

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 τ is computed to reflect government tax revenue as a share of GDP in the US, while government spending is calibrated such that the resulting debt level reflects the average US debt-to-GDP ratio between 2012 and 2022 (approximately 120%). Finally, the upper bound on the asset grid was calibrated on the amount of net wealth held by the top .1% household in the US, while the lower bound was defined as an arbitrarily low number to ensure that the borrowing constraint binds for some households. The calibration parameters are presented in table (3).

Table 2: Calibration values

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

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

Table 3: Targeted values

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

Q4 to Q6

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

Table 4: Targeted values

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

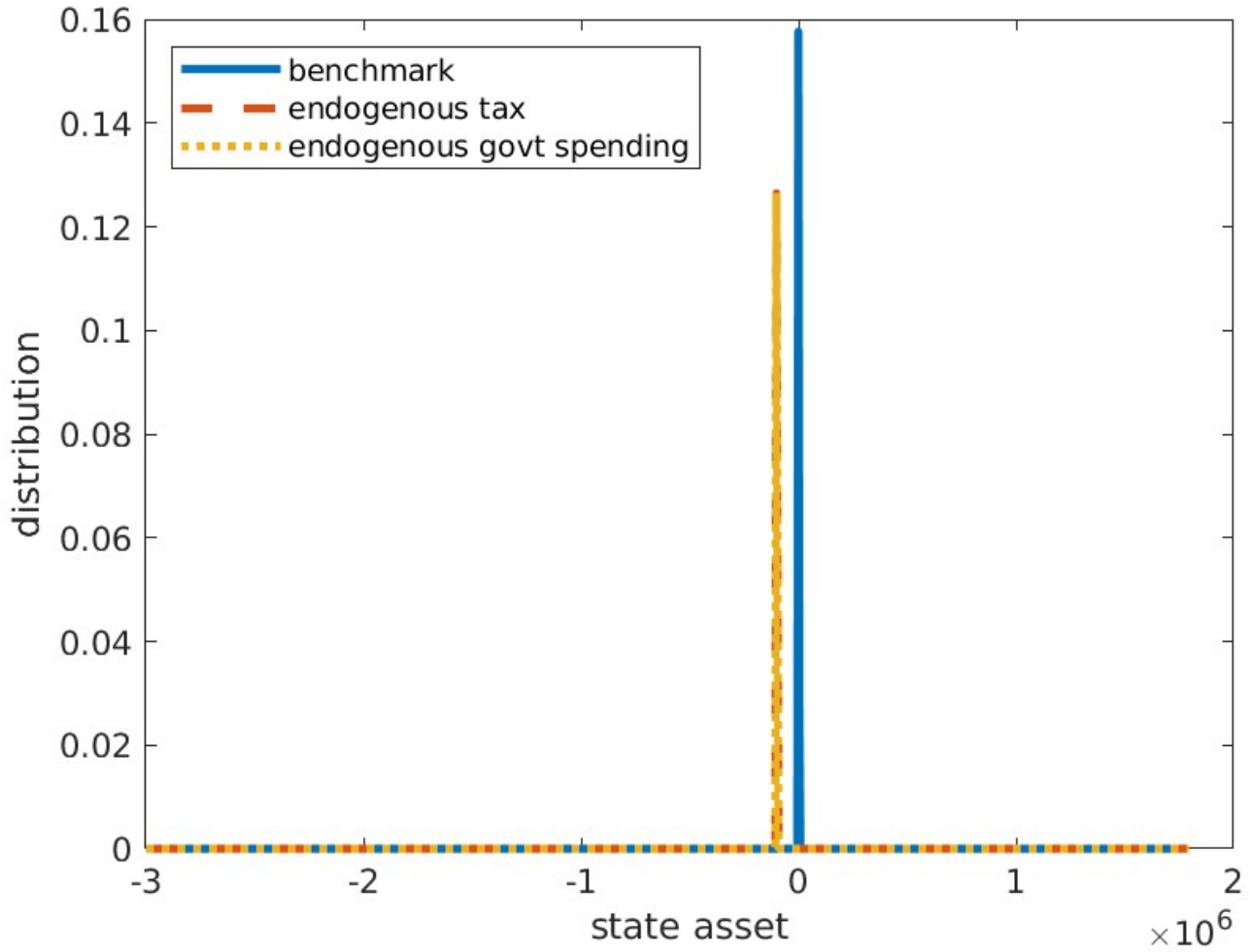


Figure 1: Utility comparative statics

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

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

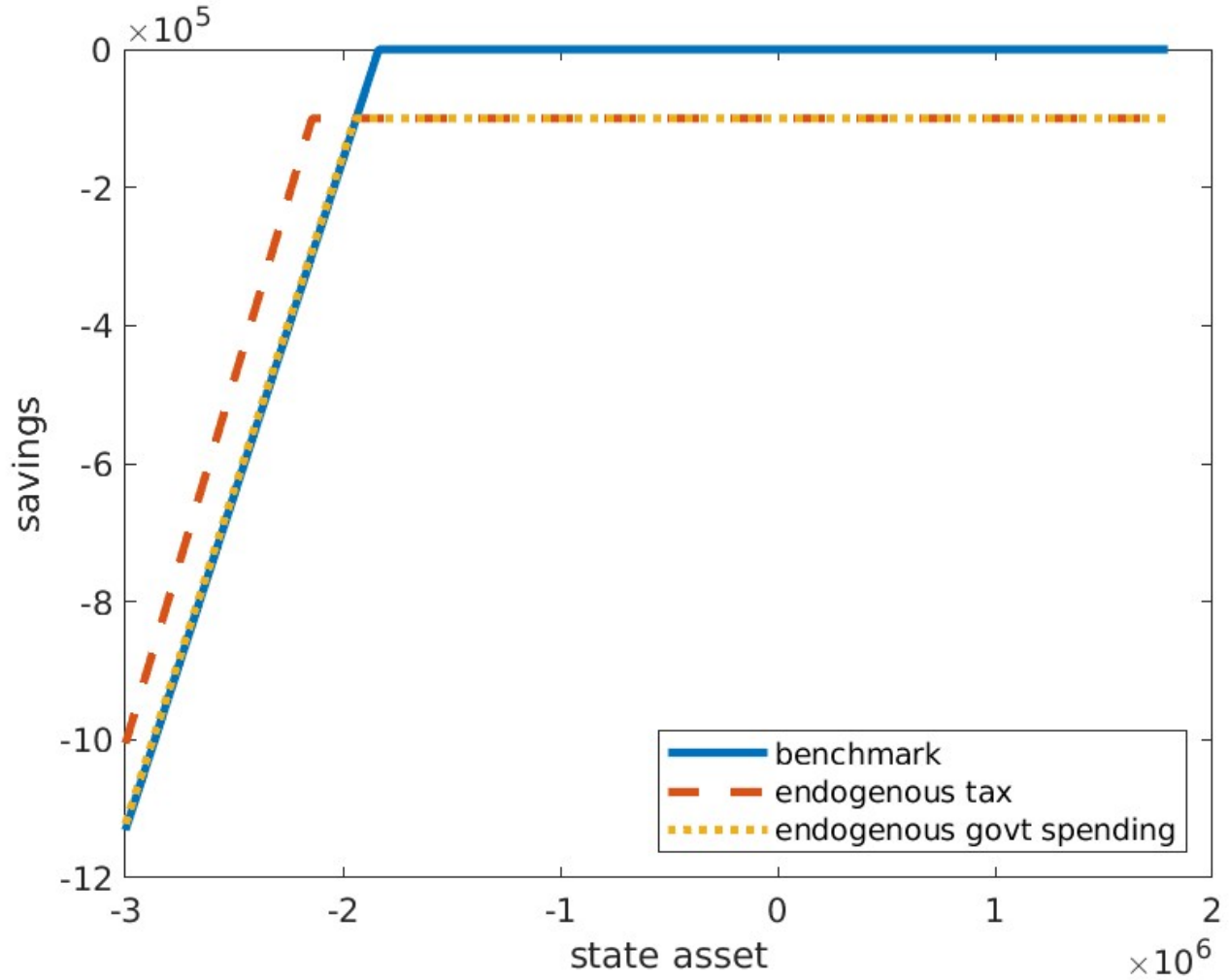


Figure 2: Savings policy function

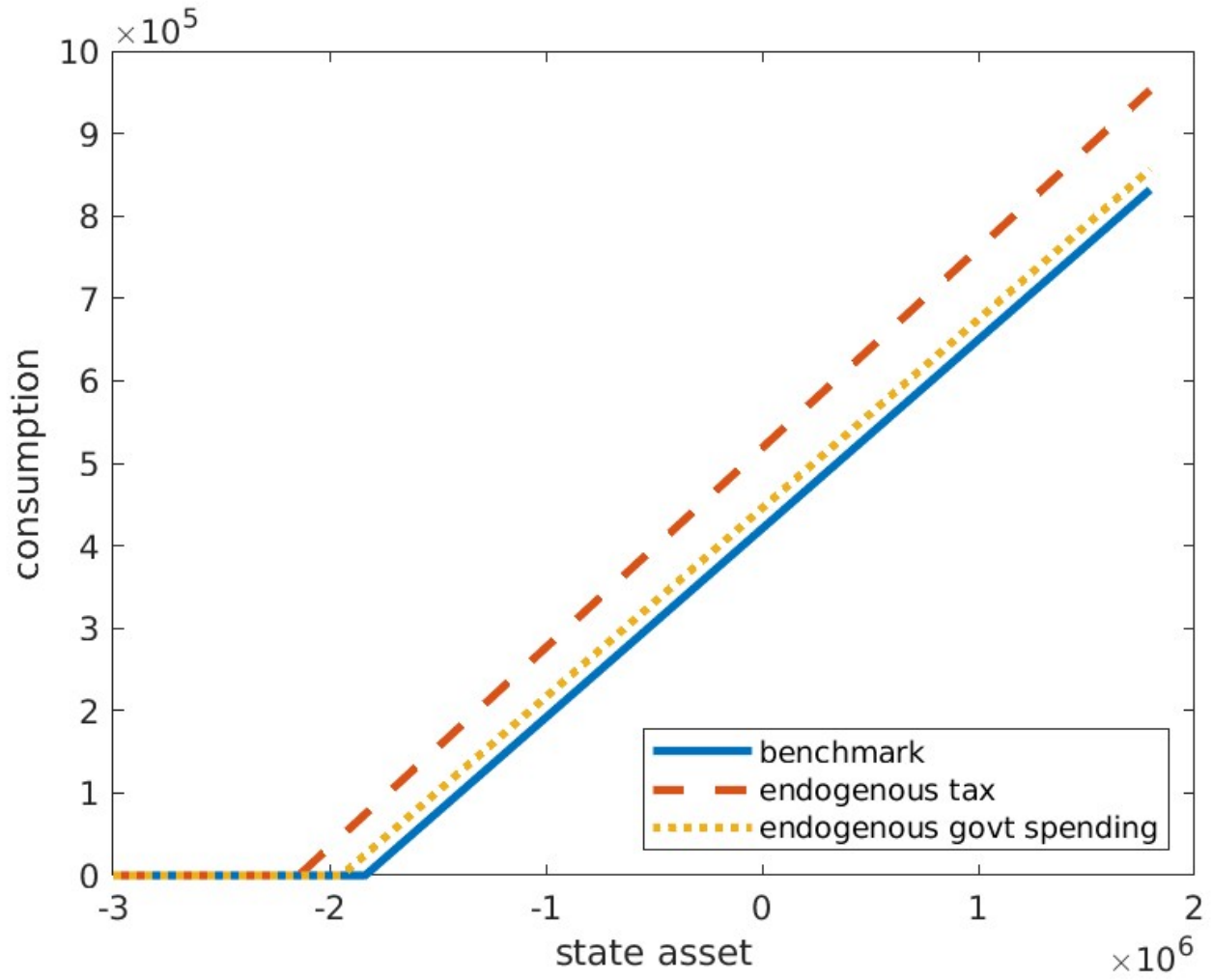


Figure 3: Consumption policy function

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

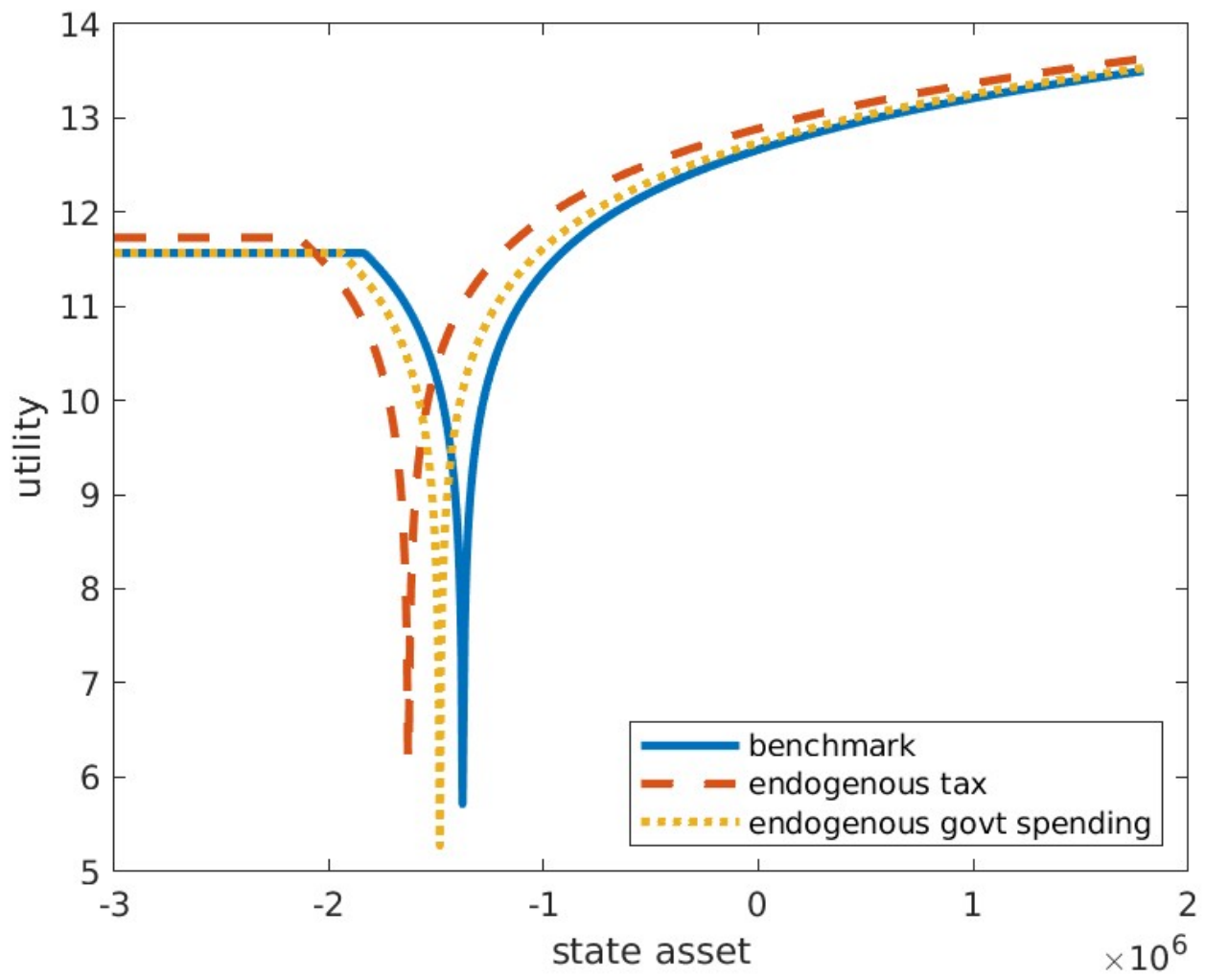


Figure 4: Utility comparative statics

Appendix A: Main File

```
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         Emupl = (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 * Emupl * 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-val(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);
```