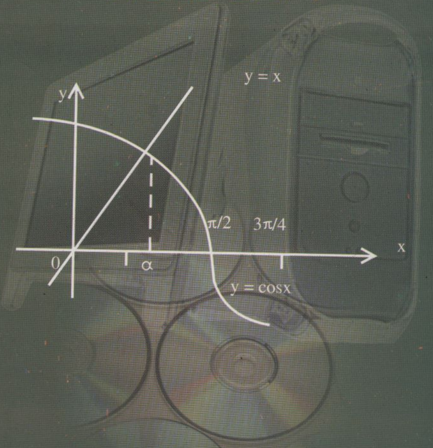
Prof.Dr. S A Bhatti
Mr. N A Bhatti

A First Course in NUMERICAL ANALYSIS With C++



Fifth Edition

Numerical Analysis is a Science

– computation is an art.

Fifth Enlarged Edition

A First Course in NUMERICAL ANALYSIS With C++

Prof. Saeed Akhter Bhatti
Mr. Naeem Akhtar Bhatti

SHAHARYAR PUBLISHERS
AL-FAZAL MARKET, URDU BAZAR, LAHORE.

CAN BE HAD FROM:
A-ONE PUBLISHERS
AL-FAZAL MARKET, URDU BAZAR, LAHORE
Phone No. 37232276 - 37357177 - 37224655
Email: aonepub@hotmail.com / info @aonepublishers.com
Website: aonepublishers.com

Book: A First Course in Numerical Analysis with C++

Author: Prof. Saeed Akhter Bhatti
Mr. Naeem Akhtar Bhatti

First Edition: May, 1990
Second Edition: Mar, 1996
Third Edition: Aug, 1999
Fourth Edition: May, 2002
Fifth Edition: Jan, 2008
Reprint: 2013

Quantity: 1100

Price: **Rs. 400/= (Rupees Four Hundred Only)**
Library Edition: **Rs. 500/= (Rupees Five Hundred Only)**

Published by:
Muhammad Azhar
(Shaharyar Publishers)

Printed at:
Ali Ejaz Printers,
Rattigun Road, Lahore

ISBN: **969-8105-01-8**

In the loving memory of our parents,

Mr and Begum Sana Ullah Sufi

[illegible] and can be avoided if [illegible] We shall primarily [illegible]

Rounding Errors

When a numerical method is actually run on a digital computer after transforming it to computer program, some kind of error called round-off error is introduced.

[illegible] of numbers to a limited number of decimal [illegible] rounding error. In simple words, the error in the result that is caused by rounding is called round-off error. For example, it would be impracticable to [illegible] the distance between two points on the earth as 15.2967 metres; it would be more [illegible] 15 metres. Thus, the error introduced by rounding is 0.2967 metres. Another example is the value of [illegible] 3.142.

[illegible] very important role in numerical analysis. [illegible] round-off, we may apply the following rules [illegible] However [illegible] these rules are not normally applied when [illegible]

[illegible] we are given a number and we want to round it [illegible] first decimal place [illegible] after the first decimal place and proceed as follows:

(a) If the first discarded digit is less than 5, the previous digit is unchanged. For example, the number 56.44, when rounded to the first decimal place, becomes [illegible]

(b) If the discarded digit is greater than 5, the previous digit is increased by 1. For example, the number 56.46, when rounded to first decimal, it becomes [illegible]

(c) If the discarded digit is exactly 5 the previous digit is unchanged, if it is even [illegible] increased by 1, if it is odd. For example, the number 56.65 becomes [illegible] and 56.75 becomes 56.8.

However, in the most commonly used rule (we are familiar with) for rounding-off the number [illegible] the discarded digit exceeds or equals 5, we add 1 to the last retained digit.

A good analysis of the round-off errors present in the final result of a numerical computation, usually termed the accumulated rounded off error, is difficult, particularly when the algorithm used is of some complexity. Except in very simple cases, the accumulated error is [illegible] the sum of the local round-off error, that is, errors [illegible] individual rounding or truncating operations. The local error at any stage [illegible] is propagated throughout the remaining part of the computation. In order to establish a round-off error bound, we may assume the worst possible [illegible] each arithmetic operation and follow the propagation of [illegible] the subsequent calculations.

Preface To The Fifth Edition

The Goal of this enlarged edition of our book on **Numerical Analysis** remains the same as for the previous editions: to give a comprehensive and state-of-the-art treatment of all the important aspects of the subject. In this, we have made modifications in all the first eight chapters and added extra problems at the end of each chapter. A new chapter, Chapter 9, based on eigenvalues and eigenvectors has been included. We have tried to cover all the basic and important procedures to compute eigenvalues and eigenvectors of a matrix. This chapter has been written especially on the request of users of the subject in various engineering universities.

We gratefully thank the users and the reviewers of the previous editions who provided valuable suggestions and ideas for the improvement of this book. Their feedback is valuable in our efforts for continuous improving this book. We are also thankful to our various teaching assistants both at BIIT and FUIMCS who checked the references and exercises in many chapters.

The authors would also like to thank Professor Akram Javed, Faculty of Science, University of Engineering and Technology, Taxila, for his many useful comments. We are thankful to Prof. Aftab Ahmad, Director, Institute of Management and Computer Sciences, Foundation University, Rawalpindi, for providing us the necessary infrastructure to complete this project. Thank you all.

The Bhattis
Islamabad,

Preface To The Fourth Edition

The Fourth Edition of this book on numerical analysis is in your hands now. It is geared specifically to the needs and background of our students. During this period, we received several comments from the users. In reviewing their comments, we have made modifications in some chapters of the book to sharpen the reader's understanding of the material presented. The plan of presentation of all chapters has been that of step by step. We start with an elementary method and then proceed to develop this or alternative, more sophisticated methods. The presentation just given is, of course, much over-simplified. In practice, a combination of conventional mathematical analysis and numerical analysis is likely to be used. Proofs of formulas are given where these are reasonably easy to follow but have been omitted in the more difficult cases.

A major change has been made in computer programs that implement the use of numerical methods presented in the book for solving problems. This edition contains computer programs written in C++. They have deliberately been kept as straightforward as possible so that the reader should understand the precise function of every step in each program. While the programs are intended primarily for educational-purposes, they can, of course, be used for solving some simple practical problems. However, for more complex practical problems, they do not offer any guarantee regarding the accuracy, adequacy or completeness of any information herein. Therefore, the user should make use of the excellent software packages now available. Hopefully the reader will appreciate this edition. We recommend them to learn and make more substantial use of their computers. We have benefited much by sitting at the feet of the wise, and we hope that, through this book, it may be possible to transmit a spark from their fire to all our readers. Good luck!

We would like to thank the users and reviewers of the previous editions whose comments and suggestions have enormously proved to be valuable in updating the material of the book. However, comments and suggestions for further improvements to the book and supporting software are welcome and can be communicated to us through the publisher. The authors would also like to express their gratitude to Prof. Akram laved, Dean, Faculty of Science, UET, Taxila, for his many useful comments received to improve the quality of this book and particularly to Dr. Jamil Sarwar, Director, BIIT, Rawalpindi, for providing necessary facilities to accomplish this reviewing exercise.

In closing, we are also grateful to our families for their continued patience and understanding during the review effort.

The Bhattis
Islamabad
May, 2002

Preface To The First Edition

The importance of Numerical Analysis to the scientists and engineers is now widely acknowledged. In the book world, there is no dearth of good books on numerical analysis written by foreign authors but the majority of these books are not available in this country. I have written this book to meet the long-felt need of indigenous students.

The main feature of the present text is to introduce numerical methods – covering the syllabi of various universities, colleges and other institutes, where this subject is being taught as a first course. In writing such an elementary book, I have inevitably been confronted by the problem of selection of material, which covers to a great extent the syllabi of the concerned institutes. Naturally, some will disagree with me over this choice of selection. I respect their prerogative. However, I shall be relieved if it is felt that the topics included do provide a reasonably solid background to the student's training and one from which he can easily proceed to further advanced courses in the subject.

The book is designed for a one-semester course in numerical analysis and consists of eight chapters. Each chapter includes a large number of thoroughly explained examples and problems of various complexity. These problems are very necessary and the students should work them out carefully. Each question has been designed to test the student's understanding of a particular formula. The answers of these problems are given at the end of the book. Proofs of formulas are included only where these are reasonably easy to follow, but the formulas are mentioned without proofs in the more difficult cases. It has been tried to keep the explanation straightforward and practically-oriented. The minimum prerequisite for using this book is elementary calculus (including some exposure to series and partial derivatives), linear algebra (determinant an matrices) and differential equations. It is also assumed that the student has taken a programming course in one of the computer languages. Fortran 77, which continues to be an excellent computer language for a wide variety of mathematical problems, is used in this book. Computer programs are given at appropriate places in the text.

No book emerges fully formed from an author's forehead. I would like to acknowledge the inspiration and encouragement I received from my colleagues and the help of many students who worked with early versions of the manuscript and checked exercise solutions and text examples.

The responsibility for any errors, omissions or lack of clarity naturally remains with me. I would appreciate having any such omissions, oversights or needed corrections called to my attention so that they can be implemented for improving the quality of this book. I would also like to thank Mr. Ghulam Shabir Qureshi and Syed Akbar Shah for their help in turning rough drafts into a beautifully prepared final manuscript.

I would like to express my gratitude to the National Book Council of Pakistan for accepting the manuscript of this book under the Creative Writer's Scheme. I also wish to thank the anonymous referees who reviewed the manuscript.

Above all, I wish to thank my family, without whose encouragement, patience and sacrifice this book would not have been completed.

Saeed A. Bhatt
Islamabad
May, 1990

Contents

Chapter 3

Interpolation

3.1 INTRODUCTION

Suppose we are given a table based on certain values of x and the corresponding values of a function f(x):

x	0	1	2	3	...	100
f(x)	10	85	90	98	...	125

The values in the table can be obtained by an experiment or generated if we know the function f(x). The process of computing an approximation value of the function at some point within the range (0, 100), but not in the table of data, is called **interpolation**. If the value of x lies outside the range, the process of estimating the value is called **extrapolation**.

Error of extrapolation increases as the point of interest goes farther from the data points. If a higher order interpolation is used for extrapolation without theoretical ba errors may increase rapidly as the order of polynomial increases. Application of extrapolation may be seen in various sections of this book: for instance, see the Newton-Cotes open integration formulas, the Romberg's integration method and the predictor-corrector methods.

In most of this chapter, we limit the interpolating function to be a **polynomial.** Interpolation has many applications in approximation theory, numerical differentiation, numerical integration, numerical solutions of ordinary and partial differential equations, and for making computer drawn curves to pass through specified points.

We are now going to describe several methods, in each case some kind of advice is given as to the circumstances under which the method should be applied.

3.1.1 Choice of a Suitable Interpolation Formula

The following are considered in choosing a method for interpolating polynomials:

a) Whether the given points x_i are equally spaced.

b) Whether the interpolation is desired towards the beginning, centre or end of a difference table.

3.1.2 Checking the Interpolated Value

The next is the question of checking the interpolated value. A single interpolation is not easy to check. One possibility is to repeat the interpolation using a different formula, but this will be more than double the labour, since the first-interpolation is usually done by the easiest formula. When possible a functional relationship such as $e^{-x} = \frac{1}{e^x}$ is a better check. This still requires two-interpolations but since they involve different tables, the formula may be used for both.

3.1.3 Type of Interpolation Formulas for Equally-Spaced Data Points

The following three types of interpolation formulas are used for equally-spaced data points:

(a) **Newton's forward difference interpolation formula.** It uses differences near the beginning of the difference table.

(b) **Newton's backward difference interpolation formula.** It uses differences near the end of the difference table.

(c) **Central difference interpolation formulas.** These formulas employ differences in the centre of the difference table. The following central difference formulas are commonly used:

i) Stirling's formula

ii) Bessel's formula

iii) Everett's formula

iv) Gauss forward and backward formulas

3.1.4 Type of Interpolation Formulas for Unequally-Spaced Data Points

The following formulas may be used for unequally-spaced data points:

i) Newton's divided difference interpolation formula;

ii) Lagrange's formula;

iii) Aitken's formula; and

iv) Hermite's formula

We shall describe only Lagrange and Aitken formulas, because they are suitable for both, equally and unequally-spaced data points. The above formulas can also be employed for extrapolation; however, the error may increase rapidly the farther we extrapolate from the given values. With the widespread use of computers tabular interpolation has lost much of its importance. The methods under the present category have been widely used.

3.2 NEWTON'S FORWARD DIFFERENCE INTERPOLATION FORMULA

The most basic formula for interpolation with equidistant points is **Newton's forward difference interpolation** (sometimes also called the **Gregory-Newton**) **formula.**

Given a set of n pairs of values:

$$(x_0, f_0), (x_1, f_1), (x_2, f_2), \ldots, (x_n, f_n).$$

We shall derive this formula with the help of two difference operators, E and Δ.

The function to be estimated is written as:

$$f_p = E^p f_0 = (1+\Delta)^p f_0 \qquad \ldots (3.1)$$

Expanding $(1+\Delta)^p$, we have

$$f_p = \left\{1 + p\Delta + \frac{1}{2!}p(p-1)\Delta^2 + \frac{1}{3!}p(p-1)(p-2)\Delta^3 + \ldots + \frac{1}{n!}p(p-1)(p-2)\cdots(p-n+1)\Delta^n\right\} f_0$$

$$= f_0 + p\Delta f_0 + \frac{1}{2!}p(p-1)\Delta^2 f_0 + \frac{1}{3!}p(p-1)(p-2)\Delta^3 f_0 + \ldots + \frac{1}{n!}p(p-1)(p-2)\cdots(p-n+1)\Delta^n f_0 \qquad \ldots (3.2)$$

where $p = \dfrac{(x_p - x_0)}{h}$, obtained from $x_p = x_0 + ph$.

The coefficient of $\Delta^n f_0$ will contain p^n which is an nth degree polynomial.

Remarks

i) This formula is used for interpolation near the beginning of a difference table, but in odd cases, it may also be applied in other parts of the table by suitably shifting the origin. Shifting the origin does not affect the result, but on the other hand it may result in a simpler formula, which is less prone to error.

ii) This formula is usually applicable for $0 < p < 1$. When working with differences, we can select any value of x in the table to be labeled as x_0. This is mostly done to keep p within the range.

Example 1 a) Compute the difference table for the following set of data-points:

x	.00	.25	.50	.75	1.00
f(x)	.0000	.2763	.5205	.7112	.8427

b) Use Newton's forward difference formula to pass a fourth degree polynomial through the above data.

c) Use the above polynomial to interpolate for f (0.125).

Solution a) The forward differences are computed as follows:

x	f(x)	Δ	Δ^2	Δ^3	Δ^4
0.00	.0000				
		2763			
0.25	.2763		–321		
		2442		–214	
0.50	.5205		–535		157
		1907		–57	
0.75	.7112		–593		
		1315			
1.00	.8427				

b) $h = x_1 - x_0 = .25 - .00 = .25$

$x_p = x_0 + ph = 0.125$

$$p = \frac{(x_p - x_0)}{h} = \frac{(.125 - .00)}{.25} = 0.5$$

Since the calculated value of p lies in the range (0, 1), it makes the forward difference formula applicable. The values to be used in formula (3.2) are underlined in the above table.

$$f_p = f_0 + p\Delta f_0 + \frac{p(p-1)}{2!}\Delta^2 f_0 + \frac{p(p-1)(p-2)}{3!}\Delta^3 f_0 + \frac{p(p-1)(p-2)(p-3)}{4!}\Delta^4 f_0$$

$$= .000 + p \times 0.2763 + \frac{(p^2 - p)}{2} \times -0.0321 + \frac{(p^3 - 3p^2 - p)}{6} \times -0.0321$$

$$+ \frac{(p^4 - 6p^3 + 11p^2 - 6p)}{6} \times 0.0157$$

$$= .2763\,p - .0125\,p^2 + .0008\,p^3 - .0006\,p^4$$

c) Inserting $p = 0.50$ in the above polynomial, we get

$$f_p = .2763 \times .50 - .0125 \times (.50)^2 + .0008 \times (.50)^3 - .0006 \times (.50)^4$$

$$= .13815 - .00313 + .0001 - .00004 = 0.1351$$

The students should be careful not to think of the answer 0.1351 as the correct answer. It is an estimate of the correct answer based on the assumption that f(x) is a fourth-degree polynomial.

Example 2 Use Newton's forward difference formula to interpolate the value for f(1.75) from the following data:

(0.5, 0.000), (1.0, 1.357), (1.5, 2.000), (2.0, 2.625), and (2.5, 4.000).

Solution The difference table is as follows:

x	f(x)	Δ	Δ^2	Δ^3	Δ^4
0.5	0.000				
		1375			
1.0	1.357		−750		
		625		750	
1.5	$\underline{2.000}$		0		0
		$\underline{625}$		750	
2.0	2.625		$\underline{750}$		
		1375			
2.5	4.000				

$x_p = 1.75;\ x_0 = 0.5;\ x_1 = 1.0;$

$h = x_1 - x_0 = 0.5$

$$p = \frac{(x_p - x_0)}{h} = \frac{(1.75 - 0.5)}{0.5} = 2.5$$

As p (= 2.5) does not lie between 0 and 1, we cannot use the origin to be $x_0 = 0.5$. Let us shift the origin to 1.0. Then, $p = \dfrac{(1.75-1)}{0.5} = 1.5$. We cannot use $x_1 = 1$ as the origin because still p > 1. Let us shift the origin to $x_0 = 1.5$. $p = \dfrac{(1.75-1.5)}{0.5} = \dfrac{0.25}{0.5} = 0.5$. So, we can use $x_0 = 1.5$ as the origin because the calculated value of p < 1.

The entries used in this case are underlined in the difference table. The reduced form of Newton's formula is as follows:

$$f_p = f_0 + p\Delta f_0 + \frac{p(p-1)}{2}\Delta^2 f_0$$

Inserting the values in the above reduced formula, we get,

$$f_p = 2.000 + 0.5 \times 0.625 + \frac{0.5(0.5-1)}{2} \times 0.750$$

$$= 2.000 + 0.313 - 0.094 = 2.219$$

Example 3 Write a computer program to implement Newton's forward difference interpolation formula. Use the following data for testing:

x	2	4	6	8	10	12	14
f(x)	23	93	259	569	1071	1813	2843

Estimate f (2.58).

Solution For computer program, see Example 4. However, the computer output for this example is given below:

Compute Output:

X	F (X)	1ST	2ND	3RD	4TH
2.00	23.00				
		70.00			
4.00	93.00		96.00		
		166.00		48.00	
6.00	259.00		144.00		.00
		310.00		48.00	
8.00	569.00		192.00		.00
		502.00		48.00	
10.00	1071.00		240.00		.00
		742.00		48.00	
12.00	1813.00		288.00		
		1030.00			
14.00	2843.00				

ANSWER = 36.23

3.3 NEWTON'S BACKWARD DIFFERENCE INTERPOLATION FORMULA

We shall derive Newton's backward difference formula using the difference operators E and ∇.

We know that, $f_p = E^p f_0 = (1-\nabla)^{-p} f_0$ ... (3.3)

Expanding $(1-\nabla)^{-p}$, we obtain,

$$f_p = \left\{1 + p\nabla + \frac{p(p+1)}{2!}\nabla^2 + \frac{p(p+1)(p+2)}{3!}\nabla^3 + \cdots + \frac{p(p+1)\cdots(p+n-1)}{n!}\nabla^n\right\}f_0$$

$$= f_0 + p\nabla f_0 + \frac{p(p+1)}{2!}\nabla^2 f_0 + \frac{p(p+1)(p+2)}{3!}\nabla^3 f_0 + \cdots$$

$$+ \frac{p(p+1)\cdots(p+n-1)}{n!}\nabla^n f_0 \qquad \ldots (3.4)$$

This is called **Newton's backward difference interpolation** (also called the **Gregory-Newton) formula.**

Remarks

i) This formula is used toward the end of the difference table but can also be applied in other parts of the table by suitably shifting the origin. This situation occurs whenever a table is being extended, for example, when the solution to a differential equation is being obtained by a step-by-step method.

ii) The formula is valid for $0 < p < 1$.

Example 4 (a) Using Newton's backward difference formula, compute f (11.8) from the following data:

x	2	4	6	8	10	12	14
f(x)	23	93	259	569	1071	1813	2843

(b) Write a computer program to implement Newton's backward difference interpolation formula.

Solution (a) The backward differences are computed in the following table:

x	f(x)	∇	∇^2	∇^3	∇^4
2	23				
		70			
4	93		96		
		166		48	
6	256		144		0
		310		48	
8	569		192		0
		502		48	
10	1071		240		0
		442		48	
12	1813		288		
		1030			
14	4843				

$x_p = 11.8$

Taking $x_0 = 14$, $p = \frac{(11.8-14)}{2} = -1.1$. As the calculated value is outside its acceptable range, we cannot accept the origin to be at $x_0 = 14$. The suitable origin may be $x_0 = 10$, which gives $p = \frac{11.8-10}{2} = \frac{1.8}{2} = 0.9$. The entries used for the backward difference formula are underlined in the above difference table. Substituting these values in formula (3.4), we get:

$$f_p = 1071 + 0.9 \times 502 + \frac{0.9(0.9+1)}{2} \times 192 + \frac{0.9(0.9+1)(0.9+2)}{6} \times 48$$

$$= 1071 + 451.8 + 164.16 + 39.67 = 1727$$

(b) This program can be used for Newton's forward and backward interpolation formulas. It is done via a main menu. Menu Choice 1 is for the forward difference formula, while Menu Choice 2 is for the backward difference formula.

Computer Program:

```
# include<iostream.h>
# include<conio.h>
# include<process.h>

float interval, x0, p, array [20][20] = {0.0};
int no, col, x,y;
void difftable( )
 {
   cout<<"\tDIFFERENETTABLE";
   cout<<"\n\n\tENTER THE FIRST VALUE : "; cin>>array[0][0];
   cout<<"\n\tENTER THE INTERVAL : "; cin>>interval;
   cout<<"\n\tENTER TOTAL NO. OF X : "; cin>>no;

   for(int i=1; i<no; i++)
      {
        array[i][0]=array[i-1][0]+interval;
      }

   cout<<"\n\tENTER FUNCTIONAL VALUES : \n";
   for(i=0;i<no;i++)
      {
        cout<<"\tX("<<i<<") = ";cin>>array[i][1];
      }
```

```
cout<<"\n\tHOW MANY COLUMNS ARE REQUIRED :  "; cin>>col;
for(i=2; i<=(col+2); i++)
    {
      for(int j=0; j<=(no-i); j++)
        {
          array[j][i]=array[j+1][i-1]-array[j][i-1];
        }
    }

clrscr( );
cout<<"\t\tDIFFERENCE TABLE\n";
cout<<" X          F(X)  ";
for(j=1;i<=col;i++)
    {
      cout<<"     col     "<<i;
    }

cout<<"\n";

for(i=0;i<no;i++)
    {
      cout<<"       "<<array[i][0]<<"\n\n";
    }

x=8; y=3;
for(i=1;i<=(col+1);i++)
  {
    gotoxy(x,y);
    for(int j=0; j<=(no-i); j++)
      {
        cout<<array[j][i];
        y+=2;
        gotoxy(x,y);
      }
    x+=9; y=i+3;
  }
}

void findx( )
 {
   float xp;
   cout<<"\n\t XP FOR WHICH VALUE OF F(X) IS REQUIRED :     "; cin>>xp;
   int i=0;
```

```
    while(((xp-array[i][0])/interval>1)&&(I<no))
      {
        i++;
      }
    x0=i;
    p=(xp-array[x0][0])/interval;
  }

void nford( )
  {
    findx( );

    cout<<"\n\ntanswer = ";
    cout<<(array[x0][1]+(p*array[x0][2]+(p*(p-1) /2 * array[x0][3])
    +p*(p-1)*(p-2) /6 * array[x0][4])+(p*(p-1)*(p-2)*(p-3) /24 * array[x0][5]);
  }

void nback( )
  {
    findx( );
    cout<<"\n\n\tanswer = ";
    cout<<(array[x0][1]+(p*array[x0-1][2]+(p*(p+1) /2 * array[x0-2][3])
    +p*(p+1)*(p+2) /6 * array[x0-3][4])+(p*(p+1)*(p+2)*(p+3) /24 * array[x0-4][5]);
  }

void main (void)
  {
    clrscr ( ); difftable ( ); getch ( );

    int choice;
    while (1)
      {
        clrscr ( );
        cout<<"\n\n\t\tMAIN MENU";
        cout<<"\n\n\tFORWARD DIFFERENCE INTERPOLATION FORMULA --- 1";
        cout<<"\n\n\tBACKWARD DIFFERENCE INTERPOLATION FORMULA --- 2";
        cout<<"\n\n\tTO EXIT --------------------------------------------------------";
        cout<<"\n\n\n\tENTER YOUR CHOICE : ";
        cin>>choice;
        switch(choice)
          {
            case 1:clrscr ( );nford( );getch( );break;
            case 2:clrscr ( );nback( );getch( );break;
            case 3:exit(0)
          }
      }
}
```

Computer Output

```
DIFFERENCE TABLE
ENTER THE FIRST VALUE : 2
ENTER THE INTERVAL : 2
ENTER TOTAL NO. OF X : 7
ENTER FUNCTIONAL VALUES:
    X(0) = 23
    X(1) = 93
    X(2) = 259
    X(3) = 569
    X(4) = 1071
    X(5) = 1813
    X(6) = 2843
HOW MANY COLUMNS ARE REQUIRED : 4
```

DIFFERENCE TABLE

X	F(X)	col 1	col 2	col 3	col 4
2	23				
		70			
4	93		96		
		166		48	
6	259		144		0
		310		48	
8	569		192		0
		502		48	
10	1071		240		0
		742		48	
12	1813		288		
		1030			
14	2843				

MAIN MENU

FORWARD DIFFERENCE INTERPOLATION FORMULA --- 1

BACKWARD DIFFERENCE INTERPOLATION FORMULA --- 2

TO EXIT -- 3

ENTER YOUR CHOIC : 1

XP, FOR WHICH VALUE OF F(X) IS REQUIRED : 2.58

A N S W E R = 36.233509

ENTER YOUR CHOICE : 2

XP, FOR WHICH VALUE OF F(X) IS REQUIRED : 11.8

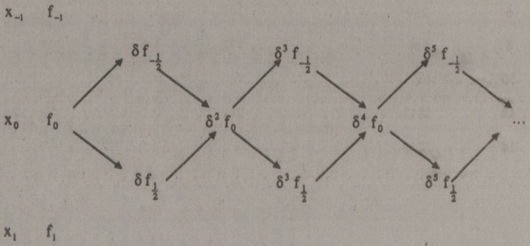
A N S W E R = 1726.63208

3.4 INTERPOLATION WITH CENTRAL DIFFERENCE FORMULAS

The two formulas by Newton are used occasionally and almost exclusively at the beginning or at the end of a table. More important are formulas which make use of central differences, a whole series of such formulas with slightly different properties can be constructed. In this section, we shall mention without proofs some well-known central difference formulas. The structure of all these formulas can easily be demonstrated by sketching a difference scheme where different quantities are represented by points. If the column to the left stands for the function values, then we have the first differences and so on.

3.4.1 Stirling's Interpolation Formula

Stirling's formula follows the path through the difference table given below:



It is expressed as follows:

$$f_p = f_0 + \frac{1}{2}p\left(\delta f_{-\frac{1}{2}} + \delta f_{\frac{1}{2}}\right) + \frac{1}{2!}p^2\delta^2 f_0 + \frac{p(p^2-1)}{2\times 3!}\left(\delta^3 f_{-\frac{1}{2}} + \delta^3 f_{\frac{1}{2}}\right)$$

$$+ \frac{p^2(p^2-1)}{4!}\delta^4 f_0 + \frac{p(p^2-1)(p^2-4)}{2\times 5!}\left(\delta^5 f_{-\frac{1}{2}} + \delta^5 f_{\frac{1}{2}}\right)$$

$$+ \frac{p^2(p^2-1)(p^2-4)}{6!}\delta^6 f_0 + \cdots \qquad \ldots \quad (3.5(a))$$

It can also be written in another form as:

$$f_p = f_0 + p\mu\delta f_0 + \frac{1}{2!}p^2\delta^2 f_0 + \frac{1}{3!}p(p^2-1)\mu\delta^3 f_0 + \frac{1}{4!}p^2(p^2-1)\delta^4 f_0$$

$$+ \frac{1}{5!}p(p^2-1)(p^2-4)\mu\delta^5 f_0 + \frac{1}{6!}p^2(p^2-1)(p^2-4)\delta^6 f_0 + \cdots$$

$$\ldots \quad (3.5(b))$$

Example 5 Use Stirling's Interpolation formula to find f(1.62) from the following table:

x	1.2	1.4	1.6	1.8	2.0
f(x)	5.6464	6.44218	7.17356	7.83327	8.41471

Solution: The difference table for is as follows:

x	f (x)	δ	δ^2	δ^3	δ^4
1.2	5.6464				
		79576			
1.4	6.44218		– 6438		
		73138		– 729	
x_0 = 1.6	7.17356		– 7167		69
		65971		– 660	
1.8	7.83327		– 7827		
		58144			
2.0	8.41471				

Taking $x_0 = 1.6$, $h = x_1 - x_0 = 1.8 - 1.6 = .2$

$$P = \frac{(1.62 - 1.6)}{0.2} = \frac{0.02}{0.2} = 0.1$$

Inserting the values in Stirling's formula 3.5(a), we get,

$$f_p = 7.17356 + \frac{1}{2} \times 0.1\,(.73138 + .65971) + \frac{1}{2} \times 0.1 \times 0.1 \times -.7167$$

$$+ \frac{0.1(0.1 \times 0.1 - 1)}{12} \times (-.00729 - .00660)$$

$$+ \frac{0.1 \times 0.1(0.1 \times 0.1 - 1)}{24} \times .00069$$

$$= 7.17356 + 0.06955 - 0.00036 + 0.00011 - 0.00000$$

$$= 7.24286$$

3.4.2 Bessel's Interpolation Formula

Bessel's formula follows the path through the difference table:

$$\begin{tikzcd}
x_0 & f_0 \arrow[dr] & & \delta^2 f_0 \arrow[dr] & & \delta^4 f_0 \\
& & \delta f_{\frac{1}{2}} \arrow[ur] \arrow[dr] & & \delta^3 f_{\frac{1}{2}} \arrow[ur] \arrow[dr] & \\
x_1 & f_1 & & \delta^2 f_1 \arrow[ur] & & \delta^4 f_1
\end{tikzcd}$$

Bessel's formula is expressed as follows:

$$f_p = f_0 + p\delta f_{\frac{1}{2}} + \frac{p(p-1)}{2.2!}\left(\delta^2 f_0 + \delta^2 f_1\right) + \frac{p(p-1)\left(p-\frac{1}{2}\right)}{3!}\delta^3 f_{\frac{1}{2}}$$

$$+ \frac{(p+1)p(p-1)(p-2)}{2.4!}\left(\delta^4 f_0 + \delta^4 f_1\right) + \cdots \qquad \ldots \quad (3.6(a))$$

It can also be written in another form as:

$$f_p = f_0 + p\,\delta f_{\frac{1}{2}} + \frac{1}{2!}p(p-1)\mu\delta^2 f_{\frac{1}{2}} + \frac{1}{3!}p(p-1)\left(p-\frac{1}{2}\right)\mu\,\delta^3 f_{\frac{1}{2}}$$

$$+ \frac{1}{4!}(p+1)p(p-1)(p-2)\mu\,\delta^4 f_{\frac{1}{2}} + \cdots \qquad \ldots \quad (3.6(b))$$

3.4.3 Everett's Interpolation Formula

Everett's formula follows the path through the difference table:

$$\begin{array}{cccccccc}
x_0 & f_0 & \text{———} & \delta^2 f_0 & \text{———} & \delta^4 f_0 & \cdots \\
x_1 & f_1 & \text{———} & \delta^2 f_1 & \text{———} & \delta^4 f_1 & \cdots
\end{array}$$

Everett's formula is expressed as follows:

$$f_p = q f_0 + \frac{q(q^2-1)}{3!}\delta^2 f_0 + \frac{q(q^2-1)(q^2-4)}{5!}\delta^4 f_0 + \cdots$$

$$+ p f_1 + \frac{p(p^2-1)}{3!}\delta^2 f_1 + \frac{p(p^2-1)(p^2-4)}{5!}\delta^4 f_1 + \cdots \quad \ldots \quad (3.7)$$

where $q = 1 - p$.

3.4.4 Gaussian Interpolation Formula

There are the following two such formulas:

a) Gauss Forward Interpolation Formula

b) Gauss Backward Interpolation Formula

Let us discuss them one by one.

a) Gauss Forward Interpolation Formula

This formula follows the zigzag path as indicated in the difference table:

$$\begin{array}{ccccccccccc}
x_0 & f_0 & & & \delta^2 f_0 & & & & \delta^4 f_0 & & \\
 & & \searrow & & \nearrow & \searrow & & \nearrow & & \searrow & \\
 & & & \delta f_{\frac{1}{2}} & & & \delta^3 f_{\frac{1}{2}} & & & & \delta^5 f_{\frac{1}{2}} \\
x_1 & f_1 & & & & & & & & &
\end{array}$$

Gauss forward formula is expressed as follows:

$$f_p = f_0 + p\delta f_{\frac{1}{2}} + \frac{p(p-1)}{2!}\delta^2 f_0 + \frac{p(p+1)(p-1)}{3!}\delta^3 f_{\frac{1}{2}}$$

$$+ \frac{(p+1)p(p-1)(p-2)}{4!}\delta^4 f_0 + \cdots \quad \ldots \quad (3.8)$$

b) Gauss Backward Interpolation Formula

This formula follows the zigzag path as indicated in the following difference table:

$$\begin{array}{ccccccccccc}
x_{-1} & f_{-1} & & & & & & & & & \\
 & & & \delta f_{-\frac{1}{2}} & & & \delta^3 f_{-\frac{1}{2}} & & & \delta^5 f_{-\frac{1}{2}} \\
 & & \nearrow & & \searrow & \nearrow & & \searrow & \nearrow & \\
x_0 & f_0 & & & \delta^2 f_0 & & & \delta^4 f_0 & &
\end{array}$$

Gauss backward formula is expressed as follows:

$$f_p = f_0 + p\delta f_{-\frac{1}{2}} + \frac{p(p+1)}{2!}\delta^2 f_0 + \frac{p(p+1)(p-1)}{3!}\delta^3 f_{-\frac{1}{2}} + \frac{(p+1)p(p-1)(p-2)}{4!}\delta^4 f_0 + \cdots \qquad \ldots \quad (3.9)$$

Note that if we take the mean of Gauss forward and backward formulas, we get Stirling's interpolation formulas.

As mentioned earlier that for most purposes formulas using central differences are to be preferred. However, some remarks about their use are in order. The two Gaussian interpolation formulas are of interest almost exclusively from theoretical standpoint. Stirling's formula is suitable for values of small values of p, for example, $-\frac{1}{4} \le p \le +\frac{1}{4}$, and Bessel's formula which is probably the most used of all interpolation formulas, is suitable for values of p not too far from $\frac{1}{2}$, for example, $-\frac{1}{4} \le p \le \frac{1}{4}$. Everett's formula which is simple and fast is perhaps the one which is most generally useful and further because even differences are used together with the function values.

Example 6 Given the following table of values:

x	2.2	2.6	3.0	3.4	3.8	4.2	4.6
f(x)	.374607	.438371	.500000	.559193	.615661	.669131	.719340

a) Construct the difference table including differences up to 4th order only.

b) Interpolate f(3.64) using the following formulas centred at x = 3.4:

(i) Stirling (ii) Bessel (iii) Everett and (iv) Gaussian forward and backward.

Solution (a) **Difference Table**

x	f (x)	δ	δ^2	δ^3	δ^4
2.2	.374607				
		63764			
2.6	.438371		− 2135		
		61629		− 301	
3.0	.500000		− 2436		12
		59193		− 289	
$x_0 = 3.4$	.559193		− 2725		16
		56468		− 273	
3.8	.615661		− 2998		10
		53470		− 263	
4.2	.669131		− 3261		
		50209			
4.6	.719340				

b) (i) **Stirling's Formula**

$x_p = 3.64, \quad x_0 = 3.4, \; h = 0.4$

$$P = \frac{(x_p - x_0)}{h} = \frac{(3.64 - 3.4)}{0.4} = 0.6$$

Substituting values in formula (3.5(a)), we get,

$$f_p = .559193 + \frac{6}{2}\,(.059193 + .056468) + \frac{.6 \times .6}{2} \times - .002725$$

$$+ \frac{0.6(0.6 \times 0.6 - 1)}{12}\,(-.000289 - .000273)$$

$$+ \frac{0.6 \times 0.6(0.6 \times 0.6 - 1)}{24} \times .000016$$

$$= .559193 + .034698 - .000491 + .000018 + .000000$$

$$= 0.593418$$

(ii) Bessel's Formula

Substituting values in formula (3.6(a)), we get,

$$f_p = .559193 + 0.6 \times .056468 + \frac{.6(.6-1)}{4}(-.002725 - .002998)$$

$$+ \frac{0.6(0.6-1)\left(0.6-\frac{1}{2}\right)}{6} \times -0.000273$$

$$+ \frac{(0.6+1)0.6(0.6-1)(0.6-2)}{48} \times (.000016 + .000010)$$

$$= .559193 + .033881 + .000343 + .000001 + .000000$$

$$= 0.593418$$

(iii) Everett's Formula

$$q = 1 - .6 = .4$$

Substituting values in formula (3.7), we get,

$$f_p = 0.4 \times .559193 \; \frac{.4(.4 \times .4 - 1)}{6} \times -0.002725$$

$$+ \frac{.4(.4 \times .4 - 1)(.4 \times .4 - 4)}{120} \times 0.000016$$

$$+ .6 \times 0.615661 + \frac{0.6(0.6 \times .6 - 1)}{6} \times -0.002998$$

$$+ \frac{0.6(0.6 \times .6 - 1)(0.6 \times .6 - 4)}{120} \times -0.000010$$

$$= .223677 + .000153 + .000000 + .369397 + .000192 + .000000$$

$$= 0.593419$$

(iv) (a) Gauss Forward Formula

Substituting values in formula (3.8), we get,

$$f_p^- = .559193 + 0.6 \times .056468 + \frac{.6(0.6-1)}{2} \times -.002725$$

$$+ \frac{0.6(0.6+1)(0.6-1)}{6} \times -.000273$$

$$+ \frac{(.6+1) \times 0.6(.6-1)(0.6-2)}{24} \times .000016$$

$$= .559193 + .033881 + .000327 + .000017 + .00000$$

$$= 0.593418$$

(b) Gauss Backward Formula

Substituting values in formula (3.9), we get,

$$f_p = .559193 + 0.6 \times .059193 + \frac{.6(0.6+1)}{2} \times -.002725$$
$$+ \frac{0.6(0.6+1)(0.6-1)}{6} \times -.000289$$
$$+ \frac{0.6(.6+1)(.6-1)(0.6-2)}{24} \times .000016$$

$$= .559193 + .035516 + .001308 + .000018 + .000000$$

$$= 0.593419$$

3.5 LAGRANGE'S FORMULA

It was mentioned in the previous sections that difference table could be used for interpolation, but this was restricted to the case of function values at equidistant intervals.

To introduce the basic idea behind the Lagrange's formula, consider the following:

Given the data points $x_0, x_1, \ldots, x_n$ (may or may not be equidistant), the problem is to find an nth degree polynomial f(x) using Lagrange's formula.

Lagrange's formula can be derived by writing:

$$\begin{aligned} f(x) = \; & A_0(x-x_1)(x-x_2)\ldots(x-x_n) \\ & + A_1(x-x_0)(x-x_2)\ldots(x-x_n) \\ & + A_2(x-x_0)(x-x_1)\ldots(x-x_n) \\ & \vdots \\ & + A_i(x-x_0)\ldots(x-x_{i-1})(x-x_{i-1})\ldots(x-x_n) \\ & \vdots \\ & + A_n(x-x_0)(x-x_1)\ldots(x-x_{n-1}) \qquad \ldots (3.10) \end{aligned}$$

where $A_0, A_1, \ldots, A_n$ are unknown constants.

If we substitute $x = x_0$ in (3.10), we get,

$$f(x_0) = A_0(x_0 - x_1)(x_0 - x_2)\ldots(x_0 - x_n)$$

$$A_0 = \frac{f(x_0)}{(x_0 - x_1)(x_0 - x_2)\cdots(x_0 - x_n)}$$

Similarly, substituting $x = x_1, x = x_2, \ldots, x = x_n$ respectively in (3.10), we get,

$$A_1 = \frac{f(x_1)}{(x_1 - x_0)(x_1 - x_2)\cdots(x_1 - x_n)}$$

$$A_2 = \frac{f(x_2)}{(x_2 - x_0)(x_2 - x_1)\cdots(x_2 - x_n)}$$

$$A_n = \frac{f(x_n)}{(x_n - x_0)(x_n - x_1)\cdots(x_n - x_{n-1})}$$

Substituting the values of $A_0, A_1, \ldots, A_n$ in (3.10), we get the following formula due to Lagrange:

$$\begin{aligned} f(x) = {} & \frac{(x - x_1)(x - x_2)\cdots(x - x_n)}{(x_0 - x_1)(x_0 - x_2)\cdots(x_0 - x_n)} f(x_0) \\ & + \frac{(x - x_0)(x - x_2)\cdots(x - x_n)}{(x_1 - x_0)(x_1 - x_2)\cdots(x_1 - x_n)} f(x_1) \\ & + \frac{(x - x_0)(x - x_1)\cdots(x - x_n)}{(x_2 - x_0)(x_2 - x_1)\cdots(x_2 - x_n)} f(x_2) \\ & + \ldots \\ & + \frac{(x - x_0)(x - x_1)\cdots(x - x_{n-1})}{(x_n - x_0)(x_n - x_1)\cdots(x_n - x_{n-1})} f(x_n) \qquad \ldots (3.11) \end{aligned}$$

It is obvious that (3.11) is a polynomial of degree n and can be written as:

$$f(x) = L_0(x) f(x_0) + L_1(x) f(x_1) + L_2(x) f(x_2) + \ldots + L_n(x) f(x_n)$$

It can be concisely represented as:

$$f(x) = \sum_{i=0}^{n} L_i(x) f(x_i) \qquad \ldots (3.11(a))$$

where $L_i(x) = \prod_{i=0}^{n}\left(\dfrac{x - x_j}{x_i - x_j}\right)$

$j = 0$

$j \neq i$

Another form of this formula is:

$$f(x) = \sum_{i=0}^{n}\left(\prod_{\substack{j=0\\ j\neq i}}^{n}\left(\frac{x - x_j}{x_i - x_j}\right)\right)f(x_i) \qquad \text{... (3.11(b))}$$

The basic formula, apparently due to Waring, is associated with the name of Lagrange. This is one of the more practical and simpler method to be used on computer; but difficult for hand calculations if data points are more. Evaluation of error is also not easy. It is meant for equispaced or unequispaced data.

Example 7 (a) Fit a polynomial using Lagrange's formula to the following data:

(1, 4), (3, 7), (4, 8) and (6, 11).

(b) Use the polynomial to estimate a value for x = 5.

(c) Write a computer program to implement Lagrange's formula.

Solution (a) The data-points are:

x	1	3	4	6
f(x)	4	7	8	11

Inserting values in Lagrange's formula, we get,

$$f(x) = \frac{(x-3)(x-4)(x-6)}{(1-3)(1-4)(1-6)}\times 4 + \frac{(x-1)(x-4)(x-6)}{(3-1)(3-4)(3-6)}\times 4$$

$$+\frac{(x-1)(x-3)(x-6)}{(4-1)(4-3)(4-6)}\times 8 + \frac{(x-1)(x-3)(x-4)}{(6-1)(6-3)(6-4)}\times 11$$

$$= \frac{-2}{15}\left(x^3 - 13x^2 + 54x - 72\right) + \frac{7}{6}\left(x^3 - 11x^2 + 34x - 24\right)$$

$$+\frac{-8}{6}\left(x^3 - 10x^2 + 27x - 18\right) + \frac{11}{30}\left(x^3 - 8x^2 + 19x - 12\right)$$

$$= \frac{1}{30}\left(2x^3 - 21x^2 + 103x + 36\right)$$

(b) The interpolated value at $x = 5$ is as follows:

$$f(5) = \frac{1}{30}\left(2\times5^3 - 21\times5^2 + 103\times5 + 36\right)$$

$$= \frac{1}{30}\times 276 = 9$$

(c) **Program No. 3** **LAGRANGE'S FORMULA**

```
# include<iostream.h>
# include<iostream.h>

void main (void)
  {
    float table[10][2], xp,temp,ans=0.0;
    int no, y=0,a=7,i,j;

    cout<<"How Many Values Of X    :    ";
    cin>>no;
    cout<<"\nEnter The Values Of X and f(x)\n";
    cout<<"\n\t     x            |            f(x)";
    cout<<"\n\t-----------------------------------";

    for(i=0;i<no;i++)                                    // Input of X & Fx
      {
        gotoxy(11,a);
        cin>>table[i][y];
        gotoxy(21,a);
        cin>>table[i][y+1];
        a++;
      }

   cout<<"\nEnter The Value Of X    :    ";
   cin>>xp;

   for(j=0;j<no;j++)                                     // calculation of formula
     {
       temp=1;
       for(i=0;i<no;i++)
         if(i!=j)
```

```
        temp*=((xp-table[i][0] / (table[j][0]-[i][0]));

      ans+=temp*table[j][1];
    }
   cout<<"\nA N S W E R =          : "<<ans;      //output
}
```

Computer Output

How Many Values of X : 4

Enter the Values of x and f(x)

x	f(x)
1	4
3	7
4	8
6	11

Enter The Value of X : 5

A N S W E R : 9.2

3.6 ITERATIVE INTERPOLATION METHOD

Like Lagrange's method, this formula is also more suitable for computer application and its use is also not limited to only uniformly spaced data. The iterative interpolation process is based on the repeated application of simple (linear) interpolation method. This method is due to **Aitken.**

Consider the following data points (equally or unequally spaced):

x	x_0	x_1	x_2	x_3	...	x_n
f(x)	f_0	f_1	f_2	f_3	...	f_n

In order to estimate the value of the function f corresponding to any value of x, we proceed as follows:

Let $f_0 = f(x_0)$

$f_1 = f(x_1)$

$\vdots$

$$f_k = f(x_k)$$

$$\vdots$$

$$f_n = f(x_n)$$

also let $f(x \mid x_0, x_1, \ldots, x_n)$ denote the unique polynomial of degree n coinciding with f(x) at $x_0, x_1, \ldots, x_n$.

Hence, $f(x \mid x_0) = f(x_0)$

$$f(x \mid x_1) = f(x_1)$$

$$\vdots$$

$$f(x \mid x_n) = f(x_n)$$

Also, $f(x \mid x_0, x_1) = \dfrac{1}{(x_1 - x_0)} \begin{vmatrix} x - x_0 & f(x \mid x_0) \\ x - x_1 & f(x \mid x_1) \end{vmatrix}$

$$= \frac{1}{x_1 - x_0} \begin{vmatrix} x - x_0 & f_0 \\ x - x_1 & f_1 \end{vmatrix}$$

$$= \frac{1}{(x_1 - x_0)} \{(x - x_0) f_1 - (x - x_1) f_0\}$$

$$f(x \mid x_0, x_2) = \frac{1}{x_2 - x_0} \begin{vmatrix} x - x_0 & f_0 \\ x - x_2 & f_2 \end{vmatrix}$$

$$= \frac{1}{(x_2 - x_0)} \{(x - x_0) f_2 - (x - x_2) f_0\}, \text{ etc.}$$

Similarly, $f(x \mid x_0, x_1, x_2) = \dfrac{1}{(x_2 - x_1)} \begin{vmatrix} x - x_1 & f(x \mid x_0, x_1) \\ x - x_2 & f(x \mid x_0, x_2) \end{vmatrix}$

$$f(x \mid x_0, x_1, x_3) = \frac{1}{(x_3 - x_1)} \begin{vmatrix} x - x_1 & f(x \mid x_0, x_1) \\ x - x_3 & f(x \mid x_0, x_3) \end{vmatrix}$$

and $f(x \mid x_0, x_1, x_4) = \dfrac{1}{(x_4 - x_1)} \begin{vmatrix} x - x_1 & f(x \mid x_0, x_1) \\ x - x_4 & f(x \mid x_0, x_4) \end{vmatrix}$

denote polynomials of degree ≤ 2 that pass through the four points $(x_0, f_0), (x_1, f_1), (x_2, f_2)$; $(x_0, f_0), (x_1, f_1), (x_3, f_3)$; and $(x_0, f_0), (x_1, f_1), (x_4, f_4)$, respectively, whereas $f(x \mid x_0, x_1, x_2, x_3) = \dfrac{1}{x_3 - x_2}\begin{vmatrix} x - x_2 & f(x \mid x_0, x_1, x_2) \\ x - x_3 & f(x \mid x_0, x_1, x_3) \end{vmatrix}$

denotes polynomial of degree ≤ 3 and so on.

Continuing the above process, we can develop the interpolating polynomials to any degree we want:

$$f(x \mid x_0, x_1, x_2, x_3, x_4) = \frac{1}{(x_4 - x_3)}\begin{vmatrix} x - x_3 & f(x \mid x_0, x_1, x_2, x_3) \\ x - x_4 & f(x \mid x_0, x_1, x_2, x_4) \end{vmatrix}$$

$$f(x \mid x_0, x_1, x_2, x_3, x_5) = \frac{1}{(x_5 - x_3)}\begin{vmatrix} x - x_3 & f(x \mid x_0, x_1, x_2, x_3) \\ x - x_5 & f(x \mid x_0, x_1, x_2, x_5) \end{vmatrix}$$

The following table illustrates the arrangement of the work needed to construct $f(x \mid x_0, x_1, \ldots, x_n)$:

x_0	$x - x_0$	$f(x \mid x_0)$				
x_1	$x - x_1$	$f(x \mid x_1)$	$f(x \mid x_0, x_1)$			
x_2	$x - x_2$	$f(x \mid x_2)$	$f(x \mid x_0, x_2)$	$f(x \mid x_0, x_1, x_2)$		
x_3	$x - x_3$	$f(x \mid x_3)$	$f(x \mid x_0, x_3)$	$f(x \mid x_0, x_1, x_3)$	$f(x \mid x_0, x_1, x_2, x_3)$	
x_4	$x - x_4$	$f(x \mid x_4)$	$f(x \mid x_0, x_4)$	$f(x \mid x_0, x_1, x_4)$	$f(x \mid x_0, x_1, x_2, x_4)$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

The tabular values are generated row-wise (or column-wise). Since the current value are generated from the previous values that is why this method is often called the **iterative interpolation method** and also named as **Neville's formula.** The rightmost value in the table is the required value of interpolation.

Example 8 (a) Using Aitken's iterative scheme, find the value of log 4.5 from the following values:

x	4.0	4.2	4.4	4.6
f(x)	0.60206	0.62325	0.64345	0.66276

(b) Write a computer program to implement Aitken's method.

Solution (a) $x = 4.5$

Aitken's table is as follows:

$x_0 = 4.0$	$x - x_0 = .5$	0.60206			
$x_1 = 4.2$	$x - x_1 = .3$	0.62325	0.65504		
$x_2 = 4.4$	$x - x_2 = .1$	0.64345	0.65380	0.65318	
$x_3 = 4.6$	$x - x_3 = -.1$	0.66276	0.65264	0.65324	0.65321

$$f(x \mid x_0, x_1) = \frac{1}{(x_1 - x_0)}\begin{vmatrix} x - x_0 & f(x \mid x_0) \\ x - x_1 & f(x \mid x_1) \end{vmatrix}$$

$$= \frac{1}{(4.2 - 4.0)}\begin{vmatrix} .5 & .60206 \\ .3 & .62325 \end{vmatrix}$$

$$= \frac{(.5 \times .62325 - .3 \times .60206)}{0.2}$$

$$= \frac{(.311625 - .180618)}{0.2} = 0.65504$$

$$f(x \mid x_0, x_2) = \frac{1}{(x_2 - x_0)}\begin{vmatrix} x - x_0 & f(x \mid x_0) \\ x - x_2 & f(x \mid x_2) \end{vmatrix}$$

$$= \frac{1}{(4.4 - 4.0)}\begin{vmatrix} .5 & .60206 \\ .1 & .64345 \end{vmatrix}$$

$$= \frac{(.321725 - .060206)}{0.4} = 0.65380$$

$$f(x \mid x_0, x_3) = \frac{1}{(x_3 - x_0)}\begin{vmatrix} x - x_0 & f(x \mid x_0) \\ x - x_3 & f(x \mid x_3) \end{vmatrix}$$

$$= \frac{1}{(4.6 - 4.0)}\begin{vmatrix} .5 & .60206 \\ -.1 & .66276 \end{vmatrix}$$

$$= \frac{(.5 \times .66276 + .1 \times .60206)}{0.6}$$

$$= \frac{(.33138 - .060206)}{0.6} = 0.65264$$

$$f(x \mid x_0, x_1, x_2) = \frac{1}{(x_2 - x_1)} \begin{vmatrix} x - x_1 & f(x \mid x_0, x_1) \\ x - x_2 & f(x \mid x_0, x_2) \end{vmatrix}$$

$$= \frac{1}{(4.4 - 4.2)} \begin{vmatrix} .3 & .65504 \\ -.1 & .65380 \end{vmatrix}$$

$$= \frac{(.3 \times .65380 - .1 \times .65504)}{0.2}$$

$$= \frac{(.19614 - .065504)}{0.2} = 0.65318$$

$$f(x \mid x_0, x_1, x_3) = \frac{1}{(x_3 - x_1)} \begin{vmatrix} x - x_1 & f(x \mid x_0, x_1) \\ x - x_3 & f(x \mid x_0, x_3) \end{vmatrix}$$

$$= \frac{1}{(0.4)} \begin{vmatrix} .3 & .65504 \\ -.1 & .65264 \end{vmatrix}$$

$$= \frac{(.195792 - .065504)}{0.4} = 0.65324$$

$$f(x \mid x_0, x_1, x_2, x_3) = \frac{1}{(x_3 - x_2)} \begin{vmatrix} x - x_2 & f(x \mid x_0, x_1, x_2) \\ x - x_3 & f(x \mid x_0, x_1, x_3) \end{vmatrix}$$

$$= \frac{1}{(0.2)} \begin{vmatrix} .1 & .65318 \\ -.1 & .65324 \end{vmatrix}$$

$$= \frac{(.195324 - .065318)}{0.2} = 0.65321$$

The rightmost entry in each row in the table gives,

$f(4.5 \mid x_0, x_1) = 0.65504$

$f(4.5 \mid x_0, x_1, x_2) = 0.65318$

$f(4.5 \mid x_0, x_1, x_2, x_3) = 0.65321$

It is seen that log 4.5 = 0.65321, which is the anticipated answer.

(b) **Program No. 4** **Aitken's Method**

```
# include<conio.h>
# include<iostream.h>
# include<complex.h>
# include<stdio.h>

void main ( )
  {
    clrscr( );
    float x[10],f[10],r[10][10],diff[10],xp;
    int i,j,l,m,n,p,k,y,z;
    double term1,term2,term3;
    cout<<"n\t\t\t Aitken Method\n\n";
    cout<<"Enter the number of  X data   :      ";
    cin>>n;
    cout<<"Enter value of  xp   :      ";
    cin>>xp;
    for(i=0;i<n;i++)
      {
        cout<<"Enter value of  X["<<i<<"]\t";
        cin>>x[i];
        diff[i] = xp - x[i];
      }

 cout<<"\n\n\nGiven the values of function\n\n\n";
 for(i=0;i<=n-1;i++)
 {
     cout<<"Enter value of  F("<<i<<")\t";
     cin>>f[i];
 }
 for(i=0;i<=n-1;i++)
     r[i][0] = f[i];
 for(i=0;i<=n-1;i++)
 {
     for(j=0;j<n-1;j++)
     {
       term1 = diff[i]*r[j+1][i];
       term2 = diff[j+1]*r[i][i];
       term3 = x[j+1] - x[i];
```

```
        if(term3 !=0)
                r[j+1][i+1] = (term1 – term2)/term3;
    }
}
y = 13;
clscr( );
gotoxy(3,5);
cout<<"                 Implementation of Aitken's Method\n"
for(i=0;i<=n-1;i++)       //loop to print the value of x differences
  {
    y = i + 13;
    gotoxy(5,y);

    cout<<setiosflags(ios::fixed)<<setiosflags(ios::showpoint)<<setprecision(5)<<x[1];
    gotoxy(15,y);
    cout<<"\t"<<setiosflags(ios::fixed)<<setprecision(5)<<diff[i];
  }
  p=0;
  m=25;
  k=13;
  for(i=0;i<=n-1;i++)
    {
      k = i + 13;
      for(j=p;j<=n-1;j++)
        {
          gotoxy(m,k);
          cout<<setiosflags(ios::fixed)<<setprecision(5)<<setw(10)<<r[j][i];
          k = k + 1;
        }
      p = p + 1;
      m = m + 11;
    }
    z = 0;

    for(y=0;y<n-1;y++)
            z = z + 1;
    cout<<"\n\n\n\tAtXp="<<setw(15)<<setiosflags(ios::fixed)<<setprecision(3)<<xp"
    function value is\t"<<setiosflags(ios::fixed)<<setprecision(5)<<setw(15)<<r[y][z];
    getch( );

}
```

Computer Output

Aitken Method

Enter the number of X data : 4

Enter value of xp : 4.5

Enter value of X[0] 4.0

Enter value of X[1] 4.2

Enter value of X[2] 4.4

Enter value of X[3] 4.6

Given the values of function

Enter value of F[0] .60206
Enter value of F[1] .62325
Enter value of F[2] .64345
Enter value of F[3] .66276

Implementation of Aitken's Method

4.00000	0.50000	0.60206			
4.20000	0.30000	0.62325	0.65504		
4.40000	0.10000	0.64345	0.65380	0.65318	
4.60000	–0.10000	0.66276	0.65264	0.65324	0.65321

At Xp = 4.500 Function value is 0.65321

3.7 ERROR ESTIMATION IN INTERPOLATION

So far, we have studied several formulas for interpolation. The basic principle in all these formulas is the approximation of a polynomial so that this polynomial passes through the set of points in a given table.

The error in an interpolation process is introduced by several sources:

(a) The truncation error due to terminating the series at the term in, say, the nth differences.

(b) The round-off errors in the function values and resulting errors in the differences causing oscillation in the differences.

(c) The round-off errors in the individual terms of the formula and their sum.

(d) Inaccuracy, usually due to rounding-off, in the given value of p.

We can estimate errors in any of the interpolation formulas from the first neglected term. However, we conclude this section by computing the error estimates in Newton's forward and backward difference formulas.

3.7.1 Error in Newton's Forward Difference Formula

If the function $y = f(x)$ is known explicitly, the remainder term in case of the nth-order forward difference formula is as follows:

$$E = \frac{h^{n+1}}{(n+1)!} p(p-1)\cdots(p-n) f^{(n+1)}(Z) \qquad \ldots (3.12)$$

where $x_0 < Z < x_n$.

If the function is specified by tabular values, the error is given by the following relation:

$$E = \frac{p(p-1)\cdots(p-n)}{(n+1)!} \Delta^{n+1} f_0 \qquad \ldots (3.13)$$

What can be done if the next term (i.e., (n+1)st) is not available? In this case, check if an additional point is available on the other side, namely f_{-1}. If it is available, $\Delta^{n+1} f_{-1}$ can be computed and used as an approximation for $\Delta^{n+1} f_0$.

Example 9 Given $f(x) = e^x$, for $x = 0(0.1)0.5$ correct to 4 dp.

i) Make a difference table and interpolate f(.175) using Newton's forward difference formula.

ii) Calculate the actual value of e^x for $x = .175$. Find the error in both the results.

iii) Use the formula (3.12) and estimate the error.

iv) What discretization size should be used if the entries are given to 6 dp?

Solution i) **Difference Table**

x	$f(x) = e^x$	Δ	Δ^2	Δ^3	Δ^4
0.0	1.0000				
		1052			
0.1	1.1052		110		
		1162		13	
0.2	1.2214		123		−2
		1285		11	
0.3	1.3499		134		5
		1419		16	
0.4	1.4918		150		
		1569			
0.5	1.6487				

$$x_p = 0.175;\ h = 0.1;\ x_0 = 0.1$$

$$p = \frac{(0.175 - 0.1)}{0.1} = 0.75$$

Using Newton's forward difference formula (3.2), we get,

$$\begin{aligned}
f_p &= 1.1052 + 0.75 \times .1162 + \frac{0.75(.75-1)}{2} \times .0123 \\
&\quad + \frac{0.75(0.75-1)(0.75-2)}{6} \times .0011 \\
&\quad + \frac{0.75(0.75-1)(0.75-2)(0.75-3)}{24} \times .0005 \\
&= 1.1052 + 0.08715 - 0.0012 + 0.00004 - 0.00001 \\
&= 1.1912
\end{aligned}$$

ii) True value, $e^x = e^{.175} = 1.1912$

Error = True value – Interpolated value

$= 1.1912 - 1.1912 = 0$

iii) $x_0 = 0.1;\ p = 0.75;\ h = 0.1$

$x_5 = 0.5,\ f(x) = e^x$

The fifth derivative is $f^{(v)}(x) = e^x$.

The maximum value, $f^{(v)}(x) = e^x = 1.64872$

$$\begin{aligned}
E &= \frac{h^5}{5!} p(p-1)(p-2)(p-3)(p-4)\, f^{(v)}(x) \\
&= \frac{0.75(0.75-1)(0.75-2)(0.75-3)(0.75-4)}{120} \times (.1)^5 \times 1.64872 \\
&= 0.0138411 \times 0.00001 \times 1.64872 = 0.0000002
\end{aligned}$$

iv) To keep the accuracy less than $\frac{1}{2} \times 10^{-6}$, h should be:

$$\begin{aligned}
h &= \left[\frac{E \times 120}{p(p-1)(p-2)(p-3)(p-4)} \times \frac{1}{1.64872}\right]^{\frac{1}{5}} \\
&= \left[\frac{0.0000005 \times 120}{1.64872 \times 1.7139}\right]^{\frac{1}{5}} = \left[\frac{0.00006}{2.8257412}\right]^{\frac{1}{5}} \\
&= \left(2.12333 \times 10^{-5}\right)^{\frac{1}{5}} = 0.1163
\end{aligned}$$

3.7.2 Error in Newton's Backward Difference Formula

If the function $y = f(x)$ is known explicitly, the remainder term in case of the nth-order backward difference formula is as follows:

$$E = \frac{h^{n+1}}{(n+1)!} p(p+1)(p+2)\cdots(p+n)\, f^{(n+1)}(Z) \qquad \ldots (3.14)$$

where $x_0 < Z < x_n$.

If the function $y = f(x)$ is not known but is specified only by tabular values, the error is given by the following relation:

$$E = \frac{p(p+1)(p+2)\cdots(p+n)}{(n+1)!} \nabla^{n+1} f_0 \qquad \ldots (3.15)$$

Let us illustrate this method with an example.

Example 10 Given $f(x) = \sin x$, compute the values of $f(x)$ for $x = 0.1(0.1)0.8$ correct to 4 dp.

a) Construct the difference table and interpolate $f(.75)$ using Newton's backward difference formula.

b) Calculate the exact value of $\sin x$ for $x = 0.75$. Find the error in both the results.

c) Use the formula (3.14) and estimate the error.

d) What discretization size should be used if the entries are given to 6 dp.

Solution $f(x) = \sin x$; $n = 0(0.1)0.8$ radians.

a) **Difference Table**

x	f (x)	∇	∇^2	∇^3	∇^4	∇^5
0.1	0.0998					
		989				
0.2	0.1987		−21			
		968		−8		
0.3	0.2955		−29		1	
		939		−7		−12
0.4	0.3894		−36		−11	
		903		−18		−8
0.5	0.4797		−54		−19	
		849		1		7
0.6	0.5646		−53		−12	
		796		−11		
$x_0 = 0.7$	0.6442		−64			
		732				
0.8	0.7174					

$x_p = 0.75;\ h = 0.1;\ x_0 = 0.7$

$$p = \frac{x_p - x_0}{h} = \frac{0.75 - 0.7}{0.1} = 0.5$$

Using Newton's backward difference formula (3.4), we get,

$$f_p = 0.6442 + 0.5 \times 0.0796 + \frac{1}{2} \times 0.5\,(0.5+1) \times -0.0053$$
$$+ \frac{1}{6} \times 0.5\,(0.5+1)\,(0.5+2) \times 0.0001$$
$$+ \frac{1}{24} \times 0.5\,(0.5+1)\,(0.5+2)\,(0.5+3) \times -0.0019$$
$$= 0.6442 + 0.0398 - 0.00199 + 0.00003 - 0.000052$$
$$= 0.68199$$

(b) $f(x) = \text{Sin } x$

$= \text{Sin } 0.75 = 0.68164$

Error $= f_p - f(x)$

$= 0.68199 - 0.68164 = 0.0035$

(c) $f^{(iv)}(x) = \sin x$

$f^{(v)}(x) = \cos x$

Maximum, $f^{(v)}(x) = f^{(v)}(0.1) = 0.9950$

$$E = \frac{h^{n+1}}{(n+1)!} p(p+1)(p+2)(p+3)(p+4)\, f^{(v)}(x)$$

$$= \frac{(.1)^5}{5!} \times 0.5(0.5+1)(0.5+2)(0.5+3)(0.5+4) \times 0.9950$$

$$= \frac{0.00001}{120} \times (0.5)(1.5)(2.5)(3.5)(4.5) \times 0.9950$$

$$= 0.00000245$$

d) $E = \frac{1}{2} \times 10^{-6} = 0.0000005$

From the error formula, we get,

$$h = \left[\frac{E \times 120}{p(p+1)(p+2)(p+3)(p+4) \times f^{(v)}(x)}\right]^{\frac{1}{5}}$$

$$= \left[\frac{0.0000005 \times 120}{0.5(1.5)(2.5)(3.5)(4.5) \times 0.9950}\right]^{\frac{1}{5}}$$

$$= (0.000002041)^{\frac{1}{5}} = 0.073$$

PROBLEMS

1. (a) Show that a curve $y = f(x)$, where f(x) is of the fourth degree, can be drawn through the points given by:

x	−1	0	1	2	3	4	5
f(x)	23	13	3	1	34	148	408

Use Newton's forward difference formula to find y exactly when x = 1.2.

(b) Given the following data:

x	−4	−2	0	2	4	6
f(x)	180	0	4	0	40	504

Use Newton's forward difference formula to find f(1.75).

(c) Consider the following table of values:

x	0.2	0.3	0.4	0.5	0.6
f(x)	0.2304	0.2788	0.3222	0.3617	0.3979

Find f(0.36) using Newton's forward difference formula.

(d) Prepare the difference table for the following data:

x	−1	0	1	2	3
f	10	2	0	10	62

Using Newton's forward difference formula, interpolate the value for f(−.05).

(e) Prepare the difference table for the following data:

x	0	0.2	0.4	0.6	0.8
f	0.12	0.46	0.74	0.9	1.2

Find the value for f(0.1) using Newton's forward difference formula.

(f) Generate the difference table for the following data:

x	2.0	2.2	2.4	2.6	2.8	3
f	0.301	0.342	0.380	0.415	0.447	0.477

Hence estimate f(2.15) using Newton's forward difference formula.

2. (i) Use Newton's backward difference interpolation formula to estimate the value of f(1.45) from the following data:

x	1.0	1.1	1.2	1.3	1.4	1.5
f(x)	2.0	2.1	2.3	2.7	3.5	4.5

(ii) Use Newton's forward difference formula to find f(1.05) from the above data.

(iii) Consider the following data:

x	−1	−0.75	−0.50	−0.25	0	0.25	0.50	0.75	1
f(x)	−0.4401	0.0447	0.4311	0.6694	0.7652	0.7522	0.6714	0.5587	0.4401

(a) Use Newton's forward difference interpolation formula to estimate f(−0.33).

(b) Use Newton's backward difference interpolation formula to estimate f(0.62).

(iv) The following table of values represents a polynomial of degree $n \leq 3$. It is given that there is an error in one of the tabular values of f(x) near the end of the table.

x	0	0.1	0.2	0.3	0.4
f(x)	2.00	2.11	2.28	2.39	2.56

(a) Locate the error and correct the value.

(b) Reconstruct the difference table and estimate f(0.35) using Newton's backward difference formula.

3. (a) The values of a low degree polynomial are given in the table below. It is suspected that there is a transposition error in one of the values. By differencing, locate and correct the error and find f(2.5):

x	2	3	4	5	6	7	8	9	10
f(x)	15	40	85	165	259	400	585	820	1111

(b) One of the functional values in the following table contains an error:

x	2.0	2.1	2.2	2.3	2.4	2.5	2.6	2.7
f(x)	1.4142	1.4491	1.4832	1.5160	1.5492	1.5811	1.6125	1.6432

i) Detect and correct the erroneous term and then reconstruct the difference with the corrected functional value.

ii) Find f(2.05) using Newton's forward difference formula.

iii) Find f(2.65) using Newton's backward difference formula.

4. (i) Using Stirling's interpolation formula, find f(3.25) from the following data:

x	1	2	3	4	5
f(x)	0.0000	0.6931	1.0986	1.3863	1.6094

(ii) The following table gives the value of p_x of a polynomial of the fourth degree for certain values of p_x:

x	5	6	7	8	9
p_x	6.195	5.919	5.630	5.326	5.006

Estimate the polynomial using Stirling's formula when x = 7.5.

5. (a) Given the following table:

x	f(x)
0.01	98.4342
0.02	48.4392
0.03	31.7775
0.04	23.4492
0.05	18.4542

Estimate the value of f(x) corresponding to x = 0.0341 using the following formulas:

(i) Stirling (ii) Bessel (iii) Everett and (iv) Gauss forward and backward.

(b) Consider the following table of values:

x	1.6	1.8	2.0	2.2	2.4	2.6	2.8	3.0
f(x)	.0495	.0605	.0739	.0903	.1102	.1346	.1644	.2009

Find the values of f(2.35) and f(2.2) using all the central difference formulas you have studied.

(c) Use the following table of current i against deflection, θ:

θ	.40	.45	.50	.55	.60	.65
i	1.268	1.449	1.639	1.839	2.052	2.281

to find i, when $\theta = .536$, from the Stirling and Everett formulas. Check the answer using Newton's forward difference formula.

(d) Consider the following data:

x	1.0	1.1	1.2	1.3	1.4	1.5	1.6
f(x)	1.54308	1.66852	1.81066	1.97091	2.15090	2.35241	2.57746

Find the value of f(1.35) using:

(i) Stirling, (ii) Bessel, (iii) Everett, and (iv) Gauss both formulas.

(e) Consider the following data:

x	1.0	1.2	1.4	1.6	1.8	2.0
f(x)	0.367879	0.301194	0.246597	0.201897	0.165299	0.135335

Find the vaiue of f(1.675) using:

(i) Stirling, (ii) Bessel, (iii) Everett, and (iv) Gauss both formulas.

(f) Kinematic viscosity of water, v, is related to temperature in the following manner:

T(°F)	40	50	60	70	80
v	1.66	1.41	1.22	1.06	0.93

Use a suitable interpolation formula to predict v at T = 62 °F.

(g) You measure the voltage drop v, across for a number of different values of current i. The results are:

i	0.25	0.75	1.25	1.75	2.25
v	– 0.23	– 0.33	0.70	1.88	6.00

Use a suitable interpolation formula to estimate the voltage drop for i = 0.9.

6. (a) Find f(x) at x = 1, from the following table using Lagrange's formula:

x	– 1	0	2	3
f(x)	6	10	12	19

Use a suitable interpolation formula to estimate the voltage drop for i = 0.9.

(b) Fit a polynomial to the following data:

(– 4, 180), (– 2, 0), (0, 4), (1, 0), (3, 40) and (5, 504).

Use the polynomial to find a value for f(2.4).

(c) Fit a polynomial for the function $f(x) = \frac{2^x}{x}$, for $x = 2$, 4 and 8. Use this polynomial to estimate f(6).

(d) Given the following table of values:

x	14	17	31	35
y	68.7	64.0	44.0	39.1

What is y(27)?

(e) Use Lagrange's interpolation formula to obtain a polynomial of least degree that assumes the following values:

x	1	2	3	4
y	7	11	28	63

Use the polynomial obtained to estimate f(4.5). Check the answer using Newton's backward difference formula.

(f) The function y = f(x) is given in the points (7, 3), (8, 1), (9, 1) and (10, 9). Find the value of y for x = 9.5 using Lagrange's interpolation formula.

(g) Use Lagrange's formula to estimate f(2.0) and f(4.5) from the following data:

x	1.6	2.9	3.7	4.8
f(x)	0.6250	0.3448	0.2703	0.2083

(h) Given the following data:

x	1	2	4	8
f(x)	1	3	7	11

Find f(7) using Lagrange's formula.

(i) Let $f(x) = \frac{8x}{2^x}$. Fit a polynomial for the function when x = 0(1)3. Estimate the value of f(1.5).

(j) Given the points:

(x_i, f_i) : (0, 1), (1, .765198), (2, .223891), (3, −.260052).

(a) Find the Lagrange's interpolation polynomial $p_3(x)$ for the given data set. Use the polynomial to approximate f(1.5).

(b) Find the Newton's interpolation polynomial $p_3(x)$ for the given data set. Use the polynomial in (a) above to approximate f(1.5).

(k) Given the points:

$$(x_i, f_i): (0, 1), (.25, .9689), (.5, .8776), (.75, -.5403).$$

(a) Find the Lagrange's interpolation polynomial $p_3(x)$ for the given data set. Use the polynomial obtained to approximate f(.6).

(b) Find the Newton's interpolation polynomial $p_3(x)$ for the given data set. Use the polynomial obtained in (a) above to approximate f(.6).

(l) Find the Lagrange's interpolation polynomial to fit the following data:

x	0	1	2	3
f(x)	0	1.7183	6.3891	19.0855

Use the polynomial to estimate its value at x = 1.5.

(m) Applying Lagrange's formula, find a cubic polynomial which approximate the following values:

$$y(1) = -3,\ y(3) = 9,\ y(4) = 30 \text{ and } y(6) = 132.$$

Find also the value of y(4.5).

7. (a) Estimate the interpolation polynomial for $f(x) = x^2 + \sin \pi x$ through (0, 0), (1, 1) and (2, 4), using Newton's forward difference formula at x = 0.5.

(b) What is the exact error when x = 0.5?

(c) What is the maximum error in (a) above?

(d) Find the largest value of h that will ensure 4 dp accuracy in the value of f(x), assuming quadratic interpolation is used.

8. Consider the following table of values:

x	1	2	3	4	5
f(x)	1.0000	1.4142	1.7320	2.0000	2.2361

(a) Use Newton's forward difference formula to estimate f(1.5). Using third-order interpolation, estimate also the maximum error.

(b) If we are known that $f(x) = \sqrt{x}$, what is the error in this case? Find also the maximum error.

(c) Find the largest value of h that will give 6 dp accuracy in the value of x, assuming third-order interpolation is used.

9. (a) Use Aitken's formula to estimate f(0.2) as accurately as possible from the following rounded values of f(x):

x	.17520	.25386	.33565	.42078	.50946
f(x)	.84147	.86742	.89121	.91276	.93204

(b) Use Aitken's formula to estimate f(1.4) correct to 4 dp from the following data:

x	1.20	1.25	1.30	1.35	1.45	1.50
f(x)	0.1823	0.2231	0.2624	0.3365	0.3716	0.4055

(c) Use Aitken's formula to estimate f(5) correct to 2 dp from the following data:

x	1	4	7	9
f(x)	2	13	122	504

(d) Use Aitken's method to evaluate log 3.63 from the following table:

x	3.50	3.60	3.70	3.80
log x	1.2527632	1.28093	1.30833	1.33500

(e) Consider the function, $f(x) = \dfrac{1}{1+20x^2}$. Calculate the values of the function correct to 4 dp, for x = 0.2(0.2)1.0. Estimate the value of f(0.55) using Aitken's method.

10.(a) Consider the following table of values:

x	− 0.1	0.1	0.3	0.5	0.7	0.9
f(x)	.7196	.8075	.8812	.9385	.9776	.9975

i) Use Newton's forward difference interpolation formula to estimate f(0.25) using upto fourth-order differences.

ii) Find the maximum error.

(b) Consider the following table:

x	.2	.4	.6	.8	1
f(x)	.19951	.39646	.58813	.72210	.94608

i) Find the value of f(0.3) using Lagrange and Aitken's formula.

ii) Use Newton's forward difference formula to estimate f(0.3) using upto third-order differences. Find the maximum error.

11.(a) Consider the following part of a difference table:

x	f(x)	δ	δ^2	δ^3	δ^4
6	1296				
		2800		1344	
8	4096		3104		384
		5904		1728	
10	10000				

Compute f(9) using Stirling's formula for interpolation.

(b) Given the following part of a difference table:

x^0	$\text{Sin } x^0$	δ	δ^2	δ^3	δ^4
25	0.422618		− 3216		23
		77382		590	
30	0.500000		− 3806		384

Estimate f(26.5) using Bessel's formula for interpolation.

12. Using table values of Q.5(d) above, do the following:

(a) Estimate the value of f(1.425) using Newton's backward difference formula with fourth-order differences.

(b) Compute the maximum error.

13. Using tabular values of Q.5(e) above, do the following:

(a) Use Newton's backward difference formula to estimate f(1.90) with fourth-order differences.

(b) Compute the maximum error if the tabular values are based on the function $f(x) = e^{-x}$.

(c) Compute the largest value of h that will give 7 dp accuracy in the value of x, assuming the same order of interpolation as used in (a) above.