# Max Kramer

I affirm that I have adhered to the honor code on this assingment.

*Hello again, scientist! I'll do all my writing in italics, and problems for you will be in* **bold.** *Comment your code, and* explain your ideas in plaintext. *As a general rule, I expect you to do at least as much writing as I do. Code should be part of your solution, but I expect variables to be clear and explanation to involve complete sentences. Cite your sources; if you work with someone in the class on a problem, that's an extremely important source. Don't work alone.*

# Problem 12.2.

This problem is pulled from the exercises in Lay 6.2.

```
v1 = [-6 -1 3 6 2 -3 -2 1]';
v2 = [-3 2 6 -3 -1 6 -1 2]';
v3 = [6 1 3 6 2 3 2 1]';
v4 = [1 -6 -2 -1 3 2 -3 6]';

A = [v1 v2 v3 v4]
```

```
A =

    -6    -3     6     1
    -1     2     1    -6
     3     6     3    -2
     6    -3     6    -1
     2    -1     2     3
    -3     6     3     2
    -2    -1     2    -3
     1     2     1     6
```

***Show that the columns of A are orthogonal.*** *You can do this in one line of code, but explain the output.*

```
proof = A' * A
```

```
proof =

   100     0     0     0
     0   100     0     0
     0     0   100     0
     0     0     0   100
```

the matrix proof is ther result of multiplying A' * A. Theorem 6 in Lay 6.2 shows that a m x n matrix has orthonormal columns iff A'*A is equal to the identity matrix. While the vectors in A are not unit length, the result of A' * A on the upper and lower triangulars are the dot products between the vectors. The fact that both triangulars are all zeros demonstrates that the columns of proof are orthogonal, but not orthonormal.

*Now let's follow Example 2 in Lay 6.2. Note that the following computation* **does not work** *unless the columns of A are orthogonal.*

```
v = [-65;-59;13;77;49;-37;-45;51];
a = A'*v
```

```
a =

        1300
        -200
           0
         700
```

***Write v as a linear combination of the columns of A. Check your answer.*** *The benefit of doing it this way, as opposed to using rref(), is that it's extremely fast and numerically stable. "Numerically stable" means that you won't have creeping errors from repeated multiplication and subtraction. Orthogonality, when you've got it, is fantastic.*

```
vec = 1300 * v1 - 200 * v2 + 700 * v4
vec = vec/100
```

```
vec =

       -6500
       -5900
        1300
        7700
        4900
       -3700
       -4500
        5100


vec =

   -65
   -59
    13
    77
    49
   -37
   -45
    51
```

Multiplying A' by v results in a projection of v onto the space spanned by the columns of A. This projection vector contains the coefficients on the vectors that express v as a linear combination of the columns of A.

***Using the same logic, try to write w as a linear combination of the columns of A. Check your answer.*** *What happened?!* ***Explain.***

```
w = [100;100;100;100;100;100;100;100];
```

```
rref(horzcat(A,w))
```

```
ans =

     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
```

When row reducing the system, we find that the last column is all zeros. The system demonstrates that it is not possible to write w as a linear combination of the columns of A.

*Published with MATLAB® R2019b*