$\frac{29}{30}$

Cristian B. Jetomo

MATH 174 – B2L

Exercise 2.2 - L^2 Norm-optimized Polynomial Interpolation

For this exercise, we ~~now~~ wish to interpolate the function $Y(x) = \arctan(x^3) + e^x$ using the concepts of Legendre and Chebyshev applied to the concept of algebraic interpolating polynomials. Further, more ? we aim to assess the accuracy of the interpolating polynomials and some error analysis, comparing the different methods or approach in performing the interpolation.

1. First, we find the algebraic interpolating polynomial (AIP) S_8 of degree 8 over the interval $[-1, 1]$ which has the minimum error with respect to the L^2 norm. As stated ~~from~~ in Theorem 3.3, this can be done by using the roots of the Legendre polynomial of degree 9 which we will denote as L_9.

   Given that $L_0 = 1, L_1 = x,$ and the recursion formula

   $$L_n(x) = \frac{2n-1}{n} x P_{n-1}(x) - \frac{n-1}{n} P_{n-2}(x)$$

   we obtain using MATLAB that $L_9(x) =$

   ```
   94.9609375*x^9 - 201.09375*x^7 + 140.765625*x^5 - 36.09375*x^3 + 2.4609375*x
   ```

   Consequently, we obtain the roots of the monic Legendre polynomial $\hat{L}_9$, which are the same as the roots of L_9, given by

   ```
                   0
   -0.968160239507625
   -0.836031107326637
   -0.613371432700588
   -0.324253423403809
    0.968160239507623
    0.836031107326639
    0.613371432700591
    0.324253423403809
   ```

   Using these as our interpolatory abscissas for the AIP S_8, we obtain that

   (2.5pts)

   ```
   S8(x) =
   2.539e-5*x^8 - 0.2603*x^7 + 0.001388*x^6 + 0.04307*x^5 + 0.04167*x^4 + 1.176*x^3 + 0.5*x^2 + 0.9989*x + 1.0
   ```

   1SF 2SF

2. Now, we wish to assess the accuracy of our obtained AIP S_8. We do this by examining the graph of S_8 together with the function $Y(x) = \arctan(x^3) + e^x$ and the scatter plot of our interpolatory points relative to S_8.

   Plotting ~~this~~ these, we obtain the figure below:

(2pts)

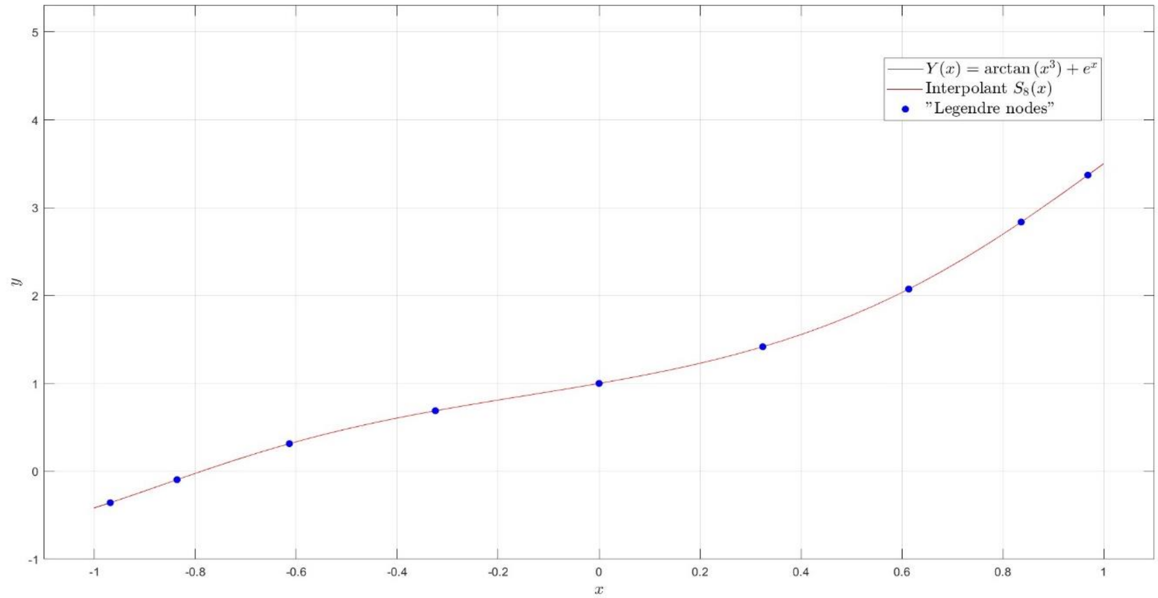

Figure 2.2.1 Graph of the interpolant $S_8(x)$, the function $Y(x) = \arctan(x^3) + e^x$, and the Legendre nodes used for interpolating $S_8(x)$ over the interval $[-1,1]$.

From the figure, we observe that there are indeed nine (9) interpolatory nodes that we obtain by solving for the roots of the 9th degree Legendre polynomial. Furthermore, we used these nodes as the interpolatory abscissas for the interpolant S_8. Hence the function $S_8(x)$ passes through all nine points. Note however that the nodes are within the interval $[-1, 1]$ and none of the nodes reaches the endpoint of the interval. Also, we can observe that the interpolant S_8 relatively replicates the graph of the function we wish to interpolate.

Now, we plot the functions P_8, Q_8, S_8 which are the interpolants relative to the roots of the 9th degree Chebyshev polynomial, the nine equally spaced points over $[-1,1]$, and the roots of the 9th degree Legendre polynomial, respectively. We plot this together with the function $Y(x)$.

(2pts)

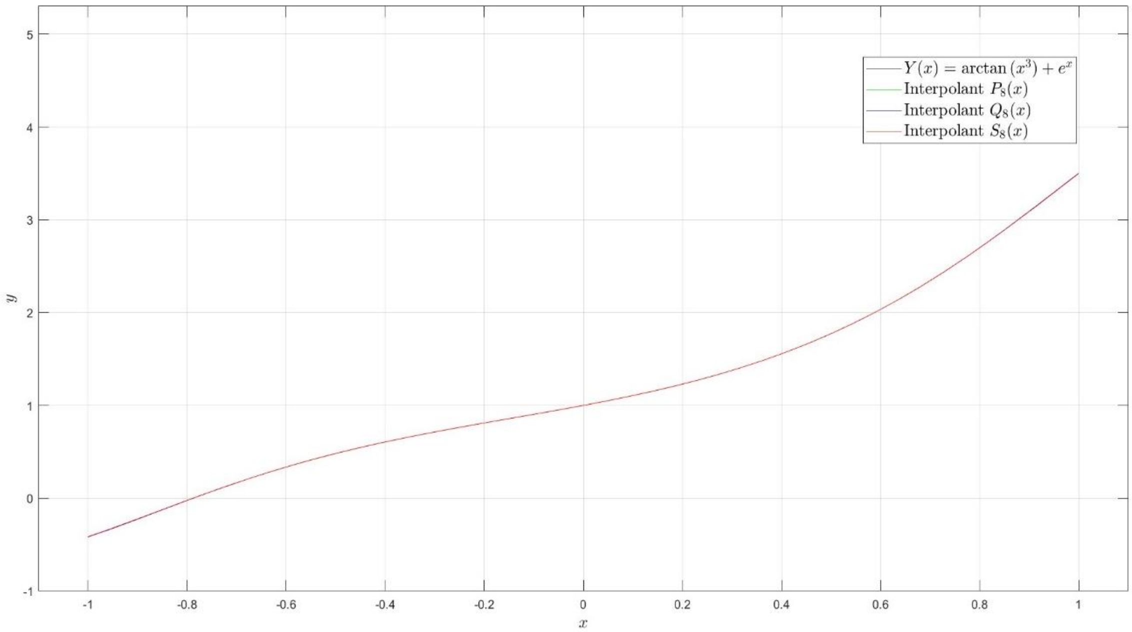

Figure 2.2.2 Graphs of the interpolants P_8, Q_8, S_8, and the function to interpolate $Y(x) = \arctan(x^3) + e^x$ over the interval $[-1,1]$.

We can observe from the figure that the plots for the three interpolants are significantly similar to graph of $Y(x)$ that each plot cannot be distinguished anymore. From here, we cannot seem to assess the accuracy of S_8 and consequently, we fail to determine which is a better approximate among the three interpolants. Hence, we calculate the absolute error functions given below and determine its plots.

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

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

$$S_e(x) = |S_8(x) - Y(x)|$$

The graphs for these functions are illustrated below.

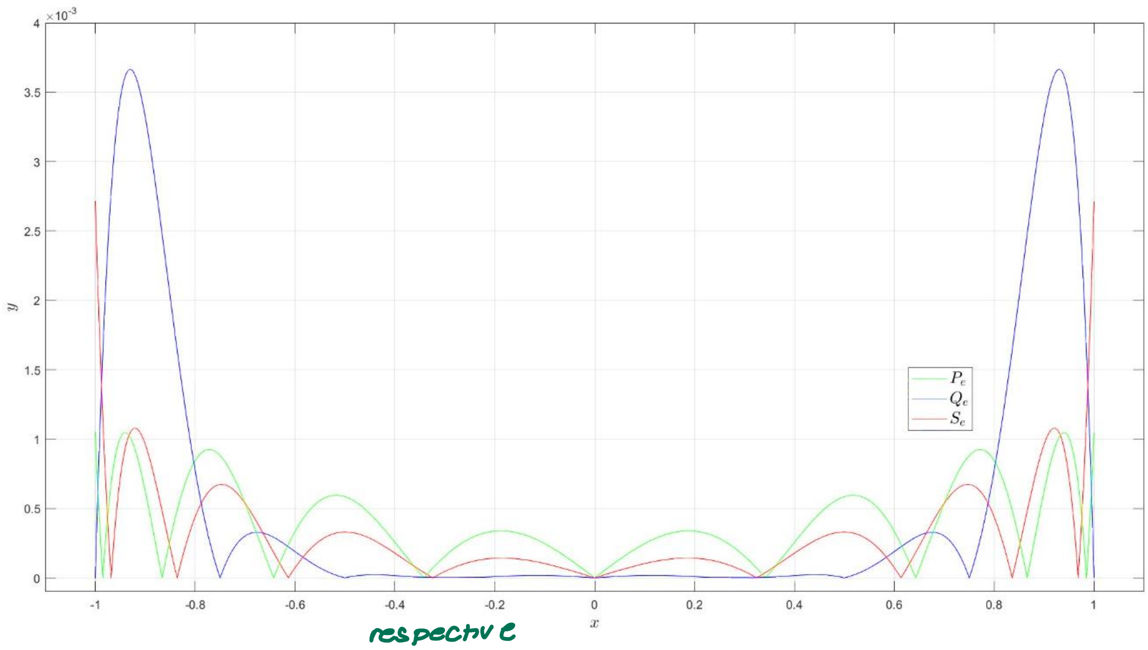

Figure 2.2.3 Graphs of the respective absolute errors P_e, Q_e, and S_e of the interpolants P_8, Q_8, and S_8, ~~respectively~~, relative to the function $Y(x)$ over the interval $[-1,1]$.

(2pts)

We can observe from the graphs that the errors are small, having values not greater than 4×10^{-3}. This is reflected to the similar plots for the interpolants P_8, Q_8, and S_8 which we obtain from Figure 2.2.2. Moreover, we can see from the graph that Q_e ~~have~~ has the highest values near the endpoints and smallest values near $x = 0$. On the other hand, P_e ~~have~~ has the smallest values at particular x-values and so is for S_e. We also see that these x values where P_e and S_e are smallest are relatively close to each other and that P_e and S_e values are higher at relatively similar x-intervals as well. We note however that P_e ~~have~~ has generally higher values at these intervals than S_e.

which values?

Unclear thought

which intervals?

These observations support the statement that L^∞ norm, corresponding to our interpolant P_8, ~~have~~ has a more local nature. That is, the absolute error P_e only have low values at certain x-values and at the other points, the errors are comparably higher. This pattern is still observable for S_e, but the difference in the lowest and highest value for S_e are smaller than that of P_e. This reflects that the L^2 norm, corresponding to our interpolant S_8, is more global in nature.

Use more concrete words to properly convey the thought.

Therefore, it can be deduced that among the three AIPs, we can infer that S_8 is a better approximation for our function $Y(x)$ as it has a more accurate result as seen by comparing the absolute errors of the interpolants.

Give proper context S_8 is better wrt L^2 norm.

3. To further justify or deny our claim, we perform some error analysis by computing for an upper bound for the relative L^2 error for our approximations from P_8, Q_8, and S_8. For all these approximations, we obtain the upper bounds as follows:

$$\frac{\left\|\frac{f^{(9)}(\xi)}{(9!)}\omega\right\|_2}{\|Y\|_2} \leq \frac{\max_{\xi\in[-1,1]}\left|\frac{f^{(9)}(\xi)}{(9!)}\right|\|\omega\|_2}{\|Y\|_2}$$

where $\omega = \prod_{i=1}^{9}(x - x_i)$ such that x_i, for $i = 1,2,\ldots,9$ are the interpolatory abscissas used to construct the interpolant which we want to find the relative error of.

Now, note that from our obtained results from Exercise 2.1, particularly from Figure 2.1.5, the absolute maximum value of $f^{(9)}$ occurs whenever $x = \pm 0.866$. Hence, the upper bound for the relative error can be obtained by

$$\frac{\left\|\frac{f^{(9)}(\xi)}{(9!)}\omega\right\|_2}{\|Y\|_2} \leq \frac{\left|\frac{f^{(9)}(0.866)}{(9!)}\right|\|\omega\|_2}{\|Y\|_2}$$

where, as defined,

$$\|\omega\|_2 = \sqrt{\int_{-1}^{1}\omega^2 dx}$$

and

$$\|Y\|_2 = \sqrt{\int_{-1}^{1}Y^2 dx}$$

~~And so~~, Using MATLAB, we obtain that the upper bound for the relative errors rel_p, rel_q and rel_s for the approximations from P_8, Q_8, and S_8, respectively, are the following:

(8pts)

```
relp =
0.09396

relq =
0.2184

rels =
0.0756
```

As observed, the S_8 ~~have~~ has the smallest relative error upper bound, justifying our hypothesis in Item 2. This is followed by the relative error upper bound for P_8, then that of Q_8.

4. Now, we need to find another AIP T_8 of degree 8 over the interval $[0,5]$ relative to the scaled Legendre nodes. The scaled nodes can be calculated using the formula below.

$$x_i = \frac{b-a}{2}r_i + \frac{a+b}{2}$$

where $a = 0, b = 5$, and r_i for $i = 1,2,\dots,9$ are the nine Legendre nodes on the interval $[-1,1]$ which we obtained from Item 1. The scaled nodes are listed below.

```
2.500000000000000
0.079599401230938
0.409922231683409
0.966571418248531
1.689366441490477
4.920400598769058
4.590077768316599
4.033428581751477
3.310633558509522
```

Using these as our interpolatory abscissas for the AIP T_8, we obtain that

(2.5pts)

```
T8(x) =
- 0.0004139*x^8 + 0.01006*x^7 - 0.06995*x^6 + 0.186*x^5 + 0.2849*x^4 - 1.945*x^3 + 4.107*x^2 - 0.1476*x + 1.07
```

3SF 3SF

Plotting the AIPs T_8, R_8 and the function $Y(x)$ where R_8 is the AIP relative to the scaled Chebyshev nodes, we obtain the illustration below.

(3pts)

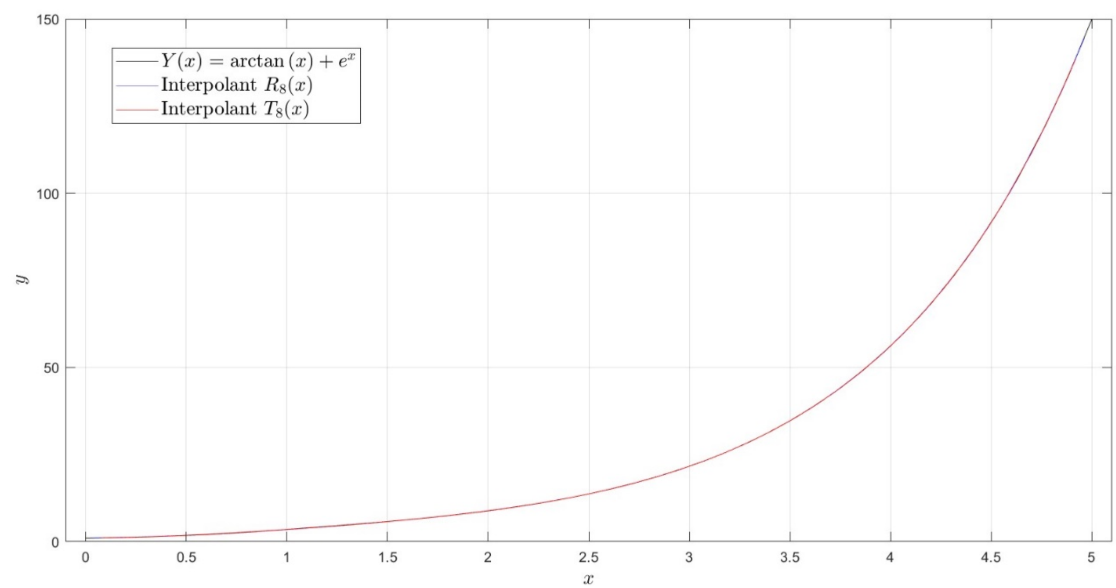

Figure 2.2.4 Graphs of the interpolants R_8 and T_8 and the function $Y(x)$ over the interval $[0,5]$.

Notice from the figure that the interpolatory intervals for R_8 and T_8 are not the same and did not reach the endpoints for the interval $[0,5]$. Notice also that the graphs of the three functions are overlapping, implying that our interpolatory intervals relatively approximates our function $Y(x)$. The errors however from these two graphs cannot be determined just by looking the figure.

However, in theory, we could infer that we should have a smaller absolute and relative error values for the AIP T_8 compared to that of R_8 since T_8 is constructed using scaled Legendre nodes which enabled us to minimize the global error of our approximation. Compared to R_8 which is constructed using Chebyshev nodes (3pts) whose goal is to minimize the local error of the approximation. Not~~able~~e however that both R_8 and T_8, in general, are better approximations to the function of interest compared to an AIP which uses nine equally spaced data points over the interval $[0,5]$ as its relative interpolatory abscissas.

APPENDIX A
Program Used in MATLAB for Item 1

(1pt)

```
lear
close all

syms x L  % Declares x and L as symbolic variables (L is a symbolic array)
%% Obtaining Chebyshev and Legendre Roots

% Chebyshev
ChebDeg = 9;  % Chebyshev Degree
Tp = zeros(1, ChebDeg);  % chebyshev nodes
for j = 1:ChebDeg
    root = cos((2*j-1)/(2*ChebDeg) * pi);
    Tp(j) = root;
end

% Legendre
LegDeg = 9;  % Legendre Degree
L0 = 1;  % L_0 = 1
L(1) = x;  % L_1 = x
L(2) = 3/2*x^2 - 1/2;
for n = 3:LegDeg
    left = ((2*n-1)/n)*x*L(n-1);
    right = ((n-1)/n)*L(n-2);
    L(n) = left - right;
end

L = L(LegDeg);
Ts = roots(sym2poly(L));  % Legendre nodes

numPts = length(Ts);  % counts number of interpolatory abscissas

%% Compute for AIP P Q and S

% AIP P
Kp = atan(Tp.^3) + exp(Tp);
P = 0;
for k = 1:numPts
    lk = 1;
    for i = 1:numPts
        if i ~= k
            numer = x - Tp(i);
            denom = Tp(k) - Tp(i);
            lk = lk * (numer/denom);
        end
    end
    P = P + Kp(k) * lk;
end

% AIP Q
Tq = linspace(-1, 1, 9);  % equally spaced points
Kq = atan(Tq.^3) + exp(Tq);
Q = 0;
for k = 1:numPts
```

```
    lk = 1;
    for i = 1:numPts
        if i ~= k
            numer = x - Tq(i);
            denom = Tq(k) - Tq(i);
            lk = lk * (numer/denom);
        end
    end
    Q = Q + Kq(k) * lk;
end

% AIP S
Ks = atan(Ts.^3) + exp(Ts);
S = 0;
for k = 1:numPts
    lk = 1;
    for i = 1:numPts
        if i ~= k
            numer = x - Ts(i);
            denom = Ts(k) - Ts(i);
            lk = lk * (numer/denom);
        end
    end
    S = S + Ks(k) * lk;
end

disp('S8(x) = ')
disp(vpa(expand(S), 4))
```

APPENDIX B
Program Used in MATLAB for Item 2

```matlab
%% Plotting
Y = atan(x^3) + exp(x);
interval = [-1, 1];

% S8, data points, and Y
figure
fplot(Y, interval, 'k')
hold on
fplot(S, interval, 'r')
hold on
scatter(Ts, Ks, 'filled', 'b')
xlim([-1.1 1.1])
ylim([-1 5.3])
grid
lgnd = legend( ...
    '$Y(x) = \arctan{(x^3)} + e^x$', ...
    'Interpolant $S_8(x)$', ...
    '"Legendre nodes"');
xlp = xlabel('$x$');
ylp = ylabel('$y$');
for label = [lgnd xlp ylp]
    set(label, 'Interpreter', 'Latex', 'Fontsize', 14)
end

% P8, Q8, S8, and K
figure
fplot(Y, interval, 'k')
hold on
fplot(P, interval, 'g')
hold on
fplot(Q, interval, 'b')
hold on
fplot(S, interval, 'r')
xlim([-1.1 1.1])
ylim([-1 5.3])
grid
lgnd = legend( ...
    '$Y(x) = \arctan{(x^3)} + e^x$', ...
    'Interpolant $P_8(x)$', ...
    'Interpolant $Q_8(x)$', ...
    'Interpolant $S_8(x)$');
xlp = xlabel('$x$');
ylp = ylabel('$y$');
for label = [lgnd xlp ylp]
    set(label, 'Interpreter', 'Latex', 'Fontsize', 14)
end

%% Plotting absolute errors
Pe = abs(P-Y);
Qe = abs(Q-Y);
Se = abs(S-Y);
```

```
figure
fplot(Pe, interval, 'g')
hold on
fplot(Qe, interval, 'b')
hold on
fplot(Se, interval, 'r')
xlim([-1.1 1.1])
ylim([-0.0001 .004])
grid
lgnd = legend('$P_e$', '$Q_e$', '$S_e$');
xlp = xlabel('$x$');
ylp = ylabel('$y$');
for label = [lgnd xlp ylp]
    set(label, 'Interpreter', 'Latex', 'Fontsize', 14)
end
```

APPENDIX C
Program Used in MATLAB for Item 3

```
%% Upper bound for Relative L2 Error

% L2 norm of omegas
omegaP = 1;
for i = 1:numPts
    omegaP = omegaP * (x - Tp(i));
end

omegaQ = 1;
for i = 1:numPts
    omegaQ = omegaQ * (x - Tq(i));
end

omegaS = 1;
for i = 1:numPts
    omegaS = omegaS * (x - Ts(i));
end

omegaPnorm = double(sqrt(int(omegaP^2, min(Tp), max(Tp))));
omegaQnorm = double(sqrt(int(omegaQ^2, min(Tq), max(Tq))));
omegaSnorm = double(sqrt(int(omegaS^2, min(Ts), max(Ts))));

% L2 Norm of Function Y(x) = atan(x^3)+exp(x)
y9Prime = diff(Y, 9);
maxy9Prime = double(abs(subs(y9Prime, 0.866)));

Ynorm = double(int(Y, -1, 1));

% Computation of Relative Errors
factor = maxy9Prime/abs(factorial(numPts));
relP = (factor * omegaPnorm) / Ynorm;
relQ = (factor * omegaQnorm) / Ynorm;
relS = (factor * omegaSnorm) / Ynorm;

disp('relp = ')
disp(vpa(relP, 4))
disp('relq = ')
disp(vpa(relQ, 4))
disp('rels = ')
disp(vpa(relS, 4))
```

APPENDIX D
Program Used in MATLAB for Item 4

```
%% Root Translation
TranslateRoots = 1;  % Indicator if I wish to scale and translate the abscissas
newInt = [0, 5];
a = newInt(1);
b = newInt(2);

if TranslateRoots == 1
    % scaled Chebyshev Nodes
    Tr = ((b-a)/2)*Tp + mean(newInt);  % mean(newInt) = (a+b)/2
    % scaled Legendre Nodes
    Tt = ((b-a)/2)*Ts + mean(newInt);
end

interpIntR = [min(Tr) max(Tr)];
interpIntT = [min(Tt) max(Tt)];

%% Computation of AIP R and T
% Translated Chebyshev
Kr = atan(Tr.^3) + exp(Tr);
R = 0;
for k = 1:numPts
    lk = 1;
    for i = 1:numPts
        if i ~= k
            numer = x - Tr(i);
            denom = Tr(k) - Tr(i);
            lk = lk * (numer/denom);
        end
    end
    R = R + Kr(k) * lk;
end

% Translated Legendre
Kt = atan(Tt.^3) + exp(Tt);
T = 0;
for k = 1:numPts
    lk = 1;
    for i = 1:numPts
        if i ~= k
            numer = x - Tt(i);
            denom = Tt(k) - Tt(i);
            lk = lk * (numer/denom);
        end
    end
    T = T + Kt(k) * lk;
end

disp('T8(x) = ')
disp(vpa(expand(T), 4))
%% Plotting AIP T, R, and K
figure
fplot(Y, newInt, 'k')
```

```
hold on
fplot(R, interpIntR, 'b')
hold on
fplot(T, interpIntT, 'r')
grid
xlim([-0.1 5.1])
ylim([0 150])
lgnd = legend('$Y(x) = \arctan{(x)} + e^x$', ...
    'Interpolant $R_8(x)$', ...
    'Interpolant $T_8(x)$');
xlp = xlabel('$x$');
ylp = ylabel('$y$');
for label = [lgnd xlp ylp]
    set(label, 'Interpreter', 'Latex', 'Fontsize', 14)
end
```