Math 3607: Homework 5

Due: 10:00PM, Wednesday, July 14, 2021

TOTAL: 30 points

You will be writing some MATLAB functions for this assignment. Include all your functions at the end of your live script.

1. (Understanding matrix multiplication) Do **LM** 12.5–3.

2. (Gram-Schmidt in MATLAB) Do **LM** 12.6–2.

3. (Adapted from **FNC** 3.3.3.) Let $x_1, x_2, \ldots, x_m$ be m equally spaced points in $[-1, 1]$. Let V be the Vandermonde-type matrix appearing on p. 12 of Module 3 lecture slides for $m = 400$ and $n = 5$. Find the thin QR factorization of $V = \widehat{Q}\widehat{R}$, and, on a single graph, plot every column of $\widehat{Q}$ as a function of the vector $\mathbf{x} = (x_1, x_2, \ldots, x_m)^{\mathrm{T}}$. (Use MATLAB to solve this problem.)

4. (Visualizing matrix norms; adapted from **LM** 9.4–26.) For $p \in [1, \infty]$, recall the definition of the matrix p-norm,
$$\|A\|_p = \max_{\|\mathbf{x}\|_p = 1} \|A\mathbf{x}\|_p . \tag{1}$$
To understand this definition, we will work in two-dimensional space so that we can easily plot the results. For this problem, use
$$A = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}. \tag{2}$$
As an illustration, we study the case $p = 2$ following the steps below.

   - Create unit vectors $\mathbf{x}_j$ in 2-norm,
   $$\mathbf{x}_j = \begin{bmatrix} \cos\theta_j \\ \sin\theta_j \end{bmatrix}, \quad 1 \leqslant j \leqslant 361 \tag{3}$$
   using 361 evenly distributed θ_j in $[0, 2\pi]$. Make sure $\mathbf{x}_1 = \mathbf{x}_{361} = (1, 0)^{\mathrm{T}}$, just as in the spiral polygon problem. Plot these points, which lie on the unit circle. Make sure the plot looks like a circle.
   - For each j, let $\mathbf{y}_j = A\mathbf{x}_j$. Plot all points $\mathbf{y}_j$. In addition, store $\|\mathbf{y}_j\|_2$ for all j in a vector.
   - Plot $\|\mathbf{y}_j\|_2$ as a function of θ_j.
   - Find the maximum value of $\|\mathbf{y}_j\|_2$ over all j. This estimates $\|A\|_2$. Compare this against the actual value computed by `norm(A,2)`.

   These steps are carried out by the following script.

```
A = [2 1; 1 3];
theta = linspace(0, 2*pi, 361);
X = [cos(theta); sin(theta)];  % x: unit vectors in 2-norm
Y = A*X;                       % y: images of x under A
norm_Y = sqrt(sum(Y.^2, 1));   % norm of vectors y

% visualization
clf
subplot(2,2,1)
plot(X(1,:), X(2,:)), axis equal
title('x: unit vectors in 2-norm')

subplot(2,2,2)
plot(Y(1,:), Y(2,:)), axis equal
title('Ax: image of unit vectors under A')

subplot(2,1,2)
plot(theta, norm_Y), axis tight
xlabel('\theta') ylabel('||y||')

% matrix norm approximation (and comparison)
fprintf(' p = 2\n')
fprintf(' approx. norm: %18.16f\n', max(norm_Y))
fprintf(' actual  norm: %18.16f\n', norm(A, 2))
```

which generates Figure 1 and the following outputs in the Command Window:

```
p = 2
approx. norm: 3.6179964204609893
actual  norm: 3.6180339887498953
```

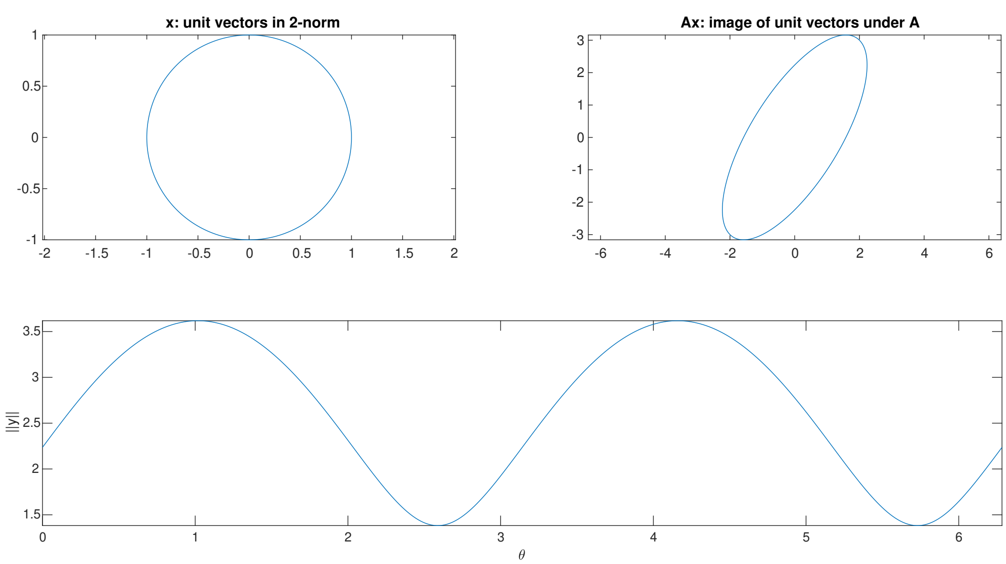

Figure 1: Plots illustrating the definition of matrix norm.

Modify and develop the script into a MATLAB function `visMatrixNorm` which takes two inputs

- A, a 2×2 matrix and
- p, a number which can be either 1, 2, or ∞,

and carries out the same tasks as above, namely,

- approximating $\|A\|_p$ using (1) and
- producing a figure such as Figure 1.

Be sure to print out the value of p, the approximate norm, and the norm computed using MATLAB's `norm` function. Then run the function with `visMatrixNorm(A, 1)` and `visMatrixNorm(A, Inf)`, where A is as defined in (2).