TAM 470 Introduction to Computational Mechanics: Assignment 1

Due: Friday, September 9, 2016.

Note: The exercises below refer to pages 8-10 in the Moin text. When a problem asks you to write a program, do so — do not work the solution by hand as a means to avoid writing code. Program your own Lagrange interpolation algorithms, do not use canned software, such as those found in Numerical Recipes or MATLAB. You may, however, use available library functions for common linear algebra operations, such as matrix multiplication or solving a matrix equation and for computing splines (except where noted). For function plots, use a computer plot package of your choice, rather than a hand sketch, and use enough data points to give a good representation of curved function plots.
**Problem 1 (15 points):** (This problem is to be done by hand.)

(a) Consider the Lagrange polynomial of degree two, \( p_2(x) \), that passes through \((x_j, f_j)\) for uniformly spaced nodes on \([0, 1]\) and \( f(x) = \sin \pi x \). Evaluate \( p_2\left(\frac{1}{4}\right) \).

What is the derivative of \( p_2 \) at \( x = \frac{1}{4} \)?

(b) Consider the biquadratic Lagrange polynomial,

\[
p(x, y) := \sum_{j=0}^{2} \sum_{i=0}^{2} f_{ij} l_i(x) l_j(y),
\]

with uniformly spaced nodes, \( \{x_i\} = [0, \frac{1}{2}, 1], \{y_i\} = [0, \frac{1}{2}, 1] \), and \( f_{ij} := f(x_i, y_i) \).

For \( f = \sin \pi x \sin \pi y \), evaluate \( p(x, y) \) for \((x, y) = (\frac{1}{4}, \frac{1}{4}), (\frac{1}{2}, \frac{1}{4}), \) and \((\frac{3}{4}, \frac{1}{4})\). Draw a sketch (by hand, if you wish) of \( p(x, y) \) for \((x, y) \in [0, 1]^2\).

Evaluate \( \frac{\partial p}{\partial x} \) at the same points. Evaluate \( \frac{\partial p}{\partial y} \) at the same points.

(c) Show (in two lines or less) that

\[
\sum_{j=0}^{n} l_j(x) \equiv 1
\]

for all \( x \) if \( l_j(x) \in \mathbb{P}_n \) are the usual Lagrange cardinal basis functions based on distinct nodes, \( x_0, x_1, \ldots, x_n \).

**Problem 2 (15 points):**

Write your own Lagrange interpolation function to answer the questions in Moin, Exercise 1, p.8 related to the Runge function

\[
f(x) = \frac{1}{1 + 25x^2},
\]

(2)

It will be most convenient for you if you write a Lagrange interpolant script (".m" file) that has the same format as the matlab `spline` function, i.e.,

\[
p = \text{lagrange}(x, f, xx);
\]

In addition to answering the questions, please provide the source for your `lagrange.m` (only) with your HW submission.
Problem 3 (15 points):

For the next two problems extend the matlab driver routine `lin_int.m` provided on the course site to run your tests. As provided in the driver, test over a range of $n$ values, $n = 1, 2, \ldots, 100$. Note that eq. (3) is already implemented in this driver.

In the following, extend Problem 2 to interpolate $f(x)$ (equation 2) with

- Lagrange interpolation on Chebyshev-Lobatto nodes, $x_j = -\cos(j\pi/n),$
- Matlab’s cubic spline routine (provided),
- and with piecewise linear interpolation (also provided).

a. Use this code to sample $f(x)$ (2) and $p(x)$ at $m$ Chebyshev points, $\tilde{x}_0, \ldots, \tilde{x}_m$, on the interval $[a, b] = [-1, 1]$, where $m = 10n$.

Plot, on a log-log scale, $e_n$ versus $n$ for the cubic spline, Chebyshev-Lagrange, and piece linear interpolation, where $e_n$ is the maximum pointwise error in your interpolant:

$$ e_n := \frac{\max_{i=0}^{m} |f(\tilde{x}_i) - p(\tilde{x}_i)|}{\max_{i=0}^{m} |f(\tilde{x}_i)|}. \quad (3) $$

What do you notice about the error behavior of these interpolants? In particular, is there anything unique about the spline convergence? What is the convergence rate of piecewise linear? Can you estimate convergence rates for the other interpolation schemes?

b. Repeat (3a) with the interpolation nodes and sampling interval taken to be $[a, b] = [-.95, 1.05]$.

As in (3a), make a log-log plot of $e_n$ versus $n$ for each scheme (three curves on a single figure). What do you notice about the convergence behavior compared to (3a)? What has changed?

c. Repeat (3b) with

$$ f(x) = e^x \cos(20x). \quad (4) $$

As in (3b), make a log-log plot of $e_n$ versus $n$ for each scheme (three curves on a single figure). Comment on the convergence behavior you see in the plots.

What conclusions can you make about the overall convergence properties of these methods?
Problem 4 (15 points):

Using the results of Moin, Exercise 10 (don’t solve the problem), write a routine to return the derivative matrix $D = \{d_{ij}\}$ that maps values $f_j$ to derivatives $p'(x_i)$ of the Lagrange polynomial. That is, if $p \in \mathbb{P}_n$ with $p(x_j) = f_j$, then

$$\left. \frac{dp}{dx} \right|_{x_i} = \sum_{j=0}^{n} d_{ij} f_j.$$  

Explain how this matrix can be used to plot $p'(x)$ for any $x$ on $[a,b]$.

Modify your routine from Problem 4 to sample $p'(x)$ and plot the derivative error,

$$d_n := \frac{\max_{i=0}^{m} |f'(\tilde{x}_i) - p'(\tilde{x}_i)|}{\max_{i=0}^{m} |f'(\tilde{x}_i)|},$$

as well as the interpolation error $e_n$ (3) for $f(x) = \sin x$ on $[0,10]$. Comment on relative sizes of $e_n$ and $d_n$.

Problem 5 (15 points):

The table above shows some of the data provided in blade.txt on the course website, which lists the nodal positions for the turbine blade cross-section pictured on the right.

Plot a parametric spline $(x(t), y(t))$ through the data. Consider two different parameterizations, one based on the index that is given (i.e., $t_i = i$), the other with $t_i$ approximating the arclength along the blade between $(x_0, y_0)$ and $(x_i, y_i)$.

Show both curves, along with the nodal points using axis equal in matlab. Also show a zoom on the region $[-1,1]^2$ (axis square; axis([-1 1 -1 1])); Comment on the quality of the results.
Problem 6 (25 points) For 4-Credit Hour Option Only:

Implement the periodic spline using equations (1.6) and (1.7) along with condition (d) on pp. 6–7 of Moin.

Note: for the questions below, $\Delta_i = h$ will be constant, which greatly simplifies the two tridiagonal matrices present in (1.7) and evaluation of (1.6). You may use this shortcut if you wish.

Compute a parametric spline as in the previous example, using $x_j = \cos(\pi j/2)$, $y_j = \sin(\pi j/2)$, for $j = 1, \ldots, 4$. Compare your periodic spline to the matlab one for the same data points by computing the maximum deviation of your interpolant from the unit circle.