

# AM205: Assignment 1 (due 5 PM, September 23)

For this assignment, first complete problems 1, 2, 3, and 4, and then complete *either* problem 5 (on image/data analysis) or problem 6 (on physics/modeling). If you submit answers for both, then your grade will be calculated using the maximum score from the two.

## 1. Polynomial approximation of the gamma function.

- (a) The **gamma function** is defined as

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

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

$n$	1	2	3	4	5	6
$\Gamma(n)$	1	1	2	6	24	120

Write the interpolant as  $g(x) = \sum_{k=0}^5 g_k x^k$ , and include the values of the coefficients  $g_k$  in your solutions.

- (b) Construct a second approximation to the gamma function by first calculating the fifth order polynomial  $p(x)$  that interpolates the points  $(n, \log(\Gamma(n)))$  for  $n = 1, 2, 3, 4, 5, 6$ . Then define the approximation by  $h(x) = \exp(p(x))$ .
- (c) Plot  $\Gamma(x)$ ,<sup>1</sup>  $g(x)$ , and  $h(x)$  on the interval  $1 \leq x \leq 6$ .
- (d) Calculate the maximum relative error between  $\Gamma(x)$  and  $g(x)$  on the interval  $1 \leq x \leq 6$ , accurate to at least three significant figures.<sup>2</sup> Repeat this for  $\Gamma(x)$  and  $h(x)$ . Which of the two approximations is more accurate?
2. (a) **Error bounds with Lagrange polynomials.** Let  $f(x) = e^{-3x} + e^{2x}$ . Write a program to calculate and plot the Lagrange polynomial  $p_{n-1}(x)$  of  $f(x)$  at the Chebyshev points  $x_j = \cos((2j-1)\pi/2n)$  for  $j = 1, \dots, n$ . For  $n = 4$ , over the range  $[-1, 1]$ , plot  $f(x)$  and Lagrange polynomial  $p_3(x)$ .
- (b) Recall from the lectures that the infinity norm for a function  $g$  on  $[-1, 1]$  is defined as  $\|g\|_\infty = \max_{x \in [-1, 1]} |g(x)|$ . Calculate  $\|f - p_3\|_\infty$  by sampling the function at 1,000 equally-spaced points over  $[-1, 1]$ .
- (c) Recall the interpolation error formula from the lectures,

$$f(x) - p_{n-1}(x) = \frac{f^{(n)}(\theta)}{n!} (x - x_1)(x - x_2) \dots (x - x_n) \quad (2)$$

<sup>1</sup>In Python, the gamma function is available in the `scipy.special` module. In MATLAB it is called `gamma`.

<sup>2</sup>This should be done numerically by sampling the functions at regular intervals that are small enough to obtain the required accuracy.

for some  $\theta \in [-1, 1]$ . Use this formula to derive an upper bound for  $\|f - p_{n-1}\|_\infty$  for any positive integer  $n$ . Your bound should be a mathematical formula, and should not rely on numerical sampling.

- (d) Find a cubic polynomial  $p_3^\dagger$  such that  $\|f - p_3^\dagger\|_\infty < \|f - p_3\|_\infty$ .
3. **The condition number of a matrix.** For a  $2 \times 2$  invertible matrix  $A$ , define the condition number to be  $\kappa(A) = \|A\| \|A^{-1}\|$  as discussed in the lectures. Assume that the matrix norm is defined using the Euclidean vector norm.
- (a) Find two  $2 \times 2$  invertible matrices  $B$  and  $C$  such that  $\kappa(B + C) < \kappa(B) + \kappa(C)$ .
- (b) Find two  $2 \times 2$  invertible matrices  $B$  and  $C$  such that  $\kappa(B + C) > \kappa(B) + \kappa(C)$ .
- (c) **Optional for the enthusiasts.** Suppose that  $A$  is a symmetric invertible  $2 \times 2$  matrix. Find  $\kappa(2A)$  and  $\kappa(A^2)$  in terms of  $\kappa(A)$ . Do the results from part (c) hold if  $A$  is not symmetric? Either prove the results, or find counterexamples.
4. **Periodic cubic splines.** In the lectures we discussed the construction of cubic splines to interpolate between a number of control points. We found that it was necessary to impose additional constraints at the end points of the spline in order to have enough constraints to determine the cubic spline uniquely. Here, we examine the construction of cubic splines on a periodic interval  $t \in [0, 4)$ , where  $t = 0$  is equivalent to  $t = 4$ . By working in a periodic interval, this simplifies the spline construction and no additional end point constraints are required.
- (a) Consider the four points  $(t, x) = (0, 0), (1, 1), (2, 0), (3, -1)$ . Construct a cubic spline  $s_x(t)$  that is piecewise cubic in the four intervals  $[0, 1), [1, 2), [2, 3)$ , and  $[3, 4)$ . At  $t = 0, 1, 2, 3$  the cubics should match the control points, giving eight constraints. At  $t = 0, 1, 2, 3$  the first and second derivatives should match, giving and additional eight constraints and allowing  $s_x(t)$  to be uniquely determined.
- (b) Plot  $s_x(t)$  and  $\sin(t\pi/2)$  on the interval  $[0, 4)$  and show that they are similar.
- (c) Construct a second cubic spline  $s_y(t)$  that goes through the four points  $(0, 1), (1, 0), (2, -1)$ , and  $(3, 0)$ . Plot  $s_y(t)$  and  $\cos(t\pi/2)$  on the interval  $0 \leq t < 4$  and show that they are similar.
- (d) In the  $xy$ -plane, plot the parametric curve  $(s_x(t), s_y(t))$  for  $t \in [0, 4)$ . Calculate the area enclosed by the parametric curve, and use it to estimate  $\pi$  to at least five decimal places,<sup>3</sup> using the relationship  $A = \pi r^2$  where  $r$  is taken to be 1.
- (e) **Optional for the enthusiasts.** In parts (a) to (c), you constructed cubic spline approximations for sine and cosine using four equally-spaced control points. Generalize this to construct cubic spline approximations for sine and cosine using  $n$  equally-spaced control points. What is the rate of convergence of the approximation of  $\pi$  as  $n$  is increased?

---

<sup>3</sup>You should calculate the estimate to at least five decimal places. However the estimate will not match  $\pi$  to five decimal places.

5. **Image reconstruction using low light.** In the program files, you will find a directory called `problem5/objects` that contains several photos of a still-life scene with different objects. One is a regular photo and the other three are low-light photos that were illuminated in different colors. In the program files on the website, you will find these photos in three different resolutions of  $(M, N) = (400, 300), (800, 600), (1600, 1200)$ . Choose one set to work with.<sup>4</sup> You will also find several code examples for how to read images into arrays.

Each pixel in the image can be represented as a three-component vector  $\mathbf{p} = (R, G, B)$  for the red, green, and blue components. Let  $\mathbf{p}_k^A$  be the  $k$ th pixel of the regular photo, and let  $\mathbf{p}_k^B$ ,  $\mathbf{p}_k^C$ , and  $\mathbf{p}_k^D$  be the  $k$ th pixel of the three low-light photos. Here,  $k$  is indexed from 0 up to  $MN - 1$ .

- (a) Consider reconstructing the regular photo from the three low-light photos. The regular photo pixel could be obtained from the low-light photo pixels using

$$\mathbf{p}_k^A = F^B \mathbf{p}_k^B + F^C \mathbf{p}_k^C + F^D \mathbf{p}_k^D + \mathbf{p}_{\text{const}} \quad (3)$$

where  $F^B$ ,  $F^C$ , and  $F^D$  are  $3 \times 3$  matrices and  $\mathbf{p}_{\text{const}}$  is a vector. Write a program to find the least-squares fit for the 30 components of the matrices  $F^B$ ,  $F^C$ ,  $F^D$ , and the vector  $\mathbf{p}_{\text{const}}$ . Specifically, your program should minimize

$$S = \frac{1}{MN} \sum_{k=0}^{MN-1} \|F^B \mathbf{p}_k^B + F^C \mathbf{p}_k^C + F^D \mathbf{p}_k^D + \mathbf{p}_{\text{const}} - \mathbf{p}_k^A\|_2^2. \quad (4)$$

Calculate  $S$  for the fitted values of  $F^B$ ,  $F^C$ ,  $F^D$ , and  $\mathbf{p}_{\text{const}}$ . Reconstruct a regular image using the pixel values given by Eq. 3 and include it in your writeup.<sup>5</sup> Compare it to the original regular image.

- (b) Chris also took a similar set of photos of two of his favorite bears, located in the directory `problem5/bears`.<sup>6</sup> Using the three low-light images of the bears, plus your fitted model from part (a), create a reconstructed regular image and include it in your writeup. In addition, calculate

$$T = \frac{1}{MN} \sum_{k=0}^{MN-1} \|\mathbf{p}_k^A - \mathbf{p}_k^{A*}\|_2^2 \quad (5)$$

where  $\mathbf{p}_k^A$  and  $\mathbf{p}_k^{A*}$  are the pixel colors in the original and reconstructed regular images, respectively.

<sup>4</sup>This choice is up to you. Higher resolution provides more accuracy, but will take longer to process.

<sup>5</sup>Typically, the pixel channel contributions will cover a fixed range, such as from 0.0 (black) up to 1.0 (full color). Some pixel colors you obtain from Eq. 3 may lie outside this range, in which case you may have to truncate them (e.g.  $1.2 \rightarrow 1$  or  $-0.3 \rightarrow 0$ ) in order to make an image.

<sup>6</sup>The polar bear on the left is called “PB the Eager” due to his eager expression, and he sometimes inspires Chris to work by sitting on his desk. The right bear is called “OG,” which stands for Orsetto Grande, roughly meaning “big teddy bear” in Italian. OG belongs to Chris’s **wife** and is originally from Italy.

- (c) **Optional for the enthusiasts.** Design an improved procedure that can reduce the difference  $T$  between the reconstructed and original regular images of the bears.
6. **Determining hidden chemical sources.** Suppose that  $\rho(\mathbf{x}, t)$  represents the concentration of a chemical diffusing in two-dimensional space, where  $\mathbf{x} = (x, y)$ . The concentration satisfies the diffusion equation

$$\frac{\partial \rho}{\partial t} = b \nabla^2 \rho = b \left( \frac{\partial^2 \rho}{\partial x^2} + \frac{\partial^2 \rho}{\partial y^2} \right), \quad (6)$$

where  $b$  is the diffusion coefficient. If a localized point source of chemical is introduced at the origin at  $t = 0$ , its concentration satisfies

$$\rho_c(\mathbf{x}, t) = \frac{1}{4\pi b t} \exp \left( -\frac{x^2 + y^2}{4bt} \right). \quad (7)$$

- (a) Show by direct calculation, or otherwise, that the concentration  $\rho_c$  satisfies Eq. 6.
- (b) Set  $b = 1$ , and now suppose that 49 point sources of chemicals are introduced at  $t = 0$  with different strengths, on a  $7 \times 7$  regular lattice covering the coordinates  $x = -3, -2, \dots, 3$  and  $y = -3, -2, \dots, 3$ . By linearity of Eq. 6 the concentration will satisfy

$$\rho(\mathbf{x}, t) = \sum_{k=0}^{48} \lambda_k \rho_c(\mathbf{x} - \mathbf{v}_k, t), \quad (8)$$

where  $\mathbf{v}_k$  is the  $k$ th lattice site and  $\lambda_k$  is the strength of the chemical introduced at that site. In the program files, you will find a file `measured-concs.txt` that contains many measurements of the concentration field,  $\rho_i = \rho(\mathbf{x}_i, 4)$ , for when  $t = 4$ . By any means necessary, determine the concentration strengths  $\lambda_k$ .

- (c) Suppose that the measurements have some experimental error, so that the measured values  $\tilde{\rho}_i$  in the file are related to the true values  $\rho_i$  according to

$$\tilde{\rho}_i = \rho_i + e_i \quad (9)$$

where the  $e_i$  are normally distributed with mean 0 and variance  $10^{-8}$ . Construct a hypothetical sample of the true  $\rho_i$ , and repeat your procedure from part (b) to determine the concentrations  $\lambda_k$ . Repeat this sampling procedure for at least 100 times, and use it to measure the standard deviation in the  $\lambda_k$  at the lattice sites  $(0, 0)$ ,  $(1, 1)$ ,  $(2, 2)$ ,  $(3, 3)$ . Which of these has the largest standard deviation and why?

- (d) **Optional for the enthusiasts.** You should find that the concentrations  $\lambda_k$  from part (b) take on special values<sup>7</sup> whereby each can be written as a **hexadecimal** expansion  $h_1 h_2 . h_3 h_4 h_5 h_6$ , where the  $h_i$  are in the range  $0, 1, 2, \dots, 9, A, B, C, D, E, F$ .

---

<sup>7</sup>There may also be a small amount of numerical error, but this should be substantially smaller than the range of special values.

In the program files, you will find a key for converting each hexadecimal number into a  $2 \times 2$  pixel block. By considering each of the  $h_i$  in turn over the  $7 \times 7$  grid, construct six  $14 \times 14$  glyphs to reveal a hidden message. Using this, complete the quotation “Beautiful \_\_\_\_\_. Beautiful \_\_\_\_\_.”