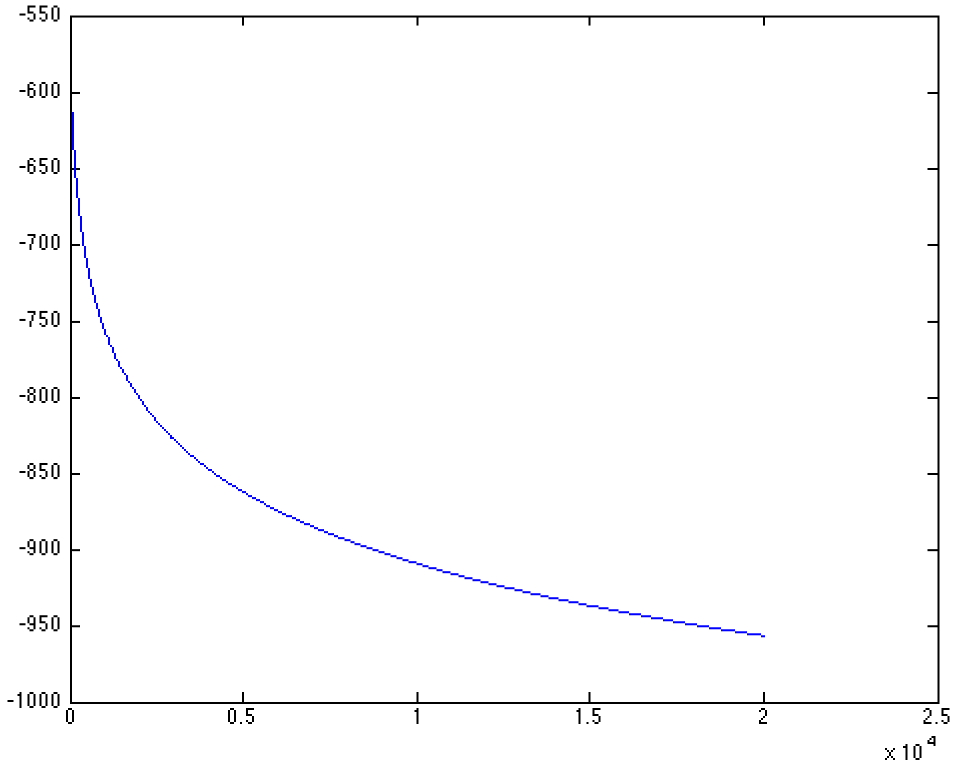
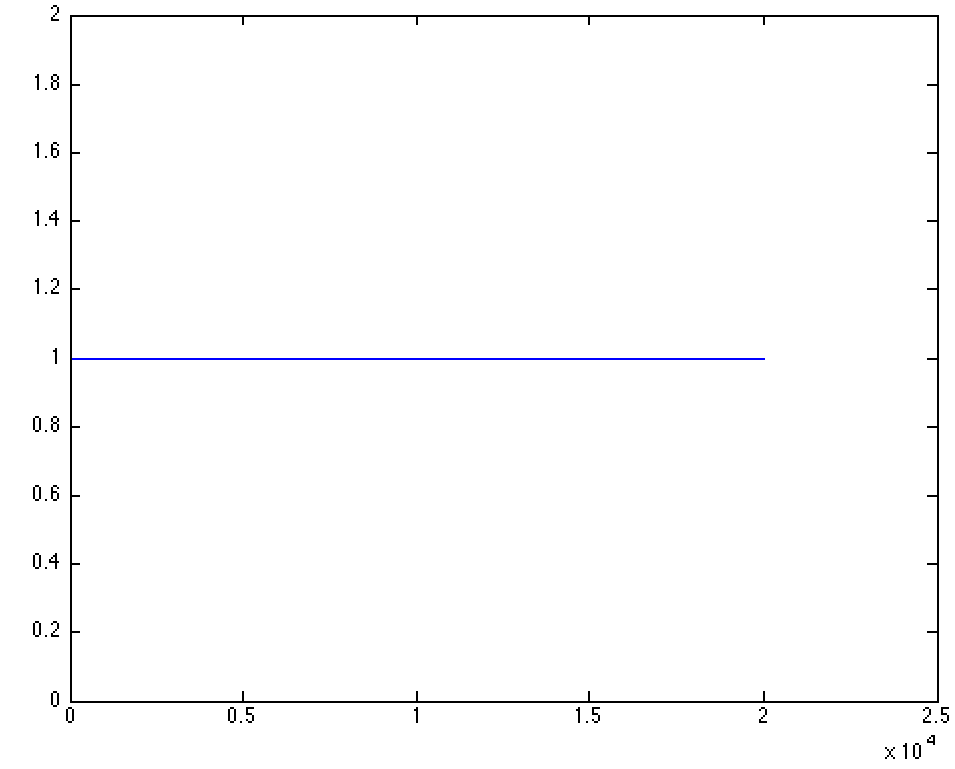
```
% this is for the 8th problem in homework2 from cpsc 546
% Gradient and Newton methods. Consider the unconstrained problem
clear;close all;
m = 40;
n = 50;
A = rand(m,n);
x0 = zeros(n,1);


f = @(x) -sum(log(1-A*x))-sum(log(ones(n,1)-x.^2));
g = @(x) A'*(1./(1-A*x))+(2.*x./(1-x.^2));
h = @(x) A'*diag((1./(1-A*x)).^2)*A + diag(1./(1+x).^2 + 1./(1-x).^2);
```

Gradient method

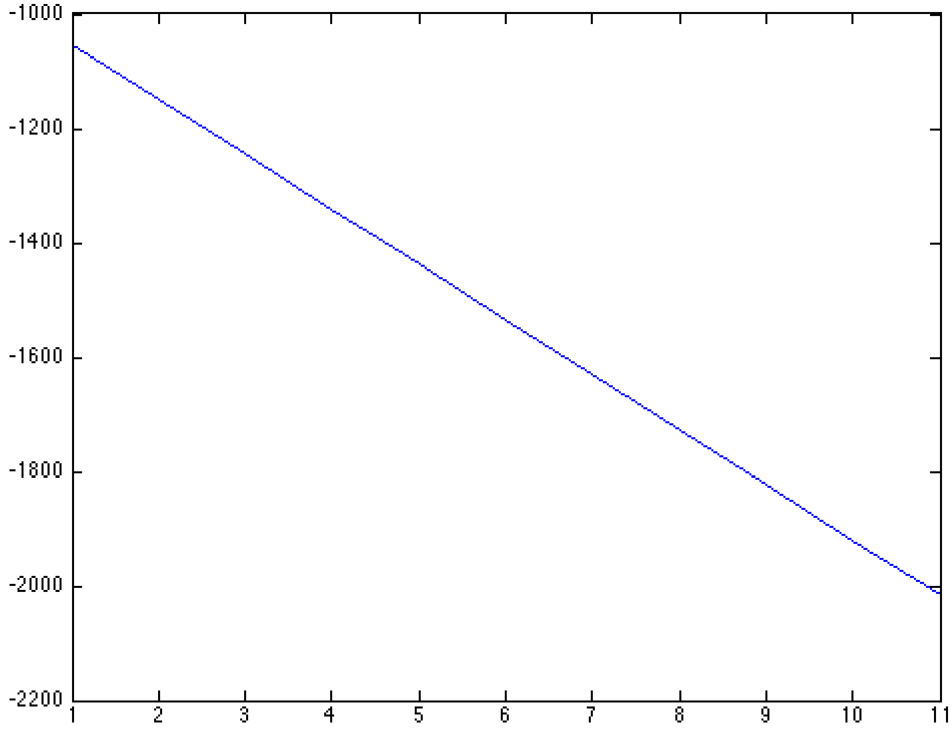
```
itnmax = 20000;
itn = 0;
tol = 10^-6;
while norm(g(x0))>=tol && itn<=itnmax
    d = -g(x0);
    % backtracking line search
    alpha = 1;
    while -abs(f(x0+alpha*d)) > -abs(f(x0)+10^-2*alpha.*g(x0)'*d)
        alpha = .5*alpha;
    end
    x0 = x0+alpha*d;
    itn = itn+1;
    plot_data(itn,:) = [f(x0),alpha];
end

figure;plot(1:1:itn,-abs(plot_data(:,1)))
figure;plot(1:1:itn,abs(plot_data(:,2)))
```

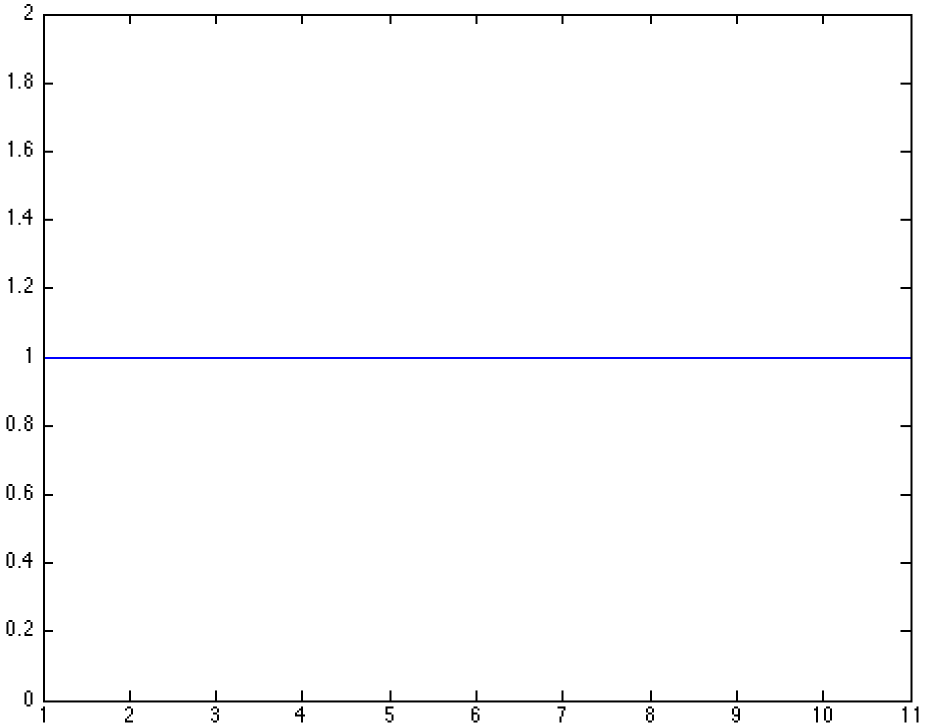

-550
-600
-650
-700
-750
-800
-850
-900
-950
-1000
0
0.5
1
1.5
2
2.5
x 10^4


2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
0.5
1
1.5
2
2.5
x 10^4

Newton method

```
clear plot_data;
itnmax = 10;
itn = 0;
tol = 10^-15;
while 1
    p = -h(x0)\g(x0);
    % backtracking line search
    alpha = 1;
    while -abs(f(x0+alpha*p)) > -abs(f(x0)+10^-4*alpha.*g(x0)'*p)
        alpha = .5*alpha;
    end
    x0 = x0+alpha*p;
    itn = itn+1;
    plot_data(itn,:) = [f(x0),alpha];

    if norm(p)<=tol || itn>itnmax
        break;
    end

end

figure;plot(1:1:itn,-abs(plot_data(:,1)))
figure;plot(1:1:itn,abs(plot_data(:,2)))
```





Published with MATLAB® 7.12