MATH 541: Midterm II, Due: 05/08/2014, at beginning of class

I, __________________________, pledge that this exam is ***completely my own work,*** and that I did not take, borrow or steal any portions from any other person. Any and all references I used are clearly cited in my solutions. I understand that if I violate this honesty pledge, I am subject to disciplinary action pursuant to the appropriate sections of the San Diego State University Policies.

____

Signature

1. So, as mentioned in class, there is nothing inherently wrong with Lagrange interpolation. What is a bad idea is always using interpolation over equi-spaced points. One way to remedy this is to interpolate over the Chebyshev points

   $$x_j = -\cos\left(\frac{\pi j}{N}\right),\ j = 0, \cdots, N.$$

   As you will note though, $-1 \leq x_j \leq 1$, so these points may at first seem of limited value. Show how to use the Chebyshev points to define a mesh over any interval $[a, b]$, *i.e.* let

   $$\tilde{x}_j = a + c(x_j + 1),\ j = 0, \cdots, N.$$

   Find c so that $\tilde{x}_0 = a$ and $\tilde{x}_N = b$.

   Now, for the function $f(x) = \lambda \cos^2(\lambda x) + \sin(3\lambda^2 x)$, on the interval $[3, 7]$, compare Lagrange interpolation using an equi-spaced mesh with step size $h = .1$ and a Chebyshev mesh with the same number of points as the equi-spaced mesh. Do this for $\lambda = 1, 3, 5$. Plot a comparison of the function and the interpolatory approximation. Plot the error in using each method. Describe your results.

2. According to lectures and the book, if we have a jth order approximation $N_j(h)$ to a quantity M, where the error is of the form

   $$\textbf{Error}(h) = M - N_j(h) = \sum_{k=j}^{\infty} C_j h^j$$

   we can combine two computations $N_j(h/2)$, and $N_j(h)$ to achieve a $(j+1)$st order approximation $N_{j+1}(h)$, using the formula

   $$N_{j+1}(h) = N_j(h/2) + \frac{N_j(h/2) - N_j(h)}{2^j - 1}.$$

Further, homework suggests that we can combine two computations $N_j(h/3)$ and $N_j(h)$ to achieve a $(j+1)$st order approximation $N_{j+1}(h)$, using the formul

$$N_{j+1}(h) = N_j(h/3) + \frac{N_j(h/3) - N_j(h)}{3^j - 1}.$$

This seems to suggest that if we have two computations $N_j(h/m)$, where $m > 1$, and $N_j(h)$ it is possible to achieve a $(j+1)$st order approximation $N_{j+1}(h)$, using the formula

$$N_{j+1}(h) = N_j(h/m) + \frac{N_j(h/m) - N_j(h)}{m^j - 1}.$$

Show that this is true.

3. The value of $f(t) = \cos(\pi + t^3) + 1$ is to be computed for $t = 10^{-6}$ in a finite precision environment.

   (a) Show that

   $$f(t) = \cos\left(\pi + t^3\right) + 1 = -\frac{\sin^2(\pi + t^3)}{\cos(\pi + t^3) - 1}$$

   Why will using this new version of the function give you a more accurate result in a finite precision environment?

   (b) Using Taylor Series arguments, find another way to compute $f(t)$ for $t = 10^{-6}$. What is the order of error in your approach?

4. Suppose, for $A > 0$, you have a sequence defined by

   $$x_n = \frac{1}{2}x_{n-1} + \frac{A}{2x_{n-1}}, \;\; n \geq 1.$$

   Further, suppose that $x_0 > 0$.

   (a) Why is $x_n > 0$ for all $n \geq 1$?

   (b) Find the function $g(x)$ such that

   $$x_n = g(x_{n-1}).$$

   (c) Show this iterative scheme is just Newton's method used on the function $f(x) = x^2 - A$. Explain what this scheme is trying to approximate.

   (d) Find all positive fixed points of $g(x)$.

   (e) If x_n converges to the fixed point, what is the rate of convergence when x_n is close to the fixed point? Explain your answer.

5. We call the function

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

the Gaussian probability distribution of mean μ with standard deviation σ. If we are using this to model grades, this means the probability that someone gets a score between a and b, say $P(a,b)$, is given by

$$P(a,b) = \int_a^b f(x)dx.$$

Find, to within 10^{-5} using both the trapezoid and Simpson's rules the probabilities

$$P(\mu-\sigma,\mu+\sigma),\ P(\mu-2\sigma,\mu+2\sigma),\ P(\mu-3\sigma,\mu+3\sigma).$$

Using the result from either 'integral' or 'quadgk' in Matlab, turn in error plots that clearly show the order of error of both the trapezoid and Simpson's rules. How far can you push either method? Can you attain machine precision?

Further, show how you would set up the integrals for use with Legendre based Gaussian quadrature. Using this method, generate a plot of the log of the error against the number of nodes used. Explain how you would compare your results to those from either trapezoid or Simpson's rules. For example, how many points in the mesh are necessary to attain machine precision in all of the above methods?

(Extra Credit): You might also ask the question of which method is fastest for a given magnitude of error. To answer this, you can use the Matlab commands 'tic' and 'toc' in the following way:

tic

Block of code you want to time...

toc

Generate a plot of the time taken against the error produced in each of the three methods. Which method is the best?