Paul Le Tran (working with Zhenghao Li)
plt377
08/05/22
ECO384G

Problem Set 6 (Problem Set 3 for spring 2022, Problem Set 1 for Chris)

1.1). We assume the parameters of the optimal consumption model to be the same as shown in class. For this part, we also do basic value function iteration. Figures 1, 2, and 3 display the results of the process below.

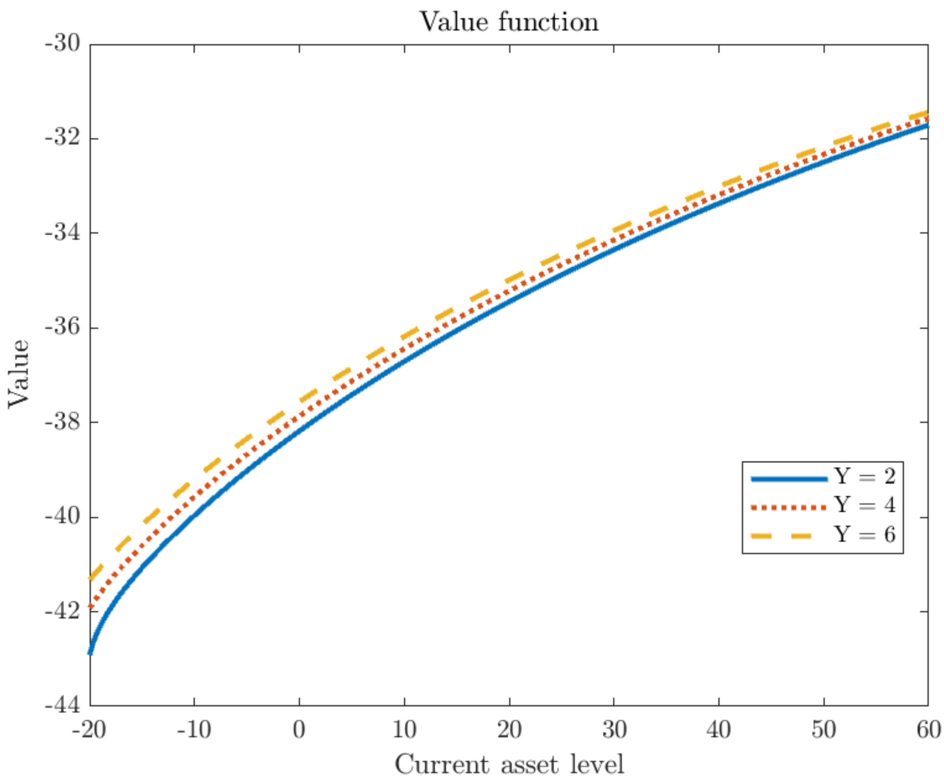

Figure 1: Value functions for different levels of income Y, given current period asset levels. Used basic value function iteration.

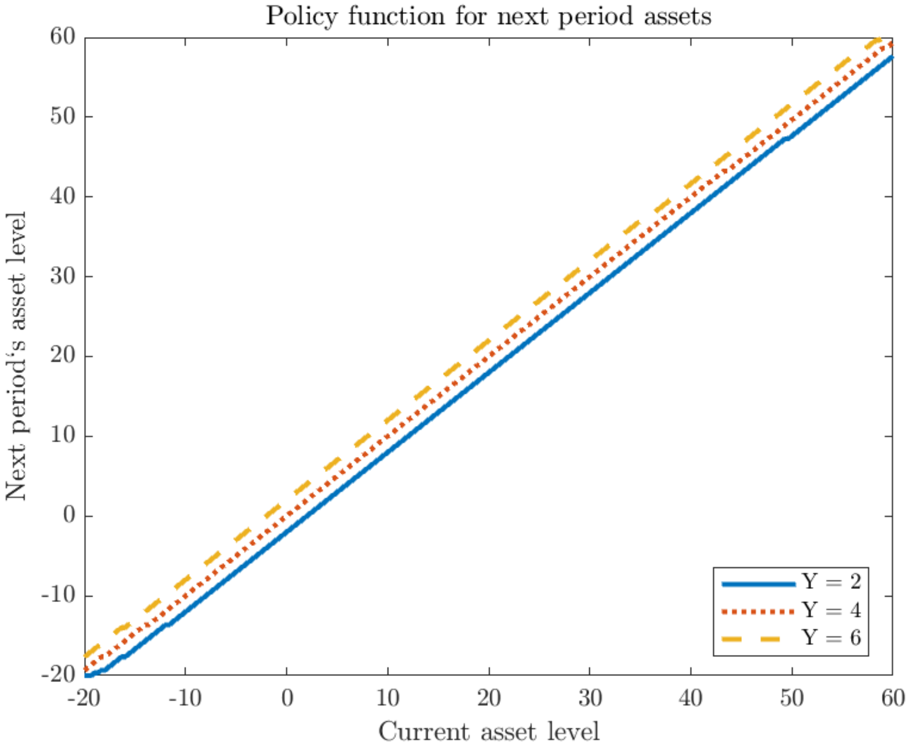

Figure 2: Policy function for choosing next period assets for different levels of income Y, given current period asset levels.

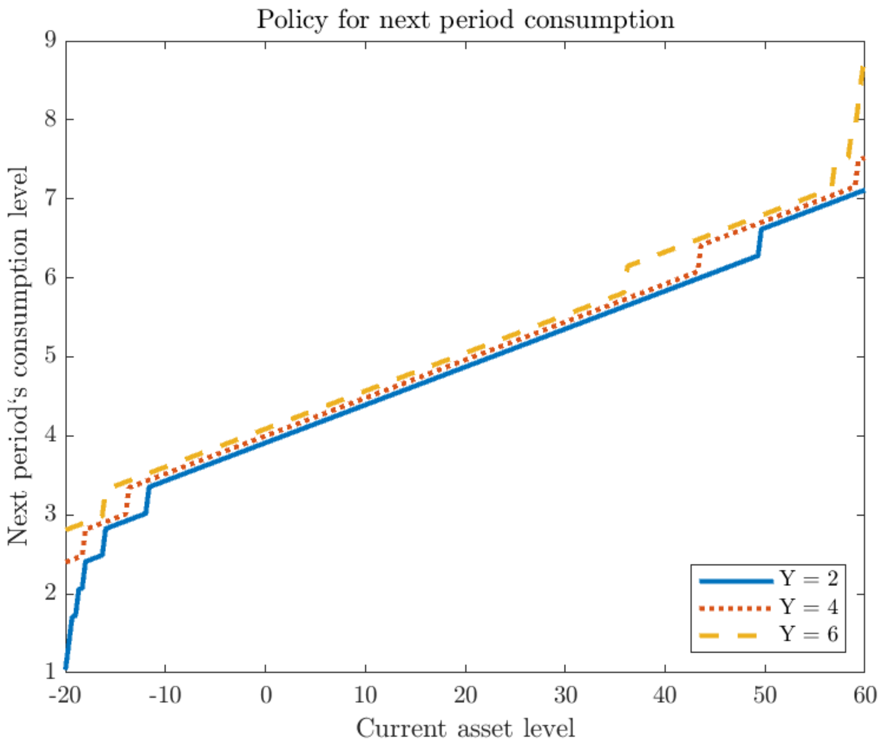

Figure 3: Policy function for choosing next period consumption for different levels of income Y, given current period asset levels. Used basic value function iteration.

1.2). We decrease the computational time of question 1 by conducting value function iteration with interpolation. Figures 4, 5, and 6 display the results below. One large difference to observe is how the interpolation process causes the policy functions in consumption (Figure 6) to remain linear at high current asset levels, which contrasts against Figure 3. Aside from that, interpolation results in smoother value and policy functions, especially for those close to or at the borrowing constraint.

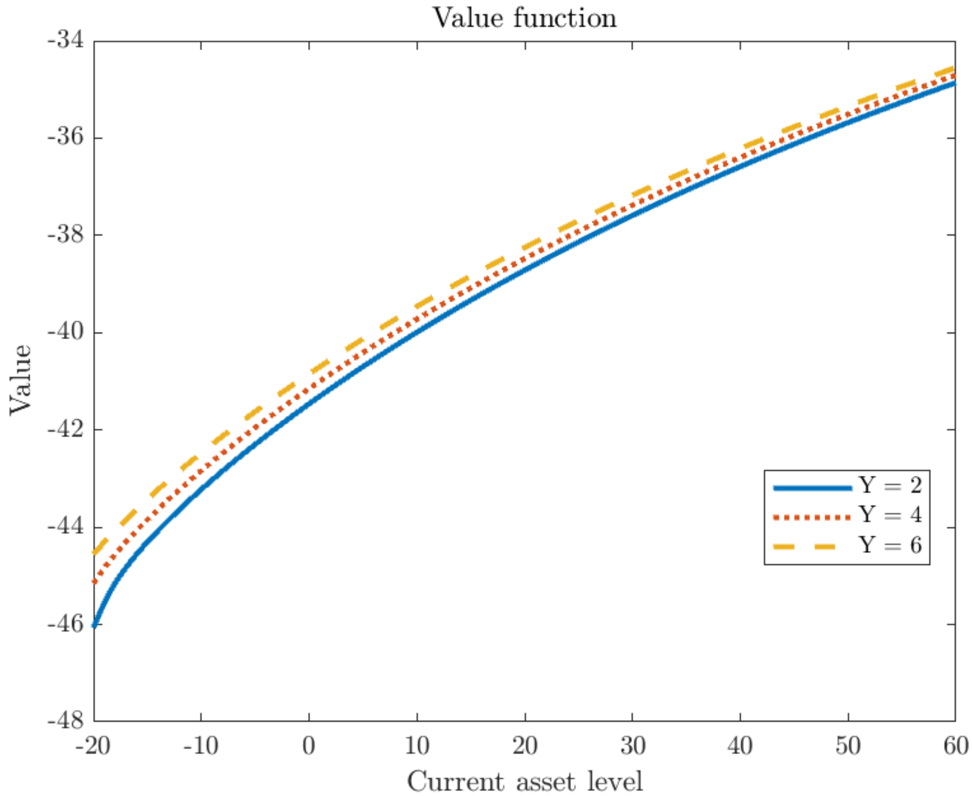

Figure 4: Value functions for different levels of income Y, given current period asset levels. Used value function iteration with interpolation.

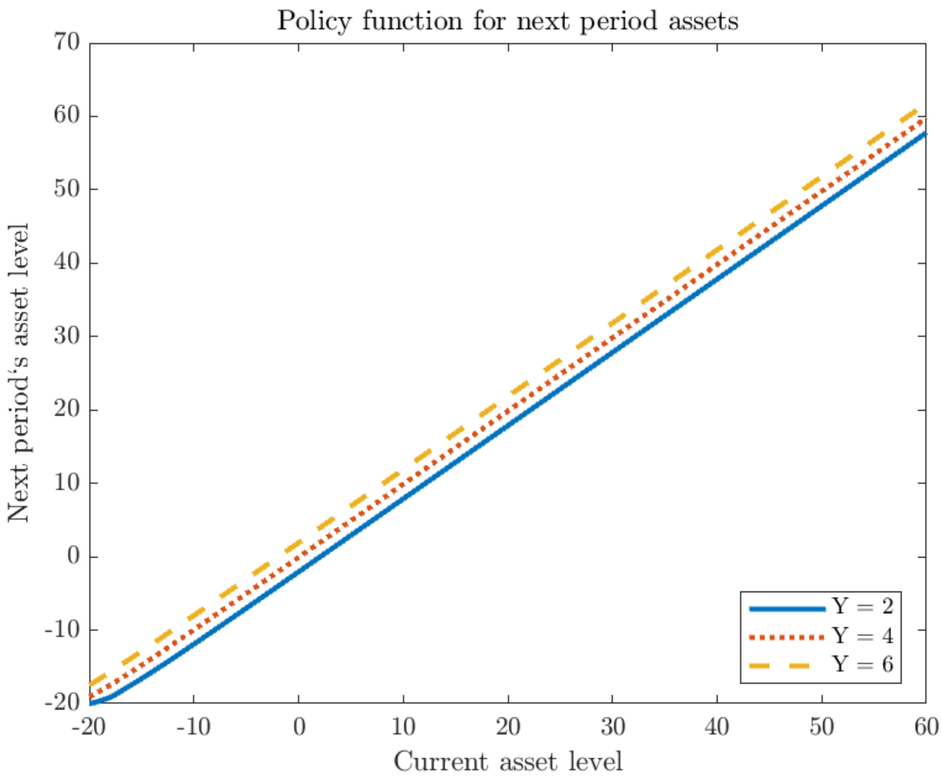

Figure 5: Policy function for choosing next period assets for different levels of income Y, given current period asset levels. Used value function iteration with interpolation.

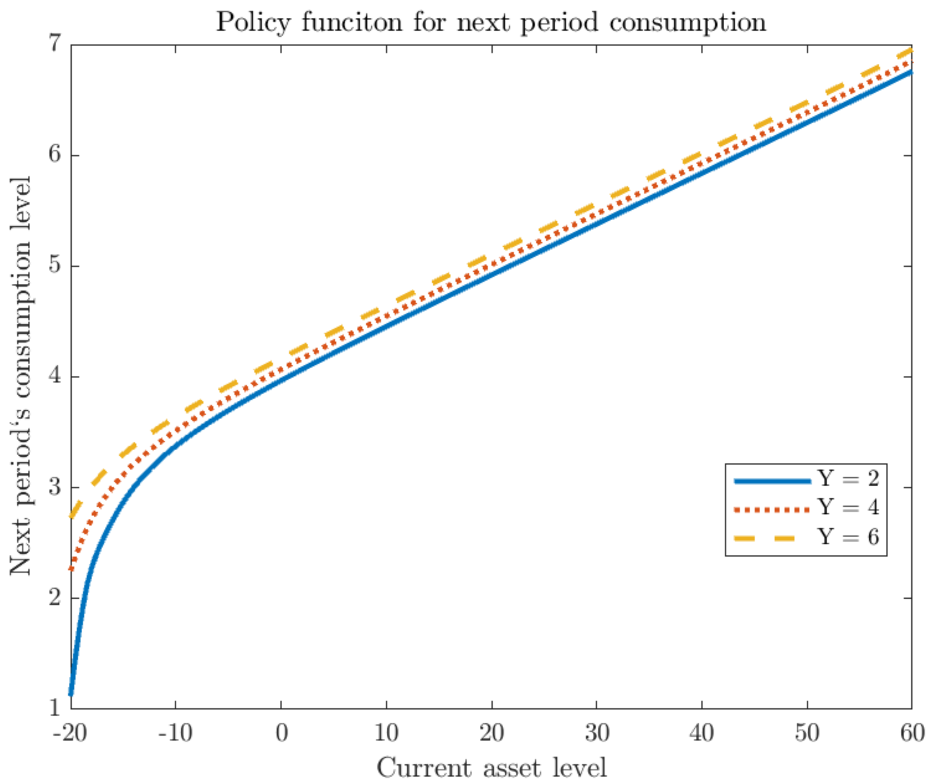

Figure 6: Policy function for choosing next period consumption for different levels of income Y, given current period asset levels. Used value function iteration with interpolation.

1.3). We now assume that income level follows an AR(1) process of the form $y_t = \rho y_{t-1} + \sigma_\epsilon \epsilon_t$, where ϵ_t follows a standard normal distribution. We use the provided code implementing Tauchen (1986)'s procedure to discretise the AR(1) process. We also conduct value function iteration through vectorisation. Figures 7, 8, and 9 assume $\rho = 0.9, \sigma_\epsilon = 0.5$. In terms of the value function, we immediately see that this new process causes there to be large gaps dependent on one's income level. This gap difference can also be seen when comparing the policy functions for consumption. However, it seems the policy functions for next period assets are closer together when assuming income following an AR(1) process.

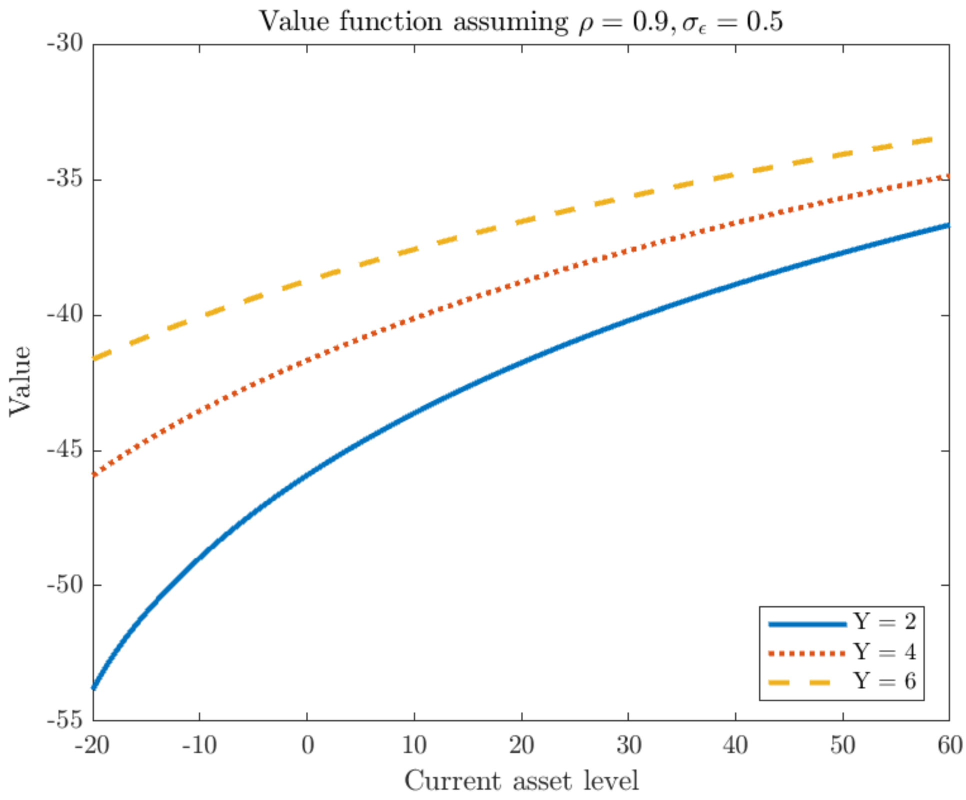

Figure 7: Value functions for different levels of income Y, given current period asset levels. Assuming Y follows an AR(1) process. Used value function iteration, vectorised.

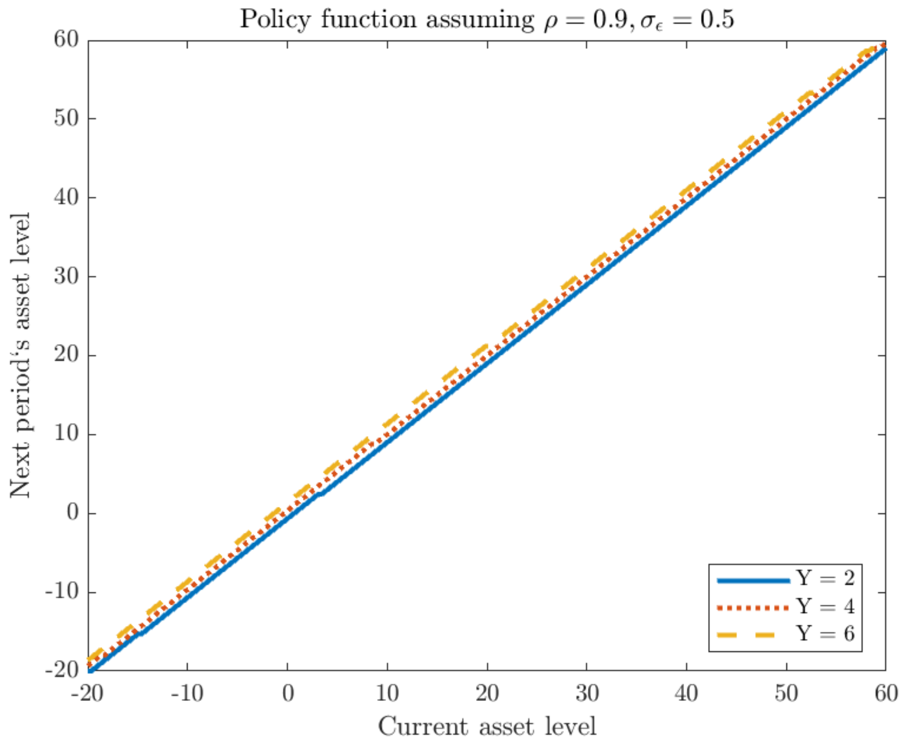

Figure 8: Policy function for choosing next period assets for different levels of income Y, given current period asset levels. Assuming Y follows an AR(1) process. Used value function iteration, vectorised.

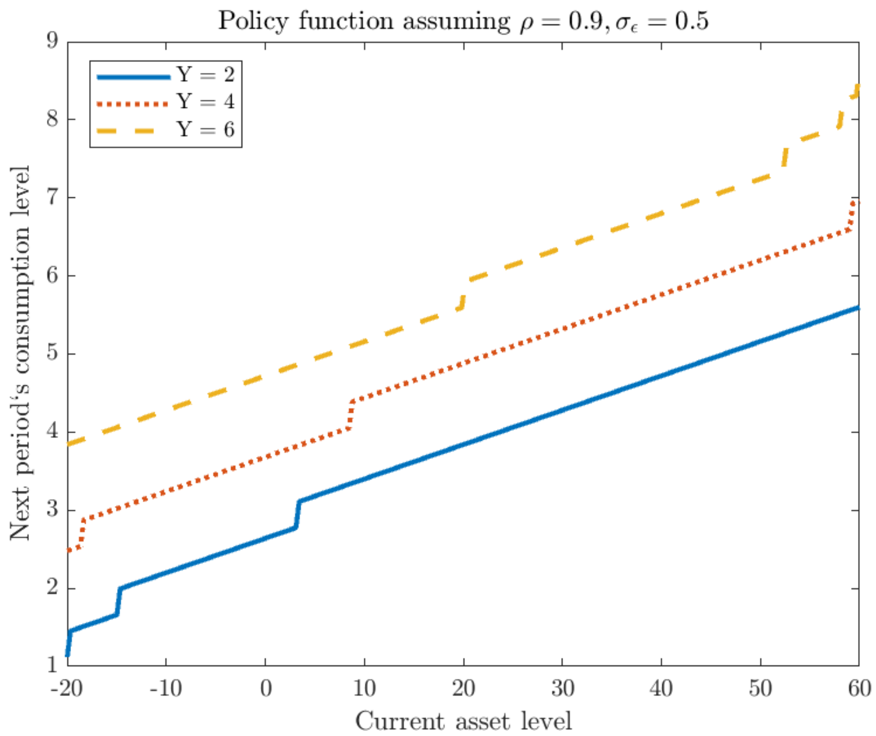

Figure 9: Policy function for choosing next period consumption for different levels of income Y, given current period asset levels. Assuming Y follows an AR(1) process. Used value function iteration, vectorised.

Interested to see how the parameters of the AR(1) process affect the iteration process, we compare the policy functions for consumption that are computed whilst varying values for ρ, σ_ϵ. To ensure the differences are only due to changes in these two parameters, we hold income at level $Y = 4$. The results are displayed in Figure 10. An interesting trend is that at the higher level of $\rho = 0.9$, increasing the standard deviation of the innovation of the AR(1) process results in the consumption function to be shifted higher upwards. However, this does not seem to be the case for the lower level of $\rho = 0.6$. We also observed that lower levels of ρ result in consumption to increase a lot steeper for those close to or at the borrowing constraint.

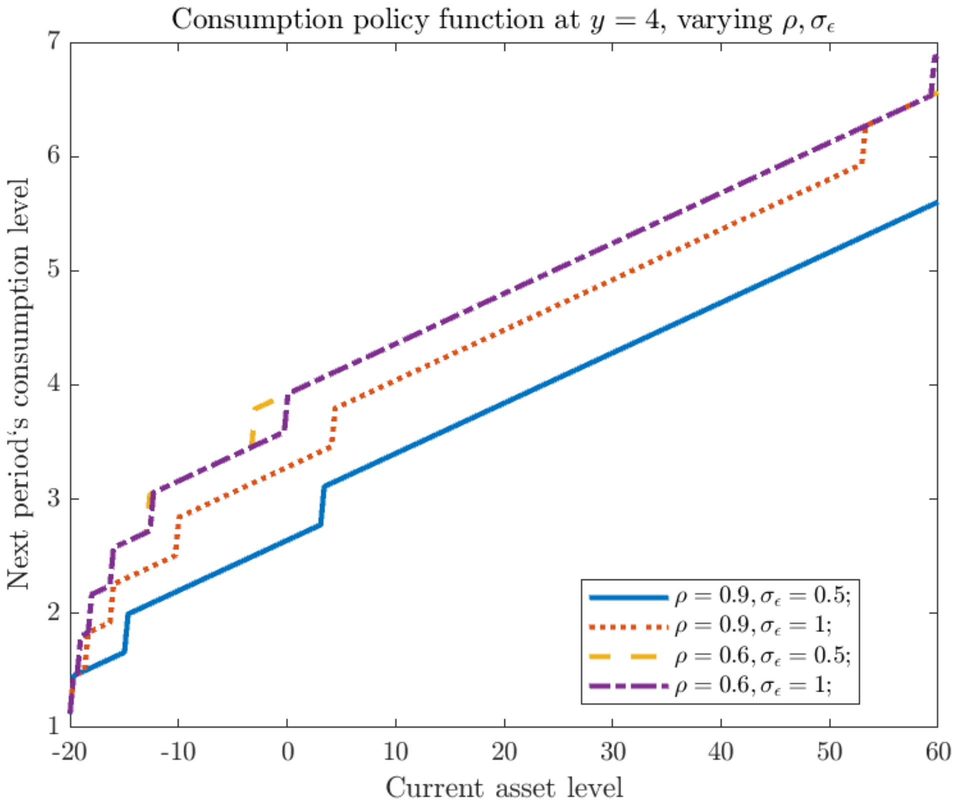

Figure 10: Policy functions for choosing consumption at the constant income level $Y = 4$, but varying parameters ρ, σ_ϵ. Used value function iteration, vectorised.

2.1). We choose to solve the model involving durable goods D and fixed costs F. We assume that utility takes the form of $u(D) = ln(D)$.With regards to parameters, we assume that the intertemporal elasticity of substitution is $\sigma = 0.75$; that $\beta = 0.95$, which leads to the interest rate of $r = \frac{1}{\beta} - 0.00215$; that the rate of depreciation is $\delta = 0.1$; that fixed costs is $F = 3$; and that the price of durable good D is $p = 1.5$. We further set income level to be $Y = 5$. For the random depreciation stock variable ϵ, we assume for simplicity that it has a probability transition matrix of $[1/3, 1/3, 1/3]$. More information about the assumptions made for the bounds of the discrete search grids of the state variables can be found in the accompanying code with this submission.

Figure 11 below displays the value functions for current asset level at $a = 13.14$. All value functions are increasing in current durable good stock level.

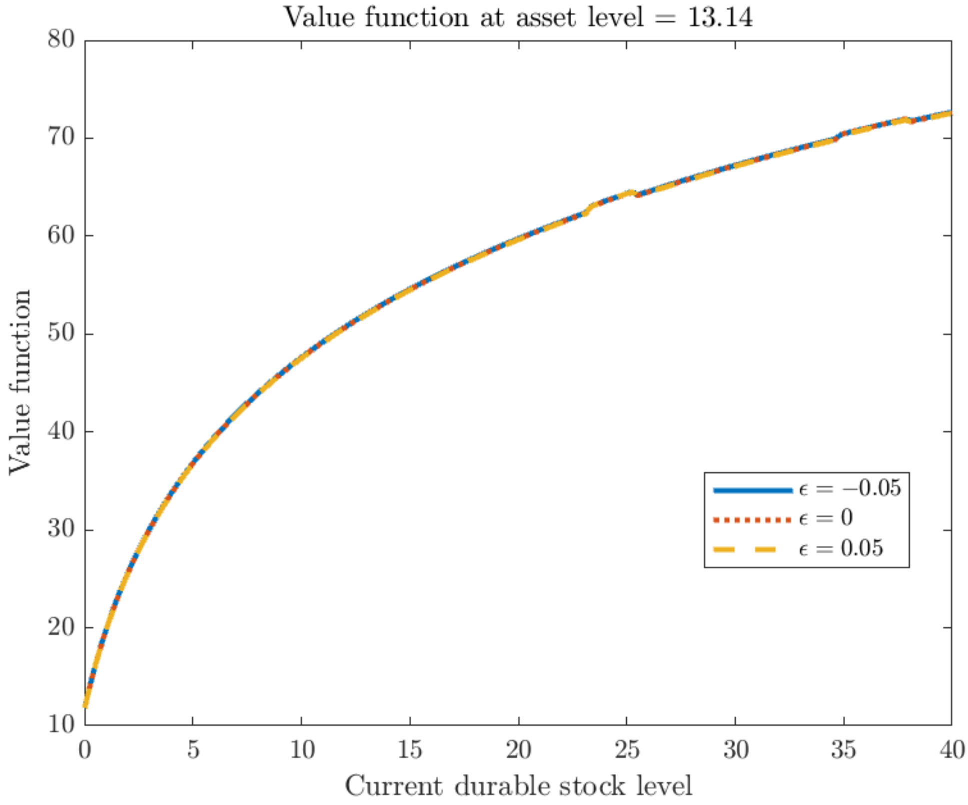

Figure 11: Value functions for different random depreciation stock values ϵ, given asset level $a = 13.14$. Used value function iteration, vectorised.

When examining the policy functions for next period durable good stock (Figure 12), we see that all three functions are fairly linear and increasing. However, the policy functions become steeper as the random depreciation stock decreases. This is because such decreases result in the actual depreciation rate of the durable good stock to be lower. Therefore, we should expect that positive ϵ will cause next period durable good levels to be lower, given any current stock level.

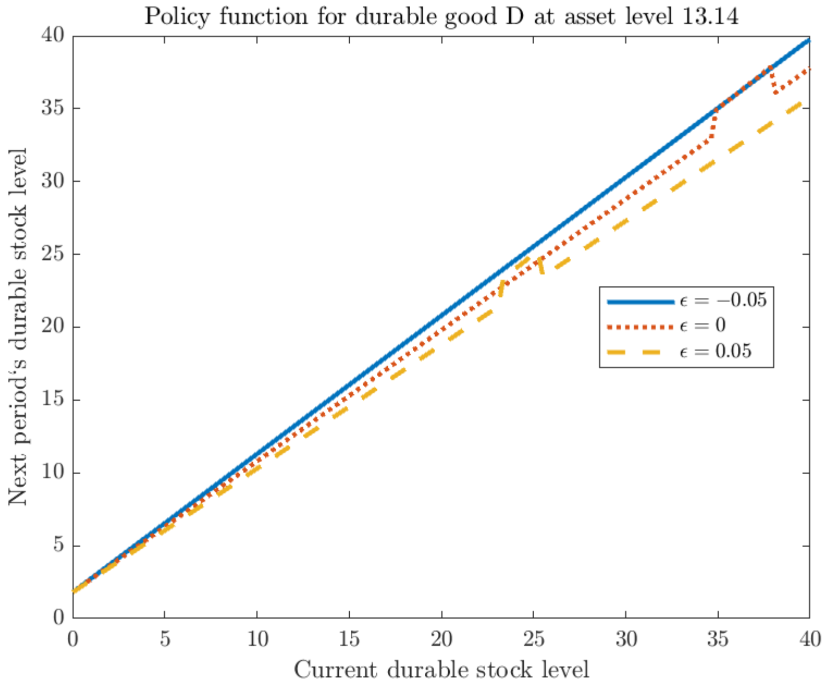

Figure 12: Policy functions for choosing next period durable good stock levels for different levels of ϵ, given asset level $a = 13.14$. Used value function iteration, vectorised.

We finally display an estimation of the long-run/ergodic distribution of current durable good stock holdings. We estimate this using by running 10,000 simulations. The results are displayed in Figure 13. Interestingly, our simulations show that the mean level of durable good stock is roughly between 15 and 16.

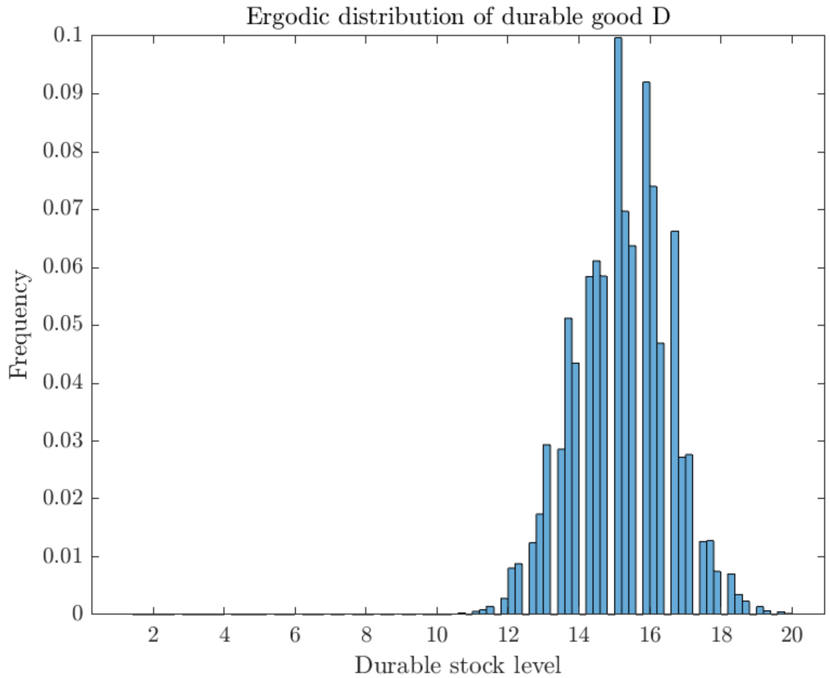

Figure 13: Estimation of ergodic distribution of durable good stock levels.

Appendix: The following contain the main Matlab code used to create the outputs discussed above for this problem set.

Code for problem 1:

```
%=========================================================================
%% Setting up workspace
clear all;
close all;
clc;

home_dir = 'Path\to\programmes';

% Setting text interpreter to latex
set(groot, 'defaultAxesTickLabelInterpreter', 'latex');
set(groot, 'defaulttextinterpreter', 'latex');
set(groot, 'defaultLegendInterpreter', 'latex');

cd(home_dir);
%=========================================================================

%=========================================================================
%% Part 1: Write your own code to solve the problem using value function
iteration
% Initialising intertemporal elasticity of substitution
sigma = 0.75;
% Initialising discount factor
beta = 0.95;
R = 1/beta - 0.00215;

% Setting iteration parameters
diff = 1;
epsi = 1e-4;

% Creating discrete search grid for state variables assets a and income Y
Amin = -20;
Amax = 60;
Na = 3*80;
Ymin = 2;
Ymax = 6;
Ny = 3;

% Discretisising state space
A = linspace(Amin, Amax, Na)';
Y = linspace(Ymin, Ymax, Ny)';

% Creating transition probability matrix
Ptran = [1/Ny 1/Ny 1/Ny];

% Initialising first guesses of value function
V = zeros(Na, Ny);

% Initialising first guesses of other variables
Vnew = zeros(Na, Ny);
Pindex = zeros(Na, Ny);

% Creating variable to keep track of iterations
iter = 0;
```

```
% Performing VFI
tic
while diff > epsi
  iter = iter + 1;
  for i = 1:Na
    for j = 1:Ny
      C = A(i) + Y(j) - (A/R);
      % For values outside of the allowed grid, we set it to negative
      % infinity
      U = -inf(Na, 1);
      U(C > 0) = (1/(1 - 1/sigma))*(C(C > 0).^(1 - 1/sigma));
      Vtemp = U + beta*V*Ptran';
      [M, I] = max(Vtemp);
      Vnew(i, j) = M;
      Pindex(i, j) = I;
    end
  end
  diff = max(max(abs(V - Vnew)));
  disp([iter diff]);

  % Saving current version of value function
  V = Vnew;
end
toc

% Plotting the value function
plot(A, V(:, 1), '-', 'Linewidth', 2);
hold on;
plot(A, V(:, 2), ':', 'Linewidth', 2);
plot(A, V(:, 3), '--', 'Linewidth', 2);
title('Value function');
xlabel('Current asset level');
ylabel('Value');
legend('Y = 2', 'Y = 4', 'Y = 6', 'Location', 'Best');
hold off;
saveas(gcf, 'Path\to\graphics\1_1a_plot.png');
close(gcf);

% Plotting the policy function for assets a'
plot(A, A(Pindex(:, 1)), '-', 'Linewidth', 2);
hold on;
plot(A, A(Pindex(:, 2)), ':', 'Linewidth', 2);
plot(A, A(Pindex(:, 3)), '--', 'Linewidth', 2);
title('Policy function for next period assets');
xlabel('Current asset level');
ylabel('Next period`s asset level');
legend('Y = 2', 'Y = 4', 'Y = 6', 'Location', 'Best');
hold off;
saveas(gcf, 'Path\to\graphics\1_1b_plot.png');
close(gcf);

% Plotting the policy function for consumption c'
Cs = repmat(A, 1, Ny) + repmat(Y', Na, 1) - [(A(Pindex(:, 1))/R)
(A(Pindex(:, 2))/R) (A(Pindex(:, 3))/R)];
plot(A, Cs(:, 1), '-', 'Linewidth', 2);
hold on;
plot(A, Cs(:, 2), ':', 'Linewidth', 2);
plot(A, Cs(:, 3), '--', 'Linewidth', 2);
title('Policy for next period consumption');
```

```
xlabel('Current asset level');
ylabel('Next period`s consumption level');
legend('Y = 2', 'Y = 4', 'Y = 6', 'Location', 'SouthEast');
hold off;
saveas(gcf, 'Path\to\graphics\1_1c_plot.png');
close(gcf);
%==========================================================================

%==========================================================================
%% Part 2: Speeding up VFI performed in part 1 using interpolation
% Initialising intertemporal elasticity of substitution
sigma = 0.75;
% Initialising discount factor
beta = 0.954;
R = 1/beta - 0.00215;

% Setting iteration parameters
diff = 1;
epsi = 1e-4;
itmax = 1000;
solverOptions = optimset('TolX', 1e-8);

% Creating utility function
global Utility;
Utility = @(c) (c > 0).*((1/(1 - 1/sigma))*c.^(1 - 1/sigma)) + (c <= 0).*-
1e10;

% Creating discrete search grid for state variables assets a and income Y
Amin = -20;
Amax = 60;
Na = 1.5*30;
Ymin = 2;
Ymax = 6;
Ny = 3;

% Discretisising state space
A = linspace(Amin, Amax, Na)';
Y = linspace(Ymin, Ymax, Ny)';

% Creating transition probability matrix
Ptran = [1/Ny 1/Ny 1/Ny]';

% Initialising first guesses of value function
V = ones(Na, Ny);
ap_pol = zeros(Na, Ny);

% Initialising first guesses of other variables
U = ones(Na, 1);
Vnew = ones(Na, Ny);

% Creating variable to keep track of iterations
iter = 0;

% Performing VFI
tic
while diff > epsi && iter <= itmax
iter = iter + 1;
V = Vnew;
vFuture = beta*V*Ptran;
```

```
for i_a = 1:Na
  for i_y = 1:Ny
    a = A(i_a);
    y = Y(i_y);
    ap_ub = a + y;
    [ap_pol(i_a,i_y), Vnew(i_a,i_y)] = fminbnd(@(ap) ...
      -value_of_ap(ap, i_a, i_y, Utility, A, Y, R, vFuture), A(1), ap_ub,
solverOptions);
  end
end
Vnew = -Vnew;
diff = max(max(abs(V - Vnew)));
disp([iter diff]);
end
V = Vnew;
toc

% Creating axis showing asset levels found in VFI via interpolation
Adet = linspace(A(1), A(end), 200);

% Plotting value function
plot(Adet, interp1(A, V(:, 1), Adet, 'pchip', 'extrap'), '-', 'Linewidth',
2);
hold on;
plot(Adet, interp1(A, V(:, 2), Adet, 'pchip', 'extrap'), ':', 'Linewidth',
2);
plot(Adet, interp1(A, V(:,3), Adet, 'pchip', 'extrap'), '--', 'Linewidth',
2);
title('Value function')
xlabel('Current asset level')
ylabel('Value')
legend('Y = 2', 'Y = 4', 'Y = 6', 'Location','Best');
hold off;
saveas(gcf, 'Path\to\graphics\1_2a_plot.png');
close(gcf);

% Plotting the policy function for assets a'
plot(Adet, interp1(A, ap_pol(:, 1), Adet, 'pchip', 'extrap'), '-',
'Linewidth', 2);
hold on;
plot(Adet, interp1(A, ap_pol(:, 2), Adet, 'pchip', 'extrap'), ':',
'Linewidth', 2);
plot(Adet, interp1(A, ap_pol(:, 3), Adet, 'pchip', 'extrap'), '--',
'Linewidth', 2);
title('Policy function for next period assets');
xlabel('Current asset level');
ylabel('Next period`s asset level');
legend('Y = 2', 'Y = 4', 'Y = 6', 'Location', 'Best');
hold off;
saveas(gcf, 'Path\to\graphics\1_2b_plot.png');
close(gcf);

% Computing consumption function
C = zeros(Na, Ny);
for i_a = 1:Na
  for i_y = 1:Ny
    C(i_a, i_y) = Y(i_y) + A(i_a) - ap_pol(i_a, i_y)/R;
  end
end
```

```
% Plotting the policy function for consumption c'
plot(Adet, interp1(A, C(:, 1), Adet, 'pchip', 'extrap'), '-', 'Linewidth',
2);
hold on;
plot(Adet, interp1(A, C(:, 2), Adet, 'pchip', 'extrap'), ':', 'Linewidth',
2);
plot(Adet, interp1(A, C(:, 3), Adet, 'pchip', 'extrap'), '--', 'Linewidth',
2);
title('Policy funciton for next period consumption');
xlabel('Current asset level');
ylabel('Next period`s consumption level');
legend('Y = 2', 'Y = 4', 'Y = 6', 'Location', 'Best');
hold off;
saveas(gcf, 'Path\to\graphics\1_2c_plot.png');
close(gcf);
%==========================================================================

%==========================================================================
%% Part 3: Assume income follows AR(1) process. Using Tauchen's procedure
to discretise the AR(1) process and then performing VFI.
% Initialising intertemporal elasticity of substitution
sigma = 0.75;
% Initialising discount factor
beta = 0.954;
R = 1/beta - 0.00215;

% Setting iteration parameters
epsi = 1e-4;

% Creating discrete search grid for state variables assets a and income Y
Amin = -20;
Amax = 60;
Na = 3*80;
Ymin = 2;
Ymax = 6;
Ny = 3;

% Discretisising state space
A = linspace(Amin, Amax, Na)';
Y = linspace(Ymin, Ymax, Ny)';

% Setting various values for income AR(1) process that Tauchen's procedure
% will use
rho = [0.9 0.6];
sd = [0.5 1];
ss_val = 4;

% Initialising first guesses
Vs = zeros(Na, Ny, length(rho)*length(sd));
Cs = zeros(Na, Ny, length(rho)*length(sd));
As = zeros(Na, Ny, length(rho)*length(sd));
Ptran = zeros(Ny, Ny, length(rho)*length(sd));
P = zeros(Na*Ny, Ny, length(rho)*length(sd));

Agrid = repmat(A, 1, Ny);
AAgrid = reshape(Agrid, Na*Ny, 1);
Ygrid = repmat(Y, 1, Na)';
YYgrid = reshape(Ygrid, Na*Ny, 1);

CCgrid = AAgrid + YYgrid - A'./R;
```

```
UUgrid = -inf(Na*Ny, Na);
UUgrid(CCgrid > 0) = (CCgrid(CCgrid > 0).^(1-1/sigma))/(1 - (1/sigma));

% Keeping track of iterations
iter = 0;

% Performing Tauchen's discretisation process then VFI via vectorisation
% inside
tic
for i = 1:length(rho)
  for j = 1:length(sd)
    diff = 1;
    iter = iter + 1;
    disp([iter rho(i) sd(j)]);
    % Using Tauchen's procedure to discretise AR(1) process of income
    Ptran(:, :, iter) = tauchen1986(rho(i), sd(j), ss_val,Y');
    P(:, :, iter) = [repmat(Ptran(1, :, iter), Na, 1); repmat(Ptran(2, :,
iter), Na, 1);...
      repmat(Ptran(3, :, iter), Na, 1)];
    VVgrid = zeros(Na*Ny,1);
    while diff > epsi
      VVtemp = UUgrid + beta*P(:, :, iter)*reshape(VVgrid, Na, Ny)';
      [VVnew, I] = max(VVtemp, [], 2);
      diff = max(abs(VVgrid - VVnew));
      VVgrid = VVnew;
    end
    Vs(:, :, iter) = reshape(VVgrid, Na, Ny);
    Pindex = reshape(I, Na, Ny);
    As(:, :, iter) = [A(Pindex(:, 1)) A(Pindex(:, 2)) A(Pindex(:, 3))];
    Cs(:, :, iter) = Agrid + Ygrid - As(:, :, iter)/R;
  end
end
disp(iter);
toc

% Plotting value function
plot(A, Vs(:, 1, 1), '-', 'Linewidth', 2);
hold on;
plot(A, Vs(:, 2, 1), ':', 'Linewidth', 2);
plot(A, Vs(:, 3, 1), '--', 'Linewidth', 2);
title('Value function assuming $\rho = 0.9, \sigma_{\epsilon} = 0.5$');
xlabel('Current asset level');
ylabel('Value');
legend('Y = 2', 'Y = 4', 'Y = 6', 'Location','Best');
hold off;
saveas(gcf, 'Path\to\graphics\1_3a_plot.png');
close(gcf);

% Plotting the policy function for assets a'
plot(A, As(:, 1, 1), '-', 'Linewidth', 2);
hold on;
plot(A, As(:, 2, 1), ':', 'Linewidth', 2);
plot(A, As(:, 3, 1), '--', 'Linewidth', 2);
title('Policy function assuming $\rho = 0.9, \sigma_{\epsilon} = 0.5$');
xlabel('Current asset level');
ylabel('Next period`s asset level');
legend('Y = 2', 'Y = 4', 'Y = 6', 'Location', 'Best');
hold off;
saveas(gcf, 'Path\to\graphics\1_3b_plot.png');
close(gcf);
```

```
% Plotting the policy function for consumption c'
plot(A, Cs(:, 1, 1), '-', 'Linewidth', 2);
hold on;
plot(A, Cs(:, 2, 1), ':' , 'Linewidth', 2);
plot(A, Cs(:, 3, 1), '--', 'Linewidth', 2);
title('Policy function assuming $\rho = 0.9, \sigma_{\epsilon} = 0.5$');
xlabel('Current asset level');
ylabel('Next period`s consumption level');
legend('Y = 2', 'Y = 4', 'Y = 6', 'Location', 'Best');
hold off;
saveas(gcf, 'Path\to\graphics\1_3c_plot.png');
close(gcf);

% Plotting policy function for consumption c' assuming different values of
\rho and SD
plot(A, Cs(:, 1, 1), '-', 'Linewidth', 2);
hold on;
plot(A, Cs(:, 1, 2), ':', 'Linewidth', 2);
plot(A, Cs(:, 1, 3), '--', 'Linewidth', 2);
plot(A, Cs(:, 1, 4), '-.', 'Linewidth', 2);
title('Consumption policy function at $y = 4$, varying $\rho,
\sigma_{\epsilon}$');
xlabel('Current asset level');
ylabel('Next period`s consumption level');
legend('$\rho = 0.9, \sigma_{\epsilon}=0.5$;',...
  '$\rho = 0.9, \sigma_{\epsilon} = 1$;',...
  '$\rho=0.6, \sigma_{\epsilon} = 0.5$;',...
  '$\rho = 0.6, \sigma_{\epsilon} = 1$;',...
  'Location', 'Best');
hold off;
saveas(gcf, 'Path\to\graphics\1_3d_plot.png');
close(gcf);
%==========================================================================
```

Code for problem 2, 1:

```
%=============================================================================
%% Setting up workspace
clear all;
close all;
clc;

home_dir = 'Path\to\programmes';

% Setting text interpreter to latex
set(groot, 'defaultAxesTickLabelInterpreter', 'latex');
set(groot, 'defaulttextinterpreter', 'latex');
set(groot, 'defaultLegendInterpreter', 'latex');

cd(home_dir);
%=============================================================================

%=============================================================================
%% Iterating to obtain next period's durable goods level
%
% Initialising intertemporal elasticity of substitution
sigma = 0.75;
% Initialising discount factor
beta = 0.95;
r = 1/beta - 0.00215;
% Initialising depreciation rate
delta = 0.1;
% Initialising fixed cost
F = 3;
% Initialising price of durable good D
p = 1.5;

% Setting iteration parameters
epsi = 1e-4;

% Creating discrete search grid for state variables assets a
Amin = 0;
Amax = 1;
Na = 3*10;

% Setting income level
y = 5;

% Creating discrete search grid for durable goods D
Dmin = 0.01;
Dmax = 40;
Nd = 3*50;

% Creating discrete search grid for random depreciation stock e
emin = -0.05;
emax = 0.05;
Ne = 3;

% Discretisising state space
A = linspace(Amin, Amax, Na)';
D = linspace(Dmin, Dmax, Nd)';
e = linspace(emin, emax, Ne)';
```

```
% More discretisation
A1 = repmat(A', Ne, Nd);
a1 = A1(:);
e1 = repmat(e, Nd*Na, 1);
D1 = repmat(D', Ne*Na, 1);
d1 = D1(:);
ddgrid = D';
d2 = repmat(D', Na*Nd*Ne, 1);

% Creating transition probability matrix
Ptran = [1/Ne 1/Ne 1/Ne];

% Initialising array to store next period's D
a2 = zeros(Nd*Na*Ne, Nd);

% Performing iteration to obtain next period's D
for i = 1:size(a2, 1)
  for j = 1:size(a2, 2)
    if ddgrid(j) == d1(i)
      a2(i, j) = (y + a1(i) - p*(ddgrid(j) - d1(i)*(1 - delta - e1(i))))*(1
+ r);
      d2(i, j) =ddgrid(j);
      if (y + a1(i) - p*(ddgrid(j) - d1(i)*(1 - delta-e1(i))))*(1 + r) <=
Amin
        d2(i, j) = (y + a1(i) - F - Amin/(1 + r))/p + d1(i)*(1 - delta -
e1(i));
        a2(i, j) = Amin;
      else
        if (y + a1(i) - p*(ddgrid(j) - d1(i)*(1 - delta - e1(i))))*(1 + r)
>= Amax
          d2(i, j) = (y + a1(i) - F - Amax/(1 + r))/p + d1(i)*(1 - delta -
e1(i));
          a2(i, j) = Amax;
        end
      end
    else
      a2(i, j) = (y + a1(i) - F - p*(ddgrid(j) - d1(i)*(1 - delta -
e1(i))))*(1 + r);
      d2(i, j) = ddgrid(j);
      if (y + a1(i) - F - p*(ddgrid(j) - d1(i)*(1 - delta - e1(i))))*(1 +
r) <= Amin
        d2(i, j) = (y + a1(i) - F - Amin/(1 + r))/p + d1(i)*(1 - delta -
e1(i));
        a2(i, j) = Amin;
      else
        if (y + a1(i) - F - p*(ddgrid(j) - d1(i)*(1 - delta - e1(i))))*(1 +
r) >= Amax
          d2(i, j)=(y + a1(i) - F-Amax/(1 + r))/p + d1(i)*(1 - delta -
e1(i));
          a2(i, j) = Amax;
        end
      end
    end
  end
end

% Assuming Utility is of the form u(D) = ln(D)
U = log(d2);

max(d2,[], "all")
```

```
min(d2,[], "all")
%==========================================================================

%==========================================================================
%% Performing VFI via vectorisation
% Setting iteration parameters
diff = 1;

% Initialising first guesses of value and policy function
v = zeros(Ne, Na, Nd);
P = repmat(Ptran, Ne, 1);

tic
while diff > epsi
  Ev = P*v(:, :);
  Ev = Ev(:);
  [vnew,b] = max(U + beta*Ev, [], 2);
  diff = max(abs(v(:) - vnew));
  v = reshape(vnew, Ne, Na, Nd);
end
toc

dp1 = zeros(length(b), 1);
ap1 = zeros(length(b), 1);

for i = 1:size(d2, 1)
  for j = 1:size(d2, 2)
    dp1(i) = d2(i, b(i));
    ap1(i) = a2(i, b(i));
  end
end

dp = reshape(dp1, Ne, Na, Nd);
ap = reshape(ap1, Ne, Na, Nd);
%==========================================================================

%==========================================================================
%% Plotting value and policy functions
plotvalue = find(A == max(A((A < median(A) + 0.2 & median(A) - 0.2))));

% Plotting value function
plot(D, reshape(v(1, plotvalue, :), 1, length(D)), '-', 'Linewidth', 2);
hold on;
plot(D, reshape(v(2, plotvalue, :), 1, length(D)), ':', 'Linewidth', 2);
plot(D, reshape(v(3, plotvalue, :), 1, length(D)), '--', 'Linewidth', 2);
title('Value function at asset level = 13.14');
xlabel('Current durable stock level');
ylabel('Value function');
legend('$\epsilon = -0.05$', '$\epsilon = 0$', '$\epsilon = 0.05$',
'Location', 'Best');
hold off;
saveas(gcf, 'Path\to\graphics\2_1a_plot.png');
close(gcf);

% Plotting policy function for durable good D'
plot(D, reshape(dp(1, plotvalue, :), 1, length(D)), '-', 'Linewidth', 2);
hold on;
plot(D, reshape(dp(2, plotvalue, :), 1, length(D)), ':', 'Linewidth', 2);
plot(D, reshape(dp(3, plotvalue, :), 1, length(D)), '--', 'Linewidth', 2);
```

```
title('Policy function for durable good D at asset level 13.14');
xlabel('Current durable stock level');
ylabel('Next period`s durable stock level');
legend('$\epsilon = -0.05$', '$\epsilon = 0$', '$\epsilon = 0.05$',
'Location', 'Best');
hold off;
saveas(gcf, 'Path\to\graphics\2_1b_plot.png');
close(gcf);
%==========================================================================

%==========================================================================
%% Performing simulation to find long-run/ergodic distribution of durable
good D
% Setting total number of simulations to perform
T = 10000;

% Initialising random number generator
rng('default');
rng(1010);

% Creating matrix of random numbers
X = rand(T, 1);

% Creating variable for standard error
Se = zeros(T, 1);
Se(X >= 0 & X < 1/3) = -0.05;
Se(X >= 1/3 & X < 2/3) = 0;
Se(X >= 2/3 & X <= 1) = 0.05;

Se_index = zeros(T, 1);
Se_index(X >= 0 & X < 1/3) = 1;
Se_index(X >= 1/3 & X < 2/3) = 2;
Se_index(X >= 2/3 & X <= 1) = 3;

% Initialising durable goods and asset levels
Ds = zeros(T, 1);
As = zeros(T, 1);
bindex = reshape(b, Ne, Na, Nd);

% Assuming initial level (i.e., at period 0) of D is 0.01, A = e = 0
D0 = 0.01;
A0 = 0;
e0 = 0;
Ds(1) = D(bindex(2, 1, 1));

if Ds(1) == D0
  As(1) = (y + A0 - p*(Ds(1)-D0*(1 - delta - Se(1))))*(1 + r);
else
  As(1) = (y + A0 - F - p*(Ds(1) - D0*(1 - delta - Se(1))))*(1 + r);
end

% Performing simulation
for i = 2:T
  [~, ida] = min(abs(As(i - 1) - A));
  idd = find(D == Ds(i - 1));
  Ds(i) = D(bindex(Se_index(i), ida, idd));
  if Ds(i) == Ds(i - 1)
    As(i) = (y + As(i - 1) - p*(Ds(i) - Ds(i - 1)*(1 - delta - Se(i))))*(1
+ r);
```

```
  else
    As(i) = (y + As(i - 1) - F - p*(Ds(i) - Ds(i - 1)*(1 - delta -
Se(i))))*(1 + r);
  end
end

% Plotting ergodic distribution of D through normalising histogram
Dh = histogram(Ds);
Dh.Normalization = 'probability';
title('Ergodic distribution of durable good D');
xlabel('Durable stock level');
ylabel('Frequency');
hold off;
saveas(gcf, 'Path\to\graphics\2_1c_plot.png');
close(gcf);
%===========================================================================
```