Max Kramer

I affirm that I have adhered to the honor code on this assignment I Acknowledge working with Sara Aragaki on this assignment. I also recieved help from Isaac on the first problem

*Hello again, scientist! I'll write in italics, and problems for you will always be in **bold**. 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.*

Problem 3.1.

Once again, let's import HM.csv as the matrix HM.

```
HM = csvread('HM.csv');
size(HM) % make sure this returns [100 100] before continuing!
```

```
ans =

   100   100
```

*The columns of HM are linearly dependent. That means that there is a nonzero vector v for which HM*v = 0. Using your knowledge of parametric vector form, and rref(), **find such a vector**. There is a MATLAB command that can do this directly; do not use it. Please, please use semicolons.*

```
A = rref(HM);
HM_homog = horzcat(HM,zeros(100,1));
sol = rref(HM_homog);
v = sol(:,100);
v = -v;
v(100) = 1;
final = rref(HM)*v;
```

The last two variables v_99 and v_100 in HM are free, which means the equation HM*v=0 has at least one nontrivial solution. By creating a homogenous linear system, we can find a nontrivial solution to the equation HM*v=0. By choosing a value (1) for X_100 and a value (0) for X_99, we can find a vector v that represents a nontrivial solution to the system HM*v=0.

***Prove to me that rref(HM)*v = 0 without just showing me a vector with 100 zeros.** There are a lot of ways to do this in MATLAB; find one!*

```
B = rref(HM) * v;
bool_ans = isequal(B,zeros(100,1))
```

```
bool_ans =

  logical

   1
```

The isequal() function does a boolean comparison on each element of two vectors to see if they are the same or not, returning 1 if the vectors are the same. The comparison above returns 1.

*The computation of row-echelon form is what's called "numerically unstable," which means that small numerical errors in the input can lead to huge errors in the output. Because row-equivalent systems have the same solution set, we should get that HM*v = 0.* ***Compute HM*v, and prove that HM*v is NOT zero without just showing me a vector with a hundred nonzero entries.***

```
C = HM*v;
bool_unstable = isequal(C,zeros(100,1))
```

```
bool_unstable =

  logical

   0
```

The above computation calculates a vector C by multiplying the non row reduced matrix HM against vector v. When compared to a vector of 100 zeros, the boolean value returned is False (0).

This is fixable, but at a significant cost. Once you've done the previous parts, ***uncomment this line of code. Then repeat the above computations, and then prove to me that this time HM*v = 0.*** *What MATLAB's doing here is treating HM as a symbolic matrix, meaning it's doing all computations exactly. Don't panic if your code takes a minute to run; even at 100x100, you're making MATLAB do roughly a million exact computations.*

```
HM = sym(HM);
A = rref(HM);
HM_homog = horzcat(HM,zeros(100,1));
sol = rref(HM_homog);
v = sol(:,100);
v = -v;
v(100) = 1;
final = rref(HM)*v;
B = rref(HM) * v;
bool_ans = isequal(B,zeros(100,1))
C = HM*v;
bool_unstable = isequal(C,zeros(100,1))
```

```
bool_ans =

  logical

   0


bool_unstable =

  logical

   0
```

Published with MATLAB® R2019b