1 exercise 2

1.1 a

first of all, proof that A is symmetric for B $\in$R $^{n\times n}$ ($A = A^T$):

$$A = B^T B \Rightarrow A^T = (B^T B)^T = B^T B = A \tag{1}$$

Then, proof that A is positive semi-definite for B $\in$R$^{n\times n}$: using the Rayleigh quotient representation of the eigenvalue

$$Ax = \lambda x \Rightarrow B^T Bx = \lambda x \tag{2}$$

Multiplying x^T at both sides:

$$x^T B^T Bx = x^T \lambda x \tag{3}$$

$$(Bx)^T Bx = \lambda ||x||^2$$

Assuming BX=u, the equation can be simplified as:

$$u^T u = \lambda ||x||^2 \Rightarrow ||u||^2 = \lambda ||x||^2 \tag{4}$$

$$||u||^2 \geq 0 \; and \; ||x||^2 \geq 0$$

$$\Rightarrow \lambda \geq 0$$

Therefore, A is positive semi-definite

1.2 b

I need to prove the following equivalent inequality instead with $f(x) = x^T Ax$:

$$f(y + \alpha(x - y)) - \alpha(x) - (1 - \alpha)f(y) \leq 0 \tag{5}$$

$$= (y^T + \alpha x^T - \alpha y^T)(Ay + \alpha Ax - \alpha Ay) - ax^T Ax - (1 - \alpha)y^T Ay$$

$$= y^T Ay + \alpha y^T Ax - \alpha y^T Ay + \alpha x^T Ay + \alpha^2 x^T Ax - \alpha^2 x^T Ax - ...$$

$$= a^2 x^T Ay - \alpha y^T Ay - \alpha^2 y^T Ax + \alpha^2 y^T Ay - \alpha x^T Ax - (1 - \alpha)y^T Ay$$

$$= (1 - 2\alpha + \alpha^2 - 1 + \alpha)y^T Ay + (\alpha^2 - \alpha)x^T Ax + (\alpha + \alpha - \alpha^2 - \alpha^2)y^T Ax$$

$$= (-2\alpha + \alpha^2 + \alpha)y^T Ay + (\alpha^2 - \alpha)x^T Ax + (2\alpha - 2\alpha^2)y^T Ax$$

$$= (\alpha^2 - \alpha)y^T Ay - 2(\alpha^2 - \alpha)y^T Ax + (\alpha^2 - \alpha)x^T Ax$$

$$= (\alpha^2 - \alpha)(x - y)^T A(x - y)$$

because, $(\alpha^2 - \alpha) \leq 0$ for $\alpha \in 0, 1$ and A is symmetric positive semidefinite matrix $(A \geq 0)$ and $(x - y)^T (x - y) \geq 0$.

Therefore, $(\alpha^2 - \alpha)(x - y)^T A(x - y) \leq 0$

2 exercise 3

2.1 a

Taylor series of function f(x) at a is defined as:

$$f(x) = f(a) + \frac{f^{'(a)}}{1!}(x-a) + \frac{f^{"}(a)}{2!}(x-a)^2 + ... \tag{6}$$

now, f(x)=cos(1/x) for x$\in R^+$, and let a=1

$$f(x) = cos(1/1) + \frac{\frac{d}{dx}(cos(\frac{1}{x}))(1)}{1!}(x-1) + \frac{\frac{d^2}{dx^2}(cos(\frac{1}{x}))(1)}{2!}(x-1)^2 + ... \tag{7}$$

$$= cos(1) + sin(1)(x-1) + \frac{-cos(1) - 2sin(1)}{2}(x-1)^2 + ...$$

2.2 b

Taylor Theorem with muti-variables can be shown as:

$$g(x+p) = g(x) + \triangledown g(x)^T p + \frac{1}{2} p^T \triangledown^2 g(x+tp)p + ... \tag{8}$$

now, g(x)=$exp(||\mathbf{x}||^2)$ for $\mathbf{x} = (x_1, x_2)$ $\in$R^2, therefore:

$$p = \begin{bmatrix} \triangledown x_1 \\ \triangledown x_2 \end{bmatrix} \ and \ \triangledown g(x) = \begin{bmatrix} 2x_1 e^{x_1^2+x_2^2} \\ 2x_2 e^{x_1^2+x_2^2} \end{bmatrix} \tag{9}$$

$$\triangledown^2 g(x) = \begin{bmatrix} \frac{\partial^2 g}{\partial x_1^2} & \frac{\partial^2 g}{\partial x_1 \partial x_2} \\ \frac{\partial^2 g}{\partial x_2 \partial x_1} & \frac{\partial^2 g}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} (4x_1^2+2)e^{x_1^2+x_2^2} & 4x_1x_2e^{x_1^2+x_2^2} \\ 4x_1x_2e^{x_1^2+x_2^2} & (4x_2^2+2)e^{x_1^2+x_2^2} \end{bmatrix} \tag{10}$$

Thus,

$$g(x+p) = e^{(x_1+p)^2+(x_2+p)^2} \tag{11}$$

$$= e^{x_1^2+x_2^2} + \begin{bmatrix} 2x_1 e^{x_1^2+x_2^2} \\ 2x_2 e^{x_1^2+x_2^2} \end{bmatrix}^T \begin{bmatrix} \triangledown x_1 \\ \triangledown x_2 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \triangledown x_1 \\ \triangledown x_2 \end{bmatrix}^T \begin{bmatrix} (4x_1^2+2)e^{x_1^2+x_2^2} & 4x_1x_2e^{x_1^2+x_2^2} \\ 4x_1x_2e^{x_1^2+x_2^2} & (4x_1^2+2)e^{x_1^2+x_2^2} \end{bmatrix} \begin{bmatrix} \triangledown x_1 \\ \triangledown x_2 \end{bmatrix}$$

3 exercise 4

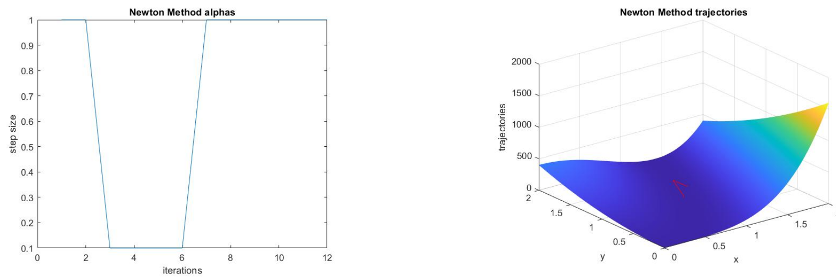
3.1 a

refer to MATLAB Grader

3.2 b

The table below shows the information for the Newton Methods:

Initial point	Optimal solution	Number of Iterations	Optimal Function Value
$[1.2, 1.2]^T$	$[1, 1]^T$	11	0

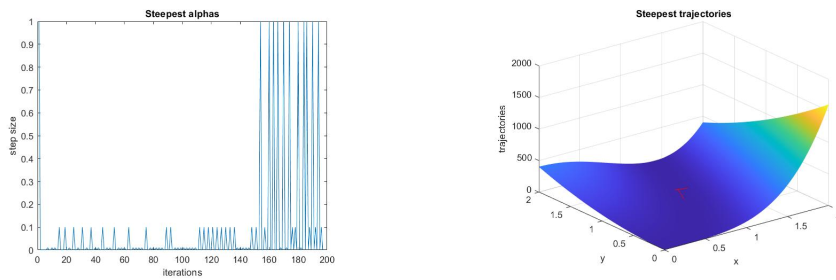


(a) Step size (b) trajectories

Figure 1: Using Newton's algorithm

The table below shows the information for the Steepest Descent Methods:

initial point	Optimal solution	Number of Iterations	Optimal Function Value
$[1.2, 1.2]^T$	$[1, 1]^T$	196	0



(a) Step size (b) trajectories

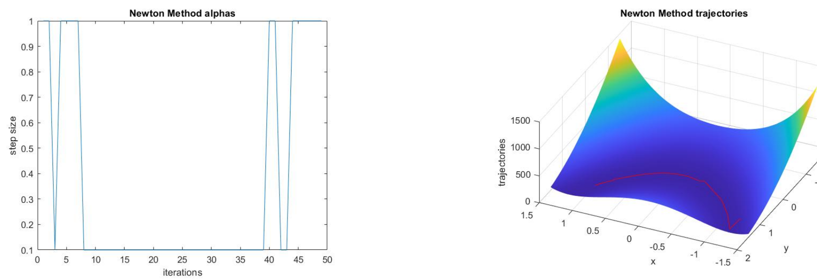
Figure 2: Using Steepest descent

Comment on the obtained trajectories:
In the figure 1, The trajectories using newton method is quite short. It rapidly converges to the true solution along the lowest area within 11 iterations. The pure Newton iteration converges rapidly when it started close enough to a solution, but its steps may not even be descent directions away from the solution. The trajectories using steepest descent methods is also quite short. It can quickly converge to the true solution within 196 iterations. The reason for that might be it starts at a easy point which is quite close to the true solution.

3.3 c

The table below shows the information for the Newton Methods:

initial point	Optimal solution	Number of Iterations	Optimal Function Value
$[-1.2, 1]^T$	$[1, 1]^T$	48	0

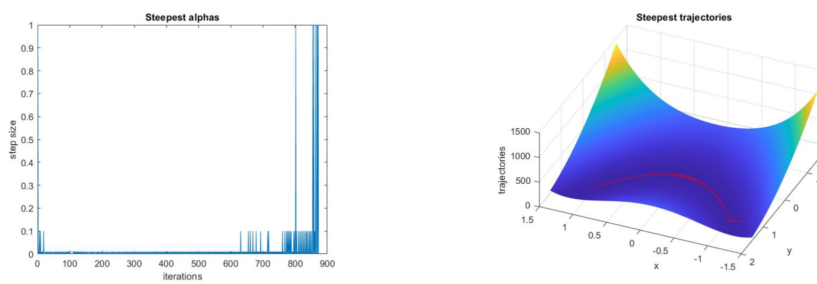


(a) Step size (b) trajectories

Figure 3: Using Newton's algorithm

The table below shows the information for the Steepest Descent Methods:

initial point	Optimal solution	Number of Iterations	Optimal Function Value
$[-1.2, 1]^T$	$[1, 1]^T$	873	0

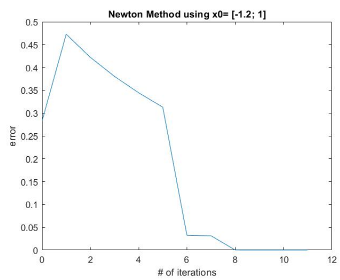


(a) Step size (b) trajectories

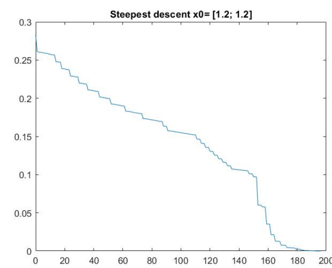
Figure 4: Using Steepest descent

Comment on the obtained trajectories:
In the figure 3b, It is noticed that the trajectories using newton method has a smooth cure along the lowest line in the graph from the initial point to the true solution. However, In the figure 4b, the iterates zigzag toward the solution. along the lowest line in the graph. It might be because the initial point is far way from the true solution. The steepest descent method requires more iterations than the newton method does.

3.4 d



(a) Newton Methods

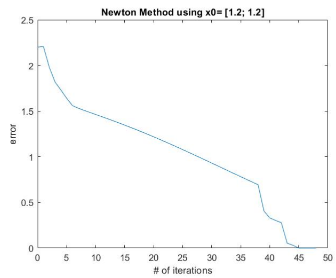


(b) Steepest

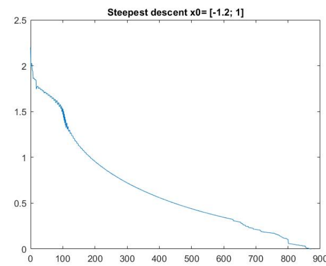
Figure 5: the convergence rate for part b

Comments on the figures 5:
For the newton method, the theoretical convergence rate is quadratic, having a fast rate of local convergence. The convergence of Newton method can be obtained using $||x_k - x^*||$ over the iterates. As you can see in the figure 5a, It is noticed that there is a rise at the first iteration. After that, it rapidly converges to the true solution within 11 iterations. The pure Newton iteration converges rapidly when it started close enough to a solution, but its steps may not even be descent directions away from the solution.
For the steepest descent method, the theoretical convergence rate is linear. The proof of the linear result is given by Luenberger. The convergence can be accessed using $||x_k - x^*||$ over the iterates, so that the norm measure the difference between the current objective value and the optimal value which is $[1, 1]^T$ in this case. As you can see in the figure 5b, the graph converges at a linear rate apart from the part after the 150th iteration. The rate of convergence behavior of the steepest descent method is essentially the same on general nonlinear functions. The step length is the global minimizer along the search direction. it is noticed that the iterates zigzag toward the solution.

Comments on the figures 6:
In the figure 6a, it can be noticed that the newton method can converge to the true solution within 48 iterations. There are two rapid drops at the beginning and last part, and one steady drop in the middle. The line is quite smooth in general. In the figure 6b, the steepest descent method can linearly converge to the true solution within 873 iterations. This is not as efficient as the newton method. It can be observed that the iterate zigzag toward the true solution.



(a) Newton Methods



(b) Steepest

Figure 6: the convergence rate for part c