Q1(a)

Rewrite $f_1(x)$ as:

$$f_1(x) = x^T(B-I)x - (b-a)^T x \qquad (1)$$

Expand the given complete square form $(x-c)^T C(x-c) + c_0$:

$$(x-c)^T C(x-c) + c_0 = (x^T Cx - c^T Cx - x^T Cc + c^T Cc) + c_0 \quad (2)$$

Now, since the above terms $c^T Cx, x^T Cc$ both evaluate to a scalar quantity, and given the transpose of a scalar remains itself:

$$c^T Cx = (c^T Cx)^T = x^T Cc \;;\; x^T Cc = (x^T Cc)^T = c^T Cx$$

Exploit this identity to simplify (2):

$$\begin{aligned}(x-c)^T C(x-c) + c_0 &= (x^T Cx - c^T Cx - c^T Cx + c^T Cc) + c_0 \\ &= x^T Cx - 2c^T Cx + c^T Cc + c_0 \qquad (3)\end{aligned}$$

Equate (3) to (1):

$$x^T(B-I)x - (b-a)^T x = x^T Cx - 2c^T Cx + c^T Cc + c_0$$

And equate the coefficients of the corresponding x terms to find:

$$C = B - I \qquad (4)$$

$$2c^T C = (b-a)^T \quad (5)$$

$$c_0 = -c^T Cc \qquad (6)$$

Now solve (4), (5) and (6) as a set of equations with C, c and c_0 as the three unknowns:

$$c = (\frac{1}{2}(b-a)^T C^{-1})^T = C^{-1}\frac{(b-a)}{2} = (B-I)^{-1}\frac{(b-a)}{2}$$

$$c_0 = -\frac{1}{2}(b-a)^T C^{-1} C(B-I)^{-1}\frac{(b-a)}{2} == -\frac{1}{2}(b-a)^T(B-I)^{-1}\frac{(b-a)}{2}$$

Substitute in the real values for B, a, b:

$$C = B - I = \begin{bmatrix} 3 & -2 \\ -2 & 3 \end{bmatrix}$$

$$c = (B-I)^{-1}\frac{(b-a)}{2} = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}\begin{bmatrix} -1 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.6 \\ -0.4 \end{bmatrix}$$

$$c_0 = -\frac{(b-a)^T}{2}(B-I)^{-1}\frac{(b-a)}{2}$$

$$c_0 = -\begin{bmatrix} -1 & 0 \end{bmatrix}\begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}\begin{bmatrix} -1 \\ 0 \end{bmatrix} = -0.6$$

Q1(b)

The hessian of $f_1(x)$ is:

$$H(f_1) = \begin{bmatrix} \frac{\partial^2 f_1}{\partial {x_1}^2} & \frac{\partial^2 f_1}{\partial x_1 x_2} \\ \frac{\partial^2 f_1}{\partial x_2 x_1} & \frac{\partial^2 f_1}{\partial {x_2}^2} \end{bmatrix} = \begin{bmatrix} 6 & -4 \\ -4 & 6 \end{bmatrix}$$

If there exists a minimum for $f_1(x)$, then its hessian must be positive semidefinite, that is $v^T H(f_1) v \geq 0$, for every direction vector v (a column vector).

In fact, the eigenvalues of the hessian $H(f_1)$ are calculated to be $\begin{cases} \lambda_1 = 2 \\ \lambda_2 = 10 \end{cases}$, both of which are positive values, thus proving the positive definiteness. Therefore a minimum value for function $f_1(x)$ can be obtained.

Use the complete square form obtained in Q1(a) to determine the function minimum, since the minimum value for quadratic form $(x-c)^T C(x-c)$ can only achieve 0, the minimum value for $f_1(x)$ is simply:

$$\min(f_1) = 0 + c_0 = c_0 = -0.6$$

The input to achieve it is:

$$argmin(f_1) = c = \begin{bmatrix} -0.6 \\ -0.4 \end{bmatrix}$$

Q1(c)(i)

Use the complete square form of $f_1(x)$:

$$f_1(x) = (x-c)^T C(x-c) + c_0$$

Let $y = x - c$, rewrite f_1 in terms of y:

$$f_1(y) = y^T C y + c_0$$

Use the chain rule to calculate the gradient:

$$\begin{aligned} \frac{df_1(x)}{dx} &= \frac{\partial f_1}{\partial y}\frac{\partial y}{\partial x} \\ &= (2y^T C)(I) \\ &= 2(x-c)^T C \end{aligned}$$

Substitute in the real values of C and c to get the element-wise expression for the gradient:

$$\frac{df_1(x)}{dx} = [6x_1 - 4x_2 + 2, \qquad 6x_2 - 4x_1]$$

Q1(c)(ii)

For $f_2(x)$, let:

$$z = x - b$$
$$y = (x-b)^T(x-b) = z^T z$$
$$k = x - a$$

Use the chain rule to express the gradient:

$$\frac{df_2(x)}{dx} = \left(\frac{\partial(\cos(y))}{\partial y}\right)\left(\frac{\partial y}{\partial z}\right)\left(\frac{\partial z}{\partial x}\right) + \left(\frac{\partial(k^T B k)}{\partial k}\right)\left(\frac{\partial k}{\partial x}\right)$$

Substitute in the expressions for z, y, k and compute each corresponding derivative to obtain the gradient:

$$\frac{df_2(x)}{dx} = (-\sin(y))(2z^T)(I) + (2k^T B)(I)$$
$$= -2\sin\left((x-b)^T(x-b)\right)(x-b)^T + 2(x-a)^T B$$

And substitute in the real values of B, a, b to obtain the element-wise expression:

$$\frac{df_2(x)}{dx} = \begin{bmatrix} 4(2x_1 - x_2 + 1) - 2(x_1+2)\sin\left((x_1+2)^2 + (x_2-1)^2\right) \\ 4(2x_2 - x_1 - 2) - 2(x_2-1)\sin\left((x_1+2)^2 + (x_2-1)^2\right) \end{bmatrix}^T$$

Q1(c)(iii)

For $f_3(x)$, let:

$$y = (x-a)^T(x-a)$$
$$z = (x-b)^T B (x-b)$$
$$p = |0.01I + xx^T|$$

First, evaluate p (the determinant) to write it out into element-wise expressions:

$$p = \left|\begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix} + \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\begin{bmatrix} x_1 & x_2 \end{bmatrix}\right|$$
$$= \left|\begin{bmatrix} x_1x_1 + 0.01 & x_1x_2 \\ x_2x_1 & x_2x_2 + 0.01 \end{bmatrix}\right|$$

$$= \frac{1}{100}({x_1}^2 + {x_2}^2) + \frac{1}{10000}$$

Now apply the chain rule for f_3:

$$\frac{df_3(x)}{dx} = -\left(\frac{\partial(e^{-y})}{\partial y}\right)\left(\frac{\partial y}{\partial x}\right) - \left(\frac{\partial(e^{-z})}{\partial z}\right)\left(\frac{\partial z}{\partial x}\right) + \left(\frac{\partial(0.1\log(p))}{\partial p}\right)\left(\frac{\partial p}{\partial x}\right) \qquad (7)$$

For each of the derivatives, compute individually:

$$\frac{\partial(e^{-y})}{\partial y} = -e^{-y} = -e^{-(x-a)^T(x-a)}$$

$$\frac{\partial y}{\partial x} = 2(x-a)^T$$

$$\frac{\partial(e^{-z})}{\partial z} = -e^{-z} = -e^{-(x-b)^T B(x-b)}$$

$$\frac{\partial z}{\partial x} = 2(x-b)^T B$$

$$\frac{\partial(0.1\log(p))}{\partial p} = 0.1\frac{1}{p} = \frac{0.1}{\frac{1}{100}({x_1}^2 + {x_2}^2) + \frac{1}{10000}}$$

$$\frac{\partial p}{\partial x} = \begin{bmatrix}\frac{1}{50}x_1 & \frac{1}{50}x_2\end{bmatrix} = \frac{1}{50}x^T$$

Now return to (7), the gradient can be obtained:

$$\frac{df_3(x)}{dx} = 2e^{(-(x-a)^T(x-a))}(x-a)^T + 2e^{(-(x-b)^T B(x-b))}(x-b)^T B + \frac{20}{100({x_1}^2 + {x_2}^2) + 1}x^T$$

It is of convenience that $\frac{df_3(x)}{dx}$ remains in the above vector form.

Q1(d)

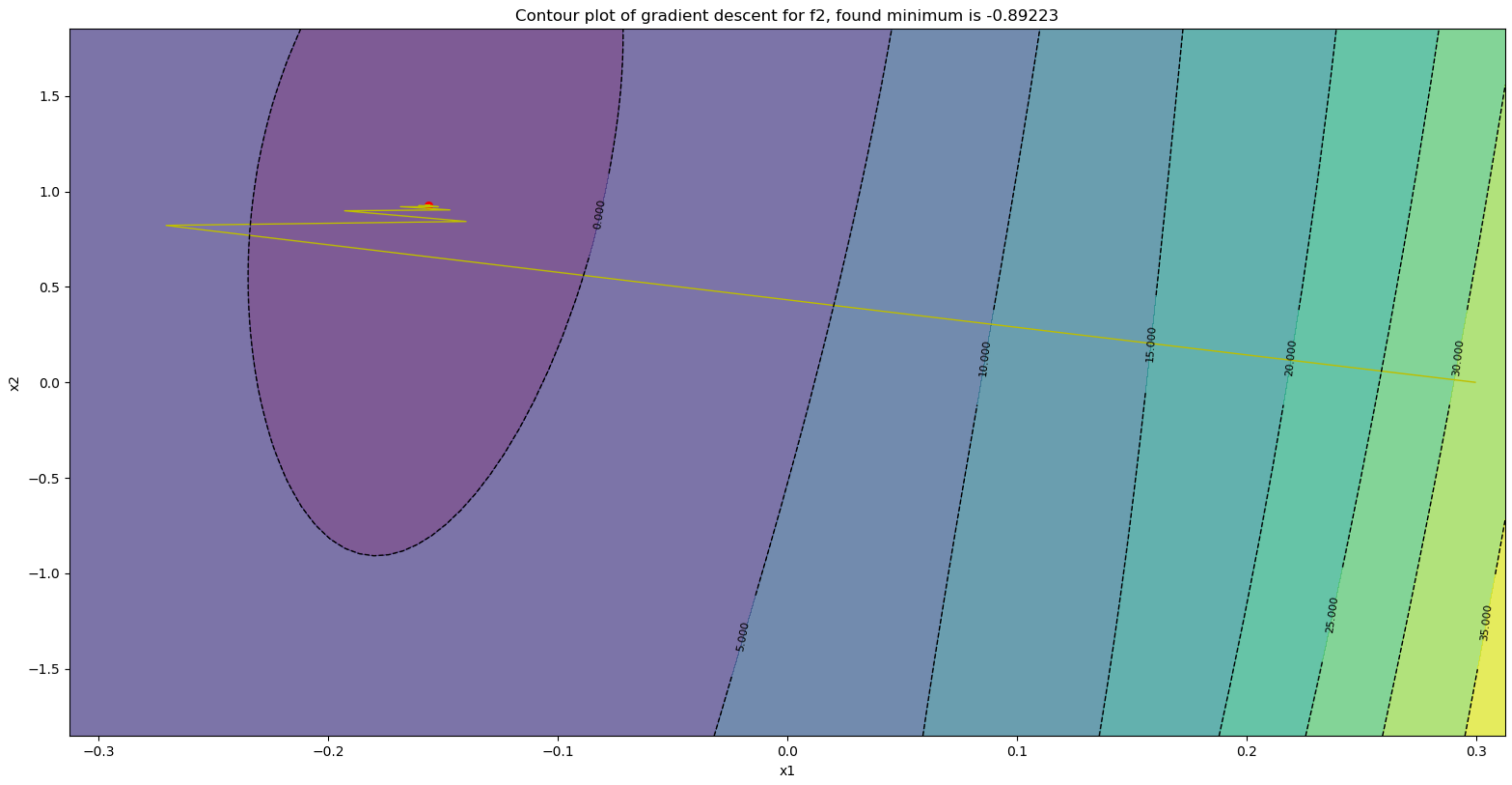

Figure 1. Countour plot of gradient descent for f2, with step size 0.07.

The above figure.1. demonstrates the result from 50-steps gradient descent algorithm for $f_2(x)$, with a constant step size value of 0.07. Note the zig-zagged yellow lines are formed by connecting each input point generated at each step of the gradient descent, ordered from the start to end. The end coordinates have been highlighted with a red dot. The final coordinates of the minimal point for $f_2(x)$ is found to be at (-0.1564, 0.9266).

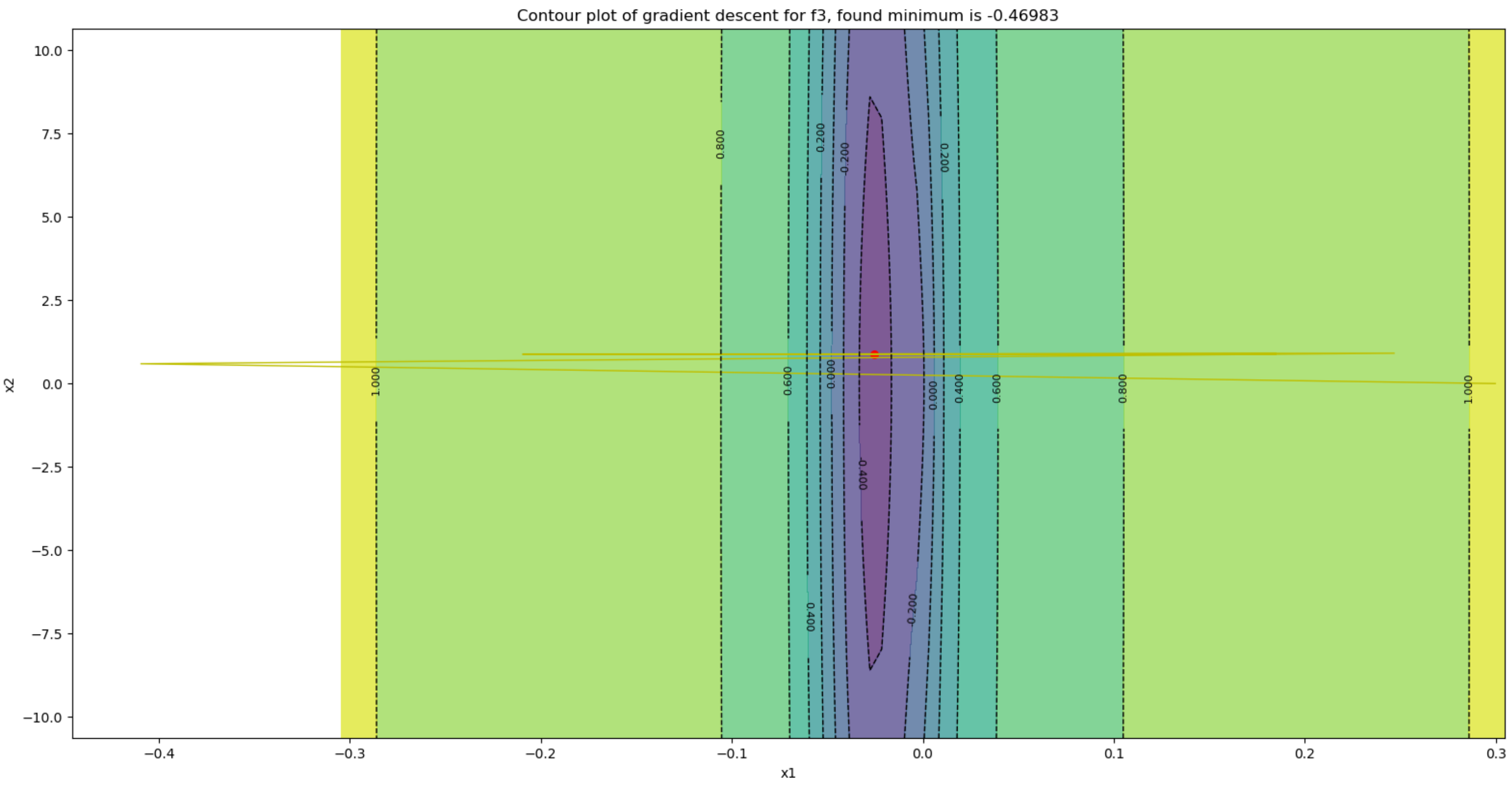

Figure 2. Countour plot of gradient descent for f3, with step size 0.885.

Figure.2. demonstrates the result from 50-steps gradient descent algorithm for $f_3(x)$, with a constant step size value of 0.885. Note the zig-zagged yellow lines are formed by connecting each input point generated at each step of the gradient descent, ordered from the start to end. The end coordinates have been highlighted with a red dot. The final coordinates of the minimal point for $f_3(x)$ is found to be at (-0.0254, 0.8873).

Q1(e)(i)

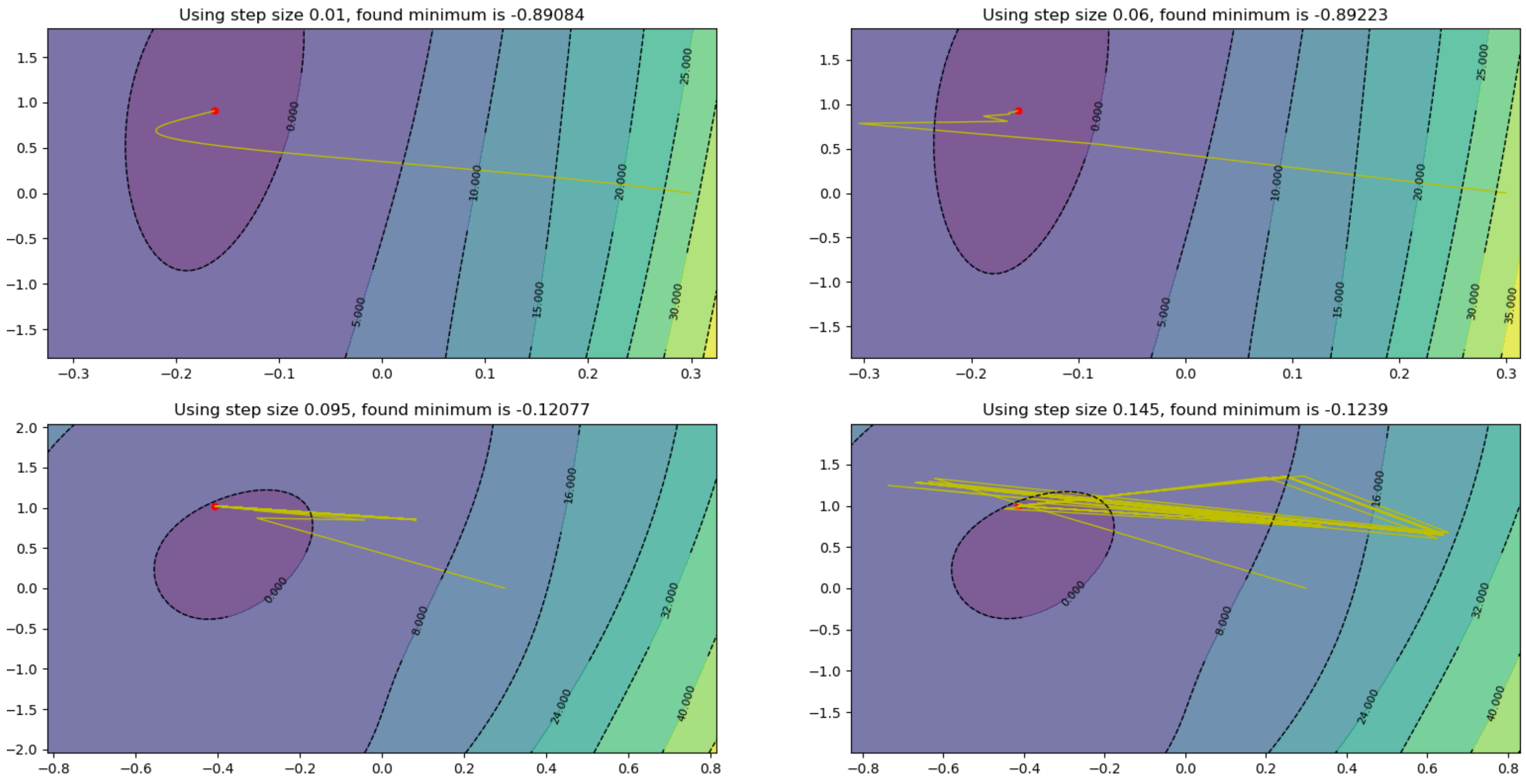

Figure 3. Contour plots of gradient descents for f2, with varying step sizes.

Figure.3. shows the change in behaviour of the algorithm for $f_2(x)$ under different step sizes. Starting from 0.01 step size, the algorithm finds the local minimum with little overshoot. The total distance covered by the yellow trace is the shortest of all four plots, reflecting the small amount of "descent" each step takes. Increasing the step size to 0.06, the overshoot starts to increase mildly, however the oscillation eventually is still able to settle down. The local minimum found in this case is a smaller value but close to the previous one. Increasing the step size to 0.095, as can be seen on the third plot, the overshoot starts to become excessively heavy, and the oscillation seems not to be able to settle to steady level. The resulting minimum value is dramatically different from what it is in the previous step sizes, and it is clearly less optimal. Finally, at step size 0.145, the oscillation is even more severe than the previous case, as indicated by the most zig-zagged trace(yellow line), and the resulting minimum is not optimal. Further increase in step sizes is anticipated to introduce more oscillation and further increase the chance of divergence for the algorithm. For step size between 0.01 and 1, for $f_2(x)$, the optimal step size to use is found to be around 0.06 to 0.08.

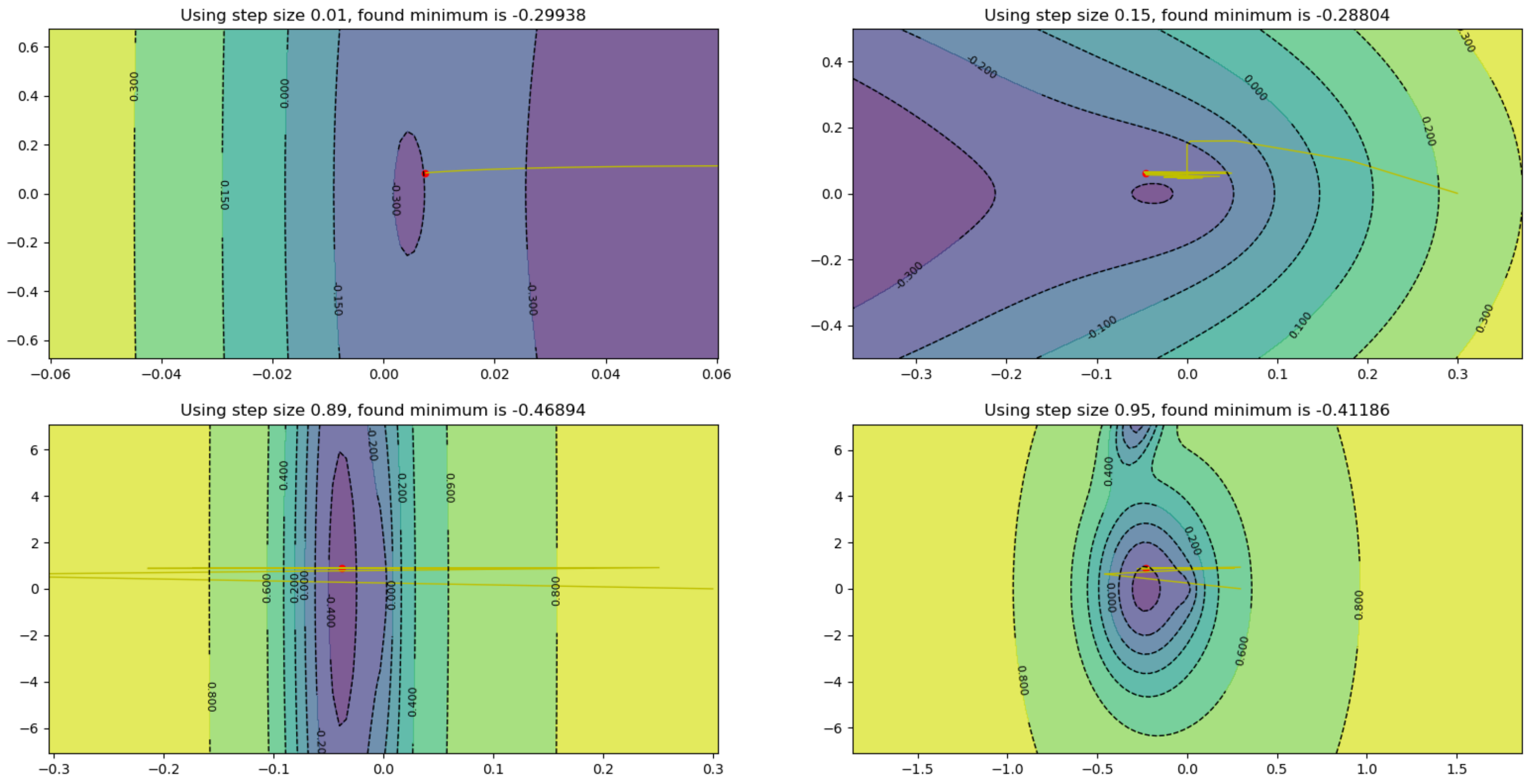

Figure 4. Contour plots of gradient descents for f3, with varying step sizes.

The above figure shows the change in behaviour for $f_3(x)$. Starting from 0.01 step size, the algorithm slowly reduces the x-input with no overshoot occurring. Increasing the step size to 0.15 (the second plot), the algorithm first makes a violent descent and then starts to oscillate around a certain x-value and the oscillation continues to grow. The final red dot appearing on the far left of the oscillation indicates that the algorithm has not been able to converge, at least not under 50 iterations. Thus the result is not optimal for this case. Increasing the step size to 0.89, the algorithm taking a large step each time immediately starts to oscillate, but this time the pattern for the oscillation is a converging one, as can be seen from the smaller and smaller zig-zagging on the third plot. Finally for step size 0.95, a similar oscillation occurs, but under 50 iterations the oscillation level does not seen to reduce to a steady level, perhaps due to the step taken each time being excessively large. The optimal step size for $f_3(x)$ seems to be just around 0.885 to 0.89.

Q1(e)(ii)

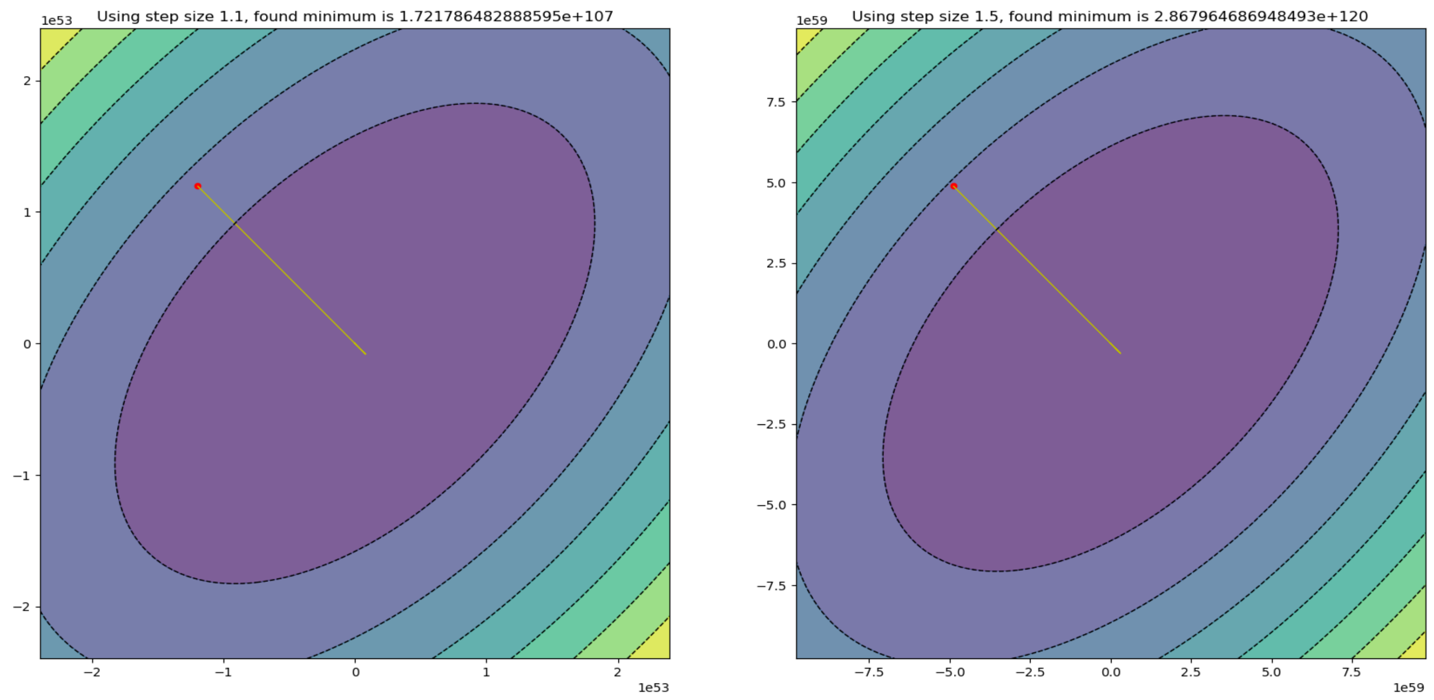

Figure 5. Contour plots of gradient descent for f2, with grossly mis-specified step sizes.

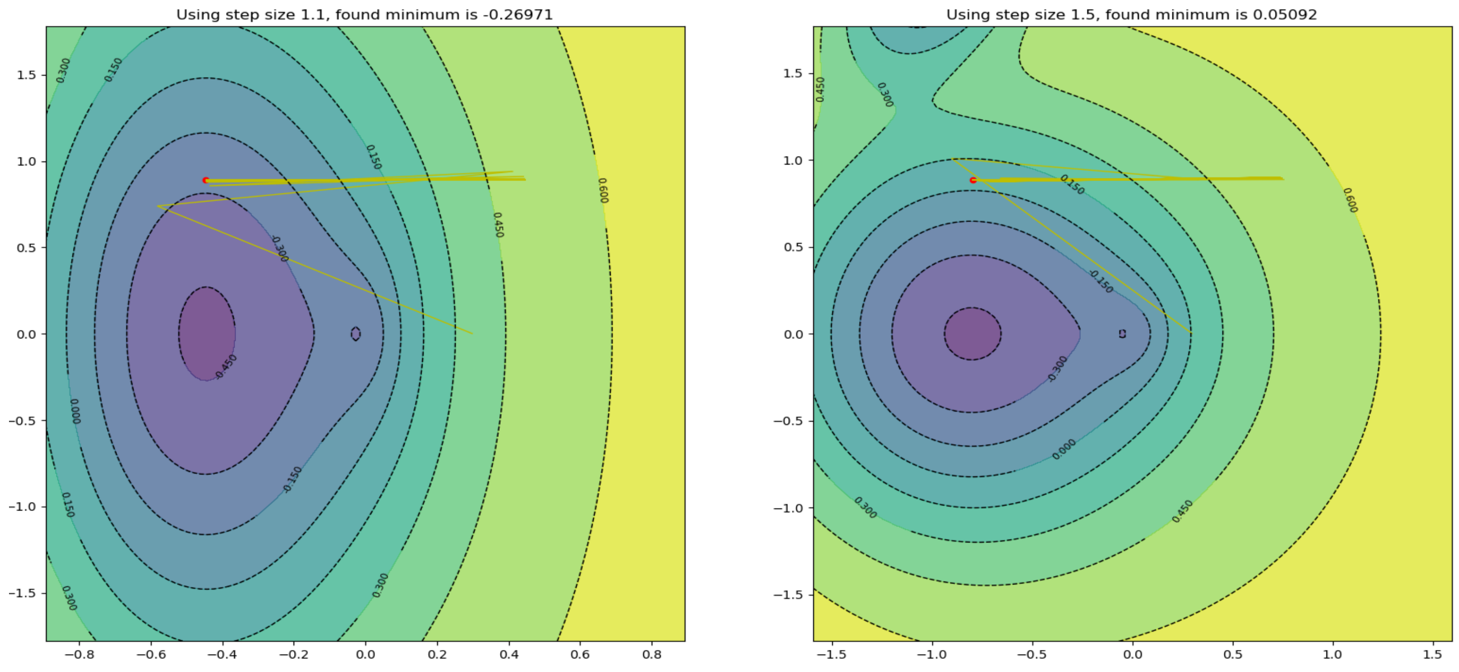

Figure 6. Contours of gradient descent for f3, with grossly mis-specified step sizes.

From figure.5., when the step size gets too large, the algorithm for $f_2(x)$ takes enormous steps each time and never changes in direction to reduce function value, thus diverging rapidly. In figure.6. for $f_3(x)$, similarly algorithm takes drastically big steps, but under step sizes 1.1/1.5 it is still able to change direction to move to both sides of the minimum, although the oscillation is too much to stabilize in 50 iterations.