[n-1][0];

    printf("\nThe value of y = f(x) at xr = %lf is yr = %lf", xr,yr);

}
/*-------------------------- END OF PROGRAM ---------------------------*/
```

First the program asks for number of total data points or entries, then the `for` loop gets the values of x and y. Next the program asks for value of x_r at which y is to be interpolated. The next statement calculates h. Thus the program is exactly similarly to that of newton's forward differences interpolation upto the statement,

r = (x_r-x[n-1])/h; /* CALCULATE VALUE OF 'r' */

This statement is the implementation of $r = \frac{x_r - x_n}{h}$.

In the statement we have taken array element `x[n-1]` equal to x_n since counting starts from zero in the array.

The next statement is `printf`, to print the values of `h` and `xr` on the screen. The next `for` loop calculates backward differences table. Observe that this `for` loop we have taken as it is from the program of backward differences.

Next, there is `for` loop to implement backward differences interpolation formula. This loop is reproduced below for convenience.

```
for (t=1; t<n; t++)      ← loop for terms in the formula
{
      fy = 1;
      facto = 1; ← variable for factorial
      for (m=0; m<t; m++)              ← loop for calculation of r (r + 1),
                r (r +1) (r+2) and factorials.
      {
            fy = fy*(r+m); ← calculates r, r (r+1), r(r+1) (r+2) etc.
            facto = facto*(m+1); ← calculates factorial in the term
      }
      fy = fy * (y [n - 1][t]/facto);← calculates value of the
       term
      sum = sum + fy;                        ← Adds all the terms
}
```

The outer `for` loop is the counter for terms in the formula.

i.e. when t = 1 it calculates $r \nabla y_n$

$$t = 2 \quad \text{it calculates } \frac{r(r+1)}{2!} \nabla y_n \quad \text{\& so on.}$$

The inner `for` loop calculates actual value of the term. In this loop the first statement is,

```
fy = fy * (r + m);
```

This calculates r (r + 1) when t = 2, r (r +1) (r +2) when t = 3 and so on. The next statement is

```
facto = facto * (m + 1);
```

This calculates factorial in the term. The program then comes out of the inner `for` loop. The next statement is,

```
fy = fy * (y[n-1][t]/facto);
```

This calculates actual value of the term.

When t = 1,

`fy = y,    facto = 1`

& `y[n-1][1]` $= \nabla y_n$

$\therefore$ This statement gives $r \nabla y_n$ at t = 1

When t = 2,

`fy = r (r + 1),    facto = 2`

& `y[n-1][2]` $= \nabla^2 y_n$

$\therefore$ This statement gives $\frac{r(r+1)}{2!} \nabla y_n$ and so on.

The last statement in the `for` loop adds all the terms.

The next statement out of the `for` loop is,

```
yr = sum + y[n-1][0];
```

Here `y[n-1][0]` $= y_n$ and it gives final interpolated value of y_r. Next there is a `printf` statement to print values of x_r and y_r.

How to run this program?

Compile the source code of the program and make EXE file. Here we will use the data of example 5.4.5 for illustration. The data points of example 5.4.5 are reproduced below.

x	$x_0 = 1$	$x_1 = 2$	$x_2 = 3$	$x_3 = 4$	$x_4 = 5$
y	$y_0 = 2.38$	$y_1 = 3.65$	$y_2 = 5.85$	$y_3 = 9.95$	$y_4 = 14.85$

And $x_r = 4.5$ at which y_r is to be calculated.

Run the EXE file of the program on your computer. The program displays,

`Enter the number of entries (max 20)=` Here enter number of data points, i.e. 5 and press 'enter' key.

Then the program displays,

x0 = Here enter `1' and press `enter' key

$y0$ = Here enter 2.38 and press `enter' key

$x1$ = 2 ↵

$y1$ = 3.65 ↵

$x2$ = 3 ↵

$y2$ = 5.85 ↵

$x3$ = 4 ↵

$y3$ = 9.95 ↵

$x4$ = 5 ↵

$y4$ = 14.85 ↵

Here symbol `↵' means press 'enter' key after typing the number before the symbol.

The program then displays,

```
Enter the value of xr at which y=f(x)
is to be interpolated, xr =
```

Here enter $x_r = 4.5$ and press 'enter' key. The program then displays value of h, r and y_r. Here is the display of all the results.

As can be seen from the results.

```
----------------------------- Results ------------------------------

    NEWTON'S BACKWARD DIFFERENCES INTERPOLATION TECHNIQUE

Enter the number of entries (max 20) = 5
x0 = 1
            y0 = 2.38
x1 = 2
            y1 = 3.65
x2 = 3
            y2 = 5.85
x3 = 4
            y3 = 9.95
x4 = 5
            y4 = 14.85

Enter the value of xr at which  y = f(x)
is to be interpolated, xr  = 4.5

The value of h = 1.000000    and  value of r = -0.500000

The value of y = f(x) at xr = 4.500000 is yr = 12.449609
--------------------------------------------------------------------
```

$$x_r = 12.449609$$

which is similar to that we obtained in example. 5.4.5.

Important Note :

This method can be used for any data for which values of x are equally spaced and x_r is close to the last value x_n. Under these conditions there is no need to change the program.

5.4.3 Errors in Polynomial Interpolation

Let the function $f(x)$ be defined by (n + 1) points $(x_i, y_i) = 0, 1, 2, \dots n$. And let this function be approximated by a polynomial $\phi_n(x)$ of degree n. This polynomial $\phi_n(x)$ passes through all the data points. But $\phi_n(x)$ will not be exactly satisfying the original function f(x) except at data points. Therefore an error in this interpolation is given as,

$$\begin{aligned} E_n &= f(x) - \phi_n(x) \\ &= \frac{(x_r - x_0)(x_r - x_1)\dots(x_r - x_n)}{(n+1)!} f^{(n+1)}(z) \qquad \dots (5.4.13) \end{aligned}$$

Here $f^{(n+1)}(z)$ is the $(n+1)^{th}$ derivative of f (x) taken at x = z.

and z is any value between x_0 & x_n.

The above expression is not much useful for practical computations.

Errors in Newton's Forward and Backward Differences Interpolation

From equation 5.4.13 we know that general error formula is given as,

$$\begin{aligned} E_n &= f(x) - \phi_n(x) \\ &= \frac{(x_r - x_0)(x_r - x_1)\dots(x_r - x_n)}{(n+1)!} f^{(n+1)}(z) \end{aligned}$$

$$x_0 < z < x_n$$

For newton's forward differences interpolation it can be shown that,

$$f^{(n+1)}(x) = \frac{1}{h^{n+1}} \Delta^{n+1} f(x)$$

On putting this in equation above,

$$\begin{aligned} E_n &= f(x) - \phi_n(x) \\ &= \frac{(x_r - x_0)(x_r - x_1)\dots(x_r - x_n)}{(n+1)!\,h^{n+1}} \Delta^{n+1} f(z) \qquad \dots (5.4.14) \end{aligned}$$

Here $\phi_n(x) =$ is the interpolated polynomial and $\Delta^{n+1} f(z)$ is the $(n+1)^{th}$ order forward difference.

We know that $\frac{x_r - x_0}{h} = r$, with this substitution above equation becomes,

$$\boxed{f(x) - \phi_n(x) = \frac{r(r-1)(r-2)\dots(r-n)}{(n+1)!} \Delta^{n+1} f(z)} \qquad \dots (5.4.15)$$

This is the expression for truncation error in polynomial interpolation for newton's forward differences interpolation.

For newton's backward differences interpolation formula we know that,

$$r = \frac{x_r - x_n}{h}$$

With this substitution equation 5.4.13 becomes,

$$\boxed{f(x) - \phi_n(x) = \frac{r(r+1)(r+2)\ldots(r+n)}{(n+1)!} h^{n+1} f^{(n+1)}(z)} \quad \ldots (5.4.16)$$

& $x_0 < z < x_n$

Exercise

1. *The population of a town decennial census was as given below. Estimate the population in the year 1895.*

Year	1891	1901	1911	1921	1931
Population	46	66	81	93	101

[Ans. : 54.85]

2. *In the table given below the values of y are consecutive terms of series of which the number 21.6 is the 6^{th} term. Find first and 10^{th} term of series.*

x	3	4	5	6	7	8	9
y	2.7	6.4	12.5	21.6	34.3	51.2	72.9

[**Hint :** To obtain first term use newtons forward differences interpolation formula with $x_0 = 3$ & $x_r = 10$. To obtain 10^{th} term use newton's backward differences interpolation formula with $x_n = 9$ and $x_r = 10$. This is the problem of extrapolation where x_r lies out of the values of x given].

[Ans. : f (10) = 100 and f (1) = 0.1]

3. *Find the cubic polynomial which takes the following values $f(0) = 1, f(1) = 0, f(2) = 1$ & $f(3) = 10$. Hence obtain $f(4)$*
[**Hint :** Use forward or backward differences interpolation].

[Ans. : $\mathbf{f(x) = x^3 - 2x^2 + 1}$ & $\mathbf{f(4) = 33}$]

4. *Apply newton's forward differences interpolation to find a polynomial of degree four or less which takes the values of x_i and y_i as shown in data below.*

x_i	1	2	3	4	5
y_i	1	–1	1	–1	1

$$\left[\textbf{Ans.: } \mathbf{f(x) = \frac{1}{3}\left(2x^4 - 24x^3 + 100x^2 - 168x + 93\right)} \right]$$

5. *From the data find y at x = 4.5 :*

x	1	2	3	4	5
y	2.38	3.65	5.85	9.95	14.85

Hint : Use newton's backward differences interpolation.

Ans. : $y(4.5) = 12.449609$ **[May - 96, 6 marks]**

University Questions

1. Derive Newton's forward difference interpolation formula. Comment on the use of Newton's forward backward difference interpolation formulae.

[May - 97, May - 99, May - 2001]

2. Given the table

x	0.15	0.17	0.19	0.21	0.23
Sin x	0.14944	0.16918	0.18886	0.20846	0.22798

evaluate sin (0.157) ans sin (0.235). **[Dec - 97, May - 99, May - 2001]**

3. From the following table, find the number of students who
   i) Obtained less than 45 marks
   ii) Obtained more than 65 marks

Marks	30-40	40-50	50-60	60-70	70-80
No. of students	30	41	52	36	31

[Dec - 97, Dec - 2000]

4. Derive Newton's backward interpolation formula. **[Dec - 2001]**

5. Find the number of students from the following data who secured marks not more than 45. **[Dec - 2003]**

Marks	30-40	40-50	50-60	60-70	70-80
Number of students	35	48	70	40	22

5.5 Newton's Divided Difference Interpolation Method

This technique uses divided differences for interpolation. This interpolation method can be used with any type of data irrespective of the spacing between values of x.

Newton's divided difference interpolation formula is given as,

$$\boxed{\begin{aligned} y_r = {} & y_0 + (x_r - x_0)\, y(x_1\ x_0) + (x_r - x_0)(x_r - x_1)\, y(x_2\ x_1\ x_0) \\ & + (x_r - x_0)(x_r - x_1)(x_r - x_2)\, y(x_3\ x_2\ x_1\ x_0) + \ldots. \\ & + (x_r - x_0)(x_r - x_1)\ldots.(x_r - x_{n-1})\, y(x_n\ x_{n-1} \ldots x_0) \end{aligned}}$$

... (5.5.1)

Here, x_r = The value of x at which we want to interpolate y

$y(x_1\ x_0)$ = First order divided difference over (x_1, x_0)

$y(x_2\ x_1\ x_0)$ = Second order divided difference over (x_2, x_1, x_0)

... & so on.

Truncation error in divided differences interpolation :

If $\phi_n(x)$ is the approximated polynomial to function f (x), then error is given as,

$$E_n = f(x) - \phi_n(x)$$

From equation 5.4.13 'E_n' is given as,

$$E_n = \frac{(x_r - x_0)(x_r - x_1)\ldots(x_r - x_n)}{(n+1)!} f^{(n+1)}(z)$$

& $\quad x_0 \le z \le x_n$

For the divided differences, above equation is equivalent to,

$$E_n = y(x_n\ x_{n-1} \cdots x_0\ x_r)(x_r - x_0)(x_r - x_1)\ldots(x_r - x_n)$$

This is the equation of error in interpolation.

5.5.1 Solved Examples

Ex. 5.5.1 *Certain corresponding values of x and* $\log_{10} x$ *are (300, 2.4771), (304, 2.4829), (305, 2.4841) and (307, 2.4871). Find* $\log_{10} 301$.

[Dec - 2001, 8 marks; Dec-2003, 8 Marks]

Sol. : Here the function is,

$$y = \log_{10} x$$

The value of x and y are given in the tabular form as shown below.

x	$x_0 = 300$	$x_1 = 304$	$x_2 = 305$	$x_3 = 307$
$y = \log_{10} x$	$y_0 = 2.4771$	$y_1 = 2.4829$	$y_2 = 2.4843$	$y_3 = 2.4871$

Here we observe that $x_2 - x_1 \ne x_1 - x_0$ (i.e. values of x are not equally spaced). Therefore we use newtons divided difference formula. Let's prepare divided difference table for this data.

Please see section 5.3.3 for how to prepare divided differences table. From table 5.3.7 of divided differences we can write the following.

$$y(x_1\ x_0) = \frac{y_1 - y_0}{x_1 - x_0} = \frac{2.4839 - 2.4771}{304 - 300} = 0.00145$$

$$y(x_2\ x_1) = \frac{y_2 - y_1}{x_2 - x_1} = \frac{2.4843 - 2.4829}{305 - 304} = 0.0014$$

$$y(x_3\ x_2) = \frac{y_3 - y_2}{x_3 - x_2} = \frac{2.4871 - 2.4843}{307 - 305} = 0.0014$$

Now let's calculate second order divided differences.

$$y(x_2\ x_1\ x_0) = \frac{y(x_2\ x_1) - y(x_1\ x_0)}{x_2 - x_0} = \frac{0.0014 - 0.00145}{305 - 300} = -0.00001$$

$$y(x_3\ x_2\ x_1) = \frac{y(x_3\ x_2) - y(x_2\ x_1)}{x_2 - x_1} = \frac{0.0014 - 0.0014}{307 - 304} = 0$$

Now let's calculate third order divided difference.

$$y(x_3\ x_2\ x_1\ x_0) = \frac{y(x_3\ x_2\ x_1) - y(x_2\ x_1\ x_0)}{x_3 - x_0} = \frac{0 + 0.00001}{307 - 300} = 1.4286 \times 10^{-6}$$

All these divided differences are tabulated in the divided difference table as shown below.

Divided differences table of example 4.5.1.

x	$y = \log_{10} x$	$y(x_n\ x_{n-1})$	$y(x_n\ x_{n-1}\ x_{n-2})$	$y(x_n\ x_{n-1}\ x_{n-2}\ x_{n-3})$
$x_0 = 300$	$y_0 = 2.4771$			
		$y(x_1\ x_0) = 0.00145$		
$x_1 = 304$	$y_1 = 2.4829$		$y(x_2\ x_1\ x_0) = -0.00001$	
		$y(x_2\ x_1) = 0.0014$		$y(x_3\ x_2\ x_1\ x_0) = 1.4286 \times 10^{-6}$
$x_2 = 305$	$y_2 = 2.4843$		$y(x_3\ x_2\ x_1) = 0$	
		$y(x_3\ x_2) = 0.0014$		
$x_3 = 307$	$y_3 = 2.4871$			

We have to find $\log_{10} 301$

i.e. we have to find $y_r = log_{10}\, x_r$ at $x_r = 301$ newton's divided differences formula is given by equation 5.5.1 as,

$$\begin{aligned} y_r &= y_0 + (x_r - x_0)\, y(x_1\ x_0) + (x_r - x_0)(x_r - x_1)\, y(x_2\ x_1\ x_0) \\ &\quad + (x_r - x_0)(x_r - x_1)(x_r - x_2)\, y(x_3\ x_2\ x_1\ x_0) + \ldots \end{aligned}$$

Putting the values of divided differences, x_r and x in above formula we get,

$$\begin{aligned} y_r &= 2.4771 + (301 - 300)(0.00145) \\ &\quad + (301 - 300)(301 - 304)(-0.00001) \\ &\quad + (301 - 300)(301 - 304)(301 - 305)(1.4286 \times 10^{-6}) \\ &= 2.4785971 \end{aligned}$$

Thus value of $y = log_{10}\, x$ at x = 301 will be 2.4785971.

Ex. 5.5.2 *For the following data, find the polynomial f(x) which passes through all the points.*

x	−1	0	3	6	7
f (x)	3	−6	39	822	1611

Sol. : We use divided differences since the values of x are not equally spaced.

The divided difference table is shown below.

x	y = f(x)	$y(x_n x_{n-1})$	$y(x_n x_{n-1} x_{n-2})$	$y(x_n x_{n-1} x_{n-2} x_{n-3})$	$y(x_n x_{n-1} \ldots x_0)$
−1	3				
		−9			
0	−6		6		
		15		5	
3	39		41		1
		261		13	
6	822		132		
		789			
7	1611				

From equation 5.5.1 newton's divided difference interpolation formula is given as,

$$y_r = y_0 + (x_r - x_0)\, y(x_1\, x_0) + (x_r - x_0)(x_r - x_1)\, y(x_2\, x_1\, x_0) + (x_r - x_0)(x_r - x_1)(x_r - x_2)\, y(x_3\, x_2\, x_1\, x_0) + \ldots$$

Putting the values of divided differences and x in above formula we get

$$y_r = 3 + (x_r + 1)(-9) + x_r (x_r + 1)(6) + x_r (x_r + 1)(x_r - 3)(5) + x_r (x_r + 1)(x_r - 3)(x_r - 6)(1)$$
$$= x_r^4 - 3x_r^3 + 5x_r^2 - 6$$

Replace x_r by x in above equation,

$$y = f(x) = x^4 - 3x^3 + 5x^2 - 6$$

This is the required polynomial.

If we want y at x = 4, then put x = 4 in above equation. We get,

$$f(4) = (4)^4 - 3(4)^3 + 5(4)^2 - 6 = 138$$

Thus at x = 4, f (x) = 138.

Ex. 5.5.3 *Find the unique polynomial of degree 2 or less, such that $f(0) = 1$, $f(1) = 3$, $f(3) = 55$ using newton's divided difference interpolation.*

Sol. : We will prepare the divided differences table as shown below.

x	y = f (x)	$y(x_n x_{n-1})$	$y(x_n x_{n-1} x_{n-2})$
0	1		
		y (1, 0) = 2	
1	3		y (3, 1, 0) = 8
		y (3, 1) = 26	
3	55		

From equation newton's divided difference interpolation formula is given as,

$$y_r = y_0 + (x_r - x_0)\, y(x_r\, x_0) + (x_r - x_0)(x_r - x_1)\, y(x_2\, x_1\, x_0) + \ldots$$

Putting the values of x and divided differences,

$$y_r = 1 + (x_r - 0)(2) + (x_r - 0)(x_r - 1)(8) = 8x_r^2 - 6x_r + 1$$

Thus the polynomial is,

$$f(x) = 8x^2 - 6x + 1$$

Ex. 5.5.4 *A cromel-alumel thermocouple gives the following output for rise in temperature :*

Temp in °C	0	10	20	30	40	50
Output in mV	0.0	0.4	0.8	1.2	1.61	2.02

Find the output of thermocouple for temperature of 45°C using Newton's Divided Difference interpolation.

Sol. : Let us rewrite the table as follows :

Temp in °C x	$x_0 = 0$	$x_1 = 10$	$x_2 = 20$	$x_3 = 30$	$x_4 = 40$	$x_5 = 50$
Output in mV y	$y_0 = 0$	$y_1 = 0.4$	$y_2 = 0.8$	$y_3 = 1.2$	$y_4 = 1.61$	$y_5 = 2.02$

The first order divided differences can be calculated as follows :

$$y(x_1 x_0) = \frac{y_1 - y_0}{x_1 - x_0} = \frac{0.4 - 0}{10 - 0} = 0.04$$

$$y(x_2 x_1) = \frac{y_2 - y_1}{x_2 - x_1} = \frac{0.8 - 0.4}{20 - 10} = 0.04$$

$$y(x_3 x_2) = \frac{y_3 - y_2}{x_3 - x_2} = \frac{1.2 - 0.8}{30 - 20} = 0.04$$

$$y(x_4 x_3) = \frac{y_4 - y_3}{x_4 - x_3} = \frac{1.61 - 1.2}{40 - 30} = 0.041$$

$$y(x_5 x_4) = \frac{y_5 - y_4}{x_5 - x_4} = \frac{2.02 - 1.61}{50 - 40} = 0.041$$

Now second order divided differences can be calculated as follows :

$$y(x_2 x_1 x_0) = \frac{y(x_2 x_1) - y(x_1 x_0)}{x_2 - x_0} = \frac{0.04 - 0.04}{20 - 0} = 0$$

$$y(x_3 x_2 x_1) = \frac{y(x_3 x_2) - y(x_2 x_1)}{x_3 - x_1} = \frac{0.04 - 0.04}{30 - 10} = 0$$

$$y(x_4 x_3 x_2) = \frac{y(x_4 x_3) - y(x_3 x_2)}{x_4 - x_2} = \frac{0.041 - 0.04}{40 - 20} = 5 \times 10^{-5}$$

$$y(x_5 x_4 x_3) = \frac{y(x_5 x_4) - y(x_4 x_3)}{x_5 - x_3} = \frac{0.041 - 0.041}{50 - 30} = 0$$

Third order divided differences can be calculated as follows :

$$y(x_3x_2x_1x_0) = \frac{y(x_3x_2x_1) - y(x_2x_1x_0)}{x_3 - x_0} = \frac{0-0}{30-0} = 0$$

$$y(x_4x_3x_2x_1) = \frac{y(x_4x_3x_2) - y(x_3x_2x_1)}{x_4 - x_1} = \frac{5\times10^{-5} - 0}{40-10} = 1.66667\times10^{-6}$$

$$y(x_5x_4x_3x_2) = \frac{y(x_5x_4x_3) - y(x_4x_3x_2)}{x_5 - x_2} = \frac{0 - 5\times10^{-5}}{50-20} = -1.66667\times10^{-6}$$

Fourth order divided differences are calculated as follows :

$$y(x_4x_3x_2x_1x_0) = \frac{y(x_4x_3x_2x_1) - y(x_3x_2x_1x_0)}{x_4 - x_0} = \frac{1.66667\times10^{-6} - 0}{40-0} = 4.166667\times10^{-8}$$

$$y(x_5x_4x_3x_2x_1) = \frac{y(x_5x_4x_3x_2) - y(x_4x_3x_2x_1)}{x_5 - x_1} = \frac{-1.66667\times10^{-6} - 1.66667\times10^{-6}}{50-10}$$

$$= -8.33335\times10^{-8}$$

Fifth order divided difference is calculated as follows :

$$y(x_5x_4x_3x_2x_1x_0) = \frac{y(x_5x_4x_3x_2x_1) - y(x_4x_3x_2x_1x_0)}{x_5 - x_0}$$

$$= \frac{-8.33335\times10^{-8} - 4.166667\times10^{-8}}{50-0} = -2.5\times10^{-9}$$

From equation 5.5.1 we can write Newton's divided difference formula upto fifth difference as follows :

$$\begin{aligned} y_r = {} & y_0 + (x_r - r_0)\, y(x_1x_0) + (x_r - x_0)(x_r - x_1)\, y(x_2x_1x_0) \\ & + (x_r - x_0)(x_r - x_1)(x_r - x_2)\, y(x_3x_2x_1x_0) \\ & + (x_r - x_0)(x_r - x_1)(x_r - x_2)(x_r - x_3)\, y(x_4x_3x_2x_1x_0) \\ & + (x_r - x_0)(x_r - x_1)(x_r - x_2)(x_r - x_3)(x_r - x_4)\, y(x_5x_4x_3x_2x_1x_0) \end{aligned}$$

Puttig the values in above equation with

$x_r = 45$ we get,

$$\begin{aligned} y_r = {} & 0 + (45-0)\,(0.04) + (45-0)\,(45-10)\,(0) \\ & + (45-0)(45-10)(45-20)(0) \\ & + (45-0)(45-10)(45-20)(45-30)\,(4.166667\times10^{-8}) \\ & + (45-0)(45-10)(45-20)(45-30)(45-40)\,(-2.5\times10^{-9}) \\ = {} & 0 + 1.8 + 0 + 0 + 0.024609 - 7.382812\times10^{-3} \\ = {} & 1.8319918 \end{aligned}$$

Thus at the temperature of 45°C, the output voltage of thermocouple will be 1.8319918 mV.

Ex. 5.5.5 *Use Newton's divided difference interpolation formula to evaluate f(3) from the following table.*

x	0	1	2	4	5	6
f(x)	1	14	15	5	6	19

Sol. : Let us rewrite the given table as follows :

x	$x_0 = 0$	$x_1 = 1$	$x_2 = 2$	$x_3 = 4$	$x_4 = 5$	$x_5 = 6$
y = f(x)	$y_0 = 1$	$y_1 = 14$	$y_2 = 15$	$y_3 = 5$	$y_4 = 6$	$y_5 = 19$

We know that how to calculate divided differences. The procedure for calculating divided differences is explained in section 5.3.3. The divided difference table is shown below :

x	$y = f(x)$	$y(x_n x_{n-1})$	$y(x_n x_{n-1} x_{n-2})$	$y(x_n x_{n-1} x_{n-2} x_{n-3})$	$y(x_n x_{n-1} x_{n-2} x_{n-3} x_{n-4})$	$y(x_n x_{n-1} \ldots\ldots x_0)$
0	1					
		$y(x_1 x_0) = 13$				
1	14		$y(x_2 x_1 x_0) = -6$			
		1		$y(x_3 x_2 x_1 x_0) = 1$		
2	15		-2		$y(x_4 x_3 x_2 x_1 x_0) = 0$	
		-5		1		$y(x_5 x_4 x_3 x_2 x_1 x_0) = 0$
4	5		2		0	
		1		1		
5	6		6			
		13				
6	19					

Newton's divided difference formula is given by equation 5.5.1. Hence we can write,

$$
\begin{aligned}
y_r = \; & y_0 + (x_r - x_0) y(x_1 x_0) + (x_r - x_0)(x_r - x_1) y(x_2 x_1 x_0) \\
& + (x_r - x_0)(x_r - x_1)(x_r - x_2) y(x_3 x_2 x_1 x_0) \\
& + (x_r - x_0)(x_r - x_1)(x_r - x_2)(x_r - x_3) y(x_4 x_3 x_2 x_1 x_0) \\
& + (x_r - x_0)(x_r - x_1)(x_r - x_2)(x_r - x_3)(x_r - x_4) y(x_5 x_4 x_3 x_2 x_1 x_0)
\end{aligned}
$$

We have $x_r = 3$ to get $y_r = f(3)$. Putting the values in above equation,

$$\begin{aligned} y_r &= 1+(3-0)(13)+(3-0)(3-1)(-6)+(3-0)(3-1)(3-2)(1) \\ &\quad +(3-0)(3-1)(3-2)(3-4)(0)+(3-0)(3-1)(3-2)(3-4)(3-5)(0) \\ &= 10 \end{aligned}$$

Thus f(3) = 10 is an interpolated value

Ex. 5.5.6 *Find the interpolating polynomial for the data :*

x	0	1	2	5
f (x)	2	3	12	147

[Dec - 99, 8 marks, May - 98, 8 marks, May - 96, 8 marks]

Derive newton's divided difference formula and use it to compute $f(1.5)$ from the following data :

x	0	1	2	5
f (x)	2	3	12	147

[May - 2001, 8 marks]

Sol. : Let us rewrite the given data as follows :

x	$x_0 = 0$	$x_1 = 1$	$x_2 = 2$	$x_3 = 5$
y = f (x)	$y_0 = 2$	$y_1 = 3$	$y_2 = 12$	$y_3 = 147$

First we can obtain the interpolating polynomial using lagrange's or divided differences interpolation and then obtain the value of $f(1.5)$. The divided difference table is as shown below.

x	$y = f(x)$	$y(x_n\, x_{n-1})$	$y(x_n x_{n-1}\, x_{n-2})$	$y(x_n x_{n-1} x_{n-2} x_{n-3})$
$x_0 = 0$	$y_0 = 2$			
		$y(x_1 x_0) = 1$		
$x_1 = 1$	$y_1 = 3$		$y(x_2 x_1 x_0) = 4$	
		9		$y(x_3 x_2 x_1 x_0) = 1$
$x_2 = 2$	$y_2 = 12$		9	
		45		
$x_3 = 5$	$y_3 = 147$			

Equation 5.5.1 gives newton's divided differences interpolation formula. For $n=3$, it can be written as,

$$y_r = y_0 + (x_r - x_0) y(x_1 x_0) + (x_r - x_0)(x_r - x_1) y(x_2 x_1 x_0) + (x_r - x_0)(x_r - x_1)(x_r - x_2) y(x_3 x_2 x_1 x_0)$$

Putting values in above equation,

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

This is the interpolating polynomial for the given data. Now to determine $f(1.5)$, put $x_r = 15$ in above equation. i.e.,

$$\begin{aligned} y_r = f(1.5) &= (1.5)^3 + (1.5)^2 - 1.5 + 2 \\ &= 6.125 \end{aligned}$$

Thus $\quad f(1.5) = 6.125$

5.5.2 Algorithm and C Program

Now we will develop the computer logic to implement divided differences interpolation.

Algorithm :

Step 1 : Real number of total data points and values of those data points, i.e. x and $y = f(x)$.

Step 2 : Read the value of $x = x_r$ *at which y is to be interpolated.*

Step 3 : Calculate divided differences

$$y(x_1 x_0) = \frac{y_1 - y_0}{x_1 - x_0}$$

$$y(x_2 x_1 x_0) = \frac{y(x_2 x_1) - y(x_1 x_0)}{x_2 - x_0}$$

$$y(x_3 x_2 x_1 x_0) = \frac{y(x_3 x_2 x_1) - y(x_2 x_1 x_0)}{x_3 - x_0} \quad \textit{and so on.}$$

Step 4 : Calculate

$$\begin{aligned} y_r = y_0 &+ (x_r - x_0) y(x_1 x_0) + (x_r - x_0)(x_r - x_1) y(x_2 x_1 x_0) \\ &+ (x_r - x_0)(x_r - x_1)(x_r - x_2) y(x_3 x_2 x_1 x_0) + \ldots\ldots \\ &+ (x_r - x_0)(x_r - x_1)\ldots\ldots(x_r - x_{n-1}) y(x_n x_{n-1} \ldots\ldots x_0) \end{aligned}$$

Step 5 : The interpolated value of y at $x = x_r$ *is equal to* y_r *obtained in step 4. i.e.* $f(x_r) = y_r$

Step 6 : Display x_r *and* $y_r = f(x_r)$ *on the screen and stop.*

Flowchart :

Based on the algorithm above we can develop the flowchart for this method. This flowchart is shown in Fig. 5.5.1.

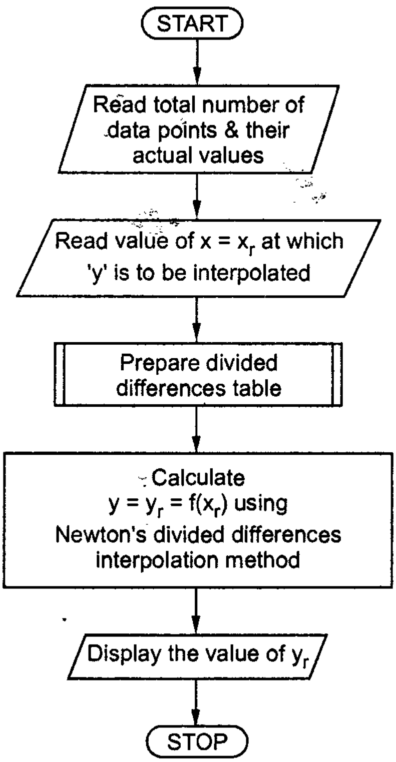


Fig. 5.5.1 Flowchart for newton's divided interpolation method

Here we have shown a single subroutine block to prepare divided differences table. We have shown a detailed flowchart to prepare divided differences table in Fig. 5.3.3. (in section 5.3.3.1).

Computer Program :

A 'C' program for newton's divided differences interpolation is shown below.

```
/* Download this program from  www.vtubooks.com                     */
/* File name : nwtn_div.cpp                                           */

/*------------ NEWTON'S DIVIDED DIFFERENCES INTERPOLATION METHOD --------*/

/*      THE PROGRAM GENERATES A DIVIDED DIFFERENCES TABLE FROM GIVEN

        DATA, AND IT CALCULATES  THE VALUE OF f(x) AT GIVEN VALUE
        OF xr.
        INPUTS :  1) Number of entries of the data.
                  2) Values of 'x' & corresponding y = f(x).
                  3) Value of xr at which f(x) is to be interpolated.
                  VALUES OF x NEED NOT BE EQUALLY SPACED.
        OUTPUTS : Interpolated value of f(x) at x = xr.                  */

/*------------------------------   PROGRAM  ----------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>

void main()
{
      double y[20][20],x[20],sum,fy,xr;
              /* ARRAY OF y[n][n] ELEMENTS FOR DIVIDED DIFFERENCE TABLE */
      int i,j,k,n,t,m;

      clrscr();
      printf("\n     NEWTON'S DIVIDED DIFFERENCES INTERPOLATION METHOD");

      printf("\n\nEnter the number of entries (max 20) = ");
                              /* ENTER THE NUMBER OF ENTRIES IN THE TABLE   */
      scanf("%d",&n);

      for(i = 0; i < n; i++)
      {                     /*  LOOP TO GET x AND y = f(x) IN THE TABLE    */

         printf("x%d = ",i);  scanf("%lf",&x[i]);
         printf("           y%d = ",i);  scanf("%lf",&y[i][0]);
      }

      printf("\nEnter the value of xr at which y = f(x) is to be"
             " calculated xr = ");
      scanf("%lf",&xr);

      k = n;
      for(j = 1; j < n; j++)
      {           /*  LOOP TO CALCULATE DIVIDED DIFFERENCES IN THE TABLE  */
         k = k - 1;
         for(i = 0; i < k; i++)
         {
            y[i][j] = (y[i+1][j-1] - y[i][j-1])/(x[i+j]-x[i]);
         }
      }
      sum = 0;
      for(t = 1; t < n; t++)
      {          /* LOOP TO CALCULATE INTERPOLATED VALUE OF 'y'   */
          fy = 1;
          for(m = 0; m < t; m++)
          {
             fy = fy * (xr - x[m]);
          }
          sum = sum + (fy * y[0][t]);
      }
      sum = sum + y[0][0];

      printf("\nThe interpolated value of y at xr = %lf"
             "-is yr = %lf\n",xr,sum);
}
/*-------------------- END OF PROGRAM --------------------------------*/
```

The initial part of the program is similar to previous programs we discussed. In this initial part there are variables declarations, then one `for` loop to get values of x and y, and `printf` and `scanf` statements to get value of x_r at which 'y' is to be interpolated.

Next, there is a `for` loop to calculate divided differences in the table. This `for` loop we have taken as it is from the program to generate divided differences table (see program given in section 5.3.3.2).

Next, there is a `for` loop to implement newton's divided differences interpolation formula.

This loop is reproduced below for explanation,

```
for  (t=1;  t<n;  t++) ← loop for terms in the formula
{
        fy  =  1;
        for  (m=0;  m<t;  m++)  ← loop to calculate value of the term
        {
              fy=fy*(xr-x[m]);  ← statement to calculate value of the terms
                                  (x_r - x_0), (x_r - x_0)(x_r - x_1)... etc.
        }
        sum=sum+(fy*y[0][t]);  ← statement to calculate value of the term.
}
sum  =  sum  +  y[0][0];    ← Interpolated value of y.
```

The outer `for` loop is the counter for terms to be summed up in the formula. In the inner `for` loop the statement is,

`fy = fy * (xr - x[m]);`

When t = 1 this statement calculates $(x_r - x_0)$

When t = 2 it calculates $(x_r - x_0)(x_r - x_1)$

When t = 3 it calculates $(x_r - x_0)(x_r - x_1)(x_r - x_2)$ and so on.

The next statement out of inner `for` loop is,

`sum = sum + fy * y[0][t];`

`fy` is the product obtained in the `for` loop. This product is multiplied by divided difference.

When t = 1, `y[0][1]` $= y(x_1\ x_0)$ and sum is the second term in formula. **i.e.** $(x_r - x_0)\, y(x_1\ x_0)$.

When t = 2, `y[0][2]=` $y\ (x_2\ \ x_1\ \ x_0)$, then

`fy * y[0][2]=` $(x_r - x_0)\,(x_r - x_1)\ y\ (x_2\ \ x_1\ \ x_0)$

Before executing this statement,

Sum $= (x_r - x_0)\, y\,(x_1\ x_0)$ from previous cycle.

After executing this statement.

```
Sum = Sum + fy * f[0][t];
```

Sum = $(x_r - x_0)\, y\,(x_1\ x_0) + (x_r - x_0)(x_r - x_1)\, y\,(x_2\ x_1\ x_0)$

Thus the statement calculates the values of individual terms plus adds the latest calculated term to previous terms.

The last statement is out of `for` loop. It is,

```
Sum = Sum + y[0][0];
```

This statement adds `y[0][0]` $= y_0$ (first term) to the sum obtained in the for loop. Thus sum after executing this statement is equal to y_r.

The last statement in the program is `printf` statement and it prints x_r and y_r (i.e. sum) on the screen.

How to run this program?

Compile and make EXE file of the 'C' program given here. Then run EXE file on your computer.

We will use data of example 5.5.1 to illustrate the program. The data points of example 5.5.1 are reproduced here for convenience.

x	$x_0 = 300$	$x_1 = 304$	$x_2 = 305$	$x_3 = 307$
$y = log_{10}\, x$	$y_0 = 2.4771$	$y_1 = 2.4829$	$y_2 = 2.4843$	$y_3 = 2.4871$

And we have to interpolate y at $x_r = 301$.

When you run the program on computer it displays names of methods and then it displays,

`Enter the number of entries (max 20) =` Here enter the number of data points, i.e. '4' and press 'enter' key.

Then the program displays,

x0 = Here enter 300 and press `enter' key

y0 = Here enter 2.4771 & press `enter' key

x1 = 304 ↵

y1 = 2.4829 ↵

x2 = 305 ↵

y2 = 2.4843 ↵

x3 = 307 ↵

y3 = 2.4871 ↵

Here symbol `↵' means press 'enter' key after typing the number.

Then the program displays,

```
Enter the value of xr at which y = f(x) is to be
interpolated xr =
```

Here enter value of $x_r = 301$ and press 'enter' key.

The program then displays the value of y_r as 2.478597. Here is the display of results.

```
------------------------- Results ---------------------------------

NEWTON'S DIVIDED DIFFERENCES INTERPOLATION METHOD

Enter the number of entries (max 20) = 4
x0 = 300
            y0 = 2.4771
x1 = 304
            y1 = 2.4829
x2 = 305
            y2 = 2.4843
x3 = 307
            y3 = 2.4871

Enter the value of xr at which y = f(x) is to be calculated xr = 301

The interpolated value of y at xr = 301.000000 is yr = 2.478597
-------------------------------------------------------------------
```

Observe that the value of y_r obtained by the program is same as that we obtained in example 5.5.1.

Important Note :

You can use this program for any type of data, there is no need to change the program. However if data points are more than 20, then you will have to increase the sizes of arrays `x[]` and `y[][]` in the program.

Exercise

1. *Using newton's divided differences interpolation find f (2). Given that*

x	1	4	5	6
y	0	1.3863	1.6094	1.7917

[Ans. : f (2) = 0.62882]

2. *Use newton's divided differences to obtain y at x = 2 from given data.*

x	1	3	4	6
y	4	7	8	11

[Ans. : 5.799998]

3. *A cromel Alumel thermocouple gives following type of output for rise in temperature.*

Temp (°C)	0	10	20	30	40	50
Output (mV)	0.0	0.4	0.8	1.2	1.61	2.02

Find the voltage output of thermocouple for temperature of 37°C using newton's divided difference formula.

[**Hint** : Refer example 5.5.4] **[Ans. : At 37° C output is 1.485 mV]**

4. *A table of a polynomial function is given below. Fit a polynomial and find the value of f(x) at x = 2.5*

x	−3	−1	0	3	5
f (x)	−30	−22	−12	330	3458

[Ans. : f(x) = −30 + 4(x + 3) + 2(x + 3)(x + 1) + 4(x + 3) (x + 1)x + 5(x + 3) (x + 1) x (x − 3) and f (2.5) = 102.7]

5. *Find the polynomial of degree three which takes the values given below.*

x_i	0	1	2	4
y_i	1	1	2	5

$$\left[\textbf{Ans.: } f(x) = \frac{1}{12}\left(-x^3 + 8x^2 - 8x + 12\right)\right]$$

University Questions

1. What is divided difference ? Define divided difference of N^{th} order. Obtain Newton's general interpolation formula with divided differences.

[Dec - 95, May - 97, May - 98, Dec- 98, Dec - 99, Dec - 2000]

2. Derive Newton's divided difference formula and use it to compute f(1.5) from the following data :

x	0	1	2	5
y(x)	2	3	12	147

[May - 2001]

3. Write a C program to get the interpolation with unevenly spaced points (use any appropriate method) **[Dec - 2001]**

4. Find $\log_{10} 301$ using Newton's divided difference formula from the following table.

x	300	304	305	307
$\log_{10} x$	2.4771	2.4829	2.4843	2.4871

[Dec - 2001]

5. Derive Newtons divided difference formula. Show that same formula works for evenly spaced points also. Write a program in C / C++ to print the divided difference table and interpolate the value of y at a particular value of x. **[Dec - 2002]**

6. Given the data f(1) = 4, f(2) = 5, f(7) = 5, f(8) = 4, find the value of x for which f(x) is maximum and find $f(x)_{max}$. (Use numerical method). **[Dec - 2002]**

7. Derive Newton's Divided difference formula. Find $\log_{10} 301$ if -

x	300	304	305	307
$\log_{10} x$	2.4771	2.4829	2.4843	2.4871

[Dec - 2003]

8. Derive Newton's Divided difference formula and use it to find f(2) if

x	1	4	6
f(x)	0	1.386294	1.791759

[May - 2004]

5.6 Lagrange's Interpolation

The most important feature of this interpolation method is that it doesnot need any differences. Lagrange's interpolation method can be applied to any type of data irrespective of the spacing between values of x.

If at the value of $x = x_r$ interpolation is required, then Lagranges interpolation formula is given as,

$$\begin{aligned} y_r = & \frac{(x_r - x_1)(x_r - x_2)\ldots(x_r - x_n)}{(x_0 - x_1)(x_0 - x_2)\ldots(x_0 - x_n)} y_0 \\ & + \frac{(x_r - x_0)(x_r - x_2)(x_r - x_3)\ldots(x_r - x_n)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)\ldots(x_1 - x_n)} y_1 \\ & + \frac{(x_r - x_0)(x_r - x_1)(x_r - x_3)\ldots(x_r - x_n)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)\ldots(x_2 - x_n)} y_2 + \ldots \\ & \ldots + \frac{(x_r - x_0)(x_r - x_1)\ldots\ldots(x_r - x_{n-1})}{(x_n - x_0)(x_n - x_1)\ldots(x_n - x_{n-1})} y_n \end{aligned} \qquad \ldots (5.6.1)$$

This formula can be written alternately as follows :

$$y_r = \sum_{k=0}^{n} y_k L_k(x) \qquad \ldots (5.6.2)$$

Where $L_k(x)$ is called Lagrange's interpolation polynomial and is given by,

$$L_k(x) = \frac{\prod\limits_{i=0, i \neq k}^{n} (x_r - x_i)}{\prod\limits_{i=0, i \neq k}^{n} (x_k - x_i)} \qquad \ldots (5.6.3)$$

Truncation error in Lagrange's Interpolation :

Let $\phi_n(x)$ be the approximated polynomial by Lagrange's interpolation. Then truncation error is given as,

$$E_n = f(x) - \phi_n(x) = \frac{(x_r - x_0)(x_r - x_1)\ldots(x_r - x_n)}{(n+1)!} = f^{(n+1)}(z)$$

& $\quad x_0 \le z \le x_n$

5.6.1 Solved Examples

Ex. 5.6.1 *For the following data find $\sqrt{1.1}$ using Lagrange's interpolation. Determine the accuracy of interpolation.*

x	1	1.2	1.3	1.4
$\sqrt{x}$	1	1.095	1.140	1.183

Sol. : Here let $\quad x_0 = 1,\ x_1 = 1.2,\ x_2 = 1.3\ \&\ x_3 = 1.4$

$\therefore \quad y_0 = 1,\ y_1 = 1.095,\ y_2 = 1.140\ \&\ y_3 = 1.183$

Let $\quad x_r = 1.1$

From the Lagrange's interpolation formula of equation 5.6.1 we have,

$$y_r = \frac{(x_r - x_1)(x_r - x_2)(x_r - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} y_0 + \frac{(x_r - x_0)(x_r - x_2)(x_r - x_3)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)} y_1 + \frac{(x_r - x_0)(x_r - x_1)(x_r - x_3)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)} y_2 + \frac{(x_r - x_0)(x_r - x_1)(x_r - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} y_3$$

Putting the values of x & y we get,

$$y_r = \frac{(1.1-1.2)(1.1-1.3)(1.1-1.4)}{(1-1.2)(1-1.3)(1-1.4)} 1 + \frac{(1.1-1)(1.1-1.3)(1.1-1.4)}{(1.2-1)(1.2-1.3)(1.2-1.4)} 1.095 + \frac{(1.1-1)(1.1-1.2)(1.1-1.4)}{(1.3-1)(1.3-1.2)(1.3-1.4)} 1.140 + \frac{(1.1-1)(1.1-1.2)(1.1-1.3)}{(1.4-1)(1.4-1.2)(1.4-1.3)} 1.183$$

$$= 0.25 + 1.6425 - 1.14 + 0.2958 = 1.0483$$

$$\therefore \quad \sqrt{1.1} = 1.0483$$

Actual value of $\sqrt{1.1}$ is 1.0488

$\therefore$ Error in interpolation $= 1.0488 - 1.0483$

$= 0.0005$

Ex. 5.6.2 *Find the polynomial of degree three which takes the values as shown below.*

x	0	1	2	4
y	1	1	2	5

Sol. : Since the data points are unequally spaced, let's use Lagrange's interpolation for this data,

From equation 5.6.1 Lagrange's interpolation formula is given as,

$$y_r = \frac{(x_r - x_1)(x_r - x_2)(x_r - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} y_0 + \frac{(x_r - x_0)(x_r - x_2)(x_r - x_3)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)} y_1$$

$$+ \frac{(x_r - x_0)(x_r - x_1)(x_r - x_3)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)} y_2 + \frac{(x_r - x_0)(x_r - x_1)(x_r - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} y_3$$

Putting the values of x and y in above equation,

$$y_r = \frac{(x_r - 1)(x_2 - 2)(x_r - 4)}{(0-1)(0-2)(0-4)}(1) + \frac{x_r (x_r - 2)(x_r - 4)}{1(1-2)(1-4)}(1)$$

$$+ \frac{x_r (x_r - 1)(x_r - 4)}{2(2-1)(2-4)} (2) + \frac{x_r (x_r - 1)(x_r - 2)}{4(4-1)(4-2)} (5)$$

$$= \frac{1}{12}\left(-x_r^3 + 9x_r^2 - 8x_r + 12\right)$$

Replacing x_r by x in the above equation

$$y = f(x) = \frac{1}{12}\left(-x^3 + 9x^2 - 8x + 12\right)$$

This is the required polynomial.

Ex. 5.6.3 *$y = x^3$ is given for x = 1, 2, ... 5. Use lagrange's formula to obtain x at y = 3.375. Compare this result with correct value, 1.5.*

Sol. : The given relation is $y = x^3$ and x = 1, 2, 3, 4, 5. Following table shows the values of x and y.

x	$y = x^3$
$x_0 = 1$	$y_0 = 1$
$x_1 = 2$	$y_1 = 8$
$x_2 = 3$	$y_2 = 27$
$x_3 = 4$	$y_3 = 64$
$x_4 = 5$	$y_4 = 125$

Here $y_r = 3.375$ and we have to find x_r. Here note that the problem is trivial, only the notations are interchanged. Hence we will use equation 5.6.1 with notations of x and y interchanged.

Such equation with interchanged x and y (of equation 5.6.1) is shown below :

$$x_r = \frac{(y_r-y_1)(y_r-y_2)(y_r-y_3)(y_r-y_4)}{(y_0-y_1)(y_0-y_2)(y_0-y_3)(y_0-y_4)}x_0$$
$$+\frac{(y_r-y_0)(y_r-y_2)(y_r-y_3)(y_r-y_4)}{(y_1-y_0)(y_1-y_2)(y_1-y_3)(y_1-y_4)}x_1$$
$$+\frac{(y_r-y_0)(y_r-y_1)(y_r-y_3)(y_r-y_4)}{(y_2-y_0)(y_2-y_1)(y_2-y_3)(y_2-y_4)}x_2$$
$$+\frac{(y_r-y_0)(y_r-y_1)(y_r-y_2)(y_r-y_4)}{(y_3-y_0)(y_3-y_1)(y_3-y_2)(y_3-y_4)}x_3$$
$$+\frac{(y_r-y_0)(y_r-y_1)(y_r-y_2)(y_r-y_3)}{(y_4-y_0)(y_4-y_1)(y_4-y_2)(y_4-y_3)}x_4$$

Now let us substitute the values of x and y in above equation i.e.

$$x_r = \frac{(3.375-8)(3.375-27)(3.375-64)(3.375-125)}{(1-8)(1-27)(1-64)(1-125)}\times 1$$
$$+\frac{(3.375-1)(3.375-27)(3.375-64)(3.375-125)}{(8-1)(8-27)(8-64)(8-125)}\times 2$$
$$+\frac{(3.375-1)(3.375-8)(3.375-64)(3.375-125)}{(27-1)(27-8)(27-64)(27-125)}\times 3$$
$$+\frac{(3.375-1)(3.375-8)(3.375-27)(3.375-125)}{(64-1)(64-8)(64-27)(64-125)}\times 4$$
$$+\frac{(3.375-1)(3.375-8)(3.375-27)(3.375-64)}{(125-1)(125-8)(125-27)(125-64)}\times 5$$

$$= 0.56666258 + 0.9495427 - 0.13564895 + 0.01585513 - 0.00090699$$

$$= 1.39550446$$

The correct value of x at y = 3.375 will be,

$$x = (y)^{1/3} = (3.375)^{1/3} = 1.5$$

The calculated value by lagrange's interpolaton is 1.39550446. Hence the error in computation is,

$$\text{Error} = |1.5 - 1.39550446| = 0.1044956$$

Ex. 5.6.4 *Obtain the missing term in the data given below using lagrange's formula.*

x	10	15	20	25	30	35
y = f(x)	43	—	29	32	—	78

Sol. : Let us rewrite the given data as follows :

x	$x_0 = 10$	$x_1 = 20$	$x_2 = 25$	$x_3 = 35$
y = f(x)	$y_0 = 43$	$y_1 = 29$	$y_2 = 32$	$y_3 = 78$

Lagrange's formula is given by equation 5.6.1 as follows :

$$y_r = \frac{(x_r - x_1)(x_r - x_2)(x_r - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} y_0 + \frac{(x_r - x_0)(x_r - x_2)(x_r - x_3)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)} y_1$$
$$+ \frac{(x_r - x_0)(x_r - x_1)(x_r - x_3)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)} y_2 + \frac{(x_r - x_0)(x_r - x_1)(x_r - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} y_3$$

Putting values of x and y in above equation

$$y_r = \frac{(x_r - 20)(x_r - 25)(x_r - 35)}{(10-20)(10-25)(10-35)} \times 43 + \frac{(x_r - 10)(x_r - 25)(x_r - 35)}{(20-10)(20-25)(20-35)} \times 29$$
$$+ \frac{(x_r - 10)(x_r - 20)(x_r - 35)}{(25-10)(25-20)(25-35)} \times 32 + \frac{(x_r - 10)(x_r - 20)(x_r - 25)}{(35-10)(35-20)(35-25)} \times 78$$

$$\therefore \quad y_r = -0.0114666\,(x_r - 20)(x_r - 25)(x_r - 35)$$
$$+ 0.0386666\,(x_r - 10)(x_r - 25)(x_r - 35)$$
$$- 0.0426666\,(x_r - 10)(x_r - 20)(x_r - 35)$$
$$+ 0.0208\,(x_r - 10)(x_r - 20)(x_r - 25) \qquad \ldots (5.6.4)$$

This is the polynomial to fit in the given data. Now missing values of $y = f(x)$ at 15 and 30 can be obtained by putting $x_r = 15$ and 30 in above equation. Putting $x_r = 15$ we get,

$$y_{15} = f(15) = y_r\big|_{x_r=15}$$
$$\therefore \quad y_r = -0.0114666\,(15-20)(15-25)(15-35)$$
$$+ 0.0386666\,(15-10)(15-25)(15-35)$$
$$- 0.0426666\,(15-10)(15-20)(15-35)$$
$$+ 0.0208\,(15-10)(15-20)(15-25)$$
$$= 34$$

Thus the missing $f(x)$ corresponding to $x = 15$ is 34. Now let us put $x_r = 30$ in equation 5.6.4 to obtain the second missing term i.e.

$$y_{30} = f(30) = y_r\big|_{x_r=30}$$
$$\therefore \quad y_{30} = -0.0114666\,(30-20)(30-25)(30-35)$$
$$+ 0.038666\,(30-10)(30-25)(30-35)$$
$$- 0.0426666\,(30-10)(30-20)(30-35)$$
$$+ 0.0208\,(30-10)(30-20)(30-25)$$
$$= 47$$

Thus missing $f(x) = 47$ corresponding to $x = 30$.

Ex. 5.6.5 *Given the following table of values :*

x	0.4	0.5	0.7	0.8
y = f(x)	− 0.916	− 0.693	− 0.357	− 0.223

Estimate the value of f(0.6) by Lagrange's interpolation, what is the order of polynomial equation?

Sol. : Let us rewrite the given table as follows :

x	$x_0 = 0.4$	$x_1 = 0.5$	$x_2 = 0.7$	$x_3 = 0.8$
y = f(x)	$y_0 = -0.916$	$y_1 = -0.693$	$y_2 = -0.357$	$y_3 = -0.223$

Lagrange's interpolation formula is given by equation 5.6.1 as follows :

$$y_r = \frac{(x_r - x_1)(x_r - x_2)(x_r - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} y_0 + \frac{(x_r - x_0)(x_r - x_2)(x_r - x_3)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)} y_1$$

$$+ \frac{(x_r - x_0)(x_r - x_1)(x_r - x_3)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)} y_2 + \frac{(x_r - x_0)(x_r - x_1)(x_r - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} y_3$$

Putting the values in above equation with $x_r = 0.6$

$$y_r = \frac{(0.6-0.5)(0.6-0.7)(0.6-0.8)}{(0.4-0.5)(0.4-0.7)(0.4-0.8)}(-0.916)$$

$$+ \frac{(0.6-0.4)(0.6-0.7)(0.6-0.8)}{(0.5-0.4)(0.5-0.7)(0.5-0.8)}(-0.693)$$

$$+ \frac{(0.6-0.4)(0.6-0.5)(0.6-0.8)}{(0.7-0.4)(0.7-0.5)(0.7-0.8)}(-0.357)$$

$$+ \frac{(0.6-0.4)(0.6-0.5)(0.6-0.7)}{(0.8-0.4)(0.8-0.5)(0.8-0.7)}(-0.223)$$

$$= 0.15266667 - 0.462 - 0.238 + 0.03716667$$

$$= -0.51016667$$

Thus $f(0.6) = -0.51016667$. The order of the polynomial equation for y_r will be three.

Ex. 5.6.6 *Using numerical method, express the rational function* $\dfrac{3x^2 + x + 1}{(x-1)(x-2)(x-3)}$ *as a sum of partial fractions.* **[Dec - 2001, 5 marks; Dec - 2002, 6 Marks]**

Sol. : Lagrange's interpolation can be used to express a rational function as a sum of partial fractions. The denominator of the function is $(x-1)(x-2)(x-3)$. Hence evaluate numerator for $x=1$, $x=2$ and $x=3$. These values are given below.

x	$x_0 = 1$	$x_1 = 2$	$x_2 = 3$
$y = 3x^2 + x + 1$ (numerator)	$y_0 = 5$	$y_1 = 15$	$y_2 = 31$

Lagrange's interpolation formula is given by equation 5.6.1. For $n = 2$ we can write,

$$y = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} y_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} y_1 + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} y_2$$

Putting the values in above equation,

$$y = \frac{(x-2)(x-3)}{(1-2)(1-3)}(5) + \frac{(x-1)(x-3)}{(2-1)(2-3)}(15) + \frac{(x-1)(x-2)}{(3-1)\ (3-2)}(31)$$

$$= \frac{5}{2}(x-2)(x-3) - 15(x-1)(x-3) + \frac{31}{2}(x-1)(x-2)$$

Thus we obtained the numerator as,

$$3x^2 + x + 1 = \frac{5}{2}(x-2)(x-3) - 15(x-1)(x-3) + \frac{31}{2}(x-1)(x-2)$$

Putting this value of numerator in the given function we get,

$$\frac{3x^2+x+1}{(x-1)(x-2)(x-3)} = \frac{\frac{5}{2}(x-2)(x-3) - 15(x-1)(x-3) + \frac{31}{2}(x-1)(x-2)}{(x-1)(x-2)(x-3)}$$

$$= \frac{\frac{5}{2}}{(x-1)} - \frac{15}{(x-2)} + \frac{\frac{31}{2}}{(x-3)} = \frac{5}{2(x-1)} - \frac{15}{x-2} + \frac{31}{2(x-3)}$$

Ex. 5.6.7 *Using lagrange's interpolation formula, express the function* $\frac{6x^2-23x+19}{x^3-3x^2+9x-6}$ *as a sum of partial fractions.* **[May - 2000, 8 marks]**

Sol. : Consider the denominator $x^3 - 3x^2 + 9x - 6$.

We can write this denominator as,

$$\left(x^3 - 3x^2 + 9x - 6\right) = (x - 0.8341)\,[x - (1.0830 + j\ 2.4537)]\,[x - (1.0830 - j\,2.4537)] \quad \ldots (5.6.5)$$

Now let us evaluate the numerator at $x = 0.8341$, $x = 1.0830 + j\,2.4537$ and $x = 1.0830 - j\ 2.4537$.

x	$y = 6x^2 - 23x + 19$
$x_0 = 0.8341$	$y_0 = 3.99$
$x_1 = 1.0830 + j\,2.4537$	$y_1 = -34.9955 - j\,24.5468$
$x_2 = 1.0830 - j\,2.4537$	$y_2 = -34.9955 + j\,24.5468$

For $n = 2$, lagranges formula becomes (see equation 5.6.1),

$$y = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} y_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} y_1 + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} y_2$$

Putting values in the above equation,

$$= \frac{[x-(1.0830+j\ 2.4537)][x-(1.0830-j\,2.4537)]}{(0.8341-1.0830-j\,2.4537)(0.8341-1.0830+j\ 2.4537)} (3.99)$$

$$+\frac{(x-0.8341)[x-(1.0830-j\ 2.4537)]}{(1.0830+j\ 2.4537-0.8341)(1.0830+j\,2.4537-1.0830+j\ 2.4537)}(-34.9955-j\ 24.5468)$$

$$+\frac{(x-0.8341)[x-(1.0830+j\,2.4537)]}{(1.0830-j\,2.4537-0.8341)(1.0830-j\,2.4537-1.0830-j\,2.4537)}(-34.9955+j\,24.5468)$$

$$= 0.65597\ [x-(1.0830+j\,2.4537)][x-(1.0830-j\,2.4537)]$$
$$+ (2.672+j\ 2.3096)(x-0.8341)[x-(1.0830-j\ 2.4537)]$$
$$+ (2.6720-j\ 2.3096)(x-0.8341)[x-(1.0830+j\ 2.4537)] \quad \ldots (5.6.6)$$

Above equation represents numerator. Equation 5.6.5 represents denominator. Putting this expressions, we get following equation after simplification.

$$\frac{6x^2-23\,x+19}{x^3-3\,x^2+9x-6} = \frac{0.65597}{(x-0.8341)} + \frac{2.6720+j\ 2.3096}{x-(1.0830+j\ 2.4537)} + \frac{2.6720-j\,2.3096}{x-(1.0830-j\,2.4537)}$$

This is the partial fraction expansion of the given function.

Ex. 5.6.8 *Using lagrange's formula find a polynomial that satisfies* $p(1)=1$, $p(3)=27$ *and* $p(4)=64$ *and hence evaluate* $p(2)$.

[May - 2001, 6 marks, Dec - 98, 8 marks, Dec - 95, 8 marks]

Sol. : Let us rewrite the given data as follows :

x	$x_0 = 1$	$x_1 = 3$	$x_2 = 4$
$y = f(x) = p(x)$	$y_0 = 1$	$y_1 = 27$	$y_2 = 64$

Here $n = 3$. Lagranges interpolation formula is given by equation 5.6.1 as,

$$y_r = \frac{(x_r-x_1)(x_r-x_2)}{(x_0-x_1)(x_0-x_2)} y_0 + \frac{(x_r-x_0)(x_r-x_2)}{(x_1-x_0)(x_1-x_2)} y_1 + \frac{(x_r-x_0)(x_r-x_1)}{(x_2-x_0)(x_2-x_1)} y_2$$

Putting values in above equation,

$$y_r = \frac{(x_r-3)(x_r-4)}{(1-3)(1-4)}(1) + \frac{(x_r-1)(x_r-4)}{(3-1)(3-4)}(27) + \frac{(x_r-1)(x_r-3)}{(4-1)(4-3)}(64)$$

Simplifying the above equation we get,

$$y_r = 8\ x_r^2 - 19\ x_r + 12$$

This is the required quadratic polynomial passing through given points. To obtain $p(2)$, we have to put $x_r = 2$ in above equation. i.e.,

$$y_r = p(2) = 8\,(2)^2 - 19\,(2) + 12 = 6$$

Ex. 5.6.9 *Find the missing term in the table.*

x	0	1	2	3	4
y	1	3	9	–	81

[Dec - 2001, 8 marks]

Sol. : Let us rewrite the given table as follows :

x	$x_0 = 0$	$x_1 = 1$	$x_2 = 2$	$x_3 = 4$
y	$y_0 = 1$	$y_1 = 3$	$y_2 = 9$	$y_3 = 81$

Since the given data is not evenly spaced, we will use lagrange's interpolation formula. For $n = 3$, this formula is given as,

$$y_r = \frac{(x_r - x_1)(x_r - x_2)(x_r - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} y_0 + \frac{(x_r - x_0)(x_r - x_2)(x_r - x_3)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)} y_1 + \frac{(x_r - x_0)(x_r - x_1)(x_r - x_3)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)} y_2 + \frac{(x_r - x_0)(x_r - x_1)(x_r - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} y_3$$

Here $x_r = 3$ and putting other values in above equation we get,

$$y_r = \frac{(3-1)(3-2)(3-4)}{(0-1)(0-2)(0-4)}(1) + \frac{(3-0)(3-2)(3-4)}{(1-0)(1-2)(1-4)}(3) + \frac{(3-0)(3-1)(3-4)}{(2-0)(2-1)(2-4)}(9) + \frac{(3-0)(3-1)(3-2)}{(4-0)(4-1)(4-2)}(81)$$

$$= \frac{1}{4} - 3 + \frac{27}{2} + \frac{81}{4} = 31$$

Thus at $x = 3$, $y = 31$.

5.6.2 C Program and Algorithm

Now we will prepare an algorithm for Lagrange's interpolation.

Algorithm :

Step 1 : ***Read total number of data points and read actual values of these data points. i.e. x & y = f (x).***

Step 2 : ***Read the value of $x = x_r$ at which value of y is to be calculated.***

Step 3 : ***If there are (n + 1) data points, then calculate,***

$$L_k(x) = \frac{\overset{n}{\underset{i=0,\, i\neq k}{P}} \quad (x_r - x_i)}{\overset{n}{\underset{i=0,\, i\neq k}{P}} \quad (x_k - x_i)}$$

then calculate,

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

S. *The interpolated value at* $\mathbf{x} = \mathbf{x}_r$ *is equal to* $\mathbf{y}_r$. *i.e.* $f(x_r) = y_r$.

St *Display values of* x_r *and* $f(x_r)$ *on the screen and stop.*

owchart :

Based on the algorithm above now we will prepare a flowchart for Lagrange's olation method. This flowchart is shown in Fig. 5.6.1.

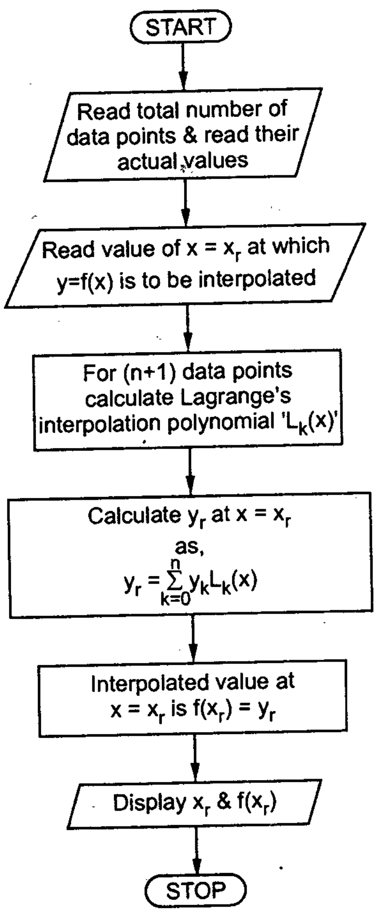


Fig. 5.6.1 Flowchart of Lagranges interpolation method

Computer Program :

A 'C' program for Lagrange's interpolation is shown below.

```
/* Download this program from  www.vtubooks.com                                  */
/* File name : lagrangs.cpp                                                       */

/*----------------- LAGRANGE'S INTERPOLATION  METHOD --------------------*/

/*      THE PROGRAM CALCULATES THE VALUE OF f(x) AT GIVEN VALUE OF x
        USING LAGRANGE'S INTERPOLATION METHOD.

        INPUTS :  1) Number of entries of the data.

                  2) Values of 'x' & corresponding y = f(x).

                  3) Value of 'xr' at which y = f(x) to be calculated.

        OUTPUTS : Interpolated value f(x) at x = xr.                             */

/*-----------------------------    PROGRAM  ----------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>

void main()
{
      double y[20],x[20],xr,fy,num,den;
      int i,j,n;

      clrscr();
      printf("\n              LAGRANGE'S INTERPOLATION TECHNIQUE");

      printf("\n\nEnter the number of entries (max 20) = ");
                                 /* ENTER THE NUMBER OF ENTRIES    */
      scanf("%d",&n);

      for(i = 0; i < n; i++)
      {                        /*  LOOP TO GET x AND y = f(x) IN THE ARRAY */

          printf("x%d = ",i);  scanf("%lf",&x[i]);
          printf("           y%d = ",i);  scanf("%lf",&y[i]);
      }

      printf("\nEnter the value of xr at which  y = f(x)\n\t\t\t"
             "is to be interpolated, xr  = ");
      scanf("%lf",&xr);

      fy = 0;
      for(j = 0; j < n; j++)
      {           /*  LOOP TO CALCULATE LAGRANGE'S INTERPOLATION  */
          num = den = 1;
          for(i = 0; i < n; i++)
          {
             if(i == j) continue;
             num = num * (xr - x[i]);
             den = den * (x[j] - x[i]);
          }
          fy = fy + ((num/den) * y[j]);
      }

      printf("\nThe value of y = f(x) at xr = %lf is yr = %lf", xr,fy);

}
/*-------------------------- END OF PROGRAM ---------------------------*/
```

The initial part of the program is similar to previous programs we discussed. First the program declares variables. Observe in the first statement that `y[20]` is declared as one dimensional array since there is no differences table. Then the program prints the name of the method. After this there is a `printf` and `scanf` statement to get number of data points. Next there is a `for` loop to get values of x and y in the arrays. The next `printf` and `scanf` statement asks for value of $x = x_r$ at which y is to be interpolated.

The next `for` loop implements lagrange's interpolation formula. This loop is reproduced below for convenience.

```
for  (j=0;  j<n,  j++)  ← This loop is for counting terms in the formula.
{
        num=den=1
        for(i=0;  i<n;  i++)← This loop is for calculating value of the term.
        {
            if  (i==j)  continue;  ← This statement avoids calculation of
                    x1 - x1, x2 - x2,... etc.
            num=num*(xr-x[i]);← Calculation of numerator in the term i.e.
                    (xr - x0)(xr - x1)(xr - x2) etc.
            den=den*(x[j]-x[i]);←Calculation of denominator in the term
                    i.e., (x1 - x0)(x1 - x2)(x1 - x3) etc.
        }
        fy=fy+((num/den)*y[j]);                  ← Summation of all the terms.
}
```

We have indicated the function of each statement in this `for` loop. The outer `for` loop is used as a counter for individual term of summation in the interpolation formula. The inner `for` loop calculates value of the particular term. The first statement in this loop is,

`if (i==j) continue;`

We don't want to calculate $x_1 - x_1$; $x_2 - x_2$, ... etc. Therefore when `i=j`; because of `continue` statement program comes out of this loop. The next statement calculates numerator of the term.

i.e.,num= num*(xr-x[i]);

When i = 0 it calculates $(x_r - x_0)$

When i = 1 it calculates $(x_r - x_0)(x_r - x_1)$ ... & so on.

The next statement calculates denominator

i.e.,

`den = den*(x[j]-x[i]);`

When j = 0 and i = 0 program comes out of loop.

When j = 0 and i = 1 it calculates $(x_0 - x_1)$

When j = 0 and i = 2 it calculates $(x_0 - x_1)(x_0 - x_2)$ . . . and so on.

The program then comes out of the inner loop.

The next statement is,

```
fy = fy+((num/den)*y[j]);
```

This statement multiplies ratio of numerator to denominator by value of y [ ] and adds the result to previous sum. The program then comes out of the loop and `printf` statement prints x_r and y_r (`fy`) on the screen.

How to Run this program?

Compile the source code of the program and make EXE file. For the illustration of this method we will use data points of example 5.6.1. Those data points are reproduced below for convenience.

x	$x_0 = 1$	$x_1 = 1.2$	$x_2 = 1.3$	$x_3 = 1.4$
$y = \sqrt{x}$	$y_0 = 1$	$y_1 = 1.095$	$y_2 = 1.140$	$y_3 = 1.183$

Here $\sqrt{1.1}$ is required means x_r = 1.1.

Run the EXE file of the program on your computer. The program first displays the name of the method, then it displays,

`Enter the number of entries (max 20) =` Here enter the number of entries as '4' and press 'enter' key.

Then it displays,

x0 = Here enter `1' and press `enter' key.

y0 = Here enter `1' and press `enter'key

```
x1 = 1.2 ↵
          y1 = 1.095 ↵
x2 = 1.3 ↵
          y2 = 1.140 ↵
x3 = 1.4 ↵
          y3 = 1.183 ↵
```

Here symbol `↵' means press 'enter' key then the program displays,

`Enter the value of xr at which y=f(x) is to be interpolated, xr =`

Here enter 1.1. and press 'enter' key.

The pogram then displays y_r as 1.048250.

Here is the complete display of result

```
---------------------------- Results ------------------------------

        LAGRANGE'S INTERPOLATION TECHNIQUE

Enter the number of entries (max 20) = 4
x0 = 1
            y0 = 1
x1 = 1.2
            y1 = 1.095
x2 = 1.3
            y2 = 1.140
x3 = 1.4
            y3 = 1.183

Enter the value of xr at which  y = f(x)
                          is to be interpolated, xr  = 1.1

The value of y = f(x) at xr = 1.100000 is yr = 1.048250
---------------------------------------------------------------------
```

Note : This program will work for any type of data.

Exercise

1. Use Lagrangian interpolation technique to find the value of t at x = 2.

x	1	4	6	5
y	0	1.3863	1.7918	1.609

[Ans. : f (2) = 0.62928] [Dec-91]

2. Find the unique polynomial of degree 2 or less, such that f (0) = 1, f (1) = 3, f (3) = 55 using Lagrange's interpolation. **[Ans. : $8x^2 - 6x + 1$]**

3. Use Lagrange's formula to produce a cubic polynomial which includes the following values of x_i , y_i. Then evaluate this polynomial for x = 2, 3, 5.

x_i	0	1	4	6
y_i	1	− 1	1	− 1

$$\textbf{[Ans. : } f(x) = \frac{(x-1)(x-4)(x-6)}{-24} - \frac{x(x-4)(x-6)}{15} + \frac{x(x-1)(x-6)}{-24} - \frac{x(x-1)(x-4)}{60}$$

And f (2) = − 1, f (3) = 0 and f (5) = 1]

University Questions

1. Using Lagrange's formula, find a unique polynomial P(x) of degree 2 or less such that

   P(1) = 1; P(3) = 27; P(4) = 64

   and hence evaluate P(2). **[Dec - 95, May - 2001]**

2. Find the interpolating polynomial for the data -

x	0	1	2	5
f(x)	2	3	12	147

[May - 96, May - 98, Dec - 99]

3. From the data find y at 4.5

x	1	2	3	4	5
y	2.38	3.65	5.85	9.95	14.85

[May - 96]

4. If $y(1) = -3$, $y(3) = 9$, $y(4) = 30$ and $y(6) = 132$ find the four-point lagrange interpolation polynomial that takes the same values as the function y at the given points. [May - 97]
5. Using Lagranges Interpolation formula, express the function.
   $\frac{6x^2 - 23x + 19}{x^3 - 3x^2 + 9x - 6}$ as the sum of partial functions [May - 2000]
6. Using Numerical method, express the rational function $\frac{3x^2 + x + 1}{(x-1)(x-2)(x-3)}$ as a sum of partial fractions. [Dec - 2001]
7. Find the missing term in the table.

x	0	1	2	3	4
y(x)	1	3	9	–	81

[Dec - 2001]

8. Using Numerical method, express the function as the sum of partial fractions.
   $\frac{3x^2+x+1}{(x-1)(x-2)(x-3)}$ [Dec - 2002]
9. Derive Language's Interpolation formula and hence find the cubic polynomial that takes f(0) = 2, f(1) = 3, f(2) = 12 and f(5) = 147 [May - 2003]
10. Write a program for Lagranges Interpolation formula. [Dec - 2003]
11. Write a program on C/C ++ interpolate a polynomial of degree 'n' using Lagranges interpolation. [May - 2004]
12. Write a C/C ++ program to interpolate a polynomial of degree 'n' using Lagranges interpolation. Express the function
    $\frac{x^2+x-3}{x^3-2x^2-x+2}$ as sum of partial fractions
    Using Lagranges method. [Dec - 2004]

5.7 Inverse Interpolation

In the interpolation problem, we are given the values of x and $y = f(x)$. Then we have to calculate y for given value of x. This can be easily calculated by any interpolation method discussed in previous sections. When we are given the value of y and we have to find corresponding x, then it is called inverse interpolation. Following example illustrates inverse interpolation.

Ex. 5.7.1 *An experiment carried out given the temperatures at time t as follows :*

time t	0	1	2	3	4	5	6	7	8
temperature °C	60.0	64.5	72.5	80.0	86.25	92.5	105.0	111.0	118.25

Using the above table evaluate temperature at time 3.5, 8.5 and time at temperature of 100°C. **[May-99, 8 marks, Dec-96, 8 marks]**

Sol. : The temperature at $t = 3.5$ can be obtained using newton's forward differences interpolation since values of x are evenly spaced. Similarly temperature at $t = 8.5$ can be obtained using newton's backward differences interpolation. The difference table for the given data can be formed as follows :

Please refer Table 5.7.1 on next page.

To obtain temperature at $t = 3.5$

Newton's forward differences interpolation formula is given by equation 5.4.4 as,

$$y_r = y_0 + r\,\Delta y_0 + \frac{r\,(r-1)}{2!}\,\Delta^2 y_0 + \frac{r\,(r-1)\,(r-2)}{3!}\,\Delta^3 y_0 + \ldots\ldots$$

From equation 5.4.1, $x_r = x_0 + r\,h$

$$\therefore \qquad r = \frac{x_r - x_0}{h}$$

Here $x_r = t = 35$, $\quad x_0 = 0$ and $h = x_1 - x_0 = 1 - 0 = 1$

$$\therefore \qquad r = \frac{3.5 - 0}{1} = 3.5$$

Putting there values and forward differences in newton's interpolation formula we get,

$$y_r = 60 + 3.5\,(4.5) + \frac{3.5\,(3.5-1)}{2\,!}\,(3.5) + \frac{3.5\,(3.5-1)\,(3.5-2)}{3\,!}\,(-4)$$

$$+ \frac{3.5\,(3.5-1)\,(3.5-2)\,(3.5-3)}{4\,!}\,(3.25)$$

$$+ \frac{3.5\,(3.5-1)\,(3.5-2)\,(3.5-3)\,(3.5-4)}{5\,!}\,(-\ 1.25)$$

$$+ \frac{3.5\,(3.5-1)\,(3.5-2)\,(3.5-3)\,(3.5-4)\,(3.5-5)}{6\,!}\,(4.25)$$

$$+ \frac{3.5\,(3.5-1)\,(3.5-2)\,(3.5-3)\,(3.5-4)\,(3.5-5)\,(3.5-6)}{7\,!}\,(-\ 31.25)$$

$$+ \frac{3.5\,(3.5-1)\,(3.5-2)\,(3.5-3)\,(3.5-4)\,(3.5-5)\,(3.5-6)\,(3.5-7)}{8\,!}\,(121.75)$$

Solving the above equation we get,

$$y_r = 83.638969$$

Forward/Backward difference table for example 5.7.1

x	y = temp	Δy or ∇y	$\Delta^2 y$ or $\nabla^2 y$	$\Delta^3 y$ or $\nabla^3 y$	$\Delta^4 y$ or $\nabla^4 y$	$\Delta^5 y$ or $\nabla^5 y$	$\Delta^6 y$ or $\nabla^6 y$	$\Delta^7 y$ or $\nabla^7 y$	$\Delta^8 y$ or $\nabla^8 y$
$x_0 = 0$	$y_0 = 60.0$	$\Delta y_0 = 4.5$							
$x_1 = 1$	$y_1 = 64.5$		$\Delta^2 y_0 = 3.5$						
		8.0		$\Delta^3 y_0 = -4$					
$x_2 = 2$	$y_2 = 72.5$		-0.5		$\Delta^4 y_0 = 3.25$				
		7.5		-0.75		$\Delta^5 y_0 = -1.25$			
$x_3 = 3$	$y_3 = 80.0$		-1.25		2		$\Delta^6 y_0 = 4.25$		
		6.25		1.25		3		$\Delta^7 y_0 = -31.25$	
$x_4 = 4$	$y_4 = 86.25$		0		5		-27		$\Delta^8 y_0 = \nabla^8 y_0 = 121.75$
		6.25		6.25		-24		$\nabla^7 y_8 = 90.5$	
$x_5 = 5$	$y_5 = 92.5$		6.25		-19		$\nabla^6 y_8 = 63.5$		
		12.5		-12.75		$\nabla^5 y_8 = 39.5$			
$x_6 = 6$	$y_6 = 105.0$		-6.5		$\nabla^4 y_8 = 20.5$				
		6		$\nabla^3 y_8 = 7.75$					
$x_7 = 7$	$y_7 = 111.0$		$\nabla^2 y_8 = 1.25$						
		$\nabla y_8 = 7.25$							
$x_8 = 8$	$y_8 = 118.25$								

Thus at $t = 3.5$, temperature = 83.638969 °C

To obtain temperature at $t = 8.5$

This can be obtained using newtons backward difference interpolation formula. From equation 5.4.12 we have,

$$r = \frac{x_r - x_n}{h}$$

Here $x_r = 8.5$, $x_n = x_8 = 8$ and $h = x_1 - x_0 = 1 - 0 = 1$

$$\therefore \quad r = \frac{8.5 - 8}{1} = 0.5$$

Newton's backward differences interpolation formula is given by equation 5.4.10 as,

$$y_r = y_n + r \nabla y_n + \frac{r(r+1)}{2!} \nabla^2 y_n + \frac{r(r+1)(r+2)}{3!} \nabla^3 y_n + \ldots\ldots$$

Putting values in above equation,

$$y_r = 118.25 + 0.5\,(7.25) + \frac{0.5\,(0.5+1)}{2\,!}(1.25) + \frac{0.5\,(0.5+1)\,(0.5+2)}{3\,!}(7.75)$$

$$+ \frac{0.5\,(0.5+1)\,(0.5+2)\,(0.5+3)}{4\,!}(20.5)$$

$$+ \frac{0.5\,(0.5+1)\,(0.5+2)\,(0.5+3)\,(0.5+4)}{5\,!}(39.5)$$

$$+ \frac{0.5\,(0.5+1)\,(0.5+2)\,(0.5+3)\,(0.5+4)\,(0.5+5)}{6\,!}(63.5)$$

$$+ \frac{0.5\,(0.5+1)\,(0.5+2)\,(0.5+3)\,(0.5+4)\,(0.5+5)\,(0.5+6)}{7\,!}(90.5)$$

$$+ \frac{0.5\,(0.5+1)\,(0.5+2)\,(0.5+3)\,(0.5+4)\,(0.5+5)\,(0.5+6)\,(0.5+7)}{8\,!}(121.75)$$

$$= 197.283119$$

That at $t = 8.5$, temperature = 197.283119°C

To obtain time at temperature 100°C (Inverse interpolation)

The temperature of 100°C comes near $x_5 = 5$. Hence let us consider following part of the data,

x	$x_0 = 5$	$x_1 = 6$	$x_2 = 7$
y	$y_0 = 925$	$y_1 = 105$	$y_2 = 111$

Here we have renamed data points for the sake of convenience. Let us form the forward difference table of above data :

x	y	Δy	$\Delta^2 y$
$x_0 = 5$	$y_0 = 925$		
		$\Delta y_0 = 12.5$	
$x_1 = 6$	$y_1 = 105$		$\Delta^2 y_0 = -6.5$
		6	
$x_2 = 7$	$y_1 = 111$		

We know that $h = x_1 - x_0 = 6 - 5 = 1$. From equation 5.4.4 newton's forward difference formula is given as,

$$y_r = y_0 + r\,\Delta y_0 + \frac{r(r-1)}{2\,!}\,\Delta^2 y_0 + \frac{r(r-1)(r-2)}{3!}\,\Delta^3 y_0 + \ldots\ldots$$

Putting values in above equation,

$$y_r = 92.5 + r\,(12.5) + \frac{r(r-1)}{2!}\,(-\,6.5)$$

$$= 92.5 + 19\,r - 3.25\,r^2 \qquad \ldots (5.7.1)$$

We know that $x_r = x_0 + r\,h$

$$\therefore \qquad \mathrm{r} = \frac{x_r - x_0}{h}$$

Here $x_0 = 5$ and $h = x_1 - x_0 = 6 - 5 = 1$. Hence,

$$r = \frac{x_r - 5}{1} = x_r - 5$$

Putting this value of r in equation 5.7.1,

$$y_r = 92.5 + 19\,(x_r - 5) - 3.25\,(x_r - 5)^2$$

$$= -\,3.25\,x_r^2 + 51.5\,x_r - 83.75$$

Here x_r is basically time and y_r is temperature. Hence putting $y_r = 100$ in above equation,

$$100 = -\,3.25\,x_r^2 + 51.5\,x_r - 83.75$$

i.e. $x_r^2 - 15.8461\,x_r + 56.538 = 0$

Solving this equation we get two roots :

$x_r = 10.420347$ and 5.4257534

By looking at the data, we can conclude that proper value of $x_r = 5.4247534$ at $y_r = 100$. Thus,

At 100°C, time $t = 5.4257534$.

More accuracy in the value of t can be obtained by taking more data points and more terms in equation 5.7.1.

Alternative method :

An alternative method is, we can *exchange* x and y variables. Since values of y (i.e. temperature) are not evenly spaced, we can use either lagrange's interpolation or divided differences interpolation.

University Questions

1. An experiment carried out gives the readings of temperature°C at time t as follows :

Time t	0	1	2	3	4	5	6	7	8
Temp °C	60.0	64.5	72.5	80.0	86.25	92.5	105.0	111.0	118.25

Using the above table, evaluate :

i) temperature at time = 3.5

ii) temperature at time = 8.5

iii) time t when the temperature = 100°C **[Dec - 96, May - 99]**

5.8 Least Squares Approximation

In the last section we studied cubic splines for polynomial approximation. Here we will study the least squares method to find a polynomial which can approximate all the given data points. The polynomial p (x) which is selected as an approximation to the function $f(x_i)$ [$f(x_i)$ = discrete data function] in such a way that it minimizes the squares of the errors. Hence the name least squares approximation is given.

5.8.1 Linear Regression (Fitting of a Straight Line)

The equation of a straight line is given as,

$$y = a + bx \qquad \text{... (5.8.1)}$$

This line can be approximated for the set of data points. Then we need to find values of 'a' and 'b'. The set of two equations called normal equations are formed as shown below.

$$\boxed{\sum_{i=1}^{n} y_i = na + b \sum_{i=1}^{n} x_i} \qquad \text{... (5.8.2)}$$

&

$$\boxed{\sum_{i=1}^{n} x_i y_i = a \sum_{i=1}^{n} x_i + b \sum_{i=1}^{n} x_i^2} \qquad \text{... (5.8.3)}$$

The above two equations are solved for a and b.

5.8.2 Polynomial Regression (Fitting a Second Degree Parabola)

We know that error involved in straight line approximation is much higher, hence we use polynomials of higher degrees to achieve the smoothness and better approximation. Let,

$$y = a + bx + cx^2 \qquad \text{... (5.8.3(a))}$$

be the second degree polynomial (also called as second degree parabola) used for approximation. We have to calculate values of a, b, and c in the above equation. The set of equations, called normal equations are formed as follows to find a, b and c, i.e.,

$$\sum_{i=1}^{n} y_i = n\,a + b \sum_{i=1}^{n} x_i + c \sum_{i=1}^{n} x_i^2$$

$$\sum_{i=1}^{n} x_i\, y_i = a \sum_{i=1}^{n} x_i + b \sum_{i=1}^{n} x_i^2 + c \sum_{i=1}^{n} x_i^3$$

& $$\sum_{i=1}^{n} x_i^2\, y_i = a \sum_{i=1}^{n} x_i^2 + b \sum_{i=1}^{n} x_i^3 + c \sum_{i=1}^{n} x_i^4 \qquad \text{... (5.8.4)}$$

Here $(x_i\,, y_i)$, $i = 1, 2, \ldots. n$ are data points.

The values of x_i and y_i are known. After putting those values, we get three equations in three unknowns. On solving those equations we get values of unknowns a, b, and c. On putting values of a, b, and c in equation 5.8.3 we get approximated polynomial for $(x_i\,, y_i)$ data points.

5.8.3 Solved Examples

Ex. 5.8.1 *Find the equation of a straight line which approximates the following data.*

x_i	1	2	3	4	6	8
$y_i = f(x_i)$	2	3	4	4	5	6

Sol. : Here we have 6 data points i.e. n = 6. Let's prepare the following table.

i	x_i	y_i	x_i^2	$x_i\, y_i$
1	1	2	1	2
2	2	3	4	6
3	3	4	9	12
4	4	4	16	16
5	6	5	36	30
6	8	6	64	48
	$\sum_{i=1}^{6} x_i = 24$	$\sum_{i=1}^{6} y_i = 24$	$\sum_{i=1}^{6} x_i^2 = 130$	$\sum_{i=1}^{6} x_i\, y_i = 114$

From equation 5.8.2 we have,

$$\sum_{i=1}^{n} y_i = n\,a + b \sum_{i=1}^{n} x_i$$

Putting values of $\sum x_i$, $\sum y_i$ & n from the table,

$$24 = 6a + 24b$$

or $$4 = a + 4b \qquad \text{... (5.8.5)}$$

From equation 5.8.3 we have,

$$\sum_{i=1}^{n} x_i\, y_i = a \sum_{i=1}^{n} x_i + b \sum_{i=1}^{n} x_i^2$$

Putting the values of $\sum x_i y_i$, $\sum x_i$ and $\sum x_i^2$ from table,

$$114 = 24a + 130b \qquad \text{... (5.8.6)}$$

Solving equation 5.8.5 and equation 5.8.6 for a and b we get,

$$a = 1.8824 \qquad \& \qquad b = 0.5294$$

The straight line approximation is given,

From equation 5.8.1 as,

$$y = a + bx$$

Putting the values of a and b,

$$y = 1.8824 + 0.5294\,x$$

This is the required straight line fit for given data.

***Important Note :** Least squares approximation works with any type of data.*

Ex. 5.8.2 *Find the least squares approximation of second degree for the discrete data given below.*

x_i	−2	−1	0	1	2
$y_i = f(x_i)$	15	1	1	3	19

Sol. : Here there are 5 data points, Hence n = 5.

Since second degree polynomial is asked we should use equation 5.8.3 (a) for approximation.

i.e. $$y = a + bx + cx^2 \qquad \text{... (5.8.7)}$$

Let's prepare the following table.

x_i	y_i	x_i^2	x_i^3	x_i^4	$x_i\, y_i$	$x_i^2\, y_i$
−2	15	4	−8	16	−30	60
−1	1	1	−1	1	−1	1
0	1	0	0	0	0	0
1	3	1	1	1	3	3
2	19	4	8	16	38	76
$\sum_{i=1}^{5} x_i = 0$	$\sum_{i=1}^{5} y_i = 39$	$\sum_{i=1}^{5} x_i^2 = 10$	$\sum_{i=1}^{5} x_i^3 = 0$	$\sum_{i=1}^{5} x_i^4 = 34$	$\sum_{i=1}^{5} x_i\, y_i = 10$	$\sum_{i=1}^{5} x_i^2\, y_i = 140$

The normal equations are given for second degree polynomial by equation 5.8.4. i.e.

$$\sum_{i=1}^{n} y_i = na + b\sum_{i=1}^{n} x_i + c\sum_{i=1}^{n} x_i^2$$

Putting the values from table,

$$39 = 5a + 0 + 10c$$

i.e. $5a+10c = 39$ ... (5.8.8)

The second equation in equation 5.8.4. is,

$$\sum_{i=1}^{n} x_i\, y_i = a \sum_{i=1}^{n} x_i + b \sum_{i=1}^{n} x_i^2 + c \sum_{i=1}^{n} x_i^3$$

Putting values from table in this equation,

$$10 = 0 + 10b$$

i.e. $b = 1$ ... (5.8.9)

The last equation in equation 5.8.4 is,

$$\sum_{i=1}^{n} x_i^2\, y_i = a \sum_{i=1}^{n} x_i^2 + b \sum_{i=1}^{n} x_i^3 + c \sum_{i=1}^{n} x_i^4$$

Putting the values from table in this equation,

$$140 = 10a + 0 + 34c$$

i.e. $10a + 34c = 140$ ... (5.8.10)

On solving equation 5.8.8, equation 5.8.9 and equation 5.8.10, for a, b and c.

We get, $a = -1.057$, $b=1$, & $c = 4.428$

Putting values of a, b and c obtained above in equation 5.8.7 we get,

$$y = -1.057 + x + 4.428\,x^2$$

This is the required polynomial of degree two which approximates given data.

Ex. 5.8.3 *Use least square approximation to fit a law $y = a+bx+cx^2$ to the following data.*

x_i	1	2	3	4	5
y_i	3.38	8.25	16.6	28.5	44.00

Sol. : The normal equations for the given approximation $y = a+bx+cx^2$ are given by equation 5.8.4. Let's prepare the coefficients table for these equations.

x	y	xy	x^2	x^2y	x^3	x^4
1	3.38	3.38	1	3.38	1	1
2	8.25	16.5	4	33.00	8	16
3	16.60	49.8	9	149.4	27	81
4	28.5	114.00	16	456	64	256
5	44.00	220.00	25	1100	125	625
$\sum x_i = 15$	$\sum y_i = 100.73$	$\sum x_i y_i = 403.68$	$\sum x_i^2 = 55$	$\sum x_i^2\, y_i = 1741.78$	$\sum x_i^3 = 225$	$\sum x_i^4 = 979$

The normal equations are given by equation 5.8.4 as,

$$\sum_{i=1}^{n} y_i = n\,a + b \sum_{i=1}^{n} x_i + c \sum_{i=1}^{n} x_i^2$$

$$\sum_{i=1}^{n} x_i\, y_i = a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2 + c\sum_{i=1}^{n} x_i^3$$

& $$\sum_{i=1}^{n} x_i^2\, y_i = a\sum_{i=1}^{n} x_i^2 + b\sum_{i=1}^{n} x_i^3 + c\sum_{i=1}^{n} x_i^4$$

Putting the values from table in above equations we get following equations.

$$5a + 15b + 55c = 100.73$$

$$15a + 55b + 225c = 403.68$$

& $$55a + 225b + 979c = 1741.78$$

On solving these equations we get,

$$a = 2.104, \quad b = -0.484, \; c = 1.772$$

Therefore the equation of the function becomes,

$$y = 2.104 - 0.484\,x + 1.772\,x^2$$

Ex. 5.8.4 *Fit a parabola of the form $y = ax^2 + bx + c$ to the following data using least square approximation criteria.*

x	1	2	3	4	5	6	7
y	−5	−2	5	16	31	50	73

Sol. : Let us prepare the coefficient table as follows :

x	y	xy	x^2	x^2y	x^3	x^4
1	−5	−5	1	−5	1	1
2	−2	−4	4	−8	8	16
3	5	15	9	45	27	81
4	16	64	16	256	64	256
5	31	155	25	775	125	625
6	50	300	36	1800	216	1296
7	73	511	49	3577	343	2401
$\sum x_i = 28$	$\sum y_i = 168$	$\sum x_i y_i = 1036$	$\sum x_i^2 = 140$	$\sum x_i^2\, y_i = 6440$	$\sum x_i^3 = 784$	$\sum x_i^4 = 4676$

The set of normal equations are given by equation 5.8.4 as,

$$\sum y_i = na + b\sum x_i + c\sum x_i^2$$

$$\sum x_i y_i = a\sum x_i + b\sum x_i^2 + c\sum x_i^3$$

$$\sum x_i^2 y_i = a\sum x_i^2 + b\sum x_i^3 + c\sum x_i^4$$

Here n = 7 data points. Putting various values in above equations we get.

$$7a + 28b + 140c = 168$$

$$28a + 140b + 784c = 1036$$

$$140a + 784b + 4676c = 6440$$

There are three unknowns and three equations. Solving the above equations for a, b and c we get,

$a = -4, \quad b = -3$ and $c = 2$

The above values of 'a' 'b' and 'c' are obtained with the set of normal equations defined by equation 5.8.4. These normal equations approximate the parabola given by equation 5.8.3 i.e.,

$$y = a + bx + cx^2$$

The above equation can be written as,

$$y = cx^2 + bx + a \qquad \text{... (5.8.11)}$$

Putting the values in above equations,

$$y = 2x^2 - 3x - 4$$

This is the required parabola for the given data.

Important note

Here in this example, the parabola of the form $y = ax^2 + bx + c$ is asked. As per our regular procedure, we have obtained the porabola of the form given by equation 5.8.11, i.e. $y = cx^2 + bx + a$. Note that both of these equations have same meaning. They represent a second degree function. The variables 'a' 'b' and 'c' are just arbitrary.

Ex. 5.8.5 *Fit a straight line for the following data :*

x	1	2	3	4	5	6	7
y	0.5	2.3	2.1	4.2	3.6	5.8	5.5

And evaluate value of y at x = 4.5.

[May-2001, 8 marks, Dec-98, 8 marks, Dec-96, 8 marks, May-2000, 10 marks, May-96, 8 marks, May-2004, 10 Marks]

Sol. : Here there are 7 data points. Let us prepare the following table :

i	x_i	y_i	x_i^2	$x_i\, y_i$
1	1	0.5	1	0.5
2	2	2.3	4	4.6
3	3	2.1	9	6.3
4	4	4.2	16	16.8

i	x_i	y_i	x_i^2	$x_i\,y_i$
5	5	3.6	25	18
6	6	5.8	36	34.8
7	7	5.5	49	38.5
	$\sum x_i = 28$	$\sum y_i = 24$	$\sum x_i^2 = 140$	$\sum x_i\,y_i = 119.5$

From equation 5.8.2 we have,

$$\sum_{i=1}^{n} y_i = na + b \sum_{i=1}^{n} x_i$$

Putting values from table,

$$24 = 7a + 28\,b \qquad \text{... (5.8.12)}$$

From equation 5.8.3 we have,

$$\sum_{i=1}^{n} x_i\,y_i = a \sum_{i=1}^{n} x_i + b \sum_{i=1}^{n} x_i^2$$

Putting values from the table,

$$119.5 = 28\,a + 140\,b$$

Solving the above equation and equation 5.8.12 for a and b, we get,

$$a = 0.071429 \quad \text{and} \quad b = 0.839286$$

The straight line equation is given as,

$$y = a + b\,x$$

Putting values in this equation

$$y = 0.071429 + 0.839286\,x$$

This is the required straight line fit for given data.

Putting $x = 4.5$ in above equation,

$$y\,(4.5) = 0.071429 + 0.839286\,(4.5) = 3.848216$$

5.8.4 Linerization of Nonlinear Relationships

Now we will determine about how nonlinear relationships are linearized.

(i) Power function $y = a\,x^b$

Let the function $y = a\,x^b$ is to be fitted to the given data. This is nonlinear function. Taking logarithm on both sides,

$$\log y = \log a + \log x^b$$

$$\therefore \quad \log y = \log a + b \log x \qquad \text{... (5.8.13)}$$

Here let $Y = \log y$, $X = \log x$ and $a_0 = \log a$

Then above equation becomes,

$$Y = a_0 + b\,X \qquad \text{... (5.8.14)}$$

This is a straight line. a_0 and b can be obtained using regular method discussed earlier.

(ii) Exponential function y = a e^{bx}

Now let the exponential function y = a e^{bx} to be fitted to the given data. Taking natural logarithm of this function we get,

$$ln\ y = ln\, a + ln\, e^{bx}$$

$$ln\ y = ln\, a + bx \qquad \ldots (5.8.15)$$

Here let Y = $ln\ y$ $\quad a_0 = ln\, a$ Then the above equations becomes,

$$Y = a_0 + bx \qquad \ldots (5.8.16)$$

This is a straight line.

Ex. 5.8.6 *Determine the constants a and b by the method of least square such that $y = a e^{bx}$ fits the following data.*

x	2	4	6	8	10
y	4.077	11.084	30.128	81.897	222.62

[Dec - 2004, 10 Marks, Dec - 2002, 8 Marks]

Sol. : Here we have to fit the above data by $y = a e^{bx}$. This exponential function can be approximated as a straight line by eq 5.8.15 i.e,

$$ln\ y = ln\, a + bx$$

Hence we will prepare the following table.

i	x_i	y_i	$Y_i = ln\, y_i$	x_i^2	$x_i Y_i$
1	2	4.077	1.405	4	2.810
2	4	11.084	2.405	19	9.620
3	6	30.128	3.405	36	20.430
4	8	81.897	4.405	64	35.240
5	10	222.62	5.405	100	54.050
	$\Sigma x_i = 30$		$\Sigma Y_i = 17.025$	$\Sigma x_i^2 = 220$	$\Sigma x_i Y_i = 122.150$

Putting the values in eq 5.8.2 and 5.8.3

$$17.025 = 5a_0 + b \times 30$$

and

$$122.150 = a_0 \times 30 + b \times 220$$

Solving above two equations for a_0 and b we get,

$$a_0 = 0.405 \text{ and } b = 0.5$$

We know that $\quad a_0 = ln\, a$

$$\therefore \quad a = e^{a0} = e^{0.405} = 1.5$$

Putting the values of a = 1.5 and b = 0.5

We get,

$$y = 1.5\, e^{0.5x}$$

This is the required exponential approximation.

Ex. 5.8.7 *Using the method of least square fit a curve of the form $y = a\,x^b$ to the following data.*

x	1	2	3	4
y	4	11	35	100

[Dec - 2003, 8 Marks]

Sol. : Here we have to fit the function y = $a\,x^b$. This can be approximated by the straight line of the form

$$\log y = \log a + b \log x$$

Where Y = log y, X = log x and a_0 = log a.

Let us form the following table.

x_i	y_i	$X_i = \log x_i$	$Y_i = \log y_i$	X_i^2	$X_i Y_i$
1	4	0	0.602	0	0
2	11	0.301	1.041	0.09	0.313
3	35	0.477	1.544	0.227	0.736
4	100	0.602	2	0.362	1.204
		$\Sigma X_i = 1.38$	$\Sigma Y_i = 5.187$	$\Sigma X_i^2 = 0.679$	$\Sigma X_i Y_i = 2.253$

Putting the values in eq 5.8.2 and 5.8.3 we get,

$$5.187 = 4\,a_0 + b \times 1.38$$

and

$$2.253 = a_0 \times 1.38 + b \times 0.679$$

Solving above equations for a_0 and b,

$$a_0 = 0.508 \text{ and } b = 2.285$$

We know that a_0 = log a. Therefore,

$$a = 10^{a_0} = 10^{0.508} = 3.22$$

Putting the values of a = 3.22 and b = 2.285 we get,

$$y = 3.22\, x^{2.285}$$

Ex 5.8.8 *A table below gives the temperature T (°C) and length l(mm) of heated rod. If $l = a_0 + a_1 T$, find the best values of a_0 and a_1.*

T	20°	30°	40°	50°	60°	70°
l	800.3	800.4	800.6	800.7	800.9	801.0

[May-2003, 8 Marks]

Sol. : Following table can be formed for the given data

T_i	l_i	T_i^2	$T_i l_i$
20	800.3	400	16006
30	800.4	900	24012
40	800.6	1600	32024
50	800.7	2500	40035
60	800.9	3600	48054
70	801.0	4900	56070
$\Sigma T_i = 270$	$\Sigma l_i = 4803.9$	$\Sigma T_i^2 = 13900$	$\Sigma T_i l_i = 216201$

Following two equations are formed from above data

$$4803.9 = 6a_0 + 270a_1$$

and

$$216201 = 270\, a_0 + 13900\, a_1$$

Solving above equations for a_0 and a_1,

$$a_0 = 800 \quad \text{and} \quad a_1 = 0.0146$$

These are the best values of a_0 and a_1.

5.8.5 Algorithm

We will now write an algorithm for least squares approximation. We will write this algorithm for fitting a second degree equation for a given set of data points.

Algorithm :

Step 1 : ***Read the total number of data points i.e. 'n' and read their actual values i.e. x_i and y_i.***

Step 2 : *Calculate*

$$\sum_{i=1}^{n} x_i, \ \sum_{i=1}^{n} y_i, \ \sum_{i=1}^{n} x_i^2, \ \sum_{i=1}^{n} x_i^3$$

$$\sum_{i=1}^{n} x_i^4, \ \sum_{i=1}^{n} x_i y_i, \ \text{and} \ \sum_{i=1}^{n} x_i^2 y_i$$

From the given data points.

Step 3 : *Solve the following equations for a, b and c.*

$$\sum_{i=1}^{n} = n\,a + b\sum_{i=1}^{n} x_i + c\sum_{i=1}^{n} x_i^2$$

$$\sum_{i=1}^{n} x_i\, y_i = a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2 + c\sum_{i=1}^{n} x_i^3$$

& $$\sum_{i=1}^{n} x_i^2\, y_i = a\sum_{i=1}^{n} x_i^2 + b\sum_{i=1}^{n} x_i^3 + c\sum_{i=1}^{n} x_i^4$$

Step 4 : *The second degree approximation to the given data is,*

$y = a + bx + cx^2$

Substitute for a, b, and c in above equation.

Step 5 : *Display the full approximated equation obtained in step 4 or display only its coefficients (i.e. a, b and c).*

Step 6 : *Stop.*

Exercise

1. Find the parabola of second degree to the following data using least squares approximation.

x	0	1	2	3	4
y	1	1.8	1.3	2.5	2.3

[Ans. : $f(x) = 0.55\,x^2 - 1.07\,x + 1.42$]

2. Obtain the polynomial of degree two to approximate the following data using least squares method.

x	1	3	5	7	9
y	1	7	21	43	73

[Ans. : $f(x) = x^2 - x + 1$]

3. Fit a straight line of the form $Y = a_0 + a_1\,x$ to the following data.

x	1	2	3	4	6	8
y	2.4	3.1	3.5	4.2	5.0	6.0

[Ans. : $a_0 = 2.016$ and $a_1 = 0.503$]

4. Use the method of least squares to fit f (x) = a + bx to the following data.

x	0	1	2	3
y	2	5	8	11

[Ans. : a = 2, b = 3]

5. Find the values of a, b and c so that $Y = a + bx + cx^2$ is the best fit to the following data.

x	0	1	2	3	4
y	1	0	3	10	21

[Ans. : a = 1, b = – 3, c = 2]

University Questions

1. Write a short note on : Least square approximation methods.

[Dec - 95, May - 97, May - 98, Dec - 98, Dec - 99, May - 2000, May - 2001]

2. Explain the methods of best data fit so as to -
   i) Minimize the sum of residual errors
   ii) Minimize the sum of square of residual errors. **[May - 96, May - 99]**
3. Fit a straight line to the x and y values in the two columns :

x_i	1	2	3	4	5	6	7
y_i	0.5	2.5	2.0	4.0	3.5	6.0	5.5

[May - 96]

4. Fit a straight line to the (x, y) values in the table below :

x_1	1	2	3	4	5	6	7
y_1	0.5	2.3	2.1	4.2	3.6	5.8	5.5

and evaluate the value of y at x = 4.5. **[Dec - 96, Dec - 98, May - 2000, May - 2001]**

5. Explain least square curve fitting procedures for fitting a straight line, non linear curve fitting of power function and a polynomial of n^{th} degree. **[Dec - 2002]**
6. Determine the constants a and b by the method of least square such that $y = ae^{bx}$ fits the following data : **[Dec - 2002]**

x :	2	4	6	8	10
y :	4.077	11.084	30.128	81.897	222.62

7. Using the method of least square fit a curve of the form $y = ax^b$ to the following data. **[Dec - 2003]**

x	1	2	3	4
y	4	11	35	100

8. Given a table of values : **[Dec - 2003]**

x	0.1	0.2	0.3	0.4	0.5	0.6
y	5.1	5.3	5.6	5.7	5.9	6.1

Obtain the regression line of y on x.

9. Fit a straight line to the x and y values given : **[May - 2004]**

x	1	2	3	4	5	6	7
y	0.5	2.5	2.0	4.0	3.5	6.0	6.5

10. Determine the constants a and b by the method of least square such that $y = ae^{bx}$ fits the following data : **[Dec - 2004]**

x	2	4	6	8	10
y	4.77	11.084	30.128	81.897	222.62

5.9 Engineering Applications

In this chapter we studied several methods of interpolation and curve fitting. These methods are largely used in electronics and electrical engineering applications. Following examples illustrate these applications.

Ex. 5.9.1 *An experiment carried out on a circuit give the following readings :*

V_{in} (volts)	0	0.01	0.02	0.03	0.04	0.05
V_{out} (volts)	0	1.52	2.65	3.84	4.65	6.25

Using the above table evaluate

i) V_{out} at V_{in} = 0.025 volts

ii) V_{in} at V_{out} = 5.00 volts.

[Dec - 99, 9 marks, May - 98, 8 marks, Dec - 95, 6 marks]

Sol. : The given data can be considered as the inputs and outputs of voltage amplifier.

i) To obtain V_{out} at $V_{in} = 0.025$ volts

The input voltages are evenly spaced, hence newton's forward differences interpolation can be used. Let us represent the given data as follows :

$x = V_{in}$	$x_0 = 0$	$x_1 = 0.01$	$x_2 = 0.02$	$x_3 = 0.03$	$x_4 = 0.04$	$x_5 = 0.05$
$y = V_{out}$	$y_0 = 0$	$y_1 = 1.52$	$y_2 = 2.65$	$y_3 = 3.84$	$y_4 = 4.65$	$y_5 = 6.25$

Let us form the forward differences table of the above data :

$x = V_{in}$	$y = V_{out}$	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$
$x_0 = 0$	$y_0 = 0$					
		$\Delta y_0 = 1.52$				
$x_1 = 0.01$	$y_1 = 1.52$		$\Delta^2 y_0 = -0.39$			
		1.13		$\Delta^3 y_0 = 0.45$		

$x_2 = 0.02$	$y_2 = 2.65$		0.06		$\Delta^4 y_0 = -0.89$	
		1.19		-0.44		$\Delta^2 y_0 = 2.5$
$x_3 = 0.03$	$y_3 = 3.84$		-0.38		1.61	
		0.81		1.17		
$x_4 = 0.04$	$y_4 = 4.65$		0.79			
		1.6				
$x_5 = 0.05$	$y_5 = 6.25$					

From equation 5.4.1 we know that,

$$x_r = x_0 + r\,h$$

Here $x_r = 0.025$, $x_0 = 0$ and $h = x_2 - x_1 = 0.02 - 0.01 = 0.01$

Hence
$$r = \frac{x_r - x_0}{h}$$
$$= \frac{0.025 - 0}{0.01} = 2.5$$

Newton's forward differences interpolation formula is given by equation 5.4.4 as,

$$y_r = y_0 + r\,\Delta y_0 + \frac{r(r-1)}{2!}\Delta^2 y_0 + \frac{r(r-1)(r-2)}{3!}\Delta^3 y_0 + \ldots\ldots$$

Putting values in above equation,

$$y_r = 0 + 2.5\,(1.52) + \frac{2.5\,(2.5-1)}{2\,!}(-\,0.39)$$
$$+ \frac{2.5\,(2.5-1)\,(2.5-2)}{3\,!}(0.45)$$
$$+ \frac{2.5\,(2.5-1)\,(2.5-2)\,(2.5-3)}{4\,!}(-\,0.89)$$
$$+ \frac{2.5\,(2.5-1)\,(2.5-2)\,(2.5-3)\,(2.5-4)}{5\,!}(2.5)$$
$$= 3.273437$$

Thus at $V_{in} = 0.025$, $V_{out} = 3.273437$ volts.

ii) To obtain V_{in} at V_{out} = 5.00 volts

This is the problem of inverse interpolation. This can be solved using lagranges interpolation by switching the V_{in} and V_{out} values. Here we will solve this problem without switching of variables. Since $V_{out} = 5.00$ volts lie near end of the data, we will consider following data points :

$x = V_{in}$	$x_0 = 0.03$	$x_1 = 0.04$	$x_2 = 0.05$
$y = V_{out}$	$y_0 = 3.84$	$y_1 = 4.65$	$y_2 = 6.25$

Let us form backward differences table as follows :

$x = V_{in}$	$y = V_{out}$	∇y	$\nabla^2 y$
$x_0 = 0.03$	$y_0 = 3.84$		
		$\nabla y_1 = 0.81$	
$x_1 = 0.04$	$y_1 = 4.65$		$\nabla^2 y_0 = 0.79$
		$\nabla y_2 = 1.6$	
$x_2 = 0.05$	$y_2 = 6.25$		

From equation 5.4.12 we have,

$$r = \frac{x_r - x_n}{h}$$

Here $x_n = 0.05$ and $h = x_1 - x_0 = 0.04 - 0.03 = 0.01$

$$\therefore \qquad r = \frac{x_r - 0.05}{0.01} = 100\,(x_r - 0.05) = 100\,x_r - 5$$

Newton's backward differences formula is given by equation 5.4.10 as,

$$y_r = y_n + r\,\nabla\,y_n + \frac{r\,(r+1)}{2!}\,\nabla^2\,y_n + \ldots\ldots\ldots$$

Putting values in above equation,

$$y_r = 6.25 + r\,(1.6) + \frac{r\,(r+1)}{2!}\,(0.79)$$

$$= 0.395\,r^2 + 1.995\,r + 6.25$$

Putting the value of 'r' in above equation,

$$y_r = 0.395\,(100\,x_r - 5)^2 + 1.995\,(100\,x_r - 5) + 6.25$$

$$= 3950\,x_r^2 - 195.5\,x_r + 6.15$$

This is the second order polynomial passing through last three points. Now we want x_r at $y_r = V_{out} = 5.00$. Hence above equation will be,

$$5 = 3950\,x_r^2 - 195.5\,x_r + 6.15$$

Solving the above equation for x_r we get,

$$x_r = 0.0425198 \quad \text{and} \quad 0.006847$$

The first value is correct since it lies near x_1 and x_2. Thus,

At $V_{out} = 5.00$ volts, $V_{in} = 0.0425198$ volts

More number of terms can be taken in the difference table for more accuracy.

Ex. 5.9.2 *A test performed on NPN transistor gives the following result :*

Base current $I_b(mA)$	0	0.01	0.02	0.03	0.04	0.05
Collector current $I_c(mA)$	0	1.2	2.5	3.6	4.3	5.34

Calculate (i) the value of the collector current for the base current of 0.005 mA
(ii) the value of base current required for a collector current of 4.0 mA.

[Dec - 2004, 20 Marks]

Sol. : i) To obtain I_c for I_b = 0.005

Here I_b = 0.005 value lies at the begining of the data points. Hence we will use Newton's forward differences interpolation. Using the procedure discussed earlier, I_c can be obtained as,

$$I_c = 0.587812 \text{ mA for } I_b = 0.005 \text{ mA}$$

ii) To obtain I_b for I_c = 4 mA

Now I_c = 4 mA lies near 3.6 mA data point. Hence consider the following data points :

$x = I_b$	$x_0 = 0.03$	$x_1 = 0.04$	$x_2 = 0.05$
$y = I_c$	$y_0 = 3.6$	$y_1 = 4.3$	$y_2 = 5.34$

Now let us form the forward differences table of above data :

$x = I_b$	$y = I_c$	Δy	$\Delta^2 y$
$x_0 = 0.03$	$y_0 = 3.6$		
		$\Delta y_0 = 0.7$	
$x_1 = 0.04$	$y_1 = 4.3$		$\Delta^2 y_0 = 0.34$
		1.4	
$x_2 = 0.05$	$y_2 = 5.34$		

Now we have, $\quad r = \dfrac{x_r - x_0}{h}$

Here $\quad x_0 = 0.03$ and h = 0.01, hence

$$r = \frac{x_r - 0.03}{0.01}$$

$$= 100(x_r - 0.3) = 100x_r - 3$$

Newton's forward differences formula is given as,

$$y_r = y_0 + r\,\Delta y_0 + \frac{r(r-1)}{2!}\Delta^2 y_0 + \cdots$$

$$= 3.6 + 0.7r + \frac{r(r-1)}{2} \times 0.34$$

Putting $y_r = 4$ in above equation and solving for 'r' we get,

$$r = 0.6281542 \text{ or } -3.7458013$$

Since $r = 100x_r - 3$, then we get x_r as,

$$x_r = 0.0362815 \text{ or } -0.00745$$

Looking at data points we find that $x_r = 0.0362815$ is more correct value of I_b for $I_c = 4$ mA.

Exercise

1. Following table shows the voltage drop across the resistance for various currents :

i (Amp)	0.25	0.75	1.25	1.5	2.0
v (volts)	– 0.45	– 0.6	0.7	1.88	6.0

Determine the voltage drop for $i = 1.1$.

Ans. : v = 0.176198 volts using lagrange's interpolation.

University Questions

1. An experiment carried out on a circuit gave the following readings.

V_{in}(volts)	0	0.01	0.02	0.03	0.04	0.05
V_{out}(volts)	0	1.52	2.65	3.84	4.65	6.25

Using the above table evaluate

i) V_{out} at V_{in} = 0.025 volts ii) V_{in} at V_{out} = 5.00 volts **[Dec - 95, May - 98, Dec - 99]**

2. For such a table given above, develop a flow chart and write a C program to generate the difference table and hence calculate the voltage V_{out} at any given value of V_{in} using Newton's forward interpolation formula. **[Dec - 95]**

3. The table below gives the temperature T (° C) and length l (mm) of heated rod. **[May - 2003]**

If $l = a_0 + a_1$ T. Find the best value of a_0 and a_1

T	20°	30°	40°	50°	60°	70°
l	800.3	800.4	800.6	800.7	800.9	801.0

4. A test performed on NPN transistor gave the following result :

Base current $I_b(mA)$	0	0.01	0.02	0.03	0.04	0.05
Collector current $I_c(mA)$	0	1.2	2.5	3.6	4.3	5.34

Calculate (i) the value of the collector current for the base current of 0.005 mA

(ii) the value of base current required for a collector current of 4.0 mA.

Use suitable interpolation methods. Derive the formula used and write the algorithm for case (i). **[Dec - 2004]**

5.10 MATLAB for Curve Fitting

Now let us see how MATLAB can be used for curve fitting. A program is given below for fitting a polynomial for given data :

```
%   Download this program from www.vtubooks.com
%   file name : curveFitting.m

%   Polynomial approximation for the given set of data points.
%                                                        Matlab Version 6
%   This program fits the polynomial to the given set of data points.

%   Inputs :       1.  Data points x and y
%                  2.  Order of the polynomial to be interpolated

%   Outputs:       1.  Coefficients of the polynomial
%                  2.  Display of data points and curve y = f(x)

%------------------------------------------------------------------------------
clc                                         % clear the screen
disp('  Curve Fitting ( Polynomial Interpolation )'); disp(' ');

x = input('Enter the values of x = ');  %   Values of x entered here
y = input('Enter the values of y = ');  %   Values of y entered here
n = input('Enter the order of the polynomial = ');
                              %   order of the polynomial entered here

p = polyfit(x,y,n);           %   calculate coefficients of the polynomial

disp(' ');
disp('The coefficients of the polynomial are as follows...'); disp(' ');
disp(p);                      %   Display the coefficients of the polynomial

%----- Next part displays the interpolated polynomial and data points ----

xn = 0:0.1:10;            %   generate points from 0 to 10 in steps of 0.1
yn = polyval(p,xn);       %   calculate values of polynomial at
%                                                  above generated points
plot(x,y,'o',xn,yn);      %   plot data points (x,y) and polynomial(xn,yn)
xlabel('x'); ylabel('y = f(x)');          %   Label the x and y axes
title('Curve Fitting');                   %   Title of the plot

%-------------- End of the program ------------------------------------
```

This program accepts the data points x, y and order 'n' of the polynomial to be interpolated. The first three `input` statements accept these values. Next, there is a statement,

```
p = polyfit(x,y,n);
```

This statement calculates coefficients of a polynomial or order `n` and assigns to variable `p`. The `polyfit` function generates a best fit polynomial (in least squares sense) for the given set of data points. The next `disp` statements display the coefficients of the polynomial.

Next, there are following statements :

```
xn = 0:0.1:10;
yn = polyval(p,xn);
```

Here `xn` is a vector of values of 0 to 10 in steps of 0.1. And `yn` represents the values of polynomial for `xn`. The `polyval` function calculates $y = f(x)$ for given `x=xn`. These values are calculated to plot the polynomial. The next `plot` statement plots the polynomial and (x, y) data points simultaneously.

To test the program :

Let us test this program with the data given in example 5.5.6. This data is given below :

x	0	1	2	5
$y = f(x)$	2	3	12	147

The 3^{rd} order polynomial is obtained for this data. i.e.,

$$y = x_r^3 + x_r^2 - x_r + 2$$

The coefficients of this polynomial are,

$$p = [1, 1, -1, 2]$$

If we given x, y from above table and $n = 3$ as input to the MATLAB program, we will get above coefficients. Here is the display of MATLAB results.

```
%--------------------- Results ----------------------------------------
  Curve Fitting ( Polynomial Interpolation )

Enter the values of x = [0 1 2 5]
Enter the values of y = [2 3 12 147]
Enter the order of the polynomial = 3

The coefficients of the polynomial are as follows...

    1.0000     1.0000    -1.0000     2.0000
Please refer Fig. 5.10.1 on next page.

%---------------------------------------------------------------------
```

In the above results, observe that exactly same coefficients are given by the MATLAB program. Fig. 5.10.1 shows the plot of the polynomial and data points. Observe that the polynomial passes through all the data points.

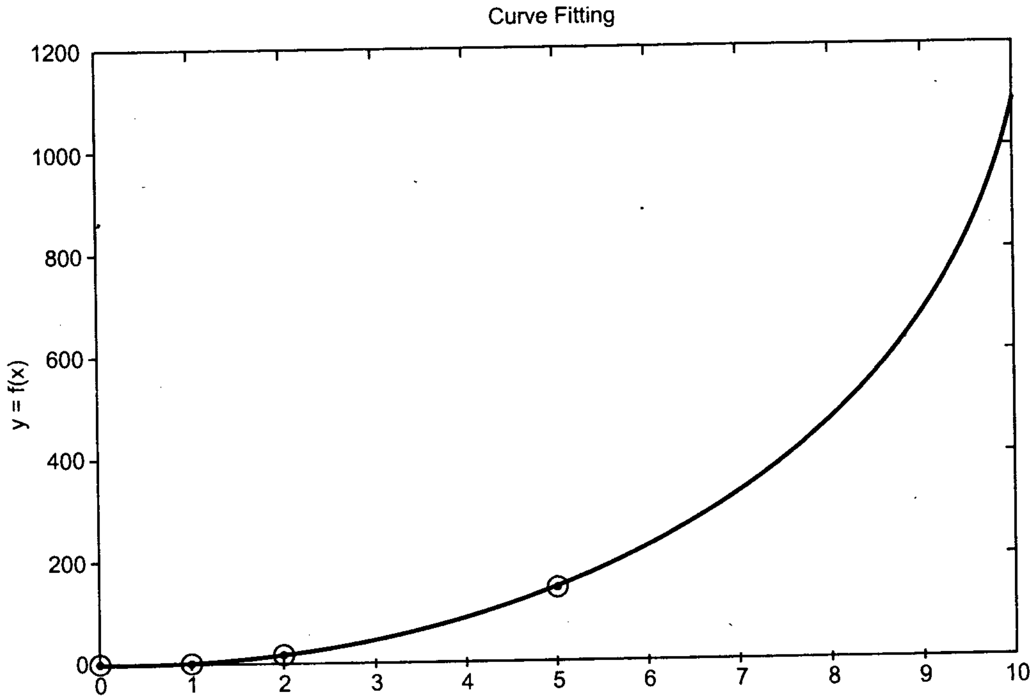


Fig. 5.10.1 Plot of x, y and y = f(x)

Computer Exercise

1. Write the program in 'C' language to generate a central differences table. Make facility in the program to shift the point x_0.
2. Write the program in 'C' language which combines the programs of newton's forward differences interpolation and newton's backward differences interpolation. Your program should ask about which interpolation is required.
3. Write a 'C' program for linear regression and polynomial regression.

❑❑❑

6
Numerical Differentiation

6.1 Introduction

Differentiation operation is required in large number of applications. For simple cases, differentiation can be calculated by direct methods. But for complicated ones, numerical differentiation is used. Numerical differentiation can work on tabulated data or interpolated polynomial. Based on this, there are three methods for numerical differentiation.

i) Methods based on interpolation

ii) Methods based on finite difference operators and

iii) Methods based on undetermined coefficients.

In this chapter we will discuss first two methods.

6.2 Methods Based on Interpolation

The methods based on interpolation can be separately studied for equispaced and nonequispaced data points.

6.2.1 Equispaced (Uniform) Data Points

Consider Newton's forward differences interpolation formula,

$$f(x_r) = y_r = y_0 + r\,\Delta y_0 + \frac{r(r-1)}{2!}\,\Delta^2 y_0 + \frac{r(r-1)(r-2)}{3!}\,\Delta^3 y_0 + \ldots\ldots \qquad \text{... (6.2.1)}$$

Here $\qquad x_r = x_0 + r\,h$

and $\qquad h = x_1 - x_0 = x_2 - x_1 = \ldots\ldots$

The derivative of equation 6.2.1 will be $\frac{dy_r}{dx_r}$. We can write this derivative as,

$$\frac{dy_r}{dx_r} = \frac{dy_r}{dr}\,\frac{dr}{dx_r} \qquad \text{... (6.2.2)}$$

We know that, $x_r = x_0 + r\,h$

Differentiating the above equation with respect to r we get,

$$1 = 0 + h\frac{dr}{dx_r}$$

$$\therefore \quad \frac{dr}{dx_r} = \frac{1}{h} \quad \text{... (6.2.3)}$$

We can write equation 6.2.1 as,

$$f(x_r) = y_r = y_0 + r\,\Delta y_0 + \frac{r^2 - r}{2!}\Delta^2 y_0 + \frac{r^3 - 3r^2 + 2r}{3!}\Delta^3 y_0 + ...$$

Differentiate this equation with respect to r we get,

$$\frac{dy_r}{dr} = \Delta y_0 + \frac{2r-1}{2!}\Delta^2 y_0 + \frac{3r^2 - 6r + 2}{3!}\Delta^2 y_0 + ... \quad \text{... (6.2.4)}$$

Putting the values of $\frac{dy_r}{dr}$ and $\frac{dr}{dx_r}$ from equation 6.2.4 and equation 6.2.3 in equation 6.2.2 we get,

$$\frac{dy_r}{dx_r} = \frac{1}{h}\left[\Delta y_0 + \frac{2r-1}{2!}\Delta^2 y_0 + \frac{3r^2 - 6r + 2}{3!}\Delta^3 y_0 + ...\right] \quad \text{... (6.2.5)}$$

Using this formula we can obtain the derivative at any point such as $x = x_r$ from given data.

If x_r is one of the value of x in given data points then let $x_r = x_0$

$$\therefore \quad x_0 = x_0 + rh \qquad (\because \; x_r = x_0 + rh)$$

$$r = 0$$

Putting this value in equation 6.2.5 we get a new equation for differentiation at given data points.

$$\left[\frac{dy}{dx}\right]_{x=x_0} = \frac{1}{h}\left[\Delta y_0 - \frac{1}{2}\Delta^2 y_0 + \frac{1}{3}\Delta^3 y_0 - \frac{1}{4}\Delta^4 y_0 + ...\right] \quad \text{... (6.2.6)}$$

Differentiating equation 6.2.5 once again we get,

$$\frac{d^2 y_r}{dx_r^2} = \frac{1}{h^2}\left[\Delta^2 y_0 + \frac{6r-6}{6}\Delta^3 y_0 + \frac{12r^2 - 36r + 22}{24}\Delta^4 y_0 + ...\right]$$

... (6.2.7)

If x_r is the point from given data points, then let $x_r = x_0$.

$$x_0 = x_0 + rh \qquad (\because \; x_r = x_0 + rh)$$

$$r = 0$$

Putting this value in equation 6.2.7 we get a new equation for second derivative at given data points,

$$\left[\frac{d^2y}{dx^2}\right]_{x=x_0} = \frac{1}{h^2}\left[\Delta^2 y_0 - \Delta^3 y_0 + \frac{11}{12}\Delta^4 y_0 - \frac{5}{6}\Delta^5 y_0 + \frac{137}{180}\Delta^6 y_0 + \ldots\right] \qquad \ldots (6.2.8)$$

To evaluate differentiation at points near x_n :

This can be done with Newton's backward differences interpolation formula. The derivative at one of the points in the given data points will be,

$$\left[\frac{dy}{dx}\right]_{x=x_n} = \frac{1}{h}\left[\nabla y_n + \frac{1}{2}\nabla^2 y_n + \frac{1}{3}\nabla^3 y_n + \frac{1}{4}\nabla^4 y_n + \frac{1}{5}\nabla^5 y_n + \ldots\ldots\right] \qquad \ldots (6.2.9)$$

Similarly second derivative at one of the given data points will be,

$$\left[\frac{d^2y}{dx^2}\right]_{x=x_n} = \frac{1}{h^2}\left[\nabla^2 y_n + \nabla^3 y_n + \frac{11}{12}\nabla^4 y_n + \frac{5}{6}\nabla^5 y_n + \frac{137}{180}\nabla^6 y_n + \ldots\right] \qquad \ldots (6.2.10)$$

6.2.2 Non-uniform Nodal Points

For non-uniform data points lagrange's interpolation is used. The lagrange's polynomial is then differentiated. Using linear interpolation, derivative is given as,

$$\frac{dy}{dx} = \frac{y_1 - y_0}{x_1 - x_0} \qquad \ldots (6.2.11)$$

Using quadratic interpolation,

$$\frac{dy}{dx} = \frac{2x - x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)} y_0 + \frac{2x - x_0 - x_2}{(x_1 - x_0)(x_1 - x_2)} y_1 + \frac{2x - x_0 - x_1}{(x_2 - x_0)(x_2 - x_1)} y_2 \qquad \ldots (6.2.12)$$

and $$\frac{d^2y}{dx^2} = 2\left\{\frac{y_0}{(x_0 - x_1)(x_0 - x_2)} + \frac{y_1}{(x_1 - x_0)(x_1 - x_2)} + \frac{y_2}{(x_2 - x_0)(x_2 - x_1)}\right\} \qquad \ldots (6.2.13)$$

6.2.3 Solved Examples

Ex. 6.2.1 *From the following table of values of x and y obtain* $\frac{dy}{dx}$ *and* $\frac{d^2y}{dx^2}$ *for* $x = 1.2$

x	1.0	1.2	1.4	1.6	1.8	2.0	2.2
y	2.7183	3.3201	4.0552	4.9530	6.0496	7.3891	9.0250

Sol. : The difference table is shown below.

Table 6.2.1 Forward/Backward differences table

x	y	Δy or ∇y	$\Delta^2 y$ or $\nabla^2 y$	$\Delta^3 y$ or $\nabla^3 y$	$\Delta^4 y$ or $\nabla^4 y$	$\Delta^5 y$ or $\nabla^5 y$	$\Delta^6 y$ or $\nabla^6 y$
1.0	2.7183						
		0.6018					
1.2	3.3201		0.1333				
		0.7351		0.0294			
1.4	4.0552		0.1627		0.0067		
		0.8978		0.0361		0.0013	
1.6	4.9530		0.1988		0.0080		0.0001
		1.0966		0.0441		0.0014	
1.8	6.0496		0.2429		0.0094		
		1.3395		0.0535			
2.0	7.3891		0.2964				
		1.6359					
2.2	9.0250						

Here take $x_0 = 1.2$ and $y_0 = 3.3201$

Here $h = 0.2$

From equation 6.2.6 we have,

$$\left[\frac{dy}{dx}\right]_{x=x_0} = \frac{1}{h}\left[\Delta y_0 - \frac{1}{2}\Delta^2 y_0 + \frac{1}{3}\Delta^3 y_0 - \frac{1}{4}\Delta^4 y_0 + \ldots\ldots\right]$$

According to $x_0 = 1.2$ and $y_0 = 3.3201$ we have from table 6.2.1,

$\Delta y_0 = 0.7351$, $\Delta^2 y_0 = 0.1627$, $\Delta^3 y_0 = 0.0361$,

$\Delta^4 y_0 = 0.0080$, $\Delta^5 y_0 = 0.0014$

Putting these values in the above equation,

$$\left[\frac{dy}{dx}\right]_{x_0=1.2} = \frac{1}{0.2}\left[0.7351 - \frac{1}{2}\times 0.1627 + \frac{1}{3}\times 0.0361 - \frac{1}{4}\times 0.0080 + \frac{1}{5}\times 0.0014\right]$$

$$= 3.3205$$

Thus $$\left.\frac{dy}{dx}\right|_{x=1.2} = 3.3205$$

The second derivative is given by equation 6.2.8 as,

$$\left[\frac{d^2y}{dx^2}\right]_{x=x_0} = \frac{1}{h^2}\left[\Delta^2 y_0 - \Delta^3 y_0 + \frac{11}{12}\Delta^4 y_0 - \frac{5}{6}\Delta^5 y_0 + \ldots\ldots\right]$$

Putting the values of forward differences in above equation,

$$\left.\frac{d^2y}{dx^2}\right|_{x=1.2} = \frac{1}{0.04}\left[0.1627 - 0.0361 + \frac{11}{12}\times 0.0080 - \frac{5}{6}\times 0.0014\right] = 3.318$$

Ex. 6.2.2 *Calculate the first and second derivatives of the function tabulated in the previous example at point $x = 2.2$ and also at $x = 2.0$.*

Sol. : Since x = 2.2 is last data point in the table, we will use backward differences to find derivative.

Let $x_n = 2.2 \quad \therefore \quad y_n = 9.0250$ and $h = 0.2$

Let's use the formula of equation 6.2.9 i.e.,

$$\left[\frac{dy}{dx}\right]_{x=x_n} = \frac{1}{h}\left[\nabla y_n + \frac{1}{2}\nabla^2 y_n + \frac{1}{3}\nabla^3 y_n + \frac{1}{4}\nabla^4 y_n + \frac{1}{5}\nabla^5 y_n + \ldots\right]$$

According to $x_n = 2.2$ and $y_n = 9.0250$, the backward differences are,

$\nabla y_n = 1.6359, \quad \nabla^2 y_n = 0.2964, \quad \nabla^3 y_n = 0.0535,$

$\nabla^4 y_n = 0.0094, \quad \nabla^5 y_n = 0.0014, \quad \nabla^6 y_n = 0.0001$

Putting those values in above equation we get,

$$\left[\frac{dy}{dx}\right]_{x=2.2} = \frac{1}{0.2}\left[\begin{array}{l}1.6359 + \frac{1}{2}\times 0.2964 + \frac{1}{3}\times 0.0535 \\ + \frac{1}{4}\times 0.0094 + \frac{1}{5}\times 0.0014 + \frac{1}{6}\times 0.0001\end{array}\right] = 9.0228$$

From equation 6.2.10 second derivative is given as,

$$\left[\frac{d^2y}{dx^2}\right]_{x=x_n} = \frac{1}{h^2}\left[\nabla^2 y_n + \nabla^3 y_n + \frac{11}{12}\nabla^4 y_n + \frac{5}{6}\nabla^5 y_n + \frac{137}{180}\nabla^6 y_n + \ldots\right]$$

Putting the values of finite differences in above equation we get,

$$\left[\frac{d^2y}{dx^2}\right]_{x=2.2} = \frac{1}{0.04}\left[0.2964 + 0.0535 + \frac{11}{12}\times 0.0094 + \frac{5}{6}\times 0.0014 + \frac{137}{180}\times 0.0001\right]$$

$$= 8.992$$

Similarly using the same formulae given above, calculate first and second derivatives at x = 2.0.

For this take $x_n = 2.0$ and $y_n = 7.3891$. The finite differences will be changed according to their new values of x_n and y_n.

Ex. 6.2.3 *Consider the following function*

x	$x_0 = 2.0$	$x_1 = 2.2$	$x_2 = 2.6$
$y = f(x)$	$y_0 = 0.69315$	$y_1 = 0.78846$	$y_0 = 0.95551$

Determine approximate value of $f'(x_0 = 2.0)$ *and* $f''(x_0 = 2.0)$.

Sol. : From equation 6.2.11,

$$\frac{dy}{dx} = \frac{y_1 - y_0}{x_1 - x_0}$$

$$= \frac{0.78846 - 0.69315}{2.2 - 2.0} = 0.47655$$

Thus $f'(2.0) = 0.47655$ using linear interpolation.

From equation 6.2.12,

$$\frac{dy}{dx} = \frac{2x_0 - x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)} y_0 + \frac{x_0 - x_2}{(x_1 - x_0)(x_0 - x_2)} y_1 + \frac{x_0 - x_1}{(x_2 - x_0)(x_2 - x_1)} y_2$$

$$= \frac{2(2.0) - 2.2 - 2.6}{(2.0 - 2.2)(2.0 - 2.6)} 0.69315$$

$$+ \frac{2.0 - 2.6}{(2.2 - 2.0)(2.0 - 2.6)} 0.78846$$

$$+ \frac{2.0 - 2.2}{(2.6 - 2.0)(2.6 - 2.2)} (0.95551)$$

$$= 0.49619$$

Thus $f'(2.0) = 0.49619$ using quadratic interpolation

From equation 6.2.13,

$$\frac{d^2 y}{dx^2} = 2\left\{\frac{y_0}{(x_0 - x_1)(x_0 - x_2)} + \frac{y_1}{(x_1 - x_0)(x_1 - x_2)} + \frac{y_2}{(x_2 - x_0)(x_2 - x_1)}\right\}$$

$$= 2\left\{\frac{0.69315}{(2.0 - 2.2)(2.0 - 2.6)} + \frac{0.78846}{(2.2 - 2.0)(2.2 - 2.6)} + \frac{0.95551}{(2.6 - 2.0)(2.6 - 2.2)}\right\}$$

$$= -0.19642$$

Thus $f''(2.0) = -0.19642$ using quadratic interpolation.

Exercise

1. *Find* $\frac{dy}{dx}$ *and* $\frac{d^2 y}{dx^2}$ *at* $x = 1.6$ *from the tabulated function of example 6.2.1.*

[Hint : Here use central differences formulae given by equation 6.2.11 and equation 6.2.12.

Take $x_0 = 1.6$ *and* $y_0 = 4.9530$.

[Ans. : $\frac{dy}{dx} = 4.9530$ **and** $\frac{d^2y}{dx^2} = 4.9525$**]**

2. *The following table gives the angular displacements θ (radians) at different intervals of time t (seconds).*

θ	0.052	0.105	0.168	0.242	0.327	0.408	0.489
t	0	0.02	0.04	0.06	0.08	0.10	0.12

Calculate the angular velocity at the instant $t = 0.06$.

[Hint : To calculate angular velocity 'ω' we have to differentiate angular displacement (θ) with respect to time (t) i.e.

Angular velocity $\omega = \frac{d\theta}{dt}$

Since differentiation is required at $t = 0.06$ *which is near center, use central differences formulae of equation 6.2.11 and equation 6.2.12.*

[Ans. : $\frac{d\theta}{dt} = 4.054$**]**

University Questions

1. Write a detailed note on Numerical differentiation.

[May - 97, Dec - 97, May - 2001, Dec - 2001]

6.3 Methods Based on Finite Differences

From equation 5.2.29 we know that,

$$E = e^{hD}$$

Here $D = f'(x) = \frac{d}{dx}$. We can write above equation as,

$$hD = \log E \quad \text{... (6.3.1)}$$

$$= \log(1+\Delta) \quad \text{since} \quad E = 1+\Delta$$

the log function can be expanded as,

$$hD = \Delta - \frac{1}{2}\Delta^2 + \frac{1}{3}\Delta^3 - \ldots\ldots$$

Let us operate above equation on y_0,

$$\therefore \quad h\,D\,(y_0) = \left(\Delta - \frac{1}{2}\Delta^2 + \frac{1}{3}\Delta^3 - \ldots\right) y_0$$

Here $Dy_0 = y_0'$ or $f'(x_0)$. Hence above equation can be written as,

$$y'_0 = \frac{1}{h}\left\{\Delta y_0 - \frac{1}{2}\Delta^2 y_0 + \frac{1}{3}\Delta^3 y_0 - \ldots\ldots\right\} \quad \ldots (6.3.2)$$

This equation uses forward finite differences.

We know that $E = (1-\nabla)^{-1}$, hence equation 6.3.1 can be written as,

$$hD = \log (1-\nabla)^{-1}$$
$$= \nabla + \frac{1}{2}\nabla^2 + \frac{1}{3}\nabla^3 + \ldots\ldots.$$

Let us operate above equation on y_n,

$$\therefore \quad hD(y_n) = \left(\nabla + \frac{1}{2}\nabla^2 + \frac{1}{3}\nabla^3 + \ldots\right) y_n$$

Here $Dy_n = y_n'$ or $f'(x_n)$. Hence above equation can be written as,

$$y_n = \frac{1}{h}\left\{\nabla y_n + \frac{1}{2}\nabla^2 y_n + \frac{1}{3}\nabla^3 y_n + \ldots\right\} \quad \ldots (6.3.3)$$

This equation uses backward finite differences. Note that above equation is same as equation 6.2.9 derived in previous section. Similarly equation 6.3.2 is same as equation 6.2.6.

University Question

1. Explain and derive Numerical differentiation for finite difference. **(Dec.-2002)**

6.4 Maximum and Minimum Values of the Function

At maximum and minimum values, the first derivative of the function is zero. This criteria is used to determine the maximum and minimum values of the function. Consider newton's forward differences formula (equation 5.4.4),

$$y_r = y_0 + r\,\Delta y_0 + \frac{r(r-1)}{2!}\Delta^2 y_0 + \frac{r(r-1)(r-2)}{3!}\Delta^3 y_0 + \ldots.$$

Differentiating above equation with respect to r, we get,

$$\frac{dy_r}{dr} = \Delta y_0 + \frac{2r-1}{2}\Delta^2 y_0 + \frac{3r^2-3r+2}{6}\Delta^3 y_0 + \ldots. \quad \ldots (6.4.1)$$

For maximum or minimum, $\frac{dy_r}{dr} = 0$. Hence above equation becomes,

$$\Delta y_0 + \frac{2r-1}{2}\Delta^2 y_0 + \frac{3r^2-3r+2}{6}\Delta^3 y_0 = 0 \quad \ldots (6.4.2)$$

For simplicity we have considered only upto third difference. Solving the above equation we get value of r. Then x_r can be obtained as,

$$x_r = x_0 + rh$$

Then we can obtain corresponding y_r by newton's forward or backward differences interpolation.

Ex. 6.4.1 *From the following table, find x, correct to two decimal places, for which y is maximum and also find the maximum value of y.*

x	1.2	1.3	1.4	1.5	1.6
y	0.9320	0.9636	0.9855	0.9975	0.9996

[Dec - 2001, 8 marks]

Sol. : Let us prepare the forward difference table of the given data :

x	y = f (x)	Δ y	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
$x_0 = 1.2$	$y_0 = 0.9320$				
		$\Delta y_0 = 0.0316$			
$x_1 = 1.3$	$y_1 = 0.9636$		$\Delta^2 y_0 = -0.0097$		
		0.0219		$\Delta^3 y_0 = -0.0002$	
$x_2 = 1.4$	$y_2 = 0.9855$		– 0.0099		$\Delta^4 y_0 = 0.0002$
		0.0120		0	
$x_3 = 1.5$	$y_3 = 0.9975$		– 0.0099		
		$\nabla y_4 = 0.0021$			
$x_4 = 1.6$	$y_4 = 0.9996$				

The value of x correct upto two digits is required. From the above table it is clear that if we consider upto second difference $(\Delta^2 y)$, x will be correct upto two digits. Putting these values in equation 6.4.2,

$$0.0316 + \frac{2r-1}{2}(-0.0097) = 0$$

Here note that we have neglected third and all the higher differences. Solving above equation for 'r' we get,

$$r = 3.757732$$

We know that, $\quad x_r = x_0 + rh$

Here $\quad h = x_1 - x_0 = 1.3 - 1.2 = 0.1$ and $x_0 = 1.2$

$$\therefore \quad x_r = 1.2 + (3.757732)(0.1)$$

$$= 1.5757732$$

Thus 'y' is maximum at x = 1.5757732.

To obtain maximum value of y :

We know that y is maximum at $x_r = 1.5757732$ observe that this value of x is near end of data points. Hence 'y' can be obtained using newton's backward differences interpolation. We know that,

$$r = \frac{x_r - x_n}{h} \quad \text{for backward differences.}$$

Here $h = 0.1$, $x_n = x_4 = 1.6$ and $x_r = 1.5757732$

$$\therefore \quad r = \frac{1.5757732 - 1.6}{0.1} = -0.242268$$

Newton's backward differences interpolation formula is given as,

$$y_r = y_n + r \nabla y_n + \frac{r(r+1)}{2!} \nabla^2 y_n + \frac{r(r+1)(r+2)}{3!} \nabla^3 y_n + \ldots..$$

Putting values in above equation,

$$y_r = 0.9996 + (-0.242268)(0.0021)$$

$$+ \frac{(-0.242268)(-0.242268+1)}{2!}(-0.0099) + 0$$

$$+ \frac{(-0.242268)(-0.242268+1)(-0.242268+2)(-0.242268+3)}{4!}(0.0002)$$

$$= 0.999993$$

Thus the maximum value of y is 0.999993.

Ex. 6.4.2 : *Given the data f(1) = 4, f(2) = 5, f(7) = 5 and f(8) = 4. Find the value of 'x' for which f(x) is maximum and find $f(x)_{max}$.*

(Dec.-2002, 8 Marks, Dec.-2004, 10 Marks)

Sol. : i) To obtain x where $f(x)_{max}$ lies

The given data is,

x	1	2	7	8
y = f(x)	4	5	5	4

The data points are not evenly spaced. Hence we have to use lagrange's interpolation method. We will use quadratic interpolation given by eq. 6.2.12. Therefore we will take the data points near maximum from above data.

Let $x_0 = 2$, $y_0 = 5$

$x_1 = 7$, $y_1 = 5$

$x_2 = 8$, $y_2 = 4$

Eq. 6.2.12 gives derivative $\frac{dy}{dx}$, i.e.

$$\frac{dy}{dx} = \frac{2x - x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)} y_0 + \frac{2x - x_0 - x_2}{(x_1 - x_0)(x_1 - x_2)} y_1 + \frac{2x - x_0 - x_1}{(x_2 - x_0)(x_2 - x_1)} y_2$$

$$= \frac{2x - 7 - 8}{(2-7)(2-8)} 5 + \frac{2x - 2 - 8}{(7-2)(7-8)} 5 + \frac{2x - 2 - 7}{(8-2)(8-7)} 4$$

$$= -2x + 9$$

At maximum value of 'y', $\frac{dy}{dx} = 0$. Hence,

$$-2x + 9 = 0 \Rightarrow x = 4.5$$

Thus at x = 4.5, 'y' will be maximum.

ii) To obtain y_{max} at x = 4.5

For this calculation we will use all the data points. i.e.,

x	$x_0 = 1$	$x_1 = 2$	$x_2 = 7$	$x_3 = 8$
y = f(x)	$y_0 = 4$	$y_1 = 5$	$y_2 = 5$	$y_3 = 4$

Now we have to find y at x = 4.5. Since the data points are not evenly spaced, we will use lagrange's interpolation. i.e.,

$$y_{max} = \frac{(4.5-2)(4.5-7)(4.5-8)}{(1-2)(1-7)(1-8)} 4 + \frac{(4.5-1)(4.5-7)(4.5-8)}{(2-1)(2-7)(2-8)} 5 + \frac{(4.5-1)(4.5-2)(4.5-8)}{(7-1)(7-2)(7-8)} 5 + \frac{(4.5-1)(4.5-2)(4.5-7)}{(8-1)(8-2)(8-7)} 4$$

$$= 6.0416667$$

Thus maximum value of y is 6.0416667 and it occurs at x = 4.5.

Exercise

1. Find the minimum value of the function from the following table :

x	0	1	2	3	4	5
f(x)	0	0.25	0	2.25	16.0	56.25

[May - 2001, 8 marks, Dec - 2000, 8 marks, Dec - 98, 8 marks]

Sol. : $y_{min} = 0.028319$ at x = 2.1561 using upto 3^{rd} order finite differences.

University Questions

1. Find the minimum, value of the function from the following table

x	0	1	2	3	4	5
f(x)	0.0	0.25	0	2.25	16.00	56.25

[Dec - 98, Dec - 2000, May - 2001]

2. From the following table, find x, correct to two decimal places for which y is maximum and also find the maximum value of y.

x	1.2	1.3	1.4	1.5	1.6
y	0.9320	0.9636	0.9855	0.9975	0.9996

[Dec - 2001]

3. Given the data f(1) =4, f(2) = 5, f(7) = 5 f(8) = 4, find the value of x for which f(x) is maximum and find $f(x)_{max}$. (Use Numerical method) **(Dec - 2004)**

6.5 Engineering Applications

Numerical differentiation is required in many engineering applications. For example, the voltage drop in the inductance is given as,

$$v_L = L\frac{di}{dt}$$

Here numerical differentiation of i gives the voltage drop in inductance. The current in the capacitor is given as,

$$i_C = C\frac{dv}{dt}$$

Here numerical differentiation of v gives current through the capacitor. Following example illustrate such application.

Ex. 6.5.1 *The current flowing through the inductance as a function of time is given below :*

time, t	0	0.1	0.2	0.3	0.5	0.7
current, i	0	0.15	0.3	0.55	0.8	1.9

Determine the voltage drop in an inductance of 4H at t = 0.3.

Sol. : The data is not evenly spaced. Hence we can use the differentiation methods for non-uniform nodal points. The derivative is required at $t = 0.3$. Hence let us consider the following part of the data.

time, x	$x_0 = 0.3$	$x_1 = 0.5$	$x_2 = 0.7$
current, y	$y_0 = 0.55$	$y_1 = 0.8$	$y_2 = 1.9$

Equation 6.2.12 gives derivative using quadratic interpolation. i.e.,

$$\frac{dy}{dx} = \frac{2\,x_0 - x_1 - x_2}{(x_0 - x_1)\,(x_0 - x_2)}\,y_0 + \frac{x_0 - x_2}{(x_1 - x_0)\,(x_1 - x_2)}\,y_1 + \frac{x_0 - x_1}{(x_2 - x_0)\,(x_2 - x_1)}\,y_2$$

This equation gives derivative at $x = x_0$.

Putting values in above equation,

$$\frac{dy}{dx} = \frac{2\,(0.3) - 0.5 - 0.7}{(0.3 - 0.5)\,(0.3 - 0.7)}\,(0.55) + \frac{0.3 - 0.7}{(0.5 - 0.3)\,(0.5 - 0.7)}\,(0.8) + \frac{0.3 - 0.5}{(0.7 - 0.3)\,(0.7 - 0.5)}\,(1.9)$$

$$= -\,0.875$$

Thus $\frac{di}{dt}$ at $t = 0.3$ is $-\,0.875$. Hence voltage drop in an inductance is,

$$\begin{aligned} v_L &= L\,\frac{di}{dt} \\ &= 4\,(-\,0.875) \\ &= -\,3.5 \text{ volts at } t = 0.3 \end{aligned}$$

Similarly, the voltage drop at other times can be obtained.

6.6 C-program and Algorithm for Numerical Differentiation

Now let us discuss an algorithm for numerical differentiation using interpolation.

6.6.1 Algorithm

Assumption : The data points are equally spaced. Differentiation is done at $x = x_0$, i.e. first data point.

Step 1 : Accept the number of data points

Step 2 : Accept the data points

Step 3 : Prepare the forward differences table and calculate $h = x_1 - x_0$.

Step 4 : Calculate derivative as follows :

$$\left[\frac{dy}{dx}\right]_{x=x_0} = \frac{1}{h}\left[\Delta y_0 - \frac{1}{2}\,\Delta^2 y_0 + \frac{1}{3}\,\Delta^3 y_0 - \frac{1}{4}\,\Delta^4 y_0 + \ldots\right]$$

Step 5 : Display the result $\frac{dy}{dx}$

Step 6 : Stop.

6.6.2 C-program

A C-program based on above algorithm is given below :

```
/* Download this program from  www.vtubooks.com          */
/* File name : nu_diffn.cpp                              */

/*--------------- NUMERICAL DIFFERENTIATION ---------------*/

/*   This program calculates numerical differentiation at x0 (i.e.
```

```
first value) of the input data using following equation.

    y'(x0) = {Dy0 - D2y0/2 + D3y0/3 - D4y0/4 + ...}/h

INPUTS :  Number of entries of the data.

OUTPUTS : Vaulue of derivative y' at x0.

ASSUMPTIONS: This program calculates derivative of the data
             at first value i.e. x = x0.                         */
/*---------------      PROGRAM  ---------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>

void main()
{
      double y[20][20],x[20],h,sum;
      int i,j,k,n;

      clrscr();
      printf("\n\t\tNUMERICAL DIFFERENTIATION");

      printf("\n\nEnter the number of entries (max 20) = ");
            /* ENTER THE NUMBER OF ENTRIES IN THE TABLE          */
      scanf("%d",&n);

      for(i = 0; i < n; i++)
      {          /*  LOOP TO GET x AND y = f(x) IN THE TABLE     */

     printf("x%d = ",i);   scanf("%lf",&x[i]);
     printf("              y%d = ",i);  scanf("%lf",&y[i][0]);
      }

      h =  x[1] - x[0];           /*   CALCULATE VALUE OF 'h'    */

      k = n;
         for(j = 1; j < n; j++)
         {     /*  LOOP TO CALCULATE FORWARD DIFFERENCES IN THE TABLE  */
     k = k - 1;
     for(i = 0; i < k; i++)
     {
        y[i][j] = y[i+1][j-1] - y[i][j-1];
     }
  }
```

```
      sum = 0;
      for(i = 1; i < n; i++)  /*THIS LOOP CALCULATES DERIVATIVE
                                AT x0 */
    {
  sum = sum + (y[0][i]/i)*(pow(-1,(i+1)));
    }
    sum = sum/h;

    printf("\nThe value of dy/dx at x0 = %lf is  %lf",
            x[0],sum);
}
/*------------------- END OF PROGRAM -------------*/
```

As shown in the above program, the first `for` loop accepts values of x and y. Then program calculates `h`. The next `for` loop calculates forward differences. The logic of calculating forward differences is discussed in previous chapter.

The next `for` loop calculates derivative as per following equation,

$$\left[\frac{dy}{dx}\right]_{x=x_0} = \frac{1}{h}\left\{\Delta y_0 - \frac{1}{2}\Delta^2 y_0 + \frac{1}{3}\Delta^3 y_0 - \frac{1}{4}\Delta^4 y_0 + \ldots\right\} \qquad \ldots (6.6.1)$$

Consider the statement in the `for` loop,

```
sum = sum+(y[0][i]/i)*(pow(-1,(i+1)));
```

This statement implements the part of equation 6.6.1 in { } brackets. `y[0][i]` contains forward differences $\Delta y_0, \Delta^2 y_0, \Delta^3 y_0, \ldots$ And `pow(-1,(i+1))` takes care of alternating sign of terms in equation 6.6.1. And lastly

```
sum = sum/h;
```

This statement implements remaining part of equation 6.6.1. After the above statement is executed,

$$\frac{dy}{dx} = \text{sum}$$

The derivative is the displayed on the screen.

To test this program :

Consider the data of example 6.2.2. This example calculates $\frac{dy}{dx}$ at $x = 1.2$. Hence the data of table 6.2.1 is to be considered as follows :

x	$x_0 = 1.2$	$x_1 = 1.4$	$x_2 = 1.6$	$x_3 = 1.8$	$x_4 = 2.0$	$x_5 = 2.2$
y	$y_0 = 3.3201$	$y_1 = 4.0552$	$y_2 = 4.9530$	$y_3 = 6.0496$	$y_4 = 7.3891$	$y_5 = 9.0250$

Run the program and enter the above data as input to the program. Then the results of the program are as follows :

```
--------------- Results ------------------

        NUMERICAL DIFFERENTIATION

Enter the number of entries (max 20) = 6

x0 = 1.2
        y0 = 3.3201
x1 = 1.4
        y1 = 4.0552
x2 = 1.6
        y2 = 4.9530
x3 = 1.8
        y3 = 6.0496
x4 = 2.0
        y4 = 7.3891
x5 = 2.2
        y5 = 9.0250

The value of dy/dx at x0 = 1.200000 is  3.320317
     ----------------------------------------------------------------
```

Observe the above program gives $\frac{dy}{dx} = 3.320317$. In example 6.2.2, we have obtained the same derivative as, $\frac{dy}{dx} = 3.3205$.

6.7 MATLAB for Differentiation

MATLAB can also be used for differentiation. A MATLAB program is given below to differentiate the given vector.

```
%    Download this program from www.vtubooks.com
%    file name : differentiation.m

%    This program determines differences between adjecent
%                             elements of a vector.  Matlab Version 6

%    Input : Data elements of a vector

%    Output: Differentiated vector

%-----------------------------------------------------------
clc                                         % clear the screen
disp('  Differentiation'); disp(' ');

y = input('Enter the vector y = ');         % Values of vector are
entered here

dy = diff(y);                               % Differentiate the vector

disp(' ');
disp('The differentiated vector is as follows...'); disp(' ');
disp(dy);                      % Display the differentiated the vector

%------------   End of the program -----------------
```

As shown above, the program accepts the vector `y` by the `input` function. Here it is expected that the data points x (though not entered) must be equally spaced. Then the next statement is,

```
dy = diff(y);
```

This statement differentiates the vector `y`. It calculates the differences between adjacent elements of a vector. The program then displays the result.

To test this program :

Consider the constantly increasing vector,

$$y = [1, 2, 3, 4, 5, 6, 7]$$

This vector has constant slope, which is $\frac{dy}{dx}$. Hence derivative will be,

$$\frac{dy}{dx} = [1, 1, 1, 1, 1, 1]$$

The above vector represents constant slope of y. This data is given as input to the program. Then the program generates following results :

```
%------------------------- Results ------------------------------------
    Differentiation

Enter the vector y = [1 2 3 4 5 6 7]

The differentiated vector is as follows...

     1     1     1     1     1     1
%----------------------------------------------------------------------
```

Observe that the same $\frac{dy}{dx}$ is generated by the program. Here note that the program assumes equally spaced arbitrary x. The `diff` function of MATLAB can be used to obtain the derivatives of the other functions also. For example let us define

```
y = sin (a*x);
```

Here y = sin (a x). This function can be differentiated by,

```
diff (y)
```

Then the program displays `cos(a*x)*a` as the derivative $\frac{dy}{dx}$.

Computer Exercise

1. Write the C program to obtain the derivative of a function with non uniform nodal points.
2. Write the C program to obtain the derivative of a function at a point which doesnot exists in the given data points. (Hint : Use generalized expression of interpolation).

❑❑❑

7
Numerical Integration

7.1 Introduction

Integration operation is required in large number of calculations. In this chapter we will study the methods of numerical integration. The function between the two points is approximated by a straight line, parabola or some higher order polynomial. Based on such approximation, there are trapezoidal rule, simpson's $\frac{1}{3}^{rd}$ rule and simpson's $\frac{3}{8}^{th}$ rule of integration. The complete range of the integration is broken into several small intervals. The numerical integration methods are derived on the basis of interpolation of a parabola or polynomial over such small sized intervals. These methods are discussed next.

Newton-Cotes Integration Formulas

Principle : The complicated function to be integrated is replaced by some approximate function. Such approximate function can be easily integrated.

Consider an integral,

$$I = \int_a^b f(x)dx$$

Here f(x) is approximated by some function or n^{th} order polynomial i.e.,

$$f(x) \approx a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1} + a_n x^n$$

Therefore the integration will be,

$$I = \int_a^b [a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1} + a_n x^n] dx$$

The function f(x) can be approximated by a straight line or parabola.

Closed forms : If the begining and ending limits of integration are included in data points, then it is called closed form of integration.

Open forms : If the beging and ending limits of integration are beyond the data points, then it is called open form of integration.

7.2 Trapezoidal Rule for Numerical Integration

7.2.1 Principle

Trapezoidal rule approximates the curve i.e. f(x) between the limits of integration by a straight line. In Fig. 7.2.1 observe that the curve f(x) is approximated by a straight line AB. The integration of f(x) is equal to shaded area under the line AB. This area is within the trapezoid formed by AB x_0 x_n. Hence the name trapezoidal rule is given.

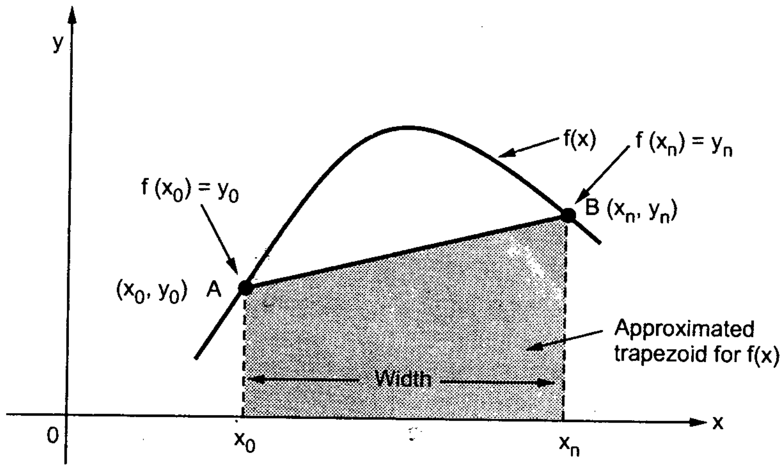


Fig. 7.2.1 Approximation of f(x) by straight line

7.2.2 Derivation of Trapezoidal Rule

The line AB can be represented as, f(x) ≈ [line AB] = $f(x_0)+\dfrac{f(x_n)-f(x_0)}{x_n-x_0}(x-x_0)$

Integrating above equation from x_0 to x_n we get,

$$I \approx \int_{x_0}^{x_n}\left[f(x_0)+\frac{f(x_n)-f(x_0)}{x_n-x_0}(x-x_0)\right]dx$$

$$\approx \int_{x_0}^{x_n}\left\{f(x_0)+\frac{f(x_n)-f(x_0)}{x_n-x_0}x-\frac{f(x_n)-f(x_0)}{x_n-x_0}x_0\right\}dx$$

$$\approx \int_{x_0}^{x_n}\left\{\frac{f(x_n)-f(x_0)}{x_n-x_0}x+f(x_0)-\frac{x_0 f(x_n)-x_0 f(x_0)}{x_n-x_0}\right\}dx$$

$$\approx \int_{x_0}^{x_n}\left\{\frac{f(x_n)-f(x_0)}{x_n-x_0}x+\frac{x_n f(x_0)-x_0 f(x_n)}{x_n-x_0}\right\}dx$$

$$\approx \frac{f(x_n)-f(x_0)}{x_n-x_0}\left[\frac{x^2}{2}\right]_{x_0}^{x_n}+\frac{x_n f(x_0)-x_0 f(x_n)}{x_n-x_0}[x]_{x_0}^{x_n}$$

$$\approx \frac{f(x_n)-f(x_0)}{x_n-x_0}\frac{1}{2}(x_n^2-x_0^2)+\frac{x_n f(x_0)-x_0 f(x_n)}{x_n-x_0}(x_n-x_0)$$

$$\approx f(x_n)-f(x_0)\cdot\frac{(x_n+x_0)}{2}+(x_n f(x_0)-x_0 f(x_n))$$

$$\approx (x_n-x_0)\cdot\frac{f(x_0)+f(x_n)}{2}$$

This is the formula for trapezoidal integration. Note that above equation indeed represents the area of trapezoid of Fig. 7.2.1. Area of the trapezoid is given as,

Area of trapezoid=Width × average height

From Fig. 7.2.1 we can write above equation as,

$$\text{Area of trapezoid} = (x_n-x_0)\cdot\frac{f(x_0)+f(x_n)}{2}$$

This is same as the earlier equation.

Since $y_0 = f(x_0)$ and $y_n = f(x_n)$ above equation will be,

$$\int_{x_0}^{x_n} y\,dx = (x_n-x_0)\cdot\frac{y_0+y_n}{2}$$

7.2.3 Composite Formula

We approximated the complete area under $f(x)$ by single trapezoid. Hence there is large area in the result. This error can be reduced by using trapezoids of smaller width. Fig. 7.2.2 shows this concept. In this figure observe that the complete integration interval $[x_0\ x_n]$ is divided into small intervals of equal width. Let this width be denoted by 'h'. i.e.,

$$h = x_5-x_4 = x_4-x_3 = x_3-x_2 = \ldots = x_1-x_0$$

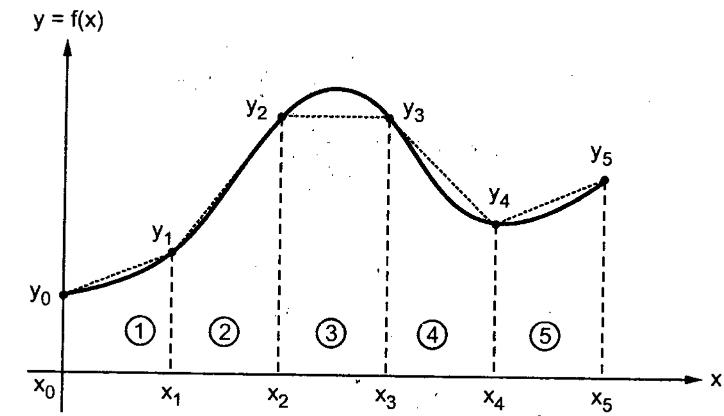


Fig. 7.2.2 Multiple application of composite formula

The total area under the curve f(x) will be equal to sum of areas of individual trapezoids. For above figure we can write,

$$\int_{x_0}^{x_n} y\,dx = \int_{x_0}^{x_1} y\,dx + \int_{x_1}^{x_2} y\,dx + \int_{x_2}^{x_3} y\,dx + \int_{x_3}^{x_4} y\,dx + \int_{x_4}^{x_5} y\,dx$$

$$= (x_1 - x_0)\,\frac{y_0 + y_1}{2} + (x_2 - x_1)\cdot\frac{y_1 + y_2}{2} + \cdots + (x_5 - x_4)\,\frac{y_4 + y_5}{2}$$

$$= h\cdot\frac{y_0 + y_1}{2} + h\cdot\frac{y_1 + y_2}{2} + \cdots + h\cdot\frac{y_4 + y_5}{2}$$

$$= \frac{h}{2}[y_0 + 2y_1 + 2y_2 + 2y_3 + 2y_4 + y_5]$$

$$= \frac{h}{2}[(y_0 + y_5) + 2(y_1 + y_2 + y_3 + y_4)]$$

This equation is derived for five trapezoids. If can be generalized as follows :

$$\boxed{\int_{x_0}^{x_n} y\,dx = \frac{h}{2}\,[(y_0 + y_n) + 2(y_1 + y_2 + y_3 + \ldots\ldots + y_{n-1})]} \qquad \ldots (7.2.1)$$

Here h is the spacing between individual values of x. i.e. $x_1 - x_0 = h$ = interval or $\frac{x_n - x_0}{h} = n$ = number of steps.

We can write the above formula as,

$$\int_{x_0}^{x_n} y\,dx = \frac{h}{2}\text{ [Sum of first and last ordinates + 2 (sum of remaining ordinates)]}$$

The error in this formula is given as

$$\boxed{E_n = -\frac{h^2}{12}(x_n - x_0)\,f''(z)} \qquad \ldots (7.2.2)$$

where $x_0 \le z \le x_n$

7.2.4 Solved Examples

Ex. 7.2.1 *Use trapezoidal rule with four steps to estimate the rule of integral.*

$$\int_0^2 \frac{x}{\sqrt{2 + x^2}}\,dx$$

Sol. : Since there are four steps,

$$n = 4$$

and we know that,

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

$\therefore \quad h = \frac{x_n - x_0}{n}$ i.e. spacing or interval of each step.

$\therefore \quad h = \frac{2-0}{4}$

$\therefore \quad h = 0.5$

Here, $\quad y = f(x) = \frac{x}{\sqrt{2+x^2}}$

Let's prepare the following table -

x	0	0.5	1	1.5	2
$y = f(x)$	0	0.33333	0.57735	0.727606	0.81645

Here $y_0 = 0$, $y_1 = 0.3333$, $y_2 = 0.57735$, $y_3 = 0.727606$ and $y_n = y_4 = 0.81645$

From equation 7.2.1 we have,

$$\int_{x_0}^{x_n} y\,dx = \frac{h}{2}[(y_0 + y_n) + 2(y_1 + y_2 + y_3 + \ldots.. y_{n-1})]$$

For $\quad n = 4$

$$\therefore \quad \int_0^2 y\,dx = \frac{h}{2}[(y_0 + y_4) + 2(y_1 + y_2 + y_3)]$$

Putting the values,

$$\int_0^2 \frac{x}{\sqrt{2+x^2}}\,dx = \frac{0.5}{2}[(0 + 0.81649) + (0.33333 + 0.57735 + 0.727606)] = 1.023267$$

Ex. 7.2.2 *Evaluate* $\int_4^{5.2} ln\ x\ dx$ *using trapezoidal rule. Take* $h = 0.2$.

Sol. : Here $\quad y = ln\ x$

Let's prepare the table of values of x and y.

x	$x_0 = 4$	$x_1 = 4.2$	$x_2 = 4.4$	$x_3 = 4.6$	$x_4 = 4.8$	$x_5 = 5.0$	$x_6 = 5.2$
$y = ln\ x$	$y_0 = 1.3863$	$y_1 = 1.4351$	$y_2 = 1.4816$	$y_3 = 1.5260$	$y_4 = 1.5686$	$y_5 = 1.6094$	$y_6 = 1.6486$

From equation 7.2.1 trapezoidal rule is given as,

$$\int_{x_0}^{x_n} y\,dx = \frac{h}{2}[(y_0 + y_n) + 2(y_1 + y_2 + y_3 + \ldots + y_{n-1})]$$

Here $n = 6$ hence above formula becomes,

$$\int_{x_0}^{x_6} y\,dx = \frac{h}{2}[(y_0 + y_6) + 2(y_1 + y_2 + y_3 + y_4 + y_5)]$$

$$= \frac{0.2}{2}\begin{bmatrix} (1.3863 + 1.6486) \\ +2\ (1.4351 + 1.4816 + 1.5260 + 1.5686 + 1.6094) \end{bmatrix} = 1.82764$$

Ex. 7.2.3 *Solve* $\int_0^1 \frac{\sin x}{x} dx$ *using trapezoidal rule and write a program for the same.*
[May-2004, 10 Marks, Dec-2002, 8 Marks]

Sol. : Here the number of steps are not given. Hence let us take 6 steps. Therefore 'h' will be,

$$h = \frac{1-0}{6} = \frac{1}{6}$$

Let us prepare the table of values of x and y.

x	0	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{3}{6}$	$\frac{4}{6}$	$\frac{5}{6}$	1
$y = \frac{\sin x}{x}$	1	0.9953	0.9815	0.9588	0.9275	0.8882	0.8414

$$\therefore \quad \int_0^1 \frac{\sin x}{x} dx = \frac{1/6}{2}[(1+0.8414)+2(0.9953+0.9815+0.9588+0.9275+0.8882)]$$

$$= 0.94533$$

Exercise

1. *Computer* $\int_1^2 \frac{dx}{x}$ *using trapezoidal rule for* $h = 0.25$. **[Ans. : 0.6970238]**

2. *Find the integral* $\int_0^{0.8} e^{x^2} dx$ *by trapezoidal rule. Take* $h = 0.1$. **[Ans. : 1.012]**

3. *Evaluate* $\int_0^1 \frac{dx}{1+x}$ *by using trapezoidal rule. Take* $h = 0.125$. **[Ans. : 0.694122]**

7.2.5 C - Program and Algorithm

Based on the illustrative problems, we will now develop an algorithm for trapezoidal rule.

Algorithm :

Assumption : The function to be integrated is predefined.

Step 1 : Read the lower and upper limits of integration.

Step 2 : Read value of 'h' or Read number of intervals 'n'.

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

Step 3 : Calculate

$$y_0 = f(x)\big|_{x=x_0} \quad y_1 = f(x)\big|_{x=x_1} \quad y_2 = f(x)\big|_{x=x_2} \text{ and so on.}$$

Step 4 : Calculate

$$I = \frac{h}{2}[(y_0 + y_n) + 2(y_1 + y_2 + y_3 + \ldots y_{n-1})]$$

Step 5 : Display the value of result of integration 'I' on the screen.

Step 6 : Stop.

Flowchart :

The flowchart of trapezoidal rule is given in Fig. 7.2.3. This flowchart also assumes that the function to be integrated is predefined.

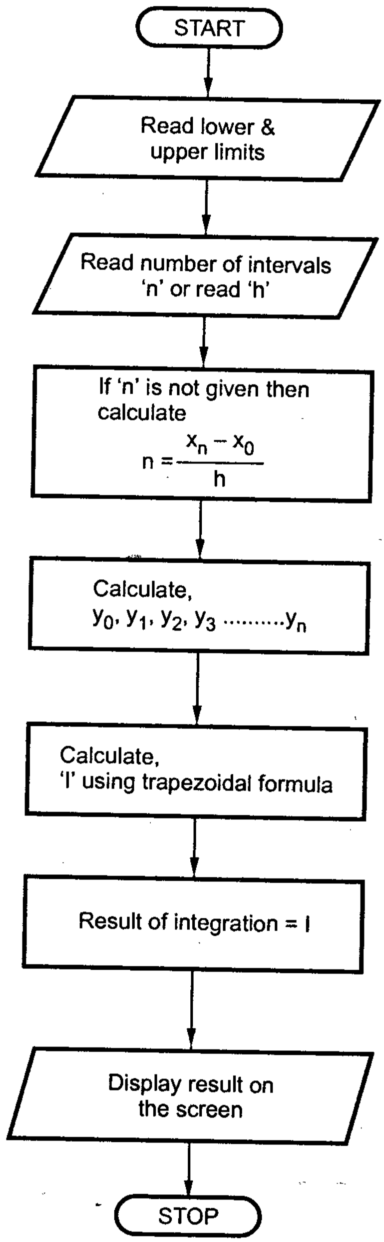


Fig. 7.2.3 Flowchart of trapezoidal rule

Computer program :

Based on the algorithm and flowchart of Fig. 7.2.3, the 'C' program for trapezoidal rule is given below.

```
/* Download this program from  www.vtubooks.com                             */
/* File name : trapez.cpp                                                    */

/*---------------- TRAPEZOIDAL RULE OF INTEGRATION ------------------*/

/*      THIS PROGRAM CALCULATES THE VALUE OF INTEGRATION USING
        TRAPEZIODAL  RULE. THE FUNCTION TO BE INTEGRATED IS,

                     f(x) = 1/x

        INPUTS :  1) Lower and upper limits of integration.

                  2) Number of intervals.

        OUTPUTS : Result of integration.                                     */

/*------------------------------  PROGRAM  -------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>

void main()
{
      double fx (double x0);

      double lo,up,f[20],h,x0,sum,result;
      int i,n;

      clrscr();
      printf("\n\t  TRAPEZOIDAL RULE OF INTEGRATION");

      printf("\n\nEnter the lower limit of integration = ");
      scanf("%lf",&lo);          /* ENTER LOWER LIMIT OF INTEGRATION       */
      printf("\n\nEnter the upper limit of integration = ");
      scanf("%lf",&up);          /* ENTER UPPER LIMIT OF INTEGRATION       */

      printf("\n\nEnter the value of h = ");
      scanf("%lf",&h);             /*    ENTER THE VALUE OF h              */

      n = (up - lo)/h;            /*   CALCULATION VALUE OF n              */
      x0 = lo;

      for(i = 0; i <= n; i++)     /* LOOP TO CALCULATE VALUE OF f(x)      */
      {
          f[i] = fx(x0);           /*    FUNCTION fx IS CALLED HERE       */
          x0 = x0 + h;       /* NEXT VALUE OF x IS CALCULATED HERE        */
      }
      sum = 0;
      for(i = 1; i <= n-1; i++)
      {
        sum = sum + 2*f[i]; /*SUMMATION OF ORDINATES FROM y(1) to y(n-1)*/
      }
      result = (h/2) * ( f[0] + f[n] + sum ); /* RESULT OF INTEGRATION  */

      printf("\n\nThe result of integration is = %lf",result);
 }

 double fx ( double x)       /* FUNCTION TO CALCULATE VALUE OF f(x)  */
 {
     double f;
     f = 1/x;
     return(f);
 }
 /*---------------------- END OF PROGRAM -----------------------------*/
```

The first statement in `main` function is,

```
double fx (double x0);
```

This is the declaration of the function to calculate $f(x)$. In this program the function used for integration is,

$$f(x) = \frac{1}{x}$$

The program first displays the name of the method and then takes values of upper and lower limits. It then takes the value of 'h'.

Observe the statement to calculate value of 'n',

```
n = (up - lO)/h;
```

This statement is the implementation of,

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

Next there is a `for` loop to calculate

$f(x)$ at $x = x_0, x_1, x_2, \ldots x_n$.

In the `for` loop the first statement is,

```
f[i] = fx(x0);
```

This statement calls function `fx` to calculate value of $f(x)$. `x0` is the value passed to the function. Function calculates $f(x)$ at `x=x0` and returns this values to the above statement. `F[i]` stores the values of $f(x)$.

`f[i]` is on array going from `i=0 to n`.

The next statement in the `for` loop is,

```
x0 = x0+h
```

This statement adds 'h' to `x0` every time `i` is incremented in the loop. Thus,

`x0` is not equal to x_0.

The next `for` loop is an implementation of trapezoidal rule. The last `printf` statement prints the result of integration.

Then there is the listing of function `fx`. This function calculates $f(x)$ by,

`f=1/x` ; Here $f = f(x)$

The value of `f` is returned to main program.

How to Run this program ?

Compile and make EXE file of this source code. To illustrate this program we will evaluate,

$$\int_1^2 \frac{1}{x}\, dx \text{ for } h = 0.25$$

Run the program on your computer. The program first displays, `Enter lower limit of integration` = Here enter '1' and press 'enter'

Then it displays,

`Enter upper limit of integration` = Here enter '2' and press 'enter'

Then it displays,

`Enter the value of h` = Here enter 0.25 and press 'enter'

The program then displays the result of integration.

Combined display of results is shown below -

```
---------------------------- Results -----------------------------------

         TRAPEZOIDAL RULE OF INTEGRATION

Enter the lower limit of integration = 1


Enter the upper limit of integration = 2


Enter the value of h = 0.25


The result of integration is = 0.697024
-------------------------------------------------------------------------
```

University Questions

1. Derive the formula for Trapezoidal Rule for numerical Integration and Evaluate - **[Dec - 2003]**

$$\int_0^6 \frac{1}{1+x} dx$$

2. *Solve* $\int_0^1 \frac{\sin x}{x} dx$ *using trapezoidal rule and write a program for the same.* **[May - 2004]**

7.3 Simpson's $\frac{1}{3}$ Rule

7.3.1 Principle

In trapezoidal rule we approximate the curve by straight line between two points as shown in Fig. 7.2.1. To be more accurate a parabola can be used. One more point is taken at the center of (x_0, y_0) and (x_n, y_n). These three points are then approximated by a parabola. The area under this parabola approximately gives the integration of the function. Fig. 7.3.1 illustrates this.

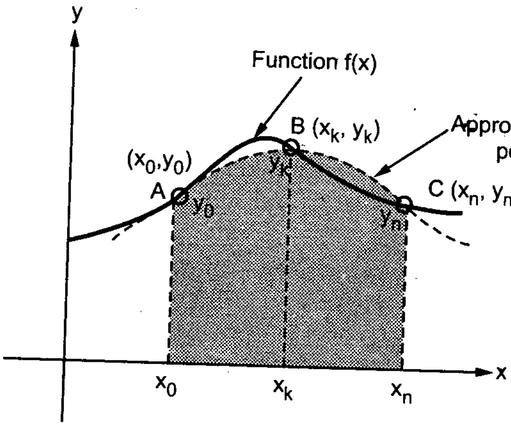


Fig. 7.3.1 Approximation of f(x) by parabola

7.3.2 Derivation of Simpson's $\frac{1}{3}$ Rule

Let the three points be (x_0, y_0), (x_k, y_k) and (x_n, y_n). We can use lagrange's interpolation to determine the second degree parabola. i.e.,

$$f(x) \approx \frac{(x-x_k)(x-x_n)}{(x_0-x_k)(x_0-x_n)} y_0 + \frac{(x-x_0)(x-x_n)}{(x_k-x_0)(x_k-x_n)} y_k + \frac{(x-x_0)(x-x_k)}{(x_n-x_0)(x_n-x_k)} y_n$$

Integration of above equation gives the required formula. i.e.,

$$\int_{x_0}^{x_n} y\,dx = \int_{x_0}^{x_n} \left\{ \frac{(x-x_k)(x-x_n)}{(x_0-x_k)(x_0-x_n)} y_0 + \frac{(x-x_0)(x-x_n)}{(x_k-x_0)(x_k-x_n)} y_k + \frac{(x-x_0)(x-x_k)}{(x_n-x_0)(x_n-x_k)} y_n \right\} dx$$

Integrating and simplifying the above equation,

$$\int_{x_0}^{x_n} y\,dx = \frac{x_n - x_0}{6} [y_0 + 4y_1 + y_2]$$

Let 'h' be the distance between two points. i.e.,

$$h = x_n - x_k = x_k - x_0$$

$$\therefore \quad x_n - x_0 = 2h$$

$$\therefore \quad \int_{x_0}^{x_n} y\,dx = \frac{h}{3} [y_0 + 4y_1 + y_2] \qquad \ldots (7.3.1)$$

The name simpson's $\frac{1}{3}$ rule is given since 'h' is divided by 3.

7.3.3 Composite Formula

The result given by eq 7.3.1 is quite approximate, since only one parabola approximates the complete range of integration. Hence many such parabolas are approximated over the complete range of integration. Each approximated parabola is then integrated independently. The individual results of integration are added to give the final result. i.e.,

$$\int_{x_0}^{x_n} y\,dx = \int_{x_0}^{x_2} y\,dx + \int_{x_2}^{x_4} y\,dx + \int_{x_4}^{x_6} y\,dx + \cdots + \int_{x_{n-2}}^{x_n} y\,dx$$

$$= \frac{h}{3}[y_0 + 4y_1 + y_2] + \frac{h}{3}[y_2 + 4y_3 + y_4] + \frac{h}{3}[y_4 + 4y_5 + y_6] + \cdots + \frac{h}{3}[y_{n-2} + 4y_{n-1} + y_n]$$

$$= \frac{h}{3}[y_0 + 4(y_1 + y_3 + y_5 + \cdots + y_{n-1}) + 2(y_2 + y_4 + y_6 + \cdots + y_{n-2}) + y_n]$$

Rearranging the above equation,

$$\boxed{\int_{x_0}^{x_n} y\,dx = \frac{h}{3}[(y_0 + y_n) + 4(y_1 + y_3 + y_5 + \ldots) + 2(y_2 + y_4 + y_6 + \ldots)]} \quad \ldots (7.3.2)$$

Here 'n' is the number of intervals taken over (x_0, x_n). 'h' is the spacing between two nearby points. They are related as,

$$n = \frac{x_n - x_0}{h} \quad \ldots (7.3.3)$$

The above integration formula can be written as,

$$\int_{x_0}^{x_n} y\,dx = \frac{h}{3}\begin{bmatrix} (Sum\ of\ first\ and\ last\ ordinates) \\ +4\ (Sum\ of\ odd\ ordinates) \\ +\ 2\ (Sum\ of\ even\ ordinates) \end{bmatrix}$$

This formula is used when there are even number of segments and an odd number of points.

An error in the simpson's $\frac{1}{3}$ rule is given as,

$$\boxed{E_n = \frac{(x_n - x_0)^5}{180\,n^4} f^{(4)}(z)} \quad \ldots (7.3.4)$$

Here $f^{(4)}(z)$ is the fourth derivative of $f(x)$ taken at $x_0 \le z \le x_n$.

7.3.4 Solved Examples

Ex. 7.3.1 *Evaluate*

$$\int_0^{\frac{3\pi}{20}} (1 + 2\sin x)\,dx$$

using simpson's $\frac{1}{3}$ rule. Take 4 segments. Compare this value with analytical result and calculate percentage relative error.

Sol. : Here $x_0 = 0$ and $x_n = \frac{3\pi}{20}$ or 27°.

Since there are 4 segments ; means n = 4.

$\therefore \qquad h = \frac{x_n - x_0}{n} = \frac{27° - 0}{4} = 6.75°$

Here $\qquad f(x) = 1 + 2 \sin x$

Let's calculate x and $f(x)$ at following points.

x	$x_0 = 0$	$x_1 = 6.75°$	$x_2 = 13.5°$	$x_3 = 20.25°$	$x_2 = 27°$
$y = f(x)$	$y_0 = 1$	$y_1 = 1.2350748$	$y_2 = 1.4668907$	$y_3 = 1.6922341$	$y_4 = 1.907981$

By Simpson's $\frac{1}{3}$ rule given by equation 7.3.2 we have,

$$\int_{x_0}^{x_n} y\, dx = \frac{h}{3}\left[(y_0 + y_n) + 4(y_1 + y_3 + y_5 + \ldots) + (y_2 + y_4 + y_6 + \ldots)\right]$$

Here $n = 4$. Hence the above equation becomes,

$$\int_{x_0}^{x_4} y\, dx = \frac{h}{3}\left[(y_0 + y_4) + 4\,(y_1 + y_3) + 2\,(y_2)\right]$$

Putting the values of y in the above equation we get,

$$h = 6.75° = \frac{3\pi}{80}$$

$$\therefore \quad \int_0^{\frac{3\pi}{20}} y\, dx = \frac{3\pi/80}{3}\left[(1 + 1.907981 + 4\,(1.2350748 + 1.6922341) + 2 \times 1.466890)\right]$$

$$= 0.689226$$

The actual value is given as,

$$\int_0^{3\pi/20} (1 + 2 \sin)\, dx = [x - 2 \cos x]_0^{3\pi/20}$$

$$= \left(\frac{3\pi}{20} - 2 \cos \frac{3\pi}{20}\right) - (0 - 2) = 0.6892258$$

$$\therefore \quad \%\ \text{Relative error} = \frac{|\text{Actual value} - \text{Approximate value}|}{|\text{Actual value}|} \times 100$$

$$= 2.18 \times 10^{-5}\ \%$$

Ex. 7.3.2 *Calculate the value of h such that the integral $\int_0^1 x\, e^x\, dx$ using simpson's $\frac{1}{3}$ rule is correct upto three significant digits after decimal point.*

Sol. : The error in Simpson's $\frac{1}{3}$ rule is given as

$$E_n = -\frac{(x_n - x_0)^5}{180\, n^4} f^{(4)}(z) \quad \text{from equation 7.3.4}$$

$$f(x) = xe^x$$
$$f'(x) = xe^x + e^x = e^x(x+1)$$
$$f''(x) = (x+1)\,e^x + e^x = e^x(x+2)$$
$$f'''(x) = (x+2)\,e^x + e^x = e^x(x+3)$$

and $$f^{(4)}(x) = (x+3)\,e^x + e^x = e^x(x+4)$$

$\therefore$ $$f^4(z) = e^z(z+4) \text{ and } x_0 \le z \le x_n$$

Putting the value of 4^{th} derivative in error equation,

$$E_n = -\frac{(x_n - x_0)^5}{180\,n^4}\,e^z\,(z+4)$$

Max $\left|e^z(z+4)\right|$ will occur if $z = x_n$ i.e. $z = 1$.

$\therefore$ $$Max\left|e^z(z+4)\right| = 5e \qquad \text{only in interval } (0, 1)$$

We have to calculate maximum value of E_n.

Here $$x_0 = 0, \quad x_n = 1 \text{ and } n = \frac{x_n - x_0}{h} = \frac{1-0}{h} = \frac{1}{h}$$

Putting these values in equation of E_n,

$$E_n = -\frac{(1-0)^5}{180\left(\frac{1}{h}\right)^4}\,e^z\,(z=4)$$

$$\therefore \text{ Maximum value of } E_n = \frac{(1-0)^5}{180\left(\frac{1}{h}\right)^4}\,5\,e = 0.0755078\,h^4$$

The answer is required to be correct to 3 digits after decimal point. Means error in the answer is 0.0001 [Here note that 3 digits are zero in error].

$\therefore$ E_n should be less than 0.0001

or $$E_n < 0.0001$$

$\therefore$ $$0.0755078\,h^4 < 0.0001$$

$\therefore$ $$h < 0.1907664$$

This is the required value of h.

Ex. 7.3.3 *Evaluate* $I = \int_1^2 \frac{dx}{x^2}$ *by dividing the interval into equally spaced intervals of width (i) 0.5 and (ii) 0.25 and fitting quadratics through the set of three points. Compare the result with exact value.*

Sol. : Since quadratic interpolation is asked, we should use simpson's $\frac{1}{3}$ rule of intergration.

(i) Taking h = 0.5

We have $h = 0.5$, $y = \frac{1}{x^2}$. Let's prepare the table

x	$x_0 = 1$	$x_1 = 1.5$	$x_2 = 2$
$y = \frac{1}{x^2}$	$y_0 = 1$	$y_1 = 0.04444$	$y_2 = 0.25$

By Simpson's $\frac{1}{3}$ rule of equation 7.3.2 we have,

$$\int_1^2 \frac{1}{x^2}\, dx = \frac{h}{3}\left[y_0 + y_2 + 4(y_1)\right]$$

Putting the values,

$$\int_1^2 \frac{1}{x^2}\, dx = \frac{0.5}{3}\left[1 + 0.25 + 4 \times 0.4444\right]$$

$$= 0.50463$$

(ii) Taking h = 0.25

Lets prepare the table

x	$x_0 = 1$	$x_1 = 1.25$	$x_2 = 1.5$	$x_3 = 1.75$	$x_4 = 2.00$
$y = \frac{1}{x^2}$	$y_0 = 1$	$y_1 = 0.64$	$y_2 = 0.4444$	$y_3 = 0.3265$	$y_4 = 0.25$

By simpson's $\frac{1}{3}$ rule of equation 7.3.2 we have,

$$\int_1^2 \frac{1}{x^2}\, dx = \frac{h}{3}\left[y_0 + y_4 + 4\,(y_1 + y_3) + 2(y_2)\right]$$

$$= \frac{0.25}{3}\left[1 + 0.25 + 4\,(0.64 + 0.3265) + 2 \times 0.4444\right]$$

$$= 0.50041$$

(iii) Exact value of integration

$$\int_1^2 \frac{1}{x^2}\, dx = \left[-\frac{1}{x}\right]_1^2 = 0.5$$

Results :

Exact value $\to$ 0.5

with h = 0.5 $\to$ 0.50463

with h = 0.25 $\to$ 0.50041

Thus as the spacing between values of x is small, integration is more accurate.

Ex. 7.3.4 *A curve is drawn to pass through the points given by the following table :*

x	1	1.5	2	2.5	3	3.5	4
y	2	2.4	2.7	2.8	3	2.6	2.1

Estimate the area bounded by the curve, the x-axis and ordinates $x = 1$, $x = 4$.

Sol. : Let us designate the values in the table as follows :

x	$x_0 = 1$	$x_1 = 1.5$	$x_2 = 2$	$x_3 = 2.5$	$x_4 = 3$	$x_5 = 3.5$	$x_6 = 4$
y	$y_0 = 2$	$y_1 = 2.4$	$y_2 = 2.7$	$y_3 = 2.8$	$y_4 = 3$	$y_5 = 2.6$	$y_6 = 2.1$

We have to calculate the area under the curve 'y' from $x = 1$ to 4. It is equivalent to integrating 'y' from $x = 1$ to 4. i.e.

$$\text{Area} \qquad A = \int_1^4 y\,dx$$

Here we have $x_0 = 1$ and $x_6 = 4$. There are $n = 6$ intervals. Hence h will be,

$$h = \frac{x_6 - x_0}{n} = \frac{4-1}{6} = 0.5$$

Here let us use simpsón's $\frac{1}{3}$ *rd* rule for $n = 6$. i.e.,

$$\int_{x_0}^{x_6} y\,dx = \frac{h}{3}\left[y_0 + y_6 + 4(y_1 + y_3 + y_5) + 2(y_2 + y_4)\right]$$

$$= \frac{0.5}{3}\left[2 + 2.1 + 4(2.4 + 2.8 + 2.6) + 2(2.7 + 3)\right]$$

$$= 7.7833333$$

Ex. 7.3.5 *The velocity of the car (running on a straight road) at intervals of 2 minutes are given below.*

Time, min	0	2	4	6	8	10	12
Velocity, km/hr	0	22	30	27	18	7	0

Find the distance covered by the car. Justify for the method used.

[Dec - 2001, 8 Marks, Dec - 2003, 8 Marks]

Sol. : Here there are 7 data points, i.e. odd. Hence we will use simpson's $\frac{1}{3}$ rd rule.

The given time is in minutes. Let us convert it to hours and rewrite the data as follows :

t hours	$t_0 = \frac{0}{60}$	$t_1 = \frac{2}{60}$	$t_2 = \frac{4}{60}$	$t_3 = \frac{6}{60}$	$t_4 = \frac{8}{60}$	$t_5 = \frac{10}{60}$	$t_6 = \frac{12}{60}$
v km/hr	$v_0 = 0$	$v_1 = 22$	$v_2 = 30$	$v_3 = 27$	$v_4 = 18$	$v_5 = 7$	$v_6 = 0$

Let r be the distance covered by the car.

Then we can write,

$$v = \frac{dr}{dt}$$

i.e., velocity is given as derivative of distance with respect to time. We can write above equation as,

$$dr = v\,dt$$

Integrating both the sides,

$$\int dr = \int v\,dt$$

Here $\int dr$ is the distance covered by the car. Equation 7.3.2 gives the simpson's $\frac{1}{3}$ rd rule,

$$\int_{x_0}^{x_n} y\,dx = \frac{h}{3}\left\{(y_0 + y_n) + 4(y_1 + y_3 + y_5 + \ldots) + 2(y_2 + y_4 + y_6 + \ldots)\right\}$$

Putting values in this equation,

$$\text{Distance} = \frac{h}{3}\left\{(v_0 + v_6) + 4(v_1 + v_3 + v_5) + 2(v_2 + v_4)\right\}$$

Here $h = t_2 - t_1 = \frac{4}{60} - \frac{2}{60} = \frac{2}{60}$. Hence above equation becomes,

$$\text{Distance} = \frac{2/60}{3}\left\{(0+0) + 4(22+27+7) + 2(30+18)\right\}$$

$$= 3.555556 \text{ km}$$

Thus the car will cover 3.555556 km of distance.

Justification for the method used :

The given data has 7 (odd) data points and 6 (even) segments. Hence simpson's $\frac{1}{3}$ *rd* rule is suitable for such data.

Ex. 7.3.6 *Use simpson's $\frac{1}{3}$ rd and trapezoidal rules to evaluate $\int_0^{12} \log_e\left(1+x^2\right) dx$ by taking suitable number of intervals. Comment on the result.* **[May - 2001, 8 marks]**

Sol. : (i) Using simpson's $\frac{1}{3}$ rule :

For simpson's $\frac{1}{3}$ rd rule, we know that even number of segments are used. Hence let us use $n = 6$ segments. Here $x_0 = 0$ and $x_n = 12$. Hence,

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

$$= \frac{12-0}{6} = 2$$

Let us form the table of data points as follows :

x	$x_0 = 0$	$x_1 = 2$	$x_2 = 4$	$x_3 = 6$	$x_4 = 8$	$x_5 = 10$	$x_6 = 12$
$y = log_e\left(1+x^2\right)$	$y_0 = 0$	$y_1 = 1.6094$	$y_2 = 2.8332$	$y_3 = 3.6109$	$y_4 = 4.1743$	$y_5 = 4.6151$	$y_6 = 4.9767$

Simpson's $\frac{1}{3}$ rd formula is given by equation 7.3.2 as,

$$\int_{x_0}^{x_n} y\ dx = \frac{h}{3}\left\{(y_0 + y_n) + 4\ (y_1 + y_3 + y_5 + \ldots) + 2\ (y_2 + y_4 + y_6 + \ldots)\right\}$$

For $n = 6$, above equation becomes,

$$\int_{x_0}^{x_6} y\ dx = \frac{h}{3}\left\{(y_0 + y_6) + 4\ (y_1 + y_3 + y_5) + 2\ (y_2 + y_4)\right\}$$

Putting values in above equation,

$$\int_0^{12} log_e\left(1+x^2\right)dx = \frac{2}{3}\{(0+4.9767)+4(1.6094+3.6109+4.6151)+2(2.8332+4.1743)\}$$

$$= 38.888867$$

(ii) Using trapezoidal rule :

Let us use the same number of segments as in previous case. Trapezoidal rule is given by equation 7.2.1 as,

$$\int_{x_0}^{x_n} y\,dx = \frac{h}{2}\left\{(y_0 + y_n) + 2\ (y_1 + y_2 + y_3 + y_4 + \ldots + y_{n-1})\right\}$$

For $n = 6$ above equation will be,

$$\int_{x_0}^{x_6} y\ dx = \frac{h}{2}\left\{(y_0 + y_6) + 2\ (y_1 + y_2 + y_3 + y_4 + y_5)\right\}$$

Putting values in above equation,

$$\int_0^{12} log_e\left(1+x^2\right)dx = \frac{2}{2}\left\{(0+4.9767)+2\ (1.6094+2.8332+3.6109+4.1743+4.6151)\right\}$$

$$= 38.6625$$

Thus we obtained the result as,

Method	Answer
Trapezoidal rule	38.6625
Simpson's $\frac{1}{3}$ rule	38.888867

The actual value of integration is 38.6961. Here note that trapezoidal method has less error compared to simpson's $\frac{1}{3}$ method.

Exercise

1. *Evaluate* $\int_0^6 \frac{dx}{1+x}$ *by simpson's* $\frac{1}{3}$ *rule dividing into 10 sub intervals.*

   Obtain error bounds.

   [Hint : For error bound calculate minimum and maximum values of E_n]

   [Ans. : 1.94836]

2. *Evaluate* $\int_0^1 \sqrt{\sin x + \cos x}\, dx$ *dividing into 6 sub-intervals.*

   [Ans. : 1.1398]

3. *Evaluate the integral* $\int_0^9 \frac{1}{1+x} dx$ *using simpson's* $\frac{1}{3}$ *rule. Take* $h = 1$.

   And hence find the value of ln 10.

   [Hint : $\int_0^9 \frac{1}{1+x} dx = [\ln(1+x)]_0^9$

   $$= \ln(1+9) - \ln(1+0)$$
   $$= \ln 10$$

   Thus $\ln 10 = \int_0^9 \frac{1}{1+x} dx$]

4. *A function* $f(t)$ *is described by at equally spaced intervals.*

t	1	1.5	2	2.5	3	3.5	4
$f(t)$	2	2.4	2.7	2.8	3	2.6	2.1

   Evaluate $I = \int_1^4 f(t)\, dt$

   [Ans. : 7.8375]

5. *Approximate the integral below using the composite trapezoidal rule and simpson's rule using 8 equal segments.* $\int_{-1}^{1} \frac{1}{1+x^2} dx.$

   [May - 2000, 8 marks, Dec - 97, 10 marks]

[Hint : Here composite trapezoidal or simpson's rules means multiple application of these rules. Equation 7.2.1, equation 7.3.2 or equation 7.4.1 are composite formulae. Since there are 8 segments, use simpson's $\frac{1}{3}$ rd rule.

Ans. : Trapezoidal rule, 1.565588 Simpson's $\frac{1}{3}$ rd rule, 1.570784

7.3.5 Algorithm and C Program

Based on the discussion of the method and illustrative problems we will prepare an algorithm for simpson's $\frac{1}{3}$ rule as follows :

Algorithm :

Assumption : The function which is to be integrated is predefined in the program.

Step 1 : Read the lower and upper limits of integration.

Step 2 : Read value of spacing 'h' or read number of intervals to be taken i.e. n.

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

Step 3 : Calculate

$$y_0 = f(x)\big|_{x=x_0}, \quad y_1 = f(x)\big|_{x=x_1},$$
$$y_2 = f(x)\big|_{x=x_2}, \quad \textit{and so on.}$$

Step 4 : Calculate

$$I = \frac{h}{3}\left[y_0 + y_n + 4(y_1 + y_3 + y_5 + \ldots) + 2(y_2 + y_4 + y_6 + \ldots)\right]$$

Step 5 : Display the value of result of integration 'I' on the screen.

Step 6 : Stop.

Flowchart :

Fig. 7.3.2 shows a simplified flowchart for simpson's $\frac{1}{3}$ rule of integration. It is assumed that the function to be integrated is predefined.

Please refer Fig. 7.3.2 on page No. 458.

Computer program for simpson's $\frac{1}{3}$ rule :

The C program source code for simpson's $\frac{1}{3}$ rule is given below -

```
/* Download this program from  www.vtubooks.com                    */
/* File name : simpl_3.cpp                                          */

/*----------------- SIMPSON'S 1/3 RULE OF INTEGRATION -----------------*/
```

```
/*      THIS PROGRAM CALCULATES THE VALUE OF INTEGRATION USING
        SIMPSON'S 1/3 RULE. THE FUNCTION TO BE INTEGRATED IS,

                        f(x)  = 1/(1+x)

        INPUTS :  1) Lower and upper limits of integration.

                  2) Number of intervals.

        OUTPUTS : Result of integration.                              */

/*------------------------------   PROGRAM  --------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>

void main()
{
      double fx (double x0);      /*   DECLARATION OF A FUNCTION fx    */

      double lo,up,f[20],h,x0,sum,result.
      int i,n;

      clrscr();
      printf("\n\t    SIMPSON'S 1/3 RULE OF INTEGRATION");

      printf("\n\nEnter the lower limit of integration = ");
      scanf("%lf",&lo);        /* ENTER LOWER LIMIT OF INTEGRATION      */
      printf("\n\nEnter the upper limit of integration = ");
      scanf("%lf",&up);        /* ENTER UPPER LIMIT OF INTEGRATION      */

      printf("\n\nEnter the value of h = ");
      scanf("%lf",&h);            /*    ENTER THE VALUE OF h            */

      n = (up - lo)/h;            /* CALCULATION VALUE OF n i.e.STRIPS  */
      x0 = lo;

      for(i = 0; i <= n; i++)     /* LOOP TO CALCULATE VALUE OF f(x)    */
      {
          f[i] = fx(x0);          /*    FUNCTION fx IS CALLED HERE      */
          x0 = x0 + h;      /* NEXT VALUE OF x IS CALCULATED HERE       */
      }
      sum = 0;
      for(i = 1; i <= n-1; i = i + 2)
      {
        sum = sum + 4*f[i];  /* THIS IS  sum = 4 * ( odd ordinates )    */
      }
      for(i = 2; i <= n-1; i = i + 2)
      {
        sum = sum + 2*f[i]; /* THIS IS  sum = 2 * ( even ordinates )    */
      }
      result = (h/3) * ( f[0] + f[n] + sum );

                /*    Result = (h/3) * (4 * sum of odd ordinates
                                    + 2 * sum of even odinates )        */

      printf("\n\nThe result of integration is = %lf",result);
}

double fx ( double x)       /* FUNCTION TO CALCULATE VALUE OF f(x)  */
{
    double f;
    f = 1/(1+x);            /*     function f(x) = 1(1 + x)         */
    return(f);
}
/*----------------------- END OF PROGRAM ----------------------------*/
```

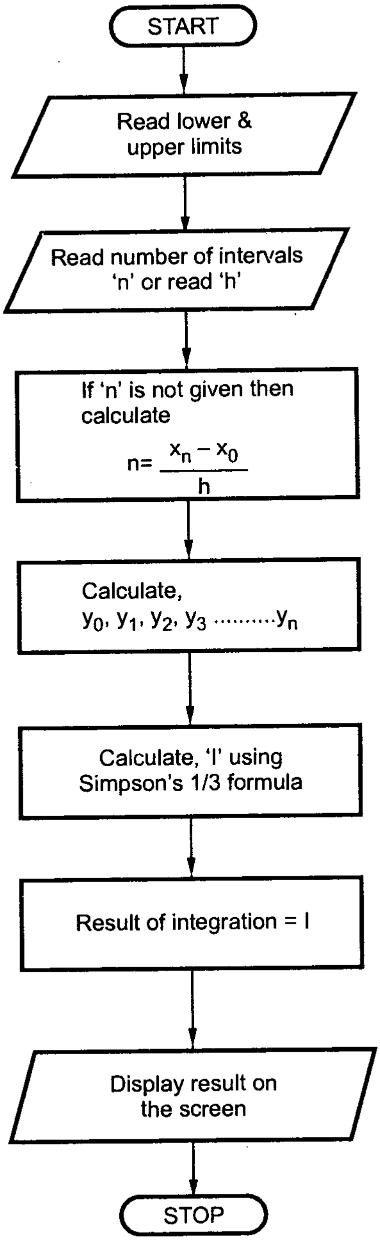


Fig. 7.3.2 Flowchart of simpson's $\frac{1}{3}$ rule

Observe this program carefully. It is almost similar to the program of trapezoidal rule. It asks for upper and lower limits. Then it asks for value of h. After this, the program calculates $y_0, y_1, y_2 \dots$ etc. with the help of contain `fx`.

The last two `for` loops implement the simpson's $\frac{1}{3}$ rule. In that the second last `for` loop in `main` program calculates `4*(odd ordinates)` part of the program and last `for` loop in `main` program calculates `2*(even ordinates)` part of the program.

After the source code of `main` program there is written function `fx`. Here the function being evaluated is

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

The statement in the function fx,

```
f = 1/(1+x);
```

Implements $$f(x) = \frac{1}{1+x}$$

If you want to use this program to integrate some other function, you will have to change the above statement in `fx`.

How to Run this program

Compile and make EXE file of the source code given here. Here we are using

$$\int_0^6 \frac{1}{1+x}\,dx \qquad \text{with} \qquad h = 0.6$$

to illustrate this program. Note that $f(x) = \frac{1}{1+x}$ is defined in the program itself.

Run the program on your computer. It first displays the name of the method.

Then it displays,

`Enter lower limit of integration` = Here enter '0' and press 'enter'

`Enter upper limit of integration` = Here enter 6 and press 'enter'

`Enter the value of h` = Here enter 0.6 and press 'enter'

The program then displays the result of integration.

The combined display of results is given below.

```
---------------------------- Results ----------------------------------

            SIMPSON'S 1/3 RULE OF INTEGRATION

Enter the lower limit of integration = 0


Enter the upper limit of integration = 6


Enter the value of h = 0.6


The result of integration is = 1.948519
------------------------------------------------------------------------
```

University Questions

1. Develop a C procedure to compute the integration on a table of values. [Dec - 96, Dec - 98, Dec - 99]

2. Evaluate the integral

$$I = \int_0^1 \frac{1}{1+x} dx$$

with h = 1/6 by using Simpson's 1/3rd rule and 3/8th rule and compare the results. [May - 97, Dec - 2000]

3. Develop a program in C to evaluate the area under the curve $y = f(x)$ over the interval [a, b] using Simpson's rule. [Dec - 97, May - 99, Dec - 2000]

4. Approximate the integral below using the composite trapezoidal rule and Simpson's rule using 8 equal subdivisions

$$\int_{-1}^{1} \frac{1}{1+x^2} dx$$

[Dec - 97, May - 2000]

5. Use simpson's $\frac{1}{3}^{rd}$ and trapezoidal rules to evaluate $\int_0^{12} \log_e\left(1+x^2\right)dx$ by taking suitable number of intervals. Comment on the result. [May - 2001]

6. The velocity of a car (running on a straight road) at intervals 2 minutes are given below :

Time (min)	0	2	4	6	8	10	12
Velocity (km/hr)	0	22	30	27	18	7	0

Find the distance covered by the car. Justify for the method used. [Dec - 2001]

7. Evaluate $\int_0^1 \frac{\sin x}{x} dx$ using Simpsons $1/3^{rd}$ and trapezoidal rule with h = 1/6. [Dec - 2002]

8. Evaluate $\int_0^{0.8} \left(0.2 + 25x - 200x^2 + 675x^3 - 900x^4\right)$

Using Simpson's $\frac{1}{3}$ rule taking 4 intervals. [May - 2003]

9. The velocity of a car (running on a straight road) at intervals of 2 minutes are given below : [Dec - 2003]

Time (min)	0	2	4	6	8	10	12
Velocity (km/hr)	0	22	30	27	18	07	0

Find the distance covered by the car. (Use Simpsons $1/3^{rd}$ rule)

10. Derive Newton cotes integration formula and use it to get trapezoidal and Simpson's rule. [May - 2004]

7.4 Simpson's $\frac{3}{8}$ Rule

7.4.1 Principle

In simpson's $\frac{1}{3}$ rule we approximate the three points by a parabola. Let's consider that four points are taken and they are interpolated by a polynomial of degree 3. This polynomial is integrated to obtain approximate integration of given function. This method is then called simpson's $\frac{3}{8}$ rule.

7.4.2 Derivation of Simpson's $\frac{3}{8}$ Rule

Let the four points be (x_0, y_0), (x_1, y_1), (x_2, y_2) and (x_3, y_3). The given function can be approximated by lagrange's 3^{rd} degree polynomial. i.e.,

$$f(x) \approx \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} y_0 + \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} y_1$$

$$+ \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} y_2 + \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} y_3$$

The integration of above equation gives required formula i.e.,

$$\int_{x_0}^{x_3} y\, dx = \int_{x_1}^{x_3} \left\{ \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} y_0 + \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} y_1 \right.$$

$$\left. + \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} y_2 + \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} y_3 \right\} dx$$

Integrating and simplifying the above equation we get,

$$\int_{x_0}^{x_3} y\, dx = \frac{x_3 - x_0}{8} [y_0 + 3y_1 + 3y_2 + y_3]$$

Let $\quad h = x_3 - x_2 = x_2 - x_1 = x_1 - x_0$ Hence

$$x_3 - x_0 = 3h$$

$$\therefore \quad \int_{x_0}^{x_3} y\, dx = \frac{3h}{8} [y_0 + 3y_1 + 3y_2 + y_3]$$

This is the required simpson's $\frac{3}{8}$ rule of integration.

7.4.3 Composite Formula

With the single polynomial, the error becomes large. Hence many such polynomials of 3^{rd} degree are taken over $[x_0\ x_n]$. Every polynomial is integrated separately. All the integrations results are then added to give the final result. i.e.,

$$\int_{x_0}^{x_n} y\,dx = \int_{x_0}^{x_3} y\,dx + \int_{x_3}^{x_6} y\,dx + \cdots + \int_{x_{n-3}}^{x_n} y\,dx$$

$$= \frac{3h}{8}[y_0 + 3y_1 + 3y_2 + y_3] + \frac{3h}{8}[y_3 + 3y_4 + 3y_5 + y_6] + \cdots + \frac{3h}{8}[y_{n-3} + 3y_{n-2} + 3y_{n-1} + y_n]$$

$$= \frac{3h}{8}[y_0 + 3(y_1 + y_2 + y_4 + y_5 + \cdots + y_{n-2} + y_{n-1}) + 2(y_3 + y_6 + \cdots + y_{n-3} + y_n]$$

Rearranging the above equation,

$$\boxed{\int_{x_0}^{x_n} y\,dx = \frac{3h}{8}[(y_0 + y_n) + 3(y_1 + y_2 + y_4 + y_5 + y_7 + \ldots) + 2(y_3 + y_6 + y_9 \ldots)]} \quad \ldots (7.4.1)$$

This formula can also be written as,

$$\int_{x_0}^{x_n} f(x)\,dx = \frac{3h}{8}\begin{bmatrix} (\textit{Sum of first and last ordinate}) \\ + 3\,(\textit{Sum of ordinates which are not multiple of } 3) \\ + 2\,(\textit{Sum of ordinates which are multiple of } 3) \end{bmatrix}$$

This formula is used when there are even number of points and odd number of segments.

Error in Simpson's $\frac{3}{8}$ rule is given as,

$$\boxed{E_n = -\frac{h^4}{80}(x_n - x_0)\, f^{(4)}(z)} \quad \ldots (7.4.2)$$

This can also be written as,

$$E_n = -\frac{(x_n - x_0)^5}{80\, n^4} f^{(4)}(z) \qquad \text{Since } h = \frac{x_n - x_0}{n}$$

and $\qquad x_0 \le z \le x_n$

7.4.4 Solved Examples

Ex. 7.4.1 *Evaluate the integral* $\int_0^{\pi} (4 + 2 \sin x)\,dx$ *using simpsons* $\frac{3}{8}$ *rule where* $n = 5$.

Compute percent relative error.

Sol. : Here $\qquad n = 5$

$$\therefore \quad h = \frac{x_n - x_0}{n} = \frac{\pi - 0}{5}$$

$$\therefore \quad h = \frac{\pi}{5} \quad \text{or} \quad h = 36°$$

Let's calculate the following table -

x	$x_0 = 0$	$x_1 = \frac{\pi}{5}$	$x_2 = \frac{2\pi}{5}$	$x_3 = \frac{3\pi}{5}$	$x_4 = \frac{4\pi}{5}$	$x_5 = \pi$
y	$y_0 = 4$	$y_1 = 5.176$	$y_2 = 5.902$	$y_3 = 5.902$	$y_4 = 5.176$	$y_5 = 4$

For $n = 5$, we can write simpson's $\frac{3}{8}$ rule as,

$$\int_{x_0}^{x_5} y\, dx = \frac{3h}{8}\left[(y_0 + y_5) + 3(y_1 + y_2 + y_4) + 2(y_3)\right]$$

Putting the values,

$$\int_0^{\pi} (4 + 2\sin x)\, dx = \frac{3 \times \frac{\pi}{5}}{8}\left[(4+4) + 3(5.176 + 5.902 + 5.176) + 2 \times 5.902\right]$$

$$= 16.155483$$

Actual value will be,

$$I = \int_0^{\pi} (4 + 2\sin x)\, dx$$

$$= \left[4x + 2(-\cos x)\right]_0^{\pi} = 16.566371$$

$\therefore$ % Relative error $= \frac{|16.566371 - 16.155483|}{16.566371} \times 100 = 2.48025\%$

Ex. 7.4.2 *Compute the value of a definite integral*

$$\int_{0.2}^{1.4} \left(\sin x - \log_e x + e^x\right) dx \text{ by using simpson's } \frac{3}{8} \text{ rule. Take } h = 0.1.$$

Sol. : We have to evaluate

$$\int_{0.2}^{1.4} \left(\sin x - \ln x + e^x\right) dx$$

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

$$= \frac{1.4 - 0.2}{0.1}$$

$\therefore$ $n = 12$

Let's calculate the following table -

x	0.2	0.3	0.4	0.5	0.6	0.7
y	3.02951	2.8493518	2.7975338	2.821294	2.8975869	3.0146653

0.8	0.9	1.0	1.1	1.2	1.3	1.4
3.1660406	3.3482905	3.5597528	3.8	4.06983	4.37049	4.7041775

From equation 7.4.1 simpson's $\frac{3}{8}$ rule is given as,

$$\int_{0.2}^{1.4} f(x)\,dx = \frac{3\times 0.1}{8}\left[\begin{array}{l}(y_0+y_{12})+3(y_1+y_2+y_4+y_5+y_7+y_8+y_{10}+y_{11})\\ +2(y_3+y_6+y_9)\end{array}\right]$$

Putting values of $y_0, y_1 \ldots y_{12}$ in above equation we obtain,

$$\int_{0.2}^{1.4} f(x)\,dx = 4.0511582$$

Ex. 7.4.3 *Compute* $\int_0^3 \frac{dx}{1+x}$ *by Simpson's* $\frac{3}{8}$ *rule dividing interval (0, 3) into 6 equal parts.*

Also obtain the error bound.

Sol. : Here $n = 6$

$$\therefore \quad h = \frac{x_n - x_0}{n} = \frac{3-0}{6}$$

$$\therefore \quad h = 0.5$$

Let's prepare the following table -

x	$x_0 = 0$	$x_1 = 0.5$	$x_2 = 1$	$x_3 = 1.5$	$x_4 = 2$	$x_5 = 2.5$	$x_6 = 3$
y	$y_0 = 1$	$y_1 = 0.66667$	$y_2 = 0.5$	$y_3 = 0.4$	$y_4 = 0.33333$	$y_5 = 0.2857$	$y_6 = 0.25$

From simpson's $\frac{3}{8}$ rule of equation 7.4.1 we have,

$$\int_0^3 \frac{dx}{1+x} = \frac{3h}{8}[(y_0+y_6)+3(y_1+y_2+y_4+y_5)+2(y_3)]$$

$$= \frac{3\times 0.5}{8}[(1+0.25)+3(0.66667+0.5+0.3333+0.2857)+2\times 0.4]$$

$$= 1.3888313$$

Here, $f(x) = \frac{1}{1+x}$

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

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

$$f'''(x) = \frac{-6}{(1+x)^4}$$

and
$$f^{(4)}(x) = \frac{24}{(1+x)^5}$$

In the interval $[0,3]\, f^{(4)}(x)$ will be maximum when $x = 0$ i.e.

$$Max\,|f^{(4)}(x)| = \frac{24}{1} = 24$$

From equation 7.4.2 we know that error in simpson's $\frac{3}{8}$ rule is given as,

$$E_n = -\frac{h^4}{80}(x_n - x_0)\, f^{(4)}(z)$$

Maximum error will be,

$$E_n \le \frac{(0.5)^4}{80}(3-0)\max|f^{(4)}(z)|$$

$$\le \frac{(0.5)^4}{80} \times 3 \times 24 \quad \text{when} \quad z = 0$$

$$\le 0.05625$$

Ex. 7.4.4 *Use simpson's $\frac{3}{8}^{th}$ rule to evaluate*

$$\int_0^{\pi/2} \sqrt{\sin x + \cos x}\; dx$$

taking $h = \frac{\pi}{6}$

Sol. : For $h = \frac{\pi}{6}$, the intervals in the range $\left[0, \frac{\pi}{2}\right]$ will be,

$x\ :\ 0, \frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{2}$ (Three intervals)

Here
$$y = \sqrt{\sin x + \cos x}$$

The values of x and y are tabulated below :

x	$x_0 = 0$	$x_1 = \frac{\pi}{6}$	$x_2 = \frac{\pi}{3}$	$x_3 = \frac{\pi}{2}$
$y = \sqrt{\sin x + \cos x}$	$y_0 = 1$	$y_1 = 1.1687709$	$y_2 = 1.1687709$	$y_3 = 1$

For $n = 3$, simpson's $\frac{3}{8}$ rule is given as,

$$\int_{x_0}^{x_3} y\, dx = \frac{3h}{8}\left[y_0 + y_3 + 3(y_1 + y_2)\right]$$

Putting the values in above equation,

$$\int_0^{\frac{\pi}{2}} \sqrt{\sin x + \cos x}\, dx = \frac{3 \cdot \frac{\pi}{6}}{8}\left[1 + 1 + 3\,(1.1687709 + 1.1687709)\right]$$

$$= 1.7696249$$

Ex. 7.4.5 *Evaluate $\log_e 7$ (logarithm of 7 to base e) by simpson's :*

i) $\frac{1}{3}^{rd}$ *rule and*

ii) $\frac{3}{8}^{th}$ *rule*

Comment on the accuracy of the results of above two methods.

Sol. : Here let us write $\log_e 7$ as $\ln 7$. Here we have to evaluate $\ln 7$ by simpson's rules. This means we should evaluate the function whose integration is $\ln 7$. We have

$$\int_0^6 \frac{1}{1+x}\, dx = \left[\ln(1+x)\right]_0^6$$

$$= \ln(1+6) - \ln(1+0)$$

$$= \ln 7$$

Thus we have to evaluate the integration of $\int_0^6 \frac{1}{1+x}\, dx$ using simpson's rules.

Let us take $h = 1$. Hence,

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

or

$$n = \frac{x_n - x_0}{h} = \frac{6-0}{1} = 6$$

Thus there will be 6 intervals

Let $y = \frac{1}{1+x}$. The following table shows values of x and y.

x	$x_0 = 0$	$x_1 = 1$	$x_2 = 2$	$x_3 = 3$	$x_4 = 4$	$x_5 = 5$	$x_6 = 6$
$y = \frac{1}{1+x}$	$y_0 = 1$	$y_1 = 0.5$	$y_2 = 0.333333$	$y_3 = 0.25$	$y_4 = 0.2$	$y_5 = 0.166667$	$y_6 = 0.1428571$

i) Using Simpson's $\frac{1}{3}^{rd}$ rule

For $n = 6$, simpson's $\frac{1}{3}^{rd}$ rule can be written as,

$$\int_{x_0}^{x_6} y\,dx = \frac{h}{3}[y_0 + y_6 + 4(y_1 + y_3 + y_5) + 2(y_2 + y_4)]$$

Putting the values,

$$\int_0^6 \frac{1}{1+x}\,dx = \frac{1}{3}[1 + 0.1428571 + 4(0.5 + 0.25 + 0.166667)$$
$$+ 2(0.333333 + 0.2)]$$
$$= 1.95873$$

ii) Using Simpson's $\frac{3}{8}^{th}$ rule

For $n = 6$, simpson's $\frac{3}{8}^{th}$ rule can be written as,

$$\int_{x_0}^{x_6} y\,dx = \frac{3h}{8}[y_0 + y_6 + 3(y_1 + y_2 + y_4 + y_5) + 2y_3]$$

Putting the values,

$$\int_0^6 \frac{1}{1+x}\,dx = \frac{3 \times 1}{8}[1 + 0.1428571 + 3(0.5 + 0.33333$$
$$+ 0.2 + 0.166667) + 2(0.25)]$$
$$= 1.9660713$$

Actual value of $ln\,7$ is 1.9459101.

$\therefore$ Absolute error in Simpson's $\frac{1}{3}^{rd}$ rule $= |1.9459101 - 1.95873| = 0.0128198$

and absolute error in Simpson's $\frac{3}{8}^{th}$ rule $= |1.9459101 - 1.9660713|$

$$= 0.0201611$$

Thus the result obtained by Simpson's $\frac{1}{3}^{rd}$ rule is better.

Ex. 7.4.6 *Evaluate*

$$\int_0^3 \frac{dx}{1+x}$$

with 7 (seven) ordinates by using simpson's $\frac{3}{8}^{th}$ *rule and hence calculate* $\log_e 2$.

Sol. : Here we have seven ordinates means '6' intervals.

Hence $n = 6$. Hence 'h' will be,

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

$$= \frac{3-0}{6} \qquad \text{since} \qquad x_0 = 0 \text{ and } x_n = 3$$

$$= 0.5$$

The function to be integrated is $y = \frac{1}{1+x}$. The table of values of x and corresponding y is given below :

x	$x_0 = 0$	$x_1 = 0.5$	$x_2 = 1$	$x_3 = 1.5$	$x_4 = 2$	$x_5 = 2.5$	$x_6 = 3$
y	$y_0 = 1$	$y_1 = 0.6666$	$y_2 = 0.5$	$y_3 = 0.4$	$y_4 = 0.3333$	$y_5 = 0.2857142$	$y_6 = 0.25$

Now Simpson's $\frac{3}{8}^{th}$ rule for $n = 6$ is given as,

$$\int_{x_0}^{x_6} y\,dx = \frac{3h}{8}[y_0 + y_6 + 3(y_1 + y_2 + y_4 + y_5) + 2(y_3)]$$

Putting the values in above equation,

$$\int_0^3 \frac{1}{1+x}dx = \frac{3 \times 0.5}{8}[1 + 0.25 + 3(0.6666 + 0.5 + 0.3333 + 0.28571) + 2(0.4)]$$

$$= 1.3888392$$

To obtain $\log_e 2$ or $\ln 2$: Here we have to obtain $\ln 2$ by integration. This means we have to integrate the function whose integration is $\ln 2$. Let us consider the integration given in this example. i.e.,

$$\int_0^3 \frac{1}{1+x}dx = [\ln(1+x)]_0^3 = \ln(1+3) - \ln(1+0)$$

$$= \ln 4$$

We have $4 = 2^2$. Hence $\ln 4 = \ln 2^2 = 2 \ln 2$

Hence above equation will be

$$2\ln 2 = \int_0^3 \frac{1}{1+x}\,dx$$

Putting the value of $\int_0^3 \frac{1}{1+x}\,dx = 1.3888392$ in above equation,

$$2\ln 2 = 1.3888392$$

$\therefore$

$$\ln 2 = 0.6944196$$

Thus

$$\log_e 2 = \ln 2 = 0.6944196$$

Ex. 7.4.7 *Calculate :*

$\int_0^{\frac{\pi}{2}} e^{\sin\theta}\,d\theta$ *by Simpson's* $\frac{3}{8}^{th}$ *rule for interval* $\left[0, \frac{\pi}{2}\right]$ *in 4 equal parts.*

Sol. : Here we have $x_0 = \theta_0 = 0$ and $x_n = \theta_n = \frac{\pi}{2}$. Number of intervals $n = 4$ (Four equal parts given).

$$y = e^{\sin\theta}$$

Hence value of h will be,

$$h = \frac{x_n - x_0}{n} = \frac{\theta_n - \theta_0}{\nu}$$

$$= \frac{\frac{\pi}{2} - 0}{4}$$

$$= \frac{\pi}{8}$$

Let us form the table of θ and y as follows :

θ	$\theta_0 = 0$	$\theta_1 = \frac{\pi}{8}$	$\theta_2 = \frac{\pi}{4}$	$\theta_3 = \frac{3\pi}{8}$	$\theta_4 = \frac{\pi}{2}$
y	$y_0 = 1$	$y_1 = 1.4662138$	$y_2 = 2.028115$	$y_3 = 2.5190442$	$y_4 = 2.7182818$

Simpson's $\frac{3}{8}^{th}$ rule for $n = 4$ is written as,

$$\int_{\theta_0}^{\theta_n} y\,d\theta = \frac{3h}{8}\left[y_0 + y_4 + 3(y_1 + y_2) + 2(y_3)\right]$$

Putting the values in above equation,

$$\int_0^{\frac{\pi}{2}} e^{\sin\theta}\,d\theta = \frac{3\times\frac{\pi}{8}}{8}\left[1 + 2.7182818 + 3(1.4662138 + 2.028115) + 2(2.5190442)\right]$$

$$= 2.8332291$$

Ex. 7.4.8 *Evaluate*

$$\int_0^{\pi} \frac{\sin^2\theta}{5+4\cos\theta}\, d\theta \text{ by Simpson's } \frac{3}{8}^{th} \text{ rule taking } h=\frac{\pi}{6}.$$

Sol. : Here $$y = \frac{\sin^2\theta}{5+4\cos\theta}$$

The values of θ and y are given in the following table :

θ	$\theta_0 = 0$	$\theta_1 = \frac{\pi}{6}$	$\theta_2 = \frac{\pi}{3}$	$\theta_3 = \frac{\pi}{2}$	$\theta_4 = \frac{2\pi}{3}$	$\theta_5 = \frac{5\pi}{6}$	$\theta_6 = \pi$
y	$y_1 = 0$	$y_1 = 0.0295365$	$y_2 = 0.1071428$	$y_3 = 0.2$	$y_4 = 0.25$	$y_5 = 0.162771$	$y_6 = 0$

Here observe that there are $n=6$ intervals. Simpson's $\frac{3}{8}^{th}$ rule for $n=6$ is given as,

$$\int_{\theta_0}^{\theta_6} y\, d\theta = \frac{3h}{8}\left[y_0+y_6+3(y_1+y_2+y_4+y_5)+2(y_3)\right]$$

Putting the values in above equation,

$$\int_0^{\pi} \frac{\sin^2\theta}{5+4\cos\theta}\, d\theta = \frac{3\,\frac{\pi}{6}}{8}\left[0+0+3\,(0.0295365+0.1071428+0.25+0.1627711)+2\,(0.2)\right]$$

$$= 0.4021928$$

Ex. 7.4.9 *Use simpson's $\frac{3}{8}$ th rule to integrate the function $f(x)=0.2+20x+25x^2+60x^3$ over the limit a = 0.0 to b = 1.0*

[Dec - 99, 8 marks, May - 98, 8 marks, Dec - 96, 6 marks]

Sol. : Here we have to use simpson's $\frac{3}{8}$ th rule. We know that this rule requires odd number of segments. Hence let us take $n=7$ segments. The limits of integration are $x_0=0$ and $x_n=1$.

Hence
$$h = \frac{x_n - x_0}{n} = \frac{1-0}{7} = \frac{1}{7}$$

Let us prepare the table of x and $y=f(x)$ as follows :

x	$x_0 = 0$	$x_1 = \frac{1}{7}$	$x_2 = \frac{2}{7}$	$x_3 = \frac{3}{7}$	$x_4 = \frac{4}{7}$	$x_5 = \frac{5}{7}$	$x_6 = \frac{6}{7}$	$x_7 = 1$
$y=f(x)$	$y_0 = 0.2$	$y_1 = 3.7422$	$y_2 = 9.3545$	$y_3 = 18.0862$	$y_4 = 30.9871$	$y_5 = 49.106$	$y_6 = 73.49$	$y_7 = 105.2$

Simpson's $\frac{3}{8}$ th rule is given by equation 7.4.1 as,

$$\int_{x_0}^{x_n} y\,dx = \frac{3h}{8}\left[(y_0+y_n)+3(y_1+y_2+y_4+y_5+y_7+\ldots)+2(y_3+y_6+y_9+\ldots)\right]$$

Putting values in above equation,

$$\int_0^1 f(x)\,dx = \frac{3h}{8}\left[(y_0+y_n)+3(y_1+y_2+y_4+y_7)+2(y_3+y_6)\right]$$

Putting values from the table,

$$\int_0^1 f(x)\,dx = \frac{3}{8}\cdot\frac{1}{7}\,\{(0.2+105.2)+3(3.7422+9.3545+30.9871+49.106)+2(18.0862+73.49)\}$$

$$= 30.435696$$

Ex. 7.4.10 *Evaluate the integral* $I = \int_0^1 \frac{1}{1+x}\,dx$ *with* $h = \frac{1}{6}$ *by using simpson's* $\frac{1}{3}$ *rd and* $\frac{3}{8}$ *th rule and compare the results. Also comment on the results.*

[Dec - 2000, 8 marks, May - 97, 10 marks]

Sol. : We know that, $n = \frac{x_n - x_0}{h}$

$$= \frac{1-0}{1/6} = 6$$

Thus there are 6 subdivisions. Let us form the table of x and $y = f(x)$ as follows.

x	$x_0 = 0$	$x_1 = \frac{1}{6}$	$x_2 = \frac{2}{6}$	$x_3 = \frac{3}{6}$	$x_4 = \frac{4}{6}$	$x_5 = \frac{5}{6}$	$x_6 = \frac{6}{6}$
$y = \frac{1}{1+x}$	$y_0 = 1$	$y_1 = 0.8571$	$y_2 = 0.75$	$y_3 = 0.6667$	$y_4 = 0.6$	$y_5 = 0.545$	$y_6 = 0.5$

i) Using simpson's $\frac{1}{3}$ rd rule :

Equation 7.3.2 gives simpson's $\frac{1}{3}$ rd rule as,

$$\int_{x_0}^{x_n} y\,dx = \frac{h}{3}\left[(y_0+y_n)+4(y_1+y_3+y_5+\ldots)+2(y_2+y_4+y_6+\ldots)\right]$$

For $n = 6$, above equation becomes,

$$\int_{x_0}^{x_6} \frac{1}{1+x}\,dx = \frac{h}{3}\left[(y_0+y_6)+4(y_1+y_3+y_5)+2(y_2+y_4)\right]$$

Putting values in above equation

$$\int_0^1 \frac{1}{1+x}\,dx = \frac{1/6}{3}\left[(1+0.5)+4(0.8571+0.6667+0.5454)+2(0.75+0.6)\right]$$

$$= 0.6931555$$

ii) Using simpson's $\frac{3}{8}$ th rule :

Equation 7.4.1 gives simpson's $\frac{3}{8}$ th rule as,

$$\int_{x_0}^{x_n} y\,dx = \frac{3h}{8}\{(y_0+y_n)+3\,(y_1+y_2+y_4+y_5+y_7+\ldots)+2(y_3+y_6+y_9+\ldots)\}$$

For $n=6$, above equation becomes,

$$\int_{x_0}^{x_6} \frac{1}{1+x}\,dx = \frac{3h}{8}\{(y_0+y_6)+3\,(y_1+y_2+y_4+y_5)+2y_3\}$$

Putting values in above equation,

$$\int_0^1 \frac{1}{1+x}\,dx = \frac{3(1/6)}{8}\{(1+0.5)+3(0.8571+0.75+0.6+0.545)+2(0.6667)\}$$

$$= 0.6931775$$

iii) Comments :

Actual value of the integral is,

$$\int_0^1 \frac{1}{1+x}\,dx = \left[ln\,(1+x)\right]_0^1$$

$$= ln\ 2 = 0.6931471$$

Method	Answer
Simpson's $\frac{1}{3}$ rule	0.6931555
Simpson's $\frac{3}{8}$ rule	0.6931775
Actual value	0.6931471

From above table, it is clear that simpson's $\frac{1}{3}$ rd rule provides more correct answer. This is because there are 6 subdivisions (segments). Simpson's $\frac{1}{3}$ rd rule provides more correct answers for even number of segments. Whereas simpson's $\frac{3}{8}th$ rule provider more correct answer with odd number of segments.

Exercise

1. *Evaluate* $\int_4^8 \frac{dx}{\sqrt{16x - x^2}}$ *by using simpson's* $\frac{3}{8}$ *rule dividing the interval into 6 equal parts.*

[Ans. : 0.5239]

2. *Evaluate* $\int_0^{\pi/2} e^{\sin\theta}\, d\theta$ *by simpson's* $\frac{3}{8}$ *rule by dividing the interval* $\left[0, \frac{\pi}{2}\right]$ *into 6 equal strips.*

[Ans. : 3.1012]

3. *Evaluate* $\int_0^{\pi/4} \cos x\, dx$ *by dividing the interval into three strips using simpson's* $\frac{3}{8}$ *rule.*

[Ans. : 0.7071332]

4. *Use simpson's* $\frac{3}{8}$ *rule to integrate*

$f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$ from a = 0 to b = 0.8.

[May - 99, 8 marks, May - 96, 10 marks]

[Ans. : 1.584212 with 7 equal segments.]

7.4.5 Algorithm and C Program*

Based on the discussion of the method and illustrative problems we will now prepare an algorithm for simpson's $\frac{3}{8}$ rule of integration.

Algorithm :

Assumption : The function which is to be integrated is predefined in the program.

Step 1 : Read the lower and upper limits of integration.

Step 2 : Read the spacing 'h' or read number of intervals to be taken i.e. 'n'.

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

Step 3 : Calculate,

$$y_0 = f(x)\big|_{x=x_0}, \quad y_1 = f(x)\big|_{x=x_1}$$

$$y_2 = f(x)\big|_{x=x_2}$$

and so on.

Step 4 : Calculate,

$$I = \frac{3h}{8}\left[(y_0 + y_n) + 3(y_1 + y_2 + y_4 + y_5 + y_7 + \ldots) + 2(y_3 + y_6 + y_9 + \ldots)\right]$$

Step 5 : Display the value of result of integration I on the screen.

Step 6 : Stop.

Flowchart :

Fig. 7.4.1 shows the simplified flowchart of simpson's $\frac{3}{8}$ rule of integration. It is assumed that the function to be integrated is predefined.

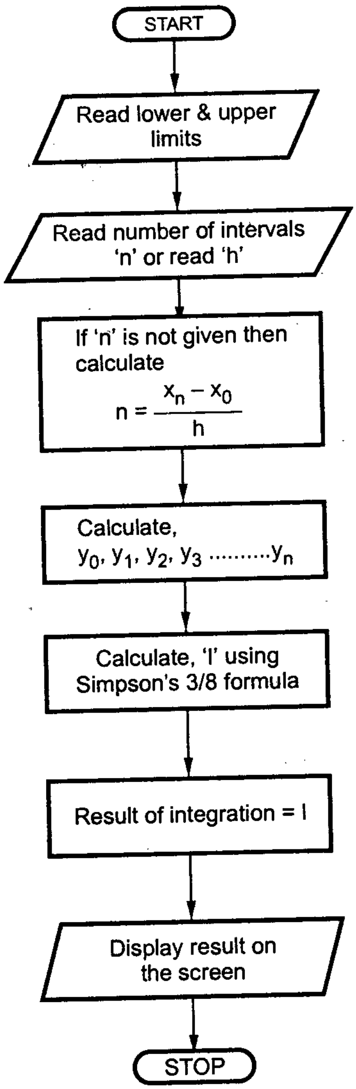


Fig. 7.4.1 Flowchart of simpson's $\frac{3}{8}$ rule

Computer program :

The source code of C program for simpson's $\frac{3}{8}$ rule is given below.

```
/* Download this program from www.vtubooks.com                        */
/* File name : simp3_8.cpp                                             */

/*----------------- SIMPSON'S 3/8 RULE OF INTEGRATION ----------------*/

/*      THIS PROGRAM CALCULATES THE VALUE OF INTEGRATION USING
        SIMPSON'S 3/8 RULE. THE FUNCTION TO BE INTEGRATED IS,

                    f(x) = 4 + 2 sin x

        INPUTS :  1) Lower and upper limits of integration.

                  2) Number of intervals.

        OUTPUTS : Result of integration.                               */

/*------------------------------   PROGRAM  ------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>

void main()
{
      double fx (double x0);       /*   DECLARATION OF A FUNCTION fx   */

      double lo,up,f[20],h,x0,sum,result;
      int i,n;

      clrscr();
      printf("\n\t   SIMPSON'S 3/8 RULE OF INTEGRATION");

      printf("\n\nEnter the lower limit of integration = ");
      scanf("%lf",&lo);         /* ENTER LOWER LIMIT OF INTEGRATION    */
      printf("\n\nEnter the upper limit of integration = ");
      scanf("%lf",&up);         /* ENTER UPPER LIMIT OF INTEGRATION    */

      printf("\n\nEnter the value of h = ");
      scanf("%lf",&h);            /*    ENTER THE VALUE OF h           */

      n = (up - lo)/h;           /* CALCULATION VALUE OF n i.e.STRIPS  */
      x0 = lo;

      for(i = 0; i <= n; i++)     /* LOOP TO CALCULATE VALUE OF f(x)   */
      {
          f[i] = fx(x0);          /*   FUNCTION fx IS CALLED HERE      */
          x0 = x0 + h;      /* NEXT VALUE OF x IS CALCULATED HERE      */
      }
      sum = 0;
      for(i = 1; i <= n-1; i++)
      {
        if(i == 3*(i/3) ) continue;
        sum = sum + 3*f[i];  /*  3 * SUM OF ORDINATES NOT MULTIPLE OF 3 */
      }
      for(i = 3; i <= n-1; i = i + 3)
      {
        sum = sum + 2*f[i]; /*2 * SUM OF ORDINATES WHICH ARE MULTIPLE OF 3*/
      }
      result = (3*h/8) * ( f[0] + f[n] + sum );

          /*Result = (3h/8) * (3 * sum of ordinates not multiple of 3
                       + 2 * sum of ordinates which are multiple of 3 )   */
```

```
    printf("\n\nThe result of integration is = %lf",result);
}

double fx ( double x)       /* FUNCTION TO CALCULATE VALUE OF f(x)  */
{
    double f;
    f = 4 + 2 * sin(x);          /*    function f(x) = 4 + 2 sin x     */
    return(f);
}
/*------------------------- END OF PROGRAM ------------------------------*/
```

This program is also similar to the previous two programs of trapezoidal and Simpson's $\frac{1}{3}$ methods. The only difference is the implementation of Simpsons $\frac{3}{8}$ formula.

You will find the program is exactly same as previous programs till the second last `for` loop. The second last for loop is given below -

```
for  (i=1;  i<=n-1;  i++)              ← Count ordinates from 1 to (n – 1)
{
    if  (i==3*(i/3))  continue  ;      ← if ordinate is multiple of three bypass next
                                          statement
    sum=sum+3*f[i];                    ← calculate 3 × (sum of ordinate not multiple of 3)
}
```

Here observe that the statement,

`if  (i==3*(i/3))  continue;`

Here if `i` is exact multiple of '3' then above condition is satisfied and continue statement bypasses the next statement. The next statement calculates,

$3 \times$ (sum of ordinates which are not multiple of 3).

The next `for` loop calculates sum of remaining ordinates.

In this program, the function being integrated is defined in subroutine `fx`. This subroutine is listed after main program. Observe the statement in `fx`,

`f = 4+2*sin(x);`

This is implementation of

$$fx = 4 + 2 \sin x$$

The limits of integration and value of 'h' should be given externally. If you want to use this program to integrate some other function, then you will have to change the above statement in subroutine fx.

How to Run this program

Compile and make EXE file of this source code. To illustrate this program we will integrate

$$\int_0^{\pi} (4 + 2\, sin\, x)\, dx \text{ with } h = \frac{\pi}{5}$$

Here $\pi = 3.1415927$ is to be entered.

and $h = \frac{\pi}{5} = 0.6283185$ is to be entered.

Run the program on your computer. It will display

`Enter the lower limit of integration` = Here enter 0 and 'enter'

`Enter the upper limit of integration` = Here enter 3.1415927 and 'enter'

`Enter the value of h` = Here enter 0.6283185 and 'enter'

The program then displays the result of integration.

Combined display of above results is shown below -

```
--------------------------- Results -----------------------------------

             SIMPSON'S 3/8 RULE OF INTEGRATION

Enter the lower limit of integration = 0

Enter the upper limit of integration = 3.1415927

Enter the value of h = 0.6283185

The result of integration is = 16.155009
-----------------------------------------------------------------------
```

This function we have integrated in example 7.4.1.

Compare the result of example and program.

7.4.6 Comparison between Trapezoidal and Simpson's Rules

Following table shows the comparison between trapezoidal, simpson's $\frac{1}{3}$ and simpson's $\frac{3}{8}$ rules.

Table 7.4.1 Comparison of integration methods

Sr. no	Trapezoidal rule	Simpson's $\frac{1}{3}$ rule	Simpson's $\frac{3}{8}$ rule
1.	Approximates f(x) by straight line.	Approximates f(x) by parabola.	Approximates f(x) by a 3rd order polynomial.
2.	$\int_{x_0}^{x_n} y\,dx = (x_n - x_0)\frac{y_0 + y_n}{2}$	$\int_{x_0}^{x_n} y\,dx = \frac{h}{3}[y_0 + 4y_k + y_n]$	$\int_{x_0}^{x_3} y\,dx = \frac{3h}{8}[y_0 + 3y_1 + 3y_2 + y_3]$
3.	$E_n = -\frac{h^2}{12}(x_n - x_0)f''(z)$ $x_0 \le z \le x_n$	$E_n = \frac{(x_n - x_0)^5}{180n^4} f^{(4)}(z)$ $x_0 \le z \le x_n$	$E_n = -\frac{(x_n - x_0)^5}{80n^4} f^{(4)}(z)$ $x_0 \le z \le x_n$
4.	Error is more	Error is less than trapezoidal but more than simpsons $\frac{3}{8}$	Error is less

University Questions

1. Discuss the relative advantages and disadvantages of the Simpson's rule over the trapezoidal rule. **[Dec - 95]**
2. Using expressions for errors compare the Trapezoidal, Simpson's 1/3rd and Simpson's 3/8th methods of integration. **[May - 96, Dec - 96, May - 98, Dec - 98]**
3. Use Simpson's 3/8 rule to integrate

   $f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$ from a = 0 to b = 0.8 where b – a is the width of integration. **[May - 96, May - 99]**
4. Use Simpson's 3/8 rule to integrate the function $f(x) = 0.2 + 20x + 25x^2 + 60x^3$ over the limits a = 0.0 to b = 1.0. **[Dec - 96, May - 98, Dec - 99]**
5. Derive Simpson's 3/8th rule

   $$\int_{x_0}^{x_3} ydx = 3/8\ h(y^0 + 3y^1 + 3y^2 + y^3)$$

   [May - 97, May - 2000]
7. Draw a flow-chart and explain the method to find the area under a curve using Simpson's 3/8th rule. **[Dec - 2001]**
8. Derive Newton Cote's integration formula and hence find Simpsons $1/3^{rd}$ and $3/8^{th}$ rule. **[Dec - 2002, Dec - 2004]**
9. Write a menu driven program in C / C++ for integration using Simpson's $1/3^{rd}$ and $3/8^{th}$ rule. **[Dec - 2002, Dec - 2004]**
10. Write a program in C/C++ for Intergration using Simpson's $\frac{3}{8}$ rule. **[May - 2003]**
11. Derive $\int_{x_0}^{x_3} y\,dx = \frac{3h}{8}(y_0 + 3y_1 + 3y_2 + y_3)$ **[May - 2003]**
12. Write the algorithm for integration using simpson's $3/8^{th}$ rule. **[Dec - 2003]**

7.5 Engineering Applications

Here we will use the integration methods to determine the RMS value of a current. We know that a sinusoidal current is given as,

$$i(t) = A\,sin\left(\frac{2\pi t}{T}\right) \quad \text{... (7.5.1)}$$

Here A is peak amplitude of current.

& T is the period of the current.

For our study let $T = 2\pi$. This waveform is shown in Fig. 7.5.1.

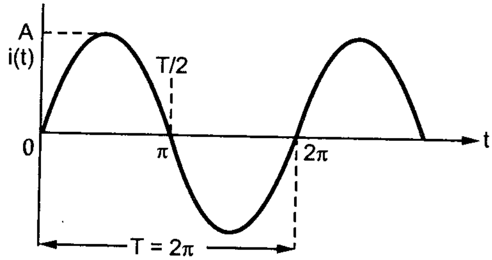


Fig. 7.5.1 Sinusoidal current of unit amplitude and period

∴ We can write equation 7.5.1 as,

$$i(t) = A \sin t \qquad \text{(with } T = 2\pi) \qquad \dots (7.5.2)$$

RMS value of the sinusoidal current is given by standard formula as,

$$I_{rms} = \left[\frac{1}{T}\int_0^T i^2(t)\, dt\right]^{\frac{1}{2}} \qquad \dots (7.5.3)$$

Putting the value of $i(t)$ and T in above formula,

$$I_{rms} = \left[\frac{1}{2\pi}\int_0^{2\pi} A^2 \sin^2 t\, dt\right]^{\frac{1}{2}} \qquad \dots (7.5.4)$$

$$= \left(\frac{A^2}{2\pi} \times I\right)^{\frac{1}{2}} \qquad \dots (7.5.5)$$

Here I is given as,

$$I = \int_0^{2\pi} \sin^2 t\, dt \qquad \dots (7.5.6)$$

Let's solve this integration using simpson's $\frac{1}{3}$ rule,

Let's divide the complete integrating interval $[0, 2\pi]$ into 12 divisions.

i.e. $\qquad n = 12$

$$\therefore \qquad h = \frac{x_n - x_0}{n} = \frac{2\pi - 0}{12} = \frac{\pi}{6}$$

Let's prepare the table of values for t and $y = i^2(t) = \sin^2 t$ as follows :

t	$y = \sin^2 t$	t	$y = \sin^2 t$
0	$y_0 = 0$	$\frac{7\pi}{6}$	$y_7 = 0.25$
$\frac{\pi}{6}$	$y_1 = 0.25$	$\frac{4\pi}{3}$	$y_8 = 0.75$
$\frac{\pi}{3}$	$y_2 = 0.75$	$\frac{3\pi}{2}$	$y_9 = 1$

$\frac{\pi}{2}$	$y_3 = 1$	$\frac{5\pi}{3}$	$y_{10} = 0.75$
$\frac{2\pi}{3}$	$y_4 = 0.75$	$\frac{11\pi}{6}$	$y_{11} = 0.25$
$\frac{5\pi}{6}$	$y_5 = 0.25$	2π	$y_{12} = 0$
π	$y_6 = 0$		

From equation 7.3.2 simpson's $\frac{1}{3}$ rule is given as,

$$\int_{x_0}^{x_n} y\,dx = \frac{h}{3}\left[\begin{array}{l}(y_0 + y_n) + 4(y_1 + y_3 + y_5 + y_7 + \ldots) \\ \quad + 2(y_2 + y_4 + y_6 + \ldots)\end{array}\right]$$

Here $h = \frac{\pi}{6}$ and putting values of y from above table,

$$\int_0^{2\pi} \sin^2 t\,dt = \frac{\pi/6}{3}\left[\begin{array}{l}(0+0) + 4(0.75 + 0.75 + 0 + 0.75 + 0.75) \\ \quad + 2(0.25 + 1 + 0.25 + 0.25 + 1 + 0.25)\end{array}\right]$$

$$= \pi \text{ or } 3.1415927$$

i.e. $I = \pi$ or 3.1415927 from equation 7.5.6 ... (7.5.7)

Putting this value of integration 'I' in equation 7.5.5,

$$I_{rms} = \left(\frac{A^2}{2\pi} \times I\right)^{1/2}$$

$$= \left(\frac{A^2}{2\pi} \times \pi\right)^{1/2} \quad \because I = \pi \text{ from equation 7.5.7}$$

$$= \frac{A}{\sqrt{2}}$$

Thus for sinusoidal current,

$$I_{rms} = \frac{A}{\sqrt{2}}$$

$$= \frac{\text{Peak amplitude}}{\sqrt{2}}$$

We know that this is the standard formula for RMS value of sinusoidal quantity. Thus simpson's $\frac{1}{3}$ rule can be used numerically to calculate RMS value. Here A can be peak amplitude of voltage or current.

7.6 MATLAB for Integration

MATLAB can also be used for numerical integration. A program is given below:

```
%   Download this program from www.vtubooks.com
%   file name : integration.m

%   This program calculates the integration from the given tabulated
%   data using trapezoidal rule.                    Matlab Version 6

%   Input : Values of x and y = f(x) of a function to be integrated

%   Output: Result of integration

%-------------------------------------------------------------------

  clc                                          % clear the screen
  disp('  Numerical Integration using trapezoidal rule'); disp(' ');

  x = input('Enter the values of x = ');   % Values of x are entered here
  y = input('Enter the values of y = ');   % Values of y are entered here

  I = trapz(x,y);                      %   Integrate using trapezoidal rule

  disp(' ');
  disp('The result of integration is as follows...'); disp(' ');
  disp(I);                             %   Display the result of integration

%-------------- End of the program -----------------------------------
```

As shown in the above program, the two `input` statements accept the values of x and $y = f(x)$ of the function to be integrated. The next statement is,

```
I = trapz (x,y);
```

This statement uses `trapz` function of MATLAB. It calculates integration of given values of x and y using trapezoidal rule. Then the result of integration is displayed on the screen.

To test the program :

Let us consider the data of example 7.2.2 to test this program. This data is,

x	$x_0 = 4$	$x_1 = 4.2$	$x_2 = 4.4$	$x_3 = 4.6$	$x_4 = 4.8$	$x_5 = 5.0$	$x_6 = 5.2$
$y = \ln x$	$y_0 = 1.3863$	$y_1 = 1.4351$	$y_2 = 1.4816$	$y_3 = 1.5260$	$y_4 = 1.5686$	$y_5 = 1.6094$	$y_6 = 1.6486$

Run the MATLAB program and give above data as input. Then the results produced by the MATLAB program are as follows :

```
%--------------------- Results ------------------------------------------
  Numerical Integration using trapezoidal rule

Enter the values of x = [4.0 4.2 4.4 4.6 4.8 5.0 5.2]
Enter the values of y = [1.3863 1.4351 1.4816 1.5260 1.5686 1.6094 1.6486]

The result of integration is as follows...

   1.8276
%-------------------------------------------------------------------------
```

As shown above, the result of integration is 1.8276. In example 7.2.2, the result obtained is 1.82764.

MATLAB also has a `quad` function for integration. This function uses adaptive simpson's rules of integration. It adapts, number of intervals and type of rule. This function gives more accurate results.

Computer Exercise

1. Write the program in C language to differentiate the function numerically. In your program implement the differentiation formula of equation 7.1.6.
2. Modify the programs of Simpson's $\frac{1}{3}$, Simpson's $\frac{3}{8}$ and trapezoidal rules of integration to calculate an error in computation for given value of z.

   where $x_0 \le z \le x_n$.
3. Make the facility in the simpson's $\frac{1}{3}$ rule of integration program to select the value of 'h' depending on given permissible error.
4. Write a general program in C to find the RMS value of current or voltage using Simpson's $\frac{3}{8}$ rule of integration.

❑❑❑

8

Solution to Ordinary Differential Equations

Differential equations play an important role in science and engineering. Most of the important phenomena and fundamental laws are described with the help of differential equations. When a differential equation contains all the derivatives with respect to a single variable, then it is called ordinary differential equation. For example consider,

$$m \frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx = 0$$

This is ordinary differential and it contains derivatives only with respect to 't'. If a differential equation contains derivatives with respect to two or more variables, then it is called partial differential equation.

The most important Faraday's law of electromagnetic induction is also defined with the help of differential equation .e.,

Voltage drop in inductance $= L\frac{di}{dt}$

Here i is current through an inductance.

It becomes necessary to solve those differential equations. For example if we want to calculate current through inductance, then we have to solve above equation.

Numerical methods can be used to obtain the solutions of ordinary differential equations. We will describe various methods in the subsequent sections.

8.1 Taylor Series Method

Consider the first order differential equation

$$y' = f(x,y) \qquad \text{... (8.1.1)}$$

Here $y' = \frac{dy}{dx}$ is the first derivative. Let the initial conditions be given as,

$$y = y_0 \quad \text{at} \quad x = x_0$$

Let $y = f(x)$ be the solution of equation 8.1.1.

Let's expand $y = f(x)$ using taylor series around $x = x_0$ then we get,

$$y = f(x_0) + (x - x_0) f'(x_0) + \frac{(x-x_0)^2}{2!} f''(x_0) + \frac{(x-x_0)^3}{3!} f'''(x_0) + \dots \qquad \text{... (8.1.2)}$$

In this equation,

$y_0 = f(x_0)$, $\quad y'_0 = f'(x_0)$, $\quad y''_0 = f''(x_0)$

$y'''_0 = f'''(x_0)$ and so on.

Here,

y'_0 is the first derivative of y taken at $x = x_0$

y''_0 is the second derivative of y taken at $x = x_0$

y'''_0 is the third derivative of y taken at $x = x_0$ ..... and so on.

Thus we can write equation 8.1.2 as,

$$\boxed{y = y_0 + (x - x_0)\, y'_0 + \frac{(x-x_0)^2}{2!} y''_0 + \frac{(x-x_0)^3}{3!} y'''_0 + \ldots} \qquad \ldots (8.1.3)$$

If derivatives at x_0 are known, then this equation gives solution of differential equation using taylor series.

The error E_n in taylor series solution is given as,

$$E_n = \frac{(x-x_i)^{n+1}}{(n+1)!} f^{(n+1)} (x_i + \theta h)$$

Here $\quad h = x - x_i$

& $\quad 0 < \theta < 1$

$\therefore$ We can write expression for E_n as

$$\boxed{E_n = \frac{h^{n-1}}{(n+1)!} f^{(n+1)} (x_i + \theta h)} \qquad \ldots (8.1.4)$$

This expression gives error in taylor series solution.

8.1.1 Solved Examples

Ex.8.1.1 *Obtain the solution of $y' = 3x + y^2$ using Taylor's series method $y = 1$ when $x = 0$. Find the value of y at $x = 0.1$.*

Sol. : Let's first calculate first four derivatives of y.

$$y' = 3x + y^2$$

$$y'' = \frac{d}{dx}(y') = \frac{d}{dx}(3x + y^2) = 3 + 2yy'$$

$$y''' = \frac{d}{dx}(y'') = \frac{d}{dx}(3 + 2yy') = 2y'y' + 2yy''$$

$$= 2(y')^2 + 2yy''$$

$$y^{(4)} = \frac{d}{dx} y''' = \frac{d}{dx}\left[2(y')^2 + 2yy''\right]$$

$$= 4y'y'' + 2y'y'' + 2yy'''$$

$$= 6y'y'' + 2yy'''$$

Thus we have,

$$y' = 3x + y^2$$

$$y'' = 3 + 2yy'$$

$$y''' = 2(y')^2 + 2yy''$$

$$y^{(4)} = 6y'y'' + 2yy'''$$

It is given that initially, $x_0 = 0, y_0 = 1$. Putting those values in above equations we get,

$$y'_0 = y'\Big|_{x_0=0, y_0=1} = 3\times 0 + 1 = 1 \qquad \therefore \boxed{y'_0 = 1}$$

$$y''_0 = y''\Big|_{x_0=0, y_0=1} = 3 + 2\times 1 \times 1 = 5 \qquad \therefore \boxed{y''_0 = 5}$$

$$y'''_0 = y'''\Big|_{x_0=0, y_0=1} = 2(1)^2 + 2\times 1\times 5 = 12 \qquad \therefore \boxed{y'''_0 = 12}$$

$$y^{(4)}_0 = y^{(4)}\Big|_{x_0=0, y_0=1} = 6\times 1\times 5 + 2\times 1\times 12 = 54 \qquad \therefore \boxed{y^{(4)}_0 = 54}$$

From equation 8.1.3 the solution is given as,

$$y = y_0 + (x - x_0)\, y'_0 + \frac{(x-x_0)^2}{2!} y''_0 + \frac{(x-x_0)^3}{3!} y'''_0 + \frac{(x-x_0)^4}{4!} y^{(4)}_0 + \ldots$$

Putting the values in above equation

$$y = 1 + (x-0) + \frac{(x-0)^2}{2}\times 5 + \frac{(x-0)^3}{6}\times 12 + \frac{(x-0)^4}{24}\times 54$$

$$= 1 + x + \frac{5x^2}{2} + 2x^3 + \frac{9}{4}x^4 \qquad \ldots (8.1.5)$$

Here we have truncated the fifth and other higher order derivatives. Therefore our answer will be correct upto 4 digits. To calculate y at x = 0.1,

Putting the value of x = 0.1 in equation 8.1.5 above,

$$y = 1 + 0.1 + \frac{5(0.1)^2}{2} + 2(0.1)^3 + \frac{9}{4}(0.1)^4$$

$$= 1.127225$$

Ex.8.1.2 *Find out the solution of* $\frac{d^2y}{dx^2} + xy = 0$, *subject to* $x = 0$, $y = c$ *and* $\frac{dy}{dx} = 0$

Sol. : The given equation is,

$$y'' = -xy$$

$$y''' = -y - xy'$$

$$y^{(4)} = -y' - xy'' - y' = -2y' - xy''$$

$$y^{(5)} = -2y'' - y'' - xy''' = -3y'' - xy'''$$

$$y^{(6)} = -3y''' - xy^{(4)} - y''' = -4y''' - xy^{(4)}$$

It is given that,

$x_0 = 0,$ $\quad y_0 = c$ & $y'_0 = 0$ (initial conditions)

Putting those values in the equations above we get,

$$y''_0 = y''\Big|_{x_0=0,y_0=c\ \&\ y_0=0} = 0 \qquad \therefore \boxed{y''_0 = 0}$$

$$y'''_0 = y''\Big|_{x_0=0,y_0=c\ \&\ y'_0=0} = -c-0 = -c \qquad \therefore \boxed{y'''_0 = -c}$$

Similarly we obtain,

$$y_0^{(4)} = -2y'_0 - x_0\, y''_0 = 0-0 = 0 \qquad \therefore \boxed{y_0^{(4)} = 0}$$

$$y_0^{(5)} = -3y''_0 - x\, y''_0{}' = 0-0 = 0 \qquad \therefore \boxed{y_0^{(5)} = 0}$$

$$y_0^{(6)} = -4y'''_0 - x\, y^{(4)} = -4(-c) - 0 = 4c \qquad \therefore \boxed{y_0^{(6)} = 4c}$$

From equation 8.1.3 solution using taylor's series is given as,

$$y = y_0 + (x-x_0)\, y'_0 + \frac{(x-x_0)^2}{2!} y''_0 + \frac{(x-x_0)^3}{3!} y'''_0 + \ldots$$

Putting the values of derivatives in above equation,

$$y = c + 0 + 0 + \frac{x^3}{3!}(-c) + \frac{x^6}{6!}(4c) + \ldots..$$

$$\therefore \qquad y = c\left[1 - \frac{x^3}{3!} + 4\,\frac{x^6}{6!} + \ldots\ldots\right]$$

This is the required solution.

Ex.8.1.3 *Find out the solution of $y' = 2y + 3e^x$ using taylor's series. Initial values are given as $x_0 = 0$ and $y_0 = 1$. Find out value of y for 0 (0.1) 0.3.*

Sol. : First calculate the derivatives upto 4^{th} derivative

$$\left.\begin{aligned} y' &= 2y + 3e^x \\ y'' &= 2y' + 3e^x \\ y''' &= 2y'' + 3e^x \\ y^{(4)} &= 2y''' + 3e^x \end{aligned}\right\} \qquad \ldots (8.1.6)$$

Here 0 (0.1) 0.3 means we have to find value of y from 0 to 0.3 in steps of 0.1.

Now,

$$y'_0 = y'\,|_{x_0=0,y_0=1} = 2y_0 + 3e^{x_0} = 2+3 = 5 \qquad \therefore \boxed{y'_0 = 5}$$

$$y''_0 = y''\,|_{x_0=0,y_0=1} = 2y'_0 + 3e^{x_0} = 2\times 5 + 3 = 13 \qquad \therefore \boxed{y''_0 = 13}$$

Similarly we can obtain,

$$\boxed{y_0''' = 29 \quad \text{and} \quad y_0^{(4)} = 61}$$

From equation 8.1.3 taylor series solution is given as,

$$y = y_0 + (x - x_0)\, y'_0 + \frac{(x-x_0)^2}{2!} y_0'' + \frac{(x-x_0)^3}{3!} y_0''' + \ldots \qquad \ldots (8.1.7)$$

Putting the values in above equation we get,

$$y = 1 + (x-0) \times 5 + \frac{(x-0)^2}{2!} \times 13 + \frac{(x-0)^3}{3!} \times 29 + \frac{(x-0)^4}{4!} \times 61$$

$$\therefore \quad y = 1 + 5x + \frac{13}{2} x^2 + \frac{29}{6} x^3 + \frac{61}{24} x^4 \qquad \ldots (8.1.8)$$

This is the required solution.

To find y at x = 0.1

Substitute x = 0.1 in equation 8.1.7 to get y i.e.,

$$y = 1 + 5 \times 0.1 + \frac{13}{2} \times (0.1)^2 + \frac{29}{6} \times (0.1)^3 + \frac{61}{24} (0.1)^4$$

$$\therefore \quad y = 1.5701$$

To calculate y at x = 0.2

For this assume $x_0 = 0.1 \quad \therefore y_0 = 1.5701$

Now let's calculate the derivatives of equation 8.1.6 again with these new initial conditions. i.e.

$$y'_0 = y' \Big|_{x_0 = 0.1,\, y_0 = 1.5701} = 2 \times 1.5701 + 3\, e^{0.1} = 6.4557$$

$$y_0'' = y'' \Big|_{x_0 = 0.1,\, y_0 = 1.5701} = 2 \times 6.4557 + 3\, e^{0.1} = 16.2270$$

Similarly we obtain,

$$y_0''' = 2y_0''' + 3e^{x_0} = 35.7695$$

$$y_0^{(4)} = 2y_0''' + 3e^{x_0} = 74.8545$$

Putting the values of derivatives in taylor series equation given by equation 8.1.7 we get,

$$y = 1.5701 + (x - 0.1) \times 6.4557 + \frac{(x-0.1)^2}{2!} \times 16.2270 + \frac{(x-0.1)^3}{3!} \times 35.7695 + \frac{(x-0.1)^4}{4!} \times 74.8545$$

to get y at x = 0.2, put x = 0.2 in above equation. We get,

$$y = 2.3025$$

To calculate y at x = 0.3

For this assume $x_0 = 0.2 \quad \therefore y_0 = 2.3025$

According to those new initial values, calculate derivatives of equation 8.1.6. i.e.,

$$y'_0 = y'\Big|_{x_0 = 0.2,\ y_0 = 2.3025} = 2y_0 + 3e^{x_0} = 8.2692$$

$$y''_0 = y''\Big|_{x_0 = 0.2,\ y_0 = 2.3025} = 2y'_0 + 3e^{x_0} = 20.2026$$

Similarly we can calculate,

$$y''' = 44.0694$$

& $$y_0^{(4)} = 91.8030$$

Putting these values of derivative in taylor series equation of equation 8.1.7 we get,

$$y = 2.3025 + (x - 0.2) \times 8.2692 + \frac{(x-0.2)^2}{2!} \times 20.2026 + \frac{(x-0.2)^3}{3!} \times 44.0694 + \frac{(x-0.2)^4}{4!} \times 91.8030$$

In this equation put x = 0.3 to find value of y.

then we get,

$$y = 3.2382$$

Thus we have,

x	y
0.1	1.5701
0.2	2.3025
0.3	3.2382

Ex.8.1.4 *Find 1 (0.1) 1.2 [i.e. Initial value = 1, $h = 0.1$ final value = 1.2], the solution of $\frac{dy}{dx} = x + y$ through (1, 0) by Taylor's series.*

Sol. : The given function is,

$$y' = x + y$$

Second derivative of y will be,

$$y'' = \frac{d}{dx}(y') = \frac{d}{dx}(x+y) = 1 + y'$$

$$y''' = \frac{d}{dx}(y'') = \frac{d}{dx}(1+y') = y''$$

$$y^{(4)} = \frac{d}{dx}y''' = y'''$$

Step 1 : To find y at $x = 1.1$

At the beginning of this step we have,

$$x_0 = 1, \quad y_0 = 0$$

$$x_0 = 1, \quad y_0 = 0$$

And we have to find y in steps of $h = 0.1$ up to $x = 1.2$. Hence first we have to find y for $x = 1.1$.

Hence let us first calculate $y_0^{'}$, $y_0^{''}$, $y_0^{'''}$, $y_0^{(4)}$. i.e.,

$$y_0^{'} = y' \Big|_{x=1,\ y=0}$$

$$= x = y = 1 + 0 = 1 \qquad \therefore \boxed{y_0^{'} = 1}$$

$$y_0^{''} = y'' \Big|_{x=1,\ y=0,\ y'=1}$$

$$= 1 + y' = 1 + 1 = 2 \qquad \therefore \boxed{y_0^{''} = 2}$$

$$y_0^{'''} = y'' = 2 \qquad \therefore \boxed{y_0^{'''} = 2}$$

$$y_0^{(4)} = y''' = 2 \qquad \therefore \boxed{y_0^{(4)} = 2}$$

From equation 8.1.3 solution is given as,

$$y = y_0 + (x - x_0) y_0^{'} + \frac{(x - x_0)^2}{2!} y_0^{''} + \frac{(x - x_0)^3}{3!} y_0^{'''} + \frac{(x - x_0)^4}{4!} y_0^{(4)} + \ldots$$

In this equation we will consider derivatives only upto 4^{th} order. With $x = 1.1$ and putting other values in above equation, we get,

$$y = 0 + (1.1 - 1)(1) + \frac{(1.1 - 1)^2}{2!}(2) + \frac{(1.1 - 1)^3}{3!}(2) + \frac{(1.1 - 1)^4}{4!}(2)$$

$$= 0.1103416$$

Step 2 : To find y at $x = 1.2$

In the previous step we obtained,

$x = 1.1, \quad y = 0.1103416$

For this step let $x_0 = 1.1$ and $y_0 = 0.1103416$

Now let us obtain y at $x = x_0 + h = 1.1 + 0.1$ i.e. $x = 1.2$.

Hence we have to calculate $y_0^{'}$, $y_0^{''}$, $y_0^{'''}$ and $y_0^{(4)}$ for this step with new initial conditions. i.e.,

$$y_0^{'} = y' \Big|_{x=1.1,\ y=0.1103416}$$

$$= x + y = 1.1 + 0.1103416$$

$$= 1.2103416 \qquad \therefore \boxed{y_0^{'} = 1.2103416}$$

$$y_0^{''} = y'' \Big|_{x=1.1,\ y=0.1103416,\ y'=1.2103416}$$

$$= 1+y'=1+1.2103416$$
$$= 2.2103416 \qquad \therefore \boxed{y_0'' = 2.2103416}$$

$$y_0''' = y'' = 2.2103416 \qquad \therefore \boxed{y_0''' = 2.2103416}$$

$$y_0^{(4)} = y''' = 2.2103416 \qquad \therefore \boxed{y_0^{(4)} = 2.2103416}$$

Taylor's series of equation 8.1.3 is given as,

$$y = y_0 + (x-x_0)y_0' + \frac{(x-x_0)^2}{2!}y_0'' + \frac{(x-x_0)^3}{3!}y_0''' + \frac{(x-x_0)^4}{4!}y_0^{(4)} + \ldots..$$

In this equation we will consider the derivatives only upto 4^{th} order. With $x=1.2$ and putting other values in above equation we get,

$$y = 0.1103416 + (1.2-1.1)(1.2103416) + \frac{(1.2-1.1)^2}{2!}(2.2103416)$$
$$+ \frac{(1.2-1.1)^3}{3!}(2.2103416) + \frac{(1.2-1.1)^4}{4!}(2.2103416)$$
$$= 0.242805$$

Thus the results are

x	y
1	0
1.1	0.1103416
1.2	0.242805

Ex. 8.1.5 *Solve the following equation by Taylor series method :*

$$xy' = x - y$$

Given $y(2) = 2$. *Find* y *at* $x = 2.1$. *Assume step size if required.*

Sol. : The given initial values are :

$$x_0 = 2, \qquad y_0 = 2$$

We have to find y at $x = 2.1$.

The given differential equation is,

$$xy' = x - y$$

$$\therefore \quad y' = 1 - \frac{y}{x} \qquad \ldots (8.1.9)$$

Let us calculate second, third and fourth derivatives of y. i.e.,

$$y'' = \frac{d}{dx}(y')$$

$$= \frac{d}{dx}\left(1-\frac{y}{x}\right)$$

$$= -\frac{y'}{x}+\frac{y}{x^2} \qquad \dots (8.1.10)$$

Similarly,

$$y''' = \frac{d}{dx}(y'')$$

$$= \frac{d}{dx}\left(-\frac{y'}{x}+\frac{y}{x^2}\right)$$

$$= -\frac{y''}{x}+\frac{y'}{x^2}-\frac{2y}{x^3}+\frac{y'}{x^2}$$

$$= -\frac{y''}{x}+\frac{2y'}{x^2}-\frac{2y}{x^3} \qquad \dots (8.1.11)$$

and

$$y^{(4)} = \frac{d}{dx}(y''')$$

$$= \frac{d}{dx}\left(-\frac{y''}{x}+\frac{2y'}{x^2}-\frac{2y}{x^3}\right)$$

$$= -\frac{y'''}{x}+\frac{y''}{x^2}-\frac{4y'}{x^3}+\frac{2y''}{x^2}+\frac{6y}{x^4}-\frac{2y'}{x^3}$$

$$= -\frac{y'''}{x}+\frac{3y''}{x^2}-\frac{6y'}{x^3}+\frac{6y}{x^4} \qquad \dots (8.1.12)$$

Now let us calculate y_0', y_0'', y_0''', $y_0^{(4)}$ by putting $x_0=2$ and $y_0=2$ in their respective equations.

By $x_0=2$ and $y_0=2$ in equation 8.1.9 we get,

$$y_0' = y'\Big|_{x_0=2,\ y_0=2}$$

$$= 1-\frac{y}{x}$$

$$= 1-\frac{2}{2}=0 \qquad \text{i.e.} \quad \boxed{y_0'=0}$$

From equation 8.1.10,

$$y_0'' = y''\Big|_{x_0=2,\,y_0=2,\,y_0'=0}$$

$$= -\frac{y'}{x}+\frac{y}{x^2}$$

$$= -\frac{0}{2}+\frac{2}{2^2}$$

$$= 0.5 \qquad \text{i.e.} \quad \boxed{y_0''=05}$$

From equation 8.1.11, $y_0''' = y'''\Big|_{x_0=2, y_0=2, y_0'=0, y_0''=0.5}$

$$= -\frac{y''}{x} + \frac{2y'}{x^2} - \frac{2y}{x^3}$$

$$= -\frac{0.5}{2} + \frac{2\,(0)}{2^2} - \frac{2(2)}{2^3} \qquad \text{i.e.} \quad \boxed{y_0''' = -\frac{3}{4}}$$

From equation 8.1.12, $y_0^{(4)} = y^{(4)}\Big|_{x_0=2, y_0=2, y_0'=0, y_0''=0.5, y_0'''=-\frac{3}{4}}$

$$= -\frac{y'''}{x} + \frac{3y''}{x^2} - \frac{6y'}{x^3} + \frac{6y}{x^4}$$

$$= -\frac{\left(-\frac{3}{4}\right)}{2} + \frac{3\,(0.5)}{2^2} - \frac{6\,(0)}{2^3} + \frac{6\,(2)}{2^4}$$

$$= \frac{3}{2} \qquad \text{i.e.} \quad \boxed{y_0^{(4)} = \frac{3}{2}}$$

From equation 8.1.3, the solution is given as,

$$y = y_0 + (x - x_0)\, y_0' + \frac{(x-x_0)^2}{2!}\, y_0'' + \frac{(x-x_0)^3}{3!}\, y_0''' + \frac{(x-x_0)^4}{4!}\, y_0^{(4)} + \ldots$$

We have to find y at $x = 2.1$. Let us put values in above equation. We will neglect 5^{th} and other higher order derivatives. Then above equation will be,

$$y = 2 + (2.1-2)\times 0 + \frac{(2.1-2)^2}{2!}\times 0.5 + \frac{(2.1-2)^3}{3!}\times\left(-\frac{3}{4}\right) + \frac{(2.1-2)^4}{4!}\times\frac{3}{2}$$

$$= 2.0023813$$

Thus at $x = 2.1$, $y = 2.0023813$

Ex. 8.1.6 *Solve* $\frac{dy}{dx} = x - y^2$ *by Taylor's series method to calculate y at* $x = 0.4$ *in two steps.*

Initial values are x = 0, y = 1.

Sol. : We have $y' = \frac{dy}{dx} = x - y^2$

$$y'' = \frac{d}{dx}(y') = \frac{d}{dx}\left(x - y^2\right) = 1 - 2y\,y'$$

$$y''' = \frac{d}{dx}(y'') = \frac{d}{dx}(1 - 2y\,y') = -2y'\,y' - 2y\,y''$$

$$= -2\,(y')^2 - 2y\,y''$$

$$y^{(4)} = \frac{d}{dx}(y''') = \frac{d}{dx}\left[-2\,(y')^2 - 2y\,y''\right]$$

$$= -4\,y'\,y'' - 2y'\,y'' - 2y\,y'''$$

$$= -6\,y'\,y'' - 2y\,y'''$$

Step 1 : To find y at x = 0.2

We have $x_0 = 0,\ y_0 = 1$

We have to find y at $x = x_0 + h$. Since we have to calculate y at $x = 0.4$ in two steps. We will consider $h = 0.2$. Hence, $x = x_0 + h = 0 + 0.2 = 0.2$

Now let us calculate $y_0', y_0'', y_0''', y_0^{(4)}$. i.e.,

$$y_0' = y'|_{x_0=0,\ y_0=1}$$

$$= x - y^2 = 0 - (1)^2 = -1 \qquad \therefore \boxed{y_0' = -1}$$

$$y_0'' = y''|_{x_0=0,\ y_0=1,\ y_0'=-1}$$

$$= 1 - 2y\ y' = 1 - 2(1)(-1) = 3 \qquad \therefore \boxed{y_0'' = 3}$$

$$y_0''' = y'''|_{x_0=0,\ y_0=1,\ y_0''=-1,\ y_0'''=3}$$

$$= -2\,(y')^2 - 2y\ y''$$

$$= -2(-1)^2 - 2(1)(3) = -8 \qquad \therefore \boxed{y_0''' = -8}$$

$$y_0^{(4)} = -6\,y'\ y'' - 2y\ y'''$$

$$= -6\,(-1)(3) - 2(1)\,(-8) = 34 \qquad \therefore \boxed{y_0^{(4)} = 34}$$

From equation 8.1.3, Taylor's series is given as,

$$y = y_0 + (x - x_0)\,y_0' + \frac{(x-x_0)^2}{2!}\,y_0'' + \frac{(x-x_0)^3}{3!}\,y_0''' + \ldots.$$

Let us write above equation upto 4^{th} derivative of y. i.e.,

$$y = y_0 + (x - x_0)\,y_0' + \frac{(x-x_0)^2}{2!}\,y_0'' + \frac{(x-x_0)^3}{3!}\,y_0''' + \frac{(x-x_0)^4}{4!}\,y_0^{(4)}$$

Putting values in above equation,

$$y = 1 + (0.2 - 0)\,(-1) + \frac{(0.2-0)^2}{2!}(3) + \frac{(0.2-0)^3}{3!}(-8) + \frac{(0.2-0)^4}{4!}\,(34)$$

$$= 0.8729333$$

Step 2 : To obtain y at x = 0.4

In the last step we obtained

$x = 0.2, \quad y = 0.8729333$

For this step we assume,

$x_0 = 0.2, \quad y_0 = 0.8729333$

And we have to find y at $x = x_0 + h = 0.2 + 0.2$ i.e. $x = 0.4$. Now let us calculate $y_0', y_0'', y_0''', y_0^{(4)}$ again for these new initial conditions. i.e.,

$$y_0' = x - y^2 = 0.2 - (0.8729333)^2$$

$$= -0.5620125 \qquad \therefore \boxed{y_0' = -0.5620125}$$

$$y_0'' = 1 - 2y\ y'$$

$$= 1 - 2\,(0.8729333)\,(-0.5620125)$$

$$= 1.9811989 \qquad \therefore \boxed{y_0'' = 1.9811989}$$

$$y_0''' = -2\,(y')^2 - 2y\ y''$$

$$= -2\,(-0.5620125)^2 - 2(0.8729333)(1.9811989)$$

$$= -2.827193 \qquad \therefore \boxed{y_0''' = -2.827193}$$

$$y_0^{(4)} = -6\,y'\,y'' - 2y\ y'''$$

$$= -6\,(-0.5620125)\,(1.9811989) - 2\,(0.8729333)\,(-2.827193)$$

$$= 11.616653 \qquad \therefore \boxed{y_0^{(4)} = 11.616653}$$

Taylor's series of equation 8.1.3 can be written upto 4 derivatives of y as follows :

$$y = y_0 + (x - x_0)\,y_0' + \frac{(x-x_0)^2}{2!}\,y_0'' + \frac{(x-x_0)^3}{3!}\,y_0''' + \frac{(x-x_0)^4}{4!}\,y_0^{(4)}$$

Putting values in above equation,

$$y = 0.872933 + (0.4 - 0.2)\,(-0.5620125) + \frac{(0.4-0.2)^2}{2!}\,(1.9811989)$$

$$+ \frac{(0.4-0.2)^3}{3!}\,(-2.827193) + \frac{(0.4-0.2)^4}{4!}\,(11.616653)$$

$$= 0.7956103$$

Thus we have the results as follows :

x	y
0	1
0.2	0.8729333
0.4	0.7956103

Exercise

1. *Solve $y' = x + y^2$, given that $y(0) = 0$ using taylor's series method.*

[Ans.: $y = \frac{x^2}{2} + \frac{x^5}{20} + \ldots$]

8.2 Picard's Method of Successive Approximation

Consider the differential equation,

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

with $y = y_0$ at $x = x_0$ are the initial conditions. Integrating the above equation,

$$y = y_0 + \int_{x_0}^{x} f(x,y)\, dx \qquad \text{... (8.2.1)}$$

This equation can be successively solved. It can be written as,

$$y^{(n)} = y_0 + \int_{x_0}^{x} f\left(x, y^{(n-1)}\right) dx \qquad \text{... (8.2.2)}$$

Here $y^{(n)}$ is the n^{th} approximation and

$y^{(n-1)}$ is the $(n-1)^{th}$ i.e. previous approximation.

And $y^{(0)} = y_0$. Following example illustrates this method.

Ex. 8.2.1 *Solve the equation $y' = x + y^2$, with $y(0) = 1$ using picard's method. Determine y at $x = 0.1$.*

Sol. : i) To obtain the solution :

Here at $x_0 = 0$, $y_0 = 1$. Hence $y^{(0)} = 1$.

With $n = 1$, equation 8.2.2 becomes,

$$y^{(1)} = y_0 + \int_{x_0}^{x} f\left(x, y^{(0)}\right) dx$$

Putting values in above equation,

$$y^{(1)} = 1 + \int_{0}^{x} \left\{x + \left[y^{(0)}\right]^2\right\} dx$$

$$= 1 + \int_{0}^{x} (x+1)\, dx$$

$$= 1 + \left[\frac{x^2}{2} + x\right]_0^x = 1 + x + \frac{x^2}{2}$$

With $n = 2$ in equation 8.2.2, we get,

$$y^{(2)} = y_0 + \int_{x_0}^{x} f\left(x, y^{(1)}\right) dx$$

Putting values in above equation,

$$y^{(2)} = 1 + \int_{0}^{1} \left\{x + \left[y^{(1)}\right]^2\right\} dx$$

$$= 1 + \int_{0}^{x} \left\{x + \left(1 + x + \frac{x^2}{2}\right)^2\right\} dx$$

$$= 1 + x + \frac{3}{2}x^2 + \frac{2}{3}x^3 + \frac{1}{4}x^4 + \frac{1}{20}x^5$$

More approximations can be obtained for better accuracy.

ii) To determine y at $x = 0.1$

Putting $x = 0.1$ in equation of $y^{(2)}$, we get,

$$y = 1 + (0.1) + \frac{3}{2}(0.1)^2 + \frac{2}{3}(0.1)^3 + \frac{1}{4}(0.1)^4 + \frac{1}{20}(0.1)^5$$

$$= 1.1156922$$

Ex. 8.2.2 *Solve* $y' = \dfrac{x^2}{1+y^2}$, *with the initial condition* $y = 0$ *when* $x = 0$. *Find the value of* y *or* $x = 0.25$ *and 0.5 correct to three decimal places.*

Sol. : With $n = 1$ in equation 8.2.2 we get,

$$y^{(1)} = y_0 + \int_{x_0}^{x} f\left(x, y^{(0)}\right) dx$$

Here $y_0 = 0$, $x_0 = 0$ and $y^{(0)} = y_0 = 0$. Hence above equation becomes,

$$y^{(1)} = \int_0^x f(x, 0)\, dx$$

We have $\quad f(x, y) = \dfrac{x^2}{1+y^2}$. Hence,

$$f(x, 0) = \frac{x^2}{1+0} = x^2$$

Therefore $y^{(1)}$ becomes,

$$y^{(1)} = \int_0^x x^2\, dx$$

$$= \left[\frac{x^3}{3}\right]_0^x = \frac{x^3}{3}$$

With $n = 2$ in equation 8.2.2, we get,

$$y^{(2)} = y_0 + \int_{x_0}^{x} f\left(x, y^{(1)}\right) dx$$

Here $y_0 = 0$, $x_0 = 0$ and $y^{(1)} = \dfrac{x^3}{3}$. We know that,

$$f(x, y) = \frac{x^2}{1+y^2}$$

$$\therefore \qquad f\left(x, y^{(1)}\right) = \frac{x^2}{1+\left(\frac{x^3}{3}\right)^2} = \frac{x^2}{1+\frac{x^6}{9}}$$

Therefore $y^{(2)}$ becomes,

$$y^{(2)} = \int_0^x \frac{x^2}{1+\frac{x^6}{9}}\, dx$$

Solving the above integration we get

$$y^{(2)} = \frac{1}{3}x^3 - \frac{1}{81}x^9 + \ldots$$

This is an infinite series. Since we want the value of y correct to three decimal places, we have to determine the number of terms of the series to be considered. For the answer to be correct to three decimal places, the maximum error will be, 0.0005.

Let us solve,

$$\frac{1}{81}x^9 \leq 0.0005$$

$$\therefore \qquad x \leq 0.70028$$

This shows that for values of $x \leq 0.7$, the second term $\frac{1}{81}x^9$ has the value less than the error (0.0005). Hence we can truncate 2^{nd} and all higher terms. Hence,

$$y = \frac{1}{3}x^3$$

The value of y in above equation will be correct to three decimal places provided that $0 \leq x \leq 0.7$. Hence

$$y\,(0.25) = \frac{1}{3}(0.25)^3 = 0.0052083$$

and

$$y\,(0.5) = \frac{1}{3}(05)^3 = 0.0416666$$

University Questions

1. *Using Picards method, solve* $\frac{dy}{dx} = 1 + xy$ *with y(0) = 2. Find y(0.1), y(0.2) y(0.3).* **[May - 2003]**

2. Using Picard's method, obtain the solution of $\frac{dy}{dx} = x\,(1 + x^3y)$, $y(0) = 3$ **[Dec. - 2004]**

8.3 Euler's Method

Consider the differential equation

$$\frac{dy}{dx} = f(x, y) \qquad \ldots (8.3.1)$$

Let $y = \phi(x)$ be the solution of the above equation.

Let's say that there are $x_i, y_i, i = 0, 1, 2, \ldots n+1$ equispaced points on the curve $y = \phi(x)$. i.e.,

$$y_i = \phi(x_i) \qquad \text{for } i = 0, 1, 2, \ldots n, n+1$$

For $i = n+1$,

$$y_{n+1} = \phi(x_{n+1}) \qquad \ldots (8.3.2)$$

When the points are equally spaced, then we can write,

$$x_{n+1} - x_n = h$$

$$\therefore \quad x_{n+1} = x_n + h \qquad \ldots (8.3.3)$$

Putting this value of x_{n+1} in equation 8.3.2,

$$y_{n+1} = \phi(x_n + h)$$

Let's expand this equation around x_n,

$$\therefore \quad y_{n+1} = \phi(x_n) + h\,\phi'(x_n) + \frac{h^2}{2!}\phi''(x_n) + \ldots\ldots \qquad \ldots (8.3.4)$$

$$= \phi(x_n) + h\,\phi'(x_n)$$

(Neglecting second and higher order terms)

$$\therefore \quad \boxed{y_{n+1} = y_n + h\,f(x_n, y_n)} \qquad \ldots (8.3.5)$$

Since

$$\phi(x_n) = y_n \quad \text{and}$$

$$\phi'(x_n) = \frac{dy_n}{dx_n} = f(x_n, y_n) \qquad \text{From equation 8.3.1}$$

Equation 8.3.5 gives the next value of y from present value and its derivative $f(x_n, y_n)$ at present value. This is known as **Euler's formula.**

8.3.1 Error in Euler's Method

Recall equation 8.3.4 of last subsection i.e.,

$$y_{n+1} = \phi(x_n) + h\,\phi'(x_n) + \frac{h^2}{2!}\phi''(x_n) + \ldots\ldots$$

$$= \phi(x_n) + h\,\phi'(x_n) + \frac{h^2}{2!}\phi''(\theta_n)$$

Here we have neglected terms higher than second and $x_n \le \theta_n \le x_{n+1}$. Therefore error in the Euler's method due to truncation of second derivative term is given as,

$$\boxed{E_n = \frac{h^2}{2!}\phi''(\theta_n)} \qquad \ldots (8.3.6)$$

or

$$E_n = \frac{h^2}{2!} f'(\theta_n, y_n) \qquad \ldots (8.3.7)$$

8.3.2. Graphical Interpretation of Euler's Method

Fig. 8.3.1 shows the interpretation of Euler's method.

From equation 8.3.5 we have Euler's formula as,

$$y_{n+1} = y_n + h\,f(x_n, y_n)$$

i.e. $$y_{n+1} = y_n + h\frac{dy_n}{dx_n} \quad \text{From equation 8.3.1} \qquad \dots (8.3.8)$$

$\frac{dy}{dx}$ is the slope of y = f (x) at (x_n, y_n).

Therefore we can write equation 8.3.8 as,

New value of y = Old value of y + Step size × Slope of y at old value

Observe in the figure that the new predicted value of y based on slope, step size and old value contains an error. This error can be minimized by selecting smaller step size h. In Fig. 8.3.1 if we have the step size 'h', then error will be reduced by considerable amount.

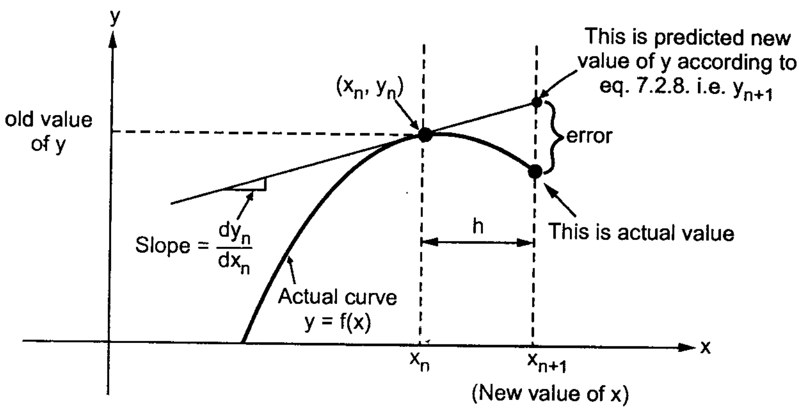


Fig. 8.3.1 Graphical interpretation of Euler's method

8.3.3 Solved Examples

Ex. 8.3.1 *Using Euler's method, obtain the solution of $y' = x - y$, given $x_0 = 0$, $y_0 = 1$ at* $x = 0.6$ taking h = 0.2.

Sol. : Here we have

$$f(x, y) = x - y, \qquad x_0 = 0 \quad \& \quad y_0 = 1$$

From equation 8.3.5 Euler's formula is given as,

$$y_{n+1} = y_n + h f(x_n, y_n)$$

i.e. $$y_{n+1} = y_n + h(x_n - y_n)$$

Let's prepare the table of calculations as shown below.

$$h = 0.2$$

n $n \neq 0$	$x_n = x_{n-1} + h$ $n \neq 0$	y_n	$f(x_n, y_n) = x_n - y_n$	$y_{n+1} = y_n + h(x_n - y_n)$
–	$x_0 = 0$ (Given)	$y_0 = 1$	$x_0 - y_0 = (0 - 1)$ $= -1$	$y_1 = y_0 + h(x_0 - y_0)$ $= 1 + 0.2(-1) = 0.8$
1	$x_1 = x_0 + h$ $= 0 + 0.2 = 0.2$	$y_1 = 0.8$	$x_1 - y_1 = (0.2 - 0.8)$ $= -0.6$	$y_2 = y_1 + h(x_1 - y_1)$ $= 0.8 + 0.2(-0.6)$ $= 0.68$
2	$x_2 = x_1 + h$ $= 0.2 + 0.2 = 0.4$	$y_2 = 0.68$	$x_2 - y_2 = (0.4 - 0.68)$ $= -0.28$	$y_3 = y_2 + h(x_2 - y_2)$ $= 0.68 + 0.2$ $(-0.28) = 0.624$
3	$x_3 = x_2 + h$ $= 0.4 + 0.2 = 0.6$	$y_3 = 0.624$		

Thus at x = 0.6 , y = 0.624

Ex. 8.3.2 *Using Eulers method, find an approximate value of y corresponding to x = 1.* $y' = x + y$ & $y = 1$ *when* $x = 0$.

Sol. : Let's consider h = 0.1

From equation 8.3.5 we have Euler's formula as,

$$y_{n+1} = y_n + h f(x_n, y_n)$$

Let's prepare the table of calculations as follows –

n	$x_n = x_{n-1} + h$	y_n	$f(x_n, y_n) = x_n + y_n$	$y_{n+1} = y_n + h(x_n + y_n)$
–	$x_0 = 0$	$y_0 = 1$	$x_0 + y_0 = 0 + 1 = 1$	$y_1 = y_0 + h(x_0 + y_0)$ $= 1 + 0.1(1) = 1.1$
1	$x_1 = x_0 + h$ $= 0 + 0.1 = 0.1$	$y_1 = 1.1$	$x_1 + y_1 = 0.1 + 1.1 = 1.2$	$y_2 = y_1 + h(x_1 + y_1)$ $= 1.1 + 0.1(1.2) = 1.22$
2	$x_2 = 0.2$	$y_2 = 1.22$	$x_2 + y_2 = 1.42$	$y_3 = 1.362$
3	$x_3 = 0.3$	$y_3 = 1.362$	$x_3 + y_3 = 1.662$	$y_4 = 1.5282$
4	$x_4 = 0.4$	$y_4 = 1.5282$	$x_4 + y_4 = 1.9282$	$y_5 = 1.7210$
5	$x_5 = 0.5$	$y_5 = 1.7210$	$x_5 + y_5 = 2.2210$	$y_6 = 1.9431$
6	$x_6 = 0.6$	$y_6 = 1.9431$	$x_6 + x_6 = 2.5431$	$y_7 = 2.1974$
7	$x_7 = 0.7$	$y_7 = 2.1974$	$x_7 + y_7 = 2.8974$	$y_8 = 2.4871$
8	$x_8 = 0.8$	$y_8 = 2.4871$	$x_8 + y_8 = 3.2871$	$y_9 = 2.8158$
9	$x_9 = 0.9$	$y_9 = 2.8158$	$x_9 + y_9 = 3.7158$	$y_{10} = 3.1874$
10	$x_{10} = 1.0$	$y_{10} = 3.1874$		

Thus at x = 1.0, y = 3.1874

8.3.4 Effect of Step Size on Euler's Method

In the last examples we selected some small value of h and then we applied Euler's formula. This value of 'h' is very important in Euler's method from stability point of view. Consider the function shown in Fig. 8.3.2. In Euler's method we know that,

New value of y = Old value of y + step size × slope of y at old value

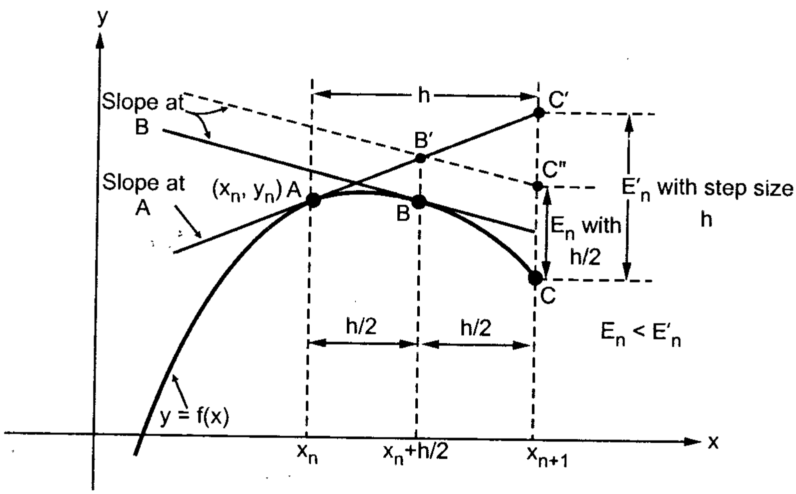


Fig. 8.3.2 Effect of reduction in step size in Euler's method

Effect with step size h :

Let's say that step size is h. Then applying Euler's formula at point A (x_n, y_n) we get point C'. Actual point on the curve is C. Observe that there is a large difference between actual value C and calculated value C'.

Thus E'_n is an error with step size h.

Effect with step size $\frac{h}{2}$:

Let's say that the step size is now $\frac{h}{2}$. Therefore we now need two iterations to reach x_{n+1}.

This is because,

$$\begin{aligned} x_{n+1} &= x_n + h \\ &= x_n + \frac{h}{2} + \frac{h}{2} \end{aligned}$$

Applying Euler's formula at point $A(x_n, y_n)$ with step size $\frac{h}{2}$, we reach to point B'. But actual point on the curve is B. Again we apply Euler's formula at point B on the curve. The tangent line at point B gives the slope. We have to draw a line parallel to this tangent line passing through B'. This line through B' is shown by dotted line. We have to do this because we have to apply Euler's formula at B' taking into consideration the slope of curve at actual value B. Thus the calculated value of y will be C''. But the actual value of y is C on the curve. The difference between the two values is E_n i.e. an error.

Observe that error E_n with step size $\frac{h}{2}$ is smaller than error E'_n with step size h.

It is quite possible that the Euler's method may be unstable. In such case the calculated values does not follow actual values.

Ex. 8.3.3 *Use Euler's method to numerically integrate,*

$$f(x,y) = -2x^3 + 12x^2 - 20x + 8.5, \quad y(0) = 1 \text{ from } x = 0 \text{ to } x = 0.5$$

[May - 96, 8 marks, May - 99, 8 marks, Dec - 2000, 8 marks, May - 2004, 10 marks]

Giving proper example illustrate the effect of stop size on stability of euler's method.

Sol. : Please refer to the theory given in this section. Consider the example,

$$\frac{dy}{dx} = -2x^3 + 12x^2 - 20x + 8.5 \qquad \text{with } y(0) = 1$$

Let's find value of y at x = 0.5

Let's take h = 0.5

$$\therefore \quad x_1 = x_0 + h = 0 + 0.5$$

$$\therefore \quad x_1 = 0.5$$

From Euler's formula we know that

$$y_{n+1} = y_n + h\,f(x_n, y_n) \qquad \text{(equation 8.3.5)}$$

$$\begin{aligned} \therefore \quad y_1 &= y_0 + h\,f(x_0, y_0) \\ &= y_0 + h\left(-2x_0^3 + 12x_0^2 - 20x_0 + 8.5\right) \\ &= 1 + 0.5\,(0 + 0 - 0 + 8.5) \\ &= 5.25 \end{aligned}$$

$$\boxed{\text{Thus with } h = 0.5 \text{ we get } y = 5.25 \text{ at } x = 0.5}$$

Now let's take h = 0.1 :

$x_0 = 0$, $y_0 = 1$ and we have to find y at x = 0.5

From equation 8.3.5 we have Euler's formula as,

$$y_1 = y_0 + h\,f(x_0, y_0)$$

$$= 1 + 0.1\ (0 + 0 - 0 + 8.5)$$
$$= 1.85$$

Similarly we can calculate other values as,

$x_2 = 0.2,\qquad y_2 = 1.85 + (0.1)\, f(x_1, y_1) = 2.5118$

$x_3 = 0.3,\qquad y_3 = 2.5118 + (0.1)\, f(x_2, y_2) = 3.0082$

$x_4 = 0.4,\qquad y_4 = 3.0082 + (0.1)\, f(x_3, y_3) = 3.3608$

$x_5 = 0.5,\qquad y_5 = 3.3608 + (0.1)\, f(x_5, y_5) = 3.59$

Thus with h = 0.1 we get y = 3.59 at x = 0.5.

Let's find exact solution of given equation :

$$\frac{dy}{dx} = -2x^3 + 12x^2 - 20x + 8.5$$

$$\therefore \quad dy = \left(-2x^3 + 12x^2 - 20x + 8.5\right) dx$$

Integrating both sides we get

$$y = -\frac{1}{2}x^4 + 4x^3 - 10x^2 + 8.5\,x + c$$

Put initial conditions y (0) = 1 to find value of constant.

$$1 = c \Rightarrow c = 1$$

$$\therefore \quad y = -\frac{1}{2}x^4 + 4x^3 - 10x^2 + 8.5\,x + 1$$

This is exact solution. Putting the value of x = 0.5, we get,

$$y = -\frac{1}{2}(0.5)^4 + 4\,(0.5)^3 - 10\,(0.5)^2 + 8.5\,(0.5) + 1$$
$$= 3.2187$$

Thus we have,

Value of h	Value of y for x = 0.5
0.5	y = 5.25
0.1	y = 3.59
—	Exact value of y = 3.2187

From the results we observe that the error is largely reduced in value of y when step size is reduced.

Ex. 8.3.4 *Apply Euler's method to solve* $y' = -xy^2$, $y(0) = 2$ *computing upto* $x = 1$ *with* $h = 0.1$.

[Dec - 99, 8 marks, May - 98, 8 marks, Dec - 95, 8 marks, May - 2003, 8 marks]

Sol. : Here we have, $f(x,y) = -xy^2$, $x_0 = 0$ and $y_0 = 2$.

From equation 8.3.5 Euler's formula is given as,

$$y_{n+1} = y_n + h\, f(x_n, y_n)$$

i.e.

$$y_{n+1} = y_n - h\, x_n\, y_n^2$$

Let us prepare the table of calculations as shown below : $(h = 0.1)$

n $n \neq 0$	$x_n = x_{n-1} + h$ $n \neq 0$	y_n	$y_{n+1} = y_n - h\, f(x_n, y_n)$ $\therefore\ y_{n+1} = y_n - h\, x_n\, y_n^2$
–	$x_0 = 0$ (Given)	$y_0 = 2$	$y_1 = y_0 - hx_0\, y_0^2$ $= 2 - (0.1)(0)(2)^2 = 2$
$n = 1$	$x_1 = x_0 + h$ $= 0 + 0.1 = 0.1$	$y_1 = 2$	$y_2 = y - h\, x_1\, y_1^2$ $= 2 - (0.1)(0.1)(2)^2 = 1.96$
$n = 2$	$x_2 = x_1 + h$ $= 0.1 + 0.1 = 0.2$	$y_2 = 1.96$	$y_3 = y_2 - h\, x_2\, y_2^2$ $= 1.96 - (0.1)(0.2)(1.96)^2 = 1.883168$
$n = 3$	$x_3 = 0.3$	$y_3 = 1.883168$	$y_4 = y_3 - h\, x_3\, y_3^2 = 1.7767783$
$n = 4$	$x_4 = 0.4$	$y_4 = 1.7767783$	$y_5 = y_4 - h\, x_4\, y_4^2 = 1.6505006$
$n = 5$	$x_5 = 0.5$	$y_5 = 1.6505006$	$y_6 = y_5 - h\, x_5\, y_5^2 = 1.514293$
$n = 6$	$x_6 = 0.6$	$y_6 = 1.514293$	$y_7 = y_6 - h\, x_6\, y_6^2 = 1.376708$
$n = 7$	$x_7 = 0.7$	$y_7 = 1.376708$	$y_8 = y_7 - h\, x_7\, y_7^2 = 1.2440353$
$n = 8$	$x_8 = 0.8$	$y_8 = 1.2440353$	$y_9 = y_8 - h\, x_8\, y_8^2 = 1.1202254$
$n = 9$	$x_9 = 0.9$	$y_9 = 1.1202254$	$y_{10} = y_9 - h\, x_9\, y_9^2 = 1.007284$
$n = 10$	$x_{10} = 1.0$	$y_{10} = 1.007284$	–

Thus at $x = 1$, $y = 1.007284$ is the required solution.

8.3.5 Solution of Higher Order Equations

Let's now see how to apply Euler's method to solve differential equations of higher order. Consider the equation of the form,

$$\frac{d^2y}{dx^2} = f\left(x, y, \frac{dy}{dx}\right) \qquad \text{... (8.3.9)}$$

Let the initial conditions be,

$$x = x_0, \qquad y = y_0 \;\; \& \;\; \frac{dy}{dx} \text{ at } x = x_0 \text{ be } y_0'$$

Let's write $$\frac{dy}{dx} = z$$

$$\therefore \quad \frac{d^2y}{dx^2} = \frac{dz}{dx} \qquad \text{... (8.3.10)}$$

$\therefore$ Equation 8.3.9 becomes,

$$\frac{dz}{dx} = f(x, y, z)$$

& $$\frac{dy}{dx} = z$$

ϕ is some other function of x, y and z.

$$\therefore \quad \frac{dy}{dx} = \phi(x, y, z)$$

Thus we have two differential equations of the same variables and same initial conditions.

i.e., $$z' = f(x, y, z) \qquad \text{... (8.3.11)}$$

& $$y' = \phi(x, y, z) \qquad \text{... (8.3.12)}$$

With $x = x_0$, $y = y_0$ and $z = z_0$.

Using Euler's formula of equation 8.3.5 is given as,

$$\boxed{y_{n+1} = y_n + h f(x_n, y_n)} \qquad \text{... (8.3.13)}$$

Applying this formula to equation 8.3.11 we get,

$$z_{n+1} = z_n + h f(x_n, y_n, z_n) \qquad \text{... (8.3.14)}$$

Applying Euler's formula of equation 8.3.13 to equation 8.3.12,

$$y_{n+1} = y_n + h\phi(x_n, y_n, z_n) \qquad \text{... (8.3.15)}$$

Equation 8.3.14 & equation 8.3.15 are the Euler's formulae for second order differential equations.

Ex. 8.3.5 *Solve the following second order differential equation*

$$\frac{d^2y}{dx^2} - 2x\frac{dy}{dx} + 4y - 4 = 0$$

Take $x_0 = 0$, $y_0 = 1$, & $\frac{dy}{dx}$ *at* $x = x_0$ *is* -2.

Find y at x = 0.1 (0.1) 0.4 using Euler's method.

Sol. : Rearrange given differential equation as,

$$\frac{d^2y}{dx^2} = 2x\frac{dy}{dx} - 4y + 4$$

Let $\frac{dy}{dx} = z$ then $\frac{d^2y}{dx^2} = \frac{dz}{dx}$

$\therefore$ The given differential equation becomes,

$$\frac{dz}{dx} = 2xz - 4y + 4$$

i.e. $$\frac{dz}{dx} = f(x,y,z) \;\;\&\;\; f(x,y,z) = 2xz - 4y + 4 \qquad \text{... (8.3.16)}$$

and we have, $$\frac{dy}{dx} = z$$

i.e., $$\frac{dy}{dx} = \phi(x,y,z) \;\&\; \phi(x,y,z) = z \qquad \text{... (8.3.17)}$$

Solution of differential equation given by equation 8.3.16 is given using Eulers formula of equation 8.3.14 as,

$$z_{n+1} = z_n + h\,f(x_n, y_n, z_n)$$

$$\therefore \quad z_{n+1} = z_n + h(2x_n z_n - 4y_n + 4) \qquad \text{... (8.3.18)}$$

Now applying Euler's formula of equation 8.3.15 to differential equation of equation 8.3.17 we get,

$$y_{n+1} = y_n + h\,\phi(x_n, y_n, z_n)$$

$$\therefore \quad y_{n+1} = y_n + h(z_n) \qquad \text{... (8.3.19)}$$

It is given that $x_0 = 0, y_0 = 1$ and $\frac{dy}{dx}$ at $x = x_0$ is -2.

i.e. $$\frac{dy}{dx} = z \text{ at } x = x_0 \quad \therefore \quad z_0 = -2$$

We are asked to find y at x = 0.1 (0.1) 0.4 means to find y at values of x from 0.1 to 0.4 in steps of 0.1.

$$\therefore \quad h = 0.1$$

Thus we have,

$x_0 = 0$, $y_0 = 1$, $z_0 = -2$ and $h = 0.1$

And two equations, equation 8.3.18 and equation 8.3.19 to solve.

Let n = 0 in equation 8.3.18 and equation 8.3.19 we have,

From equation 8.3.18 $\Rightarrow$
$$\begin{aligned} z_1 &= z_0 + h[2x_0 z_0 - 4y_0 + 4] \\ &= -2 + 0.1[2\times 0\times(-2) - 4\times 1 + 4] \\ &= -2 \end{aligned}$$

From equation 8.3.19 $\Rightarrow$
$$\begin{aligned} y_1 &= y_0 + h(z_0) \\ &= 1 + 0.1\times(-2) \\ &= 0.8 \end{aligned}$$

Let n = 1

$$x_1 = x_0 + h = 0 + 0.1 = 0.1$$

From equation 8.3.18 $\Rightarrow$ $z_2 = z_1 + h\,[2x_1\, z_1 - 4y_1 + 4]$

$= -2 + 0.1\,[2 \times 0.1 \times (-2) - 4 \times 0.8 + 4] = -1.96$

From equation 8.3.19 $\Rightarrow$ $y_2 = y_1 + h\,(z_1)$

$= 0.8 + 0.1 \times (-2) = 0.6$

n = 2

$$x_2 = x_1 + h = 0.1 + 0.1 = 0.2$$

From equation 8.3.18 $\Rightarrow$ $z_3 = z_2 + h\,[2\,x_2\, z_2 - 4y_2 + 4]$

$= -1.96 + 0.1\,[2 \times 0.2 \times (-1.96) - 4 \times 0.6 + 4]$

$= -1.8784$

From equation 8.3.19 $\Rightarrow$ $y_3 = y_2 + h\,(z_2)$

$= 0.6 + 0.1 \times (-1.96) = 0.404$

n = 3

$$x_3 = x_2 + h = 0.2 + 0.1 = 0.3$$

From equation 8.3.18 $\Rightarrow$ $z_4 = z_3 + h\,[2\,x_3\, z_3 - 4y_3 + 4]$

$= -1.8784 + 0.1\,[2 \times 0.3 \times (-1.8784) - 4 \times 0.404 + 4]$

$= -1.752704$

From equation 8.3.19 $\Rightarrow$ $y_4 = y_3 + h\,(z_3)$

$= 0.404 + 0.1 \times (-1.8784)$

$= 0.21616$

Thus we have values of y tabulates below.

x	y
0	1
0.1	0.8
0.2	0.6
0.3	0.404
0.4	0.21616

Exercise

1. *Use Euler's method to solve the ordinary differential equation.*

$$\frac{dy}{dx} = -2x^3 + 12x^2 - 20x + 8.5$$

From x = 0 to x = 4 with step size of 0.5. The initial condition is y (0) =1.

[Ans. :

x	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0
y	1	5.25	4.5	3.875	4.125	5.25	6.5	6.375	2.625

2. *Using Euler's method find approximate value of y for x = 0.1. It is given that,*

$$\frac{dy}{dx} = x - y^2 \quad \text{And} \quad y(0) = 1$$

[Hint : Here take h = 0.02] **[Ans. : y (0.1) = 0.9113]**

3. *Using Euler's method solve the following equation.*

$$\frac{dy}{dx} = x + y, \quad y(0) = 0$$

Choose h = 0.2 and compute y (0.4) & y (0.6)

[Ans. : y (0.4) = 0.0938, y (0.6) = 0.2258]

8.3.6 Algorithm and C Program

Now we will prepare an algorithm for Euler's method.

Assumption : The function $\frac{dy}{dx} = f(x, y)$ is defined in the program. Algorithm doesnot accept function.

Algorithm :

Step 1 : Read initial values of x_0 **and** y_0

Step 2 : Read value of x at which y is to be calculated.

Step 3 : Read the stepsize h.

Step 4 : Calculate

$$y_{n+1} = y_n + h\, f(x_n, y_n)$$

Step 5 : Increase x_n ***by one step.***

Step 6 : Repeat step 4 and 5 till x_n ***becomes greater than x.***

Step 7 : Display x_n, y_n ***on the screen in tabular format.***

Step 8 : Stop.

Flowchart :

Fig. 8.3.3 shows the flowchart for this algorithm. The flowchart shows all the steps in detail. The program discussed next uses the same logic discussed here.

Please refer Fig. 8.3.3 on page no. 510.

Observe in the flowchart that the value of x_0 is increased by one step in every cycle and it is compared with value of x at which we want y. When x_0 becomes greater than x, program comes out of the loop.

Computer program :

The source code of the program for Euler's method is shown below –

```
/* Download this program from  www.vtubooks.com                  */
/* File name : euler.cpp                                          */

/*------------- EULER'S METHOD TO SOLVE DIFFERENTIAL EQUATION --------*/

/*      THIS PROGRAM CALCULATES THE VALUE y AT GIVEN VALUE OF x
        USING EULER'S METHOD. THE FUNCTION y' = f(x,y) IS
        DEFINED IN THE PROGRAM.

                        y' = x - y
        Hence     f(x,y) = x - y

        INPUTS :  1) Initial values of x and y.

                  2) Step size h.

        OUTPUTS : Calculated values of y at every step.           */

/*----------------------------    PROGRAM  ------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>


 void main()
 {
      double f (double x0,double y0);  /* DECLARATION OF A FUNCTION f  */

      double y0,y1,x0,x1,h,x;
      int i,n,t;

      clrscr();
      printf("\n       EULER'S METHOD TO SOLVE DIFFERENTIAL EQUATION\n");

      printf("\n\nEnter x0 = ");
      scanf("%lf",&x0);                      /* ENTER VALUE OF x0     */
      printf("\n\nEnter y0 = ");
      scanf("%lf",&y0);                      /* ENTER VALUE OF y0     */

      printf("\n\nEnter the value of x at which y is to be found = ");
      scanf("%lf",&x);                       /* ENTER VALUE OF x      */

      printf("\n\nEnter the value of h = ");
      scanf("%lf",&h);                 /*    ENTER THE VALUE OF h     */

      i = 0;
      printf("\nPress any key to see step be step display of results...\n");
      while(x0 < x)         /* LOOP TO CALCULATE y USING EULER'S FORMULA  */
      {
          i++;
          x1 = x0 + h;
          y1 = y0 + h * f(x0,y0);    /* IMPLEMENTATION OF EULER'S FORMULA */
          printf("\nx%d = %lf      y%d = %lf",i,x1,i,y1);
          x0 = x1;   y0 = y1;
          getch();
      }
 }
 /*-----------------------------------------------------------------*/

 double f ( double x,double y)  /* FUNCTION TO CALCULATE VALUE OF f(x,y)  */
 {
     double y_dash;
     y_dash = x - y;          /*    function f(x,y) = y' = x - y        */
     return(y_dash);
 }
 /*---------------------- END OF PROGRAM ---------------------------*/
```

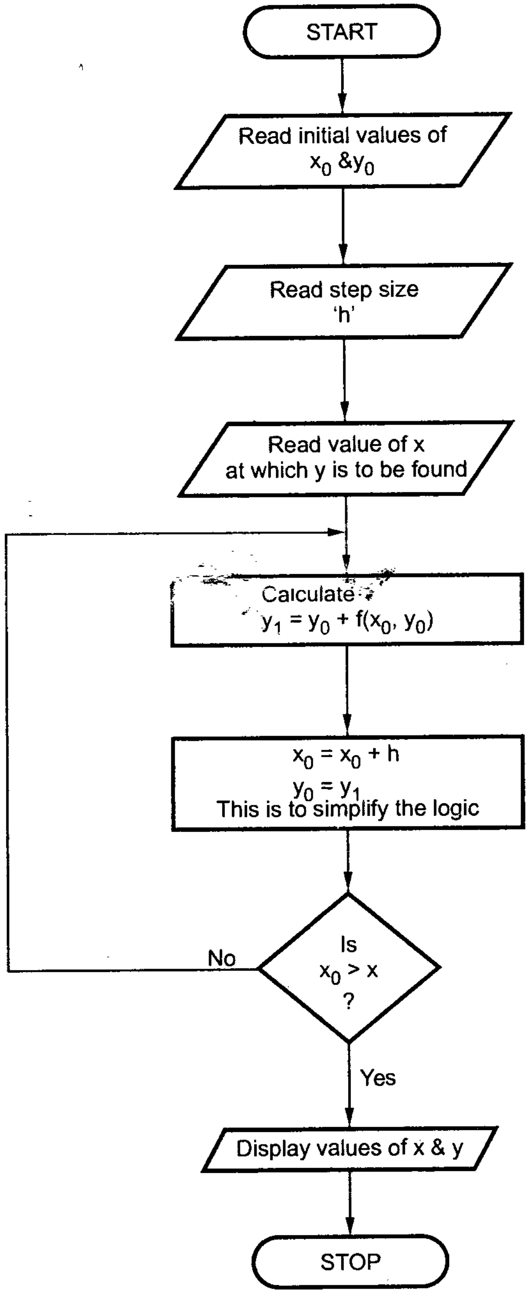


Fig. 8.3.3 Detailed flowchart of Euler's method

Observe that after the include statements, the first statement in function `main` is,

double f (double x0, double y0);

This statement is the declaration of the subroutine function `f`. This subroutine takes two values `x0`, `y0` and returns $f(x_0, y_0)$ depending upon those values. This function is written at the end of the program. The program then gets the values of x0, y0, `x` and `h` in the computer. `x0`, `y0` and `h` carry their usual meanings. `x` is the value at which y is to be calculated.

Then the program enters in the `while` loop.

The statement of `while` loop is,

while (x0<x)

This statement checks whether `x0` has reached to the value of `x`. Here note that `x0` is not x_0.

However `x0` is always updated by adding h to it.

The function `f` is written after the program. Observe the statement in this function,

y_dash = x–y;

This is implementation of

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

Thus this program solves $\frac{dy}{dx} = x - y \quad \because \quad y' = x - y$

If you want to use this program for some other function, then you will have to change the equation of `y_dash` given above.

How to run this program?

Compile the source code given here and make its EXE file. Note that we are using this program to solve

$$\frac{dy}{dx} = x - y \text{ with } x_0 = 0,\ y_0 = 1$$

[This equation is defined in the program]

This equation we have solved in example 8.3.1 let's evaluate y at x = 0.6 with h = 0.2.

Run the program on your computer.

The program then ask for,

`Enter x0` = Here enter `0' and press `enter'

`Enter y0` = Here enter 1 and press `enter'

`Enter the value of x at which y is to be found` = Here enter 0.6 and press `enter'

`Enter value of h` = Here enter 0.2 and press `enter'

Then go on pressing any key to see step by step results. The complete display is shown below.

```
------------------------------- Results ----------------------------

 EULER'S METHOD TO SOLVE DIFFERENTIAL EQUATION


Enter x0 = 0

Enter y0 = 1

Enter the value of x at which y is to be found = 0.6

Enter the value of h = 0.2

Press any key to see step be step display of results...

x1 = 0.200000       y1 = 0.800000
x2 = 0.400000       y2 = 0.680000
x3 = 0.600000       y3 = 0.624000
---------------------------------------------------------------------
```

University Questions

1. Discuss the Euler's predictor-corrector method for the solution of a differential equation. **[Dec - 95, Dec - 96, Dec - 97, May - 98, May - 99]**
2. Apply simple Euler's method to solve

   $$\frac{dy}{dx} = -xy^2; \qquad y(0) = 2$$

   Computing upto x = 1 with h = 0.1. **[Dec - 95, May - 98, Dec - 99, May - 2001]**
3. Use Euler's method to numerically integrate -

   $$f(x, y) = -2x^3 + 12x^2 - 20x + 8.5$$

   from x = 0 to x = 4 with a step size of 0.5. The initial condition at x = 0 is y = 1. **[May - 96, Dec - 98, May - 99, Dec - 2000]**
4. Given :

   $$\frac{dy}{dx} = 10 + y^2 \text{ and } y(0) = 0$$

   Solve numerically for the interval 0 < x < 1 with h = 0.1 and h = 0.2. **[Dec - 96, May - 98, May - 2000]**
5. Derive Euler's method to solve $\frac{dy}{dx} = f(x, y)$ with $y(x_0) = y_0$. **[May - 2001]**
6. Using Eulers method solve

   $$\frac{dy}{dx} = -xy^2; \qquad y(0) = 2$$

   Computing upto x = 1 with h = 0.1. **[May - 2003]**

8.4 Modified Euler's (or Heun's) Method (Euler's Predictor-Corrector Method)

Recall Euler's formula of equation 8.3.5 i.e.,

$$y_{n+1} = y_n + h f(x_n, y_n) \qquad \text{... (8.4.1)}$$

Here $f(x_n, y_n) = \phi'(x_n)$ i.e. slope at (x_n, y_n)

Since step size h is multiplied by slope at (x_n, y_n), the value y_{n+1} contains large errors.

From Euler's formula of equation 8.4.1 above, we get next value of y, i.e. y_{n+1}

and $x_{n+1} = x_n + h$

Thus $f(x_{n+1}, y_{n+1})$ is the slope at (x_{n+1}, y_{n+1}).

Let's take average of the slopes at (x_n, y_n) and (x_{n+1}, y_{n+1}) as,

$$\text{Average slope} = \frac{f(x_n, y_n) + f(x_{n+1}, y_{n+1})}{2}$$

Putting this slope in Euler's formula i.e.,

Next value of y = Old value of y + step size × Average slope

$$\therefore \quad y_{n+1} = y_n + h\left[\frac{f(x_n, y_n) + f(x_{n+1}, y_{n+1})}{2}\right]$$

In the above equation, we need value of y_{n+1} on RHS. This can be calculated by Euler's basic formula. Thus we have two equations for Euler's modified method.

$$\boxed{y_{n+1}^{(k)} = y_n + h f(x_n, y_n)} \qquad \text{... (8.4.2)}$$

This is called *Predictor equation.*

And,

$$y_{n+1}^{(k+1)} = y_n + \frac{h}{2}\left[f(x_n, y_n) + f(x_{n+1}, y_{n+1}^{(k)})\right] \qquad \text{... (8.4.3)}$$

This is called *corrector equation.*

After getting y_{n+1} by predictor equations, corrector equation is applied repeatedly on y_{n+1} till it becomes correct upto required level.

8.4.1 Graphical Interpretation of Modified Euler's Method

Fig. 8.4.1 shows the interpretation of modified Euler's method.

Please refer Fig. 8.4.1 on next page.

Observe in Fig. 8.4.1 (a). There is large error between actual and calculated values of y_{n+1}. Because of averaging of the two slopes in Fig. 8.4.1 (b), the calculated value of y_{n+1} lies almost on the actual value of the curve.

8.4.2 Error in Modified Euler's Method

The recursive equation of modified Euler's method is given by corrector equation of equation 8.4.3 i.e.

$$y_{n+1} = y_n + \frac{h}{2}\left[f(x_n, y_n) + f(x_{n+1}, y_{n+1})\right]$$

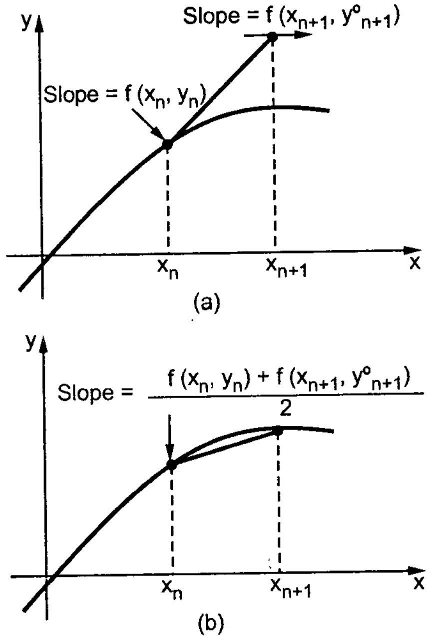


Fig. 8.4.1 Graphical interpretation of modified euler's method

This approximates the basic trapezoidal rule for two points (x_n, y_n) and (x_{n+1}, y_{n+1}). Therefore truncation error is given as,

$$\boxed{{}^{*}E_t = -\frac{f''(z)}{12}h^3} \qquad \dots (8.4.4)$$

Here $\quad x_n \leq z \leq x_{n+1}$

& $\quad h = x_{n+1} - x_n$

8.4.3 Solved Examples

Ex. 8.4.1 *Using modified Euler's method solve the following differential equation,*

$$\frac{dy}{dx} = 1 + xy \qquad \textit{with } y = 1 \quad \textit{when } x = 0$$

Find value of y at x = 0.1 and x = 0.2. Take step size of 0.1. Allowed error is 0.001. **[Dec. - 2003, 8 marks]**

Sol. : Here $\quad f(x, y) = 1 + xy$

with $\quad x_0 = 0, \quad y_0 = 1 \quad \& \quad h = 0.1$

Step 1 : To find y_1 at x_1 (n = 0)

Using predictor formula of equation 8.4.2 we can calculate $y_{n+1}^{(0)}$ as,

$$y_{n+1}^{(0)} = y_n + h\,f(x_n, y_n) \qquad \text{Here k = 0}$$

i.e. $$y_1^{(0)} = y_0 + h\,f(x_0, y_0) \qquad \text{Here n = 0}$$

* Refer [1] for proof of this equation.

$$= y_0 + h(1 + x_0 , y_0)$$
$$= 1 + 0.1\ (1 + 0) = 1.1$$

Use this value of $y_1^{(0)}$ in corrector formula of equation 8.4.3, to calculate more correct value of y_1,

$$y_{n+1}^{(1)} = y_n + \frac{h}{2}\left[f(x_n , y_n) + f(x_{n+1} , y_{n+1}^{(0)})\right] \quad \text{with k = 0}$$

i.e.
$$y_1^{(1)} = y_0 + \frac{h}{2}\left[f(x_0 , y_0) + f(x_1 , y_1^{(0)})\right] \quad \text{with n = 0}$$
$$= y_0 + \frac{h}{2}\left[1 + x_0\, y_0 + 1 + x_1\, y_1^{(0)}\right]$$
$$= 1 + \frac{0.1}{2}\ [1 + 0 + 1 + (0.1 \times 1.1)]$$

Here $x_1 = x_0 + h = 0 + 0.1 = 0.1$

$$= 1.1055$$

Now apply the same corrector formula of equation 8.4.3 with k = 1 and n = 0.

$$y_1^{(2)} = y_0 + \frac{h}{2}\left[f(x_0 , y_0) + f(x_1 , y_1^{(1)})\right]$$
$$= y_0 + \frac{h}{2}\left[1 + x_0\, y_0 + 1 + x_1\, y_1^{(1)}\right]$$
$$= 1 + \frac{0.1}{2}\ [1 + 0 + 1 + (0.1 \times 1.1055)] = 1.1055275$$

Since values of $y_1^{(1)}$ and $y_1^{(2)}$ are repeating upto 4 digits, we will find y for next value of x.

Step 2 : To find y_2 at x_2 (n = 1) :

At the end of step 1 we have $x_1 = 0.1$ and $y_1 = 1.1055275$

$$x_2 = x_1 + h = 0.1 + 0.1 = 0.2$$

Let's apply predictor formula of equation 8.4.2 i.e.,

$$y_{n+1}^{(0)} = y_n + h\, f(x_n , y_n) \quad \text{with k = 0}$$

With n = 2 above equation becomes,

$$y_2^{(0)} = y_1 + h\, f(x_1 , y_1)$$
$$= y_1 + h\,(1 + x_1\, y_1)$$
$$= 1.1055275 + 0.1\ [1 + (0.1) \times (1.1055275)] = 1.2165828$$

Now use this value in corrector formula of equation 8.4.3 to calculate more correct value of y_2,

$$y_{n+1}^{(1)} = y_n + \frac{h}{2}\left[f(x_n , y_n) + f(x_{n+1} , y_{n+1}^{(0)})\right] \quad \text{with k = 0}$$

with n = 1, $$y_2^{(1)} = y_1 + \frac{h}{2}\left[f(x_1, y_1) + f(x_2, y_2^{(0)})\right]$$

$$= y_1 + \frac{h}{2}\left[1 + x_1 y_1 + 1 + x_2 y_2^{(0)}\right]$$

$$= 1.1055275 + \frac{0.1}{2}[1 + (0.1)(1.1055275) + 1 + (0.2)(1.2165828)]$$

$$= 1.223221$$

Now apply the same corrector formula with k = 1 & n = 1 i.e.,

$$y_2^{(2)} = y_1 + \frac{h}{2}\left[f(x_1, y_1) + f(x_2, y_2^{(1)})\right]$$

$$= y_1 + \frac{h}{2}\left[1 + x_1 y_1 + 1 + x_2 y_2^{(1)}\right]$$

$$= 1.1055275 + \frac{0.1}{2}[1 + (0.1)(1.1055275) + 1 + (0.2)(1.223221)]$$

$$= 1.2232873$$

Since value of $y_2^{(2)}$ is repeating upto 4 significant digits with that of $y_2^{(1)}$, we can say that they are correct upto 4 significant digits.

Thus, at x = 0.2, y_2 = 1.2232873 (correct upto 4 decimal places)

Error allowed is 0.001. This means that the answer should be correct upto three decimal places. (i.e. number of zeros after decimal point +1). Here the answer is correct upto 4 decimal places.

Thus the answers are as shown below –

x	y (correct upto 4 decimal places
0.1	1.1055274
0.2	1.2232873

Ex. 8.4.2 *Use modified Euler's method to solve* $\frac{dy}{dx} = x^2 + y$ *with the condition y (0)= 1.*

Find value of y at x = 0.1 **[Dec - 2001, 8 marks]**

Sol. : Here let h = 0.05

$$f(x, y) = x^2 + y \quad \& \quad y(0) = 1 \text{ means } x_0 = 0, y_0 = 1$$

Step 1 : To find y_1 at x_1 :

Here take $x_1 = x_0 + h = 0 + 0.05 = 0.05$

First calculate value of y_1 by predictor formula of equation 8.4.2 i.e.,

$$y_{n+1}^{(0)} = y_n + h f(x_n, y_n) \quad \text{Here k = 0}$$

$$\therefore \quad y_1 = y_0 + h f(x_0, y_0) \quad \text{with n = 0}$$

$$= y_0 + h\,(x_0^2 + y_0)$$
$$= 1 + 0.05\ (\ 0 + 1) = 1.05$$

Now we will apply corrector formula of equation 8.4.3 on y_1 till required accuracy is achieved.

$$y_{n+1}^{(1)} = y_n + \frac{h}{2}\left[f(x_n , y_n) + f\left(x_{n+1} , y_{n+1}^{(0)}\right)\right] \qquad \text{Here k = 0}$$

$$\therefore \quad y_1^{(1)} = y_0 + \frac{h}{2}\left[f(x_0 , y_0) + f\left(x_1 , y_1^{(0)}\right)\right]$$
$$= y_0 + \frac{h}{2}\left[x_0^2 + y_0 + x_1^2 + y_1^{(0)}\right]$$
$$= 1 + \frac{0.05}{2}\left[0 + 1 + (0.05)^2 + 1.05\right] = 1.05131$$

Let's apply the same corrector formula with k = 1 and n = 0 i.e.

$$y_1^{(2)} = y_0 + \frac{h}{2}\left[x_0^2 + y_0 + x_1^2 + y_1^{(2)}\right]$$
$$= 1 + \frac{0.05}{2}\left[0 + 1 + (0.05)^2 + 1.05131\right] = 1.05134$$

Again apply corrector formula on $y_1^{(2)}$ above we get,

$$y_1^{(3)} = y_0 + \frac{h}{2}\left[x_0^2 + y_0 + x_1^2 + y_1^{(2)}\right] \quad \text{with k = 2}$$
$$= 1 + \frac{0.05}{2}\left[0 + 1 + (0.05)^2 + 1.05134\right] = 1.05134$$

Since $y_1^{(2)}$ and $y_1^{(3)}$ are same upto 5 digits after decimal point we can say that value of $y_1^{(3)}$ is correct to 5 decimal digits.

Step 2 : To find y_2 at x_2 :

From previous step we have $y_1^{(2)} = y_1 = 1.05134$ at $x_1 = 0.05$ and $x_2 = x_1 + h =$ 0.05 + 0.05 = 0.1.

First calculate value of y_2 using predictor formula of equation 8.4.2 i.e.

$$y_{n+1}^{(0)} = y_n + h\,f(x_n , y_n) \qquad \text{with k = 0}$$

$$\therefore \quad y_2^{(0)} = y_1 + h\,f(x_1 , y_1) \qquad \text{with n = 1}$$
$$= y_1 + h\,\left(x_1^2 + y_1\right)$$
$$= 1.05134 + 0.05\left[(0.05)^2 + 1.05134\right] = 1.1040$$

Now apply corrector formula of equation 8.4.4 to value of $y_2^{(0)}$ above till required accuracy is achieved. i.e.,

$$y_{n+1}^{(1)} = y_n + \frac{h}{2}\left[f(x_n , y_n) + f(x_{n+1} ,\ y_{n+1}^{(0)})\right] \text{ with k = 0}$$

$$y_2^{(1)} = y_1 + \frac{h}{2}\left[f(x_1, y_1) + f(x_2, y_2^{(0)})\right] \qquad \text{with } n = 1$$

$$= y_1 + \frac{h}{2}\left[x_1^2 + y_1 + x_2^2 + y_2^{(0)}\right]$$

$$= 1.05134 + \frac{0.05}{2}\left[(0.05)^2 + 1.05134 + (0.1)^2 + 1.1040\right]$$

$$= 1.10554$$

Reapply corrector formula with k = 1 we get

$$y_2^{(2)} = y_1 + \frac{h}{2}\left[x_1^2 + y_1 + x_2^2 + y_2^{(1)}\right]$$

$$= 1.05134 + \frac{0.05}{2}\left[(0.05)^2 + 1.05134 + (0.1)^2 + 1.10554\right]$$

$$= 1.10557$$

Reapplying corrector formula with k = 2 we get,

$$y_2^{(3)} = y_1 + \frac{h}{2}\left[x_1^2 + y_1 + x_2^2 + y_2^{(2)}\right]$$

$$= 1.05134 + \frac{0.05}{2}\left[(0.05)^2 + 1.05134 + (0.1)^2 + 1.10557\right]$$

$$= 1.10557$$

Since values of $y_2^{(2)}$ and $y_2^{(3)}$ are similar upto 5 decimal digits, we can say that value of $y_2^{(3)}$ is correct upto 5 decimal places.

Thus at x = 0.1, $y_2^{(3)} = y_2 = y = 1.10557$.

The calculations of this example are tabulated below in the table.

Table of example 8.4.2

n	x	Value of y using predictor formula	k	Value of using corrector formula	Selected value of y
–	$x_0 = 0$	—	—	—	$y_0 = 1$
1	$x_1 = x_0 + h$	$y_1^{(0)} = 1.05$	k = 0	$y_1^{(1)} = 1.05131$	
	$= 0 + 0.05$		k = 1	$y_1^{(2)} = 1.05134$	
	$= 0.05$		k = 2	$y_1^{(3)} = 1.05134$	$y_1 = y_1^{(3)} = 1.05134$
2	$x_2 = x_1 + h$	$y_2^{(0)} = 1.1040$	k = 0	$y_2^{(1)} = 1.10554$	
	$= 0.05 + 0.05$		k = 1	$y_2^{(2)} = 1.10557$	
	$= 0.1$		k = 2	$y_2^{(3)} = 1.10557$	$y_2 = y_2^{(3)} = 1.10557$

Ex. 8.4.3 *Use the predictor-corrector formula for tabulating the solution of* $10\,\dfrac{dy}{dx} = x^2 + y^2$, $y(0) = 1$ *for the range* $0.5 \le x \le 1.0$.

[May - 2000, 8 marks, Dec - 98, 8 marks, May - 97, 8 marks]

Sol. : We will use euler's predictor corrector formula to solve this example.

Here let $h = 0.5$, since we have to find y for $0.5 \le x \le 1.0$. There will be two steps.

$$10\,\frac{dy}{dx} = x^2 + y^2$$

$$\therefore \quad \frac{dy}{dx} = 0.1\,x^2 + 0.1\,y^2$$

i.e.
$$f(x, y) = 0.1\,x^2 + 0.1\,y^2$$

And
$$y(0) = 1 \Rightarrow \quad x_0 = 0, \; y_0 = 1$$

Step 1 : To find y_1 at x_1

Here take $x_1 = x_0 + h = 0 + 0.5 = 0.5$

First calculate the value of y_1 by predictor formula of equation 8.4.2. i.e.,

$$y_{n+1}^{(k)} = y_n + h\; f(x_n, y_n)$$

With $k = 0$ and $n = 0$ above equation will be,

$$y_1^{(0)} = y_0 + h\, f(x_0, y_0)$$

Putting values in above equation,

$$\begin{aligned} y_1^{(0)} &= y_0 + h\left\{0.1\,x_0^2 + 0.1\,y_0^2\right\} \\ &= 1 + 0.5\left\{0.1\,(0)^2 + 0.1\,(1)^2\right\} \\ &= 1.05 \end{aligned}$$

New let us apply the corrector formula of equation 8.4.3. i.e.,

$$y_{n+1}^{(k+1)} = y_n + \frac{h}{2}\left[f(x_n, y_n) + f\left(x_{n+1}, y_{n+1}^{(k)}\right)\right]$$

With $n = 0$ and $k = 0$ in above equation,

$$\begin{aligned} y_1^{(1)} &= y_0 + \frac{h}{2}\left[f(x_0, y_0) + f\left(x_1, y_1^{(0)}\right)\right] \\ &= y_0 + \frac{h}{2}\left\{0.1\,x_0^2 + 0.1\,y_0^2 + 0.1\,x_1^2 + 0.1\left[y_1^{(0)}\right]^2\right\} \\ &= 1 + \frac{0.5}{2}\left\{0.1(0)^2 + 0.1(1)^2 + 0.1(0.5)^2 + 0.1(1.05)^2\right\} \\ &= 1.0588125 \end{aligned}$$

Now apply the corrector formula again with $n = 0$ and $k = 1$. i.e.,

$$y_1^{(2)} = y_0 + \frac{h}{2}\left[f(x_0, y_0) + f\left(x_1, y_1^{(1)}\right)\right]$$

$$= y_0 + \frac{h}{2}\left\{0.1\,x_0^2 + 0.1\,y_0^2 + 0.1\,x_1^2 + 0.1\left[y_1^{(1)}\right]^2\right\}$$

$$= 1 + \frac{0.5}{2}\left\{0.1(0)^2 + 0.1(1)^2 + 0.1(0.5)^2 + 0.1(1.0588125)^2\right\}$$

$$= 1.0592771$$

Step 2 : To find y_2 at x_2

Here take $x_2 = x_1 + h = 0.5 + 0.5 = 1$

First calculate the value of y_1 by predictor formula of equation 8.4.2. i.e.,

$$y_{n+1}^{(k)} = y_n + h\, f\,(x_n, y_n)$$

With $n = 1$ and $k = 0$ in above equation,

$$y_2^{(0)} = y_1 + h\, f\,(x_1, y_1)$$

$$= y_1 + h\left\{0.1\ x_1^2 + 0.1\ y_1^2\right\}$$

Here $y_1 = y_1^{(2)} = 1.0592771$ as obtained in last step.

$$\therefore\quad y_2^{(0)} = 1.0592771 + 0.5\left\{0.1(0.5)^2 + 0.1\ (1.0592771)^2\right\}$$

$$= 1.1278805$$

Now apply corrector formula of equation 8.4.3. i.e.,

$$y_{n+1}^{(k+1)} = y_n + \frac{h}{2}\left[f(x_n, y_n) + f\left(x_{n+1}, y_{n+1}^{(k)}\right)\right]$$

With $k = 0$ and $n = 1$ above equation becomes,

$$y_2^{(1)} = y_1 + \frac{h}{2}\left\{f(x_1, y_1) + f\left(x_2, y_2^{(0)}\right)\right\}$$

$$= y_1 + \frac{h}{2}\left\{0.1\ x_1^2 + 0.1\ y_1^2 + 0.1\ x_2^2 + 0.1\left[y_2^{(0)}\right]^2\right\}$$

Here $y_1 = y_1^{(2)} = 1.0592771$ obtained in last stop.

$$\therefore\quad y_2^{(1)} = 1.0592771 + \frac{0.5}{2}\left\{0.1(0.5)^2 + 0.1(1.0592771)^2 + 0.1(1)^2 + 0.1(1.1278805)^2\right\}$$

$$= 1.1503817$$

Apply the corrector formula again with $n = 1$ and $k = 1$. i.e.,

$$y_2^{(2)} = y_1 + \frac{h}{2}\left[f(x_1, y_1) + f\left(x_2, y_2^{(1)}\right)\right]$$

$$= y_1 + \frac{h}{2}\left\{0.1\,x_1^2 + 0.1\ y_1^2 + 0.1\ x_2^2 + 0.1\left[y_1^{(1)}\right]^2\right\}$$

$$= 1.0592771 + \frac{0.5}{2}\left\{0.1(0.5)^2 + 0.1\ (1.0592771)^2 + 0.1(1)^2 + 0.1(1.1503817)^2\right\} = 1.1516632$$

The results are tabulated below :

n	x	Value of predictor formula	k	Value of corrector formula	Selected value of y
–	$x_0 = 0$	–	–	–	$y_0 = 1$
1	$x_1 = x_0 + h$ $= 0 + 0.5$ $= 0.5$	$y_1^{(0)} = 1.05$	$k = 0$	$y_1^{(1)} = 1.0588125$	
			$k = 1$	$y_1^{(2)} = 1.0592771$	$y_1 = 1.0592771$
2	$x_2 = x_1 + h$ $= 0.5 + 0.5$ $= 1$	$y_2^{(0)} = 1.1278805$	$k = 0$	$y_2^{(1)} = 1.1503817$	
				$y_2^{(2)} = 1.1516632$	$y_2 = 1.1516632$

The values of $y_1^{(1)}$ and $y_1^{(2)}$ repeat upto two digits after decimal point. Hence $y_1 = 1.0592771$ is correct upto two digits after decimal point. Similarly $y_2 = 1.1516632$ is also correct upto two digits.

Exercise

1. *Using Euler's modified method solve the following differential equation,*

$$\frac{dy}{dx} = \log(x+y) \text{ at } x = 1.2 \text{ and } x = 1.4.$$

Given that y (1) = 2. Take h = 0.2. Your answer should be correct upto three decimal places.

[Hint : y (1) = 2 means $x_0 = 1$ and $y_0 = 2$]

[Ans. : At x = 1.2, y = 2.2332 and at x = 1.4, y = 2.492]

2. *It is given that*

$$\frac{dy}{dx} = \frac{y-x}{y+x} \quad \text{with } y(0) = 1$$

Find value of y for x = 0.1 using

(i) Euler's method (take h = 0.02) and

(ii) Modified Euler's method (take h = 0.05)

[Hint : If value of 'h' is not given for modified Euler's method, then always select 'h' such that you will need at the most two or three iterations to reach to required value of x]

[Ans. : y (0.1) = 1.0928 by Euler's method & y (0.1) = 1.0912 by modified Euler's method]

3. *Given that*

$$\frac{dy}{dx} - \sqrt{xy} = 2, \qquad y(1) = 1$$

Find the value of y (2) in steps of 0.1 using Euler's modified method.

[May - 2001, 8 marks, Dec - 99, 8 marks, Dec - 98, 8 marks]

*[**Hint** : Here steps of 0.1 means take h = 0.1 and apply corrector formula only once to reduce calculations.]* **[Ans. : y (2) = 5.0524]**

8.4.4 Algorithm and C Program

Based on the illustration for modified Euler's method we will now prepare an algorithm for it.

Assumption : The function $\frac{dy}{dx} = f(x, y)$ is predefined in the program itself. The following algorithm does not have the provision to accept different functions.

Algorithm :

Step 1 : Read initial values of x_0 **and** y_0.

step 2 : Read the value of x at which y is to be calculated.

Step 3 : Read the step size h.

Step 4 : Calculate $y_{n+1}^{(0)}$ ***using predictor formula i.e.***

$$y_{n+1}^{(0)} = y_n + h\, f(x_n, y_n)$$

Step 5 : Calculate $y_{n+1}^{(k+1)}$ ***using corrector formula i.e.***

$$y_{n+1}^{(k+1)} = y_n + \frac{h}{2}\left[f(x_n, y_n) + f(x_{n+1}, y_{n+1}^{(k)})\right]$$

Step 6 : Repeat step 5 until y_{n+1} ***is reached to the required accuracy.***

Step 7 : Increase x_n ***by one step.***

Step 8 : Repeat steps 4 to step 7 till x_n ***becomes greater than x.***

Step 9 : Display x_n, y_n ***on the screen in tabular format.***

Step 10 : Stop.

Flowchart :

Based on the algorithm given above we will now prepare a flowchart for it. This flowchart is shown in Fig. 8.4.2. Observe that each and every step is shown in details in the flowchart. The program discussed next is based on this logic only.

Please refer Fig. 8.4.2 on next page.

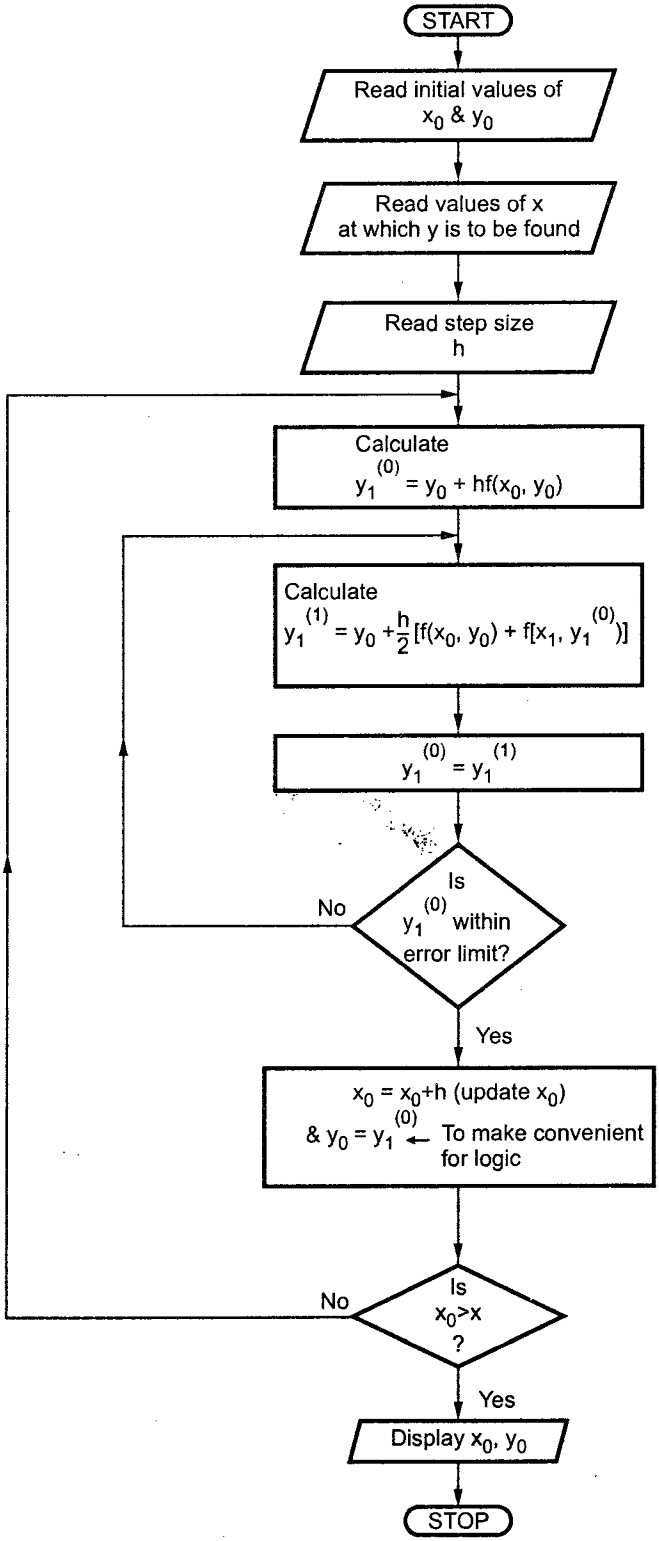


Fig. 8.4.2 Detailed flowchart of modified Euler's method

Computer Program :

The source code of the C program for modified Euler's method is shown below.

```
/* Download this program from  www.vtubooks.com                    */
/* File name : mod_eulr.cpp                                         */

/*------- MODIFIED EULER'S METHOD TO SOLVE DIFFERENTIAL EQUATION ------*/

/*      THIS PROGRAM CALCULATES THE VALUE y AT GIVEN VALUE OF x
        USING MODIFIED EULER'S METHOD. THE FUNCTION y' = f(x,y) IS
        DEFINED IN THE PROGRAM.

                        y' = x*x + y
        Hence     f(x,y) = x*x + y

        INPUTS :  1) Initial values of x and y.

                  2) Step size h.

        OUTPUTS : Calculated values of y at every step.              */

/*------------------------------    PROGRAM  ---------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>

void main()
{
      double f (double x0,double y0);  /* DECLARATION OF A FUNCTION f    */

      double y0,y1,y10,x0,x1,h,x,diff;
      int i,n;

      clrscr();

      printf("\n  MODIFIED EULER'S METHOD TO SOLVE DIFFERENTIAL EQUATION");

      printf("\n\nEnter x0 = ");
      scanf("%lf",&x0);                     /* ENTER VALUE OF x0      */
      printf("\n\nEnter y0 = ");
      scanf("%lf",&y0);                     /* ENTER VALUE OF y0      */

      printf("\n\nEnter the value of x at which y is to be found = ");
      scanf("%lf",&x);                      /* ENTER VALUE OF x       */

      printf("\n\nEnter the value of h = ");
      scanf("%lf",&h);                /*    ENTER THE VALUE OF h      */

      i = 0;
      printf("\nPress any key to see step by step display of results...\n");
      while(x0 < x) /* LOOP TO CALCULATE y USING MODIFIED EULER'S FORMULA*/
      {
           i++;
           x1 = x0 + h;
           y1 = y0 + h * f(x0,y0);        /* PREDICTOR EQUATION */
           y10 = y1;

           do {          /* LOOP TO IMPLEMENT CORRECTOR FORMULA */

                 y1 = y0 + (h/2) * ( f(x0,y0) + f(x1,y10) );
                 diff = fabs(y1 - y10);  /* CHECK FOR HOW MANY DIGITS
                                            ARE REPEATING AFTER DECIMAL POINT */
                 y10 = y1;
              } while( diff > 0.0001);
                /*REMAIN IN LOOP TILL 3 DIGITS REPEAT AFTER DECIMAL POINT */
```

```
        printf("\nx%d = %lf      y%d = %lf",i,x1,i,y1);
        x0 = x1;    y0 = y1;
        getch();
    }

/*-----------------------------------------------------------------*/

double f ( double x,double y)  /* FUNCTION TO CALCULATE VALUE OF f(x,y)  */
{
    double y_dash;
    y_dash = x*x + y;          /*   function f(x,y) = y' = x*x + y    */
    return(y_dash);
}
/*----------------------- END OF PROGRAM ----------------------------*/
```

Observe that this program is exactly similar to the program of Euler's method till `while` loop. The values of x_0 , y_0 , x & h are taken in the computer as usual. Then the program enters `while` loop. The statement of the `while` loop is, `while(x0<x)`

This means execute `while` loop till `x0` is less than `x`. Value of `x0` is incremented by step size 'h' in every cycle. Note here that `x0` is not our usual x_0, rather it is variable incremented every time to pass to function `f`.

In the `while` loop, the third statement is

$$y_1 = y0 + h * f(x0, y0);$$

This statement is the 'predictor equation' in modified Euler's method. This is an implementation of equation 8.4.2. This value is actually $y_1^{(0)}$.

∴ The next statement,

y10 = y1;

This is equivalent to $y_1^{(0)} = y_1$. This is done to make it convenient to refer in the program.

The program then enters `do-while` loop. Note that this loop is within outer `while` loop. This loop is reproduced below for explanation.

```
do{
```

```
  y1=y0+(h/2)*(f(x0,y0)+f(x1,y10));
```

This statement is corrector formula of equation 8.4.3. Here note that $y1$ = $y_n^{(1)}$, $y_n^{(2)}$, $y_n^{(3)}$... in successive cycles. and **n** depends on outer loop.

`diff=fabs(y1-y10);` ← This statement calculates absolute difference between $y_n^{(k+1)}$ & $y_n^{(k)}$ to see how many digits are repeating after decimal point.

`y10=y1;` ← This statement makes it convenient to call function `f` with the same name of variables.

`} while(diff>0.0001);` ← This is an expression of `do-while` loop. `It checks whether the difference between` $y_n^{(k+1)}$ & $y_n^{(k)}$ is reduced bellow 0.0001.This happens when three digits repeat in them.

The program then comes out of this loop and prints the value of x_n , y_n on screen.

After the program, the function `f` is written.

The statement in this function `f`,

y_dash = x*x+y;

is an implementation of

$$\frac{dy}{dx} = x^2 + y \quad \text{or} \quad y' = f(x,y) = x^2 + y$$

If you want to use this program for some other differential equation, then you will have to change the expression for `y_dash` given above. Modify this expression to implement new differential equation.

How to Run this program?

Compile and make EXE file of this source code. As just discussed, this program is written to solve,

$$\frac{dy}{dx} = x^2 + y \quad \text{with} \quad y(0) = 1$$

And we have to find y at x = 0.1. This equation is solved in example 8.4.2 with $h = 0.05$.

Let's use the same data here,

After running the program, it will display

`Enter x0` = Here enter '0' and press 'enter'

`Enter y0` = Here enter '1' and press 'enter'

`Enter value of x at which y is to be found` = Here enter `0.1' and press `enter'

`Enter value of h` = Here enter `0.05' and press `enter'

The program then displays the results step by step. You have to just press any key. The results are shown below.

```
------------------------------ Results ---------------------------

MODIFIED EULER'S METHOD TO SOLVE DIFFERENTIAL EQUATION

Enter x0 = 0

Enter y0 = 1

Enter the value of x at which y is to be found = 0.1

Enter the value of h = 0.05

Press any key to see step by step display of results...

x1 = 0.050000        y1 = 1.051345
x2 = 0.100000        y2 = 1.105580
-----------------------------------------------------------------
```

University Questions

1. Given that -

   $$\frac{dy}{dx} - \sqrt{xy} = 2, \quad y(1) = 1,$$

   find the value of y(2) in steps of 0.1 using Euler's modified method.

   [May - 97, Dec - 98, Dec - 99, May - 2001]

2. Find y(0.2) using Modified Euler's Method, given **[May - 2000]**

   $\frac{dy}{dx} = x + y,\ y(0) = 1$ step size 0.1

3. Given $\frac{dy}{dx} = x^2 + y,\ y(0) = 1$ determine y(0.02), y(0.04) and y(0.06) using Euler's modified method. **[Dec - 2001]**

4. Explain modified Euler's method and write a program to do the same **[Dec - 2001]**

5. Uisng Euler's modified method solve $\frac{dy}{dx}$ = 1 + xy for y(0.1) and y(0.2) given initial condition y(0) = 2.0. **[Dec. - 2003]**

6. Write a program to solve a differential equation using Eulers modified method.

   [Dec. - 2004]

8.5 Runge Kutta Methods (RK Methods)

Runge Kutta (RK) Methods have higher accuracy with reduced calculation efforts. Various types of Runge Kutta methods are discussed next. The most popular is 4^{th} order Runge Kutta method.

8.5.1 Second Order Runge Kutta Method

The second order equation for solution of a differential equation is given as,

$$\boxed{\begin{aligned} y_{n+1} &= y_n + \frac{h}{2}(k_1 + k_2) \\ \text{where} \quad k_1 &= f(x_n, y_n) \\ \& \quad k_2 &= f(x_n + h,\ y_n + hk_1) \end{aligned}} \quad \ldots (8.5.1)$$

8.5.2 Third Order Runge Kutta Method

In the third order Runge Kutta method, three factors are calculated to find predicted value of y_{n+1}. The percentage error is further reduced in third order Runge Kutta method compared to second order. The set of equations for this method is given below –

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 4k_2 + k_3)$$

where,

$$k_1 = f(x_n, y_n)$$

$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{hk_1}{2}\right)$$

$$k_3 = f(x_n + h, y_n - hk_1 + 2hk_2) \qquad \text{... (8.5.2)}$$

8.5.3 Fourth Order Runge Kutta Method

This is the most popular among RK methods. The percentage relative error is less than other methods and at the same time, the burden of calculations is reduced. The set of equations for this method are given below.

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

where,

$$k_1 = f(x_n, y_n)$$

$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{hk_1}{2}\right)$$

$$k_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{hk_2}{2}\right)$$

$$k_4 = f(x_n + h, y_n + hk_3) \qquad \text{... (8.5.3)}$$

8.5.4 Solved Examples

Ex. 8.5.1 *Using 4^{th} order Runge Kutta method, integrate,*

$$\frac{dy}{dx} = f(x, y) = -2x^3 + 12x^2 - 20x + 8.5$$

using a step size of 0.5 and an initial condition of $y = 1$ at $x = 0$.

Use Euler's method to integrate above equation and compare the result.

[Dec - 98, 8 marks, May - 96, 10 marks, May - 2004, 10 marks]

Sol. : Here $f(x, y) = \frac{dy}{dx}$ and we have to find value of y in steps of 0.5. i.e. h = 0.5. In these steps we can find values of y as many as we want.

We have $x_0 = 0, y_0 = 1, h = 0.5$. Let's find y at x = 1 in two steps.

Step 1 : h = 0.5, $x_1 = x_0 + h$

$= 0 + 0.5 = 0.5$

We have to find y_1.

With n = 0 in equation 8.5.3 we get,

$$y_1 = y_0 + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad \text{... (8.5.4)}$$

&

$$k_1 = f(x_0, y_0) \quad \text{... (8.5.5)}$$

$$k_2 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{hk_1}{2}\right) \quad \text{... (8.5.6)}$$

$$k_3 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{hk_2}{2}\right) \quad \text{... (8.5.7)}$$

$$k_4 = f(x_0 + h, y_0 + hk_3) \quad \text{... (8.5.8)}$$

From equation 8.5.5 we have,

$$\begin{aligned} k_1 &= f(x_0, y_0) \\ &= f(0, 1) \\ &= 0 + 0 - 0 + 8.5 \end{aligned}$$

$$\therefore \boxed{k_1 = 8.5}$$

From equation 8.5.6 we have,

$$\begin{aligned} k_2 &= f\left(x_0 + \frac{h}{2}, y_0 + \frac{hk_1}{2}\right) \\ &= f\left(0 + \frac{0.5}{2}, 1 + \frac{0.5 \times 8.5}{2}\right) \\ &= f(0.25, 3.125) \\ &= -2(0.25)^3 + 12(0.25)^2 - 20(0.25) + 8.5 \\ &= 4.21875 \end{aligned}$$

$$\therefore \boxed{k_2 = 4.21875}$$

From equation 8.5.7 we have,

$$\begin{aligned} k_3 &= f\left(x_0 + \frac{h}{2}, y_0 + \frac{hk_2}{2}\right) \\ &= f\left(0 + \frac{0.5}{2}, 1 + \frac{0.5 \times 4.21875}{2}\right) \\ &= f(0.25, 2.05468) \\ &= -2(0.25)^3 + 12(0.25)^2 - 20(0.25) + 8.5 \end{aligned}$$

$$= 4.21875 \qquad \therefore \boxed{k_3 = 4.21875}$$

From equation 8.5.8 we get,

$$k_4 = f\left(x_0 + h,\ y_0 + hk_3\right)$$
$$= f\ (0 + 0.5,\ 1 + 0.5 \times 4.21875)$$
$$= f\ (0.5,\ 3.109375)$$
$$= -2\,(0.5)^3 + 12\,(0.5)^2 - 20\,(0.5) + 8.5$$
$$= 1.25 \qquad \therefore \boxed{k_4 = 1.25}$$

Putting the values of k_1, k_2, k_3 and k_4 in equation 8.5.4 we have,

$$y_1 = y_0 + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$
$$= 1 + \frac{0.5}{6}\ [8.5 + 2\,(4.21875) + 2\,(4.21875) + 1.25]$$
$$= 3.21875$$

Step 2 :

For this step we have,

$$x_1 = 0.5, \quad y_1 = 3.21875 \qquad \text{(From previous step)}$$
$$h = 0.5 \quad \& \quad x_2 = x_1 + h$$
$$= 0.5 + 0.5$$
$$= 1$$

With n = 1 in equation 8.5.3 we get,

$$y_2 = y_1 + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \qquad \text{... (8.5.9)}$$

where

$$k_1 = f\ (x_1, y_1) \qquad \text{... (8.5.10)}$$
$$k_2 = f\left(x_1 + \frac{h}{2},\ y_1 + \frac{hk_1}{2}\right) \qquad \text{... (8.5.11)}$$
$$k_3 = f\left(x_1 + \frac{h}{2},\ y_1 + \frac{hk_2}{2}\right) \qquad \text{... (8.5.12)}$$
$$k_4 = f\left(x_1 + h,\ y_1 + hk_3\right) \qquad \text{... (8.5.13)}$$

From equation 8.5.10, we have,

$$k_1 = f\ (x_1, y_1)$$
$$= f\ (0.5,\ 3.21875)$$
$$= -2\,(0.5)^3 + 12\,(0.5)^2 - 20\,(0.5) + 8.5$$
$$= 1.25 \qquad \therefore \boxed{k_1 = 1.25}$$

From equation 8.5.11 we have,

$$k_2 = f\left(x_1 + \frac{h}{2},\ y_1 + \frac{hk_1}{2}\right)$$

$$= f\left(0.5 + \frac{0.5}{2},\ 3.21875 + \frac{0.5 \times 1.25}{2}\right)$$

$$= f(0.75,\ 3.53125)$$

$$= -2(0.75)^3 + 12(0.75)^2 - 20(0.75) + 8.5$$

$$= -0.59375 \qquad \therefore \boxed{k_2 = -0.59375}$$

From equation 8.5.12 we have,

$$k_3 = f\left(x_1 + \frac{h}{2},\ y_1 + \frac{hk_2}{2}\right)$$

$$= f\left(0.5 + \frac{0.5}{2},\ 3.21875 + \frac{0.5\,(-0.59375)}{2}\right)$$

$$= f(0.75,\ 3.07031)$$

$$= -2(0.75)^3 + 12(0.75)^2 - 20(0.75) + 8.5$$

$$= -0.59375 \qquad \therefore \boxed{k_3 = -0.59375}$$

$$x_2 = x_1 + h = 0.1 + 0.1 = 0.2$$

From equation 8.5.13 we have,

$$k_4 = f\left(x_1 + h,\ y_1 + hk_3\right)$$

$$= -2(1)^3 + 12(1)^2 - 20(1) + 8.5$$

$$= f\left(0.5 + 0.5,\ 3.21875 + 0.5\,(-0.59375)\right)$$

$$= f(1,\ 2.921875)$$

$$= -2(1)^3 + 12(1)^2 - 20(1) + 8.5$$

$$= -1.5 \qquad \boxed{\therefore\ k_4 = -1.5}$$

Putting values of k_1, k_2, k_3 and k_4 in equation 8.5.9 we have,

$$y_2 = y_1 + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

$$= 3.21875 + \frac{0.5}{6}\left(1.25 + 2(-0.59375) + 2(-0.59375) - 1.5\right)$$

$$= 3$$

Thus we have, the results are,

x	y
0	1
0.5	3.21875
1.0	3.0

Solution using Euler's Method

Here $f(x,y) = -2x^3 + 12x^2 - 20x + 8.5$

$x_0 = 0$, $y_0 = 1$ and $h = 0.5$

Euler's formula gives the value of y as,

$$y_{n+1} = y_n + hf(x_n, y_n)$$

$$\therefore \quad y_1 = y_0 + hf(x_0, y_0)$$

$$= y_0 + h\left[-2x_0^3 + 12x_0^2 - 20x_0 + 8.5\right]$$

$$= 1 + 0.5\left[-2(0)^3 + 12(0)^2 - 20(0) + 8.5\right]$$

$$= 5.25$$

Now $x_1 = x_0 + h = 0 + 0.5 = 0.5$

$$\therefore \quad y_2 = y_1 + hf(x_1, y_1)$$

$$= y_1 + h\left[-2x_1^3 + 12x_1^2 - 20x_1 + 8.5\right]$$

$$= 5.25 + 0.5\left[-2(0.5)^3 + 12(0.5)^2 - 20(0.5) + 8.5\right]$$

$$= 5.875$$

Thus the results are,

x	Euler's method	Runge kutta method
0	1	1
0.5	5.25	3.21875
1	5.875	3

Comment on results

Observe that there is large difference in the results obtained by Runge-Kutta method and Euler's method. This difference can be reduced by taking small step size. If we take h = 0.1, then results of Euler's method are as listed below :

x	y
0	1
0.1	1.85
0.2	2.5118
0.3	3.0082
0.4	3.3608
0.5	3.59
0.6	3.715
0.7	3.7538
0.8	3.7232
0.9	3.6388
1.0	3.515

Ex.8.5.2 *Solve* $10\dfrac{dy}{dx} = x^2 + y^2$ *at x = 0.4 using fourth order Runge Kutta method. Initial conditions are y = 1 at x = 0. Take h = 0.2*

Use Runge Kutta method to solve

$$10\ \frac{dy}{dx} = x^2 + y^2, \quad y\ (0) = 1$$

For the interval $0 \le x \le 0.4$ *with h = 0.1*

Sol. : Both the above statements are similar except h is different. Let's solve this problem with h = 0.1.

Thus we have,

$x_0 = 0,$ $\quad y_0 = 1,$ $\quad$ h = 0.1 & we have to find y for x = 0.1 to 0.4.

Rearrange the equation as,

$$\frac{dy}{dx} = \frac{x^2 + y^2}{10}$$

i.e.
$$f(x, y) = \frac{x^2 + y^2}{10}$$

We have 4^{th} order Runge Kutta equations from equation 8.5.3 as,

$$\left.\begin{aligned} y_{n+1} &= y_n + \frac{h}{6}\left(k_1 + 2. \quad + 2k_3 + k_4\right) \\ \text{where} \qquad k_1 &= f\ (x_n\ , y_n) \\ k_2 &= f\left(x_n + \frac{h}{2},\ y_n + \frac{hk_1}{2}\right) \\ k_3 &= f\left(x_n + \frac{h}{2},\ y_n + \frac{hk_2}{2}\right) \\ k_4 &= f\ \left(x_n + h,\ y_n + hk_3\right) \end{aligned}\right\} \qquad \text{... (8.5.14)}$$

Step 1 : To find y_1 for $x_1 = x_0 + h = 0 + 0.1$. i.e. $x_1 = 0.1$

Let n = 0 in above equations,

$$y_1 = y_0 + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \qquad \text{... (8.5.15)}$$

where

$$k_1 = f\ (x_0\ , y_0) \qquad \text{... (8.5.16)}$$

$$k_2 = f\left(x_0 + \frac{h}{2},\ y_0 + \frac{hk_1}{2}\right) \qquad \text{... (8.5.17)}$$

$$k_3 = f\left(x_0 + \frac{h}{2},\ y_0 + \frac{hk_2}{2}\right) \qquad \text{... (8.5.18)}$$

$$k_4 = f\ \left(x_0 + h\ ,\ y_0 + hk_3\right) \qquad \text{... (8.5.19)}$$

From equation 8.5.16 we have,

$$\begin{aligned} k_1 &= f\ (x_0\ ,\ y_0) \\ &= \frac{x_0^2 + y_0^2}{10} \qquad \text{By putting value of } f\,(x, y) \\ &= \frac{0+1}{10} \\ &= 0.1 \end{aligned}$$

From equation 8.5.17 we have,

$$\begin{aligned} k_2 &= f\left(x_0 + \frac{h}{2},\ y_0 + \frac{hk_1}{2}\right) \\ &= f\left(0 + \frac{0.1}{2},\ 1 + \frac{0.1 \times 0.1}{2}\right) \\ &= f\,(0.05\ ,\ 1.005) \\ &= \frac{(0.05)^2\ +\ (1.005)^2}{10} \\ &= 1.1012525 \end{aligned}$$

From equation 8.5.18 we have,

$$k_3 = f\left(x_0 + \frac{h}{2},\ y_0 + \frac{hk_2}{2}\right)$$

$$= f\left(0 + \frac{0.1}{2},\ 1 + \frac{0.1 \times 0.1012525}{2}\right)$$

$$= f(0.05\ ,\ 1.0050626)$$

$$= \frac{(0.05)^2 + (1.0050626)^2}{10}$$

$$= 0.101265$$

From equation 8.5.19 we have,

$$k_4 = f\left(x_0 + h,\ y_0 + hk_3\right)$$

$$= f(0 + 0.1\ ,\ 1 + 0.1 \times 0.101265)$$

$$= f(0.1\ ,\ 1.0101265)$$

$$= \frac{(0.1)^2 + (1.0101265)^2}{10}$$

$$= 0.1030355$$

From equation 8.5.15 we have,

$$y_1 = y_0 + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

$$= 1 + \frac{0.1}{6}\ [0.1 + 2\ (0.1012626) + 2\ (0.101265) + 0.1030355]$$

$$= 1.010345$$

Step 2 : We have $x_2 = x_1 + h = 0.1 + 0.1$ i.e. $x_1 = 0.2$

$\therefore \quad x_1 = 0.2, \qquad y_1 = 1.010345$

with n = 1 in equation 8.5.14 we get,

$$k_1 = f(x_1\ ,\ y_1)$$

$$= f(0.2\ ,\ 1.010345)$$

$$= 0.103037$$

$$k_2 = f\left(x_1 + \frac{h}{2},\ y_1 + \frac{hk_1}{2}\right)$$

$$= 0.1053306$$

$$k_3 = f\left(x_1 + \frac{h}{2},\ y_1 + \frac{hk_2}{2}\right)$$

$$= 0.1053539$$

$$k_4 = f\left(x_1 + h\ ,\ y_1 + hk_3\right)$$

$$= 0.1081767$$

$$\therefore \quad y_2 = y_1 + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$
$$= 1.0206775$$

Step 3 : We have $x_2 = x_1 + h = 0.1 + 0.1$
$$= 0.2$$

$\therefore \quad x_2 = 0.2$ and $y_2 = 1.0206775$

Apply the same formulae of equation 8.5.14 and calculate y_3.

The results of all the steps are shown below.

Step	x	y
1	0.1	1.010135
2	0.2	1.020678
3	0.3	1.031842
4	0.4	1.043845

Ex.8.5.3 *Using second order Runge Kutta method, integrate,*

$$f(x, y) = -2x^3 + 12x^2 - 20x + 8.5$$

using step size of 0.5 and initial conditions of $y = 1$ at $x = 0$. Find value of y at $x = 0.5$ (i.e. only one step)

Sol. : The function is,

$$f(x, y) = -2x^3 + 12x^2 - 20x + 8.5$$
$$x_0 = 0, \quad y_0 = 1, \quad h = 0.5$$

From equation 8.5.1 we have second order equations as,

$$\left.\begin{aligned} y_{n+1} &= y_n + \frac{h}{2}\left(k_1 + k_2\right) \\ \& \quad k_1 &= f\left(x_n, y_n\right) \\ k_2 &= f\left(x_n + h, y_n + hk_1\right) \end{aligned}\right\} \quad \ldots (8.5.20)$$

with n = 0 in above equations we get,

To find y_1 : We have $x_1 = x_0 + h = 0 + 0.5 = 0.5$

$$y_1 = y_0 + \frac{h}{2}\left(k_1 + k_2\right)$$

$$\& \quad k_1 = f\left(x_0, y_o\right)$$
$$k_2 = f\left(x_0 + h, y_0 + hk_1\right)$$
$$\therefore \quad k_1 = f\left(x_0, y_0\right)$$
$$= f(0, 1)$$

$$= -0+0-0+8.5 = 8.5 \qquad \text{By putting value of } f(x,y)$$

& $$k_2 = f\left(x_0 + h,\ y_0 + hk_1\right)$$
$$= f(0+0.5,\ 1+0.5 \times 8.5)$$
$$= f\ (0.5,\ 5.25)$$
$$= -2\ (0.5)^3 + 12\ (0.5)^2 - 20\ (0.5) + 8.5$$

By putting value of $f(x,y)$

$$= 1.25$$

$\therefore$ We can calculate value of y_1 as,

$$y_1 = y_0 + \frac{h}{2}\left(k_1 + k_2\right)$$
$$= 1 + \frac{0.5}{2}\ (8.5 + 1.25)$$
$$= 3.4375$$

Thus at $x_1 = 0.5$, $y_1 = 3.4375$.

Ex. 8.5.4 *Solve* $\frac{dy}{dx} = x + y$ *when* $y = 1$ *at* $x = 0$. *Find solution for* $x = 0.1,\ 0.2$ *by Runge-Kutta method.*

Sol. : Here $\frac{dy}{dx} = x + y \qquad \therefore \qquad f(x,y) = x + y$

$x_0 = 0$ and $y_0 = 1$

Step 1 :

Let $h = 0.1$

Hence
$$x_1 = x_0 + h$$
$$= 0 + 0.1$$
$$= 0.1$$

We have to find y_1

With $n = 0$ in equation 8.5.3 we get,

$$y_1 = y_0 + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \qquad \ldots (8.5.21)$$

$$k_1 = f\ (x_0, y_0) \qquad \ldots (8.5.22)$$

$$k_2 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{hk_1}{2}\right) \qquad \ldots (8.5.23)$$

$$k_3 = f\left(x_0 + \frac{h}{2},\ y_0 + \frac{hk_2}{2}\right) \qquad \ldots (8.5.24)$$

$$k_4 = f\left(x_0 + h,\ y_0 + h\,k_3\right) \qquad \ldots (8.5.25)$$

From 8.5.22 we have,

$$k_1 = f(x_0, y_0)$$
$$= f(0, 1)$$
$$= 0 + 1 \quad \text{since} \quad f(x,y) = \frac{dy}{dx} = x + y$$
$$= 1$$

$$\boxed{\therefore \quad k_1 = 1}$$

From equation 8.5.23, $k_2 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{hk_1}{2}\right)$

$$= f\left(0 + \frac{0.1}{2},\ 1 + \frac{0.1 \times 1}{2}\right)$$
$$= f(0.05,\ 1.05)$$
$$= 0.05 + 1.05$$
$$= 1.1$$

$$\boxed{\therefore \quad k_2 = 1.1}$$

From equation 8.5.24, $k_3 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{hk_2}{2}\right)$

$$= f\left(0 + \frac{0.1}{2}, 1 + \frac{0.1 \times 1.1}{2}\right)$$
$$= f(0.05,\ 1.055)$$
$$= 0.05 + 1.055$$

$$\boxed{\therefore \quad k_3 = 1.105}$$

From equation 8.5.25, $k_4 = f(x_0 + h,\ y_0 + hk_3)$

$$= f(0 + 0.1, 1 + 0.1 \times 1.105)$$
$$= f(0.1, 1.1105)$$
$$= 0.1 + 1.1105$$
$$= 1.2105$$

$$\boxed{\therefore \quad k_4 = 1.2105}$$

Putting values of k_1, k_2, k_3 and k_4 in equation 8.5.21 we get,

$$y_1 = y_0 + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$
$$= 1 + \frac{0.1}{6}[1 + 2(1.1) + 2(1.105) + 1.2105]$$
$$= 1.1103417$$

Step 2 :

For this step we have,

$x_1 = 0.1,\quad y_1 = 1.1103417$ from previous step

$h = 0.1$ and $x_2 = x_1 + h$

$$= 0.1 + 0.1$$

$= 0.2$

With $n = 2$ in equation 8.5.3 we get,

$$y_2 = y_1 + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \qquad \text{... (8.5.26)}$$

$$k_1 = f(x_1, y_1) \qquad \text{... (8.5.27)}$$

$$k_2 = f\left(x_1 + \frac{h}{2}, y_1 + \frac{hk_1}{2}\right) \qquad \text{... (8.5.28)}$$

$$k_3 = f\left(x_1 + \frac{h}{2}, y_1 + \frac{hk_2}{2}\right) \qquad \text{... (8.5.29)}$$

$$k_4 = f\left(x_1 + h,\ y_1 + hk_3\right) \qquad \text{... (8.5.30)}$$

From equation 8.5.27, $k_1 = f(x_1, y_1)$

$= f(0.1,\ 1.1103417)$

$= 0.1 + 1.1103417$

$= 1.2103417$ $\qquad \therefore \boxed{k_1 = 1.2103417}$

From equation 8.5.28, $k_2 = f\left(x_1 + \frac{h}{2},\ y_1 + \frac{hk_1}{2}\right)$

$$= f\left(0.1 + \frac{0.1}{2}, 1.1103417 + \frac{0.1 \times 1.2103417}{2}\right)$$

$= f(0.15,\ 1.1708588)$

$= 0.15 + 1.1708588$

$= 1.3208588$ $\qquad \therefore \boxed{k_2 = 1.3208588}$

From equation 8.5.29, $k_3 = f\left(x_1 + \frac{h}{2},\ y_1 + \frac{hk_2}{2}\right)$

$$= f\left(0.1 + \frac{0.1}{2}, 1.1103417 + \frac{0.1 \times 1.3208588}{2}\right)$$

$= f(0.15,\ 1.1763846)$

$= 0.15 + 1.1763846$

$= 1.3263846$ $\qquad \therefore \boxed{k_3 = 1.3263846}$

From equation 8.5.30, $k_4 = f\left(x_1 + h,\ y_1 + hk_3\right)$

$= f(0.1 + 0.1,\ 1.1103417 + 0.1 \times 1.3263846)$

$= f(0.2,\ 1.2429802)$

$= 0.2 + 1.2429802$

$= 1.4429802$ $\qquad \therefore \boxed{k_4 = 1.4429802}$

Putting values of k_1, k_2, k_3 and k_4 in equation 8.5.26 we get,

$$y_2 = y_1 + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

$$= 1.1103417 + \frac{0.1}{6}[1.2103417 + 2\,(1.3208588) + 2\,(1.3263846) + 1.4429802]$$

$$= 1.2428052$$

Thus the results are,

x	y
0	1
0.1	1.1103417
0.2	1.2428052

Ex. 8.5.5 *Solve the following equation by Runge-kutta method at* $x = 0.8$:

$$\frac{dy}{dx} = y - x$$

Take $x_0 = 0, y\,(0) = 2, h = 0.2$

Sol. : Here we have $x_0 = 0,\ y_0 = 2$ and $h = 0.2$

$$f\,(x, y) = \frac{dy}{dx} = y - x$$

Step 1 :

$$x_1 = x_0 + h = 0 + 0.2 = 0.2$$

We have to find y_1. Putting $n = 0$ in equation 8.5.3 we get k_1, k_2, k_3 and k_4 as follows :

$$k_1 = f\,(x_0, y_0) = y_0 - x_0 = 2 - 0 = 2 \qquad \therefore \boxed{k_1 = 2}$$

$$k_2 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{hk_1}{2}\right) = f\left(0 + \frac{0.2}{2}, 2 + \frac{0.2 \times 2}{2}\right) = f\,(0.1,\ 2.2) = 2.2 - 0.1 = 2.1 \qquad \therefore \boxed{k_2 = 2.1}$$

$$k_3 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{hk_2}{2}\right) = f\left(0 + \frac{0.2}{2}, 2 + \frac{0.2 \times 2.1}{2}\right) = f\,(0.1,\ 2.21) = 2.21 - 0.1$$

$$= 2.11 \qquad \therefore \boxed{k_3 = 2.11}$$

$$k_4 = f\left(x_0 + h,\ y_0 + hk_3\right)$$

$$= f\left(0 + 0.2,\ 2 + 0.2 \times 2.11\right)$$

$$= f\left(0.2,\ 2.422\right) = 2.422 - 0.2$$

$$= 2.222 \qquad \therefore \boxed{k_4 = 2.222}$$

Value of y_1 can be obtained by putting $n = 0$ in first equation of equation 8.5.3. i.e.,

$$y_1 = y_0 + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

Putting values of k_1, k_2, k_3 and k_4 obtained above,

$$y_1 = 2 + \frac{0.2}{6}\left[2 + 2\,(2.1) + 2\,(2.11) + 2.222\right]$$

$$= 2.4214$$

Thus we have $y_1 = 2.4214$ at $x_1 = 0.2$

Step 2 :

At this step we have, $x_1 = 0.2,\ y_1 = 2.4214$

And

$$x_2 = x_1 + h$$

$$= 0.2 + 0.2$$

$$= 0.4$$

We have to find y_2. Putting $n = 1$ in equation 8.5.3 we get k_1, k_2, k_3 and k_4 as follows :

$$k_1 = f\left(x_1, y_1\right)$$

$$= f\left(0.2,\ 2.4214\right) = 2.4214 - 0.2$$

$$= 2.2214 \qquad \therefore \boxed{k_1 = 2.2214}$$

$$k_2 = f\left(x_1 + \frac{h}{2},\ y_1 + \frac{hk_1}{2}\right)$$

$$= f\left(0.2 + \frac{0.2}{2},\ 2.4214 + \frac{0.2 \times 2.2214}{2}\right)$$

$$= f\left(0.3,\ 2.64354\right) = 2.64354 - 0.3$$

$$= 2.34354 \qquad \therefore \boxed{k_2 = 2.34354}$$

$$k_3 = f\left(x_1 + \frac{h}{2},\ y_1 + \frac{hk_2}{2}\right)$$

$$= f\left(0.2 + \frac{0.2}{2},\ 2.4214 + \frac{0.2 \times 2.34354}{2}\right)$$

$$= f\left(0.3,\ 2.655754\right) = 2.655754 - 0.3$$

$$= 2.355754 \qquad \therefore \boxed{k_3 = 2.355754}$$

$$k_4 = f\left(x_1 + h,\ y_1 + hk_3\right)$$

$$= f\,(0.2+0.2,\,2.4214+0.2\times 2.355754)$$
$$= f\,(0.4,\ 2.8925508) = 2.8925508 - 0.4$$
$$= 2.4925508 \qquad \therefore \quad \boxed{k_4 = 2.4925508}$$

y_2 can be obtained by putting $n=1$ in first equation of equation 8.5.3 i.e.,

$$y_2 = y_1 + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$
$$= 2.4214 + \frac{0.2}{6}\left[2.2214 + 2(2.34354) + 2(2.355754) + 2.4925508\right]$$
$$= 2.891818$$

Thus we have $y_2 = 2.891818$ at $x_2 = 0.4$

Similarly we can obtain y_3 at $x_3 = x_2 + h = 0.6$ and y_4 at $x_4 = x_3 + h = 0.8$

The results are tabulated below :

x	y
0	2
0.2	2.4214
0.4	2.891818
0.6	3.422106
0.8	4.025521

Ex. 8.5.6 *Solve the equation* $\frac{dy}{dx} = \sqrt{x+y}$ *, subject to* $x=0, y=1$ *to find* y *at* $x=0.2$ *taking* $h=0.1$. **[Dec - 99, 8 marks, Dec - 97, 8 marks]]**

Sol. : Here we have given as,

$$f\,(x,y) = \frac{dy}{dx} = \sqrt{x+y}$$
$$x_0 = 0,\ y_0 = 1,\ h = 0.1$$

Step 1 : To find y at x = 0.1

$$x_1 = x_0 + h$$
$$= 0 + 0.1 = 0.1$$

We have to calculate y_1 at $x_1 = 0.1$. With $n=0$ in equation 8.5.3 we can calculate k_1, k_2, k_3 and k_4 as follows :

$$k_1 = f\,(x_0, y_0)$$
$$= f\,(0,1)$$
$$= \sqrt{0+1} = 1 \qquad \therefore \quad \boxed{k_1 = 1}$$

$$k_2 = f\left(x_0 + \frac{h}{2},\ y_0 + \frac{hk_1}{2}\right)$$

$$= f\left(0 + \frac{0.1}{2}, 1 + \frac{0.1 \times 1}{2}\right)$$

$$= f\ (0.05,\ 1.05) = \sqrt{0.05 + 1.05}$$

$$= 1.0488088 \qquad \therefore \boxed{k_2 = 1.0488088}$$

$$k_3 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{hk_2}{2}\right)$$

$$= f\left(0 + \frac{0.1}{2}, 1 + \frac{0.1 \times 1.0488088}{2}\right)$$

$$= f\ (0.05,\ 1.0524404) = \sqrt{0.05 + 1.0524404}$$

$$= 1.0499716 \qquad \therefore \boxed{k_3 = 1.0499716}$$

$$k_4 = f\ (x_0 + h,\ y_0 + hk_3)$$

$$= f\ (0 + 0.1\ ,\ 1 + 0.1 \times 1.0499716)$$

$$= f\ (0.1,\ 1.1049972) = \sqrt{0.1 + 1.1049972}$$

$$= 1.0977236 \qquad \therefore \boxed{k_4 = 1.0977236}$$

y_1 can be obtained by putting $n = 0$ in first equation of equation 8.5.3. i.e.,

$$y_1 = y_0 + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

Putting values in above equation,

$$y_1 = 1 + \frac{0.1}{6}\ [1 + 2\ (1.0488088) + 2\ (1.0499716) + 1.0977236]$$

$$= 1.1049214$$

Step 2 : To find y at x = 0.2

For this step, $x_2 = x_1 + h$

$$= 0.1 + 0.1 = 0.2$$

And initial conditions are,

$$x_1 = 0.1 \text{ and } y_1 = 1.1049214$$

Let us calculate k_1, k_2, k_3 and k_4 by putting $n = 1$ in equation 8.5.3. i.e.,

$$k_1 = f\ (x_1, y_1)$$

$$= f\ (0.1,\ 1.1049214) = \sqrt{0.1 + 1.1049214}$$

$$= 1.0976891 \qquad \therefore \boxed{k_1 = 1.0976891}$$

$$k_2 = f\left(x_1 + \frac{h}{2}, y_1 + \frac{hk_1}{2}\right)$$

$$= f\left(0.1+\frac{0.1}{2}, 1.1049214+\frac{0.1\times 1.0976891}{2}\right)$$

$$= f(0.15,\ 1.1598059) = \sqrt{0.15+1.1598059}$$

$$= 1.1444675 \qquad \therefore \boxed{k_2 = 1.1444675}$$

$$k_3 = f\left(x_1+\frac{h}{2}, y_1+\frac{hk_2}{2}\right)$$

$$= f\left(0.1+\frac{0.1}{2}, 1.1049214+\frac{0.1\times 1.1444675}{2}\right)$$

$$= f(0.15,\ 1.1621448) = \sqrt{0.15+1.1621448}$$

$$= 1.1454889 \qquad \therefore \boxed{k_3 = 1.1454889}$$

$$k_4 = f(x_1+h,\ y_1+hk_3)$$

$$= f(0.1+0.1, 1.1049214+0.1\times 1.1454889)$$

$$= f(0.2,\ 1.2194703) = \sqrt{0.2+1.2194703}$$

$$= 1.1914152 \qquad \therefore \boxed{k_4 = 1.1914152}$$

y_2 can be obtained by putting $n=1$ in first equation of equation 8.5.3 i.e.,

$$y_2 = y_1+\frac{h}{6}(k_1+2k_2+2k_3+k_4)$$

Putting values in above equation,

$$y_2 = 1.1049214+\frac{0.1}{6}[1.0976891+2\,(1.1444675)+2\,(1.1454889)+1.1914152]$$

$$= 1.3171587$$

Thus the results are

x	y
0	1
0.1	1.1049214
0.2	1.3171587

8.5.5 Solution of Higher Order Differential Equations (or systems of simultaneous differential equations)

Consider the second order differential equation of the form,

$$\frac{d^2y}{dx^2} = \phi\left(x, y, \frac{dy}{dx}\right) \qquad \text{... (8.5.31)}$$

Here let

$$\frac{dy}{dx} = z \qquad \text{... (8.5.32)}$$

Then we can write equation 8.5.31 as,

$$\frac{dz}{dx} = \phi(x, y, z) \qquad \text{... (8.5.33)}$$

and we can write equation 8.5.32 as,

$$\frac{dy}{dx} = f(x, y, z) \quad \text{(Here } \phi \text{ is some function of x, y \& z)... (8.5.34)}$$

Thus we have two simultaneous differential equations of first order. These equations can be solved by our regular methods.

We can write 4^{th} order Runge Kutta formula for above system of equation by extending equation 8.5.3 to three variables x, y and z.

Solution to equation 8.5.33 by 4^{th} order Runge Kutta Method :

$$\left.\begin{aligned} z_{n+1} &= z_n + \frac{h}{6}(m_1 + 2m_2 + 2m_3 + m_4) \\ \text{where,} \quad m_1 &= \phi\ (x_n, y_n, z_n) \\ m_2 &= \phi\left(x_n + \frac{h}{2},\ y_n + \frac{hk_1}{2},\ z_n + \frac{hm_1}{2}\right) \\ m_3 &= \phi\left(x_n + \frac{h}{2},\ y_n + \frac{hk_2}{2},\ z_n + \frac{hm_2}{2}\right) \\ m_4 &= \phi\ (x_n + h,\ y_n + hk_3,\ z_n + hm_3) \end{aligned}\right\} \qquad \text{... (8.5.35)}$$

Similarly solution to equation 8.5.34 by 4^{th} order Runge Kutta Method :

$$\left.\begin{aligned} y_{n+1} &= y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ \text{where,} \quad k_1 &= f\ (x_n, y_n, z_n) \\ k_2 &= f\left(x_n + \frac{h}{2},\ y_n + \frac{hk_1}{2},\ z_n + \frac{hm_1}{2}\right) \\ k_3 &= f\left(x_n + \frac{h}{2},\ y_n + \frac{hk_2}{2},\ z_n + \frac{hm_2}{2}\right) \\ k_4 &= f\ (x_n + h,\ y_n + hk_3,\ z_n + hm_3) \end{aligned}\right\} \qquad \text{... (8.5.36)}$$

Ex.8.5.7 *Solve the following equations*

$$\frac{d^2y}{dx^2} = x\frac{dy}{dx} - y$$

using 4^{th} order runge kutta method at x = 0.2. Use initial values as x = 0, y = 1 and $\frac{dy}{dx} = 0$. Take h = 0.2.

Sol. : Let $\frac{dy}{dx} = z$ then the given equation becomes,

$$\frac{dz}{dx} = xz - y$$

Thus we have two differential equations

$$\frac{dy}{dx} = z \quad \Rightarrow f(x, y, z) = z \qquad \text{... (8.5.37)}$$

&

$$\frac{dz}{dx} = xz - y \Rightarrow \quad \phi(x, y, z) = xz - y \qquad \text{... (8.5.38)}$$

We have initial condition as,

$$x_0 = 0, \qquad y_0 = 1 \qquad \& \quad z_0 = 0 \qquad \left(\because \quad \frac{dy}{dx} = z = 0\right)$$

$$h = 0.2$$

With n = 0 let's apply equation 8.5.36 we get,

From equation 8.5.36 with n = 0 we have,

$$\begin{aligned} k_1 &= f(x_0, y_0, z_0) \\ &= f(0, 1, 0) \\ &= z \\ &= 0 \end{aligned}$$

$$\therefore \boxed{k_1 = 0}$$

From equation 8.5.35 with n = 0 we have,

$$\begin{aligned} m_1 &= \phi(x_0, y_0, z_0) \\ &= \phi(0, 1, 0) \\ &= xz - y \\ &= 0(0) - 1 \\ &= -1 \end{aligned}$$

$$\therefore \boxed{m_1 = -1}$$

Similarly,

$$\begin{aligned} k_2 &= f\left(x_0 + \frac{h}{2}, y_0 + \frac{hk_1}{2}, z_0 + \frac{hm_1}{2}\right) \\ &= f\left(0 + \frac{0.2}{2}, 1 + \frac{0.2(0)}{2}, 0 + \frac{0.2(-1)}{2}\right) \\ &= f[0.1, 1, -0.1] \\ &= z \\ &= -0.1 \end{aligned}$$

$$\therefore \boxed{k_2 = -0.1}$$

$$\begin{aligned} m_2 &= \phi\left(x_0 + \frac{h}{2}, y_0 + \frac{hk_1}{2}, z_0 + \frac{hm_1}{2}\right) \\ &= \phi\left[0 + \frac{0.2}{2}, 1 + \frac{0.2(0)}{2}, 0 + \frac{0.2(-1)}{2}\right] \\ &= \phi[0.1, 1, -0.1] \\ &= xz - y \\ &= 0.1(-0.1) - 1 \\ &= -1.01 \end{aligned}$$

$$\therefore \boxed{m_2 = -1.01}$$

$$k_3 = f\left(x_0 + \frac{h}{2},\ y_0 + \frac{hk_2}{2},\ z_0 + \frac{hm_2}{2}\right)$$

$$= f\left(0 + \frac{0.2}{2},\ 1 + \frac{0.2\,(-0.1)}{2},\ 0 + \frac{0.2\,(-1.01)}{2}\right)$$

$$= f\,[0.1,\ 0.99,\ -0.101]$$

$$= z$$

$$= -0.101 \qquad \therefore \boxed{k_3 = -0.101}$$

$$m_3 = \phi\left(x_0 + \frac{h}{2},\ y_0 + \frac{hk_2}{2},\ z_0 + \frac{hm_2}{2}\right)$$

$$= \phi\left[0 + \frac{0.2}{2},\ 1 + \frac{0.2\,(-0.1)}{2},\ 0 + \frac{0.2\,(-1.01)}{2}\right]$$

$$= \phi\,[0.1,\ 0.99,\ -0.101]$$

$$= xz - y$$

$$= 0.1\,(-0.101) - 0.99$$

$$= -1.0001 \qquad \therefore \boxed{m_3 = -1.0001}$$

$$k_4 = f\,(x_0 + h,\ y_0 + hk_3,\ z_0 + hm_3)$$

$$= f\,[0 + 0.2,\ 1 + 0.2\,(-0.101),\ 0 + 0.2\,(-1.0001)]$$

$$= f\,(0.2,\ 0.9798,\ -0.20002)$$

$$= z$$

$$= -0.20002 \qquad \therefore \boxed{k_4 = -0.20002}$$

$$m_4 = \phi\,(x_0 + h,\ y_0 + hk_3,\ z_0 + hm_3)$$

$$= \phi\,(0.2,\ 0.9798,\ -0.20002)$$

$$= xz - y$$

$$= 0.2\,(-0.20002) - 0.9798$$

$$= -1.019804 \qquad \therefore \boxed{m_4 = -1.019804}$$

$\therefore$ From equation 8.5.35,

$$z_1 = z_0 + \frac{h}{6}\,(m_1 + 2m_2 + 2m_3 + m_4)$$

$$= 0 + \frac{0.2}{6}\,[-1 + 2\,(-1.01) + 2\,(-1.0001) - 1.019804]$$

$$= -0.2013334$$

and from equation 8.5.36 we have,

$$y_1 = y_0 + \frac{h}{6}\,(k_1 + 2k_2 + 2k_3 + k_4)$$

$$= 1 + \frac{0.2}{6}\,[0 + 2\,(-0.1) + 2\,(-0.101) - 0.20002]$$

$$= 0.9799326$$

Thus we have at x = 0.2, y = 0.9799326 (Ans.)

Ex. 8.5.8 *Employ Runge-kutta method to calculate y for x = 0.2 from equation :*

$$\frac{d^2y}{dx^2} = x\left(\frac{dy}{dx}\right)^2 - y^2$$

Given : $y = 1, \frac{dy}{dx} = 0$ for $x = 0$, $h = 0.1$.

Sol. : Let $\frac{dy}{dx} = z$, then the given equation becomes,

$$\frac{dz}{dx} = xz^2 - y^2$$

Thus we have two differential equations :

$$\frac{dy}{dx} = z \quad \Rightarrow \quad f(x, y, z) = z \qquad \ldots (8.5.39)$$

and

$$\frac{dz}{dx} = xz^2 - y^2 \quad \Rightarrow \quad \phi(x, y, z) = xz^2 - y^2 \qquad \ldots (8.5.40)$$

We have initial conditions as,

$$x_0 = 0, \quad y_0 = 1, \quad z_0 = 0 \quad \text{and} \quad h = 0.1$$

Step 1 :

In this step let us calculate y_1 at x_1. Here,

$$\begin{aligned} x_1 &= x_0 + h \\ &= 0 + 0.1 \\ &= 0.1 \end{aligned}$$

We will put $n = 0$ in equation 8.5.35 and equation 8.5.36 to get z_1 and y_1. These calculations are given as follows :

$$\begin{aligned} k_1 &= f(x_0, y_0, z_0) \\ &= z_0 = 0 \end{aligned} \qquad \therefore \boxed{k_1 = 0}$$

$$\begin{aligned} m_1 &= \phi(x_0, y_0, z_0) \\ &= x_0 z_0^2 - y_0^2 = 0 - (1)^2 \\ &= -1 \end{aligned} \qquad \therefore \boxed{m_1 = -1}$$

$$\begin{aligned} k_2 &= f\left[x_0 + \frac{h}{2}, y_0 + \frac{hk_1}{2}, z_0 + \frac{hm_1}{2}\right] \\ &= f\left[0 + \frac{0.1}{2}, 1 + \frac{(0.1)\times 0}{2}, 0 + \frac{0.1(-1)}{2}\right] \\ &= f[0.05, 1, -0.05] \\ &= z = -0.05 \end{aligned} \qquad \therefore \boxed{k_2 = -0.05}$$

$$m_2 = \phi\left[x_0+\frac{h}{2}, y_0+\frac{hk_1}{2}, z_0+\frac{hm_1}{2}\right]$$
$$= \phi\,[0.05,\ 1,\ -0.05]$$
$$= xz^2-y^2 = 0.05\,(-0.05)^2-(1)^2$$
$$= -0.999875 \qquad \therefore \boxed{m_2 = -0.999875}$$

$$k_3 = f\left[x_0+\frac{h}{2}, y_0+\frac{hk_2}{2}, z_0+\frac{hm_2}{2}\right]$$
$$= f\left[0+\frac{0.1}{2}, 1+\frac{0.1\,(-0.05)}{2}, 0+\frac{0.1(-0.999875)}{2}\right]$$
$$= f\,[0.05,\ 0.9975,\ -0.0499937]$$
$$= z = -0.0499937 \qquad \therefore \boxed{k_3 = -0.0499937}$$

$$m_3 = \phi\left[x_0+\frac{h}{2}, y_0+\frac{hk_2}{2}, z_0+\frac{hm_2}{2}\right]$$
$$= \phi\,[0.05, 0.9975, -0.0499937]$$
$$= xz^2-y^2 = 0.05\,(-0.0499937)^2-(0.9975)^2$$
$$= -0.9948812 \qquad \therefore \boxed{m_3 = -0.9948812}$$

$$k_4 = f\,[x_0+h,\ y_0+h\,k_3, z_0+hm_3]$$
$$= f\,[0+0.1, 1+0.1\,(-0.0499937), 0+0.1\,(-0.9948812)]$$
$$= f\,[0.1, 0.995, -0.0994881]$$
$$= z = -\ 0.0994881 \qquad \therefore \boxed{k_4 = -0.0994881}$$

$$m_4 = \phi\,[x_0+h, y_0+hk_3, z_0+hm_3]$$
$$= \phi\,[0.1,\ 0.995,\ -0.0994881]$$
$$= xz^2-y^2 = 0.1\,(-0.0994881)^2-(0.995)^2$$
$$= -0.9890352 \qquad \therefore \boxed{m_4 = -0.9890352}$$

To calculate z_1 and y_1, and $n=0$ in first equations of equation 8.5.35 and equation 8.5.36. i.e.,

$$z_1 = x_0+\frac{h}{6}\,[m_1+2m_2+2m_3+m_4]$$
$$= 0+\frac{0.1}{6}\,[-1+2(-0.999875)+2(-0.9948812)-0.9890352]$$
$$= -0.0996424$$

And
$$y_1 = y_0+\frac{h}{6}\,[k_1+2k_2+2k_3+k_4]$$
$$= 1+\frac{0.1}{6}\,[0+2\,(-0.05)+2\,(-0.0499937)-0.0994881]$$
$$= 0.995$$

Step 2 :

At the beginning of this step we have,

$x_1 = 0.1,\ y_1 = 0.995$ And $z_1 = -0.0996424$

Now we have,
$$x_2 = x_1 + h = 0.1 + 0.1 = 0.2$$

We will put $n = 1$ in equation 8.5.35 and equation 8.5.36 get z_2 and y_2. These calculations are given as follows :

$$\begin{aligned} k_1 &= f(x_1, y_1, z_1) \\ &= f(0.1, 0.995, -0.0996424) \\ &= z = -0.0996424 \end{aligned}$$

$$\therefore \boxed{k_1 = -0.0996424}$$

$$\begin{aligned} m_1 &= \phi(x_1, y_1, z_1) \\ &= \phi(0.1, 0.995, -0.0996424) \\ &= x z^2 - y^2 = 0.1(-0.0996424)^2 - (0.995)^2 \\ &= -0.9890321 \end{aligned}$$

$$\therefore \boxed{m_1 = -0.9890321}$$

$$\begin{aligned} k_2 &= f\left[x_1 + \frac{h}{2}, y + \frac{hk_1}{2}, z_1 + \frac{hm_1}{2}\right] \\ &= f\left[0.1 + \frac{0.1}{2}, 0.995 + \frac{0.1(-0.0996424)}{2}, -0.0996424 + \frac{0.1(-0.9890321)}{2}\right] \\ &= f[0.15, 0.99, -0.149094] \\ &= z = -0.149094 \end{aligned}$$

$$\therefore \boxed{k_2 = -0.149094}$$

$$\begin{aligned} m_2 &= \phi\left[x_1 + \frac{h}{2}, y_1 + \frac{hk_1}{2}, z_1 + \frac{hm_1}{2}\right] \\ &= \phi[0.15, 0.99, -0.149094] \\ &= x z^2 - y^2 = 0.15(-0.149094)^2 - (0.99)^2 \\ &= -0.9767656 \end{aligned}$$

$$\therefore \boxed{m_2 = -0.9767656}$$

$$\begin{aligned} k_3 &= f\left[x_1 + \frac{h}{2}, y_1 + \frac{hk_2}{2}, z_1 + \frac{hm_2}{2}\right] \\ &= f\left[0.1 + \frac{0.1}{2}, 0.995 + \frac{0.1(-0.149094)}{2}, -0.0996424 + \frac{0.1(-0.9767656)}{2}\right] \end{aligned}$$

$$= f\,[0.15,\ 0.9875453\,, -0.1484806]$$

$$= z = -0.1484806 \qquad \therefore \boxed{k_3 = -0.1484806}$$

$$m_3 = \phi\left[x_1 + \frac{h}{2}, y_1 + \frac{hk_2}{2}, z_1 + \frac{hm_2}{2}\right]$$

$$= \phi\,[0.15,\ 0.9875453,\ -0.1484806]$$

$$= xz^2 - y^2 = 0.15\,(-0.1484806)^2 - (0.9875453)^3$$

$$= -0.9719387 \qquad \therefore\ m_3 = -0.9719387$$

$$k_4 = f\,[x_1 + h, y_1 + hk_3, z_1 + hm_3]$$

$$= f\,[0.1 + 0.1, 0.995 + 0.1\,(-0.1484806),$$

$$-0.0996424 + 0.1\,(-0.9719387)]$$

$$= f\,[0.2, 0.9801519, -0.1968362]$$

$$= z = -0.1968362 \qquad \therefore \boxed{k_4 = -0.1968362}$$

$$m_4 = \phi\,[x_1 + h, y_1 + h\,k_3, z_1 + h\,m_3]$$

$$= \phi\,[0.2, 0.9801519, -0.1968362]$$

$$= xz^2 - y^2 = 0.2\,(-0.1968362)^2 - (0.9801519)^2$$

$$= -0.9529488 \qquad \therefore \boxed{m_4 = -0.9529488}$$

To obtain z_2 and y_2 let us put $n = 1$ in first equations of equation 8.5.35 and equation 8.5.36. i.e.,

$$y_2 = y_1 + \frac{h}{6}\,[k_1 + 2k_2 + 2k_3 + k_4]$$

$$= 0.995 + \frac{0.1}{6}[-0.0996424 + 2(-0.149094) +$$

$$2(-0.1484806) - 0.1968362]$$

$$= 0.9801395$$

Thus we have the results as follows :

x	y
0	1
0.1	0.995
0.2	0.9801395

Exercise

1. Using 4^{th} order Runge Kutta method, solve $y' = \frac{1}{x+y}$. Given that $x_0 = 0, y_0 = 1$. Calculate y at x = 0.6, taking h = 0.2.

Ans. :

x	y
0.2	1.13967
0.4	1.25589
0.6	1.3559

2. Using Runge Kutta fourth order method solve

$\frac{dy}{dx} = x^2 + y^2$ with $y(1) = 1.5$.

Find the value of y at 1.3. Take h = 0.1.

Ans. :

x	y
1.1	1.8955
1.2	2.5005
1.3	3.4263

8.5.6 Algorithm and C Program

Based on the illustration of Runge kutta method we will now prepare an algorithm for the Runge kutta method.

Assumption : It is assumed that the function $\frac{dy}{dx} = f(x, y)$ is defined in the program. It is not given externally. The algorithm given below is written for 4^{th} order Runge kutta method.

Algorithm :

Step 1 : Read initial value x_0 , y_0.

Step 2 : Read the value of x at which y is to be found.

Step 3 : Read the step size h.

Step 4 : Calculate

$$k_1 = f(x_n, y_n)$$

$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{hk_1}{2}\right)$$

$$k_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{hk_2}{2}\right)$$

$$k_4 = f(x_n + h, y_n + hk_3)$$

Step 5 : Calculate,

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Step 6 : Increase value of x_n by one step.

Step 7 : Repeat steps 4 to 6 till $x_n \leq x$

Step 8 : Display the values of x and y in tabular format.

Step 9 : Stop.

Flowchart :

Fig. 8.5.1 shows the flowchart for 4^{th} order Runge Kutta method. All the steps of algorithm are shown in the flowchart. The program given next, is base on this logic.

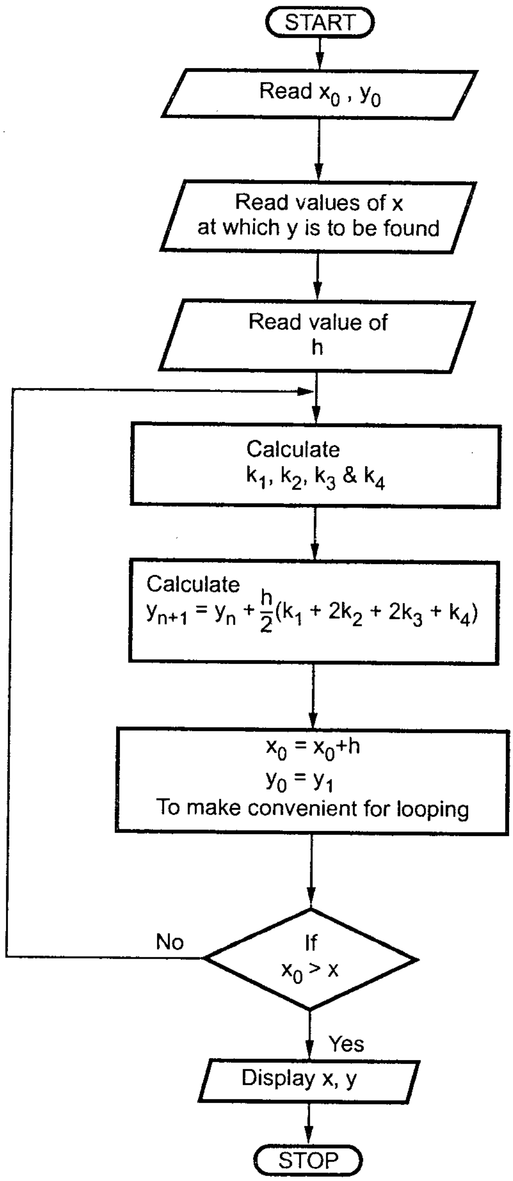


Fig. 8.5.1 Flowchart for Runge Kutta method

Computer program :

The source code of the C program for 4^{th} order Runge Kutta method is listed below.

```
/* Download this program from  www.vtubooks.com                          */
/* File name : rng_kut.cpp                                               */

/*----------- RUNG KUTTA METHOD TO SOLVE DIFFERENTIAL EQUATION --------*/

/*      THIS PROGRAM CALCULATES THE VALUE y AT GIVEN VALUE OF x
        USING FOURTH ORDER RUNG KUTTA METHOD. THE FUNCTION y' = f(x,y)
        IS DEFINED IN THE PROGRAM.
             y' = 1 + y*y
        Hence     f(x,y) = 1 + y*y

        INPUTS :  1) Initial values of x and y.

                  2) Step size h.

        OUTPUTS : Calculated values of y at every step.                   */

/*------------------------------    PROGRAM  ---------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>

void main()
{
      double f (double x0,double y0);  /* DECLARATION OF A FUNCTION f     */

      double y0,y1,x0,x1,h,x,k1,k2,k3,k4,k;
      int i,n;

      clrscr();
      printf("\n      RUNG KUTTA METHOD TO SOLVE DIFFERENTIAL EQUATION\n");

      printf("\n\nEnter x0 = ");
      scanf("%lf",&x0);                         /* ENTER VALUE OF x0      */
      printf("\n\nEnter y0 = ");
      scanf("%lf",&y0);                         /* ENTER VALUE OF y0      */

      printf("\n\nEnter the value of x at which y is to be found = ");
      scanf("%lf",&x);                          /* ENTER VALUE OF x       */

      printf("\n\nEnter the value of h = ");
      scanf("%lf",&h);                     /*   ENTER THE VALUE OF h      */

      i = 0;
      printf("\nPress any key to see step by step display of results...\n");
      while(x0 < x)      /* LOOP TO CALCULATE y USING RUNG KUTTA METHOD   */
      {
          i++;
          k1 = f(x0,y0);
          k2 = f(x0+h/2 , y0+(h*k1/2));
          k3 = f(x0+h/2 , y0+(h*k2/2));
          k4 = f(x0+h , y0+h*k3);
                           /* CALCULATION OF k USING RUNG KUTTA METHOD  */
          y1 = y0 + (h/6)*(k1 + 2*k2 + 2*k3 + k4);
                           /*     CALCULATION OF y FROM VALUES OF k     */
          x1 = x0 + h;
          printf("\nx%d = %lf       y%d = %lf",i,x1,i,y1);
          x0 = x1;    y0 = y1;
          getch();
      }
}
/*----------------------------------------------------------------------*/

double f ( double x,double y)  /* FUNCTION TO CALCULATE VALUE OF f(x,y)  */
{
    double y_dash;
    y_dash = 1 + y*y;           /*     function f(x,y) = y' = 1 + y*y     */
    return(y_dash);
}
/*----------------------- END OF PROGRAM ------------------------------*/
```

Observe that this program is exactly similar to the previous two programs upto `while` loop.

The program takes values of x_0, y_0 x & h as usual. Then it enters the `while` loop.

This loop calculates coefficients k_1, k_2, k_3, k_4 & k for every cycle. Observe the statement of `while` loop.

while(x0<x)

Here `x0` is compared with `x` in every iteration. In every iteration value of `x0` is increased by 'h'. Note that `x0` is not same as x_0, rather it is the value passed to function `f` in the cycle. The program prints values of x and y in every cycle.

After the `main` program, a function `f` is written. Observe the statement in function `f`,

y_dash = 1+y*y;

This implements,

$$\frac{dy}{dx} = 1 + y^2 \qquad \Rightarrow \qquad y' = f(x, y) = 1 + y^2$$

Thus the program solves above differential equation. If you want to use this program to solve some other differential equation, then you will have to change statement of `y_dash`.

How to Run this program?

Compile and make EXE file of this source code. As just discussed, the function `f` in the program is written to solve,

$$\frac{dy}{dx} = 1 + y^2 \qquad \text{with } y(0) = 0$$

Using 4^{th} order Runge kutta method. Take h = 0.2 and find y at x = 1.0.

Run the program on your computer. It will display,

`Enter x0 =` Here enter zero and press `enter' key

`Enter y0 =` Here enter zero and press `enter' key

`Enter the value of x at which y is to be found =` Here enter `1' and press `enter' key

`Enter the value of h =` Here enter `0.2' and press `enter' key

Then go on pressing any key to get step by step display of results. The combined results are shown below.

```
------------------------------ Results ----------------------------

      RUNG KUTTA METHOD TO SOLVE DIFFERENTIAL EQUATION

Enter x0 = 0

Enter y0 = 0

Enter the value of x at which y is to be found = 1

Enter the value of h = 0.2

Press any key to see step by step display of results...
```

```
x1 = 0.200000     y1 = 0.202707
x2 = 0.400000     y2 = 0.422789
x3 = 0.600000     y3 = 0.684133
x4 = 0.800000     y4 = 1.029637
x5 = 1.000000     y5 = 1.557352
--------------------------------------------------------------------------------
```

University Questions

1. Develop a C program for solving differential equations using Range Kutta fourth order formulae. **[Dec - 95, Dec - 96, May - 98, Dec - 99, Dec - 2000, May - 2001]**
2. Derive the second order Runge-Kutta method. **[May - 96]**
3. Use the classical fourth order Runge-Kutta method to integrate -

   $f(x, y) = -2x^3 + 12x^2 - 20x + 8.5$

   with a step size of 0.05 and an initial condition of y = 1 at x = 0 **[May - 96, Dec - 98]**
4. Derive the fourth order Runge-Kutta formula. **[May - 97, May - 99, May - 2000]**
5. Use the Runge-Kutta fourth order method to find the value or y when x = 1 given that y = 1 when x = 0 and that

   $\frac{dy}{dx} = \frac{y-x}{y+x}$ **[May - 97, Dec - 2000]**
6. Using Runge-Kutta's fourth order method, find the numerical solution at x = 0.8 for

   $\frac{dy}{dx} = \sqrt{x+y}, \; y(0.4) = 0.41$

   Assume step length h = 0.2 Also develop a suitable program for the above. **[Dec - 97, Dec - 99]**
7. Apply 4th order Runge-Kutta method to find y(1) with h = 0.1 given that

   $\frac{dy}{dx} = -xy^2; \qquad y(0) = 2$ **[May - 2001]**
8. Use Runge-Kutta fourth order method to find the value of y, when x = 1 given that y = 1 at x = 0 and

   $\frac{dy}{dx} = \frac{y-x}{y+x}$ **[Dec - 2001]**
9. Develop a C program to solve $\frac{dy}{dx} = f(x, y), y(x_0) = y_0$ with a step size 'h' at an interval $[x_0, x_n]$ using Runge Kutta 2^{nd} order method. **[Dec - 2001]**
10. Write the algorithm for RK 2^{nd} and 4^{th} order method. **[Dec - 2002]**
11. Compute y(0.1) and y(0.2) by R-K method of fourth order for differential equation

    $\frac{dy}{dx} + xy + y^2, \quad y(0) = 1$ **[May - 2003]**
12. Write a program in C/C++ for second order Runge-Kutta method. **[May - 2003]**
13. Derive the formula for solving ordinary differential equation using Runga-Kutta second order method. **[Dec - 2003]**

14. Solve $\frac{dy}{dx} = x + y^2$ with initial y = 1 when x = 0 for x = 0.2(0.2) 0.4, using Runga-Kutta fourth order method. **[Dec - 2003]**

15. Use RK 4th order method to solve :

    $f(x, y) = -2x^3 + 12x^2 - 20x + 8.5$

    Using a step size of h = 0.5 initial condition of y = 1 at x = 0 and write a program for the same. **[May - 2004]**

16. Use Euler's method to solve $\frac{dy}{dx} = -2x^3 + 12x^2 - 20x + 8.5$ using Eulers method and compare the result in part (a). Write the algorithm for solving differential equation using modified Eulers method. **[May - 2004]**

8.6 Predictor Corrector Methods

We have seen modified Euler's method. In this method the value of 'y' is first predicted. And then corrector formula is applied repeatedly to achieve higher accuracy in value of y. In this section we will discuss two predictor corrector methods. These methods need $y_n, y_{n-1}\ y_{n-2} \ldots$ to calculate y_{n+1}. Thus y_n, y_{n-1}, y_{n-2} etc. can be obtained by some other method and then predictor corrector methods are applied to obtain y_{n+1}.

8.6.1 Adams Methods

or Adams Bashforth or Adams Moulton method

These methods are also called multistep methods. The prediction formula for this method is given as,

$$\boxed{y_4^p = y_3 + \frac{h}{24}[55 f_3 - 59 f_2 + 37 f_1 - 9 f_0]} \quad \ldots (8.6.1)$$

Here

$f_0 = f(x_0, y_0)$, $f_1 = f(x_1, y_1)$, $f_2 = f(x_2, y_2)$ & so on.

This equation is called 4^{th} order Adams Bashforth formula. And the corrector formula is given as,

$$\boxed{y_4^c = y_3 + \frac{h}{24}\left[9 f_4^p + 19 f_3 - 5 f_2 + f_1\right]} \quad \ldots (8.6.2)$$

Here $f_4^p = f\left(x_4, f_4^p\right)$

This equation is called 4^{th} order Adams Moulton formula. Hence this method is some times called *Adams Bashforth* method or *Adams Moulton* method.

Error in Predictor formula,

$$E_n^p = \frac{251}{720} h^5 f_3^{(4)} \quad \ldots (8.6.3)$$

And error in corrector formula is given as,

$$E_n^c = -\frac{19}{720} h^5 f_3^{(4)} \qquad \text{... (8.6.4)}$$

8.6.2 Milne's Method

This is most common multistep method based on basic integration formula.

The predictor formula is given as,

$$y_4^p = y_0 + \frac{4h}{3} [2f_1 - f_2 + 2f_3] \qquad \text{... (8.6.5)}$$

And the corrector formula will be,

$$\boxed{y_4^c = y_2 + \frac{h}{3} \left[f_2 + 4 f_3 + f_4^p\right]} \qquad \text{... (8.6.6)}$$

The formulae given for Adams and Milne's method here give value of y_4. However if you want to evaluate y_5, then change other functions accordingly. Specific values are given for better understanding.

8.6.3 Solved Examples

Ex. 8.6.1 *Solve the following differential equation using Adams method.*

$$\frac{dy}{dx} = 1+y^2 \quad \textit{with } y = 0 \textit{ when } x = 0$$

Take h = 0.2 and find y at x = 0.8

Following values are obtained previously by Runge Kutta method.

x	0	0.2	0.4	0.6
y	0	0.2027	0.4228	0.6841

Sol. : We have

$$h = 0.2$$

$$\frac{dy}{dx} = 1+y^2$$

i.e.

$$f(x,y) = 1+y^2$$

Let's prepare the following table from given data.

From this table it is clear that values upto x_3 , y_3 are given. Hence we can find y_4 at x_4 using Adams method.

Table 8.6.1 : Table of x, y & f (x, y)

x	y	f (x, y) = 1 + y^2
$x_0 = 0$	$y_0 = 0$	$f_0 = f(x_0, y_0) = 1 + y_0^2$ $= 1 + 0 = 1$
$x_1 = 0.2$	$y_1 = 0.2027$	$f_1 = f(x_1, y_1) = 1 + y_1^2$ $= 1 + (0.2027)^2 = 1.0411$

$x_2 = 0.4$	$y_2 = 0.4228$	$f_2 = f(x_2, y_2) = 1 + y_2^2$ $1 + (0.4228)^2 = 1.1787$
$x_3 = 0.6$	$y_3 = 0.6841$	$f_3 = f(x_3, y_3) = 1 + y_3^2$ $= 1 + (0.6841)^2 = 1.4681$

To find y at x = 0.8 $\rightarrow$ $x_3 + h = x_4$,

Lets use Adam's predictor formula of equation 8.6.1,

$$y_4^p = y_3 + \frac{h}{24}[55 f_3 - 59 f_2 + 37 f_1 - 9 f_0]$$

$$y_4^p = 0.6841 + \frac{0.2}{24}[55 \times 1.4681 - 59 \times 1.1787 + 37 \times 1.0411 - 9 \times 1]$$

$$= 1.0233$$

Let's find f_4^p.

$$\therefore \quad f_4^p = f\left(x_4, y_4^p\right) = 1 + \left(y_4^p\right)^2$$

$$= 1 + 1.0233$$

$$= 2.047465$$

Now let's use corrector formula of equation 8.6.2 to improve y_4. i.e.,

$$y_4^c = y_3 + \frac{h}{24}\left[9 f_4^p + 19 f_3 - 5 f_2 + f_1\right]$$

$$= 0.6841 + \frac{0.2}{24}\begin{bmatrix} 9 \times 2.047465 + 19 \times 1.4681 \\ -5 \times 1.1787 + 1.0411 \end{bmatrix}$$

$$= 1.0296$$

Thus at $\boxed{x_4 = 0.8 \qquad y_4 = 1.0296}$ is the answer.

Ex.8.6.2 *Let's solve the same differential equation* $\frac{dy}{dx} = 1 + y^2$ *of example 8.6.1 using Milne's method at x = 0.8. The data remains same.*

Sol. : Refer to the Table 8.6.1 we have prepared in example 8.6.1. We will use this table in this example,

Let's use predictor formula of equation 8.6.5 of Milne's method.

$$y_4^p = y_0 + \frac{4h}{3}[2f_1 - f_2 + 2f_3]$$

Putting values of f_1, f_2 and f_3 from Table 8.6.1,

$$\therefore \quad y_4^p = 0 + \frac{4 \times 0.2}{3}[2 \times 1.0411 - 1.1787 + 2 \times 1.4681]$$

$$= 1.0239$$

Now let's find $\quad f_4^p = f\left(x_4, y_4^p\right)$

$$= 1 + \left(y_4^p\right)^2$$

$$= 1 + (1.0239)^2$$

$$= 2.0480$$

To correct this value of y_4^p we apply corrector formula of equation 8.6.6. i.e.,

$$y_4^c = y_2 + \frac{h}{3}\left[f_2 + 4f_3 + f_4^p\right]$$

$$= 0.4228 + \frac{0.2}{3}[1.1787 + 4 \times 1.4681 + 2.0480]$$

$$= 1.0294$$

Thus using Milne's method

At $\boxed{x_4 = 0.8, \quad y_4 = 1.0294}$ is the answer.

Ex. 8.6.3 *Using any one of the three predictor corrector methods, and given the three starting values y_1, y_2 and y_3 and $h = 0.05$. Calculate the next value y_4 for the initial value problem $y' = t^2 - y$ with $y(0) = 0$. $[a, b] = 0.5$.*

$y_1 = y\ (0.05) = 0.95127058$
$y_2 = y\ (0.10) = 0.90516258$
$y_3 = y(0.15) = 0.86179202$
$y(t) = -e^{-t} + t^2 - 2t^2 + 2$

[Dec - 2000, 8 marks, Dec - 96, 8 marks]

Sol. : Here $\qquad \dfrac{dy}{dx} = t^2 - y$

i.e. $\qquad f(x, y) = x^2 - y$

Let us prepare the following table from the given data :

x	y	$f(x, y) = x^2 - y$
$x_0 = 0$	$y_0 = 0$	$f_0 = f(x_0, y_0) = x_0^2 - y_0$ $= (0)^2 - 0 = 0$
$x_1 = 0.05$	$y_1 = 0.95127058$	$f_1 = f(x_1, y_1) = x_1^2 - y_1$ $= (0.05)^2 - 0.95127058 = -0.9487705$
$x_2 = 0.10$	$y_2 = 0.90516258$	$f_2 = f(x_2, y_2) = x_2^2 - y_2$ $= (0.1)^2 - 0.90516258 = -0.8951625$
$x_3 = 0.15$	$y_3 = 0.86179202$	$f_3 = f(x_3, y_3) = x_3^2 - y_3$ $= (0.15)^2 - 0.86179202 = -0.839292$

Now we have to find y_4. We will use Adams method. We know that,

$$x_4 = x_3 + h$$
$$= 0.15 + 0.05 = 0.2$$

Consider Adams predictor formula given by equation 8.6.1,

$$y_4^p = y_3 + \frac{h}{24}[55 f_3 - 59 f_2 + 37 f_1 - 9 f_0]$$

Putting the values from table,

$$y_4^p = 0.86179202 + \frac{0.05}{24}\{55(-0.839292) - 59(-0.8951625) + 37(-0.9487705) - 9(0)\}$$

$$= 0.8025191$$

Let us find f_4^p. i.e.,

$$f_4^p = f\left(x_4, y_4^p\right)$$
$$= x_4^2 - y_4^p$$
$$= (0.2)^2 - (0.8025191)$$
$$= -0.7625191$$

Now let us use corrector formula of equation 8.6.2 to improve y_4. i.e.,

$$y_4^c = y_3 + \frac{h}{24}\left[9 f_4^p + 19 f_3 - 5 f_2 + f_1\right]$$

$$= 0.86179202 + \frac{0.05}{24}\{9(-0.7625191) + 19(-0.839292) - 5(-0.8951625) + (-0.9487705)\}$$

$$= 0.8216208$$

Thus at $x_4 = 0.2$, $y_4 = 0.8216208$.

Exercise

1. Solve the equation $\frac{dy}{dx} = x + y$ using Adam's method. Find value of y at x = 0.4. Take h = 0.1. Tha values of x and y satisfied by this differential equation are given below.

x	0	0.1	0.2	0.3
y	1	1.110342	1.242806	1.399718

[Hint : Here you have to find y_4 at $x_4 = 0.3 + h = 0.4$] **[Ans. : 1.503650]**

2. Solve $\frac{dy}{dx} = \frac{1}{x+y}$ using Milne's method. Find value of y at x = 2.0.

The differential equation satisfies following values of x and y.

x	0	0.5	1.0	1.5
y	1	1.3571	1.5837	1.7565

[Hint : Take h = 0.5 and you have to find y_4 at $x_4 = 1.5 + h = 2.0$] **[Ans. : 2.023]**

8.6.4 Algorithm

Here let us consider an algorithm for predictor corrector methods. The predictor and corrector equations given by equation 8.6.1, 8.6.2, 8.6.5 and 8.6.6 are for specific values. The generalized equations for milne's method are given as,

Predictor formula : $y_{n+1}^{p} = y_{n-3} + \frac{4h}{3}[2f_{n-2} - f_{n-1} + 2f_n]$ ... (8.6.7)

Corrector formula : $y_{n+1}^{c} = y_{n-1} + \frac{h}{3}\left[f_{n-1} + 4f_n + f_{n+1}^{p}\right]$ ... (8.6.8)

Here $f_{n+1}^{p} = f\left(x_{n+1}, y_{n+1}^{p}\right)$ ... (8.6.9)

Here note that similar generalized equations can be written for Adams Bashforth method.

Assumptions :

Here we will assume that $\frac{dy}{dx} = f(x, y)$ is defined in the program. It is not given externally.

Algorithm :

Step 1 : Read initial values x_0 and y_0

Step 2 : Read value of x at which y is to be evaluated.

Step 3 : Read step size h.

Step 4 : Calculate y_{n+1}^{p} using predictor formula

Step 5 : Calculate $x_{n+1} = x_n + h$

Step 6 : Calculate $f_{n+1}^{p} = f\left(x_{n+1}, y_{n+1}^{p}\right)$

Step 7 : Calculate y_{n+1}^{c} using corrector formula

Step 8 : If x_{n+1} is the required value of x, then go to next step. Otherwise, $n = n + 1$ And go to step 4.

Step 9 : The required value of y is, $y = y_{n+1}^{c}$ at given x.

Step 10 : Display value of 'y' and stop.

University Questions

1. Using any one of the three predictor-corrector methods and given the three starting values y_1, y_2 and y_3 and h = 0.05, calculate the next y_4 for the initial value problem

   $y' = t^2 - y$, with y(0) = 0

   [a, b] = [0, 5]

   y_1 = y(0.05) = 0.95127058

   y_2 = y(0.10) = 0.90516258

   y_3 = y(0.15) = 0.86179202

   $y(t) = -e^{-1} + t^2 - 2t + 2$

   [Dec - 96, Dec - 2000]

2. Use the predictor-corrector formula for tabulating a solution of

   $$10\frac{dy}{dx} = x^2 + y^2,\ y(0) = 1$$

   for the range $0.5 \leq x \leq 1.0$ **[May - 97, Dec - 98, May - 2000]**

8.7 Boundary-Value and Eigen Value Problems Initial Value Problems

In the preceding sections we studied the methods to solve *initial value problems.* For initial value problems, all the conditions are specified at the same value of independent variable. For example consider Fig. 8.7.1.

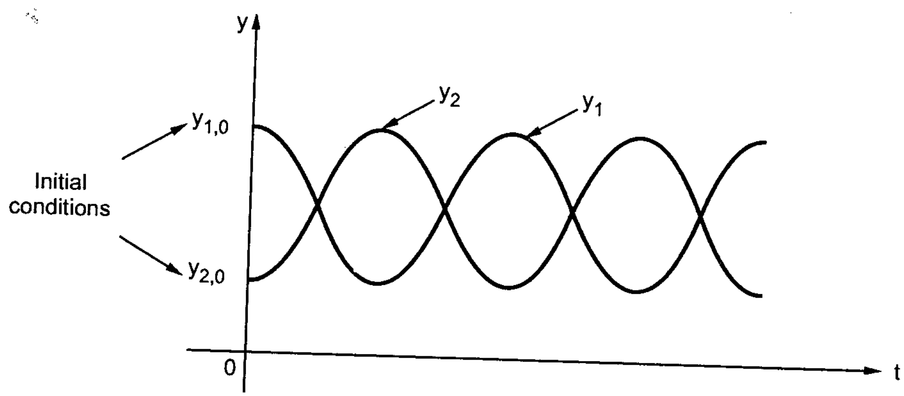


Fig. 8.7.1 Initial value problem

In the figure,

$$\frac{dy_1}{dt} = f_1(t, y_1, y_2)$$

and

$$\frac{dy_2}{dt} = f_2(t, y_1, y_2)$$

And at $t = 0$, $y_1 = y_{1,0}$ and $y_2 = y_{2,0}$. Thus the conditions for both y_1 and y_2 are specified at the same point, i.e. $t = 0$. Hence, this is an initial value problem.

Boundary value problems :

In some other applications the conditions are not known at the single point. The conditions are specified at the different values of the independent variable. Normally these conditions are specified at the boundaries. Hence such problems are called *boundary value problems.* Consider the function shown in Fig. 8.7.2.

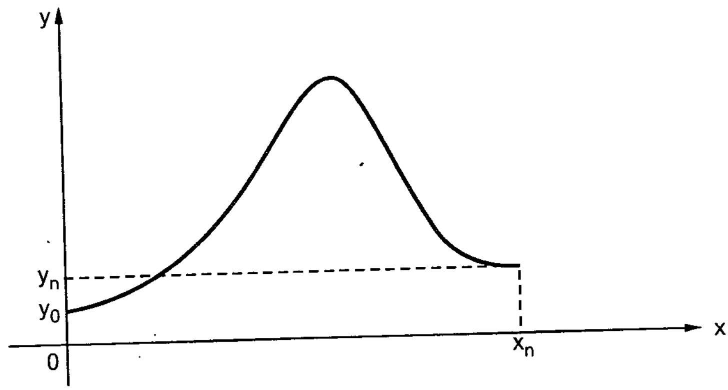


Fig. 8.7.2 Boundary value problem

In the above figure,

$$\frac{d^2y}{dx^2} = f(x,y)$$

And at $x = 0, \quad y = y_0$

and $x = x_n, \quad y = y_n$

Thus the conditions are specified at the boundaries $(x_0 \text{ and } x_n)$. Hence this is a boundary value problem. Large number of engineering application are described as boundary value problems.

Eigen-value problems :

Eigen-values are also called as characteristic values. Eigen-value problems are a special class of boundary value problems. Applications such as vibrations, oscillating systems, elasticity etc are described by eigen-value problems.

8.7.1 Boundary Value Problems

Consider the following example of two point linear boundary value problem :

$$y''(x) + f(x)\, y'(x) + g(x)\, y(x) = r(x) \qquad \text{... (8.7.1)}$$

And the boundary conditions,

$$\left.\begin{aligned} y(x_0) &= a \\ y(x_n) &= b \end{aligned}\right\} \qquad \text{... (8.7.2)}$$

These problems can be solved by finite difference method and shooting method. Here we will discuss finite difference method to solve the boundary value problems.

Finite difference method to solve boundary value problems

In this method, the derivatives in the differential equation are replaced by their finite difference approximations. Because of this, the differential equations are converted into linear equations. The linear equations are then solved by usual methods. The central difference approximation of y'_i is given as,

$$y'_i = \frac{y_{i+1} - y_{i-1}}{2h} \quad \text{... (8.7.3)}$$

Similarly central difference approximation of y''_i is given as,

$$y''_i = \frac{y_{i-1} - 2\,y_i + y_{i+1}}{h^2} \quad \text{... (8.7.4)}$$

Here $y_i = f(x_i) = f(x_0 + i\,h)$

The range $[x_0, x_n]$ is divided into n subintervals of width h. At $x = x_i$ we can write equation 8.7.1 as,

$$y''_i + f_i\; y'_i + g_i\; y_i = r_i$$

Putting for y''_i and y'_i in above equation,

$$\frac{y_{i-1} - 2\,y_i + y_{i+1}}{h^2} + f_i\,\frac{y_{i+1} - y_{i-1}}{2h} + g_i\; y_i = r_i$$

Multiplying by h^2 we get,

$$y_{i-1} - 2\,y_i + y_{i+1} + f_i\,\frac{(y_{i+1} - y_{i-1})}{2}\,h + g_i\,y_i\;h^2 = r_i\;h^2$$

Let us rearrange the above equation as,

$$\left(1 - \frac{h}{2} f_i\right) y_{i-1} + \left(-2 + g_i\,h^2\right) y_i + \left(1 + \frac{h}{2} f_i\right) y_{i+1} = r_i\;h^2 \quad \text{... (8.7.5)}$$

Here $i = 1, 2, \ldots\ldots\ldots n-1$ and

$y_0 = a, \quad y_n = b$

We get 'n' simultaneous linear equations, which can be solved to get value of y.

Ex. 8.7.1 *Solve* $\frac{d^2y}{dx^2} = y$ *with* $y(0) = 0$ *and* $y(2) = 3.627$.

Sol. : The given equation is $\frac{d^2y}{dx^2} = y$. This can be written as,

$$y''_i = y_i$$

Putting the finite differences approximation of y''_i we get,

$$\frac{y_{i-1} - 2\,y_i + y_{i+1}}{h^2} = y_i \quad \text{... (8.7.6)}$$

The given initial conditions are,

$y(0)=0$ i.e. $x_0=0$, $y_0=0$

and $y(2)=3.627$ i.e. $x_n=2$, $y_n=3.627$

Let us divide x_0 to x_n into four equal parts. Hence $n=4$.

$$\therefore \quad h = \frac{x_n - x_0}{n}$$

$$= \frac{2-0}{4} = 0.5$$

and $i=1,2,3.$

Hence equation 8.7.6 becomes,

$$\left.\begin{aligned} i=1 &\Rightarrow \frac{y_0 - 2y_1 + y_2}{(0.5)^2} = y_1 \\ i=2 &\Rightarrow \frac{y_1 - 2y_2 + y_3}{(0.5)^2} = y_2 \\ i=3 &\Rightarrow \frac{y_2 - 2y_3 + y_4}{(0.5)^2} = y_3 \end{aligned}\right\} \quad \text{... (8.7.7)}$$

From boundary conditions,

$$y_0 = 0 \quad \text{and} \quad y_n = y_4 = 3.627$$

Putting these values in equation 8.7.7,

$$\frac{0 - 2y_1 + y_2}{(0.5)^2} = y_1$$

$$\frac{y_1 - 2y_2 + y_3}{(0.5)^2} = y_2$$

$$\frac{y_2 - 2y_3 + 3.627}{(0.5)^2} = y_3$$

Simplifying the above equations,

$$-9y_1 + 4y_2 = 0$$

$$4y_1 - 9y_2 + 4y_3 = 0$$

$$4y_2 - 9y_3 = -14.508$$

Solving the above linear equations,

$$y_1 = 0.526367$$

$$y_2 = 1.184327$$

$$y_3 = 2.138367$$

Thus we obtained the solution of boundary value problem at different values o x. The results are tabulated below :

i	$x_i = x_0 + i\,h$	y_i
–	$x_0 = 0$ (Given)	$y_0 = 0$ (Given)
1	$x_1 = x_0 + 1(h)$ $= 0 + 1\,(0.5) = 0.5$	$y_1 = 0.526367$
2	$x_2 = x_0 + 2(h)$ $= 0 + 2\,(0.5) = 1.0$	$y_2 = 1.184327$
3	$x_3 = x_0 + 3\,(h)$ $= 0 + 3\,(0.5) = 1.5$	$y_3 = 2.138367$
–	$x_4 = 2$ (Given)	$y_4 = 3.627$ (Given)

8.7.2 Eigen-Value Problems

Let A be the square matrix of order n. Then,

$$AX = \lambda X \qquad \text{... (8.7.8)}$$

Here λ is called the eigen value and X is the corresponding eigen-vector. Above matrix equation represents a set of homogeneous linear equations. i.e.

$$AX - \lambda X = 0$$

i.e.

$$\{A - \lambda I\}X = 0$$

The above equation can also be written as,

$$\left.\begin{aligned} (a_{11} - \lambda)\, x_1 + a_{12}\, x_2 + \ldots\ldots\ldots\ldots + a_{1n}\, x_n &= 0 \\ a_{21}\, x_1 + (a_{22} - \lambda)\, x_2 + \ldots\ldots\ldots\ldots + a_{2n}\, x_n &= 0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ a_{n1}\, x_1 + a_{n2}\, x_2 + \ldots\ldots\ldots + (a_{nn} - \lambda)\, x_n &= 0 \end{aligned}\right\} \qquad \text{... (8.7.9)}$$

The determinant $|A - \lambda I|$ yields a polynomial in λ. Roots of this polynomial are the eigen values. For every eigen value, there is a eigen vector.

Ex. 8.7.2 *Find the eigen-values and eigen-vectors of the matrix,*

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

Sol. : The determinant $|A - \lambda I|$ becomes,

$$\begin{vmatrix} 5-\lambda & 0 & 1 \\ 0 & -2-\lambda & 0 \\ 1 & 0 & 5-\lambda \end{vmatrix} = 0$$

Solving the above determinant for λ,

$$\lambda = -2, 4, 6$$

These are the eigen values. Now let us find eigen vector for $\lambda = 4$. We know that

$$AX = \lambda X$$

Putting values in above equation,

$$\begin{bmatrix} 5 & 0 & 1 \\ 0 & -2 & 0 \\ 1 & 0 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 4 \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

This gives,

$$x_1 + x_3 = 0$$
$$-6\, x_2 = 0$$

$\therefore\; x_1 = -\, x_3$ and $x_2 = 0$. Thus the eigen-vector is,

$$X = \begin{bmatrix} x_1 \\ 0 \\ -x_1 \end{bmatrix}$$

Exercise

1. Solve $y''+y+1=0$ with boundary conditions $y(0)=0$ and $y(1)=0$ at $x=0.5$.

Ans. : $y\,(0.5) = \dfrac{1}{7}$ for $n=2$ and,

$$y\,(0.5) = \frac{63}{449} \text{ for } n=4$$

University Questions

1. Find the layout Eigen value and Eigen vector of the following matrix.

$$\begin{bmatrix} 1 & 3 & -1 \\ 3 & 2 & 4 \\ -1 & 4 & 10 \end{bmatrix}$$

[May - 2003]

8.8 Engineering Applications

Now let's see an application based on solution of differential equation. Normally a current through an inductor or voltage across capacitor exhibit exponential nature. Consider a simple RC circuit shown in Fig. 8.8.1.

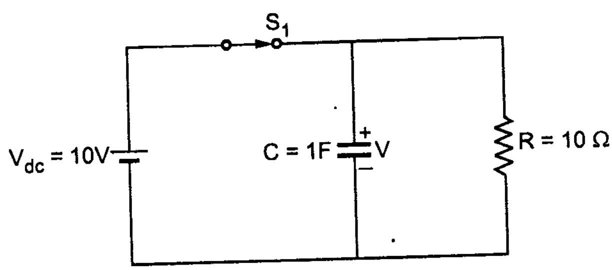


Fig. 8.8.1 Simple RC circuit with a dc source

In this circuit switch S_1 is closed as shown in figure. Let's say that switch S_1 is closed for a long period. The capacitor will be charged to the voltage equal to V_{dc}. Let's denote the voltage across capacitor as '$v\,(t)$'.

$\therefore \qquad v(t) = V_{dc}$

i.e. $\qquad v(t) = 10$ V (i.e. capacitor is charged fully)

The voltage source V_{dc} is also connected in parallel across the resistor of 10 Ω.

Now let's say that the switch S_1 is opened. When the switch is opened, the circuit will be disconnected from V_{dc}. This circuit is shown below in Fig. 8.8.2.

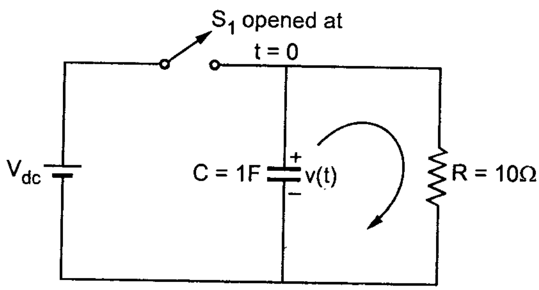


Fig. 8.8.2 Circuit with switch opened

Let's call the time at which the switch is opened as t = 0. Note that this is totally arbitrary time. We are assuming it for convenience.

Initial conditions :

At the instant when switch is opened (i.e. t = 0) the capacitor is charged to V_{dc}. i.e.,

$$\text{Voltage across capacior } v(t) = V_{dc} = 10\,V \quad \text{at } t \leq 0$$

Thus at $\boxed{t = 0, \quad v(t) = 10\,V}$

These are the *initial conditions* in the circuit.

After opening the switch the capacitor will then start discharging through the resistor R. Because of this discharging, the voltage $v(t)$ of the capacitor will go on reducing.

Relation of v (t) with t after switch is opened :

The circuit of Fig. 8.8.2 is redrawn below in Fig. 8.8.3 for better understanding. i.e.,

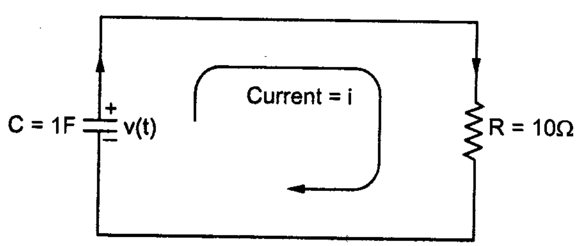


Fig. 8.8.3 Analysis of RC circuit

Apply Kirchhoff's voltage law to the above circuit. i.e,

$$v(t) = iR \qquad (\because \text{ drop across R is } = iR) \qquad \text{... (8.8.1)}$$

Here 'i' is the current flowing through R and C. The capacitor current is flowing from negative to positive terminal of the capacitor. Hence it will be treated as negative capacitor current. i.e.,

$$i = -C\frac{dv(t)}{dt} \quad \text{(By property of capacitors)} \qquad \text{... (8.8.2)}$$

Here $v(t)$ is the voltage across capacitor and C is value of capacitor in farads. From equation 8.8.1 we have,

$$\therefore \quad i = \frac{v(t)}{R} \qquad \text{... (8.8.3)}$$

Putting this value of 'i' in equation 8.8.2 we get,

$$\frac{v(t)}{R} = -C\frac{dv(t)}{dt}$$

Rearranging this equation we have,

$$\boxed{\frac{dv(t)}{dt} = -\frac{v(t)}{RC}} \qquad \text{... (8.8.4)}$$

Observe that this is an ordinary differential equation of first order. The initial conditions on this differential equation are t = 0, v (t) = 10 V as we have established earlier.

Now let's solve the above equations.

We can write equation 8.8.4 as,

$$\frac{dv(t)}{v(t)} = -\frac{dt}{RC} \qquad \text{... (8.8.5)}$$

Let's integrate both the sides of this equation from the instant when switch is opened to source time 't'.

i.e. $$\int_{t_0}^{t}\frac{dv(t)}{v(t)} = -\int_{t_0}^{t}\frac{dt}{RC}$$

$$\therefore \quad [\ln v(t)]_{t_0}^{t} = -\frac{1}{RC}[t]_{t_0}^{t}$$

t_0 is an instant when switch is opened. This instant we have called t = 0.

$$\therefore \quad t_0 = \text{instant when switch is opened}$$

$$\therefore \quad t_0 = 0$$

Then above equation becomes,

$$[\ln v(t)]_0^t = -\frac{1}{RC}[t]_0^t$$

$$\therefore \quad \ln v(t) - \ln v(0) = -\frac{1}{RC}(t-0)$$

$$\therefore \quad ln\left[\frac{v(t)}{v(0)}\right] = -\frac{1}{RC} \times t \qquad \because \quad ln(a) - ln(b) = ln\left(\frac{a}{b}\right)$$

By taking inverse logarithm above,

$$\frac{v(t)}{v(0)} = e^{-\frac{t}{RC}} \qquad \because \text{ If } ln(a) = b \text{ then } a = e^{b}$$

$$\therefore \quad \boxed{v(t) = v(0)\, e^{-\frac{t}{RC}}} \qquad \text{... (8.8.6)}$$

This is the solution of differential equation given by equation 8.8.4. Here, $v(0)$ is the voltage $v(t)$ at t = 0.

We know that at t = 0, v (t) = 10 V

$$\therefore \quad v(0) = 10\,V$$

Putting this value in equation 8.8.6 we get,

$$v(t) = 10\, e^{-\frac{t}{RC}} \qquad \text{... (8.8.7)}$$

To obtain voltage on capacitor after 10 seconds :

We have to find v (t) at t = 10 seconds.

We have R = 10 Ω, C = 1F. Putting these values in equation 8.8.7, we have,

$$v(10) = 10\, e^{-\frac{10}{10 \times 1}}$$

$$= 3.6787944$$

Thus $\quad v(10) = 3.6787944\ V \qquad$ ... (8.8.8)

This answer we have obtained using analytical method. Now let's see how numerical analysis can be used here.

To find voltage on capacitor using Euler's method :

We will see how Euler's method can be used to obtain voltage on capacitor after 10 seconds. The differential equation of the RC circuit we have obtained in equation 8.8.4. i.e.,

$$\frac{dv(t)}{dt} = -\frac{v(t)}{RC}$$

Let's represent $v(t)$ by v simply to avoid much more notations. Then we have above equation as,

$$\boxed{\frac{dv}{dt} = -\frac{v}{RC}} \qquad \text{... (8.8.9)}$$

This is a differential equation in v and at t = 0, v = 10, we have to find v at t = 10.

Let's take h = 1 sec. We can write above conditions as,

$$t_0 = 0, \quad v_0 = 10$$ From equation 8.8.9,

$$\frac{dv}{dt} = f(t, v) \qquad \text{... (8.8.10)}$$

Here $$f(t, v) = -\frac{v}{RC} \qquad \text{... (8.8.11)}$$

Putting R = 10, & C = 1 which are constants,

$$f(t, v) = -\frac{v}{10 \times 1}$$

$$\therefore \quad f(t, v) = -0.1\, v \qquad \text{... (8.8.12)}$$

Euler's equation is,

$$y_{n+1} = y_n + h\, f(x_n, y_n)$$

Here y is v and x is t. Therefore Eulers formula will be,

$$v_{n+1} = v_n + h\, f(t_n, v_n) \qquad \text{... (8.8.13)}$$

n = 0 in equation 8.8.13 :

$$\begin{aligned} v_1 &= v_0 + h\, f(t_0, v_0) \\ &= 10 + 1\,(-0.1\, v_0) \\ &= 10 + 1\,(-0.1 \times 10) \\ &= 10 - 1 \\ &= 9 \text{ V} \end{aligned}$$

& $$t_1 = t_0 + h = 0 + 1 = 1 \text{ sec}$$

$\therefore$ At $t = 1$ sec v = 9 V

n = 1 in equation 8.8.13

$$\begin{aligned} v_2 &= v_1 + h\, f(t_1, v_1) \\ &= 9 + 1(-0.1\, v_1) \\ &= 9 + 1\,(-0.1 \times 9) \\ &= 9 - 0.9 \\ &= 8.1 \end{aligned}$$

& $$t_2 = t_1 + h = 1 + 1 = 2$$

$\therefore$ At $t = 2$ sec, v = 8.1 V

n = 2 in equation 8.8.13

$$\begin{aligned} v_3 &= v_2 + h\, f(t_2, v_2) \\ &= 8.1 + 1\,(-0.1\, v_2) \\ &= 8.1 + 1\,(-0.1 \times 8.1) \\ &= 8.1 - 0.81 \\ &= 7.29 \end{aligned}$$

& $$t_3 = t_2 + h = 2 + 1 = 3$$

$\therefore$ At $t = 3$ sec, $v = 7.29$ V

n = 3 in equation 8.8.13,

$$\begin{aligned} v_4 &= v_3 + h\,f(t_3, v_3) \\ &= 7.29 + 1(-0.1\,v_3) \\ &= 7.29 + 1\,(-0.1 \times 7.29) \\ &= 7.29 - 0.729 \\ &= 6.561 \end{aligned}$$

& $t_4 = t_3 + h = 3 + 1 = 4$

$\therefore$ At $t = 4$ sec, $v = 6.561$ V

Similarly we can obtain values at t = 5, 6, 7, 8, 9, & 10, just go on putting n = 4, 5, 6, 7, 8 & 9 in equation 8.8.13.

At $\boxed{t = 10 \quad \text{we get } v = 3.48678 \text{ V}}$

All the values of t & v are tabulated below.

Table 8.8.1 Analysis of RC circuit using Euler's method

t	v	t	v
0	10	6	5.31441
1	9	7	7.82969
2	8.1	8	4.3046721
3	7.29	9	3.8742049
4	6.561	10	3.48678
5	5.9049		

Thus we have results as,

Method	t	v
Analytical	10	3.6787944
Euler (numerical)	10	3.48678

Thus using numerical methods we can obtain voltage on the capacitor which is very much close to its actual value. The error in Euler's method answer can be reduced by reducing h.

Ex. 8.8.1 *A resistance of 100 ohms and inductance of 0.5 H are connected in series with a battery of 15 V. If initially current is zero, find the current flowing in the circuit at time t = 0.001 and t = 0.01 using,*

(i) Runge-Kutta 4^{th} order method (ii) Euler's method

Compare the results with actual values. Comment on step size.

Sol. : Fig. 8.8.4 shows the RL circuit. In this circuit we have to find 'i'. The initial conditions are,

At $\boxed{t_0 = 0, \quad i_0 = 0}$ i.e. Current is zero initially.

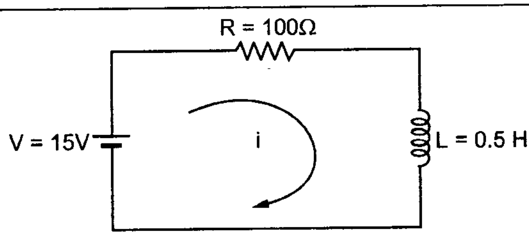


Fig. 8.8.4 RL circuit

To formulate the differential equation :

We know that the drop across the inductance is $L\,\frac{di}{dt}$. Applying KVL to the RL circuit of Fig. 8.8.4,

$$V = Ri + L\frac{di}{dt}$$

i.e.

$$\frac{di}{dt} + \frac{R}{L}i = \frac{V}{L} \qquad \ldots (8.8.14)$$

This is the differential equation of first order.

To solve differential equation analytically :

To obtain the actual values, we will solve the differential equation analytically. Hence we have to obtain 'i' as a function of 't'. Consider the linear differential equation of equation 8.8.14 above. i.e.,

$$\frac{di}{dt} + \frac{R}{L}i = \frac{V}{L}$$

The integrating factor for this equation is,

Integrating factor = $e^{\frac{R}{L}t}$

Multiplying both the sides of differential equation by Integrating factor,

$$\frac{di}{dt}\, e^{\frac{R}{L}t} + \frac{R}{L}\, i\, e^{\frac{R}{L}t} = \frac{V}{L}\, e^{\frac{R}{L}t} \qquad \ldots (8.8.15)$$

Here observe that,

$$\frac{d}{dt}\left\{ i\, e^{\frac{R}{L}t} \right\} = \frac{di}{dt}\, e^{\frac{R}{L}t} + i\,\frac{R}{L}\, e^{\frac{R}{L}t}$$

Thus LHS of equation 8.8.15 is derivative of $i\, e^{\frac{R}{L}t}$. Hence we can write,

$$\frac{d}{dt}\left\{i\,e^{\frac{R}{L}t}\right\} = \frac{V}{L}\,e^{\frac{R}{L}t}$$

$$d\left\{i\,e^{\frac{R}{L}t}\right\} = \left\{\frac{V}{L}\,e^{\frac{R}{L}t}\right\}dt$$

Integrating both sides of above equation,

$$\int d\left\{i\,e^{\frac{R}{L}t}\right\} = \int \frac{V}{L}\,e^{\frac{R}{L}t}\,dt + C$$

Here 'C' is integration constant. Above equation becomes,

$$i\,e^{\frac{R}{L}t} = \frac{V}{L}\int e^{\frac{R}{L}t} + C$$

$$= \frac{V}{L}\,\frac{e^{\frac{R}{L}t}}{\frac{R}{L}} + C$$

$$= \frac{V}{R}\,e^{\frac{R}{L}t} + C$$

$$\therefore \qquad i = \frac{V}{R} + C\,e^{-\frac{R}{L}t} \qquad \ldots (8.8.16)$$

We know that initially when $t = 0$ current is zero. i.e.,

$$0 = \frac{V}{R} + C\,e^{0}$$

$$C = -\frac{V}{R}$$

Hence equation 8.8.16 becomes,

$$i = \frac{V}{R} - \frac{V}{R}\,e^{-\frac{R}{L}t}$$

or

$$i = \frac{V}{R}\left(1 - e^{-\frac{R}{L}t}\right) \qquad \ldots (8.8.17)$$

This is the equation for current in the RL circuit. Putting the values in above equation,

$$i = \frac{15}{100}\left(1 - e^{-\frac{100}{0.5}t}\right)$$

$$i = \frac{3}{20}\left(1 - e^{-200t}\right) \qquad \ldots (8.8.18)$$

To obtain actual values of current :

By putting $t=0.001$ and $t=0.01$ in equation 8.8.18 we can get actual values of current in the circuit. i.e.,

$$i\,(0.001) = \frac{3}{20}\left(1-e^{-200\times 0.001}\right)$$

$$= 27.19 \text{ mA} \qquad \therefore \boxed{i\,(0.001) = 27.19\ mA}$$

$$i\,(0.01) = \frac{3}{20}\left(1-e^{-200\times 0.01}\right)$$

$$= 129.7 \text{ mA} \qquad \therefore \boxed{i\,(0.01) = 129.7\ mA}$$

(i) To obtain current using Runge-Kutta method

The differential equation is given by equation 8.8.14 as,

$$\frac{di}{dt}+\frac{R}{L}i = \frac{V}{L}$$

i.e.

$$\frac{di}{dt} = -\frac{R}{L}i+\frac{V}{L}$$

Putting the values in above equation,

$$\frac{di}{dt} = -\frac{100}{0.5}i+\frac{15}{0.5}$$

$$\left.\begin{aligned}\therefore \quad \frac{di}{dt} &= -200\,i+30\\ \text{or} \quad f\,(t,i) &= -200\,i+30\end{aligned}\right\} \qquad \ldots (8.8.19)$$

To obtain current at $t=0.001$, take $h=0.001$:

Here we have initial conditions as,

$t_0=0, \quad i_0=0$ and $h=0.001$.

Now we have to find i at $t=t_0+h=0+0.001=0.001$

Let us calculate k_1, k_2, k_3 and k_4 using equation 8.5.3 with $n=0$. i.e. (Here 'x' will be replaced by 't' and 'y' will be replaced by 'i'),

$$k_1 = f\,(x_0,y_0)=f\,(t_0,i_0)$$

$$= f\,(0,0) = -200\times 0+30$$

$$= 30 \qquad \therefore \boxed{k_1 = 30}$$

$$k_2 = f\left(t_0+\frac{h}{2},\ i_0+\frac{hk_1}{2}\right)$$

$$= f\left(0+\frac{0.001}{2},\ 0+\frac{0.001\times 30}{2}\right)$$

$$= f\,(0.005,\ 0.015) = -200\times 0.015+30$$

$$= 27 \qquad \therefore \boxed{k_2 = 27}$$

$$k_3 = f\left(t_0 + \frac{h}{2}, i_0 + \frac{hk_2}{2}\right)$$
$$= f\left(0 + \frac{0.001}{2}, \ 0 + \frac{0.001 \times 27}{2}\right)$$
$$= f\ (0.0005, \ 0.0135) = -200 \times 0.0135 + 30$$
$$= 27.3 \qquad \therefore \boxed{k_3 = 27.3}$$
$$k_4 = f\ (t_0 + h, \ i_0 + hk_3)$$
$$= f\ (0.001, \ 0 + 0.001 \times 27.3)$$
$$= f\ (0.001, \ 0.0273) = -200 \times 0.0273 + 30$$
$$= 24.54 \qquad \therefore \boxed{k_4 = 24.54}$$

The value of i at $t = 0.001$ can be obtained by putting $n = 0$ in first equation of equation 8.5.3. i.e., ('y' is replaced by i).

$$i_1 = i_0 + \frac{h}{6}\ (k_1 + 2k_2 + 2k_3 + k_4)$$
$$= 0 + \frac{0.001}{6}\ (30 + 2 \times 27 + 2 \times 27.3 + 24.54)$$
$$= 0.02719 \quad \text{or} \quad 27.19 \ \text{mA} \qquad \therefore \boxed{i(0.001) = 27.19\ mA}$$

To obtain current at t = 0.01, take h = 0.01 :

Here we have initial conditions as,

$t_0 = 0, \qquad i_0 = 0 \qquad$ and $\qquad h = 0.01$

Now we have to find i at $t = t_0 + h = 0 + 0.01 = 0.01$

Let us calculate k_1, k_2, k_3 and k_4 by putting $n = 0$ in equation 8.5.3. i.e.,

$$k_1 = f\ (t_0, i_0)$$
$$= f\ (0, 0) = -200 \times 0 + 30$$
$$= 30 \qquad \therefore \boxed{k_1 = 30}$$
$$k_2 = f\left(t_0 + \frac{h}{2}, i_0 + \frac{hk_1}{2}\right)$$
$$= f\left(0 + \frac{0.01}{2}, \ 0 + \frac{0.01 \times 30}{2}\right)$$
$$= f\ (0.005, \ 0.15) = -200 \times 0.15 + 30$$
$$= 0 \qquad \therefore \boxed{k_2 = 0}$$
$$k_3 = f\left(t_0 + \frac{h}{2}, i_0 + \frac{hk_2}{2}\right)$$
$$= f\left(0 + \frac{0.01}{2}, 0 + \frac{0.01 \times 0}{2}\right)$$
$$= f\ (0.005, \ 0) = -200 \times 0 + 30$$

$$= 30 \qquad \therefore \boxed{k_3 = 30}$$

$$k_4 = f\left(t_0 + h,\ i_0 + hk_3\right)$$

$$= f\,(0 + 0.01,\ 0 + 0.01 \times 30)$$

$$= f\,(0.01\,, 0.3) = -\,200 \times 0.3 + 30$$

$$= -\,30 \qquad \therefore \boxed{k_4 = -30}$$

The value of i at $t = 0.01$ can be obtained by putting $n = 0$ in first equation of equation 8.5.3. i.e.,

$$i_1 = i_0 + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

$$= 0 + \frac{0.01}{6}\,(30 + 2 \times 0 + 2 \times 30 - 30)$$

$$= 0.1 \text{ or } 100 \text{ mA} \qquad \therefore \boxed{i\,(0.01) = 100\ mA}$$

(ii) To obtain current using Euler's method

Here also we have to obtain 'i' at $t = 0.001$ and 0.01. Let us assume step size $h = 0.001$. Euler's equation is given as,

$$y_{n+1} = y_n + h\ f\,(x_n,\, y_n)$$

In this example 'y' is 'i' and 'x' is 't'. Hence above equation will be,

$$i_{n+1} = i_n + h\ f\,(t_n, i_n)$$

From equation 8.8.19, $f\,(t,i) = -\,200\ i + 30$, hence above equation becomes,

$$i_{n+1} = i_n + h\,(-\,200\ i_n + 30) \qquad \ldots (8.8.20)$$

Now let us iterate this equation.

With n = 0 in equation 8.8.20 :

$$i_1 = i_0 + h\,(-\,200\,i_0 + 30)$$

We have $i_0 = 0$ and $h = 0.001$, hence above equation becomes,

$$i_1 = 0 + 0.001\,(-\,200 \times 0 + 30)$$

$$= 0.03 \text{ or } 30 \text{ mA}$$

And $\quad t_1 = t_0 + h = 0 + 0.001 = 0.001$

Thus at $\quad t = 0.001, \quad i = 30\ mA \quad$ i.e. $\boxed{i\,(0.001) = 30\ mA}$

With n = 1 in equation 8.8.20

$$i_2 = i_1 + h\,(-200\ i_1 + 30)$$

$$= 0.03 + 0.001\ (-\,200 \times 0.03 + 30)$$

$$= 0.054 \text{ or } 54 \text{ mA}$$

Thus at $t_2 = t_1 + h = 0.001 + 0.001 = 0.002$, $i_2 = 0.054$ other values can be calculated on the similar lines. Table 8.8.2 shows the results of these calculations.

Table 8.8.2 : Analysis of RL circuit using Euler's method :

t	I
0.001	0.03
0.002	0.054
0.003	0.0732
0.004	0.08856
0.005	0.100848
0.006	0.1106784
0.007	0.1185427
0.008	0.1248341
0.009	0.1298673
0.01	0.1338938

Thus from above table,

at $t = 0.01,\ i = 0.1338938$ or 133.8938 mA i.e. $i\ (0.01) = 133.893$ mA

Results :

The results of actual, Runge-kutta and Euler's methods are given below in table 8.8.3.

Table 8.8.3 : Results of RL circuit

t	Actual current (mA)	Runge-kutta method current (mA)	Euler's method current (mA)
0.001	27.19	27.19 for h = 0.001	30 for h = 0.001
0.01	129.7	100 for h = 0.01	133.8938 for h = 0.001

Comments on Results :

1. For small step size, the results of Runge-kutta method are accurate. i.e. $i\ (0.001) = 27.19$ mA for step size of 0.001.
2. For larger step size, the error increases in Runge kutta method.
3. For Euler's method, the step size is selected as 0.001 for both the currents. Hence error is reduced in $i\ (0.01)$. That is, small step size reduces error in Euler's method.

Ex. 8.8.2 *Consider the RC circuit shown in Fig. 8.8.5. At $t = 0^-$ just before the switch is closed, $v_c = 100$ V. Obtain the current and charge at time 10 msec using RK 4th order. Also write the algorithm for the same.* **[Dec - 2004, 10 marks]**

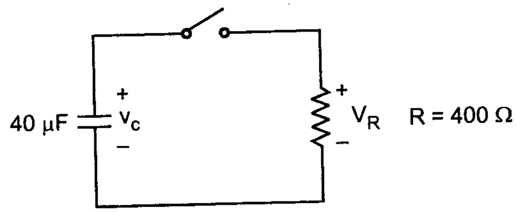


Fig. 8.8.5 RC circuit of Ex. 8.8.2

Sol. : When the switch is closed at t = 0, the capacitor will start discharging and current will flow through the circuit as shown below. The initial conditions are $t_0 = 0$,

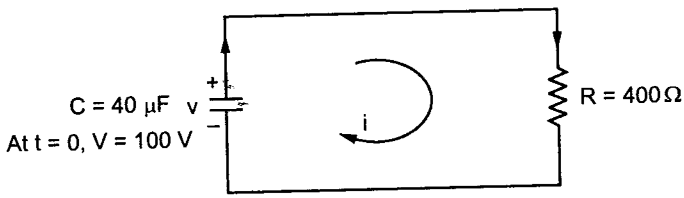


Fig. 8.8.6 RC circuit of Ex. 8.8.2

$v_0 = 100$ V. We have to determine voltage on the capacitor at 10 msec. Similar circuit is analyzed in Fig. 8.8.3. The differential equation is given as,

$$\frac{dv}{dt} = f(t,v) = -\frac{v}{RC}$$

Putting the values of R and C,

$$f(t,v) = -\frac{v}{400 \times 40 \times 10^{-6}}$$

$$\therefore \quad f(t,v) = -62.5\,v$$

Let us take h = 0.005. Hence v can be determined in two steps.

Step 1 : $h = 0.005, \quad t_0 = 0, \quad v_0 = 100$

$t_1 = t_0 + h = 0.005$, we have to obtain v_1.

$$\therefore \quad k_1 = f(t_0, v_0) = -6250$$

$$k_2 = f\left(t_0 + \frac{h}{2}, v_0 + \frac{hk_1}{2}\right) = -5273.437$$

$$k_3 = f\left(t_0 + \frac{h}{2}, v_0 + \frac{hk_2}{2}\right) = -5426.025$$

$$k_4 = f(t_0 + h, v_0 + hk_3) = -4554.367$$

$$\therefore \quad v_1 = v_0 + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 73.164$$

Step 2 : For this step we have,

$$t_1 = 0.005, \qquad v_1 = 73.164$$

$$h = 0.005 \text{ and } t_2 = t_1 + h = 0.005 + 0.005 = 0.01$$

$$\therefore \quad k_1 = f(t_1, v_1) = -4572.745$$

$$k_2 = f\left(t_1 + \frac{h}{2}, v_1 + \frac{hk_1}{2}\right) = -3858.253$$

$$k_3 = f\left(t_1 + \frac{h}{2}, v_1 + \frac{hk_2}{2}\right) = -3969.893$$

$$k_4 = f(t_1 + h, v_1 + hk_3) = -3332.153$$

$$\therefore \quad v_2 = v_1 + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 53.529 \text{ v}$$

Thus at $\quad t = 10$ msec, $\quad v_c = 53.529$ volts

To obtain the current and charge

Current is given as,

$$i = \frac{v_c}{R}$$

$$= \frac{53.529}{400} = 0.1338 \text{ A or } 133.8 \text{ mA}$$

Charge on the capacitor is given as,

$$Q = C v_c$$

$$= 40 \times 10^{-6} \times 53.529 = 0.00214 \text{ C}$$

University Questions

1. For the above circuit, when the switch SW_1 is closed at time t = 0 sec, the current in the circuit is zero.

   Evaluate with the help of a suitable numerical method,the charge Q on the capacitor after 0.5 m sec and 1.0 m sec. **[Dec - 96, Dec - 99]**

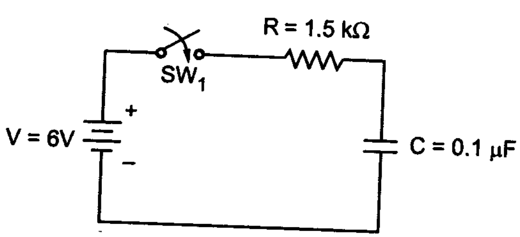


Fig. 8.8.7

2. Develop a C program for the above method to output the charge Q at anytime. t.
[Dec - 96, Dec - 99]

3. At t = 0, switch is closed $V_c(0) = 0$.
Find $V_c(t)$ at t = 2 ms. Use RK 4^{th} order method with h = 0.5. [Dec - 2002]

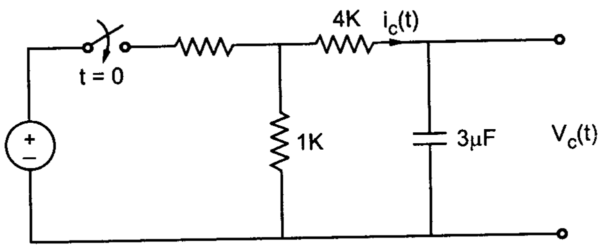


Fig. 8.8.9

4. At t = 0, just before the switch is closed in Fig. 8.8.9 V_c = 100 V. Obtain the current and charge at time 10 msc using RK 4th order. Also write the algorithm for the same.
[Dec - 2004]

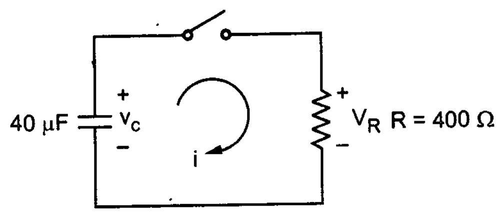


Fig. 8.8.10

8.9 MATLAB for Ordinary Differential Equations

MATLAB can also be used to solve differential equations. Following program uses MATLAB's ode45 function to solve differential equations.

```
% Download this program from www.vtubooks.com
% Solution of a differential equation. MatLab Version 6

%       This Matlab program calculates the solution of the differential
%       equation using ode function avialable in Matlab.

%       Consider the differential equation,

%               y" + 4y' + y = 0

%       Let this equation be written as,

%               y(1)" + 4*y(1)' + y(1) = 0

%       And let y(1)' = y(2) then above equation will be written as,

%               y(2)' + 4*y(2) + y(1) = 0;
%       i.e.    y(2)' = -4*y(2) - y(1)
```

```
     The ODE file is an m-file labelled F.m which describes the
                                         system to be solved.
     The file F.m is present in this directory. For the above
     function the ODE file is written as follows,

              function dy = F(t,y)
              dy = [y(2); -4*y(2)-y(1)];

     Other functions can be written in the similer fashion.

%---------- Next part calculates the values of y(1) and y(2) in the
%   interval[0 20].     The initial value of y(1) = 1
%                       and initial value of y(2) = 0   ------------

clc;
disp('                  Solution of a differential equation using ode function');
disp(' ');
disp('The function to be solved is');
type F                         %the function is displayed for reference

[T,Y] = ode45('F',[0 20],[1;0]);
%                              ode45 is used for solving differential equation
%                                    T is time vector and Y is solution vector

%------- Next part displays and plots y(1) and y(2) ---------------------

disp('The values of y(1) and y(2) are as follows...');
disp(Y);         %y(1) and y(2) are displayed on the screen
plot(Y)          %y(10 and y(2) are plotted on the screen
legend('Blue : y1','Green : y2');
xlabel('time t');
ylabel('solution y');
title('Solution of differential equation');

%---------------------- end of program -----------------------------
```

Consider the statement in the above program,

`[T,Y]=ode45('F',[020],[1;0]);`

This statement solves the differential equation using `ode45` function of MATLAB. Here `F` is the file which contains the function to be integrated. The contents of `F.m` are as follows ;

```
%------------------------- F.m -------------------------------------
% file name : F.m
% This file contains a function for differentialEq.m
function dy = F(t,y)
dy = [y(2); -4*y(2)-y(1)];
%-------------------------------------------------------------------
```

The differential equation is $y''+4\,y'+y=0$. The above function file is written as per the requirement of `ode45` function. The `ode45` function has following format :

`[T,Y]=ode45(odefun,Tspan,y0)`

Here `odefun` is the function to be integrated, it is written in file `F.m`

`Tspan` is value of independent variable over which function is integrated. In this example function is integrated over `[0  20]`, i.e. from 0 to 20.

`y0` represents initial conditions of the function. In this example `[1;0]` means $y=1$ and $y'=0$ at $t=0$.

Next part of the program displays values and plots of y and y' with respect to t.

The results of this program are as follows :

```
%---------------------------- Results -----------------------------------
                        Solution of a differential equation using ode function

The function to be solved is

function dy = F(t,y)
dy = [y(2); -4*y(2)-y(1)];

The values of y(1) and y(2) are as follows...
    1.0000         0
    1.0000   -0.0001
    1.0000   -0.0001
    1.0000   -0.0002
    1.0000   -0.0002
    1.0000   -0.0005
    1.0000   -0.0000
      .         .
      .         .
      .         .
    0.0074   -0.0020
    0.0069   -0.0018
    0.0066   -0.0018
    0.0062   -0.0017
    0.0059   -0.0016
    0.0057   -0.0015
    0.0055   -0.0015
    0.0054   -0.0014
    0.0052   -0.0014
    0.0051   -0.0014
```

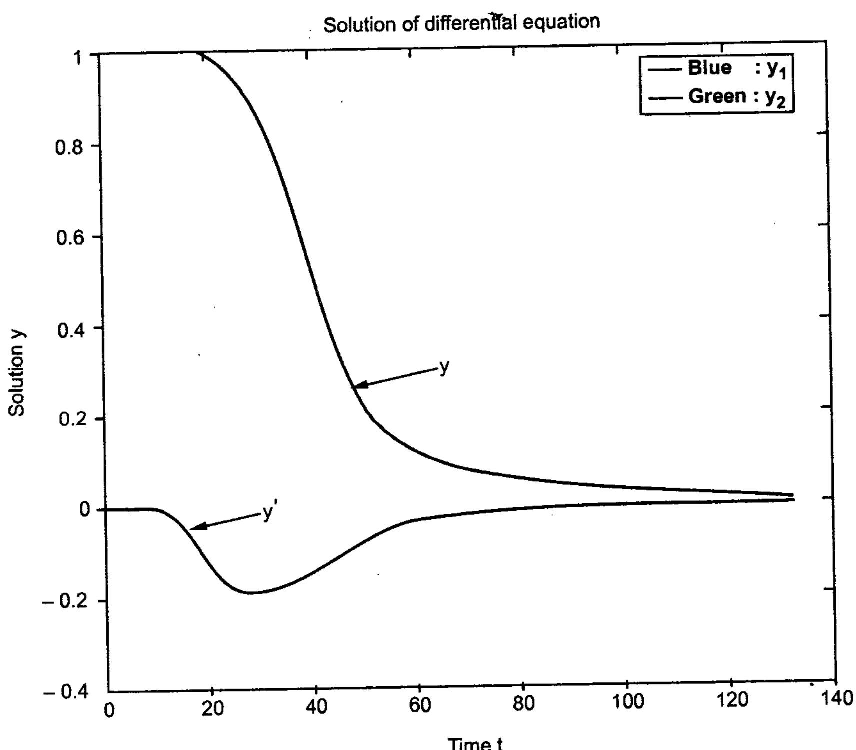


Fig. 8.9.1 Solution of the differential equation $y''+4y'+y=0$. Plots of y and y'

The above results lists some of the values of y and y'. All values can be seen by actually running the program. Fig. 8.9.1 shows the plots of y and y'.

Computer Exercise

1. Write a C program to implement second order Runge kutta method.
2. Write a C program of Euler's method which will accept a variable step size h.
3. Write a C program for predictor and corrector methods such as Milne's method.
4. Write a C program for modified Euler's method. Your program should accept allowed error in the solution and it should calculate value of y according to the permitted error.

❑❑❑

9
Optimization

9.1 Introduction

Optimization is one of the major area in engineering. In almost all the engineering applications optimization is required. Optimization broadly deals with finding the best possible result or optimum solution under the given constraints. For example we always desire an aircraft having minimum weight and maximum strength. Hence during the design of an aircraft efforts are made for maximum strength. The constraint in this design will be weight of an air craft. Because weight cannot be increased above certain limit.

Next, consider an example of satellite. Efforts are made to design a satellite which will handle large number of channels and provides broad coverage area. But the satellite have limited solar power from its solar panels. This becomes a constraint on the design of satellite. Hence optimum solution is required which will balance available solar power as well as gain and bandwidth capabilities of the satellite. There are large number of such examples like audio amplifier with maximum power output and minimum power dissipation, power supply with good regulation and minimum output impedance etc. All such problems can be mathematically formulated and solved using optimization techniques. An optimization problem can be mathematically represented as follows :

Determine value of x, which minimizes or maximizes $f(x)$ subject to following constraints :

$$d_i(x) \leq a_i \quad , \qquad i = 1, 2, \ldots . m$$
$$e_i(x) = b_i \cdot \quad , \qquad i = 1, 2, \ldots .. p$$

This problem represents *constrained optimization.* If there are no constraints in the optimization problem, then it is called unconstrained optimization. For example to determine maxima or minima of a polynomial is unconstrained optimization. In this chapter we will study some of the techniques of constrained and unconstrained optimization.

9.2 One-dimensional Unconstrained Optimization

In one-dimensional optimization, the functions depend on a single dependent variable. Here we will discuss the techniques to find minimum or maximum (i.e. optimum) of a function of single variable. Such techniques are basically one dimensional unconstrained optimization techniques. These techniques are similar to root location techniques such as bisection, regula falsi ; newton raphson method etc. Optimization techniques can be bracketing or open. Here we will discuss golden section search and quadratic interpolation, which are bracketing methods. And we will discuss newton's method which is open method.

9.2.1 Golden Section Search

Fig. 9.2.1 shows a function $f(x)$ having one maximum in the interval $[x_l, x_u]$.

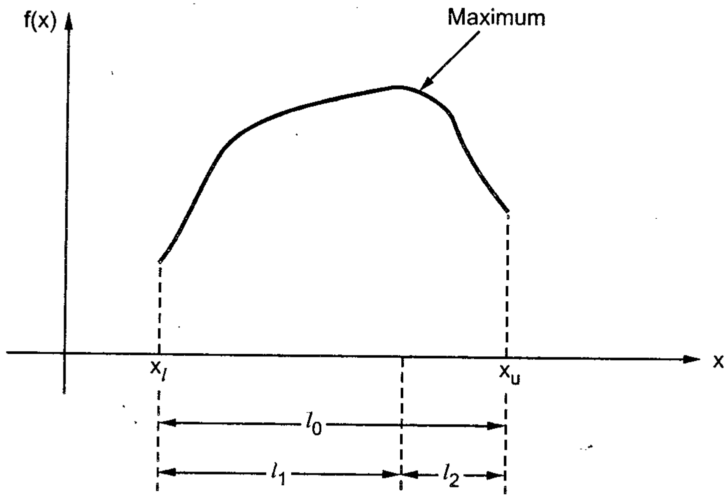


Fig. 9.2.1 Selection of interior points according to golden ratio

Golden ratio :

The concept of golden ratio is to be discussed first. In the above figure observe that,

$$l_0 = x_u - x_l$$

$$l_0 = l_1 + l_2 \quad \text{... (9.2.1)}$$

Further l_1 and l_2 are selected such that,

$$\frac{l_1}{l_0} = \frac{l_2}{l_1}$$

Putting for l_0 in above equation,

$$\frac{l_1}{l_1 + l_2} = \frac{l_2}{l_1}$$

Let $R = \frac{l_2}{l_1}$, then above equation becomes,

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

i.e. $$R^2 + R - 1 = 0$$

Solving the above equation for positive root,

$$R = \frac{-1 + \sqrt{1-4(-1)}}{2} = \frac{\sqrt{5}-1}{2} \quad \text{... (9.2.2)}$$

The above value of R is called *golden ratio.* It is used in golden section method to calculate maxima or minima efficiently.

How an optima is calculated ?

Now let us see how maximum is calculated using golden ratio. We know that maxima lies in the interval $[x_l, x_u]$. Then two additional points in this interval are calculated as follows :

$$d = R(x_u - x_l) = \frac{\sqrt{5}-1}{2}(x_u - x_l) \quad \text{... (9.2.3)}$$

Then, and

$$\left. \begin{aligned} x_1 &= x_l + d \\ x_2 &= x_u - d \end{aligned} \right\} \quad \text{... (9.2.4)}$$

Then if $f(x_2) > f(x_1)$, maxima lies in $[x_l, x_1]$ i.e. $x_u \leftarrow x_1$ and if $f(x_1) > f(x_2)$, maxima lies in $[x_2, x_u]$ i.e. $x_l \leftarrow x_2$. After establishing the new interval, new value of d is calculated and the algorithm repeats.

Ex. 9.2.1 *Using golden section search determine the maximum value of* $f(x) = 2\sin x - 0.1x^2$ *in the interval* $[0, 4]$.

(Dec-2002, 8 Marks, May-2004, 10 Marks, Dec-2004, 10 Marks)

Sol. : Here, $f(x) = 2\sin x - 0.1x^2$

And $[0, 4]$ means, $x_l = 0$, $x_u = 4$.

Iteration no. 1 : Interval is $x_l = 0$, $x_u = 4$ from equation 9.2.3, value of 'd' becomes,

$$d = \frac{\sqrt{5}-1}{2}(x_u - x_l)$$

Putting values of x_u and x_l,

$$d = \frac{\sqrt{5}-1}{2}(4-0) = 2.472136$$

From equation 9.2.4, x_1 and x_2 are given as,

$$x_1 = x_l + d = 0 + 2.472136 = 2.472136$$

$$x_2 = x_u - d = 4 - 2.472136 = 1.527864$$

Hence

$$\begin{aligned} f(x_1) &= f(2.472136) \\ &= 2\sin(2.472136) - 0.1(2.472136)^2 \\ &= 0.6299743 \end{aligned}$$

and $$f(x_2) = f(1.527864) = 2\sin(1.527864) - 0.1(1.527864)^2 = 1.7647203$$

Here $f(x_2) > f(x_1)$, hence maxima lies in $[x_l, x_1] = [0, 2.472136]$

Iteration No. 2 : New interval is $[0, 2.472136]$ hence $x_l = 0$, $x_u = 2.472136$. Here note that value of x_1 from previous iteration is assigned to x_u. Thus new interval is formed. Form equation 9.2.3, value of 'd' becomes,

$$d = \frac{\sqrt{5}-1}{2}(x_u - x_l)$$

Putting new values of x_u and x_l,

$$d = \frac{\sqrt{5}-1}{2}(2.472136 - 0) = 1.5278641$$

From equation 9.2.4, x_1 and x_2 are given as,

$$x_1 = x_l + d = 0 + 1.5278641 = 1.5278641$$

and $$x_2 = x_u - d = 2.472136 - 1.5278641 = 0.9442719$$

Hence $$f(x_1) = f(1.5278641) = 1.7647203$$

and $$f(x_2) = f(0.9442719) = 1.5309755$$

Here $f(x_1) > f(x_2)$, hence maxima lies in $[x_2, x_u] = [0.9442719, 2.472136]$

Iteration No. 3 : New interval is $[0.9442719, 2.472136]$. Hence $x_l = 0.9442719$ and $x_u = 2.472136$. From equation 9.2.3, value of 'd' becomes,

$$d = \frac{\sqrt{5}-1}{2}(x_u - x_l)$$

Putting new values of x_u and x_l,

$$d = \frac{\sqrt{5}-1}{2}(2.472136 - 0.9442719) = 0.9442719$$

From equation 9.2.4, x_1 and x_2 are given as,

$$x_1 = x_l + d = 0.9442719 + 0.9442719 = 1.8885439$$

and $$x_2 = x_u - d = 2.472136 - 0.9442719 = 1.5278641$$

Hence $$f(x_1) = f(1.8885439) = 1.5433233$$

and $$f(x_2) = f(1.5278641) = 1.767202$$

Here $f(x_2) > f(x_1)$, hence maxima lies in $[x_l, x_1] = [0.9442719, 1.8885439]$

Iteration No. 4 : New interval is $[0.9442719, 1.8885439]$. Hence $x_l = 0.9442719$ and $x_u = 1.8885439$.

$\therefore$ $$d = \frac{\sqrt{5}-1}{2}(1.8885439 - 0.9442719) = 0.5835921$$

$\therefore$ $$x_1 = x_l + d = 0.9442719 + 0.5835921 = 1.5278641$$

and $$x_2 = x_u - d = 1.8885439 - 0.5835921 = 1.3049517$$

Hence $f(x_1) = f(1.5278641) = 1.7647202$

and $f(x_2) = f(1.3049517) = 1.759452$

If we want to stop iterations here, then $f(x_1) > f(x_2)$, hence maxima occurs at $x_1 = 1.5278641$. And maximum values of $f(x)$ is 1.7647202. If we want more accurate value, then next iterations can be performed using the same procedure.

9.2.2 C-program and Algorithm for Golden Section Search

Algorithm :

Let us prepare an algorithm for golden section search.

Assumption : The function $f(x)$ whose maxima is to be determined is defined in the program.

Step 1 : Start

Step 2 : Get an interval $[x_l, x_u]$ in which the maxima lies.

Step 3 : Calculate

$$d = \frac{\sqrt{5}-1}{2}(x_u - x_l)$$

Step 4 : Calculate x_1 and x_2 such that

$$x_1 = x_l + d$$
$$x_2 = x_u - d$$

Step 5 : Evaluate $f(x)$ at x_1 and x_2. i.e. calculate $f(x_1)$ and $f(x_2)$

Step 6 : If $f(x_2) > f(x_1)$, $x_u \leftarrow x_1$ else $x_l \leftarrow x_2$

Step 7 : If the required accuracy is not achieved, then go to step 3.

Step 8 : x_1 is the maxima

Step 9 : Display the maxima and stop.

C-program :

The C-program based on above algorithm is given below :

```
/* Download this program from  www.vtubooks.com                           */
/* File name : gldn_src.cpp                                                */

/*----------- Golden Section Search Method ----------------------------*/

/*      This program calculates the maxima of the given function
        in the given interval. The function is defined in the program

                        f(x)  = 2*sin(x)-(x*x/10)

        INPUTS : Initial values of xl and xu.


        OUTPUTS: Location of the maxima                                    */

/*-----------------------------    PROGRAM  -------------------------*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>
```

```
void main()
{
    double f (double x);               /* DECLARATION OF A FUNCTION f */
    char ch;
    double xu,xl,x1,x2,d,fx2,fx1;

    clrscr();
    printf("\n\tGolden Section Search Method\n");

    printf("\n\nEnter xl = ");
    scanf("%lf",&xl);                        /* ENTER VALUE OF xl    */
    printf("\n\nEnter xu = ");
    scanf("%lf",&xu);                        /* ENTER VALUE OF yu    */

    printf("\nPress any key to see step by step display of results...\n"
           "press 'q' to stop\n\n\t  x\t\t\t  f(x)\n");

    while(ch !='q')       /* Loop to implement golden section search */
    {
       d = (sqrt(5)-1)*(xu-xl)/2; /* calculate d using golden ration */
       x1 = xl+d;                 /* calculation of x1         */
       x2 = xu-d;                 /* calculation of x2         */
       fx2 = f(x2);               /* calculation of f(x2)      */
       fx1 = f(x1);               /* calculation of f(x1)      */
       if(fx2>fx1) xu = x1;       /* maxima lies in xl, x2 x1 */
       else        xl = x2;       /* maxima lies in xu, x2 x1 */
       printf("\n\t%lf\t\t%lf",x1,fx1);
       ch = getch();
    }
}
/*-----------------------------------------------------------------------*/

double f ( double x)             /* FUNCTION TO CALCULATE VALUE OF f(x) */
{
    double fx;
    fx = 2*sin(x)-(x*x/10);
    return(fx);
}
/*----------------------- END OF PROGRAM ----------------------------*/
```

In the above program observe that values of x_l and x_u are accepted first. Note that these values contain only one maxima of $f(x)$. Then there is a `while` loop to implement golden section search. The first statement in `while` loop is,

```
d  =(sqrt(5)-1)*(xu-xl)/2;
```

This statement implements $d = \frac{\sqrt{5}-1}{2}\,(x_u - x_l)$.

Then next two statements are ;

```
x1  = xl+d;
x2  = xu-d;
```

These two statements calculate x_1 and x_2 within an interval $[x_l, x_u]$. Further `f(x2)` and `f(x1)` are calculated and new interval $[x_l, x_u]$ for the maxima is established. This `while` loop works till 'q' is pressed. There is no error check in the program. This program prints the values of x_1 and $f(x_1)$ on the screen.

To test this program :

Observe the function f at the end of the program. The statement in this function is,

```
fx=2*sin(x)-(x*x/10);
```

This statement implements $f(x) = 2\sin x - \frac{x^2}{10}$ some other function can be implemented by changing the above statement in the program. In example 9.2.1, we have obtained the maxima of this function in the interval [0, 4]. Run this program and give $x_l = 0$, $x_u = 4$ as inputs. Then the results generated by the program are given below :

```
---------------------------- Results ---------------------------------

        Golden Section Search Method


 Enter xl = 0

 Enter xu = 4

 Press any key to see step by step display of results...
 press 'q' to stop
              x                         f(x)

        2.472136                  0.629974
        1.527864                  1.764720
        1.888544                  1.543223
        1.527864                  1.764720
        1.665631                  1.713580
        1.527864                  1.764720
        1.442719                  1.775475
        1.475242                  1.773242
        1.442719                  1.775475
        1.422619                  1.775699
        1.430297                  1.775717
        1.435042                  1.775665
        1.430297                  1.775717
        1.427364                  1.775726
        1.428484                  1.775725
-----------------------------------------------------------------------
```

In the above results observe that value of x is 1.527864 in the fourth iteration. In example 9.2.1, $x_1 = 1.5278641$ in the fourth iteration, i.e. same value is obtained. Program prints value of $f(x)$ also.

9.2.3 Quadratic Interpolation

In this method a quadratic (i.e. 2^{nd} order) polynomial is interpolated near maximum. This quadratic polynomial passes through the three points which include maxima or minima. This quadratic polynomial is then differentiated and the result is set to zero. Then it is solved which gives maxima or minima. Let the three points be x_0, x_1 and x_2. A quadratic polynomial passes through these points. Then the maxima occurs at x_3, which is given as,

$$x_3 = \frac{f(x_0)\left(x_1^2 - x_2^2\right) + f(x_1)\left(x_2^2 - x_0^2\right) + f(x_2)\left(x_0^2 - x_1^2\right)}{2f(x_0)(x_1 - x_2) + 2f(x_1)(x_2 - x_0) + 2f(x_2)(x_0 - x_1)} \quad \text{... (9.2.5)}$$

Then three points are selected from x_0, x_1, x_2, x_3 for next iteration and the algorithm repeats.

Ex. 9.2.2 *Obtain the maxima of the function $f(x) = 2\sin x - 0.1x^2$ using quadratic interpolation. Take $x_0 = 0$, $x_1 = 1$ and $x_2 = 4$.*

Sol. : Iteration No. 1 : $x_0 = 0, x_1 = 1, x_2 = 4.$

Let us calculate,

$$f(x_0) = f(0) = 0$$
$$f(x_1) = f(1) = 1.582942$$

Putting values in equation 9.2.5.

$$f(x_2) = f(4) = -3.113605$$

$$x_3 = \frac{0\left(1^2 - 4^2\right) + 1.582942\left(4^2 - 0^2\right) - 3.113605\left(0^2 - 1^2\right)}{2(0)(1-4) + 2(1.582942)(4-0) + 2(-3.113605)(0-1)}$$

$= 1.505535$

Hence $f(x_3) = f(1.505535) = 1.7690789$

Here $f(x_3) > f(x_1)$, hence maxima lies in x_1, x_3 and x_2. Hence new three points become,

$x_0 = 1, \quad x_1 = 1.505535, \quad x_2 = 4$

Iteration No. 2 : $x_0 = 1, \quad x_1 = 1.505535, \quad x_2 = 4$

Hence
$$f(x_0) = f(1) = 1.582942$$
$$f(x_1) = f(1.505535) = 1.7690789$$
$$f(x_2) = f(4) = -3.113605$$

Putting these values in equation 9.2.5,

$$x_3 = \frac{1.582942\left(1.505535^2 - 4^2\right) + 1.7690789\left(4^2 - 1.582942^2\right) - 3.113605\left(1.582942^2 - 1.505535^2\right)}{2(1.582942)(1.505535-4) + 2(1.7690789)(4-1.582942) + 2(-3.113605)(1.582942-1.505535)}$$

$= 1.490253$

Here $f(x_3) = f(1.490253) = 1.7714309.$

From the values of $f(x_0)$, $f(x_1)$ and $f(x_3)$, it is clear that maxima lies in x_0, x_1 and x_3. Hence new three points become,

$x_0 = 1, \quad x_1 = 1.490253$ and $x_2 = 1.505535$

Iteration No. 3 : As per the above values of x_0, x_1 and x_2, we can calculate x_3 from equation 9.2.5. i.e.,

$$x_3 = 1.425636$$

And $f(x_3) = f(1.425636) = 1.7757217$, which is highest of the earlier values. Similar procedure can be repeated to get accurate value of maxima.

9.2.4 Newton's Method

This method is similar to newton raphson method for evaluating the root of $f(x)$. We know that the root by newton raphson method is given as,

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

The similar approach can be used to find the maxima or minima. A taylor series is written for $f(x)$. Then its derivative is set equal to zero. This gives following equation,

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)} \qquad \text{... (9.2.6)}$$

Ex. 9.2.3 *Using Newton's method determine the maximum of $f(x) = 2\sin x - 0.1x^2$. Take initial approximation as $x_0 = 2.5$.*

Sol. : Let us evaluate first and second derivatives of $f(x)$. i.e.,

$$f'(x) = 2\cos x - 0.2x$$

and $$f''(x) = -2\sin x - 0.2$$

Putting these values in equation 9.2.6,

$$x_{n+1} = x_n - \frac{2\cos x_n - 0.2x_n}{-2\sin x_n - 0.2} \qquad \text{... (9.2.7)}$$

Iteration No. 1 : With $n = 0$ in above equation,

$$x_1 = x_0 - \frac{2\cos x_0 - 0.2x_0}{-2\sin x_0 - 0.2}$$

Putting $x_0 = 2.5$ in above equation,

$$x_1 = 2.5 - \frac{2\cos(2.5) - 0.2(2.5)}{-2\sin(2.5) - 0.2} = 0.9950815$$

Iteration No. 2 : With $n = 1$ in equation 9.2.7,

$$x_2 = x_1 - \frac{2\cos x_1 - 0.2x_1}{-2\sin x_1 - 0.2}$$

Putting $x_1 = 0.9950815$ in above equation,

$$x_2 = 0.9950815 - \frac{2\cos(0.9950815) - 0.2(0.9950815)}{-2\sin(0.9950815) - 0.2}$$

$$= 1.4690107$$

Iteration No. 3 : With $n = 2$ in equation 9.2.7,

$$x_3 = x_2 - \frac{2\cos x_2 - 0.2x_2}{-2\sin x_2 - 0.2}$$

Putting $x_2 = 1.4690107$,

$$x_3 = 1.4690107 - \frac{2\cos(1.4690107) - 0.2\,(1.4690107)}{-2\sin(1.4690107) - 0.2}$$

$$= 1.4276423$$

Similarly next iterations can be performed to obtain more and more accurate value of maxima.

University Questions

1. Discuss the golden section search method for minimisation.

   [May - 98, Dec - 98, May - 99, Dec - 99, May - 2000, Dec - 2000, May - 2001]

2. Use the Golden-section search method to find the maximum of $2 \sin x - \frac{x^2}{10}$, with the interval $x_l = 0$ and $x_u = 4$. **(Dec.-2002)**
3. Write a short note on Golden section search. **(May-2003)**
4. Use the golden-section search method to find the maximum of $f(x) = 2\sin x - \frac{x^2}{10}$

   (May-2004, Dec - 2004)

9.3 Multidimensional Unconstrained Optimization

In this section we will discuss the methods to find maxima or minima of a function of two variables. These methods are classified depending upon the derivative evaluation. *Direct methods* do not require derivative evaluation. And *gradient methods* require derivatives. These methods are discussed next.

9.3.1 Direct Methods

9.3.1.1 Random Search

In this method, the function is repeatedly evaluated at the random points. A maxima or minima is selected finally from all the function values.

9.3.1.2 C-Program and Algorithm for Random Search

Now let us discuss the algorithm and C-program for random search.

Algorithm :

Assumption : The function of two variables $f(x,y)$ is predefined in the program.

Step 1 : Start and initialize registers

Step 2 : Accept the intervals $[x_l, x_u]$ and $[y_l, y_u]$ in which maxima or minima is to be located

Step 3 : Generate the x and y values randomly as follows :

$$x = x_l + (x_u - x_l)\, r$$

$$y = y_l + (y_u - y_l)\, r$$

Here r is the random number between 0 and 1.

Step 4 : Evaluate function at randomly generated points (x,y)

Step 5 : Select maxima and minima form the randomly evaluated function values

Step 6 : Display the maxima and minima values and stop.

C-program :

A C-program for random search based on above algorithm is given below :

```
/* Download this program from  www.vtubooks.com                              */
/* File name : rand_src.cpp                                                   */
/*----------------------- Random Search Method -----------------------*/

/*      This program evaluates the maxima and minima of the function
        f(x,y) using random search method.

                f(x,y) = y - x - 2*x*x - 2*x*y - y*y


        INPUTS :  Intervals [xl,xu] and [yl,yu].

        OUTPUTS : Maxima,Minima and corresponding coordinates            */

/*----------------------------  PROGRAM  --------------------------*/

#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<conio.h>

void main()
{
    double f ( double x, double y);  /* FUNCTION TO EVALUATE f(x,y)  */
    double x,y,xu,xl,yu,yl,fxy,fxyMax,fxyMin,r,xmin,ymin,xmax,ymax;
    int n;

    clrscr();

    printf("\n\t\tRandom Search Method\n");

    printf("\nMaxima and Minima of following function is calculated"
           "\n\tf(x,y) = y - x - 2*x*x - 2*x*y - y*y\n");

    randomize();/* INITIALIZATION OF PSEUDORANDOM NUMBER GENERATOR */

    printf("\nEnter the interval for x coordinate i.e. xl, xu = ");
    scanf("%lf %lf",&xl,&xu);                      /* Enter [xl, xu]*/
    printf("\nEnter the interval for y coordinate i.e. yl, yu = ");
    scanf("%lf %lf",&yl,&yu);                      /* Enter [yl, yu]*/
    printf("\nEnter the number of pseudorandom numbers\nto be "
           "used for random search = ");
                                          /* Accept random numbers */
    scanf("%d",&n);

    fxyMax = -1e100;
    fxyMin =  1e100;

    while(n-- > 0)
    {

      r = rand()/(RAND_MAX + 1.0);
                    /* CALLING PSEUDORANDOM NUMBER GENERATOR */
```

```
        x = xl + (xu-xl)*r;
        y = yl + (yu-yl)*r;

        fxy =  f(x,y);                      /* Evaluate f(x,y)*/

        if(fxy>fxyMax) {fxyMax = fxy; xmax = x; ymax = y;}
        if(fxy<fxyMin) {fxyMin = fxy; xmin = x; ymin = y;}
      }

    printf("\nMaximum = %lf at x = %lf y = %lf",fxyMax,xmax,ymax);
    printf("\n\nMinimum = %lf at x = %lf y = %lf",fxyMin,xmin,ymin);
}
/*-------------------------------------------------------------------*/

double f ( double x, double y)
{
   double fxy;

   fxy = y - x - 2*x*x - 2*x*y - y*y;           /*EVALUATION OF f(x,y)*/
   return(fxy);
}
/*------------------------ END OF PROGRAM --------------------------*/
```

The program initially accepts the values of `xl,xu` and `yl,yu`. These are the ranges of x and y over which search is to be made. Then the program accepts the number of random values of x and y to be used for search. Then there is a `while` loop to determine the maximum and minimum values. In this loop, the first statement is,

`r=rand()/(RAND-MAX+1.0);`

This calls the standard C function `rand()` and generates random numbers between 0 and 1. Then the next statements calculate x and y values using random number r. Then the next statement is,

`fxy = f (x,y)`

This statement evaluates $f(x,y)$ as per randomly calculated values of x and y. Then minima and maxima are checked by the next `if` statements. The program finally prints minimum/maximum values and their locations.

To test the program :

This program has a function `f` at the end. The statement in this function is,

`fxy=y-x-2*x*x-2*x*y-y*y;`

This statement implements $f(x,y)=y-x-2x^2-2xy-y^2$. The program calculates the maximum and minimum of this function. By changing the above statement, we can calculate maximum and minimum of other functions also.

Run this program and enter the following values of intervals :

$$[x_l, x_u] = [-2, 2] \quad \text{and}$$
$$[y_l, y_u] = [1, 3]$$

Then enter `n=500` random numbers for the search. The program then produces following results.

```
---------------------------- Result --------------------------------

                  Random Search Method

Maxima and Minima of following function is calculated
        f(x,y) = y - x - 2*x*x - 2*x*y - y*y

Enter the interval for x coordinate i.e. xl, xu = -2 2

Enter the interval for y coordinate i.e. yl, yu = 1 3

Enter the number of pseudorandom numbers
to be used for random search = 500

Maximum = 1.249476 at x = -1.012695 y = 1.493652

Minimum = -27.807508 at x = 1.990112 y = 2.995056
---------------------------------------------------------------------
```

The above results shows that maxima occurs at

x = – 1.012695 and y = 1.493652 and $f(x,y)$ = 1.249476

Similarly minima occurs at,

x = 1.990112 and y = 2.995056 and $f(x,y)$ = – 27.8075

Note that the above program evaluates global optima in the given interval.

9.3.2 Gradient Methods

The gradient methods use the derivatives of the functions to locate the maxima or minima. Determining maxima of a two dimensional function is like climbing up at the top of the mountain.

9.3.2.1 Steepest Ascent Method

The gradient of the function $f(x,y)$ is given as,

$$\nabla f = \frac{\partial f}{\partial x} i + \frac{\partial f}{\partial y} j \quad \text{... (9.3.1)}$$

The direction of this gradient represents the direction of search for the maxima. Hence search is initiated along this direction. When $f(x,y)$ stops increasing in this direction, gradient is re-evaluated. The new gradient gives new direction for travel. This process is repeated until the maxima is reached. This method is called *steepest ascent* method.

If (x_0, y_0) is the starting point, then co-ordinates of any point in the gradient direction are given as,

$$\left. \begin{aligned} x &= x_0 + \frac{\partial f}{\partial x} h \\ y &= y_0 + \frac{\partial f}{\partial y} h \end{aligned} \right\} \quad \text{... (9.3.2)}$$

Here h is the axis of travel along the gradient. In the above equation value of h is the distance along the gradient is to be substituted.

Ex. 9.3.1 *Maximize* $f(x,y) = 2xy + 2x - x^2 - 2y^2$ *with* $x_0 = -1$ *and* $y_0 = 1$.

Sol. : Let us evaluate gradient of the given function,

$$\left.\begin{aligned}\frac{\partial f}{\partial x} &= 2y+2-2x\\ \frac{\partial f}{\partial y} &= 2x-4y\end{aligned}\right\} \quad \text{... (9.3.3)}$$

Let us evaluate this gradient at the starting point $x_0 = -1$ and $y_0 = 1$. i.e.,

$$\frac{\partial f}{\partial x_0} = 2y_0+2-2x_0$$

$$= 2(1)+2-2(-1) = 6$$

$$\frac{\partial f}{\partial y_0} = 2\,x_0 - 4\,y_0$$

$$= 2\,(-1)-4\,(1) = -6$$

As per equation 9.3.1, the gradient vector is,

$$\nabla f = 6i - 6j$$

The maximum can be located along the gradient direction. For this, we have to transform the function as per equation 9.3.2. i.e.,

$$x = x_0 + \frac{\partial f}{\partial x}\,h$$

$$= -1+6h \quad \text{... (9.3.4)}$$

And $$y = y_0 + \frac{\partial f}{\partial y}\,h$$

$$= 1-6\,h \quad \text{... (9.3.5)}$$

Hence we can convert $f(x,y)$ to $g(h)$ using above subitutions. i.e.,

$$g(h) = f\,(x,y)$$

$$= f\,(-1+6h, 1-6h)$$

$$= 2(-1+6h)(1-6h)+2(-1+6h)-(-1+6h)^2-2(1-6h)^2$$

$$= -\,180h^2+72h-7$$

Here observe that we have mapped a two dimensional function $f(x,y)$ to one dimensional function $g(h)$. Here h is the axis along the direction of gradient. We proceed along this axis. Now we have to locate the maximum along h-direction. This can be done by differentiating $g(h)$ and setting it equal to zero. i.e.,

$$g'(h) = \frac{d}{dh}\,g(h) = -360h+72$$

Setting the above equation to zero, we get value of h corresponding to maxima to $f(x,y)$ along h-direction.

i.e. $\quad -360h+72 = 0 \quad \Rightarrow \quad h = 0.2$

Thus if we travel a distance of 0.2 along h-axis, then we will reach a maxima of $f(x, y)$ in h direction. This is illustrated in Fig. 9.3.1. In this figure observe that a gradient is initiated from point (x_0, y_0). Then after $h = 0.2$, a maximum (x_1, y_1) is reached. The co-ordinates of this maxima can be obtained from equation 9.3.4 and equation 9.3.5.

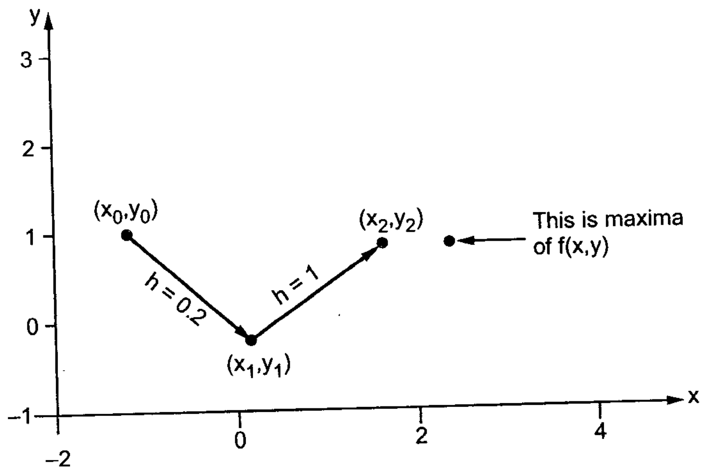


Fig. 9.3.1 Direction of travel in steepest ascent method

i.e.,

$$x_1 = -1 + 6h = -1 + 6\,(0.2) = 0.2$$

and

$$y_1 = 1 - 6h = 1 - 6\,(0.2) = -\,0.2$$

Thus a maxima of $f(x, y)$ occurs at $(0.2, -0.2)$ in h-direction.

Now we have to establish the new h-direction. Hence let us consider the starting point be $(x_1, y_1) = (0.2, -0.2)$, and evaluate gradient at this point. This can be obtained from equation 9.3.3. i.e.,

$$\frac{\partial f}{\partial x_1} = 2\,y_1 + 2 - 2x_1$$

$$= 2\,(-0.2) + 2 - 2\,(0.2) = 1.2$$

$$\frac{\partial f}{\partial y_1} = 2\,x_1 - 4\,y_1$$

$$= 2\,(0.2) - 4\,(-0.2) = 1.2$$

Thus the gradient vector is,

$$\nabla f = 1.2\,i + 1.2\,j$$

To locate maximum along this gradient we have to obtain $g(h)$. Hence from equation 9.3.2,

$$x = x_1 + \frac{\partial f}{\partial x_1} h$$

$$= 0.2 + 1.2\, h \qquad \text{... (9.3.6)}$$

$$y = y_1 + \frac{\partial f}{\partial y_1} h$$

$$= -0.2 + 1.2\, h \qquad \text{... (9.3.7)}$$

Hence $g(h)$ can be obtained from above equation as,

$$g(h) = f(x, y)$$

$$= f(0.2 + 1.2h, -0.2 + 1.2h)$$

$$= 2(0.2+1.2h)(-0.2+1.2h) + 2(0.2+1.2h) - (0.2+1.2h)^2 - 2(-0.2+1.2h)^2$$

$$= -1.44h^2 + 2.88h + 0.2$$

The maxima along this direction can be obtained by differentiating the above equation and setting to zero. i.e.,

$$g'(h) = -2.88\, h + 2.88$$

Hence, $-2.88h + 2.88 = 0 \quad \Rightarrow \quad h = 1$

Thus after travelling a distance of $h = 1$, in h-direction we reach a maxima of $f(x,y)$ in this h-direction. This is shown in Fig. 9.3.1. Thus we reach very close to the actual maxima of $f(x,y)$. The co-ordinates (x_2, y_2) of the maxima along h-direction can be obtained from equation 9.3.6 and equation 9.3.7. i.e.,

$$x_2 = 0.2 + 1.2\, h$$

$$= 0.2 + 1.2\,(1) = 1.4$$

and

$$y_2 = -0.2 + 1.2h$$

$$= -0.2 + 1.2\,(1) = 1$$

Thus $(x_2, y_2) = (1.4, 1)$ are the co-ordinates of maximum along direction of new gradient. Observe in Fig. 9.3.1 that these co-ordinates are very close to the actual maximum. The above process can be repeated by evaluating the new gradient at (x_2, y_2).

9.4 Constrained Optimization

Now let us discuss the optimization techniques where constraints appear in the problems. The constraints can be linear or nonlinear. In this chapter we will mainly concentrate on linear constraints.

9.4.1 Linear Programming

Linear programing methods are used to solve the problems having linear constraints. The linear programing problem consists of objective function and a set of constraints.

For example,

Maximize $\quad Z = c_1\, x_1 + c_2\, x_2 + \ldots\ldots + c_n\, x_n$ ... (9.4.1)

This is an objective function and constraints can be represented as,

$a_{i1}\, x_1 + a_{i2}\, x_2 + \ldots\ldots + a_{in}\, x_n < b_i$ ... (9.4.2)

Here we have to determine the values of $x_1, x_2, \ldots\ldots x_n$ which will maximize z, satisfying the above constraints. Note that equation 9.4.1 and equation 9.4.2 both are linear. Such linear programming problems can be solved graphically upto two variables. For large number of variables simplex method is used. In this section we will discuss both of these methods.

9.4.1.1 Graphical Method

This method is preferable, when the linear programming problem has two variables. The two variables (x_1, x_2) are plotted along two axes. The constraints are plotted as straight lines in the (x_1, x_2) plane of variables. A space which satisfies all the constraint lines is the solution space. The objective function z is then superimposed on this solution space. Then maximum or minimum values of z are obtained in the solution space.

Ex. 9.4.1 *Maximize* $z = 150\, x_1 + 175\, x_2$

Subject to $7\, x_1 + 11\, x_2 \leq 77$ ... (1)

$$10\, x_1 + 8\, x_2 \leq 80 \quad \ldots (2)$$

$$x_1 \leq 9 \quad \ldots (3)$$

$$x_2 \leq 6 \quad \ldots (4)$$

$$x_1 \geq 0 \quad \ldots (5)$$

$$x_2 \geq 0 \quad \ldots (6)$$

Sol. : To determine the solution space :

There are 6 constraints as numbered above and one objective function, z. Let us first plot the constraints in the $x_1 - x_2$ plane.

Consider the 1st constraint,

$$7\, x_1 + 11\, x_2 \leq 77$$

This can be written as,

$$x_2 = -\frac{7}{11}\, x_1 + 7$$

This represents a straight line and plotted in $x_1 - x_2$ plane as shown in Fig. 9.4.1. Then 2nd constraint is,

$$10\, x_1 + 8\, x_2 \leq 80$$

This can be written as,

$$x_2 = -\frac{5}{4}\, x_1 + 10$$

This line is also plotted in Fig. 9.4.1. Then the lines for other constraints are also plotted in Fig. 9.4.1. The shaded area in Fig. 9.4.1 satisfies all the constraints. Hence the solution space is the shaded area in Fig. 9.4.1. Observe that the line due to $x_1 \leq 9$ does not affect the solution space. Hence it is redundant constraint and it can be neglected.

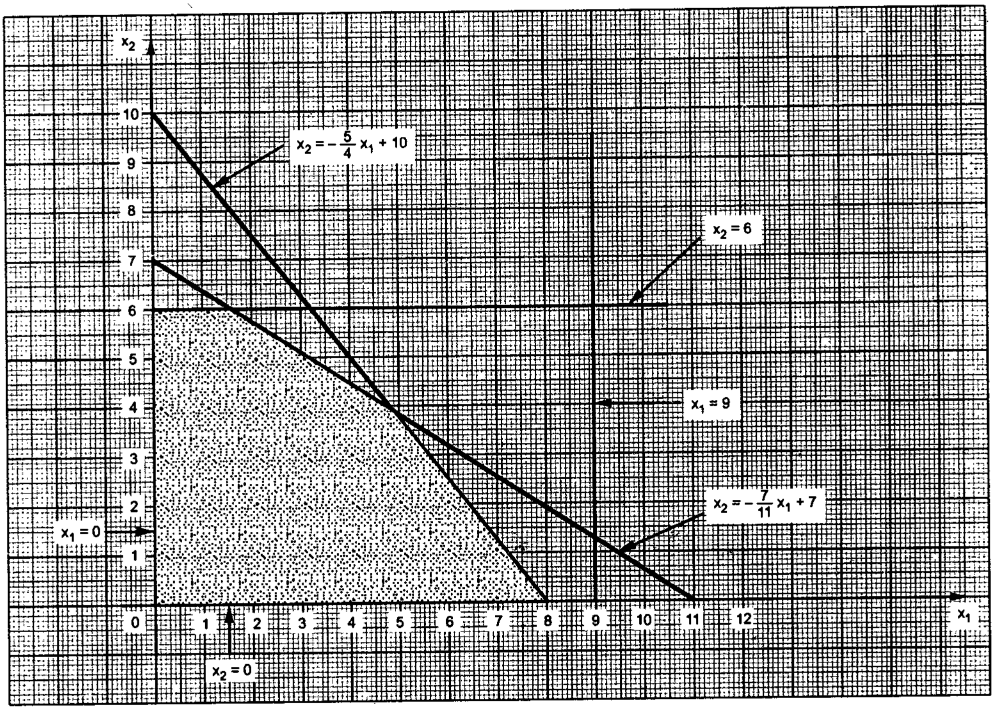


Fig. 9.4.1 Solution space of Ex. 9.4.1. The shaded area shows the solution space

To determine x_1 and x_2 for maximized z :

First let us begin with $z = 0$. Then the objective function becomes,

$$0 = 150\, x_1 + 175\, x_2$$

i.e.
$$x_2 = -\frac{150}{175} x_1$$

or
$$x_2 = -\frac{6}{7} x_1 \qquad \text{... (9.4.3)}$$

This represents a line which passes through the origin. The solution space of Fig. 9.4.1 is redrawn in Fig. 9.4.2. The line represented by above equation is plotted in Fig. 9.4.2. It is shown dotted.

Now let us increase z by 500. Then objective function becomes,

$$500 = 150\, x_1 + 175\, x_2$$

This equation can be written as,

$$x_2 = -\frac{6}{7} x_1 + \frac{20}{7} \qquad \text{... (9.4.4)}$$

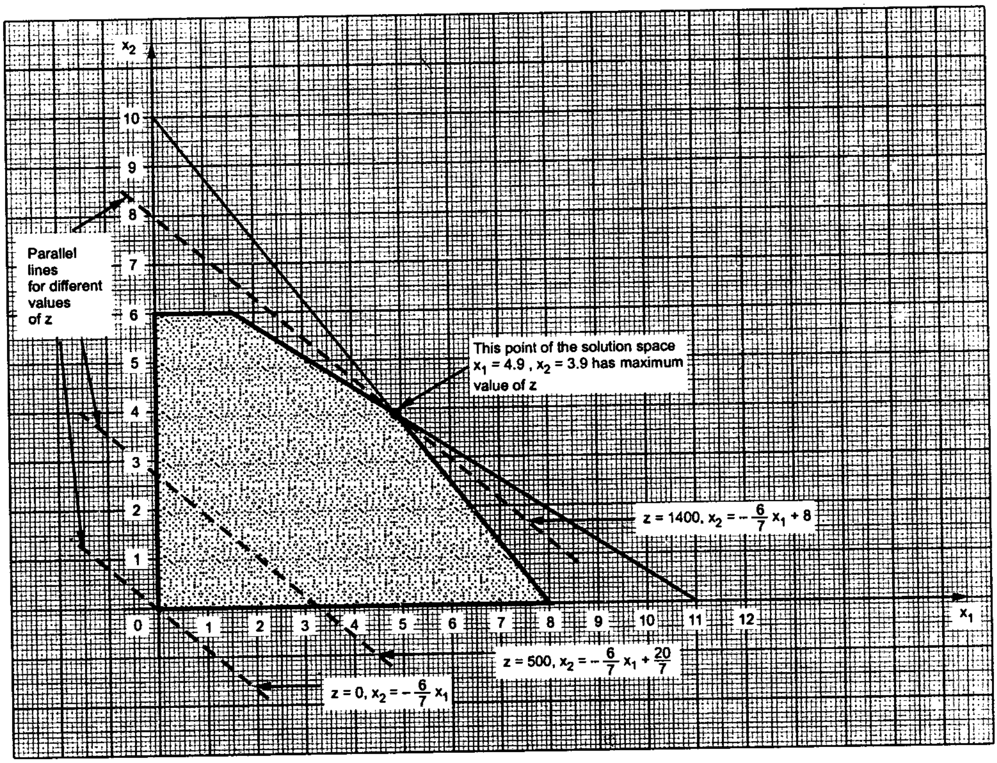


Fig. 9.4.2 Z is maximized at approximately $x_1 = 4.9$ and $x_2 = 3.9$

This represents a line parallel to line of equation 9.4.3 (since their slopes are same i.e. $-\frac{6}{7}$). Above line has the intercept of $\frac{20}{7}$ when $x_1 = 0$. This line is shown dotted in Fig. 9.4.2. In this figure observe that the above line also passes through solution space. Similar parallel lines are drawn which pass through the solution space and higher values of z. At $z = 1400$, the objective function becomes,

$$1400 = 150\, x_1 + 175\, x_2$$

This equation can be written as,

$$x_2 = -\frac{6}{7}\, x_1 + 8$$

This also represents a line parallel to the earlier ones. It has an intercept of 8 when $x_1 = 0$. This line is plotted in Fig. 9.4.2. Observe that the above line just touches the solution space at $x_1 = 4.9$ and $x_2 = 3.9$. This is the only point from solution space corresponding to maximum value of z, which is 1400. Other points give less value of z. Hence the solution is,

$$x_1 = 4.9 \quad \text{and} \quad x_2 = 3.9 \text{ with } z = 1400.$$

It can be varified easily that above values of x_1 and x_2 satisfy all the constraints.

9.4.1.2 Simplex Method

The simplex method is very efficient method to solve linear programming problems having more than two variables. We will illustrate simplex method with the help of following example.

Ex. 9.4.2 *Maximize* $F = x_1 + 2x_2 + x_3$ *subject to*

$$2x_1 + x_2 - x_3 \leq 2 \quad \text{... (1)}$$

$$-2x_1 + x_2 - 5x_3 \geq -6 \quad \text{... (2)}$$

$$4x_1 + x_2 + x_3 \leq 6 \quad \text{... (3)}$$

$$x_1, x_2 \text{ and } x_3 \geq 0 \quad \text{... (4)}$$

Sol. : Let us change the sign of the objective function so as to convert it to a minimization problem. Also change the signs of the second constraint so that its RHS (i.e. – 6) will be positive. i.e.,

Minimize $f = -x_1 - 2x_2 - x_3$ subject to

$$2x_1 + x_2 - x_3 \leq 2 \quad \text{... (1)}$$

$$2x_1 - x_2 + 5x_3 \leq 6 \quad \text{... (2)}$$

$$4x_1 + x_2 + x_3 \leq 6 \quad \text{... (3)}$$

$$x_1, x_2, x_3 \geq 0 \quad \text{... (4)}$$

Let us introduce the variables $x_4 \geq 0$, $x_5 \geq 0$ and $x_6 \geq 0$ in above system of equations so that it will be in canonical form. These variables are called slack variables. i.e.,

$$\left.\begin{aligned} 2x_1 + x_2 - x_3 + x_4 &= 2 \\ 2x_1 - x_2 + 5x_3 + x_5 &= 6 \\ 4x_1 + x_2 + x_3 + x_6 &= 6 \\ -x_1 - 2x_2 - x_3 - f &= 0 \end{aligned}\right\} \quad \text{... (9.4.5)}$$

Here x_4, x_5, x_6 and $-f$ can be taken as basic variables. The basic solution of above equations can be obtained by setting nonbasic variables to zero. i.e. $x_1 = x_2 = x_3 = 0$.

Hence we get,

$$x_4 = 2, \quad x_5 = 6, \quad x_6 = 6 \quad \text{and} \quad f = 0$$

This is a feasible solution but not optimum since coefficients of x_1, x_2 and x_3 in object function $f = -x_1 - 2x_2 - x_3$ are negative. These coefficients are -1, -2 and -1, which are negative.

Step I : Let us improve the solution. Consider the object function $f = -x_1 - 2x_2 - x_3$. In this equation x_2 has most negative coefficient, i.e. -2. Hence x_2 will be the basic variable. Now consider the constraint equation of equation 9.4.5. i.e.,

$$2x_1 + x_2 - x_3 + x_4 = 2 \quad \text{... (9.4.6 (a))}$$

$$2x_1 - x_2 + 5x_3 + x_5 = 6 \quad \text{... (9.4.6 (b))}$$

$$4x_1 + x_2 + x_3 + x_6 = 6 \quad \text{... (9.4.6 (c))}$$

Here observe that x_2 has positive coefficient in equation 9.4.6 (a) and equation 9.4.6 (c). The ratios of RHS to these coefficients are,

$$\text{Equation 9.4.6 (a)} \Rightarrow \frac{\text{R.H.S. of equation 9.4.6(a)}}{\text{Coefficient of } x_2} = \frac{2}{1} = 2$$

$$\text{Equation 9.4.6 (c)} \Rightarrow \frac{\text{R.H.S. of equation 9.4.6(c)}}{\text{Coefficient of } x_2} = \frac{6}{1} = 6$$

Since lowest ratio is given by equation 9.4.6 (a), we have to pivot on x_2 of this equation. This means we have to eliminate x_2 from all other equations except equation 9.4.6 (a). Consider equation 9.4.6 (a),

$$2x_1 + x_2 - x_3 + x_4 = 2 \qquad \text{... (9.4.7 (a))}$$

Add this equation to 2^{nd} equation of equation 9.4.5. i.e.,

$$4\,x_1 + 0\,x_2 + 4\,x_3 + x_4 + x_5 = 8 \qquad \text{... (9.4.7 (b))}$$

Subtract equation 9.4.7 (a) from 3^{rd} equation of equation 9.4.5. i.e.,

$$2x_1 + 0\,x_2 + 2\,x_3 - x_4 + x_6 = 4 \qquad \text{... (9.4.7 (c))}$$

Multiply equation 9.4.7 (a) by 2 and add it to 4^{th} (last) equation of equation 9.4.5. i.e.,

$$3x_1 + 0\,x_2 - 3\,x_3 + 2\,x_4 - f = 4 \qquad \text{... (9.4.7 (d))}$$

The equations obtained above from equation 9.4.7 (a) to (d) are written together below :

$$\left.\begin{aligned} 2\,x_1 + 1\,x_2 - x_3 + x_4 \qquad &= 2 \\ 4\,x_1 + 0\,x_2 + 4\,x_3 + x_4 + x_5 &= 8 \\ 2\,x_1 + 0\,x_2 + 2\,x_3 - x_4 + x_6 &= 4 \\ 3\,x_1 + 0\,x_2 - 3\,x_3 + 2\,x_4 \quad - f &= 4 \end{aligned}\right\} \qquad \text{... (9.4.8)}$$

Now x_2 will be basic variable. Hence from first equation of the above system, x_1, x_3 and x_4 will be nonbasic variables. i.e., $x_1 = x_3 = x_4 = 0$ (nonbasic variables).

Hence values of basic variables becomes,

$x_2 = 2, \quad x_5 = 8, \quad x_6 = 4$ and $f = -4$.

Let us consider the last equation of equation 9.4.8. i.e.,

$$3\,x_1 + 0\,x_2 - 3\,x_3 + 2\,x_4 - f = 4$$

Since x_3 has negative coefficient in above equation, the present solution is not optimum.

Step II : Let us improve the solution further.

Consider the last equation of equation 9.4.8. i.e.,

$$3\,x_1 + 0\,x_2 - 3\,x_3 + 2\,x_4 - f = 4$$

In this equation x_3 has most negative coefficient, i.e. -3. Hence x_3 will be the basic variable in this step. Now consider the first three (constraint) equations of equation 9.4.8. i.e.,

$$2x_1 + 1\,x_2 - x_3 + x_4 = 2 \quad \text{... (9.4.9 (a))}$$

$$4\,x_1 + 0\,x_2 + 4\,x_3 + x_4 + x_5 = 8 \quad \text{... (9.4.9 (b))}$$

$$2x_1 + 0\,x_2 + 2\,x_3 - x_4 + x_6 = 4 \quad \text{... (9.4.9 (c))}$$

Here observe that x_3 has positive coefficient in equation 9.4.9 (b) and equation 9.4.9 (c). The ratios of RHS to these coefficients are,

$$\text{equation 9.4.9 (b)} \Rightarrow \frac{\text{R.H.S. of equation 9.4.9(b)}}{\text{Coefficient of } x_3} = \frac{8}{4} = 2$$

$$\text{equation 9.4.9 (c)} \Rightarrow \frac{\text{R.H.S. of equation 9.4.9(c)}}{\text{Coefficient of } x_3} = \frac{4}{2} = 2$$

Since both the ratios are same we can pivot on x_3 of any equation. Let us pivot on x_3 of equation 9.4.9 (b). This means eliminate x_3 from all other equations.

Divide equation 9.4.9 (b) by the pivot i.e. coefficient of x_3 i.e. 4. Hence we get,

$$\frac{1}{4}(4\,x_1 + 0x_2 + 4\,x_3 + x_4 + x_5 = 8)$$

$$\text{i.e. } 1\,x_1 + 0\,x_2 + 1\,x_3 + \frac{1}{4}\,x_4 + \frac{1}{4}\,x_5 = 2 \quad \text{... (9.4.10 (a))}$$

Add the above equation to first equation of equation 9.4.8. i.e.,

$$3\,x_1 + 1\,x_2 + 0\,x_3 + \frac{5}{4}\,x_4 + \frac{1}{4}\,x_5 = 4 \quad \text{... (9.4.10 (b))}$$

Multiply equation 9.4.10 (a) by 2 and subtract from 3rd equation of equation 9.4.8. i.e.,

$$0\,x_1 + 0\,x_2 + 0\,x_3 - \frac{3}{2}\,x_4 - \frac{1}{2}\,x_5 + x_6 = 0 \quad \text{... (9.4.10 (c))}$$

Multiply equation 9.4.10 (a) by 3 and add it to 4th (last) equation of equation 9.4.8. i.e.,

$$6\,x_1 + 0\,x_2 + 0\,x_3 + \frac{11}{4}\,x_4 + \frac{3}{4}\,x_5 - f = 10 \quad \text{... (9.4.10 (d))}$$

Equation 9.4.10 (b) is written first, then equation 9.4.10 (a), (c) and (d) are written below :

$$\left.\begin{array}{ll} 3x_1 + 1x_2 + 0x_3 + \frac{5}{4}x_4 + \frac{1}{4}x_5 & = 4 \\ 1x_1 + 0x_2 + 1x_3 + \frac{1}{4}x_4 + \frac{1}{4}x_5 & = 2 \\ 0x_1 + 0x_2 + 0x_3 - \frac{3}{2}x_4 - \frac{1}{2}x_5 + x_6 & = 0 \\ 6x_1 + 0x_2 + 0x_3 + \frac{11}{4}x_4 + \frac{3}{4}x_5 \quad - f & = 10 \end{array}\right\} \quad \text{... (9.4.11)}$$

Now x_3 is the basic variable for this step. Hence from 2^{nd} equation of above system x_1, x_4 and x_5 will be the nonbasic variables. Hence,

$$x_1 = x_4 = x_5 = 0$$

Putting these values in above system of equations we get values of basic variables. i.e.,

$$x_2 = 4, \quad x_3 = 2, \quad x_6 = 0 \quad \text{and} \quad f = -10$$

Consider the last equation of equation 9.4.11,

$$6\, x_1 + 0\, x_2 + 0\, x_3 + \frac{11}{4}\, x_4 + \frac{3}{4}\, x_5 - f = 10$$

Since coefficients of all x's are positive in this equation, the solution is optimum.

Table form of simplex method : All the calculations given above can be done in the table form as shown below.

Basic variables	Variables						$-f$	b_i	$\frac{b_i}{a_{is}}$ for $a_{is} > 0$
	x_1	x_2	x_3	x_4	x_5	x_6			
x_4	2	$\boxed{1}$ Pirot element	-1	1	0	0	0	2	$\frac{2}{1} = 2$ This is smaller. Hence x_4 drops from basis
x_5	2	-1	5	0	1	0	0	6	
x_6	4	1	1	0	0	1	0	6	$\frac{6}{1} = 6$
$-f$	-1	-2	-1	0	0	0	1	0	
		This is most negative, hence x_2 will be basic							

Pivoting is performed on coefficient of x_2, i.e. 1. x_2 is the basic variable in place of x_4. Then the results generated are as follows :

Basic variables	Variables						$-f$	b_i	$\frac{b_i}{a_{is}}$ for $a_{is} > 0$
	x_1	x_2	x_3	x_4	x_5	x_6			
x_2	2	1	$\boxed{-1}$	1	0	0	0	2	$\frac{8}{4} = 2$ select this, hence x_5 drops from next basis
x_5	4	0	4 Pivot element	1	1	0	0	8	
x_6	2	0	2	−1	0	1	0	4	$\frac{4}{2} = 2$
$-f$	3	0	−3	2	0	0	1	4	

This is most negative, hence x_3 enters next basis

Pivoting is performed on coefficient of x_3, i.e. 4. Then x_3 will be basic variable in place of x_5. Then the results generated are as follows :

Basic variables	Variables						$-f$	b_i	$\frac{b_i}{a_{is}}$ for $a_{is} > 0$
	x_1	x_2	x_3	x_4	x_5	x_6			
x_2	3	1	0	5/4	1/4	0	0	4	
x_3	1	0	1	1/4	1/4	0	0	2	
x_6	0	0	0	−3/2	−1/2	1	0	0	
$-f$	6	0	0	11/4	3/4	0	1	10	

Since all the coefficients in equation for $-f$ are ≥ 0, this is an optimum solution. The solution is,

$$x_2 = 4, \quad x_3 = 2, \quad x_6 = 0 \quad \text{and} \quad f = -10.$$

Ex. 9.4.3 *Applying simplex method, find the minimum of $f = x_1 - 2x_2$ subject to the following constraints,*

$0 \leq x_1$

$0 \leq x_2$

$x_1 + x_2 \leq 4$

$-x_1 + x_2 \leq 1$

$x_1 + x_2 \leq 3$ **[May - 2000, 8 marks, May - 98, 8 marks, Dec - 96, 8 marks]**

Sol. : Let us introduce the variables $x_3 \geq 0$, $x_4 \geq 0$ and $x_5 \geq 0$ in above system of equation so that it will be in canonical form and inequalities will be converted to equalities. These variables are called slack variables. i.e.,

$$\left.\begin{aligned} x_1 + 2x_2 + x_3 &= 4 \\ -x_1 + x_2 + x_4 &= 1 \\ x_1 + x_2 + x_5 &= 3 \\ x_1 - 2x_2 - f &= 0 \end{aligned}\right\} \qquad \dots (9.4.12)$$

Here x_3, x_4, x_5 and $-f$ are basic variables. And x_1, x_2 are nonbasic variables. A tableau of above equations is formed as shown below :

Basic variables	Variables					$-f$	b_i	$\frac{b_i}{a_{is}}$ for $a_{is} > 0$	
	x_1	x_2	x_3	x_4	x_5				
x_3	1	2	1	0	0	0	4	$\frac{4}{2} = 2$	
x_4	−1	$\boxed{1}$ Pivot element	0	1	0	0	1	$\frac{1}{1} = 1$	This is smaller, hence x_4 drops from next basis
x_5	1	1	0	0	1	0	3	$\frac{3}{1} = 1$	
$-f$	1	−2	0	0	0	1	0		

This is most negative, hence x_2 enters next basis

Pivoting is performed on coefficient of x_2, i.e. 1. Then x_2 will be the basic variable in place of x_4. Then the results generated are as follows :

Basic variables	Variables					$-f$	b_i	$\frac{b_i}{a_{is}}$ for $a_{is} > 0$	
	x_1	x_2	x_3	x_4	x_5				
x_3	$\boxed{3}$ Pivot element	0	1	−2	0	0	2	$\frac{2}{3} = 0.667$	This is small, hence x_3 drops from next basis
x_2	−1	1	0	1	0	0	1		
x_5	2	0	0	−1	1	0	2	$\frac{2}{2} = 1$	
$-f$	−1	0	0	2	0	1	2		

This is most negative, hence x_1 enters next basis

Pivoting is performed on coefficient of x_1, i.e. 3. Then x_1 will be the basic variable in place of x_3. The results generated are given below :

Basic variables	Variables					$-f$	b_i	$\frac{b_i}{a_{is}}$ for $a_{is} > 0$
	x_1	x_2	x_3	x_4	x_5			
x_1	1	0	$\frac{1}{3}$	$-\frac{2}{3}$	0	0	$\frac{2}{3}$	
x_2	0	1	$\frac{1}{3}$	$\frac{1}{3}$	0	0	$\frac{5}{3}$	
x_5	0	0	$-\frac{2}{3}$	$\frac{1}{3}$	1	0	$\frac{2}{3}$	
$-f$	0	0	$\frac{1}{3}$	$\frac{4}{3}$	0	1	$\frac{8}{3}$	

All the coefficients in equation for $-f$ are ≥ 0. Hence this is an optimum solution. We can write the above equations as,

$$x_1 + 0\,x_2 + \frac{1}{3}\,x_3 - \frac{2}{3}\,x_4 = \frac{2}{3}$$

$$0\,x_1 + x_2 + \frac{1}{3}\,x_3 + \frac{1}{3}\,x_4 = \frac{5}{3}$$

$$0\,x_1 + 0\,x_2 - \frac{2}{3}\,x_3 + \frac{1}{3}\,x_4 + x_5 = \frac{2}{3}$$

$$0\,x_1 + 0\,x_2 + \frac{1}{3}\,x_3 + \frac{4}{3}\,x_4 - f = \frac{8}{3}$$

In the above equations x_1, x_2 and x_5 are basic variables and x_3, x_4 are non basic variables. Hence with $x_3 = x_4 = 0$ in above equations we get,

$$x_1 = \frac{2}{3}, \quad x_2 = \frac{5}{3}, \quad x_5 = \frac{2}{3} \text{ and } f = -\frac{8}{3}$$

Thus function achieves a minima of $-\frac{8}{3}$ at $x_1 = \frac{2}{3}$ and $x_2 = \frac{5}{3}$.

Exercise

1. Maximize $f = 2\,x_1 + 6\,x_2$ subject to
   $-x_1 + x_2 \leq 1$
   $2\,x_1 + x_2 \leq 2$
   $x_1 \geq 0$
   $x_2 \geq 0$
2. Maximize $f = -2\,x_1 - x_2 + 5\,x_3$ subject to
   $x_1 - 2x_2 + x_3 \leq 8$
   $3\,x_1 - 2\,x_2 \geq -18$
   $2\,x_1 + x_2 - 2\,x_3 \leq -4$

University Questions

1. Write a short note on linear programming.

[Dec - 95, May - 98, Dec - 98, Dec - 99, Dec 2000]

2. Applying the simplex method, find the minimum of $F = x_1 - 2x_2$ subject to the following constraints

$$\begin{aligned} 0 &\le x_1 \\ 0 &\le x_2 \\ x_1 + 2x_2 &\le 4 \\ -x_1 + x_2 &\le 1 \\ x_1 + x_2 &\le 3 \end{aligned}$$

[Dec - 96, May - 98, May - 2000]

3. Write a detailed note on Simplex method for minimisation.

[Dec - 97, May - 99]

9.5 Engineering Applications

Consider an electric circuit shown in Fig. 9.5.1. The power transferred to the load resistance can be derived from kirchoff's laws. It is given as,

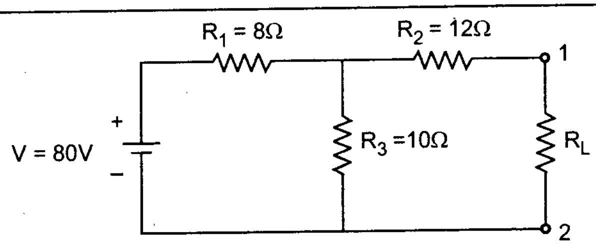


Fig. 9.5.1 An electric circuit

$$P(R_L) = \frac{\left(\dfrac{V\, R_3\, R_L}{R_1\,(R_L + R_2 + R_3) + R_3\, R_L + R_3\, R_2}\right)^2}{R_L} \qquad \text{... (9.5.1)}$$

Putting the values in above equation,

$$P\,(R_L) = \frac{\left(\dfrac{80 \times 10\, R_L}{8\,(R_L + 12 + 10) + 10\, R_L + 10 \times 12}\right)^2}{R_L}$$

$$= \frac{1975.3\, R_L}{(R_L + 16.44)^2} \qquad \text{... (9.5.2)}$$

The above equation shows that power transferred to the load is the function of load itself. Here we can calculate the value of R_L that maximizes power transferred across the load. This problem can be solved using optimization techniques.

Let us use the C-program of golden section search to determine maximum power and corresponding load resistance. We can write equation 9.5.2 as,

$$f(x) = \frac{1975.3\,x}{(x+16.44)^2}$$

Let us use the interval as $x_l = 5\Omega$ to $x_u = 50\,\Omega$. The results generated by the program are given below :

```
Golden section search method
Enter xl=5
Enter xu=50
Press any key to see step by step display of results ...
Press 'q' to stop
x               f (x)
32.811529       26.718994
22.188471       29.372804
15.623059       30.018517
18.130823       29.966163
15.623059       30.018517
16.580940       30.037470
17.172942       30.023735
16.580940       30.037470
16.807065       30.034356
16.580940       30.037470
16.441187       30.038017
16.494568       30.037935
16.441187       30.038017
16.461576       30.038004
16.441187       30.038017
16.448975       30.038015
16.441187       30.038017
16.444162       30.038017
16.441187       30.038017
```

The above results show that maximum power transfer of 30.038017 W occurs at $R_L = 16.441187\ \Omega$. Thus other engineering applications can be solved by optimization techniques.

Computer Exercise

1. Write a C-program for implementing the simplex method.
2. Write a C-program for steepest ascent method to locate maxima of a function.

❑❑❑

Index

A

B

C

D

E

Contents

Second Revised Edition
2008

Price INR 295/-
10 Digit: ISBN 81-8431-018-8
13 Digit: ISBN 978-81-8431-(

TECHNICAL PUBLICATIONS PUNE

Technical Publications Pune
1, Amit Residency, 412 Shaniwar Peth, Pune - 411030, M.S., India.
Telefax : +91 (020) 24495496/97, Email : technical@vtubooks.com
Visit us at : www.vtubooks.com