Math 3607: Homework 6

Wednesday, March 2, 2022

TOTAL: 30 points

- Problems marked with ✏ are to be done by hand; those marked with 💻 are to be solved using a computer.
- **Important note.** Do not use *Symbolic Math Toolbox*. Any work done using `sym` or `syms` will receive NO credit.
- **Another important note.** When asked write a MATLAB function, write one at the end of your live script.

1. (FLOP counting; **FNC** 2.5.5) ✏ This problem is about evaluation of a polynomial

$$p(x) = c_1 + c_2 x + c_3 x^2 + \cdots + c_n x^{n-1}.$$

(a) Here is a little code to do the evaluation.

```
y = c(1);
xpow = 1;
for i = 2:n
    xpow = xpow * x;
    y = y + c(i)*xpow;
end
```

Assuming that `x` is a scalar, how many flops does this code take, as a function of n? Provide both the exact answer and the asymptotic answer as $n \to \infty$.

(b) Here is another code to do the same task.

```
y = c(n);      % This algorithm is called Horner's rule.
for j = n-1:-1:1
    y = y*x + c(j);
end
```

Assuming that `x` is a scalar, how many flops does this code take, as a function of n? Provide both the exact answer and the asymptotic answer as $n \to \infty$. Then compare the count to the one from (a).

2. (Matrix norms) ✏ Let

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}.$$

Calculate $\|A\|_1$, $\|A\|_2$, $\|A\|_\infty$, and $\|A\|_F$ all by hand.

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

4. (Periodic fit; **FNC** 3.1.3) In this problem you are trying to find an approximation to the periodic function $f(t) = e^{\sin(t-1)}$ over one period, $0 \leqslant t \leqslant 2\pi$. In MATLAB, let `t=linspace(0,2*pi,200)'` and let `b` be a column vector of evaluations of f at those points.

   (a) Find the coefficients of the least square fit
   $$f(t) \approx c_1 + c_2 t + \cdots + c_7 t^6.$$

   (b) Find the coefficients of the least squares fit
   $$f(t) \approx d_1 + d_2 \cos(t) + d_3 \sin(t) + d_4 \cos(2t) + d_5 \sin(2t).$$

   (c) Plot the original function $f(t)$ and the two approximations from (a) and (b) together on a well-labeled graph.

5. (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
$$A = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}. \tag{2}$$

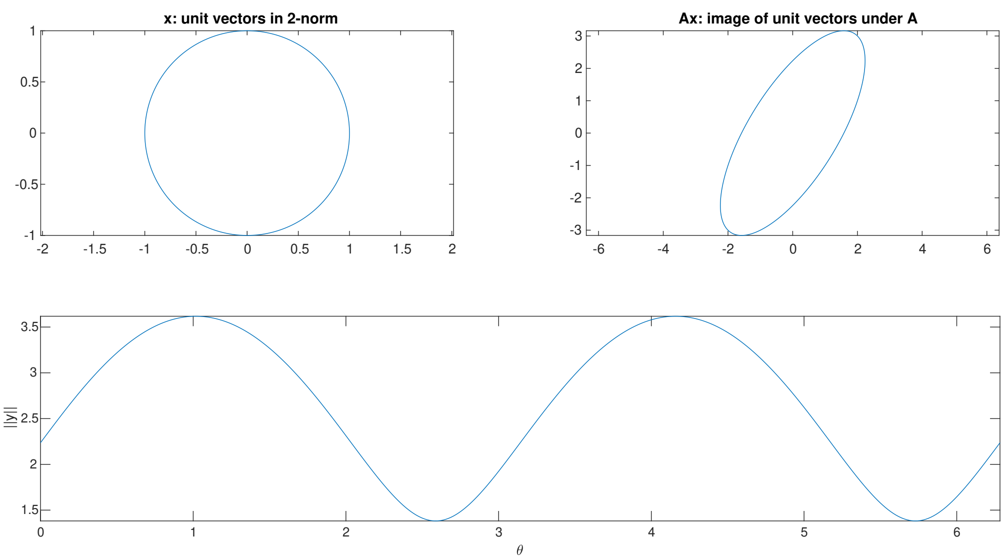

Figure 1: Plots illustrating the definition of matrix norm.

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
```

(a) 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.

(b) Then run the function with `visMatrixNorm(A, 1)` and `visMatrixNorm(A, Inf)`, where A is as defined in (2).