Homework 5. Gradient Descent Method / Newton's Method

Hyuntaek Oh

ohhyun@oregonstate.edu

Due: May 14, 2025

ECE599/ AI539 Nonlinear Optimization (Spring 2025)
Homework 5. Gradient Descent Method / Newton's Method (Due: 11:59pm on May 14, Wednesday.)

Instruction: Students should provide enough detail of the logical procedure of deriving answers. Answers without sufficient justification will receive partial or no credit. You are welcome to use **Python** instead of MATLAB if you prefer.

Reading: Section 8.6-8.8 of the textbook (Luenberger and Ye).

1. (MATLAB / Python experiment) Consider the following nonlinear optimization problem:

$$\min_{x,y\in\mathbb{R}} f(x,y) = x^2 - 5xy + y^4 - 25x - 8y \tag{1}$$

   (a) Implement the Gradient Descent method with the *backtracking* line search and use it to find a local minimum point. Explain the parameters of the algorithm you used. Plot (i) $||\nabla f(\mathbf{x}_k)||$ versus k (use the log scale for the y-axis so that will be able to recognize the small difference such as the one between 10^{-4} and 10^{-6}), and (ii) $f(\mathbf{x}_k)$ versus k and provide interpretation. Check the Hessian at the algorithm output to verify whether the algorithm output is indeed a relative minimum point.

   (1) Explain the parameters of the algorithm you used.

   As can be seen in Figure 1, there are the parameters of the backtracking algorithm.

   - $\eta\ (= 2.0)$: *a step size reduction factor, and if the Armijo's condition is not satisfied, the current* α *is reduced by using* $\alpha \leftarrow \alpha/\eta$.
   - $\epsilon\ (= 1e^{-3} = 0.001)$: *Armijo sufficient decrease condition constant.*
   - $tol\ (= 1e^{-6} = 0.000001)$: *Convergence tolerance for gradient norm. The algorithm stops when the norm of the gradient is less than tolerance.*
   - $\alpha_{init}\ (= 1.0)$: *initial step size in backtracking line search. It is essential for quadratic convergence near the optimum.*
   - *max_iter* $(= 500)$: *Safety cap on the number of iterations.*
   - $x_0 = [4, 8]$: *Initial point. This point is chosen such that Hessian is invertible, direction is descent, and Backtracking can proceed normally.*

```
eta = 2.0;
epsilon = 1e-3;
tol = 1e-6;
alpha_init = 1.0;

max_iter = 500;
x_0 = [4; 8];   % x = [x, y]
x = x_0;

f_vals = zeros(max_iter, 1);
grad_norms = zeros(max_iter, 1);
```

Figure 1: Parameters Initialization of Backtracking

(2) Plot $||\nabla f(\mathbf{x}_k)||$ versus k (use the log scale for the y-axis so that will be able to recognize the small difference such as the one between 10^{-4} and 10^{-6}), and provide interpretation.

Figure 2 shows the graph of the norm of the gradient versus iteration k. The gradient norm $||\nabla f(\mathbf{x}_k)||$ decreases steadily in a nearly linear fashion on a log scale, indicating consistent progress toward optimality. Although some oscillations are observed early on, the norm ultimately drops below the tolerance threshold of 10^{-6}, confirming convergence. The initial convergence is relatively slow, which suggests that the backtracking line search frequently adjusted the step size. Quadratic convergence seems to emerge later in the iterations, rather than in the early stage.

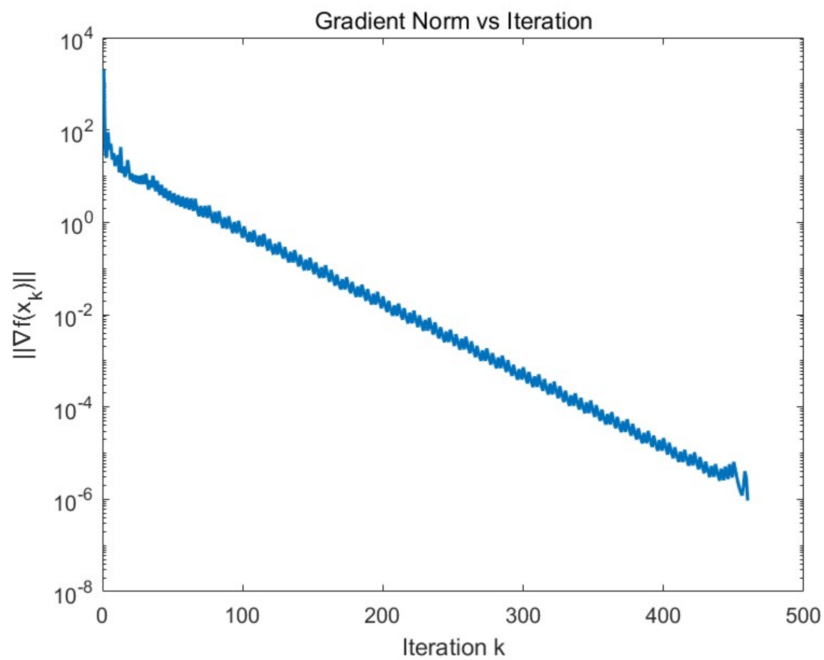

Figure 2: The graph of $||\nabla f(\mathbf{x}_k)||$ versus iteration k using Backtracking

(3) Plot $f(\mathbf{x}_k)$ versus k, and provide interpretation.

As can be seen in Figure 3, the function value $f(x_k)$ decreases sharply within the first few iterations and then flat, indicating that the algorithm rapidly approached a near-optimal region. After this rapid descent, the function value remains virtually constant, showing that further iterations only made minor refinements. This implies that while the step sizes became small due to backtracking, the descent directions were still effective. The function value stabilizes early, but the the gradient norm continues to decrease more gradually.

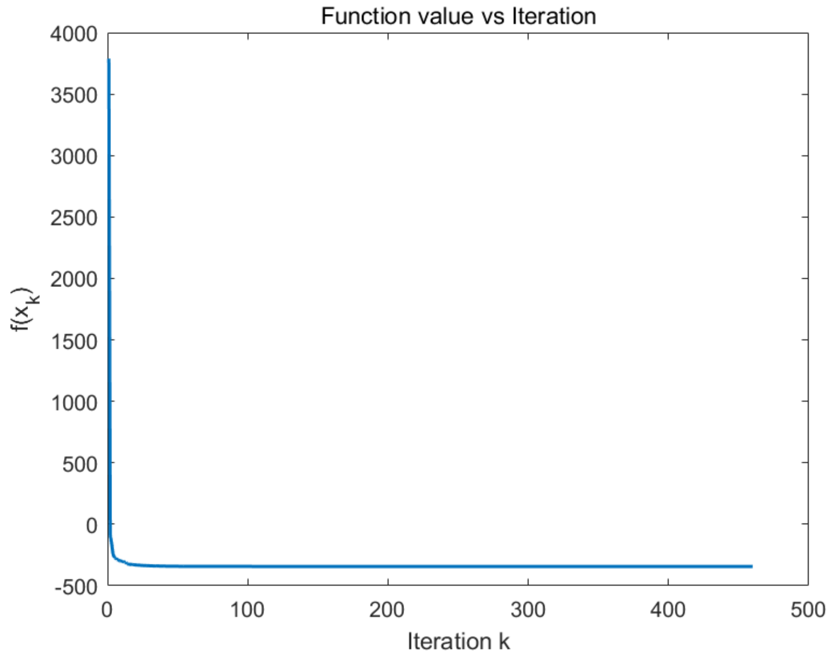

Figure 3: The graph of $f(\mathbf{x}_k)$ vs. iteration k using Backtracking

(4) Check the Hessian at the algorithm output to verify whether the algorithm output is indeed a relative minimum point.

As shown in Figure 4, the point $x^* = (20, 3)$ is a local minimum of the given function. By using this point, the gradient of the function $f(x, y)$ is:

$$\nabla f_1 = 2x - 5y - 25 = 2(20) - 5(3) - 25 = 40 - 15 - 25 = 0$$
$$\nabla f_1 = -5x + 4y^3 - 8 = -5(20) + 4(3)^3 - 8 = -100 + 108 - 8 = 0$$
$$\nabla f(x^*) = [0, 0]^T$$

This satisfies the first order necessary condition. Then, by computing the Hessian, its positive definiteness is verified. It is positive definite if the first element is greater than zero, and determinant of it is also greater than zero. Firstly, the Hessian matrix is:

$$H(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

The second derivatives are:

$$\frac{\partial^2 f}{\partial x^2} = 2, \frac{\partial^2 f}{\partial x \partial y} = -5, \frac{\partial^2 f}{\partial y^2} = 12y^2$$

So, the Hessian matrix is:

$$H(x, y) = \begin{bmatrix} 2 & -5 \\ -5 & 12y^2 \end{bmatrix}$$

As can be seen in Figure 4, a relative minimum point (x^*, y^*) is (20, 3). Applying it to the Hessian matrix:

$$H(x, y) = \begin{bmatrix} 2 & -5 \\ -5 & 12 \cdot (3)^2 \end{bmatrix} = \begin{bmatrix} 2 & -5 \\ -5 & 108 \end{bmatrix}$$

To verify that x^* is a local minimum, the Hessian at that point must be positive definite. The leading minor $H_{11} = 2 > 0$, and the determinant of the matrix is:

$$det(H) = 2 * 108 - (-5)^2 = 216 - 25 = 191 > 0$$

Thus, the point x^* is a strict local minimum since the Hessian matrix is positive definite.

```
>> HW5_1_Backtracking
Initial point x_0: [4.0, 8.0]
Initial alpha value: 1.00
Minimum point x*: (20.0000, 3.0000)
Final alpha value: 1.00e+00
Gradient norm at solution: 9.39e-07
Eigenvalues of Hessian: 1.7647, 108.2353
x* is a local minimum since Hessian is positive definite
```

Figure 4: Results of Backtracking algorithm

(b) Implement Gradient Descent with the *Goldstein rule* for line search, and use it to find a local minimum point. Explain the parameters of the algorithm you used. Plot (i) $||\nabla f(\mathbf{x}_k)||$ versus k, and (ii) $f(\mathbf{x}_k)$ versus k and provide interpretation. Check the Hessian at the algorithm output to verify whether the algorithm output is indeed a local minimum point.

(1) Explain the parameters of the algorithm you used.

Figure 5 shows parameter initialization of Goldstein rule. The parameters used in the Goldstein rule are the same as those used in the Backtracking.

- η $(= 2.0)$*: a step size reduction factor, and if the Armijo's condition is not satisfied, the current α is reduced by using $\alpha \leftarrow \alpha/\eta$.*
- ϵ $(= 1e^{-3} = 0.001)$*: Armijo sufficient decrease condition constant.*

- *tol* $(= 1e^{-6} = 0.000001)$*: Convergence tolerance for gradient norm. The algorithm stops when the norm of the gradient is less than tolerance.*
- α_{init} $(= 1.0)$*: initial step size in backtracking line search. It is essential for quadratic convergence near the optimum.*
- *max_iter* $(= 500)$*: The maximum number of iterations.*
- $x_0 = [4, 8]$*: Initial point. This point is chosen such that Hessian is invertible, direction is descent, and Backtracking can proceed normally.*

```
eta = 2.0;
epsilon = 1e-3;
tol = 1e-6;
alpha_init = 1.0;

max_iter = 500;
x_0 = [4; 8];    % x = [x, y]
x = x_0;

f_vals = zeros(max_iter, 1);
grad_norms = zeros(max_iter, 1);
```

Figure 5: Parameters Initialization of Goldstein rule

(2) Plot $||\nabla f(\mathbf{x}_k)||$ *versus* k *(use the log scale for the* y*-axis so that will be able to recognize the small difference such as the one between* 10^{-4} *and* 10^{-6}*), and provide interpretation.*

As can be seen in Figure 6, the gradient norm steadily decreases on a log scale. This graph exhibits small oscillations, especially in the early and mid-phase of iterations. Despite these fluctuations, the gradient norm eventually drops below the convergence threshold (tol) around 460 iteration.

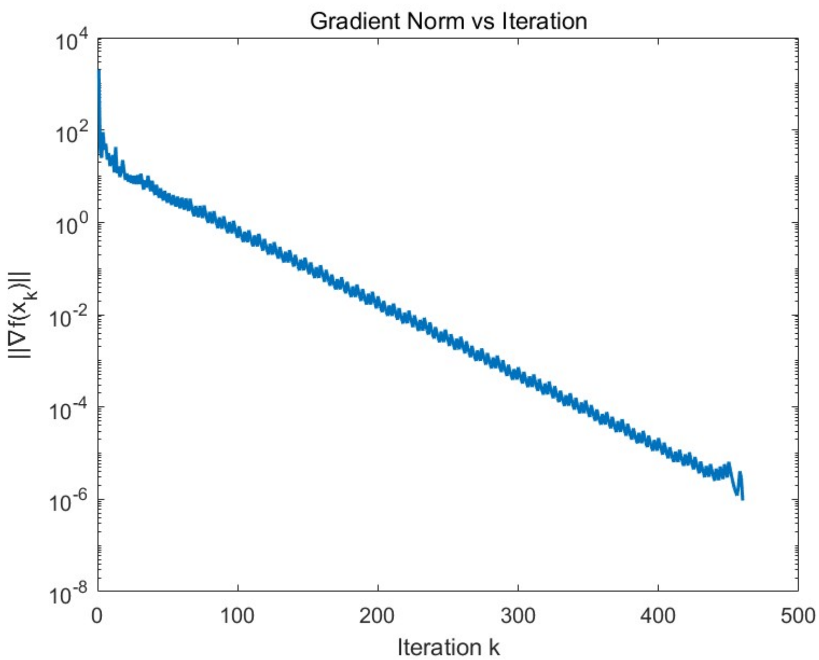

Figure 6: The graph of $||\nabla f(\mathbf{x}_k)||$ versus iteration k using Goldstein rule

(3) Plot $f(\mathbf{x}_k)$ versus k, and provide interpretation.

As shown in Figure 7, the graph shows a rapid decreases in the function value during the initial iterations. After the steep descent, the function value stabilizes and remains nearly constant, demonstrating that the iterates have approached a minimum. This flat suggests that most of the meaningful progress occurred early. The Goldstein rule successfully maintained descent.

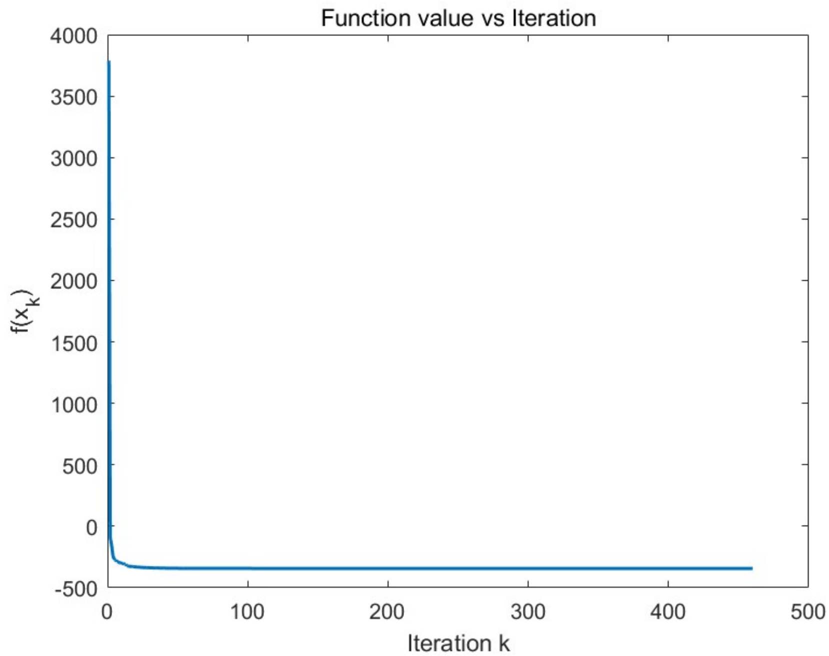

Figure 7: The graph of $f(\mathbf{x}_k)$ versus iteration k using Goldstein rule

(4) Check the Hessian at the algorithm output to verify whether the algorithm output is indeed a relative minimum point.

Similarly, the first order condition is satisfied:

$$||\nabla f(x^*)|| = 9.39 \times 10^{-7}$$

Since this value is quite close to zero and below the specified convergence tolerance (e.g. 10^{-6}), it confirms that the point x^* satisfies the first order condition:

$$\nabla f(x^*) \approx 0$$

The problem setting used in Goldstein rule is same as those used in Backtracking, so Hessian matrix is also same. According to Figure 8, the eigenvalues of the Hessian matrix at x^* are:

$$\lambda_1 = 1.7647, \quad \lambda_2 = 108.2353$$

Since both eigenvalues are strictly positive, the Hessian is positive definite. This satisfies the second order sufficient condition for a strict local minimum.
Thus, the point $x^* = (20, 3)$ is a strict local minimum of the objective function.

```
>> HW5_1_Goldstein_rule
Initial point x_0: [4.0, 8.0]
Initial alpha value: 1.00
Minimum point x*: (20.0000, 3.0000)
Final alpha value: 1.00e+00
Gradient norm at solution: 9.39e-07
Eigenvalues of Hessian: 1.7647, 108.2353
x* is a local minimum since Hessian is positive definite
```

Figure 8: Results of Goldstein rule

2. (MATLAB/Python experiment) Consider the following nonlinear optimization problem:

$$\min_{x,y\in\mathbb{R}} f(x,y) = x^2 - 5xy + y^4 - 25x - 8y \tag{2}$$

In the lecture, we discussed how we can modify Newton's method to ensure global convergence. Read Section 8.8 for more details, especially the part on the modification of Newton's method to guarantee global convergence. Implement a modification of Newton's method to ensure both global convergence and quadratic local convergence. Explain the modifications you made. Plot (i) $||\nabla f(\mathbf{x}_k)||$ versus k, and (ii) $f(\mathbf{x}_k)$ versus k and provide interpretation. Check whether the algorithm output satisfies the first and second order necessary conditions for local minimum points.

(1) Explain the modifications you made

As can be seen in Figure 9, the parameters using the modified Newton's method:

- η $(= 2.0)$*: a step size reduction factor, and if the Armijo's condition is not satisfied, the current* α *is reduced by using* $\alpha \leftarrow \alpha/\eta$.
- *bt_epsilon:* ϵ $(= 1e^{-3} = 0.001)$*: Armijo sufficient decrease condition constant.*
- *delta:* $\delta =$*1e-4: Minimum eigenvalue threshold.*
- *tol* $(= 1e^{-6} = 0.000001)$*: Convergence tolerance for gradient norm. The algorithm stops when the norm of the gradient is less than tolerance.*
- α_{init} $(= 1.0)$*: initial step size in backtracking line search. It is essential for quadratic convergence near the optimum.*
- *max_iter* $(= 500)$*: Safety cap on the number of iterations.*

- $x_0 = [4, 8]$: *Initial point. This point is chosen such that Hessian is invertible, direction is descent, and Backtracking can proceed normally.*

To ensure both global convergence and local quadratic convergence, a modified Newton's method was implemented, incorporating two enhancements: Ensuring a descent direction (Figure 10) and backtracking line search (Figure 11). At each iteration, Hessian matrix H *was computed and evaluated its smallest eigenvalue. If the minimum eigenvalue* $lambda_{min}$ *is less than a threshold* $\delta > 0$, $\epsilon_k \boldsymbol{I}$ *is added to the Hessian, where* $\epsilon_k = \delta - \lambda_{min}(\boldsymbol{F}(x_k))$. *This guarantees that the modified Hessian* H_mod *is positive definite and produces a descent direction* d. *Then, backtracking line search is applied with using the Armijo's condition to determine a suitable step size* $alpha$, *guaranteeing sufficient decrease in the objective function. These modifications allow the method to converge globally from arbitrary starting points while preserving the local quadratic convergence.*

```
eta = 2.0;
bt_epsilon = 1e-3;
delta = 1e-4;
alpha_init = 1.0;

max_iter = 500;
tol = 1e-6;
x_0 = [4; 8];   % x = [x, y]
x = x_0;

f_vals = zeros(max_iter, 1);
grad_norms = zeros(max_iter, 1);
```

Figure 9: Modified Newton's Method Parameters Initialization

```
for k = 1: max_iter
    g = grad_f(x);
    H = Hessian_f(x);

    % Ensuring a descent direction
    lambda_min = min(eig(H));
    if lambda_min < delta
        epsilon_k = delta - lambda_min;
    else
        epsilon_k = 0;
    end

    H_mod = epsilon_k * eye(2) + H;

    d = -H_mod \ g;
```

Figure 10: Modified Newton's Method - Ensuring a descent direction

```
% Backtracking line search
alpha = alpha_init;
while true
    if f(x + alpha * d) <= (f(x) + bt_epsilon * alpha * g' * d)
        break;
    else
        alpha = alpha / eta;
    end
end
```

Figure 11: Modified Newton's Method - Backtracking line search

(2) Plot $||\nabla f(\mathbf{x}_k)||$ versus k, and provide interpretation.

Figure 12 shows the convergence behavior of the modified Newton's method by plotting $||\nabla f(\mathbf{x}_k)||$, the norm of the gradient at each iteration, on a logarithmic scale.

At initial iteration $k = 1$ to 3, the gradient norm decreases steadily, indicating global convergence.

At mid iteration $k = 4$ to 5, the rate of decrease accelerates, suggesting that the algorithm has entered the local convergence region near the minimum. This is where the quadratic convergence property of Newton's method becomes obvious.

At final iterations $k = 6$ to 7, the gradient norm drops dramatically to below 10^{-6}. This rapid decrease is a typical property of Newton's method when Hessian is well-conditioned and the iterate is close to the solution.

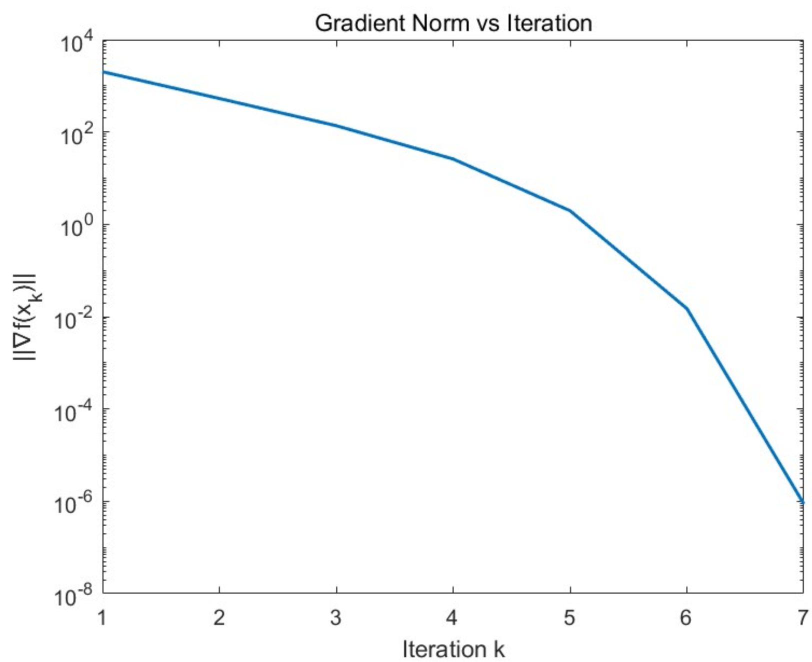

Figure 12: The graph of $||\nabla f(\mathbf{x}_k)||$ versus iteration k using the modified Newton's Method

(3) Plot $f(\mathbf{x}_k)$ versus k, and provide interpretation.

As shown in Figure 13, the plot illustrates the decrease in the objective function $f(x_k)$ at each iteration of the modified Newton's method.

At initial iteration $k = 1$ to 2, a very large drop in function value is observed from around 3800 to near 100. This implies that the initial step moved the iterate significantly closer to the region containing the local minimum.

At mid iteration $k = 2$ to 4, the function value continues to decrease steadily, approaching negative values. This phase reflects a refinement stage, where the algorithm is adjusting direction and step size to navigate a more curved region of the function.

At final iterations $k = 4$ to 7, the function value is flat near the minimum around -450, showing only marginal decreases. This plateau is consistent with the algorithm approaching the optimal point. The behavior indicates convergence toward a stationary point.

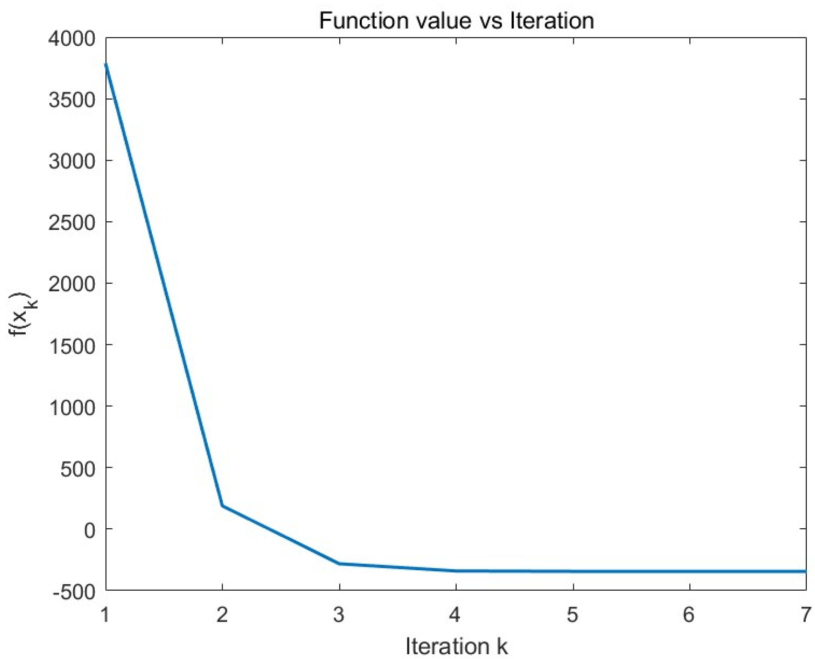

Figure 13: The graph of $f(\mathbf{x}_k)$ versus k using the modified Newton's Method

(4) Check whether the algorithm output satisfies the first and second order necessary conditions for local minimum points.

Figure 14 shows the solution obtained by the modified Newton's method is $x^* = (20.0, 3.0)$. To verify that x^* is a strict local minimum, the first- and second-order necessary conditions should be checked.

A point $x^* \in \mathbb{R}^n$ is a stationary point of a differentiable function f if:

$$\nabla f(x^*) = 0$$

From the algorithm output:

$$\nabla f(x^*) = \begin{bmatrix} 2x^* - 5y^* - 25 \\ -5x^* + 4(y^*)^3 - 8 \end{bmatrix} = \begin{bmatrix} 2(20) - 5(3) - 25 \\ -5(20) + 4(3)^3 - 8 \end{bmatrix} = \begin{bmatrix} 40 - 15 - 25 \\ -100 + 108 - 8 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Thus, the first-order necessary condition is satisfied.

The second-order necessary condition for a local minimum is:

$$\nabla^2 f(x^*) \geq 0 \text{ (i.e., positive semidefinite)}$$

The Hessian is:

$$\nabla^2 f(x^*) = \begin{bmatrix} 2 & -5 \\ -5 & 12y^2 \end{bmatrix} = \begin{bmatrix} 2 & -5 \\ -5 & 12(3)^2 \end{bmatrix} = \begin{bmatrix} 2 & -5 \\ -5 & 108 \end{bmatrix}$$

As can be seen in Figure 14, to check definiteness, the eigenvalues are:

$$\lambda_1 \approx 1.7647 \quad \lambda_2 \approx 108.2353$$

Since both $\lambda_1, \lambda_2 > 0$, the matrix is positive definite:

$$\nabla^2 f(x^*) > 0$$

Therefore, the second-order necessary condition is satisfied, and even the second-order sufficient condition holds, meaning that x^* is a strict local minimizer of the function f can be concluded.

```
>> HW5_2_Newton_method
Initial point x_0: [4.0, 8.0]
Initial alpha value: 1.00
Minimum point x*: (20.0000, 3.0000)
Final alpha value: 1.00
Gradient norm at solution: 0.00
Eigenvalues of Hessian: 1.7647, 108.2353
x* is a strict local minimum (gradient zero && Hessian positive definite)
```

Figure 14: Results of the modified Newton's Method