

# ECE-210-A HW4

Instructor: Jonathan Lam

Spring 2022

1. Professor Mintchev has just assigned you 20 tedious Gram Schmidt orthonormalization problems! Luckily, you are a MATLAB master so you decide to build a function to handle all of them for you in short time.
  - (a) Create a function called `gram_schmidt`. The input to the function should be a 2D array, each column of which is a vector in the original linearly independent set of vectors. Implement GS to create an orthonormal set of vectors from these. Store them as columns in an output matrix, similar to the input format. Feel free to use the `norm` function if needed.
  - (b) After you've created this function, you'd like a way to test if this works. Create another function called `is_orthonormal` which has a single 2D array as the input. The function should return 1 (or true) if all columns are orthonormal and 0 (or false) 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.
  - (c) Finally, we would like to estimate another vector as a linear combination of these orthonormal vectors. (Project the vector on to the space of the orthonormal vectors.) Implement a function called `ortho_proj` which takes a (column) vector to be estimated and an array of orthonormal columns as arguments and outputs the estimated 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 a vector. Compare the errors. (You can use the Euclidean distance as a measure of error.)

- (e) Uniformly sample  $\sin(x)$  on  $[0, 2\pi]$  with 1000 points. Also generate five Gaussians by sampling

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

over the same  $[0, 2\pi]$  interval, with  $\sigma = 1$  and  $\mu \in \{0, \pi/2, \pi, 3\pi/2, 2\pi\}$ . Consider using `ndgrid` or broadcasting for compact code. Plot the sinusoid and Gaussians on the same plot. Give axis labels and a title. Use `gram_schmidt` to create an orthonormal set of vectors from the Gaussians. Use `ortho_proj` to estimate the sinusoid from that set of vectors.

Create a  $2 \times 1$  subplot (using `subplot` or `tilde layout`). Plot the sinusoid and the estimated sinusoid together on the upper plot. Plot the orthonormal basis functions on the lower plot. Give all plots proper labels and titles.

- (f) (*Optional*) For the first part, you could assume that the input vectors were a linearly independent set. This simplifies the code a little bit. How would you deal with a linearly dependent set? Implement it and test on a linearly dependent set.
- (g) (*Optional*) If you haven't already done so already, you might find it neater to use the `ortho_proj` function in your implementation of `gram_schmidt`.