AM205: Take-home midterm exam (Fall 2017)

This exam was posted at 5 PM on November 9th. Answers are due at 5 PM on November 11th. Solutions should be uploaded to Canvas.

For queries, contact the teaching staff using a ***private*** *message on Piazza—do not post questions publicly. Any clarifications will be posted on Piazza.*

The exam is open book—any class notes, books, or online resources can be used. The exam must be completed by yourself and no collaboration with classmates or others is allowed. The exam will be graded out of forty points. Point values for each question are given in square brackets.

1. **Polynomial approximation of the gamma function [6].**

   (a) The gamma function is defined as

   $$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t}\, dt \tag{1}$$

   and satisfies $(n-1)! = \Gamma(n)$ for integers n. Construct an approximation to the gamma function by finding the polynomial interpolant of the following points:

n	1	2	3	4	5
$\Gamma(n)$	1	1	2	6	24

   Write the interpolant as $g(x) = \sum_{k=0}^{4} g_k x^k$, and include the values of the coefficients[1] g_k in your solutions.

   (b) Construct a second approximation to the gamma function by first calculating the fourth order polynomial $p(x)$ that interpolates the points $(n, \log(\Gamma(n)))$ for $n = 1, 2, 3, 4, 5$. Then define the approximation by $h(x) = \exp(p(x))$.

   (c) Plot $\Gamma(x)$,[2] $g(x)$, and $h(x)$ on the interval $1 \le x \le 5$.

   (d) Calculate the maximum relative error between $\Gamma(x)$ and $g(x)$ on the interval $1 \le x \le 5$, accurate to at least three significant figures.[3] Repeat this for $\Gamma(x)$ and $h(x)$. Which of the two approximations is more accurate?

2. **Least-squares spline fitting [8].** For $k = 0, 1, \ldots, 8$, define the function $s_k(x)$ to be a cubic spline on the domain $[0, 8]$. Each spline has control points at $x = 0, 1, \ldots, 8$ with corresponding function values

   $$s_k(j) = \begin{cases} 1 & \text{if } j = k, \\ 0 & \text{if } j \neq k. \end{cases} \tag{2}$$

   Each spline is defined in the standard way, where its first and second derivatives are continuous at the interior control points, and the natural end point conditions $s_k''(0) = s_k''(8) = 0$ are

[1] You can use numerical linear algebra and/or library functions to find the polynomial, but be sure to include these coefficients in the writeup.
[2] In Python, the gamma function is available in the `scipy.special` module. In MATLAB it is called `gamma`.
[3] This should be done numerically.

used.[4] Define a linear combination of these splines as

$$S(x;b) = \sum_{k=0}^{8} b_k s_k(x) \tag{3}$$

where $b = (b_0, b_1, \ldots, b_8)$ is a vector of parameters.

(a) A text file `sdata1.txt` is provided containing a number of data points (x_i, y_i) for $i = 1, \ldots, N$. Find the value of b that minimizes the residual

$$r(b) = \sum_{i=1}^{N} |S(x_i; b) - y_i|^2. \tag{4}$$

Plot the points and the spline.

(b) Repeat part (a) for the second set of data points in the `sdata2.txt`. Plot the points and the spline.

(c) You should find that one of your two plots looks reasonable, whereas the other does not. Explain the problem with the plot that does not look reasonable.

3. **Stability of a scheme for the heat equation [13].**

(a) Let f be a smooth function and consider the finite difference formula

$$f''_{\text{diff}}(x) = \frac{af(x) + b(a)\left(f(x+h) + f(x-h)\right) + c(a)\left(f(x+2h) + f(x-2h)\right)}{h^2} \tag{5}$$

where $a \in \mathbb{R}$. Here, $b(a)$ and $c(a)$ are affine functions of a. Calculate $b(a)$ and $c(a)$ in order to make $f''_{\text{diff}}(x)$ equal to the true second derivative, $f''(x)$, up to $O(h^2)$ accuracy.

(b) Consider solving the diffusion equation $u_t = u_{xx}$ with the discretization

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \frac{aU_j^n + b(a)(U_{j+1}^n + U_{j-1}^n) + c(a)(U_{j+2}^n + U_{j-2}^n)}{h^2} \tag{6}$$

where the numerical values U_j^n approximate the continuous function according to $U_j^n \approx u(jh, n\Delta t)$. Define the constant $\mu = \Delta t / h^2$. Perform a stability analysis by substituting the ansatz $U_j^n(k) = [\lambda(k)]^n e^{ikjh}$ into Eq. 6. Find the amplification factor $\lambda(k)$, and write your solution so that it only involves μ, a, and the parameter s defined as $s = \sin^2 \frac{hk}{2}$.

(c) Consider the case when $a = -2$. Show that in this case, Eq. 5 reduces to the usual forward Euler discretization for the heat equation discussed in lecture 16, and your formula from part (b) is consistent with the stability analysis in the lecture derivation when the parameter θ is set to zero.

(d) Use your answer from part (b) to show that there is a value a_* such that if and only if $a > a_*$, the numerical scheme in Eq. 5 is always unstable, regardless of the value of μ.

[4] You can either write your own routine to compute the splines, or you can use a library function. If you use a library function you should ensure that the spline definition exactly matches the specified conditions.

(e) Use your answer from part (b) to show that if $a \le a_*$ the numerical scheme is conditionally stable, and find the range of μ as function of a for which stability is achieved.

(f) Show that setting $a = a_*$ allows for the largest stable value of μ to be chosen. By considering the initial condition $U_j^0 = (-1)^j$, or otherwise, explain why even though this scheme is formally stable, it may not be appropriate for practical usage.

4. **Solving an elliptic problem on an irregular domain [13].** Consider the two-dimensional domain Ω that consists of the square $[0,3]^2$ with three regions

$$\begin{aligned} R &= \{\mathbf{x} \in \mathbb{R}^2 \mid x > 2 \text{ and } y > 1\}, \\ S &= \{\mathbf{x} \in \mathbb{R}^2 \mid x < 1 \text{ and } y < 1\}, \\ T &= \{\mathbf{x} \in \mathbb{R}^2 \mid x < 1 \text{ and } y > 2\} \end{aligned} \tag{7}$$

removed from it. Here $\mathbf{x} = (x, y)$ is a two-dimensional vector. The domain Ω is shown in Fig. 1. Consider the steady state reaction–diffusion equation

$$-\nabla^2 v + v = f \tag{8}$$

for an unknown function $v(x, y)$ and a source term $f(x, y)$, using the boundary condition of $v(x, y) = 0$ on the boundary $\partial\Omega$. Introduce a discretized solution $V_{j,k} \approx v(jh, kh)$ and a discretized source term $F_{j,k} = f(jh, kh)$ on a $M \times M$ grid where $M = 52$ and the grid spacing is $h = 3/(M-1)$. The gridpoints (j, k) for $j = 0, \ldots, M-1$ and $k = 0, \ldots, M-1$ fall into three categories:

- they lie in the *interior* of Ω (*e.g.* $(j, k) = (26, 26)$);
- they lie on the *boundary* of Ω (*e.g.* $(j, k) = (17, 17)$);
- they are *exterior* to Ω (*e.g.* $(j, k) = (35, 18)$).

(a) Suppose that the Laplacian operator is discretized as

$$\begin{aligned} (\nabla^2 v)_{j,k} \approx \alpha V_{j,k} + \beta\big(&V_{j-2,k} + V_{j-1,k} + V_{j+1,k} + V_{j+2,k} \\ &+ V_{j,k-2} + V_{j,k-1} + V_{j,k+1} + V_{j,k+2}\big). \end{aligned} \tag{9}$$

To begin, assume that all of the values of V on the right hand side are non-exterior. By referencing question 3(a), or otherwise, find the values of α and β in terms of h, in order to make Eq. 9 accurate up to second order.

(b) Consider an interior gridpoint (j, k) where the neighbor $(j+1, k)$ is a boundary, and $(j+2, k)$ is exterior. Using a ghost value of $V^{\mathrm{g}}_{j+2,k} = -V_{j,k}$, write down the modified version of discretized Laplacian at the point (j, k). In the subsequent sections the same procedure can be used to handle any missing points in the stencil.

(c) Write a program to solve Eq. 8 numerically using the discretizations from parts (a) and (b). Your program should represent the solution as a vector V consisting only of the values $V_{j,k}$ at interior gridpoints.[5] Let F be the corresponding source vector and assemble the matrix A such that

$$AV = F. \tag{10}$$

[5] This will require re-indexing the two-dimensional interior gridpoints into a one-dimensional vector, and you will need to devise a procedure to do this. Hint: the resulting vector V should have exactly 1344 components.

Verify that A is symmetric by showing that $||A - A^\mathsf{T}||_F$ is zero to within numerical roundoff. Use a source term of $f(x, y) = 3 - (x - 1.5)^2 + (y - 1.5)^2$. Solve Eq. 10 by finding the Cholesky factorization[6] $A = LL^\mathsf{T}$ and then directly solving the triangular systems $LB = F$ and $L^\mathsf{T} V = B$.[7] Make a plot of the numerically computed solution in the (x, y) plane, either as a surface plot or a contour plot.

(d) Find the index (j, k) where the solution $V_{j,k}$ from part (c) is maximized, and report the maximum value to at least six significant figures.

[6] You can use a library function to compute the Cholesky factorization.

[7] The triangular solves can be performed either with a library function or by writing your own routines to do the forward and back substitutions. Since the aim of the question is to use Cholesky, you should not use a direct black-box solver like `numpy.linalg.solve`.

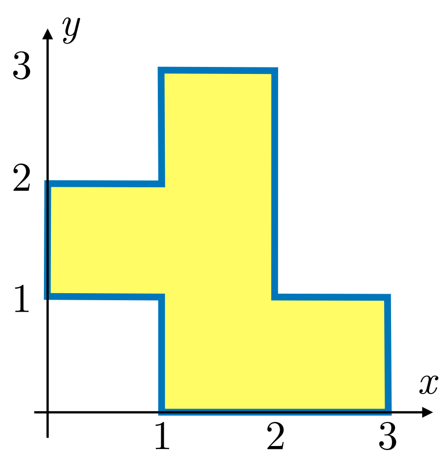

Figure 1: A diagram of the two-dimensional domain Ω considered in question 4. The interior of the domain is shown in yellow, and the boundary $\partial\Omega$ is shown in blue.