Max Kramer

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

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 5.2.

The exact calculation of determinants is a numerical nightmare. The cofactor expansion algorithm for an n-by-n matrix requires n! multiplications and there's not really any way to cheat. Worse, every time you multiply numbers you propagate error. Consider the following.

```
A = [3301 4423; 2133 2858];
```

What is the exact value of det(A)? det(A^2)? det(A^3)? det(A^n)? Explain. *Don't use det() to find these values, you're about to be disappointed by it.*

```
detA = A(1,1)*A(2,2) - A(1,2)*A(2,1)
detAsq = detA^2
detAcu = detA^3
```

```
detA =

    -1


detAsq =

     1


detAcu =

    -1
```

Using the definition of the determinant of a 2x2 matrix, the determinant detA is calculated. Given that if one row of A is multiplied by k to produce B, then det(B) = k *det A, the determinants of A^2 and A^3 are then manually calculated by squaring the determinant of A.

Go ahead, ask MATLAB what det(A^3) is. ***Uncomment this.***

```
det(A^3)
```

```
ans =

   6.7872e+03
```

Wow that is really terrible. That is not even sort of close. Let's force MATLAB to do it exactly by using symbolic matrices. ***Uncomment this.***

```
B = sym(A)
det(B^3)
```

```
B =

[ 3301, 4423]
[ 2133, 2858]


ans =

-1
```

It's 2020, right? We have powerful computers. Why worry about decimal precision and stuff when you can just do everything symbolically? Let's find out.

```
syms a s d f g h j k l;
C = [a s d; f g h; j k l]
```

```
C =

[ a, s, d]
[ f, g, h]
[ j, k, l]
```

Once you're ready, uncomment this. Explain the output. *Note that it may take much longer to run the first time you publish this than any time afterwards, because MATLAB keeps calculations in memory. Make a note of how long it takes the first time you run the code.*

```
tic
simplify(det(C^10)-det(C)^10)
toc
```

```
ans =

0

Elapsed time is 0.585602 seconds.
```

The code took approximately 107.45 seconds to run. The code simplifies the algebraic expression that results from subtracting det(C)^10 from the determinant of C^10.

Your computer is probably devoting about a teraflop = 10^12 calculations to MATLAB each second. ***Roughly how many calculations did you just force MATLAB to do?***

Given that it took approximately 107.45 seconds to run and assuming a teraflop per second computation rate, my laptop performed approximately 107.45(10e12) = 1.0745e14 computations overall.

Each time you multiply by C, you force MATLAB to do another column's worth of work, which increases the amount of stuff to do by a factor of about 3. ***Don't uncomment the following code.***

```
% tic
% simplify(det(C^20)-det(C)^20)
% toc
```

If you were bad at reading directions and did uncomment the code, roughly ***how many days would the calculation take?***

Assuming the code above would multiply the above computations by a factor of 3 for each additional multiplication of C, the overall time required to complete would be 3^10(107.45) = 6344815.05 seconds = 73.435 days.

Published with MATLAB® R2019b