ECE210A – Signals and Systems

Assignment 04: Getting Funcy

1. Make an anonymous dot function that returns the dot product of two input column vectors. This function should work regardless of the size of the input vectors.

   Note: You could multiply corresponding elements and use sum(), but there is a much simpler way of obtaining your dot product. Think matrix multiplication. Also, we will be using this function on complex data, so there should be conjugation (or a transpose) somewhere.

2. Use this function to make another function is_orthonormal that takes a two-dimensional matrix as an input and checks if the columns of this matrix are orthonormal vectors, returning a logical 1 if the columns are orthonormal and a logical 0 otherwise.

   Hint: As soon as you find a column that has a norm not equal to 1 or two columns that are not orthogonal, you immediately know your function should return 0. You can make sure it does with the following two lines:

   ```
   * = false;
   return
   ```

   where * is the name you assign to the output of your function. Here, `return` causes your function to return the values assigned to your output variables, skipping the rest of the function.

   Also, due to numerical instability, your norms may be very close to 1, but not exactly 1. This will cause your function to yield false, even if it should yield true. To solve this issue, instead of checking if your norms are identical to 1, use `eps`, which is how MATLAB defines the smallest positive value it can store (2^{-52}).
   (i.e. check if `abs(dot(*, *) - 1) <= 1000 * eps;` or something). The same applies for checking if two vectors are orthogonal.

3. Make a gram_schmidt function that takes a two-dimensional matrix as an input, uses your is_orthonormal function to check if the columns of this two-dimensional matrix are orthonormal, and then uses your dot function to perform Gram-Schmidt on the columns of the input matrix if they are found not to be orthonormal.

4. Generate a matrix using `randi(15, 4, 4) + 1j * randi(15, 4, 4);` and run your gram_schmidt function on it, displaying the output. Then run your is_orthonormal function on your output and display the result.