Santa Clara University

AMTH 118

Numerical Methods

Project 3

Authors:
Daniel Call, Ryan Schulz,
Ruben Contreras

December 4, 2015

Introduction

The first objective of this project was to construct a natural cubic spline to represent the top and bottom parts of an airfoil. Using the constructed cubic splines, the second objective was to approximate the area enclosed by the airfoil using the trapezoid rule.

Part 1

Construction of Cubic Splines

Cubic splines are created using equation (1). Taking the derivative of (1) with respect to x, the first derivative (2) and second derivative (3) are also considered.

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3 \tag{1}$$

$$S'_j(x) = b_j(x - x_j) + 2c_j(x - x_j) + 3d_j(x - x_j)^2 \tag{2}$$

$$S''_j(x) = 2c_j + 6d_j(x - x_j) \tag{3}$$

By plugging in x_j values into the respective x, equation (1) shows:

$$a_j = f(x_j) \tag{4}$$

Thus, our a_j can be are equal to our y coordinates from the nodes.

Defining the distance between each node point, h_j is found by the following:

$$h_j = x_{j+1} - x_j \tag{5}$$

Through substituting h_j into (1) and it's two derivatives yield the following new equations.

$$a_{j+1} = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3 \tag{6}$$

$$b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2 \tag{7}$$

$$2c_{j+1} = 2c_j + 6d_j h_j \tag{8}$$

Solving (8) for d_j shows that:

$$d_j = \frac{c_{j+1} - c_j}{3h_j} \tag{9}$$

Substituting (9) into (6) results in

$$b_j = \frac{a_{j+1} - a_j}{h_j} - \frac{h_j}{3}(2c_j + c_{j+1}) \tag{10}$$

Subtracting 1 from the j values derives:

$$b_{j-1} = \frac{a_j - a_{j-1}}{h_{j-1}} - \frac{h_{j-1}}{3}(2c_{j-1} + c_j) \tag{11}$$

Substituting (9) into (7) derives

$$b_{j+1} = b_j + h_j(c_j + c_{j+1}) \tag{12}$$

Taking (10) and substituting it into (7) gives

$$h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_j c_{j+1} = \frac{3}{h_j}(a_{j+1} - a_j) - \frac{3}{h_{j-1}}(a_j - a_{j-1}) \tag{13}$$

By solving this into a system of matrix equations and MATLAB, the c_j values can be found. Using the a_j and c_j which were previously found, the b_j and
d_j values can also be found for every cubic spline by substituting a_j and c_j into equations (9) and (10).

Airfoil

Figure 1 shows the airfoil considered. To find the interpolating cubic spline for the top curve of the airfoil, points where chosen along the curve through which the approximating cubic spline was to pass through using Adobe Photoshop CS6. Figure 2 shows the airfoil with the respective nodes chosen. Table 1 lists the exact coordinates of 25 data points that where chosen. For the purposes of this project, the fin on the top curve was disregarded.

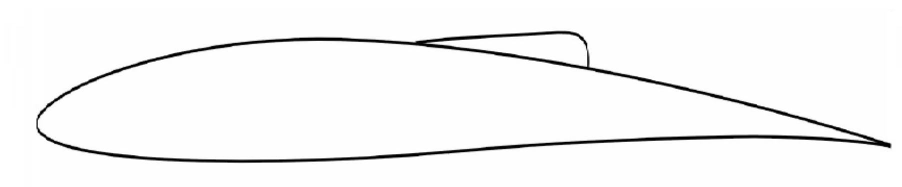

Figure 1: The object under consideration: an airfoil.

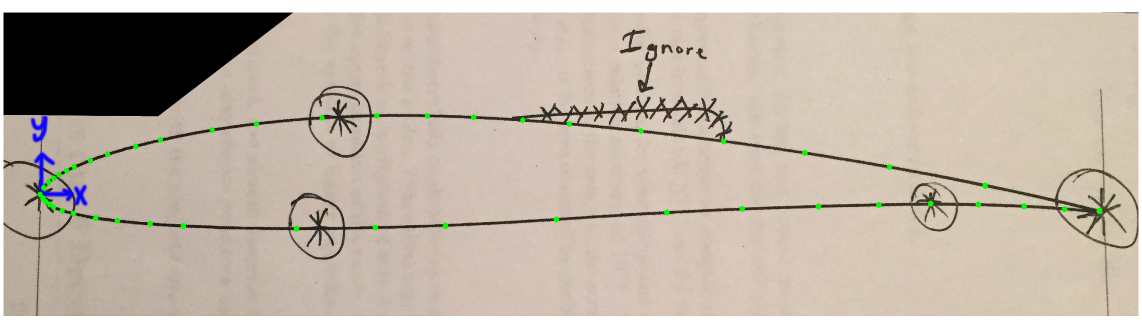

Figure 2: The nodes on the airfoil that were chosen using Adobe Photoshop CS6.

Top Curve		
Point	x (in)	y (in)
1	0.4	1.52
2	0.51	1.74
3	0.69	1.92
4	0.86	2.03
5	1.04	2.15
6	1.34	2.29
7	1.58	2.41
8	2.15	2.62
9	2.75	2.82
10	3.72	3.08
11	4.58	3.29
12	6.4	3.61
13	7.94	3.8
14	10.25	4.01
15	12.17	4.05
16	13.92	4.08
17	15.53	4.02
18	17.25	3.93
19	18.89	3.8
20	21.4	3.57
21	24.26	3.24
22	27.13	2.86
23	30.08	2.41
24	33.41	1.82
25	37.38	0.98

Bottom Curve		
Point	x	y
1	0.4	1.52
2	0.48	1.42
3	0.58	1.3
4	0.7	1.21
5	0.83	1.13
6	1.15	1.01
7	1.54	0.91
8	2.35	0.77
9	3.08	0.68
10	4.23	0.57
11	5.38	0.51
12	7.5	0.44
13	9.34	0.41
14	12.11	0.46
15	14.55	0.53
16	18.43	0.73
17	22.26	0.93
18	24.93	1.06
19	27.57	1.14
20	29.7	1.21
21	31.51	1.22
22	33.17	1.21
23	34.77	1.13
24	36.18	1.05
25	37.38	0.98

Table 1: Node coordinates of the airfoil that created the cubic spline

Finding the interpolating cubic spline for the bottom of the airfoil was done in a similar fashion for both top and bottom splines. Points where chosen along the curves through which the approximating cubic curve was to pass through. Table 1 lists the coordinates of data points that where chosen relative to the an origin placed to the lower left side of the airfoil shown of Figure 2.

Top Curve Constants				
j	a	b	c	d
1	1.52	1.9485	0	4.2528
2	1.74	1.1029	1.4034	-10.9732
3	1.92	1.1886	-4.5221	7.8620
4	2.03	0.7053	-0.5125	1.6557
5	2.15	0.4621	0.3816	-1.2212
6	2.29	0.5947	-0.7175	1.3463
7	2.41	0.3180	0.2518	-0.2865
8	2.62	0.4223	-0.2381	0.1498
9	2.82	0.2681	0.0316	-0.0327
10	3.08	0.2787	-0.0635	0.0272
11	3.29	0.1853	0.0067	-0.0065
12	3.61	0.1606	-0.0290	0.0031
13	3.8	0.1136	-0.0145	0.0020
14	4.01	0.0416	-0.0003	-0.0055
15	4.05	0.0463	-0.0318	0.0086
16	4.08	-0.0364	0.0135	-0.0087
17	4.02	-0.0214	-0.0286	0.0062
18	3.93	-0.0770	0.0032	-0.0028
19	3.8	-0.0745	-0.0105	0.0015
20	3.57	-0.1109	0.0006	-0.0007
21	3.24	-0.1192	-0.0058	0.0004
22	2.86	-0.1452	-0.0022	-0.0001
23	2.41	-0.1654	-0.0032	-0.0001
24	1.82	-0.2001	-0.0043	0.0004
25	0.98	--	0	--

Bottom Curve Constants				
j	a	b	c	d
1	1.52	-1.2535	0	0.5396
2	1.42	-1.1931	0.1295	-1.9856
3	1.3	-0.8268	-0.4662	9.2155
4	1.21	-0.8559	2.8514	-7.7010
5	1.13	-0.3992	-0.1520	0.7116
6	1.01	-0.3997	0.5311	-0.4198
7	0.91	-0.2120	0.0400	0.0103
8	0.77	-0.1609	0.0650	-0.0184
9	0.68	-0.1174	0.0247	-0.0050
10	0.57	-0.0647	0.0074	0.0031
11	0.51	-0.0599	0.0180	-0.0025
12	0.44	-0.0200	0.0019	0.0001
13	0.41	0.0052	0.0022	0.0009
14	0.46	0.0139	0.0095	-0.0014
15	0.53	0.0472	-0.0009	0.0005
16	0.73	0.0413	0.0051	-0.0006
17	0.93	0.0504	-0.0016	0.0004
18	1.06	0.0347	0.0013	-0.0011
19	1.14	0.0391	-0.0076	0.0022
20	1.21	0.0066	0.0063	-0.0038
21	1.22	0.0077	-0.0145	0.0037
22	1.21	-0.0418	0.0042	-0.0058
23	1.13	-0.0357	-0.0236	0.0062
24	1.05	-0.0603	0.0025	-0.0007
25	0.98	--	0	--

Table 2: Coefficients produced from data points in table .

Using equations (4) and (5), the a_j and the h_j were found from the nodes. Ultimately after finding h_j, a matrix was used to solve for c_j values shown in equation (13). Our program then used the equations (7) and (9). The constants for the top and bottom spline were found as shown in Table 2.

The resulting coefficients allow for 48 unique equations to be made for the top and bottom splines, with 24 equations connecting each of the 25 nodes for both parts of the airfoil. By plugging each of these 48 unique sets of coefficients into (13), the equations are found in the form of:

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$$

Results and Discussion

Considering the 48 equations from the coefficients in Table 2 and plotting the results using the written MATLAB program yielded the cubic spline representation of the airfoil shown in Figure 3.

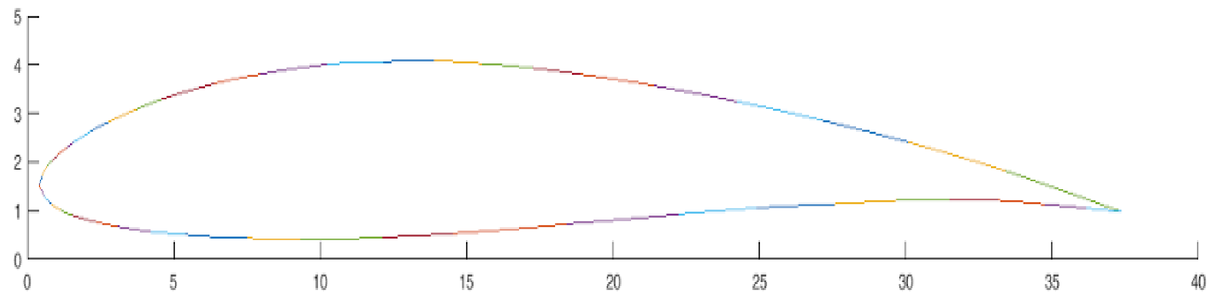

Figure 3: Cubic spline representation of airfoil.

The generated spline curve is nearly identical to the profile shown in Figure 1. Each of the unique cubic splines are plotted with a different color in Figure 3. Deviations from an identical representation of the airfoil can be attributed to the number of nodes considered. A more accurate representation for the airfoil can be obtained by increasing the number of data points.

Part 2

Area Approximation Using Trapezoidal Rule

The Trapezoidal Rule is a numerical integration technique used to approximate definite integrals. This numerical integration technique approximates the area under a curve as the area of a trapezoid. The error of this approximation varies based on the second derivative of the function that is being approximated.

$$A = \frac{dx}{2}[f(1) + 2f(2) + 2f(3) + ... + 2f(n-1) + f(n)] \tag{14}$$

The cubic splines obtained were of a general cubic polynomial form with four constants:

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x_{j+1} - x_j)^2 + d_j(x_{j+1} - x_j)^3 \tag{15}$$

To obtain the area between the splines, the area of each individual spline had to be calculated. The spacing between each nodes for the Trapezoidal rule was found using:

$$dx = \frac{b-a}{n} \tag{16}$$

The specific equidistant x values were then calculated using:

$$x_i = x_1 + dx(i-1) \tag{17}$$

with $i = 1 : n$ number of nodes.

Matlab was used to compare which of the 24 cubic spline equations is needed to find the y values for the respective n number of equidistant x values.

Once the respective y values are found, MATLAB was used to calculate the top area using equation (14). The other splines area was then found using the same process. Area A_k was then found:

$$A_k = A_{top} - A_{bottom} \tag{18}$$

In order to reach an accuracy of 10^{-8}, the n of nodes were then doubled to $2n$. A new $2n$ number of equidistant x values and their respective y values were then found to calculate A_{k+1}. The following percent error formula was used to determine if the relative accuracy of 10^{-8} was reached.

$$Error = |\frac{(A_{k+1} - A_k)}{A_{k+1}}| \tag{19}$$

If the relative accuracy was not 10^{-8}, the $2n$ number of nodes used to calculate A_{k+1} is doubled to calculate A_{k+2}. Following iterations are continued until the number of nodes finds the area A_k to be within an accuracy of 10^{-8}.

Running the MATLAB program for part two, the area between the two spline functions was calculated to be **84.71147833** from equation (18) with an error of **4.9843E-9** from equation (19).