Problem 1 (20 points)

1. Write a MATLAB routine for computing the Newton polynomial interpolant of degree $n$ for a given set of datapoints $\{(x_j, f(x_j)), \ 0 \leq j \leq n\}$ where $f$ is a given function.

2. Let

$$p_n(x) = \sum_{i=0}^{n} a_n x^n.$$ 

Horner’s method is an algorithm for calculating polynomials, which consists of transforming the monomial form into a computationally efficient form. To evaluate $p_n(x_0)$, define a sequence of constants as follows:

$$b_n := a_n \quad \quad (1)$$

$$b_{n-1} := a_{n-1} + b_n x_0 \quad \quad (2)$$

... \quad \quad (3)

$$b_0 := a_0 + b_1 x_0. \quad \quad (4)$$

Then $b_0$ is the value of $p_n(x_0)$. Write another routine to evaluate the polynomial using Horner’s nested evaluation scheme. Your evaluation scheme should be written in vectorized form so it can accept a an arbitrary vector argument $x$ and produce a vector of function values $f$ with $f(i) = p_n(x(i))$.

3. Write a third MATLAB routine that will add a new data point $(x_{n+1}, f(x_{n+1}))$ to the existing set of points and update the Newton interpolant without recomputing anything done with the first set of data.

4. Apply your routines for computing and evaluating the Newton polynomial interpolant to the Gamma function (gamma in matlab) on the interval $[1, 4]$. First, construct the
polynomial interpolants at 5, 10, 15 equally spaced points in [.1, 4]. Include the end points (automatic with linspace). Plot the Gamma function on this interval and then plot the three polynomial interpolants you constructed on the same graph (using hold on). Your graph should include a legend identifying the gamma function and the three different interpolants. Be sure to use different line types for each of these. Use 1000 equally spaced points in the plot.

Second, compute the interpolant on the interval [.5, 3.5] using 10 equally spaced points in this interval. Plot the Gamma function and the interpolating polynomial on the same graph as above. Examine the result and judiciously add a few more interpolation points with \(x_j\) in the interval [.1, 4] to improve the quality of the interpolation approximation. When you have a decent fit, plot the improved interpolating polynomial on the same graph as the gamma function and first interpolating polynomial. This plot should be over the interval [.1, 4] using 1000 equally spaced points as before. Again a legend and distinct line types are required.

You should turn in the listing of your three codes together with the two plots.

**Problem 2 (30 points)** Let \(\{x_i = ih\}_{i=0}^n\) with \(h = 1/n\) be a uniform partition of the interval [0, 1]. Consider a function \(f \in C^2[0, 1]\), and its have the piecewise linear polynomial interpolation \(p(x)\). You are going to prove that the \(L^2\) error estimate for piecewise linear interpolation is

\[
\|f - p\|_{L^2([0,1])} = \sqrt{\int_0^1 [f(x) - p(x)]^2 dx} \leq h^2 \|f''\|_{L^2([0,1])}.
\]

For that, follow the steps below:

1. Show that the Taylor expansion of \(f(x)\) for \(x\) close to zero can be written as

\[
f(x) = f(0) + f'(0)x + \int_0^x f''(s)(x-s)ds.
\]

Hint: You should use the integration by part formula

\[
\int u(x)v'(x)dx = u(x)v(x) - \int u'(x)v(x)dx.
\]

2. Then show that, since \(p(x)\) is the linear interpolation of \(f \in C[0,h]\),

\[
p(x) = f(0) + f'(\alpha)x, \quad \alpha \in [0,h].
\]
3. Use results form points 1 and 2 to show that for \( x \in [0, h] \),

\[
|f(x) - p(x)| \leq \sqrt{3h} x \left( \int_0^h |f''(s)|^2 ds \right)^{1/2}
\]

Hint: notice that

\[ g(x) - g(0) = \int_0^x g'(s) ds. \]

Hint: use the following result (without proof)

\[
\int_0^h |f(x)| dx \leq \sqrt{h} \left( \int_0^h f^2(x) dx \right)^{1/2}.
\]

4. Proof now that

\[
\|f - p\|_{L^2([0,1])} \leq h^2 \|f''\|_{L^2([0,1])}.
\]

Problem 3 (50 points). This problem is pledged!

The Hermite interpolant \( h_n \in P_{2n+1} \) of \( f \in C^1[a,b] \) at the points \( \{x_j : j = 0, 1, 2, \ldots, n\} \) is the interpolating polynomial such that \( h_n(x_j) = f(x_j) \) and \( h'_n(x_j) = f'(x_j) \). It can be written in the form

\[
h_n(x) = \sum_{j=0}^n A_j(x)f(x_j) + B_j(x)f'(x_j)
\]

where the functions \( A_j \) and \( B_j \) generalize the Lagrange basis functions:

\[
A_j(x) = (1 - 2l'_j(x_j)(x-x_j))l^2_j(x) \quad (5)
\]

\[
B_j(x) = (x-x_j)l^2_j(x) \quad (6)
\]

with

\[
l_j(x) = \prod_{k=0, k \neq j}^n \frac{(x-x_k)}{(x_j-x_k)}.
\]

• Calculate explicitly (by hand) the cubic hermite polynomials \( A_0, A_1, B_0 \) and \( B_1 \) for \( x_0 = 0 \) and \( x_1 = h, h > 0 \). For that, Assume for example that \( A_0 \) is of the form

\[
A_0(x) = a + bx + cx^2 + dx^3.
\]

Then, calculate coefficients \( a, b, c \) and \( d \) by imposing the constraints \( A_0(0) = 1, A_0(h) = 0, A'(0) = A'(h) = 0 \). Do do the same for \( A_1, B_0 \) and \( B_1 \) to get the final results. Verify that the results correspond to formulas (5) and (6).
• We want now to build a piecewise Hermite interpolation $H_N(x)$ of degree 3 for the function $f(x) = 1/(1 + x^2)$ in $x = [-4, 4]$. For that, we divide $[-4, 4]$ in $N$ intervals of size $8/N$. In each of those intervals, we approximate $f$ using cubic Hermite polynomials. The result is the piecewise cubic function $H_N(x)$.

Write a MATLAB program that computes $H_N(x)$ and plot $H_N(x)$, $x \in [-4, 4]$ for different values of $N$. Superimpose $f$ to that graph.

Write a MATLAB program that computes $H'_N(x)$ and plot $H'_N(x)$, $x \in [-4, 4]$ for different values of $N$. Superimpose $f'$ to that graph.

Write a MATLAB program that computes $H''_N(x)$ and plot $H''_N(x)$, $x \in [-4, 4]$ for different values of $N$. Superimpose $f''$ to that graph.

Compute $e_N = \max_{x \in [-4, 4]} |H_N(x) - f(x)|$ for $N = 1, 2, 3 \ldots$. Plot $e_N$ vs. $N$ using a log scale on the $y$ axis. Comment that graph.
Problem 1 (Solution)

A MATLAB code can be downloaded at [http://www.caam.rice.edu/~caam453/matlab/HW3_P1.m](http://www.caam.rice.edu/~caam453/matlab/HW3_P1.m).

Problem 2 (Solution)

Let us show that

\[ f(x) = f(0) + f'(0)x + \int_0^x f''(s)(x-s)ds. \]

We integrate by parts the integral as

\[ \int_0^x f''(s)(x-s)ds = -\int_0^x f'''(s)\frac{(x-s)^2}{2}ds + f''(s)\frac{(x-s)^2}{2}\big|_0^x = f''(0)\frac{x^2}{2} + \int_0^x f'''(x-s)^2ds. \]

In general

\[ n \int_0^x f^{(n)}(s)(x-s)^{n-1}ds = f^{(n)}(0)x^n + \int_0^x f^{(n+1)}(x-s)^n ds. \]

Then,

\[ \frac{p(x) - f(0)}{x - 0} = \frac{p(x) - p(0)}{x - 0} = \frac{dp}{dx}. \]

The Mean value theorem states that, if a function \( f \in C^1[a, b] \), then there exists a point \( \alpha \in [a, b] \) such that

\[ f'(\alpha) = \frac{f(b) - f(a)}{b - a}. \]

We have then two results that we combine:

\[ |E(x)| = |f(x) - p(x)| = x(f'(0) - f'(\alpha)) + \int_0^x f''(s)(x-s)ds. \]

Using both Cauchy-Schwarz and the inequality that has been provided, we have

\[ |E(x)| \leq x\int_0^x |f''(s)|ds + \int_0^x |f''(s)(x-s)|ds \]
\[ \leq \sqrt{h}x\left(\int_0^h |f''(s)|^2 ds\right)^{1/2} + \left(\int_0^x |f''(s)|^2 ds\right)^{1/2} \]
\[ \leq \left(\sqrt{h} + \frac{\sqrt{h}}{3}\right)x\left(\int_0^h |f''(s)|^2 ds\right)^{1/2} \]
\[ \leq \sqrt{3hx}\left(\int_0^h |f''(s)|^2 ds\right)^{1/2}. \]

Finally,

\[ \int_0^h |E(x)|^2 dx \leq 3h \int_0^h x^2 dx \int_0^h |f''(s)|^2 ds = h^4 \int_0^h |f''(s)|^2 ds = h^4 \|f''\|_{L^2}^2. \]
Problem 3 (Solution)

Cubic Hermite functions are given as

\[
\begin{align*}
A_0(x) &= (2t + 1)(t - 1)^2 \\
B_0(x) &= Lt(t - 1)^2 \\
A_1(x) &= t^2(-2t + 3) \\
B_1(x) &= -L(1 - t)t^2 .
\end{align*}
\]  

(7)

with \( t = x/L \).

A MATLAB code can be downloaded at [http://www.caam.rice.edu/~caam453/matlab/HW3_P3.m](http://www.caam.rice.edu/~caam453/matlab/HW3_P3.m).