Numerical Analysis II

Exam 1 **MATH 6371-13338 (S2013)** **Feb. 28 – Mar. 2, 2013**

Name and ID: ______________________________

- This is a 72 hour take-home exam. Please turn it in to me (Jiwen He) in a digital form at jiwenhe@math.uh.edu by 3:00pm on Sat., Mar. 2, i.e., 72 hours after you receive it. The exam is not supposed to take anything close to 72 hours; I am giving you 72 hours so you can take a number of long breaks (to eat, sleep, study for other classes, takes other exams, etc.).
- You may use any books, notes, but you may not discuss the exam with anyone until Mar. 2, after everyone has taken the exam.
- Please make a copy of your exam before handing it in.
- The problems are all equally weighted. This is to make things simple, and not because the problems are all of equal difficulty.
- Keep in mind that there are multiple approaches (and valid solutions) to some of the problems, and these approaches can differ considerably in complexity. Also, there are several tricky parts and at least one tricky problem, *so don't expect to be able to solve everything.*
- Since this is a take home exam, you will be graded on clarity and conciseness as well as accuracy and correctness. Please take the time and make the effort to make your solutions clear. Please try to use standard and simple notation, introducing only as many new symbols as is absolutely necessary. The solutions are not long, so if you find that your solution to a problem goes on and on for many pages, you should try to figure out a simpler one. I expect neat, legible exams from everyone.
- Please attach the cover page to the front of your exam. Assemble your solutions in order (problem 1, problem 2, problem 3, . . .), starting a new page for each problem.
- Please respect the honor code. Although I allow you to work on homework assignments in small groups, you cannot discuss the exam with anyone, at least until everyone has taken it.

20 points 1. *Exponential Interpolation.*

Some modeling considerations have mandated a search for a function

$$u(x) = c_0 e^{c_1 x + c_2 x^2}$$

where the unknown coefficients c_1 and c_2 are expected to be nonpositive. Given are data pairs to be interpolated, (x_0, z_0), (x_1, z_1), and (x_2, z_2), where $z_i > 0$, $i = 0, 1, 2$. Thus, we require $u(x_i) = z_i$.

The function $u(x)$ is not linear in its coefficients, but $v(x) = \ln(u(x))$ is linear in its.

Find a quadratic polynomial $v(x)$ that interpolates appropriately defined three data pairs, and then give a formula for $u(x)$ in terms of the original data.

20 points

2. *Weak Line Search.*

A popular technique arising in methods for minimizing functions in several variables involves a *weak line search*, where an approximate minimum x^* is found for a function in one variable, $f(x)$, for which the values of $f(0)$, $f'(0)$, and $f(1)$ are given. The function $f(x)$ is defined for all nonnegative x, has a continuous second derivative, and satisfies $f(0) < f(1)$ and $f'(0) < 0$. We then interpolate the given values by a quadratic polynomial and set x^* as the minimum of the interpolant.

(a) Find x^* for the values $f(0) = 1$, $f'(0) = -1$, $f(1) = 2$.

(b) Show that the quadratic interpolant has a unique minimum satisfying $0 < x^* < 1$. Can you show the same for the function f itself?

20 points

3. *Forward Difference Operator.*

Given a sequence y_0, y_1, y_2, . . . , define the *forward difference operator* Δ by

$$\Delta y_i = y_{i+1} - y_i.$$

Powers of Δ are defined recursively by

$$\begin{aligned}\Delta^0 y_i &= y_i,\\ \Delta^j y_i &= \Delta(\Delta^{j-1} y_i), \quad j = 1, 2, \cdots\end{aligned}$$

Thus, $\Delta^2 y_i = \Delta(y_{i+1} - y_i) = y_{i+2} - 2y_{i+1} + y_i$, etc.
Consider polynomial interpolation at equispaced points, $x_i = x_0 + ih$, $i = 0, 1, \cdots, n$.

(a) Show that

$$f[x_0, x_1, \ldots, x_j] = \frac{1}{j!h^j}\Delta^j f(x_0).$$

[Hint: use mathematical induction.]

(b) Show that the interpolating polynomial of degree at most n is given by the *Newton forward difference formula*

$$p_n(x) = \sum_{j=0}^{n} \binom{s}{j} \Delta^j f(x_0),$$

where

$$s = \frac{x - x_0}{h}, \quad \binom{s}{j} = \frac{s(s-1)\cdots(s-j+1)}{j!} \quad (\text{ with } \binom{s}{0} = 1)$$

$\boxed{\text{20 points}}$ 4. *Hermite Cubic Interpolation.*

Verify that the Hermite cubic interpolating $f(x)$ and its derivative at the points t_i and t_{i+1} can be written explicitly as

$$s_i(x) = f_i + (h_i f_i')\tau + \Big(3(f_{i+1} - f_i) - h_i(f_{i+1}' + 2f_i')\Big)\tau^2$$
$$+ \Big(h_i(f_{i+1}' + f_i') - 2(f_{i+1} - f_i)\Big)\tau^3,$$

where $h_i = t_{i+1} - t_i$, $f_i = f(t_i)$, $f_i' = f'(t_i)$, $f_{i+1} = f(t_{i+1})$, $f_{i+1}' = f'(t_{i+1})$, and $\tau = \frac{x-t_i}{h_i}$

20 points 5. *Complete Interpolating Cubic Spline.*

Given function values $f(t_0)$, $f(t_1)$, $\cdots$, $f(t_r)$, as well as those of $f'(t_0)$ and $f'(t_r)$, for some $r \geq 2$, it is possible to construct the complete interpolating cubic spline.
Suppose that we were instead to approximate $f'(t_i)$ by the divided difference $f[t_{i-1}, t_{i+1}]$, for $i = 1, 2, \cdots, r-1$, and then use these values to construct a Hermite piecewise cubic interpolant.
State one advantage and one disadvantage of this procedure over a complete cubic spline interpolation.

20 points

6. *Orthogonal Polynomials.*

Let $\phi_0(x)$, $\phi_1(x)$, $\phi_2(x)$, $\cdots$ be a sequence of orthogonal polynomials on an interval $[a, b]$ with respect to a positive weight function $w(x)$. Let x_1, $\cdots$, x_n be the n zeros of $\phi_n(x)$; it is known that these roots are real and $a < x_1 < \cdots < x_n < b$.

(a) Show that the Lagrange polynomials of degree $n-1$ based on these points are orthogonal to each other, so we can write

$$\int_a^b w(x)L_j(x)L_k(x)dx = 0, \quad j \neq k,$$

where

$$L_j(x) = \prod_{k\neq j} \frac{(x-x_k)}{(x_j-x_k)}, \quad 1 \leq j \leq n.$$

(b) For a given function $f(x)$, let $y_k = f(x_k)$, $k = 1, \cdots, n$. Show that the polynomial $p_{n-1}(x)$ of degree at most $n-1$ that interpolates the function $f(x)$ at the zeros x_1, $\cdots$, x_n of the orthogonal polynomial $\phi_n(x)$ satisfies

$$\|p_{n-1}\|^2 = \sum_{k=1}^{n} y_k^2 \|L_k\|^2$$

in the weighted least squares norm. This norm is defined by

$$\|g\|^2 = \int_a^b w(x)[g(x)]^2 dx$$

for any suitably integrable function $g(x)$.

20 points

7. *Best Approximation.*

Jane works for a famous bioinformatics company. Last year she was required by Management to approximate an important but complicated formula, $g(x)$, defined on the interval $[-1, 1]$, by a polynomial of degree $n + 1$. She did so, and called the result $f(x) = p_{n+1}(x)$.

Last week, Management decided that they really needed a polynomial of degree n, not $n + 1$, to represent g. Alas, the original g had been lost by this time and all that was left was $f(x)$. Therefore, Jane is looking for the polynomial of degree n which is closest (in the maximum norm) to f on the interval $[-1, 1]$. Please help her find it.