A2, Question 3

We are to write a Matlab function

$$[a, b, c] = MySpline(x, y)$$

where

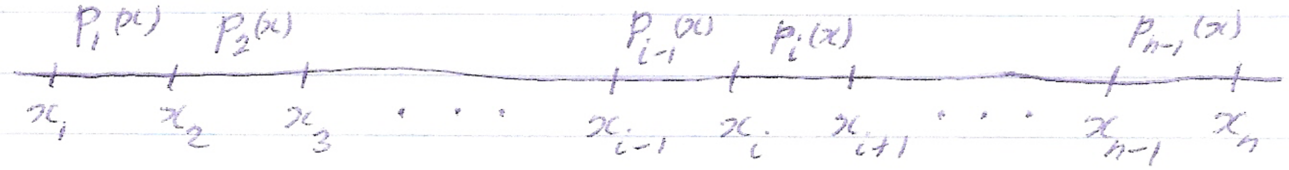
x, y – input vectors of (x_i, y_i) data $(1 \le i \le n)$

a, b, c – output vectors of the coefficients of the natural cubic spline which interpolates (x, y), using the <u>special representation</u>:

$$p_i(x) = a_{i-1}\frac{(x_{i+1}-x)^3}{6h_i} + a_i\frac{(x-x_i)^3}{6h_i} + b_i(x_{i+1}-x) + c_i(x-x_i),$$

$$\left[\text{where } h_i = x_{i+1} - x_i\right], \qquad \text{for } i = 1, \ldots, n-1.$$

Schematically,



In order to solve for the coefficients $\{a_i\}_{i=0}^{n-1}$, $\{b_i\}_{i=1}^{n-1}$ and $\{c_i\}_{i=1}^{n-1}$, apply the cubic spline conditions to $p_i(x)$, $1 \le i \le n-1$.

We will need to use the ~~the~~ derivative formulas:

$$p_i'(x) = -3a_{i-1}\frac{(x_{i+1}-x)^2}{6h_i} + 3a_i\frac{(x-x_i)^2}{6h_i} - b_i + c_i$$

$$p_i''(x) = a_{i-1}\frac{(x_{i+1}-x)}{h_i} + a_i\frac{(x-x_i)}{h_i}$$

Interpolation conditions:

(1) $p_i(x_i) = y_i \implies \frac{1}{6} a_{i-1} h_i^2 + b_i h_i = y_i$, for $i=1,\dots,n-1$

(2) $p_i(x_{i+1}) = y_{i+1} \implies \frac{1}{6} a_i h_i^2 + c_i h_i = y_{i+1}$, for $i=1,\dots,n-1$

Note that if we know a_i's then we can solve equation (1) for b_i and equation (2) for c_i:

$$\boxed{b_i = \frac{y_i}{h_i} - \frac{1}{6} a_{i-1} h_i}; \qquad \boxed{c_i = \frac{y_{i+1}}{h_i} - \frac{1}{6} a_i h_i};$$

for $i=1,\dots,n-1$.

1st derivative conditions

$p_i'(x_{i+1}) = p_{i+1}'(x_{i+1})$, for $i=1,\dots,n-2$

$$\implies 3a_i \frac{h_i^2}{6h_i} - b_i + c_i = -3a_i \frac{h_{i+1}^2}{6h_{i+1}} - b_{i+1} + c_{i+1}$$

$$\implies \frac{1}{2} a_i h_i - b_i + c_i = -\frac{1}{2} a_i h_{i+1} - b_{i+1} + c_{i+1}$$

Plugging in the above formulas for b_i and c_i in terms of a_i's we get, after rearranging terms to put terms involving a_i's on the left hand side:

$$\frac{1}{6} h_i a_{i-1} + \frac{1}{3}(h_i + h_{i+1}) a_i + \frac{1}{6} h_{i+1} a_{i+1} = r_i, \quad i=1,\dots,n-2$$

where the right hand side values are

$$r_i = \frac{(y_{i+2} - y_{i+1})}{h_{i+1}} - \frac{(y_{i+1} - y_i)}{h_i}.$$

We have a linear system of $(n-2)$ equations in the n unknowns $\{a_i\}_{i=0}^{n-1}$.

We must add two <u>boundary conditions</u> before we can solve.

First note that the linear system is <u>tridiagonal</u> so for the two extra equations, let us add them in the form:

1st eqn: $t_0 a_0 + t_1 a_1 = r_0$

last eqn: $t_2 a_{n-2} + t_3 a_{n-1} = r_{n-1}$

Where $t_0, t_1, t_2, t_3, r_0, r_{n-1}$ are constants to be chosen, depending on the desired boundary conditions.

We then have the following <u>$n \times n$ tridiagonal system</u>:

$$T \underline{a} = \underline{r}$$

where $\underline{a} = [a_0 \; a_1 \; \dots \; a_{n-1}]^T$; $\underline{r} = [r_0 \; r_1 \; \dots \; r_{n-1}]^T$;

$$T = \begin{bmatrix} t_0 & t_1 & & & & \\ \frac{1}{6}h_1 & \frac{1}{3}(h_1+h_2) & \frac{1}{6}h_2 & & 0 & \\ & \frac{1}{6}h_2 & \frac{1}{3}(h_2+h_3) & \frac{1}{6}h_3 & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & \frac{1}{6}h_{n-2} & \frac{1}{3}(h_{n-2}+h_{n-1}) & \frac{1}{6}h_{n-1} \\ & & & & t_2 & t_3 \end{bmatrix}.$$

2nd derivative conditions

$$p_i''(x_{i+1}) = p_{i+1}''(x_{i+1}) \quad , \quad \text{for } i=1,\ldots,n-2$$

$$\Longrightarrow \quad a_i \frac{h_i}{h_i} = a_i \frac{h_{i+1}}{h_{i+1}} \Longrightarrow a_i = a_i \quad \text{(an identity!)}.$$

In other words, the special form for $p_i(x)$ is such that the second derivatives automatically match at interior points

Boundary conditions

We are asked to compute the natural cubic spline so we want the boundary conditions:

$$p_1''(x_1) = 0 \quad \text{and} \quad p_{n-1}''(x_n) = 0 .$$

I.e., $a_0 \dfrac{h_1}{h_1} = 0$ and $a_{n-1} \dfrac{h_{n-1}}{h_{n-1}} = 0$

$$\Longrightarrow \boxed{a_0 = 0} \quad \text{and} \quad \boxed{a_{n-1} = 0} .$$

Therefore, in our general tridiagonal system $T\underline{a} = \underline{r}$ we set

$$t_0 = 1, \quad c_1 = 0, \quad t_2 = 0, \quad t_3 = 1, \quad r_0 = 0, \quad r_{n-1} = 0 .$$

Matlab function MySpline is now easy to write. Define the matrix T and vector $\underline{r}$, and solve $T\underline{a} = \underline{r}$ for $\underline{a}$. Then calculate $\underline{b}$ and $\underline{c}$ from $\underline{a}$ (and $\underline{x}$, $\underline{y}$).

Note: For efficiency, one could develop the code for Gaussian elimination (without pivoting) specifically for a tridiagonal system.

```
%
% function [a, b, c] = MySpline(x, y)
%
% Input:
%
%   x and y are vectors (same size) of x-values and y-values, corresponding
%   to points in the xy plane.  The x-values must be in increasing order.
%
% Output:
%
%   a, b and c are vectors of parameters corresponding to the special form
%   for cubic polynomials (see Course Notes, Section 2.2.3). Note that "a"
%   is indexed starting at 1, so a_0 from the Course Notes is stored in
%   position a(1), etc. The vectors "b" and "c" are indexed the same as
%   in the Course Notes.
%
%   Hence...
%     a(1) = a_0
%     a(2) = a_1           b(1) = b_1         c(1) = c_1
%          :                    :                  :
%     a(n) = a_(n-1)     b(n-1) = b_(n-1)   c(n-1) = c_(n-1)
%
%   The polynomial piece is evaluated using
%
%     p_k(x) = a(k)*(x(k+1)-xvals(m))^3/(6*hk) + ...
%              a(k+1)*(xvals(m)-x(k))^3/(6*hk) + ...
%              b(k)*(x(k+1)-xvals(m)) + c(k)*(xvals(m)-x(k));
%
%   where hk = x(k+1) - x(k).  See the function
%
%     EvaluateMySpline
%
%   for more details.
%
%
function [a b c] = MySpline(x, y)

    n = length(x);
    a = zeros(n,1);
    b = zeros(n-1,1);
    c = zeros(n-1,1);
    h = zeros(n-1,1);
    r = zeros(n,1);
    T = zeros(n);

    % compute h

    for i=1:n-1
        h(i) = x(i+1) - x(i) ;
    end

    % compute r

    for j=2:n-1
        r(j) = (( y(j+1) - y(j) )/ h(j)) - (( y(j) - y(j-1) )/ h(j-1) ) ;
    end
```

```
    % compute T

    for m=2:n-1
        T(m,m-1) = h(m-1)/6;
        T(m,m) = ( h(m-1) + h(m) )/3;
        T(m,m+1) = h(m)/6;
    end

    T(1,1) = 1;
    T(n,n) = 1;

    % compute a

    a=T\r;

    % compute b and c

    for k=1:n-1
        b(k) = y(k)/h(k) - (a(k) * h(k))/6;
        c(k) = y(k+1)/h(k) - (a(k+1) * h(k))/6;
    end
end
```