**% In-Lecture #3 Assignment related to Homework 5.1 Steepest Descent**

% Copy this file into a Matlab script window, add your code and answers to the

% questions as Matlab comments, hit "Publish", and upload the resulting PDF file

% to this page for the tune-up assignment.  Please do not submit a link to a file

% but instead upload the file itself.   **Late penalty:** 2 points per minute late.

% This assignment introduces steepest descent algorithms.  
% Please see Fig. 6.15 on page 116 in JSK’s *Software Receiver Design* book.

% See [steepest descent slides](http://users.ece.utexas.edu/~bevans/courses/realtime/lectures/tuneups/fall2024/InLectureWork3Slides.pptx) and [Midterm Problem 2.1 in Spring 2016](http://users.ece.utexas.edu/~bevans/courses/realtime/lectures/MidtermTwoSpring2016.pdf).

% Consider performing an iterative minimization of objective function

% *J*(*x*) = *x*^2 – 14*x* + 49 = (*x* - 7)^2  
% via the steepest descent algorithm (JSK equation (6.5) on page 116).  
%

% a. Visualize and analyze the shape of the objective function *J*(*x*).

% 1) Plot *J*(*x*) for 5 < *x* < 9. Give the Matlab code for your answer.

x = [5 : 0.01 : 9];

J = x.^2 - 14\*x + 49;

figure;

plot(x, J); %% At end of document

% 2) Describe the plot.

% *Answer:* It’s a concave up parabola (bowl)

% 3) How many local minima do you see?

% *Answer:* 1 at *x* = 7

% 4) Of the local minima, how many are global minima?

% *Answer:* The local minimum is also a global minimum.

% b. As first step in deriving steepest descent update equation,

% compute the first derivative of *J*(*x*) with respect to x.

% *Answer:* dJ(x)/dx = 2x - 14

% c. Implement the steepest descent algorithm in Matlab with *x*[0] = 5.

% 1) What value of *x* did steepest descent reach in 50 iterations with mu=0.01?

% *Answer:* *x* = 6.2568

% 2) What value of *x* did steepest descent reach in 50 iterations with mu=0.1?

% *Answer:* *x* = 7.0

% 3) Is the above value the global minimum of *J*(*x*)? Why or why not?

% *Answer:* Yes, the objective function has only one minimum.

% polyconverge.m find the minimum of J(x) via steepest descent

N=50;                      % number of iterations

mu=0.01;               % algorithm stepsize

x=zeros(1,N);              % initialize sequence of x values to zero

x(1)=5.0;                 % starting point x(1)

for k=1:N-1

  x(k+1)= x(k) - (2\*x(k)-14)\*mu;    % update equation

end

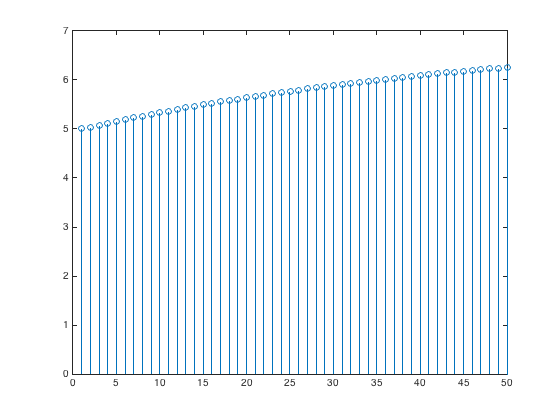
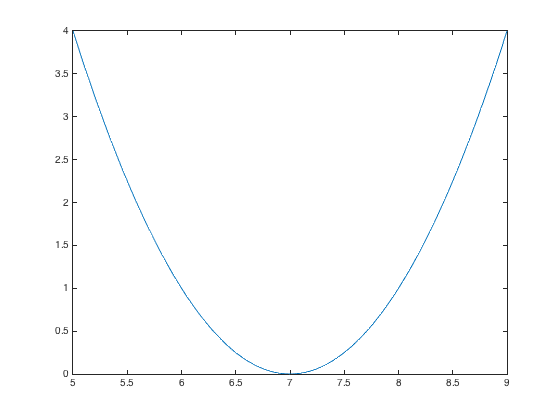
figure;

stem(x);          % to visualize approximation

x(N)

***See plots and comments on the next page…***

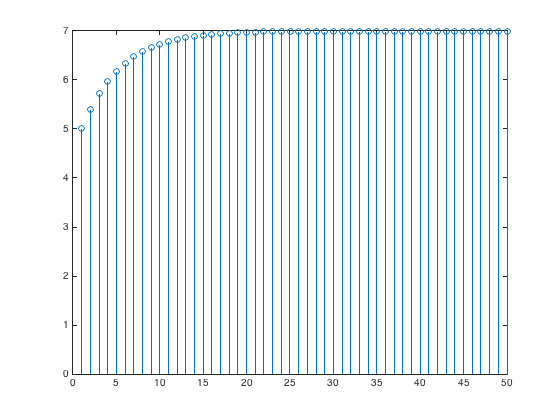
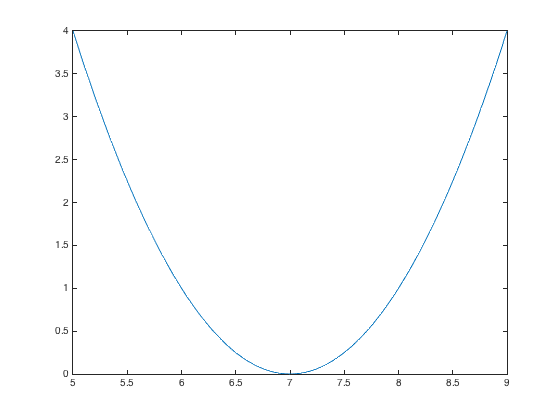
Plots for mu = 0.01



iterations

Plot of *J*(*x*) vs. x Plot of *x* vs. iterations

Plots for mu = 0.1



iterations

Plot of *J*(*x*) vs. x Plot of *x* vs. iterations

**Convergence.** In general, the convergence of the steepest descent algorithm depends on the initial guess and the value of the step size mu.

In this problem, the steepest descent algorithm is a first-order IIR filter with a pole at :

The current output is , previous output is , and current input is where is the unit step function. For the first-order IIR filter to be BIBO stable, the pole has to be inside the unit circle, i.e. or equivalently . We choose a small positive value for the step size so that the steepest descent algorithm converges. See additional analysis next.

**The first derivative acts like a highpass filter**: As an LTI system, the first derivative is a highpass FIR filter.  Recall that the first-order FIR difference filter from homework 1.1(b) and 2.1(b) is a discrete-time approximation of the first derivative and is a highpass filter.

Often in practice, the first derivative is calculated by formula or estimated numerically using measured data, e.g. using a continuous-time analog signal that has been converted to a discrete-time digital signal.  The first derivative, as a highpass filter, will amplify high-frequency components of noise and measurement error.

In this problem, **steepest descent** acts like a lowpass filter if the step size is small enough.  For a lowpass filter, we want the pole location to be at say 0.9 which would mean a = 0.05.  By choosing an appropriate value, we can equalize the highpass response of the first derivative.