

# Problem Set 1

1068576

October 2023

Q1.

The social planner's problem is:

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

subject to:

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

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:

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

The FOCs with respect to  $x_t$  and  $\pi_t$  are, respectively:

$$x_t + k \cdot \mu_t = 0 \quad (4)$$

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

We obtain the optimal policy by combining them:

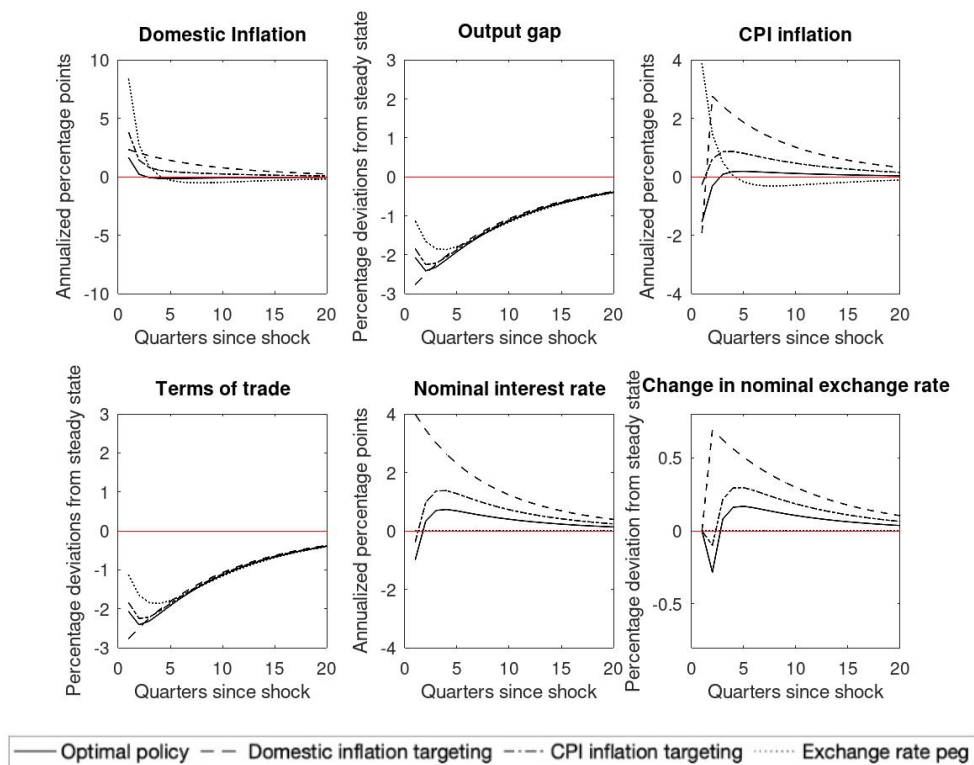
$$\kappa \lambda_{\pi} \pi_{h,t} = -(x_t - x_{t-1}) \quad (6)$$

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

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

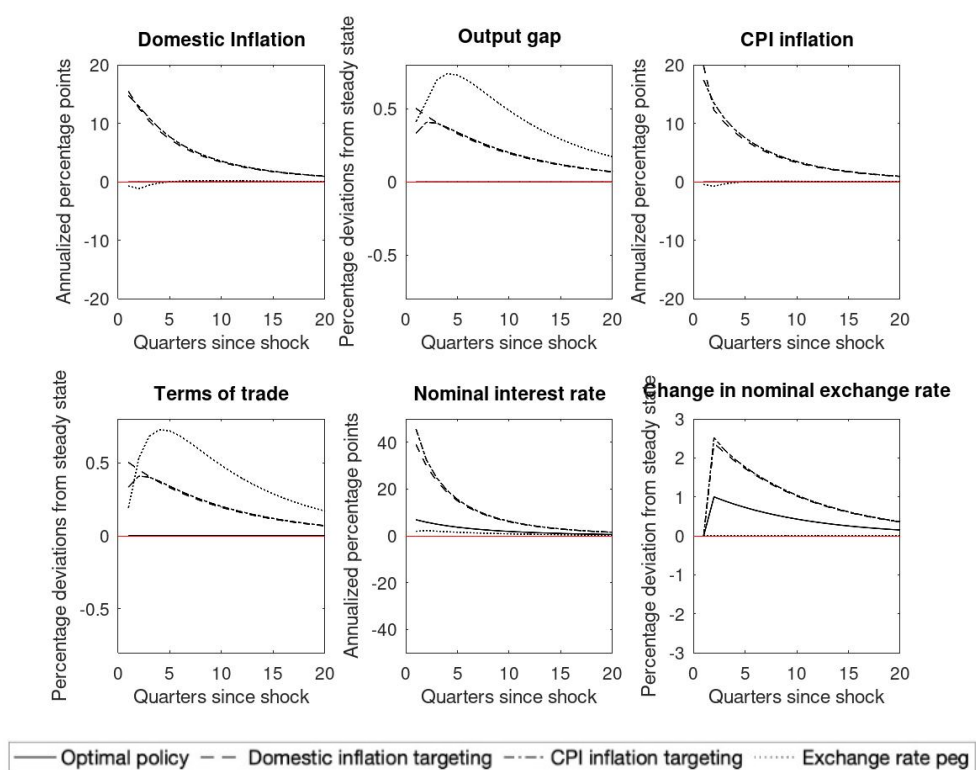
The optimal targeting rule sets the target inflation in the opposite direction to the change in the output gap.

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

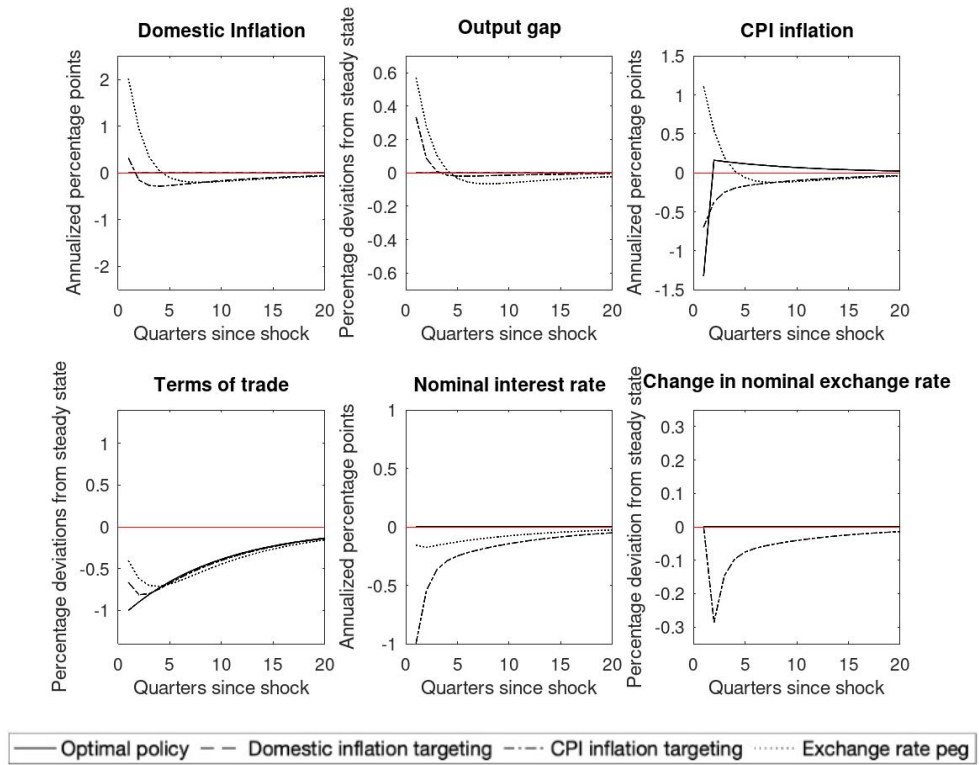


Q2.

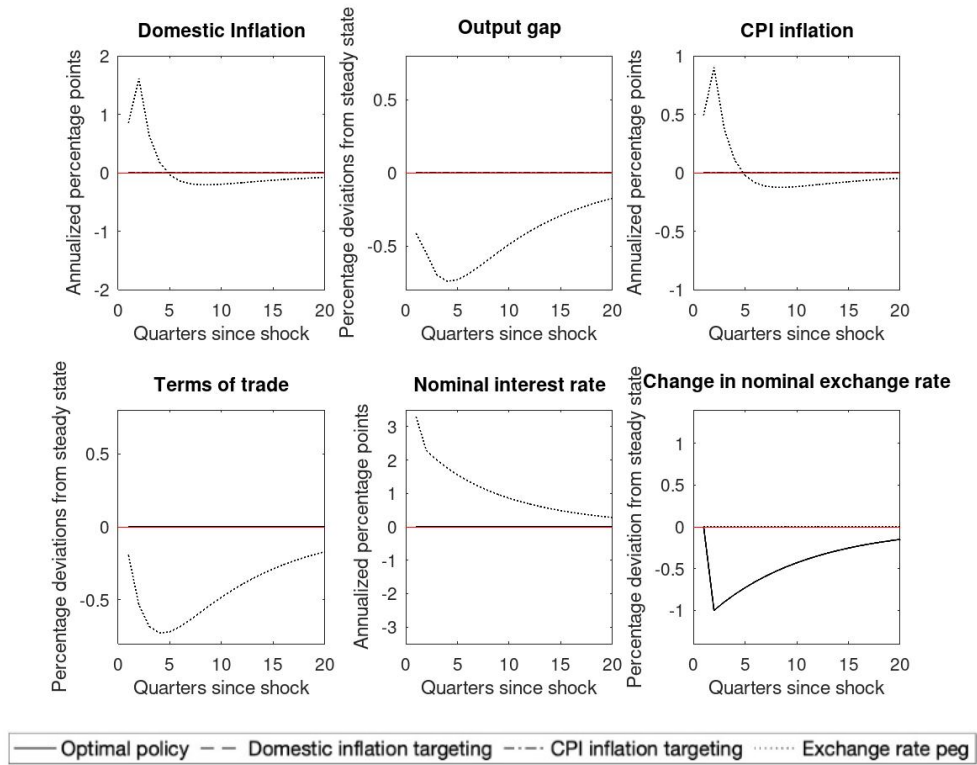
Impulse response functions for a unit standard deviation technology shock ( $r_t^n$ )



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:

$$\pi_{h,t} = \kappa x_t + \beta E_t (\pi_{h,t+1}) + u_t \quad (8)$$

$$x_t = x_{t+1} - (i_t - \pi_{h,t+1} - r_t^n) \quad (9)$$

$$\pi_t = \pi_{h,t} + \alpha(\tau_t - \tau_{t-1}) \quad (10)$$

$$x_t = z_t + \tau_t \quad (11)$$

$$i_t = i_t^* + \Delta e_t \quad (12)$$

$$(13)$$

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

$$x_{ss} = 0$$

$$\tau_{ss} = 0$$

$$\pi_{ss} = 0$$

$$\pi_{h,ss} = 0$$

$$i_{ss} = 0$$

$$\Delta e_{ss} = 0$$

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:

$$u_t = \rho u_{t-1} + \epsilon_{u,t}$$

$$r_t = \rho r_{t-1} + \epsilon_{r,t}$$

$$z_t = \rho z_{t-1} + \epsilon_{z,t}$$

$$i_t^* = \rho i_{t-1}^* + \epsilon_{i^*,t}$$

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:

$$i_t = \phi_{\pi} \pi_{H,t} \quad (14)$$

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:

$$i_t = \phi_{\pi} \pi_t \quad (15)$$

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,  $\phi_{pi}$ , 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:

$$i_t = i_t^* \quad (16)$$

where  $i_t$  is now an exogenous process. Therefore,  $\pi_t$  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:

$$\tau_t = \frac{P_{h,t}^*}{P_t} \quad (17)$$

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

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

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

$$\tau_t = \tau_{t-1} - \pi_{h,t} \quad (19)$$

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:

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

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

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

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:

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

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,  $\phi_\pi$ , 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', 'k');
    plot(citr_results.oo_.irfs.x_err_u(1:20), '-.', 'Linewidth', 1, 'Color', 'k');
    plot(peg2_results.oo_.irfs.x_err_u(1:20), ':', 'Linewidth', 1, 'Color', 'k');
    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', 'k');
    hold on
    plot(ditr_results.oo_.irfs.tau_err_u(1:20), '—', 'Linewidth', 1, 'Color', 'k');
    plot(citr_results.oo_.irfs.tau_err_u(1:20), '-.', 'Linewidth', 1, 'Color', 'k');
    plot(peg2_results.oo_.irfs.tau_err_u(1:20), ':', 'Linewidth', 1, 'Color', 'k');
    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, 'Color', 'b');
        hold on
        plot(ditr_results.oo_.irfs.delta_e_err_u(1:20), '—', 'Linewidth', 1, 'Color', 'b');
        plot(citr_results.oo_.irfs.delta_e_err_u(1:20), '—', 'Linewidth', 1, 'Color', 'b');
        plot(peg2_results.oo_.irfs.delta_e_err_u(1:20), ':', 'Linewidth', 1, 'Color', 'b');
        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-interest-rate', 'Foreign-interest-rate', 'Domestic-inflation', 'Foreign-inflation'}, 'location', 'south', 'orientation', 'horizontal', 'fontsize', 10);
    % 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', 'k');
    plot(citr_results.oo_.irfs.x_err_r(1:20), '—.', 'LineWidth', 1, 'Color', 'k');
    plot(peg2_results.oo_.irfs.x_err_r(1:20), ':', 'LineWidth', 1, 'Color', 'k');
    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', 'k');
    hold on
    plot(ditr_results.oo_.irfs.tau_err_r(1:20), '—', 'LineWidth', 1, 'Color', 'k');
    plot(citr_results.oo_.irfs.tau_err_r(1:20), '—.', 'LineWidth', 1, 'Color', 'k');
    plot(peg2_results.oo_.irfs.tau_err_r(1:20), ':', 'LineWidth', 1, 'Color', 'k');
    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, 'Color', 'k');
    hold on
    plot(ditr_results.oo_.irfs.delta_e_err_r(1:20), '—', 'Linewidth', 1, 'Color', 'k');
    plot(citr_results.oo_.irfs.delta_e_err_r(1:20), '—.', 'Linewidth', 1, 'Color', 'k');
    plot(peg2_results.oo_.irfs.delta_e_err_r(1:20), ':', 'Linewidth', 1, 'Color', 'k');
    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 inflation', 'Domestic interest rate', 'Domestic technology shock'}, 'location', 'south', 'orientation', 'horizontal', 'fontsize', 10);
% 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', 'k');
    plot(citr_results.oo_.irfs.x_err_z(1:20), '—.', 'LineWidth', 1, 'Color', 'k');
    plot(peg2_results.oo_.irfs.x_err_z(1:20), ':', 'LineWidth', 1, 'Color', 'k');
    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', 'k');
    hold on
    plot(ditr_results.oo_.irfs.tau_err_z(1:20), '—', 'LineWidth', 1, 'Color', 'k');
    plot(citr_results.oo_.irfs.tau_err_z(1:20), '—.', 'LineWidth', 1, 'Color', 'k');
    plot(peg2_results.oo_.irfs.tau_err_z(1:20), ':', 'LineWidth', 1, 'Color', 'k');
    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, 'Color', 'k');
    hold on
    plot(ditr_results.oo_.irfs.delta_e_err_z(1:20), '—', 'Linewidth', 1, 'Color', 'k');
    plot(citr_results.oo_.irfs.delta_e_err_z(1:20), '—.', 'Linewidth', 1, 'Color', 'k');
    plot(peg2_results.oo_.irfs.delta_e_err_z(1:20), ':', 'Linewidth', 1, 'Color', 'k');
    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 inflation', 'Domestic interest rate', 'Domestic exchange rate'}, 'location', 'south', 'orientation', 'horizontal', 'fontsize', 12);
% 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) * 4;
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) * 4;
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', 'k');
    hold on
    plot(ditr_results.oo_.irfs.x_err_istar(1:20), '—', 'LineWidth', 1, 'Color', 'k');
    plot(citr_results.oo_.irfs.x_err_istar(1:20), '—.', 'LineWidth', 1, 'Color', 'k');
    plot(peg2_results.oo_.irfs.x_err_istar(1:20), ':', 'LineWidth', 1, 'Color', 'k');
    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, 'Color', 'k');
    hold on
    plot(ditr_results.oo_.irfs.tau_err_istar(1:20), '—', 'LineWidth', 1, 'Color', 'k');
    plot(citr_results.oo_.irfs.tau_err_istar(1:20), '—.', 'LineWidth', 1, 'Color', 'k');
    plot(peg2_results.oo_.irfs.tau_err_istar(1:20), ':', 'LineWidth', 1, 'Color', 'k');
    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, 'Color', 'k');
    hold on
    plot(ditr_results.oo_.irfs.delta_e_err_istar(1:20), '—', 'LineWidth', 1, 'Color', 'k');
    plot(citr_results.oo_.irfs.delta_e_err_istar(1:20), '—.', 'LineWidth', 1, 'Color', 'k');
    plot(peg2_results.oo_.irfs.delta_e_err_istar(1:20), ':', 'LineWidth', 1, 'Color', 'k');
    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-interest-rate', 'Foreign-interest-rate', 'Domestic-credit-growth', 'Foreign-credit-growth'}, 'location', 'south', 'orientation', 'horizontal', 'fontsize', 12);
% 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, optimal_results.oo_.irfs.y_err_u);
loss_ditr_u = calculate_welfare_loss(ditr_results.oo_.irfs.x_err_u, ditr_results.oo_.irfs.y_err_u);
loss_citr_u = calculate_welfare_loss(citr_results.oo_.irfs.x_err_u, citr_results.oo_.irfs.y_err_u);
loss_peg1_u = calculate_welfare_loss(peg1_results.oo_.irfs.x_err_u, peg1_results.oo_.irfs.y_err_u);
loss_peg2_u = calculate_welfare_loss(peg2_results.oo_.irfs.x_err_u, peg2_results.oo_.irfs.y_err_u);

% shock to r
loss_optimal_r = calculate_welfare_loss(optimal_results.oo_.irfs.x_err_r, optimal_results.oo_.irfs.y_err_r);
loss_ditr_r = calculate_welfare_loss(ditr_results.oo_.irfs.x_err_r, ditr_results.oo_.irfs.y_err_r);

```

```

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, optim
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;  
lambda = ((1 - theta) * (1 - beta * theta)) / theta;  
kappa = lambda * (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:

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

subject to:

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

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

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

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

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

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

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

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

and the cost of default is defined as:

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

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

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

where we assume a risk-neutral lender, such that  $\sigma = 1$ .  $\beta^*$  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:

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

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	$\gamma$	2.0
subjective discount factor	$\beta$	0.97
prob. of good state	$\pi$	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.



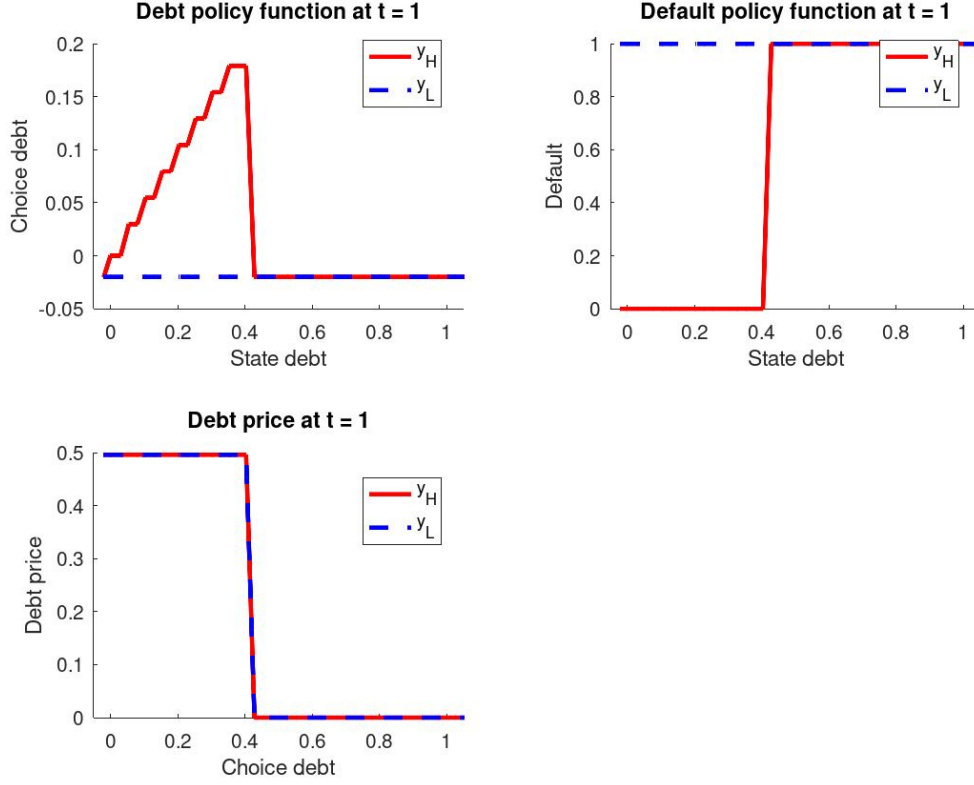


Figure 1: Policy functions and debt price

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.

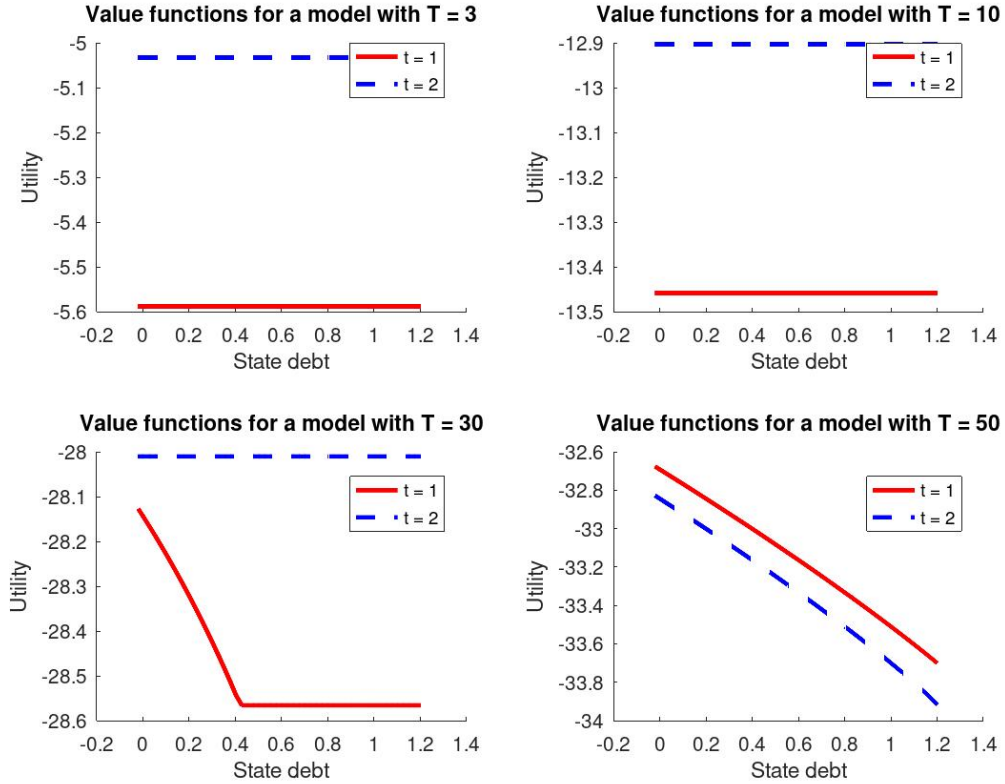


Figure 2: Value function at  $t = 1$  and  $t = 2$  for various time horizons

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.

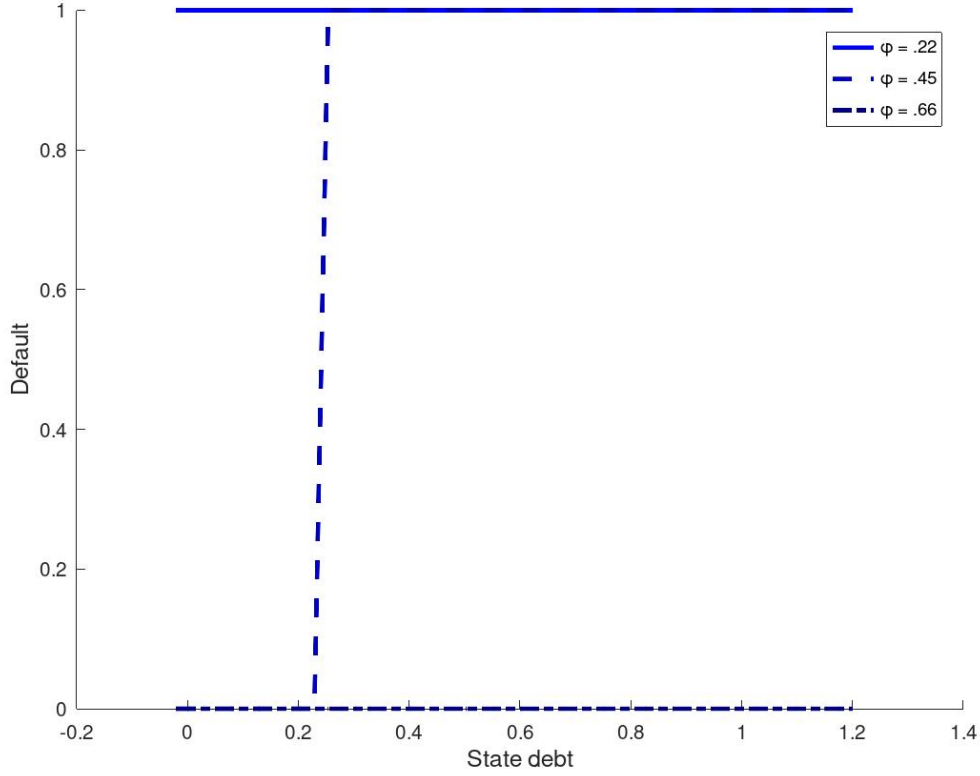


Figure 3: Default policy function given various default costs

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.

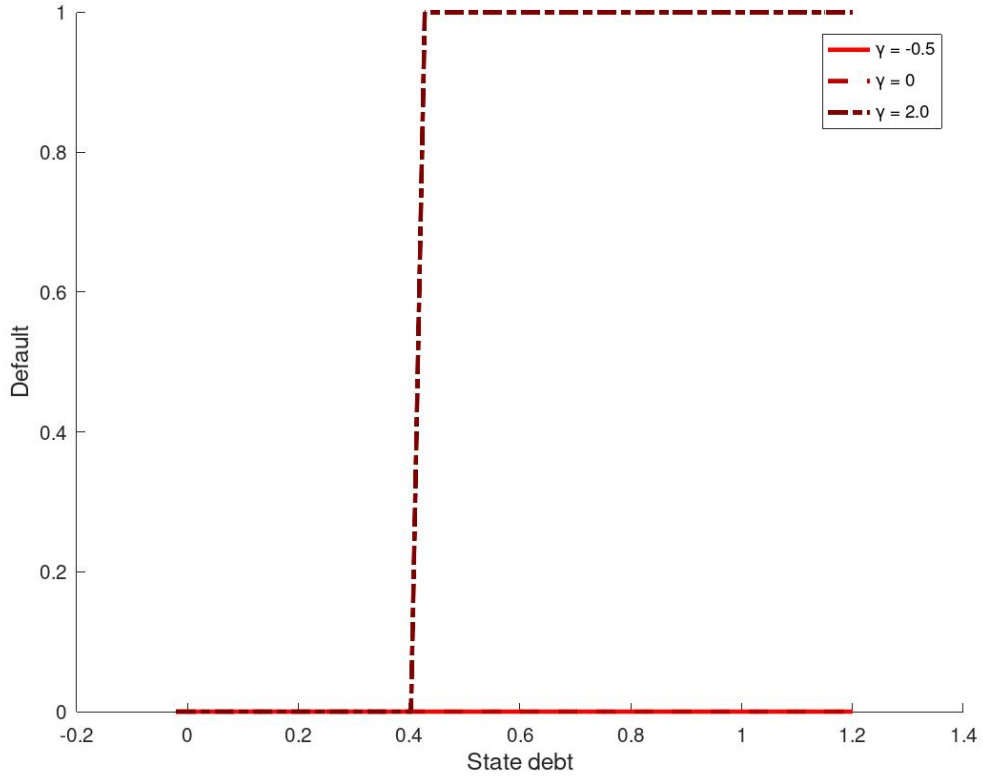


Figure 4: Default policy function given various borrower risk aversions

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.

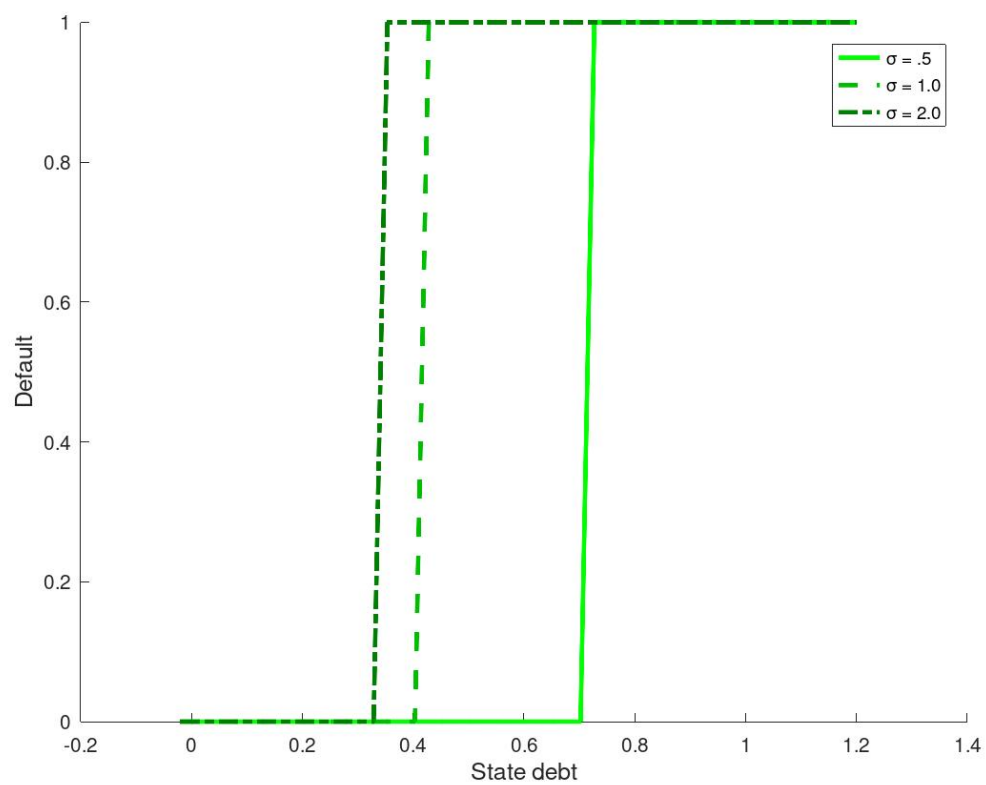


Figure 5: Default policy function given various lender risk aversions

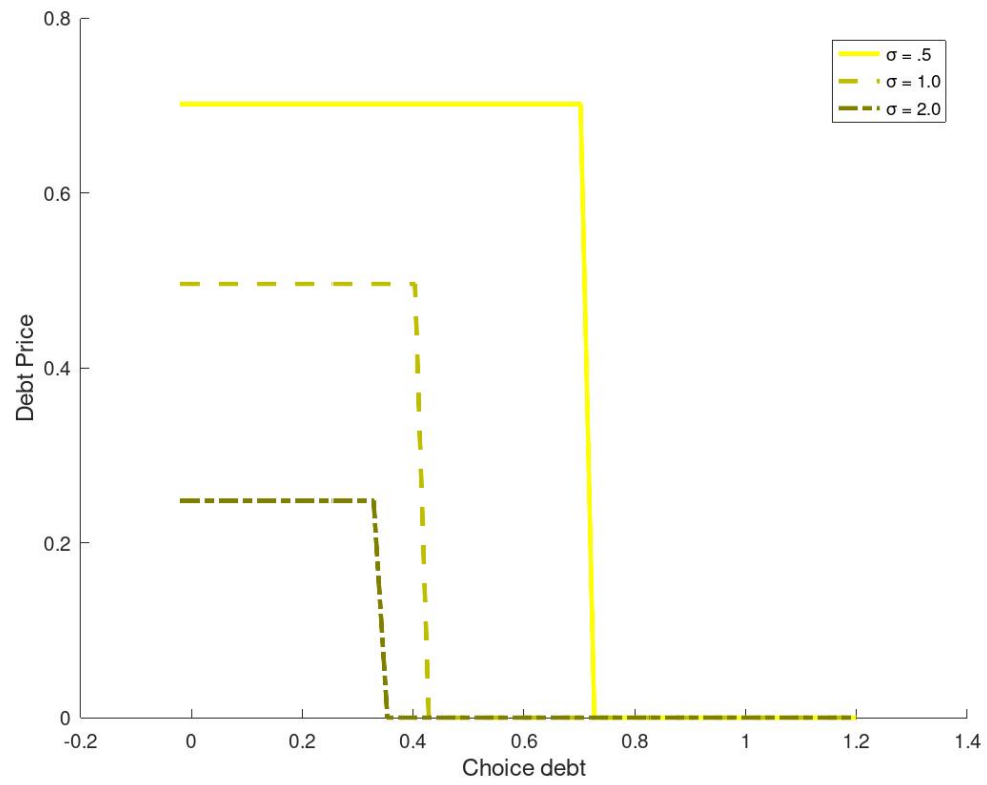


Figure 6: Price of debt given various lender risk aversions

# Appendix A: Code

## Main file

```
1
2
3 %-----
4 %
5 % Title: International Macro-Finance Problem Set 2, main file
6 % Author: Rinto Fujimoto
7 % Date: 25/11/2023
8 % Description: Sovereign default model with T periods
9 %
10 %-----
11
12 %-----
13 % 0. Housekeeping (close all graphic windows)
14 %-----
15
16 close all;
17 clear all;
18
19 cd '/home/rinto/Desktop/International Macro/PS2'
20
21 %-----
22 % 1. Defining parameters, grid and variables
23 %-----
24
25 % parameters:
26 par.gamma = 2 % household's RRA preference
27 par.beta = .97; % subjective discount factor
28 par.pi = .5; % probability of being in the good state
29
30
31 % constants:
32 cons.phi_hat = .5; % cost of default in the good state
33 cons.yh = 1.1; % endowment in good state
34 cons.yl = .9; % endowment in bad state
35 cons.rf = par.beta^(-1) - 1; % risk-free rate
36
37 % number of time periods:
38 t = 30;
39
40 % define convergence criterion
41 maxit_q = 1000; % max iteration to solve for q
42 maxit_v = 1000; % max iteration to solve for v
43 err_tol_q = 1e-10; % error tolerance
44 err_tol_v = 1e-10;
45 lambda_q = .5; % dampening parameter
46 lambda_v = .5;
47
48
49 % transition matrix
50 Pr = [par.pi, 1 - par.pi;
51       par.pi, 1 - par.pi];
52
53
```

```

54 % derived values:
55 qrf = (1 + repmat(cons.rf, [2, 1])).^(1/4); % quarterly risk-free rate
56
57
58 %-----
59 % 2. Defining grid
60 %-----
61
62 n = 50; % grid points for b
63
64 b_min = -.02; % min value for b
65 b_max = 1.2; % max value for b
66
67 b_vec = linspace(b_min, b_max, n); % grid for debt, 1 x n vector
68 y_vec = [cons.yh cons.yl].'; % grid for endowment, 2 x 1 vector
69
70 % Ensure presence of 0 on b-vector
71 [~, i_b_zero] = min(abs(b_vec));
72 % i_b_zero captures the index of the cell containing 0
73 b_vec(i_b_zero) = 0;
74
75 % grid:
76 grid.b_state3 = repmat(b_vec, [2, 1, n]); % state debt, 2 x n x n
77 grid.b_choice2 = repmat(b_vec, [2, 1]); % choice debt, 2 x n
78 grid.y_state3 = repmat(y_vec, [1, n, n]); % state output, 2 x n x n
79
80 grid.borr_choice2 = zeros(2, n, t); % grid which will contain debt value
81 grid.borr_choice3 = zeros(2, n, n, t);
82 % grid.borr_choice2 augmented by 1 dimension to account for choice debt
83
84
85
86 %-----
87 % 3. Setting Initial Guesses
88 %-----
89
90 guess.v = zeros(2, n, t); % value function
91 guess.v_new = zeros(2, n, t);
92
93 guess.i_b = zeros(2, n, t); % debt choice index (to be extracted from the grid)
94
95 guess.q = qrf(1, 1) * ones(2, n, t); % price of debt, 2 x n x t matrix
96 guess.def = zeros(2, n, t); % default choice
97
98
99 %-----
100 % 4. Defining functions
101 %-----
102
103
104 function u = util(c, gamma)
105 % this is the utility function (CRRA)
106 u = c.^(1 - gamma) / (1 - gamma);
107 end
108
109
110
111 %-----

```



```

112 % 5. Other variables
113 %
114
115 % storage value
116 store.v = zeros(2, n, t);
117 store.i_b = zeros(2, n, t);
118 store.q = qrf(1, 1) * ones(2, n, t);
119 store.def = zeros(2, n, t);
120
121 % policy functions
122 policy.b = zeros(2, n, t);
123 policy.def = zeros(2, n, t);
124
125 % others
126 con_choice = zeros(2, n, n, t); % consumption choice for non-default
127 util_choice = zeros(2, n, n, t); % utility choice for non-default
128 borr_maximand = zeros(2, n, n, t); % to be maximised over choice debt
129 e_def = zeros(2, n, t); % expected default prob.
130 e_v = zeros(2, n); % continuation value
131 e_v3 = zeros(2, n, n); % continuation value, augmented
132 e_v_def = zeros(2, n); % continuation value after default
133
134 % sum of discount factors (i.e.  $(1 - \beta^{T-t}) / (1 - \beta)$ )
135 discount = zeros(1, t);
136 for i = 1:t
137     discount(:, i) = par.beta * ((1 - par.beta^(i)) / (1 - par.beta)); % scalar
138 end
139 discount = flip(discount, 2);
140
141 % consumption in default
142 con_def = [cons.yh - cons.phi_hat, cons.yl].'; % 2 x 1
143 con_def3 = repmat(con_def, [1, n, n]);
144
145
146 % Expected value of exclusion state
147 e_v_def = discount.' * Pr(1, :) * util(con_def, par.gamma); % t x 1
148
149 % evaluate value function in default state
150 v_def = repmat(util(con_def, par.gamma), [1, n, t]) + permute(repmat(e_v_def, [1, n,
    2]), [3, 2, 1]);
151 % 2 x n x t
152
153
154
155 %
156 % 5. Value function iteration
157 %
158
159
160 err_q = 7;
161 err_v = 7;
162 iter_q = 1;
163 iter_v = 1;
164
165
166
167 while err_q > err_tol_q && iter_q < maxit_q
168

```

```

169 while err_v > err_tol_v && iter_v < maxit_v
170
171     for i = 1:t
172
173         % Expected continuation value
174         e_v = Pr * guess.v(:, :, i);
175         e_v3 = permute(repmat(e_v(:, :), [1, 1, n]), [1, 3, 2]); % 2 x n x n matrix
176         e_v_def = Pr * v_def(:, :, i);
177
178         optimal_choice = max(e_v3(:, :, i), e_v_def);
179
180         % Resources from borrowing
181         grid.borr_choice2(:, :, i) = guess.q(:, :, i) .* grid.b_choice2; % 2 x n
matrix
182         grid.borr_choice3(:, :, :, i) = repmat(grid.borr_choice2(:, :, i), [1, 1, n]);
% 2 x n x n
183         grid.borr_choice3(:, :, :, i) = permute(grid.borr_choice3(:, :, :, i), [1, 3,
2]);
184
185         if i < t
186             % Consumption implied by choice and state b and guess q
187             con_choice(:, :, :, i) = grid.y_state3 - grid.b_state3 + grid.borr_choice3
(:, :, :, i); % 2 x n x n matrix
188         else
189             % Terminal condition
190             con_choice(:, :, :, i) = max(grid.y_state3 - grid.b_state3, con_def3);
191         end
192
193         con_choice(con_choice < eps) = eps; % rule out negative consumption
194
195         % Period-utility implied by state and choice b
196         util_choice(:, :, :, i) = util(con_choice(:, :, :, i), par.gamma); % 2 x 100 x
100
197
198         % Formulate maximand in borrowing choice
199         borr_maximand(:, :, :, i) = util_choice(:, :, :, i) + par.beta * e_v3(:, :, :);
% 2 x 100 x 100
200
201         % Store old choice for borrowing
202         store.i_b(:, :, i) = guess.i_b(:, :, i); % 2 x n
203
204         % Maximise over b' and update value function
205         [guess.v_new(:, :, i), guess.i_b(:, :, i)] = max(borr_maximand(:, :, :, i),
[], 3); % 2 x n and 2 x n
206
207         % Ensures that HH cannot borrow once defaulted
208         guess.i_b(:, :, i) = guess.def(:, :, i) .* ones(2, n) * i_b_zero + (1 - guess.
def(:, :, i)) .* guess.i_b(:, :, i);
209
210     end
211
212     %Store old default policy function
213     store.def = guess.def;
214
215     % Evaluate policy function for default (indicator function)
216     guess.def = v_def > guess.v_new;
217     % compare two 2 x n x t matrices
218

```

```

219 % Evaluate value function including discrete choice for default
220 guess.v_new = max(v_def, guess.v_new);
221
222 % Store old value-function guesses
223 store.v = guess.v;
224
225 % Update value-function guesses
226 guess.v = lambda_v * guess.v_new + (1 - lambda_v) * store.v;
227
228 % Evaluate change in v and compare to error tolerance
229 err_v = max(abs(guess.v(:) - store.v(:)));
230
231 iter_v
232 err_v
233
234 iter_v = iter_v + 1;
235
236 end
237
238 % expected probability of default
239 for i = 1:t
240     e_def(:, :, i) = Pr * guess.def(:, :, i);
241 end
242
243 % Store old sovereign debt price
244 store.q = guess.q;
245
246 guess.q = qrf(1, 1) * (1 - e_def);
247
248 % Evaluate change in q-g
249 err_q = max(abs(guess.q(:) - store.q(:)));
250
251 % Update sovereign debt price
252 guess.q = lambda_q * guess.q + (1 - lambda_q) * store.q;
253
254 iter_q
255 err_q
256
257 iter_q = iter_q + 1;
258
259 % Reset counter and diff for inner-v loop
260 err_v = 7;
261 iter_v = 1;
262
263 end
264
265
266 policy.b = repmat(grid.b_choice2, [1, 1, t]);
267 policy.b = policy.b(store.i_b);
268
269 policy.def = store.def;
270
271
272 %-----
273 % 6. Plot the value function and policy functions
274 %-----
275
276

```

```

277 % Policy functions and debt price at t = 1
278 figure
279 subplot(2, 2, 1);
280 hold on;
281 plot(b_vec, policy.b(1, :, 1), 'r-', 'LineWidth', 2.5);
282 plot(b_vec, policy.b(2, :, 1), 'b--', 'LineWidth', 2.5);
283 xlim([-0.05 1.05]);
284 xlabel('State debt'), ylabel('Choice debt');
285 title('Debt policy function at t = 1');
286 legend('y_{H}', 'y_{L}', 'Location');
287 hold off;
288 subplot(2, 2, 2);
289 hold on;
290 plot(b_vec, policy.def(1, :, 1), 'r-', 'LineWidth', 2.5);
291 plot(b_vec, policy.def(2, :, 1), 'b--', 'LineWidth', 2.5);
292 xlim([-0.05 1.05]);
293 xlabel('State debt'), ylabel('Default');
294 title('Default policy function at t = 1');
295 legend('y_{H}', 'y_{L}', 'Location');
296 hold off;
297 subplot(2, 2, 3);
298 hold on;
299 plot(b_vec, store.q(1, :, 1), 'r-', 'LineWidth', 2.5);
300 plot(b_vec, store.q(2, :, 1), 'b--', 'LineWidth', 2.5);
301 xlim([-0.05 1.05]);
302 xlabel('Choice debt'), ylabel('Debt price');
303 title('Debt price at t = 1');
304 legend('y_{H}', 'y_{L}', 'Location');
305 hold off;
306
307 saveas(gcf, 'Policy_functions_Q2.jpg');
308
309
310
311 figure;
312 hold on;
313 % Value function for high output at t = 1 (solid red line)
314 plot(b_vec, store.v(1, :, 1), 'r', 'LineWidth', 2.5);
315 % Value function for high output at t = 2 (solid blue line)
316 plot(b_vec, store.v(1, :, 2), 'b', 'LineWidth', 2.5);
317 % Value function for low output at t = 1 (dashed red line)
318 plot(b_vec, store.v(2, :, 1), 'r', '--', 'LineWidth', 2.5);
319 % Value function for low output at t = 2 (dashed blue line)
320 plot(b_vec, store.v(2, :, 2), 'b', '--', 'LineWidth', 2.5);
321 xlabel('Debt levels', 'FontSize', 12);
322 ylabel('Value function', 'FontSize', 12);
323 title('Value function by time period and endowment levels', 'FontSize', 14);
324 % Adding a legend
325 legend('High endowment at t=1', 'High endowment at t=2', 'Low endowment at t=1', '
Low endowment at t=2', 'Location', 'best');
326 hold off;
327
328 saveas(gcf, 'Value_functions_Q2.jpg');

```

## Q3-Q4 Main file

```
1
2
3 %-----
4 %
5 % Title: International Macro–Finance Problem Set 2, Q4
6 % Author: Rinto Fujimoto
7 % Date: 25/11/2023
8 % Description: Sovereign default model with T periods
9 %
10 %-----
11
12
13 %-----
14 % 0. Housekeeping
15 %-----
16
17 close all;
18 clear all;
19
20
21 %-----
22 % 1. Defining functions
23 %-----
24
25 function u = util(c, gamma)
26 % this is the utility function (CRRA)
27 u = c.^(1 - gamma) / (1 - gamma);
28 end
29
30
31
32 function [value, default, debt_price, b_vec] = model(time, gamma, sigma, phi_hat)
33 % This function performs value function iteration for different
34 % parameters.
35 % Inputs: time horizon, borrower's risk aversion, lender's risk aversion, cost of
36 % default
37 % Outputs: value function, default policy function, debt price, grid for b
38 % Note: sigma appears in the debt pricing equation
39
40 par.gamma = gamma;
41 par.beta = .97;
42 par.pi = .5;
43
44 cons.phi_hat = phi_hat;
45 cons.yh = 1.1;
46 cons.yl = .9;
47 cons.rf = par.beta^(-1) - 1;
48
49 t = time;
50
51 maxit_q = 1000;
52 maxit_v = 1000;
53 err_tol_q = 1e-10;
54 err_tol_v = 1e-10;
55 lambda_v = .5;
```

```

56 lambda_q = .5;
57
58 Pr = [par.pi, 1 - par.pi;
59       par.pi, 1 - par.pi];
60
61 qrf = (1 + repmat(cons.rf, [2, 1])).^(1/4);
62
63 n = 50;
64
65 b_min = -.02;
66 b_max = 1.2;
67
68 b_vec = linspace(b_min, b_max, n);
69 y_vec = [cons.yh cons.yl].';
70
71 [~, i_b_zero] = min(abs(b_vec));
72 b_vec(i_b_zero) = 0;
73
74 grid.b_state3 = repmat(b_vec, [2, 1, n]);
75 grid.b_choice2 = repmat(b_vec, [2, 1]);
76 grid.y_state3 = repmat(y_vec, [1, n, n]);
77
78 grid.borr_choice2 = zeros(2, n, t);
79 grid.borr_choice3 = zeros(2, n, n, t);
80
81 guess.v = zeros(2, n, t);
82 guess.v_new = zeros(2, n, t);
83
84 guess.i_b = zeros(2, n, t);
85
86 guess.q = qrf(1, 1) * ones(2, n, t);
87 guess.def = zeros(2, n, t);
88
89 store.v = zeros(2, n, t);
90 store.i_b = zeros(2, n, t);
91 store.q = qrf(1, 1) * ones(2, n, t);
92 store.def = zeros(2, n, t);
93
94 policy.b = zeros(2, n, t);
95 policy.def = zeros(2, n, t);
96
97 con_choice = zeros(2, n, n, t);
98 util_choice = zeros(2, n, n, t);
99 borr_maximand = zeros(2, n, n, t);
100 e_def = zeros(2, n, t);
101 e_v = zeros(2, n);
102 e_v3 = zeros(2, n, n);
103 e_v_def = zeros(2, n);
104
105 discount = zeros(1, t);
106 for i = 1:t
107     discount(:, i) = par.beta * ((1 - par.beta^(i)) / (1 - par.beta));
108 end
109 discount = flip(discount, 2);
110
111 con_def = [cons.yh - cons.phi_hat, cons.yl].';
112 con_def3 = repmat(con_def, [1, n, n]);
113

```

```

114 e_v_def = discount.' * Pr(1, :) * util(con_def, par.gamma);
115
116 v_def = repmat(util(con_def, par.gamma), [1, n, t]) + permute(repmat(e_v_def, [1,
    n, 2]), [3, 2, 1]);
117
118 err_q = 7;
119 err_v = 7;
120 iter_q = 1;
121 iter_v = 1;
122
123
124
125 while err_q > err_tol_q && iter_q < maxit_q
126
127     while err_v > err_tol_v && iter_v < maxit_v
128
129         for i = 1:t
130
131             e_v = Pr * guess.v(:, :, i);
132             e_v3 = permute(repmat(e_v(:, :), [1, 1, n]), [1, 3, 2]);
133             e_v_def = Pr * v_def(:, :, i);
134
135             optimal_choice = max(e_v3(:, :, i), e_v_def);
136
137             grid.borr_choice2(:, :, i) = guess.q(:, :, i) .* grid.b_choice2;
138             grid.borr_choice3(:, :, :, i) = repmat(grid.borr_choice2(:, :, i), [1, 1, n
139 ]);
140             grid.borr_choice3(:, :, :, i) = permute(grid.borr_choice3(:, :, :, i), [1,
141 3, 2]);
142
143             if i < t
144                 con_choice(:, :, :, i) = grid.y_state3 - grid.b_state3 + grid.borr_choice3
145 (:, :, :, i);
146             else
147                 con_choice(:, :, :, i) = max(grid.y_state3 - grid.b_state3, con_def3);
148             end
149
150             con_choice(con_choice < eps) = eps;
151
152             util_choice(:, :, :, i) = util(con_choice(:, :, :, i), par.gamma);
153
154             borr_maximand(:, :, :, i) = util_choice(:, :, :, i) + par.beta * e_v3(:, :,
155 :);
156
157             store.i_b(:, :, i) = guess.i_b(:, :, i);
158
159             [guess.v_new(:, :, i), guess.i_b(:, :, i)] = max(borr_maximand(:, :, :, i),
160 [], 3); % 2 x n and 2 x n
161
162             guess.i_b(:, :, i) = guess.def(:, :, i) .* ones(2, n) * i_b_zero + (1 -
163 guess.def(:, :, i)) .* guess.i_b(:, :, i);
164
165         end
166
167         store.def = guess.def;
168
169         guess.def = v_def > guess.v_new;

```

```

165     guess.v_new = max(v_def, guess.v_new);
166
167     store.v = guess.v;
168
169     guess.v = lambda_v * guess.v_new + (1 - lambda_v) * store.v;
170
171     err_v = max(abs(guess.v(:) - store.v(:)));
172
173     iter_v
174     err_v
175
176     iter_v = iter_v + 1;
177
178 end
179
180     for i = 1:t
181         e_def(:, :, i) = Pr * guess.def(:, :, i);
182     end
183
184     store.q = guess.q;
185
186     % sigma is the lender's risk aversion
187     guess.q = qrf(1, 1) * (1 - e_def).^sigma;
188
189     err_q = max(abs(guess.q(:) - store.q(:)));
190
191     guess.q = lambda_q * guess.q + (1 - lambda_q) * store.q;
192
193     iter_q
194     err_q
195
196     iter_q = iter_q + 1;
197
198     err_v = 7;
199     iter_v = 1;
200
201 end
202
203
204 policy.b = repmat(grid.b_choice2, [1, 1, t]);
205 policy.b = policy.b(store.i_b);
206 policy.def = store.def;
207
208 default = policy.def;
209 debt_price = store.q;
210 value = store.v;
211
212 end
213
214
215
216 %
217 % 2. Running functions with different parameters
218 %
219
220 [value1, ~, ~, b_vec] = model(20, 2, 1.0, .5);
221 value2 = model(30, 2, 1.0, .5);
222 value3 = model(40, 2, 1.0, .5);

```



```

223 value4 = model(50, 2, 1.0, .5);
224
225 [~, default1] = model(30, -.5, 1.0, .5);
226 [~, default2] = model(30, 0, 1.0, .5);
227 [~, default3] = model(30, 2.0, 1.0, .5);
228
229 [~, default4, debt_price1] = model(30, 2, .5, .5);
230 [~, default5, debt_price2] = model(30, 2, 1.0, .5);
231 [~, default6, debt_price3] = model(30, 2, 2.0, .5);
232
233 [~, default7] = model(30, 2, 1.0, .22);
234 [~, default8] = model(30, 2, 1.0, .45);
235 [~, default9] = model(30, 2, 1.0, .66);
236
237
238
239 %
240 % 3. Plot the value function and policy functions
241 %
242
243
244 % Value function at t=1 and t=2 for various time horizons
245 figure
246     subplot(2, 2, 1);
247     hold on;
248     plot(b_vec, value1(1, :, 1), 'r-', 'LineWidth', 2.5);
249     plot(b_vec, value1(2, :, 1), 'b--', 'LineWidth', 2.5);
250     xlabel('State debt'), ylabel('Utility');
251     title('Value functions for a model with T = 3');
252     legend('t = 1', 't = 2');
253     hold off;
254     subplot(2, 2, 2);
255     hold on;
256     plot(b_vec, value2(1, :, 1), 'r-', 'LineWidth', 2.5);
257     plot(b_vec, value2(2, :, 1), 'b--', 'LineWidth', 2.5);
258     xlabel('State debt'), ylabel('Utility');
259     title('Value functions for a model with T = 10');
260     legend('t = 1', 't = 2');
261     hold off;
262     subplot(2, 2, 3);
263     hold on;
264     plot(b_vec, value3(1, :, 1), 'r-', 'LineWidth', 2.5);
265     plot(b_vec, value3(2, :, 1), 'b--', 'LineWidth', 2.5);
266     xlabel('State debt'), ylabel('Utility');
267     title('Value functions for a model with T = 30');
268     legend('t = 1', 't = 2');
269     hold off;
270     subplot(2, 2, 4);
271     hold on;
272     plot(b_vec, value4(1, :, 1), 'r-', 'LineWidth', 2.5);
273     plot(b_vec, value4(2, :, 1), 'b--', 'LineWidth', 2.5);
274     xlabel('State debt'), ylabel('Utility');
275     title('Value functions for a model with T = 50');
276     legend('t = 1', 't = 2');
277     hold off;
278
279     saveas(gcf, 'Value-functions-Q3.jpg');
280

```

```

281
282 figure;
283 hold on;
284 % Default policy function for risk-loving borrower (darker red)
285 plot(b_vec, default1(1, :, 1), 'Color', [1 0 0], 'LineWidth', 2.5);
286 % Default policy function for risk-neutral borrower
287 plot(b_vec, default2(1, :, 1), 'Color', [.75 0 0], '--', 'LineWidth', 2.5);
288 % Default policy function for risk-averse borrower (lighter red)
289 plot(b_vec, default3(1, :, 1), 'Color', [.5 0 0], '-.', 'LineWidth', 2.5);
290 xlabel('State debt', 'FontSize', 12);
291 ylabel('Default', 'FontSize', 12)
292 % Adding a legend
293 legend('\gamma = -0.5', '\gamma = 0', '\gamma = 2.0', 'Location', 'best');
294 hold off;
295
296 saveas(gcf, 'Default_risk_borrower-Q4.jpg');
297
298
299
300 figure;
301 hold on;
302 % Default policy function for risk-loving lender (darker green)
303 plot(b_vec, default4(1, :, 1), 'Color', [0 1 0], 'LineWidth', 2.5);
304 % Default policy function for risk-neutral lender
305 plot(b_vec, default5(1, :, 1), 'Color', [0 .75 0], '--', 'LineWidth', 2.5);
306 % Default policy function for risk-averse lender (lighter green)
307 plot(b_vec, default6(1, :, 1), 'Color', [0 .5 0], '-.', 'LineWidth', 2.5);
308 xlabel('State debt', 'FontSize', 12);
309 ylabel('Default', 'FontSize', 12);
310 % Adding a legend
311 legend('\sigma = .5', '\sigma = 1.0', '\sigma = 2.0', 'Location', 'best');
312 hold off;
313
314 saveas(gcf, 'Default_risk_lender-Q4.jpg');
315
316
317 figure;
318 hold on;
319 % Debt price for risk-loving lender (darker yellow)
320 plot(b_vec, debt_price1(1, :, 1), 'Color', [1 1 0], 'LineWidth', 2.5);
321 % Debt price for risk-neutral lender
322 plot(b_vec, debt_price2(1, :, 1), 'Color', [.75 .75 0], '--', 'LineWidth', 2.5);
323 % Debt price for risk-averse lender (lighter yellow)
324 plot(b_vec, debt_price3(1, :, 1), 'Color', [.5 .5 0], '-.', 'LineWidth', 2.5);
325 xlabel('Choice debt', 'FontSize', 12);
326 ylabel('Debt Price', 'FontSize', 12);
327 % Adding a legend
328 legend('\sigma = .5', '\sigma = 1.0', '\sigma = 2.0', 'Location', 'best');
329 hold off;
330
331 saveas(gcf, 'Debt_price_risk_lender-Q4.jpg');
332
333
334
335 figure;
336 hold on;
337 % Default policy function for low default cost (darker blue)
338 plot(b_vec, default7(1, :, 1), 'Color', [0 0 1], 'LineWidth', 2.5);

```

```

339 % Default policy function for medium default cost
340 plot(b_vec, default8(1, :, 1), 'Color', [0 0 .75], '--', 'LineWidth', 2.5);
341 % Default policy function for high default cost (lighter blue)
342 plot(b_vec, default9(1, :, 1), 'Color', [0 0 .5], '-.', 'LineWidth', 2.5);
343 xlabel('State debt', 'FontSize', 12);
344 ylabel('Default', 'FontSize', 12);
345 % Adding a legend
346 legend('\phi = .22', '\phi = .45', '\phi = .66', 'Location', 'best');
347 hold off;
348
349 saveas(gcf, 'Default_cost-Q4.jpg');

```

# IMF, Problem Set 3

1068576

17 December 2023

## Q1

The household's problem is:

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

where:

$$C_{i,t} \equiv (C_{it}^H)^{\alpha_H} (C_{i,t}^F)^{(1-\alpha_H)} \quad (2)$$

subject to the budget constraint:

$$P_t C_{it} = P_t^H C_{i,t}^H + P_t^F C_{i,t}^F = (1 - \tau_t) W_t s_{i,t} L_{it} + P_t^H B_{i,t}^H R_t - P_t^H B_{i,t+1}^H \quad (3)$$

and the borrowing constraint:

$$B_{i,t+1} \geq -\kappa \quad (4)$$

The Lagrangian is:

$$\begin{aligned} \mathcal{L} = \sum_{t=0}^{\infty} \beta^t & \left[ \log \left( (C_{it}^H)^{\alpha_H} (C_{i,t}^F)^{(1-\alpha_H)} - \frac{L_{i,t}^{1+\eta}}{1+\eta} \right) + \dots \right. \\ & \lambda_{i,t} \left( (1 - \tau_t) W_t s_{i,t} L_{it} + P_t^H B_{i,t}^H R_t - P_t^H B_{i,t+1}^H - P_t^H C_{i,t}^H - P_t^F C_{i,t}^F \right) + \dots \\ & \left. \mu_{i,t} (B_{i,t+1} + \kappa) \right] \quad (5) \end{aligned}$$

The first order conditions are:

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

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

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

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

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

with the complementary slackness condition:

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

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

$$P_t^H C_t^H = P_t^F C_t^F \quad (12)$$

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

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

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

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

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

Next, I look at the firm's problem:

$$\max_{Y_t, L_t} P_t^H Y_t - W_t L_t \quad (15)$$

given the production technology:

$$Y_t = L_t \quad (16)$$

The firm's first order condition is:

$$W_t = P_t^H \quad (17)$$

## 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:

$$C_t^{H*} = \gamma Y_t \quad (18)$$

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:

$$C_t^H = Y_t - C_t^{H*} \quad (19)$$

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:

$$C_t^F = \omega Y_t^* \quad (20)$$

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  $\tau$  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	$\beta$	0.99	
Frisch elasticity	$\eta$	3	Chetty et al. (2011)
home bias	$\alpha_H$	0.5	assumption
tax revenue (% of GDP)	$\tau$	.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
$\tau$	0.30	0.18	0.30
$g$	6.61	6.61	6.75

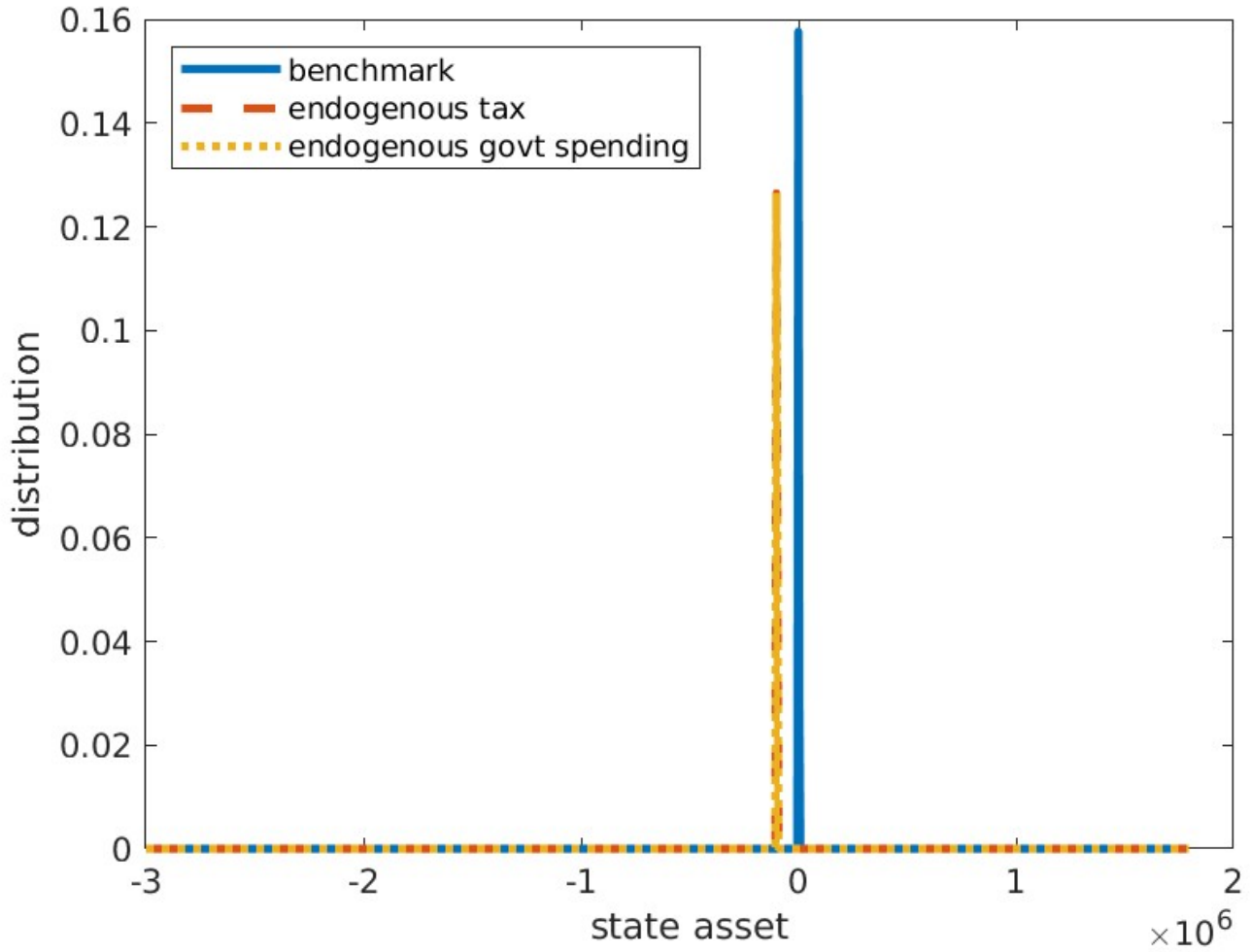


Figure 1: Utility comparative statics

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.

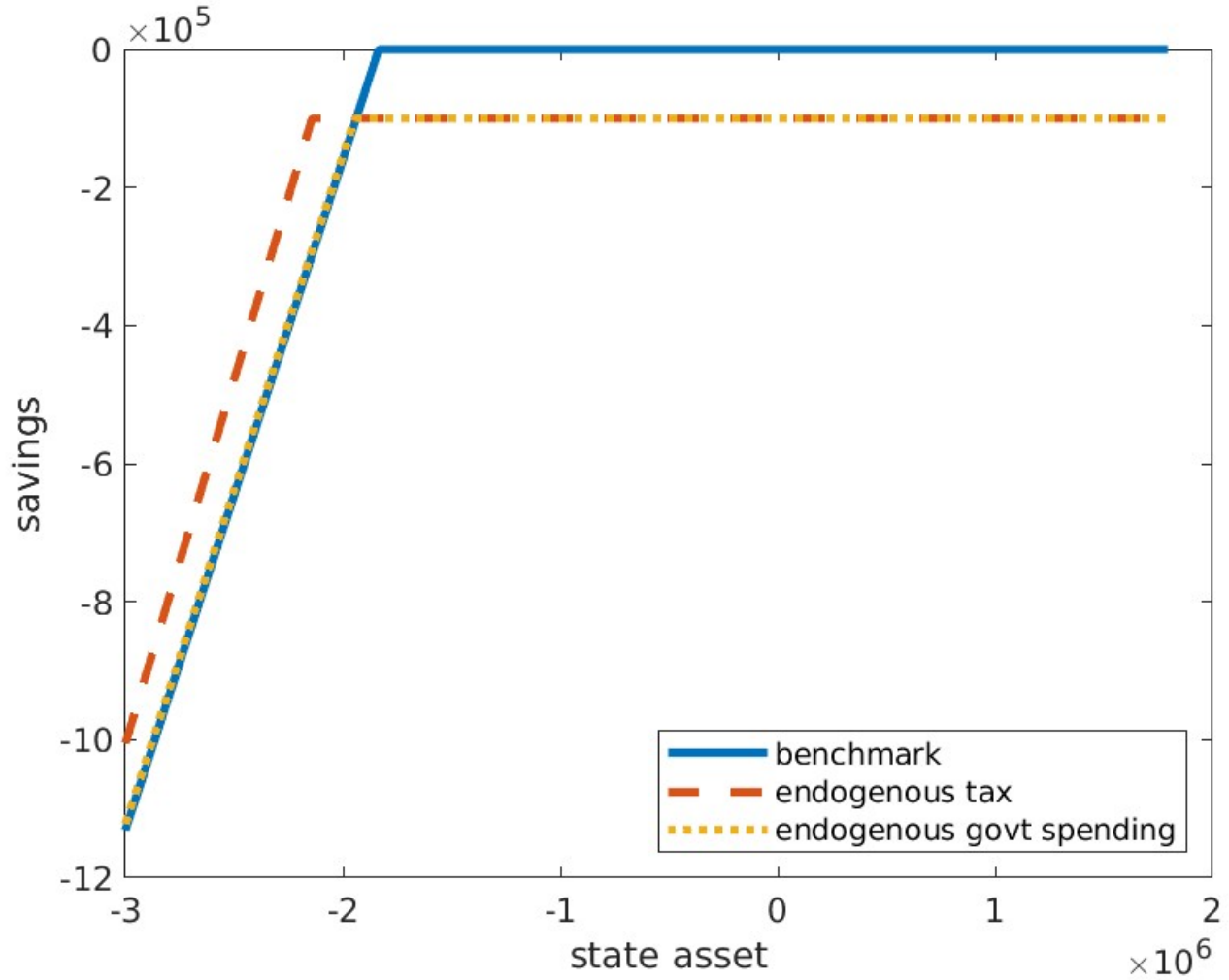


Figure 2: Savings policy function

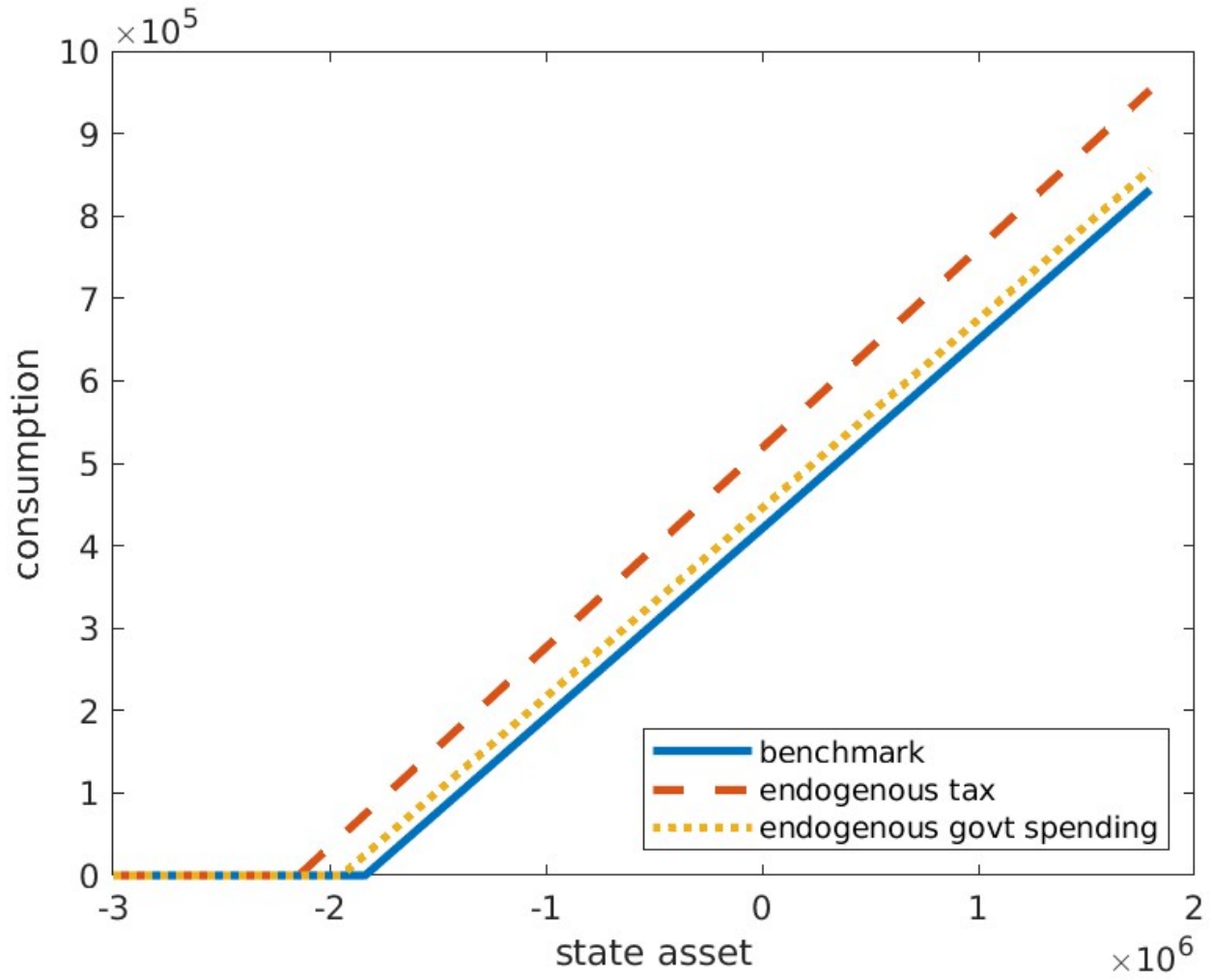


Figure 3: Consumption policy function

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.

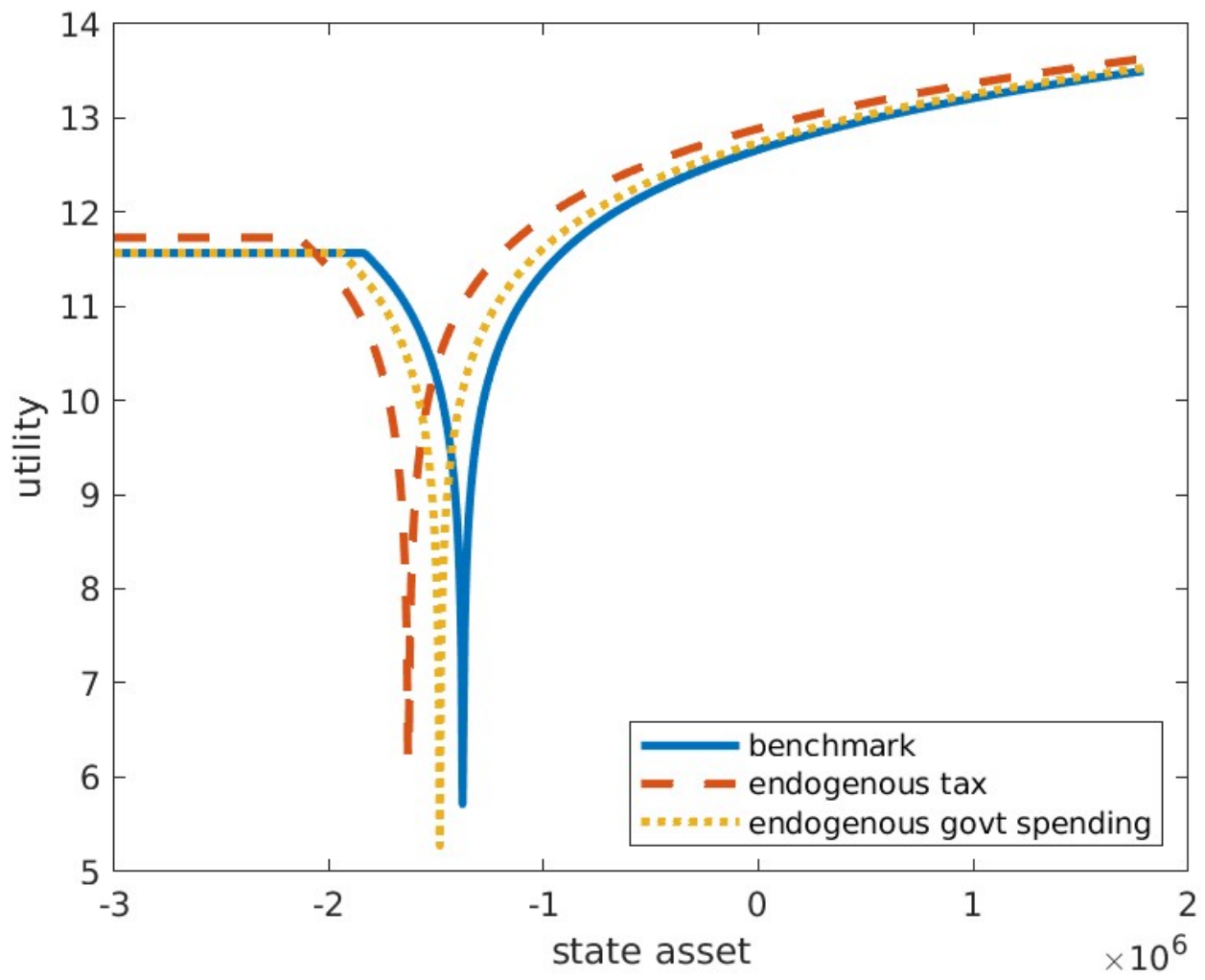


Figure 4: Utility comparative statics

# Appendix A: Main File

```
1
2
3 %
4 %
5 % Title: International Macro–Finance Problem Set 3, main file
6 % Author: —
7 % Date: 25/11/2023
8 % Description: Open economy heterogeneous agent model
9 %
10 %
11
12
13 %
14 %% 0. Housekeeping (close all graphic windows)
15 %
16
17 close all;
18 clear all;
19
20 cd '/home/—/Desktop/International Macro/PS3';
21
22 %
23 %% 1. Loading results
24 %
25
26 load("PS3Q4.mat");
27
28 eta = par.eta;
29 grid_b = grid.b_fine';
30
31 r0Q4 = guess.r0;
32 ciQ4 = policy.ci;
33 biQ4 = policy.bi;
34 G0Q4 = G0;
35 liQ4 = policy.li;
36 yQ4 = y;
37 totQ4 = guess.ph_pf;
38 tauQ4 = par.tau;
39 gQ4 = par.g;
40 bgQ4 = bg;
41 b_starQ4 = par.b_star;
42
43 clearvars -except grid_b eta *Q4
44
45
46 load("PS3Q5.mat");
47
48 r0Q5 = guess.r0;
49 ciQ5 = policy.ci;
50 biQ5 = policy.bi;
51 G0Q5 = G0;
52 liQ5 = policy.li;
53 yQ5 = y;
54 totQ5 = guess.ph_pf;
55 tauQ5 = guess.tau;
56 gQ5 = par.g;
57 bgQ5 = par.bg;
58 b_starQ5 = b_star;
59
60 clearvars -except grid_b eta *Q4 *Q5
61
62
63 load("PS3Q6.mat");
64
65 r0Q6 = guess.r0;
66 ciQ6 = policy.ci;
67 biQ6 = policy.bi;
68 G0Q6 = G0;
69 liQ6 = policy.li;
70 yQ6 = y;
71 totQ6 = guess.ph_pf;
72 tauQ6 = par.tau;
73 gQ6 = g;
```

```

74 bgQ6 = par.bg;
75 b_starQ6 = b_star;
76
77 clearvars -except grid_b eta pi *Q4 *Q5 *Q6
78
79 %-----
80 %% 2. Comparative statics
81 %-----
82
83 G0Q41 = G0Q4 * pi';
84 G0Q51 = G0Q5 * pi';
85 G0Q61 = G0Q6 * pi';
86
87
88
89 figure
90     plot(grid_b, G0Q41, '-', 'LineWidth', 2.5)
91     hold on
92     plot(grid_b, G0Q51, '-', 'LineWidth', 2.5)
93     plot(grid_b, G0Q61, ':', 'LineWidth', 2.5)
94     xlabel('state asset')
95     ylabel('distribution')
96     legend('benchmark', 'endogenous tax', 'endogenous govt spending', 'Location', 'northwest')
97
98
99 saveas(gcf, 'Distribution-Q4.jpg');
100
101
102 utilQ4 = log(ciQ4 - liQ4.^(1 + eta)/ (1 + eta));
103 utilQ41 = utilQ4 * pi';
104
105 utilQ5 = log(ciQ5 - liQ5.^(1 + eta)/ (1 + eta));
106 utilQ51 = utilQ5 * pi';
107
108 utilQ6 = log(ciQ6 - liQ6.^(1 + eta)/ (1 + eta));
109 utilQ61 = utilQ6 * pi';
110
111
112 figure
113     plot(grid_b, utilQ41, '-', 'LineWidth', 2.5)
114     hold on
115     plot(grid_b, utilQ51, '-', 'LineWidth', 2.5)
116     plot(grid_b, utilQ61, ':', 'LineWidth', 2.5)
117     % title('Utility: comparative statics')
118     xlabel('state asset')
119     ylabel('utility')
120     legend('benchmark', 'endogenous tax', 'endogenous govt spending', 'Location', 'southeast')
121
122 saveas(gcf, 'Utility-Q5Q6.jpg');
123
124 % %%
125 % figure
126 %     plot(grid_b, sum(ciQ4 .* G0Q4, 2), '-', 'LineWidth', 2.5)
127 %     hold on
128 %     plot(grid_b, sum(ciQ5 .* G0Q5, 2), '-', 'LineWidth', 2.5)
129 %     plot(grid_b, sum(ciQ6 .* G0Q6, 2), ':', 'LineWidth', 2.5)
130 %     title('Consumption Policy Function')
131 %     xlabel('state asset')
132 %     ylabel('consumption')
133 %     legend('benchmark', 'endogenous tax', 'endogenous govt spending')
134 % %%
135
136
137
138 figure
139     plot(grid_b, ciQ4(:, 1), '-', 'LineWidth', 2.5)
140     hold on
141     plot(grid_b, ciQ5(:, 1), '-', 'LineWidth', 2.5)
142     plot(grid_b, ciQ6(:, 1), ':', 'LineWidth', 2.5)
143     % title('Consumption Policy Function')
144     xlabel('state asset')
145     ylabel('consumption')
146     legend('benchmark', 'endogenous tax', 'endogenous govt spending', 'Location', 'southeast')
147

```



```

148 saveas(gcf, 'Consumption_policy-Q4Q5Q6.jpg');
149
150
151
152 figure
153     plot(grid_b, biQ4(:, 1), '-', 'LineWidth', 2.5)
154     hold on
155     plot(grid_b, biQ5(:, 1), '—', 'LineWidth', 2.5)
156     plot(grid_b, biQ6(:, 1), ':', 'LineWidth', 2.5)
157     % title('Lending Policy Function')
158     xlabel('state asset')
159     ylabel('lending')
160     legend('benchmark', 'endogenous tax', 'endogenous govt spending', 'Location', 'southeast')
161
162 saveas(gcf, 'Debt_policy-Q4Q5Q6.jpg');

```

## Appendix B: Q4 model file

```
1
2
3 %
4 %
5 % Title: International Macro–Finance Problem Set 3, Q4 model file
6 % Author: —
7 % Date: 10/12/2023
8 % Description: Open economy heterogeneous agent model
9 %
10 %
11
12
13 %
14 %% 0. Housekeeping (close all graphic windows)
15 %
16
17 close all;
18 clear all;
19
20 cd '/home/—/Desktop/International Macro/PS3';
21
22 %
23 %% 1. Calibration & defining parameters
24 %
25
26
27 % preferences
28 par.beta = .99;
29 par.eta = 3; % standard Frisch elasticity of 3 according to macro literature (Chetty et al. 2011)
30 par.alphah = .5; % no home bias at home
31
32 % fiscal policy
33 par.tau = .3; % US govt revenue as a share of gdp, 2012–22 average
34 par.g = 6.61174; % adjust govt spending until I reach a debt to GDP ratio of 120%
35
36
37 % rest of the world
38 par.y.star = 75; % world excld. US (RoW) GDP, 2022 in trillion USD (FRED)
39 par.b.star = 0; % foreign holding of home bonds
40 par.gamma = .119; % US export to RoW as share of US GDP, 2012–22 average (FRED)
41 par.omega = .047; % US import from RoW as a share of RoW GDP, 2012–22 average (FRED)
42
43 % others
44 par.kappa = 1;
45 b_ss = 138; % aggregate net worth in the US in 2022, trillion USD (FRED)
46 par.b.shareTop01 = .13; % share of net worth held by top .1% in 2022 (SCF)
47 par.b.shareBot01 = -.0007 * 5; % share of net worth held by bottom .1% in 2022 (SCF)
48
49
50 %
51 %% 2. Defining the exogenous shock process
52 %
53
54
55 % Exogeneous shock
56 % par.scale = 1;
57 % grid.s1 = [.5 1.5]; % value
58 % grid.s1 = grid.s1 * par.scale ;
59 % Pr = [.8 .2];
60 %      .2 .8]; % Markov chain
61 % ns = length(grid.s1); % number of states
62 % pi = ones(1, ns) / ns;
63 % dis = 1; % initial distribution of HHs
64 % tol=1e-20;
65 % % compute invariant distribution for s
66 % while dis>tol
67 %     pi_2 = pi * Pr;
68 %     dis = max(abs(pi_2 - pi));
69 %     pi = pi_2;
70 % end
71
72 ns      = 10;          % Number of points for labour productivity process
73 par.rhos = .995;       % Calibrate persistence of AR(1), from FRED time series
```

```

74 par.sigs = sqrt(.6);           % Calibrate standard deviation of innovation
75 par.scale = 1e+5;             % scaling factor for GDP
76 [grid.s1, Pr] = rouwenhorst(ns, 0, par.rhos, par.sigs);
77 % grid.s1 = rescale(grid.s1', .5 * par.scale, 1.5 * par.scale);
78 grid.s1 = exp(grid.s1); % labour productivity process (rule out negatives)
79
80 maxIter_pi = 1e+5;
81 Pr_ergo = Pr^maxIter_pi; % Ergodic distribution of exogenous labour productivity process
82 pi = Pr_ergo(1,:);
83 grid.s1 = (grid.s1 / (pi * grid.s1) + par.scale)'; % Normalize mean to 1 (or to 100)
84
85
86
87 %-----
88 %% 3. Setting up the grids
89 %-----
90
91 % Asset grid:
92 b_con = -par.kappa * par.scale;
93 b_min = 30 * b_con; % ensures it is significantly below b_con
94 b_max = par.b_shareTop01 * b_ss * par.scale; % approximately the net worth of the top .1%
95 nb = 1000;
96 nb_fine = 1000;
97 grid.b = linspace(b_min, b_max, nb);
98 grid.b = [grid.b, b_con]';
99 grid.b = sort(grid.b);
100 Ind_b_min = find(grid.b == b_con);
101 nb = length(grid.b); % update grid point number to account for kappa
102 grid.b_fine = linspace(b_min, b_max, nb_fine);
103
104
105 % Build some usefull matrices
106 bg = ones(nb, ns);
107 grid.b2 = repmat(grid.b, [1 ns]); % nb x ns matrix
108 grid.s2 = repmat(grid.s1, [nb 1]); % nb x 2 matrix
109
110
111 %-----
112 %% 4. Guesses & policy functions
113 %-----
114
115 % consumption
116 guess.ci = ones(nb, ns);
117
118 % terms of trade
119 guess.ph_pf = 2;
120
121 % interest rate
122 guess.r0 = 1 / par.beta;
123
124 % consumption
125 cf = par.omega * par.y_star; % home consumption of foreign goods (i.e. US import from RoW)
126
127 % distribution
128 G0 = ones(nb_fine, ns) / (nb_fine * ns);
129
130 % policy functions
131 policy.ci = ones(nb_fine, ns);
132 policy.bi = ones(nb_fine, ns);
133
134 % convergence parameters
135 maxIter_c = 1e+3;
136 maxIter_r = 1e+3;
137 maxIter_tot = 1e+3;
138 lambda = .5;
139 errtol_c = 1e-5;
140 errtol_r = 1e-2;
141 errtol_tot = 1e-5;
142
143
144 %-----
145 %% 5. Iterations
146 %-----
147

```

```

148 iter_tot = 1;
149 err_tot = 1;
150
151 while err_tot > errtol_tot && iter_tot <= maxIter_tot
152
153     w_p0 = par.alphah^par.alphah * (1 - par.alphah)^(1 - par.alphah) * ...
154         guess.ph_pf^(1 - par.alphah); % update real wage from the tot
155
156 %% Compute labour supply
157 policy.li = ((1 - par.tau) .* w_p0 .* grid.s1).^(1 / par.eta);
158 l0 = pi * policy.li.';
159
160 iter_r = 1;
161 err_r = 1;
162 %%
163
164 while err_r > errtol_r && iter_r <= maxIter_r
165
166     %% Solve for consumption
167     y = l0;
168
169     % Compute constrained consumption given R
170     c_constrained = (1 - par.tau) .* guess.ph_pf .* grid.s2 .* policy.li + ...
171         w_p0 .* grid.b2 .* guess.r0 - ...
172         w_p0 .* b_con;
173     c_constrained = max(c_constrained, 1e-5); % rule out consumption below 0
174
175     iter_c = 1;
176     err_c = 1;
177
178     guess.ci = ones(nb, ns);
179     %%
180
181     while err_c > errtol_c && iter_c <= maxIter_c
182         %%
183
184         % expected marginal utility at t+2
185         Emup1 = (guess.ci * par.scale - policy.li.^(1 + par.eta) / (1 + par.eta)).^(-1);
186         % expected marginal utility at t+1 (scale up consumption)
187         Mup = par.beta * guess.r0 * Emup1 * Pr';
188         % expected consumption at t+1
189         Ec = Mup.^(-1) + policy.li.^(1 + par.eta) / (1 + par.eta);
190         % state debt tomorrow (i.e. choice debt today)
191         bi_state = (Ec ./ w_p0 + grid.b2 - (1 - par.tau) .* grid.s2 .* ...
192             policy.li) ./ (guess.r0);
193
194         c_new = ones(nb, ns);
195
196         for j=1:ns
197
198             c_new(:,j) = (grid.b > bi_state(Ind_b_min, j)) .* ... % indicator function on whether borrowing
199                 constraint is binding
200                 interp1(bi_state(:, j), Ec(:, j), grid.b, 'pchip') + ... % interpolate c_s = f(
201                 bi_state) at each grid point
202                 (grid.b <= bi_state(Ind_b_min, j)) .* c_constrained(:,j); % if constraint is binding,
203             then c_constrained
204             c_new(:,j) = max(c_new(:,j), 1e-5); % rules out negative values
205         end
206
207         err_c = max(max(abs(c_new - guess.ci)));
208
209         guess.ci = c_new * lambda + (1 - lambda) * guess.ci;
210
211         iter_c
212         err_c
213
214         iter_c = iter_c + 1;
215         %%
216     end
217
218     % Write the policy function for consumption
219     for j=1:ns
220         policy.ci(:,j) = interp1(grid.b, guess.ci(:,j), grid.b_fine);
221     end

```

```

219 %% Solve for interest
220
221
222 % Write the policy function for assets
223 bi_choice = (grid.b2 * guess.r0 + (1 - par.tau) * grid.s2 .* policy.li - ...
224             guess.ci ./ w_p0);
225
226 for j=1:ns
227     policy.bi(:,j) = interp1(grid.b, bi_choice(:,j), grid.b_fine);
228 end
229
230 % Compute the endogenous distribution
231 trows = zeros(nb_fine * ns * ns * 2, 1);
232 tcols = trows;
233 tvals = tcols;
234 index = 0;
235 for j=1:ns
236     for bi = 1:nb_fine
237         [vals,inds] = basefun(grid.b_fine, nb_fine, policy.bi(bi,j));
238         for jp=1:ns
239             index = index + 1;
240             trows(index) = bi + (j - 1) * nb_fine;
241             tcols(index) = inds(1) + (jp - 1) * nb_fine;
242             tvals(index) = Pr(j, jp) * vals(1);
243             index = index+1;
244             trows(index) = bi + (j - 1) * nb_fine;
245             tcols(index) = inds(2) + (jp - 1) * nb_fine;
246             tvals(index) = Pr(j, jp) * vals(2);
247         end
248     end
249 end
250 transMat = sparse(trows, tcols, tvals, nb_fine * ns, nb_fine * ns);
251 [EigVec, EigVal] = eigs(transMat.', 1);
252 EigVec = EigVec / sum(EigVec);
253 EigVec(EigVec < 0) = 0;
254 EigVec = EigVec / sum(EigVec);
255 G0 = reshape(EigVec / sum(EigVec), [nb_fine ns]); % distr. of HHs across assets & states
256
257 % update guess for r
258 b = sum(sum(policy.bi .* G0)); % aggregate HH borrowing
259 bg = -(b + par.b_star); % solve for govt borrowing from bond market clearing
260 r_new = (bg + par.g - par.tau * y) / bg; % update r from govt BC
261
262 err_r = abs(r_new - guess.r0);
263
264 guess.r0 = lambda * r_new + (1 - lambda) * guess.r0;
265
266 iter_r
267 err_r
268
269 iter_r = iter_r + 1;
270 %%
271
272 end
273
274 %% Solve for terms of trade
275 ch_star = par.gamma * y; % US export to the RoW
276 ch = y - par.g - ch_star; % consumption of home goods from market clearing
277 ph_pf_new = cf / ch; % domestic price of foreign goods from the FOC
278
279 err_tot = max(max(abs(ph_pf_new - guess.ph_pf)));
280
281 guess.ph_pf = lambda * ph_pf_new + (1 - lambda) * guess.ph_pf;
282
283 iter_tot
284 err_tot
285
286 iter_tot = iter_tot + 1;
287
288 end
289
290
291 %-----
292 % 6. Export results

```

```

293 %
294
295 save("PS3Q4.mat");
296
297
298 %
299 % 7. Defining functions
300 %
301
302 function [Z,PI] = rouwenhorst(N,mu,rho,sigma)
303 % Code to approximate AR(1) process using the Rouwenhorst method as in
304 % Kopecky & Suen, Review of Economic Dynamics (2010), Vol 13, p 701–714
305 %
306 %Purpose:      Finds a Markov chain whose sample paths approximate those of
307 %              the AR(1) process
308 %               $z(t+1) = (1-\rho)*\mu + \rho * z(t) + \text{eps}(t+1)$ 
309 %              where eps are normal with stddev sigma
310 %
311 %Format:       [Z, PI] = rouwenhorst(N,mu,rho,sigma)
312 %
313 %Input:        N          scalar, number of nodes for Z
314 %              mu         scalar, unconditional mean of process
315 %              rho         scalar
316 %              sigma      scalar, std. dev. of epsilons
317 %
318 %Output:       Z          N*1 vector, nodes for Z
319 %              PI         N*N matrix, transition probabilities
320 %
321 % Code and comment by Martin Floden, Stockholm University, August 2010
322 %
323 % Comment on this method:
324 % As opposed to the methods suggested by Tauchen and Tauchen and Hussey
325 % (see M. Floden, Economic Letters, 2008, 99, 516–520), the Rouwenhorst
326 % method perfectly matches both the conditional and unconditional variances
327 % and autocorrelations of the AR(1) process. The method however tends to
328 % generate errors eps that are further away from the normal distribution
329 % than the Tauchen methods (the kurtosis of the simulated eps is too high
330 % with the Rouwenhorst method).
331
332 sigmaz = sigma / sqrt(1-rho^2);
333
334 p = (1+rho)/2;
335 PI = [p 1-p; 1-p p];
336
337 for n = 3:N
338     PI = p*[PI zeros(n-1,1); zeros(1,n)] + ...
339         (1-p)*[zeros(n-1,1) PI; zeros(1,n)] + ...
340         (1-p)*[zeros(1,n); PI zeros(n-1,1)] + ...
341         p*[zeros(1,n); zeros(n-1,1) PI];
342     PI(2:end-1,:) = PI(2:end-1,+)/2;
343 end
344
345 fi = sqrt(N-1)*sigmaz;
346 Z = linspace(-fi,fi,N)';
347 Z = Z + mu;
348
349 end
350
351
352 function [vals, inds]=basefun(grid_x,npx,x)
353 %Linear interpolation
354 jl=1;
355 ju=npx;
356 while((ju-jl>1))
357     jm=round((ju+jl)/2);
358     if(x>=grid_x(jm))
359         jl=jm;
360     else
361         ju=jm;
362     end
363 end
364
365 i=jl+1;
366 vals(2)=(x-grid_x(i-1))/(grid_x(i)-grid_x(i-1));

```

```
367     vals(2)=max(0.0d0,min(1.0d0,vals(2)));
368     vals(1)=1.0d0-vals(2);
369     inds(2)=i;
370     inds(1)=i-1;
371
372 end
```

## Appendix C: Q5 model file

```
1
2
3 %
4 %
5 % Title: International Macro–Finance Problem Set 3, Q5 model file
6 % Author: —
7 % Date: 10/12/2023
8 % Description: Open economy heterogeneous agent model
9 %
10 %
11
12
13 %
14 %% 0. Housekeeping (close all graphic windows)
15 %
16
17 close all;
18 clear all;
19
20 cd '/home/—/Desktop/International Macro/PS3';
21
22 %
23 %% 1. Retrieve parameters from Q4
24 %
25
26 load("PS3Q4.mat");
27
28
29 %
30 %% 2. Set government debt as a parameter & foreign lending
31 %
32
33 par.bg = bg;
34 clear bg;
35
36 guess.tau = 0;
37
38
39 %
40 %% 3. Iterations
41 %
42
43 iter_tot = 1;
44 err_tot = 1;
45
46 while err_tot > errtol_tot && iter_tot <= maxIter_tot
47
48     w_p0 = par.alphah^par.alphah * (1 - par.alphah)^(1 - par.alphah) * ...
49         guess.ph_pf^(1 - par.alphah); % update real wage from the tot
50
51     %% Compute labour supply
52     policy.li = ((1 - guess.tau) .* w_p0 .* grid.s1).^(1 / par.eta);
53     l0 = pi * policy.li.';
54
55     iter_r = 1;
56     err_r = 1;
57     %%
58
59     while err_r > errtol_r && iter_r <= maxIter_r
60
61         %% Solve for consumption
62         y = l0;
63
64         % Compute constrained consumption given R
65         c_constrained = (1 - guess.tau) .* w_p0 .* grid.s2 .* policy.li + ...
66             w_p0 .* grid.b2 .* guess.r0 - ...
67             w_p0 .* b_con;
68         c_constrained = max(c_constrained, 1e-5); % rule out consumption below 0
69
70         iter_c = 1;
71         err_c = 1;
72
73         guess.ci = ones(nb,ns);
```



```

74 %%
75
76 while err_c > errtol_c && iter_c <= maxIter_c
77
78
79     Emup1 = (guess.ci * par.scale - policy.li.^(1 + par.eta) / (1 + par.eta)).^(-1); % expected marginal
utility at t+2
80     Emup = par.beta * guess.r0 * Emup1 * Pr'; % expected marginal utility at t+1
81     Ec = Emup.^(-1) + policy.li.^(1 + par.eta) / (1 + par.eta); % consumption tomorrow
82     bi_state = (Ec ./ w_p0 + grid.b2 - (1 - guess.tau) .* grid.s2 .* ...
83                 policy.li) ./ (guess.r0); % state debt tomorrow (i.e. choice debt today)
84
85     c_new = ones(nb, ns);
86
87     for j=1:ns
88
89         c_new(:,j) = (grid.b > bi_state(Ind_b_min, j)) .* ... % indicator function on whether borrowing
constraint is binding
90                     interp1(bi_state(:, j), Ec(:, j), grid.b, 'pchip') + ... % interpolate c_s = f(
bi_state) at each grid point
91                     (grid.b <= bi_state(Ind_b_min, j)) .* c_constrained(:,j); % if constraint is binding,
then c_constrained
92         c_new(:,j) = max(c_new(:,j), 1e-5); % rules out negative values
93     end
94
95     guess.ci = c_new * lambda + (1 - lambda) * guess.ci;
96
97     err_c = max(max(abs(c_new - guess.ci)));
98
99     iter_c
100     err_c
101
102     iter_c = iter_c + 1;
103
104 end
105
106 % Write the policy function for consumption
107 for j=1:ns
108     policy.ci(:,j) = interp1(grid.b, guess.ci(:,j), grid.b_fine);
109 end
110
111 %% Solve for interest
112
113 % Write the policy function for assets
114 bi_choice = (grid.b2 * guess.r0 + (1 - guess.tau) * grid.s2 .* policy.li - ...
115             guess.ci ./ w_p0);
116
117 for j=1:ns
118     policy.bi(:,j) = interp1(grid.b, bi_choice(:,j), grid.b_fine);
119 end
120
121 % Compute the endogenous distribution
122 trows = zeros(nb_fine * ns * ns * 2, 1);
123 tcols = trows;
124 tvals = tcols;
125 index = 0;
126 for j=1:ns
127     for bi = 1:nb_fine
128         [vals, inds] = basefun(grid.b_fine, nb_fine, policy.bi(bi, j));
129         for jp=1:ns
130             index = index + 1;
131             trows(index) = bi + (j - 1) * nb_fine;
132             tcols(index) = inds(1) + (jp - 1) * nb_fine;
133             tvals(index) = Pr(j, jp) * vals(1);
134             index = index+1;
135             trows(index) = bi + (j - 1) * nb_fine;
136             tcols(index) = inds(2) + (jp - 1) * nb_fine;
137             tvals(index) = Pr(j, jp) * vals(2);
138         end
139     end
140 end
141 transMat = sparse(trows, tcols, tvals, nb_fine * ns, nb_fine * ns);
142 [EigVec, EigVal] = eigs(transMat.', 1);
143 EigVec = EigVec / sum(EigVec);

```

```

144 EigVec(EigVec < 0) = 0;
145 EigVec = EigVec / sum(EigVec);
146 G0 = reshape(EigVec / sum(EigVec), [nb_fine ns]); % distr. of HHs across assets & states
147
148 % update guess for r
149 b = sum(sum(policy.bi .* G0)); % aggregate HH lending
150 c = sum(sum(policy.ci .* G0));
151 b_star = -(b + par.bg); % solve for aggregate borrowing from bond market clearing
152 tau_new = (par.g + par.bg * (1 - guess.r0)) / y; % solve for tax from govt BC
153 guess.tau = lambda * tau_new + (1 - lambda) * guess.tau;
154
155 r_new = (par.bg + par.g - guess.tau * y) / par.bg; % update r from govt BC
156 % r_new = 1 + c / (b * w_p0) - (1 - guess.tau) * l0 / b; % solve for interest from HH BC
157
158 err_r = abs(r_new - guess.r0);
159
160 guess.r0 = lambda * r_new + (1 - lambda) * guess.r0; % update
161
162 iter_r
163 err_r
164
165 iter_r = iter_r + 1;
166 %%
167
168 end
169
170 %% Solve for terms of trade
171 ch_star = par.gamma * y;
172 ch = y - par.g - ch_star; % consumption of home goods from market clearing
173 ph_pf_new = cf / ch; % domestic price of foreign goods from the FOC
174
175 err_tot = max(max(abs(ph_pf_new - guess.ph_pf)));
176
177 guess.ph_pf = lambda * ph_pf_new + (1 - lambda) * guess.ph_pf; % update
178
179 iter_tot
180 err_tot
181
182 iter_tot = iter_tot + 1;
183
184 end
185
186 %
187 %-----
188 % 6. Export results
189 %-----
190
191 save("PS3Q5.mat");
192
193 %
194 %-----
195 % 5. Defining functions
196 %-----
197
198 function [vals, inds]=basefun(grid_x,npx,x)
199 %Linear interpolation
200 jl=1;
201 ju=npx;
202 while((ju-jl>1))
203     jm=round((ju+jl)/2);
204     if(x>=grid_x(jm))
205         jl=jm;
206     else
207         ju=jm;
208     end
209 end
210
211 i=jl+1;
212 vals(2)=(x-grid_x(i-1))/(grid_x(i)-grid_x(i-1));
213 vals(2)=max(0.0d0,min(1.0d0,vals(2)));
214 vals(1)=1.0d0-vals(2);
215 inds(2)=i;
216 inds(1)=i-1;
217

```



## Appendix D: Q6 model file

```
1
2
3 %
4 %
5 % Title: International Macro–Finance Problem Set 3, model file
6 % Author: —
7 % Date: 10/12/2023
8 % Description: Open economy heterogeneous agent model
9 %
10 %
11
12
13 %
14 %% 0. Housekeeping (close all graphic windows)
15 %
16
17 close all;
18 clear all;
19
20 cd '/home/—/Desktop/International Macro/PS3';
21
22 %
23 %% 1. Retrieve parameters from Q4
24 %
25
26 load("PS3Q4.mat");
27
28
29 %
30 %% 2. Set government debt as parameter & set foreign lending
31 %
32
33 par.bg = bg;
34 clear bg;
35
36
37 %
38 %% 3. Iterations
39 %
40
41 iter_tot = 1;
42 err_tot = 1;
43
44
45 g_tol = 1e-4;          % Tolerance for convergence of par_g
46 max_iter_g = 1000;     % Maximum iterations for finding par_g
47
48 while err_tot > errtol_tot && iter_tot <= maxIter_tot
49
50     w_p0 = par.alphah^par.alphah * (1 - par.alphah)^(1 - par.alphah) * ...
51           guess.ph_pf^(1 - par.alphah); % update real wage from the tot
52
53     %% Compute labour supply
54     policy.li = ((1 - par.tau) .* w_p0 .* grid.s1).^(1 / par.eta);
55     l0 = pi * policy.li.';
56
57     iter_r = 1;
58     err_r = 1;
59     %%
60
61     while err_r > errtol_r && iter_r <= maxIter_r
62
63         %% Solve for consumption
64         y = l0;
65
66         % Compute constrained consumption given R
67         c_constrained = (1 - par.tau) .* w_p0 .* grid.s2 .* policy.li + ...
68                         w_p0 .* grid.b2 .* guess.r0 - ...
69                         w_p0 .* b_con;
70         c_constrained = max(c_constrained, 1e-5); % rule out consumption below 0
71
72         iter_c = 1;
73         err_c = 1;
```

```

74 guess.ci = ones(nb,ns);
75 %%
76
77
78 while err_c > errtol_c && iter_c <= maxIter_c
79
80
81     Emup1 = (guess.ci * par.scale - policy.li.^(1 + par.eta) / (1 + par.eta)).^(-1); % expected marginal
utility at t+2
82     Emup = par.beta * guess.r0 * Emup1 * Pr'; % expected marginal utility at t+1
83     Ec = Emup.^(-1) + policy.li.^(1 + par.eta) / (1 + par.eta); % consumption tomorrow
84     bi_state = (Ec ./ w_p0 + grid.b2 - (1 - par.tau) .* grid.s2 .* ...
85                 policy.li) ./ (guess.r0); % state debt tomorrow (i.e. choice debt today)
86
87     c_new = ones(nb, ns);
88
89     for j=1:ns
90
91         c_new(:,j) = (grid.b > bi_state(Ind_b_min, j)) .* ... % indicator function on whether borrowing
constraint is binding
92                     interp1(bi_state(:, j), Ec(:, j), grid.b, 'pchip') + ... % interpolate c_s = f(
bi_state) at each grid point
93                     (grid.b <= bi_state(Ind_b_min, j)) .* c_constrained(:,j); % if constraint is binding,
then c_constrained
94         c_new(:,j) = max(c_new(:,j), 1e-5); % rules out negative values
95     end
96
97     guess.ci = c_new * lambda + (1 - lambda) * guess.ci;
98
99     err_c = max(max(abs(c_new - guess.ci)));
100
101     iter_c
102     err_c
103
104     iter_c = iter_c + 1;
105
106 end
107
108 % Write the policy function for consumption
109 for j=1:ns
110     policy.ci(:,j) = interp1(grid.b, guess.ci(:,j), grid.b_fine);
111 end
112
113 %% Solve for interest
114
115 % Write the policy function for assets
116 bi_choice = (grid.b2 * guess.r0 + (1 - par.tau) * grid.s2 .* policy.li - ...
117             guess.ci ./ w_p0);
118
119 for j=1:ns
120     policy.bi(:,j) = interp1(grid.b, bi_choice(:,j), grid.b_fine);
121 end
122
123 % Compute the endogenous distribution
124 trows = zeros(nb_fine * ns * ns * 2, 1);
125 tcols = trows;
126 tvals = trows;
127 index = 0;
128 for j=1:ns
129     for bi = 1:nb_fine
130         [vals,inds] = basefun(grid.b_fine, nb_fine, policy.bi(bi,j));
131         for jp=1:ns
132             index = index + 1;
133             trows(index) = bi + (j - 1) * nb_fine;
134             tcols(index) = inds(1) + (jp - 1) * nb_fine;
135             tvals(index) = Pr(j, jp) * vals(1);
136             index = index+1;
137             trows(index) = bi + (j - 1) * nb_fine;
138             tcols(index) = inds(2) + (jp - 1) * nb_fine;
139             tvals(index) = Pr(j, jp) * vals(2);
140         end
141     end
142 end
143 transMat = sparse(trows, tcols, tvals, nb_fine * ns, nb_fine * ns);

```

```

144 [EigVec, EigVal] = eigs(transMat.', 1);
145 EigVec = EigVec / sum(EigVec);
146 EigVec(EigVec < 0) = 0;
147 EigVec = EigVec / sum(EigVec);
148 G0 = reshape(EigVec / sum(EigVec), [nb-fine ns]); % distr. of HHs across assets & states
149
150 % update guess for r
151 c = sum(sum(policy.ci .* G0)); % aggregate HH consumption
152 b = sum(sum(policy.bi .* G0));
153 b_star = -(b + par.bg); % solve for aggregate borrowing from bond market clearing
154 g = par.tau * y + par.bg * (guess.r0 - 1); % solve for govt spending from govt BC
155
156 r_new = (par.bg + g - par.tau * y) / par.bg; % update r from govt BC
157 % r_new = 1 + c / (b * w_p0) - (1 - par.tau) * l0 / b; % solve for interest from HH BC
158
159 err_r = abs(r_new - guess.r0);
160
161 guess.r0 = lambda * r_new + (1 - lambda) * guess.r0; % update
162
163 iter_r
164 err_r
165
166 iter_r = iter_r + 1;
167 %%
168
169 end
170
171 %% Solve for terms of trade
172 ch_star = par.gamma * y;
173 ch = y - g - ch_star; % consumption of home goods from market clearing
174 ph_pf_new = cf / ch; % domestic price of foreign goods from the FOC
175
176 err_tot = max(max(abs(ph_pf_new - guess.ph_pf)));
177
178 guess.ph_pf = lambda * ph_pf_new + (1 - lambda) * guess.ph_pf; % update
179
180 iter_tot
181 err_tot
182
183 iter_tot = iter_tot + 1;
184
185 end
186
187
188 %-----
189 % 4. Export results
190 %-----
191
192 save("PS3Q6.mat");
193
194
195 %-----
196 % 5. Defining functions
197 %-----
198
199 function [vals, inds]=basefun(grid_x,npx,x)
200 %Linear interpolation
201 j1=1;
202 ju=npx;
203 while((ju-j1>1))
204     jm=round((ju+j1)/2);
205     if(x>=grid_x(jm))
206         j1=jm;
207     else
208         ju=jm;
209     end
210 end
211
212 i=j1+1;
213 vals(2)=(x-grid_x(i-1))/(grid_x(i)-grid_x(i-1));
214 vals(2)=max(0.0d0,min(1.0d0,vals(2)));
215 vals(1)=1.0d0-vals(2);
216 inds(2)=i;
217 inds(1)=i-1;

```

218

219 **end**

## Appendix E: AR(1) Fitting

```
1 %% Calibrate labour productivity from FRED data to match US GDP
2 % (https://fred.stlouisfed.org/series/OPHNFB)
3
4 % Load series
5 lab_prod = xlsread("OPHNFB.xls");
6 lab_prod = lab_prod(:,2);
7 restricted_lab_prod = lab_prod(200:307);
8
9 % Difference the data
10 Dlab_prod = diff(lab_prod);
11
12 % Fit an AR(1) process to the differenced data
13 %model1 = arima(1,1,0);
14 model2 = arima(1,0,0); % AR(1)
15
16 %estmodel1 = estimate(model1, lab_prod);
17 %estmodel2 = estimate(model2, lab_prod);
18
19 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:

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

subject to:

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

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

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

subject to:

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

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:

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

the household's labour supply decision in Country A:

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

and the budget constraints

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

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

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

$$\max_{l_{it}} P_{i,t} y_{i,t} - W_t l_{i,t} \quad (7)$$

subject to:

$$y_{i,t} = l_{i,t}^{1-\alpha}$$

$$P_{i,t} = P_{T,t} \left( \frac{y_{i,t}}{y_{T,t}} \right)^{-\theta}$$

The first order condition is:

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

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

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

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  $\{P_{T,t}, P_{O,t}, P_{T,t}^s, P_{O,t}^s, W_t, E_t\}_{t=0}^{\infty}$  such that the FOCs 3 to 9 hold and market-clearing is satisfied on the tradeable goods market:

$$c_{A,T,t} + c_{B,T,t} = y_{T,t} \quad (10)$$

the oil market:

$$c_{A,O,t} + c_{B,O,t} = y_{O,t} \quad (11)$$

and the labour market:

$$l_t = \int_0^1 l_{i,t} di \quad (12)$$

## 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:

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

subject to:

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

and:

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

subject to:

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

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

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

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  $\{P_{T,t}, P_{O,t}, P_{T,t}^s, P_{O,t}^s, W_t, E_t, R_t^s\}_{t=0}^{\infty}$  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  $\chi$  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:

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

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

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

and HTM households' FOCs by:

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

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

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

### 0.4 TANBNR FER

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

$$W_t = (1 - \theta)(1 - \alpha)P_{T,t} \frac{l_{i,t}^{(1-\theta)(1-\alpha)-1}}{y_{T,t}^{-\theta}} - \phi \left[ \frac{P_{T,t}}{P_{T,t-1}} \frac{l_{i,t}^{-\theta(1-\alpha)}}{y_{T,t}^{-\theta}} \left( \frac{y_{i,t-1}}{y_{T,t-1}} \right)^{\theta} - 1 \right] \dots \quad (22)$$

$$P_{T,t} Y_{T,t} (-\theta)(1 - \alpha) \left[ \frac{P_{T,t}}{P_{T,t-1}} \frac{l_{i,t}^{-\theta(1-\alpha)}}{y_{T,t}^{-\theta}} \left( \frac{y_{i,t-1}}{y_{T,t-1}} \right)^{\theta} \right]$$

## 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:

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

The Euler equation for Ricardian households is:

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

and labour supply decision for both households is:

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

## 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,  $\chi$ , 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,  $\theta$ , and the intercept of the AR(1) process for oil endowment  $\gamma$ , 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
$\alpha$	0.3	0.3	0.3	0.3	0.3	0.3
$\beta$	0.99	0.99	0.99	0.99	0.99	0.99
$\psi$	1.5	1.5	1.5	1.5	1.5	1.5
$\phi$				10	10	
$\eta$	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
$\theta$	1.0	1.0	1.0	1.0	1.0	1.0
$\rho$	0.9	0.9	0.9	0.9	0.9	0.9
$\gamma$	0.1	0.1	0.1	0.1	0.1	0.1
$\chi$	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:

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

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.

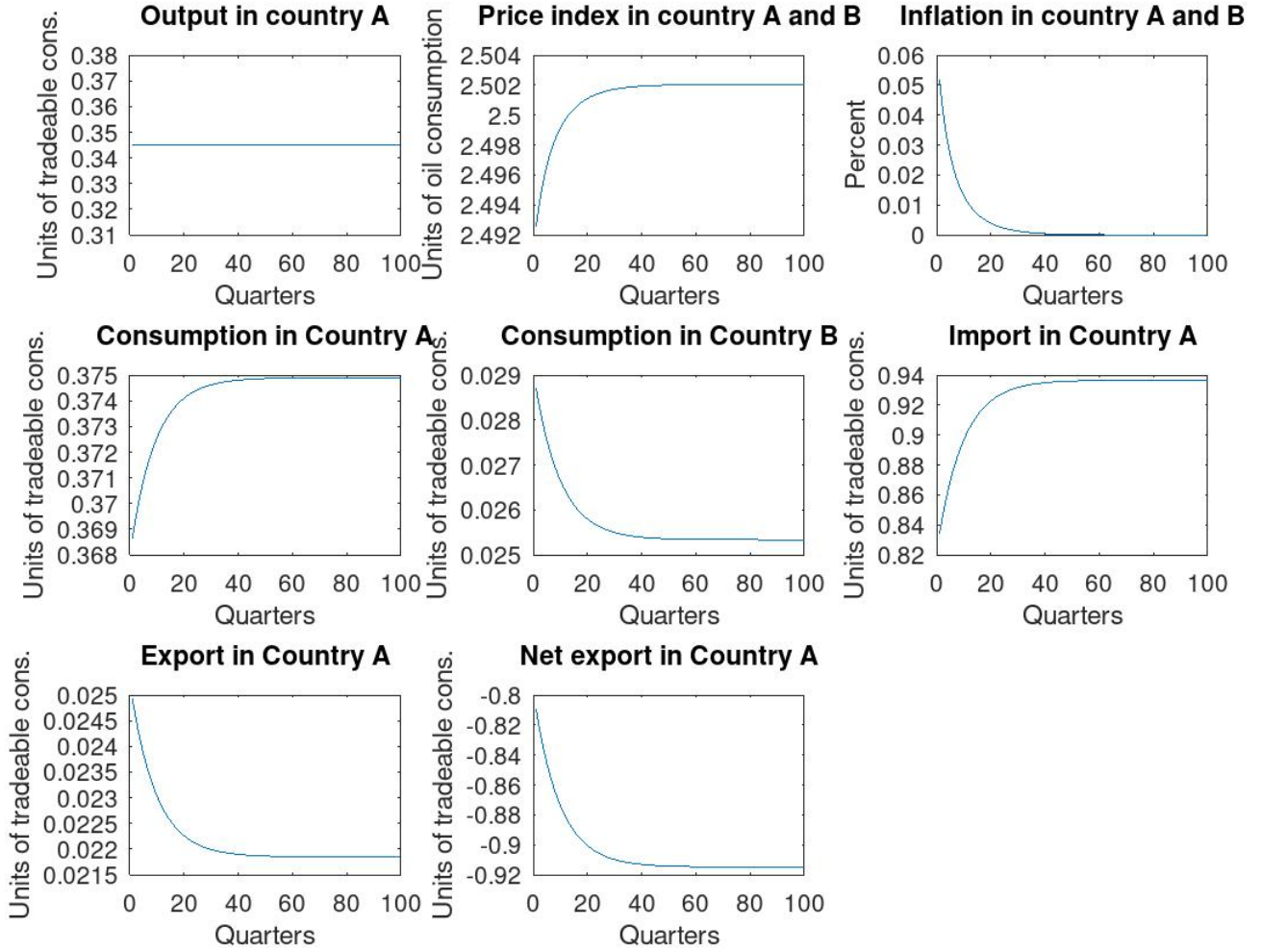


Figure 1: IRFs for RAFAFP

## 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).

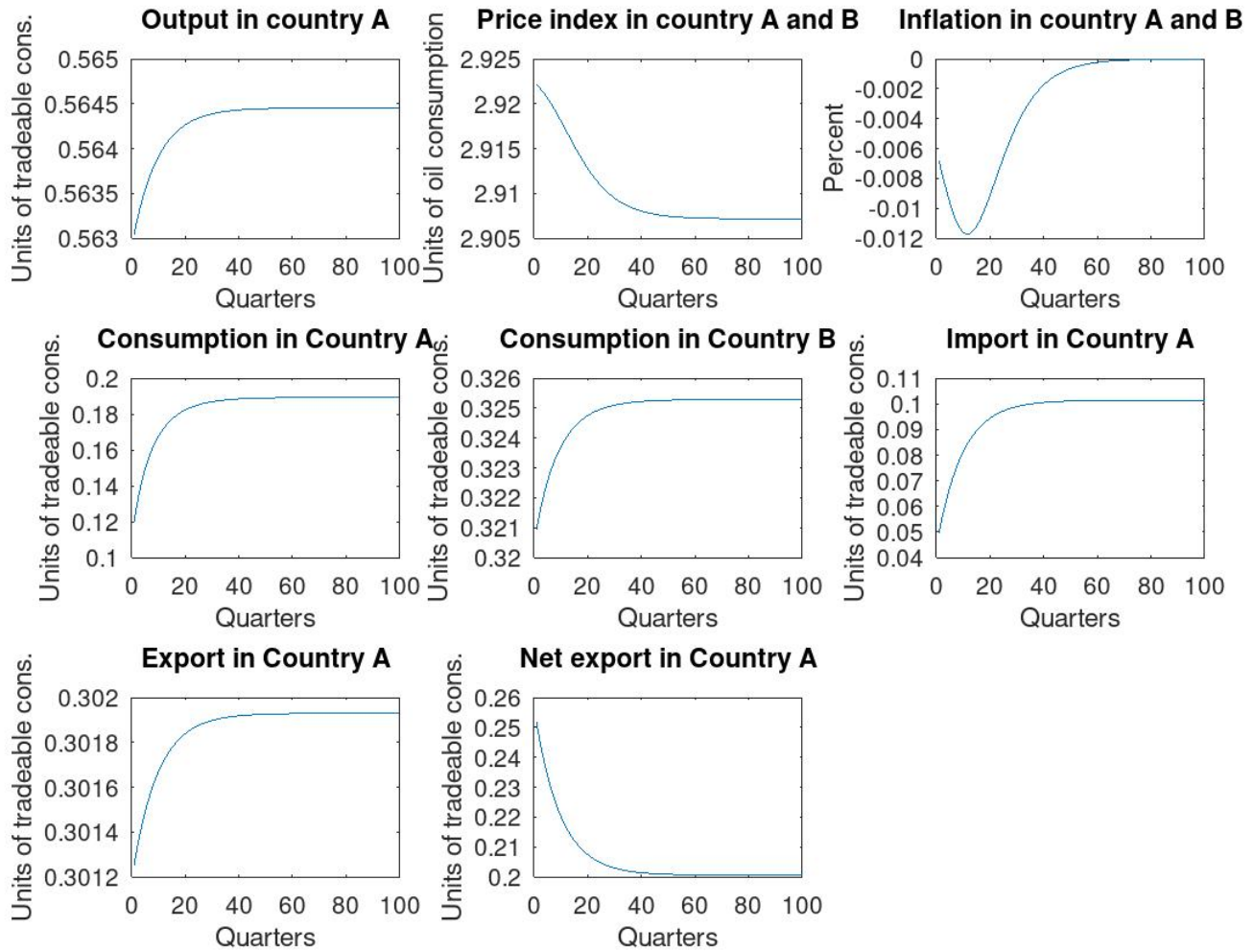


Figure 2: IRFs for RANBFP

## 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.

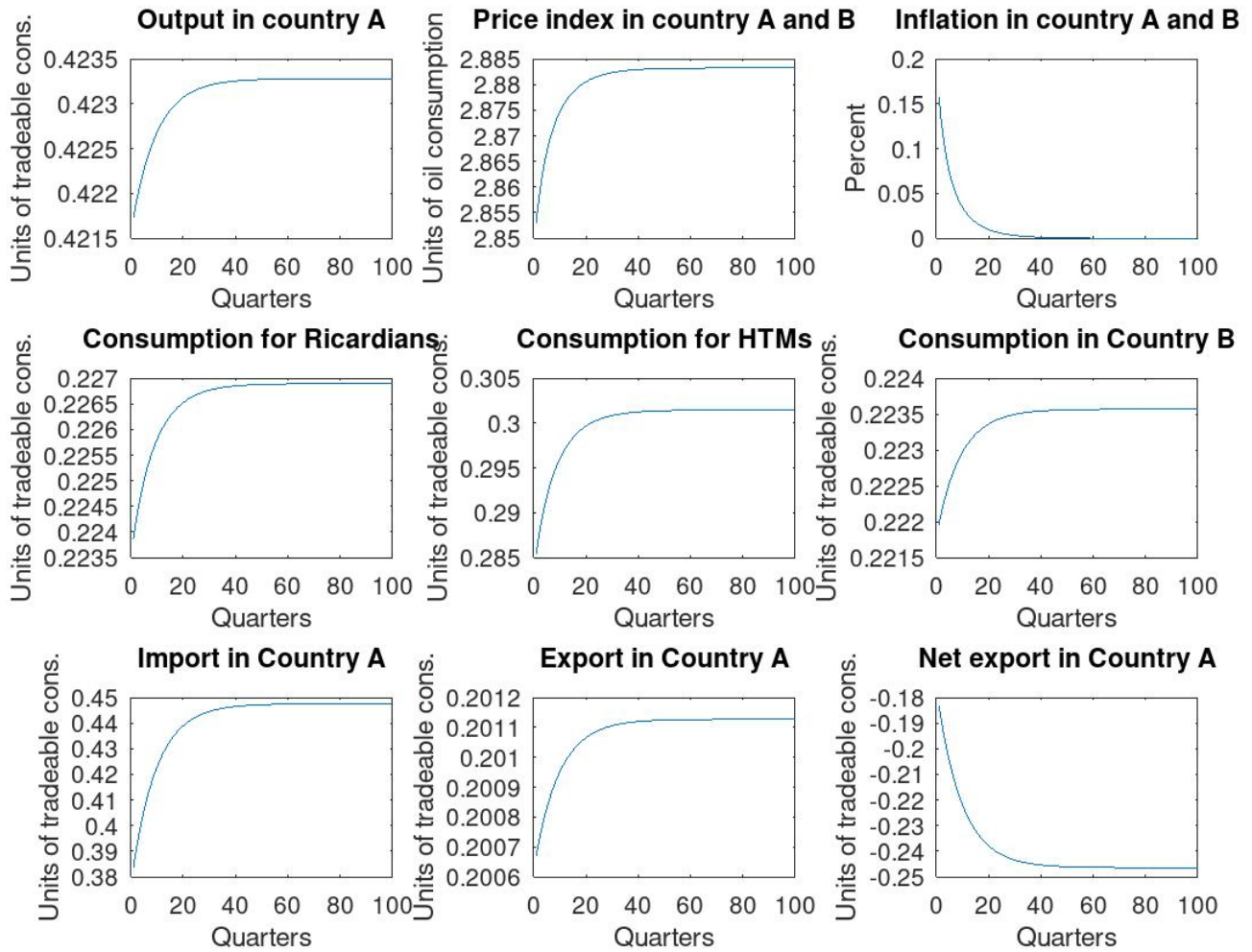


Figure 3: IRFs for TANBFP

## 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.

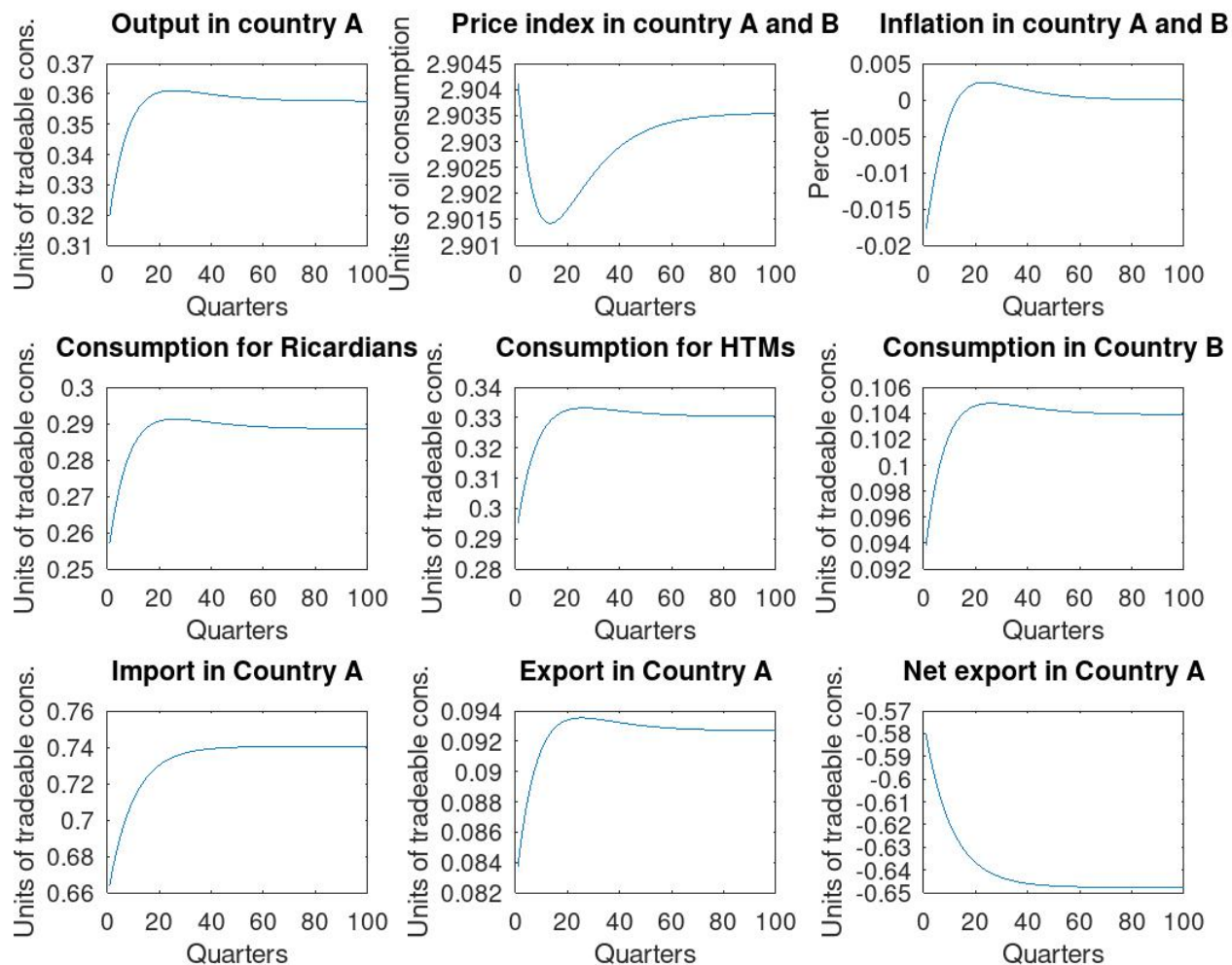


Figure 4: IRFs for TANBNR FER

### 0.11 TANBNR PIT

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

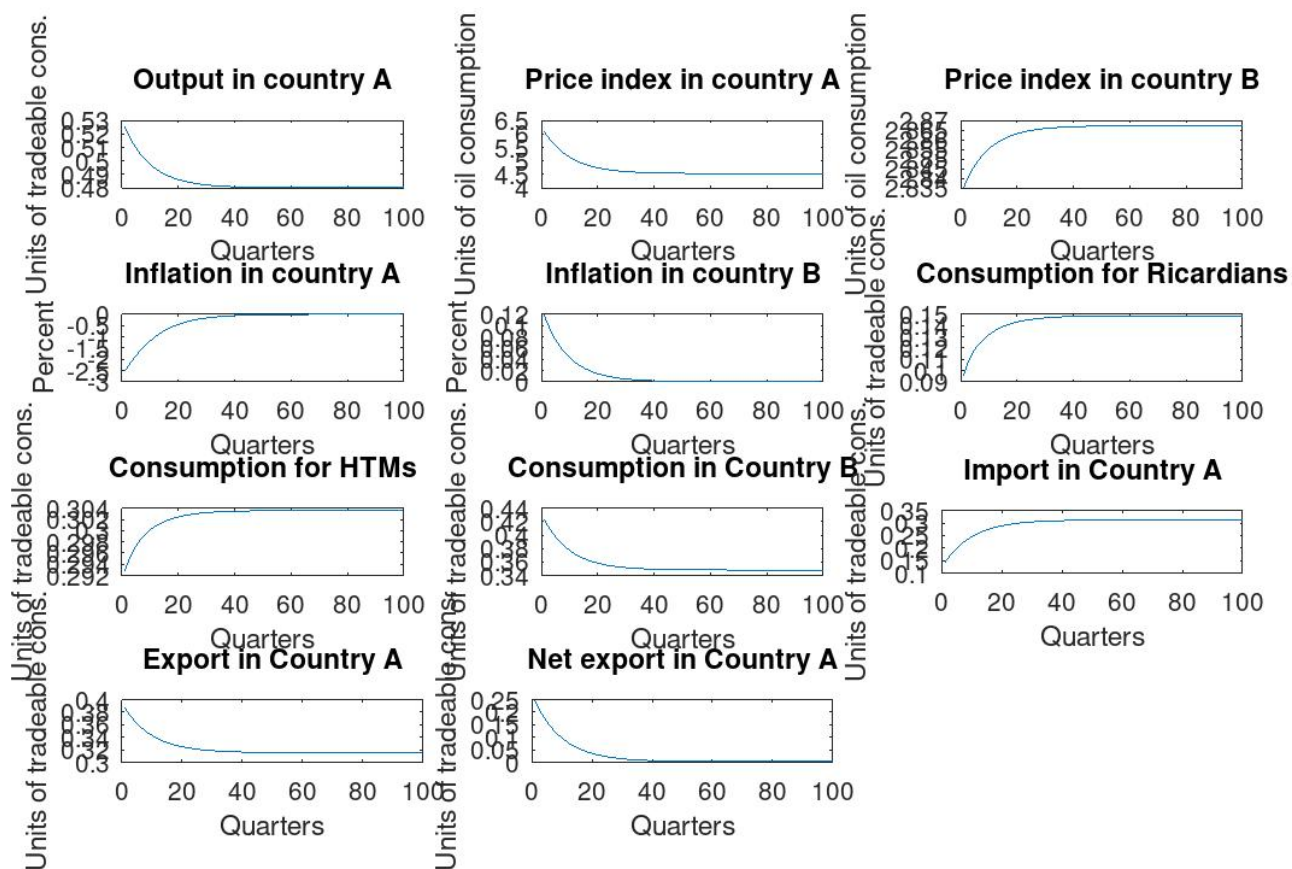


Figure 5: IRFs for TANBNR PIT



## 0.12 TANBFPNH

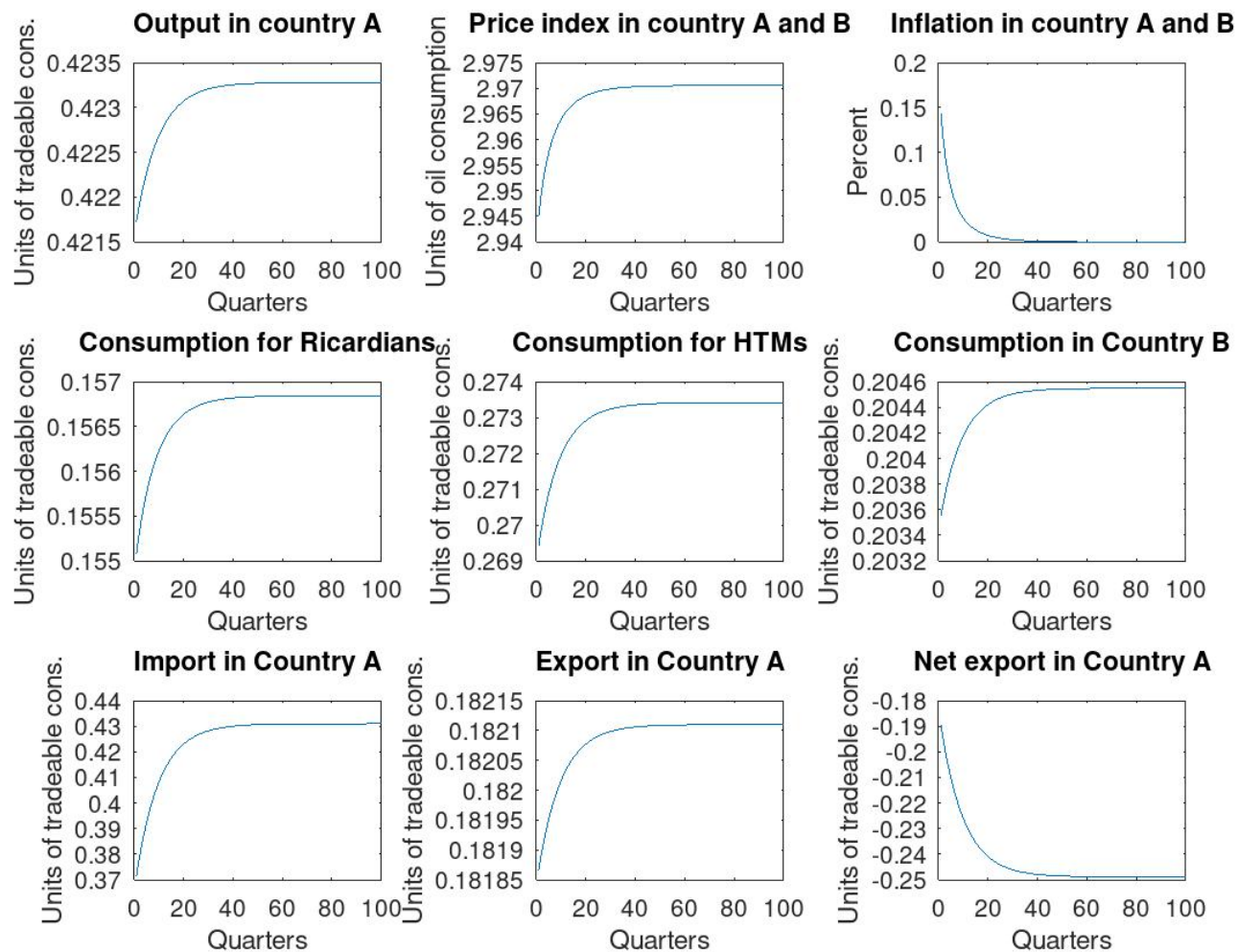


Figure 6: IRFs for TANBFPNH

# Appendix A: Main File

```
1
2 %
3 %
4 % Title: International Macro–Finance Problem Set 4, main file
5 % Author: —
6 % Date: 10/01/2024
7 % Description: Main file for the final problem set
8 %
9 %
10
11
12 %
13 % 0. Housekeeping (close all graphic windows)
14 %
15
16 close all;
17 clear all;
18
19
20 %
21 % 1. Load parameters
22 %
23
24 parameters;
25 save parameters par;
26
27
28 %
29 % 1. Run models
30 %
31
32 dynare rafafp;
33 dynare ranbfp;
34 dynare tanbfp;
35 dynare tanbnr_fer;
36 dynare tanbnr_pit;
37 dynare tanbfpnh;
38
39
40 %
41 % 2. Load results
42 %
43
44
45 rafafp_results = load("rafafp_results.mat");
46
47 RAFAFP.c_aoss = rafafp_results.oo_.steady_state(1);
48 RAFAFP.c_booss = rafafp_results.oo_.steady_state(2);
49 RAFAFP.c_atss = rafafp_results.oo_.steady_state(3);
50 RAFAFP.c_btss = rafafp_results.oo_.steady_state(4);
51 RAFAFP.lss = rafafp_results.oo_.steady_state(5);
52 RAFAFP.y_tss = rafafp_results.oo_.steady_state(6);
53 RAFAFP.y_oss = rafafp_results.oo_.steady_state(7);
54 RAFAFP.p_tss = rafafp_results.oo_.steady_state(8);
55 RAFAFP.wss = rafafp_results.oo_.steady_state(9);
56 RAFAFP.piss = rafafp_results.oo_.steady_state(10);
57
58 RAFAFP.c_ao = rafafp_results.oo_.irfs.c_ao_err_yo;
59 RAFAFP.c_at = rafafp_results.oo_.irfs.c_at_err_yo;
60 RAFAFP.c_bo = rafafp_results.oo_.irfs.c_bo_err_yo;
61 RAFAFP.c_bt = rafafp_results.oo_.irfs.c_bt_err_yo;
62 RAFAFP.l = rafafp_results.oo_.irfs.l_err_yo;
63 RAFAFP.p_t = rafafp_results.oo_.irfs.p_t_err_yo;
64 RAFAFP.pi = rafafp_results.oo_.irfs.pi_err_yo;
65 RAFAFP.w = rafafp_results.oo_.irfs.w_err_yo;
66 RAFAFP.y_o = rafafp_results.oo_.irfs.y_o_err_yo;
67 RAFAFP.y_t = rafafp_results.oo_.irfs.y_t_err_yo;
68
69 psi = rafafp_results.M_.params(2);
70 RAFAFP.eta = rafafp_results.M_.params(3);
71 RAFAFP.s_t = rafafp_results.M_.params(4);
72
73
```



```

74 ranbfp_results = load(" ranbfp_results.mat");
75
76 RANBFP.c_aoss = ranbfp_results.oo_.steady_state(1);
77 RANBFP.c_boss = ranbfp_results.oo_.steady_state(2);
78 RANBFP.c_atss = ranbfp_results.oo_.steady_state(3);
79 RANBFP.c_btss = ranbfp_results.oo_.steady_state(4);
80 RANBFP.lss = ranbfp_results.oo_.steady_state(5);
81 RANBFP.y_tss = ranbfp_results.oo_.steady_state(6);
82 RANBFP.y_oss = ranbfp_results.oo_.steady_state(7);
83 RANBFP.p_tss = ranbfp_results.oo_.steady_state(8);
84 RANBFP.wss = ranbfp_results.oo_.steady_state(9);
85 RANBFP.piss = ranbfp_results.oo_.steady_state(10);
86 RANBFP.n_ass = ranbfp_results.oo_.steady_state(11);
87 RANBFP.n_bss = ranbfp_results.oo_.steady_state(12);
88 RANBFP.r_starss = ranbfp_results.oo_.steady_state(13);
89
90 RANBFP.c_ao = ranbfp_results.oo_.irfs.c_ao_err_yo;
91 RANBFP.c_at = ranbfp_results.oo_.irfs.c_at_err_yo;
92 RANBFP.c_bo = ranbfp_results.oo_.irfs.c_bo_err_yo;
93 RANBFP.c_bt = ranbfp_results.oo_.irfs.c_bt_err_yo;
94 RANBFP.l = ranbfp_results.oo_.irfs.l_err_yo;
95 RANBFP.p_t = ranbfp_results.oo_.irfs.p_t_err_yo;
96 RANBFP.pi = ranbfp_results.oo_.irfs.pi_err_yo;
97 RANBFP.w = ranbfp_results.oo_.irfs.w_err_yo;
98 RANBFP.y_o = ranbfp_results.oo_.irfs.y_o_err_yo;
99 RANBFP.y_t = ranbfp_results.oo_.irfs.y_t_err_yo;
100
101 beta = ranbfp_results.M_.params(2);
102 RANBFP.eta = ranbfp_results.M_.params(4);
103 RANBFP.s_t = ranbfp_results.M_.params(5);
104
105
106 tanbfp_results = load(" tanbfp_results.mat");
107
108 TANBFP.c_ahoss = ranbfp_results.oo_.steady_state(1);
109 TANBFP.c_aross = tanbfp_results.oo_.steady_state(2);
110 TANBFP.c_boss = tanbfp_results.oo_.steady_state(3);
111 TANBFP.c_ahtss = tanbfp_results.oo_.steady_state(4);
112 TANBFP.c_artss = tanbfp_results.oo_.steady_state(5);
113 TANBFP.c_btss = tanbfp_results.oo_.steady_state(6);
114 TANBFP.l_hss = tanbfp_results.oo_.steady_state(7);
115 TANBFP.l_rss = tanbfp_results.oo_.steady_state(8);
116 TANBFP.lss = tanbfp_results.oo_.steady_state(9);
117 TANBFP.y_tss = tanbfp_results.oo_.steady_state(10);
118 TANBFP.y_oss = tanbfp_results.oo_.steady_state(11);
119 TANBFP.p_tss = tanbfp_results.oo_.steady_state(12);
120 TANBFP.wss = tanbfp_results.oo_.steady_state(13);
121 TANBFP.piss = tanbfp_results.oo_.steady_state(14);
122 TANBFP.n_ass = tanbfp_results.oo_.steady_state(15);
123 TANBFP.n_bss = tanbfp_results.oo_.steady_state(16);
124 TANBFP.rstarss = tanbfp_results.oo_.steady_state(17);
125 TANBFP.pirss = tanbfp_results.oo_.steady_state(18);
126 TANBFP.pihss = tanbfp_results.oo_.steady_state(19);
127
128 TANBFP.c_aro = tanbfp_results.oo_.irfs.c_aro_err_yo;
129 TANBFP.c_art = tanbfp_results.oo_.irfs.c_art_err_yo;
130 TANBFP.c_aho = tanbfp_results.oo_.irfs.c_aho_err_yo;
131 TANBFP.c_aht = tanbfp_results.oo_.irfs.c_aht_err_yo;
132 TANBFP.c_bo = tanbfp_results.oo_.irfs.c_bo_err_yo;
133 TANBFP.c_bt = tanbfp_results.oo_.irfs.c_bt_err_yo;
134 TANBFP.l_r = tanbfp_results.oo_.irfs.l_r_err_yo;
135 TANBFP.l_h = tanbfp_results.oo_.irfs.l_h_err_yo;
136 TANBFP.l = tanbfp_results.oo_.irfs.l_err_yo;
137 TANBFP.p_t = tanbfp_results.oo_.irfs.p_t_err_yo;
138 TANBFP.pi = tanbfp_results.oo_.irfs.pi_err_yo;
139 TANBFP.w = tanbfp_results.oo_.irfs.w_err_yo;
140 TANBFP.y_o = tanbfp_results.oo_.irfs.y_o_err_yo;
141 TANBFP.y_t = tanbfp_results.oo_.irfs.y_t_err_yo;
142
143 TANBFP.eta = tanbfp_results.M_.params(4);
144 TANBFP.s_t = tanbfp_results.M_.params(5);
145 chi = tanbfp_results.M_.params(9);
146
147

```

```

148 tanbnr_fer_results = load("tanbnr_fer_results.mat");
149
150 TANBNR_FER.c_ahoss = tanbnr_fer_results.oo_.steady_state(1);
151 TANBNR_FER.c_aross = tanbnr_fer_results.oo_.steady_state(2);
152 TANBNR_FER.c_boss = tanbnr_fer_results.oo_.steady_state(3);
153 TANBNR_FER.c_ahoss = tanbnr_fer_results.oo_.steady_state(4);
154 TANBNR_FER.c_artss = tanbnr_fer_results.oo_.steady_state(5);
155 TANBNR_FER.c_btss = tanbnr_fer_results.oo_.steady_state(6);
156 TANBNR_FER.l_hss = tanbnr_fer_results.oo_.steady_state(7);
157 TANBNR_FER.l_rss = tanbnr_fer_results.oo_.steady_state(8);
158 TANBNR_FER.lss = tanbnr_fer_results.oo_.steady_state(9);
159 TANBNR_FER.y_tss = tanbnr_fer_results.oo_.steady_state(10);
160 TANBNR_FER.y_oss = tanbnr_fer_results.oo_.steady_state(11);
161 TANBNR_FER.p_tss = tanbnr_fer_results.oo_.steady_state(12);
162 TANBNR_FER.wss = tanbnr_fer_results.oo_.steady_state(13);
163 TANBNR_FER.piss = tanbnr_fer_results.oo_.steady_state(14);
164 TANBNR_FER.n_oss = tanbnr_fer_results.oo_.steady_state(15);
165 TANBNR_FER.n_bss = tanbnr_fer_results.oo_.steady_state(16);
166 TANBNR_FER.r_starss = tanbnr_fer_results.oo_.steady_state(17);
167 TANBNR_FER.pirss = tanbnr_fer_results.oo_.steady_state(18);
168 TANBNR_FER.pihss = tanbnr_fer_results.oo_.steady_state(19);
169
170 TANBNR_FER.c_aro = tanbnr_fer_results.oo_.irfs.c_aro_err_yo;
171 TANBNR_FER.c_art = tanbnr_fer_results.oo_.irfs.c_art_err_yo;
172 TANBNR_FER.c_aho = tanbnr_fer_results.oo_.irfs.c_aho_err_yo;
173 TANBNR_FER.c_ah = tanbnr_fer_results.oo_.irfs.c_ah_err_yo;
174 TANBNR_FER.c_bo = tanbnr_fer_results.oo_.irfs.c_bo_err_yo;
175 TANBNR_FER.c_bt = tanbnr_fer_results.oo_.irfs.c_bt_err_yo;
176 TANBNR_FER.l_r = tanbnr_fer_results.oo_.irfs.l_r_err_yo;
177 TANBNR_FER.l_h = tanbnr_fer_results.oo_.irfs.l_h_err_yo;
178 TANBNR_FER.l = tanbnr_fer_results.oo_.irfs.l_err_yo;
179 TANBNR_FER.p_t = tanbnr_fer_results.oo_.irfs.p_t_err_yo;
180 TANBNR_FER.pi = tanbnr_fer_results.oo_.irfs.pi_err_yo;
181 TANBNR_FER.w = tanbnr_fer_results.oo_.irfs.w_err_yo;
182 TANBNR_FER.y_o = tanbnr_fer_results.oo_.irfs.y_o_err_yo;
183 TANBNR_FER.y_t = tanbnr_fer_results.oo_.irfs.y_t_err_yo;
184
185 TANBNR_FER.eta = tanbnr_fer_results.M_.params(5);
186 TANBNR_FER.s_t = tanbnr_fer_results.M_.params(6);
187
188
189 tanbnr_pit_results = load("tanbnr_pit_results.mat");
190
191 TANBNR_PIT.c_ahoss = tanbnr_pit_results.oo_.steady_state(1);
192 TANBNR_PIT.c_aross = tanbnr_pit_results.oo_.steady_state(2);
193 TANBNR_PIT.c_boss = tanbnr_pit_results.oo_.steady_state(3);
194 TANBNR_PIT.c_ahoss = tanbnr_pit_results.oo_.steady_state(4);
195 TANBNR_PIT.c_artss = tanbnr_pit_results.oo_.steady_state(5);
196 TANBNR_PIT.c_btss = tanbnr_pit_results.oo_.steady_state(6);
197 TANBNR_PIT.l_hss = tanbnr_pit_results.oo_.steady_state(7);
198 TANBNR_PIT.l_rss = tanbnr_pit_results.oo_.steady_state(8);
199 TANBNR_PIT.lss = tanbnr_pit_results.oo_.steady_state(9);
200 TANBNR_PIT.y_tss = tanbnr_pit_results.oo_.steady_state(10);
201 TANBNR_PIT.y_oss = tanbnr_pit_results.oo_.steady_state(11);
202 TANBNR_PIT.p_tss = tanbnr_pit_results.oo_.steady_state(12);
203 TANBNR_PIT.ess = tanbnr_pit_results.oo_.steady_state(13);
204 TANBNR_PIT.wss = tanbnr_pit_results.oo_.steady_state(14);
205 TANBNR_PIT.piss = tanbnr_pit_results.oo_.steady_state(15);
206 TANBNR_PIT.n_oss = tanbnr_pit_results.oo_.steady_state(16);
207 TANBNR_PIT.n_bss = tanbnr_pit_results.oo_.steady_state(17);
208 TANBNR_PIT.r_starss = tanbnr_pit_results.oo_.steady_state(18);
209 TANBNR_PIT.pirss = tanbnr_pit_results.oo_.steady_state(19);
210 TANBNR_PIT.pihss = tanbnr_pit_results.oo_.steady_state(20);
211
212 TANBNR_PIT.c_aro = tanbnr_pit_results.oo_.irfs.c_aro_err_yo;
213 TANBNR_PIT.c_art = tanbnr_pit_results.oo_.irfs.c_art_err_yo;
214 TANBNR_PIT.c_aho = tanbnr_pit_results.oo_.irfs.c_aho_err_yo;
215 TANBNR_PIT.c_ah = tanbnr_pit_results.oo_.irfs.c_ah_err_yo;
216 TANBNR_PIT.c_bo = tanbnr_pit_results.oo_.irfs.c_bo_err_yo;
217 TANBNR_PIT.c_bt = tanbnr_pit_results.oo_.irfs.c_bt_err_yo;
218 TANBNR_PIT.l_r = tanbnr_pit_results.oo_.irfs.l_r_err_yo;
219 TANBNR_PIT.l_h = tanbnr_pit_results.oo_.irfs.l_h_err_yo;
220 TANBNR_PIT.l = tanbnr_pit_results.oo_.irfs.l_err_yo;
221 TANBNR_PIT.p_t = tanbnr_pit_results.oo_.irfs.p_t_err_yo;

```

```

222 TANBNR_PIT.e      = tanbnr_pit_results.oo_.irfs.e_err_yo;
223 TANBNR_PIT.pi     = tanbnr_pit_results.oo_.irfs.pi_err_yo;
224 TANBNR_PIT.w      = tanbnr_pit_results.oo_.irfs.w_err_yo;
225 TANBNR_PIT.y_o    = tanbnr_pit_results.oo_.irfs.y_o_err_yo;
226 TANBNR_PIT.y_t    = tanbnr_pit_results.oo_.irfs.y_t_err_yo;
227
228 TANBNR_PIT.eta = tanbnr_pit_results.M_.params(5);
229 TANBNR_PIT.s_t = tanbnr_pit_results.M_.params(6);
230
231
232 tanbfpnh_results = load("tanbfpnh_results.mat");
233
234 TANBFPNH.c_ahoss = tanbfpnh_results.oo_.steady_state(1);
235 TANBFPNH.c_aross = tanbfpnh_results.oo_.steady_state(2);
236 TANBFPNH.c_boss = tanbfpnh_results.oo_.steady_state(3);
237 TANBFPNH.c_ahtss = tanbfpnh_results.oo_.steady_state(4);
238 TANBFPNH.c_artss = tanbfpnh_results.oo_.steady_state(5);
239 TANBFPNH.c_btss = tanbfpnh_results.oo_.steady_state(6);
240 TANBFPNH.l_hss = tanbfpnh_results.oo_.steady_state(7);
241 TANBFPNH.l_rss = tanbfpnh_results.oo_.steady_state(8);
242 TANBFPNH.lss = tanbfpnh_results.oo_.steady_state(9);
243 TANBFPNH.y_tss = tanbfpnh_results.oo_.steady_state(10);
244 TANBFPNH.y_oss = tanbfpnh_results.oo_.steady_state(11);
245 TANBFPNH.p_tss = tanbfpnh_results.oo_.steady_state(12);
246 TANBFPNH.wss = tanbfpnh_results.oo_.steady_state(13);
247 TANBFPNH.piss = tanbfpnh_results.oo_.steady_state(14);
248 TANBFPNH.n_ass = tanbfpnh_results.oo_.steady_state(15);
249 TANBFPNH.n_bss = tanbfpnh_results.oo_.steady_state(16);
250 TANBFPNH.rstarss = tanbfpnh_results.oo_.steady_state(17);
251 TANBFPNH.pirss = tanbfpnh_results.oo_.steady_state(18);
252 TANBFPNH.pihss = tanbfpnh_results.oo_.steady_state(19);
253
254 TANBFPNH.c_aro = tanbfpnh_results.oo_.irfs.c_aro_err_yo;
255 TANBFPNH.c_art = tanbfpnh_results.oo_.irfs.c_art_err_yo;
256 TANBFPNH.c_aho = tanbfpnh_results.oo_.irfs.c_aho_err_yo;
257 TANBFPNH.c_aht = tanbfpnh_results.oo_.irfs.c_aht_err_yo;
258 TANBFPNH.c_bo = tanbfpnh_results.oo_.irfs.c_bo_err_yo;
259 TANBFPNH.c_bt = tanbfpnh_results.oo_.irfs.c_bt_err_yo;
260 TANBFPNH.l_r = tanbfpnh_results.oo_.irfs.l_r_err_yo;
261 TANBFPNH.l_h = tanbfpnh_results.oo_.irfs.l_h_err_yo;
262 TANBFPNH.l = tanbfpnh_results.oo_.irfs.l_err_yo;
263 TANBFPNH.p_t = tanbfpnh_results.oo_.irfs.p_t_err_yo;
264 TANBFPNH.pi = tanbfpnh_results.oo_.irfs.pi_err_yo;
265 TANBFPNH.w = tanbfpnh_results.oo_.irfs.w_err_yo;
266 TANBFPNH.y_o = tanbfpnh_results.oo_.irfs.y_o_err_yo;
267 TANBFPNH.y_t = tanbfpnh_results.oo_.irfs.y_t_err_yo;
268
269 TANBFPNH.eta = tanbfpnh_results.M_.params(4);
270 TANBFPNH.s_art = tanbfpnh_results.M_.params(5);
271 TANBFPNH.s_aht = tanbfpnh_results.M_.params(6);
272 TANBFPNH.s_bt = tanbfpnh_results.M_.params(7);
273
274 TANBFPNH.s_t = chi * TANBFPNH.s_art + (1 - chi) * TANBFPNH.s_aht;
275
276
277 %
278 % 4. Function definition
279 %
280
281 % CES aggregator:
282
283 function cons = ces_agg(c_t, c_o, eta, s_t)
284
285     cons = (s_t * c_t.^(1 - 1 / eta) +
286             (1 - s_t) * c_o.^(1 - 1 / eta)).^(eta / (eta - 1));
287
288 endfunction
289
290
291
292 %
293 % 3. Create key variables
294 %
295

```

```

296 % output in country A
297
298 RAFAFP.output = RAFAFP.y_t + RAFAFP.y_tss;
299 RANBFP.output = RANBFP.y_t + RANBFP.y_tss;
300 TANBFP.output = TANBFP.y_t + TANBFP.y_tss;
301 TANBNR_FER.output = TANBNR_FER.y_t + TANBNR_FER.y_tss;
302 TANBNR_PIT.output = TANBNR_PIT.y_t + TANBNR_PIT.y_tss;
303 TANBFPNH.output = TANBFPNH.y_t + TANBFPNH.y_tss;
304
305
306 % price levels
307
308 % re-express eveything as levels
309 RAFAFP.p_t = RAFAFP.p_t + RAFAFP.p_tss;
310 RAFAFP.p_o = 1;
311
312 % create a price index
313 RAFAFP.price = ces_agg(RAFAFP.p_t, RAFAFP.p_o, RAFAFP.eta, RAFAFP.s_t);
314
315
316 RANBFP.p_t = RANBFP.p_t + RANBFP.p_tss;
317 RANBFP.p_o = 1;
318
319 RANBFP.price = ces_agg(RANBFP.p_t, RANBFP.p_o, RANBFP.eta, RANBFP.s_t);
320
321
322 TANBFP.p_t = TANBFP.p_t + TANBFP.p_tss;
323 TANBFP.p_o = 1;
324
325 TANBFP.price = ces_agg(TANBFP.p_t, TANBFP.p_o, TANBFP.eta, TANBFP.s_t);
326
327
328 TANBNR_FER.p_t = TANBNR_FER.p_t + TANBNR_FER.p_tss;
329 TANBNR_FER.p_o = 1;
330
331 TANBNR_FER.price = ces_agg(TANBNR_FER.p_t, TANBNR_FER.p_o, TANBNR_FER.eta, TANBNR_FER.s_t);
332
333
334 TANBNR_PIT.p_at = TANBNR_PIT.p_tss;
335 TANBNR_PIT.p_ao = TANBNR_PIT.e + TANBNR_PIT.ess;
336
337 TANBNR_PIT.price_a = ces_agg(TANBNR_PIT.p_at, TANBNR_PIT.p_ao, TANBNR_PIT.eta, TANBNR_PIT.s_t);
338
339
340 TANBNR_PIT.p_bt = TANBNR_PIT.p_tss ./ (TANBNR_PIT.e + TANBNR_PIT.ess);
341 TANBNR_PIT.p_bo = 1;
342
343 TANBNR_PIT.price_b = ces_agg(TANBNR_PIT.p_bt, TANBNR_PIT.p_bo, TANBNR_PIT.eta, TANBNR_PIT.s_t);
344
345
346 TANBFPNH.p_t = TANBFPNH.p_t + TANBFPNH.p_tss;
347 TANBFPNH.p_o = 1;
348
349 TANBFPNH.price = ces_agg(TANBFPNH.p_t, TANBFPNH.p_o, TANBFPNH.eta, TANBFPNH.s_t);
350
351
352
353 % Inflation
354 RAFAFP.infl = diff(RAFAFP.price) ./ RAFAFP.price(1:(end-1)) * 100;
355 RANBFP.infl = diff(RANBFP.price) ./ RANBFP.price(1:(end-1)) * 100;
356 TANBFP.infl = diff(TANBFP.price) ./ TANBFP.price(1:(end-1)) * 100;
357 TANBNR_FER.infl = diff(TANBNR_FER.price) ./ TANBNR_FER.price(1:(end-1)) * 100;
358 TANBNR_PIT.infl_a = diff(TANBNR_PIT.price_a) ./ TANBNR_PIT.price_a(1:(end-1)) * 100;
359 TANBNR_PIT.infl_b = diff(TANBNR_PIT.price_b) ./ TANBNR_PIT.price_b(1:(end-1)) * 100;
360 TANBFPNH.infl = diff(TANBFPNH.price) ./ TANBFPNH.price(1:(end-1)) * 100;
361
362
363 % consumption
364
365 % re-express eveything as levels
366 RAFAFP.c_at = RAFAFP.c_at + RAFAFP.c_atss;
367 RAFAFP.c_ao = RAFAFP.c_ao + RAFAFP.c_aoss;
368 RAFAFP.c_bt = RAFAFP.c_bt + RAFAFP.c_btss;
369 RAFAFP.c_bo = RAFAFP.c_bo + RAFAFP.c_boss;

```

```

370
371 % rule out negative consumption
372 RAFAFP.c_at = max(RAFAFP.c_at, eps);
373 RAFAFP.c_ao = max(RAFAFP.c_ao, eps);
374 RAFAFP.c_bt = max(RAFAFP.c_bt, eps);
375 RAFAFP.c_bo = max(RAFAFP.c_bo, eps);
376
377 % compute consumption basket
378 RAFAFP.c_a = ces_agg(RAFAFP.c_at, RAFAFP.c_ao, RAFAFP.eta, RAFAFP.s_t);
379 RAFAFP.c_b = ces_agg(RAFAFP.c_bt, RAFAFP.c_bo, RAFAFP.eta, RAFAFP.s_t);
380
381
382 RANBFP.c_at = RANBFP.c_at + RANBFP.c_atss;
383 RANBFP.c_ao = RANBFP.c_ao + RANBFP.c_aoss;
384 RANBFP.c_bt = RANBFP.c_bt + RANBFP.c_btss;
385 RANBFP.c_bo = RANBFP.c_bo + RANBFP.c_booss;
386
387 RANBFP.c_at = max(RANBFP.c_at, eps);
388 RANBFP.c_ao = max(RANBFP.c_ao, eps);
389 RANBFP.c_bt = max(RANBFP.c_bt, eps);
390 RANBFP.c_bo = max(RANBFP.c_bo, eps);
391
392 RANBFP.c_a = ces_agg(RANBFP.c_at, RANBFP.c_ao, RANBFP.eta, RANBFP.s_t);
393 RANBFP.c_b = ces_agg(RANBFP.c_bt, RANBFP.c_bo, RANBFP.eta, RANBFP.s_t);
394
395
396 TANBFP.c_art = TANBFP.c_art + TANBFP.c_artss;
397 TANBFP.c_aro = TANBFP.c_aro + TANBFP.c_aross;
398 TANBFP.c_aht = TANBFP.c_aht + TANBFP.c_ahtss;
399 TANBFP.c_aho = TANBFP.c_aho + TANBFP.c_ahoss;
400 TANBFP.c_bt = TANBFP.c_bt + TANBFP.c_btss;
401 TANBFP.c_bo = TANBFP.c_bo + TANBFP.c_booss;
402
403 TANBFP.c_art = max(TANBFP.c_art, eps);
404 TANBFP.c_aro = max(TANBFP.c_aro, eps);
405 TANBFP.c_aht = max(TANBFP.c_aht, eps);
406 TANBFP.c_aho = max(TANBFP.c_aho, eps);
407 TANBFP.c_bt = max(TANBFP.c_bt, eps);
408 TANBFP.c_bo = max(TANBFP.c_bo, eps);
409
410 TANBFP.c_ar = ces_agg(TANBFP.c_art, TANBFP.c_aro, TANBFP.eta, TANBFP.s_t);
411 TANBFP.c_ah = ces_agg(TANBFP.c_aht, TANBFP.c_aho, TANBFP.eta, TANBFP.s_t);
412 TANBFP.c_b = ces_agg(TANBFP.c_bt, TANBFP.c_bo, TANBFP.eta, TANBFP.s_t);
413
414
415 TANBNR_FER.c_art = TANBNR_FER.c_art + TANBNR_FER.c_artss;
416 TANBNR_FER.c_aro = TANBNR_FER.c_aro + TANBNR_FER.c_aross;
417 TANBNR_FER.c_aht = TANBNR_FER.c_aht + TANBNR_FER.c_ahtss;
418 TANBNR_FER.c_aho = TANBNR_FER.c_aho + TANBNR_FER.c_ahoss;
419 TANBNR_FER.c_bt = TANBNR_FER.c_bt + TANBNR_FER.c_btss;
420 TANBNR_FER.c_bo = TANBNR_FER.c_bo + TANBNR_FER.c_booss;
421
422 TANBNR_FER.c_art = max(TANBNR_FER.c_art, eps);
423 TANBNR_FER.c_aro = max(TANBNR_FER.c_aro, eps);
424 TANBNR_FER.c_aht = max(TANBNR_FER.c_aht, eps);
425 TANBNR_FER.c_aho = max(TANBNR_FER.c_aho, eps);
426 TANBNR_FER.c_bt = max(TANBNR_FER.c_bt, eps);
427 TANBNR_FER.c_bo = max(TANBNR_FER.c_bo, eps);
428
429 TANBNR_FER.c_ar = ces_agg(TANBNR_FER.c_art, TANBNR_FER.c_aro, TANBNR_FER.eta, TANBNR_FER.s_t);
430 TANBNR_FER.c_ah = ces_agg(TANBNR_FER.c_aht, TANBNR_FER.c_aho, TANBNR_FER.eta, TANBNR_FER.s_t);
431 TANBNR_FER.c_b = ces_agg(TANBNR_FER.c_bt, TANBNR_FER.c_bo, TANBNR_FER.eta, TANBNR_FER.s_t);
432
433
434 TANBNR_PIT.c_art = TANBNR_PIT.c_art + TANBNR_PIT.c_artss;
435 TANBNR_PIT.c_aro = TANBNR_PIT.c_aro + TANBNR_PIT.c_aross;
436 TANBNR_PIT.c_aht = TANBNR_PIT.c_aht + TANBNR_PIT.c_ahtss;
437 TANBNR_PIT.c_aho = TANBNR_PIT.c_aho + TANBNR_PIT.c_ahoss;
438 TANBNR_PIT.c_bt = TANBNR_PIT.c_bt + TANBNR_PIT.c_btss;
439 TANBNR_PIT.c_bo = TANBNR_PIT.c_bo + TANBNR_PIT.c_booss;
440
441 TANBNR_PIT.c_art = max(TANBNR_PIT.c_art, eps);
442 TANBNR_PIT.c_aro = max(TANBNR_PIT.c_aro, eps);
443 TANBNR_PIT.c_aht = max(TANBNR_PIT.c_aht, eps);

```

```

444 TANBNR_PIT.c_aho = max(TANBNR_PIT.c_aho, eps);
445 TANBNR_PIT.c_bt = max(TANBNR_PIT.c_bt, eps);
446 TANBNR_PIT.c_bo = max(TANBNR_PIT.c_bo, eps);
447
448 TANBNR_PIT.c_ar = ces_agg(TANBNR_PIT.c_art, TANBNR_PIT.c_aro, TANBNR_PIT.eta, TANBNR_PIT.s_t);
449 TANBNR_PIT.c_ah = ces_agg(TANBNR_PIT.c_aht, TANBNR_PIT.c_aho, TANBNR_PIT.eta, TANBNR_PIT.s_t);
450 TANBNR_PIT.c_b = ces_agg(TANBNR_PIT.c_bt, TANBNR_PIT.c_bo, TANBNR_PIT.eta, TANBNR_PIT.s_t);
451
452
453 TANBFPNH.c_art = TANBFPNH.c_art + TANBFPNH.c_artss;
454 TANBFPNH.c_aro = TANBFPNH.c_aro + TANBFPNH.c_arooss;
455 TANBFPNH.c_aht = TANBFPNH.c_aht + TANBFPNH.c_ahtss;
456 TANBFPNH.c_aho = TANBFPNH.c_aho + TANBFPNH.c_ahoss;
457 TANBFPNH.c_bt = TANBFPNH.c_bt + TANBFPNH.c_btss;
458 TANBFPNH.c_bo = TANBFPNH.c_bo + TANBFPNH.c_booss;
459
460 TANBFPNH.c_art = max(TANBFPNH.c_art, eps);
461 TANBFPNH.c_aro = max(TANBFPNH.c_aro, eps);
462 TANBFPNH.c_aht = max(TANBFPNH.c_aht, eps);
463 TANBFPNH.c_aho = max(TANBFPNH.c_aho, eps);
464 TANBFPNH.c_bt = max(TANBFPNH.c_bt, eps);
465 TANBFPNH.c_bo = max(TANBFPNH.c_bo, eps);
466
467 TANBFPNH.c_ar = ces_agg(TANBFPNH.c_art, TANBFPNH.c_aro, TANBFPNH.eta, TANBFPNH.s_art);
468 TANBFPNH.c_ah = ces_agg(TANBFPNH.c_aht, TANBFPNH.c_aho, TANBFPNH.eta, TANBFPNH.s_aht);
469 TANBFPNH.c_b = ces_agg(TANBFPNH.c_bt, TANBFPNH.c_bo, TANBFPNH.eta, TANBFPNH.s_bt);
470
471
472
473 % labour supply
474
475 RAFAFP.l = RAFAFP.l + RAFAFP.lss;
476
477 RANBFP.l = RANBFP.l + RANBFP.lss;
478
479 TANBFP.l = TANBFP.l + TANBFP.lss;
480 TANBFP.l_r = TANBFP.l_r + TANBFP.l_rss;
481 TANBFP.l_h = TANBFP.l_h + TANBFP.l_hss;
482
483 TANBNR_FER.l = TANBNR_FER.l + TANBNR_FER.lss;
484 TANBNR_FER.l_r = TANBNR_FER.l_r + TANBNR_FER.l_rss;
485 TANBNR_FER.l_h = TANBNR_FER.l_h + TANBNR_FER.l_hss;
486
487 TANBNR_PIT.l = TANBNR_PIT.l + TANBNR_PIT.lss;
488 TANBNR_PIT.l_r = TANBNR_PIT.l_r + TANBNR_PIT.l_rss;
489 TANBNR_PIT.l_h = TANBNR_PIT.l_h + TANBNR_PIT.l_hss;
490
491 TANBFPNH.l = TANBFPNH.l + TANBFPNH.lss;
492 TANBFPNH.l_r = TANBFPNH.l_r + TANBFPNH.l_rss;
493 TANBFPNH.l_h = TANBFPNH.l_h + TANBFPNH.l_hss;
494
495
496 % discount vector
497 betas = zeros(1, 100);
498
499 for t = 1:length(betas)
500
501     betas(1, t) = beta^(t - 1);
502
503 endfor
504
505
506
507 % welfare
508
509 RAFAFP.utility_a = log(RAFAFP.c_a) + psi * log(1 - RAFAFP.l);
510 RAFAFP.welfare_a = betas * RAFAFP.utility_a.';
511
512 RAFAFP.utility_b = log(RAFAFP.c_b);
513 RAFAFP.welfare_b = betas * RAFAFP.utility_b.';
514
515
516 RANBFP.utility_a = log(RANBFP.c_a) + psi * log(1 - RANBFP.l);
517 RANBFP.welfare_a = betas * RANBFP.utility_a.';

```

```

518
519 RANBFP.utility_b = log(RANBFP.c_b);
520 RANBFP.welfare_b = betas * RANBFP.utility_b.';
521
522
523 TANBFP.utility_ar = log(TANBFP.c_ar) + psi * log(1 - TANBFP.l_r);
524 TANBFP.welfare_ar = betas * TANBFP.utility_ar.';
525
526 TANBFP.utility_ah = log(TANBFP.c_ah) + psi * log(1 - TANBFP.l_h);
527 TANBFP.welfare_ah = betas * TANBFP.utility_ah.';
528
529 TANBFP.utility_b = log(TANBFP.c_b);
530 TANBFP.welfare_b = betas * TANBFP.utility_b.';
531
532
533 TANBNR_FER.utility_ar = log(TANBNR_FER.c_ar) + psi * log(1 - TANBNR_FER.l_r);
534 TANBNR_FER.welfare_ar = betas * TANBNR_FER.utility_ar.';
535
536 TANBNR_FER.utility_ah = log(TANBNR_FER.c_ah) + psi * log(1 - TANBNR_FER.l_h);
537 TANBNR_FER.welfare_ah = betas * TANBNR_FER.utility_ah.';
538
539 TANBNR_FER.utility_b = log(TANBNR_FER.c_b);
540 TANBNR_FER.welfare_b = betas * TANBNR_FER.utility_b.';
541
542
543 TANBNR_PIT.utility_ar = log(TANBNR_PIT.c_ar) + psi * log(1 - TANBNR_PIT.l_r);
544 TANBNR_PIT.welfare_ar = betas * TANBNR_PIT.utility_ar.';
545
546 TANBNR_PIT.utility_ah = log(TANBNR_PIT.c_ah) + psi * log(1 - TANBNR_PIT.l_h);
547 TANBNR_PIT.welfare_ah = betas * TANBNR_PIT.utility_ah.';
548
549 TANBNR_PIT.utility_b = log(TANBNR_PIT.c_b);
550 TANBNR_PIT.welfare_b = betas * TANBNR_PIT.utility_b.';
551
552
553 TANBFPNH.utility_ar = log(TANBFPNH.c_ar) + psi * log(1 - TANBFPNH.l_r);
554 TANBFPNH.welfare_ar = betas * TANBFPNH.utility_ar.';
555
556 TANBFPNH.utility_ah = log(TANBFPNH.c_ah) + psi * log(1 - TANBFPNH.l_h);
557 TANBFPNH.welfare_ah = betas * TANBFPNH.utility_ah.';
558
559 TANBFPNH.utility_b = log(TANBFPNH.c_b);
560 TANBFPNH.welfare_b = betas * TANBFPNH.utility_b.';
561
562
563
564 % import/export, trade balance in country A:
565 RAFAFP.import = RAFAFP.c_ao;
566 RAFAFP.export = RAFAFP.c_bt;
567 RAFAFP.nex = RAFAFP.export - RAFAFP.import;
568
569
570 RANBFP.import = RANBFP.c_ao;
571 RANBFP.export = RANBFP.c_bt;
572 RANBFP.nex = RANBFP.export - RANBFP.import;
573
574
575 TANBFP.import = chi * TANBFP.c_aro + (1 - chi) * TANBFP.c_aho;
576 TANBFP.export = TANBFP.c_bt;
577 TANBFP.nex = TANBFP.export - TANBFP.import;
578
579
580 TANBNR_FER.import = chi * TANBNR_FER.c_aro + (1 - chi) * TANBNR_FER.c_aho;
581 TANBNR_FER.export = TANBNR_FER.c_bt;
582 TANBNR_FER.nex = TANBNR_FER.export - TANBNR_FER.import;
583
584
585 TANBNR_PIT.import = chi * TANBNR_PIT.c_aro + (1 - chi) * TANBNR_PIT.c_aho;
586 TANBNR_PIT.export = TANBNR_PIT.c_bt;
587 TANBNR_PIT.nex = TANBNR_PIT.export - TANBNR_PIT.import;
588
589
590 TANBFPNH.import = chi * TANBFPNH.c_aro + (1 - chi) * TANBFPNH.c_aho;
591 TANBFPNH.export = TANBFPNH.c_bt;

```



```

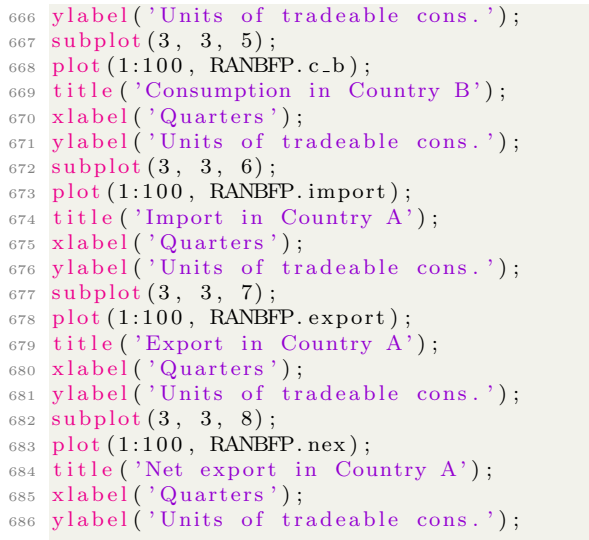
592 TANBFPNH.nex = TANBFPNH.export - TANBFPNH.import;
593
594
595 %
596 % 4. Plot IRFs
597 %
598
599 % RAFAFP
600
601 subplot(3, 3, 1);
602 plot(1:100, RAFAFP.output, 1:100);
603 title('Output in country A');
604 xlabel('Quarters');
605 ylabel('Units of tradeable cons. ');
606 subplot(3, 3, 2);
607 plot(1:100, RAFAFP.price, 1:100, RAFAFP.p_tss);
608 title('Price index in country A and B');
609 xlabel('Quarters');
610 ylabel('Units of oil consumption ');
611 subplot(3, 3, 3);
612 plot(1:99, RAFAFP.infl);
613 title('Inflation in country A and B');
614 xlabel('Quarters');
615 ylabel('Percent ');
616 subplot(3, 3, 4);
617 plot(1:100, RAFAFP.c_a);
618 title('Consumption in Country A');
619 xlabel('Quarters');
620 ylabel('Units of tradeable cons. ');
621 subplot(3, 3, 5);
622 plot(1:100, RAFAFP.c_b);
623 title('Consumption in Country B');
624 xlabel('Quarters');
625 ylabel('Units of tradeable cons. ');
626 subplot(3, 3, 6);
627 plot(1:100, RAFAFP.import);
628 title('Import in Country A');
629 xlabel('Quarters');
630 ylabel('Units of tradeable cons. ');
631 subplot(3, 3, 7);
632 plot(1:100, RAFAFP.export);
633 title('Export in Country A');
634 xlabel('Quarters');
635 ylabel('Units of tradeable cons. ');
636 subplot(3, 3, 8);
637 plot(1:100, RAFAFP.nex);
638 title('Net export in Country A');
639 xlabel('Quarters');
640 ylabel('Units of tradeable cons. ');
641
642 saveas(gcf, 'rafafp.jpg');
643
644
645 % RANBFP
646
647 subplot(3, 3, 1);
648 plot(1:100, RANBFP.output);
649 title('Output in country A');
650 xlabel('Quarters');
651 ylabel('Units of tradeable cons. ');
652 subplot(3, 3, 2);
653 plot(1:100, RANBFP.price);
654 title('Price index in country A and B');
655 xlabel('Quarters');
656 ylabel('Units of oil consumption ');
657 subplot(3, 3, 3);
658 plot(1:99, RANBFP.infl);
659 title('Inflation in country A and B');
660 xlabel('Quarters');
661 ylabel('Percent ');
662 subplot(3, 3, 4);
663 plot(1:100, RANBFP.c_a);
664 title('Consumption in Country A');
665 xlabel('Quarters');

```



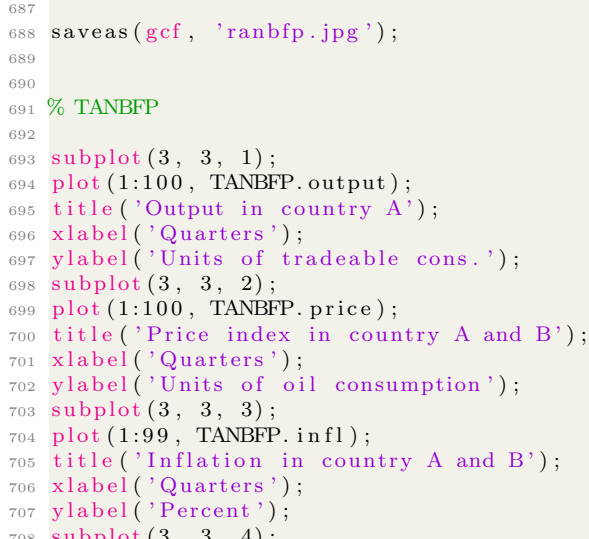
```

666 ylabel('Units of tradeable cons.');
```



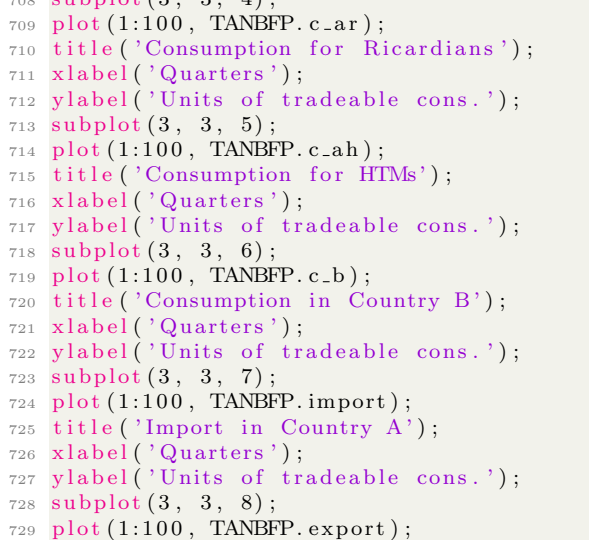
```

667 subplot(3, 3, 5);
668 plot(1:100, RANBFP.c_b);
669 title('Consumption in Country B');
670 xlabel('Quarters');
671 ylabel('Units of tradeable cons.');
```



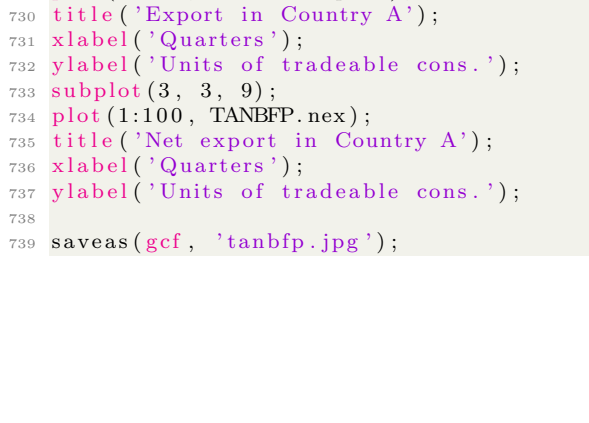
```

672 subplot(3, 3, 6);
673 plot(1:100, RANBFP.import);
674 title('Import in Country A');
675 xlabel('Quarters');
676 ylabel('Units of tradeable cons.');
```



```

677 subplot(3, 3, 7);
678 plot(1:100, RANBFP.export);
679 title('Export in Country A');
680 xlabel('Quarters');
681 ylabel('Units of tradeable cons.');
```



```

682 subplot(3, 3, 8);
683 plot(1:100, RANBFP.nex);
684 title('Net export in Country A');
685 xlabel('Quarters');
686 ylabel('Units of tradeable cons.');
```

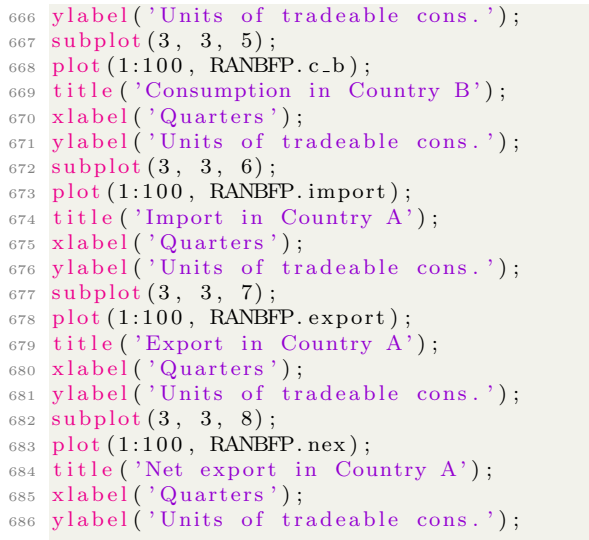


```

687
688 saveas(gcf, 'ranbfp.jpg');
```

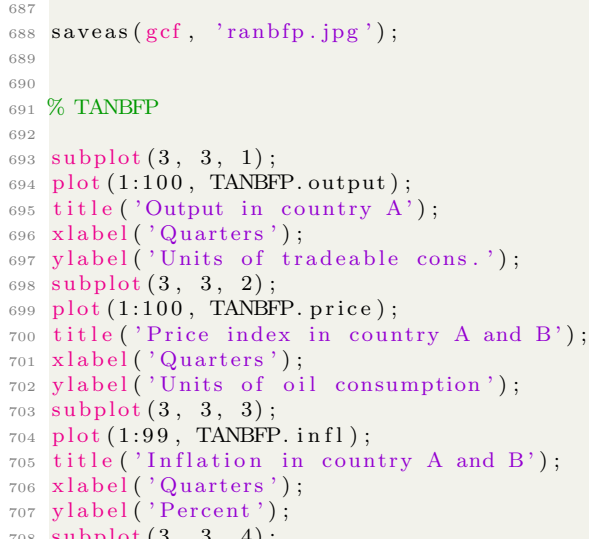
```

689
690
691 % TANBFP
692
693 subplot(3, 3, 1);
694 plot(1:100, TANBFP.output);
695 title('Output in country A');
696 xlabel('Quarters');
697 ylabel('Units of tradeable cons.');
```



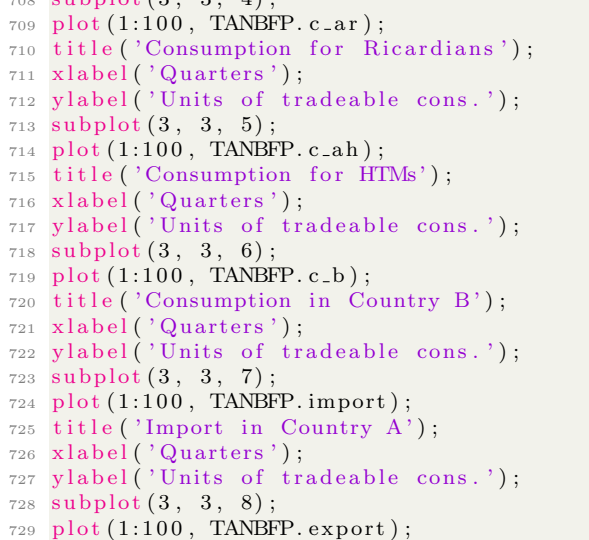
```

698 subplot(3, 3, 2);
699 plot(1:100, TANBFP.price);
700 title('Price index in country A and B');
```



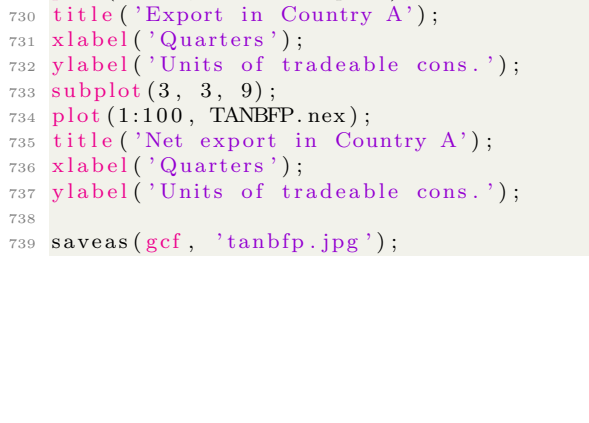
```

701 xlabel('Quarters');
702 ylabel('Units of oil consumption');
```



```

703 subplot(3, 3, 3);
704 plot(1:99, TANBFP.infl);
705 title('Inflation in country A and B');
```



```

706 xlabel('Quarters');
707 ylabel('Percent');
```



```

708 subplot(3, 3, 4);
709 plot(1:100, TANBFP.c_ar);
710 title('Consumption for Ricardians');
```



```

711 xlabel('Quarters');
```



```

712 ylabel('Units of tradeable cons.');
```



```

713 subplot(3, 3, 5);
714 plot(1:100, TANBFP.c_ah);
715 title('Consumption for HTMs');
```



```

716 xlabel('Quarters');
```



```

717 ylabel('Units of tradeable cons.');
```



```

718 subplot(3, 3, 6);
719 plot(1:100, TANBFP.c_b);
720 title('Consumption in Country B');
```



```

721 xlabel('Quarters');
```



```

722 ylabel('Units of tradeable cons.');
```



```

723 subplot(3, 3, 7);
724 plot(1:100, TANBFP.import);
725 title('Import in Country A');
```



```

726 xlabel('Quarters');
```



```

727 ylabel('Units of tradeable cons.');
```



```

728 subplot(3, 3, 8);
729 plot(1:100, TANBFP.export);
730 title('Export in Country A');
```



```

731 xlabel('Quarters');
```



```

732 ylabel('Units of tradeable cons.');
```



```

733 subplot(3, 3, 9);
734 plot(1:100, TANBFP.nex);
735 title('Net export in Country A');
```



```

736 xlabel('Quarters');
```



```

737 ylabel('Units of tradeable cons.');
```



```

738
739 saveas(gcf, 'tanbfp.jpg');
```

```

740
741
742
743
744
745 % TANBNR_FER
746
747 subplot(3, 3, 1);
748 plot(1:100, TANBNR_FER.output);
749 title('Output in country A');
750 xlabel('Quarters');
751 ylabel('Units of tradeable cons.');
```

```

752 subplot(3, 3, 2);
753 plot(1:100, TANBNR_FER.price);
754 title('Price index in country A and B');
755 xlabel('Quarters');
756 ylabel('Units of oil consumption');
```

```

757 subplot(3, 3, 3);
758 plot(1:99, TANBNR_FER.infl);
759 title('Inflation in country A and B');
760 xlabel('Quarters');
761 ylabel('Percent');
```

```

762 subplot(3, 3, 4);
763 plot(1:100, TANBNR_FER.c_ar);
764 title('Consumption for Ricardians');
```

```

765 xlabel('Quarters');
```

```

766 ylabel('Units of tradeable cons.');
```

```

767 subplot(3, 3, 5);
768 plot(1:100, TANBNR_FER.c_ah);
769 title('Consumption for HTMs');
```

```

770 xlabel('Quarters');
```

```

771 ylabel('Units of tradeable cons.');
```

```

772 subplot(3, 3, 6);
773 plot(1:100, TANBNR_FER.c_b);
774 title('Consumption in Country B');
```

```

775 xlabel('Quarters');
```

```

776 ylabel('Units of tradeable cons.');
```

```

777 subplot(3, 3, 7);
778 plot(1:100, TANBNR_FER.import);
779 title('Import in Country A');
```

```

780 xlabel('Quarters');
```

```

781 ylabel('Units of tradeable cons.');
```

```

782 subplot(3, 3, 8);
783 plot(1:100, TANBNR_FER.export);
784 title('Export in Country A');
```

```

785 xlabel('Quarters');
```

```

786 ylabel('Units of tradeable cons.');
```

```

787 subplot(3, 3, 9);
788 plot(1:100, TANBNR_FER.nex);
789 title('Net export in Country A');
```

```

790 xlabel('Quarters');
```

```

791 ylabel('Units of tradeable cons.');
```

```

792
793 saveas(gcf, 'tanbnr_fer.jpg');
```

```

794
795
796
797 % TANBNR_PIT
798
799 subplot(5, 3, 1);
800 plot(1:100, TANBNR_PIT.output);
801 title('Output in country A');
```

```

802 xlabel('Quarters');
```

```

803 ylabel('Units of tradeable cons.');
```

```

804 subplot(5, 3, 2);
805 plot(1:100, TANBNR_PIT.price_a);
806 title('Price index in country A');
```

```

807 xlabel('Quarters');
```

```

808 ylabel('Units of oil consumption');
```

```

809 subplot(5, 3, 3);
810 plot(1:100, TANBNR_PIT.price_b);
811 title('Price index in country B');
```

```

812 xlabel('Quarters');
```

```

813 ylabel('Units of oil consumption');
```

```

814 subplot(5, 3, 4);
815 plot(1:99, TANBNR_PIT.infl_a);
816 title('Inflation in country A');
817 xlabel('Quarters');
818 ylabel('Percent');
819 subplot(5, 3, 5);
820 plot(1:99, TANBNR_PIT.infl_b);
821 title('Inflation in country B');
822 xlabel('Quarters');
823 ylabel('Percent');
824 subplot(5, 3, 6);
825 plot(1:100, TANBNR_PIT.c_ar);
826 title('Consumption for Ricardians');
827 xlabel('Quarters');
828 ylabel('Units of tradeable cons. ');
829 subplot(5, 3, 7);
830 plot(1:100, TANBNR_PIT.c_ah);
831 title('Consumption for HTMs');
832 xlabel('Quarters');
833 ylabel('Units of tradeable cons. ');
834 subplot(5, 3, 8);
835 plot(1:100, TANBNR_PIT.c_b);
836 title('Consumption in Country B');
837 xlabel('Quarters');
838 ylabel('Units of tradeable cons. ');
839 subplot(5, 3, 9);
840 plot(1:100, TANBNR_PIT.import);
841 title('Import in Country A');
842 xlabel('Quarters');
843 ylabel('Units of tradeable cons. ');
844 subplot(5, 3, 10);
845 plot(1:100, TANBNR_PIT.export);
846 title('Export in Country A');
847 xlabel('Quarters');
848 ylabel('Units of tradeable cons. ');
849 subplot(5, 3, 11);
850 plot(1:100, TANBNR_PIT.nex);
851 title('Net export in Country A');
852 xlabel('Quarters');
853 ylabel('Units of tradeable cons. ');
854
855 saveas(gcf, 'tanbnr_pit.jpg');
856
857
858
859 % TANBFPNH
860
861 subplot(3, 3, 1);
862 plot(1:100, TANBFP.output);
863 title('Output in country A');
864 xlabel('Quarters');
865 ylabel('Units of tradeable cons. ');
866 subplot(3, 3, 2);
867 plot(1:100, TANBFPNH.price);
868 title('Price index in country A and B');
869 xlabel('Quarters');
870 ylabel('Units of oil consumption');
871 subplot(3, 3, 3);
872 plot(1:99, TANBFPNH.infl);
873 title('Inflation in country A and B');
874 xlabel('Quarters');
875 ylabel('Percent');
876 subplot(3, 3, 4);
877 plot(1:100, TANBFPNH.c_ar);
878 title('Consumption for Ricardians');
879 xlabel('Quarters');
880 ylabel('Units of tradeable cons. ');
881 subplot(3, 3, 5);
882 plot(1:100, TANBFPNH.c_ah);
883 title('Consumption for HTMs');
884 xlabel('Quarters');
885 ylabel('Units of tradeable cons. ');
886 subplot(3, 3, 6);
887 plot(1:100, TANBFPNH.c_b);

```

```

888 title('Consumption in Country B');
889 xlabel('Quarters');
890 ylabel('Units of tradeable cons.');
```



```

891 subplot(3, 3, 7);
892 plot(1:100, TANBFPNH.import);
893 title('Import in Country A');
894 xlabel('Quarters');
895 ylabel('Units of tradeable cons.');
```



```

896 subplot(3, 3, 8);
897 plot(1:100, TANBFPNH.export);
898 title('Export in Country A');
899 xlabel('Quarters');
900 ylabel('Units of tradeable cons.');
```



```

901 subplot(3, 3, 9);
902 plot(1:100, TANBFPNH.nex);
903 title('Net export in Country A');
904 xlabel('Quarters');
905 ylabel('Units of tradeable cons.');
```



```

906
907 saveas(gcf, 'tanbfpnh.jpg');
```

## Appendix A2: Parameter File

```

1
2 %-----
3 %
4 % Title: International Macro–Finance Problem Set 4, parameter file
5 % Author: —
6 % Date: 10/01/2024
7 % Description: Parameter file for the final problem set
8 %
9 %-----
10
11
12 %-----
13 % 0. Housekeeping (close all graphic windows)
14 %-----
15
16 close all;
17 clear all;
18
19 %-----
20 % 1. Defining parameters
21 %-----
22
23
24 % key moments to target:
25 % y_o = 12.1 * 365 * 90 * 10^(-6) = .4 Saudi oil production in 2022, tn euro
26 % c_ao = .46 EU import of oil from Saudi Arabia in 2022, tn euro
27 % l = .22; average weekly work hours in the EU (37.5 / (24 * 7))
28 % y_t * p_t = 16.75 EU GDP in 2022, tn USD
29
30 % this implies:
31 % y_t = l^(1 - alpha) = .35 with alpha = .3
32 % p_t = 16.75 / .35 = 47.86
33
34
35 par.alpha = .3; % income share of capital
36 par.beta = .99;
37 par.psi = 1.5; % elasticity of labour supply, normally between 1 and 2 (see Blundell et al. 2000)
38 par.phi = 10; % strength of the nominal rigidity
39 par.eta = .4; % elasticity of substitution between the tradeable good and oil
40 par.s_t = .75; % home bias towards tradeable in both countries
41 par.s_art = .78; % home bias towards tradeable for ricardians
42 par.s_aht = .9; % home bias towards tradeable for htms
43 par.s_bt = .8; % home bias towards tradeable in country b
44 par.theta = .4; % elasticity of substitution between goods in the tradeable sector
45 par.rho = .9; % persistence of shock
46 par.gamma = .1; % mean oil endowment of country b (y_bbar = gamma / (1 - rho))
47 par.chi = .8; % proportion of ricardian HHs in country a
```

## Appendix B: RAFAFP mod file

```
1
2 %
3 %
4 % Title: International Macro–Finance Final Assignment, model file
5 % Author: —
6 % Date: 25/12/2023
7 % Description: Representative agent, financial autarky, flexible prices
8 %
9 %
10
11
12 %
13 % 0. Housekeeping (close all graphic windows)
14 %
15
16 close all;
17
18 %
19 % 1. Defining variables
20 %
21
22 var c_ao c_bo c_at c_bt l y_t y_o p_t w pi;
23
24 varexo err_yo;
25
26
27 %
28 % 2. Calibration
29 %
30
31 parameters alpha psi eta s_t theta rho gamma;
32
33 load parameters;
34
35 set_param_value('alpha', par.alpha);
36 set_param_value('psi', par.psi);
37 set_param_value('eta', par.eta);
38 set_param_value('s_t', par.s_t);
39 set_param_value('theta', par.theta);
40 set_param_value('rho', par.rho);
41 set_param_value('gamma', par.gamma);
42
43
44 %
45 % 3. Model
46 %
47
48
49 model;
50
51 % consumption allocation between oil and tradeable in country a
52 (1 - s_t) * c_ao^(-1 / eta) - s_t * c_at^(-1 / eta) / p_t;
53
54 % consumption allocation between oil and tradeable in country b
55 (1 - s_t) * c_bo^(-1 / eta) - s_t * c_bt^(-1 / eta) / p_t;
56
57 % HH's labour supply decision in country a
58 psi - (1 - s_t) * c_ao^(-1 / eta) / (s_t * c_at^(1 - 1 / eta) +
59      (1 - s_t) * c_ao^(1 - 1 / eta)) *
60      w * (1 - l);
61
62 % HH's budget constraint in country a
63 p_t * c_at + c_ao - w * l - pi;
64
65 % HH's budget constraint in country b
66 c_bo + p_t * c_bt - y_o;
67
68 % firm's FOC
69 (1 - theta) * (1 - alpha) * p_t *
70      l^((1 - theta) * (1 - alpha) - 1) * y_t^theta - w;
71
72 % firm's profit
73 p_t * l^(1 - alpha) - w * l - pi;
```

```

74
75 % tradeable goods market-clearing condition
76 y_t = c_at + c_bt;
77
78 % oil market-clearing condition
79 y_o = c_ao + c_bo;
80
81 % oil endowment, stochastic process
82 gamma + rho * y_o(-1) - err_yo - y_o;
83
84 end;
85
86
87 %
88 % 4. Steady state
89 %
90
91
92 initval;
93
94 l = .22;
95 c_ao = .46;
96 y_t = l^(1 - alpha);
97 p_t = 16.75 / y_t;
98 c_at = s_t^eta * c_ao / (p_t^eta * (1 - s_t)^eta);
99 c_bt = y_t - c_at;
100 y_o = gamma / (1 - rho);
101 c_bo = y_o - c_ao;
102 w = (1 - theta) * (1 - alpha) * p_t *
103     l^((1 - theta) * (1 - alpha) - 1) * y_t^theta;
104 pi = p_t * l^(1 - alpha) - w * l;
105
106 end;
107
108 steady(maxit = 1000, solve_algo = 1);
109
110
111 %
112 % 5. Impulse response function
113 %
114
115
116 shocks;
117
118 var err_yo;
119 stderr .1;
120
121 end;
122
123
124 stoch_simul(order=1, irf=100, irf_plot_threshold=0) c_ao
125     c_bo c_at c_bt l y_t y_o p_t w pi;

```

## Appendix C: RANBFP mod file

```
1
2 %
3 %
4 % Title: International Macro–Finance Final Assignment, model file
5 % Author: —
6 % Date: 25/12/2023
7 % Description: Representative agent, nominal bond, flexible prices
8 %
9 %
10
11
12 %
13 % 0. Housekeeping (close all graphic windows)
14 %
15
16 close all;
17
18 %
19 % 1. Defining variables
20 %
21
22 var c_ao c_bo c_at c_bt l y_t y_o p_t w pi n_a n_b r_star;
23
24 varexo err_yo;
25
26
27 %
28 % 2. Calibration
29 %
30
31 parameters alpha beta psi eta s_t theta rho gamma;
32
33 load parameters;
34
35 set_param_value('alpha', par.alpha);
36 set_param_value('beta', par.beta);
37 set_param_value('psi', par.psi);
38 set_param_value('eta', par.eta);
39 set_param_value('s_t', par.s_t);
40 set_param_value('theta', par.theta);
41 set_param_value('rho', par.rho);
42 set_param_value('gamma', par.gamma);
43
44 s_t = .8;
45 theta = .65;
46
47
48 %
49 % 3. Model
50 %
51
52
53 model;
54
55 % consumption allocation between oil and tradeable in country a
56 (1 - s_t) * c_ao^(-1 / eta) - s_t * c_at^(-1 / eta) / p_t;
57
58 % consumption allocation between oil and tradeable in country b
59 (1 - s_t) * c_bo^(-1 / eta) - s_t * c_bt^(-1 / eta) / p_t;
60
61 % Euler equation in country a
62 c_ao^(-1 / eta) / (s_t * c_at^(1 - 1 / eta) +
63 (1 - s_t) * c_ao^(1 - 1 / eta))
64 - beta * r_star *
65 c_ao(+1)^(-1 / eta) / (s_t * c_at(+1)^(1 - 1 / eta) +
66 (1 - s_t) * c_ao(+1)^(1 - 1 / eta));
67
68 % Euler equation in country b
69 c_bo^(-1 / eta) / (s_t * c_bt^(1 - 1 / eta) +
70 (1 - s_t) * c_bo^(1 - 1 / eta)) -
71 beta * r_star *
72 c_bo(+1)^(-1 / eta) / (s_t * c_bt(+1)^(1 - 1 / eta) +
73 (1 - s_t) * c_bo(+1)^(1 - 1 / eta));
```

```

74
75 % HH's labour supply decision in country a
76 psi = (1 - s_t) * c_ao^(-1 / eta) / (s_t * c_at^(1 - 1 / eta) +
77     (1 - s_t) * c_ao^(1 - 1 / eta)) *
78     w * (1 - l);
79
80 % HH's budget constraint in country a
81 p_t * c_at + c_ao - w * l - pi = (n_a(-1) - n_a / r_star);
82
83 % HH's budget constraint in country b
84 c_bo + p_t * c_bt - y_o = (n_b(-1) - n_b / r_star);
85
86 % firm's FOC
87 (1 - theta) * (1 - alpha) * p_t *
88     l^((1 - theta) * (1 - alpha) - 1) * y_t^theta - w;
89
90 % firm's profit
91 p_t * l^(1 - alpha) - w * l - pi;
92
93 % tradeable goods market-clearing condition
94 y_t - c_at - c_bt;
95
96 % oil market-clearing condition
97 y_o - c_ao - c_bo;
98
99 % bond market clearing condition
100 n_a + n_b;
101
102 % oil endowment, stochastic process
103 gamma + rho * y_o(-1) - err_yo - y_o;
104
105 end;
106
107
108 %-----
109 % 4. Steady state
110 %-----
111
112
113 initval;
114
115 l = .22;
116 c_ao = .46;
117 y_t = l^(1 - alpha); % from production function
118 p_t = 16.75 / y_t; % from target GDP in country a
119 c_at = s_t^eta * c_ao / (p_t^eta * (1 - s_t)^eta); % from tradeable-oil tradeoff
120 c_bt = y_t - c_at; % from tradeable MC
121 y_o = gamma / (1 - rho); % from oil stochastic process
122 c_bo = y_o - c_ao; % from oil MC
123 w = (1 - theta) * (1 - alpha) * p_t *
124     l^((1 - theta) * (1 - alpha) - 1) * y_t^theta; % from firm's FOC
125 pi = p_t * l^(1 - alpha) - w * l; % from firm's profit
126 r_star = 1 / beta; % from SS Euler
127 n_a = r_star / (r_star - 1) * (p_t * c_at + c_ao - w * l - pi); % from the BC
128 n_b = - n_a; % from bond MC
129
130 end;
131
132 steady(maxit = 1000);
133
134
135 %-----
136 % 5. Impulse response function
137 %-----
138
139 initval;
140
141 c_ao = 0.425983;
142 c_bo = 0.174017;
143 c_at = 0.318506;
144 c_bt = 0.130112;
145 l = 0.318186;
146 y_t = 0.448618;
147 y_o = 0.6;

```



```

148 p_t = 30.8469;
149 w = 4.56664;
150 pi = 12.3854;
151 n_a = -358.757;
152 n_b = 358.757;
153 r_star = 1.0101;
154
155
156 end;
157
158 endval;
159
160 n_a = -358.757;
161 n_b = 358.757;
162
163 end;
164
165
166 shocks;
167
168 var err_yo;
169 stderr .1;
170
171 end;
172
173
174 stoch_simul(order=1, irf=100, irf_plot_threshold=0) c_ao
175 c_bo c_at c_bt l y_t y_o p_t w pi r_star
176 n_a n_b;
177
178 model_diagnostics;

```

## Appendix D: TANBFP mod file

```
1
2 %
3 %
4 % Title: International Macro-Finance Final Assignment, model file
5 % Author: —
6 % Date: 25/12/2023
7 % Description: Two agent, nominal bond, flexible prices
8 %
9 %
10
11
12 %
13 % 0. Housekeeping (close all graphic windows)
14 %
15
16 close all;
17
18 %
19 % 1. Defining variables
20 %
21
22 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
23     n_a n_b r_star pi_r pi_h;
24
25 varexo err_yo;
26
27
28 %
29 % 2. Calibration
30 %
31
32 parameters alpha beta psi eta s_t theta rho gamma chi;
33 load parameters;
34
35 set_param_value('alpha', par.alpha);
36 set_param_value('beta', par.beta);
37 set_param_value('psi', par.psi);
38 set_param_value('eta', par.eta);
39 set_param_value('s_t', par.s_t);
40 set_param_value('theta', par.theta);
41 set_param_value('rho', par.rho);
42 set_param_value('gamma', par.gamma);
43 set_param_value('chi', par.chi);
44
45 s_t = .8;
46 theta = .5;
47
48
49 %
50 % 3. Model
51 %
52 %
53
54 model;
55
56
57 % consumption allocation between oil and tradeable for htm HHs in country a
58 (1 - s_t) * c_aho^(-1 / eta) - s_t * c_aht^(-1 / eta) / p_t;
59
60 % consumption allocation between oil and tradeable for ricardian HHs in country a
61 (1 - s_t) * c_aro^(-1 / eta) - s_t * c_art^(-1 / eta) / p_t;
62
63 % consumption allocation between oil and tradeable in country b
64 (1 - s_t) * c_bo^(-1 / eta) - s_t * c_bt^(-1 / eta) / p_t;
65
66 % Euler equation for ricardian HHs in country a
67 c_aro^(-1 / eta) / (s_t * c_art^(1 - 1 / eta) +
68     (1 - s_t) * c_aro^(1 - 1 / eta)) -
69     beta * r_star *
70     c_aro(+1)^(-1 / eta) / (s_t * c_art(+1)^(1 - 1 / eta) +
71     (1 - s_t) * c_aro(+1)^(1 - 1 / eta));
72
73 % Euler equation in country b
```

```

74 c_bo^(-1 / eta) / (s_t * c_bt^(1 - 1 / eta) +
75 (1 - s_t) * c_bo^(1 - 1 / eta)) -
76 beta * r_star *
77 c_bo(+1)^(1 - 1 / eta) / (s_t * c_bt(+1)^(1 - 1 / eta) +
78 (1 - s_t) * c_bo(+1)^(1 - 1 / eta));
79
80 % labour supply decision for htm HHs in country a
81 psi - (1 - s_t) * c_aho^(-1 / eta) / (s_t * c_aht^(1 - 1 / eta) +
82 (1 - s_t) * c_aho^(1 - 1 / eta)) *
83 w * (1 - l_h);
84
85 % labour supply decision for ricardian HHs in country a
86 psi - (1 - s_t) * c_aro^(-1 / eta) / (s_t * c_art^(1 - 1 / eta) +
87 (1 - s_t) * c_aro^(1 - 1 / eta)) *
88 w * (1 - l_r);
89
90 % htm's budget constraint in country a
91 p_t * c_aht + c_aho - w * l_h - pi_h;
92
93 % ricardian's budget constraint in country a
94 p_t * c_art + c_aro - w * l_r - pi_r - (n_a(-1) - n_a / r_star);
95
96 % HH's budget constraint in country b
97 c_bo + p_t * c_bt - y_o - (n_b(-1) - n_b / r_star);
98
99 % firm's FOC
100 (1 - theta) * (1 - alpha) * p_t *
101 l^((1 - theta) * (1 - alpha) - 1) * y_t^theta - w;
102
103 % firm's profit
104 p_t * l^(1 - alpha) - w * l - pi;
105
106 % profit distribution
107 chi * pi_r + (1 - chi) * pi_h - pi;
108
109 % fix ricardian's profit
110 pi_r - pi;
111
112 % tradeable goods market-clearing condition
113 y_t - (1 - chi) * c_aht - chi * c_art - c_bt;
114
115 % oil market-clearing condition
116 y_o - (1 - chi) * c_aho - chi * c_aro - c_bo;
117
118 % labour market-clearing condition
119 chi * l_r + (1 - chi) * l_h - l;
120
121 % bond market clearing condition
122 chi * n_a + n_b;
123
124 % oil endowment, stochastic process
125 gamma + rho * y_o(-1) - err_yo - y_o;
126
127 end;
128
129
130 %
131 % 4. Steady state
132 %
133
134
135 initval;
136
137 l_r = .22;
138 l_h = .22;
139 c_aro = .46;
140 c_aho = .46;
141 l = chi * l_r + (1 - chi) * l_h;
142 y_t = l^(1 - alpha); % from production function
143 p_t = 16.75 / y_t; % from target GDP in country a
144 c_art = s_t^eta * c_aro / (p_t^eta * (1 - s_t)^eta); % from tradeable-oil tradeoff
145 c_aht = s_t^eta * c_aho / (p_t^eta * (1 - s_t)^eta);
146 c_bt = y_t - chi * c_art - (1 - chi) * c_aht; % from tradeable MC
147 y_o = gamma / (1 - rho); % from oil stochastic process

```

```

148 c_bo = y_o - chi * c_aro - (1 - chi) * c_aho; % from oil MC
149 w = (1 - theta) * (1 - alpha) * p_t *
150     l^((1 - theta) * (1 - alpha) - 1) * y_t^theta; % from firm's FOC
151 pi = p_t * l^(1 - alpha) - w * l; % from firm's profit
152 pi_r = pi;
153 pi_h = (pi - chi * pi_r) / (1 - chi);
154 r_star = 1 / beta; % from SS Euler
155 n_a = r_star / (r_star - 1) * (p_t * c_art + c_aro - w * l - pi_r); % from the BC
156 n_b = - chi * n_a; % from bond MC
157
158 end;
159
160 steady(maxit = 1000, solve_algo = 3);
161
162
163
164 %
165 % 5. Impulse response function
166 %
167
168
169 shocks;
170
171 var err_yo;
172 stderr .1;
173
174 end;
175
176
177 stoch_simul(order=1, irf=100, irf_plot_threshold=0)
178     c_aho c_aro c_bo c_aht c_art c_bt l_h
179     l_r l y_t y_o p_t w pi n_a n_b r_star;

```

## Appendix E: TANBFP mod file

```
1
2 %
3 %
4 % Title: International Macro-Finance Final Assignment, model file
5 % Author: —
6 % Date: 25/12/2023
7 % Description: Two agent, nominal bond, flexible prices
8 %
9 %
10
11
12 %
13 % 0. Housekeeping (close all graphic windows)
14 %
15
16 close all;
17
18 %
19 % 1. Defining variables
20 %
21
22 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
23     n_a n_b r_star pi_r pi_h;
24
25 varexo err_yo;
26
27
28 %
29 % 2. Calibration
30 %
31
32 parameters alpha beta psi eta s_t theta rho gamma chi;
33 load parameters;
34
35 set_param_value('alpha', par.alpha);
36 set_param_value('beta', par.beta);
37 set_param_value('psi', par.psi);
38 set_param_value('eta', par.eta);
39 set_param_value('s_t', par.s_t);
40 set_param_value('theta', par.theta);
41 set_param_value('rho', par.rho);
42 set_param_value('gamma', par.gamma);
43 set_param_value('chi', par.chi);
44
45 s_t = .8;
46 theta = .5;
47
48
49
50 %
51 % 3. Model
52 %
53
54
55 model;
56
57 % consumption allocation between oil and tradeable for htm HHs in country a
58 (1 - s_t) * c_aho^(-1 / eta) - s_t * c_aht^(-1 / eta) / p_t;
59
60 % consumption allocation between oil and tradeable for ricardian HHs in country a
61 (1 - s_t) * c_aro^(-1 / eta) - s_t * c_art^(-1 / eta) / p_t;
62
63 % consumption allocation between oil and tradeable in country b
64 (1 - s_t) * c_bo^(-1 / eta) - s_t * c_bt^(-1 / eta) / p_t;
65
66 % Euler equation for ricardian HHs in country a
67 c_aro^(-1 / eta) / (s_t * c_art^(1 - 1 / eta) +
68     (1 - s_t) * c_aro^(1 - 1 / eta)) -
69     beta * r_star *
70     c_aro(+1)^(-1 / eta) / (s_t * c_art(+1)^(1 - 1 / eta) +
71     (1 - s_t) * c_aro(+1)^(1 - 1 / eta));
72
73 % Euler equation in country b
```

```

74 c_bo^(-1 / eta) / (s_t * c_bt^(1 - 1 / eta) +
75 (1 - s_t) * c_bo^(1 - 1 / eta)) -
76 beta * r_star *
77 c_bo(+1)^(-1 / eta) / (s_t * c_bt(+1)^(1 - 1 / eta) +
78 (1 - s_t) * c_bo(+1)^(1 - 1 / eta));
79
80 % labour supply decision for htm HHs in country a
81 psi - (1 - s_t) * c_aho^(-1 / eta) / (s_t * c_aht^(1 - 1 / eta) +
82 (1 - s_t) * c_aho^(1 - 1 / eta)) *
83 w * (1 - l_h);
84
85 % labour supply decision for ricardian HHs in country a
86 psi - (1 - s_t) * c_aro^(-1 / eta) / (s_t * c_art^(1 - 1 / eta) +
87 (1 - s_t) * c_aro^(1 - 1 / eta)) *
88 w * (1 - l_r);
89
90 % htm's budget constraint in country a
91 p_t * c_aht + c_aho - w * l_h - pi_h;
92
93 % ricardian's budget constraint in country a
94 p_t * c_art + c_aro - w * l_r - pi_r - (n_a(-1) - n_a / r_star);
95
96 % HH's budget constraint in country b
97 c_bo + p_t * c_bt - y_o - (n_b(-1) - n_b / r_star);
98
99 % firm's FOC
100 (1 - theta) * (1 - alpha) * p_t *
101 l^((1 - theta) * (1 - alpha) - 1) * y_t^theta - w;
102
103 % firm's profit
104 p_t * l^(1 - alpha) - w * l - pi;
105
106 % profit distribution
107 chi * pi_r + (1 - chi) * pi_h - pi;
108
109 % fix ricardian's profit
110 pi_r - pi;
111
112 % tradeable goods market-clearing condition
113 y_t - (1 - chi) * c_aht - chi * c_art - c_bt;
114
115 % oil market-clearing condition
116 y_o - (1 - chi) * c_aho - chi * c_aro - c_bo;
117
118 % labour market-clearing condition
119 chi * l_r + (1 - chi) * l_h - l;
120
121 % bond market clearing condition
122 chi * n_a + n_b;
123
124 % oil endowment, stochastic process
125 gamma + rho * y_o(-1) - err_yo - y_o;
126
127 end;
128
129
130 %
131 % 4. Steady state
132 %
133
134
135 initval;
136
137 l_r = .22;
138 l_h = .22;
139 c_aro = .46;
140 c_aho = .46;
141 l = chi * l_r + (1 - chi) * l_h;
142 y_t = l^(1 - alpha); % from production function
143 p_t = 16.75 / y_t; % from target GDP in country a
144 c_art = s_t^eta * c_aro / (p_t^eta * (1 - s_t)^eta); % from tradeable-oil tradeoff
145 c_aht = s_t^eta * c_aho / (p_t^eta * (1 - s_t)^eta);
146 c_bt = y_t - chi * c_art - (1 - chi) * c_aht; % from tradeable MC
147 y_o = gamma / (1 - rho); % from oil stochastic process

```

```

148 c_bo = y_o - chi * c_aro - (1 - chi) * c_aho; % from oil MC
149 w = (1 - theta) * (1 - alpha) * p_t *
150     l^((1 - theta) * (1 - alpha) - 1) * y_t^theta; % from firm's FOC
151 pi = p_t * l^(1 - alpha) - w * l; % from firm's profit
152 pi_r = pi;
153 pi_h = (pi - chi * pi_r) / (1 - chi);
154 r_star = 1 / beta; % from SS Euler
155 n_a = r_star / (r_star - 1) * (p_t * c_art + c_aro - w * l - pi_r); % from the BC
156 n_b = - chi * n_a; % from bond MC
157
158 end;
159
160 steady(maxit = 1000, solve_algo = 3);
161
162
163
164 %
165 % 5. Impulse response function
166 %
167
168
169 shocks;
170
171 var err_yo;
172 stderr .1;
173
174 end;
175
176
177 stoch_simul(order=1, irf=100, irf_plot_threshold=0)
178     c_aho c_aro c_bo c_aht c_art c_bt l_h
179     l_r l y_t y_o p_t w pi n_a n_b r_star;

```

## Appendix F: TANBNR FER mod file

```
1
2 %
3 %
4 % Title: International Macro–Finance Final Assignment, model file
5 % Author: —
6 % Date: 25/12/2023
7 % Description: Two agent, nominal bond, nominal rigidities, fixed exchange rate
8 %
9 %
10
11
12 %
13 % 0. Housekeeping (close all graphic windows)
14 %
15
16 close all;
17
18 %
19 % 1. Defining variables
20 %
21
22 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
23      n_a n_b r_star pi_r pi_h;
24
25 varexo err_yo;
26
27
28 %
29 % 2. Calibration
30 %
31
32 parameters alpha beta psi phi eta s_t theta rho gamma chi;
33
34 load parameters;
35
36 set_param_value('alpha', par.alpha);
37 set_param_value('beta', par.beta);
38 set_param_value('psi', par.psi);
39 set_param_value('phi', par.phi);
40 set_param_value('eta', par.eta);
41 set_param_value('s_t', par.s_t);
42 set_param_value('theta', par.theta);
43 set_param_value('rho', par.rho);
44 set_param_value('gamma', par.gamma);
45 set_param_value('chi', par.chi);
46
47 s_t = .8;
48 theta = .5;
49
50
51 %
52 % 3. Model
53 %
54
55
56 model;
57
58 % consumption allocation between oil and tradeable for htm HHs in country a
59 (1 - s_t) * c_aho^(-1 / eta) - s_t * c_aht^(-1 / eta) / p_t;
60
61 % consumption allocation between oil and tradeable for ricardian HHs in country a
62 (1 - s_t) * c_aro^(-1 / eta) - s_t * c_art^(-1 / eta) / p_t;
63
64 % consumption allocation between oil and tradeable in country b
65 (1 - s_t) * c_bo^(-1 / eta) - s_t * c_bt^(-1 / eta) / p_t;
66
67 % Euler equation for ricardian HHs in country a
68 c_aro^(-1 / eta) / (s_t * c_art^(1 - 1 / eta) +
69      (1 - s_t) * c_aro^(1 - 1 / eta)) -
70      beta * r_star *
71      c_aro(+1)^(-1 / eta) / (s_t * c_art(+1)^(1 - 1 / eta) +
72      (1 - s_t) * c_aro(+1)^(1 - 1 / eta));
73
```



```

74 % Euler equation in country b
75 c_bo^(-1 / eta) / (s_t * c_bt^(1 - 1 / eta) +
76 (1 - s_t) * c_bo^(1 - 1 / eta)) -
77     beta * r_star *
78     c_bo(+1)^(1 - 1 / eta) / (s_t * c_bt(+1)^(1 - 1 / eta) +
79 (1 - s_t) * c_bo(+1)^(1 - 1 / eta));
80
81 % labour supply decision for htm HHs in country a
82 psi - (1 - s_t) * c_aho^(-1 / eta) / (s_t * c_aht^(1 - 1 / eta) +
83 (1 - s_t) * c_aho^(1 - 1 / eta)) *
84     w * (1 - l_h);
85
86 % labour supply decision for ricardian HHs in country a
87 psi - (1 - s_t) * c_aro^(-1 / eta) / (s_t * c_art^(1 - 1 / eta) +
88 (1 - s_t) * c_aro^(1 - 1 / eta)) *
89     w * (1 - l_r);
90
91 % htm's budget constraint in country a
92 p_t * c_aht + c_aho - w * l_h - pi_h;
93
94 % ricardian's budget constraint in country a
95 p_t * c_art + c_aro - w * l_r - pi_r - (n_a(-1) - n_a / r_star);
96
97 % HH's budget constraint in country b
98 c_bo + p_t * c_bt - y_o - (n_b(-1) - n_b / r_star);
99
100 % firm's FOC
101 p_t * (1 - theta) * (1 - alpha) * l^((1 - theta) * (1 - alpha) - 1) * y_t^theta -
102     w - phi * (p_t * y_t^theta * l(-1)^(theta * (1 - alpha)) /
103 (p_t(-1) * l^(theta * (1 - alpha)) * y_t(-1)^theta) - 1) *
104     p_t * y_t * (-theta) * (1 - alpha) * p_t * y_t^theta *
105     l(-1)^(theta * (1 - alpha)) / (p_t(-1) *
106     l^(theta * (1 - alpha) + 1) * y_t(-1)^theta);
107
108 % firm's profit
109 p_t * l^(1 - alpha) - w * l - phi / 2 * (p_t / p_t(-1) - 1)^2 * p_t * y_t - pi;
110
111 % profit distribution
112 chi * pi_r + (1 - chi) * pi_h - pi;
113
114 % impose profit for ricardians
115 pi_r - pi;
116
117 % tradeable goods market-clearing condition
118 y_t - (1 - chi) * c_aht - chi * c_art - c_bt;
119
120 % oil market-clearing condition
121 y_o - (1 - chi) * c_aho - chi * c_aro - c_bo;
122
123 % labour market-clearing condition
124 chi * l_r + (1 - chi) * l_h - l;
125
126 % bond market clearing condition
127 chi * n_a + n_b;
128
129 % oil endowment, stochastic process
130 gamma + rho * y_o(-1) - err_yo - y_o;
131
132 end;
133
134
135 %
136 % 4. Steady state
137 %
138
139
140 initval;
141
142 l_r = .22;
143 l_h = .22;
144 c_aro = .46;
145 c_aho = .46;
146 l = chi * l_r + (1 - chi) * l_h;
147 y_t = l^(1 - alpha); % from production function

```

```

148 p_t = 16.75 / y_t; % from target GDP in country a
149 c_art = s_t^eta * c_aro / (p_t^eta * (1 - s_t)^eta); % from tradeable-oil tradeoff
150 c_aht = s_t^eta * c_aho / (p_t^eta * (1 - s_t)^eta);
151 c_bt = y_t - chi * c_art - (1 - chi) * c_aht; % from tradeable MC
152 y_o = gamma / (1 - rho); % from oil stochastic process
153 c_bo = y_o - chi * c_aro - (1 - chi) * c_aho; % from oil MC
154 w = (1 - theta) * (1 - alpha) * p_t *
155       l^((1 - theta) * (1 - alpha) - 1) * y_t^theta; % from firm's FOC at SS
156 pi = p_t * l^(1 - alpha) - w * l; % from firm's profit at SS
157 pi_r = pi;
158 pi_h = (pi - chi * pi_r) / (1 - chi);
159 r_star = 1 / beta; % from SS Euler
160 n_a = r_star / (r_star - 1) * (p_t * c_art + c_aro - w * l - pi_r); % from the BC
161 n_b = - n_a; % from bond MC
162
163 end;
164
165 steady(maxit = 1000, solve_algo = 3);
166
167
168
169
170 %-----
171 % 5. Impulse response function
172 %-----
173
174 shocks;
175
176 var err_yo;
177 stderr .1;
178
179 end;
180
181
182 stoch_simul(order=1, irf=100, irf_plot_threshold=0)
183           c_aro c_bo c_bo c_aht c_art c_bt l_h
184           l_r l y_t y_o p_t w pi n_a n_b r_star;
185
186 model_diagnostics;

```

## Appendix G: TANBNR PIT mod file

```
1 %
2 %
3 %
4 % Title: International Macro–Finance Final Assignment, model file
5 % Author: —
6 % Date: 25/12/2023
7 % Description: Two agent, nominal bond, nominal rigidities,
8 %               price inflation targeting
9 %
10 %
11 %
12 %
13 %
14 % 0. Housekeeping (close all graphic windows)
15 %
16
17 close all;
18
19 %
20 % 1. Defining variables
21 %
22
23 var c_aho c_aro c_bo c_ahc c_art c_bt l_h l_r l_yt y_o p_t
24     e w pi n_a n_b r_star pi_r pi_h;
25
26 varexo err_yo;
27
28 %
29 % 2. Calibration
30 %
31 %
32
33 parameters alpha beta psi phi eta s_t theta rho gamma chi;
34
35 load parameters;
36
37 set_param_value('alpha', par.alpha);
38 set_param_value('beta', par.beta);
39 set_param_value('psi', par.psi);
40 set_param_value('phi', par.phi);
41 set_param_value('eta', par.eta);
42 set_param_value('s_t', par.s_t);
43 set_param_value('theta', par.theta);
44 set_param_value('rho', par.rho);
45 set_param_value('gamma', par.gamma);
46 set_param_value('chi', par.chi);
47
48
49 s_t = .8;
50 theta = .562;
51
52 %
53 % 3. Model
54 %
55 %
56
57 model;
58
59 % consumption allocation between oil and tradeable for htm HHs in country a
60 (1 - s_t) * c_aho^(-1 / eta) / e - s_t * c_ahc^(-1 / eta) / p_t;
61
62 % consumption allocation between oil and tradeable for ricardian HHs in country a
63 (1 - s_t) * c_aro^(-1 / eta) / e - s_t * c_art^(-1 / eta) / p_t;
64
65 % consumption allocation between oil and tradeable in country b
66 % here, p_tstar = p_t / e by LOOP
67 (1 - s_t) * c_bo^(-1 / eta) - s_t * c_bt^(-1 / eta) / (p_t / e);
68
69 % Euler equation for ricardian HHs in country a
70 c_aro^(-1 / eta) / ((s_t * c_art^(1 - 1 / eta) +
71     (1 - s_t) * c_aro^(1 - 1 / eta)) * e) -
72     beta * r_star *
```

```

74         c_aro(+1)^(-1 / eta) / ((s_t * c_art(+1)^(1 - 1 / eta) +
75             (1 - s_t) * c_aro(+1)^(1 - 1 / eta)) * e(+1));
76
77 % Euler equation in country b
78 c_bo^(-1 / eta) / (s_t * c_bt^(1 - 1 / eta) +
79     (1 - s_t) * c_bo^(1 - 1 / eta)) -
80     beta * r_star *
81         c_bo(+1)^(-1 / eta) / (s_t * c_bt(+1)^(1 - 1 / eta) +
82             (1 - s_t) * c_bo(+1)^(1 - 1 / eta));
83
84 % labour supply decision for htm HHs in country a
85 psi - (1 - s_t) * c_aho^(-1 / eta) / ((s_t * c_aht^(1 - 1 / eta) +
86     (1 - s_t) * c_aho^(1 - 1 / eta)) * e) *
87     w * (1 - l_h);
88
89 % labour supply decision for ricardian HHs in country a
90 psi - (1 - s_t) * c_aro^(-1 / eta) / ((s_t * c_art^(1 - 1 / eta) +
91     (1 - s_t) * c_aro^(1 - 1 / eta)) * e) *
92     w * (1 - l_r);
93
94 % htm's budget constraint in country a
95 p_t * c_aht + e * c_aho - w * l_h - pi_h;
96
97 % ricardian's budget constraint in country a
98 p_t * c_art + e * c_aro - w * l_r - pi_r - e * (n_a(-1) - n_a / r_star);
99
100 % HH's budget constraint in country b
101 c_bo + (p_t / e) * c_bt - y_o - (n_b(-1) - n_b / r_star);
102
103 % firm's FOC
104 p_t * (1 - theta) * (1 - alpha) * l^((1 - theta) * (1 - alpha) - 1) * y_t^theta -
105     w - phi * (p_t * y_t^theta * l(-1)^(theta * (1 - alpha)) /
106         (p_t(-1) * l^(theta * (1 - alpha)) * y_t(-1)^theta - 1) *
107             p_t * y_t * (-theta) * (1 - alpha) * p_t * y_t^theta *
108                 l(-1)^(theta * (1 - alpha)) / (p_t(-1) *
109                     l^(theta * (1 - alpha) + 1) * y_t(-1)^theta);
110
111 % firm's profit
112 p_t * l^(1 - alpha) - w * l - pi;
113
114 % profit distribution
115 chi * pi_r + (1 - chi) * pi_h - pi;
116
117 % impose profit for ricardians
118 pi_r - pi;
119
120 % tradeable goods market-clearing condition
121 y_t - (1 - chi) * c_aht - chi * c_art - c_bt;
122
123 % oil market-clearing condition
124 y_o - (1 - chi) * c_aho - chi * c_aro - c_bo;
125
126 % labour market-clearing condition
127 chi * l_r + (1 - chi) * l_h - l;
128
129 % bond market clearing condition
130 chi * n_a + n_b;
131
132 % oil endowment, stochastic process
133 gamma + rho * y_o(-1) - err_yo - y_o;
134
135 % monetary policy rule
136 p_t - p_t(-1);
137
138 end;
139
140
141 %
142 % 4. Steady state
143 %
144
145
146 initval;
147

```

```

148 l_r = .22;
149 l_h = .22;
150 c_aro = .46;
151 c_aho = .46;
152 e = 1;
153 l = chi * l_r + (1 - chi) * l_h;
154 y_t = l^(1 - alpha); % from production function
155 p_t = 16.75 / y_t; % from target GDP in country a
156 c_art = s_t^eta * e^eta * c_aro / (p_t^eta * (1 - s_t)^eta); % from tradeable-oil tradeoff
157 c_aht = s_t^eta * e^eta * c_aho / (p_t^eta * (1 - s_t)^eta);
158 c_bt = y_t - chi * c_art - (1 - chi) * c_aht; % from tradeable MC
159 y_o = gamma / (1 - rho); % from oil stochastic process
160 c_bo = y_o - chi * c_aro - (1 - chi) * c_aho; % from oil MC
161 w = (1 - theta) * (1 - alpha) * p_t *
162       l^((1 - theta) * (1 - alpha) - 1) * y_t^theta; % from firm's FOC at SS
163 pi = p_t * l^(1 - alpha) - w * l; % from firm's profit at SS
164 pi_r = pi;
165 pi_h = (pi - chi * pi_r) / (1 - chi);
166 r_star = 1 / beta; % from SS Euler
167 n_a = r_star / ((r_star - 1) * e) * (p_t * c_art + e * c_aro - w * l - pi_r); % from the BC
168 n_b = - n_a; % from bond MC
169
170
171 end;
172
173 steady(maxit = 1000);
174
175
176
177
178 %-----
179 % 5. Impulse response function
180 %-----
181
182 shocks;
183
184 var err_yo;
185 stderr .1;
186
187 end;
188
189
190 stoch_simul(order=1, irf=100, solve_algo = 3, irf_plot_threshold=0)
191           c_aho c_aro c_bo c_aht c_art c_bt l_h
192           l_r l_yt y_o p_t e w pi n_a n_b r_star;
193
194
195 model_diagnostics;

```

## Appendix H: TANBFPNH mod file

```
1
2 %
3 %
4 % Title: International Macro–Finance Final Assignment, model file
5 % Author: —
6 % Date: 25/12/2023
7 % Description: Two agent, nominal bond, flexible prices,
8 %               non–homothetic preferences
9 %
10 %
11
12
13 %
14 % 0. Housekeeping (close all graphic windows)
15 %
16
17 close all;
18
19 %
20 % 1. Defining variables
21 %
22
23 var c_aho c_aro c_bo c_aht c_art c_bt l_h l_r l_yt y_o p_t w pi
24     n_a n_b r_star pi_r pi_h;
25
26 varexo err_yo;
27
28
29 %
30 % 2. Calibration
31 %
32
33 parameters alpha beta psi eta s_art s_aht s_bt theta rho gamma chi;
34 load parameters;
35
36 set_param_value('alpha', par.alpha);
37 set_param_value('beta', par.beta);
38 set_param_value('psi', par.psi);
39 set_param_value('eta', par.eta);
40 set_param_value('theta', par.theta);
41 set_param_value('rho', par.rho);
42 set_param_value('gamma', par.gamma);
43 set_param_value('chi', par.chi);
44
45 chi = .7;
46 s_t = .8;
47 s_aht = .75; % home bias towards tradeable for htm
48 s_art = (s_t - (1 - chi) * s_aht) / chi; % home bias towards tradeable for ricardians
49 s_bt = s_t; % home bias towards tradeable in country b
50 theta = .68;
51
52
53 %
54 % 3. Model
55 %
56
57
58 model;
59
60 % consumption allocation between oil and tradeable for htm HHs in country a
61 (1 - s_aht) * c_aho^(-1 / eta) - s_aht * c_aht^(-1 / eta) / p_t;
62
63 % consumption allocation between oil and tradeable for ricardian HHs in country a
64 (1 - s_art) * c_aro^(-1 / eta) - s_art * c_art^(-1 / eta) / p_t;
65
66 % consumption allocation between oil and tradeable in country b
67 (1 - s_bt) * c_bo^(-1 / eta) - s_bt * c_bt^(-1 / eta) / p_t;
68
69 % Euler equation for ricardian HHs in country a
70 c_aro^(-1 / eta) / (s_art * c_art^(1 - 1 / eta) +
71     (1 - s_art) * c_aro^(1 - 1 / eta)) -
72     beta * r_star *
73     c_aro(+1)^(-1 / eta) / (s_art * c_art(+1)^(1 - 1 / eta) +
```

```

74         (1 - s_art) * c_aro(+1)^(1 - 1 / eta));
75
76 % Euler equation in country b
77 c_bo^(-1 / eta) / (s_bt * c_bt^(1 - 1 / eta) +
78     (1 - s_bt) * c_bo^(1 - 1 / eta)) -
79     beta * r_star *
80     c_bo(+1)^(-1 / eta) / (s_bt * c_bt(+1)^(1 - 1 / eta) +
81     (1 - s_bt) * c_bo(+1)^(1 - 1 / eta));
82
83 % labour supply decision for htm HHs in country a
84 psi - (1 - s_aht) * c_aho^(-1 / eta) / (s_aht * c_aht^(1 - 1 / eta) +
85     (1 - s_aht) * c_aho^(1 - 1 / eta)) *
86     w * (1 - l_h);
87
88 % labour supply decision for ricardian HHs in country a
89 psi - (1 - s_art) * c_aro^(-1 / eta) / (s_art * c_art^(1 - 1 / eta) +
90     (1 - s_art) * c_aro^(1 - 1 / eta)) *
91     w * (1 - l_r);
92
93 % htm's budget constraint in country a
94 p_t * c_aht + c_aho - w * l_h - pi_h;
95
96 % ricardian's budget constraint in country a
97 p_t * c_art + c_aro - w * l_r - pi_r - (n_a(-1) - n_a / r_star);
98
99 % HH's budget constraint in country b
100 c_bo + p_t * c_bt - y_o - (n_b(-1) - n_b / r_star);
101
102 % firm's FOC
103 (1 - theta) * (1 - alpha) * p_t *
104     l^((1 - theta) * (1 - alpha) - 1) * y_t^theta - w;
105
106 % firm's profit
107 p_t * l^(1 - alpha) - w * l - pi;
108
109 % profit distribution
110 chi * pi_r + (1 - chi) * pi_h - pi;
111
112 % impose profit for ricardians
113 pi_r - pi;
114
115 % tradeable goods market-clearing condition
116 y_t - (1 - chi) * c_aht - chi * c_art - c_bt;
117
118 % oil market-clearing condition
119 y_o - (1 - chi) * c_aho - chi * c_aro - c_bo;
120
121 % labour market-clearing condition
122 chi * l_r + (1 - chi) * l_h - l;
123
124 % bond market clearing condition
125 chi * n_a + n_b;
126
127 % oil endowment, stochastic process
128 gamma + rho * y_o(-1) - err_yo - y_o;
129
130 end;
131
132
133 %
134 % 4. Steady state
135 %
136
137
138 initval;
139
140 l_r = .22;
141 l_h = .22;
142 c_aro = .46;
143 c_aho = .46;
144 l = chi * l_r + (1 - chi) * l_h;
145 y_t = l^(1 - alpha); % from production function
146 p_t = 16.75 / y_t; % from target GDP in country a
147 c_art = s_art^eta * c_aro / (p_t^eta * (1 - s_art)^eta); % from tradeable-oil tradeoff

```

```

148 c_aht = s_aht^eta * c_aho / (p_t^eta * (1 - s_aht)^eta);
149 c_bt = y_t - chi * c_art - (1 - chi) * c_aht; % from tradeable MC
150 y_o = gamma / (1 - rho); % from oil stochastic process
151 c_bo = y_o - chi * c_aro - (1 - chi) * c_aho; % from oil MC
152 w = (1 - theta) * (1 - alpha) * p_t *
153       l^((1 - theta) * (1 - alpha) - 1) * y_t^theta; % from firm's FOC
154 pi = p_t * l^(1 - alpha) - w * l; % from firm's profit
155 pi_r = pi;
156 pi_h = (pi - chi * pi_r) / (1 - chi);
157 r_star = 1 / beta; % from SS Euler
158 n_a = r_star / (r_star - 1) * (p_t * c_art + c_aro - w * l - pi_r); % from the BC
159 n_b = - n_a; % from bond MC
160
161 end;
162
163 steady(maxit = 1000, solve_algo = 3);
164
165
166 %
167 %-----
168 % 5. Impulse response function
169 %-----
170
171
172 shocks;
173
174 var err_yo;
175 stderr .1;
176
177 end;
178
179
180 stoch_simul(order=1, irf=100, irf_plot_threshold=0)
181           c_aho c_aro c_bo c_aht c_art c_bt l_h
182           l_r l y_t y_o p_t w pi n_a n_b r_star;
183
184
185 model_diagnostics;

```