CAAM 453: Numerical Analysis I

Problem Set 2
Due: Monday, October 9th, 2017

Note: Turn in all MATLAB *code that you have written and turn in all output generated by your* MATLAB *functions/scripts.* MATLAB *functions/scripts must be commented, output must be formatted nicely, and plots must be labeled.*

Problem 1 (10 points)

Let $\mathbf{x} \in \mathbb{R}^n$. Recall for each p $(1 \leq p \leq \infty)$ the *p-norms* for the vector x, which are defined as

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}, \qquad p < \infty,$$
$$\|\mathbf{x}\|_\infty = \max_{i=1,\ldots,n} |x_i| .$$

Each norm has its own "unit sphere," the collection of points which are distance 1 from the origin, with respect to that norm. Draw the unit sphere in $\mathbb{R}^2$ using MATLAB for $p = 1, 1.5, 2, 5, \infty$, all on one plot.

Problem 2 (35 points)

Before proceeding to the problems, let's make some definitions.

Definition. Suppose $\|\mathbf{x}\|_a$ and $\|\mathbf{x}\|_b$ are two different vector norms acting on vectors $\mathbf{x} \in \mathbb{R}^k$. They are *equivalent* if there are two constants $c, C > 0$ such that for every $\mathbf{x}$,

$$c \|\mathbf{x}\|_b \leq \|\mathbf{x}\|_a \leq C \|\mathbf{x}\|_b . \qquad (*)$$

In other words, two equivalent norms $\|\mathbf{x}\|_a$ and $\|\mathbf{x}\|_b$ are not independent; they are always within a constant factor of each other. Of course, C and $1/c$ may be large.

Two matrix norms are likewise called equivalent if they satisfy an inequality like $(*)$ for some c, C.

Fun fact: any two vector or matrix norms in finite-dimensional spaces are always equivalent. However, you don't have to prove this.

(a) Prove that the 2-norm and ∞-norm on $\mathbb{R}^k$ are equivalent. Specifically, prove that for every k and $\mathbf{x} \in \mathbb{R}^k$,

$$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \sqrt{k} \|\mathbf{x}\|_\infty .$$

(b) The bound in part (a) is sharp; it cannot be improved. Find all vectors where $\|\mathbf{x}\|_\infty = \|\mathbf{x}\|_2$ (and prove you found them all). Similarly, find all vectors where $\|\mathbf{x}\|_2 = \sqrt{k}\,\|\mathbf{x}\|_\infty$.

(c) Prove that if two vector norms are equivalent, their induced matrix norms are equivalent.

(d) From parts (a) and (c), it follows that the 2-matrix norm and ∞-matrix norm are equivalent. In fact, if $A \in \mathbb{R}^{m\times n}$,

$$\frac{1}{\sqrt{n}}\,\|A\|_\infty \le \|A\|_2 \le \sqrt{m}\,\|A\|_\infty\,.$$

Find two nonzero matrices (with general m and n) for which each inequality above is an equality.

(e) Prove that $\|A\|_1 = \left\|A^\top\right\|_\infty$ for any matrix $A \in \mathbb{R}^{m\times n}$.

Problem 3 (55 points) *(This problem is pledged.)*

The goal of this problem is for you to write your own code to invert an $n \times n$ matrix A using the QR factorization and Householder reflections. Your code should include the following functions:

- `[Q,R] = QRHouseholder(A)`. Input: real or complex $n\times n$ matrix `A`. Output: $n\times n$ unitary `Q` and $n\times n$ upper triangular `R` such that `QR` = `A`. Use Householder reflections to calculate `Q` and `R`. Remember to allow for complex inputs.

- `Z = InvertUpperTriangular(R)`. Input: upper triangular $n \times n$ matrix `R`. Output: $n \times n$ matrix `Z` = `R`$^{-1}$, or an error if `R` is singular.

- `Y = Invert(A)`. Input: real or complex $n\times n$ matrix `A`. Output: $n\times n$ matrix `Y` = `A`$^{-1}$, or an error if `A` is singular. Use `QRHouseholder` and `InvertUpperTriangular` to invert `A`.

You can use all scalar and vector operations, and matrix multiplication. Of course, you cannot use any of Matlab's commands for solving linear equations.

Test your code by creating several random large matrices; for each matrix A_i you create, calculate the matrix norm $\|A_i \cdot \texttt{Invert}(A_i) - I\|_2$, where I is the identity matrix. You can use MATLAB's `norm` command to calculate the 2-matrix norm.

Compute the total operation count, to leading order, required by your algorithm for computing A^{-1} in terms of n. In other words, your operation count should be a polynomial in the variable n; you should find the highest order term in the polynomial, including the coefficient. Operations include $+$, $-$, $*$, $/$, and powers.