

```
%%%%%%%%%%%% Gradient Descent Implementation %%%%%%%%%%%%%%
%%%%%%%%%%%% By Max Yi Ren and Emrah Bayrak %%%%%%%%%%%%%%
```

```
function solution = gradient(f,g,H,x0,opt)
% Set initial conditions
x = x0; % Set current solution to the initial guess
iter = 0; % Set iteration counter to 0

% Initialize a structure to record search process
solution = [];

% Calculate the norm of the gradient
gnorm = norm(g(x),2); % this needs to be a scalar

% Set the termination criterion:
while gnorm>opt.eps % if not terminated
    iter = iter + 1

    % save current step
    solution.x([1,2],iter) = x;
    % solution.x is an array of solutions, i.e., a matrix
    % opt.linesearch switches line search on or off.
    % You can first set the variable "a" to different constant values and see how
    it % affects the convergence.
    if opt.linesearch
        a = lineSearch1(f,g,H,x,opt);
    else
        a = 0.001;
    end

    % Gradient descent:
    d = -1*g(x);
    x = x + a*d; % update x based on gradient info
    % Update termination criterion:
    gnorm = norm(g(x),2); % update the norm of gradient
end
disp(x);
```