24/30

Cristian B. Jetomo

MATH 174 – B2L

Exercise 2.1 - L^∞ Norm-optimized Polynomial Interpolation

For this exercise, we wish to interpolate the function $Y(x) = \arctan(x^3) + e^x$ using concepts of Chebyshev and Algebraic Polynomial Interpolation. Further, we aim to assess the accuracy of the interpolating polynomials and do some error analysis.

can be used interchangeably in our context

1. First, we find the algebraic interpolating polynomial (AIP) P_8 of degree $n = 8$ over the interval $[-1, 1]$ which has the minimum error with respect to the L^∞ norm. By Theorem 3.1, we can attain this by using the roots of the monic Chebyshev polynomial $\hat{T}_9$ of degree $n + 1 = 9$ as our interpolating abscissas for P_8.

   Now, given that $T_0 = 1$, $T_1 = x$, and the recursion function

   $$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

   we obtain using MATLAB that $T_9(x)$ is equal to

   ```
   256.0*x^9 - 576.0*x^7 + 432.0*x^5 - 120.0*x^3 + 9.0*x
   ```

   Note that the monic Chebyshev polynomial is obtained using the piecewise function

   $$\hat{T}_{n+1} = \begin{cases} T_0(x), & n = 0 \\ 2^{1-(n+1)}T_{n+1}(x), & n \geq 1 \end{cases} = \begin{cases} T_0(x), & n = 0 \\ 2^{-n}T_{n+1}(x), & n \geq 1 \end{cases}$$

   Thus, using MATLAB, we computed the monic Chebyshev polynomial $\hat{T}_9$ to be equal to

   ```
   x^9 - 2.25*x^7 + 1.688*x^5 - 0.4688*x^3 + 0.03516*x
   ```

   By Remark 3.1, we can calculate the roots of $\hat{T}_9$ as follows

   $$x_i = \cos\left(\frac{2i-1}{2(9)} \cdot \pi\right), \text{ for } i = 1, 2, \ldots, 9$$

Calculating these using MATLAB, we obtain the roots to be

```
 0.984807753012208
 0.866025403784439
 0.642787609686539
 0.342020143325669
 0.000000000000000
-0.342020143325669
-0.642787609686539
-0.866025403784439
-0.984807753012208
```

Using these as our interpolatory abscissas for the AIP $P_8(x)$, we obtain that

(1.5pts)

```
P8(x) =
2.543e-5*x^8 - 0.2315*x^7 + 0.001388*x^6 + 0.0001674*x^5 + 0.04167*x^4 + 1.194*x^3 + 0.5*x^2 + 0.9973*x + 1.0
```

1 SF 2 SF

Secondly, we then find another AIP $Q_8(x)$ relative to the nine equally spaced points over $[-1, 1]$. Using MATLAB, we obtain that

(1.5pts)

```
Q8(x) =
2.532e-5*x^8 - 0.2849*x^7 + 0.001389*x^6 + 0.0864*x^5 + 0.04167*x^4 + 1.159*x^3 + 0.5*x^2 + 1.0*x + 1.0
```

1 SF

2. Now, we wish to determine the accuracy of the two generated AIPs by examining their plots in comparison with the plot of the function $Y(x) = \arctan(x^3) + e^x$.

Plotting $P_8(x)$ using MATLAB, we obtain the following figure:

(2pts)

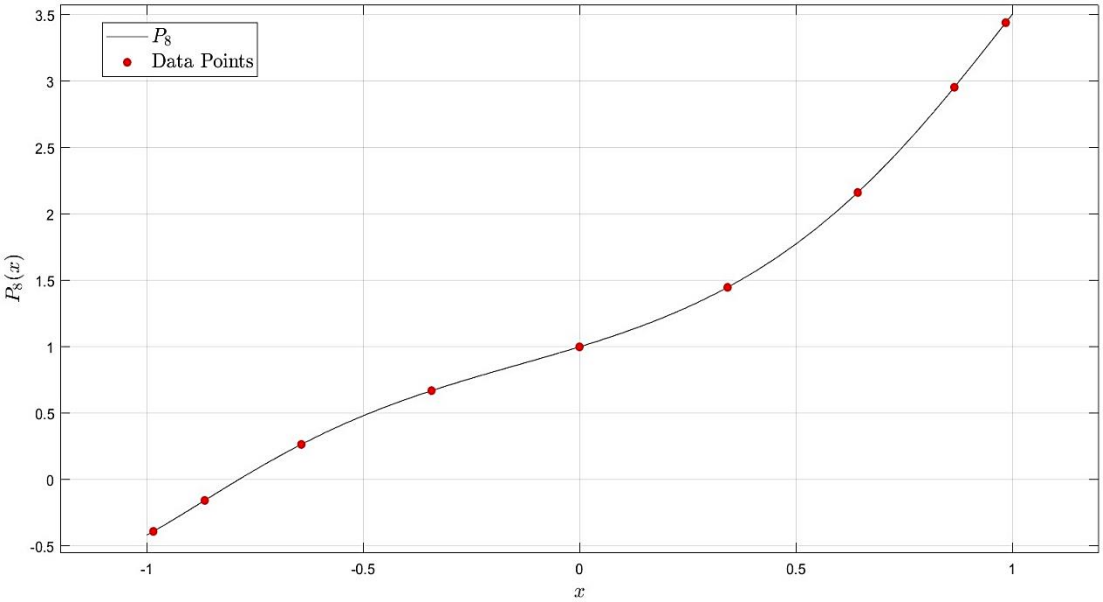


Figure 2.1.1. Graph of the interpolant $P_8(x)$ and a scatter plot of its interpolatory data points over the interval $[-1, 1]$.

Observe, that $P_8(x)$ is generated correctly as the graph of the function passed through each of the interpolatory abscissas relative to it.

Similarly, plotting $Q_8(x)$ using MATLAB, we obtain the following illustration:

(2pts)

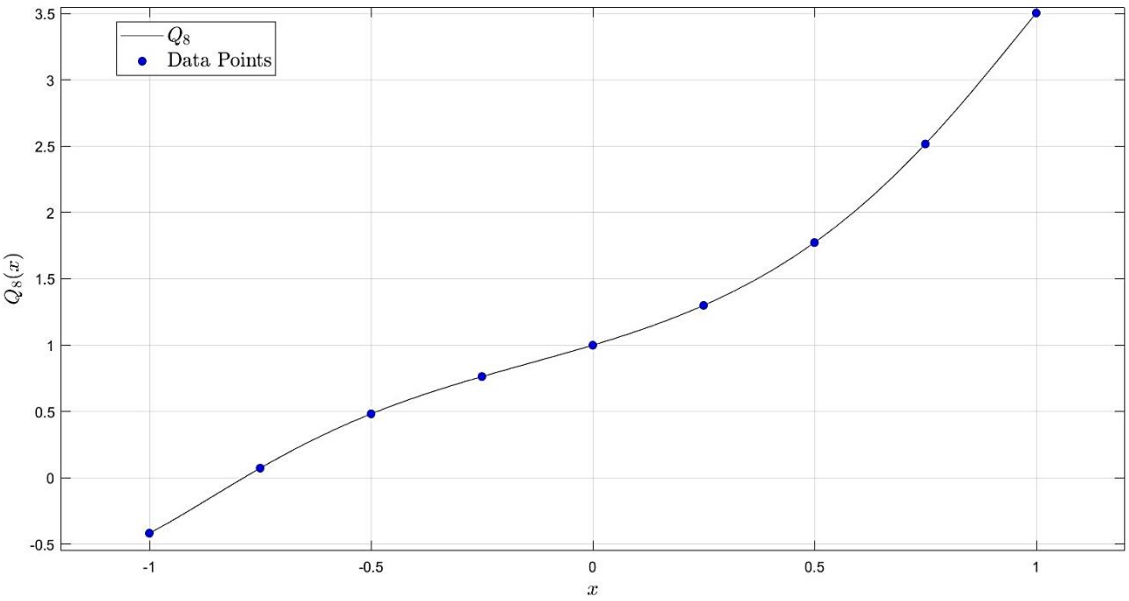
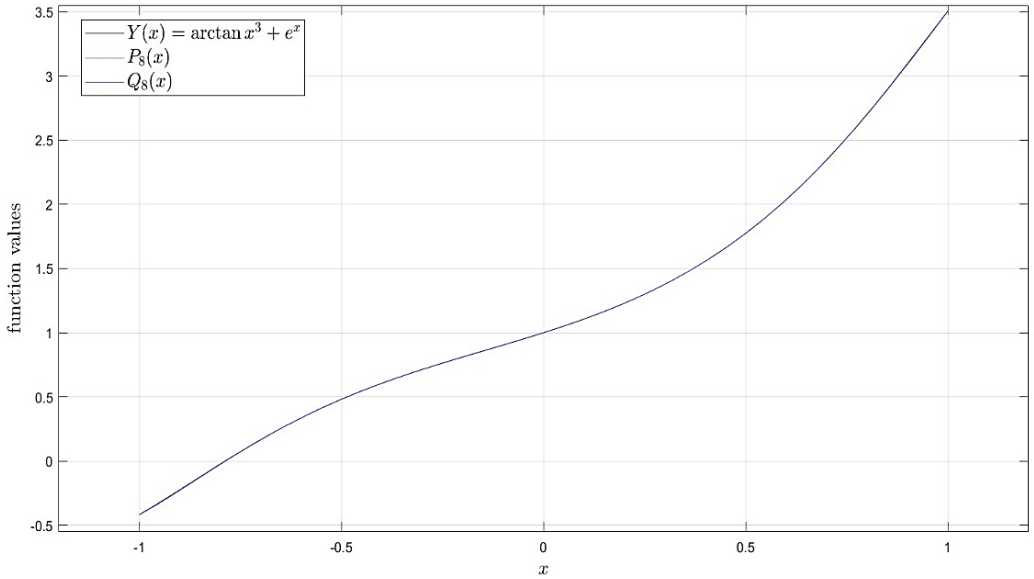


Figure 2.1.2. Graph of the interpolant $Q_8(x)$ and a scatter plot of its interpolatory data points over the interval $[-1, 1]$.

Likewise, the generated algebraic interpolating polynomial $Q_8(x)$ is correct as it passes through all its interpolatory abscissas.
To assess the accuracy of the two polynomials, we then generate the illustrated plot together with the graph of the function $Y(x)$. The figure for this is illustrated as follows:

(1pt)



should be on the same page as the figure being described.

Figure 2.1.3. Plots of the algebraic interpolatory polynomials P_8 and Q_8 and the function $Y(x) = \arctan(x^3) + e^x$ over the interval $[-1, 1]$.

We can observe from the figure that the plots for the three functions are significantly similar to each other that each of the plots cannot be distinguished anymore. From here, we cannot seem to assess the accuracy of our approximations from the two polynomials and consequently, failure to determine which is a better approximate.

Hence, we calculate the absolute error functions given below:

$$P_e(x) = |P_8(x) - Y(x)|$$
$$Q_e(x) = |Q_8(x) - Y(x)|$$

Using MATLAB, we then obtain the plots of these absolute error functions illustrated as follows:

(2pts)

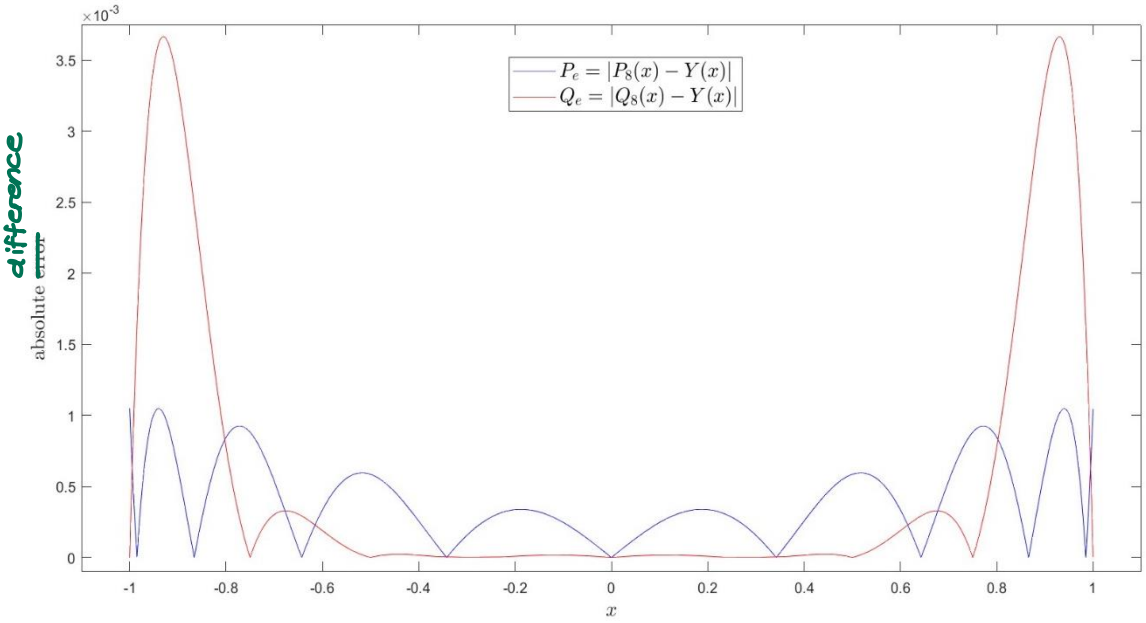


Figure 2.1.4. Graph of the absolute error functions $P_e = |P_8(x) - Y(x)|$ and $Q_e = |Q_8(x) - Y(x)|$ over the interval $[-1, 1]$.

(2pts)

Based on the figure, we can observe that the value of the absolute errors of the AIP P_8 over the interval $[-1, 1]$ is consistently within $[0 \times 10^{-3}, 1 \times 10^{-3}]$. On the other hand, the value of the absolute errors of Q_8 over the same interval is almost 3.5×10^{-3} near the endpoints and almost 0 near $x = 0$. We can infer from this that P_8 is a better approximation since, in general, it has smaller absolute errors than the errors relative to Q_8. This is aligned with the theory that minimizing the local error, in this case the L^∞ norm, by using the Chebyshev nodes as the interpolating

abscissas will allow us to construct a more accurate algebraic interpolating polynomial.

3. Now, we will perform some error analysis by examining the upper bound for the relative L^{∞} errors for the approximations from our computed AIPs P_8 and Q_8. Note that the relative errors can be computed as follows:

$$\frac{\left\| \frac{f^{(9)}(\xi)}{9!}\omega \right\|_{\infty}}{\|Y\|_{\infty}} \Longrightarrow \frac{\max_{x\in[-1,1]} \left| \frac{f^{(9)}(\xi)}{9!}\omega \right|}{\max_{x\in[-1,1]} |Y|}$$

Due to incapability of MATLAB to plot the required function, we used an online derivative calculator and Desmos. We obtain the graph of $f^{(9)}(x)$ to be

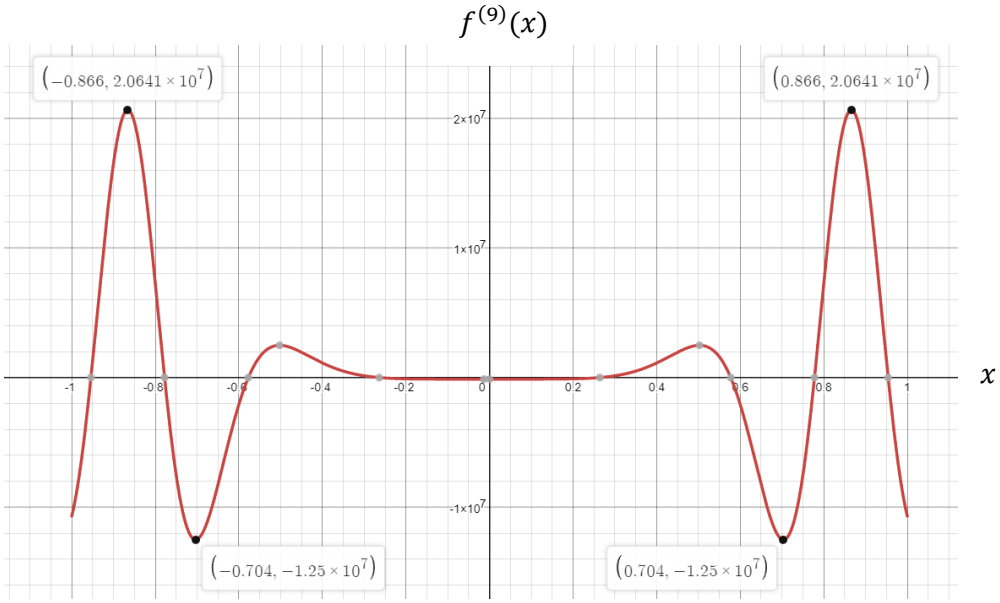


Figure 2.1.5. Graph of $f^{(9)}(x)$ where $f(x) = \arctan x^3 + e^x$ over the interval $[-1, 1]$.

(3pts) It can be observed that the absolute maximum of $f^{(9)}(x)$ occurs whenever $x = \pm 0.866$. Recall also from Figure 2.3.1 that the graph of $Y(x)$ is increasing over the interval $[-1,1]$. Hence,

$$\frac{\left\| \frac{f^{(9)}(\xi)}{9!}\omega \right\|_{\infty}}{\|Y\|_{\infty}} = \frac{\frac{f^{(9)}(0.866)}{9!} \max_{x\in[-1,1]} |\omega|}{Y(1)}$$

Now using the Extreme Value Theorem to obtain the maximum values of the $\omega(x)$ for the AIPs P_8 and Q_8, we calculated that

(1pt)
```
An upper bound for the relative L-infty error for P8 is
    0.0634
```

(1pt)
```
An upper bound for the relative L-infty error for Q8 is
    0.3053
```

As observed, the upper bound for the relative L^∞ error for the approximation using the AIP P_8 is smaller than that of the AIP Q_8. This justifies our hypothesis in item 2 that P_8 is a better approximation for our function $Y(x) = \arctan x^3 + e^x$.

4. Now, we compute a new AIP R_8 of degree 8 for the function $Y(x)$ over the interval $[0, 5]$, this time, relative to the scaled Chebyshev nodes which is given as follows:

$$x_i = \frac{b-a}{2}\cos\left(\frac{2i-1}{2(9)}\pi\right) + \frac{a+b}{2} \quad , \text{for } i = 1, 2, \ldots, 9$$

Using MATLAB, we calculated the polynomial R_8 relative to the scaled Chebyshev nodes to be

(2pts)
```
R_8(x) =
- 0.001032*x^8 + 0.02388*x^7 - 0.1972*x^6 + 0.8061*x^5 - 1.427*x^4 + 0.7043*x^3 + 1.995*x^2 + 0.5527*x + 1.015
```

The plot of this polynomial is also illustrated below.

(2pts)

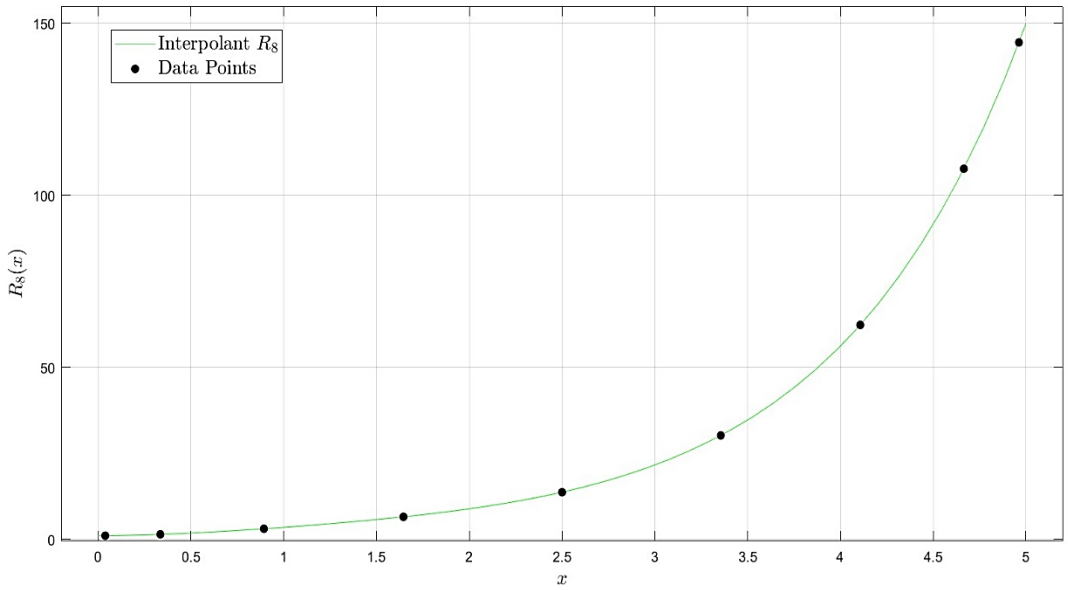


Figure 2.1.6. Graph of the interpolant $R_8(x)$ and a scatter plot of the scaled Chebyshev nodes over the interval $[0, 5]$.

Clearly from the figure, it can be observed that the obtained scaled Chebyshev nodes are in fact within the interval $[0, 5]$ and that the generated polynomial R_8 accurately interpolates the stated data points.

Now, we wish to compute the $||\omega||_\infty$ for R_8. As defined, we know that $\omega(x)$ is a polynomial. Hence, we can apply Extreme Value Theorem on it over the interval $[0, 5]$. Using this concept, we were then able to determine that the $||\omega||_\infty = \max_{x \in [0,5]} |\omega|$, or the

(2pts)

```
L-infty norm of omega for R8 is
   14.9012
```

Using the same process, we were also able to determine that the

```
L-infty norm of omega for P8 is
    0.0039
```

We observe from here that the L^∞ norm changes when the Chebyshev nodes are translated. Thus, it can be inferred that using Chebyshev polynomials is not entirely an efficient way if we want to approximate a function over the entire interval $[-\infty, \infty]$. This is because the local error, i.e., the value of L^∞ norm in this case, changes as we alter our interval. This means that the error over an arbitrary interval $[a, b] \in \mathbb{R}$ is not constant, thus resulting with varying errors over different intervals.

(1pt)

Appendix A
Program Used in MATLAB for Item 1

```
clear
close all

syms x
%%  Frequently used variables
Xq = linspace(-1, 1, 9);
Y = atan(x.^3) + exp(x);
numPts = length(Xq);
interval = [-1 1];

%% Calculation of AIP Q
Yq = atan(Xq.^3) + exp(Xq);
Q = 0;
for k = 1:numPts
    lk = 1;
    for i = 1:numPts
        if i ~= k
            lk = lk * ((x-Xq(i))/(Xq(k)-Xq(i)));
        end
    end
    Q = Q + Yq(k) * lk;
end
Q = simplify(Q);

%% Calculation of AIP P

% obtaining roots of T^hat_9
for j = 1:numPts
    root = cos((2*j-1)/(2*numPts) * pi);
    Xp(j) = root;
end
disp(Xp');

% final calculation for P
Yp = atan(Xp.^3) + exp(Xp);
P = 0;
for k = 1:numPts
    lk = 1;
    for i = 1:numPts
        if i ~= k
            lk = lk * ((x-Xp(i))/(Xp(k)-Xp(i)));
        end
    end
    P = P + Yp(k) * lk;
end
P = simplify(P);

%% Displaying polynomials
disp('P8(x) = ')
disp(vpa(expand(P), 4));
disp('Q8(x) = ')
disp(vpa(expand(Q), 4));
```

Appendix B
Program Used in MATLAB for Item 2

```
%% Plotting P8 and data points
figure
fplot(P, interval, 'k')
hold on
scatter(Xp, Yp, 'filled', 'r')
xlim([-1.2 1.2])
ylim([-0.55 3.575])
grid
lgnd = legend('$P_8$', 'Data Points');
xlp = xlabel('$x$');
ylp = ylabel('Interpolant $P_8(x)$');
for label = [lgnd xlp ylp]
    set(label, 'Interpreter', 'Latex', 'Fontsize', 14)
end
hold off

%% Plotting Q8 and data points
figure
fplot(Q, interval, 'k')
hold on
scatter(Xq, Yq, 'filled', 'b')
xlim([-1.2 1.2])
ylim([-0.55 3.55])
grid
lgnd = legend('Interpolant $Q_8$', 'Data Points');
xlp = xlabel('$x$');
ylp = ylabel('$Q_8(x)$');
for label = [lgnd xlp ylp]
    set(label, 'Interpreter', 'Latex', 'Fontsize', 14)
end

%% Plot of P8, Q8 and Y
figure
fplot(Y, interval, 'k')
hold on
fplot(P, interval, 'r')
hold on
fplot(Q, interval, 'b')
xlim([-1.2 1.2])
ylim([-0.55 3.55])
grid

legend('$Y(x) = \arctan{x^3} + e^x$', '$P_8(x)$', '$Q_8(x)$', ...
    'Interpreter', 'Latex', 'Fontsize', 14)
xlabel('$x$', 'Interpreter', 'Latex', 'Fontsize', 14)
ylabel('function values','Interpreter', 'Latex', 'Fontsize', 14)

%% Plotting Pe and Qe
Pe = abs(P - Y);
Qe = abs(Q - Y);

figure
```

```
fplot(Pe, interval, 'b')
hold on
fplot(Qe, interval, 'r')
grid
xlim([-1.1 1.1])
ylim([-0.1 3.75]*10^(-3))
grid

lgnd = legend('$P_e = |P_8(x) - Y(x)|$', '$Q_e = |Q_8(x) - Y(x)|$');
xlp = xlabel('$x$');
ylp = ylabel('absolute error');
for label = [lgnd xlp ylp]
    set(label, 'Interpreter', 'Latex', 'Fontsize', 14)
end
```

Appendix C
Program Used in MATLAB for Item 3

```
%% Error Analysis
% due to the incapability of MATLAB to plot a directly differentiated
% f9Prime, the f9Prime is obtained using an online derivative calculator
% https://www.symbolab.com/solver/derivative-calculator/
% f9Prime is graphed using Desmos
% from then we obtained that f9Prime is maximum whenever x=+-0.866

f9Prime = diff(Y, 9);
maxf9 = abs(double(subs(f9Prime, 0.866)));
maxY = double(subs(Y, 1));

% relative L-infty error for P8
omegaP = 1;
for i = 1:numPts
    omegaP = omegaP * (x - Xp(i));
end
omegaPprime = diff(omegaP);
CN = roots(sym2poly(omegaPprime));
CN = CN(CN >= min(interval) & CN <= max(interval));
CN = [CN' interval];
Pv = abs(double(subs(omegaP, CN)));
maxPv = max(Pv);

relP = (maxf9 * maxPv) / (factorial(9) * maxY);
disp('An upper bound for the relative L-infty error for P8 is')
disp(relP)

% relative L-infty error for Q8
omegaQ = 1;
for i = 1:numPts
    omegaQ = omegaQ * (x - Xq(i));
end
omegaQprime = diff(omegaQ);
CN = roots(sym2poly(omegaQprime));
CN = CN(CN >= min(interval) & CN <= max(interval));
CN = [CN' interval];
Qv = abs(double(subs(omegaQ, CN)));
maxQv = max(Qv);

relQ = (maxf9 * maxQv) / (factorial(9) * maxY);
disp('An upper bound for the relative L-infty error for Q8 is')
disp(relQ)
```

Appendix D
Program Used in MATLAB for Item 4

```
%% Calculation of AIP R8

% obtaining translated roots/nodes  [x_i = c*root + d]
intR = [0 5];  % interval for R
for j = 1:numPts
    c = (max(intR) - min(intR)) / 2;
    d = (min(intR) + max(intR)) /2 ;
    root = c * (cos((2*j-1)/(2*numPts) * pi)) + d;
    Xr(j) = root;
end

% final calculation for R
Yr = atan(Xr.^3) + exp(Xr);
R = 0;
for k = 1:numPts
    lk = 1;
    for i = 1:numPts
        if i ~= k
            lk = lk * ((x-Xr(i))/(Xr(k)-Xr(i)));
        end
    end
    R = R + Yr(k) * lk;
end
R = simplify(R);
disp('R_8(x) = ')
disp(vpa(expand(R), 4));

%% Plot for R8
figure
fplot(R, intR, 'g')
hold on
scatter(Xr, Yr, 'filled', 'k')
xlim([-0.2 5.2])
ylim([-0.5 155])
grid
lgnd = legend('Interpolant $R_8$', 'Data Points');
xlp = xlabel('$x$');
ylp = ylabel('$R_8(x)$');
for label = [lgnd xlp ylp]
    set(label, 'Interpreter', 'Latex', 'Fontsize', 14)
end

%% Calculation of L-infty norm of omega for R8
omega = 1;
for i = 1:numPts
    omega = omega * (x - Xr(i));
end

omegaPrime = diff(omega);
CN = roots(sym2poly(omegaPrime));
CN = CN(CN >= min(intR) & CN <= max(intR));
CN = [CN' intR];
```

```
Rvalues = abs(double(subs(omega, CN)));
maxRvalue = max(Rvalues);
disp('L-infty norm of omega for R8 is')
disp(maxRvalue)

%% Calculation of L-infty norm of omega for P8
omega = 1;
for i = 1:numPts
    omega = omega * (x - Xp(i));
end

omegaPrime = diff(omega);
CN = roots(sym2poly(omegaPrime));
CN = CN(CN >= min(interval) & CN <= max(interval));
CN = [CN' interval];
Pvalues = abs(double(subs(omega, CN)));
maxPvalue = max(Pvalues);
disp('L-infty norm of omega for P8 is')
disp(maxPvalue)
```