AE 370: HW2

Linyi Hou

February 12, 2020

Problem 1

1.a

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

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

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

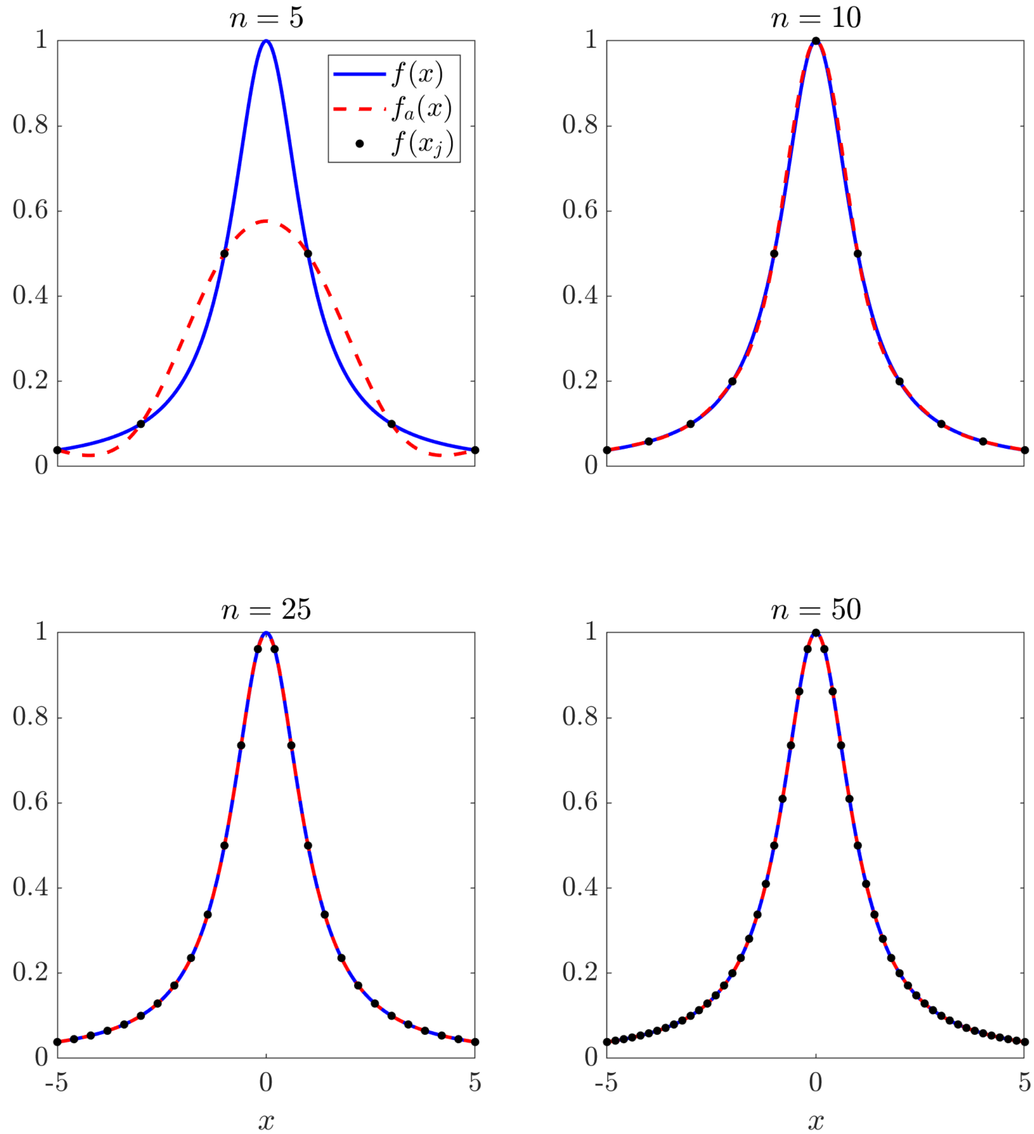

Figure 1: Problem 1.b Function Plots

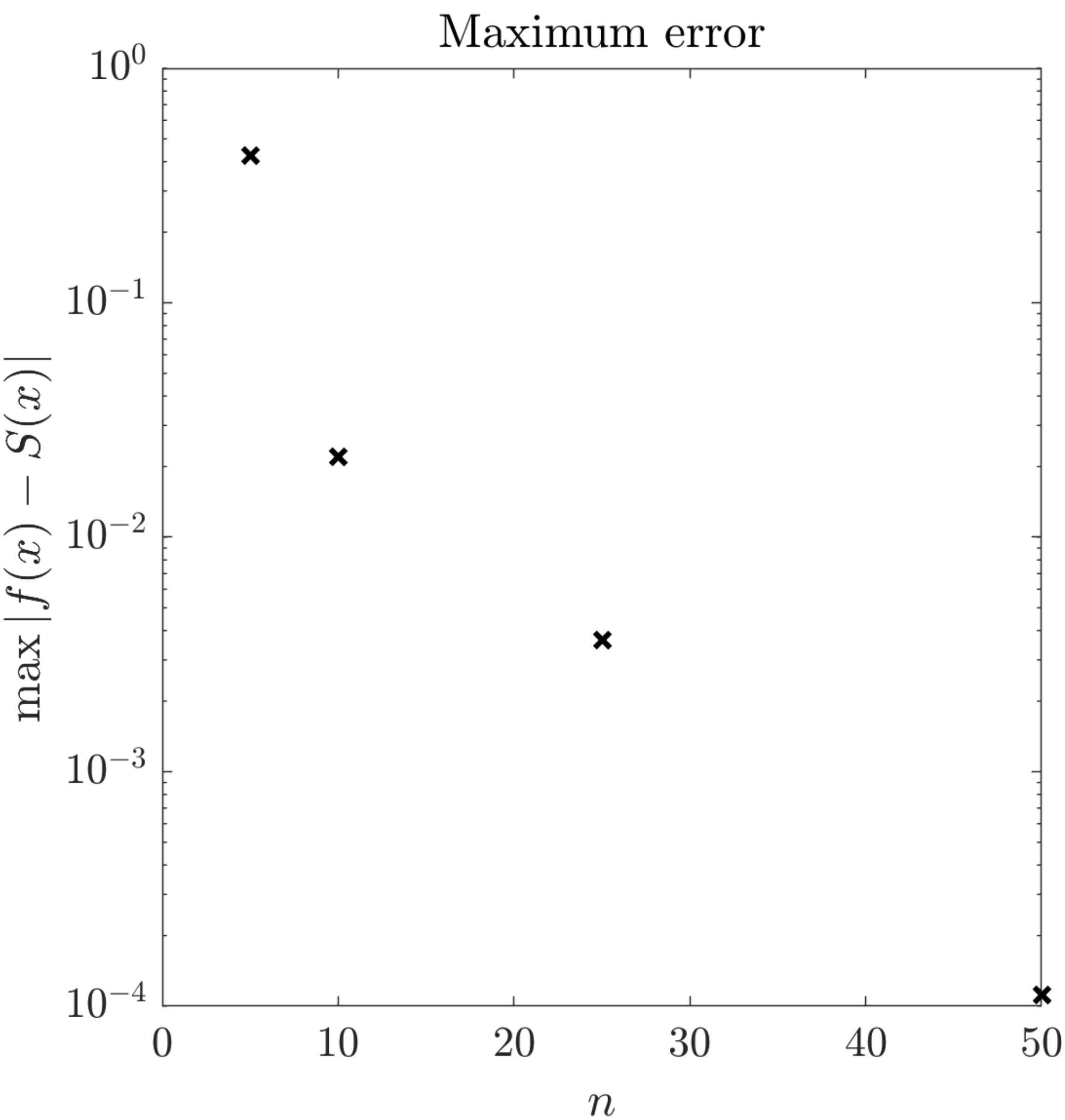

Figure 2: Problem 1.b Error Plots

Appendix A: Plots for 2.a

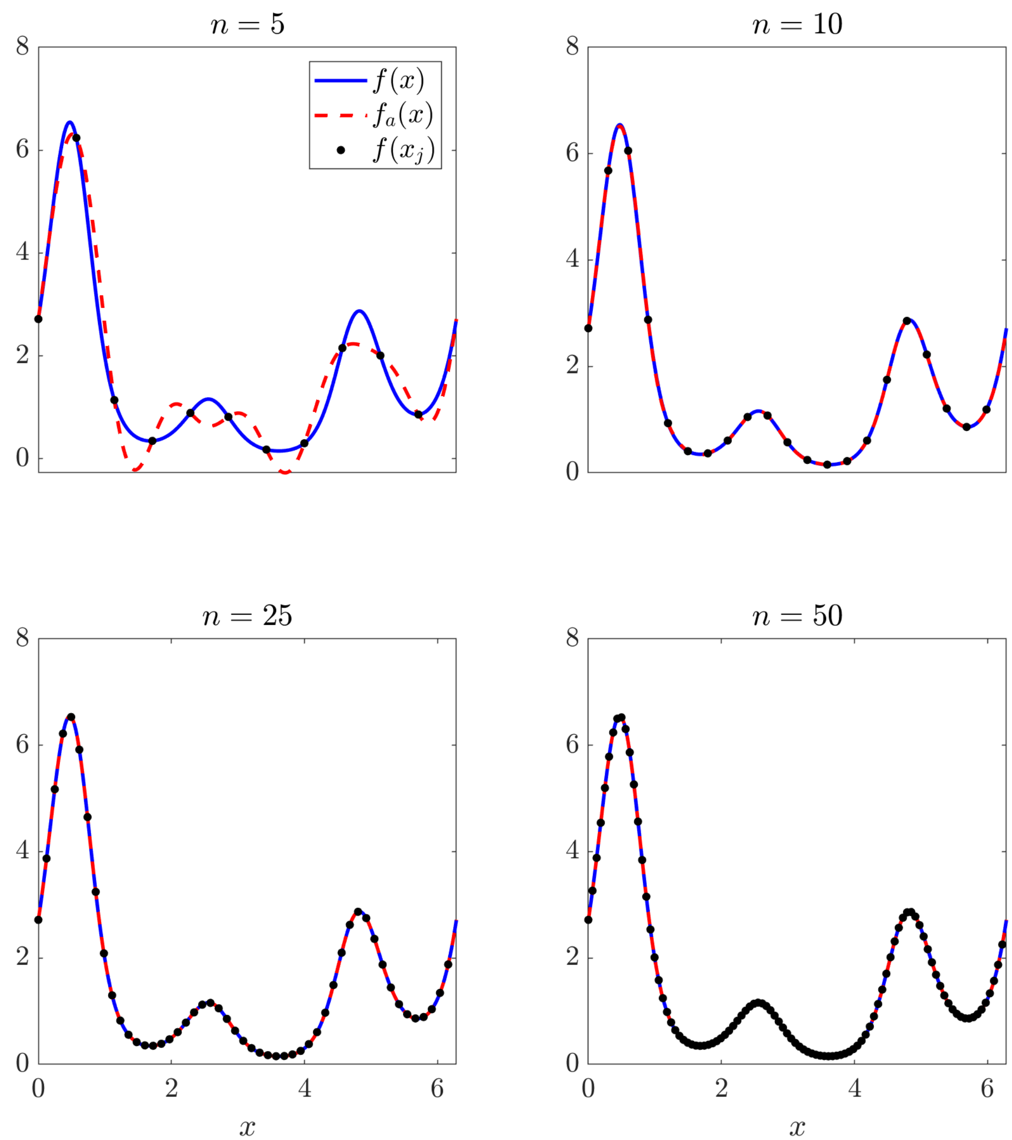

Figure 3: Problem 2.a Function Plots

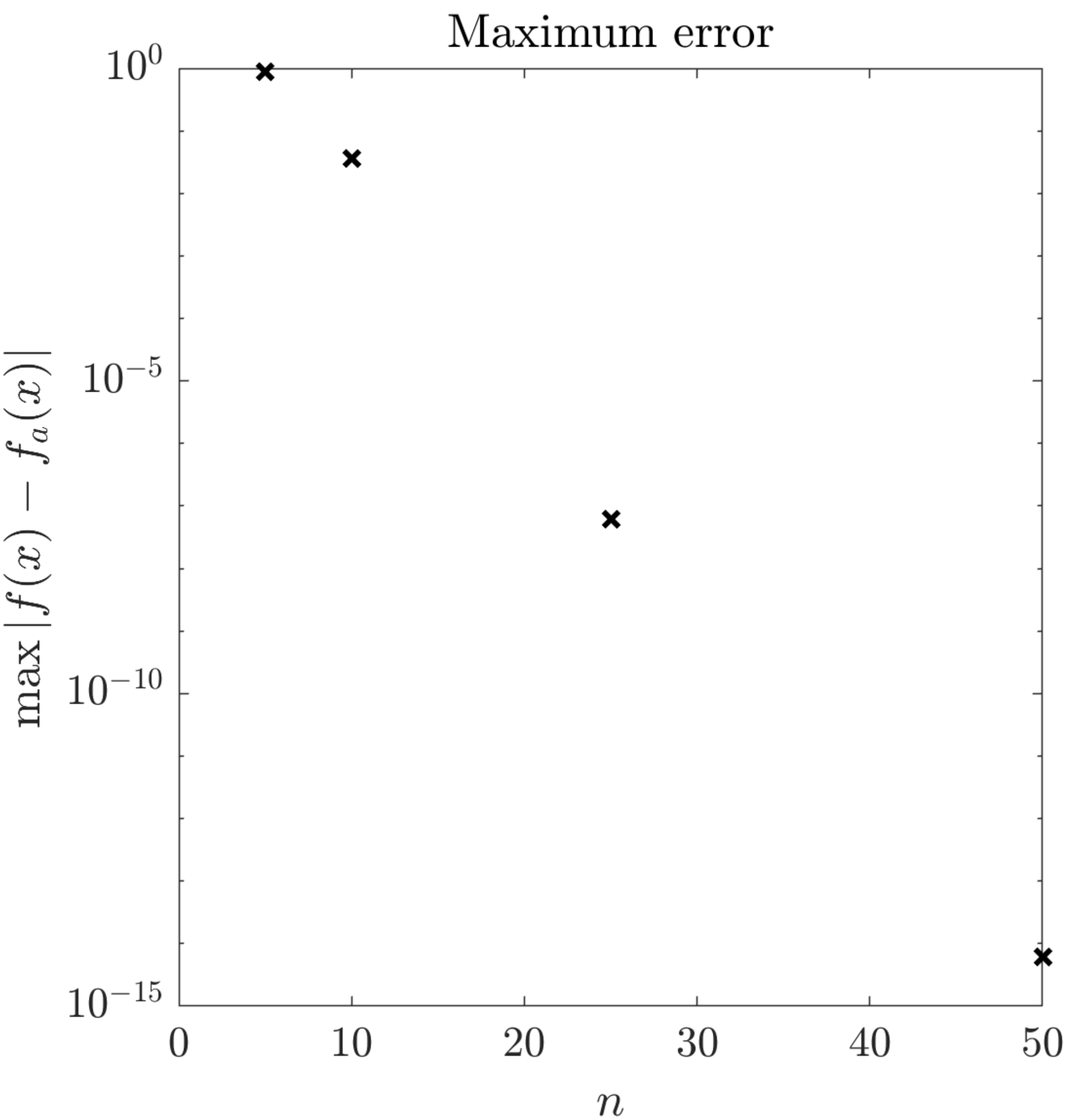

Figure 4: Problem 2.a Error Plots

Appendix A: Plots for 2.b

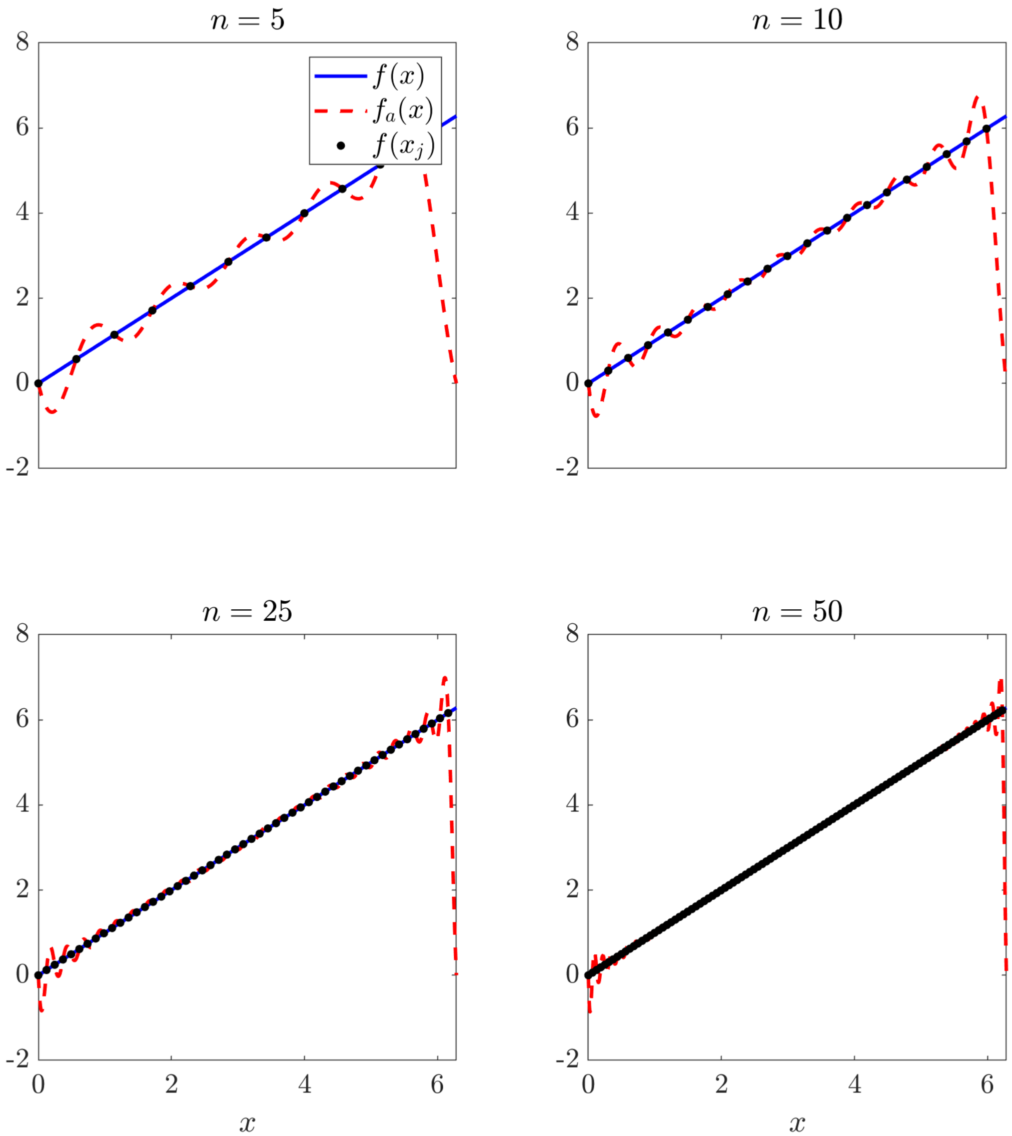

Figure 5: Problem 2.b Function Plots

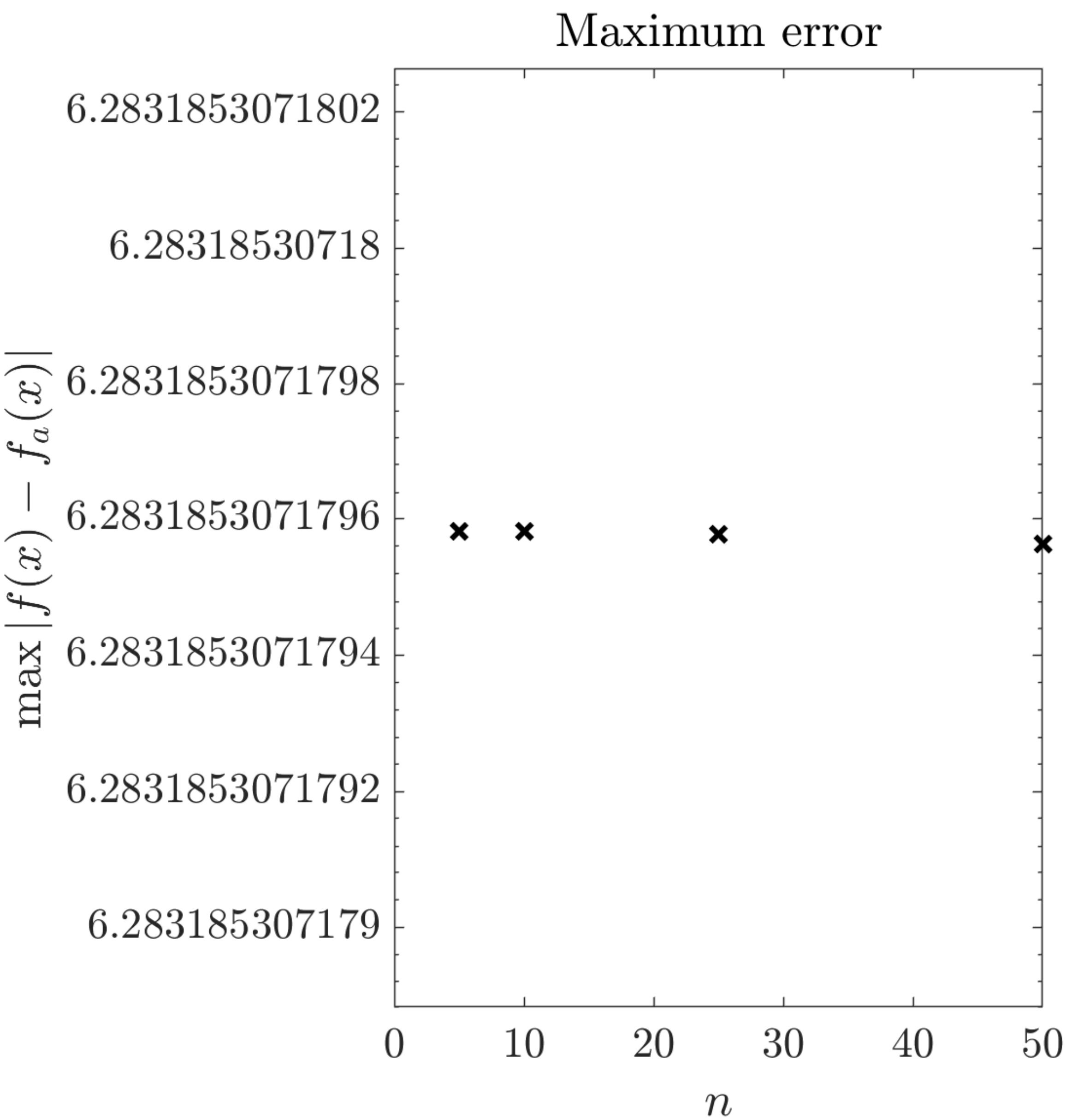

Figure 6: Problem 2.b Error Plots

Appendix B: Code

Appendix B: Code for 1.b

```matlab
%% Problem 1
clear;clc

nvect = [5, 10, 25, 50];
xl = -5; xr = 5;
%function to approx
f = @(x) 1./(1+x.^2) ;
%error vector:
err = zeros(size(nvect));
for j = 1 : length( nvect )
    %define current n
    n = nvect( j );
    %define interp points (equally spaced)
    xj = (xl : (xr-xl)/n : xr)';
    dx = xj(2)-xj(1);
    %--build & solve lin system for the c_{i,k} (i = 1,...,n; k = 1,...,4)
    %--for natural splines
    A = zeros( 4*n ); %initialize matrix
    g = zeros( 4*n, 1 ); %initialize RHS vector
    %Build A matrix & f vector
    for jj = 1 : n
        ind = 4*(jj - 1) + 1;
        %condition (1) from partial solution doc
        A( ind, ind ) = dx^2 / 6;
        A( ind, ind + 1 ) = 0;
        A( ind, ind + 2 ) = xj(jj);
        A( ind, ind + 3 ) = 1;
        g( ind ) = f(xj(jj));
        %condition (2) from partial solution doc
        A( ind + 1, ind ) = 0;
        A( ind + 1, ind + 1 ) = dx^2 / 6;
        A( ind + 1, ind + 2 ) = xj(jj + 1);
        A( ind + 1, ind + 3 ) = 1;
        g( ind + 1 ) = f(xj(jj + 1));
        %derivative conditions
```

```
        %(careful here! index on derivs only goes to n-1...)
        if jj ~= n
            %condition (3) from partial solution doc
            A( ind+2, ind ) = 0;
            A( ind+2, ind + 1 ) = dx/2;
            A( ind+2, ind + 2 ) = 1;
            A( ind+2, ind + 3 ) = 0;
            A( ind+2, ind + 4 ) = dx/2;
            A( ind+2, ind + 5 ) = 0;
            A( ind+2, ind + 6 ) = -1;
            A( ind+2, ind + 7 ) = 0;
            %condition (4) from partial solution doc
            A( ind+3, ind ) = 0;
            A( ind+3, ind + 1 ) = 1;
            A( ind+3, ind + 2 ) = 0;
            A( ind+3, ind + 3 ) = 0;
            A( ind+3, ind + 4 ) = -1;
            A( ind+3, ind + 5 ) = 0;
            A( ind+3, ind + 6 ) = 0;
            A( ind+3, ind + 7 ) = 0;
        else
            %s_1''(x_1) = 0 (eqn (11))
            A( ind+2, 1 ) = 1;
            A( ind+2, 2 ) = 0;
            A( ind+2, 3 ) = 0;
            A( ind+2, 4 ) = 0;
            %s_n''(x_n+1) = 0 (eqn (12))
            A( ind+3, ind ) = 0;
            A( ind+3, ind + 1 ) = 1;
            A( ind+3, ind + 2 ) = 0;
            A( ind+3, ind + 3 ) = 0;
        end
        g( ind + 2 ) = 0;
        g( ind + 3 ) = 0;
    end
    %solve for coeffs:
    c = A \ g;
    %--
    %--plot the spline interpolant S(x)
    xx = linspace( xl, xr, 1000 );
```

```
    S = zeros( size( xx ) );
    for jj = 1 : n
        ind = 4*(jj - 1) + 1;
        indxx = ( xx >= xj( jj ) & xx <= xj( jj + 1 ) );
        xxc = xx( indxx ~= 0 );
        S( indxx ~= 0 ) = c( ind ) * ( xxc - xj(jj+1) ).^3./ ...
            ( 6.*(xj(jj) - xj(jj+1)) ) + c( ind+1 ) * ( xxc - ...
            xj(jj) ).^3./( 6.*(xj(jj+1) - xj(jj)) ) + c( ind + 2 ) ...
            .* xxc + c( ind + 3 );
    end
    if j <= 4
        figure(1)
        subplot(2,2,j)
        plot( xx, f(xx), 'b-', 'linewidth', 2 ), hold on
        plot( xx, S, 'r--', 'linewidth', 2 )
        plot( xj, f(xj), 'k.', 'markersize', 16 )
        %make plot pretty
        title( ['$n = ', num2str( n ),'$'] ,'interpreter', 'latex',...
            'fontsize', 16)
        if j == 1
            h = legend( '$f(x)$', '$f_a(x)$', '$f(x_j)$');
        end
        if j <= 2
            set( gca, 'XTick', [] )
        else
            xlabel( '$x$', 'interpreter', 'latex', 'fontsize', 16)
        end
        set(h, 'location', 'NorthEast', 'Interpreter', 'Latex', 'fontsize', 16 )
        set(gca, 'TickLabelInterpreter','latex', 'fontsize', 16 )
        set(gcf, 'PaperPositionMode', 'manual')
        set(gcf, 'Color', [1 1 1])
        set(gca, 'Color', [1 1 1])
        set(gcf, 'PaperUnits', 'centimeters')
        set(gcf, 'PaperSize', [25 25])
        set(gcf, 'Units', 'centimeters' )
        set(gcf, 'Position', [0 0 25 25])
        set(gcf, 'PaperPosition', [0 0 25 25])
    end
    %--
```

```
    %--compute error
    err(j) = max(abs( f(xx) - S ) );
end
figure(1)
print( '-dpng', 'p1_vary_n', '-r200' )
%plot error
figure(100)
semilogy( nvect, err, 'kx', 'markersize', 8, 'linewidth', 2 )
%make plot pretty
title( 'Maximum error' ,'interpreter', 'latex','fontsize', 16)
xlabel( '$n$', 'interpreter', 'latex', 'fontsize', 16)
ylabel( '$\max|f(x) - S(x)|$', 'interpreter', 'latex', 'fontsize', 16)
set(gca, 'TickLabelInterpreter','latex', 'fontsize', 16 )
set(gcf, 'PaperPositionMode', 'manual')
set(gcf, 'Color', [1 1 1])
set(gca, 'Color', [1 1 1])
set(gcf, 'PaperUnits', 'centimeters')
set(gcf, 'PaperSize', [15 15])
set(gcf, 'Units', 'centimeters' )
set(gcf, 'Position', [0 0 15 15])
set(gcf, 'PaperPosition', [0 0 15 15])
svnm = 'p1_error';
print( '-dpng', svnm, '-r200' )
```

Appendix B: Code for 2.a

```
%% Problem 2a
close all
clear;clc

nvect = [5, 10, 25, 50];
xl = 0; xr = 2*pi;
%function to approx
f = @(x) exp(cos(x)+sin(3*x));
%error vector:
err = zeros(size(nvect));
for j = 1 : length( nvect )
    %define current n
    n = nvect( j );
    %define interp points (equally spaced)
    xj = (xl : (xr-xl)/(2*n+1) : xr)'; % 2n+2 terms
    % don't get repeated points at 2*pi
    xj = xj(1:end-1); % 2n+1 terms
    g = zeros( 2*n+1, 1 ); %initialize RHS vector
    %bulid f vector and use fft() to find c vector
    for jj = 1:2*n+1
        g(jj) = f(xj(jj));
    end
    %solve for coeffs:
    c = fft(g) / (2*n+1);
    %reshape c to sort coefficients
    c = [c(n+2:end);c(1:n+1)];
%      c = real(c);
    %--
    %--plot the periodic approximation
    xx = linspace( xl, xr, 1000 );
    fa = zeros(size(xx));
    for jj = -n:n
        fa = fa + c(jj+n+1).*exp(1i*jj.*xx);
    end
    fa = real(fa);
    if j <= 4
        figure(2)
        subplot(2,2,j)
```

```
        plot( xx, f(xx), 'b-', 'linewidth', 2 ), hold on
        plot( xx, fa, 'r--', 'linewidth', 2 )
        plot( xj, f(xj), 'k.', 'markersize', 16 )
        %make plot pretty
        title( ['$n = ', num2str( n ),'$'] ,'interpreter', 'latex',...
            'fontsize', 16)
        if j == 1
            h = legend( '$f(x)$', '$f_a(x)$', '$f(x_j)$');
        end
        if j <= 2
            set( gca, 'XTick', [] )
        else
            xlabel( '$x$', 'interpreter', 'latex', 'fontsize', 16)
        end
        set(h, 'location', 'NorthEast', 'Interpreter', 'Latex', 'fontsize', 16 )
        set(gca, 'TickLabelInterpreter','latex', 'fontsize', 16 )
        set(gcf, 'PaperPositionMode', 'manual')
        set(gcf, 'Color', [1 1 1])
        set(gca, 'Color', [1 1 1])
        set(gcf, 'PaperUnits', 'centimeters')
        set(gcf, 'PaperSize', [25 25])
        set(gcf, 'Units', 'centimeters' )
        set(gcf, 'Position', [0 0 25 25])
        set(gcf, 'PaperPosition', [0 0 25 25])
    end
    %--
    %--compute error
    err(j) = max(abs( f(xx) - fa ) );
end
figure(2)
print( '-dpng', 'p2a_vary_n', '-r200' )
%plot error
figure(200)
semilogy( nvect, err, 'kx', 'markersize', 8, 'linewidth', 2 )
%make plot pretty
title( 'Maximum error' ,'interpreter', 'latex','fontsize', 16)
xlabel( '$n$', 'interpreter', 'latex', 'fontsize', 16)
ylabel( '$\max|f(x) - f_a(x)|$', 'interpreter', 'latex', 'fontsize', 16)
set(gca, 'TickLabelInterpreter','latex', 'fontsize', 16 )
```

```
set(gcf, 'PaperPositionMode', 'manual')
set(gcf, 'Color', [1 1 1])
set(gca, 'Color', [1 1 1])
set(gcf, 'PaperUnits', 'centimeters')
set(gcf, 'PaperSize', [15 15])
set(gcf, 'Units', 'centimeters' )
set(gcf, 'Position', [0 0 15 15])
set(gcf, 'PaperPosition', [0 0 15 15])
svnm = 'p2a_error';
print( '-dpng', svnm, '-r200' )
```

Appendix B: Code for 2.b

```
%% Problem 2b
close all
clear;clc

nvect = [5, 10, 25, 50];
xl = 0; xr = 2*pi;
%function to approx
f = @(x) x;
%error vector:
err = zeros(size(nvect));
for j = 1 : length( nvect )
    %define current n
    n = nvect( j );
    %define interp points (equally spaced)
    xj = (xl : (xr-xl)/(2*n+1) : xr)'; % 2n+2 terms
    % don't get repeated points at 2*pi
    xj = xj(1:end-1); % 2n+1 terms
    g = zeros( 2*n+1, 1 ); %initialize RHS vector
    %bulid f vector and use fft() to find c vector
    for jj = 1:2*n+1
        g(jj) = f(xj(jj));
    end
    %solve for coeffs:
    c = fft(g) / (2*n+1);
    %reshape c to sort coefficients
    c = [c(n+2:end);c(1:n+1)];
%     c = real(c);
    %--
    %--plot the periodic approximation
    xx = linspace( xl, xr, 1000 );
    fa = zeros(size(xx));
    for jj = -n:n
        fa = fa + c(jj+n+1).*exp(1i*jj.*xx);
    end
    fa = real(fa);
    if j <= 4
        figure(3)
        subplot(2,2,j)
```

```
        plot( xx, f(xx), 'b-', 'linewidth', 2 ), hold on
        plot( xx, fa, 'r--', 'linewidth', 2 )
        plot( xj, f(xj), 'k.', 'markersize', 16 )
        %make plot pretty
        title( ['$n = ', num2str( n ),'$'] ,'interpreter', 'latex',...
            'fontsize', 16)
        if j == 1
            h = legend( '$f(x)$', '$f_a(x)$', '$f(x_j)$');
        end
        if j <= 2
            set( gca, 'XTick', [] )
        else
            xlabel( '$x$', 'interpreter', 'latex', 'fontsize', 16)
        end
        set(h, 'location', 'NorthEast', 'Interpreter', 'Latex', 'fontsize', 16 )
        set(gca, 'TickLabelInterpreter','latex', 'fontsize', 16 )
        set(gcf, 'PaperPositionMode', 'manual')
        set(gcf, 'Color', [1 1 1])
        set(gca, 'Color', [1 1 1])
        set(gcf, 'PaperUnits', 'centimeters')
        set(gcf, 'PaperSize', [25 25])
        set(gcf, 'Units', 'centimeters' )
        set(gcf, 'Position', [0 0 25 25])
        set(gcf, 'PaperPosition', [0 0 25 25])
    end
    %--
    %--compute error
    err(j) = max(abs( f(xx) - fa ) );
end
figure(3)
print( '-dpng', 'p2b_vary_n', '-r200' )
%plot error
figure(300)
semilogy( nvect, err, 'kx', 'markersize', 8, 'linewidth', 2 )
%make plot pretty
title( 'Maximum error' ,'interpreter', 'latex','fontsize', 16)
xlabel( '$n$', 'interpreter', 'latex', 'fontsize', 16)
ylabel( '$\max|f(x) - f_a(x)|$', 'interpreter', 'latex', 'fontsize', 16)
set(gca, 'TickLabelInterpreter','latex', 'fontsize', 16 )
```

```
set(gcf, 'PaperPositionMode', 'manual')
set(gcf, 'Color', [1 1 1])
set(gca, 'Color', [1 1 1])
set(gcf, 'PaperUnits', 'centimeters')
set(gcf, 'PaperSize', [15 15])
set(gcf, 'Units', 'centimeters' )
set(gcf, 'Position', [0 0 15 15])
set(gcf, 'PaperPosition', [0 0 15 15])
svnm = 'p2b_error';
print( '-dpng', svnm, '-r200' )
```