Assignment 2.

Due Thurs., Oct. 12, at 11:59 pm.

Reading: Lectures 3-5 in the text.

1. p. 24, Exercises 3.2. Note that an *eigenvalue* of A is any nonzero scalar λ such that $Ax = \lambda x$ for at least one nonzero vector x.
2. p. 30, Exercises 4.4
3. p. 37, Exercise 5.3.
4. **Flag compressions.** (This concept comes from an unpublished paper by Townsend and Strang.) We can represent simple geometric patterns on flags by using matrices where every entry in the matrix is either '1' or '0'. As an example, here are a few flags (I've made the flags square, which is a bit strange): A represents a monotone flag of a single color. B represents the basic cross pattern found in the Union Jack, C represents the subsquare + stripe pattern found on the American flag, D represents the 'X' pattern found on, for example, the Scottish flag, and E represents a diagonal stripe pattern, as found in the flag of Trinidad and Tobago.

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad E = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

A matrix $A \in \mathbb{C}^{m\times n}$ is said to be "low rank" if $A = XY^*$ for some $X \in \mathbb{C}^{m\times p}$ and $Y \in \mathbb{C}^{n\times p}$, and $mp+np < mn$. Some flag patterns are low rank, and others are not.

   (a) Use the MATLAB code `[u, s, v] = svd(A)` to compute the full SVD for each flag. Use the singular values of each flag to determine the rank of each flag, and create a table that shows the rank of each flag. Which ones are low rank? (Note:

you can use the code `sv = diag(s)` to extract the diagonal of the matrix s and save it as a vector sv. Also, you can use MATLAB to create a table directly and then paste it into your writeup, though it's fine to create your table other ways.) Attach to your solution sheet a printout of the code you used to compute the svds for your flags.

(b) For each rank-1 flag matrix, provide a low rank representation of the form ab^T, where a and b are not necessarily obviously related to the singular vectors of the matrix (i.e., these are vectors you can figure out with just your eyes and the definition of the outer product).

(c) Do the same for each rank-2 flag, writing out the matrix in the form $ab^T + cd^T$.

(d) Comment on the patterns of the flags that are highly compressible (low rank) vs. those that are full rank or do not admit low rank factorizations. (Hint: What is the rank of the identity matrix?)

(e) Use the command `surf(B); shading flat; view(3), colorbar, shg` to create a plot of the matrix $B \in \mathbb{C}^{m\times m}$ with entries $B_{jk} = 1/(x_j - y_k)$, where $x_j = (j-1)/m$ and $y_k = (k+1/2)/m$, with $m = 500$. Please include an image of the plot in your write up. Just by looking at it, do you think the matrix is low rank? Why or why not?

Notes on code snippets: `surf(B)` treats the entries of a matrix like points on a square surface. It colors them according to height. `shading flat` is just an asthetic choice that smooths the plot, `view(3)` sets the perspective so we have a vertical view, `colorbar` includes the colorbar in the legend, and `shg` is the "show graph" command: it makes matlab pull the graph window to the front of your screen.

5. **Low rank approximation. NOTE: Any plots you are asked to create must be included in your solution write up.** As in the last example, let A be an $m \times m$ matrix defined to have entries $A_{jk} = 1/(x_j - y_k)$, where $x = [x_1, x_2, \ldots, x_m]^T$ and $y = [y_1, y_2, \ldots, y_m]^T$. Such a matrix is called a Cauchy matrix. Instead of interlacing the entries of x and y like we did in the last problem, we let $x_j = (j-1)/2m$ and that $y_k = x_m + (k+1/2)/m$, with $m = 500$. In infinite precision, A is a full rank matrix.

   (a) Use MATLAB to find the singular value decomposition $USV^* = A$. Using the command `s = diag(S)`, you can extract from S its diagonal s, which is a vector of the singular values of A. Plot the magnitude of the singular values on a logarithmic scale against the indices of the singular values. To do this, you can use the command `semilogy(idx, s)`, where `idx` is a vector containing the indices. What do you notice about these singular values?

   (b) We will use the singular value decomposition to build low rank approximations to A. Let $A^{(j)}$ denote a rank j approximation to A, where

$$A^{(j)} = U(:, 1\!:\!j)S(1\!:\!j, 1\!:\!j)[V(:, 1\!:\!j)]^*.$$

   Create a loop that constructs $A^{(j)}$ and computes the error $E(j) = \|A - A^{(j)}\|_2$ for $j = 1, 2, \ldots, 10$. You should end up with a vector E of length 10. Plot E on a semilog scale along with the the second through eleventh singular values of A. You can choose how to make this plot visually compelling (be sure to plot the error and singular values as markers of some sort without a continuous line connecting them). What do you notice? *Hint: Remember the Eckart-Mirsky-Young Theorem (Theorem 5.8 in your book)*

(c) Plot the singular values of the matrix B from Q4(e) on a semilog scale. Do you think that a rank 10 matrix exists that effectively approximates this matrix? Justify your reasoning and relate your justification to the Eckart-Mirsky-Young Theorem. Include in your argument a lower bound on the 2-norm error for any rank-10 approximation to B, as well as a lower bound on the rank-10 error in the Frobenius norm.