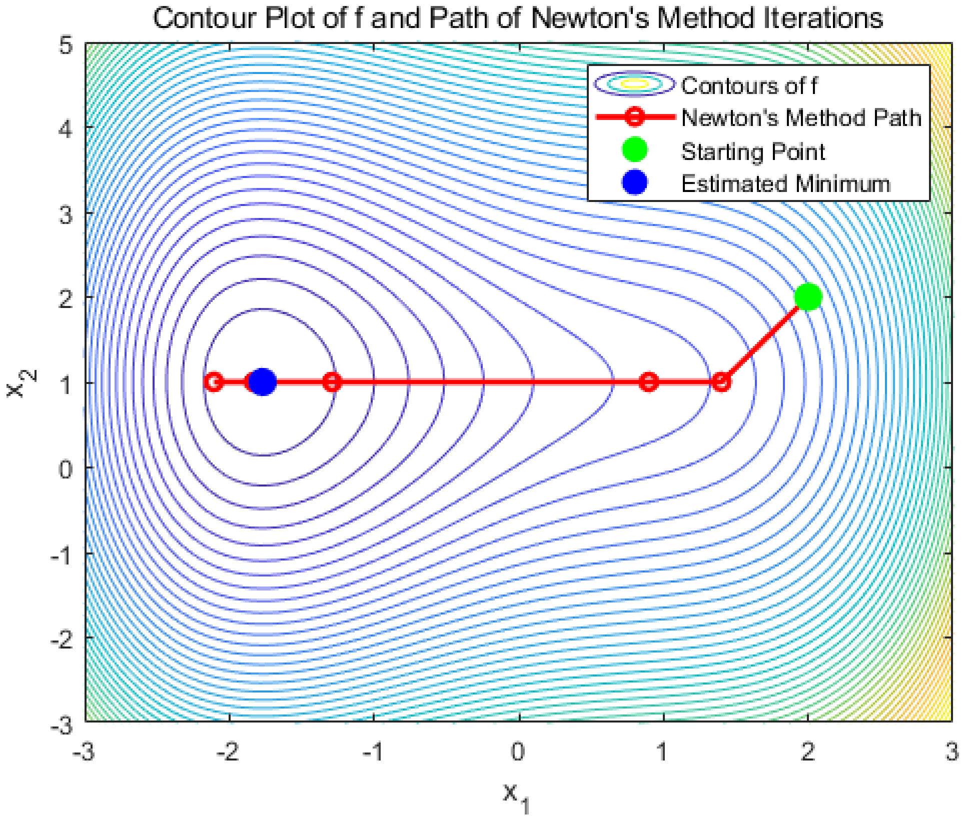
```
% Homework3 Problem3(b)
% Newton's method for minimizing the function f

% Clear workspace and command window
clear; clc;

% Define the function f as an anonymous function
f = @(x) (x(1).^4)/4 - x(1).^2 + 2*x(1) + (x(2) - 1).^2;

% Define the gradient of f as an anonymous function
grad_f = @(x) [ x(1).^3 - 2*x(1) + 2 ; 2*(x(2) - 1) ];

% Define the Hessian matrix of f as an anonymous function
Hessian_f = @(x) [ 3*x(1).^2 - 2, 0 ; 0, 2 ];

% Initial guess for the minimizer
x_initial = [2 ; 2]; % Starting point x0

% Set tolerance for the gradient norm
tolerance = 1e-6;

% Set maximum number of iterations
max_iterations = 100;

% Iteration counter
iteration = 0;

% Initialize a matrix to store all iterates for plotting
iterate_history = x_initial;

% Newton's method
while norm(grad_f(x_initial)) > tolerance && iteration < max_iterations

    % Compute the gradient at the current point
    gradient_current = grad_f(x_initial);

    % Compute the Hessian matrix at the current point
    Hessian_current = Hessian_f(x_initial);

    % Check if the Hessian is positive definite
    eigenvalues = eig(Hessian_current);
    if any(eigenvalues <= 0)
        error('Hessian is not positive definite at iteration %d', iteration);
    end

    % Update the current point
    x_initial = x_initial - inv(Hessian_current) * gradient_current;

    % Store the new iterate
    iterate_history = [iterate_history, x_initial];

    % Increment iteration counter
    iteration = iteration + 1;

end

% Display the number of iterations taken
fprintf('Number of iterations: %d\n', iteration);

% Plotting section

% Create a grid of values for x1 and x2 for contour plotting
x1_values = linspace(-3, 3, 100);
x2_values = linspace(-3, 5, 100);
[X1_grid, X2_grid] = meshgrid(x1_values, x2_values);

% Compute the function values over the grid for contour plotting
F_values = (X1_grid.^4)/4 - X1_grid.^2 + 2*X1_grid + (X2_grid - 1).^2;

% Generate contour plot of the function f
figure;
contour_levels = 50; % Number of contour levels
contour(X1_grid, X2_grid, F_values, contour_levels); % Plot contours
hold on; % Retain current plot when adding new plots

% Plot the path traced by the Newton's method iterations
plot(iterate_history(1,:), iterate_history(2,:), 'ro-', 'LineWidth', 2);

% Mark the starting point
plot(iterate_history(1,1), iterate_history(2,1), 'go', 'MarkerFaceColor',
'g', 'MarkerSize', 10);

% Mark the ending point (estimated minimum)
plot(iterate_history(1,end), iterate_history(2,end), 'bo',
'MarkerFaceColor', 'b', 'MarkerSize', 10);

% Add labels and title to the plot
xlabel('x_1');
ylabel('x_2');
title('Contour Plot of f and Path of Newton''s Method Iterations');

% Add legend to the plot
legend('Contours of f', 'Newton''s Method Path', 'Starting Point',
'Estimated Minimum');

% Release the hold on the current plot
hold off;

% Display the estimated minimum point and function value at that point
x_min = x_initial;
f_min = f(x_min);
fprintf('Estimated minimum at x = [%f, %f], f(x) = %f\n', x_min(1),
x_min(2), f_min);
```

```
Number of iterations: 8
Estimated minimum at x = [-1.769292, 1.000000], f(x) = -4.219136
```



Published with MATLAB® R2023b