

CS30 Spring 2014

Lab 6

Use the command `diary` to record your answers and submit them. Submit code for the scripts and functions you write. Submit any figures.

- (25 points) Linear systems. Consider the following 5 equations and 5 unknowns x_1, x_2, \dots, x_5 .

$$\begin{aligned}x_1 + 2x_2 + x_3 - x_4 &= 1 \\-7x_1 + 2x_2 + 6x_3 - x_5 &= 10 \\5x_1 - 2x_2 + 5x_3 - x_4 - 7x_5 &= -1 \\11x_1 + 3x_3 - x_4 + 3x_5 &= 2 \\9x_1 + 2x_2 - x_3 - x_4 + 2x_5 &= 9\end{aligned}$$

- Write down this linear system in the form $\mathbf{Ax} = \mathbf{b}$ identifying the matrix \mathbf{A} and vectors \mathbf{x} and \mathbf{b} .
- Write a function `SolveSystem` which takes as input the right-hand-side (rhs) vector \mathbf{b} and returns as output the solution vector \mathbf{x} . Use the Matlab matrix right divide operator, `\`.
- What is the solution for the rhs given above? What is the solution for the following rhs?

$$\mathbf{b} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}$$

- (45 points) We will explore *Laplacian smoothing* of a grayscale image with a simple example. Write your code in a script called `LaplacianSmoothing.m`.
 - Create a 9×9 array for the image tile named `tile`. The first three rows should be black (0), the middle three rows white (255), and the last three rows black. Use the Matlab functions `zeros` and `ones` to create the tile. Save an image of your tile (use `colormap('Gray')`) to set the colormap to gray.
 - Make a 10×2 tiling of your tile and save it in a variable named `stripes`. Use the Matlab function `repmat`. Save an image of the tiling.
 - Create a variable called `stripesVector` that is the `stripes` array flattened into a column vector.
 - What is the average value in `stripesVector`?

- (e) We will use a smoothing matrix to apply the Laplacian smoothing to the image. The smoothing matrix will multiply `stripesVector` resulting a vector of the same length. Given this information, what size should the smoothing matrix be?
- (f) Create the smoothing matrix, and name it `smoothingMatrix`. The matrix has -2 on the main diagonal, and 1 on the diagonals above and below the main diagonal. For example, the 5×5 smoothing matrix would be

$$\begin{pmatrix} -2 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & -2 \end{pmatrix}.$$

Use the Matlab function `eye` to construct the smoothing matrix.

- (g) Smoothing step. If \mathbf{A} is the smoothing matrix, and \mathbf{x} is the image vector, a single smoothing step is given mathematically as

$$\mathbf{x}^{smooth} = \mathbf{x} + \alpha \mathbf{A} \mathbf{x}$$

where $\alpha < 1$ is a smoothing parameter. Define `numSmoothingSteps` to be the number of smoothing steps. Write your smoothing step inside a `for` loop, i.e.,

```
for i=[1:numSmoothingSteps]
    % smoothing step goes here
end
```

- (h) After applying the smoothing step, reshape the resulting vector back into the 2D image for drawing. Use `numSmoothingSteps = 100` and $\alpha = 0.05$. Display the result. Save an image of your result.
- (i) What is the average value in `stripesVector` after the smoothing step?
- (j) Experiment with different values of $\alpha < 1$. What happens for larger α ? Smaller α ?
- (k) Experiment with different numbers of smoothing steps. What happens as you take more smoothing steps?

3. (30 points) Consider the following table of grade data.

Name	HW1	HW2	Exam 1	HW3	HW4	Exam 2
Kermit	21	51	76	6	16	69
Piggy	21	20	93	5	87	71
Animal	47	14	24	37	3	26
Fozzie	55	1	51	57	2	32
Gonzo	76	68	14	44	18	64

- (a) Create a 5×6 numeric array storing the grade data give.
- (b) Create a char array storing the names.
- (c) Sort the grade data by the score on Exam 1.
- (d) Reorder the names by the score on Exam 1, using array indexing, and indices returned in (c).
- (e) What are the average grades on the HW and Exams? What were the minimum and maximum grades? Use Matlab functions to compute the average, min and max.
- (f) Write code to sort the grade data by the score on HW1, and then the score on HW3 when HW1 scores are the same.