IMF, Problem Set 3

1068576

17 December 2023

Q1

The household's problem is:

$$\max_{\{C_{i,t},C^H_{i,t},C^F_{i,t},L_{i,t},B_{i,t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t \log\left(C_{i,t} - \frac{L_{i,t}^{1+\eta}}{1+\eta}\right) \tag{1}$$

where:

$$C_{i,t} \equiv (C^H_{it})^{\alpha_H}(C^F_{i,t})^{(1-\alpha_H)} \tag{2}$$

subject to the budget constraint:

$$P_t C_{it} = P^H_t C^H_{i,t} + P^F_t C^F_{i,t} = (1-\tau_t)W_t s_{i,t} L_{it} + P^H_t B^H_{i,t} R_t - P^H_t B^H_{i,t+1} \tag{3}$$

and the borrowing constraint:

$$B_{i,t+1} \geq -\kappa \tag{4}$$

The Lagrangian is:

$$\begin{aligned}\mathcal{L} = \sum_{t=0}^{\infty} \beta^t \Bigg[& \log\left((C^H_{it})^{\alpha_H}(C^F_{i,t})^{(1-\alpha_H)} - \frac{L_{i,t}^{1+\eta}}{1+\eta}\right) + ... \\ & \lambda_{i,t}\left((1-\tau_t)W_t s_{i,t} L_{it} + P^H_t B^H_{i,t} R_t - P^H_t B^H_{i,t+1} - P^H_t C^H_{i,t} - P^F_t C^F_{i,t}\right) + ... \\ & \mu_{i,t}(B_{i,t+1} + \kappa)\Bigg]\end{aligned} \tag{5}$$

The first order conditions are:

$$\frac{\partial \mathcal{L}}{\partial C_{i,t}} = \frac{1}{C_t - \frac{L_{i,t}^{1+\eta}}{1+\eta}} - \lambda_{i,t} P_t = 0 \tag{6}$$

$$\frac{\partial \mathcal{L}}{\partial C_{i,t}^H} = \frac{\alpha_H (C_{i,t}^H)^{(\alpha_H - 1)} (C_{i,t}^F)^{(1-\alpha_H)}}{(C_{it}^H)^{\alpha_H} (C_{i,t}^F)^{(1-\alpha_H)} - \frac{L_{i,t}^{1+\eta}}{1+\eta}} - \lambda_{i,t} P_t^H = 0 \tag{7}$$

$$\frac{\partial \mathcal{L}}{\partial C_{i,t}^F} = \frac{(1-\alpha_H)(C_{i,t}^H)^{\alpha_H} (C_{i,t}^F)^{-\alpha_H}}{(C_{it}^H)^{\alpha_H} (C_{i,t}^F)^{(1-\alpha_H)} - \frac{L_{i,t}^{1+\eta}}{1+\eta}} - \lambda_{i,t} P_t^F = 0 \tag{8}$$

$$\frac{\partial \mathcal{L}}{\partial L_{i,t}} = \frac{-L_{i,t}^{\eta}}{C_{i,t} - \frac{L_{i,t}^{1+\eta}}{1+\eta}} - \lambda_{i,t}(1-\tau_t) W_t s_{i,t} = 0 \tag{9}$$

$$\frac{\partial \mathcal{L}}{\partial B_{i,t}} = -\beta^t \lambda_{i,t} P_t^H + \beta^t \mu_{i,t} + \beta^{t+1} \lambda_{i,t+1} P_{t+1}^H R_{t+1} = 0 \tag{10}$$

with the complementary slackness condition:

$$\begin{gathered} \mu_{i,t}(B_{i,t+1} + \kappa) = 0, \\ \text{with } \mu_{i,t} > 0 \text{ or } (B_{i,t+1} + \kappa) > 0 \end{gathered} \tag{11}$$

Combining equations (7) and (8), I have the home-foreign consumption allocation:

$$P_t^H C_t^H = P_t^F C_t^F \tag{12}$$

From equations (6) and (9), I obtain the labour-consumption trade-off:

$$L_{i,t} = \left[\frac{(1-\tau_t) s_{i,t} W_t}{P_t} \right]^{\frac{1}{\eta}} \tag{13}$$

Finally, equations (6) and (10) give us the Euler equation:

$$\frac{1}{\left[C_t - \frac{L_{i,t}^{1+\eta}}{1+\eta} \right]} = \beta \mathbb{E} \left[R_{t+1} \frac{1}{\left[C_{t+1} - \frac{L_{i,t+1}^{1+\eta}}{1+\eta} \right]} \right] + \frac{\mu_{i,t}}{P_t^H} \tag{14}$$

Those equations define the household's optimality conditions, in addition to the complementary slackness condition.

Next, I look at the firm's problem:

$$\max_{Y_t, L_t} P_t^H Y_t - W_t L_t \tag{15}$$

given the production technology:

$$Y_t = L_t \tag{16}$$

The firm's first order condition is:

$$W_t = P_t^H \tag{17}$$

Q2

To close the model, I need to define the foreign demand of home goods and foreign output. For simplicity, I fix the foreign demand of home goods using US export to the rest of the world (RoW) as a percentage of US GDP. From the Federal Reserve Economic Data (FRED), I observe that the average share of US export between 2012 and 2022 was $\gamma = 0.119$, or 11.9% of GDP. I therefore include the following equation to the model:

$$C_t^{H*} = \gamma Y_t \tag{18}$$

Foreign consumption of home goods can then be used to derive home consumption of home goods as a function of output from the market clearing condition:

$$C_t^{H} = Y_t - C_t^{H*} \tag{19}$$

Moreover, I define foreign output as $Y_t^* = 75$, which corresponds to world output excluding the US in 2022, in trillion of US dollars. I also specify the share of RoW output consumed by the US, which was approximately $\omega = 0.047$, or 4.7% of RoW output. Therefore, I add:

$$C_t^{F} = \omega Y_t^* \tag{20}$$

to the model as well.

Q3

Household productivity is modelled as an AR(1) process, which was discretised using Tauchen's method with 10 grid points. The parameters used to calibrate the process were obtained by fitting an AR(1) model on a time series of labour productivity obtained from FRED, for the period 1996-2022. The output of the AR(1) regression are presented in table (1).

Table 1: Regression output on labour productivity

	Value	Standard Error	T-statistic
Constant	0.841	0.502	1.676
AR(1)	0.995	0.005	187.29
Variance	0.587	0.037	15.734

Moreover, I scale the mean of the productivity shock by a factor of 10,000 so that output is comparable in magnitude to US GDP.

All preference parameters are calibrated according to values that are standard in the macroeconomic literature. I assume no home bias, which implies $\alpha_H = \frac{1}{2}$. The tax rate τ is computed to reflect government tax revenue as a share of GDP in the US, while government spending is calibrated such that the resulting debt level reflects the average US debt-to-GDP ratio between 2012 and 2022 (approximately 120%). Finally, the upper bound on the asset grid was calibrated on the amount of net wealth held by the top .1% household in the US, while the lower bound was defined as an arbitrarily low number to ensure that the borrowing constraint binds for some households. The calibration parameters are presented in table (3).

Table 2: Calibration values

Description	Notation	Calibration value	Source
subjective discount factor	β	0.99	
Frisch elasticity	η	3	Chetty et al. (2011)
home bias	α_H	0.5	assumption
tax revenue (% of GDP)	τ	.30	OECD
government spending (tn. USD)	g	6.61	
net wealth upper bound (tn. USD)	b_{max}	18	FRED
net wealth lower bound (tn. USD)	b_{min}	-3	

Note that government spending in this model is calibrated at 6.61 trillion US dollar, which is approximately equal to the the federal government expenditure of 6.03 trillion US dollar in 2022 (FRED). The output targeted in the calibration are as follows:

Table 3: Targeted values

Description	Notation	Model value	Data value (2022)
US GDP (tn. USD)	y	25.47	25.46
government debt	$-bg$	30.26	30.8

Q4 to Q6

The benchmark model was obtained under a tax rate and government spending set exogenously, as described in Q3, with government debt endogenous. Moreover, bond market clearing was enforced such that the economy does not trade bonds with the RoW. By contrast, the model in Q5 treats government bond and government spending as exogenous variables and the government adjusts tax to satisfy its budget constraint. Finally, Q6 treats government debt and tax as exogenous and government spending as an endogenous variable. Both Q5 and Q6 allow foreign holding of home bonds to adjust such that the bond market always clears. The resulting real interest rates and terms of trade are shown in table (4), while the wealth distribution of household is shown in figure (1).

Table 4: Targeted values

	Benchmark	Endogenous Tax	Endogenous govt expenditure
Y	25.47	26.54	25.50
r	0.03	-0.06	0.03
$\frac{P^H}{P^F}$	0.223	0.210	0.224
τ	0.30	0.18	0.30
g	6.61	6.61	6.75

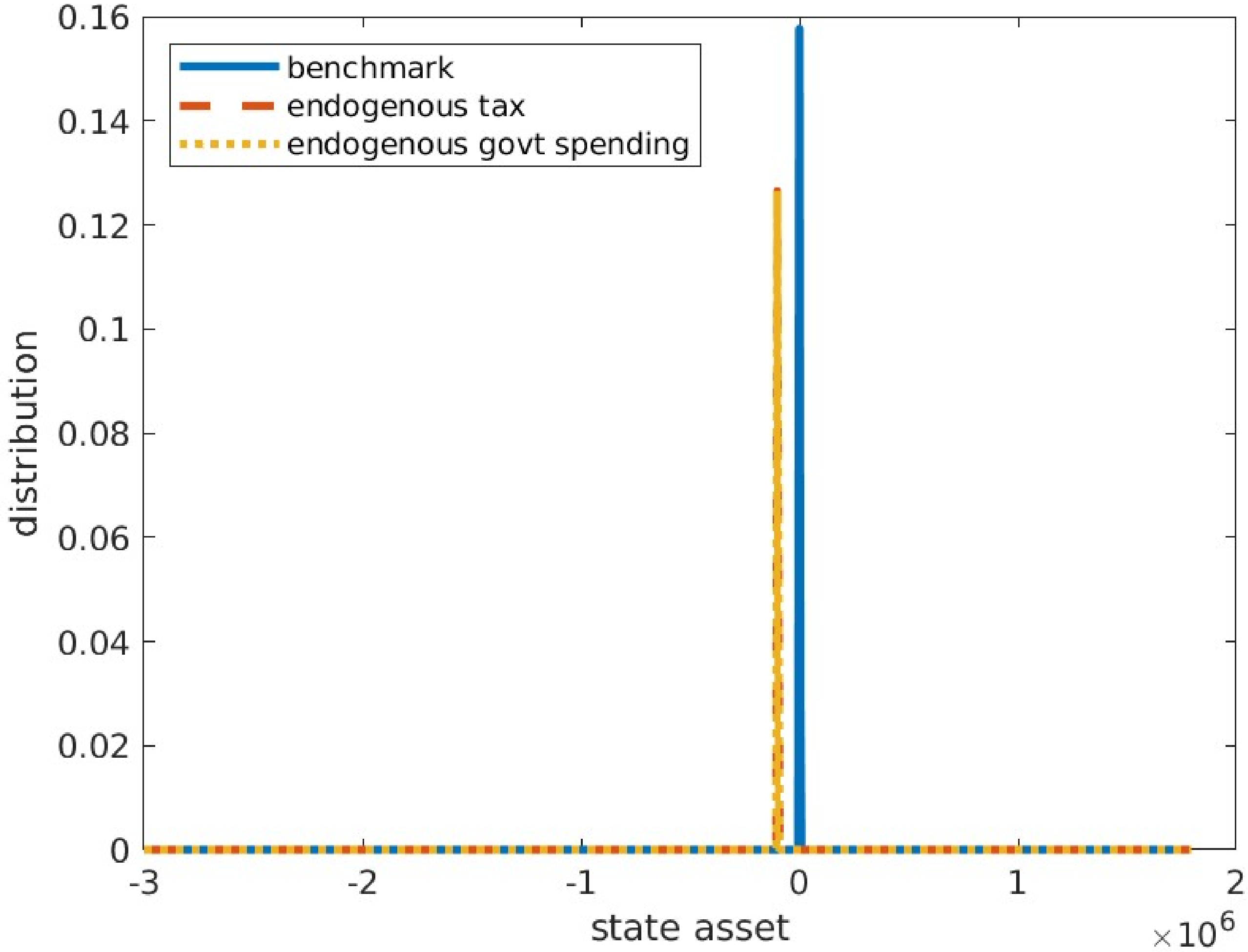

Figure 1: Utility comparative statics

When foreign lending increases and government bond supply is fixed, the real interest decreases for the bond market to clear. As a result, domestic borrowing increases, as shown in the policy functions in figure (2). Moreover, the decrease in interest leaves more fiscal room for the government, which is a net debtor. In the case where tax is endogenous, the government decreases income tax. As a result, labour supply increases as wage net of tax increases (see eq. 13). The resulting surge in labour supply increases home output, allowing households to consume more home goods (see figure 3). From equation 12, this implies that the terms of trade falls.

In the case of endogenous government spending, interest rate also decreases initially for the bond market to clear. To satisfy its budget constraint, the government increases government spending, which crowds out private consumption in the home goods market. As a result, keeping home output constant, home goods consumption decreases relative to foreign goods consumption, which increases the terms of trade compared to the benchmark. At the same time, households react to the decrease in the interest rate by borrowing more, which brings interest back to equilibrium.

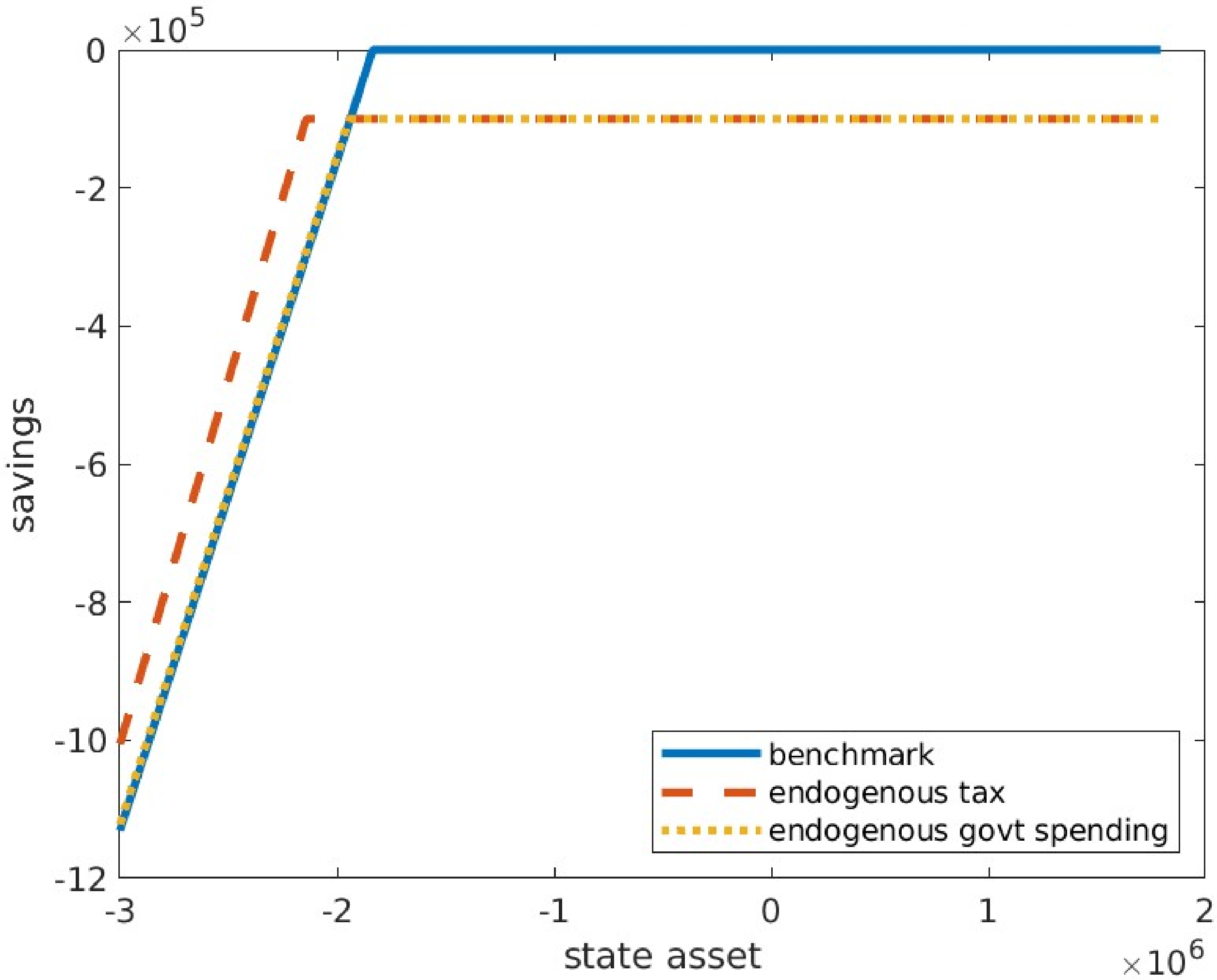

Figure 2: Savings policy function

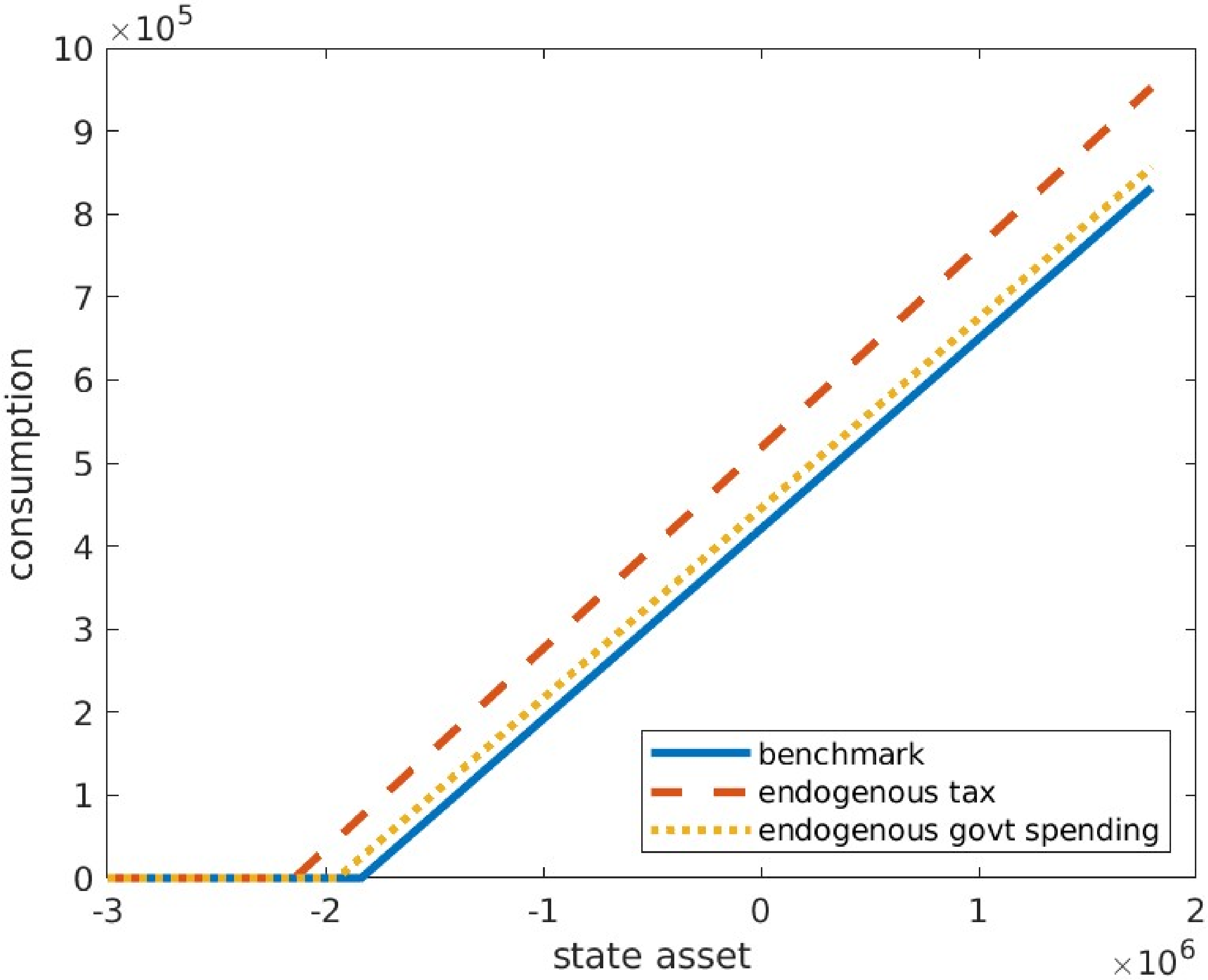

Figure 3: Consumption policy function

By comparing utility across wealth levels, we observe that the decrease in interest rate results in borrowers being significantly better off compared to the benchmark between state debt of approximately $-1.5e6$ and 0. For lenders on the other hand, the benefits from higher consumption are partially cancelled out by the effects of lower interest compared to the benchmark.

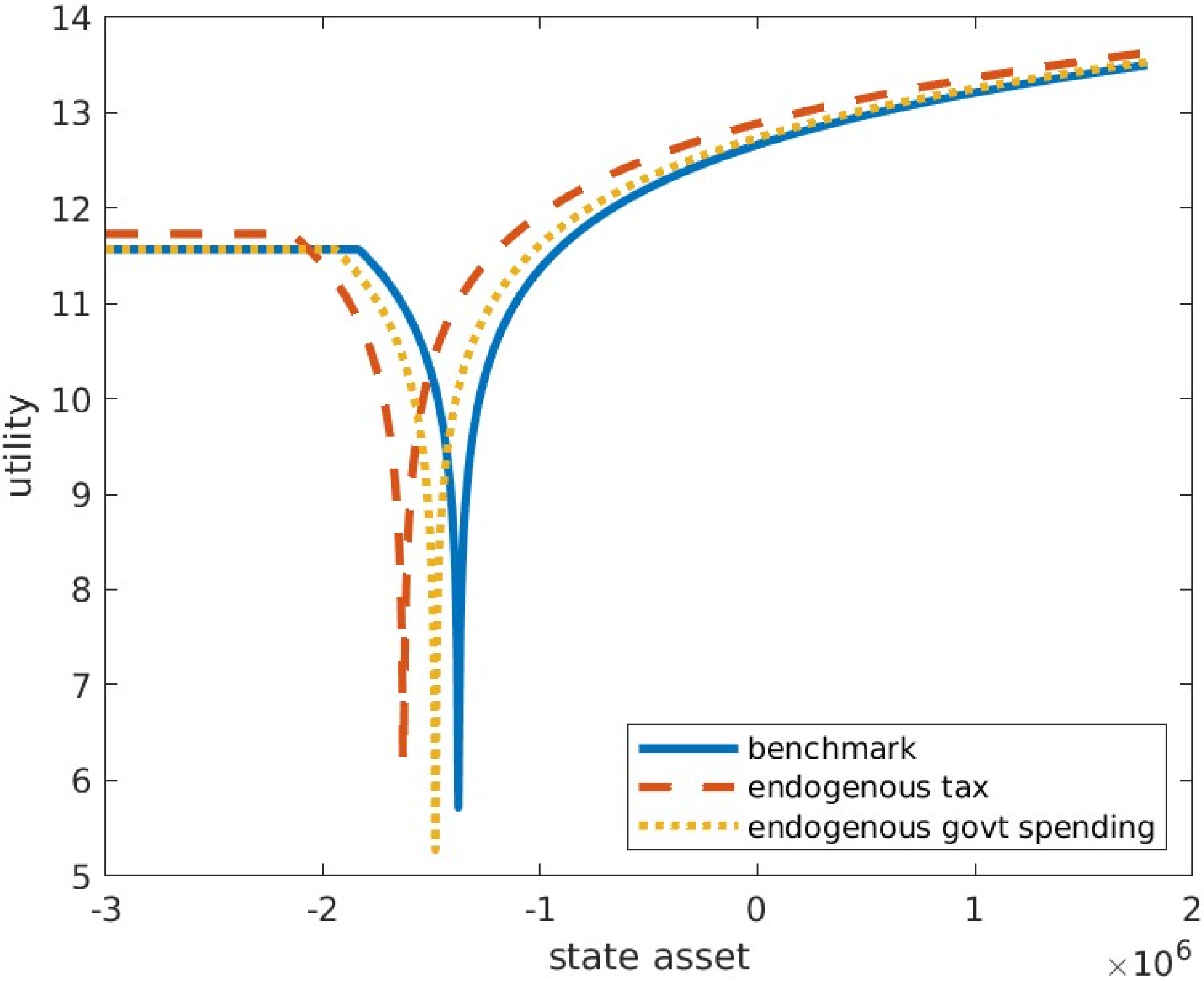

Figure 4: Utility comparative statics

Appendix A: Main File

```
%-----------------------------------------------------------------------
%
% Title: International Macro-Finance Problem Set 3, main file
% Author: ----
% Date: 25/11/2023
% Description: Open economy heterogeneous agent model
%
%-----------------------------------------------------------------------


%-----------------------------------------------------------------------
%% 0. Housekeeping (close all graphic windows)
%-----------------------------------------------------------------------

close all;
clear all;

cd '/home/----/Desktop/International Macro/PS3';

%-----------------------------------------------------------------------
%% 1. Loading results
%-----------------------------------------------------------------------

load("PS3Q4.mat");

eta = par.eta;
grid_b = grid.b_fine';

r0Q4 = guess.r0;
ciQ4 = policy.ci;
biQ4 = policy.bi;
G0Q4 = G0;
liQ4 = policy.li;
yQ4 = y;
totQ4 = guess.ph_pf;
tauQ4 = par.tau;
gQ4 = par.g;
bgQ4 = bg;
b_starQ4 = par.b_star;

clearvars -except grid_b eta *Q4


load("PS3Q5.mat");

r0Q5 = guess.r0;
ciQ5 = policy.ci;
biQ5 = policy.bi;
G0Q5 = G0;
liQ5 = policy.li;
yQ5 = y;
totQ5 = guess.ph_pf;
tauQ5 = guess.tau;
gQ5 = par.g;
bgQ5 = par.bg;
b_starQ5 = b_star;

clearvars -except grid_b eta *Q4 *Q5


load("PS3Q6.mat");

r0Q6 = guess.r0;
ciQ6 = policy.ci;
biQ6 = policy.bi;
G0Q6 = G0;
liQ6 = policy.li;
yQ6 = y;
totQ6 = guess.ph_pf;
tauQ6 = par.tau;
gQ6 = g;
```

```matlab
bgQ6 = par.bg;
b_starQ6 = b_star;

clearvars -except grid_b eta pi *Q4 *Q5 *Q6

%-------------------------------------------------------------------------
%% 2. Comparative statics
%-------------------------------------------------------------------------

G0Q41 = G0Q4 * pi';
G0Q51 = G0Q5 * pi';
G0Q61 = G0Q6 * pi';



figure
    plot(grid_b, G0Q41, '-','LineWidth', 2.5)
    hold on
    plot(grid_b, G0Q51, '--', 'LineWidth', 2.5)
    plot(grid_b, G0Q61, ':', 'LineWidth', 2.5)
    xlabel('state asset')
    ylabel('distribution')
    legend('benchmark', 'endogenous tax', 'endogenous govt spending', 'Location', 'northwest' )


saveas(gcf, 'Distribution_Q4.jpg');


utilQ4 = log(ciQ4 - liQ4.^(1 + eta)/ (1 + eta));
utilQ41 = utilQ4 * pi';

utilQ5 = log(ciQ5 - liQ5.^(1 + eta)/ (1 + eta));
utilQ51 = utilQ5 * pi';

utilQ6 = log(ciQ6 - liQ6.^(1 + eta)/ (1 + eta));
utilQ61 = utilQ6 * pi';


figure
    plot(grid_b, utilQ41, '-','LineWidth', 2.5)
    hold on
    plot(grid_b, utilQ51, '--', 'LineWidth', 2.5)
    plot(grid_b, utilQ61, ':', 'LineWidth', 2.5)
   % title('Utility: comparative statics')
    xlabel('state asset')
    ylabel('utility')
    legend('benchmark', 'endogenous tax', 'endogenous govt spending', 'Location', 'southeast' )

saveas(gcf, 'Utility_Q5Q6.jpg');

% %%
% figure
%     plot(grid_b, sum(ciQ4 .* G0Q4, 2), '-','LineWidth', 2.5)
%     hold on
%     plot(grid_b, sum(ciQ5 .* G0Q5, 2), '--', 'LineWidth', 2.5)
%     plot(grid_b, sum(ciQ6 .* G0Q6, 2), ':', 'LineWidth', 2.5)
%     title('Consumption Policy Function')
%     xlabel('state asset')
%     ylabel('consumption')
%     legend('benchmark', 'endogenous tax', 'endogenous govt spending')
% %%



figure
    plot(grid_b, ciQ4(:, 1), '-','LineWidth', 2.5)
    hold on
    plot(grid_b, ciQ5(:, 1), '--', 'LineWidth', 2.5)
    plot(grid_b, ciQ6(:, 1), ':', 'LineWidth', 2.5)
   % title('Consumption Policy Function')
    xlabel('state asset')
    ylabel('consumption')
    legend('benchmark', 'endogenous tax', 'endogenous govt spending', 'Location', 'southeast' )

```

```
saveas(gcf, 'Consumption_policy_Q4Q5Q6.jpg');



figure
    plot(grid_b, biQ4(:, 1), '-','LineWidth', 2.5)
    hold on
    plot(grid_b, biQ5(:, 1), '--', 'LineWidth', 2.5)
    plot(grid_b, biQ6(:, 1), ':', 'LineWidth', 2.5)
    % title('Lending Policy Function')
    xlabel('state asset')
    ylabel('lending')
    legend('benchmark', 'endogenous tax', 'endogenous govt spending', 'Location', 'southeast')

saveas(gcf, 'Debt_policy_Q4Q5Q6.jpg');
```

Appendix B: Q4 model file

```


%----------------------------------------------------------------
%
% Title: International Macro-Finance Problem Set 3, Q4 model file
% Author: ---
% Date: 10/12/2023
% Description: Open economy heterogeneous agent model
%
%----------------------------------------------------------------


%----------------------------------------------------------------
%% 0. Housekeeping (close all graphic windows)
%----------------------------------------------------------------

close all;
clear all;

cd '/home/---/Desktop/International Macro/PS3';

%----------------------------------------------------------------
%% 1. Calibration & defining parameters
%----------------------------------------------------------------


% preferences
par.beta = .99;
par.eta = 3; % standard Frisch elasticity of 3 according to macro literature (Chetty et al. 2011)
par.alphah = .5; % no home bias at home

% fiscal policy
par.tau = .3; % US govt revenue as a share of gdp, 2012-22 average
par.g = 6.61174; % adjust govt spending until I reach a debt to GDP ratio of 120%


% rest of the world
par.y_star = 75; % world excld. US (RoW) GDP, 2022 in trillion USD (FRED)
par.b_star = 0; % foreign holding of home bonds
par.gamma = .119; % US export to RoW as share of US GDP, 2012-22 average (FRED)
par.omega = .047; % US import from RoW as a share of RoW GDP, 2012-22 average (FRED)

% others
par.kappa = 1;
b_ss = 138; % aggregate net worth in the US in 2022, trillion USD (FRED)
par.b_shareTop01 = .13; % share of net worth held by top .1% in 2022 (SCF)
par.b_shareBot01 = -.0007 * 5; % share of net worth held by bottom .1% in 2022 (SCF)


%----------------------------------------------------------------
%% 2. Defining the exogenous shock process
%----------------------------------------------------------------


% Exogeneous shock
% par.scale = 1;
% grid.s1 = [.5 1.5];  % value
% grid.s1 = grid.s1 * par.scale ;
% Pr = [.8 .2;
%       .2 .8]; % Markov chain
% ns = length(grid.s1); % number of states
% pi = ones(1, ns) / ns;
% dis = 1; % initial distribution of HHs
% tol=1e-20;
% % compute invariant distribution for s
% while dis>tol
%     pi_2 = pi * Pr;
%     dis = max(abs(pi_2 - pi));
%     pi = pi_2;
% end

ns    = 10;        % Number of points for labour productivity process
par.rhos  = .995;       % Calibrate persistence of AR(1), from FRED time series
```

```
par.sigs  = sqrt(.6);       % Calibrate standard deviation of innovation
par.scale = 1e+5;         % scaling factor for GDP
[grid.s1, Pr] = rouwenhorst(ns, 0, par.rhos, par.sigs);
% grid.s1 = rescale(grid.s1', .5 * par.scale, 1.5 * par.scale);
grid.s1 = exp(grid.s1);  % labour productivity process (rule out negatives)

maxIter_pi = 1e+5;
Pr_ergo = Pr^maxIter_pi; % Ergodic distribution of exogenous labour productivity process
pi = Pr_ergo(1,:);
grid.s1 = (grid.s1 / (pi * grid.s1) + par.scale)'; % Normalize mean to 1 (or to 100)



%-------------------------------------------------------------------------
%% 3. Setting up the grids
%-------------------------------------------------------------------------

% Asset grid:
b_con = -par.kappa * par.scale;
b_min = 30 * b_con; % ensures it is significantly below b_con
b_max = par.b_shareTop01 * b_ss * par.scale; % approximately the net worth of the top .1%
nb = 1000;
nb_fine = 1000;
grid.b = linspace(b_min, b_max, nb);
grid.b = [grid.b, b_con]';
grid.b = sort(grid.b);
Ind_b_min = find(grid.b == b_con);
nb = length(grid.b); % update grid point number to account for kappa
grid.b_fine = linspace(b_min, b_max, nb_fine);


% Build some usefull matrices
bg = ones(nb, ns);
grid.b2 = repmat(grid.b, [1 ns]); % nb x ns matrix
grid.s2 = repmat(grid.s1, [nb 1]); % nb x 2 matrix


%-------------------------------------------------------------------------
%% 4. Guesses & policy functions
%-------------------------------------------------------------------------

% consumption
guess.ci = ones(nb,ns);

% terms of trade
guess.ph_pf = 2;

% interest rate
guess.r0 = 1 / par.beta;

% consumption
cf = par.omega * par.y_star; % home consumption of foreign goods (i.e. US import from RoW)

% distribution
G0 = ones(nb_fine, ns) / (nb_fine * ns);

% policy functions
policy.ci = ones(nb_fine, ns);
policy.bi = ones(nb_fine,ns);

% convergence parameters
maxIter_c = 1e+3;
maxIter_r = 1e+3;
maxIter_tot= 1e+3;
lambda =.5;
errtol_c = 1e-5;
errtol_r = 1e-2;
errtol_tot = 1e-5;


%-------------------------------------------------------------------------
%% 5. Iterations
%-------------------------------------------------------------------------

```

```
iter_tot = 1;
err_tot = 1;

while err_tot > errtol_tot && iter_tot <= maxIter_tot

  w_p0 = par.alphah^par.alphah * (1 - par.alphah)^(1 - par.alphah) * ...
           guess.ph_pf^(1 - par.alphah); % update real wage from the tot

  %% Compute labour supply
  policy.li = ((1 - par.tau) .* w_p0 .* grid.s1).^(1 / par.eta);
  l0 = pi * policy.li.';

  iter_r = 1;
  err_r = 1;
  %%

  while err_r > errtol_r && iter_r <= maxIter_r

    %% Solve for consumption
    y = l0;

    % Compute constrained consumption given R
    c_constrained = (1 - par.tau) .* guess.ph_pf .* grid.s2 .* policy.li + ...
                    w_p0 .* grid.b2 .* guess.r0 - ...
                    w_p0 .* b_con;
    c_constrained = max(c_constrained, 1e-5); % rule out consumption below 0

    iter_c = 1;
    err_c = 1;

    guess.ci = ones(nb,ns);
    %%

    while err_c > errtol_c && iter_c <= maxIter_c
        %%

      % expected marginal utility at t+2
      Emup1 = (guess.ci * par.scale - policy.li.^(1 + par.eta) / (1 + par.eta)).^(-1);
      % expected marginal utility at t+1 (scale up consumption)
      Mup = par.beta * guess.r0 * Emup1 * Pr';
      % expected consumption at t+1
      Ec = Mup.^(-1) + policy.li.^(1 + par.eta) / (1 + par.eta);
      % state debt tomorrow (i.e. choice debt today)
      bi_state = (Ec ./ w_p0 + grid.b2 - (1 - par.tau) .* grid.s2 .* ...
                        policy.li) ./ (guess.r0);

      c_new = ones(nb, ns);

      for j=1:ns

          c_new(:,j) =  (grid.b > bi_state(Ind_b_min, j)) .* ... % indicator function on whether borrowing constraint is binding
                        interp1(bi_state(:, j), Ec(:, j), grid.b, 'pchip') + ... % interpolate c_s = f(bi_state) at each grid point
                        (grid.b <= bi_state(Ind_b_min, j)) .* c_constrained(:,j); % if constraint is binding, then c_constrained
          c_new(:,j) =  max(c_new(:,j), 1e-5); % rules out negative values
      end

      err_c = max(max(abs(c_new - guess.ci)));

      guess.ci = c_new * lambda + (1 - lambda) * guess.ci;

      iter_c
      err_c

      iter_c = iter_c + 1;
          %%
    end

    % Write the policy function for consumption
    for j=1:ns
      policy.ci(:,j) = interp1(grid.b, guess.ci(:,j), grid.b_fine);
    end
```

```
    %% Solve for interest

    % Write the policy function for assets
    bi_choice  = (grid.b2 * guess.r0 + (1 - par.tau) * grid.s2 .* policy.li - ...
                guess.ci ./ w_p0);

    for j=1:ns
        policy.bi(:,j) = interp1(grid.b, bi_choice(:,j), grid.b_fine);
    end

    % Compute the endogenous distribution
    trows = zeros(nb_fine * ns * ns * 2, 1);
    tcols = trows;
    tvals = tcols;
    index = 0;
    for j=1:ns
        for bi = 1:nb_fine
            [vals,inds] = basefun(grid.b_fine, nb_fine, policy.bi(bi,j));
            for jp=1:ns
                index = index + 1;
                trows(index) = bi + (j - 1) * nb_fine;
                tcols(index) = inds(1) + (jp - 1)* nb_fine;
                tvals(index) = Pr(j, jp) * vals(1);
                index = index+1;
                trows(index) = bi + (j - 1) * nb_fine;
                tcols(index) = inds(2) + (jp - 1) * nb_fine;
                tvals(index) = Pr(j, jp) * vals(2);
            end
        end
    end
    transMat = sparse(trows, tcols, tvals, nb_fine * ns, nb_fine * ns);
    [EigVec, EigVal] = eigs(transMat.', 1);
    EigVec = EigVec / sum(EigVec);
    EigVec(EigVec < 0) = 0;
    EigVec = EigVec / sum(EigVec);
    G0 = reshape(EigVec / sum(EigVec), [nb_fine ns]); % distr. of HHs across assets & states

    % update guess for r
    b = sum(sum(policy.bi .* G0)); % aggregate HH borrowing
    bg = -(b + par.b_star); % solve for govt borrowing from bond market clearing
    r_new = (bg + par.g - par.tau * y) / bg; % update r from govt BC

    err_r = abs(r_new - guess.r0);

    guess.r0 = lambda * r_new + (1 - lambda) * guess.r0;

    iter_r
    err_r

    iter_r = iter_r + 1;
    %%

  end

  %% Solve for terms of trade
  ch_star = par.gamma * y; % US export to the RoW
  ch = y - par.g - ch_star; % consumption of home goods from market clearing
  ph_pf_new = cf / ch; % domestic price of foreign goods from the FOC

  err_tot = max(max(abs(ph_pf_new - guess.ph_pf)));

  guess.ph_pf = lambda * ph_pf_new + (1 - lambda) * guess.ph_pf;

  iter_tot
  err_tot

  iter_tot = iter_tot + 1;

end


%-----------------------------------------------------------------
% 6. Export results
```

```matlab
%-------------------------------------------------------------------------

save("PS3Q4.mat");


%-------------------------------------------------------------------------
% 7. Defining functions
%-------------------------------------------------------------------------

function [Z,PI] = rouwenhorst(N,mu,rho,sigma)
    % Code to approximate AR(1) process using the Rouwenhorst method as in
    % Kopecky & Suen, Review of Economic Dynamics (2010), Vol 13, p 701-714
    %
    %Purpose:    Finds a Markov chain whose sample paths approximate those of
    %            the AR(1) process
    %                z(t+1) = (1-rho)*mu + rho * z(t) + eps(t+1)
    %            where eps are normal with stddev sigma
    %
    %Format:     [Z, PI] = rouwenhorst(N,mu,rho,sigma)
    %
    %Input:      N       scalar, number of nodes for Z
    %            mu      scalar, unconditional mean of process
    %            rho     scalar
    %            sigma   scalar, std. dev. of epsilons
    %
    %Output:     Z       N*1 vector, nodes for Z
    %            PI      N*N matrix, transition probabilities
    %
    % Code and comment by Martin Floden, Stockholm University, August 2010
    %
    % Comment on this method:
    % As opposed to the methods suggested by Tauchen and Tauchen and Hussey
    % (see M. Floden, Economic Letters, 2008, 99, 516-520), the Rouwenhorst
    % method perfectly matches both the conditional and unconditional variances
    % and autocorrelations of the AR(1) process. The method however tends to
    % generate errors eps that are further away from the normal distribution
    % than the Tauchen methods (the kurtosis of the simulated eps is too high
    % with the Rouwenhorst method).

    sigmaz = sigma / sqrt(1-rho^2);

    p  = (1+rho)/2;
    PI = [p 1-p; 1-p p];

    for n = 3:N
        PI = p*[PI zeros(n-1,1); zeros(1,n)] + ...
             (1-p)*[zeros(n-1,1) PI; zeros(1,n)] + ...
             (1-p)*[zeros(1,n); PI zeros(n-1,1)] + ...
             p*[zeros(1,n); zeros(n-1,1) PI];
        PI(2:end-1,:) = PI(2:end-1,:)/2;
    end

    fi = sqrt(N-1)*sigmaz;
    Z  = linspace(-fi,fi,N)';
    Z  = Z + mu;

end


function [vals, inds]=basefun(grid_x,npx,x)
  %Linear interpolation
  jl=1;
  ju=npx;
  while((ju-jl>1))
    jm=round((ju+jl)/2);
    if(x>=grid_x(jm))
      jl=jm;
    else
      ju=jm;
    end
  end

  i=jl+1;
  vals(2)=( x-grid_x(i-1) )/(grid_x(i)-grid_x(i-1));
```

```
  vals(2)=max(0.0d0,min(1.0d0,vals(2)));
  vals(1)=1.0d0-vals(2);
  inds(2)=i;
  inds(1)=i-1;

end
```

Appendix C: Q5 model file

```
%--------------------------------------------------------------------
%
% Title: International Macro-Finance Problem Set 3, Q5 model file
% Author: ---
% Date: 10/12/2023
% Description: Open economy heterogeneous agent model
%
%--------------------------------------------------------------------


%--------------------------------------------------------------------
%% 0. Housekeeping (close all graphic windows)
%--------------------------------------------------------------------

close all;
clear all;

cd '/home/---/Desktop/International Macro/PS3';

%--------------------------------------------------------------------
%% 1. Retrieve parameters from Q4
%--------------------------------------------------------------------

load("PS3Q4.mat");


%--------------------------------------------------------------------
%% 2. Set government debt as a parameter & foreign lending
%--------------------------------------------------------------------

par.bg = bg;
clear bg;

guess.tau = 0;


%--------------------------------------------------------------------
%% 3. Iterations
%--------------------------------------------------------------------

iter_tot = 1;
err_tot = 1;

while err_tot > errtol_tot && iter_tot <= maxIter_tot

  w_p0 = par.alphah^par.alphah * (1 - par.alphah)^(1 - par.alphah) * ...
           guess.ph_pf^(1 - par.alphah); % update real wage from the tot

  %% Compute labour supply
  policy.li = ((1 - guess.tau) .* w_p0 .* grid.s1).^(1 / par.eta);
  l0 = pi * policy.li.';

  iter_r = 1;
  err_r = 1;
  %%

  while err_r > errtol_r && iter_r <= maxIter_r

    %% Solve for consumption
    y = l0;

    % Compute constrained consumption given R
    c_constrained = (1 - guess.tau) .* w_p0 .* grid.s2 .* policy.li + ...
                    w_p0 .* grid.b2 .* guess.r0 - ...
                    w_p0 .* b_con;
    c_constrained = max(c_constrained, 1e-5); % rule out consumption below 0

    iter_c = 1;
    err_c = 1;

    guess.ci = ones(nb,ns);
```

```
%%

while err_c > errtol_c && iter_c <= maxIter_c


  Emup1 = (guess.ci * par.scale - policy.li.^(1 + par.eta) / (1 + par.eta)).^(-1); % expected marginal utility at t+2
  Emup = par.beta * guess.r0 * Emup1 * Pr'; % expected marginal utility at t+1
  Ec = Emup.^(-1) + policy.li.^(1 + par.eta) / (1 + par.eta); % consumption tomorrow
  bi_state = (Ec ./ w_p0 + grid.b2 - (1 - guess.tau) .* grid.s2 .* ...
                  policy.li) ./ (guess.r0); % state debt tomorrow (i.e. choice debt today)

  c_new = ones(nb, ns);

  for j=1:ns

      c_new(:,j) =  (grid.b > bi_state(Ind_b_min, j)) .* ... % indicator function on whether borrowing constraint is binding
                 interp1(bi_state(:, j), Ec(:, j), grid.b, 'pchip') + ... % interpolate c_s = f(bi_state) at each grid point
                 (grid.b <= bi_state(Ind_b_min, j)) .* c_constrained(:,j); % if constraint is binding, then c_constrained
      c_new(:,j) =  max(c_new(:,j), 1e-5); % rules out negative values
  end

  guess.ci = c_new * lambda + (1 - lambda) * guess.ci;

  err_c = max(max(abs(c_new - guess.ci)));

  iter_c
  err_c

  iter_c = iter_c + 1;

end

% Write the policy function for consumption
for j=1:ns
  policy.ci(:,j) = interp1(grid.b, guess.ci(:,j), grid.b_fine);
end

%% Solve for interest

% Write the policy function for assets
bi_choice  = (grid.b2 * guess.r0 + (1 - guess.tau) * grid.s2 .* policy.li - ...
          guess.ci ./ w_p0);

for j=1:ns
    policy.bi(:,j) = interp1(grid.b, bi_choice(:,j), grid.b_fine);
end

% Compute the endogenous distribution
trows = zeros(nb_fine * ns * ns * 2, 1);
tcols = trows;
tvals = tcols;
index = 0;
for j=1:ns
    for bi = 1:nb_fine
        [vals,inds] = basefun(grid.b_fine, nb_fine, policy.bi(bi,j));
        for jp=1:ns
            index = index + 1;
            trows(index) = bi + (j - 1) * nb_fine;
            tcols(index) = inds(1) + (jp - 1)* nb_fine;
            tvals(index) = Pr(j, jp) * vals(1);
            index = index+1;
            trows(index) = bi + (j - 1) * nb_fine;
            tcols(index) = inds(2) + (jp - 1) * nb_fine;
            tvals(index) = Pr(j, jp) * vals(2);
        end
    end
end
transMat = sparse(trows, tcols, tvals, nb_fine * ns, nb_fine * ns);
[EigVec, EigVal] = eigs(transMat.', 1);
EigVec = EigVec / sum(EigVec);
```

```
    EigVec(EigVec < 0) = 0;
    EigVec = EigVec / sum(EigVec);
    G0 = reshape(EigVec / sum(EigVec), [nb_fine ns]); % distr. of HHs across assets & states

    % update guess for r
    b = sum(sum(policy.bi .* G0)); % aggregate HH lending
    c = sum(sum(policy.ci .* G0));
    b_star = -(b + par.bg); % solve for aggregate borrowing from bond market clearing
    tau_new = (par.g + par.bg * (1 - guess.r0)) / y; % solve for tax from govt BC
    guess.tau = lambda * tau_new + (1 - lambda) * guess.tau;

    r_new = (par.bg + par.g - guess.tau * y) / par.bg; % update r from govt BC
    % r_new = 1 + c / (b * w_p0) - (1 - guess.tau) * l0 / b; % solve for interest from HH BC

    err_r = abs(r_new - guess.r0);

    guess.r0 = lambda * r_new + (1 - lambda) * guess.r0; % update

    iter_r
    err_r

    iter_r = iter_r + 1;
    %%

  end

  %% Solve for terms of trade
  ch_star = par.gamma * y;
  ch = y - par.g - ch_star; % consumption of home goods from market clearing
  ph_pf_new = cf / ch; % domestic price of foreign goods from the FOC

  err_tot = max(max(abs(ph_pf_new - guess.ph_pf)));

  guess.ph_pf = lambda * ph_pf_new + (1 - lambda) * guess.ph_pf; % update

  iter_tot
  err_tot

  iter_tot = iter_tot + 1;

end


%--------------------------------------------------------------------
% 6. Export results
%--------------------------------------------------------------------

save("PS3Q5.mat");


%--------------------------------------------------------------------
% 5. Defining functions
%--------------------------------------------------------------------

function [vals, inds]=basefun(grid_x,npx,x)
  %Linear interpolation
  jl=1;
  ju=npx;
  while((ju-jl>1))
    jm=round((ju+jl)/2);
    if(x>=grid_x(jm))
      jl=jm;
    else
      ju=jm;
    end
  end

  i=jl+1;
  vals(2)=( x-grid_x(i-1) )/(grid_x(i)-grid_x(i-1));
  vals(2)=max(0.0d0,min(1.0d0,vals(2)));
  vals(1)=1.0d0-vals(2);
  inds(2)=i;
  inds(1)=i-1;

```

```
end
```

Appendix D: Q6 model file

```
%-------------------------------------------------------------------------------
%
% Title: International Macro-Finance Problem Set 3, model file
% Author: ---
% Date: 10/12/2023
% Description: Open economy heterogeneous agent model
%
%-------------------------------------------------------------------------------


%-------------------------------------------------------------------------------
%% 0. Housekeeping (close all graphic windows)
%-------------------------------------------------------------------------------

close all;
clear all;

cd '/home/---/Desktop/International Macro/PS3';

%-------------------------------------------------------------------------------
%% 1. Retrieve parameters from Q4
%-------------------------------------------------------------------------------

load("PS3Q4.mat");


%-------------------------------------------------------------------------------
%% 2. Set government debt as parameter & set foreign lending
%-------------------------------------------------------------------------------

par.bg = bg;
clear bg;


%-------------------------------------------------------------------------------
%% 3. Iterations
%-------------------------------------------------------------------------------

iter_tot = 1;
err_tot = 1;


g_tol = 1e-4;          % Tolerance for convergence of par_g
max_iter_g = 1000;     % Maximum iterations for finding par_g

while err_tot > errtol_tot && iter_tot <= maxIter_tot

  w_p0 = par.alphah^par.alphah * (1 - par.alphah)^(1 - par.alphah) * ...
            guess.ph_pf^(1 - par.alphah); % update real wage from the tot

  %% Compute labour supply
  policy.li = ((1 - par.tau) .* w_p0 .* grid.s1).^(1 / par.eta);
  l0 = pi * policy.li.';

  iter_r = 1;
  err_r = 1;
  %%

  while err_r > errtol_r && iter_r <= maxIter_r

    %% Solve for consumption
    y = l0;

    % Compute constrained consumption given R
    c_constrained = (1 - par.tau) .* w_p0 .* grid.s2 .* policy.li + ...
                    w_p0 .* grid.b2 .* guess.r0 - ...
                    w_p0 .* b_con;
    c_constrained = max(c_constrained, 1e-5); % rule out consumption below 0

    iter_c = 1;
    err_c = 1;
```

```
    guess.ci = ones(nb,ns);
    %%

    while err_c > errtol_c && iter_c <= maxIter_c


      Emup1 = (guess.ci * par.scale - policy.li.^(1 + par.eta) / (1 + par.eta)).^(-1); % expected marginal utility at t+2
      Emup = par.beta * guess.r0 * Emup1 * Pr'; % expected marginal utility at t+1
      Ec = Emup.^(-1) + policy.li.^(1 + par.eta) / (1 + par.eta); % consumption tomorrow
      bi_state = (Ec ./ w_p0 + grid.b2 - (1 - par.tau) .* grid.s2 .* ...
                      policy.li) ./ (guess.r0); % state debt tomorrow (i.e. choice debt today)

      c_new = ones(nb, ns);

      for j=1:ns

          c_new(:,j) =  (grid.b > bi_state(Ind_b_min, j)) .* ... % indicator function on whether borrowing constraint is binding
                      interp1(bi_state(:, j), Ec(:, j), grid.b, 'pchip') + ... % interpolate c_s = f(bi_state) at each grid point
                      (grid.b <= bi_state(Ind_b_min, j)) .* c_constrained(:,j); % if constraint is binding, then c_constrained
          c_new(:,j) =  max(c_new(:,j), 1e-5); % rules out negative values
      end

      guess.ci = c_new * lambda + (1 - lambda) * guess.ci;

      err_c = max(max(abs(c_new - guess.ci)));

      iter_c
      err_c

      iter_c = iter_c + 1;

    end

    % Write the policy function for consumption
    for j=1:ns
      policy.ci(:,j) = interp1(grid.b, guess.ci(:,j), grid.b_fine);
    end

    %% Solve for interest

    % Write the policy function for assets
    bi_choice  = (grid.b2 * guess.r0 + (1 - par.tau) * grid.s2 .* policy.li - ...
                guess.ci ./ w_p0);

    for j=1:ns
        policy.bi(:,j) = interp1(grid.b, bi_choice(:,j), grid.b_fine);
    end

    % Compute the endogenous distribution
    trows = zeros(nb_fine * ns * ns * 2, 1);
    tcols = trows;
    tvals = tcols;
    index = 0;
    for j=1:ns
        for bi = 1:nb_fine
            [vals,inds] = basefun(grid.b_fine, nb_fine, policy.bi(bi,j));
            for jp=1:ns
                index = index + 1;
                trows(index) = bi + (j - 1) * nb_fine;
                tcols(index) = inds(1) + (jp - 1)* nb_fine;
                tvals(index) = Pr(j, jp) * vals(1);
                index = index+1;
                trows(index) = bi + (j - 1) * nb_fine;
                tcols(index) = inds(2) + (jp - 1) * nb_fine;
                tvals(index) = Pr(j, jp) * vals(2);
            end
        end
    end
    transMat = sparse(trows, tcols, tvals, nb_fine * ns, nb_fine * ns);
```

```
    [EigVec, EigVal] = eigs(transMat.', 1);
    EigVec = EigVec / sum(EigVec);
    EigVec(EigVec < 0) = 0;
    EigVec = EigVec / sum(EigVec);
    G0 = reshape(EigVec / sum(EigVec), [nb_fine ns]); % distr. of HHs across assets & states

    % update guess for r
    c = sum(sum(policy.ci .* G0)); % aggregate HH consumption
    b = sum(sum(policy.bi .* G0));
    b_star = -(b + par.bg); % solve for aggregate borrowing from bond market clearing
    g = par.tau * y + par.bg * (guess.r0 - 1); % solve for govt spending from govt BC

    r_new = (par.bg + g - par.tau * y) / par.bg; % update r from govt BC
    % r_new = 1 + c / (b * w_p0) - (1 - par.tau) * l0 / b; % solve for interest from HH BC

    err_r = abs(r_new - guess.r0);

    guess.r0 = lambda * r_new + (1 - lambda) * guess.r0; % update

    iter_r
    err_r

    iter_r = iter_r + 1;
    %%

  end

  %% Solve for terms of trade
  ch_star = par.gamma * y;
  ch = y - g - ch_star; % consumption of home goods from market clearing
  ph_pf_new = cf / ch; % domestic price of foreign goods from the FOC

  err_tot = max(max(abs(ph_pf_new - guess.ph_pf)));

  guess.ph_pf = lambda * ph_pf_new + (1 - lambda) * guess.ph_pf; % update

  iter_tot
  err_tot

  iter_tot = iter_tot + 1;

end


%-------------------------------------------------------------------
% 4. Export results
%-------------------------------------------------------------------

save("PS3Q6.mat");


%-------------------------------------------------------------------
% 5. Defining functions
%-------------------------------------------------------------------

function [vals, inds]=basefun(grid_x,npx,x)
  %Linear interpolation
  jl=1;
  ju=npx;
  while((ju-jl>1))
    jm=round((ju+jl)/2);
    if(x>=grid_x(jm))
      jl=jm;
    else
      ju=jm;
    end
  end

  i=jl+1;
  vals(2)=( x-grid_x(i-1) )/(grid_x(i)-grid_x(i-1));
  vals(2)=max(0.0d0,min(1.0d0,vals(2)));
  vals(1)=1.0d0-vals(2);
  inds(2)=i;
  inds(1)=i-1;
```

```
end
```

Appendix E: AR(1) Fitting

```
%% Calibrate labour productivity from FRED data to match US GDP
% (https://fred.stlouisfed.org/series/OPHNFB)

% Load series
lab_prod = xlsread("OPHNFB.xls");
lab_prod = lab_prod(:,2);
restricted_lab_prod = lab_prod(200:307);

% Difference the data
Dlab_prod = diff(lab_prod);

% Fit an AR(1) process to the differenced data
%model1 = arima(1,1,0);
model2 = arima(1,0,0); % AR(1)

%estmodel1 = estimate(model1, lab_prod);
%estmodel2 = estimate(model2, lab_prod);

estmodel3 = estimate(model2, restricted_lab_prod);
```