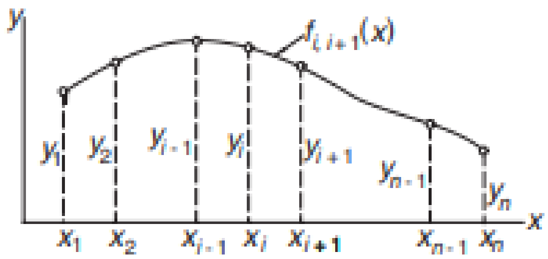
## 1- Cubic Spline Interpolation:




Cubic spline interpolation is the method to generate different cubic polynomial that spans the segment between every two knots. Therefore, the cubic spline is not a perfect method when there are more than a few data points to evaluate. But a cubic spline has less tendency than a polynomial one to oscillate between data points. So, slopes are continuous at each node. Also, the second derivatives of the start and end points are zero.

### 1.1 Generalized Formulation of Cubic Spline Interpolation:

We can denote the second derivative of the spline at knot $i$ by $k_i$ , and must have the same value $k_i$ for the next and previous cubic polynomial;

$$f''_{(i-1),i}(x_i) = f''_{i,(i+1)}(x_i) = k_i$$

For the starting and end points;

$$k_1 = k_n = 0$$

If Lagrange interpolation is applied between two points;

$$f''_{i,(i+1)}(x_i) = \big(k_i * l_i(x)\big) + \big(k_{i+1} * l_{i+1}(x)\big) \;\;,\;\; l_i(x) = \frac{x - x_{i+1}}{x_i - x_{i+1}} \;\;,\;\; l_i(x) = \frac{x - x_i}{x_{i+1} - x_i}$$

$$f''_{i,(i+1)}(x_i) = \frac{k_i(x_i - x_{i+1}) - k_{i+1}(x - x_i)}{x_i - x_{i+1}}$$

Integrated twice;

$$f_{i,(i+1)}(x_i) = \frac{k_i(x - x_{i+1})^3 - k_{i+1}(x - x_i)^3}{6(x_i - x_{i+1})} + A(x - x_{i+1}) - B(x - x_i)$$

For $f_{i,(i+1)}(x_i) = y_i$;

$$A = \frac{y_i}{x_i - x_{i+1}} - \frac{k_i}{6}(x_i - x_{i+1})$$

For $f_{i,(i+1)}(x_{i+1}) = y_{i+1}$;

$$B = \frac{y_{i+1}}{x_i - x_{i+1}} - \frac{k_{i+1}}{6}(x_i - x_{i+1})$$

After A and B are substituted into $f_{i,(i+1)}(x_i)$;

$$k_{i-1}(x_{i-1} - x_i) + 2k_i(x_{i-1} - x_{i+1}) + k_{i+1}(x_i - x_{i+1}) = 6\left(\frac{y_{i-1} - y_i}{x_{i-1} - x_i} - \frac{y_i - y_{i+1}}{x_i - x_{i+1}}\right), i = 2,3, \dots, n-1$$

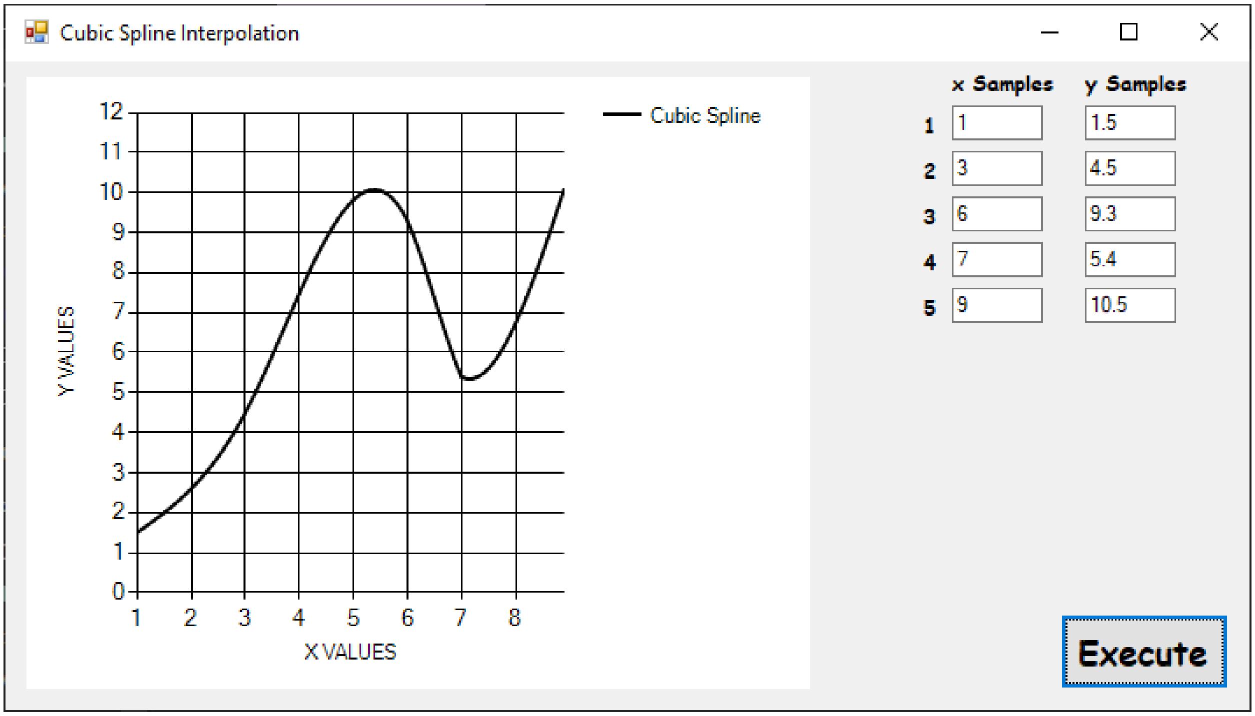
In the program, to solve the linear equation and find k values, the inverse matrix method is used;

$$k_{i-1}(x_{i-1} - x_i) + 2k_i(x_{i-1} - x_{i+1}) + k_{i+1}(x_i - x_{i+1}) = 6\left(\frac{y_{i-1} - y_i}{x_{i-1} - x_i} - \frac{y_i - y_{i+1}}{x_i - x_{i+1}}\right), i = 2,3, \dots, n-1$$

$$\rightarrow \begin{bmatrix} 2(x_1 - x_2) & x_2 - x_3 & 0 \\ x_2 - x_3 & 2(x_2 - x_4) & x_3 - x_4 \\ 0 & 2(x_3 - x_5) & x_4 - x_5 \end{bmatrix} . \begin{bmatrix} k_2 \\ k_3 \\ k_4 \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} 2(x_1 - x_2) & x_2 - x_3 & 0 \\ x_2 - x_3 & 2(x_2 - x_4) & x_3 - x_4 \\ 0 & 2(x_3 - x_5) & x_4 - x_5 \end{bmatrix}^{-1} . \begin{bmatrix} E_1 \\ E_2 \\ E_3 \end{bmatrix} = \begin{bmatrix} k_2 \\ k_3 \\ k_4 \end{bmatrix}$$

### 1.2 C# Console Application for Cubic Spline Interpolation;




First, x and y values must be written to the “x Samples” and “y Samples” columns in order and then execute. Given x values must be in increased order.