Interpolation and Polynomial Approximation - Lecture 03

Dr. Serhat Hosder

Associate Professor of Aerospace Engineering

Mechanical and Aerospace Engineering

290B Toomey Hall

Missouri S&T

Rolla, MO 65409

Phone: 573-341-7239

E-mail: hosders@mst.edu

Outline

- Review the tri-diagonal equations for obtaining the solution of cubic spline coefficients
- Review the implementation summary for the cubic splines
- Work on examples to show
    - Comparison with Lagrange interpolation
    - The effect of the boundary condition

The Linear System for Natural Cubic Splines

The tri-diagonal system of equation can be written in matrix form as:

$$
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & \ldots & 0 \\
h_0 & 2(h_0+h_1) & h_1 & 0 & 0 & 0 & 0 & \ldots & \ldots & 0 \\
0 & h_1 & 2(h_1+h_2) & h_2 & 0 & 0 & 0 & \ldots & \ldots & 0 \\
0 & 0 & \ldots & \ldots & \ldots & 0 & 0 & \ldots & \ldots & 0 \\
0 & 0 & 0 & \ldots & \ldots & \ldots & 0 & \ldots & \ldots & 0 \\
0 & 0 & 0 & 0 & \ldots & \ldots & \ldots & \ldots & \ldots & 0 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & 0 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\text{x}
\begin{bmatrix}
c_0 \\ c_1 \\ c_2 \\ . \\ . \\ . \\ . \\ . \\ . \\ c_n
\end{bmatrix}
=
\begin{bmatrix}
0 \\
\frac{3}{h_1}(a_2-a_1)-\frac{3}{h_0}(a_1-a_0) \\
\ldots. \\ \ldots. \\ \ldots. \\ \ldots. \\ \ldots. \\ \ldots. \\
\frac{3}{h_{n-1}}(a_n-a_{n-1})-\frac{3}{h_{n-2}}(a_{n-1}-a_{n-2}) \\
0
\end{bmatrix}
$$

The boundary values used here are for a *natural* cubic spline.

Note, once this system is solved for the c_k, then one can generate b_k and d_k directly and since the a_k are known, this completes the definition of the cubics for each interval.

The Linear System for Cubic Splines with Clamped Ends

This tri-diagonal system of equation can be written in matrix form as:

$$\begin{bmatrix} 2h_0 & h_0 & 0 & 0 & 0 & 0 & 0 & \dots & \dots & 0 \\ h_0 & 2(h_0+h_1) & h_1 & 0 & 0 & 0 & 0 & \dots & \dots & 0 \\ 0 & h_1 & 2(h_1+h_2) & h_2 & 0 & 0 & 0 & \dots & \dots & 0 \\ 0 & 0 & \dots & \dots & \dots & 0 & 0 & \dots & \dots & 0 \\ 0 & 0 & 0 & \dots & \dots & \dots & 0 & \dots & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{n-1} & 2h_{n-1} \end{bmatrix} \text{x} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ . \\ . \\ . \\ . \\ . \\ . \\ c_n \end{bmatrix} = \begin{bmatrix} \frac{3}{h_0}(a_1-a_0)-3f'(x_0) \\ \frac{3}{h_1}(a_2-a_1)-\frac{3}{h_0}(a_1-a_0) \\ \dots. \\ \dots. \\ \dots. \\ \dots. \\ \dots. \\ \dots. \\ \frac{3}{h_{n-1}}(a_n-a_{n-1})-\frac{3}{h_{n-2}}(a_{n-1}-a_{n-2}) \\ 3f'(x_n)-\frac{3}{h_{n-1}}(a_n-a_{n-1}) \end{bmatrix}$$

The boundary values used here are for a cubic spline with *clamped* ends

Implementation Summary

1. Solve the tri-diagonal equation for the coefficients c_k $(k=0,1,2,\ldots,n)$

2. Obtain b_k and d_k $(k=0,1,2,\ldots,n-1)$

$$b_k = \frac{a_{k+1} - a_k}{h_k} - \frac{h_k}{3}\left(2c_k + c_{k+1}\right) \qquad d_k = \frac{c_{k+1} - c_k}{3h_k}$$

3. Note that $a_k = f(x_k)$ $(k=0,1,2,\ldots,n)$

4. Approximate the function with the set of cubic splines

$$f(x) \approx S(x) = \left\{ s_k(x) \mid k = 0, 1, \cdots, n-1 \right\}$$

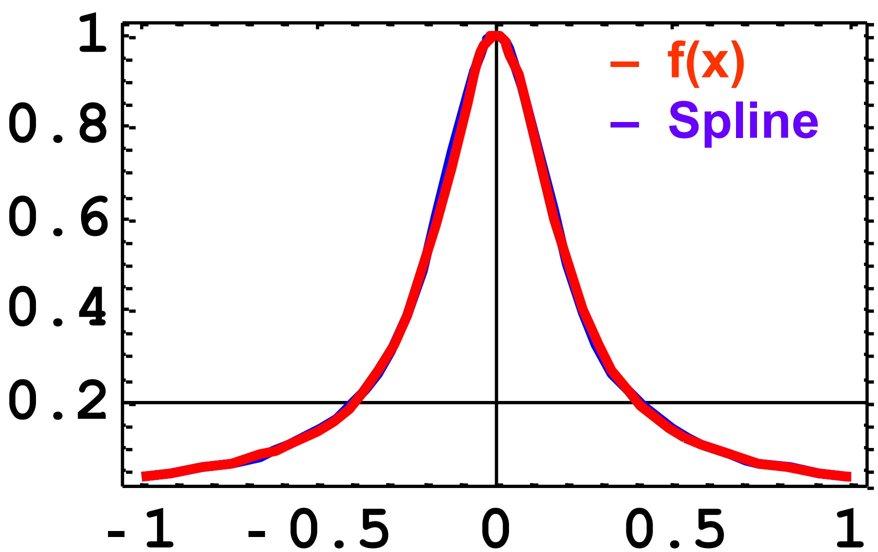
5. For any x, select the correct spline fit (choose k) and its coefficients and interpolate.

Example – Cubic Spline Interpolation

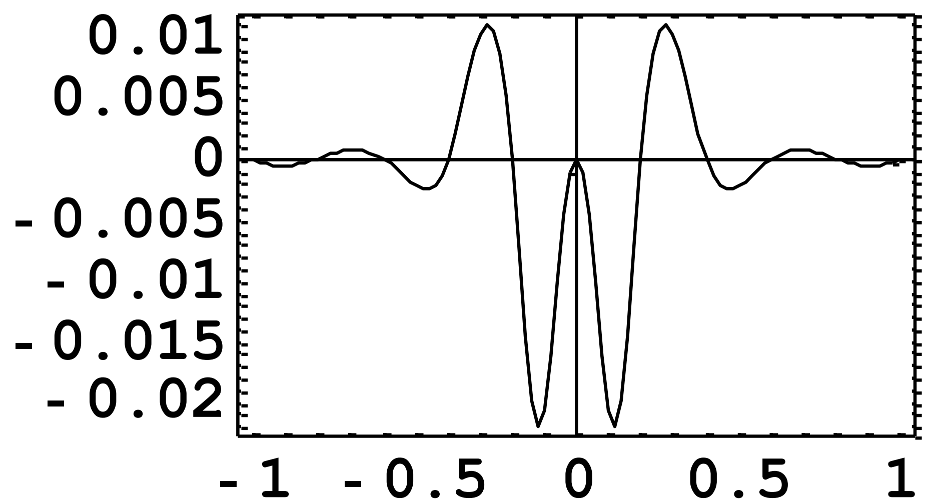
The function that we are approximating is: $f(x) = \dfrac{1}{1+25x^2}$

(using 11 evenly spaced points on {-1,+1})

A comparison of the function and the cubic spline approximation



Error – Cubic Spline

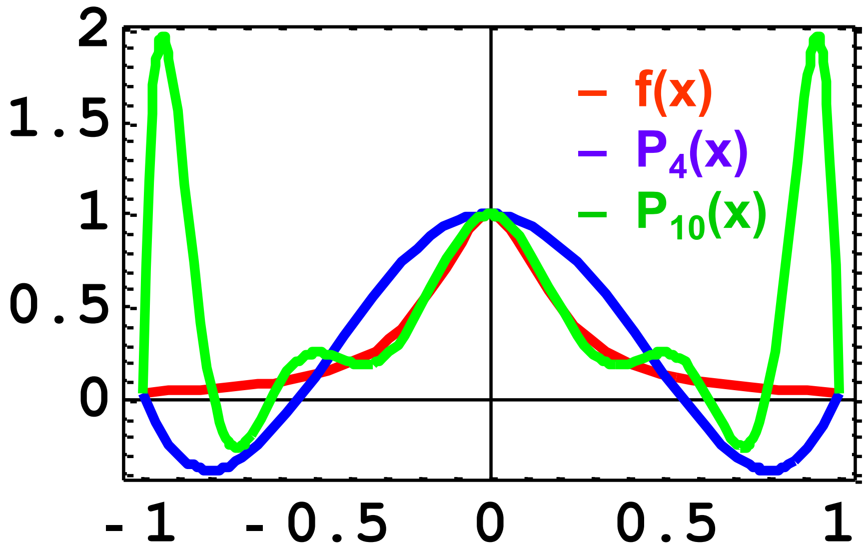


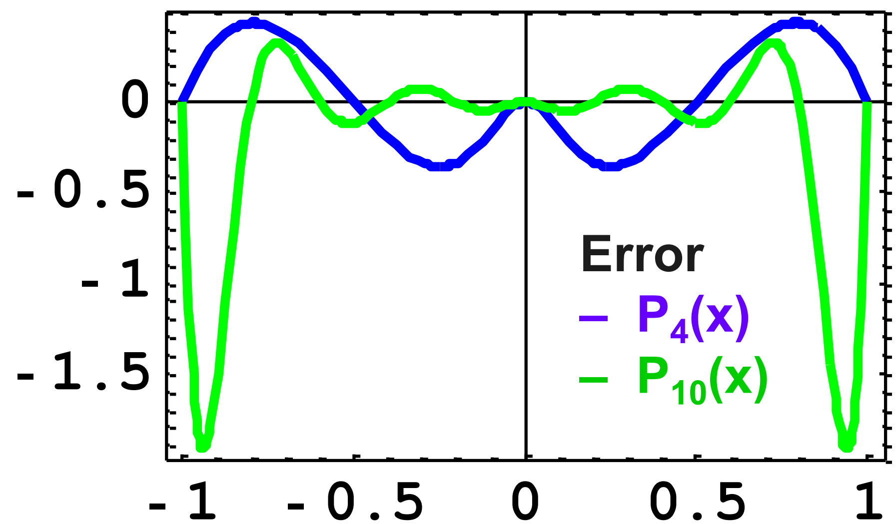
Note that the error is much smaller than both the fourth and tenth order Lagrange polynomials (see next slide for reminder). Cubic splines are commonly used because of the simplicity and performance of the method.

Lagrange Interpolation – High Degree

The function that we are approximating is: $$f(x) = \frac{1}{1+25x^2}$$

A comparison of the function and polynomial approximations

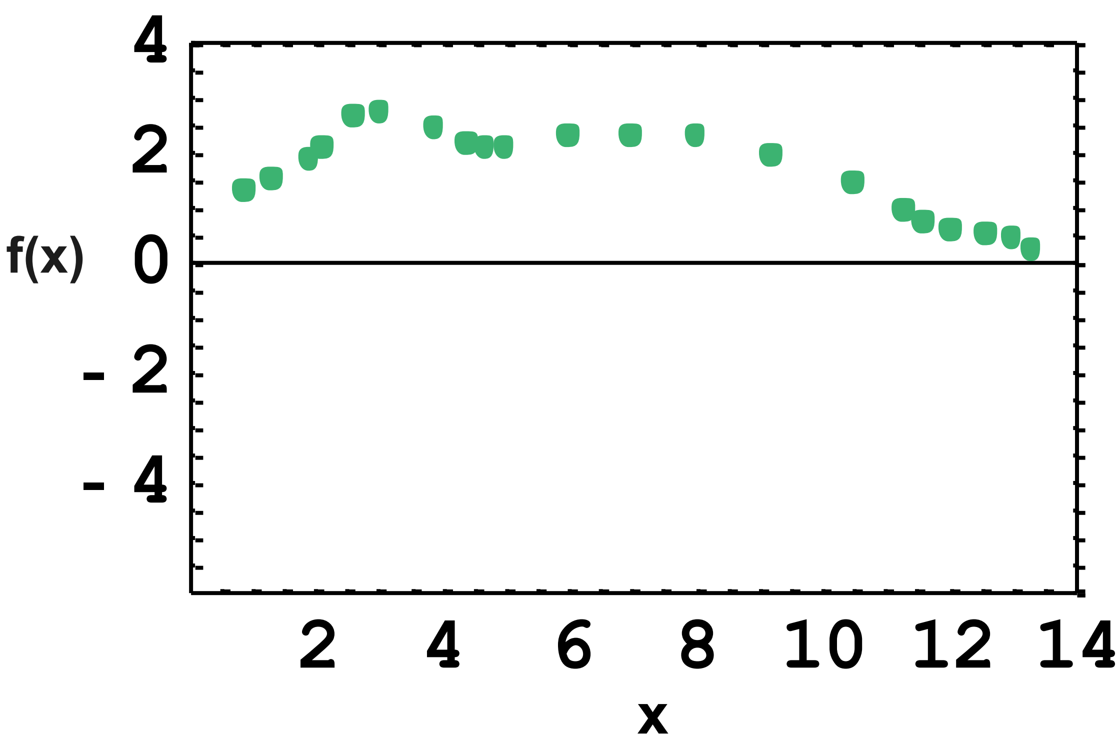




Note that although the 10^{th} degree polynomial matches the function at eleven data points, the approximation is unacceptable. Because of this, high degree polynomial approximation is not commonly used. The large wiggles near the endpoints are also typical of high degree polynomial approximations.

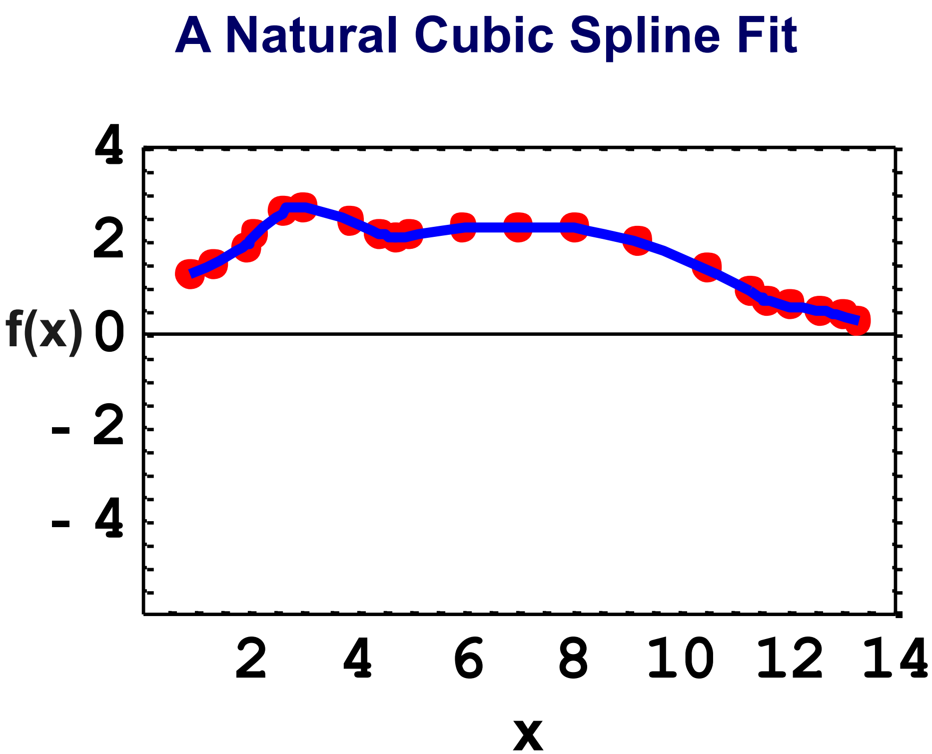
Example – Ruddy Duck

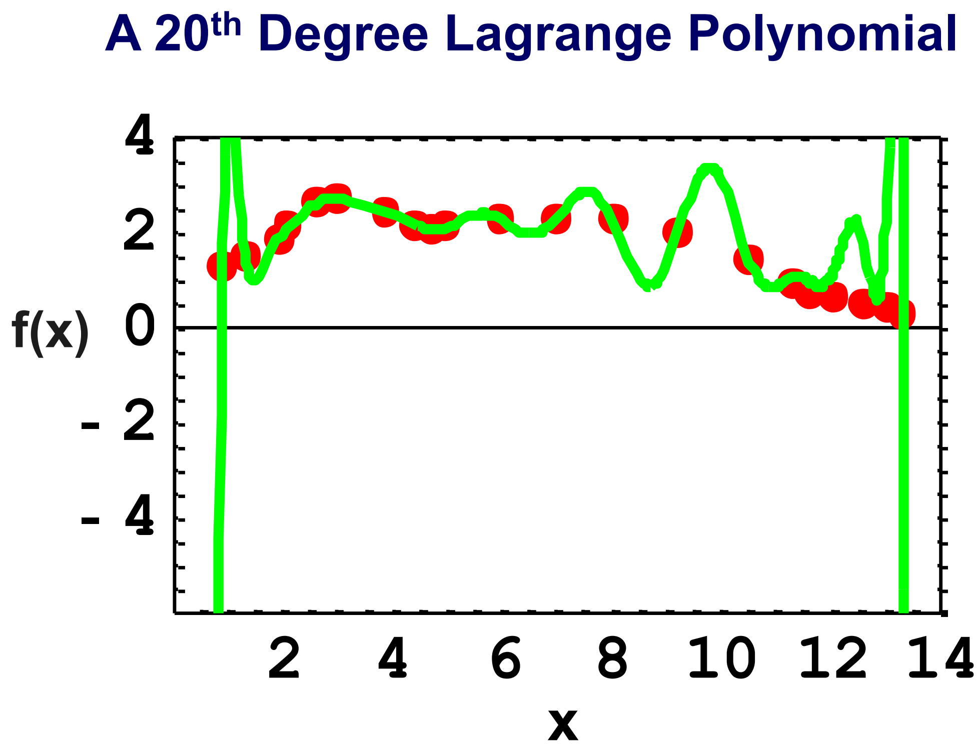
On pages 149-151, data is given that approximates the upper surface of a ruddy duck in flight. 21 data pairs are given in the form {x, f(x)}. The below is a plot of the data.points.



x	f(x)
0.9	1.3
1.3	1.5
1.9	1.85
2.1	2.1
2.6	2.6
3.	2.7
3.9	2.4
4.4	2.15
4.7	2.05
5.	2.1
6.	2.25
7.	2.3
8.	2.25
9.2	1.95
10.5	1.4
11.3	0.9
11.6	0.7
12.	0.6
12.6	0.5
13.	0.4
13.3	0.25

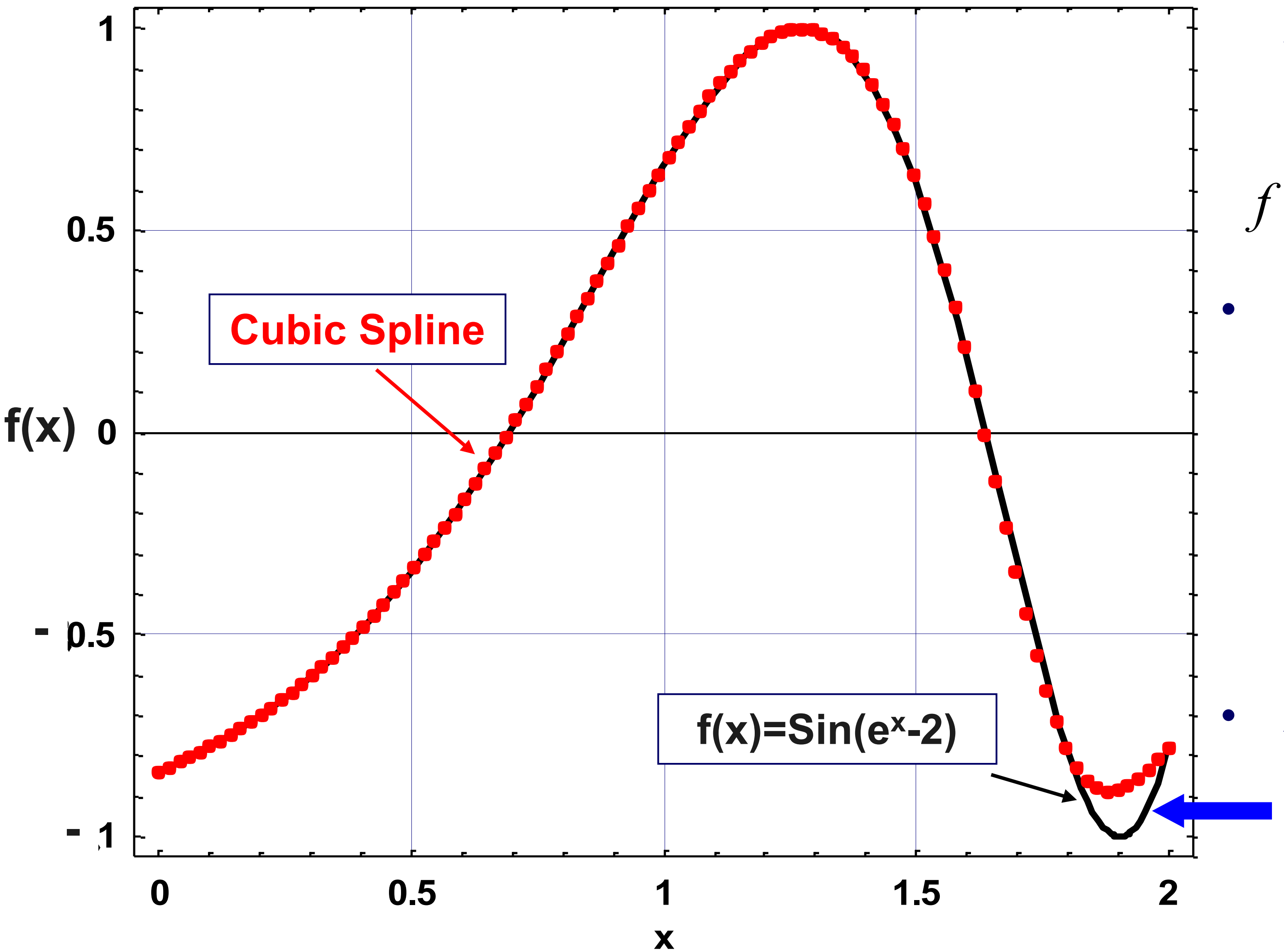
Example – Ruddy Duck





Notice once again the large oscillations in the figure on the right that are often associated with high degree polynomial interpolation.

Example - Natural Cubic Spline

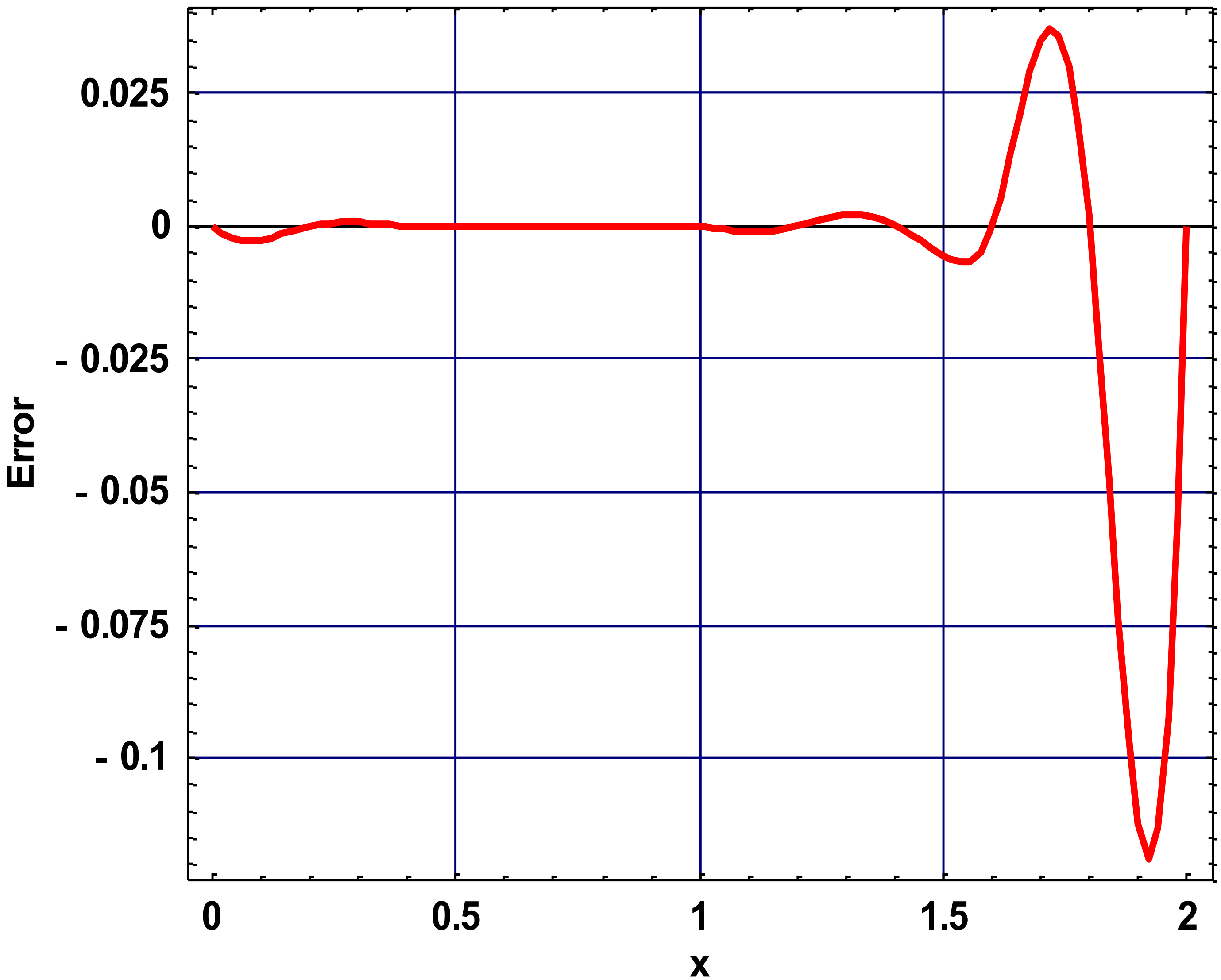


- The function that we try to approximate is:

$$f(x) = Sin(e^x - 2)$$

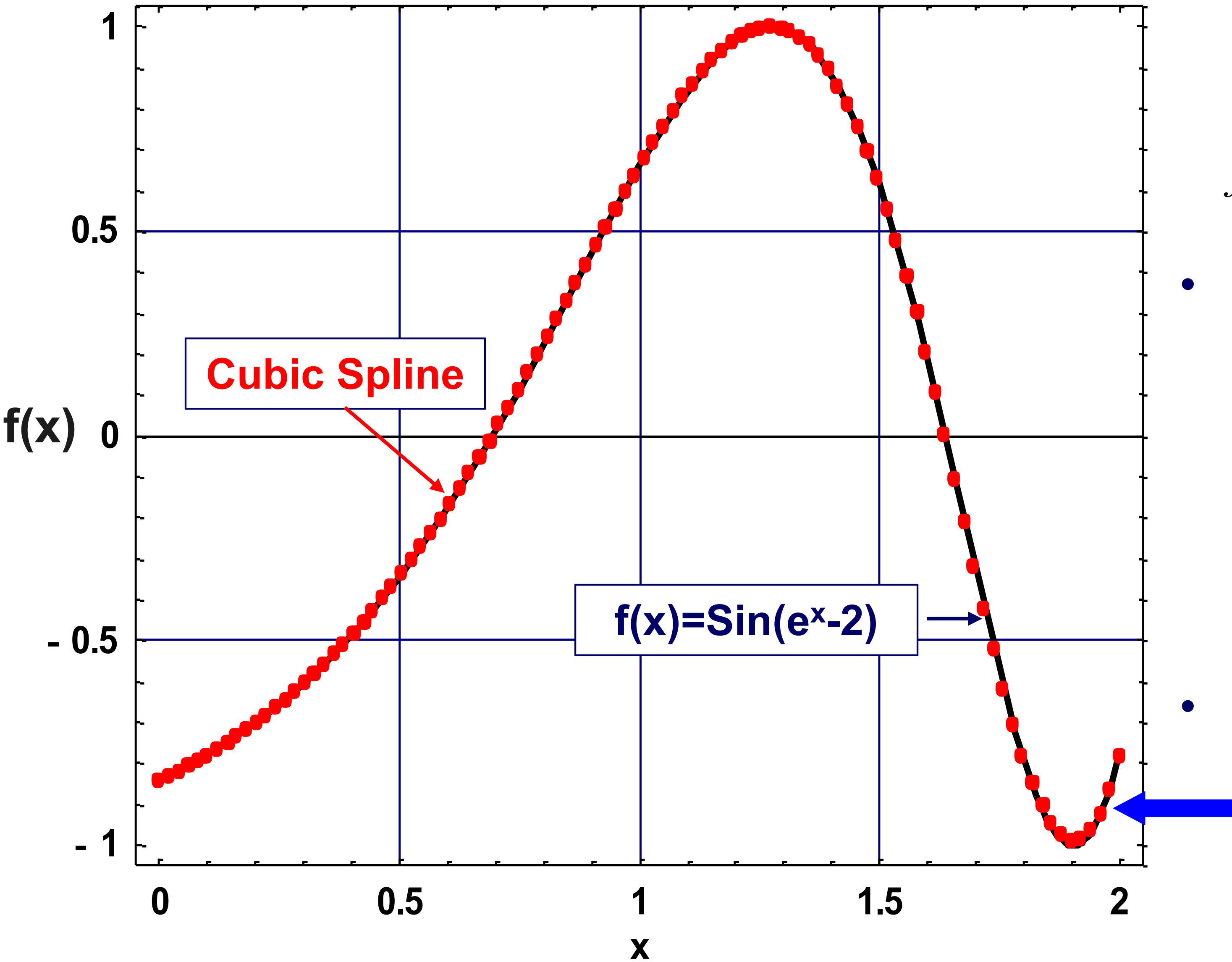
- We use 11 evenly spaced points on [0,2] and natural boundary condition at the end points
- Approximation is not accurate at the right end point

Error for Natural Cubic Spline



- Error calculated as the difference between the original function and the cubic spline approximation
- Large error can be seen towards the right end point around x=2.0

Example- Cubic Spline with Clamped Ends



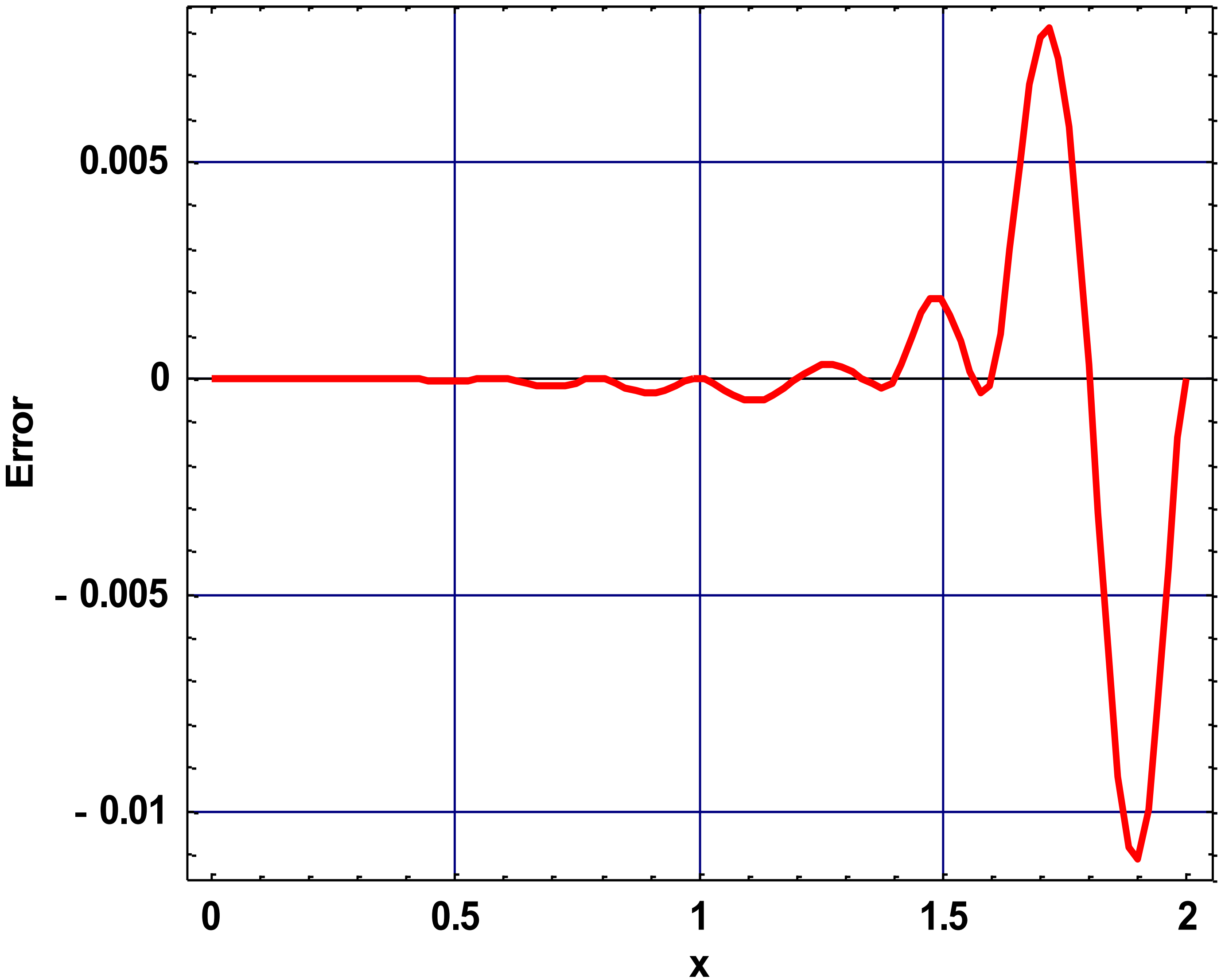
- We try to approximated the same function

$$f(x) = Sin(e^x - 2)$$

- We use 11 evenly spaced points on [0,2] and clamped boundary condition at the end points

- Better approximation towards the right end point

Error for Cubic Spline with Clamped Ends



- Error around x=2.0 is much smaller for this case compared to the natural cubic spline.
- Clamped boundary condition supply more information (its derivative) about the function near the end points

Summary

- Reviewed the implementation of cubic splines
- Worked on an example in which both low and high order Lagrange interpolation gave poor results. Cubic spline approximation was much better for the same example.
- Worked on an example to show the impact of the boundary conditions used in cubic spline approximation