Computational Economics Problem Set 5

Names: Di Lu(5706079),), Zexi Sun(5917910), Penghui Yin(5702358), Shiwei Yu(5913037)

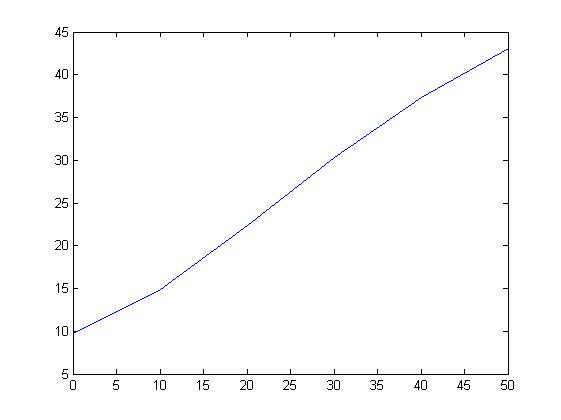
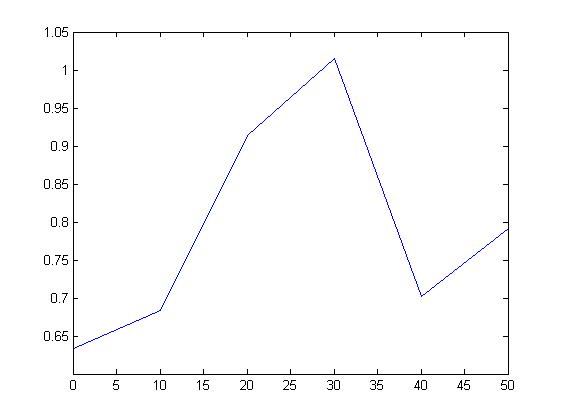
**Problem 2**

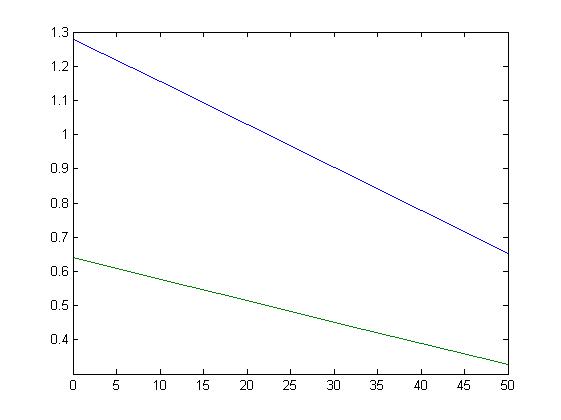
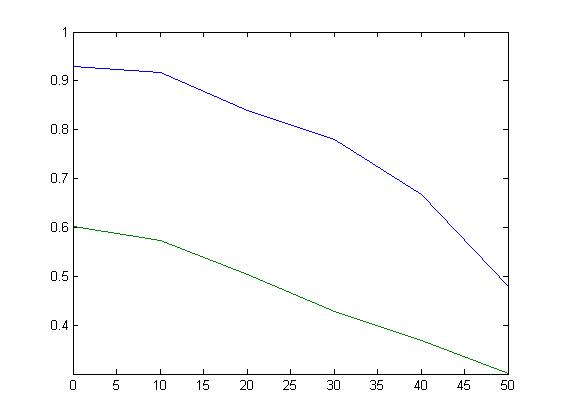
2.1

Very straight forward if we plug in the second term of W1 into the first formula and take the first derivative respect to αi , and then we get n equations exactly the same as we are asked to verify.

2.2

We first think of this problem with an implicit assumption that the sum of shares equal to one. Then this problem becomes a univariate one, and we try different method to solve this problem. We first implement Newton Method manually and get a robust answer in terms of initial guess. Then we also apply the fzero, fsolve, fminunc to this problem but all were found sensitive to the initial guess. So we use the first method to solve the problem with different amount of minimum wealth and get the picture in the up-left. We find it really strange, so head to another method afterwards which is fixed point iteration. As we thought there was no constraint on the share, any number might be a solution. Actually the fixed point iteration’s answer is also stable within the initial guess range from 3 to 13. Thus we also plot a solution of this method in the up-right. After we were informed of the online solution, we figured out there was no constraint on the sum of 1. We revised our code and get a reasonable result similar to the reference code. However, we find the solutions are not close with the same parameters and initial guess. Furthermore, they are both sensitive to the initial guess. In the down-left, we plot the initial guess of [1.2,1] to simulate the sample answer given, which is in the down-right with initial guess of [1,1]. As the fix-point iteration doesn’t get a stable answer within the reasonable range in the first case, and we are constrained by the time, we didn’t do that again. In order to save space, we just include the last part of code in the Appendix.





2.3

**Appendix**

PS5Q2

|  |
| --- |
| clc; close all; clear all;  %Parameters setting  r\_f=0.02;ru=0.5;mu=[0.04,0.06]';delta\_sd=[0.1,0.2];w0=100;wmin=0;gemma=2;a0=6;  rng(321);%set the seed  cov\_m=diag(delta\_sd);cov=sqrt(ru)\*delta\_sd(1)\*delta\_sd(2);  cov\_m(2)=cov;cov\_m(3)=cov;  r=mu+cov\_m\*rand(2,1);  %Approximate the expectation by Gauss-Hermite integration using m = 7  %nodes by the MF-function qnwnorm.  m=7;n=[m,m];  [x,w] = qnwnorm(n,mu',cov\_m);  %E=w'\*f(x),e.g.w'\*exp(x(:,1)+x(:,2))    syms A  A0=[1.2,1]';%very sensitive to initial guess  Astar=fsolve(@(A)-w'.\*1/(1-gemma)\*((1+r\_f+(x-r\_f)\*A)\*w0-wmin).^(1-gemma),A0);  wm=(0:10:50);astar1=zeros(2,length(wm));A1=A0;  for i=1:1:length(wm)  astar1(:,i) = fsolve(@(A)-w'.\*1/(1-gemma)\*((1+r\_f+(x-r\_f)\*A)\*w0-wm(i)).^(1-gemma),A1);  A1=astar1(:,i);  end  plot(wm,astar1) |