Matlab in Math 461, part four

The rank of a matrix

If A is a matrix, the matlab command `rank(A)` computes an estimate of the rank of A. (I say an estimate, because roundoff errors can make it difficult to compute the rank. If an entry is, say, 10^{-16} is it a pivot or should it really be zero?) Matlab uses a sophisticated method (the singular value decomposition in sec 7.4) to estimate the rank, attempting to account for roundoff error. For example, the matlab command

```
>> rank([1 0; 0 .000000000000000001] )
```

will return 1 for the rank since it figures your matrix is close enough to the rank 1 matrix [1 0;0 0]. If for some reason you want to defeat this sophistication, you can put in a second argument of 0 (to indicate 0 tolerance for deviation). For example

```
>> rank([1 0; 0 .000000000000000001] ,0)
```

will return 2. Don't defeat this unless you really know what you are doing.

As you might expect, if you generate a random $m \times n$ matrix, it would always have rank as big as possible, the minimum of m and n. Try this by typing in `rank(rand(4,6))`, `rank(rand(7,5))`, etc. But there is a way of generating random matrices with smaller rank. Suppose, for example, you wish to generate a random 4×7 matrix with rank 2. Then as we will see below, `A = rand(4,2)*rand(2,7);` will give you such a matrix. Try it, and check that the rank is in fact 2. This works because of the following facts:

Fact 1. *Suppose A is an $m \times n$ matrix with rank k. Then we may write $A = BC$ where B is an $m \times k$ matrix and C is a $k \times n$ matrix.*

Fact 2. *Suppose $A = BC$ where B is an $m \times k$ matrix of rank k and C is a $k \times n$ matrix of rank k. Then A has rank k.*

To see Fact 1, let $\mathcal{B} = \{b_1, b_2, \ldots, b_k\}$ be a basis for the column space of A. We let $B = P_{\mathcal{B}}$ and if a_i is the i-th column of A, we let the i-th column of C be $[a_i]_{\mathcal{B}}$, the $\mathcal{B}$ coordinates of a_i. Then the i-th column of BC is $P_{\mathcal{B}}[a_i]_{\mathcal{B}} = a_i$, so $A = BC$.

To see Fact 2, we will show that $\mathrm{Nul}(BC) = \mathrm{Nul}(C)$. Then

$$\mathrm{rank}(BC) = n - \dim(\mathrm{Nul}(BC)) = n - \dim(\mathrm{Nul}(C)) = \mathrm{rank}(C) = k$$

So let us see why $\mathrm{Nul}(BC) = \mathrm{Nul}(C)$. Pick any v in $\mathrm{Nul}(C)$. Then $Cv = 0$, so $BCv = 0$ so v is in $\mathrm{Nul}(BC)$. Conversely, pick any w in $\mathrm{Nul}(BC)$. Then $BCw = 0$ which means that Cw is in $\mathrm{Nul}(B)$. But $\dim(\mathrm{Nul}(B)) = k - \mathrm{rank}(B) = 0$ which means that $\mathrm{Nul}(B) = \{0\}$ is just the zero vector. Consequently, $Cw = 0$ which means w is in $\mathrm{Nul}(C)$. So we have shown $\mathrm{Nul}(BC) = \mathrm{Nul}(C)$, every vector in one is in the other.

The Null space

While you can, with some effort, read off a basis for the Null space of a matrix from its reduced echelon form, matlab can also find a basis for you. The matlab command `null(A)` will produce a matrix whose columns are a basis for the Null space of A. It will not be the one you get from rref, it will be an orthonormal basis (we'll learn about these later in chapter 6). It will also attempt to account for roundoff error, so the vectors it produces may not evaluate exactly to zero, but their images under A will be extremely small. So, for example, `null([1 0; 0 .000000000000000001])` will give you the vector e_2 as a basis for the Null space since its image, although nonzero, is extremely small.

Matlab project 4 due Tuesday, March 30

Problem 1: Generate random 6×8, 7×5, and 9×9 matrices (without printing them out!) and check that their ranks are as expected.

Problem 2: Generate a random 6×8 matrix with rank 3. Check that its rank is 3. Ask matlab for a basis for its null space. Check that the answer matlab gives you is in fact a basis for the null space. Some things to check: Are all vectors in the null space? Are they linearly independent? Do they span the null space? You may need to use some theorems to see how to check these.

Problem 3: page 271 problem 36 in the text.