MM8 Differential Equations and Euler Method

- kl.8.15-9.10 review of MM7 and some examples
- kl.9.10 – 10.40 exercise (see notes)
- kl.10.50-11.30 MM8 lecture (I)

Spline Function

Piecewise polynomials

Let $a=x_0 < x_1 < \dots x_N=b$. A function $s: [a,b] \to R$ is a **spline** or **spline function** of degree m with knots (or nodes) $x_0, x_1, \dots x_N$ if

1) s is a piecewise polynomial such that, on each subinterval $[x_k,x_{k+1}]$, s has degree at most m, and
2) s is $m-1$ times differentiable everywhere

Linear Spline Interpolation

- Spline of degree 1

At the nodes $a = x_0 < x_1 < x_2 \cdots < x_n = b$, there is

$$s(x_k) = f(x_k) \quad for \quad k = 0,1,2\cdots,n$$

In the node interval $[x_k, x_{k+1}]$, s is a polynomial of degree 1 passing through the points $(x_k, f(x_k))$ and $(x_{k+1}, f(x_{k+1}))$, thereby

$$s(x) = f(x_k) + \frac{f(x_{k+1}) - f(x_k)}{x_{k+1} - x_k}(x - x_k), \quad x \in [x_k, x_{k+1}]$$

simply

$$s_k(x) = a_k + b_k(x - x_k), \quad k = 0,1,2\cdots,n-1$$

where

$$a_k = f(x_k)$$

$$b_k = \frac{f(x_{k+1}) - f(x_k)}{x_{k+1} - x_k}$$

Linear Spline Interpolation - Example

Example: Find the linear spline which takes the following values:

$x_0 = 0.0, \ f(x_0) = 0.00; \quad x_1 = 0.2, \ f(x_1) = 0.18;$

$x_2 = 0.3, \ f(x_2) = 0.26; \quad x_3 = 0.5, \ f(x_3) = 0.41;$

There are three intervals and $x_0 = 0.0, x_1 = 0.2, x_2 = 0.3, x_3 = 0.5$.
according to general formula:

$s_k(x) = a_k + b_k(x - x_k), \quad k = 0,1,2\cdots,n-1$ where

$$a_k = f(x_k), \quad b_k = \frac{f(x_{k+1}) - f(x_k)}{x_{k+1} - x_k}$$

there is

$$a_0 = f(x_0) = 0.00, \ b_0 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{0.18 - 0.00}{0.2 - 0.0} = 0.90$$

$$a_1 = f(x_1) = 0.18, \ b_1 = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{0.26 - 0.18}{0.3 - 0.2} = 0.80$$

$$a_2 = f(x_2) = 0.26, \ b_2 = \frac{f(x_3) - f(x_2)}{x_3 - x_2} = \frac{0.41 - 0.26}{0.5 - 0.3} = 0.75$$

thereby

$$s(x) = \begin{cases} 0.00 + 0.9(x - 0.0) = 0.9x & 0.0 \le x \le 0.2 \\ 0.18 + 0.8(x - 0.2) = 0.8x + 0.02 & 0.2 \le x \le 0.3 \\ 0.26 + 0.75(x - 0.3) = 0.75x + 0.035 & 0.3 \le x \le 0.5 \end{cases}$$

Matlab implementation - exercise

```
clear, clc

x=[0.0,0.2,0.3,0.5];
y=[0.0,0.18,0.26,0.41];

X=0.0:0.01:0.5;
Y=interp1(x,y,X);
Y1=interp1(x,y,0.1);
Y2=interp1(x,y,0.4);
%Y3=interp1(x,y,0.8);

plot(x,y,'r*',X,Y)
hold on
plot(0.1,Y1,'bo')
plot(0.4,Y2,'bo')
%plot(0.8,Y3,'bo')

hold off
```

Cubic Spline Interpolation

At the nodes $a = x_0 < x_1 < x_2 \cdots < x_n = b,$ the piecewise polynomial has the form

$$s_k(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3 \quad k = 0,1,2,\cdots n-1$$

The following conditions should be satisfied

$(i) \quad s_k(x_k) = f(x_k)$ and $s_k(x_{k+1}) = f(x_{k+1}),$ for $k = 0,1,2\cdots,n-1$
$(ii) \quad s_k'(x_{k+1}) = s_{k+1}'(x_{k+1})$ for $k = 0,1,2\cdots,n-2$
$(ii) \quad s_k''(x_{k+1}) = s_{k+1}''(x_{k+1})$ for $k = 0,1,2\cdots,n-2$

There are 4n coefficients, but 4n-2 equations ---- 2 degrees of freedom!

Natural Cubic Spline

The natural cubic spline is defined by imposing the additional conditions:

s’’(a)=s’’(b)=0

Cubic Spline Formula (I)

Nodes $a = x_0 < x_1 < x_2 \cdots < x_n = b,$

$$s_k(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3 \quad k = 0,1,2,\cdots n-1$$

The following conditions should be satisfied

(i) $s_k(x_k) = f(x_k)$ and $s_k(x_{k+1}) = f(x_{k+1}),$ for $k = 0,1,2\cdots,n-1$

$a_k = f(x_k)$ for $k = 0,1,2\cdots,n-1$

$s_k(x_{k+1}) = f(x_{k+1}) \Rightarrow b_k(x_{k+1} - x_k) + c_k(x_{k+1} - x_k)^2 + d_k(x_{k+1} - x_k)^3 = f(x_{k+1}) - f(x_k)$

denote $h_k = x_{k+1} - x_k$, there is

$$b_k + c_k h_k + d_k h_k^2 = \frac{f(x_{k+1}) - f(x_k)}{h_k} \mathrel{\hat=} \delta_k$$

(ii) $s_k'(x_{k+1}) = s_{k+1}'(x_{k+1})$ for $k = 0,1,2\cdots,n-2$

$b_k + 2c_k h_k + 3d_k h_k^2 = b_{k+1}$ for $k = 0,1,2\cdots,n-2$

(iii) $s_k''(x_{k+1}) = s_{k+1}''(x_{k+1})$ for $k = 0,1,2\cdots,n-2$

$2c_k + 6d_k h_k = 2c_{k+1}$ for $k = 0,1,2\cdots,n-2$

$$d_k = \frac{c_{k+1} - c_k}{3h_k}$$

Cubic Spline Formula (II)

$$s_k(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3 \quad k = 0,1,2,\cdots n-1$$

$a_k = f(x_k)$ for $k = 0,1,2\cdots,n-1$

Denote $h_k = x_{k+1} - x_k$, there is $b_k + c_k h_k + d_k h_k^2 = \dfrac{f(x_{k+1}) - f(x_k)}{h_k} \mathrel{\hat=} \delta_k \quad \Leftrightarrow \quad b_k = \delta_k - c_k h_k - d_k h_k^2$

$2c_k + 6d_k h_k = 2c_{k+1}$ for $k = 0,1,2\cdots,n-2 \quad \Leftrightarrow \quad d_k = \dfrac{c_{k+1} - c_k}{3h_k}$

$b_k + 2c_k h_k + 3d_k h_k^2 = b_{k+1} \Leftrightarrow$ for $k = 0,1,2\cdots,n-2 \quad \Leftrightarrow \quad c_k h_k + 2(h_k + h_{k+1})c_{k+1} + h_{k+1}c_{k+2} = 3(\delta_{k+1} - \delta_k)$

$$H\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-1} \end{bmatrix} = 3\begin{bmatrix} \delta_1 - \delta_0 \\ \delta_2 - \delta_1 \\ \vdots \\ \delta_{n-1} - \delta_{n-2} \end{bmatrix},$$

H is the tridiagonal matrix: $H = \begin{bmatrix} 2(h_0 + h_1) & h_1 & & \\ h_1 & 2(h_1 + h_2) & h_2 & \\ & h_2 & \ddots & h_{n-2} \\ & & h_{n-2} & 2(h_{n-2} + h_{n-1}) \end{bmatrix}$

Cubic Spline - Example

Find the natural cubic spline which fits the data

x	25	36	49	64	81
$f(x)$	5	6	7	8	9

$$s_k(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3 \quad k = 0,1,2,\cdots n-1$$

(a) Determine $n = 4$, then $a_k = f(x_k)$ for $k = 0,1,2,3$

(b) Determine steplengths: h_k, for $k = 0,1,2,3$

(c) Determine the divided differences δ_k, for $k = 0,1,2,3$

(d) Introduce $c_0 = c_n = 0$, and construct the tridiagonal system to slove c_1, $...c_{n\text{-}1}$

(e) Use $d_k = \dfrac{c_{k+1} - c_k}{3h_k}$ for $k = 0,1,2,3$

(f) Use $b_k = \delta_k - c_k h_k - d_k h_k^{\,2}$ for $k = 0,1,2,3$

$$H\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = 3\begin{bmatrix} \delta_1 - \delta_0 \\ \delta_2 - \delta_1 \\ \\ \delta_3 - \delta_2 \end{bmatrix}, \qquad H \text{ is the tridiagonal matrix: } H = \begin{bmatrix} 2(h_0 + h_1) & h_1 & \\ h_1 & 2(h_1 + h_2) & h_2 \\ & h_2 & 2(h_2 + h_3) \\ & & \end{bmatrix}$$

Cubic Spline – Matlab Codes

```
function s=cspline(nodes,data,x)

% computes coefficients for natural cubic spline
% at nodes with values data
% evaluate the spline at all points of the array x

N=length(nodes)-1;
P=length(x);

% compute the steplength
h=diff(nodes);
%compute first divided differences of data
D=diff(data)./h;
% compute of D for rhs of linear system
dD3=3*diff(D);

a=data(1:N)
```

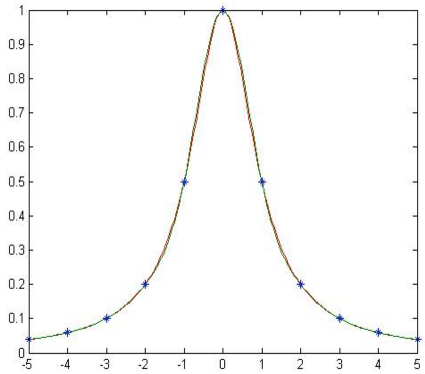
```
clear clc
X=[25,36,49,64,81]; Y=[5,6,7,8,9];
x=[25:0.4:81];
y=cspline(X,Y,x);
plot(x,y)
```

```
% generate tridiagonal system
H=diag(2*(h(2:N)+h(1:N-1)));
for k=1:N-2
   H(k,k+1)=h(k+1);
   H(k+1,k)=h(k+1);
end
c=zeros(1,N+1);
c(2:N)=H\ dD3';

b=D-h.*(c(2:N+1)+2*c(1:N))/3;
d=(c(2:N+1)-c(1:N))./(3*h);

% evaluattion

for i=1:P
   k=1;
   while (x(i)>nodes(k+1)) & (k<N)
      k=k+1;
   end
   z=x(i)-nodes(k);
   s(i)=a(k)+z*(b(k)+ z*(c(k)+z*d(k)));
end
```

Another Example

- Example 16: use cubic spline for the data from the function 1/(1+x^2) with integer nodes in [-5,5]

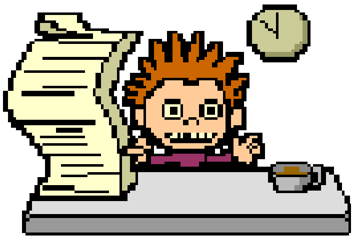
```
clear clc

X=-5:5;
Y=runge(X);
x=-5:0.05:5;

plot(X,Y,'*',x,runge(x),'r',x,cspline(X,Y,x))

figure

plot(x,runge(x)-cspline(X,Y,x))
```



Matlab Built-in Functions

- YI = INTERP1(X,Y,XI,METHOD)
  - 'nearest' - nearest neighbor interpolation
  - 'linear' - linear interpolation
  - 'spline' - piecewise cubic spline interpolation (SPLINE)
  - 'pchip' - shape-preserving piecewise cubic interpolation
  - 'cubic' - same as 'pchip'
  - 'v5cubic' - the cubic interpolation from MATLAB 5, which does not extrapolate and uses 'spline' if X is not equally spaced.
- PP = SPLINE(X,Y) provides the piecewise polynomial form of the cubic spline interpolant to the data values Y at the data sites X, for use with the evaluator PPVAL and the spline utility UNMKPP.

Exercises (MM7)



Question One (Exercise 4.4.2 and 4.4.3, pp.114):

Find the linear spline which interpolates the data

x	0	1	3	4	6
$f(x)$	5	4	3	2	1

- What are its values at 2, 3.5, and 4.5?
- Write a Matlab m-file to realize the linear spline interpolation for a given set of data;
- Use your developed m-file to plot the linear spline of above data.

Question Two (Exercise 4.4.4, pp.114):

Find the natural cubic spline interpolation for the following data

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

(see the textbook for the hints)