Sage and Linear Algebra Worksheet

FCLA Section NM

Robert Beezer
Department of Mathematics and Computer Science
University of Puget Sound

Spring 2020

First, a guaranteed nonsingular 5×5 matrix, created at random.

```
A = random_matrix(QQ, 5, algorithm="unimodular",
    upper_bound=20)
A
```

Demonstration 1 Augment with the zero vector, using the matrix method `.augment()` and the vector constructor `zero_vector(QQ, 5)`. Then row-reduce to use Definition NM. Or instead, do not augment and apply Theorem NMRRI.

Demonstration 2 Build some random vectors with `random_vector(QQ, 5)`, augment the matrix and row-reduce. There will always be a unique solution to the linear system represented by the augmented matrix. This is Theorem NMUS.

Instead—cheap, easy and powerful:

```
A.is_singular()
```

Now, a carefully crafted singular matrix.

```
B = matrix(QQ, [[ 7, -1, -12, 21, -8],
                [-3,  3,   0, -9,  6],
                [ 3,  3, -12,  9,  0],
                [ 2,  3, -10,  6,  1],
                [-2,  2,   0, -6,  4]])
B
```

Demonstration 3 Augment with the zero vector and row-reduce (Definition NM), or don't augment and row-reduce (Theorem NMRRI).]

Demonstration 4 A random vector of constants will only rarely build a consistent system when paired with `B`. Try it. But this is not a theorem, see the vector `c` below.

Instead—cheap, easy and powerful:

```
B.is_singular()
```

Two carefully crafted vectors for linear systems with `B` as coefficient matrix.

```
c = vector(QQ, [17,-15,-3,-5,-10])
d = vector(QQ, [-3,1,-2,1,2])
```

Demonstration 5 Which of these two column vectors will create a consistent system for this singular coefficient matrix? (Stay tuned.)

A null space is called a **right kernel** in Sage. It's description contains a lot of things we do not understand yet.

```
NS = B.right_kernel()
NS
```

Demonstration 6 But we can test membership in the null space, which is the most basic property of a set. Try `u in NS` and then repeat with `v`.

```
u = vector(QQ, [0,0,3,4,6])
```

```
v = vector(QQ, [1,0,0,5,-2])
```