MATLAB Assignment 4

Spring 2019, Section B

This problem set has become a bit of a staple of the MATLAB seminar- I hope you enjoy it! Note that at a minimum, all plots should have a title, x-axis and y-axis labels, and if there is more than one function in the same figure, a legend as well. Additionally, make sure your plot's axis bounds are adequate.

Please submit this homework as *.m* files with suppressed output (obviously, the plots will still be displayed). Remember that all lectures and homeworks may be found at *github.com/guybaryosef/ECE210-materials*. This homework is due at 11:59PM on February 27th to guybymatlab@gmail.com.

1. Who Gives a Schmidt!? Professor Mintchev has just assigned you 20 tedious Gram Schmidt orthonormalization problems! Luckily, you are a master of MATLAB so you decide to build a function which can handle them all for you.

a. Create a function called *gramSchmidt*. The input to the function should be a 2-D array, where each column is a vector. The set of input vectors can be assumed to be linearly independent. Implement GS to create an orthonormal set of vectors from these input vectors. Store them as columns in an output matrix in the same format as the input 2-D array. Feel free to use the ***norm*** function as needed.

b. After you've created this function, you'd like a way to test if it works. Create another function called *isOrthonormal* which takes in a 2-D array as input. This function should return a logical 1 if all the columns are orthonormal and logical 0 otherwise. Be careful with this - direct floating point equality comparison is a bad idea. Instead apply a threshold to the difference of the two numbers like so: if $|x - \hat{x}| > \epsilon$ then... The ***eps*** function might be useful here. You can add a nice big fudge factor to make the tolerance big enough that it works, just don't make it huge (Note that there is also the matter of spanning the same space as the original matrix, don't worry about this condition).

c. Finally, we would like to estimate another vector as a linear combination of these orthonormal vectors (i.e. to project a vector onto the space of the orthonormal vectors). Implement a function called *orthoProj* which takes as input a vector to be estimated as well as a 2-D array of orthonormal vectors (similar format as previously) and returns as its output the estimated projected vector.

d. Test all of the above functions on some random complex vectors (use ***rand*** to make a random vector). First test the case where there are more elements in each vector than the number of vectors. Then test the case where the number of vectors is equal to the number of elements in each vector. Compare the errors.

e. Uniformly sample *sin(x)* over $[0, 2\pi]$. Generate 5 Gaussian graphs using the pdf:

$$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x-\mu)^2}{\sigma^2}\right)$$

Give each Gaussian a standard deviation of 1 ($\sigma = 1$) and pick the mean from a linearly spaced vector ranging from 0 to 2π ($\mu \in \{0, \pi/2, \pi, 3\pi/2, 2\pi\}$). Consider using ***ndgrid*** for compact code.

(a) Plot the Sinusoid and Gaussians on the same plot.

(b) Use *gramSchmidt* to create an orthonormal set of vectors from the generated Gaussians. Use *orthoProj* to estimate the sinusoid from that set of vectors.

(c) Create a 2x1 subplot. Plot the sinusoid and the estimated sinusoid together on the upper plot. Plot the orthonormal basis functions on the lower plot. Remember to properly format all your plots!