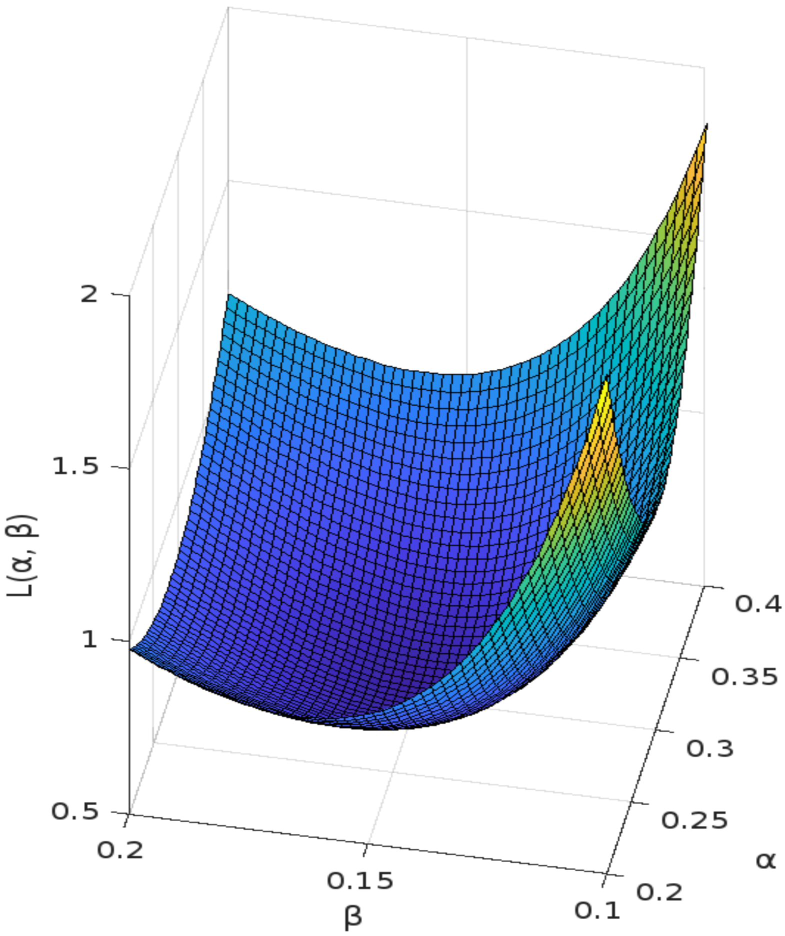
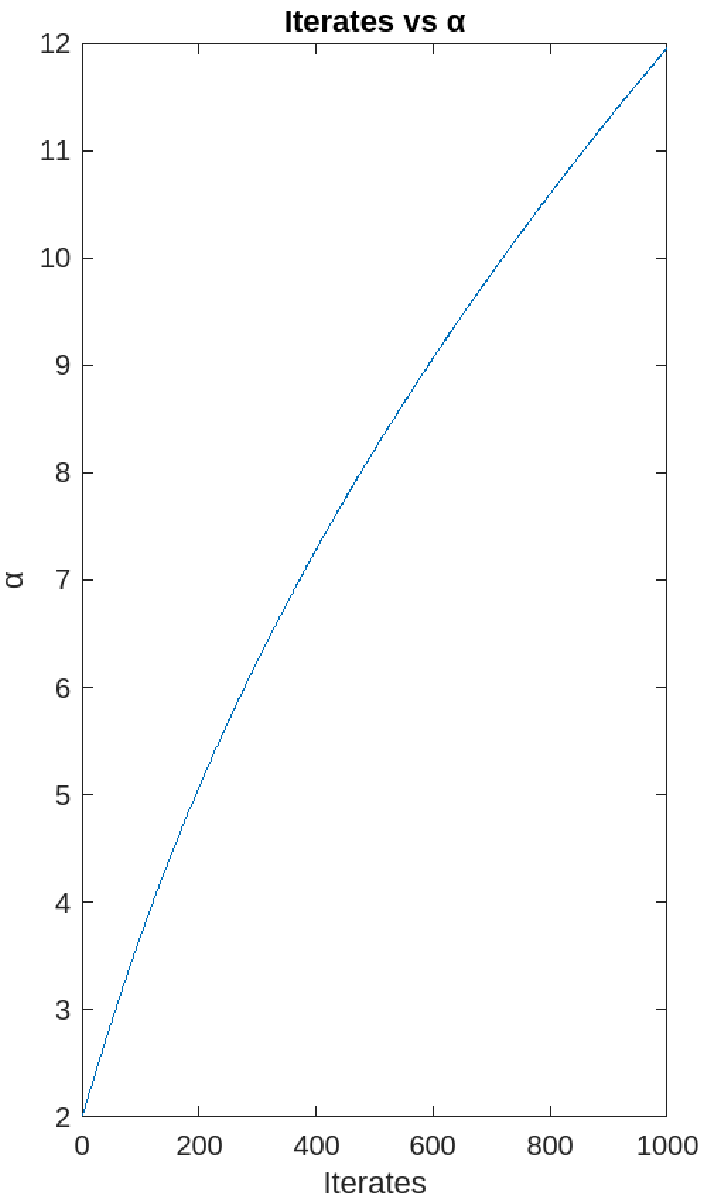
(c) I used MATLAB with sample code provided on Canvas to do the homework.

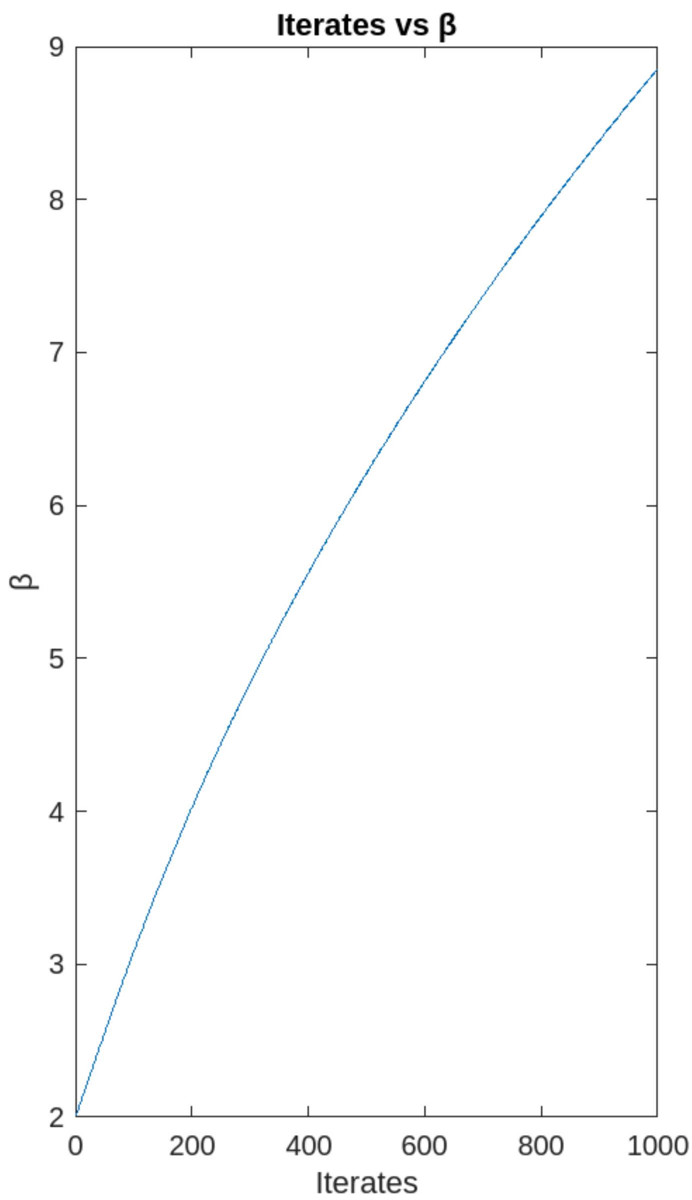
```
close all;
clear all;
x = [.30753, .56678, -.25177, .37243, .26375];
n = 5;
% create a grid of points for α, β
a = linspace(.2,.4,50);
b = linspace(.1,.2,50);
[alpha_grid, beta_grid] = meshgrid(a, b);
L = zeros(1, n);
disp(size(alpha_grid, 1));
disp(size(alpha_grid, 2));
% along with row number
for i = 1:size(alpha_grid, 1)
   % along with column number
   for j = 1:size(alpha_grid, 2)
       alpha = alpha_grid(i, j);
       beta = beta_grid(i, j);
       % calculate f(x|α,β) for each x value
       f_x = exp(-(x - alpha) ./ beta) ./ (beta * (1 + exp(-(x - alpha) ./ beta)).^2);
       % calculate L(α, β)
       L(i, j) = -log(prod(f_x));
   end
end
subplot(121);
surf(alpha_grid, beta_grid, L);
xlabel('α');
ylabel('β');
zlabel('L(α, β)');
```



(d)

```
n = 5;
x = [0.30753, 0.56678, -0.25177, 0.37243, 0.26375];
eta = 0.001;
iterates = 1000;
p = [2.0; 2.0];

% Gradient
gradient = @(alpha, beta) [
   -n/beta + 2*n/beta^2 * sum((x - alpha) .* exp(-(x - alpha) / beta) ./ (1 + exp(-(x -
alpha) / beta)));
   -1/beta^2 * sum(x - alpha) + n/beta + 2*n/beta^2 * sum(exp(-(x - alpha) / beta) .* (x
- alpha) ./ (1 + exp(-(x - alpha) / beta)))
];

% 1000 loops
alpha_values = zeros(iterates, 1);
beta_values = zeros(iterates, 1);
for i = 1:iterates
   grad = gradient(p(1), p(2));
   p = p - eta * grad;
   alpha_values(i) = p(1);
   beta_values(i) = p(2);
end

% final estimates of alpha and beta
fprintf('estimate of α: %.3f\n', p(1));
fprintf('estimate of β: %.3f\n', p(2));

% Plot
figure;
subplot(1, 2, 1);
plot(1:iterates, alpha_values);
xlabel('Iterates');
ylabel('α');
title('Iterates vs α');
subplot(1, 2, 2);
plot(1:iterates, beta_values);
xlabel('Iterates');
ylabel('β');
title('Iterates vs β');
```

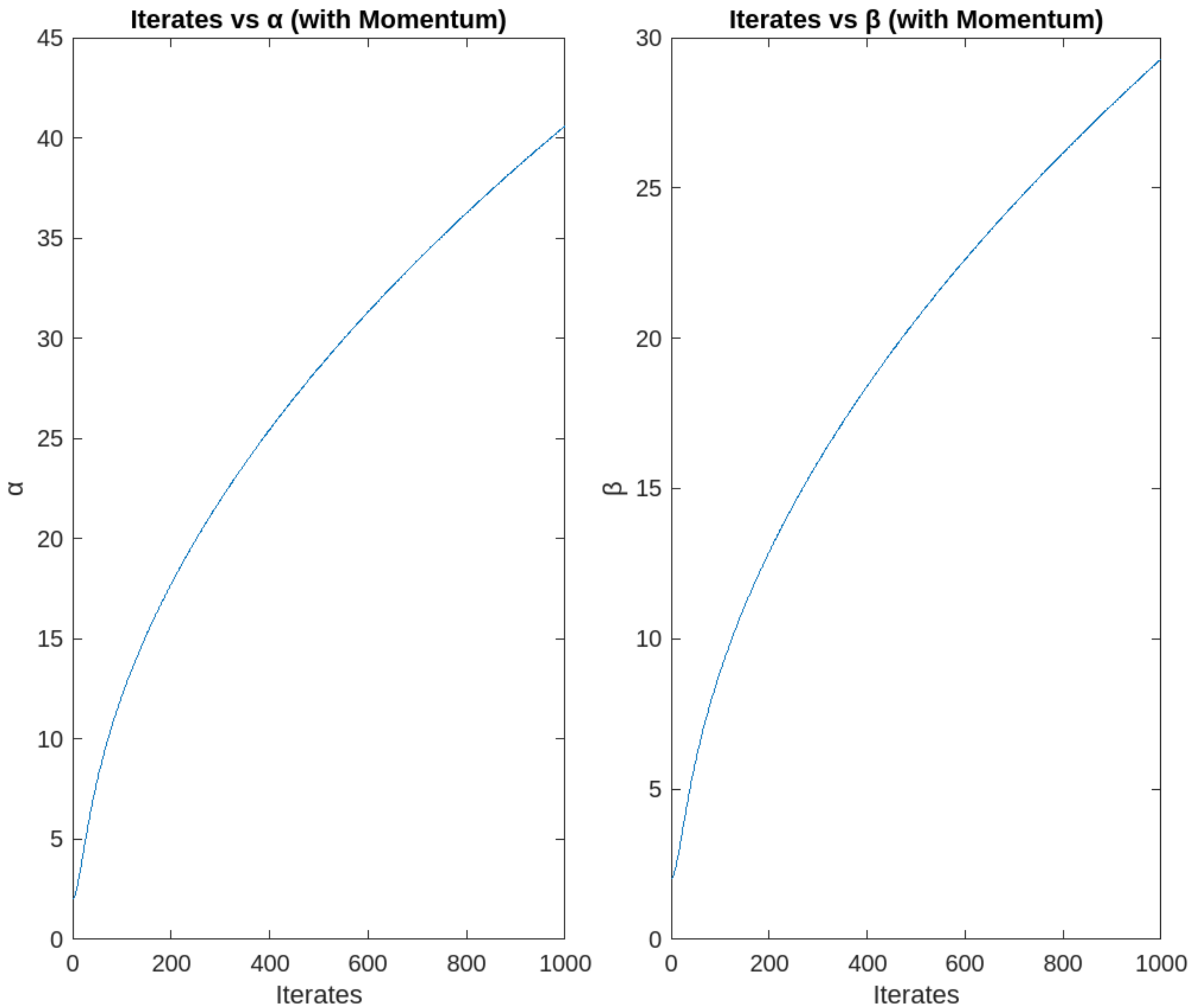





```
The estimate of α: 11.958
The estimate of β: 8.857
```

(e)

```
n = 5;
x = [.30753, .56678, -.25177, .37243, .26375];
eta = .001;
gamma = .9;
iterates = 1000;
p = [2.0; 2.0];
% Gradient
gradient = @(alpha, beta) [
   -n/beta + 2*n/beta^2 * sum((x - alpha) .* exp(-(x - alpha) / beta) ./ (1 +
exp(-(x - alpha) / beta)));
   -1/beta^2 * sum(x - alpha) + n/beta + 2*n/beta^2 * sum(exp(-(x - alpha) / beta)
.* (x - alpha) ./ (1 + exp(-(x - alpha) / beta)))
];
theta = [0;0];
% 1000 loops
alpha_values = zeros(iterates, 1);
beta_values = zeros(iterates, 1);
for i = 1:iterates
   grad = gradient(p(1), p(2));
   theta = gamma * theta + eta * grad;
   p = p - theta;
   alpha_values(i) = p(1);
   beta_values(i) = p(2);
end

% final estimates of alpha and beta
fprintf('estimate of α: %.3f\n', p(1));
fprintf('estimate of β: %.3f\n', p(2));

% Plot iterative values of alpha and beta
figure;
subplot(1, 2, 1);
plot(1:iterates, alpha_values);
xlabel('Iterates');
ylabel('α');
title('Iterates vs α (with Momentum)');
subplot(1, 2, 2);
plot(1:iterates, beta_values);
xlabel('Iterates');
ylabel('β');
title('Iterates vs β (with Momentum)');
```




```
The estimate of α with momentum: 40.606
The estimate of β with momentum: 29.296
```

From my plots, there seems no explicit difference. However, the momentum should be faster.