Very short note on using basic linear algebra in solving linear equations (Humphrey 2021)

In this course we use linear algebra to represent a set of linear equations. A solvable set of equations is a list of linear equations involving the same number of unknown variables as equations. These equation sets are typically referred to as: 'n' equations in 'n' unknowns. We then use linear algebra to solve for these unknown values. This is the bulk of the linear algebra we do in this course.

To make this clearer, consider a single linear equation in the 1 unknown x:
10*x = 20
this is linear since the unknown 'x' is not something like 'x^3' or 'sin(x)', but merely just 'x'. To solve this in a very pedantic way, we can multiply both sides by the *inverse* of the multiplier of 'x'. In other words we multiply both sides by 1/10. Which results in:
1/10*10*x = 1/10 * 20 or just x = 2.

To continue to a simple set of linear equations, we use an example of 2 equations in 2 unknowns x,y:
2x + 4y = 20
3x – 9y = 15
Important: Note that we can write the left-hand side of the above set of equations in matrix and vector notation, using '[]' to indicate a matrix and '{}' to indicate a column vector:

$$\begin{bmatrix} 2 & 4 \\ 3 & -9 \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{Bmatrix} 2x + 4y \\ 3x - 9y \end{Bmatrix}$$

Using standard matrix math where you multiply the terms in the row of the matrix, by the column values of the vector, and where the results are added together and placed in the row position in the resultant vector. This can be done in numpy with the 'dot' product; numpy.dot(A,B), where A and B are scalars, vectors or matrices.

With this aside on matrix multiplication, we can write our example of 2 equations in 2 unknowns:
2x + 4y = 20
3x – 9y = 15
Using linear algebra notation and implied matrix multiplication, this equation set becomes:

$$\begin{bmatrix} 2 & 4 \\ 3 & -9 \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{Bmatrix} 20 \\ 15 \end{Bmatrix}$$

Both sets of equations are equivalent, just written differently. However, in the matrix form we can easily solve by multiplying both sides by the *inverse* of the matrix, similar to the single equation case (note finding a matrix inverse is done with a specific algorithm, not discussed here, but the inverse is given below, or can be calculated with 'numpy.linalg.inv(matrix name)') :

$$\begin{bmatrix} 3/10 & 2/15 \\ 1/10 & -1/15 \end{bmatrix} \begin{bmatrix} 2 & 4 \\ 3 & -9 \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{bmatrix} 3/10 & 2/15 \\ 1/10 & -1/15 \end{bmatrix} \begin{Bmatrix} 20 \\ 15 \end{Bmatrix}$$

The left-hand side is matrix-matrix multiplication, and a matrix times its inverse is the identity matrix, so we get:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{bmatrix} 3/10 & 2/15 \\ 1/10 & -1/15 \end{bmatrix} \begin{Bmatrix} 20 \\ 15 \end{Bmatrix}$$

And carrying out the matrix-vector multiplication on the right-hand side, we get:

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{bmatrix} 3/10 & 2/15 \\ 1/10 & -1/15 \end{bmatrix} \begin{Bmatrix} 20 \\ 15 \end{Bmatrix} = \begin{Bmatrix} 8 \\ 1 \end{Bmatrix}$$

Or x = 8, y=1. This approach works for any set of linear equations with n equations in n unknowns. It does not work if there are more OR less equations than unknowns. It also does NOT work if 2 or more of the equations are effectively the same, such as the 2 equations: x+y=2 and 2x+2y=4. These 2 equations are the 'same' since we can just multiply the first by 2 to get the second, both equations have the same information on x and y.