CME 108/MATH 114
Introduction to Scientific Computing
Summer 2017

Final Examination – August 2017, Due Friday 8/18, 5 pm PDT

Problem	Score
1	/ 13
2	/ 17
3	/ 15
4	/ 15
5	/ 20
6	/ 20
7	/ 15
Total	/ 115

Full Name

Student Number

Honor Code (Signature)

Exam Rules

1. First 6 problems are required; problem 7 is the bonus problem and will be up to 15% of the total. The maximum for the final is 115.
2. Open book and notes. You are welcome to use all Matlab scripts in posted homework solutions or in lecture materials, but do not share your exam scripts with others.
3. You may not communicate with other students in class during the exam period.
4. Piazza will be closed for questions.
5. There will be no assistance from teaching staff on technical questions. If you need clarification on exam questions, you may email the teaching staff; best effort will be made to respond in a timely manner.
6. Submission deadline on Gradescope will be strictly enforced. Please take responsibility to avoid technical glitches at time of submission (WiFi access, scanning issues, Matlab access, etc.).
7. Observe the Honor Code.

1. (13 points) Let f be a smooth function. Using the Taylor exansions of $f(x)$, $f(x+h)$, $f(x+2h)$, $f(x-h)$, and $f(x-2h)$, develop a fourth order central difference approximation to $f''(x)$ of the form

$$f''(x_j) = \alpha_{-2} f_{j-2} + \alpha_{-1} f_{j-1} + \alpha_0 f_j + \alpha_1 f_{j+1} + \alpha_2 f_{j+2} + O(h^4),$$

where the α_k are coefficients to be determined and as usual we assume a uniformly spaced grid and we let $f_{j+b} = f(x_j + bh)$.

Report each α_k as a rational multiple of an integer power of h.

2. (17 points) Consider the following scenario. Let $x_0, x_1, \ldots, x_n$ denote chosen points on a given interval. We are given both the function values $f(x_i)$ **and** the first derivative values $f'(x_i)$, for $i = 0, 1, \ldots n$. We would like to find a polynomial P of degree at most $2n+1$ which interpolates both the the function and its first derivative. That is, we would like to find P such that

$$P(x_i) = f(x_i), \text{ and } P'(x_i) = f'(x_i). \tag{1}$$

Define

$$\begin{aligned} H_i(x) =& (1 - 2L'_{n,i}(x_i)(x - x_i))L^2_{n,i}(x) \text{ and} \\ \hat{H}_i(x) =& (x - x_i)L^2_{n,i}(x), \end{aligned}$$

where $L_{n,i}$ denotes the ith degree n Lagrange basis polynomial.

Show that the polynomial

$$P(x) = \sum_{i=0}^{n} H_i(x) f(x_i) + \sum_{i=0}^{n} \hat{H}_i(x) f'(x_i)$$

satisfies $\deg(P) \leq 2n+1$ and the conditions listed in 1 by first showing that

$$H_i(x_j) = \delta_{ij}, \text{ and } H'_i(x_j) = 0,$$

and then deriving similar relations for $\hat{H}_i$. (Note that δ_{ij} denotes the Kronecker delta. You may find its definition here.)

3. (15 points) Many of the quadrature rules we have studied take function values from a uniformly spaced grid. It turns out, however, that such choice of points is far from optimal.

Define a new quadrature by

$$I = \int_a^b f(x)dx \approx \frac{b-a}{2}\left[f\left(\frac{b+a}{2} - \sqrt{\frac{1}{3}}\frac{b-a}{2}\right) + f\left(\frac{b+a}{2} + \sqrt{\frac{1}{3}}\frac{b-a}{2}\right)\right]. \tag{2}$$

Show that, with only two function evaluations, the quadrature given in (2) can integrate polynomials of degree at most 3 exactly.

Hint: Consider first applying the change of variables $x = \frac{b-a}{2}t + \frac{b+a}{2}$ to transform I to an integral over a symmetric interval. Further, recall that the definite integral is linear and that every polynomial of degree at most 3 is a linear combination of terms of the form x^j, for $j = 0, 1, 2, 3$.

4. (15 points) Use the Newton-Raphson method to solve the following system of nonlinear equations:

$$\begin{aligned} 8x^2 + y^2 - z &= -1, \\ 2x + 3y^2 + z &= 1, \text{ and} \\ 4x^2 + 9y^2 + z^2 - 4x &= 1. \end{aligned}$$

Set `tol = 1.e-6` and `max_iter = 100`. Use $x^{(0)} = [1, 1, 1]^T$ as a starting point and note that your algorithm should converge to a solution.

5. (20 points) Consider the function $f(x, y) = 2x^2 - 2x + 4y^2 - 6y - 2xy + 7$. Show that the Newton-Raphson method for optimization converges in this case in a single step, regardless of the starting point $x^{(0)}$.

   Do this by finding the minimum of f analytically (as a critical point of the function), computing the Hessian and gradient by hand and carrying out the necessary calculations.

   Then verify your computations by using an implementation of the optimization method in Matlab, using `tol = 1.e-6`.

6. (20 points) Consider the following boundary value problem

$$\begin{aligned} &(x-1)y'' - xy' + y = -(1-x)^2; \quad x \in [0, 2] \\ &y(0) = 0; \quad y(2) + \frac{1}{2}y'(2) = -\frac{1}{4} \end{aligned}$$

   Using $N = 41$ node points,

   (a) Derive the tri-diagogal system to solve the ODE using central difference for both 1st- and 2nd-derivatives. Identify elements of vectors a, b, c and f.

   (b) and plot your numerical solution against analytical solutions. You can obtain the analytical solution from Wolfram/Alpha, and evaluate the arbitrary constants.

7. (15 points) Bonus PDE question (up to 15% of total points of problems 1 through 6)

   The steady state temperature distribution $T(x, y)$ in a rectangular copper plate, $0 \leq x \leq 2$ and $0 \leq y \leq 1$, satisfies the Laplace equation

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0,$$

   subject to the following boundary conditions:

   - The upper and lower boundaries ($y = 0$ and $y = 1$) are perfectly insulated, i.e., the normal derivative of T is zero at these boundaries.
   - The left side ($x = 0$) is kept at $0°$, and the right side at $f(y) = y$.

   You need to write a program to evaluate $T(x, y)$ numerically.

(a) Discretize the Laplace equation using second-order finite difference approximation of the second derivatives. At the upper and lower boundaries ($y = 0$ and $y = 1$), also use second-order finite difference for the Neumann boundary conditions. Show equation for $T_{i,j}^{(k+1)}$ for interior points, and similar equations for Neumann boundary points.

(b) Then write a program to compute the solution using the **SOR method** with $\omega = 1.25$ and $\omega = 1.75$. You should use $M = N = 11$ points in the x- and y-directions (including boundary points), respectively (note: the grid spacing is not the same in x and y directions).

The program should iterate following the specifications below:

- Start with initial temperature distribution $T_{i,j} = 0$ for all i and j, except for the boundary elements corresponding to $T = y$.
- Iterations should stop when the solution reaches steady-state – the residues between iterations are $\leq 10^{-5}$.

Record the total number of iterations required. Monitor your solution at the center of the plate, T_c at $(x, y) = (1, 0.5)$, at each iteration k and compare with the exact solution:

$$\epsilon(k) = \left| \frac{T_c^{\text{exact}} - T_c(k)}{T_c^{\text{exact}}} \right|.$$

Plot the relative error $\epsilon(k)$ as the iteration progresses (similar to Fig. 8.7).

The exact solution at $(1, 0.5)$ is $T(1, 0.5) = 0.25$.