

mat220rowoperations

Welcome to the MAT220 Introduction to Sage worksheet. This tutorial will guide you through the basic operations necessary to use Sage to handle basic matrix operations. Sage is a powerful tool; we'll only just begin to see its capabilities here. If you are interested you can learn much more about its capabilities at <http://www.sagemath.org>.

We'll begin by learning how to interact with the worksheet interface. It's quite straightforward with the exception that to evaluate a command you must either press "shift-enter" or click the "evaluate" link below the command entry cell. Pressing "enter" alone will merely create a new line in the command entry cell.

Click on the cell below, hold the shift key down, and press the enter key.

```
a=1
```

You may be surprised that nothing appeared to happen. In fact, the value 1 was assigned to the variable a, but this result was not echoed. Repeat on the cell below; click on the cell, then use "shift-enter" or click on "evaluate".

```
b=2; b
```

2

This time the value 2 was assigned to b and then the value of b was echoed. The semicolon is used to separate commands appearing on the same line. The same thing could be done by typing "b=2", pressing enter, typing "b" and then using "shift-enter".

```
b=2
b
```

2

Feel free to use whichever of these approaches is most comfortable to you.

Now, let's enter a matrix. The function `matrix()` is used to do this. For now the first argument to the function should be "QQ", this means that the matrix will contain either integers or rational numbers (fractions). The second argument is the matrix data.

```
M=matrix(QQ, [[2, 4, 0, 8], [-1, 3, 3, -2], [0, 1, 1, 0]]); M
```

$$\begin{pmatrix} 2 & 4 & 0 & 8 \\ -1 & 3 & 3 & -2 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

The matrix M is defined with three rows and four columns. Notice how the matrix was specified row-by-row, with each row inside a pair of square brackets and all three rows enclosed in another set of square brackets;

commas are used to separate matrix entries and rows.

Perhaps the most confusing thing about using Sage for matrix work is that rows and columns are numbered starting at 0 rather than 1 as is usually done for matrices in mathematical work. This means that the rows in M are numbered 0, 1, and 2 while the columns are numbered 0, 1, 2, and 3. For example, we can access the first and second rows with

```
M.row(0); M.row(1)
```

```
(2, 4, 0, 8)
```

```
(-1, 3, 3, -2)
```

Notice how the function row() is used; it is "attached" to the matrix variable with a dot. This means that the row function operates on the matrix M.

Suppose M is the augmented matrix of a linear system. We can solve the linear system by performing elementary row operations on M. In Sage these row operations are implemented with the functions

- swap_rows() - interchange two rows
- rescale_row() - scale a row by using a scale factor
- add_multiple_of_row() - add a multiple of one row to another row, replacing the row

Remember, row numbering starts at 0. **Pay careful attention to the changes made to the matrix by each of the following commands.**

We want to start with a 1 in the first position of the first column, so we begin by scaling the first row by 1/2.

```
M.rescale_row(0, 1/2); M
```

$$\begin{pmatrix} 1 & 2 & 0 & 4 \\ -1 & 3 & 3 & -2 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

The first argument to rescale_row() is the index of the row to be scaled and the second argument is the scale factor. We could, of course, use 0.5 rather than 1/2 for the scale factor.

Now that there is a 1 in the first position of the first column we continue by eliminating the entry below it.

```
M.add_multiple_of_row(1, 0, 1); M
```

$$\begin{pmatrix} 1 & 2 & 0 & 4 \\ 0 & 5 & 3 & 2 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

The first argument is the index of the row to be replaced, the second argument is the row to form a multiple of, and the final argument is the scale factor. Thus M.add_multiple_of_row(n,m,a) would replace row n with (row n)+a*(row m).

Since the last entry in the first column is already zero we can move on to the second column. Our first step is to get a 1 in the second position of the second column. Normally we would do this by scaling the row but in this

case we can swap the second and third rows.

```
M.swap_rows(1,2); M
```

$$\begin{pmatrix} 1 & 2 & 0 & 4 \\ 0 & 1 & 1 & 0 \\ 0 & 5 & 3 & 2 \end{pmatrix}$$

The arguments to `swap_rows()` are fairly obvious, just remember that row 1 is the second row and row 2 is the third row.

Now we want to eliminate the 5 below the 1. This is done by multiplying the second row by -5 and adding it to the third row.

```
M.add_multiple_of_row(2,1,-5); M
```

$$\begin{pmatrix} 1 & 2 & 0 & 4 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & -2 & 2 \end{pmatrix}$$

To get a 1 as the last entry in the third column we can scale the third row by $-1/2$.

```
M.rescale_row(2,-1/2); M
```

$$\begin{pmatrix} 1 & 2 & 0 & 4 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

At this point the matrix is in **echelon form** (well, having the 1's down the diagonal of the matrix is not required, but it does make our work easier). All that remains to find the solution is to put the matrix in reduced echelon form which requires that we replace all entries in the first three columns that are not on the main diagonal (where the 1's are) with zeros. We will start with the third column and work our way up and to the left. Remember that this is an augmented matrix and we are going to ignore the right-most column; it just "goes along for the ride."

```
M.add_multiple_of_row(1,2,-1); M
```

$$\begin{pmatrix} 1 & 2 & 0 & 4 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

```
M.add_multiple_of_row(0,1,-2); M
```

$$\begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

At this point our solution is complete. We see that the solution to our linear system is $x_1 = 2$, $x_2 = 1$, and $x_3 = -1$.

There is an easy way to check your work, or to carry out these steps in the future. First, let's reload the matrix M.

```
M=matrix(QQ, [[2,4,0,8],[0,1,1,0],[0,1,1,0]]); M
```

$$\begin{pmatrix} 2 & 4 & 0 & 8 \\ -1 & 3 & 3 & -2 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

The function `echelon_form()` will display the echelon form of a matrix without changing the matrix.

```
M.echelon_form()
```

$$\begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

Notice that the matrix M is unchanged.

```
M
```

$$\begin{pmatrix} 2 & 4 & 0 & 8 \\ -1 & 3 & 3 & -2 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

To replace M with its reduced echelon form, use the `echelonize()` function.

```
M.echelonize(); M
```

$$\begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

This brings us to the end of this demonstration.

Example:

A certain three digit number $N = zyx$, equals fifteen times the sum of its digits. If its digits are reversed, the resulting number exceeds N by 396. The one's digit is one larger than the sum of the other two. **Find N.**

Solution:

Here is the augmented matrix for the linear system of 4 equations in the 4 variables x, y, z, N :

```
A = matrix(QQ, [[1, 10, 100, -1, 0], [15, 15, 15, -1, 0],
[100, 10, 1, -1, 396], [1, -1, -1, 0, 1]])
```

```
A
```

$$\begin{pmatrix} 1 & 10 & 100 & -1 & 0 \\ 15 & 15 & 15 & -1 & 0 \\ 100 & 10 & 1 & -1 & 396 \\ 1 & -1 & -1 & 0 & 1 \end{pmatrix}$$

```
A.echelonize()
```

```
A
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 5 \\ 0 & 1 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 135 \end{pmatrix}$$

```
# some new definitions to save some typing....
```

```
def add(M,a,b,c): M.add_multiple_of_row(a,b,c)
```

```
def mul(M,a,c): M.rescale_row(a,c)
```

```
def swap(M,a,b): M.swap_rows(a,b)
```

```
A = matrix(QQ, [[1,10,100,-1,0],[15,15,15,-1,0],
[100,10,1,-1,396],[1,-1,-1,0,1]])
```

```
A
```

$$\begin{pmatrix} 1 & 10 & 100 & -1 & 0 \\ 15 & 15 & 15 & -1 & 0 \\ 100 & 10 & 1 & -1 & 396 \\ 1 & -1 & -1 & 0 & 1 \end{pmatrix}$$

```
add(A,1,0,-15);A
```

$$\begin{pmatrix} 1 & 10 & 100 & -1 & 0 \\ 0 & -135 & -1485 & 14 & 0 \\ 100 & 10 & 1 & -1 & 396 \\ 1 & -1 & -1 & 0 & 1 \end{pmatrix}$$

```
add(A,2,0,-100);A
```

$$\begin{pmatrix} 1 & 10 & 100 & -1 & 0 \\ 0 & -135 & -1485 & 14 & 0 \\ 0 & -990 & -9999 & 99 & 396 \\ 1 & -1 & -1 & 0 & 1 \end{pmatrix}$$

```
add(A,3,0,-1)
```

```
mul(A,1,-1/135)
```

```
A
```

$$\begin{pmatrix} 1 & 10 & 100 & -1 & 0 \\ 0 & 1 & 11 & -\frac{14}{135} & 0 \\ 0 & -990 & -9999 & 99 & 396 \\ 0 & -11 & -101 & 1 & 1 \end{pmatrix}$$

```
add(A,2,1,990)
```

```
add(A,3,1,11)
```

```
A
```

$$\begin{pmatrix} 1 & 10 & 100 & -1 & 0 \\ 0 & 1 & 11 & -\frac{14}{135} & 0 \\ 0 & 0 & 891 & -\frac{11}{3} & 396 \\ 0 & 0 & 20 & -\frac{19}{135} & 1 \end{pmatrix}$$

```
mul(A,2,1/891);A
```

$$\begin{pmatrix} 1 & 10 & 100 & -1 & 0 \\ 0 & 1 & 11 & -\frac{14}{135} & 0 \\ 0 & 0 & 1 & -\frac{1}{243} & \frac{4}{9} \\ 0 & 0 & 20 & -\frac{19}{135} & 1 \end{pmatrix}$$

```
add(A, 3, 2, -20); A
```

$$\begin{pmatrix} 1 & 10 & 100 & -1 & 0 \\ 0 & 1 & 11 & -\frac{14}{135} & 0 \\ 0 & 0 & 1 & -\frac{1}{243} & \frac{4}{9} \\ 0 & 0 & 0 & -\frac{71}{1215} & -\frac{71}{9} \end{pmatrix}$$

```
mul(A, 3, -1215/71); A
```

$$\begin{pmatrix} 1 & 10 & 100 & -1 & 0 \\ 0 & 1 & 11 & -\frac{14}{135} & 0 \\ 0 & 0 & 1 & -\frac{1}{243} & \frac{4}{9} \\ 0 & 0 & 0 & 1 & 135 \end{pmatrix}$$

Backsubstitution:

```
N=135
z= 4/9 + N/243
y = -11*z + (14/135)*N
x = -10*y-100*z+N
x, y, z, N
```

(5, 3, 1, 135)