Problem 1. Let \( f(x) = e^{\sin(x)} \). Suppose we want to construct a table over one period of this function, \([0, 2\pi]\), with step size \( h \). We want to ensure that \(|E(t)| \leq 10^{-8}\). The second and sixth derivatives of \( f \) were found using MATLAB but can be computed by iterating the chain rule and the product rule for derivatives.

(a) Suppose first that we want to so with linear interpolation. So we will utilize the error formula with \( n = 1 \). For some \( \xi \in [0, 2\pi] \) determined by \( t \), we can compute

\[
|E(t)| = \left| \frac{(t - x_0)(t - x_1)}{2!} f''(\xi) \right| \\
= \left| \frac{(t - x_0)(t - x_1)}{2!} \right| \left| \cos^2(\xi)e^{\sin(\xi)} - \sin(\xi)e^{\sin(\xi)} \right|.
\]

Using MATLAB, we conclude that \( |f''(x)| \) has a maximum of approximately 2.7183 on the interval \([0, 2\pi]\) which will be obtained in at least one interval \([x_i, x_{i+1}]\). For details, see the attached sheet of MATLAB computations.

Further, per Atkinson page 136,

\[
\max_{x_0 \leq t \leq x_1} \{(t - x_0)(t - x_1)\} = \frac{h^2}{4}.
\]

So, for our desired result, we must have that

\[
10^{-8} \geq |E(t)| \geq \frac{h^2 \cdot 2.7183}{8}.
\]

Solving for \( h \), we must require our step size satisfies

\[
h \leq \sqrt{\frac{8 \cdot 10^{-8}}{2.7183}} \approx 0.00017,
\]

to obtain the desired bound on the error of our linear interpolation.
(b) Suppose now that we want to so with 5th degree polynomial interpolation. So we will utilize the error formula with \( n = 5 \). For some \( \xi \in [0, 2\pi] \) determined by \( t \), we can compute

\[
|E(t)| = \left| \frac{(t-x_0)(t-x_1)(t-x_2)(t-x_3)(t-x_4)(t-x_5) f^{(6)}(\xi)}{6!} \right|
\]

where \( f^{(6)}(\xi) \) will not be explicitly stated but was computed via MATLAB and is included in the attached MATLAB computations. Using MATLAB, we conclude that \( |f^{(6)}(x)| \) has a maximum of approximately 84.2667 on the interval \([0, 2\pi]\) which will be obtained in at least one interval \([x_i, x_{i+1}]\). For details, see the attached sheet of MATLAB computations.

Following Atkinson's process for maximizing \(|(t-x_0)(t-x_1)|\), we will bound

\[
|(t-x_0)(t-x_1)(t-x_2)(t-x_3)(t-x_4)(t-x_5)|.
\]

In order to maximize this polynomial, assume \( t \in [x_0, x_1] \) since increasing the distance of \( t \) to the rest of the nodes as much as possible will maximize the value of the specified polynomial. Then as before, we can conclude

\[
\max_{x_0 \leq t \leq x_1} \{(t-x_0)(t-x_1)\} = \frac{h^2}{4}.
\]

We can then see that \( x \) cannot be further from \( x_2 \) that \( 2h \) so \(|x - x_2| \leq 2h\). Continuing this process, we can conclude that \( x \) cannot be further from \( x_i \) that \( ih \) so \(|x - x_i| \leq ih\). Therefore,

\[
|(t-x_0)(t-x_1)(t-x_2)(t-x_3)(t-x_4)(t-x_5)| \leq \frac{h^2}{4} \prod_{i=2}^{5} \frac{ih}{h^6} = \frac{h^6 5!}{4} = 30h^6.
\]

Hence, our error formula becomes

\[
10^{-8} = |E(t)| = \left| \frac{(t-x_0)(t-x_1)(t-x_2)(t-x_3)(t-x_4)(t-x_5) f^{(6)}(\xi)}{6!} \right| \leq \frac{30h^6}{6!} (84.2667).
\]

Rearranging this formula to find \( h \), we must require our step size satisfies

\[
h \leq \sqrt[6]{\frac{10^{-8} 6!}{30 \cdot 84.2667}} = 0.03764.
\]
Problem 2. Let \( \Psi(x) = \prod_{i=0}^{n} (x - x_i) \) and \( w_j = \frac{1}{\Psi(x_j)} \). Consider the function

\[
\ell_i(x) = \frac{\prod_{k=0, k \neq i}^{n} (x - x_k)}{\prod_{k=0, k \neq i}^{n} (x_i - x_k)}
\]

\[
= \frac{1}{\prod_{k=0, k \neq i}^{n} (x_i - x_k)} \cdot \prod_{k=0}^{n} (x - x_k)
\]

\[
= w_i \frac{\prod_{k=0}^{n} (x - x_k)}{x - x_i}
\]

Using \( n \)th degree polynomial interpolation, we can express the constant function \( f(x) = 1 \) as

\[
1 = \sum_{i=0}^{n} 1 \cdot \ell_i(x) = \sum_{i=0}^{n} w_i \frac{\prod_{k=0}^{n} (x - x_k)}{x - x_i} = \prod_{k=0}^{n} (x - x_k) \left( \sum_{i=0}^{n} \frac{w_i}{x - x_i} \right)
\]

Now, let \( p_n(x) \) be the \( n \)th degree polynomial interpolant for some arbitrary set of data points. Then we have

\[
p_n(x) = \sum_{i=0}^{n} f(x_i) \ell_i(x)
\]

\[
= \sum_{i=0}^{n} f(x_i) w_i \frac{\prod_{k=0}^{n} (x - x_k)}{x - x_i}
\]

\[
= \prod_{k=0}^{n} (x - x_k) \sum_{i=0}^{n} \frac{f(x_i) w_i}{x - x_i}
\]

\[
= \frac{1}{\prod_{k=0}^{n} (x - x_k)} \left( \sum_{i=0}^{n} \frac{w_i}{x - x_i} \right)
\]

which gives us the barycentric form of Lagrange’s interpolation polynomial.

Problem 3. See the attached MATLAB code for all parts for this problem.
Prior to annotation, I did obtain the expected result per the attached figure. We have \( n+1 \) data points, \( n \) is the degree of the polynomial. We will use 41 data points to obtain a degree 40 polynomial.

\[
\begin{align*}
n &= 40; \\
x &= \text{linspace}(-1,1,n+1); \\
y &= \frac{1}{1+16\times x^2}; \quad \text{polyfit fits a polynomial of degree 40 to the function we defined in } y \\
p &= \text{polyfit}(x,y,n); \\
x_{i} &= \text{linspace}(-1,1,201); \quad \text{xi plots a smoother graph using 201 equispaced points on the interval -1 to 1} \\
z &= \text{polyval}(p,x_{i}); \quad \text{z evaluates the polynomial fitted to the data from above at each of the 201 points equispaced between -1 and 1} \\
\end{align*}
\]

The following line plots the points used for interpolation as circles and the polynomial determined by those points.

\[
\begin{align*}
\text{plot}(x,y,'-o',x_{i},z) \quad \text{plot(x,y,'-o',xi,z)}; \\
\text{This sets up the axis for viewing the graph as [minimum x, maximum x, minimum y, maximum y]} \quad \text{axis}([-1,1,-4,4]); \\
\text{The following line adds a title to the figure.} \quad \text{title('Runge Phenomenon')};
\end{align*}
\]
% Homework 1, Problem 3, Part C
%
% We have n+1 data points, n is the degree of the polynomial. We will use
% 41 data points to obtain a degree 40 polynomial.
% n = 40;
% linspace equally spreads 41 points between 0 and 40, so we get the integers between 0
% and 40, inclusive
x = linspace(0,n,n+1);
% This gives us the 41 input values which are not equispaced but are spaced
% to reduce the Runge Phenomenon
x_n = -cos(x*pi/n);
% y defines the function we wish to approximate with a polynomial
y = 1./(1+16*x_n.^2);
% polyfit fits a polynomial of degree 40 to the function we defined in y
% with the points 41 non-equispaced points between -1 and 1
p = polyfit(x_n,y,n);
% xi plots a smoother graph using 201 equispaced points on the interval -1
% to 1
x_i = linspace(-1,1,201);
% z evaluates the polynomial fitted to the data from above at each of the
% 201 points equispaced between -1 and 1
z = polyval(p,x_i);
% The following line plots the points used for interpolation as circles and
% the polynomial determined by those points
plot(x_n,y,'o',x_i,z);
% This sets up the axis for viewing the graph as [minimum x, maximum x,
% minimum y, maximum y]
axis([-1,1,-4,4]);
% The following line adds a title to the figure.
title('Interpolation using Chebyshev nodes');
Problem 4. Let $\omega(x) = \prod_{i=0}^{n} (x - x_i)$ over the interval $[-1, 1]$ and consider

$$\ln |\omega(x)| = \ln \left| \prod_{i=0}^{n} (x - x_i) \right| = \sum_{i=0}^{n} \ln |x - x_i| = \frac{n}{2} \sum_{i=0}^{n} \frac{2}{n} \ln |x - x_i|.$$  

The sum is strongly reminiscent of a numerical integral across the interval $[-1, 1]$ with step size $\frac{2}{n}$. Hence, larger $n$ give a closer approximation to

$$\int_{-1}^{1} \ln |x - x_i| \, dx.$$  

With a careful application of integration including a change of variable for $u = x - x_i$, we can compute

$$\int_{-1}^{1} \ln |x - x_i| \, dx = \int_{-1}^{x} \ln |x - x_i| \, dx + \int_{x}^{1} \ln |x - x_i| \, dx$$

$$= - \int_{x+1}^{0} \ln |u| \, du - \int_{0}^{x-1} \ln |u| \, du$$

$$= \int_{0}^{x+1} \ln |u| \, du + \int_{x-1}^{0} \ln |u| \, du$$

$$= \int_{0}^{x+1} \ln(u) \, du + \int_{0}^{1-x} \ln(u) \, du$$

$$= u \ln(u) \Bigg|_{0}^{x+1} + u \ln(u) \Bigg|_{0}^{1-x}$$

$$= (x + 1) \ln(x + 1) + (1 - x) \ln(1 - x) - 2.$$  

Exponentiating, we obtain

$$|\omega(x)| \approx e^{\frac{n}{2} ((x+1) \ln(x+1) + (1-x) \ln(1-x) - 2) = e^{\frac{1}{2} ((x+1) \ln(x+1) + (1-x) \ln(1-x) - 1)^{n}},}$$

as desired.