HW1

Amir Livne 201301405

Carmel Rabinovitz 302958145

1) Analytical Differentiation

Task 1

Given the gradient and hessian of φ, we will calculate the gradient and hessian of $f1(x) = \varphi(Ax)$, as follows:

Gradient:

$$df1(x) = < \nabla f1(x), dx >$$

$$df1(x) = d\ \varphi(Ax) = < \nabla\varphi(Ax), dAx > = Tr(\nabla\varphi(Ax)^T Adx) = Tr((\mathrm{A}^T \nabla\varphi(Ax))^T dx)$$

$$\Rightarrow \quad < \nabla f1(x), dx > = < \nabla\varphi(Ax) \cdot A, dx >$$

$$\Rightarrow \quad \nabla f1(x) = A^T \nabla\varphi(Ax)$$

Hessian:

$$\nabla^2 f1(x)\ dx = d\ \nabla f1(x) = dA^T \cdot \nabla\varphi(Ax) = A^T \cdot \nabla^2\varphi(Ax) A\ dx$$

$$\Rightarrow \quad \nabla^2 f1(x) = A^T \cdot \nabla^2\varphi(Ax) \cdot A$$

Task 2

Given the gradient and hessian of φ, and first and second derivatives of h, we will calculate the gradient and hessian of $f2(x) = h(\varphi(x))$, as follows:

Gradient:

$$< \nabla f2(x), dx > = \nabla f2^T(x) \cdot dx = dh(\varphi(x)) = h'(\varphi(x)) d\varphi(x) = h'(\varphi(x)) \nabla\varphi^T(x) dx$$

$$\Rightarrow \quad \nabla f2(x) = h'(\varphi(x)) \cdot \nabla\varphi(x)$$

Hessian:

$$\begin{aligned} \nabla^2 f2(x)\ dx &= d\ \nabla f2(x) = d\ h'(\varphi(x)) \cdot \nabla\varphi(x) \\ &= h''(\varphi(x))\ \nabla\varphi^T(x) dx \cdot \nabla\varphi(x) + h'(\varphi(x)) \cdot \nabla^2\varphi(x) dx \end{aligned}$$

$$\Rightarrow \quad \nabla^2 f2(x) = h''(\varphi(x))\ \nabla\varphi(x) \nabla\varphi^T(x) + h'(\varphi(x)) \cdot \nabla^2\varphi(x)$$

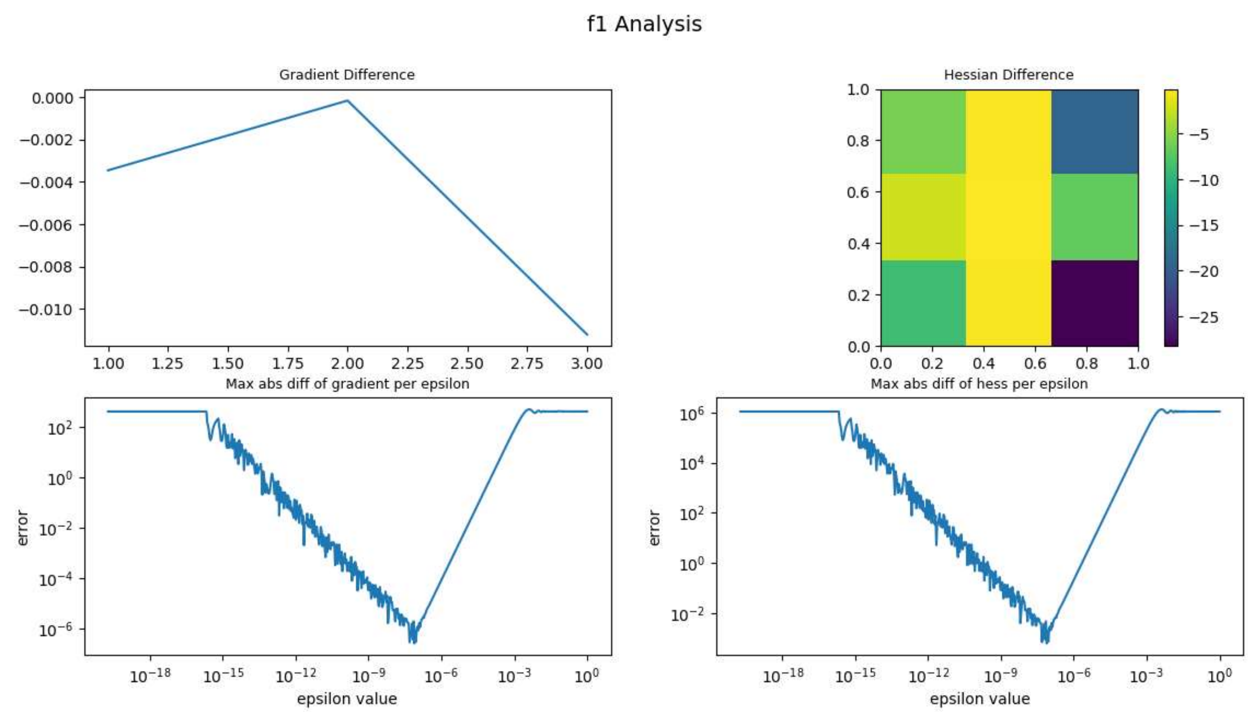
3) Comparison and Conclusions

Task 5

The following graphs were generated by the main.py file, using the functions from the functions.py file.

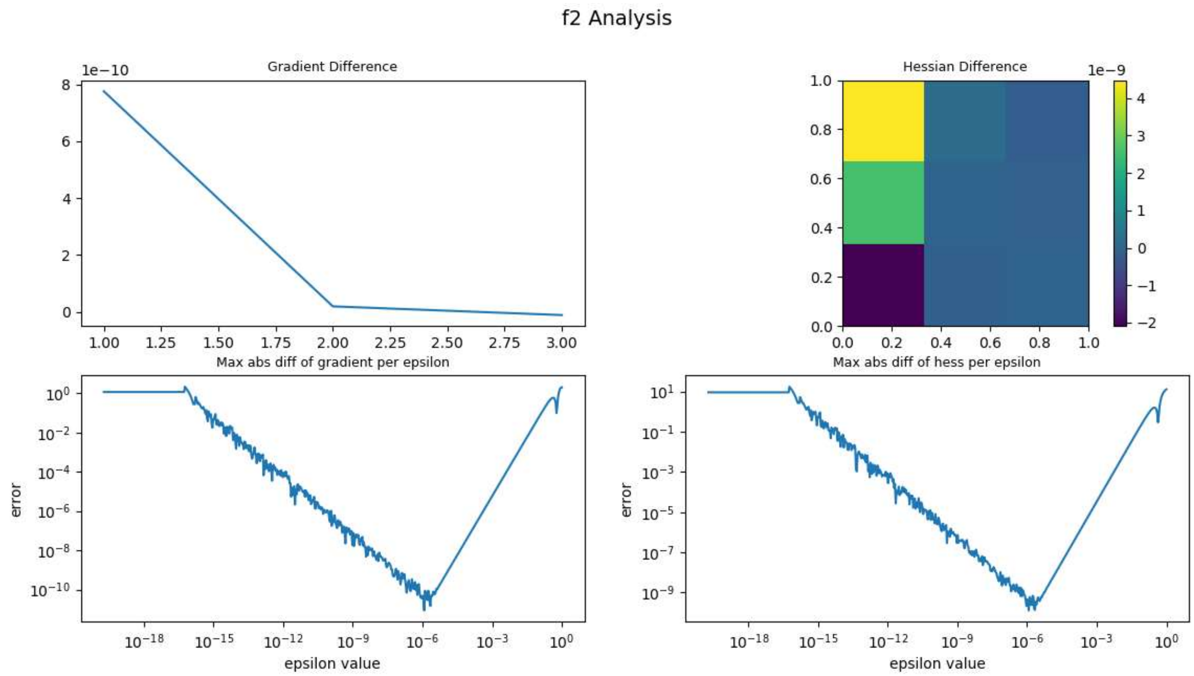
The gradient and hessian difference graphs are calculated according to the default epsilon value - $\varepsilon_{machine}^{1/3} * \max(abs(input))$. On the bottom subplots are the differences for a range of differnat epsilon values.

Following are the graphs for f1:



Where the optimal epsilon is 7.70545511460442e-08.

Following are the graphs for f2:



Where the optimal epsilon is 1.18122304839436944e-06.

Theoretically, we might have expected the error per epsilon value function would be monotonic, and would get lower value for lower epsilons. However, we found that for too small epsilons, the error is rising, to our estimation as a cause of computational limitations in small values in the computer \ python interpreter.

On the other hand, as expected, if we will keep enlarge the epsilon values, we will get bigger errors, and the optimal value we found was around 10^{-6}.

To conclude, given the optimal conditions, we can say that the numerical calculation represent in a very good matter the analytical result. But with that being said, we need to be careful as the epsilon value can damage the accuracy significantly if being chosen wrong.