Name:	Aneesh Panchal	**Assignment No:**	Assignment 2
SR No:	06-18-01-10-12-24-1-25223	**Course Code:**	E0230
Email ID:	aneeshp@iisc.ac.in	**Course Name:**	Computational Methods of Optimization
Date:	October 20, 2024	**Term:**	AUG 2024

Solution 1

1. Given, A is $n \times n$ symmetric PD matrix.
   Yes, the system $Ax = b$ do have a unique solution because,
   $rank(A) = rank(Aug(A, b)) = n$, where $Aug(A, b)$ is augmented matrix.
   Required convex quadratic minimization problem is as follows,

$$\arg\min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = \arg\min_{\mathbf{x}} \left(\frac{\mathbf{1}}{\mathbf{2}} \mathbf{x}^{\mathbf{T}} \mathbf{A} \mathbf{x} - \mathbf{b}^{\mathbf{T}} \mathbf{x} \right)$$

$$\nabla f(x) = Ax - b$$

   i.e. at optimal point x^*,
   $Ax^* - b = 0 \implies Ax^* = b$

2. Using Conjugate Gradient Descent algorithm, on the oracle $f1(srno, True)$ we get,
   Optimal value of x^*: $[\mathbf{2}, \mathbf{6}, \mathbf{6}, \mathbf{1}, \mathbf{9}]$
   Number of iterations required: **5**

   Assumptions and Considerations to be taken for implementations are as follows,
   Termination Condition: $\|Ax - b\|_2 \equiv |Ax - b| \leq 10^{-10}$
   Max. number of iterations: 10000

3. Given, A is $m \times n$ matrix with $m > n$ and b is $m \times 1$ vector. **No**, the given system will not have unique solution **until** A is full rank matrix i.e. $rank(A^T A) = n$.
   Objective is to minimize $\|Ax - b\|$,

$$\arg\min_x \|Ax - b\| = \arg\min_x \|Ax - b\|^2 = \arg\min_x (Ax - b)^T (Ax - b) = \arg\min_x (x^T A^T - b^T)(Ax - b)$$

$$= \arg\min_x \left(x^T A^T Ax - x^T A^T b - b^T Ax + b^T b \right)$$

$$= \arg\min_x \left(x^T A^T Ax - 2x^T A^T b \right) = \arg\min_x \left(\frac{1}{2} x^T A^T Ax - x^T A^T b \right)$$

   because, $x^T A^T b$ and $b^T Ax$ both are scalar.
   Also, $\arg\min_x (k(ax + b)) = \arg\min_x (ax + b) = \arg\min_x (ax)$, where k is positive constant.

$$\arg\min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\| = \arg\min_{\mathbf{x}} \left(\frac{\mathbf{1}}{\mathbf{2}} \mathbf{x}^{\mathbf{T}} \mathbf{A}^{\mathbf{T}} \mathbf{A} \mathbf{x} - \mathbf{x}^{\mathbf{T}} \mathbf{A}^{\mathbf{T}} \mathbf{b} \right)$$

   The equation derived above will have unique solution when,
   A is full rank or, $rank(A^T A) = rank(Aug(A^T A, A^T b)) = n$.

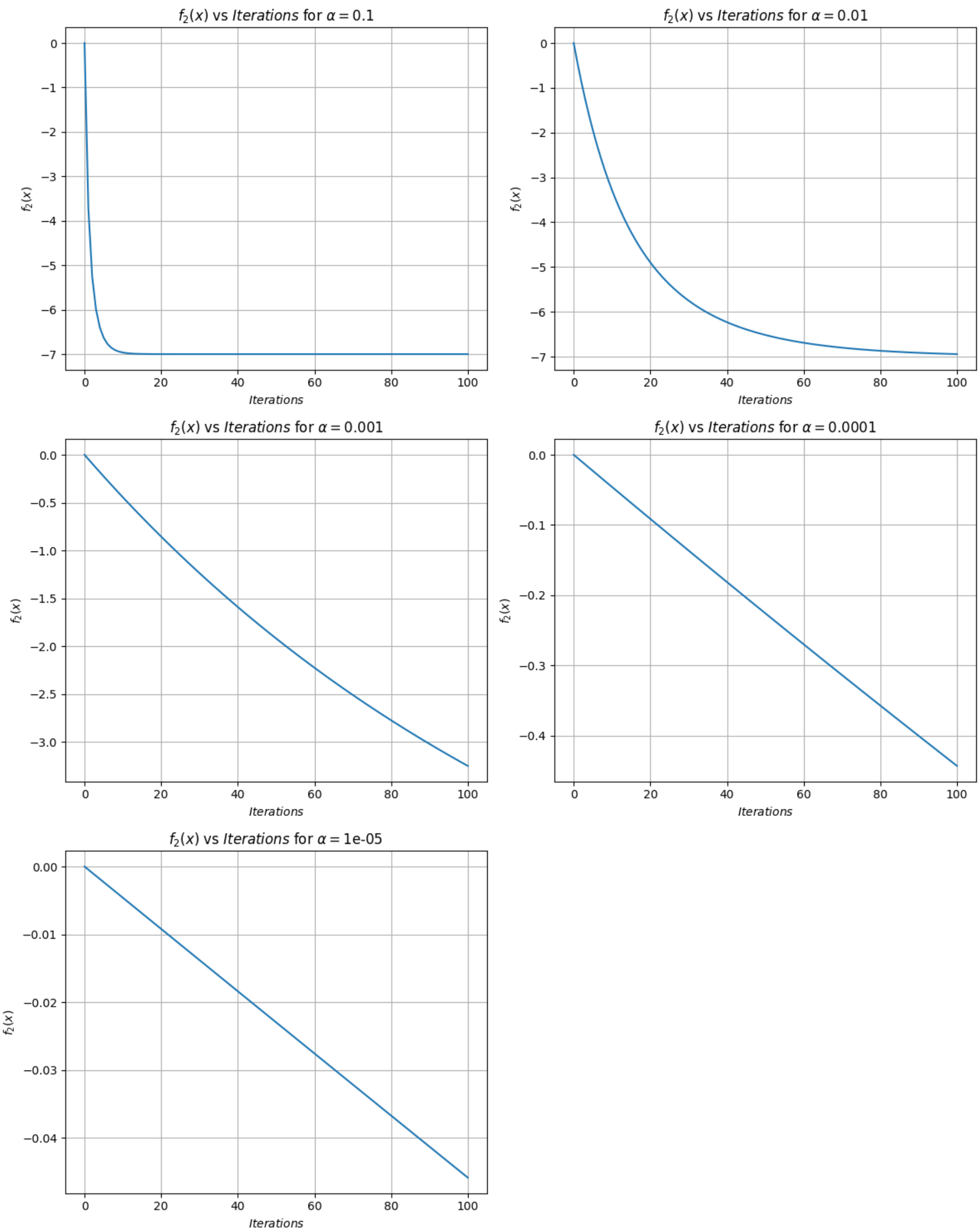
4. Using Conjugate Gradient Descent algorithm, on the oracle $f1(srno, False)$ we get,
   Optimal value of x^*: $[\mathbf{2.0}, \mathbf{6.0}, -\mathbf{8.8817842} \times \mathbf{10^{-16}}, \mathbf{9.0}, \mathbf{7.0}]$
   Number of iterations required: **1**

   Assumptions and Considerations to be taken for implementations are as follows,
   Termination Condition: $\|Ax - b\|_2 \equiv |A^T Ax - A^T b| \leq 10^{-10}$
   Max. number of iterations: 10000

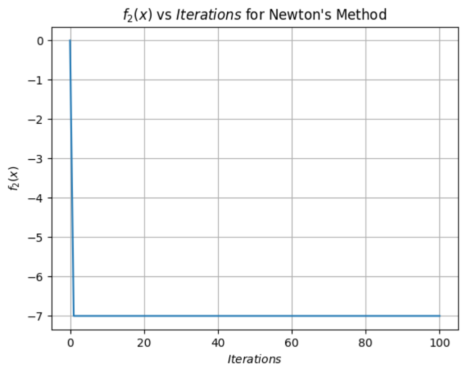
Solution 2

1. **Gradient Descent Implementation for different Step Sizes**
   Choices of step size assumed, α: $[10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}]$.
   Number of Iterations: 100.
   Initial Point, x_0: [0, 0, 0, 0, 0].

α	x_{100}	$f_2(x_{100})$
10^{-1}	[1, 1, 1, 1, 1]	-6.999999999999999
10^{-2}	[0.86738044, 0.99407947, 0.86738044, 0.86738044, 0.95244749]	-6.943756667002219
10^{-3}	[0.1814332, 0.39422956, 0.1814332, 0.1814332, 0.25951574]	-3.2499751965318433
10^{-4}	[0.01980329, 0.04878247, 0.01980329, 0.01980329, 0.02955883]	-0.44297215486087077
10^{-5}	[0.00199802, 0.00498765, 0.00199802, 0.00199802, 0.00299555]	-0.0458253740335511

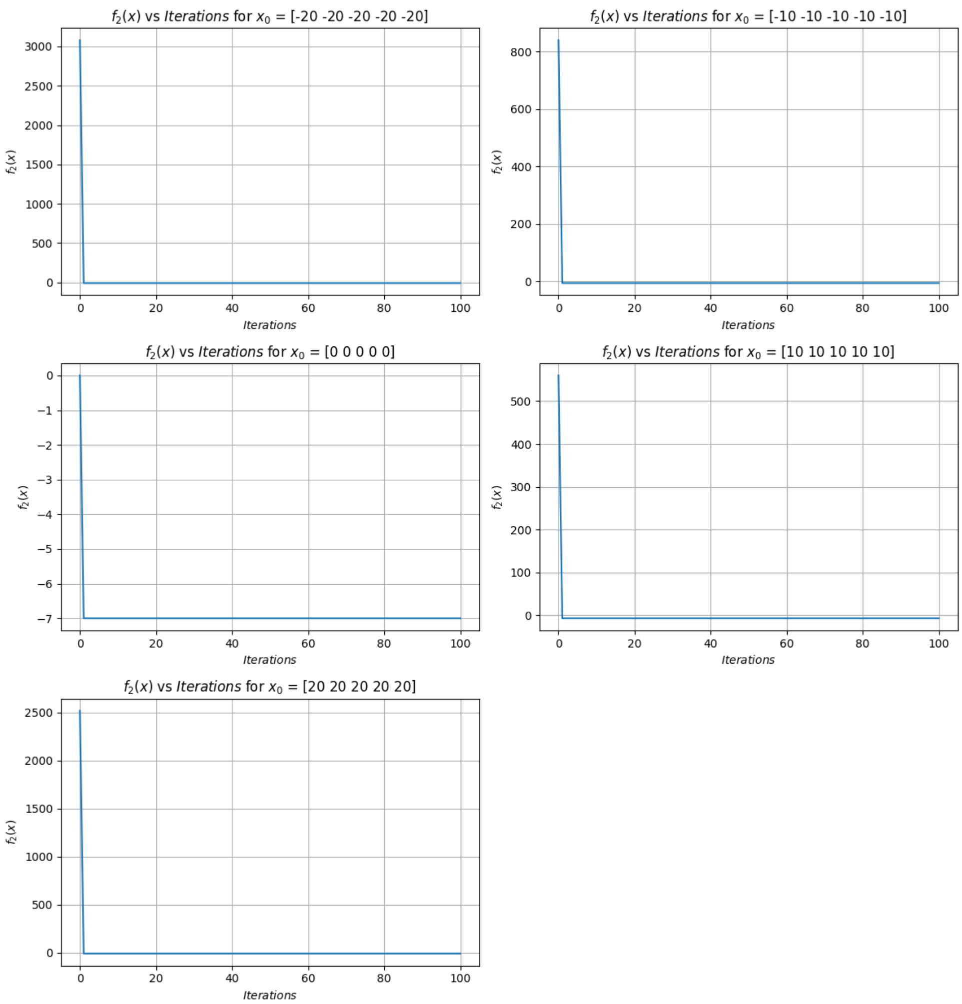


2. **Newton Method Implementation**
   Number of Iterations: 100.
   Initial Point, x_0: [0, 0, 0, 0, 0].
   Final Point, x_{100}: [1, 1, 1, 1, 1].
   Final Value of function, $f_2(x_{100})$: -7.0



   Comparison:
   From Graphs of Newton Method and Gradient Descent Method, it is easily observable that Newton method converges to optimal solution in single iteration but gradient descent converges for $\alpha = 0.1$ only, out of 5 different choices of α.

3. **Newton Method Implementation for different Initial Points**



Number of Iterations: 100.
Various Initial Points to be considered for different cases are as follows,
$\{[-20, -20, -20, -20, -20], [-10, -10, -10, -10, -10], [0, 0, 0, 0, 0], [10, 10, 10, 10, 10], [20, 20, 20, 20, 20]\}$.

Initial Point, x_0	x_{100}	$f_2(x_{100})$
$[-20, -20, -20, -20, -20]$	[1, 1, 1, 1, 1]	-7.0
$[-10, -10, -10, -10, -10]$	[1, 1, 1, 1, 1]	-7.0
[0, 0, 0, 0, 0]	[1, 1, 1, 1, 1]	-7.0
[10, 10, 10, 10, 10]	[1, 1, 1, 1, 1]	-7.0
[20, 20, 20, 20, 20]	[1, 1, 1, 1, 1]	-7.0

Observations:
The observation for Newton's method using 5 different initial points are as follows,

(i). **Single Iteration Convergence**: Newton method converges in single iteration for any choice of starting point.

(ii). **Obtained Global Minima**: As for every initial guess, the Newton method converges to same final point, hence, we can say that this point may be global minima for the given oracle.

4. **Observations Analysis**
According to above observations, the function in the oracle must be **Convex with Hessian matrix as Symmetric PD matrix**. Now, let us assume general cases,

(i). **Single Iteration Convergence**:
Let us consider $f(x)$ to be convex function and A be symmetric PD matrix to minimize the following,

$$f(x) = \frac{1}{2}x^T Ax - b^T x$$

Let x^* be the point where $f(x)$ takes minimum value. Then at x^*, $\nabla f(x^*) = 0$ i.e. $x^* = A^{-1}b$.
So, the Newton iterations becomes,

$$x_{i+1} = x_i - \left((\nabla^2 f(x_i))^{-1}\nabla f(x_i)\right)$$
$$x_{i+1} = x_i - A^{-1}(Ax_i - b) = x_i - A^{-1}(Ax_i - Ax^*)$$
$$x_{i+1} = x_i - x_i + x^* = x^*$$

That is, we can reach minima in **1** step using Newton method when hessian is symmetric PD matrix.

(ii). **Obtained Global Minima**:
Let us consider $f(x)$ to be convex function and A symmetric PD matrix to minimize the following,

$$f(x) = \frac{1}{2}x^T Ax - b^T x$$

For any x and x_0, Taylor expansion of $f(x)$ is as follows,

$$f(x) = f(x_0) + \nabla f(x_0)^T(x - x_0) + \frac{1}{2}(x - x_0)^T A(x - x_0) + O(\|x - x_0\|^2)$$

As A is symmetric PD matrix hence,

$$f(x) \geq f(x_0) + \nabla f(x_0)^T(x - x_0)$$

but for local minimum point $\hat{x}$ we have, $\nabla f(\hat{x}) = 0$, (substitute x_0 by $\hat{x}$)

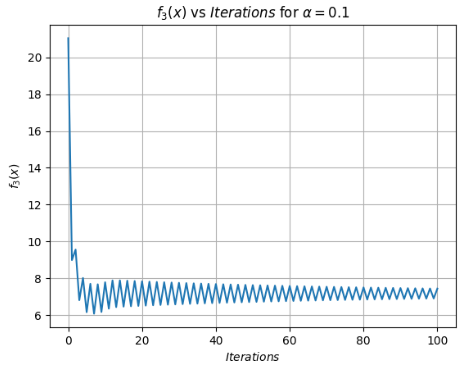
$$f(x) \geq f(\hat{x})$$

for every value of x. i.e. local minima $\hat{x}$ is the global minima.
Hence, $\hat{x}$ is the **global minima** of the function $f(x)$.

Name:	Aneesh Panchal	**Assignment No:**	Assignment 2
SR No:	06-18-01-10-12-24-1-25223	**Course Code:**	E0230
Email ID:	aneeshp@iisc.ac.in	**Course Name:**	Computational Methods of Optimization
Date:	October 20, 2024	**Term:**	AUG 2024

Solution 3

1. **Gradient Descent Implementation**
   Step Size, α: 0.1
   Number of iterations: 100.
   Initial Point, x_0: [1,1,1,1,1]
   Final Point, x_{100}: [-4.79441025e-01, 1.77602946e-01, 2.17408789e-17, 2.17408789e-17, 1.83807186e-17]
   Final Value of function, $f_3(x_{100})$: 7.439017189589639
   Best Value of function over iterations, $f_3(x)$: 6.075005868387142



2. **Oscillations**
   Yes, we observe oscillations in the implementation of the Gradient Descent algorithm. The reasons for the oscillations of gradient descent algorithm may be the **Assumption of large and constant step size** (α), due to which the final iterates oscillates around the optimal solution (Valley like structure).
   There are several ways to overcome these oscillations, some of which are,

   (i). Using Optimal (or, Low) learning rate (α) or, using Diminishing Gradient Descent algorithm.
   (ii). Using momentum term along with learning rate (α) which prevents avalanche effect in gradient descent.

3. **Newton Method Implementation**
   Number of iterations: 100.
   Initial Point, x_0: [1,1,1,1,1]

$Iteration$	Value of x	Value of $f_3(x)$
01.	[1, 1, 1, 1, 1]	21.03882095510255
02.	[-1.82388703e+06, -1.10032329e+03, -5.82247930, -5.82247930, -3.26188596e+01]	inf
03.	[inf, inf, 1.62776810e+09, 1.62776810e+09, 8.27871698e+83]	inf
04.	[nan, nan, nan, nan, nan]	nan
05.	[nan, nan, nan, nan, nan]	nan
06.	[nan, nan, nan, nan, nan]	nan
07.	[nan, nan, nan, nan, nan]	nan
08.	[nan, nan, nan, nan, nan]	nan
09.	[nan, nan, nan, nan, nan]	nan
10.	[nan, nan, nan, nan, nan]	nan

   No, the Newton algorithm did not converge, function value diverges to infinity.
   Possible reasons for divergence or, not convergence of Newton algorithm may be,

   (i). **Nearly Singular Hessian**:
   When function is having nearly constant gradient value, then the Hessian is very very small and hence, a very large inverse hessian is obtained, which leads to divergence of the iterates to infinity.
   (ii). **Ill conditioned Hessian**:
   When very small change in input of Hessian will lead to large change in output (i.e. Inverse of Hessian), which leads to very unstable solution or sometimes may lead the iterates to infinity.

Name:	Aneesh Panchal	**Assignment No:**	Assignment 2
SR No:	06-18-01-10-12-24-1-25223	**Course Code:**	E0230
Email ID:	aneeshp@iisc.ac.in	**Course Name:**	Computational Methods of Optimization
Date:	October 20, 2024	**Term:**	AUG 2024

4. **Gradient Descent and Newton Method − Game Cost Estimation**
Values of α and K taken into consideration are as follows,
α = {0.5, 0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001, 0.00005, 0.00001}
K = {0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100}
Number of Iterations: 100
x_0 = [1, 1, 1, 1, 1]
All the solutions that leads to nan or, inf values are neglected from the table.

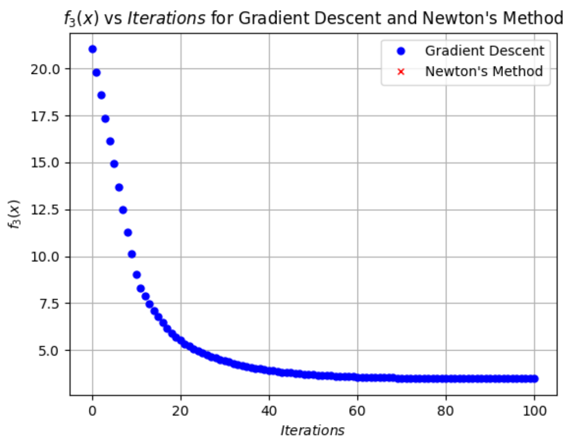
(i). **Least Observed Cost: 100**
$\alpha = 0.01$
$K = 100$
$\mathbf{f(x_{100}) = 3.4697}$
x_{100} = [-1.12433803e-18, 9.20676749e-12, 0.0314217271, 0.0314217271, 0.000299441658]

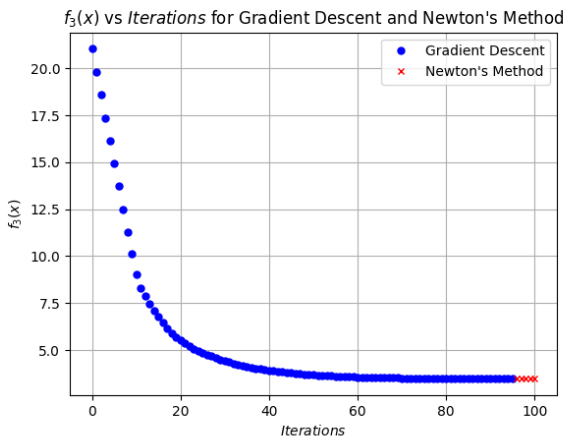
For this case priority of solution is according to the cost followed by function value.
[Solution is represented by Image 1 below, where all the 100 iterations are according to Gradient Descent.]

(ii). **Least Observed Value, $\mathbf{f(x_{100})}$: 3.4657**
There are 2 cases when Least value is observed with same cost,
$\alpha = 0.01$
$K = 95$
Cost = 220
x_{100} = [-1.12433803e-18, 9.91878741e-18, 8.94646618e-19, 8.94646618e-19, 4.20962946e-18]

$\alpha = 0.005$
$K = 95$
Cost = 220
x_{100} = [5.03817640e-18, 5.69283375e-19, -1.38135621e-17, -1.38135621e-17, 1.50502757e-17]

For this case priority of solution is according to the function value followed by cost.
[Solution is represented by Image 2 below, where 95 iterations are according to Gradient Descent and the remaining 5 are according to the Newton Method (using, $\alpha = 0.01$).]





Name:	Aneesh Panchal	**Assignment No:**	Assignment 2
SR No:	06-18-01-10-12-24-1-25223	**Course Code:**	E0230
Email ID:	aneeshp@iisc.ac.in	**Course Name:**	Computational Methods of Optimization
Date:	October 20, 2024	**Term:**	AUG 2024

α	**K**	$\mathbf{x_{100}}$	$\mathbf{f_3(x_{100})}$	**Cost**
0.5	100	[1.00000685, 1.01068444, -0.02771324, -0.02771324, 0.73162187]	17.6501	100
0.1	100	[-0.479441025, 0.177602946, 0, 0, 1.83807186e-17]	7.4390	100
0.05	100	[0.215959842, 0, 4.33173664e-10, 4.33173664e-10, 1.00121664e-17]	4.7365	100
0.01	30	[6.02675561e-19, 1.79791213e-18, 1.01932474e-18, 1.01932474e-18, 2.63811058e-18]	3.4657	1780
0.01	35	[-1.12433803e-18, 2.93458535e-18, 1.29189465e-17, 1.29189465e-17, -7.43243176e-18]	3.4657	1660
0.01	40	[-1.12433803e-18, -7.35569688e-19, 2.80738482e-19, 2.80738482e-19, -4.29154537e-18]	3.4657	1540
0.01	45	[-1.12433803e-18, -4.85036893e-19, 1.04850758e-17, 1.04850758e-17, 7.90098381e-19]	3.4657	1420
0.01	50	[-1.12433803e-18, -3.74049747e-18, -1.09910634e-17, -1.09910634e-17, 3.22088701e-18]	3.4657	1300
0.01	55	[-1.12433803e-18, 4.27446707e-18, 1.31595039e-17, 1.31595039e-17, -2.66532753e-18]	3.4657	1180
0.01	60	[-1.12433803e-18, 5.51138928e-18, 6.64582050e-18, 6.64582050e-18, 6.41994505e-18]	3.4657	1060
0.01	65	[-1.12433803e-18, 2.04089942e-18, 1.42971750e-17, 1.42971750e-17, 7.26344869e-18]	3.4657	940
0.01	70	[-1.12433803e-18, 4.41732976e-18, 7.55614269e-18, 7.55614269e-18, -2.53591076e-18]	3.4657	820
0.01	75	[-1.12433803e-18, -2.41782742e-18, 8.12155869e-18, 8.12155869e-18, 3.45303569e-18]	3.4657	700
0.01	80	[-1.12433803e-18, 5.77273426e-18, -2.20141133e-18, -2.20141133e-18, -6.16466609e-18]	3.4657	580
0.01	85	[-1.12433803e-18, -1.69554241e-19, 1.70803012e-17, 1.70803012e-17, 7.70261487e-18]	3.4657	460
0.01	90	[-1.12433803e-18, 4.42172409e-18, 2.17452983e-17, 2.17452983e-17, -1.52638191e-18]	3.4657	340
0.01	**95**	**[-1.12433803e-18, 9.91878741e-18, 8.94646618e-19, 8.94646618e-19, 4.20962946e-18]**	**3.4657**	**220**
0.01	**100**	**[-1.12433803e-18, 9.20676749e-12, 0.0314217271, 0.0314217271, 0.000299441658]**	**3.4697**	**100**
0.005	55	[1.00053285e-18, 6.32632255e-18, -7.91166375e-18, -7.91166375e-18, -2.44481943e-18]	3.4657	1180
0.005	60	[2.70182280e-18, 5.60227591e-19, 1.73617270e-17, 1.73617270e-17, 3.28657055e-19]	3.4657	1060
0.005	65	[1.15278026e-18, 1.53800007e-18, 6.34965114e-18, 6.34965114e-18, 3.39969457e-18]	3.4657	940
0.005	70	[5.40825287e-18, 4.87606427e-18, -9.82219405e-18, -9.82219405e-18, 3.85908212e-18]	3.4657	820
0.005	75	[-5.12938716e-19, 4.77910713e-18, 8.08492948e-19, 8.08492948e-19, 3.13995114e-18]	3.4657	700
0.005	80	[1.76918638e-18, -9.29687316e-19, 9.71655317e-18, 9.71655317e-18, 1.92551765e-18]	3.4657	580
0.005	85	[2.87940940e-18, -1.34995026e-18, 6.88180852e-18, 6.88180852e-18, 1.46976363e-17]	3.4657	460
0.005	90	[3.50568069e-19, -1.64374531e-18, 9.13200324e-18, 9.13200324e-18, 4.54390108e-18]	3.4657	340
0.005	**95**	**[5.03817640e-18, 5.69283375e-19, -1.38135621e-17, -1.38135621e-17, 1.50502757e-17]**	**3.4657**	**220**
0.005	100	[5.03817640e-18, 3.16882015e-05, 0.234771897, 0.234771897, 0.0351972225]	3.6841	100
0.001	100	[0.11410838, 0.50129288, 0.81063945, 0.81063945, 0.7040509]	9.1066	100
0.0005	100	[0.55000535, 0.75010019, 0.90435805, 0.90435805, 0.85119128]	14.9312	100
0.0001	100	[0.91000001, 0.95000588, 0.98074707, 0.98074707, 0.97016228]	19.8157	100
0.00005	100	[0.955, 0.97500258, 0.99036655, 0.99036655, 0.98507756]	20.4272	100
0.00001	100	[0.991, 0.99500047, 0.99807222, 0.99807222, 0.99701497]	20.9165	100

Solution 4

1. **Derivation**
Assume the Newton update as follows,
$$x_{i+1} = x_i - H(x_i)^{-1} g_i$$
where, $H(x_i)$ is the hessian of the objective function at $x = x_i$ and $g_i = \nabla f(x_i)$. Then we can assume Hessian as $H(x_i) = m_i I$, where I is the identity matrix and m_i is positive constant.
Hence, the updated expression will be,
$$x_{i+1} = x_i - \frac{g_i}{m_{i+1}}$$
Now, assume the notations,
   (i). $\gamma_i = g_{i+1} - g_i$
   (ii). $\delta_i = x_{i+1} - x_i$

From Taylor expansion we have,
$$g_{i+1} = g_i + m_{i+1}(x_{i+1} - x_i) + O(\|x_{i+1} - x_i\|)$$
Hence, the objective function becomes,
$$\arg\min_{m_{i+1}} \left\| \frac{\gamma_i}{m_{i+1}} - \delta_i \right\| \equiv \arg\min_{m_{i+1}} \|\gamma_i - m_{i+1}\delta_i\|$$
because $\arg\min_x(k(ax+b)) = \arg\min_x(ax+b) = \arg\min_x(ax)$, where k is positive constant.

   (i). **Case I**: (Say, NewUpdate1)
$$\arg\min_{m_{i+1}} \left\| \frac{\gamma_i}{m_{i+1}} - \delta_i \right\| \equiv \arg\min_{m_{i+1}} \left\| \frac{\gamma_i}{m_{i+1}} - \delta_i \right\|^2$$
Let $f(m_{i+1}) = \left\| \frac{\gamma_i}{m_{i+1}} - \delta_i \right\|^2$, then we have,
$$\nabla f(m_{i+1}) = 2\left(\frac{\gamma_i}{m_{i+1}} - \delta_i\right)^T \left(-\frac{\gamma_i}{m_{i+1}^2}\right) = 0$$
$$m_{i+1} = \frac{\gamma_i^T\gamma_i}{\delta_i^T\gamma_i}$$
Hence, final update expression for NewUpdate1 is as follows,
$$\mathbf{x_{i+1} = x_i - \frac{\gamma_i^T\gamma_i}{\delta_i^T\gamma_i} g_i}$$

   (ii). **Case II**: (Say, NewUpdate2)
$$\arg\min_{m_{i+1}} \|\gamma_i - m_{i+1}\delta_i\| \equiv \arg\min_{m_{i+1}} \|\gamma_i - m_{i+1}\delta_i\|^2$$
Let $f(m_{i+1}) = \|\gamma_i - m_{i+1}\delta_i\|^2$, then we have,
$$\nabla f(m_{i+1}) = 2(\gamma_i - m_{i+1}\delta_i)^T(-\delta_i) = 0$$
$$m_{i+1} = \frac{\delta_i^T\gamma_i}{\delta_i^T\delta_i}$$
Hence, final update expression for NewUpdate2 is as follows,
$$\mathbf{x_{i+1} = x_i - \frac{\delta_i^T\delta_i}{\delta_i^T\gamma_i} g_i}$$

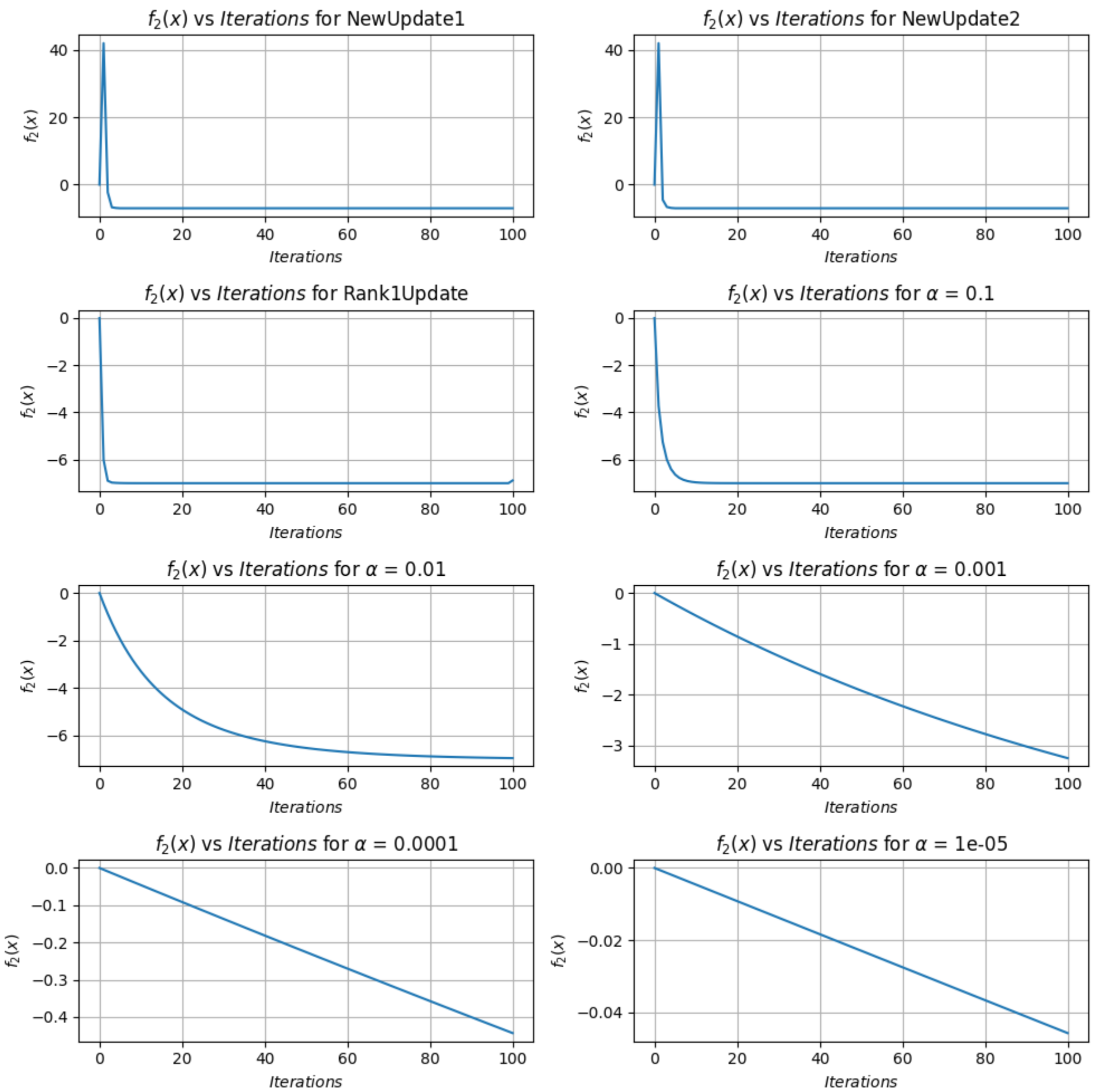
2. **Comparisons of Results**
   Assumptions for different parameters and implementations:

   (i). Quasi Newton Rank 1 update assumes Exact Line Search for α update.

   (ii). Initial Point, x_0: [0, 0, 0, 0, 0].

   (iii). Number of Iterations: 100.

   From Table below, it is clear that NewUpdate1 and NewUpdate2 provides the best results followed by, Gradient Descent ($\alpha = 0.1$), Gradient Descent ($\alpha = 0.01$) and Quasi Newton Rank 1 update.



$Methods$	x_{100}	$f_2(x_{100})$
New Update 1	[1, 1, 1, 1, 1]	-7
New Update 2	[1, 1, 1, 1, 1]	-7
Quasi Newton Rank 1 Update	[1, 1, 1, 1, 1.27586207]	-6.88585017835909
Gradient Descent (α = 1e-1)	[1, 1, 1, 1, 1]	-6.999999999999999
Gradient Descent (α = 1e-2)	[0.86738044, 0.99407947, 0.86738044, 0.86738044, 0.95244749]	-6.943756667002219
Gradient Descent (α = 1e-3)	[0.1814332, 0.39422956, 0.1814332, 0.1814332, 0.25951574]	-3.2499751965318433
Gradient Descent (α = 1e-4)	[0.01980329, 0.04878247, 0.01980329, 0.01980329, 0.02955883]	-0.44297215486087077
Gradient Descent (α = 1e-5)	[0.00199802, 0.00498765, 0.00199802, 0.00199802, 0.00299555]	-0.0458253740335511