**Numerical Methods for Science and Engineering**

**Lecture Note 5**

**Interpolation**

**5.1 Introduction**

On many occasions we are given only a few discrete set of values. To study the behavior of the function through those points a technique known as **interpolation** is introduced. Polynomial is a function which is easy to handle. The method of finding a polynomial that fits a selected set of points which behaves nearly the same way as the true function will be considered.

**5.2 Polynomial Interpolation**

Given the values of a function *f*(*x*) at (*n*+1) distinct points  we can construct a **unique** polynomial of degree less than equal to *n* which satisfies the conditions.



**General Form:** An *n*th degree polynomial can be taken as



To fit this polynomial to (*n*+1) set of points we have to solve (*n*+1) simultaneous equations and is very tedious.

**Newton Interpolating Polynomial:** A form which is convenient to use is suggested by Newton is



The unknown coefficients can be determined successively by substituting the set of values given. This form of representation is convenient in determining the unknown coefficients and plays an important role in the derivation of an interpolating polynomial.

**Example 5.1** : Find the polynomial of least degree which takes the values

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *x* | −1 | 1 | 2 | 5 |
| *f*(*x*) | 9 | 3 | 6 | 39 |

**Solution:** There are four set of values given. Let the approximated polynomial be



Using the values of *x* and *f*(*x*) in turn, we get

From , we get 

From , we get  or

From , we get or 

From , we get or



Thus, the polynomial is 





**5.3 Divided Differences**

Interpolating polynomials can be expressed in a variety of forms, and among these the Newton divided difference form is probably the convenient and efficient one.

Let the values of  corresponding to the arguments  be

.

The first divided difference for arguments  and  is defined by:



The second divided difference for arguments, and  is defined as:



Similarly higher divided differences are defined. The *n*th divided differences with (*n*+1) arguments is defined by 

**Property 1:** The divided differences are symmetric about their arguments i.e. does not depend on the order of the arguments.

**Property 2:** The *n*th divided differences of a polynomial of degree *n* is constant

**5.4 Interpolation Formula using Divided Differences**

**5.4.1 Newton Divided Difference Interpolation**

The interpolating polynomial  through the points  can be written in the Newton form as



Substituting , we have



 or 



or 

or 

Continuing the process, it can be shown that 

Thus in terms of the divided differences interpolating polynomial can be written as





This is known as **Newton’s divided difference interpolation formula**.

If is a polynomial through (*n*+1) points , then the polynomial  through those points with an extra point is 

The constant *b* can be calculated by substituting 

Example 5.2

The table below gives the values of *x* and *f*(*x*):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *x* : | −1 | 1 | 2 | 3 | 4 |
| *f*(*x*) : | −7 | −1 | 8 | 29 | 68 |

(i) Construct a divided-difference table for the above data.

(ii) Find the polynomial of least degree that incorporates the values in the table and find .

1. Find by linear interpolation a real root of .

(iv) Find the polynomial  that takes the values of the above table and .

**Solution:**

(i) The divided difference table for the given data is as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *x* | *f*(*x*) | f1[ ] | f2[ ] | f3[ ] | f4[ ] |
| -1 | -7 |  |  |  |  |
| 1 | -1 | 3 |  |  |  |
| 2 | 8 | 9 | 2 |  |  |
| 3 | 29 | 21 | 6 | 1 |  |
| 4 | 68 | 39 | 9 | 1 | 0 |

(ii) The needed differences are enclosed by the double lined box.

By Newton’s divided difference formula, we get

****

and ****

****

(iii) Here ****

Thus a root is in (1, 2).

From the table, we have

*x* *f*(*x*) 1DD

1 −1

1. 8 9

Thus the root is the solution of

****

or  ****

(iv) The polynomial  can be written as



where *b* is a constant.

Taking , we have

****

or 

Hence 

The required polynomial is

****

**5.4.2 Newton Backward Divided Difference Formula**

If the nodes are reordered as , the divided differences interpolating polynomial can be written as





and is called the **Newton Backward Divided Difference** formula.

**5.5 Lagrange Interpolating Polynomial**

Lagrange polynomial of degree one passing through two points  and  is written as



Lagrange polynomial of degree two passing through three points , and  is written as



Lagrange polynomial of degree three passing through four points ,,  and  is written as





In general, the Lagrange polynomial of degree *n* passing through  points ,, ⋅ ⋅ ⋅ ,  is written as





**Example 5.3**

The following table gives the values of an empirical function *f*(*x*) for certain values of *x*.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *x* | 0 | 1 | 2 | 3 |
| *f*(*x*) | −4 | −1 | 8 | 29 |

Use the Lagrange interpolation formula to estimate

1. the value of 
2. the root of the equation *f*(*x*) = 0 to 3 decimal places.

(i) Applying Lagrange’s formula, we have

****

****

and ****

****

****

****

(ii) Let . Then the root of  corresponds to . To find the root let us use the Lagrange formula in reverse order i.e. consider the polynomial in terms of *y*.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *y* | −4 | −1 | 8 | 29 |
| *x* | 0 | 1 | 2 | 3 |

Then

****

When , then

****

****

****

**Exercise 5.4** The upward velocity of a rocket is given as a function of time below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *t* (s) | 10 | 15 | 20 | 22.5 | 30 |
| *v*(*t*) (m/s) | 227 | 363 | 517 | 603 | 903 |

i. Construct a divided-difference table for the above data.

ii. Determine the value of the velocity at seconds using two suitable points.

iii. Determine the value of the velocity at seconds using three suitable points.

iv. Find the polynomial which passes through all the points and find .

v. Use Lagrange interpolating polynomial to estimate

a. the value of t for using two suitable points.

b. the value of *t* for using three suitable points.

vi. Write down MATLAB codes using “**polyfit(x, y, n)**” and “**polyval(p, x)**” for the following.

a. Find the polynomial of least degree that incorporates all the values in the table. and estimate the velocities corresponding to . seconds.

b. Draw the figure showing fitted polynomial and the given points.

**Solution:**

i.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *t* | *v*(*t*) | *v*1[ ] | *v*2[ ] | *v*3[ ] | *v*4[ ] |
| 10 | 227 |  |  |  |  |
| 15 | 363 | 27.2 |  |  |  |
| 20 | 517 | 30.8 | 0.36 |  |  |
| 22.5 | 603 | 34.4 | 0.48 | 0.0096 |  |
| 30 | 903 | 40.0 | 0.56 | 0.0053 | 0.00021 |

ii. Note that and using the relevant part of the table

|  |  |  |
| --- | --- | --- |
| *t* | *v*(*t*) |  |
| 15 | 363 |  |
| 20 | 517 | 30.8 |

we have the linear polynomial

And

iii. Note that . Thus we may choose points corresponding to . Collecting the relevant part of the table

|  |  |  |  |
| --- | --- | --- | --- |
| *t* | *v*(*t*) | *v*1[ ] | *v*2[ ] |
| 15 | 363 |  |  |
| 20 | 517 | 30.8 |  |
| 22.5 | 603 | 34.4 | 0.48 |

The polynomial with 3 points is

And

iv. Polynomial passing through all the points is

And

v. For a given *v* we need to calculate the value of *t*, so consider the Lagrange polynomial in reverse order.

a. With two points consider

and the Lagrange polynomial is

For ,

b. With three points consider

and the Lagrange polynomial is

For ,

vi. MATLAB CODES

a. >> t=[10 15 20 22.5 30];

>> v=[227 363 517 603 903];

>> pt=polyfit(t,v,4)

pt = -0.0002 0.0240 -0.4267 28.2000 -34.2000

>> t1=[17 25 30];

>> v1=polyval(pt,t1);

>> % Output value of v for t

>> t\_v =[t1',v1']

t\_v = 17.0000 421.9875

25.0000 695.8000

30.0000 903.0000

b. >> t=[10 15 20 22.5 30];

>> v=[227 363 517 603 903];

>> pt=polyfit(t,v,4);

>> t1=linspace(5,35,500); % generates 500 values

>> v1=polyval(pt,t1); % calculates values of v

>> plot(t, v,'o',t1,v1);

>> title('Graph of v against t');

>> xlabel('Time (t)');

>> ylabel('Velocity v(t)');

**Exercise 5.1**

1. The table below gives the velocity *v* at time *t*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *t(s)* | 1 | 3 | 4 | 7 |
| *v(m/s)* | 3 | 5 | 21 | 201 |

i. Construct a divided-difference table for the above data.

ii. Find the polynomial of least degree that incorporates the values in the table.

iii. Find the acceleration at time .

iv. Find the distance function when .

2. . The table below gives the values of

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *x* | -2 | 0 | 3 | 6 | 7 |
| *f*(*x*) | 2 | -4 | -58 | 842 | 1802 |

1. Construct a divided-difference table for the above data.
2. Find the polynomial which passes through all the points of the table and find .
3. Find the polynomial  that takes the values of the above table and
4. Use Lagrange interpolating polynomial to estimate
   1. the value of using two suitable points.
   2. the value of *x* for *f* using three suitable points.
5. Write down MATLAB codes using “**polyfit(x, y, n)**” and “**polyval(p, x)**” for the following.

Find the polynomial of least degree that incorporates all the values in the table and estimate the values corresponding to .

3. The table below gives the values of

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 4 | 5 | 7 | 9 | 11 |
|  | 62 | 95 | 185 | 307 | 461 |

1. Construct a divided-difference table for the above data.
2. Find the polynomial which passes through all the points of the table and find .
3. Find the polynomial  that takes the values of the above table and
4. Use Lagrange interpolating polynomial to estimate
5. the value of using two points.
6. the value of *x* for *f* using three points.
7. Write down MATLAB codes using “**polyfit(x, y, n)**” and “**polyval(p, x)**” for the following.

Find the polynomial of least degree that incorporates all the values in the table and estimate the values corresponding to .

4. The table below gives the values of *x* and *f*(*x*):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *x* | -2 | -1 | 0 | 3 |
| *f*(*x*) | 12 | 14 | 10 | 22 |

1. Construct a divided-difference table for the above data.
2. Find the polynomial of least degree that incorporates the values in the table and find .
3. Given , find the polynomial  that also takes the values of the above table.
4. Use Lagrange interpolation formula to find
   1. a real root of using linear approximation.
   2. a real root of using all the points.
5. Write down MATLAB codes using “**polyfit(x, y, n)**” and “**polyval(p, x)**” to plot the figure showing fitted polynomial and the given points.