Rewrite:

\[ p_j(x) = A_j x^3 + B_j x^2 + C_j x + D_j \]

for \( j = 1, \ldots, n-1 \)

**Unknowns:** \( A_j, B_j, C_j, D_j \) for \( j = 1, \ldots, n-1 \)

\[ \Rightarrow 4(n-1) \text{ unknowns} \]

**Need:** \( 4(n-1) \) equations to (hopefully) have a unique solution: a unique value for each \( A_j, B_j, C_j, D_j \); \( j = 1, \ldots, n+1 \)

**Impose conditions of smoothness**

\[ (C_1) \text{ Continuity} \]

\[ p_j(x_j) = y_j \rightarrow \text{hits the data} \]

\[ p_j(x_{j+1}) = y_{j+1} \]

\( j = 1, \ldots, n-1 \)

\[ 2(n-1) \text{ eqns} \]

\[ (C_2) \text{ Differentiability} \]

\[ p_j'(x_{j+1}) = p_{j+1}'(x_{j+1}) \]

\( \text{for } j = 1, \ldots, n-2 \)

\[ n-2 \text{ eqns} \]

\[ (C_3) \text{ Twice-differentiability at } x_1, \ldots, x_{n-1} \]

\[ p_j''(x_{j+1}) = p_{j+1}''(x_{j+1}) \]

\( \text{for } j = 1, \ldots, n-2 \)

\[ n-2 \text{ eqns} \]

At this pt: we have

\[ 2(n-1) + (n-2) + (n-2) = 4n - 6 \]

\( \text{eqns} \rightarrow \text{need two more!} \)

\[ (C_4) \text{ Impose } p_i''(x_1) = p_i''(x_n) = 0. \]

\[ \text{2 eqns!} \]
So, we have:

4(n-1) eqns; 4(n-1) unknowns!

**Example:** Set \( n = 3 \) and solve.

Unknowns: \( A_j, B_j, C_j, D_j \); \( j = 1/2 \)

\( \Rightarrow \) **8 unknowns!**

The corresponding eqns:

(C1) \( P_1(x_1) = y_1 \):
\[ A_1 x_1^3 + B_1 x_1^2 + C_1 x_1 + D_1 = y_1 \]

(C2) \( P_1'(x_2) = P_2'(x_2) \)
\[ P_j'(x) = 3A_j x^2 + 2B_j x + C_j \]

Then:
\[ 3A_1 x_1^2 + 2B_1 x_1 + C_1 = 0 \] (1 eqn)

(C3) \( P_1''(x_2) = P_2''(x_2) \)
\[ P_j''(x) = 6A_j x + 2B_j \]

Then:
\[ 6A_1 x_2^2 + 2B_1 = 6A_2 x_2 + 2B_2 \]

\( \Rightarrow \)
\[ 6A_1 x_2^2 + 2B_1 - 6A_2 x_2 - 2B_2 = 0 \] (2 eqns)

(C4) \( P_1''(x_1) = 0 \)
\[ \Rightarrow \quad 6A_1 x_1 + 2B_1 = 0 \] (2 eqns)

\( \Rightarrow \)
\[ 6A_2 x_1 + 2B_2 = 0 \] (2 eqns)

\( \Rightarrow \quad 8 \) eqns; 8 unknowns!
Let's write the linear system using matrix notation.

$$\begin{bmatrix}
x_1^3 & x_1^2 & x_1 & 1 & 0 & 0 & 0 & 0 \\
x_2^3 & x_2^2 & x_2 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & x_3^3 & x_3^2 & x_3 & 1 \\
0 & 0 & 0 & 0 & x_4^3 & x_4^2 & x_4 & 1 \\
3x_2^2 & 2x_2 & 1 & 0 & -3x_3^2 & -2x_3 & -1 & 0 \\
6x_2 & 2 & 0 & 0 & -6x_2 & -2 & 0 & 0 \\
6x_1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 6x_3 & 2 & 0 & 0
\end{bmatrix} \begin{bmatrix}
A_1 \\
B_1 \\
C_1 \\
D_1 \\
A_2 \\
B_2 \\
C_2 \\
D_2
\end{bmatrix} = \begin{bmatrix}
y_1 \\
y_2 \\
y_3 \\
0 \\
0 \\
0
\end{bmatrix}$$

Need to solve: $Sa = b$

where we know $S$ and $b$; want to find $a$, which will then give us $p_1(x)$ and $p_2(x)$.

(See MATLAB experiment)
A more efficient approach and numerically stable method for solving the same cubic spline interpolation problem.

Problem: Given \((x_1, y_1), \ldots, (x_n, y_n)\) find \(P_j(x)\) (cubic polynomial), \(j = 1, \ldots, n-1\) s.t. \((C1)-(C4)\) are satisfied.

New approach: (5.2.6)

Define:

\(A_j(x) := \frac{x_{j+1}-x}{x_{j+1}-x_j}\)

\(B_j(x) := 1 - A_j(x)\)