

AE 370: HW2

Linyi Hou

February 12, 2020

Problem 1

1.a

The four conditions used to solve for coefficients for cubic splines are:

$$\begin{aligned} s_i(x_i) &= f(x_i) & , i = 1, \dots, n \\ s_i(x_{i+1}) &= f(x_{i+1}) & , i = 1, \dots, n \\ s'_i(x_{i+1}) &= s'_{i+1}(x_{i+1}), i = 1, \dots, n-1 \\ s''_i(x_{i+1}) &= s''_{i+1}(x_{i+1}), i = 1, \dots, n-1 \end{aligned}$$

for each $s_i(x)$ defined as

$$s_i(x) = \frac{c_{i,1}}{6(x_i - x_{i+1})}(x - x_{i+1})^3 + \frac{c_{i,2}}{6(x_{i+1} - x_i)}(x - x_i)^3 + c_{i,3}x + c_{i,4}$$

When $n = 2$, the expanded equations become:

$$\begin{aligned} s_1(x_1) &= f(x_1) \\ s_1(x_2) &= f(x_2) \\ s'_1(x_2) &= s'_2(x_2) \\ s''_1(x_2) &= s''_2(x_2) \\ s_2(x_2) &= f(x_2) \\ s_2(x_3) &= f(x_3) \end{aligned}$$

with natural conditions

$$\begin{aligned} s''_1(x_1) &= 0 \\ s''_2(x_3) &= 0 \end{aligned}$$

The linear system thus becomes:

$$\begin{bmatrix} \frac{\Delta x^2}{6} & 0 & x_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{\Delta x^2}{6} & x_2 & 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{\Delta x^2}{6} & x_2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\Delta x^2}{6} & 0 & x_2 & 1 \\ 0 & 0 & 0 & 0 & 0 & \frac{\Delta x^2}{6} & x_3 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_{1,1} \\ c_{1,2} \\ c_{1,3} \\ c_{1,4} \\ c_{2,1} \\ c_{2,2} \\ c_{2,3} \\ c_{2,4} \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ 0 \\ 0 \\ f(x_2) \\ f(x_3) \\ 0 \\ 0 \end{bmatrix}$$

1.b

The plots and code have been attached in Appendices A and B, respectively.

1.c

Splines do not suffer from the same issues from large numbers of equispaced points as global polynomial variables. This is because each segment of the spline is essentially interpolated independently from a small number of points. For each individual segment, a small number of interpolation points creates a good local approximation, and by enforcing continuity and smoothness conditions, the combined function is well-behaved and a good approximation.

Problem 2

2.a, 2.b

The plots and code have been attached in Appendices A and B, respectively.

2.c

Trigonometric interpolation by construction creates periodic function with a whole number of periods as the approximation. Therefore the original function must have equal values at the beginning and end of the approximation domain ($f(x_1) = f(x_{2n+1})$).

A choice of function whose beginning and end values are not equal will cause the slope of the function at its bounds to increase dramatically as the number of interpolation points increases. The function in 2.a had equal values at $f(0)$ and $f(2\pi)$ and is also periodic, thus trigonometric approximation is well-suited. On the other hand, the function in 2.b does not have equal values at $f(0)$ and $f(2\pi)$ and is a linear function with no periodicity, thus it is not a good candidate for trigonometric approximation.

Appendix A: Figures

Appendix A: Plots for 1.b

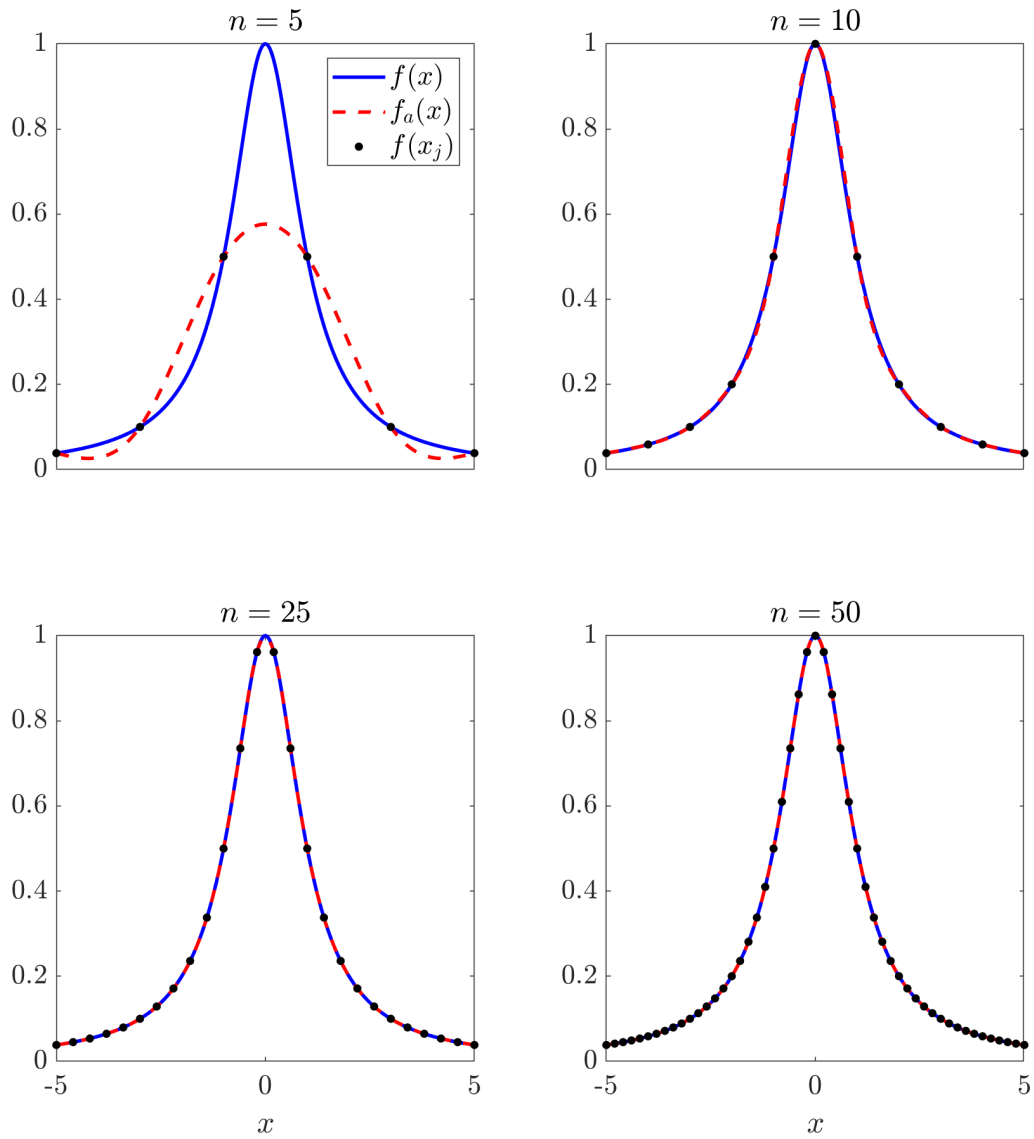


Figure 1: Problem 1.b Function Plots

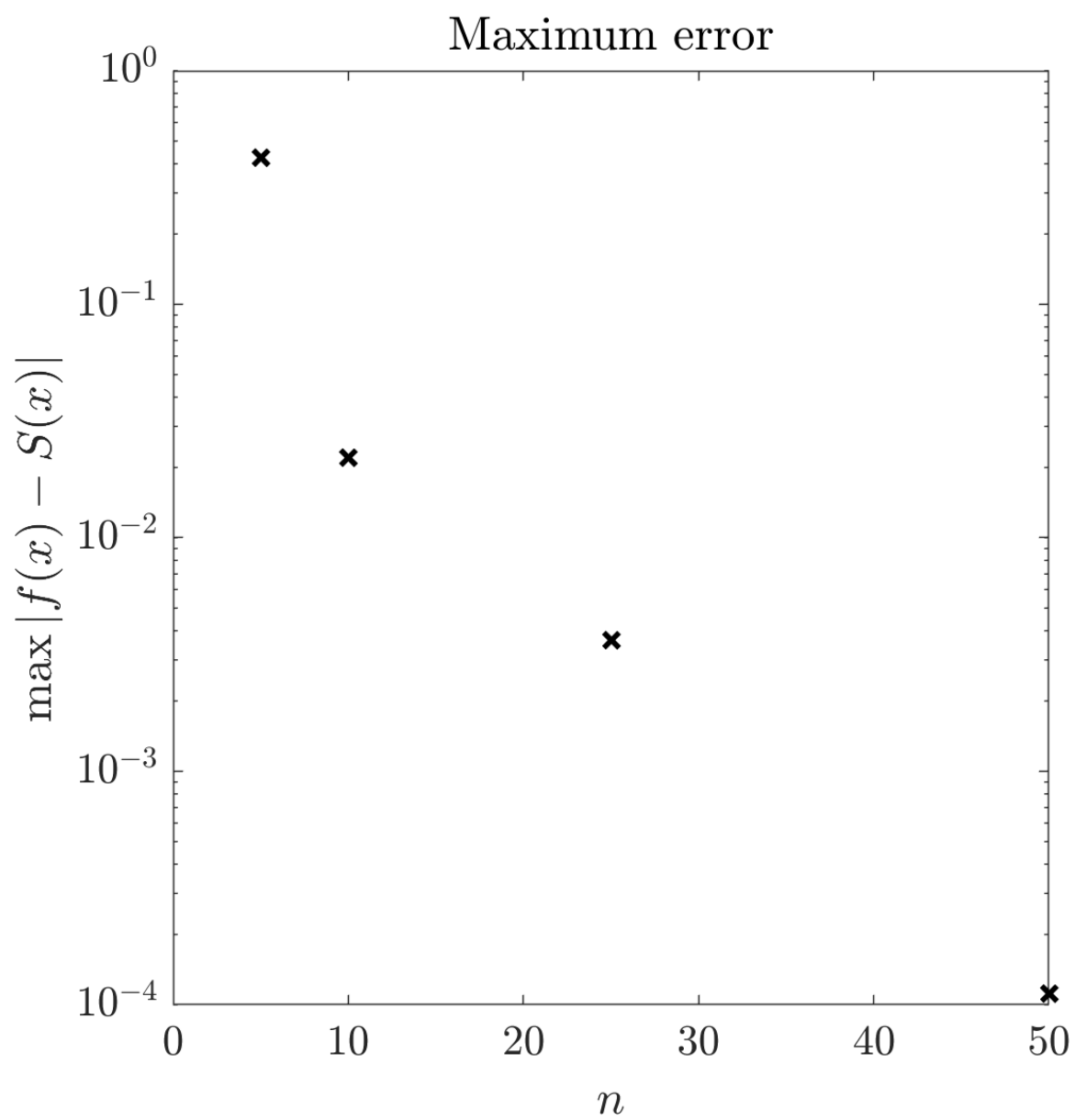


Figure 2: Problem 1.b Error Plots

Appendix A: Plots for 2.a

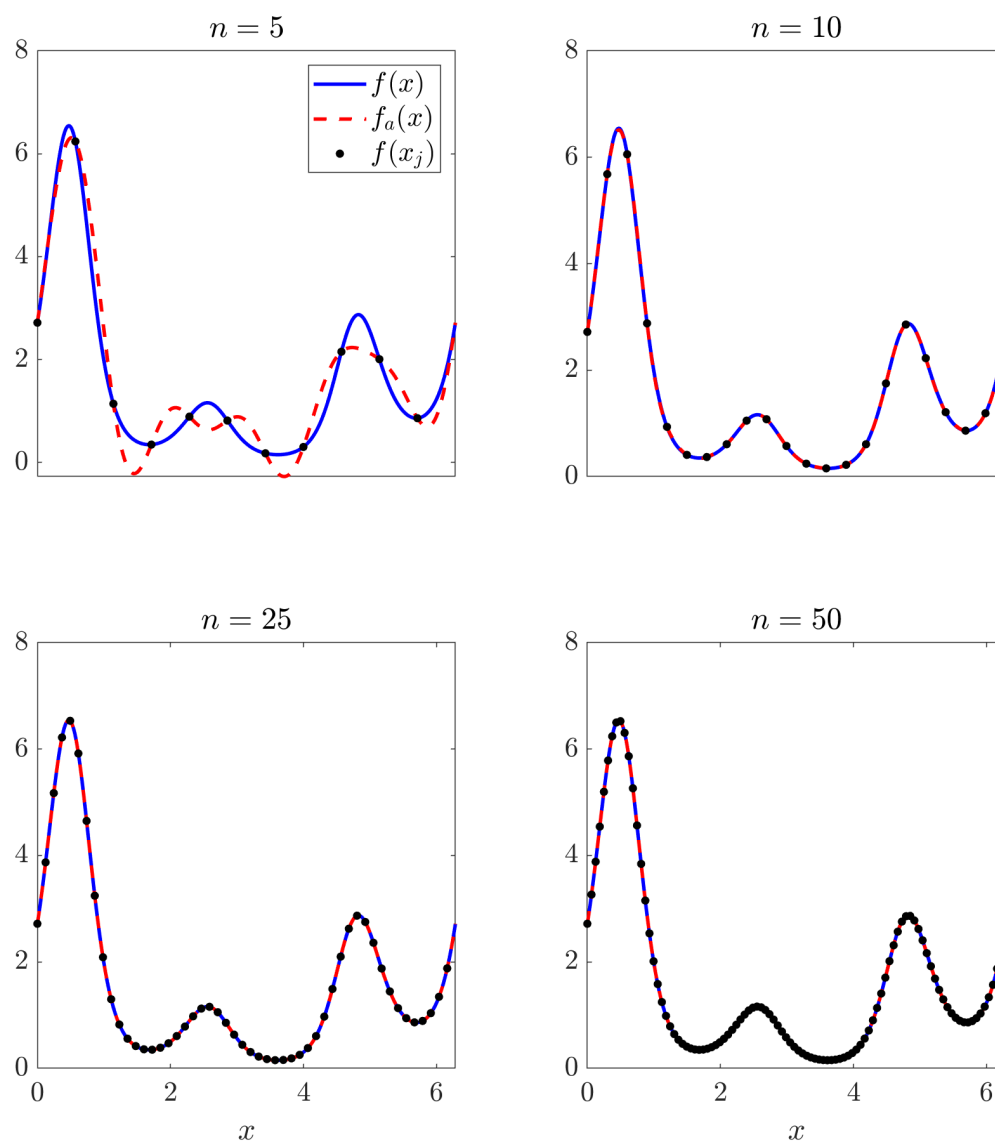


Figure 3: Problem 2.a Function Plots

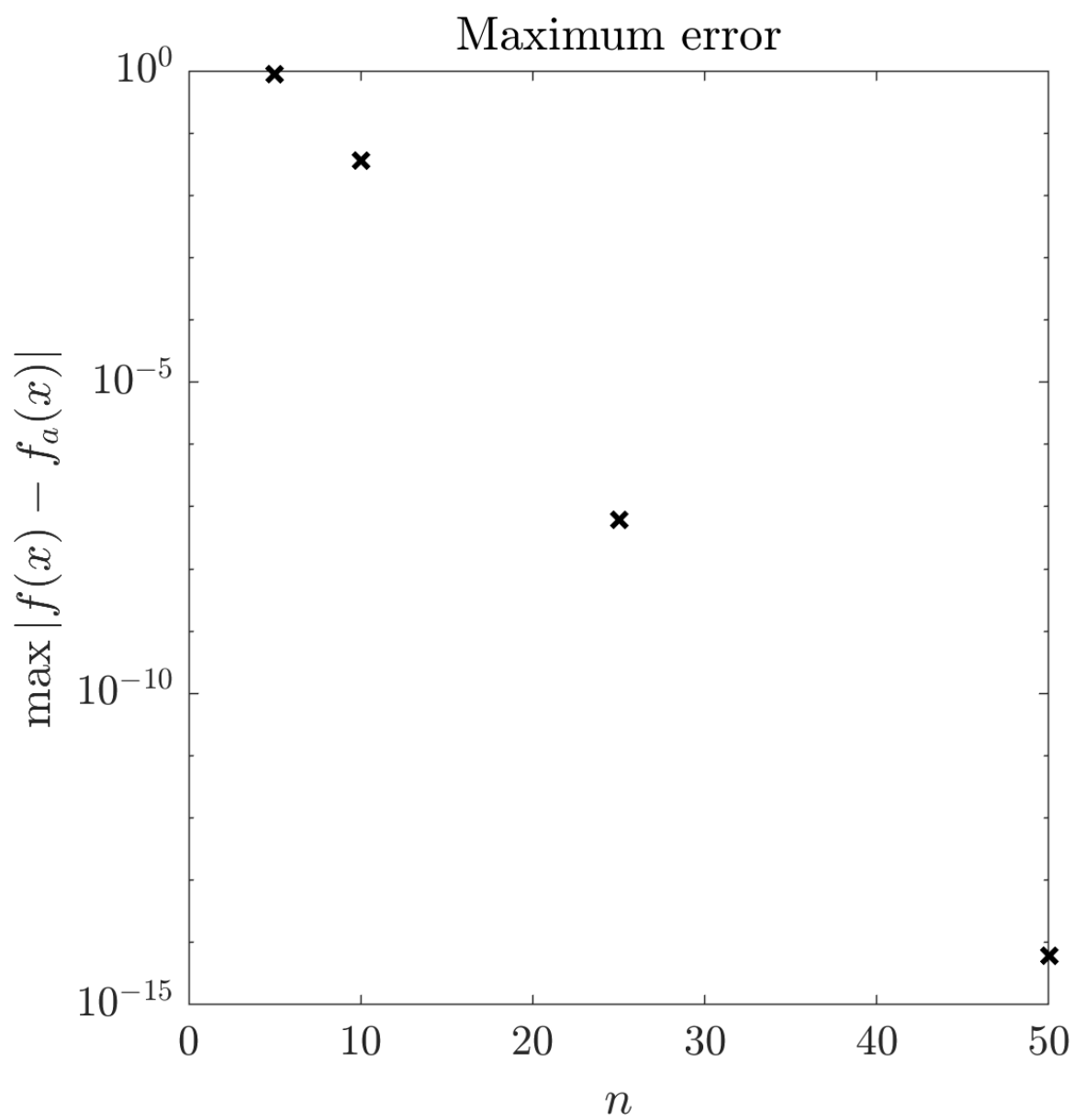


Figure 4: Problem 2.a Error Plots

Appendix A: Plots for 2.b

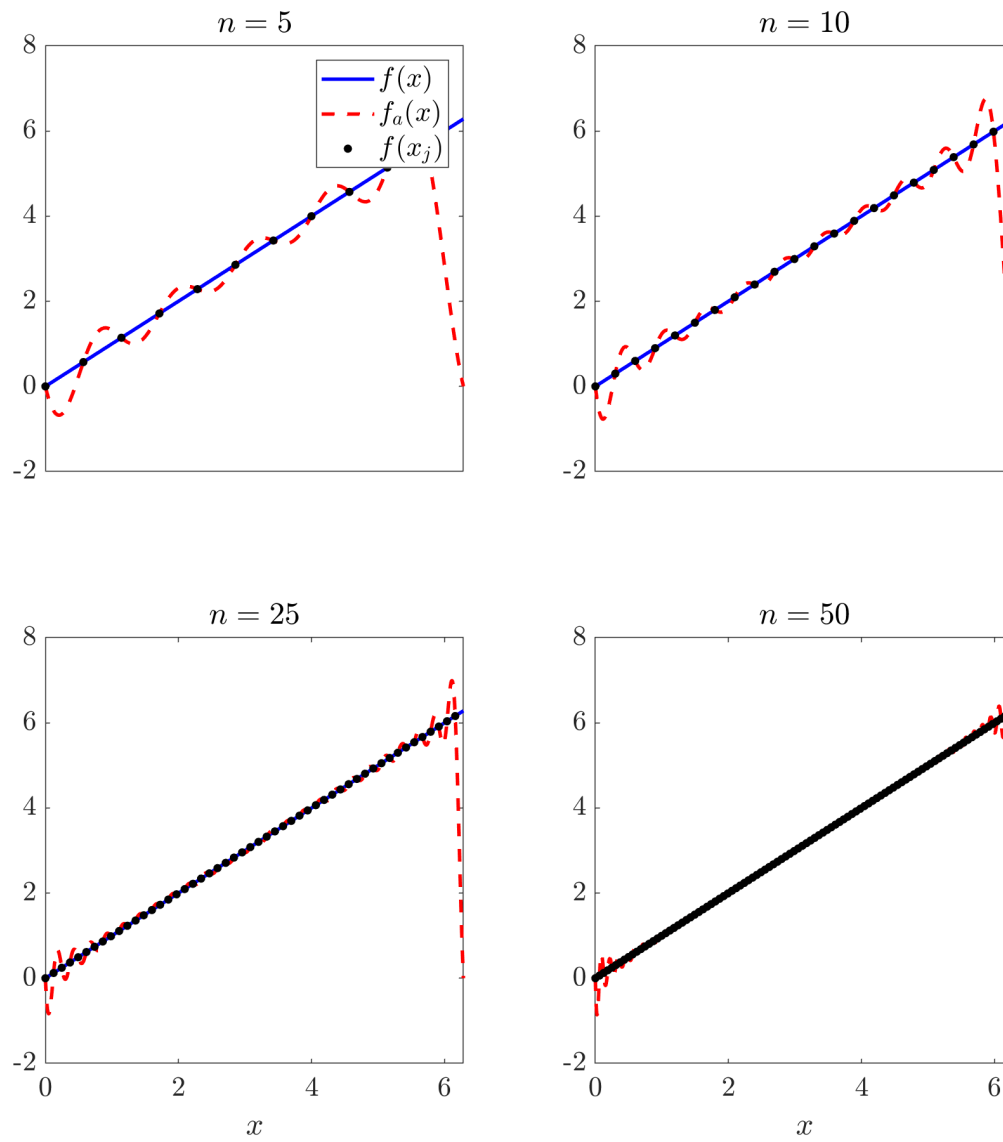


Figure 5: Problem 2.b Function Plots

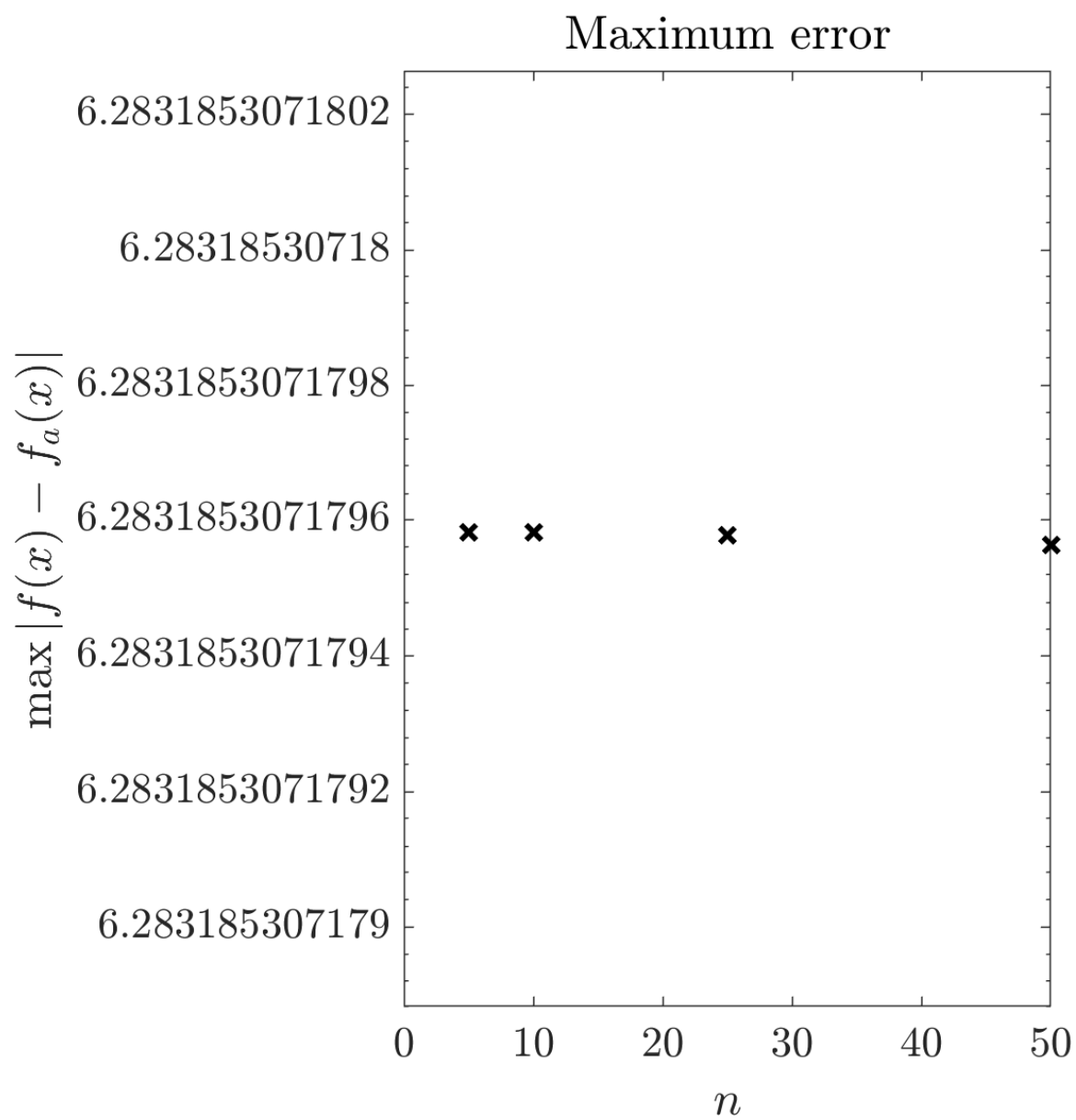


Figure 6: Problem 2.b Error Plots

Appendix B: Code

Appendix B: Code for 1.b

```
1 %% Problem 1
2 clear;clc
3
4 nvect = [5, 10, 25, 50];
5 xl = -5; xr = 5;
6 %function to approx
7 f = @(x) 1./(1+x.^2) ;
8 %error vector:
9 err = zeros(size(nvect));
10 for j = 1 : length( nvect )
11     %define current n
12     n = nvect( j );
13     %define interp points (equally spaced)
14     xj = (xl : (xr-xl)/n : xr)';
15     dx = xj(2)-xj(1);
16     %—build & solve lin system for the c_{i,k} (i = 1,...,n; k =
        1,...,4)
17     %—for natural splines
18     A = zeros( 4*n ); %initialize matrix
19     g = zeros( 4*n, 1 ); %initialize RHS vector
20     %Build A matrix & f vector
21     for jj = 1 : n
22         ind = 4*(jj - 1) + 1;
23         %condition (1) from partial solution doc
24         A( ind, ind ) = dx^2 / 6;
25         A( ind, ind + 1 ) = 0;
26         A( ind, ind + 2 ) = xj(jj);
27         A( ind, ind + 3 ) = 1;
28         g( ind ) = f(xj(jj));
29         %condition (2) from partial solution doc
30         A( ind + 1, ind ) = 0;
31         A( ind + 1, ind + 1 ) = dx^2 / 6;
32         A( ind + 1, ind + 2 ) = xj(jj + 1);
33         A( ind + 1, ind + 3 ) = 1;
34         g( ind + 1 ) = f(xj(jj + 1));
35         %derivative conditions
```

```

36      %(careful here! index on derivs only goes to n-1...)
37      if jj ~= n
38          %condition (3) from partial solution doc
39          A( ind+2, ind ) = 0;
40          A( ind+2, ind + 1 ) = dx/2;
41          A( ind+2, ind + 2 ) = 1;
42          A( ind+2, ind + 3 ) = 0;
43          A( ind+2, ind + 4 ) = dx/2;
44          A( ind+2, ind + 5 ) = 0;
45          A( ind+2, ind + 6 ) = -1;
46          A( ind+2, ind + 7 ) = 0;
47          %condition (4) from partial solution doc
48          A( ind+3, ind ) = 0;
49          A( ind+3, ind + 1 ) = 1;
50          A( ind+3, ind + 2 ) = 0;
51          A( ind+3, ind + 3 ) = 0;
52          A( ind+3, ind + 4 ) = -1;
53          A( ind+3, ind + 5 ) = 0;
54          A( ind+3, ind + 6 ) = 0;
55          A( ind+3, ind + 7 ) = 0;
56      else
57          %s_1''(x_1) = 0 (eqn (11))
58          A( ind+2, 1 ) = 1;
59          A( ind+2, 2 ) = 0;
60          A( ind+2, 3 ) = 0;
61          A( ind+2, 4 ) = 0;
62          %s_n''(x_{n+1}) = 0 (eqn (12))
63          A( ind+3, ind ) = 0;
64          A( ind+3, ind + 1 ) = 1;
65          A( ind+3, ind + 2 ) = 0;
66          A( ind+3, ind + 3 ) = 0;
67      end
68      g( ind + 2 ) = 0;
69      g( ind + 3 ) = 0;
70  end
71  %solve for coeffs:
72  c = A \ g;
73  %—
74  %—plot the spline interpolant S(x)
75  xx = linspace( xl, xr, 1000 );

```

```

76 S = zeros( size( xx ) );
77 for jj = 1 : n
78     ind = 4*(jj - 1) + 1;
79     indxx = ( xx >= xj( jj ) & xx <= xj( jj + 1 ) );
80     xxc = xx( indxx ~= 0 );
81     S( indxx ~= 0 ) = c( ind ) * ( xxc - xj(jj+1) ).^3./ ...
82         ( 6.*(xj(jj) - xj(jj+1)) ) + c( ind+1 ) * ( xxc - ...
83         xj(jj) ).^3./( 6.*(xj(jj+1) - xj(jj)) ) + c( ind + 2 ) ...
84         .* xxc + c( ind + 3 );
85 end
86 if j <= 4
87     figure(1)
88     subplot(2,2,j)
89     plot( xx, f(xx), 'b-', 'linewidth', 2 ), hold on
90     plot( xx, S, 'r—', 'linewidth', 2 )
91     plot( xj, f(xj), 'k.', 'markersize', 16 )
92     %make plot pretty
93     title( ['$n = ', num2str( n ), '$'] , 'interpreter', 'latex', ...
94         'fontsize', 16)
95     if j == 1
96         h = legend( '$f(x)$', '$f_a(x)$', '$f(x_j)$');
97     end
98     if j <= 2
99         set( gca, 'XTick', [] )
100     else
101         xlabel( '$x$', 'interpreter', 'latex', 'fontsize', 16)
102     end
103     set(h, 'location', 'NorthEast', 'Interpreter', 'Latex', 'fontsize',
104         ', 16 )
105     set(gca, 'TickLabelInterpreter','latex', 'fontsize', 16 )
106     set(gcf, 'PaperPositionMode', 'manual')
107     set(gcf, 'Color', [1 1 1])
108     set(gca, 'Color', [1 1 1])
109     set(gcf, 'PaperUnits', 'centimeters')
110     set(gcf, 'PaperSize', [25 25])
111     set(gcf, 'Units', 'centimeters' )
112     set(gcf, 'Position', [0 0 25 25])
113     set(gcf, 'PaperPosition', [0 0 25 25])
114 end
%—

```

```

115     %—compute error
116     err(j) = max(abs( f(xx) - S ) );
117 end
118 figure(1)
119 print( '-dpng', 'pl_vary_n', '-r200' )
120 %plot error
121 figure(100)
122 semilogy( nvect, err, 'kx', 'markersize', 8, 'linewidth', 2 )
123 %make plot pretty
124 title( 'Maximum error', 'interpreter', 'latex', 'fontsize', 16)
125 xlabel( '$n$', 'interpreter', 'latex', 'fontsize', 16)
126 ylabel( '$\max|f(x) - S(x)|$', 'interpreter', 'latex', 'fontsize', 16)
127 set(gca, 'TickLabelInterpreter','latex', 'fontsize', 16 )
128 set(gcf, 'PaperPositionMode', 'manual')
129 set(gcf, 'Color', [1 1 1])
130 set(gca, 'Color', [1 1 1])
131 set(gcf, 'PaperUnits', 'centimeters')
132 set(gcf, 'PaperSize', [15 15])
133 set(gcf, 'Units', 'centimeters' )
134 set(gcf, 'Position', [0 0 15 15])
135 set(gcf, 'PaperPosition', [0 0 15 15])
136 svnm = 'pl_error';
137 print( '-dpng', svnm, '-r200' )

```

Appendix B: Code for 2.a

```
1 %% Problem 2a
2 close all
3 clear;clc
4
5 nvect = [5, 10, 25, 50];
6 xl = 0; xr = 2*pi;
7 %function to approx
8 f = @(x) exp(cos(x)+sin(3*x));
9 %error vector:
10 err = zeros(size(nvect));
11 for j = 1 : length( nvect )
12     %define current n
13     n = nvect( j );
14     %define interp points (equally spaced)
15     xj = (xl : (xr-xl)/(2*n+1) : xr)'; % 2n+2 terms
16     % don't get repeated points at 2*pi
17     xj = xj(1:end-1); % 2n+1 terms
18     g = zeros( 2*n+1, 1 ); %initialize RHS vector
19     %bulid f vector and use fft() to find c vector
20     for jj = 1:2*n+1
21         g(jj) = f(xj(jj));
22     end
23     %solve for coeffs:
24     c = fft(g) / (2*n+1);
25     %reshape c to sort coefficients
26     c = [c(n+2:end);c(1:n+1)];
27 %     c = real(c);
28 %—
29 %—plot the periodic approximation
30 xx = linspace( xl, xr, 1000 );
31 fa = zeros(size(xx));
32 for jj = -n:n
33     fa = fa + c(jj+n+1).*exp(1i*jj.*xx);
34 end
35 fa = real(fa);
36 if j <= 4
37     figure(2)
38     subplot(2,2,j)
```

```

39     plot( xx, f(xx), 'b-', 'linewidth', 2 ), hold on
40     plot( xx, fa, 'r—', 'linewidth', 2 )
41     plot( xj, f(xj), 'k.', 'markersize', 16 )
42     %make plot pretty
43     title( ['$n = ', num2str( n ), '$'] , 'interpreter', 'latex', ...
44           'fontsize', 16)
45     if j == 1
46         h = legend( '$f(x)$', '$f_a(x)$', '$f(x_j)$');
47     end
48     if j <= 2
49         set( gca, 'XTick', [] )
50     else
51         xlabel( '$x$', 'interpreter', 'latex', 'fontsize', 16)
52     end
53     set(h, 'location', 'NorthEast', 'Interpreter', 'Latex', 'fontsize
54         ', 16 )
55     set(gca, 'TickLabelInterpreter','latex', 'fontsize', 16 )
56     set(gcf, 'PaperPositionMode', 'manual')
57     set(gcf, 'Color', [1 1 1])
58     set(gca, 'Color', [1 1 1])
59     set(gcf, 'PaperUnits', 'centimeters')
60     set(gcf, 'PaperSize', [25 25])
61     set(gcf, 'Units', 'centimeters' )
62     set(gcf, 'Position', [0 0 25 25])
63     set(gcf, 'PaperPosition', [0 0 25 25])
64     end
65     %—
66     %—compute error
67     err(j) = max(abs( f(xx) - fa ) );
68 end
69 figure(2)
70 print( '-dpng', 'p2a_vary_n', '-r200' )
71 %plot error
72 figure(200)
73 semilogy( nvect, err, 'kx', 'markersize', 8, 'linewidth', 2 )
74 %make plot pretty
75 title( 'Maximum error' , 'interpreter', 'latex', 'fontsize', 16)
76 xlabel( '$n$', 'interpreter', 'latex', 'fontsize', 16)
77 ylabel( '$\max|f(x) - f_a(x)|$', 'interpreter', 'latex', 'fontsize', 16)
78 set(gca, 'TickLabelInterpreter','latex', 'fontsize', 16 )

```

```
78 set(gcf, 'PaperPositionMode', 'manual')
79 set(gcf, 'Color', [1 1 1])
80 set(gca, 'Color', [1 1 1])
81 set(gcf, 'PaperUnits', 'centimeters')
82 set(gcf, 'PaperSize', [15 15])
83 set(gcf, 'Units', 'centimeters' )
84 set(gcf, 'Position', [0 0 15 15])
85 set(gcf, 'PaperPosition', [0 0 15 15])
86 svnm = 'p2a_error';
87 print( '-dpng', svnm, '-r200' )
```

Appendix B: Code for 2.b

```
1 %% Problem 2b
2 close all
3 clear;clc
4
5 nvect = [5, 10, 25, 50];
6 xl = 0; xr = 2*pi;
7 %function to approx
8 f = @(x) x;
9 %error vector:
10 err = zeros(size(nvect));
11 for j = 1 : length( nvect )
12     %define current n
13     n = nvect( j );
14     %define interp points (equally spaced)
15     xj = (xl : (xr-xl)/(2*n+1) : xr)'; % 2n+2 terms
16     % don't get repeated points at 2*pi
17     xj = xj(1:end-1); % 2n+1 terms
18     g = zeros( 2*n+1, 1 ); %initialize RHS vector
19     %bulid f vector and use fft() to find c vector
20     for jj = 1:2*n+1
21         g(jj) = f(xj(jj));
22     end
23     %solve for coeffs:
24     c = fft(g) / (2*n+1);
25     %reshape c to sort coefficients
26     c = [c(n+2:end);c(1:n+1)];
27 %     c = real(c);
28 %—
29 %—plot the periodic approximation
30 xx = linspace( xl, xr, 1000 );
31 fa = zeros(size(xx));
32 for jj = -n:n
33     fa = fa + c(jj+n+1).*exp(1i*jj.*xx);
34 end
35 fa = real(fa);
36 if j <= 4
37     figure(3)
38     subplot(2,2,j)
```



```

39     plot( xx, f(xx), 'b-', 'linewidth', 2 ), hold on
40     plot( xx, fa, 'r—', 'linewidth', 2 )
41     plot( xj, f(xj), 'k.', 'markersize', 16 )
42     %make plot pretty
43     title( ['$n = ', num2str( n ), '$'] , 'interpreter', 'latex', ...
44           'fontsize', 16)
45     if j == 1
46         h = legend( '$f(x)$', '$f_a(x)$', '$f(x_j)$');
47     end
48     if j <= 2
49         set( gca, 'XTick', [] )
50     else
51         xlabel( '$x$', 'interpreter', 'latex', 'fontsize', 16)
52     end
53     set(h, 'location', 'NorthEast', 'Interpreter', 'Latex', 'fontsize
54         ', 16 )
55     set(gca, 'TickLabelInterpreter','latex', 'fontsize', 16 )
56     set(gcf, 'PaperPositionMode', 'manual')
57     set(gcf, 'Color', [1 1 1])
58     set(gca, 'Color', [1 1 1])
59     set(gcf, 'PaperUnits', 'centimeters')
60     set(gcf, 'PaperSize', [25 25])
61     set(gcf, 'Units', 'centimeters' )
62     set(gcf, 'Position', [0 0 25 25])
63     set(gcf, 'PaperPosition', [0 0 25 25])
64     end
65     %—
66     %—compute error
67     err(j) = max(abs( f(xx) - fa ) );
68 end
69 figure(3)
70 print( '-dpng', 'p2b_vary_n', '-r200' )
71 %plot error
72 figure(300)
73 semilogy( nvect, err, 'kx', 'markersize', 8, 'linewidth', 2 )
74 %make plot pretty
75 title( 'Maximum error' , 'interpreter', 'latex', 'fontsize', 16)
76 xlabel( '$n$', 'interpreter', 'latex', 'fontsize', 16)
77 ylabel( '$\max|f(x) - f_a(x)|$', 'interpreter', 'latex', 'fontsize', 16)
78 set(gca, 'TickLabelInterpreter','latex', 'fontsize', 16 )

```

```
78 set(gcf, 'PaperPositionMode', 'manual')
79 set(gcf, 'Color', [1 1 1])
80 set(gca, 'Color', [1 1 1])
81 set(gcf, 'PaperUnits', 'centimeters')
82 set(gcf, 'PaperSize', [15 15])
83 set(gcf, 'Units', 'centimeters' )
84 set(gcf, 'Position', [0 0 15 15])
85 set(gcf, 'PaperPosition', [0 0 15 15])
86 svnm = 'p2b_error';
87 print( '-dpng', svnm, '-r200' )
```