Numerical Analysis

Homework 9. Spline Interpolations

102061125 Kuan-Chun Chen

1. Objective

In this homework, I will find the functions that approximate the simulated waveform shown below.

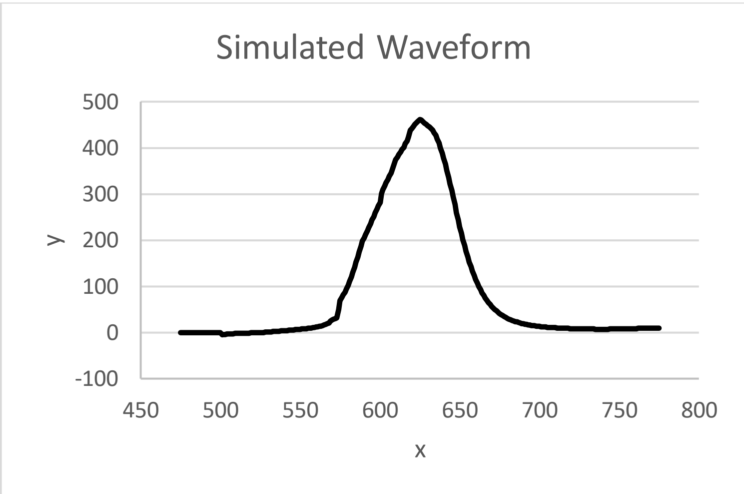

Figure 1. A simulated waveform

The data for this waveform are given in the file f301.dat. Other support points data for approximating the simulated waveform are given in f3.dat, f5.dat, f7.dat, f13.dat, and f21.dat.

2. Approach

I implement the following functions for Spline Interpolation.

$\mathbf{void\ \ splineM(int\ \ N, VEC\ \ \&X, VEC\ \ \&Y, VEC\ \ \&M);}$
$\mathbf{double\ \ spline(double\ \ x, int\ \ N, VEC\ \ \&X, VEC\ \ \&Y, VEC\ \ \&M);}$

For both functions, **X** and **Y** are two **N** vectors that contain the support points. The function **splineM** calculates the momentum vectors **M** that $\mathbf{M[\ i\ ]}$ is the second derivative at $\mathbf{X[\ i\ ]}$. Once the momentum vector is calculated, function **spline** applies interpolation to find the value at point **x**, $\mathbf{X[\ 0\ ] \le x \le X[\ N-1\ ]}$.

Algorithm. splineM

$\mathbf{void\ \ splineM(int\ \ N, VEC\ \ \&X, VEC\ \ \&Y, VEC\ \ \&M);}$

Input : N, dimension of the vectors;
X, support points $\{x_i, 0 \le \mathrm{i} \le \mathrm{n}\}$ on the x-axis;
Y, support points $\{y_i, 0 \le \mathrm{i} \le \mathrm{n}\}$ on the y-axis;
M, momentum vector;
Output : none;

Solve M vector

$$\begin{bmatrix} 2 & \lambda_0 & 0 & 0 & \cdots & 0 & 0 \\ \mu_1 & 2 & \lambda_1 & 0 & \cdots & 0 & 0 \\ 0 & \mu_2 & 2 & \lambda_2 & \cdots & 0 & 0 \\ \vdots & & & & \ddots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & \mu_{n-1} & 2 & \lambda_{n-1} \\ 0 & 0 & \cdots & \cdots & 0 & \mu_n & 2 \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ \vdots \\ M_{n-1} \\ M_n \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix}$$

where

$$h_i = x_i - x_{i-1} \ , \quad \mu_i = \frac{h_i}{h_i + h_{i+1}} \ ,$$
$$\lambda_i = \frac{h_{i+1}}{h_i + h_{i+1}} \ , \quad d_i = \frac{6}{h_i + h_{i+1}} \left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i} \right) \ ,$$
$$\lambda_0 = 0 \ , \quad d_0 = 0 \ ,$$
$$\mu_n = 0 \ , \quad d_n = 0 \ .$$

Algorithm. spline

$\mathbf{double\ \ spline(double\ \ x, int\ \ N, VEC\ \ \&X, VEC\ \ \&Y, VEC\ \ \&M);}$

Input : x, x-coordinate of the point we want to interpolate;
N, dimension of the vectors;
X, support points $\{x_i, 0 \le \mathrm{i} \le \mathrm{n}\}$ on the x-axis;
Y, support points $\{y_i, 0 \le \mathrm{i} \le \mathrm{n}\}$ on the y-axis;
M, momentum vector;
Output : y, y-coordinate of the point we want to interpolate;

$$y = \alpha_i + \beta_i (x - x_{i-1}) + \gamma_i (x - x_{i-1})^2 + \delta_i (x - x_{i-1})^3, \ \text{for} \ \ x \in [x_{i-1}, x_i] \ .$$

where

$$\alpha_i = y_{i-1} \ ,$$
$$\beta_i = \frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{6}(M_i + 2M_{i-1}) \ ,$$
$$\gamma_i = \frac{M_{i-1}}{2} \ ,$$
$$\delta_i = \frac{M_i - M_{i-1}}{6h_i} \ .$$

3. Results

3.1. Interpolated Values

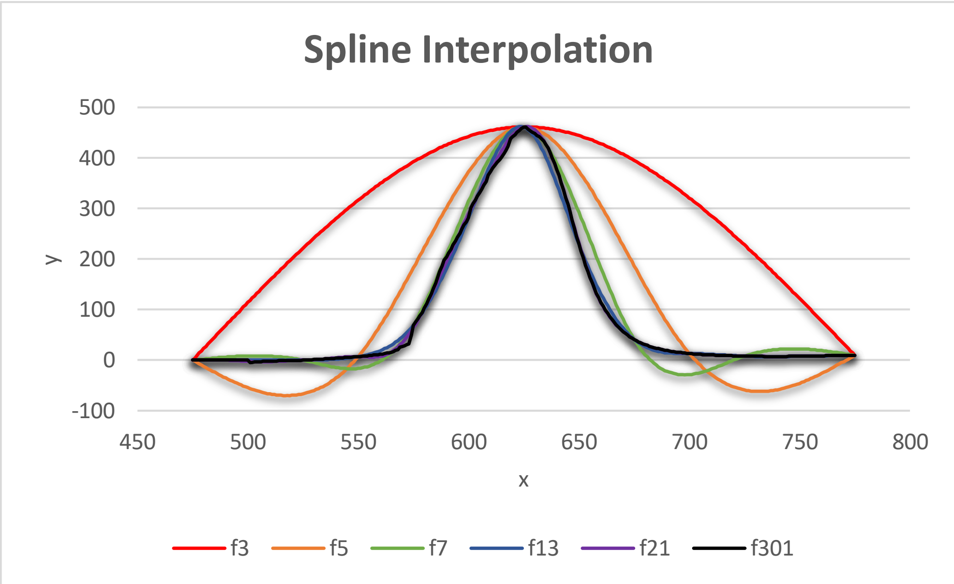

Figure 2. Spline Interpolations with different support points data

3.2. Maximum Absolute Error

f3	f5	f7	f13	f21
354.947	190.82	73.7436	29.0448	19.6417

Table 1. Maximum Absolute Error

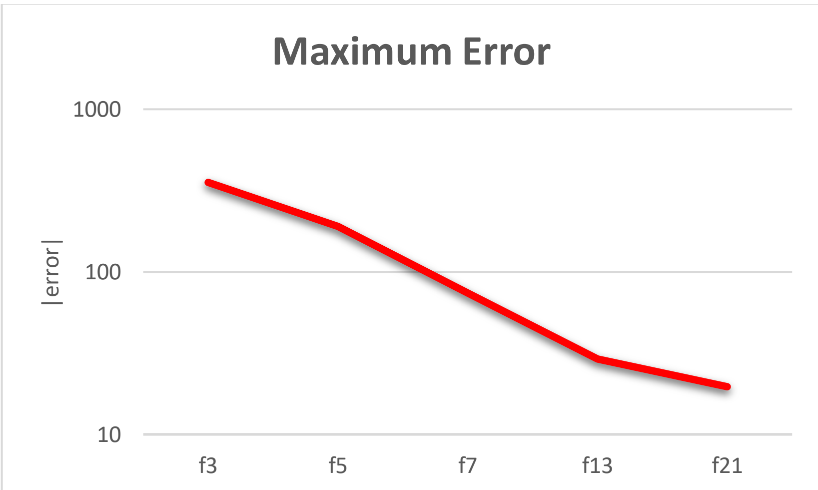

Figure 3. Maximum Absolute Error

4. Observations

f301, the black waveform, is the data of the simulated waveform, and now I am going to compare the other waveforms against it.

We can see that if we have more support points, the interpolation curve will be more accurate on the whole range.

This Figure 4 shows the absolute error on the whole range.

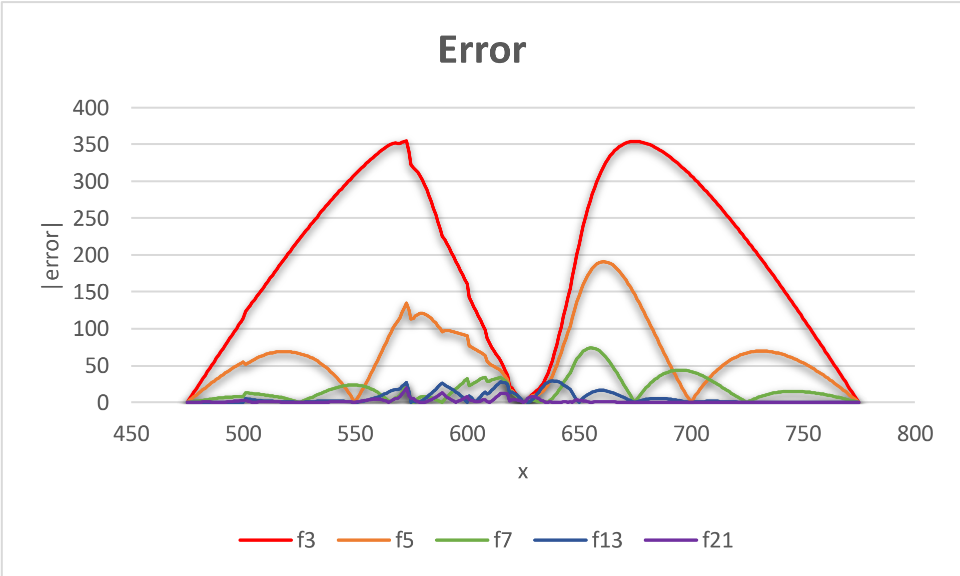

Figure 4. Absolute Error

The number of peaks is equal to the number of support points minus one. Moreover, not only the peak error but the whole range error decrease as support points increase.