# Problem Set 5. Macroeconomics - Computational Assignment
# Income Fluctuation Problem

Roxana Mihet

March 3, 2015

## 1 Approximating the Income Process - Tauchen Method

The income process is:

$$y_t = exp(w_t)$$
$$w_t = \bar{w} + \rho w_{t-1} + \epsilon_t$$

where $\rho = 0.97$, and $\epsilon_t \sim N(0, \sigma_\epsilon^2)$, where $\sigma_\epsilon^2 = 0.06$. The income process in levels $y_t$ has always mean 1, independent of the variance. So, $y_t$ is log normal.

The mean of a log-normal random variable is $exp(\mu + \frac{\sigma^2}{2})$ . Solving for $\mu$ and $\sigma^2$:

$$\mathbb{E}(w) = \bar{w} + \rho\mathbb{E}(w) \Rightarrow \boxed{\mathbb{E}(w) = \frac{\bar{w}}{1-\rho} = \mu}$$

$$Var(w) = var(\bar{w}) + \rho^2 var(w) + var(\epsilon_t) = 0 + \rho^2 Var(w) + \sigma_\epsilon^2 \Rightarrow \boxed{Var(w) = \frac{\sigma_\epsilon^2}{1-\rho^2}}$$

Now that we know the mean and the variance of the random variable $w$ we can compute:

$$y_t = exp(w_t) = exp\left(\frac{\bar{w}}{1-\rho} + \frac{\sigma_\epsilon^2}{2(1-\rho^2)}\right) = 1 \Rightarrow$$
$$\frac{\bar{w}}{1-\rho} + \frac{\sigma_\epsilon^2}{2(1-\rho^2)} = 0 \Rightarrow$$
$$\frac{\bar{w}}{1-\rho} = \frac{-\sigma_\epsilon^2}{2(1-\rho)(1+\rho)} \Rightarrow \boxed{\bar{w} = \frac{-\sigma_\epsilon^2}{2(1+\rho)}}$$

We can now approximate the AR(1) process of $w_t$ with a Markov chain using the Tauchen method. We can assume that the extreme values of the grid (notice that the grid is symmetric) are:

$$w_1 = -2\sigma_w + \mu(w) = \mu(w) - 3\sqrt{\frac{-\sigma_\epsilon^2}{2(1+\rho)}}$$
$$w_N = \mu(w) + 3\sqrt{\frac{-\sigma_\epsilon^2}{2(1+\rho)}}$$

We can compute the transition probabilities $\pi_{jk} = Pr(w_t = k | w_{t-1} = j)$ by approximating the

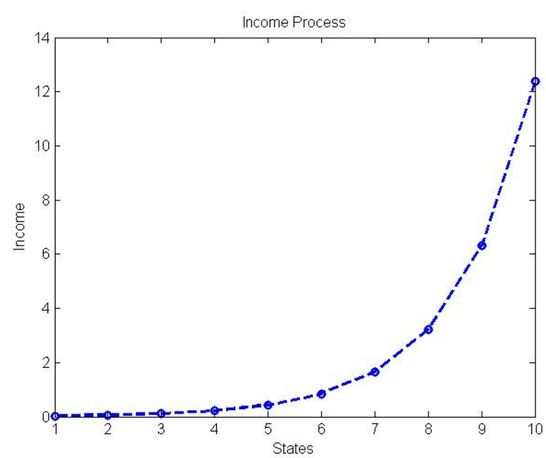
AR(1) variable to the closest grid point. Then

$$\begin{aligned}
\pi_{j1} &= P\left(\bar{w} + \rho w_j + \epsilon_t < w_1 + \frac{d}{2}\right) \\
&= P\left(\epsilon_t < w_1 + \frac{d}{2} - \bar{w} - \rho w_j\right) \\
&= F\left(w_1 + \frac{d}{2} - \bar{w} - \rho w_j\right)
\end{aligned}$$

$$\begin{aligned}
\pi_{jn} &= P\left(\bar{w} + \rho w_j + \epsilon_t > w_1 - \frac{d}{2}\right) \\
&= P\left(\epsilon_t > w_1 - \frac{d}{2} - \bar{w} - \rho w_j\right) \\
&= 1 - F\left(w_1 - \frac{d}{2} - \bar{w} - \rho w_j\right)
\end{aligned}$$

$$\begin{aligned}
\pi_{jk} &= P\left(w_k - \frac{d}{2} < \bar{w} + \rho w_j + \epsilon_t < w_1 + \frac{d}{2}\right) \\
&= F\left(w_k + \frac{d}{2} - \bar{w} - \rho w_j\right) - F\left(w_k - \frac{d}{2} - \bar{w} - \rho w_j\right)
\end{aligned}$$

The income process will look like:




While the state vectors and the transition probabilities matrices are: (please see the code and the printout at the end).

Grid:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| -3.5304 | -2.8586 | -2.1869 | -1.5152 | -0.8435 | -0.1718 | 0.5000 | 1.1717 | 1.8434 | 2.5151 |

Income states

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.0293 | 0.0573 | 0.1123 | 0.2198 | 0.4302 | 0.8422 | 1.6487 | 3.2275 | 6.3181 | 12.3684 |

Transition probabilities:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.8416 | 0.1583 | 0.0001 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.0485 | 0.8121 | 0.1393 | 0.0001 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0.0574 | 0.8207 | 0.1219 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0.0675 | 0.8264 | 0.106 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0.0789 | 0.8293 | 0.0917 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0.0917 | 0.8293 | 0.0789 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0.106 | 0.8264 | 0.0675 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0.1219 | 0.8207 | 0.0574 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0.0001 | 0.1393 | 0.8121 | 0.0485 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0001 | 0.1583 | 0.8416 |

## 2 Computing decision rule $a'(a, y)$

The consumption savings problem discussed in class was:

$$V(a, y) = \max_{c,a'} u(c) + \beta \sum_{y'} \pi(y'|y) V(a', y')$$

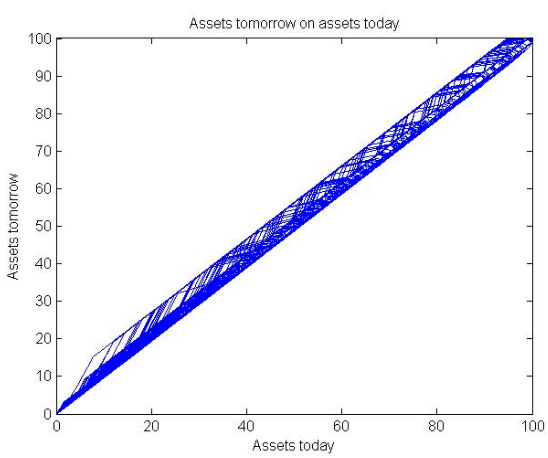
$$\text{s.t.} \quad c + a = (1 + r)a + y$$

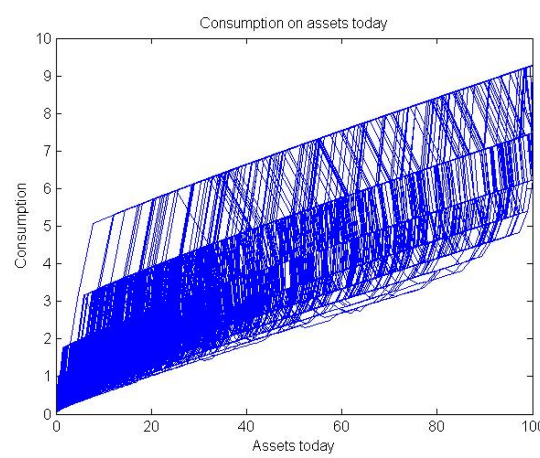
$$\text{and } a' \geq -\phi$$

$$\text{where} \quad u(c) = \frac{c^{1-\gamma}}{1-\gamma}, \ \gamma = 2, \ \beta = 0.95, \ r = 0.02$$

I will use the endogenous grid method (see code attached).

The figures below show the simulated asset and consumption policies. Notice how steep the consumption function is in the vecinity of the borrowing constraint.



## 3 Standard deviation of consumption across different risk-aversion levels

The table below shows the standard deviation when using:

- Equally spaced grid for assets
- More weight on points near the borrowing constraint
- Logarithmic spaced grid for assets

| Std $\rightarrow$ | Equally-spaced grid | More weight on points near $\phi$ | Logarithmic-spaced grid |
|---|---|---|---|
| $\gamma = 2$ | 1.59 | 0.94 | 0.82 |
| $\gamma = 5$ | 1.28 | 0.55 | 0.49 |
| $\gamma = 10$ | 0.99 | 0.35 | 0.31 |

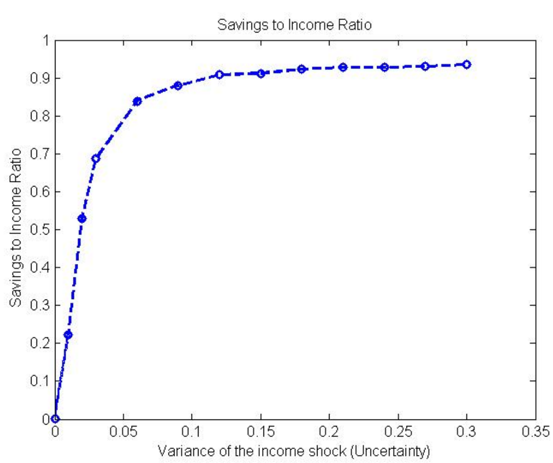
Notice that with higher levels of risk-aversion, the standard deviation of consumption decreases. This is in line with what we can expect. More risk-averse agents are going to insure themselves more and smooth consumption more. The less risk-averse agents are more willing to have a higher volatility of consumption, but the very risk-averse agents will smooth their consumption a lot.

## 4 Saving-income ratio as uncertainty increases

| Variance of shock | $\sigma_\epsilon^2 = 0.00$ | $\sigma_\epsilon^2 = 0.06$ | $\sigma_\epsilon^2 = 0.12$ |
|---|---|---|---|
| Saving-to-income ratio | 0.0000 | 0.8381 | 0.9146 |

When income is not stochastic (variance is zero), there are no savings and the savings to income ratio will be 0. There are no savings, because by consuming the entire income the agents perfectly smooth consumption and achieve the highest possible utility. On the other hand, the higher the variance of future income is, then the higher the saving-to-income ratio is. This is because of prudence. Agents will ensure against the more uncertain cases by saving more.

(see code and printout of output)



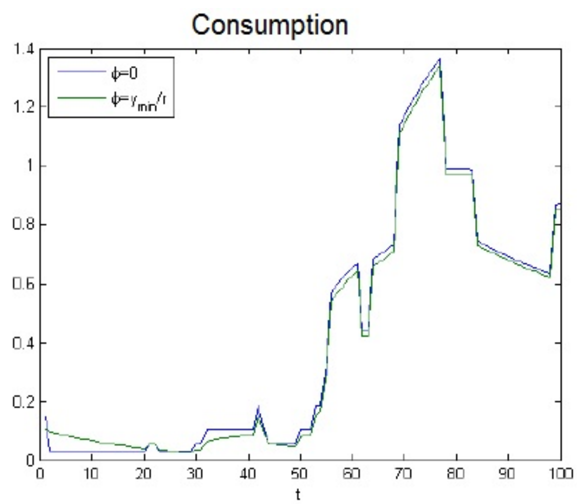
# 5 Consumption and Savings when the natural borrowing limit is different

In this section I solved the model again but imposed the natural borrowing limit as $\phi = -y_{min}/r$. Comparing the model with an ad-hoc no-borrowing constraint and the model with a natural borrowing constraint we notice that:

| | Average consumption |
|---|---|
| $\phi = 0$ | 1.91 |
| $\phi = -y_{min}/r$ | 1.41 |

Notice two things:

1. When no borrowing is allowed, the probability to hit the no-borrowing constraint is higher and therefore agents will be more prudent and save more.
2. Because agents save more, they will also have a higher consumption in the case when they are not allowed to borrow.




# 6 Insurance coefficient when the natural borrowing limit is different

Now define an insurance coefficient as the fraction of the shock $\epsilon_t$ that does not translate into consumption growth:

$$\psi = 1 - \frac{cov(\Delta ln(c_t), \epsilon_t)}{var(\epsilon_t)}$$

I solved the model again, first imposing the no-borrowing constraint and then the natural borrowing limit as $\phi = -y_{min}/r$. Comparing the model with an ad-hoc no-borrowing constraint and the model with a natural borrowing constraint notice that:

| | Insurance coefficient $\psi$ |
|---|---|
| $\phi = 0$ | 0.5442 |
| $\phi = -y_{min}/r$ | 0.5736 |

Based on the values of $\psi$, when we relax the borrowing constraint the insurance coefficient increases (after doing this exercise for several values of $\psi$ in between 0 and the natural borrowing limit, I realize that it increases in a convex way).

Notice that the model where the borrowing limit is looser (i.e. the natural borrowing limit) displays more consumption insurance. This result seems strange initially, but the possible explanation could me that with a looser borrowing constraint a bigger fraction of the shock translates into the actual consumption growth, as opposed to saving. This makes sense as there will be a smaller precautionary motive, when the agents have the option to borrow.

## 7 Ergodic distribution of assets in this economy

Now we are trying to simulate the distribution of assets/consumption directly.

This part isn't working yet.

```
ans =

27-Feb-2015 17:26:10

Part A. Tauchen N= 10
Grid:
   -3.5304   -2.8586   -2.1869   -1.5152   -0.8435   -0.1718    0.5000
1.1717    1.8434    2.5151

Income states
    0.0293    0.0573    0.1123    0.2198    0.4302    0.8422    1.6487
3.2275    6.3181   12.3684

Transition probabilities:
    0.8416    0.1583    0.0001    0.0000         0         0         0
0         0         0
    0.0485    0.8121    0.1393    0.0001    0.0000         0         0
0         0         0
    0.0000    0.0574    0.8207    0.1219    0.0000    0.0000         0
0         0         0
    0.0000    0.0000    0.0675    0.8264    0.1060    0.0000    0.0000
0         0         0
    0.0000    0.0000    0.0000    0.0789    0.8293    0.0917    0.0000
0.0000         0         0
    0.0000    0.0000    0.0000    0.0000    0.0917    0.8293    0.0789
0.0000    0.0000         0
    0.0000    0.0000    0.0000    0.0000    0.0000    0.1060    0.8264
0.0675    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.1219
0.8207    0.0574    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0001
0.1393    0.8121    0.0485
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
0.0001    0.1583    0.8416

Verify accuracy of Tauchen algorithm:
Estimated variance of w: 1.4615
Estimated rho: 0.9738

Part b) and c)
```

```
 With gamma =  2, sd of consumption is 1.54
With gamma =  5, sd of consumption is 1.29
With gamma =  10, sd of consumption is 0.98

Part d)
With var(eps) =  0.00, savings to income ratio is 0.0000
With var(eps) =  0.06, savings to income ratio is 0.8556
With var(eps) =  0.12, savings to income ratio is 0.9084

Part e) and f)
With phi =  0.00, average consumption is 1.5596, while psi = 0.5771
With phi =  -1.46, average consumption is 1.3582, while psi = 0.5634

Part g) and f)
Error using  .*
Matrix dimensions must agree.

>>
```

```
%========================================================================
% Macroeconomics II (NYU) -- Schaal - PS #5
% Roxana Mihet
% 02/25/2014
% Income Fluctuation Problem, Computational Assignment
% TO DO Spring Break: Do this in GE - see Aiyagari code.
%========================================================================

%% -----------------------------------------------------------------------
% Income Fluctuation Problem
% -----------------------------------------------------------------------
clc;
clear all;
datestr(now)                                % Displays the date and time
%% Part A (Tauchen 1986, http://www.sciencedirect.↙
com/science/article/pii/0165176586901680)
% income process: y_t=exp(w_t)
% law of motions for w: w_t=w_bar+rho*w_(t-1)+eps_t
% eps_t ~ N(0,(sigma_eps)^2)
% Find w_bar so that the income process has mean 1 (see notes)
% Compute the process for 10 states
fprintf ('Part A. Tauchen ');
% Declare parameters
rho         = 0.97;                       % Persistence coefficient
sigma_eps   = sqrt(0.06);                 % Volatility of shock
m = 3;                                    % To look at `m' standard↙
deviations
w_bar = -sigma_eps^2/(2*(1+rho));         % Mean of the random variable
N=10;                                     % Number of states (grid points)

% Tauchen algorithm
fprintf ('N= %.0f \n',N);                  % See Tauchen function

[X{2},P{2}]    = tauchen_Roxana(N,m,rho,sigma_eps,w_bar);
disp('Grid:');
disp(X{2});

disp('Income states');
disp(exp(X{2}));

figure                       % Plot the discretized income process
```

```
plot(exp(X{2}),'b--o','LineWidth',2)
title('Income Process')
xlabel('States')
ylabel('Income')

disp('Transition probabilities:');
disp(P{2});

% Verify accuracy of Tauchen algorithm
disp('Verify accuracy of Tauchen algorithm:');
[sample_states10, sample_10] = simulate_markov(X{2},P{2},20000,5,700);
var_10 = var(sample_10);
fprintf ('Estimated variance of w: %.4f \n',var_10);
rho_10 = cov(sample_10(1,1:end-1),sample_10(1,2:end))./var_10;
fprintf ('Estimated rho: %.4f \n',rho_10(2,1));
fprintf ('\n');

%% Part B, C - Solving the income fluctuation problem, finding the policy↙
functions C and A'
% I will use the endogenous grid method to solve the problem.
% In part c) we will find a'(a,y) using different gammas

% Define the parameters and the asset grid
N=10;       % compute the state space for w and transition probabilities↙
(will be same for y)
[W,P_y]    = tauchen_Roxana(N,m,rho,sigma_eps,w_bar);

y       = exp(W);          % state space for income
phi     = 0;               % borrowing limit

a_max       = 100;         % Highest value in asset grid
a_nstates   = 1000;        % Density of asset grid

r       = 0.02;            % interest rate
beta     = 0.95;            % discount factor
gammas  = [2, 5, 10];      % risk-aversion parameters

eps      = 0.01;           % Convergence Tolerance

fprintf ('Part b) and c) \n ');
% We will use the policy function from the endogenous grid method to
```

```
% simulate the economy. Since we would like to compare the three models
% besed on just hte value of gamma, we will use the same stochastic chain
% of income process for all three models:
[w_states, w_sample] = simulate_markov(W,P_y,20000,5,700);
states_seq  = w_states;
y_seq       = exp(w_sample);

for i = 1:length(gammas)
    gamma = gammas(i);
    [A{i}, C{i}] = endogeneousgrid_Roxana(y,P_y,a_max,a_nstates,phi,↙
gamma,beta,1+r,eps);

    [a_seq{i}, c_seq{i}, y_seq_burn] = sim_seq(phi,A{i},C{i},y_seq,↙
states_seq,1+r,500,500);

    sd_c(i)=std(c_seq{i});

    % assets_today=a_seq{i}(1:length(a_seq{i})-1);
    % assets_tomorrow=a_seq{i}(2:length(a_seq{i}));
    % consumption=c_seq{i};
    % plot(assets_today,assets_tomorrow)
    % title('Assets tomorrow on assets today')
    % xlabel('Assets today')
    % ylabel('Assets tomorrow')

    % plot(assets_today,consumption)
    % title('Consumption on assets today')
    % xlabel('Assets today')
    % ylabel('Consumption')

    fprintf('With gamma =  %.0f, sd of consumption is %.2f\n', gamma,↙
sd_c(i));
end;

%plot(A{1},A_star,'LineWidth',3)


fprintf ('\n');


%% Part D - Saving-to-income ratio as uncertainty increases
```

```
fprintf ('Part d) \n');
clear A C;
gamma = 2;
N=10;
sigmas   = [0  0.06   0.12 ];
% sigmas   = [0 0.01 0.02 0.03 0.06 0.09 0.12 0.15 0.18 0.21 0.24 0.27↙
0.30
% ]; % Use this when trying to plot saving-income to uncertainty

for i=1:length(sigmas)
    sigma_eps    = sqrt(sigmas(i));
    w_bar        = -sigma_eps^2/(2*(1+rho));
    [W,P_y]      = tauchen_Roxana(N,m,rho,sigma_eps,w_bar); % compute the↙
state space for w and stransition probabilities (will be same for y)
    y            = exp(W);        % state space for income
    [A{i}, C{i}] = endogeneousgrid_Roxana(y,P_y,a_max,a_nstates,phi,↙
gamma,beta,1+r,eps);

    [w_states, w_sample]     = simulate_markov(W,P_y,20000,5,700);
    states_seq               = w_states;
    y_seq                    = exp(w_sample);

    [a_seq{i}, c_seq{i}, y_seq_burn] = sim_seq(phi,A{i},C{i},y_seq,↙
states_seq,1+r,500,500);
    savings =  a_seq{i}(2:end);
    sav_inc{i} = mean(savings./((1+r)*a_seq{i}(1:end-1) + y_seq_burn'));



    fprintf('With var(eps) =  %.2f, savings to income ratio is %.4f\n',↙
sigma_eps^2, sav_inc{i});
end;

% Plot of saving-income to uncertainty
    % savtoinc=[sav_inc{1} sav_inc{2} sav_inc{3} sav_inc{4} sav_inc{5}↙
sav_inc{6} sav_inc{7} sav_inc{8} sav_inc{9} sav_inc{10} sav_inc{11}↙
sav_inc{12} sav_inc{13}];
    % plot(sigmas',savtoinc','b--o','LineWidth',2)
    % title('Savings to Income Ratio')
    % xlabel('Variance of the income shock (Uncertainty)')
    % ylabel('Savings to Income Ratio')
```

```
%% Part E and F
fprintf ('\n');
fprintf ('Part e) and f) \n');

N=10;
m=3;
rho=0.97;
sigma_eps   = sqrt(0.06);
w_bar = -sigma_eps^2/(2*(1+rho));

% compute the state space for w and transition prob (will be same for y)
[W,P_y]      = tauchen_Roxana(N,m,rho,sigma_eps,w_bar);
y            = exp(W);       % state space for income
phis = [0 -y(1)/r];

for i=1:length(phis)
    phi = phis(i);

    [A{i}, C{i}] = endogeneousgrid_Roxana(y,P_y,a_max,a_nstates,phi,↙
gamma,beta,1+r,eps);
    [w_states, w_sample]    = simulate_markov(W,P_y,20000,5,700);
    states_seq              = w_states;
    y_seq                   = exp(w_sample);
    burn = 500;
    [a_seq{i}, c_seq{i}, y_seq_burn] = sim_seq(phi,A{i},C{i},y_seq,↙
states_seq,1+r,500,burn);

    avg_c{i}=mean(c_seq{i});

    error = w_sample(burn+2:end)-rho*w_sample(burn+1:end-1);
    log_diff_c = log(c_seq{i}(2:end))-log(c_seq{i}(1:end-1));
    cov_matrix = cov(error,log_diff_c);
    cov_e = cov_matrix(2,1);
    psi{i} = 1-cov_e/var(error);

    fprintf('With phi =  %.2f, average consumption is %.4f, while psi =↙
%.4f\n', phi, avg_c{i},psi{i});

end;
```

```
%% Part G and H
%
fprintf ('\n');
fprintf ('Part g) and f) \n');

gamma=2;
[A , C ] = endogeneousgrid_Roxana(y,P_y,a_max,a_nstates,phi,gamma,beta,↙
1+r,eps);

Adist = zeros(a_nstates,1);
Adist(1) = 1;
A_Ydist = kron(Adist,pi');

aa_policy = A;
delta=1;
iter =1;

while delta>1e-5
    for m=1:a_nstates
        for n=1:a_nstates
        A_Ydist_new(m,n) = sum( sum( (aa_policy==A(m)).*A_Ydist.*repmat↙
(PI(:,n)',[a_nstates 1]) ));
        end
    end
Adist_new = sum(A_Ydist_new,2);

delta= norm(Adist_new(:)-Adist(:));
iter = iter +1;
Adist = Adist_new;
A_Ydist = A_Ydist_new;

if iter>2500
   break
end
end

Amean = Adist'*(A');
Amean2 = Adist'*(A'-Amean).^2;
Amean3 = Adist'*(A'-Amean).^3;
Amean4 = Adist'*(A'-Amean).^4;
```

```
Cmean =sum( A_Ydist(:).*c_policy(:));
Cmean2 =sum( A_Ydist(:).*(c_policy(:)-Cmean).^2 );
Cmean3 =sum( A_Ydist(:).*(c_policy(:)-Cmean).^3 );
Cmean4 =sum( A_Ydist(:).*(c_policy(:)-Cmean).^4 );


stat_dist2 = [Amean,Amean2,Amean3,Amean4; ...
             Cmean,Cmean2,Cmean3,Cmean4];

fig = figure;
[Aa,Yy] = meshgrid(A(1:500),Y);

%hist3([Asimul(:,t) Ysimul(:,t)],{A,Y});
pdf = hist3([Asimul(:,t) Ysimul(:,t)],{A(1:500),Y});
pdf(end,:) = zeros(1,N);
C = gradient(pdf');
surf(Aa,Yy,pdf',C)
view([0.1 0.5 1])
p = fig2plotly(fig);
```

```
%% Tauchen
function [X, P] = tauchen_Roxana(N,m,rho,sigma_eps,x_bar)
%{
Use Tauchen method to approximate the AR1 process of x with discrete↙
finite Markov process.

Arguments:
    N           - number of values for x
    m           - multiplied by the variance, m determines the extreme↙
values of x in the state space
    rho         - AR1 persistence coefficient
    sigma_eps   - variance of the error term
Returns:
    X - state space for x
    P - matrix of transition probabilities
%}

%% Compute the state space for x
mean_x = x_bar/(1-rho);
sigma_x = sqrt((sigma_eps)^2/(1-rho^2));   % compute variance of x
X       = linspace(mean_x-m*sigma_x,mean_x+m*sigma_x,N);   % state space↙
with extreme values
                                                           % N states↙
equally spaced
d       = X(2)-X(1); % compute the distance between grid points

%% Compute the transition probabilities
P = zeros(N);  % Create a matrix of NxN zeros
% Approximate the values that are beyond the grid with the extreme
% points, and the ones within the grid with the closest point.
for i=1:N                                                           %↙
This part fills in the rows
    P(i,1) = normcdf((X(1)+d/2-x_bar-rho*X(i)),0,sigma_eps);        % pi_↙
{j1} = CDF of Epsilon_t
        for k=2:N-1                                                 %↙
This part fills in the columns
        P(i,k) = normcdf(X(k)+d/2-x_bar-rho*X(i),0,sigma_eps) - ...
                   normcdf(X(k)-d/2-x_bar-rho*X(i),0,sigma_eps);
        end
    P(i,N) = 1 - normcdf(X(N)-d/2-x_bar-rho*X(i),0,sigma_eps);      %↙
This fills in last column
```

```
end

%% Endogenous Grid
function [A, C] = endogeneousgrid_Roxana(y,P_y,a_max,a_nstates,phi,gamma,↙
beta,R,eps)
%{
This function computes the optimal policy functions. Assuming:
utility function is u(c)=(c^(1-gamma))/(1-gamma)
hence u'(c)=c^(-gamma)

Then the Euler Equation is u'(c) = beta R E[u'(c')]
From the budget constraint c=R a + y -a', so we get:
u'(R a + y -a') = beta*R*E[u'(R a' + y' - a'')]

Arguments:
    Y           - state space for the income process
    P_y         - transition probability matrix for the income process
    a_max       - upper bound on borrowing limit
    a_nstates   - how fine the grid for borrowing is
    phi         - lower bound on the borrowing limit
    gamma       - risk aversion coefficient
    beta        - discount factor
    R           - gross interest rate
    eps         - tolerance criteria
Returns (policy rules):
    A           - a matrix of decision rule for future assets a'(a,y)
    C           - a matrix for decision rules for consumption

%}

%% Step 1: Determine a process for y and set up a grid for state space A↙
x Y
% Construct a state space grid, putting more points near where the
% constraint binds: near phi.
 a = linspace(phi,a_max,a_nstates);
% a = linspace(phi,a_max,a_nstates)./sqrt(linspace(1,a_max,a_nstates));
% a  = linspace(0,log(a_max+1-phi),a_nstates);
[Y,A] = meshgrid(y,a);

%% Step 2: Essentially we'll guess the initial consumption and solve the↙
model, then
```

```
% iterate on our consumption guess until the consumption decision is
% essentialy the same in two consecutive periods.
% A good initial guess is C=RA+Y. \\No borrowing, no saving.
C = R*A + Y;

found = 0;
while found == 0
    %% Step 3&4: Because we're looking for a steady state solution
a'=a'', then the EE becomes
    % u'(c(a',y)) = beta*R*E[u'(c(a',y'))]  if the constraint doesn't
bind
    % Call the RHS B(a',y), then  B(a',y) = beta*R*SUM(c(a',y')^(-gamma)
*P_y(y',y))
    B = zeros(a_nstates,length(y));
    for j=1:length(y)
       B(:,j) = beta*R*(C.^(-gamma)*P_y(j,:)');
    end
    % Now we'll recover c(a',y) from the LHS of the equation
    C_tilde = B.^(-1/gamma);
    %% Step 5: From the budget constraint we can now solve for A*
    A_star = (C_tilde + A - Y)/R;
    %% Step 6: In case A* is not on the grid, we interpolate by assigning
it values closest on the grid
    C_tilde_interp = zeros(size(C_tilde));
    for j=1:length(y)
       C_tilde_interp(:,j) = interp1(A_star(:,j),C_tilde(:,j),A(:,
j),'linear','extrap');

       % if A_star(1,j) > phi, then interpolations for values in A
between phi
       % and A_star(1,j) don't make sense (since the borrowing constraint
       % already binds.
       % Hence, overwrite those values in C_tilde_interp just using the
       % borrowing constraint:
       i = 1;
       while A(i,j) < A_star(1,j)
          C_tilde_interp(i,j) = R*A(i,j)+Y(i,j)-phi;
          i = i+1;
       end
    end
    % Check if the two consumptions are the same by comparing the maximum
```

```
    % difference between decision rules
    if max(max(abs(C_tilde_interp-C))) < eps
        found = 1;
    end
    C = C_tilde_interp;
end;
%% Sim_seq
function [a_seq, c_seq, y_seq] = sim_seq(phi,A,C,y_seq,states_seq,R,↙
a_ini,burn)
%{
This function computes the sequence of assets and consumption taking the
income process as given.
Arguments:
    phi          - borrowing limit
    A            - decision rule for assets (on A x Y grid)
    C            - decision rule for consumption (on A x Y grid)
    y_seq        - sequence of income
    states_seq   - sequence of income states
    R            - gross interest rate
    a_ini        - starting point for asset holdings (wouldn't matter if↙
burn is high)
    burn         - how many sample points to exclude from the observation
Returns:
    a_seq        - sequence for asset holdings
    c_seq        - sequence of consumption
%}
T = length(y_seq);
c_seq = zeros(T,1);
a_seq = zeros(T+1,1);
a_seq(1) = A(a_ini,1);

for t=0:T-1
   c_seq(t+1) = interp1(A(:,states_seq(t+1)),C(:,states_seq(t+1)),a_seq↙
(t+1),'linear','extrap');
   a_seq(t+2) = R*a_seq(t+1) + y_seq(t+1) - c_seq(t+1);
   if a_seq(t+2) < phi
    a_seq(t+2) = phi;
   % c_seq(t+1) = R*a_seq(t+1) + y_seq(t+1)-a_seq(t+2);
   end;
   if a_seq(t+2) > max(max(A));
    a_seq(t+2) = max(max(A));
```

```
    % c_seq(t+1) = R*a_seq(t+1) + y_seq(t+1)-a_seq(t+2));
    end;
end

c_seq = c_seq(burn+1:end);
a_seq = a_seq(burn+1:end);
y_seq = y_seq(burn+1:end);

%% Simulate Markov
function [states,X_sim] = simulate_markov(X,P,N,start,burn)
%{
This function simulates a Markov process.
Arguments:
    X       - state space
    P       - transition probability matrix
    N       - desireable sample size
    start   - starting point (1 to size of state space!) (wouldn't matter↙
if burn is high)
    burn    - how many sample points to exclude from the observation
Returns:
    X_sim   - a simulated Markov process
%}

transC = [zeros(size(P,1),1), cumsum(P,2)];
states = zeros(1,N); %storage of states
states(1) = start; %start at state 1 (or whatever)
rr = rand(1,N+burn); %array of random numbers uniformely distributed on↙
(0,1)
for ii = 2:N+burn
     [~,states(ii)] = histc(rr(ii),transC(states(ii-1),:));
     if ii>burn, X_sim(ii-burn) = X(states(ii)); end;
end
states = states(1,burn+1:end);
```