Mathematics for Machine Learning

Coursework 1 - Differentiation

Stanislas Hannebelle

October 2018

1 Question A

Let's prove that:

$$f_1(x) = (x - c)^{\mathrm{T}} C (x - c) + c_0$$

With:

$$c = \frac{1}{6} \times \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$
$$C = \begin{pmatrix} 4 & -1 \\ -1 & 4 \end{pmatrix}$$
$$c_0 = -\frac{1}{6}$$

According to the definition of f_1:

$$f_1(x) = x^{\mathrm{T}} x + x^{\mathrm{T}} B x - a^{\mathrm{T}} x + b^{\mathrm{T}} x$$

With:

$$B = \begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix}$$
$$a = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$
$$b = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$

So, if we write x as $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ with $x_1 \in \mathbb{R}$ and $x_2 \in \mathbb{R}$:

$$f_1(x) = x^{\mathrm{T}} x + x^{\mathrm{T}} B x - x_1 - x_2$$

Then,

$$f_1(x) = x^{\mathrm{T}} x + x^{\mathrm{T}} B x - \frac{1}{6} \times (3x_1 + 3x_2) - \frac{1}{6} \times (3x_1 + 3x_2)$$

Then,

$$f_1(x) = x^{\mathrm{T}}x + x^{\mathrm{T}}Bx - \frac{1}{6} \times \begin{pmatrix}1\\1\end{pmatrix}^{\mathrm{T}} \begin{pmatrix}4 & -1\\-1 & 4\end{pmatrix} x - \frac{1}{6} \times x^{\mathrm{T}} \begin{pmatrix}4 & -1\\-1 & 4\end{pmatrix} \begin{pmatrix}1\\1\end{pmatrix}$$

So,

$$f_1(x) = x^{\mathrm{T}} \left(\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix} + B \right) x - \begin{pmatrix}\frac{1}{6} & \frac{1}{6}\end{pmatrix} \begin{pmatrix}4 & -1\\-1 & 4\end{pmatrix} x + x^{\mathrm{T}} \begin{pmatrix}4 & -1\\-1 & 4\end{pmatrix} \begin{pmatrix}\frac{1}{6}\\\frac{1}{6}\end{pmatrix}$$

Then, by factorizing,

$$f_1(x) = \left(x - \begin{pmatrix}\frac{1}{6}\\\frac{1}{6}\end{pmatrix} \right)^{\mathrm{T}} \begin{pmatrix}4 & -1\\-1 & 4\end{pmatrix} \left(x - \begin{pmatrix}\frac{1}{6}\\\frac{1}{6}\end{pmatrix} \right) - \begin{pmatrix}\frac{1}{6} & \frac{1}{6}\end{pmatrix} \begin{pmatrix}4 & -1\\-1 & 4\end{pmatrix} \begin{pmatrix}\frac{1}{6}\\\frac{1}{6}\end{pmatrix}$$

Thus,

$$f_1(x) = (x-c)^{\mathrm{T}}C(x-c) + c_0$$

With:

$$c = \frac{1}{6} \times \begin{pmatrix}1\\1\end{pmatrix}$$
$$C = \begin{pmatrix}4 & -1\\-1 & 4\end{pmatrix}$$
$$c_0 = -c^{\mathrm{T}}Cc$$

Remark: The numerical value of c_0 is $-\frac{1}{6}$.

2 Question B

C is a symmetrical matrix. Let's have a look at its eigenvalues by computing its characteristic polynomial P_C. Let λ be real number.

$$P_C(\lambda) = \det\left(C - \lambda \mathbb{I}_2\right)$$

Then,

$$P_C(\lambda) = \begin{vmatrix}4-\lambda & -1\\-1 & 4-\lambda\end{vmatrix}$$

So,

$$P_C(\lambda) = \lambda^2 - 8\lambda + 15$$

The discriminant of this polynomial is 4 and its roots are 3 and 5. So, the eigenvalues of the symmetrical matrix C are strictly positive. So, C is a positive-definite matrix.
Hence, $\forall x \neq \begin{pmatrix}0\\0\end{pmatrix}$, $x^{\mathrm{T}}Cx > 0$
So,$\forall x \neq c$, $(x-c)^{\mathrm{T}}C(x-c) > 0$
Then,$\forall x \neq c$, $f_1(x) > c_0$
But, $f_1(c) = c_0$
We can conclude that f_1 has a minimum point, that the input which achieves this minimum is c and that the minimum value of f_1 is $f_1(c) = c_0$ which equals $-\frac{1}{6}$.

3 Question C

By developing f_1, we find that:

$$f_1\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) = 4x_1^2 + 4x_2^2 - 2x_1x_2 - x_1 - x_2$$

So,

$$\frac{\partial f_1}{\partial x_1}\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) = 8x_1 - 2x_2 - 1$$

and

$$\frac{\partial f_1}{\partial x_2}\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) = 8x_2 - 2x_1 - 1$$

Thus,

$$\nabla f_1\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) = \begin{pmatrix} 8x_1 - 2x_2 - 1 \\ 8x_2 - 2x_1 - 1 \end{pmatrix}$$

Thanks to this expression of the gradient of f_1, we can implement the python function grad_f1(x).

4 Question D

By developing f_2, we find that:

$$f_2\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) = \sin(x_1^2 + x_2^2 - 2x_1 + 1) + 3x_1^2 + 3x_2^2 - 2x_1x_2 - 2x_1 + 6x_2 + 3$$

So,

$$\frac{\partial f_2}{\partial x_1}\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) = 2((x_1 - 1)(\cos(x_1^2 + x_2^2 - 2x_1 + 1)) + 3x_1 - x_2 - 1)$$

and

$$\frac{\partial f_2}{\partial x_2}\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) = 2(x_2\cos(x_1^2 + x_2^2 - 2x_1 + 1) + 3x_2 - x_1 + 3)$$

Thus,

$$\nabla f_2\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) = \begin{pmatrix} 2((x_1 - 1)(\cos(x_1^2 + x_2^2 - 2x_1 + 1)) + 3x_1 - x_2 - 1) \\ 2(x_2\cos(x_1^2 + x_2^2 - 2x_1 + 1) + 3x_2 - x_1 + 3) \end{pmatrix}$$

Thanks to this expression of the gradient of f_2, we can implement the python function grad_f2(x).

5 Question E

We implement the python function grad_f3(x) by using autograd.

6 Question F

6.1 Function f_2

Here is a graph illustrating a gradient descent for f_2 starting from the point (1,-1). We chose a step of 0.065 which might not be the optimal value for efficiency. However, it enables us to have a good visualization of the gradient descent algorithm.

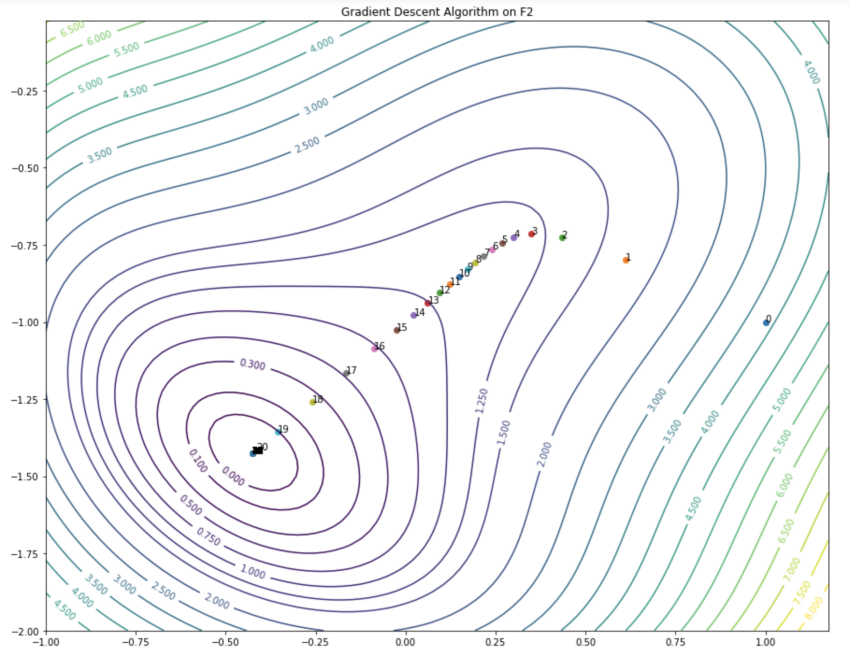

Figure 1: Contour Plot of f_2 and Iterations of Gradient Descent

It looks like there is only one minimum. Here, the gradient descent algorithm converges to this minimum.

6.2 Function f_3

Here is a graph illustrating a gradient descent for f_3 starting from the point (1,-1). We chose a step of 0.2 which, again, might not be the optimal value for efficiency. However, it enables us to have a good visualization of the gradient descent algorithm.

As we can see on the contour plot, f_3 presents two local minima. Our gradient descent algorithm converges to one of these two minima.

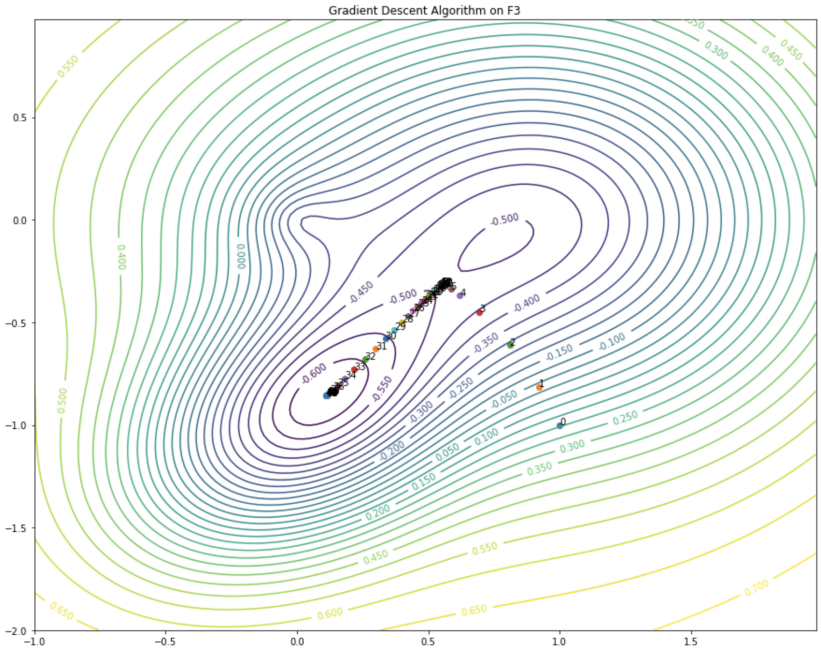

Figure 2: Contour Plot of f_3 and Iterations of Gradient Descent

Remark: Here, it looks like gradient descent algorithm converges to the global minimum and not the other minimum.

7 Question G

7.1 Step size variation

In this section, we will observe that the convergence of gradient descent depends a lot of the step size.

7.1.1 Function f_2

Here, we have one minimum. So, the algorithm can either converge to the minimum or diverge.

Firstly, we observe that for when the step size is lower than 0.025, 50 iterations are not enough to converge to the minimum. We can observe that on Figure 3 showing the case of a step size equal to 0.01.

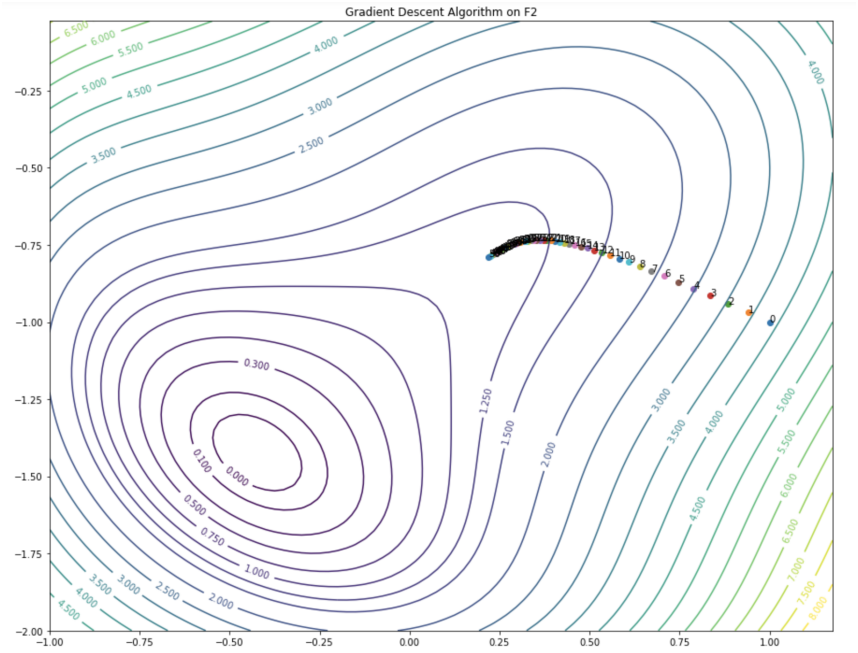

Figure 3: Contour Plot of f_2 and Iterations of Gradient Descent (Step size:0.01)

When the step size is between 0.025 and 0.2, the gradient descent algorithm converges as we saw on Figure 1 (Section 6.1).

When the step size is between 0.2 and 0.27, the step size of the gradient descent algorithm has the same magnitude that the variations of f_2. This is an issue for our gradient algorithm which will usually not converge under these conditions. Figure 4 show the case of a step size equal to 0.19. In that case, iteration points will go from one side of the minimum to the other, never converging towards it.

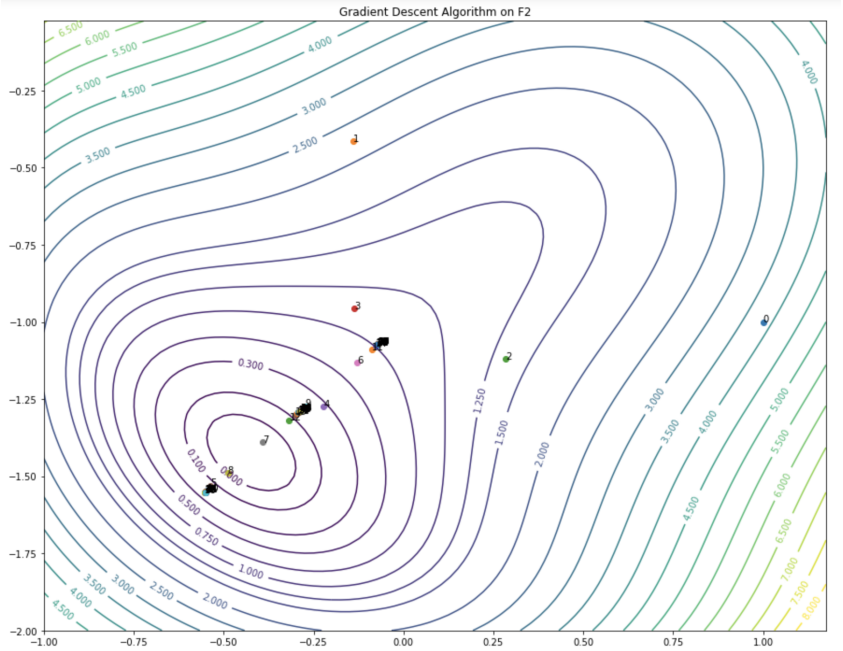

Figure 4: Contour Plot of f_2 and Iterations of Gradient Descent (Step size:0.19)

When the step size is higher than 0.27, the step size of the gradient descent algorithm is greater than the variations of f_2. This is an issue for our gradient descent algorithm which diverges. Figure 5 show the case of a step size equal to 0.3. As we can see, the iteration points go extremely far from the minimum.

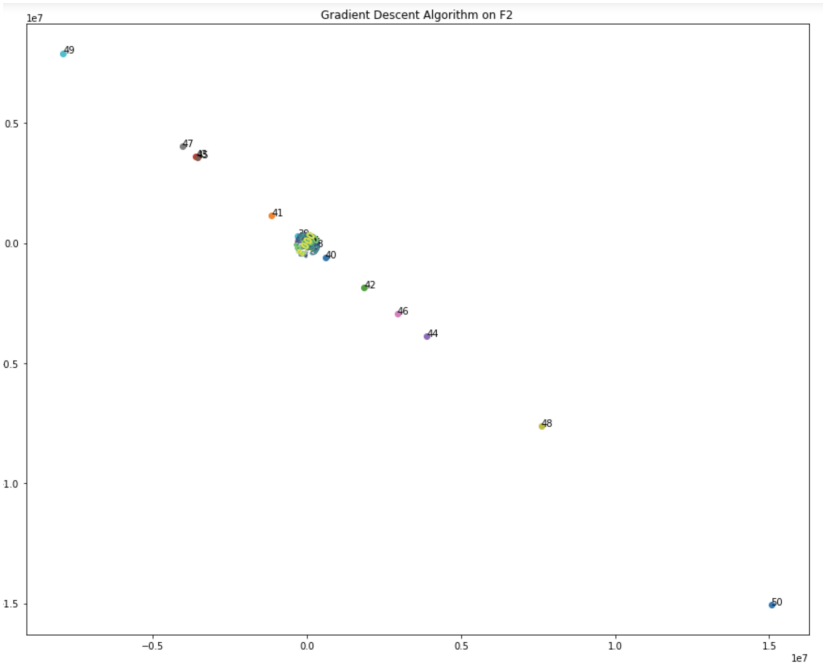

Figure 5: Contour Plot of f_2 and Iterations of Gradient Descent (Step size:0.3)

7.1.2 Function f_3

Here, we have two local minima. The algorithm can converge to the one minimum, to the other one or diverge.

Firstly, we observe that if the step size is below 0.1, 50 iterations is not enough to converge towards a minimum. Figure 6 illustrates the case of a step size equal to 0.05.

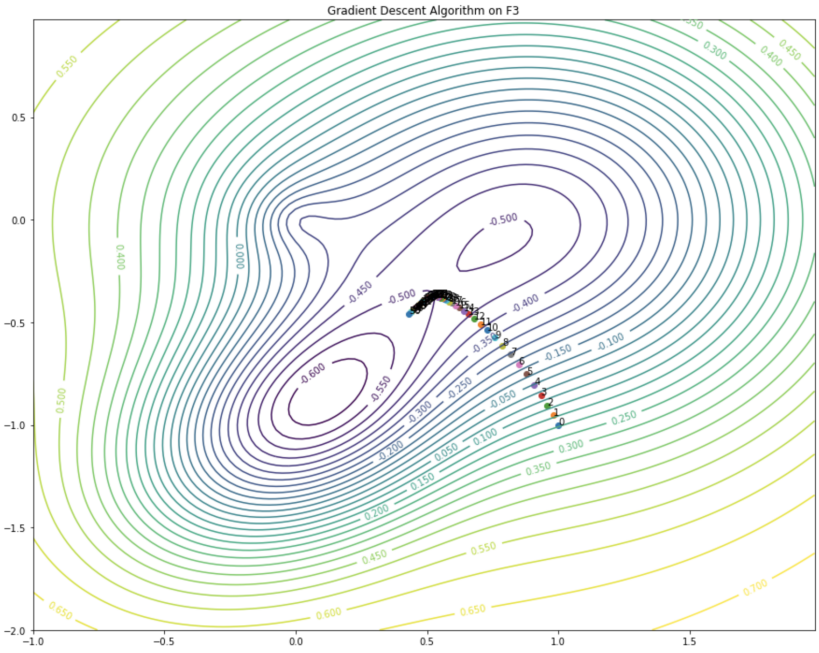

Figure 6: Contour Plot of f_3 and Iterations of Gradient Descent (Step size:0.05)

When the step size value is between 0.1 and 0.25, the gradient descent algorithm converge to one of the two minima. Figure 2 of Section 6.2 illustrates the case of a step size value of 0.2.

However, when the step size value is between 0.2 and 0.85, the algorithm converges towards the other minimum. The reason for this is that the gradient at the starting point is more in the direction of the second minimum than the first one. So, with a higher step size the first iteration point is already close to the second minimum. Figure 7 shows the case of a step size of 0.8.

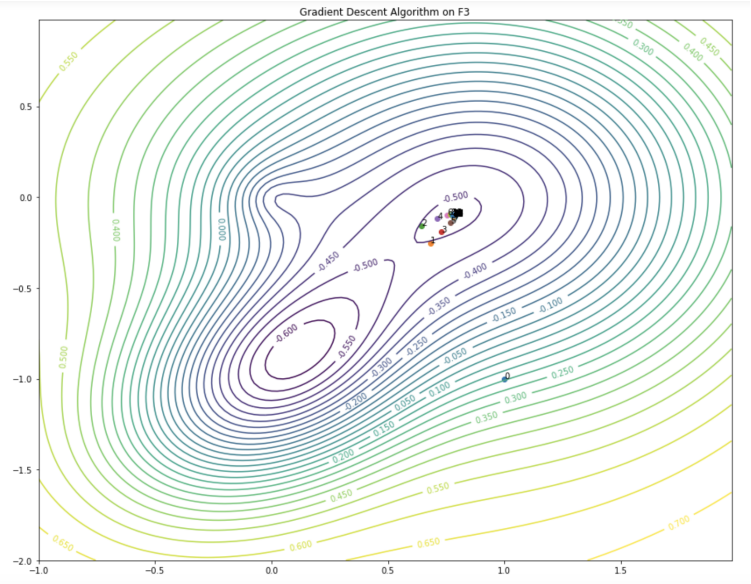

Figure 7: Contour Plot of f_3 and Iterations of Gradient Descent (Step size:0.8)

Finally, when the step size value is greater than 0.85, the gradient descent algorithm diverges. Figure 8 shows the case of step size equals 1.0.

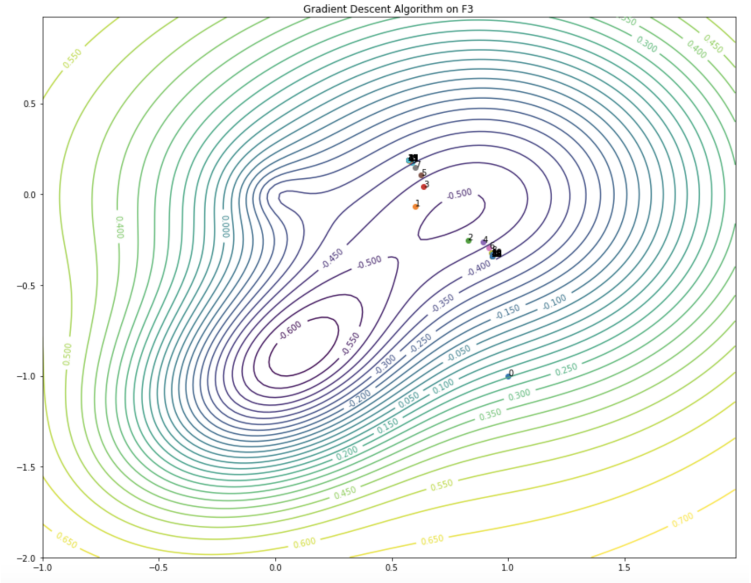

Figure 8: Contour Plot of f_3 and Iterations of Gradient Descent (Step size:1.0)

7.2 Mis-specified Step Sizes

7.2.1 Function f_2

Here is a figure showing the case of a step size of 1.1.

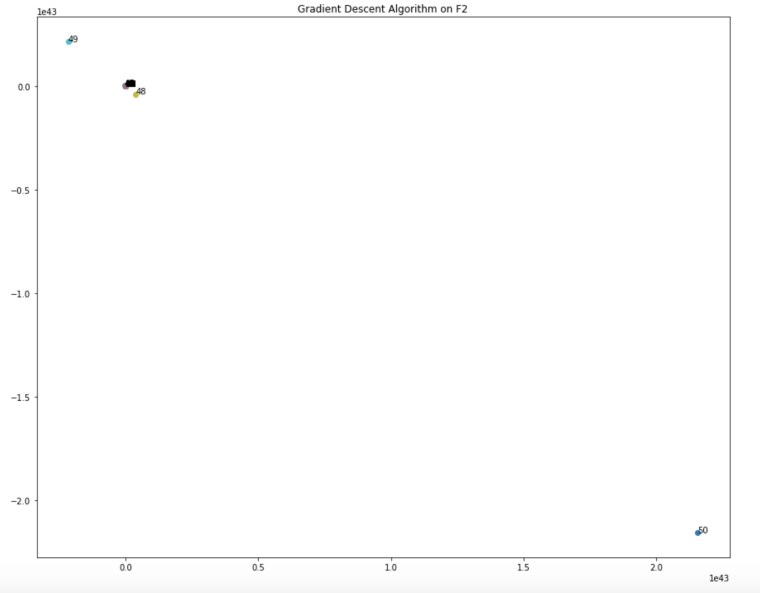

Figure 9: Contour Plot of f_2 and Iterations of Gradient Descent (Step size:1.1)

7.2.2 Function f_3

Here is a figure showing the case of a step size of 8.0.

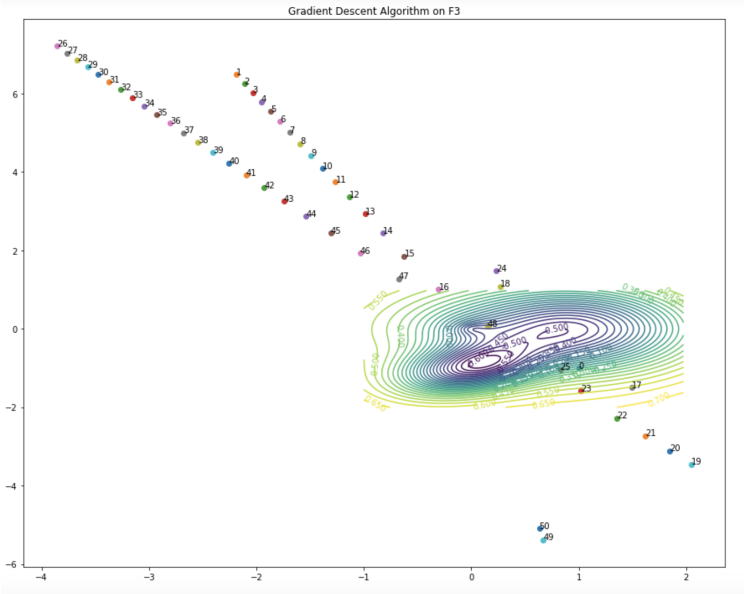

Figure 10: Contour Plot of f_3 and Iterations of Gradient Descent (Step size:8.0)

7.2.3 Observation

On these plots, we can observe that, for f_2 which has a single minimum and looks like a less complex function than f_3, the iterations diverge along an axis. However, in the case of f_3, the iterations seems to diverge along several axis.

8 Conclusion

To conclude, we saw that, the choice of the step size value is decisive for the convergence of the gradient descent algorithm. To converge, the step size as to have a smaller magnitude that the variations of the function under study.Also, the step size has to be high enough for a convergence in the given number of iterations, 50 in our study.
When, the algorithm converges, it can converge towards any local minimum depending on the step size, among other parameters (starting point...).