**Numerical Methods For Science and Engineering**

**Lecture Note 5**

**Spline Interpolation**

**5.1 Introduction**

Spline interpolation function is a piecewise polynomial function joined together with certain conditions satisfied by them.

A function of the form

is called a **spline** of degree *m* if

(i) the domain of is the interval

(ii) are all continuous functions on

(iii) is a polynomial of degree less than equal to *m* on each subinterval , .

**5.1.1 Linear Spline Interpolation**

For a linear spline through we may take is of the form

Since the line passes through and we have

and

where and .

The resulting linear spline curve can be written as

**Example 5.1**

Find the linear spline for the following data set

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *X* | −1 | 1 | 2 | 5 |
| *Y* | 2.2 | 3.5 | 5.4 | 1.5 |

Hence estimate the value of .

**Solution**

Linear spline functions in different intervals are

Linear spline function is

The value is in . Thus

**5.2 Cubic Spline Interpolation**

Cubic spline interpolation is used very often. It gives smoother curves than other types. To determine the cubic spline, we need to use cubic polynomial for each subintervals.

Consider the cubic polynomial in each subinterval of the form

where are to be determined.

Since the spline passes through and ,

We can see that there are

conditions but we need to determine constants. So we need to add two boundary conditions to get unique solution.

Normally we use three types of boundary conditions:

1. Second derivatives at end points are known

The special case

give spline called natural **cubic spline**.

2. First derivatives at end points are known

give spline called **clampedcubic spline**.

3. Automatically adjusted boundary conditions known as **not-a-knot** cubic spline.

This condition assumes that are continuous at the second and last but one points.

Note that minimum number of data points is four for this condition to be used..

**MATLAB Spline Interpolation Functions**

(1) MATLAB function **spline**

yy=spline(x, Y, xx) for **not-a-knot** cubic spline

x, Y are inputs and xx expolant.

yy=spline(x, [dY0, Y, dYn], xx) for clamped cubic spline

`

(2) **csape** spline interpolation with various end conditions

Syntax: sp=csape(X, Y, conds)

some of the conds are

‘second’ adjusted second derivatives if not mentioned it uses

‘clamped’ adjusted first derivatives

‘not-a-knot’ uses not-a-knot condtion

**Example 5.2**

A natural cubic spline is defined by

1. Use continuity and boundary conditions to estimate .
2. Find the value of from the spline curve’
3. Use MATLB function “**sp=csape(x, y, ‘conds’)**” to construct natural cubic spline for the data set ). Find using “**fnval(**s**p,x)**”’

Plot the spline curve using “**fnplt(sp)**” along with the data points.

**Solution**

Let

and

Then

and

1. Conditions at the interior point give

For natural cubic spline the boundary conditions give

From (3), or

From (1), 8(1) or C

From (2), or

The natural cubic spline function is

(ii)



(ii)

>> clear

>> x=[-1 1 2];

>> y=[1 -1 10];

>> sp=csape(x,y,'second');

>> y1=fnval(sp,1.4)

y1 =

2.6320

>> fnplt(sp, 2, [-2, 4])

>> hold on % used to plot in the same figure

>> plot(x, y,'O')

>> hold off

**Example:5.3**

A cubic spline which interpolates the data is defined by

1. If the spline satisfies the not-a-knot boundary conditions, find .
2. Using continuity of at , find .
3. Using continuity of at , find .
4. Show that passes through .
5. Use MATLB function “**sp=spline(x, y)**” to construct the spline curve and find coefficients by **“sp.coefs**”.
6. Write down MATLAB codes using “**fnval(**s**p,x)**” to estimate the values of for from the spline curve.
7. Write down MATLAB codes using “**fnplt(sp)**” to plot the spline curve and the given data points.

**Solution :** Let the cubic spline be

where

Then

And

Also

i. Spline curve satisfies not-a-knot boundary conditions. Thus

ii. Continuity of gives

And

1. Continuity of gives

And

and

>> x=[1 2 4 5];

>> y=[-10 -6 2 18];

>> sp=spline(x,y);

>> format short g

>> Coefficients = sp.coefs

Coefficients =

1 -4 7 -10

1 -1 2 -6

1 5 10 2

>> x1=[1.4, 2.5, 4.8];

>> y1=fnval(sp,x1);

>> xy\_value =[x1', y1']

xy\_value =

1.4 -7.776

2.5 -5.125

4.8 13.712

>> fnplt(sp, [0, 6])

>> hold on

>> plot(x,y,'o')

**Exercise 5**

1. In a chemical reaction the concentration level y of the product at time ***t*** (minute) was measured every half hour. The following results were found:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *t* | 1.0 | 1.5 | 2.0 | 2.5 |
| *y* | 0.24 | 0.27 | 0.31 | 0.36 |

Construct a linear spline interpolation to estimate the concentration level at 2.25 minute.

2. Use the portion of the given steam table for superheated H2O at 200 MPa to find the corresponding entropy, *s*, for a specific volume, *v*, of 0.118 m3/kg with linear spline.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| V (m3/kg) | 0.2037 | 0.2114 | 0.32547 | 0.33213 |
| S (kJ/kg K) | 6.5147 | 6.6453 | 6.8664 | 6.9513 |

3. Given the following set of values of and :

|  |
| --- |
|  |
| Find the linear spline passing through the above points. |
| A natural cubic spline through the above points are defined by |
|  |
| Use continuity and boundary conditions to find equations satisfied by . |
| Solve for . |
| Find the value of from linear spline and from cubic curve. |
| Write MATLAB codes using “**sp=csape(x, y, ‘second’)**” to construct natural cubic spline for the above data set. Plot the spline curve using “**fnplt(sp)**” along with the data points. |
|  |
| 4. A natural cubic spline is defined by |
| i. Use continuity and boundary conditions to find the values of *A, B, C, D,* and *E*. |
| ii. Estimate the value of |
| Iii. Write a MATLB code using function “**spline(x, y)**” to construct the spline curve and “**fnval(function,x)**” to estimate the values of for . |

|  |  |  |
| --- | --- | --- |
| 5. |  | A natural cubic spline is defined by |
|  |  | i. Use continuity and boundary conditions to find the values of *A, B, C, D* and *E*. |
|  |  | ii. Estimate from the spline curve. |
|  |  | iii. Use MATLAB function “**sp=spline(x, y)**” and “**fnval(**s**p,x)**” to construct the spline curve from the data and find the values of for from the spline curve. |
| **6.** |  | A clamped cubic spline function  through is defined by  Given that . |
|  | i. | Use continuity and boundary conditions to find the values of  *A, B, C* and *D*. |
|  | ii. | Estimate from the spline curve. |
|  | iii. | Use MATLB function “**sp=spline(x, y)**” and “**fnval(**s**p,x)**” to construct the spline curve from the data and find the values of for from the spline curve. |