System of Equations

System of Linear Equations

As you recall from linear regression, we need to teach computer how to solve a system of linear equation.

$$
\begin{aligned}
2a + 3b + 4c &= 20 \\
-4a + b + c &= 1 \\
2a + 2b - c &= 3
\end{aligned}
$$

This system of equation can be written in matrix form as

$$
\begin{bmatrix} 2 & 3 & 4 \\ -4 & 1 & 1 \\ 2 & 2 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 20 \\ 1 \\ 3 \end{bmatrix}
$$

Gaussian Elimination

The easiest way to solve this is by doing Gaussian Elimination. The idea comes from subtracting equations and hope that some variable will be gone. Eventually we should be left with one equation with one unknown where we know how to solve it. This process is called forward elimination.

Then with that one variable we can begin to do subtitution and solve for all others. This process is called backward substitution. Let us do a concrete example for this. It is important to keep in mind when you go through each step that you will need to write a program to do it later. So, try to think about the process in terms of code. Ask yourself every step how do I teach a computer to make the same specific decision.

Forward Elimination

1. First, we want to eliminate a from row 1 and row 2 (the first row is row 0). This means that we need to make the column 0 of row 1 and row 2 row zero. These elements that we want to make them zero are show in red.

$$
\underbrace{\begin{bmatrix} 2 & 3 & 4 \\ \mathbf{-4} & 1 & 1 \\ \mathbf{2} & 2 & -1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} a \\ b \\ c \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} 20 \\ 1 \\ 3 \end{bmatrix}}_{\mathbf{B}}
$$

We want to do something along the line of

$$k \times r_0 + r_1 \to r_1$$

where r_0 is the first row, r_1 is the second row. We hope that the result will make the first column zero. This has the same effect as multiplying k to the first equation and add it to the second equation.

We want to make the first column zero. Thus, the constant k can be found by dividing the first column of the two rows

$$k = -\frac{r_{1,0}}{r_{0,0}}$$

where $r_{1,0}$ is the element at row 1 column 0 and $r_{0,0}$ is the element at row 0 column 0. Since we use $r_{0,0}$ to eliminate all the column 0 of all other rows, $r_{0,0}$ is called the pivot element for this iteration.

Using this idea, the constant for the row 1 and row 2 are 2 and -1 accordingly. So on the matrix we want to do

$$\begin{aligned} 2 \times r_0 + r_1 &\to r_1 \\ -1 \times r_0 + r_1 &\to r_0 \end{aligned}$$

Thus our matrix becomes.

$$\begin{bmatrix} 2 & 3 & 4 \\ \mathbf{0} & \mathbf{7} & \mathbf{9} \\ \mathbf{0} & \mathbf{-1} & \mathbf{-5} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 20 \\ \mathbf{41} \\ \mathbf{-17} \end{bmatrix} \begin{array}{l} \\ 2r_0 + r_1 \to r_1 \\ -4r_0 + r_2 \to r_2 \end{array}$$

2. Now we need to do it again getting rid of the second column on the last row. When we are done with this process, the last row will be just an equation of 1 unknown which we can solve. Similar to what we did previously, We can achieve this by

$$r_2 \times \frac{1}{7} + r_3 \to r_3$$

With this the end result is

$$\begin{bmatrix} 2 & 3 & 4 \\ 0 & 7 & 9 \\ \mathbf{0} & \mathbf{0} & \mathbf{26/7} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 20 \\ 41 \\ \mathbf{-78/7} \end{bmatrix} r_1 \times \tfrac{1}{7} + r_2 \to r_2$$

Notice that the last row is just an equation of one unknown. We can solve for c easily.

Backward subtitution

1. After we are done with forward elimination, the last row is guarantee to be a simple equation of one unknown. c can be found easily.

$$c = \frac{78/7}{26/7} = 3$$

2. With c from the above equation and the matrix we had earlier. We can solve for our second number b using the second row.

$$\begin{bmatrix} 2 & 3 & 4 \\ \mathbf{0} & \mathbf{7} & \mathbf{9} \\ 0 & 0 & 26/7 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 20 \\ \mathbf{41} \\ -78/7 \end{bmatrix}$$

   The second row tells us that

$$7b + 9c = 41$$

   Thus b is simply

$$b = \frac{1}{7}(41 - 9c) = \frac{1}{7}(41 - 27) = 2.$$

3. The same method can be use to find a.

$$\begin{bmatrix} 2 & 3 & 4 \\ 0 & 7 & 9 \\ \mathbf{0} & \mathbf{0} & \mathbf{26/7} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 20 \\ \mathbf{41} \\ \mathbf{-78/7} \end{bmatrix}$$

   Thus,

$$a = \frac{1}{2}(20 - 3b - 4c) = 1.$$

The process can then be generalized to any number of equation.

There are a bunch of code on the internet doing exactly this. Trust me you will regret it badly if you do that missing the joy of writing this code bug free. You don't want a spoiler. Code this up for generic n by n matrix as as a homework The trick is to keep you sane while writing the code is naming the variable nicely.

How to avoid zero?

Sometimes when we try to do forward elimination we will run into a situation where we cannot find the constant to find one row to eliminate another row. For example,

$$\begin{bmatrix} 0 & 1 & 2 \\ 5 & 10 & 20 \\ 7 & 10 & 90 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

If we try to eliminate the first column of row 1 and row 2 of A, we just can't since no matter what we multiply row 0 with we can't make the first column 5 nor 7.

We can avoid this easily by switching the row. This has the same effect as renaming what we call first equation and second equation and so on.

$$\begin{bmatrix} 0 & 1 & 2 \\ 5 & 10 & 20 \\ 7 & 10 & 90 \end{bmatrix} \xrightarrow{r_0, r_1 \to r_1, r_0} \begin{bmatrix} \mathbf{5} & \mathbf{10} & \mathbf{20} \\ \mathbf{0} & \mathbf{1} & \mathbf{2} \\ 7 & 10 & 90 \end{bmatrix}$$

After the swapm we should be able to continue the process of forward elimination. If we can't get it to work by all possible swapping, then that means we get a linearly dependent system of equations; unsolvable system of equations.

System of Non Linear Equations

Unfortunately, not all equation can be written in matrix form. For example

$$x^2 + xy = 10 \tag{1}$$
$$y + 3xy^2 = 57 \tag{2}$$

Trying to do solve this by hand looks pretty hope less. Let us define u and v as

$$u(x,y) = x^2 + xy - 10 \tag{3}$$
$$v(x,y) = y + 3xy^2 - 57 \tag{4}$$

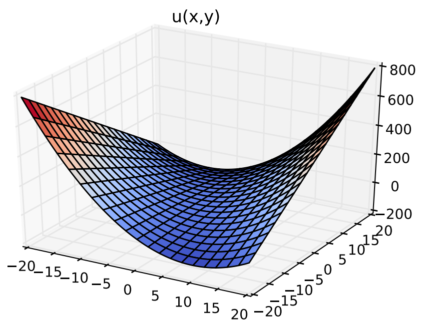
Thus our problem now turn in to finding x, y such that

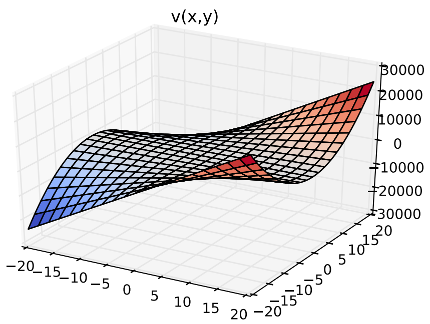
$$u(x,y) = 0$$
$$v(x,y) = 0$$

simultaneously.

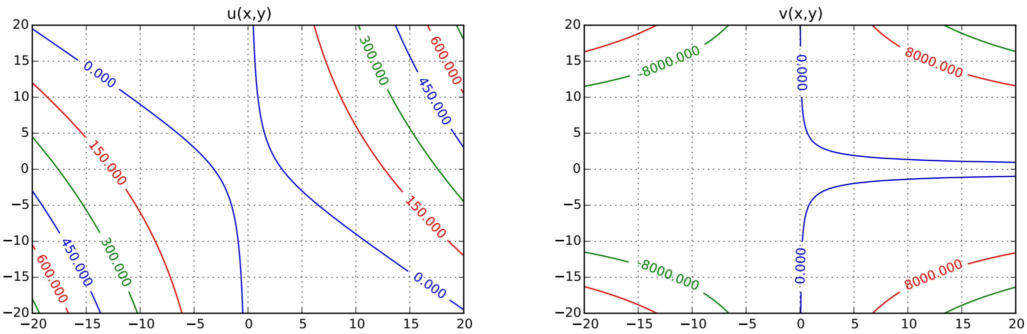
Contour Plot

You can visualize u and v. Your first though is to use some kind of 3d plot. Matplotlib does provide that but despite looking really cool, it doesn't convey much information. The left hand side shows $u(x,y)$ and the right hand side shows $v(x,y)$.





A much better way to do something less fancy. One way to do that is to make a contour plot. A contour plot of $u(x,y)$ and $v(x,y)$ are shown below.



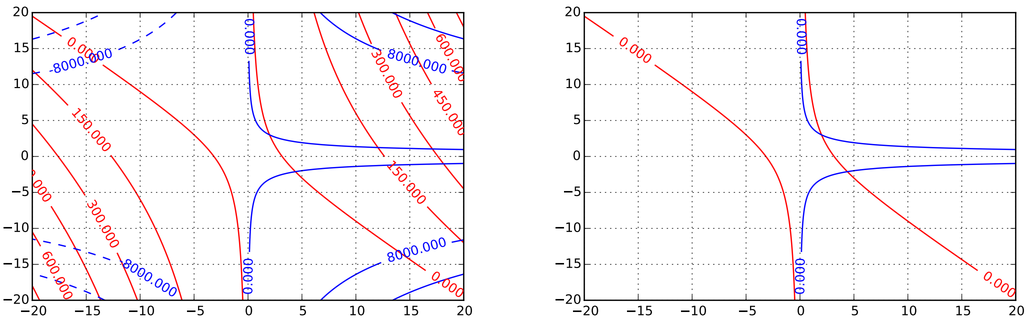
The contour plot is a set of lines constant value. For example, the blue line with 0.000 label on it tell you that all the x and y value on that line, if you plug those into the functions you will get 0.000.

The problem that we want to solve is

$$
\begin{aligned}
u(x,y) =& 0 \\
v(x,y) =& 0
\end{aligned}
$$

That is the value of x and y that makes $u(x,y)$ and $v(x,y)$ zero simultaneously. This means that we want the intersection of the 0.000 line from both $u(x,y)$ and $v(x,y)$ contour plot.

Let us plot the two on the same axis. Here I made the contour lines from u red and contour lines from v blue. On the right figure, I get rid of all other lines.



From the picture, we can see that there are two solution one of them around $(-2.5, 5)$ and the other one around $(3, -2.5)$.

Newton's Method Upgraded

Recall the idea of Newton's method is to keep improving the solution given the old solution. Let us go throught the derivation of the formula one more time.

We start with the Taylor expansion of x_{i+1} around x_i to the first order.

$$
f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i)
$$

The goal is to figure out how to find x_{i+1} such that $f(x_i) = 0$. This can be done by just setting $f(x_{i+1}) = 0$ and solve for x_{i+1}.

$$
\begin{aligned}
0 =& f(x_i) + f'(x_i)(x_{i+1} - x_i) \\
x_{i+1} =& x_i - \frac{f(x_i)}{f'(x_i)}
\end{aligned}
$$

This process can be generalized to solve higher dimension problem. All we need to do is to use Taylor series in higher dimension to the first order. This is actually pretty easy to guess.

$$u(x, y) = u(x_0, y_0) + u_x(x_0, y_0)(x - x_0) + u_y(x_0, y_0)(y - y_0) + \text{higher order} \tag{5}$$

where

$$
\begin{aligned}
u_x =& \frac{\partial}{\partial x} u(x, y) \\
u_y =& \frac{\partial}{\partial y} u(x, y)
\end{aligned}
$$

We can apply the same trick we did in the case of 1D to u and v. First, we expand them to the first order.

$$
\begin{aligned}
u(x_{i+1}, y_{i+1}) =& u(x_i, y_i) + u_x(x_i, y_i)(x_{i+1} - x_i) + u_y(x_i, y_i)(y_{i+1} - y_i) && (6) \\
v(x_{i+1}, y_{i+1}) =& v(x_i, y_i) + v_x(x_i, y_i)(x_{i+1} - x_i) + v_y(x_i, y_i)(y_{i+1} - y_i) && (7) \\
& && (8)
\end{aligned}
$$

The expression looks a bit scary but keep in mind that all the things with x_i and y_i are just numbers.

We then set $u(x_{i+1}, y_{i+1}) = 0$ and $v(x_{i+1}, y_{i+1}) = 0$. The two equations above becomes

$$
\begin{aligned}
0 =& u(x_i, y_i) + u_x(x_i, y_i)(x_{i+1} - x_i) + u_y(x_i, y_i)(y_{i+1} - y_i) && (9) \\
0 =& v(x_i, y_i) + v_x(x_i, y_i)(x_{i+1} - x_i) + v_y(x_i, y_i)(y_{i+1} - y_i) && (10)
\end{aligned}
$$

Moving all x_i and y_i to one side and x_{i+1} and y_{i+1} to another gives.

$$
\begin{aligned}
u_x(x_i, y_i)x_{i+1} + u_y(x_i, y_i)y_{i+1} =& -u(x_i, y_i) + u_x(x_i, y_i)x_i + u_y(x_i, y_i)y_i && (11) \\
v_x(x_i, y_i)x_{i+1} + v_y(x_i, y_i)y_{i+1} =& -v(x_i, y_i) + v_x(x_i, y_i)x_i + v_y(x_i, y_i)y_i && (12)
\end{aligned}
$$

Keep in mind that all the scary looing things on the right hand side are just number and our goal is to solve for x_{i+1} and y_{i+1}. The equation can be written in the matrix form as

$$\begin{bmatrix} u_x(x_i,y_i) & u_y(x_i,y_i) \\ v_x(x_i,y_i) & v_y(x_i,y_i) \end{bmatrix} \begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} -u(x_i,y_i) + u_x(x_i,y_i)x_i + u_y(x_i,y_i)y_i \\ -v(x_i,y_i) + v_x(x_i,y_i)x_i + v_y(x_i,y_i)y_i \end{bmatrix} \tag{13}$$

$$A \begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = C \tag{14}$$

Again the scary looking thing on the right hand side is just a number like 3.145. The above equation is just system of linear equations we can solve using gaussian elimination we learned in the previous section. We can then repeat the process and get closer and closer to the solution.

However, like Newton's method in 1D. This method doesn't guarantee convergence. It will converge only if you start near the solution.

At this point you may ask, can we generalize our favorite method, bisection, to higher dimension. The answer is not so easy and it is quite inefficient. See `http://stackoverflow.com/questions/3513660/multivariate-bisection-method`.

Example

Let us summarize what we learn with a concrete example. Let us consider $u(x, y)$ and $v(x, y)$ defined in Equation 3 and Equation 4.

$$u(x,y) = x^2 + xy - 10 \tag{15}$$

$$v(x,y) = y + 3xy^2 - 57 \tag{16}$$

1. From our visualization, we know there is a solution near (3.-2.5). This is our (x_0, y_0).
2. To compute the next guess (x_1, y_1). We need to compute the two matrices in Equation 13. This involves calculating the partial derivative of u and v. This can be done easily. (This can also be done numerically.)

$$\begin{aligned} u_x(x,y) &= 2x + y \\ u_y(x,y) &= x \\ v_x(x,y) &= 3y^2 \\ v_y(x,y) &= 1 + 6xy \end{aligned}$$

3. Thus, the big 2x2 matrix A on the left hand side becomes

$$\begin{bmatrix} u_x(x_0,y_0) & u_y(x_0,y_0) \\ v_x(x_0,y_0) & v_y(x_0,y_0) \end{bmatrix} = \begin{bmatrix} 3.5 & 3 \\ 18.75 & -44 \end{bmatrix} \tag{17}$$

4. The, matrix C on the right hand side becomes

$$\begin{bmatrix} -u(x_i,y_i) + u_x(x_i,y_i)x_i + u_y(x_i,y_i)y_i \\ -v(x_i,y_i) + v_x(x_i,y_i)x_i + v_y(x_i,y_i)y_i \end{bmatrix} = \begin{bmatrix} 11.5 \\ 169.5 \end{bmatrix} \tag{18}$$

5. x_1 and y_1 can be found by solving

$$\begin{bmatrix} 3.5 & 3 \\ 18.75 & -44 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 11.5 \\ 169.5 \end{bmatrix} \tag{19}$$

   which gives $x_1 = 4.82$ and $y_1 = -1.79$

6. The process is then repeated

i	x_i	y_i	$u(x,y)$	$v(x,y)$
0	3	-2.5	-8.5	-3.25
1	4.82520808561	-1.79607609988	4.61619214994	-12.0993138801
2	4.42937827197	-2.1084377963	0.280323313454	-0.0359023767696
3	4.39391755743	-2.11768337994	0.00158531727786	-0.00302078428503
4	4.39374419843	-2.11778101182	4.69787213575e-08	-8.94120972816e-08
5	4.39374419329	-2.11778101471	-1.24344978758e-14	7.1054273576e-15

7. The other solution can be found by starting from (2.5,2.5).

i	x_i	y_i	$u(x,y)$	$v(x,y)$
1	2.02325581395	2.93023255814	0.0221741481882	-1.95314877935
2	1.99979839644	3.00016056829	-0.00109008009376	0.000497302586794
3	2.00000000567	2.99999998429	8.2710691629e-09	-4.28061767366e-07
4	2.0	3.0	-3.5527136788e-15	-7.1054273576e-15

8. We can verify that this is indeed our solution by plotting the two contour lines and the solution points.