Last time we learned how to find a quadratic spline that fits \( n = 3 \) given points. Let us consider how to generalize our method for a general \( n \) to plot the quadratic spline curve on Matlab.

Given \( n \) data points \((x_1, y_1), \ldots, (x_n, y_n)\), we want to find \( n-1 \) parabolas:

\[
s_j(x) = a_j x^2 + b_j x + c_j,
\]

\[
s_j(x) = a_j x^2 + b_j x + c_j,
\]

\[
s_{n-1}(x) = a_{n-1} x^2 + b_{n-1} x + c_{n-1}.
\]

Here \( s_j(x) \) is a parabola joining two points \((x_j, y_j)\) and \((x_{j+1}, y_{j+1})\). We need to find \( a_j, b_j, c_j \) for \( j = 1, 2, \ldots, n-1 \). These are "local" parameters because they are parameters of local curves \( s_1, s_2, \ldots, s_{n-1} \).

Our strategy is to determine them through the "global" parameters \( M_1, M_2, \ldots, M_n \), where \( M_j \) is the slope of the spline curve at point \((x_j, y_j)\).

The three conditions

\[
s_j'(x_j) = M_j,
\]

\[
s_j'(x_{j+1}) = M_{j+1},
\]

\[
s_j(x_j) = y_j
\]

help us determine \( a_j, b_j, c_j \) in terms of \( M_j \) and \( M_{j+1} \). Specifically,

\[
a_j = \frac{M_j - M_{j+1}}{x_j - x_{j+1}}
\]

\[
b_j = \frac{x_j M_{j+1} - x_{j+1} M_j}{x_j - x_{j+1}}
\]

\[
c_j = y_j - a_j x_j^2 - b_j.
\]

[Detail calculation is found in Lecture 23.]

The condition the parabola must pass through the right point \((x_{j+1}, y_{j+1})\)
gives us an equation which \( M_j \) and \( M_{j+1} \) must satisfy. That is

\[ a_j \frac{x_j}{x_{j+1}} + b_j \frac{x_j}{x_{j+1}} + c_j = y_{j+1} \]

or equivalently

\[ a_j \left( \frac{x_j}{x_{j+1}} - \frac{x_j}{x_j} \right) + b_j \left( \frac{x_{j+1}}{x_{j+1}} - \frac{x_j}{x_j} \right) = y_{j+1} - y_j. \]

We have used the fact that \( c_j = y_j - a_j \frac{x_j}{x_j} - b_j \frac{x_j}{x_j} \).

After plugging \( a_j \) and \( b_j \) into this formula and reducing, we get

\[ M_j + M_{j+1} = \frac{2(y_{j+1} - y_j)}{x_{j+1} - x_j}. \]

There are \( n-1 \) such equations since \( j \) runs from 1 to \( n-1 \). Note that the right hand side is completely known. There are \( n-1 \) equations to solve for \( n \) unknowns \( M_1, M_2, \ldots, M_n \). One has the freedom to choose \( M_1 \). Then \( M_2, M_3, \ldots, M_n \) are determined by

\[ M_2 = \frac{2(y_2 - y_1)}{x_2 - x_1} - M_1, \]

\[ M_3 = \frac{2(y_3 - y_2)}{x_3 - x_2} - M_2, \]

\[ \vdots \]

How do we plot the spline which we have found?

In other words, how do we plot the parabola \( S_1, S_2, \ldots, S_{n-1} \) on the same graph?

The idea is to run a ‘for’ loop to plot each curve \( S_j \) individually while using the command ‘hold on’ to make sure that all plots are drawn on the same graph. For example,

```matlab
for j = 1:n-1
    u = xj : 0.01 : xj+1;
    s = aj*u.^2 + bj*u + cj;
    plot(u,s);
    hold on
end
```