Problem Set 1

1068576

October 2023

Q1.

The social planner's problem is:

$$\max_{\{\pi_t, x_t\}_{t=0}^{\infty}} -\frac{\omega}{2}\sum_{t=0}^{\infty}\beta^t\left(x_t^2+\lambda_h\cdot\pi_{h,t}^2\right) \tag{1}$$

subject to:

$$\pi_{h,t}=\kappa x_t+\beta E_t\left(\pi_{h,t+1}\right)+u_t \tag{2}$$

We can determine $\{x_t, \pi_{h,t}\}_{t=0}^{\infty}$ by maximising the loss function subject to the AS function. $\{i_t\}_{t=0}^{\infty}$ can be derived by substituting the target output gap and inflation into the AD equation to obtain the optimal policy plan.

The Langrangian is:

$$\mathcal{L}=-\frac{\omega}{2}\sum_{t=0}^{\infty}\beta^t\left[\left(x_t^2+\lambda_h\cdot\pi_{h,t}^2\right)+2\mu_t\left(\kappa x_t+\beta\pi_{t+1}+u_t-\pi_t\right)\right] \tag{3}$$

The FOCs with respect to x_t and π_t are, respectively:

$$x_t+k\cdot\mu_t=0 \tag{4}$$

$$\pi_{h,t}\cdot\lambda_h=\mu_t-\mu_{t-1} \tag{5}$$

We obtain the optimal policy by combining them:

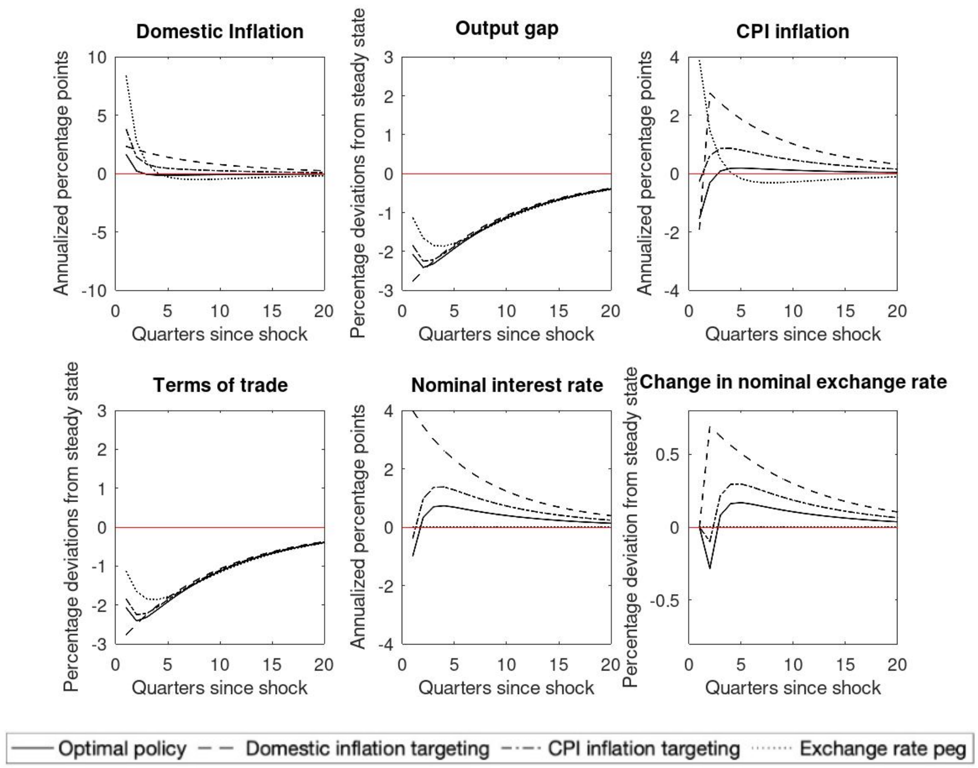
$$\kappa\lambda_\pi\pi_{h,t}=-(x_t-x_{t-1}) \tag{6}$$

By substituting $\kappa\equiv\kappa(1+\phi)$ into equation (6) and solving for $\pi_{h,t}$, the optimal policy expression simplifies to:

$$\pi_{h,t}=\frac{x_{t-1}-x_t}{\epsilon} \tag{7}$$

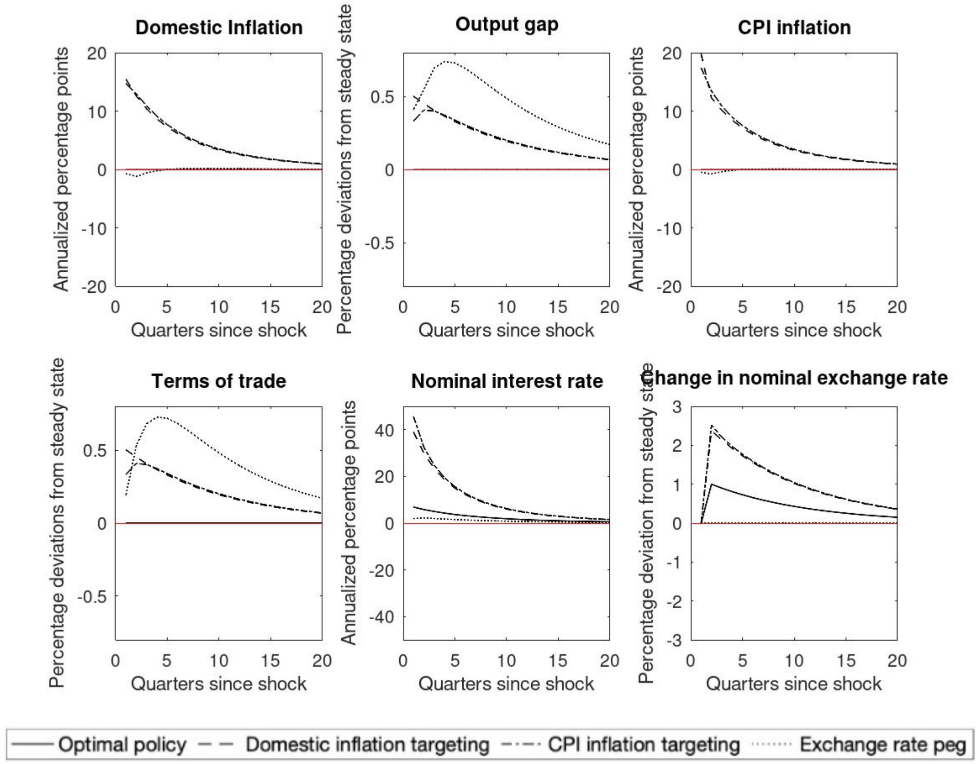
The optimal targeting rule sets the target inflation in the opposite direction to the change in the output gap.

Impulse response functions for a unit standard deviation cost-push shock (u_t)

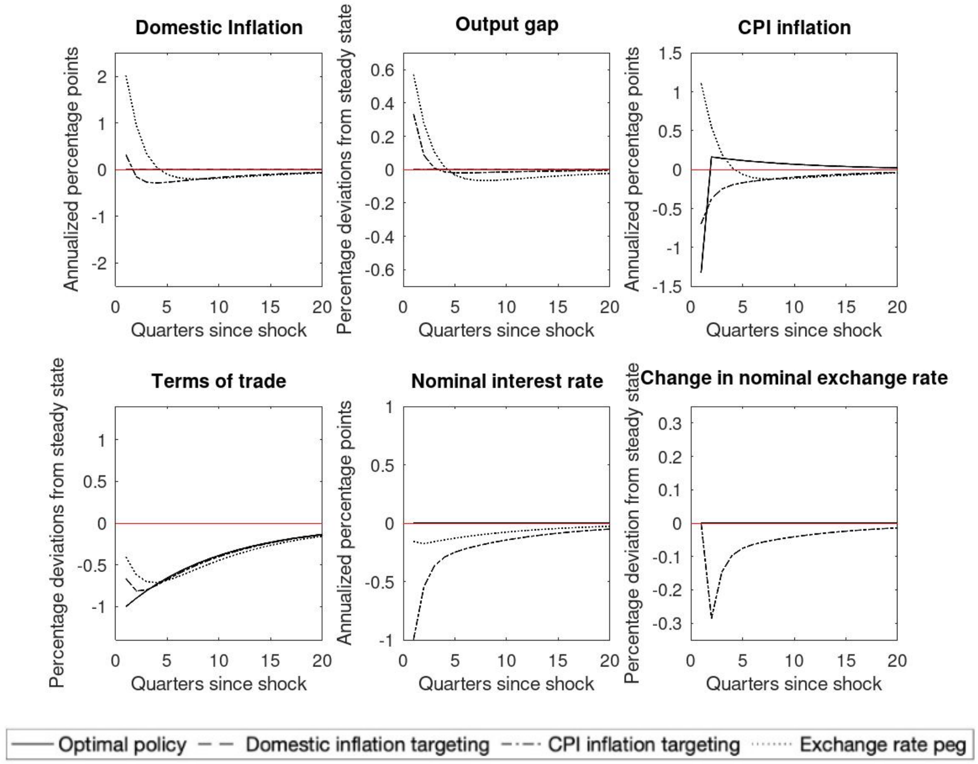


Q2.

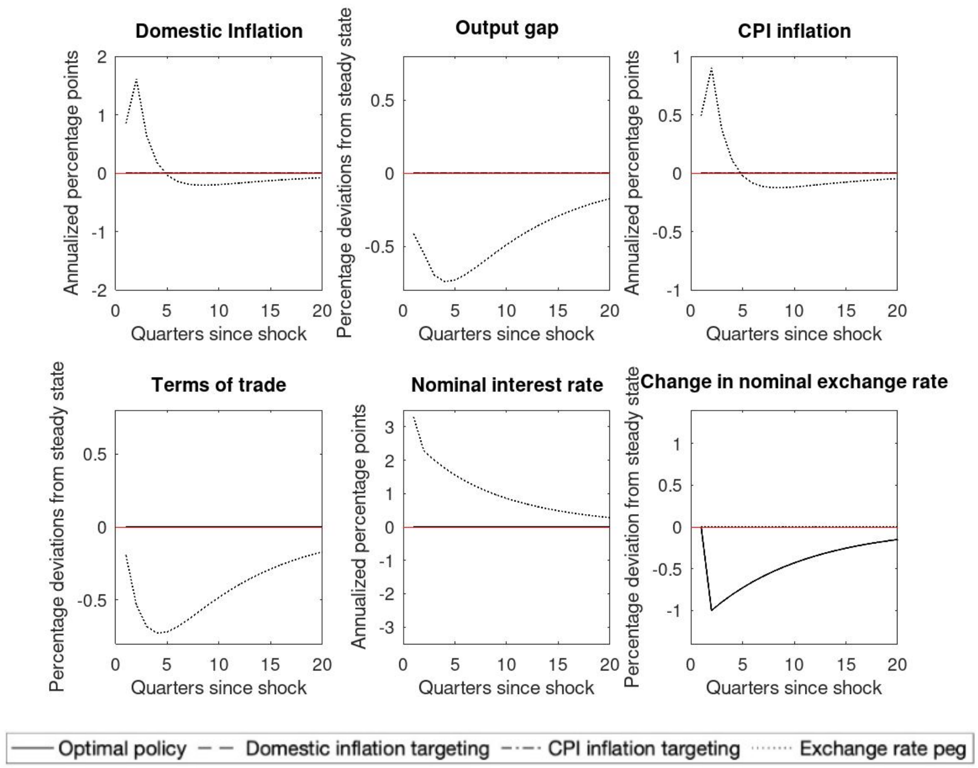
Impulse response functions for a unit standard deviation technology shock (r_t^n)



Impulse response functions for a unit standard deviation foreign output (z_t)



Impulse response functions for a unit standard deviation foreign interest shock (i_t^*)



We begin with the following system of equations:

$$\pi_{h,t} = \kappa x_t + \beta E_t\left(\pi_{h,t+1}\right) + u_t \tag{8}$$
$$x_t = x_{t+1} - (i_t - \pi_{h,t+1} - r_t^n) \tag{9}$$
$$\pi_t = \pi_{h,t} + \alpha(\tau_t - \tau_{t-1}) \tag{10}$$
$$x_t = z_t + \tau_t \tag{11}$$
$$i_t = i_t^* + \Delta e_t \tag{12}$$
$$\tag{13}$$

We can easily compute the steady state from the system above:

$$
\begin{aligned}
x_{ss} &= 0\\
\tau_{ss} &= 0\\
\pi_{ss} &= 0\\
\pi_{h,ss} &= 0\\
i_{ss} &= 0\\
\Delta e_{ss} &= 0
\end{aligned}
$$

We now solve for $\{x_t, \tau_t, \pi_t, \pi_{h,t}, i_t, \Delta e_t\}_{t=0}^{\infty}$ in the above system of equations, using the steady-states as the initial values. To obtain the impulse response functions under optimal policy, add the optimal policy rule (7) to the system above, where $\{u_t, r_t^n, z_t, i_t^*\}_{t=0}^{\infty}$ are exogenous AR(1) processes:

$$
\begin{aligned}
u_t &= \rho u_{t-1} + \epsilon_{u,t}\\
r_t &= \rho r_{t-1} + \epsilon_{r,t}\\
z_t &= \rho z_{t-1} + \epsilon_{z,t}\\
i_t^* &= \rho i_{t-1}^* + \epsilon_{i^*,t}
\end{aligned}
$$

with persistence $\rho = .9$ and $\epsilon_{i,t} \sim \mathcal{N}(0,1)$. The impulse response functions for $\{\pi_{h,t}, x_t, \pi_t, \tau_t, i_t, \Delta e_t\}_{t=0}^{20}$ are plotted in solid lines in figures **??** **??**.

Next, we solve for the endogenous variables under the domestic inflation Taylor rule:

$$i_t = \phi_\pi \pi_{H,t} \tag{14}$$

where $\phi_\pi = 1.5$. The impulse response functions are plotted in dashed lines. Finally, we solve for the endogenous variables under the CPI inflation Taylor rule:

$$i_t = \phi_\pi \pi_t \tag{15}$$

The impulse response functions are plotted in dashed-dotted lines.

In response to a cost-push shock, the central bank reacts by increasing nominal interest under domestic targeting. Under the optimal policy rule and CPI targeting by contrast, the central bank initially decreases interest to react to a decrease in CPI inflation caused by the decline in the terms of trade. By construction (see eq. 13), the change in the nominal exchange rate follows the same pattern as the movement in the nominal interest. Moreover, the increase in domestic inflation results in a decline in the term of trade, by definition. Through equation 12, this implies a decline in the output gap.

Domestic inflation targeting results in lower domestic inflation compared to CPI inflation targeting, which is unsurprising given that CPI inflation is a weighted average of domestic and inflation of imported goods. However, note that optimal policy results in lower domestic inflation than under an explicit domestic inflation targeting. This might be because the central bank chooses the smallest possible target inflation given an output gap under optimal policy, such that the welfare loss function is minimised. Under domestic inflation targeting on the other hand, the Taylor rule coefficient, $\phi_p i$, of 1.5 is chosen arbitrarily and may not result in the lowest possible inflation.

Q3.

Under all policies, the shock in the natural rate of interest results in an increase in the output gap through the AD equation (eq.(10)). The AS equation (eq. (9)) implies that domestic inflation increases drastically, in part through the expectation channel. The central bank reacts by increasing interest rate, although the increase in nominal interest is less drastic under the optimal policy rule. Terms of trade follows the path of the output gap through equation (12), while the nominal exchange rate follows the nominal interest rate through equation (13).

Under CPI targeting, the shock in the difference between foreign and domestic natural output results in a decrease in the terms of trade through equation (12) and consequently, an increase in the domestic inflation through equation (9). Since the central bank does not react to domestic inflation, the output gap also increases. Moreover, the effects of the terms of trade outweigh the effects of domestic inflation, such that CPI inflation decreases from equation (12), resulting in the central bank decreasing its policy rate. Nominal exchange rate as a result, decreases through the UIP. We do not observe similar phenomena under the other policy rules because the central bank immediately stabilises output and terms of trade absorbs most of the shock.

Under all policies, a shock in the foreign interest rate is fully absorbed by the nominal exchange rate through the UIP. All other variables remain at steady state.

Q4.

The Blanchard-Kahn condition does not hold under an exchange rate peg policy rule because the uncovered interest rate parity condition becomes:

$$i_t = i_t^* \tag{16}$$

where i_t is now an exogenous process. Therefore, π_t is unrestricted and the eigenvalues of this system will have an absolute value that is greater than one. This implies that inflation becomes explosive given a sunspot shock, such that the system does not converge to a unique solution.

There are two solutions to this problem. The first solution is to assume that the rest of the world fully stabilises inflation, such that $p_t^* = 0$. From the definition of terms of trade, we know that:

$$\tau_t = \frac{P_{h,t}^*}{P_t} \tag{17}$$

where $P_{h,t}$ is domestic price level and P_t^* is the price of imported goods. From this equation, we derive that:

$$\tau_t = \tau_{t-1} + \pi_t^* - \pi_{h,t} \tag{18}$$

From our assumption of constant price of imported goods, we can add the following equation to the system of equations:

$$\tau_t = \tau_{t-1} - \pi_{h,t} \tag{19}$$

which imposes an additional on $\pi_{h,t}$ through the terms of trade. The impulse response functions are shown in dotted lines in figures **??** to **??**.

The second solution is to add a risk premium to the uncovered interest rate parity condition to reflect risks due to inflation. The UIP states that through arbitrage, the returns on a domestic bond equals the returns earned through a carry trade selling the domestic bond and buying a foreign bond. In a world with inflation, the real returns on a domestic bond falls, while the returns through the carry trade is protected from inflation, assuming the future exchange rate adjusts to future inflation (i.e. the domestic currency depreciates). This results in higher returns on the carry trade and therefore, results in a risk premium on domestic returns such that:

$$i_t = i_t^* + f(E_t(\pi_{t+1})) \tag{20}$$

Empirically, risk premiums are estimated at around 5-12 percentage points, so re-writing the interest rate equation as:

$$i_t = i_t^* + 0.05 \cdot \mathbb{E}[\pi_{h,t+1}] \tag{21}$$

satisfies the Blanchard Khan conditions.

For a given cost-push shock, all variables under an exchange rate peg co-move with the variables under a CPI Taylor rule because CPI targeting is an indirect form of exchange rate peg.

In reaction to shock in the difference between foreign output and the domestic natural output, terms of trade decreases while the output gap increases, similar to CPI targeting. However, unlike CPI inflation targeting, the effects of the domestic inflation outweigh the effects of terms of trade, such that CPI inflation increases. This difference is observed because from the UIP equation (eq. 13), the central bank cannot raise nominal interest in reaction to the increase in inflation under a strict peg, resulting in a large increase in domestic inflation. Consequently, the output gap also increases by more than under CPI inflation targeting through the expectation channel of the AD equation.

For a given shock in foreign interest, the domestic nominal interest increases given the perfect peg, resulting in the negative output gap. Terms of trade decreases as a result (see eq. 12). It is difficult to explain the increase in both domestic and CPI inflation, as we expected a decrease given a negative output gap and decrease in terms of trade.

Q5.

To compute the welfare losses, we take the output gap and inflation at each period and compute them into the welfare loss function. Since the impulse response functions eventually converge to steady-state, the loss function should also converge for a large enough time horizon. In this case, $T = 100$ was chosen for this computation. Hence, we compute:

$$\mathcal{L} = -\frac{\omega}{2} \sum_{t=0}^{20} \beta^t \left(x_t^2 + \lambda_h \pi_{h,t}^2 \right) \tag{22}$$

The results are shown in the table below.

Table 1: Welfare Impact of Shocks on Different Policies

Shocks	Optimal	Domestic Inflation	CPI	Peg ($\pi^* = 0$)	Peg with Risk
u shock	-47.288	-68.166	-54.251	-54.67	-69.468
r^n shock	0	-264.11	-273.8	-71.904	-9.486
z shock	0	0	-1.0111	0	-5.4733
i^* shock	0	0	0	-71.904	-9.486

The optimal policy produces the least welfare loss by construction. Given a cost-push shock, domestic inflation targeting performs poorly compared to CPI targeting or an interest rate peg because nominal interest may be over-reacting to inflation and decreasing the output gap more than necessary (note that the coefficient in the Taylor rule, ϕ_π, was chosen arbitrarily). Given a shock in r^n, the interest rate peg performs better than domestic inflation or CPI targeting because it manages to keep domestic inflation at a significantly lower level, which is again puzzling. For a shock in i_t^*, the currency peg performs poorly compared to all other policy because the exchange rate cannot absorb any of the foreign shocks.

Appendix A: Code

Main file

```
pkg load dataframe

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

clear all;
close all;


%---------------------------------------------------------------------------
% 1. Run Models
%---------------------------------------------------------------------------

dynare optimal;
dynare ditr;
dynare citr;
dynare peg1;
dynare peg2;

%---------------------------------------------------------------------------
% 2. Load Models
%---------------------------------------------------------------------------

optimal_results = load("optimal_results.mat");
ditr_results = load("ditr_results.mat");
citr_results = load ("citr_results.mat");
peg1_results = load ("peg1_results.mat");
peg2_results = load("peg2_results.mat");


%---------------------------------------------------------------------------
% 3. Define Functions
%---------------------------------------------------------------------------

% Function to setup axes for subplots
function setup_subplot_axes()
    xlims = xlim; % Get the current x-axis limits
    line(xlims, [0 0], 'Color', 'r', 'LineWidth',0.5); % Draw grey line at y=0 u
    max_ylim = max(abs(ylim)); % Get max absolute value from y-axis
    ylim([-max_ylim max_ylim]); % Set symmetric y limits around 0
```

```
    xlabel('Quarters since shock');
endfunction

function loss0 = calculate_welfare_loss(x, pih)
  alpha = .4;
  beta = .99;
  epsilon = 6;
  theta = .75;
  phi = 3;
  lambda = (1 - theta) * (1 - beta * theta) / theta;
  omega = (1 - alpha) * (1 + phi);
  lambda_pi = epsilon / (lambda * (1 + phi));
  loss = zeros(1, length(x));

  for t = 1:length(x)
    loss(t) = - omega / 2 * beta ^ t * (x(t)^2 + lambda_pi * pih(t) ^ 2);

  endfor

  loss0 = sum(loss);

endfunction
%--------------------------------------------------------------------
% 3. Plot the Results
%--------------------------------------------------------------------


% annualise inflation and convert to percentage points
optimal_pih_err_u = (exp(optimal_results.oo_.irfs.pih_err_u(1:20)) - 1) * 4;
ditr_pih_err_u = (exp(ditr_results.oo_.irfs.pih_err_u(1:20)) - 1) * 4;
citr_pih_err_u = (exp(citr_results.oo_.irfs.pih_err_u(1:20)) - 1) * 4;
peg2_pih_err_u = (exp(peg2_results.oo_.irfs.pih_err_u(1:20)) - 1) * 4;

optimal_pi_err_u = (exp(optimal_results.oo_.irfs.pi_err_u(1:20)) - 1) * 4;
ditr_pi_err_u = (exp(ditr_results.oo_.irfs.pi_err_u(1:20)) - 1) * 4;
citr_pi_err_u = (exp(citr_results.oo_.irfs.pi_err_u(1:20)) - 1) * 4;
peg2_pi_err_u = (exp(peg2_results.oo_.irfs.pi_err_u(1:20)) - 1) * 4;

optimal_i_err_u = (exp(optimal_results.oo_.irfs.i_err_u(1:20)) - 1) * 4;
ditr_i_err_u = (exp(ditr_results.oo_.irfs.i_err_u(1:20)) - 1) * 4;
citr_i_err_u = (exp(citr_results.oo_.irfs.i_err_u(1:20)) - 1) * 4;
peg2_i_err_u = (exp(peg2_results.oo_.irfs.i_err_u(1:20)) - 1) * 4;

% Plotting the cost-push shock Graph
figure('Position', [100, 100, 1500, 800]);
    % pih IRFs
```

```
subplot(2,3,1)
    optimal = plot(optimal_pih_err_u, 'Linewidth', 1, 'Color', 'k');
    hold on
    ditr = plot(ditr_pih_err_u, '--', 'LineWidth',1, 'Color', 'k');
    citr = plot(citr_pih_err_u, '-.', 'LineWidth',1, 'Color', 'k');
    peg2 = plot(peg2_pih_err_u, ':', 'LineWidth',1, 'Color', 'k');
    setup_subplot_axes();
    title('Domestic Inflation');
    ylabel('Annualized percentage points');
% x IRFs
subplot(2,3,2)
    plot(optimal_results.oo_.irfs.x_err_u(1:20), 'Linewidth', 1, 'Color', 'k
    hold on
    plot(ditr_results.oo_.irfs.x_err_u(1:20), '--', 'LineWidth',1, 'Color',
    plot(citr_results.oo_.irfs.x_err_u(1:20), '-.', 'LineWidth',1, 'Color',
    plot(peg2_results.oo_.irfs.x_err_u(1:20), ':', 'LineWidth',1, 'Color', '
    setup_subplot_axes();
    title('Output gap');
    ylabel('Percentage deviations from steady state');
% pi
   subplot(2,3,3)
    plot(optimal_pi_err_u, 'Linewidth', 1, 'Color', 'k');
    hold on
    plot(ditr_pi_err_u, '--', 'LineWidth',1, 'Color', 'k');
    plot(citr_pi_err_u, '-.', 'LineWidth',1, 'Color', 'k');
    plot(peg2_pi_err_u, ':', 'LineWidth',1, 'Color', 'k');
    setup_subplot_axes();
    title('CPI inflation');
    ylabel('Annualized percentage points');
% tau
subplot(2,3,4)
    plot(optimal_results.oo_.irfs.tau_err_u(1:20), 'Linewidth', 1, 'Color',
    hold on
    plot(ditr_results.oo_.irfs.tau_err_u(1:20), '--', 'LineWidth',1, 'Color'
    plot(citr_results.oo_.irfs.tau_err_u(1:20), '-.', 'LineWidth',1, 'Color'
    plot(peg2_results.oo_.irfs.tau_err_u(1:20), ':', 'LineWidth',1, 'Color',
    setup_subplot_axes();
    title('Terms of trade');
    ylabel('Percentage deviations from steady state');
% i
subplot(2,3,5)
    plot(optimal_i_err_u, 'Linewidth', 1, 'Color', 'k');
    hold on
    plot(ditr_i_err_u, '--', 'LineWidth',1, 'Color', 'k');
    plot(citr_i_err_u, '-.', 'LineWidth',1, 'Color', 'k');
    plot(peg2_i_err_u, ':', 'LineWidth',1, 'Color', 'k');
```

```
        setup_subplot_axes();
        title('Nominal interest rate');
        ylabel('Annualized percentage points');
    % delta_e
    subplot(2,3,6)
        plot(optimal_results.oo_.irfs.delta_e_err_u(1:20), 'Linewidth', 1, 'Colo
        hold on
        plot(ditr_results.oo_.irfs.delta_e_err_u(1:20), '--', 'LineWidth',1, 'Co
        plot(citr_results.oo_.irfs.delta_e_err_u(1:20), '-.', 'LineWidth',1, 'Co
        plot(peg2_results.oo_.irfs.delta_e_err_u(1:20), ':', 'LineWidth',1, 'Col
        setup_subplot_axes();
        title('Change in nominal exchange rate');
        ylabel('Percentage deviation from steady state');
        %{
    % Add a global legend using the plots from the first subplot
        lgd = legend([optimal, ditr, citr, peg2], {'Optimal policy', 'Domestic i
        set(lgd, 'location', 'south', 'orientation', 'horizontal', 'fontsize', 1
    % Manually adjust the position to be at the bottom center of the figure
        legendPosition = [0.25, 0.01, 0.5, 0.05];
        set(lgd, 'Position', legendPosition);
        %}
    % Saving the graph
    saveas(gcf, 'Cost-push shock.jpg');



% annualise inflation and convert to percentage points
%{
optimal_pih_err_r = (exp(optimal_results.oo_.irfs.pih_err_r(1:20)) - 1) * 4;
ditr_pih_err_r = (exp(ditr_results.oo_.irfs.pih_err_r(1:20)) - 1) * 4;
citr_pih_err_r = (exp(citr_results.oo_.irfs.pih_err_r(1:20)) - 1) * 4;
peg2_pih_err_r = (exp(peg2_results.oo_.irfs.pih_err_r(1:20)) - 1) * 4;
%}

optimal_pih_err_r = (exp(optimal_results.oo_.irfs.pih_err_r(1:20)) - 1) * 4;
ditr_pih_err_r = (exp(ditr_results.oo_.irfs.pih_err_r(1:20)) - 1) * 4;
citr_pih_err_r = (exp(citr_results.oo_.irfs.pih_err_r(1:20)) - 1) * 4;
peg2_pih_err_r = (exp(peg2_results.oo_.irfs.pih_err_r(1:20)) - 1) * 4;

optimal_pi_err_r = (exp(optimal_results.oo_.irfs.pi_err_r(1:20)) - 1) * 4;
ditr_pi_err_r = (exp(ditr_results.oo_.irfs.pi_err_r(1:20)) - 1) * 4;
citr_pi_err_r = (exp(citr_results.oo_.irfs.pi_err_r(1:20)) - 1) * 4;
peg2_pi_err_r = (exp(peg2_results.oo_.irfs.pi_err_r(1:20)) - 1) * 4;

optimal_i_err_r = (exp(optimal_results.oo_.irfs.i_err_r(1:20)) - 1) * 4;
ditr_i_err_r = (exp(ditr_results.oo_.irfs.i_err_r(1:20)) - 1) * 4;
```

```
citr_i_err_r = (exp(citr_results.oo_.irfs.i_err_r(1:20)) - 1) * 4;
peg2_i_err_r = (exp(peg2_results.oo_.irfs.i_err_r(1:20)) - 1) * 4;

% Plotting the technology shock graph
figure('Position', [100, 100, 1200, 800]);
    % pih IRFs
    subplot(2,3,1)
        optimal = plot(optimal_pih_err_r, 'Linewidth', 1, 'Color', 'k');
        hold on
        ditr = plot(ditr_pih_err_r, '--', 'LineWidth',1, 'Color', 'k');
        citr = plot(citr_pih_err_r, '-.', 'LineWidth',1, 'Color', 'k');
        peg2 = plot(peg2_pih_err_r, ':', 'LineWidth',1, 'Color', 'k');
        setup_subplot_axes();
        title('Domestic Inflation');
        ylabel('Annualized percentage points');
    % x IRFs
    subplot(2,3,2)
        plot(optimal_results.oo_.irfs.x_err_r(1:20), 'Linewidth', 1, 'Color', 'k
        hold on
        plot(ditr_results.oo_.irfs.x_err_r(1:20), '--', 'LineWidth',1, 'Color',
        plot(citr_results.oo_.irfs.x_err_r(1:20), '-.', 'LineWidth',1, 'Color',
        plot(peg2_results.oo_.irfs.x_err_r(1:20), ':', 'LineWidth',1, 'Color', '
        setup_subplot_axes();
        title('Output gap');
        ylabel('Percentage deviations from steady state');
    % pi
      subplot(2,3,3)
        plot(optimal_pi_err_r, 'Linewidth', 1, 'Color', 'k');
        hold on
        plot(ditr_pi_err_r, '--', 'LineWidth',1, 'Color', 'k');
        plot(citr_pi_err_r, '-.', 'LineWidth',1, 'Color', 'k');
        plot(peg2_pi_err_r, ':', 'LineWidth',1, 'Color', 'k');
        setup_subplot_axes();
        title('CPI inflation');
        ylabel('Annualized percentage points');
    % tau
    subplot(2,3,4)
        plot(optimal_results.oo_.irfs.tau_err_r(1:20), 'Linewidth', 1, 'Color',
        hold on
        plot(ditr_results.oo_.irfs.tau_err_r(1:20), '--', 'LineWidth',1, 'Color'
        plot(citr_results.oo_.irfs.tau_err_r(1:20), '-.', 'LineWidth',1, 'Color'
        plot(peg2_results.oo_.irfs.tau_err_r(1:20), ':', 'LineWidth',1, 'Color',
        setup_subplot_axes();
        title('Terms of trade');
        ylabel('Percentage deviations from steady state');
    % i
```

```
    subplot(2,3,5)
        plot(optimal_i_err_r, 'Linewidth', 1, 'Color', 'k');
        hold on
        plot(ditr_i_err_r, '--', 'LineWidth',1, 'Color', 'k');
        plot(citr_i_err_r, '-.', 'LineWidth',1, 'Color', 'k');
        plot(peg2_i_err_r, ':', 'LineWidth',1, 'Color', 'k');
        setup_subplot_axes();
        title('Nominal interest rate');
        ylabel('Annualized percentage points');
    % delta_e
    subplot(2,3,6)
        plot(optimal_results.oo_.irfs.delta_e_err_r(1:20), 'Linewidth', 1, 'Colo
        hold on
        plot(ditr_results.oo_.irfs.delta_e_err_r(1:20), '--', 'LineWidth',1, 'Co
        plot(citr_results.oo_.irfs.delta_e_err_r(1:20), '-.', 'LineWidth',1, 'Co
        plot(peg2_results.oo_.irfs.delta_e_err_r(1:20), ':', 'LineWidth',1, 'Col
        setup_subplot_axes();
        title('Change in nominal exchange rate');
        ylabel('Percentage deviation from steady state');
    %{
    % Add a global legend using the plots from the first subplot
        lgd = legend([optimal, ditr, citr, peg2], {'Optimal policy', 'Domestic i
        set(lgd, 'location', 'south', 'orientation', 'horizontal', 'fontsize', 1
    % Manually adjust the position to be at the bottom center of the figure
        legendPosition = [0.25, 0.01, 0.5, 0.05];
        set(lgd, 'Position', legendPosition);
        %}
% Saving the graph
saveas(gcf, 'Technology shock.jpg');




% annualise inflation and convert to percentage points
optimal_pih_err_z = (exp(optimal_results.oo_.irfs.pih_err_z(1:20)) - 1) * 4;
ditr_pih_err_z = (exp(ditr_results.oo_.irfs.pih_err_z(1:20)) - 1) * 4;
citr_pih_err_z = (exp(citr_results.oo_.irfs.pih_err_z(1:20)) - 1) * 4;
peg2_pih_err_z = (exp(peg2_results.oo_.irfs.pih_err_z(1:20)) - 1) * 4;

optimal_pi_err_z = (exp(optimal_results.oo_.irfs.pi_err_z(1:20)) - 1) * 4;
ditr_pi_err_z = (exp(ditr_results.oo_.irfs.pi_err_z(1:20)) - 1) * 4;
citr_pi_err_z = (exp(citr_results.oo_.irfs.pi_err_z(1:20)) - 1) * 4;
peg2_pi_err_z = (exp(peg2_results.oo_.irfs.pi_err_z(1:20)) - 1) * 4;

optimal_i_err_z = (exp(optimal_results.oo_.irfs.i_err_z(1:20)) - 1) * 4;
ditr_i_err_z = (exp(ditr_results.oo_.irfs.i_err_z(1:20)) - 1) * 4;
```

```
citr_i_err_z = (exp(citr_results.oo_.irfs.i_err_z(1:20)) - 1) * 4;
peg2_i_err_z = (exp(peg2_results.oo_.irfs.i_err_z(1:20)) - 1) * 4;

% Plotting the demand shock graph
figure('Position', [100, 100, 1200, 800]);
    % pih IRFs
    subplot(2,3,1)
        optimal = plot(optimal_pih_err_z, 'Linewidth', 1, 'Color', 'k');
        hold on
        ditr = plot(ditr_pih_err_z, '--', 'LineWidth',1, 'Color', 'k');
        citr = plot(citr_pih_err_z, '-.', 'LineWidth',1, 'Color', 'k');
        peg2 = plot(peg2_pih_err_z, ':', 'LineWidth',1, 'Color', 'k');
        setup_subplot_axes();
        title('Domestic Inflation');
        ylabel('Annualized percentage points');
    % x IRFs
    subplot(2,3,2)
        plot(optimal_results.oo_.irfs.x_err_z(1:20), 'Linewidth', 1, 'Color', 'k
        hold on
        plot(ditr_results.oo_.irfs.x_err_z(1:20), '--', 'LineWidth',1, 'Color',
        plot(citr_results.oo_.irfs.x_err_z(1:20), '-.', 'LineWidth',1, 'Color',
        plot(peg2_results.oo_.irfs.x_err_z(1:20), ':', 'LineWidth',1, 'Color', '
        setup_subplot_axes();
        title('Output gap');
        ylabel('Percentage deviations from steady state');
    % pi
       subplot(2,3,3)
        plot(optimal_pi_err_z, 'Linewidth', 1, 'Color', 'k');
        hold on
        plot(ditr_pi_err_z, '--', 'LineWidth',1, 'Color', 'k');
        plot(citr_pi_err_z, '-.', 'LineWidth',1, 'Color', 'k');
        plot(peg2_pi_err_z, ':', 'LineWidth',1, 'Color', 'k');
        setup_subplot_axes();
        title('CPI inflation');
        ylabel('Annualized percentage points');
    % tau
    subplot(2,3,4)
        plot(optimal_results.oo_.irfs.tau_err_z(1:20), 'Linewidth', 1, 'Color',
        hold on
        plot(ditr_results.oo_.irfs.tau_err_z(1:20), '--', 'LineWidth',1, 'Color'
        plot(citr_results.oo_.irfs.tau_err_z(1:20), '-.', 'LineWidth',1, 'Color'
        plot(peg2_results.oo_.irfs.tau_err_z(1:20), ':', 'LineWidth',1, 'Color',
        setup_subplot_axes();
        title('Terms of trade');
        ylabel('Percentage deviations from steady state');
    % i
```

```
    subplot(2,3,5)
        plot(optimal_i_err_z, 'Linewidth', 1, 'Color', 'k');
        hold on
        plot(ditr_i_err_z, '--', 'LineWidth',1, 'Color', 'k');
        plot(citr_i_err_z, '-.', 'LineWidth',1, 'Color', 'k');
        plot(peg2_i_err_z, ':', 'LineWidth',1, 'Color', 'k');
        setup_subplot_axes();
        title('Nominal interest rate');
        ylabel('Annualized percentage points');
    % delta_e
    subplot(2,3,6)
        plot(optimal_results.oo_.irfs.delta_e_err_z(1:20), 'Linewidth', 1, 'Colo
        hold on
        plot(ditr_results.oo_.irfs.delta_e_err_z(1:20), '--', 'LineWidth',1, 'Co
        plot(citr_results.oo_.irfs.delta_e_err_z(1:20), '-.', 'LineWidth',1, 'Co
        plot(peg2_results.oo_.irfs.delta_e_err_z(1:20), ':', 'LineWidth',1, 'Col
        setup_subplot_axes();
        title('Change in nominal exchange rate');
        ylabel('Percentage deviation from steady state');
    %{
    % Add a global legend using the plots from the first subplot
        lgd = legend([optimal, ditr, citr, peg2], {'Optimal policy', 'Domestic i
        set(lgd, 'location', 'south', 'orientation', 'horizontal', 'fontsize', 1
    % Manually adjust the position to be at the bottom center of the figure
        legendPosition = [0.25, 0.01, 0.5, 0.05];
        set(lgd, 'Position', legendPosition);
        %}
% Saving the graph
saveas(gcf, 'Demand shock.jpg');




% annualise inflation and convert to percentage points
optimal_pih_err_istar = (exp(optimal_results.oo_.irfs.pih_err_istar(1:20)) - 1)
ditr_pih_err_istar = (exp(ditr_results.oo_.irfs.pih_err_istar(1:20)) - 1) * 4;
citr_pih_err_istar = (exp(citr_results.oo_.irfs.pih_err_istar(1:20)) - 1) * 4;
peg2_pih_err_istar = (exp(peg2_results.oo_.irfs.pih_err_istar(1:20)) - 1) * 4;

optimal_pi_err_istar = (exp(optimal_results.oo_.irfs.pi_err_istar(1:20)) - 1) *
ditr_pi_err_istar = (exp(ditr_results.oo_.irfs.pi_err_istar(1:20)) - 1) * 4;
citr_pi_err_istar = (exp(citr_results.oo_.irfs.pi_err_istar(1:20)) - 1) * 4;
peg2_pi_err_istar = (exp(peg2_results.oo_.irfs.pi_err_istar(1:20)) - 1) * 4;

optimal_i_err_istar = (exp(optimal_results.oo_.irfs.i_err_istar(1:20)) - 1) * 4;
```

```
ditr_i_err_istar = (exp(ditr_results.oo_.irfs.i_err_istar(1:20)) - 1) * 4;
citr_i_err_istar = (exp(citr_results.oo_.irfs.i_err_istar(1:20)) - 1) * 4;
peg2_i_err_istar = (exp(peg2_results.oo_.irfs.i_err_istar(1:20)) - 1) * 4;

% Plotting the foreign shock graph
figure('Position', [100, 100, 1200, 800]);
    % pih IRFs
    subplot(2,3,1)
        optimal = plot(optimal_pih_err_istar, 'Linewidth', 1, 'Color', 'k');
        hold on
        ditr = plot(ditr_pih_err_istar, '--', 'LineWidth',1, 'Color', 'k');
        citr = plot(citr_pih_err_istar, '-.', 'LineWidth',1, 'Color', 'k');
        peg2 = plot(peg2_pih_err_istar, ':', 'LineWidth',1, 'Color', 'k');
        setup_subplot_axes();
        title('Domestic Inflation');
        ylabel('Annualized percentage points');
    % x IRFs
    subplot(2,3,2)
        plot(optimal_results.oo_.irfs.x_err_istar(1:20), 'Linewidth', 1, 'Color'
        hold on
        plot(ditr_results.oo_.irfs.x_err_istar(1:20), '--', 'LineWidth',1, 'Colo
        plot(citr_results.oo_.irfs.x_err_istar(1:20), '-.', 'LineWidth',1, 'Colo
        plot(peg2_results.oo_.irfs.x_err_istar(1:20), ':', 'LineWidth',1, 'Color
        setup_subplot_axes();
        title('Output gap');
        ylabel('Percentage deviations from steady state');
    % pi
       subplot(2,3,3)
        plot(optimal_pi_err_istar, 'Linewidth', 1, 'Color', 'k');
        hold on
        plot(ditr_pi_err_istar, '--', 'LineWidth',1, 'Color', 'k');
        plot(citr_pi_err_istar, '-.', 'LineWidth',1, 'Color', 'k');
        plot(peg2_pi_err_istar, ':', 'LineWidth',1, 'Color', 'k');
        setup_subplot_axes();
        title('CPI inflation');
        ylabel('Annualized percentage points');
    % tau
    subplot(2,3,4)
        plot(optimal_results.oo_.irfs.tau_err_istar(1:20), 'Linewidth', 1, 'Colo
        hold on
        plot(ditr_results.oo_.irfs.tau_err_istar(1:20), '--', 'LineWidth',1, 'Co
        plot(citr_results.oo_.irfs.tau_err_istar(1:20), '-.', 'LineWidth',1, 'Co
        plot(peg2_results.oo_.irfs.tau_err_istar(1:20), ':', 'LineWidth',1, 'Col
        setup_subplot_axes();
        title('Terms of trade');
        ylabel('Percentage deviations from steady state');
```

```
    % i
    subplot(2,3,5)
        plot(optimal_i_err_istar, 'Linewidth', 1, 'Color', 'k');
        hold on
        plot(ditr_i_err_istar, '--', 'LineWidth',1, 'Color', 'k');
        plot(citr_i_err_istar, '-.', 'LineWidth',1, 'Color', 'k');
        plot(peg2_i_err_istar, ':', 'LineWidth',1, 'Color', 'k');
        setup_subplot_axes();
        title('Nominal interest rate');
        ylabel('Annualized percentage points');
    % delta_e
    subplot(2,3,6)
        plot(optimal_results.oo_.irfs.delta_e_err_istar(1:20), 'Linewidth', 1, '
        hold on
        plot(ditr_results.oo_.irfs.delta_e_err_istar(1:20), '--', 'LineWidth',1,
        plot(citr_results.oo_.irfs.delta_e_err_istar(1:20), '-.', 'LineWidth',1,
        plot(peg2_results.oo_.irfs.delta_e_err_istar(1:20), ':', 'LineWidth',1,
        setup_subplot_axes();
        title('Change in nominal exchange rate');
        ylabel('Percentage deviation from steady state');
    %{
    % Add a global legend using the plots from the first subplot
        lgd = legend([optimal, ditr, citr, peg2], {'Optimal policy', 'Domestic i
        set(lgd, 'location', 'south', 'orientation', 'horizontal', 'fontsize', 1
    % Manually adjust the position to be at the bottom center of the figure
        legendPosition = [0.25, 0.01, 0.5, 0.05];
        set(lgd, 'Position', legendPosition);
        %}
% Saving the graph
saveas(gcf, 'Foreign shock.jpg');


%------------------------------------------------------------------------
% 3. Welfare Loss Calculation
%------------------------------------------------------------------------

% cost-push shock
loss_optimal_u = calculate_welfare_loss(optimal_results.oo_.irfs.x_err_u, optima
loss_ditr_u = calculate_welfare_loss(ditr_results.oo_.irfs.x_err_u, ditr_results
loss_citr_u = calculate_welfare_loss(citr_results.oo_.irfs.x_err_u, citr_results
loss_peg1_u = calculate_welfare_loss(peg1_results.oo_.irfs.x_err_u, peg1_results
loss_peg2_u = calculate_welfare_loss(peg2_results.oo_.irfs.x_err_u, peg2_results

% shock to r
loss_optimal_r = calculate_welfare_loss(optimal_results.oo_.irfs.x_err_r, optima
loss_ditr_r = calculate_welfare_loss(ditr_results.oo_.irfs.x_err_r, ditr_results
```

```
loss_citr_r = calculate_welfare_loss(citr_results.oo_.irfs.x_err_r, citr_results
loss_peg1_r = calculate_welfare_loss(peg1_results.oo_.irfs.x_err_r, peg1_results
loss_peg2_r = calculate_welfare_loss(peg2_results.oo_.irfs.x_err_r, peg2_results

% shock to z
loss_optimal_z = calculate_welfare_loss(optimal_results.oo_.irfs.x_err_z, optima
loss_ditr_z = calculate_welfare_loss(ditr_results.oo_.irfs.x_err_z, ditr_results
loss_citr_z = calculate_welfare_loss(citr_results.oo_.irfs.x_err_z, citr_results
loss_peg1_z = calculate_welfare_loss(peg1_results.oo_.irfs.x_err_z, peg1_results
loss_peg2_z = calculate_welfare_loss(peg2_results.oo_.irfs.x_err_z, peg2_results

% shock to i
loss_optimal_istar = calculate_welfare_loss(optimal_results.oo_.irfs.x_err_istar
loss_ditr_istar = calculate_welfare_loss(ditr_results.oo_.irfs.x_err_istar, ditr
loss_citr_istar = calculate_welfare_loss(citr_results.oo_.irfs.x_err_istar, citr
loss_peg1_istar = calculate_welfare_loss(peg1_results.oo_.irfs.x_err_istar, peg1
loss_peg2_istar = calculate_welfare_loss(peg2_results.oo_.irfs.x_err_istar, peg2

% create table

loss_table = {"Shocks", "Optimal Policy", "Domestic Inflation Targeting", "CPI T
              "Cost-push shock (u)", loss_optimal_u, loss_ditr_u, loss_citr_u, l
              "Technology shock (r^n)", loss_optimal_r, loss_ditr_r, loss_citr_r
              "Demand shock (z)", loss_optimal_z, loss_ditr_z, loss_citr_z, loss
              "Foreign shock (i*)", loss_optimal_istar, loss_ditr_istar, loss_ci
              };

dataframe(loss_table);
```

Optimal policy mod file

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

close all;

%----------------------------------------------------------------
% 1. Defining variables
%----------------------------------------------------------------

var x z i istar r delta_e pi pih tau u;

varexo err_u err_r err_z err_istar;

parameters alpha beta epsilon theta kappa lambda phi rho;

%----------------------------------------------------------------
% 2. Calibration
%----------------------------------------------------------------

alpha = .4;
beta = .99;
epsilon = 6;
theta = .75;
phi = 3;
rho = .9;
lambda = (1 - theta) * (1 - beta * theta) / theta;
kappa = lambda * (1 + phi);


%----------------------------------------------------------------
% 3. Model
%----------------------------------------------------------------

model;

  pih = kappa * x + beta * pih(+1) + u; % AS function
  x = -(i - pih(+1) - r) + x(+1); % AD function
  pi = pih + alpha * (tau - tau(-1)); % domestic inflation & term of trade
  x = z + tau; % domestic output gap
  i(-1) = istar(-1) + delta_e; % uncovered interest rate parity
  pih = - (x - x(-1)) / epsilon; % optimal policy rule
```

```
  istar = rho *  istar(-1) + err_istar; % shocks, AR(1)
  r = rho * r(-1) + err_r;
  u = rho * u(-1) + err_u;
  z = rho * z(-1) + err_z;

end;


initval;

  x = 0;
  pi = 0;
  pih = 0;
  tau = 0;
  i = 0;
  delta_e = 0;

  istar = 0;
  r = 0;
  u = 0;
  z = 0;


end;


shocks;

  var err_u;
  stderr 1;
  var err_r;
  stderr 1;
  var err_z;
  stderr 1;
  var err_istar;
  stderr 1;

  end;

1;


stoch_simul(order=1, irf=100, irf_plot_threshold=0) x pih pi tau i delta_e;


%------------------------------------------------------------------------
```

```
% 4. Save the Results
%------------------------------------------------------------------------

optimal_results = oo_;
save optimal_results;
```

Domestic inflation mod file

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

close all;

%----------------------------------------------------------------
% 1. Defining variables
%----------------------------------------------------------------

var x z i istar r delta_e pi pih tau u;

varexo err_u err_r err_z err_istar;

parameters alpha beta epsilon theta kappa lambda phi psi_pi rho;

%----------------------------------------------------------------
% 2. Calibration
%----------------------------------------------------------------

alpha = .4;
beta = .99;
epsilon = 6;
theta = .75;
phi = 3;
psi_pi = 1.5;
rho = .9;
lambda = (1 - theta) * (1 - beta * theta) / theta;
kappa = lambda * (1 + phi);

%----------------------------------------------------------------
% 3. Model
%----------------------------------------------------------------

model;

  pih = kappa * x + beta * pih(+1) + u; % AS function
  x = -(i - pih(+1) - r) + x(+1); % AD function
  pi = pih + alpha * (tau - tau(-1)); % domestic inflation & term of trade
  x = z + tau; % domestic output gap
  i(-1) = istar(-1) + delta_e; % uncovered interest rate parity
  i = psi_pi * pih; % monetary policy rule
```

```
  istar = rho *  istar(-1) + err_istar; % shocks, AR(1)
  r = rho * r(-1) + err_r;
  u = rho * u(-1) + err_u;
  z = rho * z(-1) + err_z;

end;


initval;

  x = 0;
  pi = 0;
  pih = 0;
  i = 0;
  tau = 0;
  delta_e = 0;

  istar = 0;
  r = 0;
  u = 0;
  z = 0;

end;


shocks;

  var err_u;
  stderr 1;
  var err_r;
  stderr 1;
  var err_z;
  stderr 1;
  var err_istar;
  stderr 1;

end;


stoch_simul(order=1, irf=100, irf_plot_threshold=0) x pih pi tau i delta_e;


%-------------------------------------------------------------------
% 4. Save the Results
%-------------------------------------------------------------------
```

```
ditr_results = oo_;
save ditr_results;
```

CPI mod file

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

close all;

%----------------------------------------------------------------
% 1. Defining variables
%----------------------------------------------------------------

var x z i istar r delta_e pi pih tau u;

varexo err_u err_r err_z err_istar;

parameters alpha beta epsilon theta kappa lambda phi psi_pi rho;

%----------------------------------------------------------------
% 2. Calibration
%----------------------------------------------------------------

alpha = .4;
beta = .99;
epsilon = 6;
theta = .75;
phi = 3;
psi_pi = 1.5;
rho = .9;
lambda = (1 - theta) * (1 - beta * theta) / theta;
kappa = lambda * (1 + phi);

%----------------------------------------------------------------
% 3. Model
%----------------------------------------------------------------

model;

  pih = kappa * x + beta * pih(+1) + u; % AS function
  x = -(i - pih(+1) - r) + x(+1); % AD function
  pi = pih + alpha * (tau - tau(-1)); % domestic inflation & term of trade
  x = z + tau; % domestic output gap
  i(-1) = istar(-1) + delta_e; % uncovered interest rate parity
  i = psi_pi * pi; % monetary policy rule
  istar = rho *  istar(-1) + err_istar; % shocks, AR(1)
```

```
  r = rho * r(-1) + err_r;
  u = rho * u(-1) + err_u;
  z = rho * z(-1) + err_z;

end;


initval;

  x = 0;
  pi = 0;
  pih = 0;
  i = 0;
  tau = 0;
  delta_e = 0;

  istar = 0;
  r = 0;
  u = 0;
  z = 0;

end;


shocks;

  var err_u;
  stderr 1;
  var err_r;
  stderr 1;
  var err_z;
  stderr 1;
  var err_istar;
  stderr 1;

end;


stoch_simul(order=1, irf=100, irf_plot_threshold=0) x pih pi tau i delta_e;


%-----------------------------------------------------------------
% 4. Save the Results
%-----------------------------------------------------------------

citr_results = oo_;
```

```
save citr_results;
```

Peg with $i^* = 0$ mod file

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

close all;

%----------------------------------------------------------------
% 1. Defining variables
%----------------------------------------------------------------

var x z i istar r delta_e pi pih tau u;

varexo err_u err_r err_z err_istar;

parameters alpha beta epsilon theta kappa lambda phi rho sigma;

%----------------------------------------------------------------
% 2. Calibration
%----------------------------------------------------------------

alpha = .4;
beta = .99;
epsilon = 6;
theta = .75;
phi = 3;
rho = .9;
sigma = 0.01;
lambda = (1 - theta) * (1 - beta * theta) / theta;
kappa = lambda * (1 + phi);

%----------------------------------------------------------------
% 3. Model
%----------------------------------------------------------------

model;

  pih = kappa * x + beta * pih(+1) + u; % AS function
  x = -(i - pih(+1) - r) + x(+1); % AD function
  pi = pih + alpha * (tau - tau(-1)); % domestic inflation & term of trade
  x = z + tau; % domestic output gap
  i(-1) = istar(-1) + delta_e + sigma * pih; % uncovered interest rate parity
  delta_e = 0;
  istar = rho *  istar(-1) + err_istar; % shocks, AR(1)
```

```
  r = rho * r(-1) + err_r;
  u = rho * u(-1) + err_u;
  z = rho * z(-1) + err_z;

end;


initval;

  x = 0;
  pi = 0;
  pih = 0;
  i = 0;
  tau = 0;
  delta_e = 0;

  istar = 0;
  r = 0;
  u = 0;
  z = 0;

end;


shocks;

  var err_u;
  stderr 1;
  var err_r;
  stderr 1;
  var err_z;
  stderr 1;
  var err_istar;
  stderr 1;

end;


stoch_simul(order=1, irf=100, irf_plot_threshold=0) x pih pi tau i delta_e;


%----------------------------------------------------------------
% 4. Save the Results
%----------------------------------------------------------------

peg_results = oo_;
```

```
save peg_results;
```

Peg with risk premium mod file

```
%——————————————————————————————————————————
% Close all graphic windows
%——————————————————————————————————————————

close all;

% ——————————————————————————————————————————
% Declaring variables
% ——————————————————————————————————————————

var x tau pi pih i istar r u z delta_e pistar y yn;

varexo err_r err_u err_z err_istar;

parameters alpha beta epsilon theta phi lambda kappa ro;

alpha = 0.4;
beta = 0.99;
epsilon = 6;
theta = 0.75;
phi = 3;
lamba = ((1 − theta) ∗ (1 − beta ∗ theta)) / theta;
kappa = lamba ∗ (1 + phi);
ro = 0.9;

% ——————————————————————————————————————————
% MODEL
% ——————————————————————————————————————————

model;

    x = y − yn;
    pih = kappa ∗ x + beta ∗ pih(+1) + u;
    pi = pih + alpha ∗ (tau − tau(−1));
    x = x(+1) − (i − pih(+1) − r);
    x = z + tau;
    i = istar;
    tau = tau(−1) − pih + pistar;

    delta_e = 0;
    pistar = 0;

    r = 0.9 ∗ r(−1) + err_r;
    u = 0.9 ∗ u(−1) + err_u;
```

```
    z = 0.9 * z(-1) + err_z;
    istar = 0.9 * istar(-1) + err_istar;

end;

initval;
    x = 0;
    tau = 0;
    pi = 0;
    pih = 0;
    i = 0;
    istar = 0;
    r = 0;
    u = 0;
    z = 0;
end;


shocks;

    var err_u;
    stderr 1;
    var err_r;
    stderr 1;
    var err_z;
    stderr 1;
    var err_istar;
    stderr 1;

end;

stoch_simul(order = 1, irf=100, irf_plot_threshold = 0) pih, x, pi, tau, i, delt
```

IMF, Problem Set 2

1068576

4 December 2023

Q1.

The household's problem is:

$$\max_{\{c_t, b_{t+1}\}_{t=1}^{T}} u(c_t) + \beta \mathbb{E}[u(c_{t+1})] \tag{1}$$

subject to:

$$c_t = y_t - b_t + q(b_{t+1})b_{t+1} \tag{2}$$

$$c_T = \max_{D \in \{0,1\}} \{(1-D)[y_T - b_T] + D[y_t - \phi(y_T)]\} \tag{3}$$

where equation (3) is the terminal condition at time period T. This problem can be rewritten recursively as:

$$v^c(b_t) = max_{b_{t+1}} u(y_t - b_t + q(b_{t+1}b_{t+1}) + \beta max\{\mathbb{E}V^b_{t+1}, \mathbb{E}V^c(b_{t+1})\} \tag{4}$$

when the household has access to the international debt market and as:

$$v^b_t = u(y_t - \phi(y_t)) + \sum_{j=1}^{T-t} \beta^j [\pi u(y_H - \hat{\phi}) + (1-\pi)u(y_L)] \tag{5}$$

after the household loses access to the debt market. y_t follows the following distribution:

$$y_t = \begin{cases} y_H, & \text{w.p. } \pi \\ y_L, & \text{w.p. } 1-\pi \end{cases} \tag{6}$$

and the cost of default is defined as:

$$\phi(y_t) = \begin{cases} \hat{\phi}, & \text{if } y_t = y_H \\ 0, & \text{if } y_t = y_L \end{cases} \tag{7}$$

Finally, the price of debt is given by the function:

$$q(b_t) = \beta^* \begin{cases} 1, & \text{if } b_t =\leq 0 \\ (1 - \mathbb{E}[\pi_{def}])^\sigma, & \text{if } b_t \in (0, \hat{\phi}) \\ 0, & \text{if } b_t \geq \hat{\phi} \end{cases} \tag{8}$$

where we assume a risk-neutral lender, such that $\sigma = 1$. β^* is the quarterly risk-free rate derived from the subjective discount factor, and $\mathbb{E}[\pi_{def}]$ is the expected probability of default, which is solved numerically. The household's default decision follows:

$$D(b_t) = argmax_{D\in\{0,1\}}\{(1 - D)V^c(b_t) + DV_t^b\}, \quad t < T \tag{9}$$

where we assume no possibility of re-entry into the debt market, such that if the household defaults, it receives utility given by the value function (5) until T.

Q2.

We solve for the model above using a value function iteration algorithm (see Appendix). The parameters and constants used to calibrate the model are presented in table (1).

Table 1: Calibration values

Description	Notation	Calibration value
risk aversion	γ	2.0
subjective discount factor	β	0.97
prob. of good state	π	0.5
endowment in good state	y_H	1.1
endowment in bad state	y_L	0.90
time horizon	T	30

The policy functions for default and borrowing, as well as the equilibrium price of debt are given in figure 1.

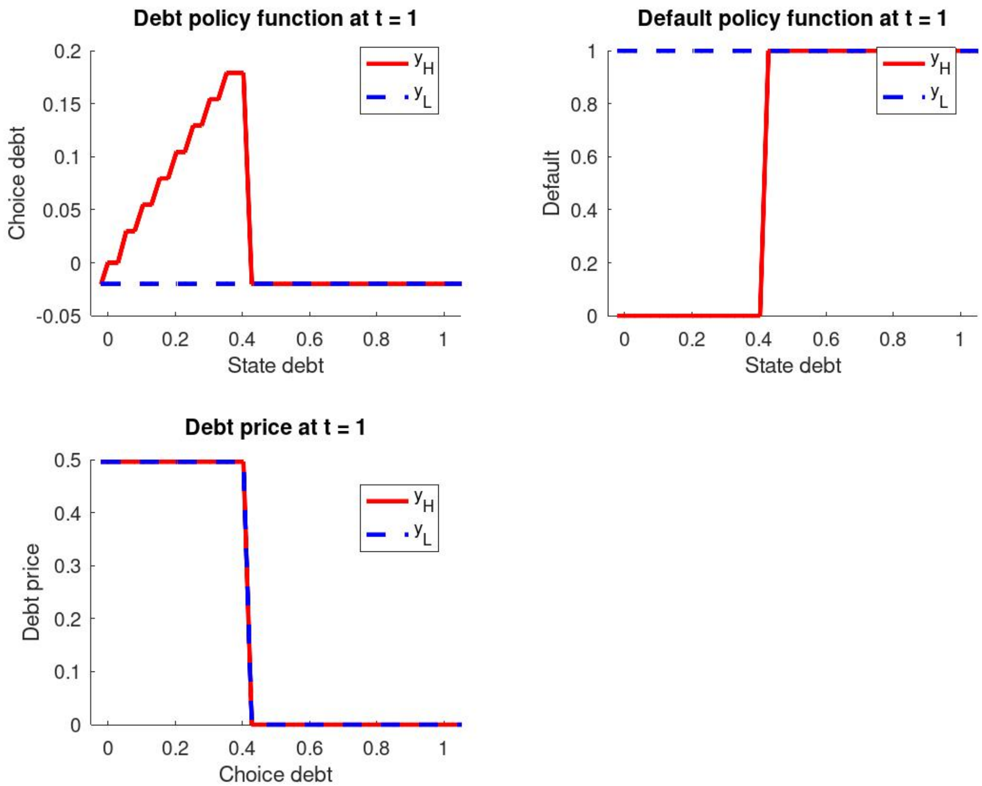


Figure 1: Policy functions and debt price

From the top-right panel of figure (1), we observe that the household defaults at all debt levels when endowment is low to benefit from the low cost of default. Consequently, the household is excluded from the debt market, such that choice debt equals zero.

When endowment is high on the other hand, the household borrows more when the state debt level is higher, to smoothe consumption. Beyond a debt level of approximately 0.4, the household decides to default because the cost of repaying the debt accumulated from the previous period outweighs the benefits of repaying.

The price of debt reflects the current default policy function under rational expectation. The risk-neutral lender bases her belief of the likelihood of default tomorrow on today's default policy function of the borrower.

Q3.

Figure (2) shows the value function when the endowment is high, under the same calibration as in Q2. but with varying time horizons. We observe that the difference between the value functions in the first and second periods becomes smaller as the model's terminal period is extended. This could be because under a long time horizon, the cost of defaulting early is high due to the cumulative cost of exclusion. Therefore, the household decides not to default in both the first and second period, and the small difference between the value functions only reflects the higher risk premium the household in the second period has to pay to the risk-neutral lender to compensate for the higher perceived risk of default.

For shorter time horizons by contrast, it is optimal for the household to default because the cost of future exclusion is low. However, defaulting in the second period is significantly less costly relative to defaulting in the first period due to the short time horizon, which explains the large difference between the value functions.

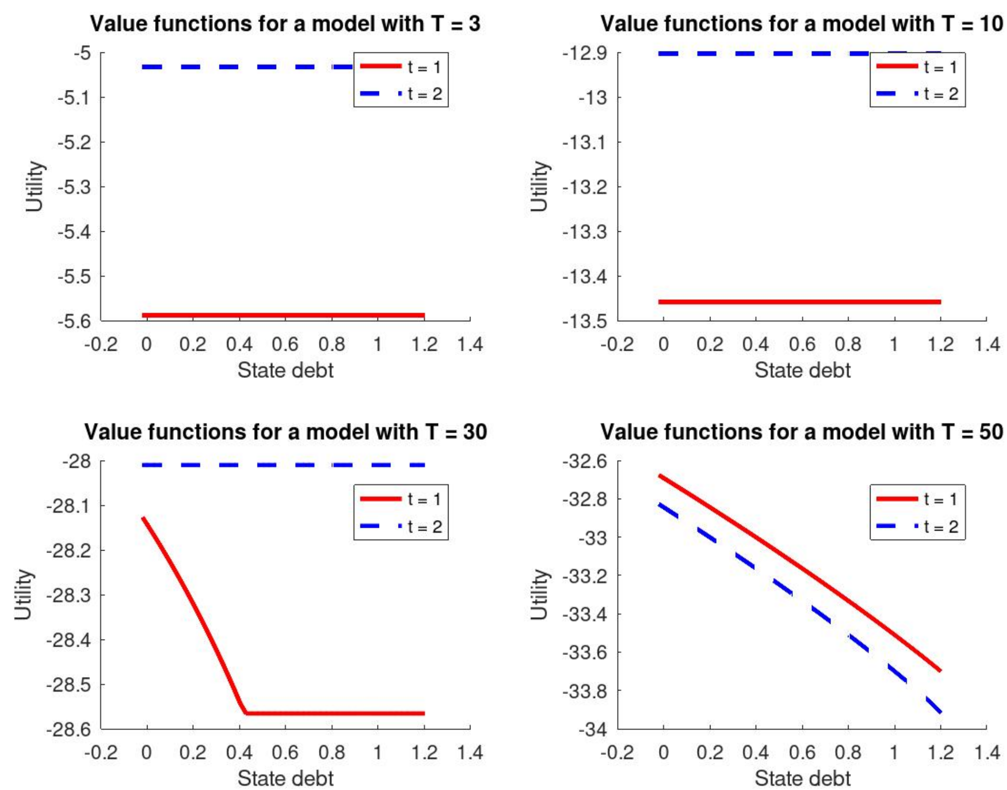


Figure 2: Value function at t = 1 and t = 2 for various time horizons

Q4.

Figures (3) to (5) show the default policy function evaluated at various parameter values. All parameters other than the parameter of interest are calibrated according to table (1).

Unsurprisingly, figure (3) shows that the household defaults at all levels of state debt when the cost of default is low, and does not default when the cost of default is high. At an intermediary cost of default, the household defaults whenever the state debt is large enough such that the benefit of not repaying outweighs the cost of default.

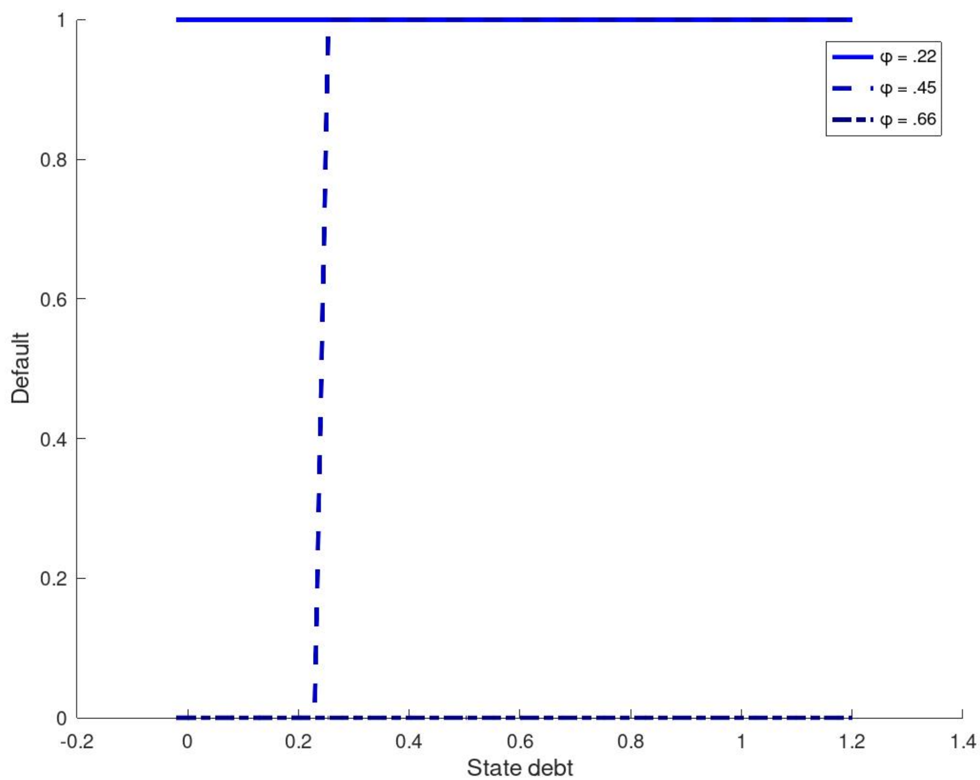


Figure 3: Default policy function given various default costs

From figure (4), we observe that a more risk-averse household defaults at a lower level of state debt. This result is puzzling given that a higher level of risk aversion implies a higher preference for consumption smoothing, which should imply that the household has an incentive to retain access to the debt market.

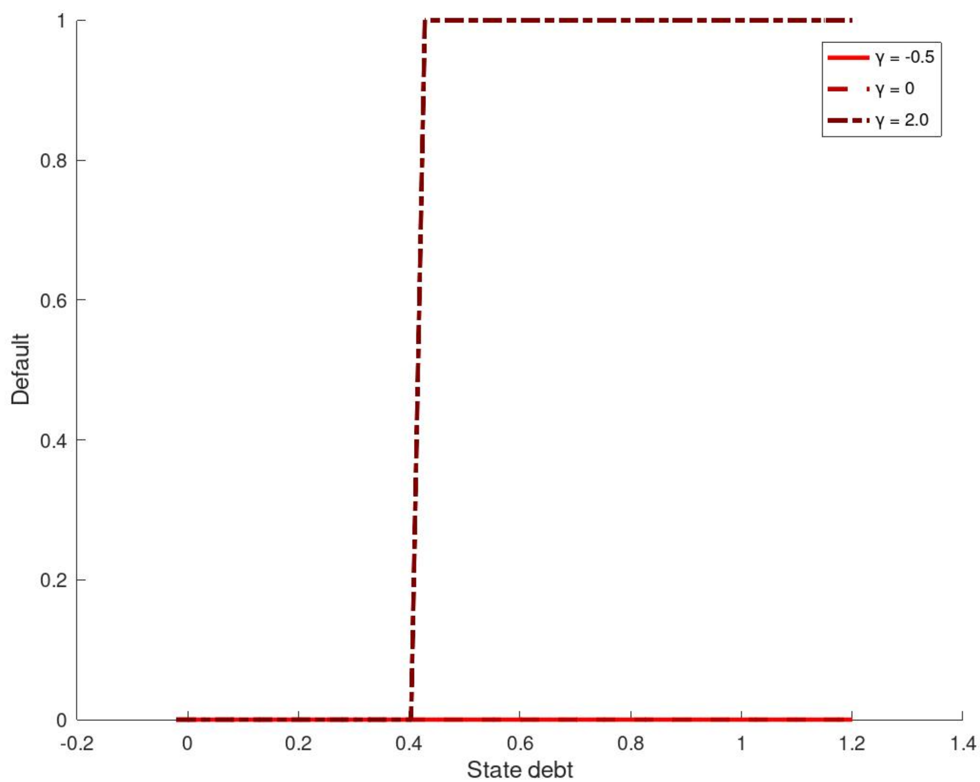


Figure 4: Default policy function given various borrower risk aversions

Figure (5) shows the default policy function for various levels of lender risk aversion. From equation (8), we can set $\sigma < 1$ to model a risk-loving agent and $\sigma > 1$ to model a risk-neutral agent.

From the figure, we immediately observe that a more risk-averse lender implies that the borrower defaults at a lower level of state debt. From figure (6), we can infer that a higher level of lender risk aversion leads to a lower debt price since the lender will require a higher risk premium, making borrowing more costly to the household. Therefore, the cost of being excluded from the debt market is lower for the household if the lender is risk-neutral, which leads to a lower threshold for default.

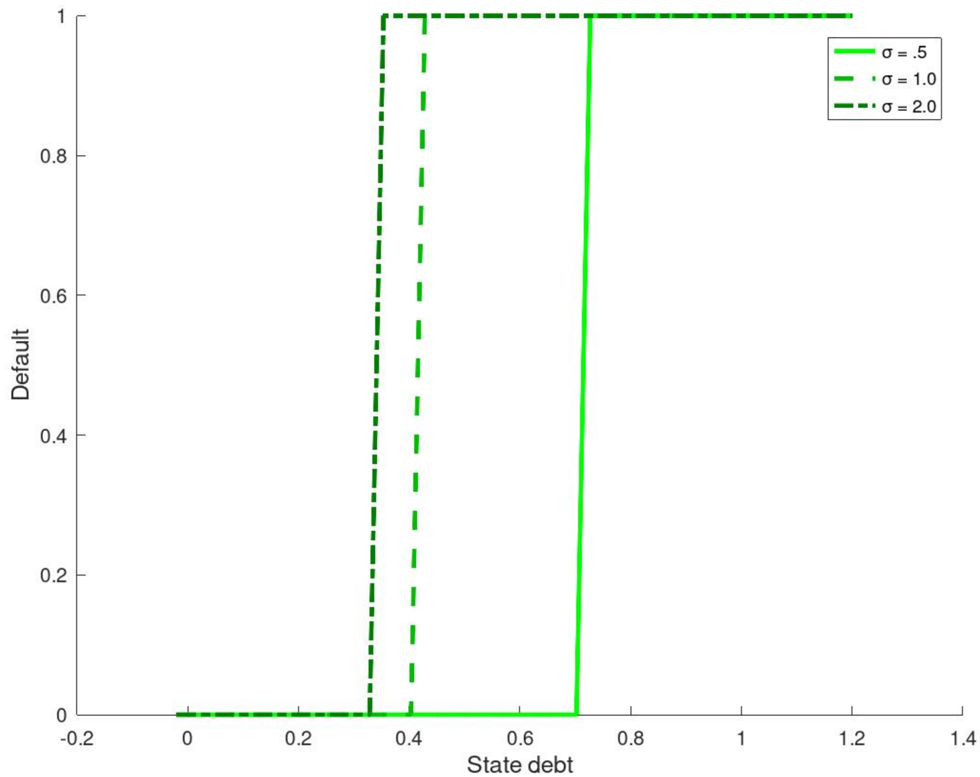


Figure 5: Default policy function given various lender risk aversions

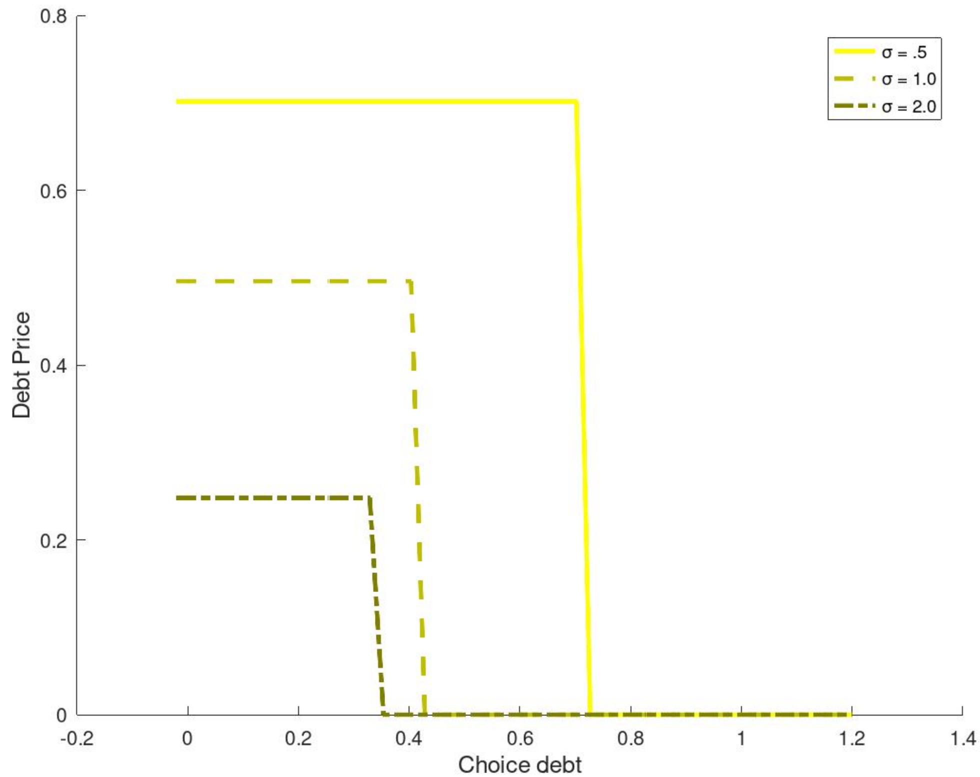


Figure 6: Price of debt given various lender risk aversions

Appendix A: Code

Main file

```
%--------------------------------------------------------------------------
%
% Title: International Macro-Finance Problem Set 2, main file
% Author: Rinto Fujimoto
% Date: 25/11/2023
% Description: Sovereign default model with T periods
%
%--------------------------------------------------------------------------

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

close all;
clear all;

cd '/home/rinto/Desktop/International Macro/PS2'

%--------------------------------------------------------------------------
% 1. Defining parameters, grid and variables
%--------------------------------------------------------------------------

% parameters:
par.gamma = 2 % household's RRA preference
par.beta = .97; % subjective discount factor
par.pi = .5; % probability of being in the good state


% constants:
cons.phi_hat = .5; % cost of default in the good state
cons.yh = 1.1; % endowment in good state
cons.yl = .9; % endowment in bad state
cons.rf = par.beta^(-1) - 1; % risk-free rate

% number of time periods:
t = 30;

% define convergence criterion
maxit_q = 1000; % max iteration to solve for q
maxit_v = 1000; % max iteration to solve for v
err_tol_q = 1e-10; % error tolerance
err_tol_v = 1e-10;
lambda_q = .5; % dampening parameter
lambda_v = .5;


% transition matrix
Pr = [par.pi, 1 - par.pi;
      par.pi, 1 - par.pi];


```

```
% derived values:
qrf = (1 + repmat(cons.rf, [2, 1])).^-(1/4); % quarterly risk-free rate


%----------------------------------------------------------------------
% 2. Defining grid
%----------------------------------------------------------------------

n = 50; % grid points for b

b_min = -.02; % min value for b
b_max = 1.2; % max value for b

b_vec = linspace(b_min, b_max, n); % grid for debt, 1 x n vector
y_vec = [cons.yh cons.yl].'; % grid for endowment, 2 x 1 vector

% Ensure presence of 0 on b-vector
[~,i_b_zero] = min(abs(b_vec));
% i_b_zero captures the index of the cell containing 0
b_vec(i_b_zero) = 0;

% grid:
grid.b_state3 = repmat(b_vec, [2, 1, n]); % state debt, 2 x n x n
grid.b_choice2 = repmat(b_vec, [2, 1]); % choice debt, 2 x n
grid.y_state3 = repmat(y_vec, [1, n, n]); % state output, 2 x n x n

grid.borr_choice2 = zeros(2, n, t); % grid which will contain debt value
grid.borr_choice3 = zeros(2, n, n, t);
% grid.borr_choice2 augmented by 1 dimension to account for choice debt



%----------------------------------------------------------------------
% 3. Setting Initial Guesses
%----------------------------------------------------------------------

guess.v = zeros(2, n, t); % value function
guess.v_new = zeros(2, n, t);

guess.i_b = zeros(2, n, t); % debt choice index (to be extracted from the grid)

guess.q = qrf(1, 1) * ones(2, n, t); % price of debt, 2 x n x t matrix
guess.def = zeros(2, n, t); % default choice


%----------------------------------------------------------------------
% 4. Defining functions
%----------------------------------------------------------------------


function u = util(c, gamma)
% this is the utility function (CRRA)
  u = c.^(1 - gamma) / (1 - gamma);
end



%----------------------------------------------------------------------
```

```
% 5. Other variables
%--------------------------------------------------------------------------

% storage value
store.v = zeros(2, n, t);
store.i_b = zeros(2, n, t);
store.q = qrf(1, 1) * ones(2, n, t);
store.def = zeros(2, n, t);

% policy functions
policy.b = zeros(2, n, t);
policy.def = zeros(2, n, t);

% others
con_choice = zeros(2, n, n, t); % consumption choice for non-default
util_choice = zeros(2, n, n, t); % utility choice for non-default
borr_maximand = zeros(2, n, n, t); % to be maximised over choice debt
e_def = zeros(2, n, t); % expected default prob.
e_v = zeros(2, n); % continuation value
e_v3 = zeros(2, n, n); % continuation value, augmented
e_v_def = zeros(2, n); % continuation value after default

% sum of discount factors (i.e. (1 - beta^(T-t)) / (1 - beta))
discount = zeros(1, t);
for i = 1:t
  discount(:, i) = par.beta * ((1 - par.beta^(i)) / (1 - par.beta)); % scalar
end
discount = flip(discount, 2);

% consumption in default
con_def = [cons.yh - cons.phi_hat, cons.yl].'; % 2 x 1
con_def3 = repmat(con_def, [1, n, n]);


% Expected value of exclusion state
e_v_def =  discount.' * Pr(1, :) * util(con_def, par.gamma); % t x 1

% evaluate value function in default state
v_def = repmat(util(con_def, par.gamma), [1, n, t]) + permute(repmat(e_v_def, [1, n, 2]), [3, 2, 1]);
% 2 x n x t



%--------------------------------------------------------------------------
% 5. Value function iteration
%--------------------------------------------------------------------------


err_q = 7;
err_v = 7;
iter_q = 1;
iter_v = 1;



while err_q > err_tol_q && iter_q < maxit_q

```

```
while err_v > err_tol_v && iter_v < maxit_v

  for i = 1:t

    % Expected continuation value
    e_v = Pr * guess.v(:, :, i);
    e_v3 = permute(repmat(e_v(:, :), [1, 1, n]), [1, 3, 2]); % 2 x n x n matrix
    e_v_def = Pr * v_def(:, :, i);

    optimal_choice = max(e_v3(:, :, i), e_v_def);

    % Resources from borrowing
    grid.borr_choice2(:, :, i) = guess.q(:, :, i) .* grid.b_choice2; % 2 x n matrix
    grid.borr_choice3(:, :, :, i) = repmat(grid.borr_choice2(:, :, i), [1, 1, n]); % 2 x n x n
    grid.borr_choice3(:, :, :, i) = permute(grid.borr_choice3(:, :, :, i), [1, 3, 2]);

    if i < t
      % Consumption implied by choice and state b and guess q
      con_choice(:, :, :, i) = grid.y_state3 - grid.b_state3 + grid.borr_choice3(:, :, :, i); % 2 x n x n matrix
    else
      % Terminal condition
      con_choice(:, :, :, i) = max(grid.y_state3 - grid.b_state3, con_def3);
    end

    con_choice(con_choice < eps) = eps; % rule out negative consumption

    % Period-utility implied by state and choice b
    util_choice(:, :, :, i) = util(con_choice(:, :, :, i), par.gamma); % 2 x 100 x 100

    % Formulate maximand in borrowing choice
    borr_maximand(:, :, :, i) = util_choice(:, :, :, i) + par.beta * e_v3(:, :, :); % 2 x 100 x 100

    % Store old choice for borrowing
    store.i_b(:, :, i) = guess.i_b(:, :, i); % 2 x n

    % Maximise over b' and update value function
    [guess.v_new(:, :, i), guess.i_b(:, :, i)] = max(borr_maximand(:, :, :, i), [], 3); % 2 x n and 2 x n

    % Ensures that HH cannot borrow once defaulted
    guess.i_b(:, :, i) = guess.def(:, :, i) .* ones(2, n) * i_b_zero + (1 - guess.def(:, :, i)) .* guess.i_b(:, :, i);

  end

  %Store old default policy function
  store.def = guess.def;

  % Evaluate policy function for default (indicator function)
  guess.def = v_def > guess.v_new;
  % compare two 2 x n x t matrices

```

```
    % Evaluate value function including discrete choice for default
    guess.v_new = max(v_def, guess.v_new);

    % Store old value-function guesses
    store.v = guess.v;

    % Update value-function guesses
    guess.v = lambda_v * guess.v_new + (1 - lambda_v) * store.v;

    % Evaluate change in v and compare to error tolerance
    err_v = max(abs(guess.v(:) - store.v(:)));

    iter_v
    err_v

    iter_v = iter_v + 1;

end

  % expected probability of default
  for i = 1:t
    e_def(:, :, i) = Pr * guess.def(:, :, i);
  end


  % Store old sovereign debt price
  store.q = guess.q;

  guess.q = qrf(1, 1) * (1 - e_def);

  % Evaluate change in q_g
  err_q = max(abs(guess.q(:) - store.q(:)));

  % Update sovereign debt price
  guess.q = lambda_q * guess.q + (1 - lambda_q) * store.q;

  iter_q
  err_q

  iter_q = iter_q + 1;

  % Reset counter and diff for inner-v loop
  err_v = 7;
  iter_v = 1;

end


policy.b = repmat(grid.b_choice2, [1, 1, t]);
policy.b = policy.b(store.i_b);

policy.def = store.def;

%--------------------------------------------------------------------------
% 6.Plot the value function and policy functions
%--------------------------------------------------------------------------


```

```matlab
% Policy functions and debt price at t = 1
figure
  subplot(2, 2, 1);
    hold on;
    plot(b_vec, policy.b(1, :, 1), 'r-', 'LineWidth', 2.5);
    plot(b_vec, policy.b(2, :, 1), 'b--', 'LineWidth', 2.5);
    xlim([-.05 1.05]);
    xlabel('State debt'), ylabel('Choice debt');
    title('Debt policy function at t = 1');
    legend('y_{H}', 'y_{L}', 'Location');
    hold off;
  subplot(2, 2, 2);
    hold on;
    plot(b_vec, policy.def(1, :, 1), 'r-', 'LineWidth', 2.5);
    plot(b_vec, policy.def(2, :, 1), 'b--', 'LineWidth', 2.5);
    xlim([-.05 1.05]);
    xlabel('State debt'), ylabel('Default');
    title('Default policy function at t = 1');
    legend('y_{H}', 'y_{L}', 'Location');
    hold off;
  subplot(2, 2, 3);
    hold on;
    plot(b_vec, store.q(1, :, 1), 'r-', 'LineWidth', 2.5);
    plot(b_vec, store.q(2, :, 1), 'b--', 'LineWidth', 2.5);
    xlim([-.05 1.05]);
    xlabel('Choice debt'), ylabel('Debt price');
    title('Debt price at t = 1');
    legend('y_{H}', 'y_{L}', 'Location');
    hold off;

saveas(gcf, 'Policy_functions_Q2.jpg');



figure;
  hold on;
  % Value function for high output at t = 1 (solid red line)
  plot(b_vec, store.v(1, :, 1), 'r', 'LineWidth', 2.5);
  % Value function for high output at t = 2 (solid blue line)
  plot(b_vec, store.v(1, :, 2), 'Color', 'b', 'LineWidth', 2.5);
  % Value function for low output at t = 1 (dashed red line)
  plot(b_vec, store.v(2, :, 1), 'Color', 'r', '--', 'LineWidth', 2.5);
  % Value function for low output at t = 2 (dashed blue line)
  plot(b_vec, store.v(2, :, 2), 'Color', 'b', '--', 'LineWidth', 2.5);
  xlabel('Debt levels', 'FontSize', 12);
  ylabel('Value function', 'FontSize', 12);
  title('Value function by time period and endowment levels', 'FontSize', 14);
  % Adding a legend
  legend('High endowment at t=1', 'High endowment at t=2', 'Low endowment at t=1', 'Low endowment at t=2', 'Location', 'best');
  hold off;

saveas(gcf, 'Value_functions_Q2.jpg');
```

Q3-Q4 Main file

```


%--------------------------------------------------------------------------
%
% Title: International Macro-Finance Problem Set 2, Q4
% Author: Rinto Fujimoto
% Date: 25/11/2023
% Description: Sovereign default model with T periods
%
%--------------------------------------------------------------------------


%--------------------------------------------------------------------------
% 0. Housekeeping
%--------------------------------------------------------------------------

close all;
clear all;


%--------------------------------------------------------------------------
% 1. Defining functions
%--------------------------------------------------------------------------

function u = util(c, gamma)
% this is the utility function (CRRA)
  u = c.^(1 - gamma) / (1 - gamma);
end



function [value, default, debt_price, b_vec] = model(time, gamma, sigma, phi_hat)
  % This function performs value function iteration for different
  % parameters.
  % Inputs: time horizon, borrower's risk aversion, lender's risk aversion, cost of default
  % Outputs: value function, default policy function, debt price, grid for b
  % Note: sigma appears in the debt pricing equation


  par.gamma = gamma;
  par.beta = .97;
  par.pi = .5;

  cons.phi_hat = phi_hat;
  cons.yh = 1.1;
  cons.yl = .9;
  cons.rf = par.beta^(-1) - 1;

  t = time;

  maxit_q = 1000;
  maxit_v = 1000;
  err_tol_q = 1e-10;
  err_tol_v = 1e-10;
  lambda_v = .5;
```

```
lambda_q = .5;

Pr = [par.pi, 1 - par.pi;
      par.pi, 1 - par.pi];

qrf = (1 + repmat(cons.rf, [2, 1])).^-(1/4);

n = 50;

b_min = -.02;
b_max = 1.2;

b_vec = linspace(b_min, b_max, n);
y_vec = [cons.yh cons.yl].';

[~,i_b_zero] = min(abs(b_vec));
b_vec(i_b_zero) = 0;

grid.b_state3 = repmat(b_vec, [2, 1, n]);
grid.b_choice2 = repmat(b_vec, [2, 1]);
grid.y_state3 = repmat(y_vec, [1, n, n]);

grid.borr_choice2 = zeros(2, n, t);
grid.borr_choice3 = zeros(2, n, n, t);

guess.v = zeros(2, n, t);
guess.v_new = zeros(2, n, t);

guess.i_b = zeros(2, n, t);

guess.q = qrf(1, 1) * ones(2, n, t);
guess.def = zeros(2, n, t);

store.v = zeros(2, n, t);
store.i_b = zeros(2, n, t);
store.q = qrf(1, 1) * ones(2, n, t);
store.def = zeros(2, n, t);

policy.b = zeros(2, n, t);
policy.def = zeros(2, n, t);

con_choice = zeros(2, n, n, t);
util_choice = zeros(2, n, n, t);
borr_maximand = zeros(2, n, n, t);
e_def = zeros(2, n, t);
e_v = zeros(2, n);
e_v3 = zeros(2, n, n);
e_v_def = zeros(2, n);

discount = zeros(1, t);
for i = 1:t
  discount(:, i) = par.beta * ((1 - par.beta^(i)) / (1 - par.beta));
end
discount = flip(discount, 2);

con_def = [cons.yh - cons.phi_hat, cons.yl].';
con_def3 = repmat(con_def, [1, n, n]);

```

```
e_v_def =  discount.' * Pr(1, :) * util(con_def, par.gamma);

v_def = repmat(util(con_def, par.gamma), [1, n, t]) + permute(repmat(e_v_def, [1, n, 2]), [3, 2, 1]);

err_q = 7;
err_v = 7;
iter_q = 1;
iter_v = 1;



while err_q > err_tol_q && iter_q < maxit_q

  while err_v > err_tol_v && iter_v < maxit_v

    for i = 1:t

      e_v = Pr * guess.v(:, :, i);
      e_v3 = permute(repmat(e_v(:, :), [1, 1, n]), [1, 3, 2]);
      e_v_def = Pr * v_def(:, :, i);

      optimal_choice = max(e_v3(:, :, i), e_v_def);

      grid.borr_choice2(:, :, i) = guess.q(:, :, i) .* grid.b_choice2;
      grid.borr_choice3(:, :, :, i) = repmat(grid.borr_choice2(:, :, i), [1, 1, n]);
      grid.borr_choice3(:, :, :, i) = permute(grid.borr_choice3(:, :, :, i), [1, 3, 2]);

      if i < t
        con_choice(:, :, :, i) = grid.y_state3 - grid.b_state3 + grid.borr_choice3(:, :, :, i);
      else
        con_choice(:, :, :, i) = max(grid.y_state3 - grid.b_state3, con_def3);
      end

      con_choice(con_choice < eps) = eps;

      util_choice(:, :, :, i) = util(con_choice(:, :, :, i), par.gamma);

      borr_maximand(:, :, :, i) = util_choice(:, :, :, i) + par.beta * e_v3(:, :, :);

      store.i_b(:, :, i) = guess.i_b(:, :, i);

      [guess.v_new(:, :, i), guess.i_b(:, :, i)] = max(borr_maximand(:, :, :, i), [], 3); % 2 x n and 2 x n

      guess.i_b(:, :, i) = guess.def(:, :, i) .* ones(2, n) * i_b_zero + (1 - guess.def(:, :, i)) .* guess.i_b(:, :, i);

    end

    store.def = guess.def;

    guess.def = v_def > guess.v_new;

```

```
      guess.v_new = max(v_def, guess.v_new);

      store.v = guess.v;

      guess.v = lambda_v * guess.v_new + (1 - lambda_v) * store.v;

      err_v = max(abs(guess.v(:) - store.v(:)));

      iter_v
      err_v

      iter_v = iter_v + 1;

  end

    for i = 1:t
      e_def(:, :, i) = Pr * guess.def(:, :, i);
    end

    store.q = guess.q;

    % sigma is the lender's risk aversion
    guess.q = qrf(1, 1) * (1 - e_def).^sigma;

    err_q = max(abs(guess.q(:) - store.q(:)));

    guess.q = lambda_q * guess.q + (1 - lambda_q) * store.q;

    iter_q
    err_q

    iter_q = iter_q + 1;

    err_v = 7;
    iter_v = 1;

  end


  policy.b = repmat(grid.b_choice2, [1, 1, t]);
  policy.b = policy.b(store.i_b);
  policy.def = store.def;

  default = policy.def;
  debt_price = store.q;
  value = store.v;

end



%-----------------------------------------------------------------------
% 2. Running functions with different parameters
%-----------------------------------------------------------------------

[value1, ~, ~, b_vec] = model(20, 2, 1.0, .5);
value2 = model(30, 2, 1.0, .5);
value3 = model(40, 2, 1.0, .5);
```

```
value4 = model(50, 2, 1.0, .5);

[~, default1] = model(30, -.5, 1.0, .5);
[~, default2] = model(30, 0, 1.0, .5);
[~, default3] = model(30, 2.0, 1.0, .5);

[~, default4, debt_price1] = model(30, 2, .5, .5);
[~, default5, debt_price2] = model(30, 2, 1.0, .5);
[~, default6, debt_price3] = model(30, 2, 2.0, .5);

[~, default7] = model(30, 2, 1.0, .22);
[~, default8] = model(30, 2, 1.0, .45);
[~, default9] = model(30, 2, 1.0, .66);



%------------------------------------------------------------------------
% 3.Plot the value function and policy functions
%------------------------------------------------------------------------


% Value function at t=1 and t=2 for various time horizons
figure
  subplot(2, 2, 1);
    hold on;
    plot(b_vec, value1(1, :, 1), 'r-', 'LineWidth', 2.5);
    plot(b_vec, value1(2, :, 1), 'b--', 'LineWidth', 2.5);
    xlabel('State debt'), ylabel('Utility');
    title('Value functions for a model with T = 3');
    legend('t = 1', 't = 2');
    hold off;
  subplot(2, 2, 2);
    hold on;
    plot(b_vec, value2(1, :, 1), 'r-', 'LineWidth', 2.5);
    plot(b_vec, value2(2, :, 1), 'b--', 'LineWidth', 2.5);
    xlabel('State debt'), ylabel('Utility');
    title('Value functions for a model with T = 10');
    legend('t = 1', 't = 2');
    hold off;
  subplot(2, 2, 3);
    hold on;
    plot(b_vec, value3(1, :, 1), 'r-', 'LineWidth', 2.5);
    plot(b_vec, value3(2, :, 1), 'b--', 'LineWidth', 2.5);
    xlabel('State debt'), ylabel('Utility');
    title('Value functions for a model with T = 30');
    legend('t = 1', 't = 2');
    hold off;
  subplot(2, 2, 4);
    hold on;
    plot(b_vec, value4(1, :, 1), 'r-', 'LineWidth', 2.5);
    plot(b_vec, value4(2, :, 1), 'b--', 'LineWidth', 2.5);
    xlabel('State debt'), ylabel('Utility');
    title('Value functions for a model with T = 50');
    legend('t = 1', 't = 2');
    hold off;

    saveas(gcf, 'Value_functions_Q3.jpg');

```

```
figure;
  hold on;
  % Default policy function for risk-loving borrower (darker red)
  plot(b_vec, default1(1, :, 1), 'Color', [1 0 0], 'LineWidth', 2.5);
  % Default policy function for risk-neutral borrower
  plot(b_vec, default2(1, :, 1), 'Color', [.75 0 0], '--', 'LineWidth', 2.5);
  % Default policy function for risk-averse borrower (lighter red)
  plot(b_vec, default3(1, :, 1), 'Color', [.5 0 0], '-.', 'LineWidth', 2.5);
  xlabel('State debt', 'FontSize', 12);
  ylabel('Default', 'FontSize', 12)
  % Adding a legend
  legend('\gamma = -0.5', '\gamma = 0', '\gamma = 2.0', 'Location', 'best');
  hold off;

saveas(gcf, 'Default_risk_borrower_Q4.jpg');



figure;
  hold on;
  % Default policy function for risk-loving lender (darker green)
  plot(b_vec, default4(1, :, 1), 'Color', [0 1 0], 'LineWidth', 2.5);
  % Default policy function for risk-neutral lender
  plot(b_vec, default5(1, :, 1), 'Color', [0 .75 0], '--', 'LineWidth', 2.5);
  % Default policy function for risk-averse lender (lighter green)
  plot(b_vec, default6(1, :, 1), 'Color', [0 .5 0], '-.', 'LineWidth', 2.5);
  xlabel('State debt', 'FontSize', 12);
  ylabel('Default', 'FontSize', 12);
  % Adding a legend
  legend('\sigma = .5', '\sigma = 1.0', '\sigma = 2.0', 'Location', 'best');
  hold off;

saveas(gcf, 'Default_risk_lender_Q4.jpg');


figure;
  hold on;
  % Debt price for risk-loving lender (darker yellow)
  plot(b_vec, debt_price1(1, :, 1), 'Color', [1 1 0], 'LineWidth', 2.5);
  % Debt price for risk-neutral lender
  plot(b_vec, debt_price2(1, :, 1), 'Color', [.75 .75 0], '--', 'LineWidth', 2.5);
  % Debt price for risk-averse lender (lighter yellow)
  plot(b_vec, debt_price3(1, :, 1), 'Color', [.5 .5 0], '-.', 'LineWidth', 2.5);
  xlabel('Choice debt', 'FontSize', 12);
  ylabel('Debt Price', 'FontSize', 12);
  % Adding a legend
  legend('\sigma = .5', '\sigma = 1.0', '\sigma = 2.0', 'Location', 'best');
  hold off;

saveas(gcf, 'Debt_price_risk_lender_Q4.jpg');



figure;
  hold on;
  % Default policy function for low default cost (darker blue)
  plot(b_vec, default7(1, :, 1), 'Color', [0 0 1], 'LineWidth', 2.5);
```

```
  % Default policy function for medium default cost
  plot(b_vec, default8(1, :, 1), 'Color', [0 0 .75], '--', 'LineWidth', 2.5);
  % Default policy function for high default cost (lighter blue)
  plot(b_vec, default9(1, :, 1), 'Color', [0 0 .5], '-.', 'LineWidth', 2.5);
  xlabel('State debt', 'FontSize', 12);
  ylabel('Default', 'FontSize', 12);
  % Adding a legend
  legend('\phi = .22', '\phi = .45', '\phi = .66', 'Location', 'best');
  hold off;

saveas(gcf, 'Default_cost_Q4.jpg');
```

IMF, Problem Set 3

1068576

17 December 2023

Q1

The household's problem is:

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

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:

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

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

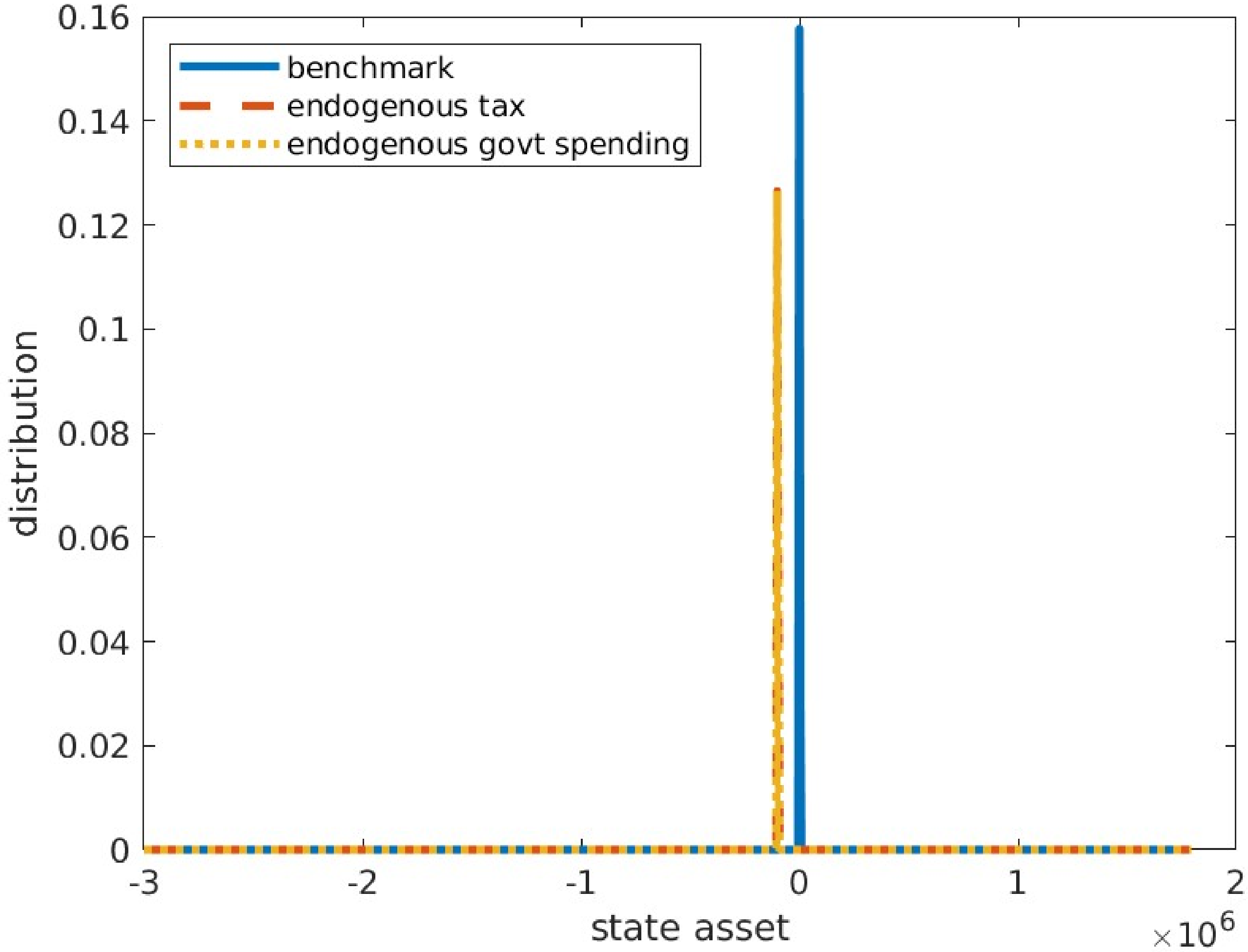


Figure 1: Utility comparative statics

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.

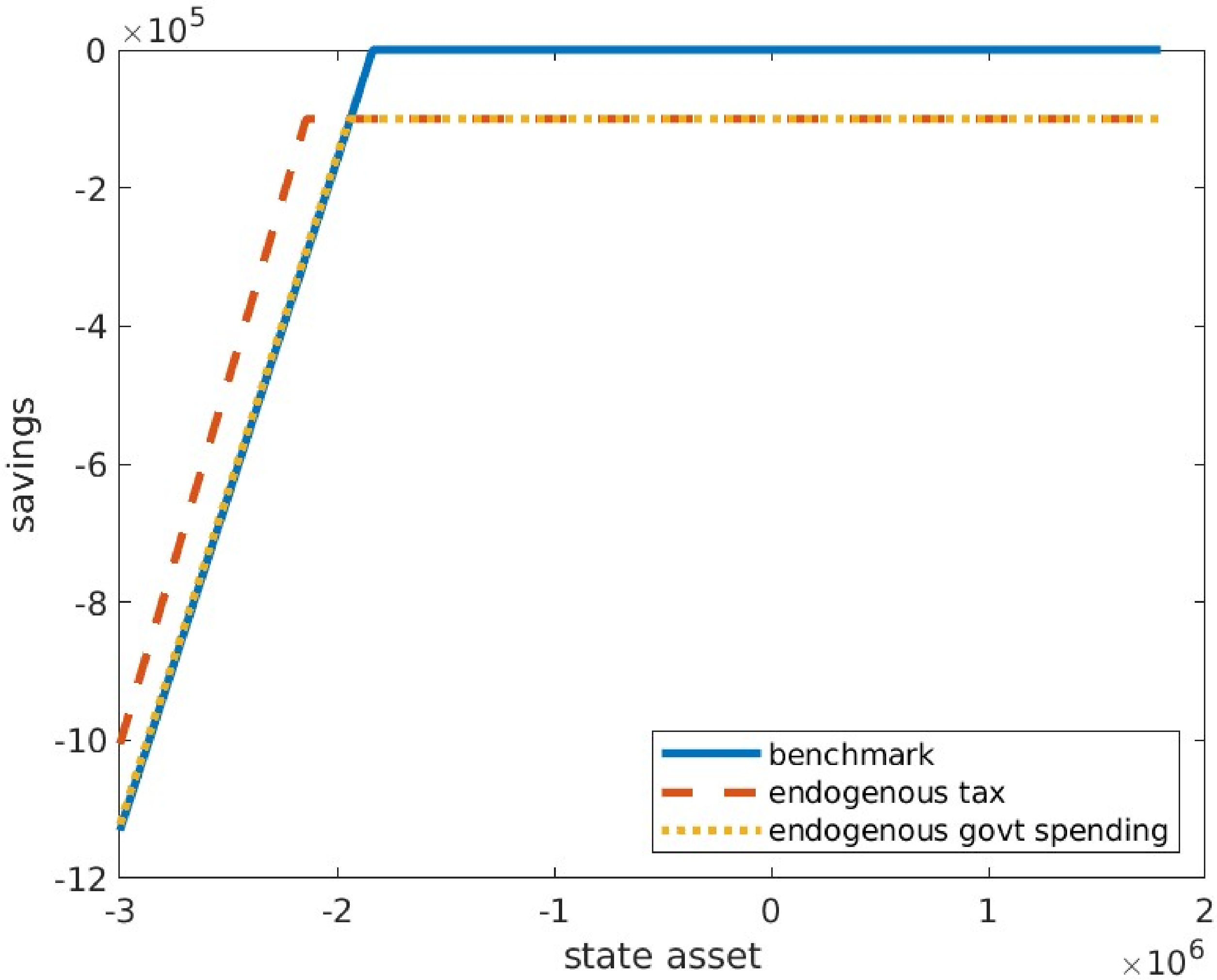


Figure 2: Savings policy function

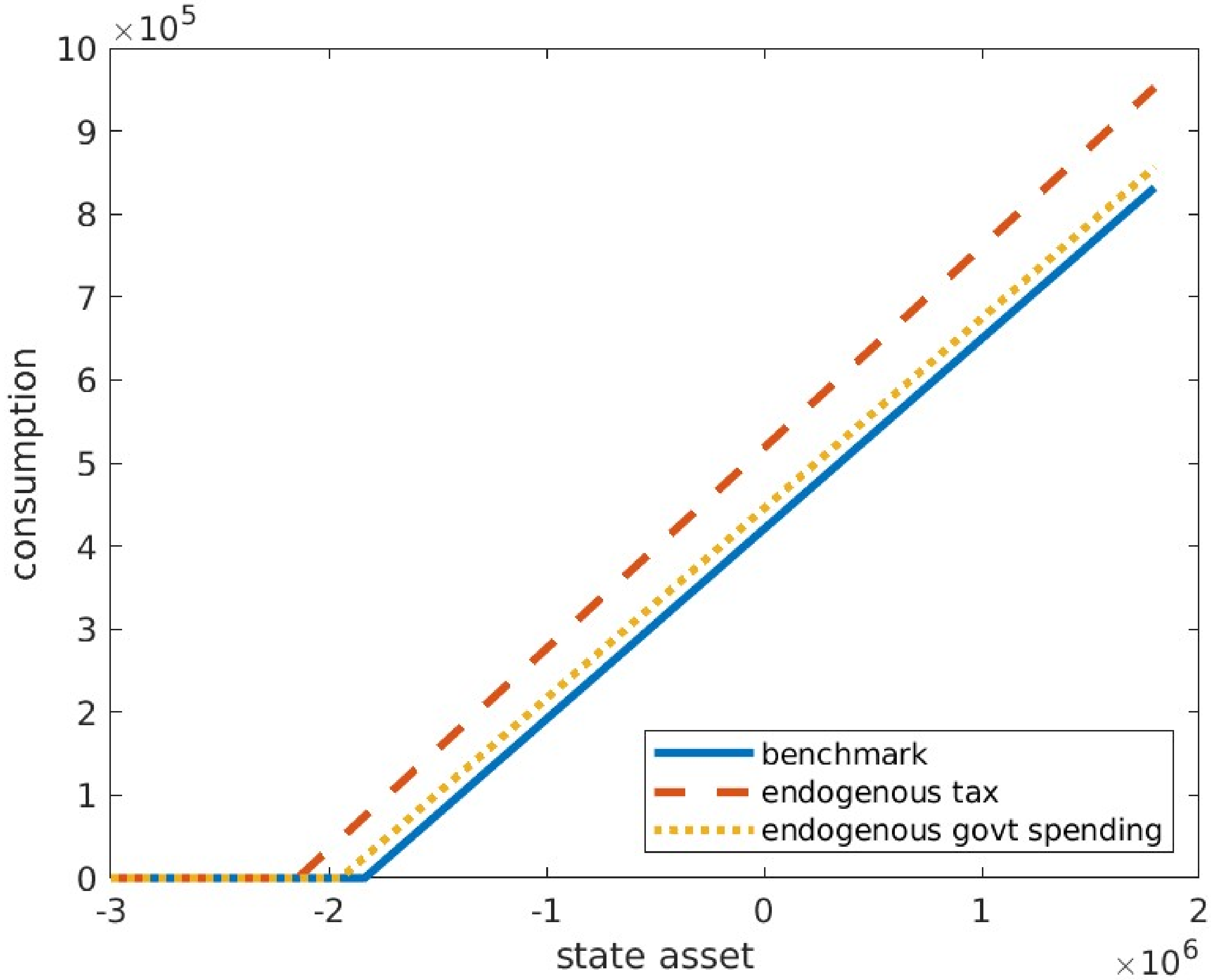


Figure 3: Consumption policy function

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.

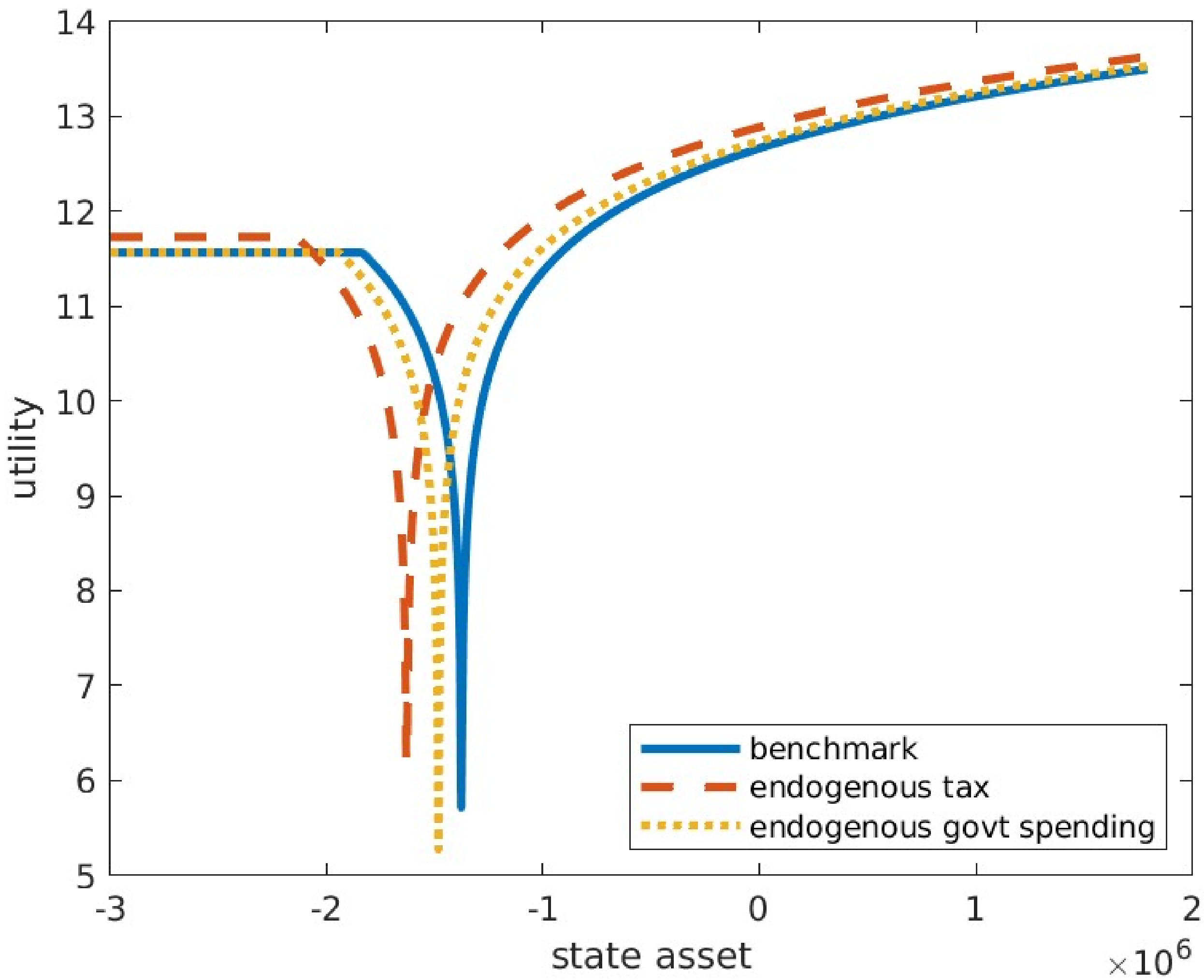


Figure 4: Utility comparative statics

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;
```

```
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
```

```
%---------------------------------------------------------------------

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

```matlab


%--------------------------------------------------------------------------
%
% 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;
```

```matlab

    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);
```

IMF, Problem Set 3

1068576

15 January 2024

Q1

0.1 RAFAFP

The household's problem in Country A is:

$$\max_{\{C_{A,T,t},C_{A,O,t},l_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t \Big(\log(c_{A,t}) + \phi \log(1-l_t) \Big) \tag{1}$$

subject to:

$$\begin{aligned} P_{O,t}c_{A,O,t} + P_{T,t}c_{A,T,t} =& W_t l_t + \Pi_t + E_t(n_{A,t} - \frac{n_{A,t+1}}{R_t^{\$}}) \\ n_{A,t} =& 0 \\ c_{A,Ot}, c_{A,T,t} \geq& \; 0 \\ l_t \in& (0,1) \end{aligned}$$

Similarly, the household's problem in Country B is:

$$\max_{\{C_{B,T,t},C_{B,O,t}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t \log(c_{B,t}) \tag{2}$$

subject to:

$$\begin{aligned} P_{O,t}c^{\$}_{B,O,t} + P^{\$}_{T,t}c_{B,T,t} =& y_{O,t} + (n_{B,t} - \frac{n_{B,t+1}}{R_t^{\$}}) \\ n_{B,t} =& 0 \\ c_{B,O,t}, c_{B,T,t} \geq& \; 0 \end{aligned}$$

where for each household, consumption is aggregated as:

$$c_{j,t} = \left[s_T c_{j,T,t}^{\frac{\eta-1}{\eta}} + (1 - s_T) c_{j,O,t}^{\frac{\eta-1}{\eta}} \right]^{\frac{\eta}{\eta-1}}, \quad j \in \{A, B\}$$

Moreover, it is assumed that the law of one price holds for both oil and tradeable goods, such that:

$$\begin{aligned} P_{T,t} &= E_t P_{T,t}^{\$} = P_{T,t}^{\$} \\ P_{O,t} &= E_t P_{O,t}^{\$} = 1 \end{aligned}$$

since $E_t = 1$ due to the assumption of fixed exchange rate and $P_{O,t} = 1$ due to normalisation.

The households' first order conditions consist of their oil-tradeable Euler equation:

$$(1 - s_T) c_{J,O,t}^{-\frac{1}{\eta}} = \frac{s_T c_{J,T,t}^{-\frac{1}{\eta}}}{P_{T,t}}, \quad j \in \{A, B\} \tag{3}$$

the household's labour supply decision in Country A:

$$\frac{(1 - s_T) c_{A,O,t}^{-\frac{1}{\eta}}}{s_T c_{A,T,t}^{1-\frac{1}{\eta}} + (1 - s_T) c_{A,O,t}^{1-\frac{1}{\eta}}} W_t = \frac{\psi}{1 - l_t} \tag{4}$$

and the budget constraints

$$P_{O,t} c_{A,O,t} + P_{T,t} c_{A,T,t} = W_t l_t + \Pi_t \tag{5}$$

$$P_{O,t} c_{B,O,t}^{\$} + P_{T,t}^{\$} c_{B,T,t} = y_{O,t} \tag{6}$$

Next, the monopolistically competitive firm's problem with flexible price is:

$$\max_{l_{it}} P_{i,t} y_{i,t} - W_t l_{i,t} \tag{7}$$

subject to:

$$
\begin{aligned}
y_{i,t} =& l_{i,t}^{1-\alpha} \\
P_{i,t} =& P_{T,t}\left(\frac{y_{i,t}}{y_{T,t}}\right)^{-\theta}
\end{aligned}
$$

The first order condition is:

$$P_{T,t}(1-\theta)(1-\alpha)l_{i,t}^{(1-\theta)(1-\alpha)-1} = y_{T,t}^{-\theta}W_t \tag{8}$$

Finally, define the oil endowment in Country B as an AR(1) process, such that:

$$y_{O,t} = \gamma + \rho y_{O,t-1} + \epsilon_{O,t} \tag{9}$$

I can now define an equilibrium. An equilibrium in this economy consists of sequences of quantities $\{c_{A,O,t}, c_{A,T,t}, c_{B,O,t}, c_{B,T,t}, l_t, y_{T,t}, y_{O,t}, \Pi_t\}_{t=0}^{\infty}$ and prices $\{P_{T,t}, P_{O,t}, P_{T,t}^{\$}, P_{O,t}^{\$}, W_t, E_t\}_{t=0}^{\infty}$ such that the FOCs 3 to 9 hold and market-clearing is satisfied on the tradeable goods market:

$$c_{A,T,t} + c_{B,T,t} = y_{T,t} \tag{10}$$

the oil market:

$$c_{A,O,t} + c_{B,O,t} = y_{O,t} \tag{11}$$

and the labour market:

$$l_t = \int_0^1 l_{i,t}\, di \tag{12}$$

0.2 RANBFP

The problem is similar to the RAFAFP economy, except households in Country A and B are no longer in financial autarky. Therefore, the households' problems are now:

$$\max_{\{C_{A,T,t},C_{A,O,t},l_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t \Big(\log(c_{A,t}) + \phi \log(1-l_t) \Big) \tag{13}$$

subject to:

$$\begin{aligned} P_{O,t}c_{A,O,t} + P_{T,t}c_{A,T,t} =& W_t l_t + \Pi_t + E_t(n_{A,t} - \frac{n_{A,t+1}}{R_t^{\$}}) \\ c_{A,Ot}, c_{A,T,t} \geq& 0 \\ l_t \in& (0,1) \end{aligned}$$

and:

$$\max_{\{C_{B,T,t},C_{B,O,t}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t \log(c_{B,t}) \tag{14}$$

subject to:

$$\begin{aligned} P_{O,t}c_{B,O,t}^{\$} + P_{T,t}^{\$}c_{B,T,t} =& y_{O,t} + (n_{B,t} - \frac{n_{B,t+1}}{R_t^{\$}}) \\ c_{B,O,t}, c_{B,T,t} \geq& 0 \end{aligned}$$

where $n_{A,t}, n_{B,t} \in (-\infty, \infty)$. The FOCs are identical to RAFAFP with the addition of the intertemporal Euler equations:

$$\frac{c_{J,O,t}^{-\frac{1}{\eta}}}{s_T c_{J,T,t}^{1-\frac{1}{\eta}} + (1-s_T)c_{J,O,t}^{1-\frac{1}{\eta}}} = \beta R_t^{\$} \frac{c_{J,O,t+1}^{-\frac{1}{\eta}}}{s_T c_{J,T,t+1}^{1-\frac{1}{\eta}} + (1-s_T)c_{J,O,t+1}^{1-\frac{1}{\eta}}}, \quad j \in \{A, B\} \tag{15}$$

The equilibrium is now defined as follows. An equilibrium in this economy consists of sequences of quantities $\{c_{A,O,t}, c_{A,T,t}, c_{B,O,t}, c_{B,T,t}, l_t, y_{T,t}, y_{O,t}, \Pi_t, n_{A,t}, n_{B,t}\}_{t=0}^{\infty}$ and prices $\{P_{T,t}, P_{O,t}, P_{T,t}^{\$}, P_{O,t}^{\$}, W_t, E_t, R_t^{\$}\}_{t=0}^{\infty}$ such that the FOCs (3) to (9), (15) hold and market-clearing conditions (10) to (12) are satisfied.

0.3 TANBFP

In the TANBFP economy, there is a mass of χ Ricardian and $(1-\chi)$ hand-to-mouth (HTM) households in country A. Ricardian households' problem is defined as in equations (1) and HTM households' problem is defined as in equations (13). Ricardian households' FOCs are characterised by:

$$(1-s_T)c_{AR,O,t}^{-\frac{1}{\eta}} = \frac{s_T c_{AR,T,t}^{-\frac{1}{\eta}}}{P_{T,t}} \tag{16}$$

$$\frac{c_{AR,O,t}^{-\frac{1}{\eta}}}{s_T c_{AR,T,t}^{1-\frac{1}{\eta}} + (1-s_T)c_{AR,O,t}^{1-\frac{1}{\eta}}} = \beta R_t^{\$} \frac{c_{AR,O,t+1}^{-\frac{1}{\eta}}}{s_T c_{AR,T,t+1}^{1-\frac{1}{\eta}} + (1-s_T)c_{AR,O,t+1}^{1-\frac{1}{\eta}}}, \quad j \in \{A,B\} \tag{17}$$

$$P_{O,t}c_{AR,O,t} + P_{T,t}c_{AR,T,t} = W_t l_{Rt} + \Pi_t + E_t(n_{AR,t} - \frac{n_{AR,t+1}}{R_t^{\$}}) \tag{18}$$

and HTM households' FOCs by:

$$(1-s_T)c_{AH,O,t}^{-\frac{1}{\eta}} = \frac{s_T c_{AH,T,t}^{-\frac{1}{\eta}}}{P_{T,t}} \tag{19}$$

$$\frac{(1-s_T)c_{AH,O,t}^{-\frac{1}{\eta}}}{s_T c_{AH,T,t}^{1-\frac{1}{\eta}} + (1-s_T)c_{AH,O,t}^{1-\frac{1}{\eta}}} W_t = \frac{\psi}{1-l_t} \tag{20}$$

$$P_{O,t}c_{AH,O,t} + P_{T,t}c_{AH,T,t} = W_t l_{Ht} + \Pi_t \tag{21}$$

0.4 TANBNR FER

In the TANBNR fixed exchange rate economy, the firm's FOC becomes:

$$\begin{aligned} W_t = (1-\theta)(1-\alpha)P_{T,t}\frac{l_{i,t}^{(1-\theta)(1-\alpha)-1}}{y_{T,t}^{-\theta}} - \phi\left[\frac{P_{T,t}}{P_{T,t-1}}\frac{l_{i,t}^{-\theta(1-\alpha)}}{y_{T,t}^{-\theta}}\left(\frac{y_{i,t-1}}{y_{T,t-1}}\right)^{\theta} - 1\right]... \\ P_{T,t}Y_{T,t}(-\theta)(1-\alpha)\left[\frac{P_{T,t}}{P_{T,t-1}}\frac{l_{i,t}^{-\theta(1-\alpha)}}{y_{T,t}^{-\theta}}\left(\frac{y_{i,t-1}}{y_{T,t-1}}\right)^{\theta}\right] \end{aligned} \tag{22}$$

0.5 TANBNR PIT

The TANBNR price inflation targeting economy is identical to the TANBNR FER economy, except E_t is no longer fixed and the monetary authority in Country A fixes the price of the tradeable goods, such that $P_{T,t} = P_{T,t-1}$. The oil-tradeable Euler equation of households in Country B becomes:

$$\frac{(1-s_T)c_{B,O,t}^{-\frac{1}{\eta}}}{E_t} = \frac{s_T c_{B,T,t}^{-\frac{1}{\eta}}}{P_{T,t}} \tag{23}$$

The Euler equation for Ricardian households is:

$$\frac{c_{AR,O,t}^{-\frac{1}{\eta}}}{s_T c_{AR,T,t}^{1-\frac{1}{\eta}} + (1-s_T)c_{AR,O,t}^{1-\frac{1}{\eta}}} = \beta R_t^{\$} \frac{E_t}{E_{t+1}} \frac{c_{AR,O,t+1}^{-\frac{1}{\eta}}}{s_T c_{AR,T,t+1}^{1-\frac{1}{\eta}} + (1-s_T)c_{AR,O,t+1}^{1-\frac{1}{\eta}}} \tag{24}$$

and labour supply decision for both households is:

$$\frac{(1-s_T)c_{J,O,t}^{-\frac{1}{\eta}}}{s_T c_{J,T,t}^{1-\frac{1}{\eta}} + (1-s_T)c_{J,O,t}^{1-\frac{1}{\eta}}} \frac{W_t}{E_t} = \frac{\psi}{1-l_t}, \quad J \in \{AH, AR\} \tag{25}$$

0.6 TANBFPNH

The TANBFPNH economy is identical to the TANBFP economy (i.e. no nominal rigidity) except s_T, the weight given to the tradeable goods in the CES aggregator is allowed to vary between types of households, such that preferences are non-homothetic.

Q2

The model is calibrated such that steady-state nominal output of Country A, $P_T^* y_T^*$, equals the nominal GDP of the EU in 2022, which was at 16.75 trillion euros. Moreover, average labour supply, l^*, is targeted to reflect the average work hour in the EU of 37.5 hours per week, resulting in a target value of 0.22. Finally, the model is calibrated to obtain steady-state oil endowment that is equal to the annual GDP of Saudi Arabia in 2022 denominated in euro, since I assume a fixed exchange rate between Country A and B throughout most models and the price of oil is the numeraire. This yields a target of 1 trillion euros for y_O^*.

For the parameters, $\alpha = 0.3$ and $\beta = 0.99$ were chosen to reflect standard values in the macroeconomic literature. The elasticity of labour supply was set to $\psi = 1.5$ following Blundell et al. (2000). The elasticity of substitution between the tradeable good and oil was set at a low value such that $\eta = 0.4$ such that tradeable goods and oil have low levels of substitutability in the consumption basket. The proportion of Ricardian households, χ, was set at 0.8 to reflect the proportion of households that are not credit-constrained in the EU according to the HFCS survey data. I assumed a highly persistent oil endowment process, such that $\rho = 0.9$. Finally, the weight given to the tradeable goods, s_T, as well as the elasticity of substitution between the varieties of tradeables, θ, and the intercept of the AR(1) process for oil endowment γ, were calibrated to attain the target moments described above.

Table 1: Targeted values

	Benchmark	RANBFP	TANBFP	TANBNR FER	TANBNR PIT	TANBFPNH
$P_T Y_T$	16.75	19.9	16.7	13.3	16.75	23.0
l_t	0.22	0.22	0.26	0.23	0.27	0.19
Y_O	1.0	1.0	1.0	1.0	1.0	1.0

Table 2: Calibrated values

	Benchmark	RANBFP	TANBFP	TANBNR FER	TANBNR PIT	TANBFPNH
α	0.3	0.3	0.3	0.3	0.3	0.3
β	0.99	0.99	0.99	0.99	0.99	0.99
ψ	1.5	1.5	1.5	1.5	1.5	1.5
ϕ				10	10	
η	0.4	0.4	0.4	0.4	0.4	0.4
s_T	0.8	0.8	0.8	0.8	0.8	0.8
$s_{AR,T}$						0.81
$s_{AH,T}$						0.75
$s_{B,T}$						0.8
θ	1.0	1.0	1.0	1.0	1.0	1.0
ρ	0.9	0.9	0.9	0.9	0.9	0.9
γ	0.1	0.1	0.1	0.1	0.1	0.1
χ	0.8	0.8	0.8	0.8	0.8	0.8

Q3

The price index in all models is calculated from the CES aggregator, such that:

$$P_{j,t} = \left[s_T P_{j,T,t}^{\frac{\eta-1}{\eta}} + (1 - s_T) P_{j,O,t}^{\frac{\eta-1}{\eta}} \right]^{\frac{\eta}{\eta-1}}, \quad j \in \{A, B\} \tag{26}$$

Therefore, it follows that the price level is identical in Country A and B for all models, except TANBFPNH.

0.7 RAFAFP

Given a 10 percent negative shock to oil endowment, the price level in Country A and B increases due to the increase in the price of oil relative to the price of the tradeable goods. Since the price of tradeable goods relative to oil decreases while output stays constant, consumption in Country A decreases. Country B on the other hand, experiences an increase in consumption due to the increase in the price of oil. In fact, because of the low substitutability of oil and tradeables, the price effect dominates over the volume effect and Country B is able to consume more. This transfer of consumption from Country A to Country B is reflected by an increase in net export for Country A.

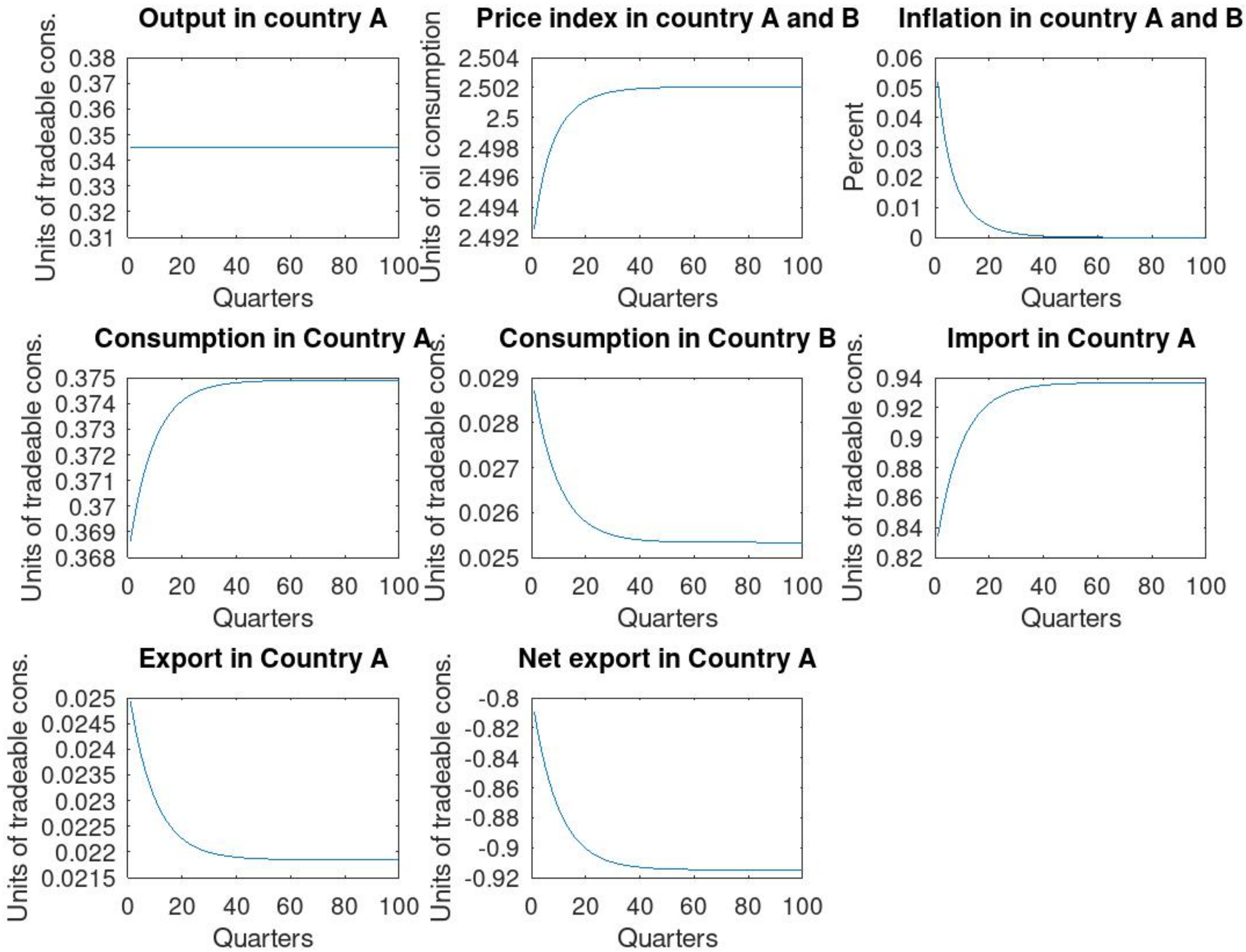


Figure 1: IRFs for RAFAFP

0.8 RANBFP

By opening financial markets, Country A borrows from Country B to smooth consumption. As a result, the shock to consumption in Country A should be less significant than in the RAFAFP model and as a result, the shock to all other variables should be damped compared to RAFAFP (which is indeed the case when we observe inflation).

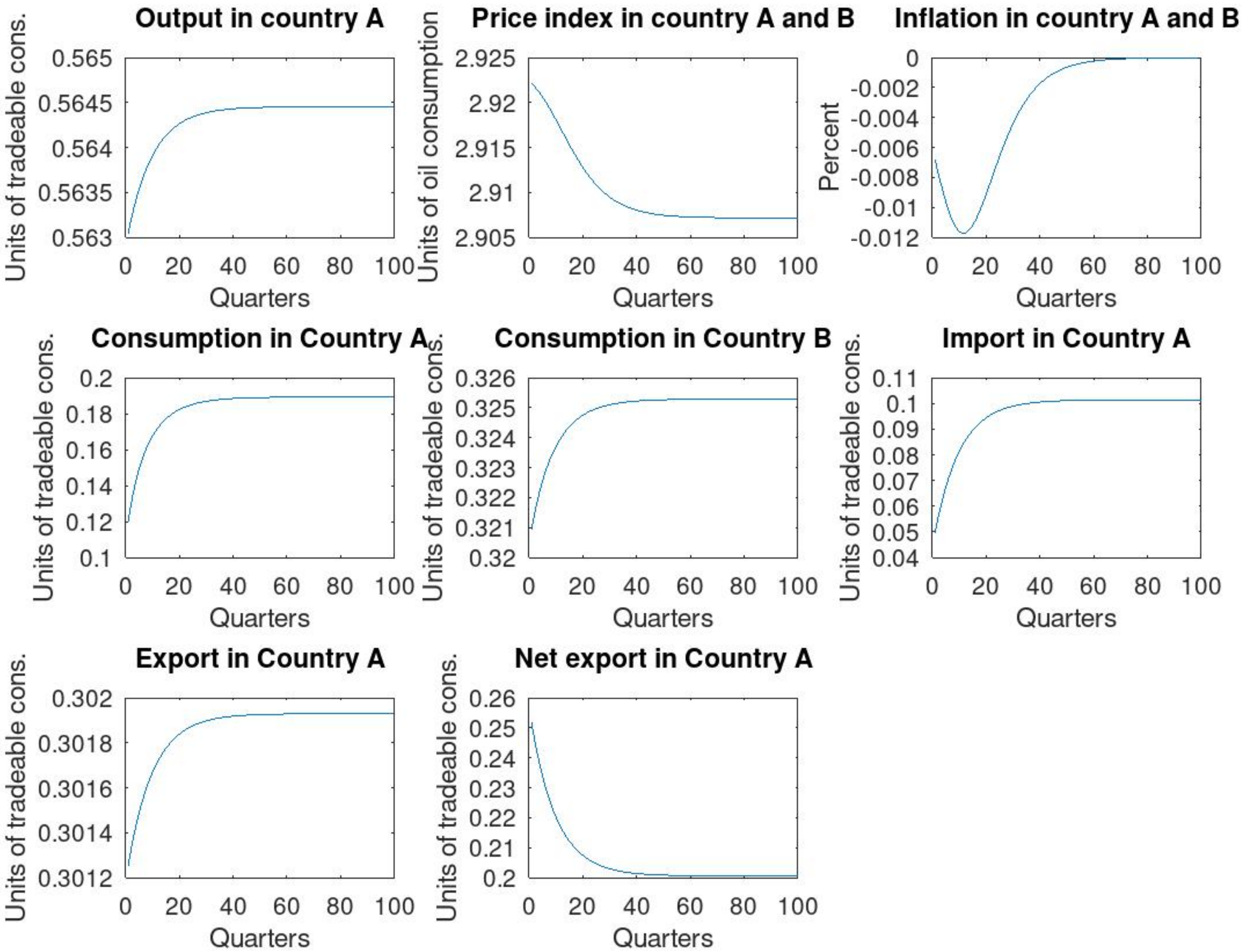


Figure 2: IRFs for RANBFP

0.9 TANBFP

Under TANBFP, I obtain an intermediary outcome between the RAFAFP and RANBFP models in terms of magnitude of the shock, since some of the shock is dampened by the share of Ricardian households.

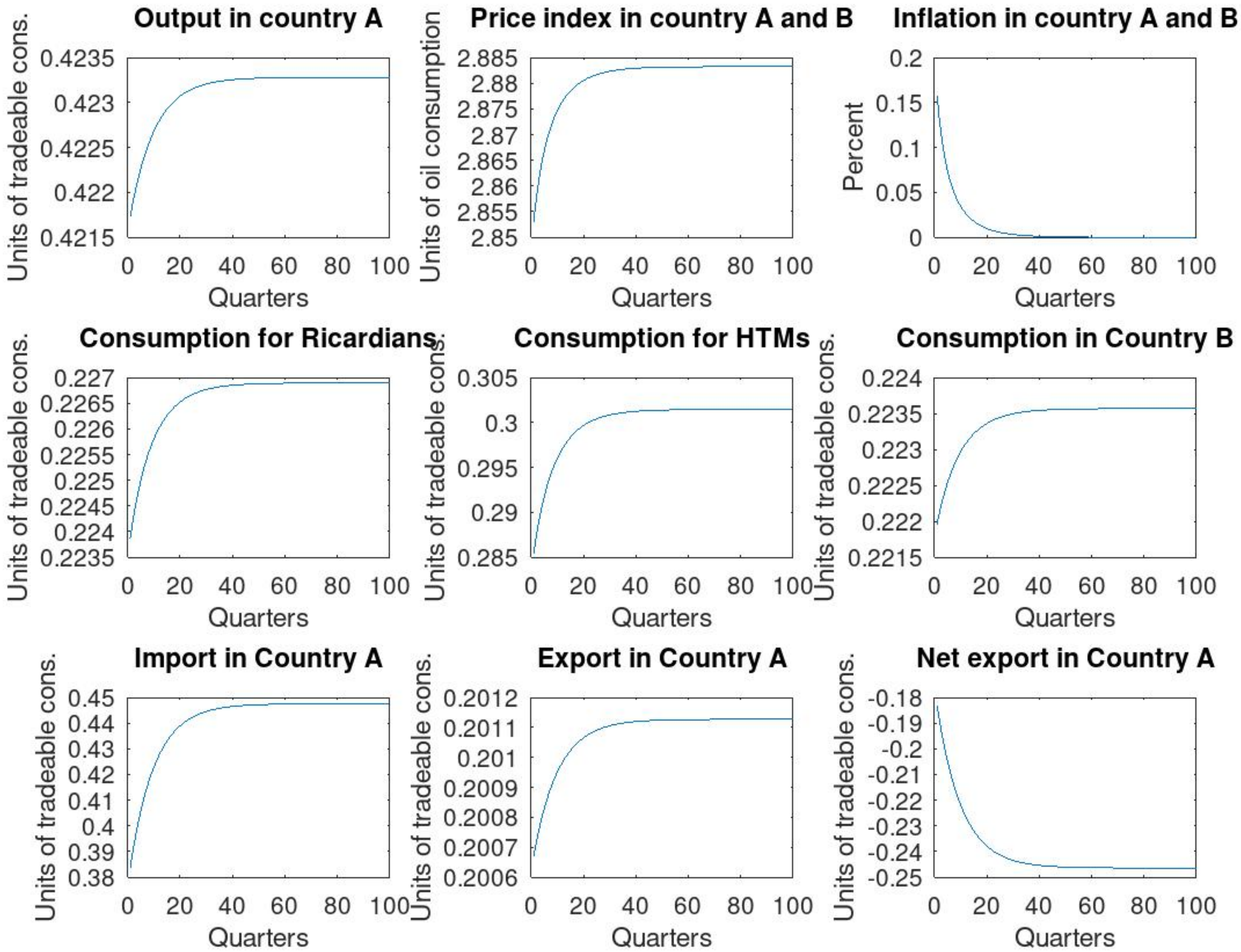


Figure 3: IRFs for TANBFP

0.10 TANBNR FER

The shock is more persistent when nominal rigidity is introduced compared to RAFAFP, as price take more time to adjust to their equilibrium level.

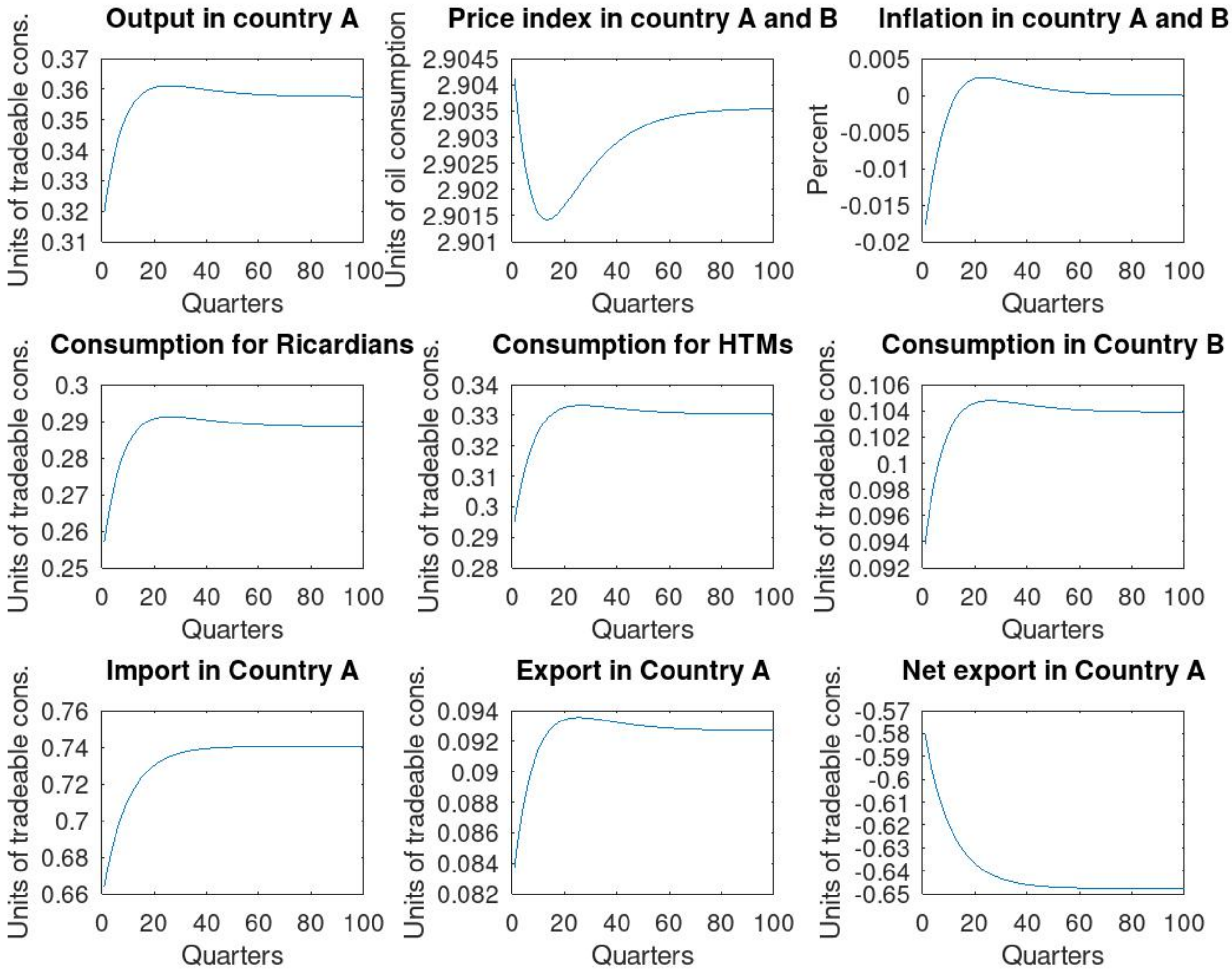


Figure 4: IRFs for TANBNR FER

0.11 TANBNR PIT

In theory, the effects of the foreign shock should be dampened by the exchange ratein Country A, which we do not observe in the IRFs.

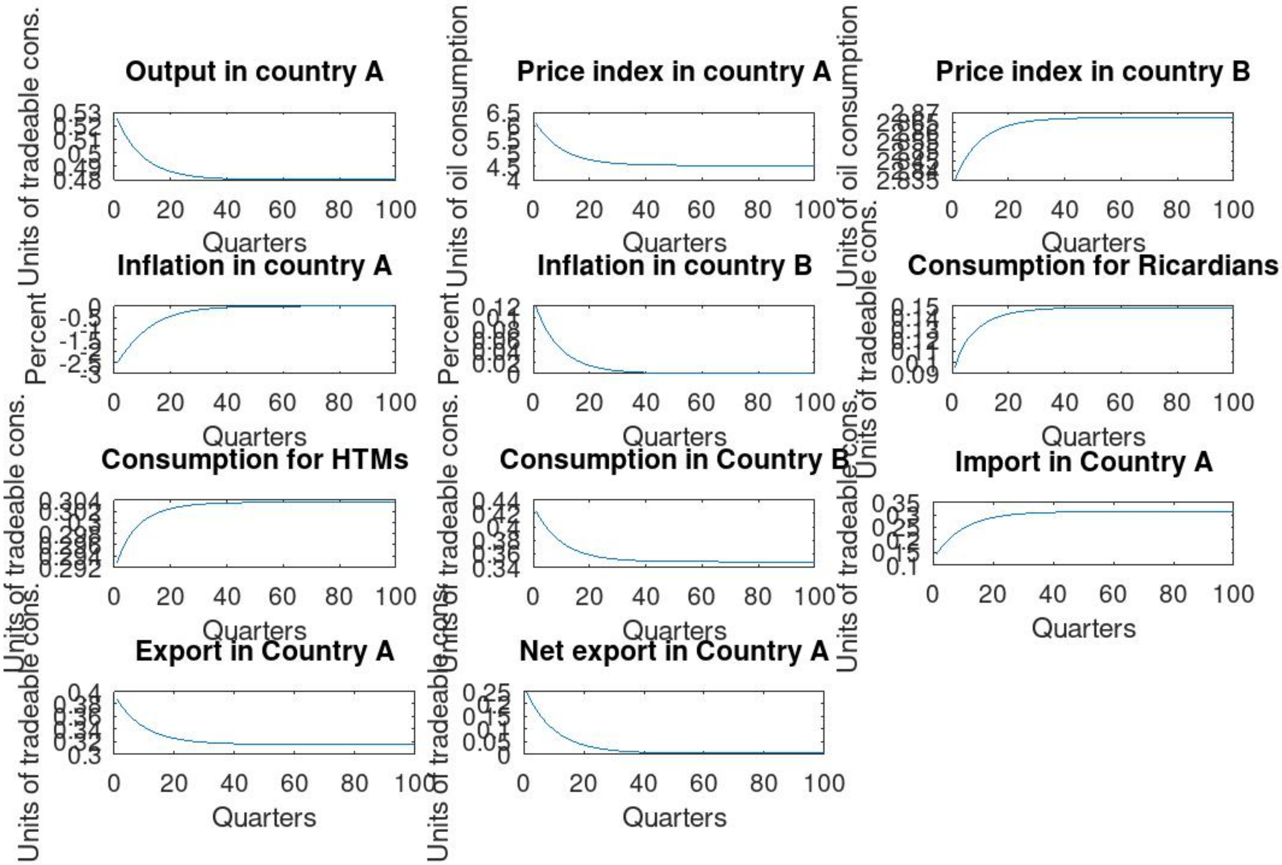


Figure 5: IRFs for TANBNR PIT

0.12 TANBFPNH

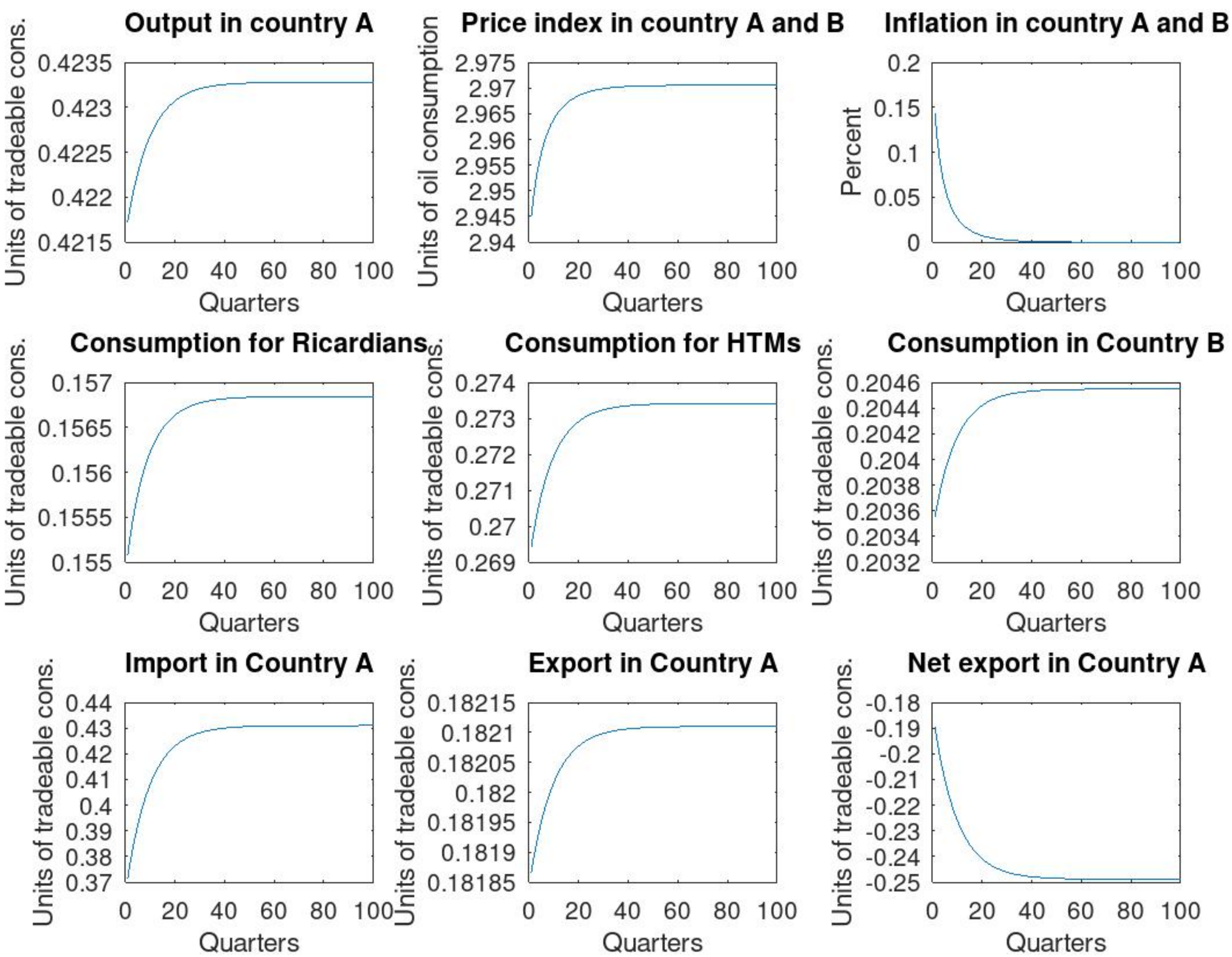


Figure 6: IRFs for TANBFPNH

Appendix A: Main File

```
%--------------------------------------------------------------
%
% Title: International Macro-Finance Problem Set 4, main file
% Author: ---
% Date: 10/01/2024
% Description: Main file for the final problem set
%
%--------------------------------------------------------------


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

close all;
clear all;


%--------------------------------------------------------------
% 1. Load parameters
%--------------------------------------------------------------

parameters;
save parameters par;


%--------------------------------------------------------------
% 1. Run models
%--------------------------------------------------------------

dynare rafafp;
dynare ranbfp;
dynare tanbfp;
dynare tanbnr_fer;
dynare tanbnr_pit;
dynare tanbfpnh;


%--------------------------------------------------------------
% 2. Load results
%--------------------------------------------------------------


rafafp_results = load("rafafp_results.mat");

RAFAFP.c_aoss = rafafp_results.oo_.steady_state(1);
RAFAFP.c_boss = rafafp_results.oo_.steady_state(2);
RAFAFP.c_atss = rafafp_results.oo_.steady_state(3);
RAFAFP.c_btss = rafafp_results.oo_.steady_state(4);
RAFAFP.lss = rafafp_results.oo_.steady_state(5);
RAFAFP.y_tss = rafafp_results.oo_.steady_state(6);
RAFAFP.y_oss = rafafp_results.oo_.steady_state(7);
RAFAFP.p_tss = rafafp_results.oo_.steady_state(8);
RAFAFP.wss = rafafp_results.oo_.steady_state(9);
RAFAFP.piss = rafafp_results.oo_.steady_state(10);

RAFAFP.c_ao = rafafp_results.oo_.irfs.c_ao_err_yo;
RAFAFP.c_at = rafafp_results.oo_.irfs.c_at_err_yo;
RAFAFP.c_bo = rafafp_results.oo_.irfs.c_bo_err_yo;
RAFAFP.c_bt = rafafp_results.oo_.irfs.c_bt_err_yo;
RAFAFP.l    = rafafp_results.oo_.irfs.l_err_yo;
RAFAFP.p_t  = rafafp_results.oo_.irfs.p_t_err_yo;
RAFAFP.pi   = rafafp_results.oo_.irfs.pi_err_yo;
RAFAFP.w    = rafafp_results.oo_.irfs.w_err_yo;
RAFAFP.y_o  = rafafp_results.oo_.irfs.y_o_err_yo;
RAFAFP.y_t  = rafafp_results.oo_.irfs.y_t_err_yo;

psi = rafafp_results.M_.params(2);
RAFAFP.eta = rafafp_results.M_.params(3);
RAFAFP.s_t = rafafp_results.M_.params(4);

```

```
ranbfp_results = load("ranbfp_results.mat");

RANBFP.c_aoss = ranbfp_results.oo_.steady_state(1);
RANBFP.c_boss = ranbfp_results.oo_.steady_state(2);
RANBFP.c_atss = ranbfp_results.oo_.steady_state(3);
RANBFP.c_btss = ranbfp_results.oo_.steady_state(4);
RANBFP.lss = ranbfp_results.oo_.steady_state(5);
RANBFP.y_tss = ranbfp_results.oo_.steady_state(6);
RANBFP.y_oss = ranbfp_results.oo_.steady_state(7);
RANBFP.p_tss = ranbfp_results.oo_.steady_state(8);
RANBFP.wss = ranbfp_results.oo_.steady_state(9);
RANBFP.piss = ranbfp_results.oo_.steady_state(10);
RANBFP.n_ass = ranbfp_results.oo_.steady_state(11);
RANBFP.n_bss = ranbfp_results.oo_.steady_state(12);
RANBFP.r_starss = ranbfp_results.oo_.steady_state(13);

RANBFP.c_ao = ranbfp_results.oo_.irfs.c_ao_err_yo;
RANBFP.c_at = ranbfp_results.oo_.irfs.c_at_err_yo;
RANBFP.c_bo = ranbfp_results.oo_.irfs.c_bo_err_yo;
RANBFP.c_bt = ranbfp_results.oo_.irfs.c_bt_err_yo;
RANBFP.l    = ranbfp_results.oo_.irfs.l_err_yo;
RANBFP.p_t  = ranbfp_results.oo_.irfs.p_t_err_yo;
RANBFP.pi   = ranbfp_results.oo_.irfs.pi_err_yo;
RANBFP.w    = ranbfp_results.oo_.irfs.w_err_yo;
RANBFP.y_o  = ranbfp_results.oo_.irfs.y_o_err_yo;
RANBFP.y_t  = ranbfp_results.oo_.irfs.y_t_err_yo;

beta = ranbfp_results.M_.params(2);
RANBFP.eta = ranbfp_results.M_.params(4);
RANBFP.s_t = ranbfp_results.M_.params(5);


tanbfp_results = load("tanbfp_results.mat");

TANBFP.c_ahoss = ranbfp_results.oo_.steady_state(1);
TANBFP.c_aross = tanbfp_results.oo_.steady_state(2);
TANBFP.c_boss = tanbfp_results.oo_.steady_state(3);
TANBFP.c_ahtss = tanbfp_results.oo_.steady_state(4);
TANBFP.c_artss = tanbfp_results.oo_.steady_state(5);
TANBFP.c_btss = tanbfp_results.oo_.steady_state(6);
TANBFP.l_hss = tanbfp_results.oo_.steady_state(7);
TANBFP.l_rss = tanbfp_results.oo_.steady_state(8);
TANBFP.lss = tanbfp_results.oo_.steady_state(9);
TANBFP.y_tss = tanbfp_results.oo_.steady_state(10);
TANBFP.y_oss = tanbfp_results.oo_.steady_state(11);
TANBFP.p_tss = tanbfp_results.oo_.steady_state(12);
TANBFP.wss = tanbfp_results.oo_.steady_state(13);
TANBFP.piss = tanbfp_results.oo_.steady_state(14);
TANBFP.n_ass = tanbfp_results.oo_.steady_state(15);
TANBFP.n_bss = tanbfp_results.oo_.steady_state(16);
TANBFP.rstarss = tanbfp_results.oo_.steady_state(17);
TANBFP.pirss = tanbfp_results.oo_.steady_state(18);
TANBFP.pihss = tanbfp_results.oo_.steady_state(19);

TANBFP.c_aro = tanbfp_results.oo_.irfs.c_aro_err_yo;
TANBFP.c_art = tanbfp_results.oo_.irfs.c_art_err_yo;
TANBFP.c_aho = tanbfp_results.oo_.irfs.c_aho_err_yo;
TANBFP.c_aht = tanbfp_results.oo_.irfs.c_aht_err_yo;
TANBFP.c_bo  = tanbfp_results.oo_.irfs.c_bo_err_yo;
TANBFP.c_bt  = tanbfp_results.oo_.irfs.c_bt_err_yo;
TANBFP.l_r   = tanbfp_results.oo_.irfs.l_r_err_yo;
TANBFP.l_h   = tanbfp_results.oo_.irfs.l_h_err_yo;
TANBFP.l   = tanbfp_results.oo_.irfs.l_err_yo;
TANBFP.p_t   = tanbfp_results.oo_.irfs.p_t_err_yo;
TANBFP.pi    = tanbfp_results.oo_.irfs.pi_err_yo;
TANBFP.w     = tanbfp_results.oo_.irfs.w_err_yo;
TANBFP.y_o   = tanbfp_results.oo_.irfs.y_o_err_yo;
TANBFP.y_t   = tanbfp_results.oo_.irfs.y_t_err_yo;

TANBFP.eta = tanbfp_results.M_.params(4);
TANBFP.s_t = tanbfp_results.M_.params(5);
chi = tanbfp_results.M_.params(9);
```

```
tanbnr_fer_results = load("tanbnr_fer_results.mat");

TANBNR_FER.c_ahoss = tanbnr_fer_results.oo_.steady_state(1);
TANBNR_FER.c_aross = tanbnr_fer_results.oo_.steady_state(2);
TANBNR_FER.c_boss = tanbnr_fer_results.oo_.steady_state(3);
TANBNR_FER.c_ahtss = tanbnr_fer_results.oo_.steady_state(4);
TANBNR_FER.c_artss = tanbnr_fer_results.oo_.steady_state(5);
TANBNR_FER.c_btss = tanbnr_fer_results.oo_.steady_state(6);
TANBNR_FER.l_hss = tanbnr_fer_results.oo_.steady_state(7);
TANBNR_FER.l_rss = tanbnr_fer_results.oo_.steady_state(8);
TANBNR_FER.lss = tanbnr_fer_results.oo_.steady_state(9);
TANBNR_FER.y_tss = tanbnr_fer_results.oo_.steady_state(10);
TANBNR_FER.y_oss = tanbnr_fer_results.oo_.steady_state(11);
TANBNR_FER.p_tss = tanbnr_fer_results.oo_.steady_state(12);
TANBNR_FER.wss = tanbnr_fer_results.oo_.steady_state(13);
TANBNR_FER.piss = tanbnr_fer_results.oo_.steady_state(14);
TANBNR_FER.n_ass = tanbnr_fer_results.oo_.steady_state(15);
TANBNR_FER.n_bss = tanbnr_fer_results.oo_.steady_state(16);
TANBNR_FER.r_starss = tanbnr_fer_results.oo_.steady_state(17);
TANBNR_FER.pirss = tanbnr_fer_results.oo_.steady_state(18);
TANBNR_FER.pihss = tanbnr_fer_results.oo_.steady_state(19);

TANBNR_FER.c_aro = tanbnr_fer_results.oo_.irfs.c_aro_err_yo;
TANBNR_FER.c_art = tanbnr_fer_results.oo_.irfs.c_art_err_yo;
TANBNR_FER.c_aho = tanbnr_fer_results.oo_.irfs.c_aho_err_yo;
TANBNR_FER.c_aht = tanbnr_fer_results.oo_.irfs.c_aht_err_yo;
TANBNR_FER.c_bo  = tanbnr_fer_results.oo_.irfs.c_bo_err_yo;
TANBNR_FER.c_bt  = tanbnr_fer_results.oo_.irfs.c_bt_err_yo;
TANBNR_FER.l_r   = tanbnr_fer_results.oo_.irfs.l_r_err_yo;
TANBNR_FER.l_h   = tanbnr_fer_results.oo_.irfs.l_h_err_yo;
TANBNR_FER.l   = tanbnr_fer_results.oo_.irfs.l_err_yo;
TANBNR_FER.p_t   = tanbnr_fer_results.oo_.irfs.p_t_err_yo;
TANBNR_FER.pi    = tanbnr_fer_results.oo_.irfs.pi_err_yo;
TANBNR_FER.w     = tanbnr_fer_results.oo_.irfs.w_err_yo;
TANBNR_FER.y_o   = tanbnr_fer_results.oo_.irfs.y_o_err_yo;
TANBNR_FER.y_t   = tanbnr_fer_results.oo_.irfs.y_t_err_yo;

TANBNR_FER.eta = tanbnr_fer_results.M_.params(5);
TANBNR_FER.s_t = tanbnr_fer_results.M_.params(6);


tanbnr_pit_results = load("tanbnr_pit_results.mat");

TANBNR_PIT.c_ahoss = tanbnr_pit_results.oo_.steady_state(1);
TANBNR_PIT.c_aross = tanbnr_pit_results.oo_.steady_state(2);
TANBNR_PIT.c_boss = tanbnr_pit_results.oo_.steady_state(3);
TANBNR_PIT.c_ahtss = tanbnr_pit_results.oo_.steady_state(4);
TANBNR_PIT.c_artss = tanbnr_pit_results.oo_.steady_state(5);
TANBNR_PIT.c_btss = tanbnr_pit_results.oo_.steady_state(6);
TANBNR_PIT.l_hss = tanbnr_pit_results.oo_.steady_state(7);
TANBNR_PIT.l_rss = tanbnr_pit_results.oo_.steady_state(8);
TANBNR_PIT.lss = tanbnr_pit_results.oo_.steady_state(9);
TANBNR_PIT.y_tss = tanbnr_pit_results.oo_.steady_state(10);
TANBNR_PIT.y_oss = tanbnr_pit_results.oo_.steady_state(11);
TANBNR_PIT.p_tss = tanbnr_pit_results.oo_.steady_state(12);
TANBNR_PIT.ess = tanbnr_pit_results.oo_.steady_state(13);
TANBNR_PIT.wss = tanbnr_pit_results.oo_.steady_state(14);
TANBNR_PIT.piss = tanbnr_pit_results.oo_.steady_state(15);
TANBNR_PIT.n_ass = tanbnr_pit_results.oo_.steady_state(16);
TANBNR_PIT.n_bss = tanbnr_pit_results.oo_.steady_state(17);
TANBNR_PIT.r_starss = tanbnr_pit_results.oo_.steady_state(18);
TANBNR_PIT.pirss = tanbnr_pit_results.oo_.steady_state(19);
TANBNR_PIT.pihss = tanbnr_pit_results.oo_.steady_state(20);

TANBNR_PIT.c_aro = tanbnr_pit_results.oo_.irfs.c_aro_err_yo;
TANBNR_PIT.c_art = tanbnr_pit_results.oo_.irfs.c_art_err_yo;
TANBNR_PIT.c_aho = tanbnr_pit_results.oo_.irfs.c_aho_err_yo;
TANBNR_PIT.c_aht = tanbnr_pit_results.oo_.irfs.c_aht_err_yo;
TANBNR_PIT.c_bo  = tanbnr_pit_results.oo_.irfs.c_bo_err_yo;
TANBNR_PIT.c_bt  = tanbnr_pit_results.oo_.irfs.c_bt_err_yo;
TANBNR_PIT.l_r   = tanbnr_pit_results.oo_.irfs.l_r_err_yo;
TANBNR_PIT.l_h   = tanbnr_pit_results.oo_.irfs.l_h_err_yo;
TANBNR_PIT.l   = tanbnr_pit_results.oo_.irfs.l_err_yo;
TANBNR_PIT.p_t   = tanbnr_pit_results.oo_.irfs.p_t_err_yo;
```

```
TANBNR_PIT.e     = tanbnr_pit_results.oo_.irfs.e_err_yo;
TANBNR_PIT.pi    = tanbnr_pit_results.oo_.irfs.pi_err_yo;
TANBNR_PIT.w     = tanbnr_pit_results.oo_.irfs.w_err_yo;
TANBNR_PIT.y_o   = tanbnr_pit_results.oo_.irfs.y_o_err_yo;
TANBNR_PIT.y_t   = tanbnr_pit_results.oo_.irfs.y_t_err_yo;

TANBNR_PIT.eta = tanbnr_pit_results.M_.params(5);
TANBNR_PIT.s_t = tanbnr_pit_results.M_.params(6);


tanbfpnh_results = load("tanbfpnh_results.mat");

TANBFPNH.c_ahoss = tanbfpnh_results.oo_.steady_state(1);
TANBFPNH.c_aross = tanbfpnh_results.oo_.steady_state(2);
TANBFPNH.c_boss = tanbfpnh_results.oo_.steady_state(3);
TANBFPNH.c_ahtss = tanbfpnh_results.oo_.steady_state(4);
TANBFPNH.c_artss = tanbfpnh_results.oo_.steady_state(5);
TANBFPNH.c_btss = tanbfpnh_results.oo_.steady_state(6);
TANBFPNH.l_hss = tanbfpnh_results.oo_.steady_state(7);
TANBFPNH.l_rss = tanbfpnh_results.oo_.steady_state(8);
TANBFPNH.lss = tanbfpnh_results.oo_.steady_state(9);
TANBFPNH.y_tss = tanbfpnh_results.oo_.steady_state(10);
TANBFPNH.y_oss = tanbfpnh_results.oo_.steady_state(11);
TANBFPNH.p_tss = tanbfpnh_results.oo_.steady_state(12);
TANBFPNH.wss = tanbfpnh_results.oo_.steady_state(13);
TANBFPNH.piss = tanbfpnh_results.oo_.steady_state(14);
TANBFPNH.n_ass = tanbfpnh_results.oo_.steady_state(15);
TANBFPNH.n_bss = tanbfpnh_results.oo_.steady_state(16);
TANBFPNH.rstarss = tanbfpnh_results.oo_.steady_state(17);
TANBFPNH.pirss = tanbfpnh_results.oo_.steady_state(18);
TANBFPNH.pihss = tanbfpnh_results.oo_.steady_state(19);

TANBFPNH.c_aro = tanbfpnh_results.oo_.irfs.c_aro_err_yo;
TANBFPNH.c_art = tanbfpnh_results.oo_.irfs.c_art_err_yo;
TANBFPNH.c_aho = tanbfpnh_results.oo_.irfs.c_aho_err_yo;
TANBFPNH.c_aht = tanbfpnh_results.oo_.irfs.c_aht_err_yo;
TANBFPNH.c_bo  = tanbfpnh_results.oo_.irfs.c_bo_err_yo;
TANBFPNH.c_bt  = tanbfpnh_results.oo_.irfs.c_bt_err_yo;
TANBFPNH.l_r   = tanbfpnh_results.oo_.irfs.l_r_err_yo;
TANBFPNH.l_h   = tanbfpnh_results.oo_.irfs.l_h_err_yo;
TANBFPNH.l   = tanbfpnh_results.oo_.irfs.l_err_yo;
TANBFPNH.p_t   = tanbfpnh_results.oo_.irfs.p_t_err_yo;
TANBFPNH.pi    = tanbfpnh_results.oo_.irfs.pi_err_yo;
TANBFPNH.w     = tanbfpnh_results.oo_.irfs.w_err_yo;
TANBFPNH.y_o   = tanbfpnh_results.oo_.irfs.y_o_err_yo;
TANBFPNH.y_t   = tanbfpnh_results.oo_.irfs.y_t_err_yo;

TANBFPNH.eta = tanbfpnh_results.M_.params(4);
TANBFPNH.s_art = tanbfpnh_results.M_.params(5);
TANBFPNH.s_aht = tanbfpnh_results.M_.params(6);
TANBFPNH.s_bt = tanbfpnh_results.M_.params(7);

TANBFPNH.s_t = chi * TANBFPNH.s_art + (1 - chi) * TANBFPNH.s_aht;


%---------------------------------------------------------------------
% 4. Function definition
%---------------------------------------------------------------------

% CES aggregator:

function cons = ces_agg(c_t, c_o, eta, s_t)

  cons = (s_t * c_t.^(1 - 1 / eta) +
                    (1 - s_t) * c_o.^(1 - 1 / eta)).^(eta / (eta - 1));

endfunction



%---------------------------------------------------------------------
% 3. Create key variables
%---------------------------------------------------------------------

```

```
% output in country A

RAFAFP.output = RAFAFP.y_t + RAFAFP.y_tss;
RANBFP.output = RANBFP.y_t + RANBFP.y_tss;
TANBFP.output = TANBFP.y_t + TANBFP.y_tss;
TANBNR_FER.output = TANBNR_FER.y_t + TANBNR_FER.y_tss;
TANBNR_PIT.output = TANBNR_PIT.y_t + TANBNR_PIT.y_tss;
TANBFPNH.output = TANBFPNH.y_t + TANBFPNH.y_tss;


% price levels

% re-express eveything as levels
RAFAFP.p_t = RAFAFP.p_t + RAFAFP.p_tss;
RAFAFP.p_o = 1;

% create a price index
RAFAFP.price = ces_agg(RAFAFP.p_t, RAFAFP.p_o, RAFAFP.eta, RAFAFP.s_t);


RANBFP.p_t = RANBFP.p_t + RANBFP.p_tss;
RANBFP.p_o = 1;

RANBFP.price = ces_agg(RANBFP.p_t, RANBFP.p_o, RANBFP.eta, RANBFP.s_t);


TANBFP.p_t = TANBFP.p_t + TANBFP.p_tss;
TANBFP.p_o = 1;

TANBFP.price = ces_agg(TANBFP.p_t, TANBFP.p_o, TANBFP.eta, TANBFP.s_t);


TANBNR_FER.p_t = TANBNR_FER.p_t + TANBNR_FER.p_tss;
TANBNR_FER.p_o = 1;

TANBNR_FER.price = ces_agg(TANBNR_FER.p_t, TANBNR_FER.p_o, TANBNR_FER.eta, TANBNR_FER.s_t);


TANBNR_PIT.p_at = TANBNR_PIT.p_tss;
TANBNR_PIT.p_ao = TANBNR_PIT.e + TANBNR_PIT.ess;

TANBNR_PIT.price_a = ces_agg(TANBNR_PIT.p_at, TANBNR_PIT.p_ao, TANBNR_PIT.eta, TANBNR_PIT.s_t);


TANBNR_PIT.p_bt = TANBNR_PIT.p_tss ./ (TANBNR_PIT.e + TANBNR_PIT.ess);
TANBNR_PIT.p_bo = 1;

TANBNR_PIT.price_b = ces_agg(TANBNR_PIT.p_bt, TANBNR_PIT.p_bo, TANBNR_PIT.eta, TANBNR_PIT.s_t);


TANBFPNH.p_t = TANBFPNH.p_t + TANBFPNH.p_tss;
TANBFPNH.p_o = 1;

TANBFPNH.price = ces_agg(TANBFPNH.p_t, TANBFPNH.p_o, TANBFPNH.eta, TANBFPNH.s_t);



% Inflation
RAFAFP.infl = diff(RAFAFP.price) ./ RAFAFP.price(1:(end-1)) * 100;
RANBFP.infl = diff(RANBFP.price) ./ RANBFP.price(1:(end-1)) * 100;
TANBFP.infl = diff(TANBFP.price) ./ TANBFP.price(1:(end-1)) * 100;
TANBNR_FER.infl = diff(TANBNR_FER.price) ./ TANBNR_FER.price(1:(end-1)) * 100;
TANBNR_PIT.infl_a = diff(TANBNR_PIT.price_a) ./ TANBNR_PIT.price_a(1:(end-1)) * 100;
TANBNR_PIT.infl_b = diff(TANBNR_PIT.price_b) ./ TANBNR_PIT.price_b(1:(end-1)) * 100;
TANBFPNH.infl = diff(TANBFPNH.price) ./ TANBFPNH.price(1:(end-1)) * 100;


% consumption

% re-express eveything as levels
RAFAFP.c_at = RAFAFP.c_at + RAFAFP.c_atss;
RAFAFP.c_ao = RAFAFP.c_ao + RAFAFP.c_aoss;
RAFAFP.c_bt = RAFAFP.c_bt + RAFAFP.c_btss;
RAFAFP.c_bo = RAFAFP.c_bo + RAFAFP.c_boss;
```

```
% rule out negative consumption
RAFAFP.c_at = max(RAFAFP.c_at, eps);
RAFAFP.c_ao = max(RAFAFP.c_ao, eps);
RAFAFP.c_bt = max(RAFAFP.c_bt, eps);
RAFAFP.c_bo = max(RAFAFP.c_bo, eps);

% compute consumption basket
RAFAFP.c_a = ces_agg(RAFAFP.c_at, RAFAFP.c_ao, RAFAFP.eta, RAFAFP.s_t);
RAFAFP.c_b = ces_agg(RAFAFP.c_bt, RAFAFP.c_bo, RAFAFP.eta, RAFAFP.s_t);


RANBFP.c_at = RANBFP.c_at + RANBFP.c_atss;
RANBFP.c_ao = RANBFP.c_ao + RANBFP.c_aoss;
RANBFP.c_bt = RANBFP.c_bt + RANBFP.c_btss;
RANBFP.c_bo = RANBFP.c_bo + RANBFP.c_boss;

RANBFP.c_at = max(RANBFP.c_at, eps);
RANBFP.c_ao = max(RANBFP.c_ao, eps);
RANBFP.c_bt = max(RANBFP.c_bt, eps);
RANBFP.c_bo = max(RANBFP.c_bo, eps);

RANBFP.c_a = ces_agg(RANBFP.c_at, RANBFP.c_ao, RANBFP.eta, RANBFP.s_t);
RANBFP.c_b = ces_agg(RANBFP.c_bt, RANBFP.c_bo, RANBFP.eta, RANBFP.s_t);


TANBFP.c_art = TANBFP.c_art + TANBFP.c_artss;
TANBFP.c_aro = TANBFP.c_aro + TANBFP.c_aross;
TANBFP.c_aht = TANBFP.c_aht + TANBFP.c_ahtss;
TANBFP.c_aho = TANBFP.c_aho + TANBFP.c_ahoss;
TANBFP.c_bt = TANBFP.c_bt + TANBFP.c_btss;
TANBFP.c_bo = TANBFP.c_bo + TANBFP.c_boss;

TANBFP.c_art = max(TANBFP.c_art, eps);
TANBFP.c_aro = max(TANBFP.c_aro, eps);
TANBFP.c_aht = max(TANBFP.c_aht, eps);
TANBFP.c_aho = max(TANBFP.c_aho, eps);
TANBFP.c_bt = max(TANBFP.c_bt, eps);
TANBFP.c_bo = max(TANBFP.c_bo, eps);

TANBFP.c_ar = ces_agg(TANBFP.c_art, TANBFP.c_aro, TANBFP.eta, TANBFP.s_t);
TANBFP.c_ah = ces_agg(TANBFP.c_aht, TANBFP.c_aho, TANBFP.eta, TANBFP.s_t);
TANBFP.c_b = ces_agg(TANBFP.c_bt, TANBFP.c_bo, TANBFP.eta, TANBFP.s_t);


TANBNR_FER.c_art = TANBNR_FER.c_art + TANBNR_FER.c_artss;
TANBNR_FER.c_aro = TANBNR_FER.c_aro + TANBNR_FER.c_aross;
TANBNR_FER.c_aht = TANBNR_FER.c_aht + TANBNR_FER.c_ahtss;
TANBNR_FER.c_aho = TANBNR_FER.c_aho + TANBNR_FER.c_ahoss;
TANBNR_FER.c_bt = TANBNR_FER.c_bt + TANBNR_FER.c_btss;
TANBNR_FER.c_bo = TANBNR_FER.c_bo + TANBNR_FER.c_boss;

TANBNR_FER.c_art = max(TANBNR_FER.c_art, eps);
TANBNR_FER.c_aro = max(TANBNR_FER.c_aro, eps);
TANBNR_FER.c_aht = max(TANBNR_FER.c_aht, eps);
TANBNR_FER.c_aho = max(TANBNR_FER.c_aho, eps);
TANBNR_FER.c_bt = max(TANBNR_FER.c_bt, eps);
TANBNR_FER.c_bo = max(TANBNR_FER.c_bo, eps);

TANBNR_FER.c_ar = ces_agg(TANBNR_FER.c_art, TANBNR_FER.c_aro, TANBNR_FER.eta, TANBNR_FER.s_t);
TANBNR_FER.c_ah = ces_agg(TANBNR_FER.c_aht, TANBNR_FER.c_aho, TANBNR_FER.eta, TANBNR_FER.s_t);
TANBNR_FER.c_b = ces_agg(TANBNR_FER.c_bt, TANBNR_FER.c_bo, TANBNR_FER.eta, TANBNR_FER.s_t);


TANBNR_PIT.c_art = TANBNR_PIT.c_art + TANBNR_PIT.c_artss;
TANBNR_PIT.c_aro = TANBNR_PIT.c_aro + TANBNR_PIT.c_aross;
TANBNR_PIT.c_aht = TANBNR_PIT.c_aht + TANBNR_PIT.c_ahtss;
TANBNR_PIT.c_aho = TANBNR_PIT.c_aho + TANBNR_PIT.c_ahoss;
TANBNR_PIT.c_bt = TANBNR_PIT.c_bt + TANBNR_PIT.c_btss;
TANBNR_PIT.c_bo = TANBNR_PIT.c_bo + TANBNR_PIT.c_boss;

TANBNR_PIT.c_art = max(TANBNR_PIT.c_art, eps);
TANBNR_PIT.c_aro = max(TANBNR_PIT.c_aro, eps);
TANBNR_PIT.c_aht = max(TANBNR_PIT.c_aht, eps);
```

```
TANBNR_PIT.c_aho = max(TANBNR_PIT.c_aho, eps);
TANBNR_PIT.c_bt = max(TANBNR_PIT.c_bt, eps);
TANBNR_PIT.c_bo = max(TANBNR_PIT.c_bo, eps);

TANBNR_PIT.c_ar = ces_agg(TANBNR_PIT.c_art, TANBNR_PIT.c_aro, TANBNR_PIT.eta, TANBNR_PIT.s_t);
TANBNR_PIT.c_ah = ces_agg(TANBNR_PIT.c_aht, TANBNR_PIT.c_aho, TANBNR_PIT.eta, TANBNR_PIT.s_t);
TANBNR_PIT.c_b = ces_agg(TANBNR_PIT.c_bt, TANBNR_PIT.c_bo, TANBNR_PIT.eta, TANBNR_PIT.s_t);


TANBFPNH.c_art = TANBFPNH.c_art + TANBFPNH.c_artss;
TANBFPNH.c_aro = TANBFPNH.c_aro + TANBFPNH.c_aross;
TANBFPNH.c_aht = TANBFPNH.c_aht + TANBFPNH.c_ahtss;
TANBFPNH.c_aho = TANBFPNH.c_aho + TANBFPNH.c_ahoss;
TANBFPNH.c_bt = TANBFPNH.c_bt + TANBFPNH.c_btss;
TANBFPNH.c_bo = TANBFPNH.c_bo + TANBFPNH.c_boss;

TANBFPNH.c_art = max(TANBFPNH.c_art, eps);
TANBFPNH.c_aro = max(TANBFPNH.c_aro, eps);
TANBFPNH.c_aht = max(TANBFPNH.c_aht, eps);
TANBFPNH.c_aho = max(TANBFPNH.c_aho, eps);
TANBFPNH.c_bt = max(TANBFPNH.c_bt, eps);
TANBFPNH.c_bo = max(TANBFPNH.c_bo, eps);

TANBFPNH.c_ar = ces_agg(TANBFPNH.c_art, TANBFPNH.c_aro, TANBFPNH.eta, TANBFPNH.s_art);
TANBFPNH.c_ah = ces_agg(TANBFPNH.c_aht, TANBFPNH.c_aho, TANBFPNH.eta, TANBFPNH.s_aht);
TANBFPNH.c_b = ces_agg(TANBFPNH.c_bt, TANBFPNH.c_bo, TANBFPNH.eta, TANBFPNH.s_bt);



% labour supply

RAFAFP.l = RAFAFP.l + RAFAFP.lss;

RANBFP.l = RANBFP.l + RANBFP.lss;

TANBFP.l = TANBFP.l + TANBFP.lss;
TANBFP.l_r = TANBFP.l_r + TANBFP.l_rss;
TANBFP.l_h = TANBFP.l_h + TANBFP.l_hss;

TANBNR_FER.l = TANBNR_FER.l + TANBNR_FER.lss;
TANBNR_FER.l_r = TANBNR_FER.l_r + TANBNR_FER.l_rss;
TANBNR_FER.l_h = TANBNR_FER.l_h + TANBNR_FER.l_hss;

TANBNR_PIT.l = TANBNR_PIT.l + TANBNR_PIT.lss;
TANBNR_PIT.l_r = TANBNR_PIT.l_r + TANBNR_PIT.l_rss;
TANBNR_PIT.l_h = TANBNR_PIT.l_h + TANBNR_PIT.l_hss;

TANBFPNH.l = TANBFPNH.l + TANBFPNH.lss;
TANBFPNH.l_r = TANBFPNH.l_r + TANBFPNH.l_rss;
TANBFPNH.l_h = TANBFPNH.l_h + TANBFPNH.l_hss;


% discount vector
betas = zeros(1, 100);

for t = 1:length(betas)

  betas(1, t) = beta^(t - 1);

endfor



% welfare

RAFAFP.utility_a = log(RAFAFP.c_a) + psi * log(1 - RAFAFP.l);
RAFAFP.welfare_a = betas * RAFAFP.utility_a.';

RAFAFP.utility_b = log(RAFAFP.c_b);
RAFAFP.welfare_b = betas * RAFAFP.utility_b.';


RANBFP.utility_a = log(RANBFP.c_a) + psi * log(1 - RANBFP.l);
RANBFP.welfare_a = betas * RANBFP.utility_a.';
```

```
RANBFP.utility_b = log(RANBFP.c_b);
RANBFP.welfare_b = betas * RANBFP.utility_b.';


TANBFP.utility_ar = log(TANBFP.c_ar) + psi * log(1 - TANBFP.l_r);
TANBFP.welfare_ar = betas * TANBFP.utility_ar.';

TANBFP.utility_ah = log(TANBFP.c_ah) + psi * log(1 - TANBFP.l_h);
TANBFP.welfare_ah = betas * TANBFP.utility_ah.';

TANBFP.utility_b = log(TANBFP.c_b);
TANBFP.welfare_b = betas * TANBFP.utility_b.';


TANBNR_FER.utility_ar = log(TANBNR_FER.c_ar) + psi * log(1 - TANBNR_FER.l_r);
TANBNR_FER.welfare_ar = betas * TANBNR_FER.utility_ar.';

TANBNR_FER.utility_ah = log(TANBNR_FER.c_ah) + psi * log(1 - TANBNR_FER.l_h);
TANBNR_FER.welfare_ah = betas * TANBNR_FER.utility_ah.';

TANBNR_FER.utility_b = log(TANBNR_FER.c_b);
TANBNR_FER.welfare_b = betas * TANBNR_FER.utility_b.';


TANBNR_PIT.utility_ar = log(TANBNR_PIT.c_ar) + psi * log(1 - TANBNR_PIT.l_r);
TANBNR_PIT.welfare_ar = betas * TANBNR_PIT.utility_ar.';

TANBNR_PIT.utility_ah = log(TANBNR_PIT.c_ah) + psi * log(1 - TANBNR_PIT.l_h);
TANBNR_PIT.welfare_ah = betas * TANBNR_PIT.utility_ah.';

TANBNR_PIT.utility_b = log(TANBNR_PIT.c_b);
TANBNR_PIT.welfare_b = betas * TANBNR_PIT.utility_b.';


TANBFPNH.utility_ar = log(TANBFPNH.c_ar) + psi * log(1 - TANBFPNH.l_r);
TANBFPNH.welfare_ar = betas * TANBFPNH.utility_ar.';

TANBFPNH.utility_ah = log(TANBFPNH.c_ah) + psi * log(1 - TANBFPNH.l_h);
TANBFPNH.welfare_ah = betas * TANBFPNH.utility_ah.';

TANBFPNH.utility_b = log(TANBFPNH.c_b);
TANBFPNH.welfare_b = betas * TANBFPNH.utility_b.';



% import/export, trade balance in country A:
RAFAFP.import = RAFAFP.c_ao;
RAFAFP.export = RAFAFP.c_bt;
RAFAFP.nex = RAFAFP.export - RAFAFP.import;


RANBFP.import = RANBFP.c_ao;
RANBFP.export = RANBFP.c_bt;
RANBFP.nex = RANBFP.export - RANBFP.import;


TANBFP.import = chi * TANBFP.c_aro + (1 - chi) * TANBFP.c_aho;
TANBFP.export = TANBFP.c_bt;
TANBFP.nex = TANBFP.export - TANBFP.import;


TANBNR_FER.import = chi * TANBNR_FER.c_aro + (1 - chi) * TANBNR_FER.c_aho;
TANBNR_FER.export = TANBNR_FER.c_bt;
TANBNR_FER.nex = TANBNR_FER.export - TANBNR_FER.import;


TANBNR_PIT.import = chi * TANBNR_PIT.c_aro + (1 - chi) * TANBNR_PIT.c_aho;
TANBNR_PIT.export = TANBNR_PIT.c_bt;
TANBNR_PIT.nex = TANBNR_PIT.export - TANBNR_PIT.import;


TANBFPNH.import = chi * TANBFPNH.c_aro + (1 - chi) * TANBFPNH.c_aho;
TANBFPNH.export = TANBFPNH.c_bt;
```

```matlab
TANBFPNH.nex = TANBFPNH.export - TANBFPNH.import;


%---------------------------------------------------------------
% 4. Plot IRFs
%---------------------------------------------------------------

% RAFAFP

subplot(3, 3, 1);
plot(1:100, RAFAFP.output, 1:100);
title('Output in country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 2);
plot(1:100, RAFAFP.price, 1:100, RAFAFP.p_tss);
title('Price index in country A and B');
xlabel('Quarters');
ylabel('Units of oil consumption');
subplot(3, 3, 3);
plot(1:99, RAFAFP.infl);
title('Inflation in country A and B');
xlabel('Quarters');
ylabel('Percent');
subplot(3, 3, 4);
plot(1:100, RAFAFP.c_a);
title('Consumption in Country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 5);
plot(1:100, RAFAFP.c_b);
title('Consumption in Country B');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 6);
plot(1:100, RAFAFP.import);
title('Import in Country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 7);
plot(1:100, RAFAFP.export);
title('Export in Country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 8);
plot(1:100, RAFAFP.nex);
title('Net export in Country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');

saveas(gcf, 'rafafp.jpg');


% RANBFP

subplot(3, 3, 1);
plot(1:100, RANBFP.output);
title('Output in country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 2);
plot(1:100, RANBFP.price);
title('Price index in country A and B');
xlabel('Quarters');
ylabel('Units of oil consumption');
subplot(3, 3, 3);
plot(1:99, RANBFP.infl);
title('Inflation in country A and B');
xlabel('Quarters');
ylabel('Percent');
subplot(3, 3, 4);
plot(1:100, RANBFP.c_a);
title('Consumption in Country A');
xlabel('Quarters');
```

```
ylabel('Units of tradeable cons.');
subplot(3, 3, 5);
plot(1:100, RANBFP.c_b);
title('Consumption in Country B');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 6);
plot(1:100, RANBFP.import);
title('Import in Country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 7);
plot(1:100, RANBFP.export);
title('Export in Country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 8);
plot(1:100, RANBFP.nex);
title('Net export in Country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');

saveas(gcf, 'ranbfp.jpg');


% TANBFP

subplot(3, 3, 1);
plot(1:100, TANBFP.output);
title('Output in country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 2);
plot(1:100, TANBFP.price);
title('Price index in country A and B');
xlabel('Quarters');
ylabel('Units of oil consumption');
subplot(3, 3, 3);
plot(1:99, TANBFP.infl);
title('Inflation in country A and B');
xlabel('Quarters');
ylabel('Percent');
subplot(3, 3, 4);
plot(1:100, TANBFP.c_ar);
title('Consumption for Ricardians');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 5);
plot(1:100, TANBFP.c_ah);
title('Consumption for HTMs');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 6);
plot(1:100, TANBFP.c_b);
title('Consumption in Country B');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 7);
plot(1:100, TANBFP.import);
title('Import in Country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 8);
plot(1:100, TANBFP.export);
title('Export in Country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 9);
plot(1:100, TANBFP.nex);
title('Net export in Country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');

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

```
% TANBNR_FER

subplot(3, 3, 1);
plot(1:100, TANBNR_FER.output);
title('Output in country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 2);
plot(1:100, TANBNR_FER.price);
title('Price index in country A and B');
xlabel('Quarters');
ylabel('Units of oil consumption');
subplot(3, 3, 3);
plot(1:99, TANBNR_FER.infl);
title('Inflation in country A and B');
xlabel('Quarters');
ylabel('Percent');
subplot(3, 3, 4);
plot(1:100, TANBNR_FER.c_ar);
title('Consumption for Ricardians');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 5);
plot(1:100, TANBNR_FER.c_ah);
title('Consumption for HTMs');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 6);
plot(1:100, TANBNR_FER.c_b);
title('Consumption in Country B');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 7);
plot(1:100, TANBNR_FER.import);
title('Import in Country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 8);
plot(1:100, TANBNR_FER.export);
title('Export in Country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 9);
plot(1:100, TANBNR_FER.nex);
title('Net export in Country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');

saveas(gcf, 'tanbnr_fer.jpg');



% TANBNR_PIT

subplot(5, 3, 1);
plot(1:100, TANBNR_PIT.output);
title('Output in country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(5, 3, 2);
plot(1:100, TANBNR_PIT.price_a);
title('Price index in country A');
xlabel('Quarters');
ylabel('Units of oil consumption');
subplot(5, 3, 3);
plot(1:100, TANBNR_PIT.price_b);
title('Price index in country B');
xlabel('Quarters');
ylabel('Units of oil consumption');
```

```
subplot(5, 3, 4);
plot(1:99, TANBNR_PIT.infl_a);
title('Inflation in country A');
xlabel('Quarters');
ylabel('Percent');
subplot(5, 3, 5);
plot(1:99, TANBNR_PIT.infl_b);
title('Inflation in country B');
xlabel('Quarters');
ylabel('Percent');
subplot(5, 3, 6);
plot(1:100, TANBNR_PIT.c_ar);
title('Consumption for Ricardians');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(5, 3, 7);
plot(1:100, TANBNR_PIT.c_ah);
title('Consumption for HTMs');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(5, 3, 8);
plot(1:100, TANBNR_PIT.c_b);
title('Consumption in Country B');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(5, 3, 9);
plot(1:100, TANBNR_PIT.import);
title('Import in Country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(5, 3, 10);
plot(1:100, TANBNR_PIT.export);
title('Export in Country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(5, 3, 11);
plot(1:100, TANBNR_PIT.nex);
title('Net export in Country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');

saveas(gcf, 'tanbnr_pit.jpg');



% TANBFPNH

subplot(3, 3, 1);
plot(1:100, TANBFP.output);
title('Output in country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 2);
plot(1:100, TANBFPNH.price);
title('Price index in country A and B');
xlabel('Quarters');
ylabel('Units of oil consumption');
subplot(3, 3, 3);
plot(1:99, TANBFPNH.infl);
title('Inflation in country A and B');
xlabel('Quarters');
ylabel('Percent');
subplot(3, 3, 4);
plot(1:100, TANBFPNH.c_ar);
title('Consumption for Ricardians');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 5);
plot(1:100, TANBFPNH.c_ah);
title('Consumption for HTMs');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 6);
plot(1:100, TANBFPNH.c_b);
```

```
title('Consumption in Country B');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 7);
plot(1:100, TANBFPNH.import);
title('Import in Country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 8);
plot(1:100, TANBFPNH.export);
title('Export in Country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');
subplot(3, 3, 9);
plot(1:100, TANBFPNH.nex);
title('Net export in Country A');
xlabel('Quarters');
ylabel('Units of tradeable cons.');

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

Appendix A2: Parameter File

```

%-----------------------------------------------------------------------
%
% Title: International Macro-Finance Problem Set 4, parameter file
% Author: ---
% Date: 10/01/2024
% Description: Parameter file for the final problem set
%
%-----------------------------------------------------------------------


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

close all;
clear all;

%-----------------------------------------------------------------------
% 1. Defining parameters
%-----------------------------------------------------------------------


% key moments to target:
% y_o = 12.1 * 365 * 90 * 10^(-6) = .4 Saudi oil production in 2022, tn euro
% c_ao  = .46 EU import of oil from Saudi Arabia in 2022, tn euro
% l = .22; average weekly work hours in the EU (37.5 / (24 * 7))
% y_t * p_t = 16.75 EU GDP in 2022, tn USD

% this implies:
% y_t = l^(1 - alpha) = .35 with alpha = .3
% p_t = 16.75 / .35 = 47.86


par.alpha = .3; % income share of capital
par.beta  = .99;
par.psi = 1.5; % elasticity of labour supply, normally between 1 and 2 (see Blundell et al. 2000)
par.phi = 10; % strength of the nominal rigidity
par.eta = .4; % elasticity of substitution between the tradeable good and oil
par.s_t = .75; % home bias towards tradeable in both countries
par.s_art = .78; %  home bias towards tradeable for ricardians
par.s_aht = .9; % home bias towards tradeable for htms
par.s_bt  = .8; % home bias towards tradeable in country b
par.theta = .4; % elasticity of substitution between goods in the tradeable sector
par.rho = .9; % persistence of shock
par.gamma = .1; % mean oil endowment of country b (y_bbar = gamma / (1 - rho))
par.chi = .8; % proportion of ricardian HHs in country a
```

Appendix B: RAFAFP mod file

```

%----------------------------------------------------------------
%
% Title: International Macro-Finance Final Assignment, model file
% Author: ---
% Date: 25/12/2023
% Description: Representative agent, financial autarky, flexible prices
%
%----------------------------------------------------------------


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

close all;

%----------------------------------------------------------------
% 1. Defining variables
%----------------------------------------------------------------

var c_ao c_bo c_at c_bt l y_t y_o p_t w pi;

varexo err_yo;


%----------------------------------------------------------------
% 2. Calibration
%----------------------------------------------------------------

parameters alpha psi eta s_t theta rho gamma;

load parameters;

set_param_value('alpha'   ,par.alpha);
set_param_value('psi'     ,par.psi);
set_param_value('eta'     ,par.eta);
set_param_value('s_t'     ,par.s_t);
set_param_value('theta'   ,par.theta);
set_param_value('rho'     ,par.rho);
set_param_value('gamma'   ,par.gamma);


%----------------------------------------------------------------
% 3. Model
%----------------------------------------------------------------


model;

% consumption allocation between oil and tradeable in country a
(1 - s_t) * c_ao^(-1 / eta) - s_t * c_at^(-1 / eta) / p_t;

% consumption allocation between oil and tradeable in country b
(1 - s_t) * c_bo^(-1 / eta) - s_t * c_bt^(-1 / eta) / p_t;

% HH's labour supply decision in country a
psi - (1 - s_t) * c_ao^(-1 / eta) / (s_t * c_at^(1 - 1 / eta) +
                        (1 - s_t) * c_ao^(1 - 1 / eta)) *
                                   w * (1 - l);

% HH's budget constraint in country a
p_t * c_at + c_ao - w * l - pi;

% HH's budget constraint in country b
c_bo + p_t * c_bt  - y_o;

% firm's FOC
(1 - theta) * (1 - alpha) * p_t *
          l^((1 - theta) * (1 - alpha) - 1) * y_t^theta - w;

% firm's profit
p_t * l^(1 - alpha) - w * l - pi;
```

```
% tradeable goods market-clearing condition
y_t - c_at - c_bt;

% oil market-clearing condition
y_o - c_ao - c_bo;

% oil endowment, stochastic process
gamma + rho * y_o(-1) - err_yo - y_o;

end;


%----------------------------------------------------------------------
% 4. Steady state
%----------------------------------------------------------------------


initval;

l = .22;
c_ao = .46;
y_t = l^(1 - alpha);
p_t = 16.75 / y_t;
c_at = s_t^eta * c_ao / (p_t^eta * (1 - s_t)^eta);
c_bt = y_t - c_at;
y_o = gamma / (1 - rho);
c_bo = y_o - c_ao;
w = (1 - theta) * (1 - alpha) * p_t *
          l^((1 - theta) * (1 - alpha) - 1) * y_t^theta;
pi = p_t * l^(1 - alpha) - w * l;

end;

steady(maxit = 1000, solve_algo = 1);


%----------------------------------------------------------------------
% 5. Impulse response function
%----------------------------------------------------------------------


shocks;

var err_yo;
stderr .1;

end;


stoch_simul(order=1, irf=100, irf_plot_threshold=0) c_ao
                                c_bo c_at c_bt l y_t y_o p_t w pi;
```

Appendix C: RANBFP mod file

```

%----------------------------------------------------------------
%
% Title: International Macro-Finance Final Assignment, model file
% Author: ---
% Date: 25/12/2023
% Description: Representative agent, nominal bond, flexible prices
%
%----------------------------------------------------------------


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

close all;

%----------------------------------------------------------------
% 1. Defining variables
%----------------------------------------------------------------

var c_ao c_bo c_at c_bt l y_t y_o p_t w pi n_a n_b r_star;

varexo err_yo;


%----------------------------------------------------------------
% 2. Calibration
%----------------------------------------------------------------

parameters alpha beta psi eta s_t theta rho gamma;

load parameters;

set_param_value('alpha'   ,par.alpha);
set_param_value('beta'   ,par.beta);
set_param_value('psi'     ,par.psi);
set_param_value('eta'     ,par.eta);
set_param_value('s_t'     ,par.s_t);
set_param_value('theta'   ,par.theta);
set_param_value('rho'     ,par.rho);
set_param_value('gamma'   ,par.gamma);

s_t = .8;
theta = .65;


%----------------------------------------------------------------
% 3. Model
%----------------------------------------------------------------


model;

% consumption allocation between oil and tradeable in country a
(1 - s_t) * c_ao^(-1 / eta) - s_t * c_at^(-1 / eta) / p_t;

% consumption allocation between oil and tradeable in country b
(1 - s_t) * c_bo^(-1 / eta) - s_t * c_bt^(-1 / eta) / p_t;

% Euler equation in country a
c_ao^(-1 / eta) / (s_t * c_at^(1 - 1 / eta) +
     (1 - s_t) * c_ao^(1 - 1 / eta))
         - beta * r_star *
                  c_ao(+1)^(-1 / eta) / (s_t * c_at(+1)^(1 - 1 / eta) +
                             (1 - s_t) * c_ao(+1)^(1 - 1 / eta));

% Euler equation in country b
c_bo^(-1 / eta) / (s_t * c_bt^(1 - 1 / eta) +
     (1 - s_t) * c_bo^(1 - 1 / eta)) -
         beta * r_star *
                  c_bo(+1)^(-1 / eta) / (s_t * c_bt(+1)^(1 - 1 / eta) +
                             (1 - s_t) * c_bo(+1)^(1 - 1 / eta));
```

```
% HH's labour supply decision in country a
psi - (1 - s_t) * c_ao^(-1 / eta) / (s_t * c_at^(1 - 1 / eta) +
                      (1 - s_t) * c_ao^(1 - 1 / eta)) *
                                w * (1 - l);

% HH's budget constraint in country a
p_t * c_at + c_ao - w * l - pi - (n_a(-1) - n_a / r_star);

% HH's budget constraint in country b
c_bo + p_t * c_bt  - y_o - (n_b(-1) - n_b / r_star);

% firm's FOC
(1 - theta) * (1 - alpha) * p_t *
          l^((1 - theta) * (1 - alpha) - 1) * y_t^theta - w;

% firm's profit
p_t * l^(1 - alpha) - w * l - pi;

% tradeable goods market-clearing condition
y_t - c_at - c_bt;

% oil market-clearing condition
y_o - c_ao - c_bo;

% bond market clearing condition
n_a + n_b;

% oil endowment, stochastic process
gamma + rho * y_o(-1) - err_yo - y_o;

end;


%-----------------------------------------------------------------
% 4. Steady state
%-----------------------------------------------------------------


initval;

l = .22;
c_ao = .46;
y_t = l^(1 - alpha); % from production function
p_t = 16.75 / y_t; % from target GDP in country a
c_at = s_t^eta * c_ao / (p_t^eta * (1 - s_t)^eta); % from tradeable-oil tradeoff
c_bt = y_t - c_at; % from tradeable MC
y_o = gamma / (1 - rho); % from oil stochastic process
c_bo = y_o - c_ao; % from oil MC
w = (1 - theta) * (1 - alpha) * p_t *
          l^((1 - theta) * (1 - alpha) - 1) * y_t^theta; % from firm's FOC
pi = p_t * l^(1 - alpha) - w * l; % from firm's profit
r_star = 1 / beta; % from SS Euler
n_a = r_star / (r_star - 1) * (p_t * c_at + c_ao - w * l - pi); % from the BC
n_b = - n_a; % from bond MC

end;

steady(maxit = 1000);


%-----------------------------------------------------------------
% 5. Impulse response function
%-----------------------------------------------------------------

initval;

c_ao = 0.425983;
c_bo = 0.174017;
c_at = 0.318506;
c_bt = 0.130112;
l = 0.318186;
y_t = 0.448618;
y_o = 0.6;
```

```
p_t = 30.8469;
w = 4.56664;
pi = 12.3854;
n_a = -358.757;
n_b = 358.757;
r_star = 1.0101;


end;

endval;

n_a = -358.757;
n_b = 358.757;

end;


shocks;

var err_yo;
stderr .1;

end;


stoch_simul(order=1,  irf=100, irf_plot_threshold=0) c_ao
                                 c_bo c_at c_bt l y_t y_o p_t w pi r_star
                                 n_a n_b;

model_diagnostics;
```

Appendix D: TANBFP mod file

```

%----------------------------------------------------------------
%
% Title: International Macro-Finance Final Assignment, model file
% Author: ---
% Date: 25/12/2023
% Description: Two agent, nominal bond, flexible prices
%
%----------------------------------------------------------------


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

close all;

%----------------------------------------------------------------
% 1. Defining variables
%----------------------------------------------------------------

var c_aho c_aro c_bo c_aht c_art c_bt l_h l_r l y_t y_o p_t w pi
          n_a n_b r_star pi_r pi_h;

varexo err_yo;


%----------------------------------------------------------------
% 2. Calibration
%----------------------------------------------------------------

parameters alpha beta psi eta s_t theta rho gamma chi;
load parameters;

set_param_value('alpha'    ,par.alpha);
set_param_value('beta'     ,par.beta);
set_param_value('psi'      ,par.psi);
set_param_value('eta'      ,par.eta);
set_param_value('s_t'   ,par.s_t);
set_param_value('theta'    ,par.theta);
set_param_value('rho'      ,par.rho);
set_param_value('gamma'    ,par.gamma);
set_param_value('chi'      ,par.chi);

s_t = .8;
theta = .5;



%----------------------------------------------------------------
% 3. Model
%----------------------------------------------------------------


model;

% consumption allocation between oil and tradeable for htm HHs in country a
(1 - s_t) * c_aho^(-1 / eta) - s_t * c_aht^(-1 / eta) / p_t;

% consumption allocation between oil and tradeable for ricardian HHs in country a
(1 - s_t) * c_aro^(-1 / eta) - s_t * c_art^(-1 / eta) / p_t;

% consumption allocation between oil and tradeable in country b
(1 - s_t) * c_bo^(-1 / eta) - s_t * c_bt^(-1 / eta) / p_t;

% Euler equation for ricardian HHs in country a
c_aro^(-1 / eta) / (s_t * c_art^(1 - 1 / eta) +
     (1 - s_t) * c_aro^(1 - 1 / eta)) -
          beta * r_star *
                c_aro(+1)^(-1 / eta) / (s_t * c_art(+1)^(1 - 1 / eta) +
                          (1 - s_t) * c_aro(+1)^(1 - 1 / eta));

% Euler equation in country b
```

```
c_bo^(-1 / eta) / (s_t * c_bt^(1 - 1 / eta) +
     (1 - s_t) * c_bo^(1 - 1 / eta)) -
       beta * r_star *
                   c_bo(+1)^(-1 / eta) / (s_t * c_bt(+1)^(1 - 1 / eta) +
                              (1 - s_t) * c_bo(+1)^(1 - 1 / eta));

% labour supply decision for htm HHs in country a
psi - (1 - s_t) * c_aho^(-1 / eta) / (s_t * c_aht^(1 - 1 / eta) +
                      (1 - s_t) * c_aho^(1 - 1 / eta)) *
                                w * (1 - l_h);

% labour supply decision for ricardian HHs in country a
psi - (1 - s_t) * c_aro^(-1 / eta) / (s_t * c_art^(1 - 1 / eta) +
                      (1 - s_t) * c_aro^(1 - 1 / eta)) *
                                w * (1 - l_r);

% htm's budget constraint in country a
p_t * c_aht + c_aho - w * l_h - pi_h;

% ricardian's budget constraint in country a
p_t * c_art + c_aro - w * l_r - pi_r - (n_a(-1) - n_a / r_star);

% HH's budget constraint in country b
c_bo + p_t * c_bt  - y_o - (n_b(-1) - n_b / r_star);

% firm's FOC
(1 - theta) * (1 - alpha) * p_t *
          l^((1 - theta) * (1 - alpha) - 1) * y_t^theta - w;

% firm's profit
p_t * l^(1 - alpha) - w * l - pi;

% profit distribution
chi * pi_r + (1 - chi) * pi_h - pi;

% fix ricardian's profit
pi_r - pi;

% tradeable goods market-clearing condition
y_t - (1 - chi) * c_aht - chi * c_art - c_bt;

% oil market-clearing condition
y_o - (1 - chi) * c_aho - chi * c_aro - c_bo;

% labour market-clearing condition
chi * l_r + (1 - chi) * l_h - l;

% bond market clearing condition
chi * n_a + n_b;

% oil endowment, stochastic process
gamma + rho * y_o(-1) - err_yo - y_o;

end;


%----------------------------------------------------------------
% 4. Steady state
%----------------------------------------------------------------


initval;

l_r = .22;
l_h = .22;
c_aro = .46;
c_aho = .46;
l = chi * l_r + (1 - chi) * l_h;
y_t = l^(1 - alpha); % from production function
p_t = 16.75 / y_t; % from target GDP in country a
c_art = s_t^eta * c_aro / (p_t^eta * (1 - s_t)^eta); % from tradeable-oil tradeoff
c_aht = s_t^eta * c_aho / (p_t^eta * (1 - s_t)^eta);
c_bt = y_t - chi * c_art - (1 - chi) * c_aht; % from tradeable MC
y_o = gamma / (1 - rho); % from oil stochastic process
```

```
c_bo = y_o - chi * c_aro - (1 - chi) * c_aho; % from oil MC
w = (1 - theta) * (1 - alpha) * p_t *
          l^((1 - theta) * (1 - alpha) - 1) * y_t^theta; % from firm's FOC
pi = p_t * l^(1 - alpha) - w * l; % from firm's profit
pi_r = pi;
pi_h = (pi - chi * pi_r) / (1 - chi);
r_star = 1 / beta; % from SS Euler
n_a = r_star / (r_star - 1) * (p_t * c_art + c_aro - w * l - pi_r); % from the BC
n_b = - chi * n_a; % from bond MC

end;

steady(maxit = 1000, solve_algo = 3);



%-----------------------------------------------------------------------
% 5. Impulse response function
%-----------------------------------------------------------------------


shocks;

var err_yo;
stderr .1;

end;


stoch_simul(order=1, irf=100, irf_plot_threshold=0)
                      c_aho c_aro c_bo c_aht c_art c_bt l_h
                             l_r l y_t y_o p_t w pi n_a n_b r_star;
```

Appendix E: TANBFP mod file

```

%--------------------------------------------------------------------------
%
% Title: International Macro-Finance Final Assignment, model file
% Author: ---
% Date: 25/12/2023
% Description: Two agent, nominal bond, flexible prices
%
%--------------------------------------------------------------------------


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

close all;

%--------------------------------------------------------------------------
% 1. Defining variables
%--------------------------------------------------------------------------

var c_aho c_aro c_bo c_aht c_art c_bt l_h l_r l y_t y_o p_t w pi
          n_a n_b r_star pi_r pi_h;

varexo err_yo;


%--------------------------------------------------------------------------
% 2. Calibration
%--------------------------------------------------------------------------

parameters alpha beta psi eta s_t theta rho gamma chi;
load parameters;

set_param_value('alpha'   ,par.alpha);
set_param_value('beta'    ,par.beta);
set_param_value('psi'     ,par.psi);
set_param_value('eta'     ,par.eta);
set_param_value('s_t'   ,par.s_t);
set_param_value('theta'   ,par.theta);
set_param_value('rho'     ,par.rho);
set_param_value('gamma'   ,par.gamma);
set_param_value('chi'     ,par.chi);

s_t = .8;
theta = .5;



%--------------------------------------------------------------------------
% 3. Model
%--------------------------------------------------------------------------


model;

% consumption allocation between oil and tradeable for htm HHs in country a
(1 - s_t) * c_aho^(-1 / eta) - s_t * c_aht^(-1 / eta) / p_t;

% consumption allocation between oil and tradeable for ricardian HHs in country a
(1 - s_t) * c_aro^(-1 / eta) - s_t * c_art^(-1 / eta) / p_t;

% consumption allocation between oil and tradeable in country b
(1 - s_t) * c_bo^(-1 / eta) - s_t * c_bt^(-1 / eta) / p_t;

% Euler equation for ricardian HHs in country a
c_aro^(-1 / eta) / (s_t * c_art^(1 - 1 / eta) +
     (1 - s_t) * c_aro^(1 - 1 / eta)) -
          beta * r_star *
                c_aro(+1)^(-1 / eta) / (s_t * c_art(+1)^(1 - 1 / eta) +
                          (1 - s_t) * c_aro(+1)^(1 - 1 / eta));

% Euler equation in country b
```

```
c_bo^(-1 / eta) / (s_t * c_bt^(1 - 1 / eta) +
     (1 - s_t) * c_bo^(1 - 1 / eta)) -
       beta * r_star *
                 c_bo(+1)^(-1 / eta) / (s_t * c_bt(+1)^(1 - 1 / eta) +
                           (1 - s_t) * c_bo(+1)^(1 - 1 / eta));

% labour supply decision for htm HHs in country a
psi - (1 - s_t) * c_aho^(-1 / eta) / (s_t * c_aht^(1 - 1 / eta) +
                      (1 - s_t) * c_aho^(1 - 1 / eta)) *
                                w * (1 - l_h);

% labour supply decision for ricardian HHs in country a
psi - (1 - s_t) * c_aro^(-1 / eta) / (s_t * c_art^(1 - 1 / eta) +
                      (1 - s_t) * c_aro^(1 - 1 / eta)) *
                                w * (1 - l_r);

% htm's budget constraint in country a
p_t * c_aht + c_aho - w * l_h - pi_h;

% ricardian's budget constraint in country a
p_t * c_art + c_aro - w * l_r - pi_r - (n_a(-1) - n_a / r_star);

% HH's budget constraint in country b
c_bo + p_t * c_bt  - y_o - (n_b(-1) - n_b / r_star);

% firm's FOC
(1 - theta) * (1 - alpha) * p_t *
          l^((1 - theta) * (1 - alpha) - 1) * y_t^theta - w;

% firm's profit
p_t * l^(1 - alpha) - w * l - pi;

% profit distribution
chi * pi_r + (1 - chi) * pi_h - pi;

% fix ricardian's profit
pi_r - pi;

% tradeable goods market-clearing condition
y_t - (1 - chi) * c_aht - chi * c_art - c_bt;

% oil market-clearing condition
y_o - (1 - chi) * c_aho - chi * c_aro - c_bo;

% labour market-clearing condition
chi * l_r + (1 - chi) * l_h - l;

% bond market clearing condition
chi * n_a + n_b;

% oil endowment, stochastic process
gamma + rho * y_o(-1) - err_yo - y_o;

end;


%-----------------------------------------------------------------
% 4. Steady state
%-----------------------------------------------------------------


initval;

l_r = .22;
l_h = .22;
c_aro = .46;
c_aho = .46;
l = chi * l_r + (1 - chi) * l_h;
y_t = l^(1 - alpha); % from production function
p_t = 16.75 / y_t; % from target GDP in country a
c_art = s_t^eta * c_aro / (p_t^eta * (1 - s_t)^eta); % from tradeable-oil tradeoff
c_aht = s_t^eta * c_aho / (p_t^eta * (1 - s_t)^eta);
c_bt = y_t - chi * c_art - (1 - chi) * c_aht; % from tradeable MC
y_o = gamma / (1 - rho); % from oil stochastic process
```

```
c_bo = y_o - chi * c_aro - (1 - chi) * c_aho; % from oil MC
w = (1 - theta) * (1 - alpha) * p_t *
          l^((1 - theta) * (1 - alpha) - 1) * y_t^theta; % from firm's FOC
pi = p_t * l^(1 - alpha) - w * l; % from firm's profit
pi_r = pi;
pi_h = (pi - chi * pi_r) / (1 - chi);
r_star = 1 / beta; % from SS Euler
n_a = r_star / (r_star - 1) * (p_t * c_art + c_aro - w * l - pi_r); % from the BC
n_b = - chi * n_a; % from bond MC

end;

steady(maxit = 1000, solve_algo = 3);



%-----------------------------------------------------------------
% 5. Impulse response function
%-----------------------------------------------------------------


shocks;

var err_yo;
stderr .1;

end;


stoch_simul(order=1, irf=100, irf_plot_threshold=0)
                       c_aho c_aro c_bo c_aht c_art c_bt l_h
                              l_r l y_t y_o p_t w pi n_a n_b r_star;
```

Appendix F: TANBNR FER mod file

```

%----------------------------------------------------------------------------
%
% Title: International Macro-Finance Final Assignment, model file
% Author: ---
% Date: 25/12/2023
% Description: Two agent, nominal bond, nominal rigidities, fixed exchange rate
%
%----------------------------------------------------------------------------


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

close all;

%----------------------------------------------------------------------------
% 1. Defining variables
%----------------------------------------------------------------------------

var c_aho c_aro c_bo c_aht c_art c_bt l_h l_r l y_t y_o p_t w pi
          n_a n_b r_star pi_r pi_h;

varexo err_yo;


%----------------------------------------------------------------------------
% 2. Calibration
%----------------------------------------------------------------------------

parameters alpha beta psi phi eta s_t theta rho gamma chi;

load parameters;

set_param_value('alpha'   ,par.alpha);
set_param_value('beta'    ,par.beta);
set_param_value('psi'     ,par.psi);
set_param_value('phi'     ,par.phi);
set_param_value('eta'     ,par.eta);
set_param_value('s_t'   ,par.s_t);
set_param_value('theta'   ,par.theta);
set_param_value('rho'     ,par.rho);
set_param_value('gamma'   ,par.gamma);
set_param_value('chi'     ,par.chi);

s_t = .8;
theta = .5;


%----------------------------------------------------------------------------
% 3. Model
%----------------------------------------------------------------------------


model;

% consumption allocation between oil and tradeable for htm HHs in country a
(1 - s_t) * c_aho^(-1 / eta) - s_t * c_aht^(-1 / eta) / p_t;

% consumption allocation between oil and tradeable for ricardian HHs in country a
(1 - s_t) * c_aro^(-1 / eta) - s_t * c_art^(-1 / eta) / p_t;

% consumption allocation between oil and tradeable in country b
(1 - s_t) * c_bo^(-1 / eta) - s_t * c_bt^(-1 / eta) / p_t;

% Euler equation for ricardian HHs in country a
c_aro^(-1 / eta) / (s_t * c_art^(1 - 1 / eta) +
     (1 - s_t) * c_aro^(1 - 1 / eta)) -
          beta * r_star *
                 c_aro(+1)^(-1 / eta) / (s_t * c_art(+1)^(1 - 1 / eta) +
                           (1 - s_t) * c_aro(+1)^(1 - 1 / eta));

```

```
% Euler equation in country b
c_bo^(-1 / eta) / (s_t * c_bt^(1 - 1 / eta) +
     (1 - s_t) * c_bo^(1 - 1 / eta)) -
       beta * r_star *
                  c_bo(+1)^(-1 / eta) / (s_t * c_bt(+1)^(1 - 1 / eta) +
                            (1 - s_t) * c_bo(+1)^(1 - 1 / eta));

% labour supply decision for htm HHs in country a
psi - (1 - s_t) * c_aho^(-1 / eta) / (s_t * c_aht^(1 - 1 / eta) +
                      (1 - s_t) * c_aho^(1 - 1 / eta)) *
                               w * (1 - l_h);

% labour supply decision for ricardian HHs in country a
psi - (1 - s_t) * c_aro^(-1 / eta) / (s_t * c_art^(1 - 1 / eta) +
                      (1 - s_t) * c_aro^(1 - 1 / eta)) *
                               w * (1 - l_r);

% htm's budget constraint in country a
p_t * c_aht + c_aho - w * l_h - pi_h;

% ricardian's budget constraint in country a
p_t * c_art + c_aro - w * l_r - pi_r - (n_a(-1) - n_a / r_star);

% HH's budget constraint in country b
c_bo + p_t * c_bt  - y_o - (n_b(-1) - n_b / r_star);

% firm's FOC
p_t * (1 - theta) * (1 - alpha) * l^((1 - theta) * (1 - alpha) - 1) * y_t^theta -
          w - phi * (p_t * y_t^theta * l(-1)^(theta * (1 - alpha)) /
                (p_t(-1) * l^(theta * (1 - alpha)) * y_t(-1)^theta) - 1) *
                          p_t * y_t * (-theta) * (1 - alpha) * p_t * y_t^theta *
                                l(-1)^(theta * (1 - alpha)) / (p_t(-1) *
                                      l^(theta * (1 - alpha) + 1) * y_t(-1)^theta);

% firm's profit
p_t * l^(1 - alpha) - w * l - phi / 2 * (p_t / p_t(-1) - 1)^2 * p_t * y_t - pi;

% profit distribution
chi * pi_r + (1 - chi) * pi_h - pi;

% impose profit for ricardians
pi_r - pi;

% tradeable goods market-clearing condition
y_t - (1 - chi) * c_aht - chi * c_art - c_bt;

% oil market-clearing condition
y_o - (1 - chi) * c_aho - chi * c_aro - c_bo;

% labour market-clearing condition
chi * l_r + (1 - chi) * l_h - l;

% bond market clearing condition
chi * n_a + n_b;

% oil endowment, stochastic process
gamma + rho * y_o(-1) - err_yo - y_o;

end;


%----------------------------------------------------------------
% 4. Steady state
%----------------------------------------------------------------


initval;

l_r = .22;
l_h = .22;
c_aro = .46;
c_aho = .46;
l = chi * l_r + (1 - chi) * l_h;
y_t = l^(1 - alpha); % from production function
```

```
p_t = 16.75 / y_t; % from target GDP in country a
c_art = s_t^eta * c_aro / (p_t^eta * (1 - s_t)^eta); % from tradeable-oil tradeoff
c_aht = s_t^eta * c_aho / (p_t^eta * (1 - s_t)^eta);
c_bt = y_t - chi * c_art - (1 - chi) * c_aht; % from tradeable MC
y_o = gamma / (1 - rho); % from oil stochastic process
c_bo = y_o - chi * c_aro - (1 - chi) * c_aho; % from oil MC
w = (1 - theta) * (1 - alpha) * p_t *
          l^((1 - theta) * (1 - alpha) - 1) * y_t^theta; % from firm's FOC at SS
pi = p_t * l^(1 - alpha) - w * l; % from firm's profit at SS
pi_r = pi;
pi_h = (pi - chi * pi_r) / (1 - chi);
r_star = 1 / beta; % from SS Euler
n_a = r_star / (r_star - 1) * (p_t * c_art + c_aro - w * l - pi_r); % from the BC
n_b = - n_a; % from bond MC

end;

steady(maxit = 1000, solve_algo = 3);




%---------------------------------------------------------------
% 5. Impulse response function
%---------------------------------------------------------------

shocks;

var err_yo;
stderr .1;

end;


stoch_simul(order=1, irf=100, irf_plot_threshold=0)
                      c_aho c_aro c_bo c_aht c_art c_bt l_h
                             l_r l y_t y_o p_t w pi n_a n_b r_star;

model_diagnostics;
```

Appendix G: TANBNR PIT mod file

```

%--------------------------------------------------------------------------
%
% Title: International Macro-Finance Final Assignment, model file
% Author: ---
% Date: 25/12/2023
% Description: Two agent, nominal bond, nominal rigidities,
%                      price inflation targeting
%
%--------------------------------------------------------------------------


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

close all;

%--------------------------------------------------------------------------
% 1. Defining variables
%--------------------------------------------------------------------------

var c_aho c_aro c_bo c_aht c_art c_bt l_h l_r l y_t y_o p_t
          e w pi n_a n_b r_star pi_r pi_h;

varexo err_yo;


%--------------------------------------------------------------------------
% 2. Calibration
%--------------------------------------------------------------------------

parameters alpha beta psi phi eta s_t theta rho gamma chi;

load parameters;

set_param_value('alpha'   ,par.alpha);
set_param_value('beta'    ,par.beta);
set_param_value('psi'     ,par.psi);
set_param_value('phi'     ,par.phi);
set_param_value('eta'     ,par.eta);
set_param_value('s_t'   ,par.s_t);
set_param_value('theta'   ,par.theta);
set_param_value('rho'     ,par.rho);
set_param_value('gamma'   ,par.gamma);
set_param_value('chi'     ,par.chi);


s_t = .8;
theta = .562;


%--------------------------------------------------------------------------
% 3. Model
%--------------------------------------------------------------------------


model;

% consumption allocation between oil and tradeable for htm HHs in country a
(1 - s_t) * c_aho^(-1 / eta) / e - s_t * c_aht^(-1 / eta) / p_t;

% consumption allocation between oil and tradeable for ricardian HHs in country a
(1 - s_t) * c_aro^(-1 / eta) / e - s_t * c_art^(-1 / eta) / p_t;

% consumption allocation between oil and tradeable in country b
% here, p_tstar = p_t / e by LOOP
(1 - s_t) * c_bo^(-1 / eta) - s_t * c_bt^(-1 / eta) / (p_t / e);

% Euler equation for ricardian HHs in country a
c_aro^(-1 / eta) / ((s_t * c_art^(1 - 1 / eta) +
     (1 - s_t) * c_aro^(1 - 1 / eta)) * e) -
        beta * r_star *
```

```
            c_aro(+1)^(-1 / eta) / ((s_t * c_art(+1)^(1 - 1 / eta) +
                 (1 - s_t) * c_aro(+1)^(1 - 1 / eta)) * e(+1));

% Euler equation in country b
c_bo^(-1 / eta) / (s_t * c_bt^(1 - 1 / eta) +
     (1 - s_t) * c_bo^(1 - 1 / eta)) -
       beta * r_star *
                 c_bo(+1)^(-1 / eta) / (s_t * c_bt(+1)^(1 - 1 / eta) +
                           (1 - s_t) * c_bo(+1)^(1 - 1 / eta));

% labour supply decision for htm HHs in country a
psi - (1 - s_t) * c_aho^(-1 / eta) / ((s_t * c_aht^(1 - 1 / eta) +
                 (1 - s_t) * c_aho^(1 - 1 / eta)) * e) *
                           w * (1 - l_h);

% labour supply decision for ricardian HHs in country a
psi - (1 - s_t) * c_aro^(-1 / eta) / ((s_t * c_art^(1 - 1 / eta) +
                 (1 - s_t) * c_aro^(1 - 1 / eta)) * e) *
                           w * (1 - l_r);

% htm's budget constraint in country a
p_t * c_aht + e * c_aho - w * l_h - pi_h;

% ricardian's budget constraint in country a
p_t * c_art + e * c_aro - w * l_r - pi_r - e * (n_a(-1) - n_a / r_star);

% HH's budget constraint in country b
c_bo + (p_t / e) * c_bt  - y_o - (n_b(-1) - n_b / r_star);

% firm's FOC
p_t * (1 - theta) * (1 - alpha) * l^((1 - theta) * (1 - alpha) - 1) * y_t^theta -
          w - phi * (p_t * y_t^theta * l(-1)^(theta * (1 - alpha)) /
                (p_t(-1) * l^(theta * (1 - alpha)) * y_t(-1)^theta) - 1) *
                          p_t * y_t * (-theta) * (1 - alpha) * p_t * y_t^theta *
                                l(-1)^(theta * (1 - alpha)) / (p_t(-1) *
                                      l^(theta * (1 - alpha) + 1) * y_t(-1)^theta);

% firm's profit
p_t * l^(1 - alpha) - w * l - pi;

% profit distribution
chi * pi_r + (1 - chi) * pi_h - pi;

% impose profit for ricardians
pi_r - pi;

% tradeable goods market-clearing condition
y_t - (1 - chi) * c_aht - chi * c_art - c_bt;

% oil market-clearing condition
y_o - (1 - chi) * c_aho - chi * c_aro - c_bo;

% labour market-clearing condition
chi * l_r + (1 - chi) * l_h - l;

% bond market clearing condition
chi * n_a + n_b;

% oil endowment, stochastic process
gamma + rho * y_o(-1) - err_yo - y_o;

% monetary policy rule
p_t - p_t(-1);

end;


%----------------------------------------------------------------
% 4. Steady state
%----------------------------------------------------------------


initval;

```

```
l_r = .22;
l_h = .22;
c_aro = .46;
c_aho = .46;
e = 1;
l = chi * l_r + (1 - chi) * l_h;
y_t = l^(1 - alpha); % from production function
p_t = 16.75 / y_t; % from target GDP in country a
c_art = s_t^eta * e^eta * c_aro / (p_t^eta * (1 - s_t)^eta); % from tradeable-oil tradeoff
c_aht = s_t^eta * e^eta * c_aho / (p_t^eta * (1 - s_t)^eta);
c_bt = y_t - chi * c_art - (1 - chi) * c_aht; % from tradeable MC
y_o = gamma / (1 - rho); % from oil stochastic process
c_bo = y_o - chi * c_aro - (1 - chi) * c_aho; % from oil MC
w = (1 - theta) * (1 - alpha) * p_t *
          l^((1 - theta) * (1 - alpha) - 1) * y_t^theta; % from firm's FOC at SS
pi = p_t * l^(1 - alpha) - w * l; % from firm's profit at SS
pi_r = pi;
pi_h = (pi - chi * pi_r) / (1 - chi);
r_star = 1 / beta; % from SS Euler
n_a = r_star / ((r_star - 1) * e) * (p_t * c_art + e * c_aro - w * l - pi_r); % from the BC
n_b = - n_a; % from bond MC


end;

steady(maxit = 1000);




%------------------------------------------------------------------
% 5. Impulse response function
%------------------------------------------------------------------

shocks;

var err_yo;
stderr .1;

end;


stoch_simul(order=1, irf=100, solve_algo = 3, irf_plot_threshold=0)
                      c_aho c_aro c_bo c_aht c_art c_bt l_h
                             l_r l y_t y_o p_t e w pi n_a n_b r_star;


model_diagnostics;
```

Appendix H: TANBFPNH mod file

```

%--------------------------------------------------------------------------
%
% Title: International Macro-Finance Final Assignment, model file
% Author: ---
% Date: 25/12/2023
% Description: Two agent, nominal bond, flexible prices,
%                                          non-homothetic prefences
%
%--------------------------------------------------------------------------


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

close all;

%--------------------------------------------------------------------------
% 1. Defining variables
%--------------------------------------------------------------------------

var c_aho c_aro c_bo c_aht c_art c_bt l_h l_r l y_t y_o p_t w pi
          n_a n_b r_star pi_r pi_h;

varexo err_yo;


%--------------------------------------------------------------------------
% 2. Calibration
%--------------------------------------------------------------------------

parameters alpha beta psi eta s_art s_aht s_bt theta rho gamma chi;
load parameters;

set_param_value('alpha'   ,par.alpha);
set_param_value('beta'    ,par.beta);
set_param_value('psi'     ,par.psi);
set_param_value('eta'     ,par.eta);
set_param_value('theta'   ,par.theta);
set_param_value('rho'     ,par.rho);
set_param_value('gamma'   ,par.gamma);
set_param_value('chi'     ,par.chi);

chi = .7;
s_t = .8;
s_aht = .75; % home bias towards tradeable for htms
s_art = (s_t - (1 - chi) * s_aht) / chi; %  home bias towards tradeable for ricardians
s_bt  = s_t; % home bias towards tradeable in country b
theta = .68;


%--------------------------------------------------------------------------
% 3. Model
%--------------------------------------------------------------------------


model;

% consumption allocation between oil and tradeable for htm HHs in country a
(1 - s_aht) * c_aho^(-1 / eta) - s_aht * c_aht^(-1 / eta) / p_t;

% consumption allocation between oil and tradeable for ricardian HHs in country a
(1 - s_art) * c_aro^(-1 / eta) - s_art * c_art^(-1 / eta) / p_t;

% consumption allocation between oil and tradeable in country b
(1 - s_bt) * c_bo^(-1 / eta) - s_bt * c_bt^(-1 / eta) / p_t;

% Euler equation for ricardian HHs in country a
c_aro^(-1 / eta) / (s_art * c_art^(1 - 1 / eta) +
     (1 - s_art) * c_aro^(1 - 1 / eta)) -
          beta * r_star *
                  c_aro(+1)^(-1 / eta) / (s_art * c_art(+1)^(1 - 1 / eta) +
```

```
                            (1 - s_art) * c_aro(+1)^(1 - 1 / eta));

% Euler equation in country b
c_bo^(-1 / eta) / (s_bt * c_bt^(1 - 1 / eta) +
     (1 - s_bt) * c_bo^(1 - 1 / eta)) -
       beta * r_star *
                  c_bo(+1)^(-1 / eta) / (s_bt * c_bt(+1)^(1 - 1 / eta) +
                            (1 - s_bt) * c_bo(+1)^(1 - 1 / eta));

% labour supply decision for htm HHs in country a
psi - (1 - s_aht) * c_aho^(-1 / eta) / (s_aht * c_aht^(1 - 1 / eta) +
                       (1 - s_aht) * c_aho^(1 - 1 / eta)) *
                                 w * (1 - l_h);

% labour supply decision for ricardian HHs in country a
psi - (1 - s_art) * c_aro^(-1 / eta) / (s_art * c_art^(1 - 1 / eta) +
                       (1 - s_art) * c_aro^(1 - 1 / eta)) *
                                 w * (1 - l_r);

% htm's budget constraint in country a
p_t * c_aht + c_aho - w * l_h - pi_h;

% ricardian's budget constraint in country a
p_t * c_art + c_aro - w * l_r - pi_r - (n_a(-1) - n_a / r_star);

% HH's budget constraint in country b
c_bo + p_t * c_bt  - y_o - (n_b(-1) - n_b / r_star);

% firm's FOC
(1 - theta) * (1 - alpha) * p_t *
          l^((1 - theta) * (1 - alpha) - 1) * y_t^theta - w;

% firm's profit
p_t * l^(1 - alpha) - w * l - pi;

% profit distribution
chi * pi_r + (1 - chi) * pi_h - pi;

% impose profit for ricardians
pi_r - pi;

% tradeable goods market-clearing condition
y_t - (1 - chi) * c_aht - chi * c_art - c_bt;

% oil market-clearing condition
y_o - (1 - chi) * c_aho - chi * c_aro - c_bo;

% labour market-clearing condition
chi * l_r + (1 - chi) * l_h - l;

% bond market clearing condition
chi * n_a + n_b;

% oil endowment, stochastic process
gamma + rho * y_o(-1) - err_yo - y_o;

end;


%----------------------------------------------------------------
% 4. Steady state
%----------------------------------------------------------------


initval;

l_r = .22;
l_h = .22;
c_aro = .46;
c_aho = .46;
l = chi * l_r + (1 - chi) * l_h;
y_t = l^(1 - alpha); % from production function
p_t = 16.75 / y_t; % from target GDP in country a
c_art = s_art^eta * c_aro / (p_t^eta * (1 - s_art)^eta); % from tradeable-oil tradeoff
```

```
c_aht = s_aht^eta * c_aho / (p_t^eta * (1 - s_aht)^eta);
c_bt = y_t - chi * c_art - (1 - chi) * c_aht; % from tradeable MC
y_o = gamma / (1 - rho); % from oil stochastic process
c_bo = y_o - chi * c_aro - (1 - chi) * c_aho; % from oil MC
w = (1 - theta) * (1 - alpha) * p_t *
          l^((1 - theta) * (1 - alpha) - 1) * y_t^theta; % from firm's FOC
pi = p_t * l^(1 - alpha) - w * l; % from firm's profit
pi_r = pi;
pi_h = (pi - chi * pi_r) / (1 - chi);
r_star = 1 / beta; % from SS Euler
n_a = r_star / (r_star - 1) * (p_t * c_art + c_aro - w * l - pi_r); % from the BC
n_b = - n_a; % from bond MC

end;

steady(maxit = 1000, solve_algo = 3);



%----------------------------------------------------------------
% 5. Impulse response function
%----------------------------------------------------------------


shocks;

var err_yo;
stderr .1;

end;


stoch_simul(order=1, irf=100, irf_plot_threshold=0)
                      c_aho c_aro c_bo c_aht c_art c_bt l_h
                             l_r l y_t y_o p_t w pi n_a n_b r_star;


model_diagnostics;
```