Estimated Dynamic Optimization (EDO) Model

Gary Young (editor)

June 28, 2016

Contents

1 EDO model packages **5**

2 Documentation **7**

Appendices **9**

A Original Files **11**

A.1 Dynare_edo.mod . . . 11
A.1.1 model . . . 12
A.1.2 prelim . . . 13
A.1.3 labor . . . 14
A.1.4 identities . . . 14
A.1.5 hours . . . 15
A.1.6 durables . . . 15
A.1.7 housing . . . 16
A.1.8 measure . . . 17
A.2 Dynare_edo_steadystate.m . . . 18
A.2.1 result return . . . 19
A.3 linearized.mod . . . 21
A.3.1 model . . . 22
A.3.2 prelim . . . 23
A.3.3 labor . . . 24
A.3.4 identities . . . 25
A.3.5 hours . . . 25
A.3.6 durables . . . 25
A.3.7 housing . . . 26
A.3.8 measure . . . 27
A.4 linearized_steadystate.m . . . 28
A.4.1 result return . . . 29
A.5 common model . . . 32
A.5.1 setup . . . 32
A.5.2 stoch sim . . . 33
A.5.3 var . . . 34
A.5.4 varexo . . . 36

A.5.5 parameters . . . 37
A.5.6 est params . . . 41
A.5.7 calib params . . . 42
A.5.8 free params . . . 43
A.5.9 varobs . . . 43
A.5.10 shocks . . . 44
A.5.11 steady params . . . 45
A.5.12 stderr . . . 46
A.6 common steady state . . . 47
A.6.1 values . . . 47
A.6.2 trends . . . 50
A.7 readme.txt . . . 51

B Notes, Bibliography and Indexes **53**
B.1 Chunks . . . 53
B.2 Index . . . 54

Chapter 1

EDO model packages

The Estimated Dynamic Optimization (EDO) Model is available from the Federal Reserve Board of Governors website:

> The model package zip file (link below) contains the following files:
>
> - A readme file with basic instructions.
> - Dynare mod files to run two versions of the EDO model, one with variables in levels and the other with variables in log deviations from steady state. Both versions include the nonlinear equations derived from household and firm optimization problems. The version in levels more closely follows the derivations described in the article "Unemployment During the Great Recession in the EDO Model of the U.S. Economy", while the version in log deviations facilitates the reporting of simulation results. These programs solve the model, report some basic model statistics, and run some basic impulse response simulations.

Notice that the edo (zip) is actually contained in the "EDO variable listing (ZIP)" link and the mentioned "variable listing zip file" doesn't seem to be available anywhere:

> The variable listing zip file (link below) contains an HTML representation of the EDO model, showing linkages between variables, parameters, and equations.

EDO variable listing (ZIP)

NOTE: The programs for simulating the EDO model are written for use with the Dynare software package. The Dynare package can be downloaded without cost at www.dynare.org While Dynare itself is free, it requires the installation of either Matlab or Octave. Matlab is a commercial product available at www.mathworks.com. Octave is free-ware, and is available at www.gnu.org/software/octave

Dynare and octave were available in The Ubuntu Software Center for my computer. From the readme file, to execute the model:

5) Run the command "dynare linearized" or "dynare Dynare_edo" from the Matlab/Octave command line to run the two model versions

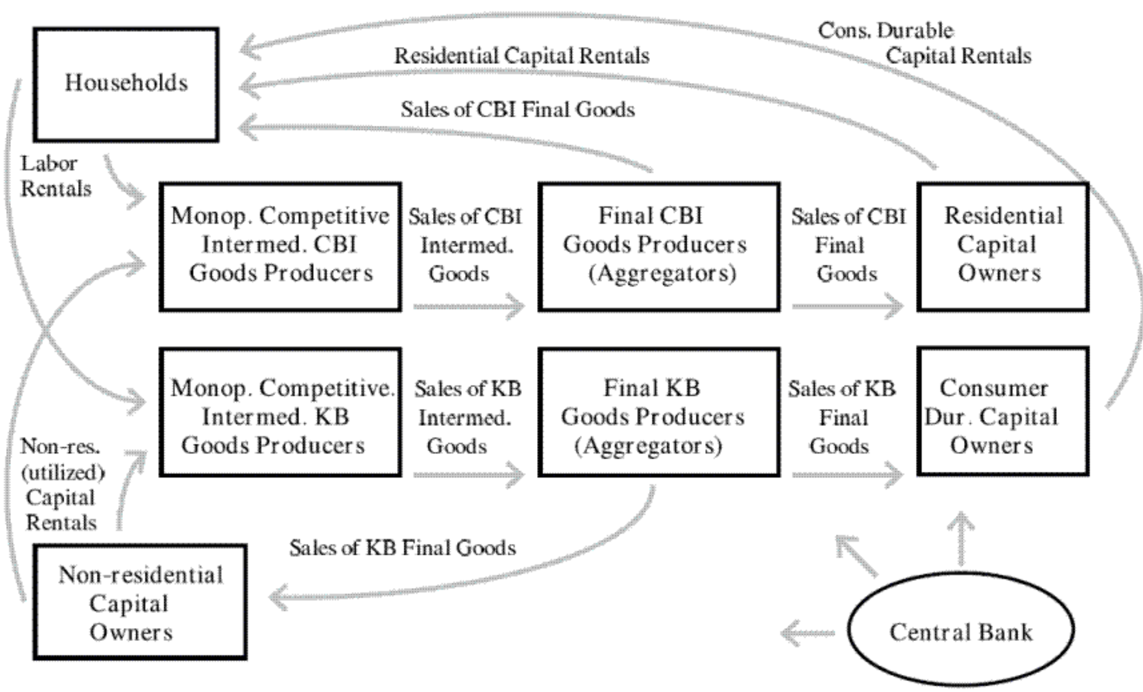
The model fails for me and when googling for the error "dynare linearized trans_A undefined" we get

At the moment, Octave 4 is not supported yet. See https://github.com/DynareTeam/dynare/issues/1113

at "Trans_A Matrix error" and "Make Dynare compatible with Octave 4.0 #1113".

Chapter 2

Documentation



$$\min_{\{X_t^s(j)\}_{j=0}^1} \int_0^1 P_t^s(j) X_t^s(j) dj \text{ subject to } \left(\int_0^1 \left(X_t^s(j)\right)^{\frac{\Theta_t^s - 1}{\Theta_t^s}} dj \right)^{\frac{\Theta_t^s}{\Theta_t^s - 1}} \geq X_t^s, \text{ for } s = cbi, kb. \tag{2.1}$$

Appendices

Appendix A

Original Files

A.1 Dynare_edo.mod

11 ⟨*srcedo/Dynare.edo.mod* 11⟩≡
   ⟨*common setup* 32⟩
   ⟨*edo model* 12⟩
   ⟨*common stoch sim* 33⟩

This code is written to file `srcedo/Dynare.edo.mod`.

A.1.1 Dynare EDO Model

12 ⟨*edo model* 12⟩≡ (11)

```
⟨edo model prelim 13⟩

// XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
// labor block
// TOTAL LABOR INPUT (called "L" in the paper, I kept the "H" notation of the origina
⟨edo model labor 14a⟩

// Identities
⟨edo model identities 14b⟩


// XXXXXXXXXXXXXXXXXXXXXX
// Aggregate hours equals agg hours in each sector
⟨edo model hours 15a⟩

// See Section 8: Data Identities

// new equations
// Durable Block

⟨edo model durables 15b⟩

// Housing Block
⟨edo model housing 16⟩

//measurement_equations;
⟨edo model measurement 17⟩
//end_measurement_equations;
end;
```

Uses L 34.

A.1.2 Dynare EDO Model Prelim

13 ⟨*edo model prelim* 13⟩≡ (12)

```
RC-MCC*YC/UC/KC(-1)*alpha_*MUK=0;
RK-MCK*YK/UK/KK(-1)*alpha_*MUK=0;
WC-MCC*YC/HC*(1-alpha_)=0;
WK-MCK*YK/HK*(1-alpha_)=0;
YC-(UC*KC(-1)/MUK)^alpha_*(HC)^(1-alpha_)=0;
YK-(UK*KK(-1)/MUK)^alpha_*(HK)^(1-alpha_)=0;
MCC*YC*theta_c-(theta_c-1)*YC-100*phi_pc*(INFC-gam_pc*INFC(-1)-(1-gam_pc)*INFCSS)*INFC*YC+beta_
MCK*YK*theta_k/PKB-(theta_k-1)*YK-100*phi_pc*(INFK-gam_pc*INFK(-1)-(1-gam_pc)*INFKSS)*INFK*YK+b
QK-beta_*(1/EFFK)*(((1-delta_)*QK(+1)+RC(+1)*UC(+1))*L(+1)/MUK(+1)/L)=0;
QK-beta_*(1/EFFK)*(((1-delta_)*QK(+1)+RK(+1)*UK(+1))*L(+1)/MUK(+1)/L)=0;
L-betas*R/rpr/INFC(+1)/MUC(+1)*L(+1)=0;
ln(R/RSS)-rho_R*ln(R(-1)/RSS)-(1-rho_R)*(r_inf*ln(INFCNA/INFCNASS)+r_dinf*(ln(INFCNA)-ln(INFCNA
L-eta_cnn/(EC-h*EC(-1)/MUC)+eta_cnn*beta_*h/(MUC(+1)*EC(+1)-h*EC)=0;
KK-(1-delta_)*KK(-1)/MUK+KC-(1-delta_)*KC(-1)/MUK-1*EIK+mu_*((UK^(1+1/phi_u)-1)/(1+1/phi_u))*KK
```

Uses alpha_ 37, beta_ 37, betas 34, delta_ 37, EC 34, EFFK 34, EIK 34, ePMKC 36, ePMKK 36, eR 36, eta_cnn 37, gam_pc 37, h 37, HC 34, HK 34, INFC 34, INFCNA 34, INFCNASS 37, INFCSS 37, INFK 34, INFKSS 37, KC 34, KCSS 37, KK 34, KKSS 37, L 34, MCC 34, MCK 34, mu_ 37, MUC 34, MUK 34, PFGAP 34, phi_pc 37, phi_u 37, PKB 34, QK 34, R 34, r_dinf 37, r_inf 37, r_y 37, RC 34, rho_R 37, RK 34, rpr 37, RSS 37, theta_c 37, theta_k 37, UC 34, UK 34, WC 34, WK 34, YC 34, YCSS 37, YK 34, and YKSS 37.

A.1.3 Dynare EDO Model Labor

14a ⟨*edo model labor* 14a⟩≡ (12)

```
-100+UHC*theta_wc-(theta_wc-1)*WC-100*phi_wc*(INFWC-gam_wc*INFWC(-1)-(1-gam_wc)*INFW
UHSC-WC+phi_H/10*(HSC/HSK-gam_h*HSC(-1)/HSK(-1)-(1-gam_h)*HSCSS/HSKSS);//+100*eXiL=0
-100+UHK*theta_wk-(theta_wk-1)*WK-100*phi_wc*(INFWK-gam_wc*INFWK(-1)-(1-gam_wc)*INFW
UHSK-WK-phi_H/10*(HSC/HSK-gam_h*HSC(-1)/HSK(-1)-(1-gam_h)*HSCSS/HSKSS);//+100*eXiL=0
UHC*L*Lpref-A_HC*((1+sigman)/(1+sigman/(1+sigmah)))*(HC)^(-1+(1+sigman)/(1+sigman/(1
UHSC*L*Lpref-A_HC*((1+sigman)/(1+sigman/(1+sigmah)))*(HSC)^(-1+(1+sigman)/(1+sigman/
UHK*L*Lpref-A_HK*((1+sigman)/(1+sigman/(1+sigmah)))*(HK)^(-1+(1+sigman)/(1+sigman/(1
UHSK*L*Lpref-A_HK*((1+sigman)/(1+sigman/(1+sigmah)))*(HSK)^(-1+(1+sigman)/(1+sigman/
empC-((1+sigmah)/sigmah*xsi_NC/xsi_HrC)^(-1/(1+sigmah+sigman))*HC^(1/(1+sigman/(1+si
HrC-((1+sigmah)/sigmah*xsi_NC/xsi_HrC)^(1/(1+sigmah))*empC^(sigman/(1+sigmah))=0;
empK-((1+sigmah)/sigmah*xsi_NK/xsi_HrK)^(-1/(1+sigmah+sigman))*HK^(1/(1+sigman/(1+si
HrK-((1+sigmah)/sigmah*xsi_NK/xsi_HrK)^(1/(1+sigmah))*empK^(sigman/(1+sigmah))=0;
empSC-((1+sigmah)/sigmah*xsi_NC/xsi_HrC)^(-1/(1+sigmah+sigman))*HSC^(1/(1+sigman/(1+
HrSC-((1+sigmah)/sigmah*xsi_NC/xsi_HrC)^(1/(1+sigmah))*empSC^(sigman/(1+sigmah))=0;
empSK-((1+sigmah)/sigmah*xsi_NK/xsi_HrK)^(-1/(1+sigmah+sigman))*HSK^(1/(1+sigman/(1+
HrSK-((1+sigmah)/sigmah*xsi_NK/xsi_HrK)^(1/(1+sigmah))*empSK^(sigman/(1+sigmah))=0;
unemp-(empSC+empSK-(empC+empK))/(empSC+empSK)=0;
PKB-(1-100*phi_ic*(EIK-gam_ic*EIK(-1)-(1-gam_ic)*EIKSS)/(KC(-1)+KK(-1))*MUK)*QK-beta_
YC-EC-ECH-0.2*YCSS*HG=0;
ln(INFWC)-ln(WC)+ln(WC(-1))-ln(MUC)-ln(INFC)=0;
ln(INFWK)-ln(WK)+ln(WK(-1))-ln(MUC)-ln(INFC)=0;
ln(INFK)-ln(INFC)-ln(PKB)+ln(PKB(-1))+ln(MUK)-ln(MUC)=0;
YK-EIK-ECD-0.2*YKSS*HG=0;
ln(DIFFNORMGDP)-(1-s_k)*(ln(YC)-ln(YC(-1)))-s_k*(ln(YK)-ln(YK(-1)))=0;
ln(NORMINFGDP)-s_k*(ln(PKB)-ln(PKB(-1)))=0;
ln(DIFFREALGDP)-ln(DIFFNORMGDP)-(1-s_k)*ln(MUC)-s_k*ln(MUK)=0;
ln(DIFFREALEC)-ln(EC)+ln(EC(-1))-ln(MUC)=0;
ln(DIFFREALEIK)-ln(EIK)+ln(EIK(-1))-ln(MUK)=0;
```

Uses A_HC 37, A_HK 37, beta_ 37, DIFFNORMGDP 34, DIFFREALEC 34, DIFFREALEIK 34, DIFFREALGDP 34, EC 34, ECD 34, ECH 34, EFFK 34, EIK 34, EIKSS 37, empC 34, empK 34, empSC 34, empSK 34, eXiL 36, gam_h 37, gam_ic 37, gam_wc 37, HC 34, HCSS 37, HG 34, HK 34, HKSS 37, HrC 34, HrK 34, HrSC 34, HrSK 34, HSC 34, HSCSS 37, HSK 34, HSKSS 37, INFC 34, INFK 34, INFWC 34, INFWCSS 37, INFWK 34, INFWKSS 37, KC 34, KK 34, L 34, Lpref 34, MUC 34, MUK 34, NORMINFGDP 34, phi_H 37, phi_ic 37, phi_wc 37, PKB 34, QK 34, s_k 37, sigmah 37, sigman 37, theta_wc 37, theta_wk 37, UHC 34, UHK 34, UHSC 34, UHSK 34, unemp 34, WC 34, WK 34, XiL 34, xsi_HrC 37, xsi_HrK 37, xsi_NC 37, xsi_NK 37, YC 34, YCSS 37, YK 34, and YKSS 37.

A.1.4 Dynare EDO Model Identities

14b ⟨*edo model identities* 14b⟩≡ (12)

```
ln(DIFFREALW)-HCSS/AHSS*(ln(INFWC))-HKSS/AHSS*(ln(INFWK))+ln(INFC)=0;
```

Uses AHSS 37, DIFFREALW 34, HCSS 37, HKSS 37, INFC 34, INFWC 34, and INFWK 34.

A.1.5 Dynare EDO Model Hours

15a ⟨*edo model hours* 15a⟩≡ (12)

```
AH-HC-HK=0;
ln(INFGDP)-ln(INFC)-ln(YC*MUC/YC(-1))+ln(DIFFREALGDP)-ln((1+PKB*YK/YC)/(1+PKB(-1)*YK(-1)/YC(-1)
ln(INFCNA)-(1-s_ecdc)*ln(INFC)-s_ecdc*ln(INFK)=0;
ln(INFCOR)-(1-s_ecdc)*ln(INFC)-s_ecdc*ln(INFK)=0;
ln(GAP)-(1-s_k)*ln(YC/YCSS)-s_k*ln(YK/YKSS)=0;
ln(PFGAP)-(1-alpha_)*((1-s_k)*ln(HC/HCSS)+s_k*ln(HK/HKSS))-alpha_*((1-s_k)*ln(UC/USS)+s_k*ln(UK
ln(INFC10)-betarl*ln(INFC10(+1))-(1-betarl)*ln(INFCOR)=0;
```

Uses AH 34, alpha_ 37, betarl 37, DIFFREALGDP 34, GAP 34, HC 34, HCSS 37, HK 34, HKSS 37, INFC 34, INFC10 34, INFCNA 34, INFCOR 34, INFGDP 34, INFK 34, MUC 34, PFGAP 34, PKB 34, s_ecdc 37, s_k 37, UC 34, UK 34, USS 37, YC 34, YCSS 37, YK 34, and YKSS 37.

A.1.6 Dynare EDO Model Durables

15b ⟨*edo model durables* 15b⟩≡ (12)

```
KD-(1-delta_cd)*KD(-1)/MUK-ECD=0;
L*RCD-eta_cd/(KD(-1)/MUK-h_cd*LAGKD(-1)/(MUK(-1)*MUK))+beta_*eta_cd*h_cd/(KD-h_cd*KD(-1)/MUK)=0
QCD-beta_*(1/EFFECD)*L(+1)/L/MUK(+1)*(RCD(+1)+(1-delta_cd)*QCD(+1))=0;
PKB-QCD*(1-100*phi_cd*(ECD-gam_icd*ECD(-1)-(1-gam_icd)*ECDSS)/KD(-1)*MUK) - beta_*(1/EFFECD)*10
```

Uses beta_ 37, delta_cd 37, ECD 34, ECDSS 37, EFFECD 34, eta_cd 37, gam_icd 37, h_cd 37, KD 34, L 34, LAGKD 34, MUK 34, phi_cd 37, PKB 34, QCD 34, and RCD 34.

A.1.7 Dynare EDO Model Housing

16 ⟨*edo model housing* 16⟩≡ (12)

```
L*RCH-eta_ch/(KCH(-1)/MUC-h_ch*LAGKCH(-1)/(MUC*MUC(-1)))+beta_*eta_ch*h_ch/(KCH-h_ch*
QCH-beta_*(1/EFFECH)*L(+1)/L/MUC(+1)*(RCH(+1)+(1-delta_ch)*QCH(+1))=0;
1*ECH+(1-delta_ch)*KCH(-1)/MUC-KCH=0;
1-QCH*(1-100*phi_ech*(ECH-gam_ech*ECH(-1)-(1-gam_ech)*ECHSS)/KCH(-1)*MUC) - beta_*(1/
ln(KD(-1))-ln(LAGKD)=0;
ln(KCH(-1))-ln(LAGKCH)=0;
RK-QK*mu_*UK^(1/phi_u)=0;
RC-QK*mu_*UC^(1/phi_u)=0;
ln(DIFFREALECH)-ln(MUC)-ln(ECH)+ln(ECH(-1))=0;
ln(DIFFREALECD)-ln(MUK)-ln(ECD)+ln(ECD(-1))=0;
ln(betas/beta_)-rho_B*ln(betas(-1)/beta_)-eB=0;
ln(XiL)-rho_XiL*ln(XiL(-1))-eXiL=0;
ln(Lpref)-rho_lpref*ln(Lpref(-1))-eLpref=0;
ln(EFFK)-rho_EFFK*ln(EFFK(-1))-eEFFK=0;
ln(MUZK/MUZKSS)-eMUZK=0;
ln(MUZM/MUZMSS)-eMUZM=0;
ln(HG)-rho_HG*ln(HG(-1))-eHG=0;
ln(MUC)-ln(MUZM)-alpha_*ln(MUZK)=0;
ln(MUK)-ln(MUZM)-ln(MUZK)=0;
ln(EFFECD)-rho_EFFECD*ln(EFFECD(-1))-eEFFECD=0;
ln(EFFECH)-rho_EFFECH*ln(EFFECH(-1))-eEFFECH=0;
ln(STAR)-rho_STAR*ln(STAR(-1))-eSTAR=0;
ln(RL1) - ln(R(+1))=0;
ln(RL2) - ln(RL1(+1))=0;
ln(RL3) - ln(RL2(+1))=0;
ln(RL4) - ln(RL3(+1))=0;
ln(RL5) - ln(RL4(+1))=0;
ln(RL6) - ln(RL5(+1))=0;
ln(RL7) - ln(RL6(+1))=0;
ln(RT2) - tp2 - 0.125*(ln(R) + ln(RL1) + ln(RL2) + ln(RL3) + ln(RL4) + ln(RL5) + ln(R
```

Uses alpha_ 37, beta_ 37, betas 34, delta_ch 37, DIFFREALECD 34, DIFFREALECH 34, eB 36, ECD 34, ECH 34, ECHSS 37, eEFFECD 36, eEFFECH 36, eEFFK 36, EFFECD 34, EFFECH 34, EFFK 34, eHG 36, eLpref 36, eMUZK 36, eMUZM 36, eSTAR 36, eta_ch 37, eXiL 36, gam_ech 37, h_ch 37, HG 34, KCH 34, KD 34, L 34, LAGKCH 34, LAGKD 34, Lpref 34, mu_ 37, MUC 34, MUK 34, MUZK 34, MUZKSS 37, MUZM 34, MUZMSS 37, phi_ech 37, phi_u 37, QCH 34, QK 34, R 34, RC 34, RCH 34, rho_B 37, rho_EFFECD 37, rho_EFFECH 37, rho_EFFK 37, rho_HG 37, rho_lpref 37, rho_STAR 37, rho_XiL 37, RK 34, RL1 34, RL2 34, RL3 34, RL4 34, RL5 34, RL6 34, RL7 34, RT2 34, STAR 34, tp2 37, UC 34, UK 34, and XiL 34.

A.1.8 Dynare EDO Model Measurement

17 ⟨*edo model measurement* 17⟩≡ (12)

```
ln(DIFFREALGDP_obs/DIFFREALGDPSS_obs) = ln(DIFFREALGDP/DIFFREALGDPSS);
ln(DIFFREALEC_obs/DIFFREALECSS_obs)   = ln(DIFFREALEC/DIFFREALECSS);
ln(DIFFREALEIK_obs/DIFFREALEIKSS_obs) = ln(DIFFREALEIK/DIFFREALEIKSS);
ln(DIFFREALECD_obs/DIFFREALECDSS_obs) = ln(DIFFREALECD/DIFFREALECDSS);
ln(DIFFREALECH_obs/DIFFREALECHSS_obs) = ln(DIFFREALECH/DIFFREALECHSS);
ln(DIFFREALW_obs/DIFFREALWSS_obs)     = ln(DIFFREALW/DIFFREALWSS);
ln(AH_obs)                            = ln(AH/AHSS);
ln(INFCNA_obs/INFCNASS_obs)           = ln(INFCNA/INFCNASS);
ln(INFCOR_obs/INFCORSS_obs)           = ln(INFCOR/INFCORSS);
ln(INFK_obs/INFKSS_obs)               = ln(INFK/INFKSS);
ln(R_obs/RSS_obs)                     = ln(R/RSS);
ln(RT2_obs/RT2SS_obs)                 = ln(RT2/RT2SS);
ln(unemp_obs/unempSS_obs)             = ln(unemp/unempSS);
```

Uses AH 34, AH_obs 34, AHSS 37, DIFFREALEC 34, DIFFREALEC_obs 34, DIFFREALECD 34, DIFFREALECD_obs 34, DIFFREALECDSS 37, DIFFREALECDSS_obs 37, DIFFREALECH 34, DIFFREALECH_obs 34, DIFFREALECHSS 37, DIFFREALECHSS_obs 37, DIFFREALECSS 37, DIFFREALECSS_obs 37, DIFFREALEIK 34, DIFFREALEIK_obs 34, DIFFREALEIKSS 37, DIFFREALEIKSS_obs 37, DIFFREALGDP 34, DIFFREALGDP_obs 34, DIFFREALGDPSS 37, DIFFREALGDPSS_obs 37, DIFFREALW 34, DIFFREALW_obs 34, DIFFREALWSS 37, DIFFREALWSS_obs 37, INFCNA 34, INFCNA_obs 34, INFCNASS 37, INFCNASS_obs 37, INFCOR 34, INFCOR_obs 34, INFCORSS 37, INFCORSS_obs 37, INFK 34, INFK_obs 34, INFKSS 37, INFKSS_obs 37, R 34, R_obs 34, RSS 37, RSS_obs 37, RT2 34, RT2_obs 34, RT2SS 37, RT2SS_obs 37, unemp 34, unemp_obs 34, unempSS 37, and unempSS_obs 37.

A.2 Dynare_edo_steadystate.m

18 ⟨*srcedo/Dynare.edo.steadystate.m* 18⟩≡

```
function [ys,check] = unlinearized_edo_steadystate(ys,exe)
        global M_

check = 0;

NumberofParameters=M_.param_nbr;
for i=1:NumberofParameters
    paramname=deblank(M_.param_names(i,:));
    eval([paramname ' =M_.params(' int2str(i) ');']);
end;

%start_steady_state;

⟨common steady state values 47⟩

%end_steady_state;

%trends;

⟨common steady state trends 50⟩

%end_trends;


for i=1:NumberofParameters
    paramname=deblank(M_.param_names(i,:));
    eval(['M_.params(' int2str(i) ')=' paramname ';']);
end;


⟨edo steady state result return 19⟩
```

This code is written to file `srcedo/Dynare.edo.steadystate.m`.
Defines:
`unlinearized_edo_steadystate`, never used.

A.2.1 EDO Steady State Result Return

19 ⟨*edo steady state result return* 19⟩≡ (18)

```
ys = [
RCSS
RKSS
WCSS
WKSS
YCSS
YKSS
MCCSS
MCKSS
KCSS
KKSS
PKBSS
RSS
LSS
QKSS
HCSS
HSCSS
HKSS
HSKSS
UHCSS
UHSCSS
UHKSS
UHSKSS
empCSS
HrCSS
empKSS
HrKSS
empSCSS
HrSCSS
empSKSS
HrSKSS
unempSS
EIKSS
ECSS
INFWCSS
INFWKSS
INFCSS
INFKSS
ONE
ONE
DIFFREALGDPSS
DIFFREALECSS
DIFFREALEIKSS
```

```
DIFFREALWSS
AHSS
INFGDPSS
INFCNASS
INFCORSS
ONE
ONE
INFC10SS
ECDSS
KCDSS
RCDSS
QCDSS
KCHSS
RCHSS
ECHSS
QCHSS
KCDSS
KCHSS
USS
USS
DIFFREALECHSS
DIFFREALECDSS
beta_
ONE
ONE
ONE
MUZKSS
MUZMSS
ONE
MUCSS
MUKSS
ONE
ONE
ONE
RL1SS
RL2SS
RL3SS
RL4SS
RL5SS
RL6SS
RL7SS
RT2SS
DIFFREALGDPSS_obs
DIFFREALECSS_obs
DIFFREALEIKSS_obs
DIFFREALECDSS_obs
```

```
  DIFFREALECHSS_obs
  DIFFREALWSS_obs
  ONE
  INFCNASS_obs
  INFCORSS_obs
  INFKSS_obs
  RSS_obs
  RT2SS_obs
  unempSS_obs
  ];
```

Uses AHSS 37, beta_ 37, DIFFREALECDSS 37, DIFFREALECDSS_obs 37, DIFFREALECHSS 37, DIFFREALECHSS_obs 37, DIFFREALECSS 37, DIFFREALECSS_obs 37, DIFFREALEIKSS 37, DIFFREALEIKSS_obs 37, DIFFREALGDPSS 37, DIFFREALGDPSS_obs 37, DIFFREALWSS 37, DIFFREALWSS_obs 37, ECDSS 37, ECHSS 37, ECSS 37, EIKSS 37, empCSS 37, empKSS 37, empSCSS 37, empSKSS 37, HCSS 37, HKSS 37, HrCSS 37, HrKSS 37, HrSCSS 37, HrSKSS 37, HSCSS 37, HSKSS 37, INFC10SS 37, INFCNASS 37, INFCNASS_obs 37, INFCORSS 37, INFCORSS_obs 37, INFCSS 37, INFGDPSS 37, INFKSS 37, INFKSS_obs 37, INFWCSS 37, INFWKSS 37, KCDSS 37, KCHSS 37, KCSS 37, KKSS 37, LSS 37, MCCSS 37, MCKSS 37, MUCSS 37, MUKSS 37, MUZKSS 37, MUZMSS 37, ONE 37, PKBSS 37, QCDSS 37, QCHSS 37, QKSS 37, RCDSS 37, RCHSS 37, RCSS 37, RKSS 37, RL1SS 37, RL2SS 37, RL3SS 37, RL4SS 37, RL5SS 37, RL6SS 37, RL7SS 37, RSS 37, RSS_obs 37, RT2SS 37, RT2SS_obs 37, UHCSS 37, UHKSS 37, UHSCSS 37, UHSKSS 37, unempSS 37, unempSS_obs 37, USS 37, WCSS 37, WKSS 37, YCSS 37, and YKSS 37.

A.3 linearized.mod

21 ⟨*srcedo/linearized.mod* 21⟩≡
  ⟨*common setup* 32⟩
  ⟨*linearized model* 22⟩
  ⟨*common stoch sim* 33⟩

This code is written to file `srcedo/linearized.mod`.

A.3.1 Linearized Model

22 ⟨*linearized model* 22⟩≡ (21)

```
⟨linearized model prelim 23⟩

// XXXXXXXXXXXXXXXXXXXXXXXXXXXX
// labor block
// TOTAL LABOR INPUT (called "(LSS*exp(L))" in the paper, I kept the "H" notation of
⟨linearized model labor 24⟩

// Identities
⟨linearized model identities 25a⟩


// XXXXXXXXXXXXXXXXXXXX
// Aggregate hours equals agg hours in each sector
⟨linearized model hours 25b⟩

// See Section 8: Data Identities

// new equations
// Durable Block

⟨linearized model durables 25c⟩

// Housing Block
⟨linearized model housing 26⟩

//measurement_equations;
⟨linearized model measurement 27⟩

//end_measurement_equations;

end;
```

Uses L 34 and LSS 37.

A.3.2 Linearized Model Prelim

23 ⟨*linearized model prelim* 23⟩≡ (22)

```
(RCSS*exp(RC))-(MCCSS*exp(MCC))*(YCSS*exp(YC))/(USS*exp(UC))/(KCSS*exp(KC(-1)))*alpha_*(MUKSS*e
(RKSS*exp(RK))-(MCKSS*exp(MCK))*(YKSS*exp(YK))/(USS*exp(UK))/(KKSS*exp(KK(-1)))*alpha_*(MUKSS*e
(WCSS*exp(WC))-(MCCSS*exp(MCC))*(YCSS*exp(YC))/(HCSS*exp(HC))*(1-alpha_)=0;
(WKSS*exp(WK))-(MCKSS*exp(MCK))*(YKSS*exp(YK))/(HKSS*exp(HK))*(1-alpha_)=0;
(YCSS*exp(YC))-((USS*exp(UC))*(KCSS*exp(KC(-1)))/(MUKSS*exp(MUK)))^alpha_*((HCSS*exp(HC)))^(1-a
(YKSS*exp(YK))-((USS*exp(UK))*(KKSS*exp(KK(-1)))/(MUKSS*exp(MUK)))^alpha_*((HKSS*exp(HK)))^(1-a
(MCCSS*exp(MCC))*(YCSS*exp(YC))*theta_c-(theta_c-1)*(YCSS*exp(YC))-100*phi_pc*((INFCSS*exp(INFC
(MCKSS*exp(MCK))*(YKSS*exp(YK))*theta_k/(PKBSS*exp(PKB))-(theta_k-1)*(YKSS*exp(YK))-100*phi_pc*
(QKSS*exp(QK))-beta_*(1/(ONE*exp(EFFK)))*(((1-delta_)*(QKSS*exp(QK(+1)))+(RCSS*exp(RC(+1)))*(US
(QKSS*exp(QK))-beta_*(1/(ONE*exp(EFFK)))*(((1-delta_)*(QKSS*exp(QK(+1)))+(RKSS*exp(RK(+1)))*(US
(LSS*exp(L))-(beta_*exp(betas))*(RSS*exp(R))/rpr/(INFCSS*exp(INFC(+1)))/(MUCSS*exp(MUC(+1)))*(L
ln((RSS*exp(R))/RSS)-rho_R*ln((RSS*exp(R(-1)))/RSS)-(1-rho_R)*(r_inf*ln((INFCNASS*exp(INFCNA))/
(LSS*exp(L))-eta_cnn/((ECSS*exp(EC))-h*(ECSS*exp(EC(-1)))/(MUCSS*exp(MUC)))+eta_cnn*beta_*h/((M
(KKSS*exp(KK))-(1-delta_)*(KKSS*exp(KK(-1)))/(MUKSS*exp(MUK))+(KCSS*exp(KC))-(1-delta_)*(KCSS*e
```

Uses alpha_ 37, beta_ 37, betas 34, delta_ 37, EC 34, ECSS 37, EFFK 34, EIK 34, EIKSS 37, ePMKC 36, ePMKK 36, eR 36, eta_cnn 37, gam_pc 37, h 37, HC 34, HCSS 37, HK 34, HKSS 37, INFC 34, INFCNA 34, INFCNASS 37, INFCSS 37, INFK 34, INFKSS 37, KC 34, KCSS 37, KK 34, KKSS 37, L 34, LSS 37, MCC 34, MCCSS 37, MCK 34, MCKSS 37, mu_ 37, MUC 34, MUCSS 37, MUK 34, MUKSS 37, ONE 37, PFGAP 34, phi_pc 37, phi_u 37, PKB 34, PKBSS 37, QK 34, QKSS 37, R 34, r_dinf 37, r_inf 37, r_y 37, RC 34, RCSS 37, rho_R 37, RK 34, RKSS 37, rpr 37, RSS 37, theta_c 37, theta_k 37, UC 34, UK 34, USS 37, WC 34, WCSS 37, WK 34, WKSS 37, YC 34, YCSS 37, YK 34, and YKSS 37.

A.3.3 Linearized Model Labor

24 ⟨*linearized model labor* 24⟩≡ (22)

```
-100+(UHCSS*exp(UHC))*theta_wc-(theta_wc-1)*(WCSS*exp(WC))-100*phi_wc*((INFWCSS*exp(I
(UHSCSS*exp(UHSC))-(WCSS*exp(WC))+phi_H/10*((HSCSS*exp(HSC))/(HSKSS*exp(HSK))-gam_h*(
-100+(UHKSS*exp(UHK))*theta_wk-(theta_wk-1)*(WKSS*exp(WK))-100*phi_wc*((INFWKSS*exp(I
(UHSKSS*exp(UHSK))-(WKSS*exp(WK))-phi_H/10*((HSCSS*exp(HSC))/(HSKSS*exp(HSK))-gam_h*(
(UHCSS*exp(UHC))*(LSS*exp(L))*(ONE*exp(Lpref))-A_HC*((1+sigman)/(1+sigman/(1+sigmah))
(UHSCSS*exp(UHSC))*(LSS*exp(L))*(ONE*exp(Lpref))-A_HC*((1+sigman)/(1+sigman/(1+sigmah
(UHKSS*exp(UHK))*(LSS*exp(L))*(ONE*exp(Lpref))-A_HK*((1+sigman)/(1+sigman/(1+sigmah))
(UHSKSS*exp(UHSK))*(LSS*exp(L))*(ONE*exp(Lpref))-A_HK*((1+sigman)/(1+sigman/(1+sigmah
(empCSS*exp(empC))-((1+sigmah)/sigmah*xsi_NC/xsi_HrC)^(-1/(1+sigmah+sigman))*(HCSS*ex
(HrCSS*exp(HrC))-((1+sigmah)/sigmah*xsi_NC/xsi_HrC)^(1/(1+sigmah))*(empCSS*exp(empC))
(empKSS*exp(empK))-((1+sigmah)/sigmah*xsi_NK/xsi_HrK)^(-1/(1+sigmah+sigman))*(HKSS*ex
(HrKSS*exp(HrK))-((1+sigmah)/sigmah*xsi_NK/xsi_HrK)^(1/(1+sigmah))*(empKSS*exp(empK))
(empSCSS*exp(empSC))-((1+sigmah)/sigmah*xsi_NC/xsi_HrC)^(-1/(1+sigmah+sigman))*(HSCSS
(HrSCSS*exp(HrSC))-((1+sigmah)/sigmah*xsi_NC/xsi_HrC)^(1/(1+sigmah))*(empSCSS*exp(emp
(empSKSS*exp(empSK))-((1+sigmah)/sigmah*xsi_NK/xsi_HrK)^(-1/(1+sigmah+sigman))*(HSKSS
(HrSKSS*exp(HrSK))-((1+sigmah)/sigmah*xsi_NK/xsi_HrK)^(1/(1+sigmah))*(empSKSS*exp(emp
(unempSS*exp(unemp))-((empSCSS*exp(empSC))+(empSKSS*exp(empSK))-((empCSS*exp(empC))+(
(PKBSS*exp(PKB))-(1-100*phi_ic*((EIKSS*exp(EIK))-gam_ic*(EIKSS*exp(EIK(-1)))-(1-gam_i
(YCSS*exp(YC))-(ECSS*exp(EC))-(ECHSS*exp(ECH))-0.2*YCSS*(ONE*exp(HG))=0;
ln((INFWCSS*exp(INFWC)))-ln((WCSS*exp(WC)))+ln((WCSS*exp(WC(-1))))-ln((MUCSS*exp(MUC)
ln((INFWKSS*exp(INFWK)))-ln((WKSS*exp(WK)))+ln((WKSS*exp(WK(-1))))-ln((MUCSS*exp(MUC)
ln((INFKSS*exp(INFK)))-ln((INFCSS*exp(INFC)))-ln((PKBSS*exp(PKB)))+ln((PKBSS*exp(PKB(
(YKSS*exp(YK))-(EIKSS*exp(EIK))-(ECDSS*exp(ECD))-0.2*YKSS*(ONE*exp(HG))=0;
ln((ONE*exp(DIFFNORMGDP)))-(1-s_k)*(ln((YCSS*exp(YC)))-ln((YCSS*exp(YC(-1)))))-s_k*(l
ln((ONE*exp(NORMINFGDP)))-s_k*(ln((PKBSS*exp(PKB)))-ln((PKBSS*exp(PKB(-1)))))=0;
ln((DIFFREALGDPSS*exp(DIFFREALGDP)))-ln((ONE*exp(DIFFNORMGDP)))-(1-s_k)*ln((MUCSS*exp
ln((DIFFREALECSS*exp(DIFFREALEC)))-ln((ECSS*exp(EC)))+ln((ECSS*exp(EC(-1))))-ln((MUCS
ln((DIFFREALEIKSS*exp(DIFFREALEIK)))-ln((EIKSS*exp(EIK)))+ln((EIKSS*exp(EIK(-1))))-ln
```

Uses A_HC 37, A_HK 37, beta_ 37, DIFFNORMGDP 34, DIFFREALEC 34, DIFFREALECSS 37, DIFFREALEIK 34, DIFFREALEIKSS 37, DIFFREALGDP 34, DIFFREALGDPSS 37, EC 34, ECD 34, ECDSS 37, ECH 34, ECHSS 37, ECSS 37, EFFK 34, EIK 34, EIKSS 37, empC 34, empCSS 37, empK 34, empKSS 37, empSC 34, empSCSS 37, empSK 34, empSKSS 37, eXiL 36, gam_h 37, gam_ic 37, gam_wc 37, HC 34, HCSS 37, HG 34, HK 34, HKSS 37, HrC 34, HrCSS 37, HrK 34, HrKSS 37, HrSC 34, HrSCSS 37, HrSK 34, HrSKSS 37, HSC 34, HSCSS 37, HSK 34, HSKSS 37, INFC 34, INFCSS 37, INFK 34, INFKSS 37, INFWC 34, INFWCSS 37, INFWK 34, INFWKSS 37, KC 34, KCSS 37, KK 34, KKSS 37, L 34, Lpref 34, LSS 37, MUC 34, MUCSS 37, MUK 34, MUKSS 37, NORMINFGDP 34, ONE 37, phi_H 37, phi_ic 37, phi_wc 37, PKB 34, PKBSS 37, QK 34, QKSS 37, s_k 37, sigmah 37, sigman 37, theta_wc 37, theta_wk 37, UHC 34, UHCSS 37, UHK 34, UHKSS 37, UHSC 34, UHSCSS 37, UHSK 34, UHSKSS 37, unemp 34, unempSS 37, WC 34, WCSS 37, WK 34, WKSS 37, XiL 34, xsi_HrC 37, xsi_HrK 37, xsi_NC 37, xsi_NK 37, YC 34, YCSS 37, YK 34, and YKSS 37.

A.3.4 Linearized Model Identities

25a ⟨*linearized model identities* 25a⟩≡ (22)

```
ln((DIFFREALWSS*exp(DIFFREALW)))-HCSS/AHSS*(ln((INFWCSS*exp(INFWC))))-HKSS/AHSS*(ln((INFWKSS*ex
```

Uses AHSS 37, DIFFREALW 34, DIFFREALWSS 37, HCSS 37, HKSS 37, INFC 34, INFCSS 37, INFWC 34, INFWCSS 37, INFWK 34, and INFWKSS 37.

A.3.5 Linearized Model Hours

25b ⟨*linearized model hours* 25b⟩≡ (22)

```
(AHSS*exp(AH))-(HCSS*exp(HC))-(HKSS*exp(HK))=0;
ln((INFGDPSS*exp(INFGDP)))-ln((INFCSS*exp(INFC)))-ln((YCSS*exp(YC))*(MUCSS*exp(MUC))/(YCSS*exp(
ln((INFCNASS*exp(INFCNA)))-(1-s_ecdc)*ln((INFCSS*exp(INFC)))-s_ecdc*ln((INFKSS*exp(INFK)))=0;
ln((INFCORSS*exp(INFCOR)))-(1-s_ecdc)*ln((INFCSS*exp(INFC)))-s_ecdc*ln((INFKSS*exp(INFK)))=0;
ln((ONE*exp(GAP)))-(1-s_k)*ln((YCSS*exp(YC))/YCSS)-s_k*ln((YKSS*exp(YK))/YKSS)=0;
ln((ONE*exp(PFGAP)))-(1-alpha_)*((1-s_k)*ln((HCSS*exp(HC))/HCSS)+s_k*ln((HKSS*exp(HK))/HKSS))-a
ln((INFC10SS*exp(INFC10)))-betarl*ln((INFC10SS*exp(INFC10(+1))))-(1-betarl)*ln((INFCORSS*exp(IN
```

Uses AH 34, AHSS 37, alpha_ 37, betarl 37, DIFFREALGDP 34, DIFFREALGDPSS 37, GAP 34, HC 34, HCSS 37, HK 34, HKSS 37, INFC 34, INFC10 34, INFC10SS 37, INFCNA 34, INFCNASS 37, INFCOR 34, INFCORSS 37, INFCSS 37, INFGDP 34, INFGDPSS 37, INFK 34, INFKSS 37, MUC 34, MUCSS 37, ONE 37, PFGAP 34, PKB 34, PKBSS 37, s_ecdc 37, s_k 37, UC 34, UK 34, USS 37, YC 34, YCSS 37, YK 34, and YKSS 37.

A.3.6 Linearized Model Durables

25c ⟨*linearized model durables* 25c⟩≡ (22)

```
(KCDSS*exp(KD))-(1-delta_cd)*(KCDSS*exp(KD(-1)))/(MUKSS*exp(MUK))-(ECDSS*exp(ECD))=0;
(LSS*exp(L))*(RCDSS*exp(RCD))-eta_cd/((KCDSS*exp(KD(-1)))/(MUKSS*exp(MUK))-h_cd*(KCDSS*exp(LAGK
(QCDSS*exp(QCD))-beta_*(1/(ONE*exp(EFFECD)))*(LSS*exp(L(+1)))/(LSS*exp(L))/(MUKSS*exp(MUK(+1)))
(PKBSS*exp(PKB))-(QCDSS*exp(QCD))*(1-100*phi_cd*((ECDSS*exp(ECD))-gam_icd*(ECDSS*exp(ECD(-1)))-
```

Uses beta_ 37, delta_cd 37, ECD 34, ECDSS 37, EFFECD 34, eta_cd 37, gam_icd 37, h_cd 37, KCDSS 37, KD 34, L 34, LAGKD 34, LSS 37, MUK 34, MUKSS 37, ONE 37, phi_cd 37, PKB 34, PKBSS 37, QCD 34, QCDSS 37, RCD 34, and RCDSS 37.

A.3.7 Linearized Model Housing

26 ⟨*linearized model housing* 26⟩≡ (22)

```
(LSS*exp(L))*(RCHSS*exp(RCH))-eta_ch/((KCHSS*exp(KCH(-1)))/(MUCSS*exp(MUC))-h_ch*(KC
(QCHSS*exp(QCH))-beta_*(1/(ONE*exp(EFFECH)))*(LSS*exp(L(+1)))/(LSS*exp(L))/(MUCSS*ex
1*(ECHSS*exp(ECH))+(1-delta_ch)*(KCHSS*exp(KCH(-1)))/(MUCSS*exp(MUC))-(KCHSS*exp(KCH
1-(QCHSS*exp(QCH))*(1-100*phi_ech*((ECHSS*exp(ECH))-gam_ech*(ECHSS*exp(ECH(-1)))-(1-
ln((KCDSS*exp(KD(-1))))-ln((KCDSS*exp(LAGKD)))=0;
ln((KCHSS*exp(KCH(-1))))-ln((KCHSS*exp(LAGKCH)))=0;
(RKSS*exp(RK))-(QKSS*exp(QK))*mu_*(USS*exp(UK))^(1/phi_u)=0;
(RCSS*exp(RC))-(QKSS*exp(QK))*mu_*(USS*exp(UC))^(1/phi_u)=0;
ln((DIFFREALECHSS*exp(DIFFREALECH)))-ln((MUCSS*exp(MUC)))-ln((ECHSS*exp(ECH)))+ln((E
ln((DIFFREALECDSS*exp(DIFFREALECD)))-ln((MUKSS*exp(MUK)))-ln((ECDSS*exp(ECD)))+ln((E
ln((beta_*exp(betas))/beta_)-rho_B*ln((beta_*exp(betas(-1)))/beta_)-eB=0;
ln((ONE*exp(XiL)))-rho_XiL*ln((ONE*exp(XiL(-1))))-eXiL=0;
ln((ONE*exp(Lpref)))-rho_lpref*ln((ONE*exp(Lpref(-1))))-eLpref=0;
ln((ONE*exp(EFFK)))-rho_EFFK*ln((ONE*exp(EFFK(-1))))-eEFFK=0;
ln((MUZKSS*exp(MUZK))/MUZKSS)-eMUZK=0;
ln((MUZMSS*exp(MUZM))/MUZMSS)-eMUZM=0;
ln((ONE*exp(HG)))-rho_HG*ln((ONE*exp(HG(-1))))-eHG=0;
ln((MUCSS*exp(MUC)))-ln((MUZMSS*exp(MUZM)))-alpha_*ln((MUZKSS*exp(MUZK)))=0;
ln((MUKSS*exp(MUK)))-ln((MUZMSS*exp(MUZM)))-ln((MUZKSS*exp(MUZK)))=0;
ln((ONE*exp(EFFECD)))-rho_EFFECD*ln((ONE*exp(EFFECD(-1))))-eEFFECD=0;
ln((ONE*exp(EFFECH)))-rho_EFFECH*ln((ONE*exp(EFFECH(-1))))-eEFFECH=0;
ln((ONE*exp(STAR)))-rho_STAR*ln((ONE*exp(STAR(-1))))-eSTAR=0;
ln((RL1SS*exp(RL1))) - ln((RSS*exp(R(+1))))=0;
ln((RL2SS*exp(RL2))) - ln((RL1SS*exp(RL1(+1))))=0;
ln((RL3SS*exp(RL3))) - ln((RL2SS*exp(RL2(+1))))=0;
ln((RL4SS*exp(RL4))) - ln((RL3SS*exp(RL3(+1))))=0;
ln((RL5SS*exp(RL5))) - ln((RL4SS*exp(RL4(+1))))=0;
ln((RL6SS*exp(RL6))) - ln((RL5SS*exp(RL5(+1))))=0;
ln((RL7SS*exp(RL7))) - ln((RL6SS*exp(RL6(+1))))=0;
ln((RT2SS*exp(RT2))) - tp2 - 0.125*(ln((RSS*exp(R))) + ln((RL1SS*exp(RL1))) + ln((RL
```

Uses `alpha_` 37, `beta_` 37, `betas` 34, `delta_ch` 37, `DIFFREALECD` 34, `DIFFREALECDSS` 37, `DIFFREALECH` 34, `DIFFREALECHSS` 37, `eB` 36, `ECD` 34, `ECDSS` 37, `ECH` 34, `ECHSS` 37, `eEFFECD` 36, `eEFFECH` 36, `eEFFK` 36, `EFFECD` 34, `EFFECH` 34, `EFFK` 34, `eHG` 36, `eLpref` 36, `eMUZK` 36, `eMUZM` 36, `eSTAR` 36, `eta_ch` 37, `eXiL` 36, `gam_ech` 37, `h_ch` 37, `HG` 34, `KCDSS` 37, `KCH` 34, `KCHSS` 37, `KD` 34, `L` 34, `LAGKCH` 34, `LAGKD` 34, `Lpref` 34, `LSS` 37, `mu_` 37, `MUC` 34, `MUCSS` 37, `MUK` 34, `MUKSS` 37, `MUZK` 34, `MUZKSS` 37, `MUZM` 34, `MUZMSS` 37, `ONE` 37, `phi_ech` 37, `phi_u` 37, `QCH` 34, `QCHSS` 37, `QK` 34, `QKSS` 37, `R` 34, `RC` 34, `RCH` 34, `RCHSS` 37, `RCSS` 37, `rho_B` 37, `rho_EFFECD` 37, `rho_EFFECH` 37, `rho_EFFK` 37, `rho_HG` 37, `rho_lpref` 37, `rho_STAR` 37, `rho_XiL` 37, `RK` 34, `RKSS` 37, `RL1` 34, `RL1SS` 37, `RL2` 34, `RL2SS` 37, `RL3` 34, `RL3SS` 37, `RL4` 34, `RL4SS` 37, `RL5` 34, `RL5SS` 37, `RL6` 34, `RL6SS` 37, `RL7` 34, `RL7SS` 37, `RSS` 37, `RT2` 34, `RT2SS` 37, `STAR` 34, `tp2` 37, `UC` 34, `UK` 34, `USS` 37, and `XiL` 34.

A.3.8 Linearized Model Measurement

27 ⟨*linearized model measurement* 27⟩≡ (22)

```
DIFFREALGDP_obs = DIFFREALGDP + DIFFREALGDPSS_obs;
DIFFREALEC_obs = DIFFREALEC + DIFFREALECSS_obs;
DIFFREALEIK_obs = DIFFREALEIK + DIFFREALEIKSS_obs;
DIFFREALECD_obs = DIFFREALECD + DIFFREALECDSS_obs;
DIFFREALECH_obs = DIFFREALECH + DIFFREALECHSS_obs;
DIFFREALW_obs = DIFFREALW + DIFFREALWSS_obs;
AH_obs = AH;
INFCNA_obs = INFCNA + INFCNASS_obs;
INFCOR_obs = INFCOR + INFCORSS_obs;
INFK_obs = INFK + INFKSS_obs;
R_obs = R + RSS_obs;
RT2_obs = RT2 + RT2SS_obs;
unemp_obs = unemp + unempSS_obs;
```

Uses AH 34, AH_obs 34, DIFFREALEC 34, DIFFREALEC_obs 34, DIFFREALECD 34, DIFFREALECD_obs 34, DIFFREALECDSS_obs 37, DIFFREALECH 34, DIFFREALECH_obs 34, DIFFREALECHSS_obs 37, DIFFREALECSS_obs 37, DIFFREALEIK 34, DIFFREALEIK_obs 34, DIFFREALEIKSS_obs 37, DIFFREALGDP 34, DIFFREALGDP_obs 34, DIFFREALGDPSS_obs 37, DIFFREALW 34, DIFFREALW_obs 34, DIFFREALWSS_obs 37, INFCNA 34, INFCNA_obs 34, INFCNASS_obs 37, INFCOR 34, INFCOR_obs 34, INFCORSS_obs 37, INFK 34, INFK_obs 34, INFKSS_obs 37, R 34, R_obs 34, RSS_obs 37, RT2 34, RT2_obs 34, RT2SS_obs 37, unemp 34, unemp_obs 34, and unempSS_obs 37.

A.4 linearized_steadystate.m

28 ⟨*srcedo/linearized.steadystate.m* 28⟩≡

```
function [ys,check] = linearized_steadystate(ys,exe)
        global M_

check = 0;

NumberofParameters=M_.param_nbr;
for i=1:NumberofParameters
    paramname=deblank(M_.param_names(i,:));
    eval([paramname ' =M_.params(' int2str(i) ');']);
end;

%start_steady_state;

⟨common steady state values 47⟩

%end_steady_state;

%trends;

⟨common steady state trends 50⟩

%end_trends;


for i=1:NumberofParameters
    paramname=deblank(M_.param_names(i,:));
    eval(['M_.params(' int2str(i) ')=' paramname ';']);
end;

⟨linearized steady state result return 29⟩
```

This code is written to file `srcedo/linearized.steadystate.m`.

A.4.1 Linearized Steady State Result Return

29 ⟨*linearized steady state result return* 29⟩≡ (28)

```
ys = [
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
```

```
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
DIFFREALGDPSS_obs
DIFFREALECSS_obs
DIFFREALEIKSS_obs
DIFFREALECDSS_obs
```

```
  DIFFREALECHSS_obs
  DIFFREALWSS_obs
  0
  INFCNASS_obs
  INFCORSS_obs
  INFKSS_obs
  RSS_obs
  RT2SS_obs
  unempSS_obs
  ];
```

Uses DIFFREALECDSS_obs 37, DIFFREALECHSS_obs 37, DIFFREALECSS_obs 37, DIFFREALEIKSS_obs 37, DIFFREALGDPSS_obs 37, DIFFREALWSS_obs 37, INFCNASS_obs 37, INFCORSS_obs 37, INFKSS_obs 37, RSS_obs 37, RT2SS_obs 37, and unempSS_obs 37.

A.5 Common Model Routines

A.5.1 Common Model Setup

32 ⟨*common setup* 32⟩≡ (11 21)

```
⟨common var 34⟩


⟨common varexo 36⟩


⟨common parameters 37⟩


//estimated_params;
⟨common estimated params 41⟩
//end_estimated_params;

//calibrated_params;
⟨common calibrated params 42⟩
//end_calibrated_params;

//free_params;
⟨common free params 43a⟩
//end_free_params;

//calibrated ME


//****************************
//MODEL BLOCK
//****************************

model;
```

A.5.2 Common Stochastic Simulation

33 ⟨*common stoch sim* 33⟩≡ (11 21)

```
⟨common varobs 43b⟩

shocks;
⟨common shocks 44⟩
end;

steady;

estimated_params;
⟨common steady estimated params 45⟩

⟨common stderr 46⟩
end;


options_.order = 1;
options_.jacobian_flag = 1;
options_.nonlin = 1;

stoch_simul(order=1,irf=40,nograph);
```

Defines:
`jacobian_flag`, never used.
`nonlin`, never used.
`options_`, never used.
`order`, never used.
`stoch_simul`, never used.

A.5.3 Common Var

34 ⟨*common var* 34⟩≡ (32)

```
var RC RK WC WK YC YK MCC MCK KC  KK PKB R L QK HC  HSC  HK HSK  UHC UHSC UHK UHSK en
 DIFFREALGDP_obs DIFFREALEC_obs DIFFREALEIK_obs DIFFREALECD_obs DIFFREALECH_obs DIFFI
```

Defines:

AH, used in chunks 15a, 17, 25b, and 27.
AH_obs, used in chunks 17, 27, 43b, and 44.
betas, used in chunks 13, 16, 23, and 26.
DIFFNORMGDP, used in chunks 14a and 24.
DIFFREALEC, used in chunks 14a, 17, 24, and 27.
DIFFREALEC_obs, used in chunks 17, 27, 43b, and 44.
DIFFREALECD, used in chunks 16, 17, 26, and 27.
DIFFREALECD_obs, used in chunks 17, 27, 43b, and 44.
DIFFREALECH, used in chunks 16, 17, 26, and 27.
DIFFREALECH_obs, used in chunks 17, 27, 43b, and 44.
DIFFREALEIK, used in chunks 14a, 17, 24, and 27.
DIFFREALEIK_obs, used in chunks 17, 27, 43b, and 44.
DIFFREALGDP, used in chunks 14a, 15a, 17, 24, 25b, and 27.
DIFFREALGDP_obs, used in chunks 17, 27, 43b, and 44.
DIFFREALW, used in chunks 14b, 17, 25a, and 27.
DIFFREALW_obs, used in chunks 17, 27, 43b, and 44.
EC, used in chunks 13, 14a, 23, and 24.
ECD, used in chunks 14–16 and 24–26.
ECH, used in chunks 14a, 16, 24, and 26.
EFFECD, used in chunks 15b, 16, 25c, and 26.
EFFECH, used in chunks 16 and 26.
EFFK, used in chunks 13, 14a, 16, 23, 24, and 26.
EIK, used in chunks 13, 14a, 23, and 24.
empC, used in chunks 14a and 24.
empK, used in chunks 14a and 24.
empSC, used in chunks 14a and 24.
empSK, used in chunks 14a and 24.
GAP, used in chunks 15a and 25b.
HC, used in chunks 13–15 and 23–25.
HG, used in chunks 14a, 16, 24, and 26.
HK, used in chunks 13–15 and 23–25.
HrC, used in chunks 14a and 24.
HrK, used in chunks 14a and 24.
HrSC, used in chunks 14a and 24.
HrSK, used in chunks 14a and 24.
HSC, used in chunks 14a and 24.
HSK, used in chunks 14a and 24.
INFC, used in chunks 13–15 and 23–25.
INFC10, used in chunks 15a and 25b.
INFCNA, used in chunks 13, 15a, 17, 23, 25b, and 27.
INFCNA_obs, used in chunks 17, 27, 43b, and 44.
INFCOR, used in chunks 15a, 17, 25b, and 27.
INFCOR_obs, used in chunks 17, 27, 43b, and 44.
INFGDP, used in chunks 15a and 25b.
INFK, used in chunks 13–15, 17, 23–25, and 27.
INFK_obs, used in chunks 17, 27, 43b, and 44.
INFWC, used in chunks 14, 24, and 25a.
INFWK, used in chunks 14, 24, and 25a.
KC, used in chunks 13, 14a, 23, and 24.
KCH, used in chunks 16 and 26.
KD, used in chunks 15b, 16, 25c, and 26.

KK, used in chunks 13, 14a, 23, and 24.
L, used in chunks 12–16 and 22–26.
LAGKCH, used in chunks 16 and 26.
LAGKD, used in chunks 15b, 16, 25c, and 26.
Lpref, used in chunks 14a, 16, 24, and 26.
MCC, used in chunks 13 and 23.
MCK, used in chunks 13 and 23.
MUC, used in chunks 13–16 and 23–26.
MUK, used in chunks 13–16 and 23–26.
MUZK, used in chunks 16 and 26.
MUZM, used in chunks 16 and 26.
NORMINFGDP, used in chunks 14a and 24.
PFGAP, used in chunks 13, 15a, 23, and 25b.
PKB, used in chunks 13–15 and 23–25.
QCD, used in chunks 15b and 25c.
QCH, used in chunks 16 and 26.
QK, used in chunks 13, 14a, 16, 23, 24, and 26.
R, used in chunks 13, 16, 17, 23, 26, and 27.
R_obs, used in chunks 17, 27, and 43b.
RC, used in chunks 13, 16, 23, and 26.
RCD, used in chunks 15b and 25c.
RCH, used in chunks 16 and 26.
RK, used in chunks 13, 16, 23, and 26.
RL1, used in chunks 16 and 26.
RL2, used in chunks 16 and 26.
RL3, used in chunks 16 and 26.
RL4, used in chunks 16 and 26.
RL5, used in chunks 16 and 26.
RL6, used in chunks 16 and 26.
RL7, used in chunks 16 and 26.
RT2, used in chunks 16, 17, 26, and 27.
RT2_obs, used in chunks 17, 27, 43b, and 44.
STAR, used in chunks 16 and 26.
UC, used in chunks 13, 15a, 16, 23, 25b, and 26.
UHC, used in chunks 14a and 24.
UHK, used in chunks 14a and 24.
UHSC, used in chunks 14a and 24.
UHSK, used in chunks 14a and 24.
UK, used in chunks 13, 15a, 16, 23, 25b, and 26.
unemp, used in chunks 14a, 17, 24, and 27.
unemp_obs, used in chunks 17, 27, 43b, and 44.
WC, used in chunks 13, 14a, 23, and 24.
WK, used in chunks 13, 14a, 23, and 24.
XiL, used in chunks 14a, 16, 24, and 26.
YC, used in chunks 13–15 and 23–25.
YK, used in chunks 13–15 and 23–25.

A.5.4 Common VarExo

36 ⟨*common varexo* 36⟩≡ (32)

```
varexo eHG eXiL eLpref eR eMUZK eMUZM ePMKC ePMKK eEFFECH eEFFECD eEFFK eB eSTAR;
```

Defines:
- `eB`, used in chunks 16, 26, 44, and 46.
- `eEFFECD`, used in chunks 16, 26, 44, and 46.
- `eEFFECH`, used in chunks 16, 26, 44, and 46.
- `eEFFK`, used in chunks 16, 26, 44, and 46.
- `eHG`, used in chunks 16, 26, 44, and 46.
- `eLpref`, used in chunks 16, 26, 44, and 46.
- `eMUZK`, used in chunks 16, 26, 44, and 46.
- `eMUZM`, used in chunks 16, 26, 44, and 46.
- `ePMKC`, used in chunks 13, 23, 44, and 46.
- `ePMKK`, used in chunks 13, 23, 44, and 46.
- `eR`, used in chunks 13, 23, 44, and 46.
- `eSTAR`, used in chunks 16, 26, 44, and 46.
- `eXiL`, used in chunks 14a, 16, 24, 26, 44, and 46.

A.5.5 Common Parameters

37 ⟨*common parameters* 37⟩≡ (32)

```
parameters
h r_inf r_y r_dy phi_pc phi_H phi_wc phi_ic phi_cd phi_ech gam_pc gam_wc gam_ic gam_icd rho_R r
rho_EFFECD rho_HG rho_EFFECH tp2 ONE MUZMSS MUZKSS  r_dinf rpr phi_u rho_MUZK rho_MUZM pbeta de
theta_k theta_wc theta_wk g_y a_ks s_AS gam_h gam_ech  s_k s_ecdc eta_cnn eta_cd eta_ch
icoef mu_ betarl MUZCSS RCSS RKSS WCSS WKSS YCSS YKSS MCCSS MCKSS KCSS KKSS LSS HCSS HKSS QKSS
MUCSS MUKSS AHSS ECDSS KCDSS QCDSS RCDSS ECHSS KCHSS QCHSS RCHSS UKSS UCSS USS MUKSShabit MUCSS
INFCNASS INFCORSS INFC10SS  RT2SS beta_0 beta_2 beta_ PYSS AA DD RR
eta_cd_eta_cnn eta_ch_eta_cnn Rnr ycbi_ykb hc_hk HSS ycbi ykb YYSS s_k_ecd s_c_ech s_k_eik s_yc
sig_HG sig_XiL sig_lpref sig_R sig_MUZK sig_MUZM sig_PMKC sig_PMKK sig_EFFECH  sig_EFFECD sig_E
HSKSS HSCSS HrCSS HrKSS A_HC sigman sigmah A_HK xsi_NC xsi_HrC xsi_NK xsi_HrK rho_XiL rho_lpref
empCSS empKSS HrSKSS HrSCSS empSCSS empSKSS UHCSS UHKSS UHSCSS UHSKSS unempSS DIFFREALGDPSS DIF
DIFFREALECHSS DIFFREALEIKSS DIFFREALWSS  RL1SS RL2SS RL3SS RL4SS RL5SS
RL6SS RL7SS DIFFREALGDPSS_obs DIFFREALECSS_obs DIFFREALEIKSS_obs DIFFREALECDSS_obs
DIFFREALECHSS_obs DIFFREALWSS_obs INFCNASS_obs INFCORSS_obs INFKSS_obs
RSS_obs RT2SS_obs unempSS_obs;
```

Defines:
- A_HC, used in chunks 14a, 24, 43a, and 47.
- A_HK, used in chunks 14a, 24, 43a, and 47.
- a_ks, used in chunk 42.
- AA, used in chunk 47.
- AHSS, used in chunks 14b, 17, 19, 25, and 47.
- alpha_, used in chunks 13, 15a, 16, 23, 25b, 26, 42, and 47.
- beta_, used in chunks 13–16, 19, 23–26, and 47.
- beta_0, used in chunk 47.
- beta_2, used in chunk 47.
- betarl, used in chunks 15a, 25b, and 42.
- DD, used in chunk 47.
- delta_, used in chunks 13, 23, 42, and 47.
- delta_cd, used in chunks 15b, 25c, 42, and 47.
- delta_ch, used in chunks 16, 26, 42, and 47.
- DIFFREALECDSS, used in chunks 17, 19, 26, and 47.
- DIFFREALECDSS_obs, used in chunks 17, 19, 27, 29, and 50.
- DIFFREALECHSS, used in chunks 17, 19, 26, and 47.
- DIFFREALECHSS_obs, used in chunks 17, 19, 27, 29, and 50.
- DIFFREALECSS, used in chunks 17, 19, 24, and 47.
- DIFFREALECSS_obs, used in chunks 17, 19, 27, 29, and 50.
- DIFFREALEIKSS, used in chunks 17, 19, 24, and 47.
- DIFFREALEIKSS_obs, used in chunks 17, 19, 27, 29, and 50.
- DIFFREALGDPSS, used in chunks 17, 19, 24, 25b, and 47.
- DIFFREALGDPSS_obs, used in chunks 17, 19, 27, 29, and 50.
- DIFFREALWSS, used in chunks 17, 19, 25a, and 47.
- DIFFREALWSS_obs, used in chunks 17, 19, 27, 29, and 50.
- ECDSS, used in chunks 15b, 19, 24–26, and 47.
- ECHSS, used in chunks 16, 19, 24, 26, and 47.
- ECSS, used in chunks 19, 23, 24, and 47.
- EIKSS, used in chunks 14a, 19, 23, 24, and 47.
- empCSS, used in chunks 19, 24, and 47.
- empKSS, used in chunks 19, 24, and 47.
- empSCSS, used in chunks 19, 24, and 47.
- empSKSS, used in chunks 19, 24, and 47.

eta_cd, used in chunks 15b, 25c, 43a, and 47.
eta_cd_eta_cnn, used in chunk 47.
eta_ch, used in chunks 16, 26, 43a, and 47.
eta_ch_eta_cnn, used in chunk 47.
eta_cnn, used in chunks 13, 23, 43a, and 47.
g_y, used in chunk 42.
gam_ech, used in chunks 16, 26, and 42.
gam_h, used in chunks 14a, 24, and 42.
gam_ic, used in chunks 14a, 24, and 42.
gam_icd, used in chunks 15b, 25c, and 42.
gam_pc, used in chunks 13, 23, 41, and 45.
gam_wc, used in chunks 14a, 24, 41, and 45.
h, used in chunks 13, 23, 41, 45, and 47.
h_cd, used in chunks 15b, 25c, 42, and 47.
h_ch, used in chunks 16, 26, 42, and 47.
hc_hk, used in chunk 47.
HCSS, used in chunks 14, 15a, 19, 23–25, and 47.
HKSS, used in chunks 14, 15a, 19, 23–25, and 47.
HrCSS, used in chunks 19, 24, and 47.
HrKSS, used in chunks 19, 24, and 47.
HrSCSS, used in chunks 19, 24, and 47.
HrSKSS, used in chunks 19, 24, and 47.
HSCSS, used in chunks 14a, 19, 24, and 47.
HSKSS, used in chunks 14a, 19, 24, and 47.
HSS, used in chunk 47.
icoef, used in chunk 42.
IMPHSSS, used in chunk 47.
INFC10SS, used in chunks 19, 25b, and 47.
INFCNASS, used in chunks 13, 17, 19, 23, 25b, and 47.
INFCNASS_obs, used in chunks 17, 19, 27, 29, and 50.
INFCORSS, used in chunks 17, 19, 25b, and 47.
INFCORSS_obs, used in chunks 17, 19, 27, 29, and 50.
INFCSS, used in chunks 13, 19, 23–25, 47, and 50.
INFGDPSS, used in chunks 19, 25b, and 47.
INFKSS, used in chunks 13, 17, 19, 23–25, 47, and 50.
INFKSS_obs, used in chunks 17, 19, 27, 29, and 50.
INFWCSS, used in chunks 14a, 19, 24, 25a, and 47.
INFWKSS, used in chunks 14a, 19, 24, 25a, and 47.
KCDSS, used in chunks 19, 25c, 26, and 47.
KCHSS, used in chunks 19, 26, and 47.
KCSS, used in chunks 13, 19, 23, 24, and 47.
KKSS, used in chunks 13, 19, 23, 24, and 47.
LSS, used in chunks 19, 22–26, and 47.
MCCSS, used in chunks 19, 23, and 47.
MCKSS, used in chunks 19, 23, and 47.
mu_, used in chunks 13, 16, 23, 26, 43a, and 47.
MUCSS, used in chunks 19, 23–26, 47, and 50.
MUCSShabit, used in chunk 47.
MUKSS, used in chunks 19, 23–26, 47, and 50.
MUKSShabit, used in chunk 47.
MUZCSS, used in chunk 47.
MUZKSS, used in chunks 16, 19, 26, 42, and 47.
MUZMSS, used in chunks 16, 19, 26, 42, and 47.
ONE, used in chunks 19, 23–26, 42, and 47.
pbeta, used in chunks 42 and 47.
phi_cd, used in chunks 15b, 25c, 41, and 45.
phi_ech, used in chunks 16, 26, 41, and 45.

phi_H, used in chunks 14a, 24, 41, and 45.
phi_ic, used in chunks 14a, 24, 41, and 45.
phi_pc, used in chunks 13, 23, 41, and 45.
phi_u, used in chunks 13, 16, 23, 26, and 42.
phi_wc, used in chunks 14a, 24, 41, and 45.
PKBSS, used in chunks 19, 23–25, and 47.
PYSS, used in chunk 47.
QCDSS, used in chunks 19, 25c, and 47.
QCHSS, used in chunks 19, 26, and 47.
QKSS, used in chunks 19, 23, 24, 26, and 47.
r_dinf, used in chunks 13, 23, and 42.
r_dy, used in chunk 42.
r_inf, used in chunks 13, 23, 41, and 45.
r_y, used in chunks 13, 23, 41, and 45.
RCDSS, used in chunks 19, 25c, and 47.
RCHSS, used in chunks 19, 26, and 47.
RCSS, used in chunks 19, 23, 26, and 47.
rho_B, used in chunks 16, 26, 41, and 45.
rho_EFFECD, used in chunks 16, 26, 41, and 45.
rho_EFFECH, used in chunks 16, 26, 41, and 45.
rho_EFFK, used in chunks 16, 26, 41, and 45.
rho_HG, used in chunks 16, 26, 41, and 45.
rho_lpref, used in chunks 16, 26, 41, and 45.
rho_MUZK, used in chunk 42.
rho_MUZM, used in chunk 42.
rho_R, used in chunks 13, 23, 41, and 45.
rho_STAR, used in chunks 16, 26, 41, and 45.
rho_XiL, used in chunks 16, 26, 41, and 45.
RKSS, used in chunks 19, 23, 26, and 47.
RL1SS, used in chunks 19, 26, and 47.
RL2SS, used in chunks 19, 26, and 47.
RL3SS, used in chunks 19, 26, and 47.
RL4SS, used in chunks 19, 26, and 47.
RL5SS, used in chunks 19, 26, and 47.
RL6SS, used in chunks 19, 26, and 47.
RL7SS, used in chunks 19, 26, and 47.
Rnr, used in chunk 47.
rpr, used in chunks 13, 23, 42, and 47.
RR, used in chunk 47.
RSS, used in chunks 13, 17, 19, 23, 26, 47, and 50.
RSS_obs, used in chunks 17, 19, 27, 29, and 50.
RT2SS, used in chunks 17, 19, 26, 47, and 50.
RT2SS_obs, used in chunks 17, 19, 27, 29, and 50.
s_AS, used in chunks 42 and 47.
s_c_ech, used in chunk 47.
s_ecdc, used in chunks 15a, 25b, 43a, 47, and 50.
s_k, used in chunks 14a, 15a, 24, 25b, 43a, 47, and 50.
s_k_ecd, used in chunk 47.
s_k_eik, used in chunk 47.
s_yc, used in chunk 47.
sig_B, used in chunks 41 and 44.
sig_EFFECD, used in chunks 41 and 44.
sig_EFFECH, used in chunks 41 and 44.
sig_EFFK, used in chunks 41 and 44.
sig_HG, used in chunks 41 and 44.
sig_lpref, used in chunks 41 and 44.
sig_MUZK, used in chunks 41 and 44.

sig_MUZM, used in chunks 41 and 44.
sig_PMKC, used in chunks 41 and 44.
sig_PMKK, used in chunks 41 and 44.
sig_R, used in chunks 41 and 44.
sig_STAR, used in chunks 41 and 44.
sig_XiL, used in chunks 41 and 44.
sigmah, used in chunks 14a, 24, 41, 45, and 47.
sigman, used in chunks 14a, 24, 41, 45, and 47.
theta_c, used in chunks 13, 23, 42, and 47.
theta_k, used in chunks 13, 23, 42, and 47.
theta_wc, used in chunks 14a, 24, 43a, and 47.
theta_wk, used in chunks 14a, 24, 43a, and 47.
tp2, used in chunks 16, 26, 41, 45, and 47.
UCSS, used in chunk 47.
UHCSS, used in chunks 19, 24, and 47.
UHKSS, used in chunks 19, 24, and 47.
UHSCSS, used in chunks 19, 24, and 47.
UHSKSS, used in chunks 19, 24, and 47.
UKSS, used in chunk 47.
unempSS, used in chunks 17, 19, 24, 42, 47, and 50.
unempSS_obs, used in chunks 17, 19, 27, 29, and 50.
USS, used in chunks 15a, 19, 23, 25b, 26, and 47.
WCSS, used in chunks 19, 23, 24, and 47.
WKSS, used in chunks 19, 23, 24, and 47.
xsi_HrC, used in chunks 14a, 24, 43a, and 47.
xsi_HrK, used in chunks 14a, 24, 43a, and 47.
xsi_NC, used in chunks 14a, 24, 43a, and 47.
xsi_NK, used in chunks 14a, 24, 43a, and 47.
ycbi, used in chunk 47.
ycbi_ykb, used in chunk 47.
YCSS, used in chunks 13–15, 19, 23–25, and 47.
ykb, used in chunk 47.
YKSS, used in chunks 13–15, 19, 23–25, and 47.
YYSS, used in chunk 47.

A.5.6 Common Estimated Params

41 ⟨*common estimated params* 41⟩≡ (32)

```
h               = 0.715162417869797;
r_inf           = 1.46344163969035;
r_y             = 0.263123294207851;
phi_pc          = 3.54471453295450;
phi_H           = 3.22894079106560;
phi_wc          = 5.49395755514723;
phi_ic          = 0.253308786976374;
phi_cd          = 0.470089385005009;
phi_ech         = 9.13986886546163;
gam_pc          = 0.314488926051065;
gam_wc          = -0.230018833252054;
sigman          = 39.4075260618789;
sigmah          = 21.8859803402692;
rho_R           = 0.833200065745674;
rho_XiL         = 0.263567746111198;
rho_lpref       = 0.979092048897712;
rho_B           = 0.895267027146152;
rho_STAR        = 0.909187927454138;
rho_EFFK        = 0.937829274540004;
rho_EFFECD      = -0.240286975088701;
rho_HG          = 0.582395471123139;
rho_EFFECH      = 0.877235725078934;
tp2             = 0.000307314910763576;
sig_HG          = 0.579315931803017;
sig_XiL         = 2.49313873916751;
sig_lpref       = 5.66476748114241;
sig_R           = 0.124100461010359;
sig_MUZK        = 0.936167718269030;
sig_MUZM        = 0.597390920898135;
sig_PMKC        = 0.451830653200989;
sig_PMKK        = 0.685376191952156;
sig_EFFECH      = 0.514704527091087;
sig_EFFECD      = 9.11199585973990;
sig_EFFK        = 0.402779878811407;
sig_B           = 0.295232712196573;
sig_STAR        = 0.104877885500673;
```

Uses gam_pc 37, gam_wc 37, h 37, phi_cd 37, phi_ech 37, phi_H 37, phi_ic 37, phi_pc 37, phi_wc 37, r_inf 37, r_y 37, rho_B 37, rho_EFFECD 37, rho_EFFECH 37, rho_EFFK 37, rho_HG 37, rho_lpref 37, rho_R 37, rho_STAR 37, rho_XiL 37, sig_B 37, sig_EFFECD 37, sig_EFFECH 37, sig_EFFK 37, sig_HG 37, sig_lpref 37, sig_MUZK 37, sig_MUZM 37, sig_PMKC 37, sig_PMKK 37, sig_R 37, sig_STAR 37, sig_XiL 37, sigmah 37, sigman 37, and tp2 37.

A.5.7 Common Calibrated params

42 ⟨*common calibrated params* 42⟩≡ (32)

```
r_dy = 0;
ONE = 1;
MUZKSS = 1.009250;
MUZMSS = 1.001000;
gam_ic = 1.0;
gam_icd = 1.0;
r_dinf = 0;
rpr = 0.965;
phi_u = 1;
rho_MUZK = 0;
rho_MUZM = 0;
pbeta = 0.99862;
delta_ = 0.03;
h_cd = 0.0;
h_ch = 0.0;
delta_cd = 0.055;
delta_ch = 0.0035;
alpha_ = 0.26;
theta_c = 7;
theta_k = 7;
unempSS = .06;
g_y = 0.0;
a_ks = 0.2;
s_AS = 0.2;
gam_h = 1;
gam_ech = 1;
icoef = 3;
betarl = .958;
```

Uses a_ks 37, alpha_ 37, betarl 37, delta_ 37, delta_cd 37, delta_ch 37, g_y 37, gam_ech 37, gam_h 37, gam_ic 37, gam_icd 37, h_cd 37, h_ch 37, icoef 37, MUZKSS 37, MUZMSS 37, ONE 37, pbeta 37, phi_u 37, r_dinf 37, r_dy 37, rho_MUZK 37, rho_MUZM 37, rpr 37, s_AS 37, theta_c 37, theta_k 37, and unempSS 37.

A.5.8 Common Free Params

43a ⟨*common free params* 43a⟩≡ (32)

```
//A_HC;
//A_HK;
//xsi_NC;
//xsi_HrC;
//xsi_NK;
//xsi_HrK;
//theta_wc;
//theta_wk;
//infkbar;
//infcbar;
//infwcbar;
//infwkbar;
//Pybar;
//Yybar;
//mu_yc;
//mu_yk;
//s_k;
//s_ecdc;
//eta_cnn;
//eta_cd;
//eta_ch;
//mu_;
```

Uses `A_HC` 37, `A_HK` 37, `eta_cd` 37, `eta_ch` 37, `eta_cnn` 37, `mu_` 37, `s_ecdc` 37, `s_k` 37, `theta_wc` 37, `theta_wk` 37, `xsi_HrC` 37, `xsi_HrK` 37, `xsi_NC` 37, and `xsi_NK` 37.

A.5.9 common Varobs

43b ⟨*common varobs* 43b⟩≡ (33)

```
varobs DIFFREALGDP_obs DIFFREALEC_obs DIFFREALEIK_obs DIFFREALECD_obs DIFFREALECH_obs DIFFREALW
```

Uses `AH_obs` 34, `DIFFREALEC_obs` 34, `DIFFREALECD_obs` 34, `DIFFREALECH_obs` 34, `DIFFREALEIK_obs` 34, `DIFFREALGDP_obs` 34, `DIFFREALW_obs` 34, `INFCNA_obs` 34, `INFCOR_obs` 34, `INFK_obs` 34, `R_obs` 34, `RT2_obs` 34, and `unemp_obs` 34.

A.5.10 Common Shocks

44 ⟨*common shocks* 44⟩≡ (33)

```
var eHG;
stderr sig_HG;
var eXiL;
stderr sig_XiL;
var eLpref;
stderr sig_lpref;
var eR;
stderr sig_R;
var eMUZK;
stderr sig_MUZK;
var eMUZM;
stderr sig_MUZM;
var ePMKC;
stderr sig_PMKC;
var ePMKK;
stderr sig_PMKK;
var eEFFECH;
stderr sig_EFFECH;
var eEFFECD;
stderr sig_EFFECD;
var eEFFK;
stderr sig_EFFK;
var eB;
stderr sig_B;
var eSTAR;
stderr sig_STAR;


var  DIFFREALGDP_obs;
stderr 0.3;
var  DIFFREALEC_obs;
stderr 0.1;
var  DIFFREALEIK_obs;
stderr 1.5;
var DIFFREALECD_obs;
stderr 1.5;
var DIFFREALECH_obs;
stderr 1.5;
var DIFFREALW_obs;
stderr 0.3;
var AH_obs;
stderr 0.3;
var INFCNA_obs;
```

```
stderr 0.5;
var INFCOR_obs;
stderr 0.05;
var INFK_obs;
stderr 0.2;
var RT2_obs;
stderr 0.1;
var unemp_obs;
stderr 4;
```

Uses AH_obs 34, DIFFREALEC_obs 34, DIFFREALECD_obs 34, DIFFREALECH_obs 34, DIFFREALEIK_obs 34, DIFFREALGDP_obs 34, DIFFREALW_obs 34, eB 36, eEFFECD 36, eEFFECH 36, eEFFK 36, eHG 36, eLpref 36, eMUZK 36, eMUZM 36, ePMKC 36, ePMKK 36, eR 36, eSTAR 36, eXiL 36, INFCNA_obs 34, INFCOR_obs 34, INFK_obs 34, RT2_obs 34, sig_B 37, sig_EFFECD 37, sig_EFFECH 37, sig_EFFK 37, sig_HG 37, sig_lpref 37, sig_MUZK 37, sig_MUZM 37, sig_PMKC 37, sig_PMKK 37, sig_R 37, sig_STAR 37, sig_XiL 37, and unemp_obs 34.

A.5.11 Common Steady Estimated params

45 ⟨*common steady estimated params* 45⟩≡ (33)

```
h              , .673     , -1     , 1    , uniform_pdf  ,,,-1      ,1;
r_inf          , 1.461    , -999   , 999  , normal_pdf   , 1.5000   , 0.062
r_y            , 0.214    , -999   , 999  , normal_pdf   , 0.125    , 0.125
phi_pc         , 3.126    , 0      , 999  , gamma_pdf    , 4.0000   , 4.000
phi_H          , 4.064    , 0      , 999  , gamma_pdf    , 4.0000   , 4.000
phi_wc         , 5.119    , 0      , 999  , gamma_pdf    , 4.0000   , 4.000
phi_ic         , .325     , 0      , 999  , gamma_pdf    , 4.0000   , 4.000
phi_cd         , .651     , 0      , 999  , gamma_pdf    , 4.0000   , 4.000
phi_ech        , 10.948   , 0      , 999  , gamma_pdf    , 4.0000   , 4.000
gam_pc         , 0.386    , -999   , 999  , normal_pdf   , 0.000    , 0.250
gam_wc         , 0.213    , -999   , 999  , normal_pdf   , 0.000    , 0.250
sigman         , 1.25     , 0      , 999  , gamma_pdf    , 1.25     , 12.5^
sigmah         , 10       , 0      , 999  , gamma_pdf    , 10       , 100^.
rho_R          , 0.654    , -1     , 1    , normal_pdf   , 0.5      , 0.25;
rho_XiL        , 0.654    , -1     , 1    , normal_pdf   , 0.5      , 0.25;
rho_lpref      , 0.954    , -1     , 1    , normal_pdf   , 0.5      , 0.25;
rho_B          , 0.825    , -1     , 1    , normal_pdf   , 0        , 0.5;
rho_STAR       , 0.825    , -1     , 1    , normal_pdf   , 0        , 0.5;
rho_EFFK       , 0.850    , -1     , 1    , normal_pdf   , 0        , 0.5;
rho_EFFECD     , .230     , -1     , 1    , normal_pdf   , 0        , 0.5;
rho_HG         , 0.596    , 0      , 1    , beta_pdf     , 0.5      , 0.015
rho_EFFECH     , 0.844    , -1     , 1    , normal_pdf   , 0        , 0.5;
tp2            , 0.001    , -999   , 999  , normal_pdf   , 0.0      , 0.000
```

Uses gam_pc 37, gam_wc 37, h 37, phi_cd 37, phi_ech 37, phi_H 37, phi_ic 37, phi_pc 37, phi_wc 37, r_inf 37, r_y 37, rho_B 37, rho_EFFECD 37, rho_EFFECH 37, rho_EFFK 37, rho_HG 37, rho_lpref 37, rho_R 37, rho_STAR 37, rho_XiL 37, sigmah 37, sigman 37, and tp2 37.

A.5.12 Common Stderr

46 ⟨*common stderr* 46⟩≡ (33)

```
stderr eHG      , .745       , 0.0001     , 999  , inv_gamma_pdf , 1.772454
stderr eXiL     , 3.621      , 0.0001     , 999  , inv_gamma_pdf , 1.772454
stderr eLpref   , 1.621      , 0.0001     , 999  , inv_gamma_pdf , 1.772454
stderr eR       , 0.165      , 0.0001     , 999  , inv_gamma_pdf , 0.354491
stderr eMUZK    , .834       , 0.0001     , 999  , inv_gamma_pdf , 0.443113
stderr eMUZM    , .484       , 0.0001     , 999  , inv_gamma_pdf , 0.443113
stderr ePMKC    , .391       , 0.0001     , 999  , inv_gamma_pdf , 0.354491
stderr ePMKK    , .552       , 0.0001     , 999  , inv_gamma_pdf , 0.354491
stderr eEFFECH  , .526       , 0.0001     , 999  , inv_gamma_pdf , 1.772454
stderr eEFFECD  , 13.349     , 0.0001     , 999  , inv_gamma_pdf , 1.772454
stderr eEFFK    , .499       , 0.0001     , 999  , inv_gamma_pdf , 1.772454
stderr eB       , 0.5        , 0.0001     , 999  , inv_gamma_pdf , 1.772454
stderr eSTAR    , 0.05       , 0.0001     , 999  , inv_gamma_pdf , 0.354491
```

Uses eB 36, eEFFECD 36, eEFFECH 36, eEFFK 36, eHG 36, eLpref 36, eMUZK 36, eMUZM 36, ePMKC 36, ePMKK 36, eR 36, eSTAR 36, and eXiL 36.

A.6 Common Steady State Routines

A.6.1 Common Steady State Values

47 ⟨*common steady state values* 47⟩≡ (18 28)

```
beta_0 = pbeta;
beta_2 = pbeta*rpr; % s.s. funds rate premium
beta_ = beta_2;
MUZCSS=1;
ONE=1;
USS=1;
MUKSS=MUZKSS*MUZMSS;
MUCSS=MUZKSS^alpha_*MUZMSS;
MUKSShabit=MUKSS;
MUCSShabit=MUCSS;
PKBSS=theta_k/(theta_k-1)*(theta_c-1)/theta_c;
PYSS=1;
MCCSS=(theta_c-1)/theta_c;
MCKSS=(theta_k-1)/theta_k;
RKSS=MUKSS/beta_2-(1-delta_);
RCSS=MUKSS/beta_2-(1-delta_);
RCHSS=MUCSS/beta_2-(1-delta_ch); % Housing sector
RCDSS=MUKSS/beta_2-(1-delta_cd); % Durable sector
USS=1;
mu_=RCSS;
AA=alpha_/RKSS*MCKSS;
DD = 0.135;
RR = 0.075;
eta_cnn=1;
eta_cd_eta_cnn=DD/((MUKSShabit-beta_2*h_cd)/(1-beta_2*h/MUCSShabit)*(1-h/MUCSShabit)/(1-h_cd/MU
eta_ch_eta_cnn=RR/((MUCSShabit-beta_2*h_ch)/(1-beta_2*h/MUCSShabit)*(1-h/MUCSShabit)/(1-h_ch/MU
eta_ch=eta_ch_eta_cnn;
eta_cd=eta_cd_eta_cnn;
DD=eta_cd_eta_cnn*(MUKSShabit-beta_2*h_cd)/(1-beta_2*h/MUCSShabit)*(1-h/MUCSShabit)/(1-h_cd/MUK
RR=eta_ch_eta_cnn*(MUCSShabit-beta_2*h_ch)/(1-beta_2*h/MUCSShabit)*(1-h/MUCSShabit)/(1-h_ch/MUC
Rnr=(1-(1-delta_)/MUKSS)*AA*MUKSS;
ycbi_ykb=((1-s_AS)-Rnr)/((DD*(1-s_AS)/(1+RR))+Rnr);
hc_hk=ycbi_ykb*(RCSS*MCKSS/(RKSS*MCCSS))^(alpha_/(1-alpha_));
HSS=0.25;
AHSS=HSS;
HKSS=HSS/(1+hc_hk);
HCSS=HSS-HKSS;
HrCSS=1/3;
HrKSS=1/3;
empCSS=HCSS/HrCSS;
empKSS=HKSS/HrKSS;
```

```
ycbi=HCSS*(AA)^(alpha_/(1-alpha_));
ykb=HKSS*(AA)^(alpha_/(1-alpha_));
YCSS=ycbi;
YKSS=ykb;
KCSS=AA*ycbi*MUKSS;
KKSS=AA*ykb*MUKSS;
ECHSS=RR/(1+RR)*ycbi*(1-s_AS);
ECSS=1/(1+RR)*ycbi*(1-s_AS);
ECDSS=DD*PKBSS*ECSS;
EIKSS=(1-(1-delta_)/MUKSS)*(KCSS+KKSS);
KCDSS=ECDSS/(1-(1-delta_cd)/MUKSS);
KCHSS=ECHSS/(1-(1-delta_ch)/MUCSS);
YYSS=(YCSS+YKSS*PKBSS)/PYSS;
s_k_ecd=ECDSS/YKSS;
s_c_ech=ECHSS/YCSS;
s_k_eik=EIKSS/YKSS;
s_yc = (YCSS/YYSS);
s_ecdc=PKBSS*ECDSS/(ECSS+PKBSS*ECDSS+(MUCSS/beta_2-1+delta_ch)*KCHSS);
INFCNASS=exp(.02/4);
INFCSS = INFCNASS*((MUZCSS/MUZKSS)^(1-alpha_))^(-s_ecdc);
INFCORSS=INFCNASS;
INFKSS=INFCSS*(MUZCSS/MUZKSS)^(1-alpha_);
INFWCSS=INFCSS*MUZKSS^alpha_*MUZMSS;
INFWKSS=INFWCSS;
RSS=INFCSS/beta_0*MUCSS;
RT2SS=exp(tp2)*RSS;
INFC10SS = INFCNASS;
IMPHSSS = RCHSS*KCHSS;
s_k=PKBSS*YKSS/YYSS;
INFGDPSS=INFCSS^(YCSS/YYSS)*INFKSS^(YKSS*PKBSS/(YYSS));
LSS=eta_cnn/(ECSS*(1-h/MUCSShabit))-eta_cnn*beta_2*h/(ECSS*(MUCSShabit-h));
WCSS=MCCSS*(1-alpha_)*YCSS/HCSS;
WKSS=MCKSS*(1-alpha_)*YKSS/HKSS;
xsiN_xsiH_C = ((HrCSS/empCSS)^(1+sigmah))/(1+1/sigmah);
xsiN_xsiH_K = ((HrKSS/empKSS)^(1+sigmah))/(1+1/sigmah);
gC = (1/(1+sigman) + 1/sigmah)*(xsiN_xsiH_C*(1+sigmah)/sigmah)^(-(1+sigman)/(1+sigmar
markup_xsiN_C = (HCSS^((1+sigmah)*(1+sigman)/(1+sigmah+sigman)-1))*gC/(LSS*WCSS);
gK = (1/(1+sigman) + 1/sigmah)*(xsiN_xsiH_K*(1+sigmah)/sigmah)^(-(1+sigman)/(1+sigmar
markup_xsiN_K = (HKSS^((1+sigmah)*(1+sigman)/(1+sigmah+sigman)-1))*gK/(LSS*WKSS);
markup_w = (1-unempSS)^((1+sigmah+sigman)/(1+sigmah) - 1 - sigman);
theta_wc = markup_w/(markup_w -1); theta_wk = theta_wc;
A_HC=LSS*(theta_wc-1)/theta_wc*WCSS/(((1+sigman)/(1+sigman/(1+sigmah)))*HCSS^(-1+(1+s
A_HK=LSS*(theta_wk-1)/theta_wk*WKSS/(((1+sigman)/(1+sigman/(1+sigmah)))*HKSS^(-1+(1+s
xsi_NC=A_HC/((1/(1+sigman)+1/sigmah)*(HCSS^sigman/HrCSS^(1+sigman+sigmah))^((1+sigmar
xsi_NK=A_HK/((1/(1+sigman)+1/sigmah)*(HKSS^sigman/HrKSS^(1+sigman+sigmah))^((1+sigmar
xsi_HrC=xsi_NC*(1+sigmah)/sigmah*(HCSS^sigman/HrCSS^(1+sigman+sigmah));
```

```
xsi_HrK=xsi_NK*(1+sigmah)/sigmah*(HKSS^sigman/HrKSS^(1+sigman+sigmah));
UHCSS=A_HC*((1+sigman)/(1+sigman/(1+sigmah)))*HCSS^(-1+(1+sigman)/(1+sigman/(1+sigmah)))/LSS;
UHKSS=A_HK*((1+sigman)/(1+sigman/(1+sigmah)))*HKSS^(-1+(1+sigman)/(1+sigman/(1+sigmah)))/LSS;
HSCSS=(WCSS*LSS/(A_HC*((1+sigman)/(1+sigman/(1+sigmah)))))^(1/(-1+(1+sigman)/(1+sigman/(1+sigma
HSKSS=(WKSS*LSS/(A_HK*((1+sigman)/(1+sigman/(1+sigmah)))))^(1/(-1+(1+sigman)/(1+sigman/(1+sigma
empSCSS=((1+sigmah)/sigmah*xsi_NC/xsi_HrC)^(-1/(1+sigmah+sigman))*HSCSS^(1/(1+sigman/(1+sigmah)
empSKSS=((1+sigmah)/sigmah*xsi_NK/xsi_HrK)^(-1/(1+sigmah+sigman))*HSKSS^(1/(1+sigman/(1+sigmah)
HrSCSS=HSCSS/empSCSS;
HrSKSS=HSKSS/empSKSS;
UHSCSS=A_HC*((1+sigman)/(1+sigman/(1+sigmah)))*HSCSS^(-1+(1+sigman)/(1+sigman/(1+sigmah)))/LSS;
UHSKSS=A_HK*((1+sigman)/(1+sigman/(1+sigmah)))*HSKSS^(-1+(1+sigman)/(1+sigman/(1+sigmah)))/LSS;
unempSS=(empSCSS+empSKSS-(empCSS+empKSS))/(empSCSS+empSKSS);
QKSS=1;
QCDSS=1;
QCHSS=1;
UCSS=1;
UKSS=1;
XiBSS=1;
XiDSS=1;
XiHSS=1;
RL1SS=RSS;
RL2SS=RSS;
RL3SS=RSS;
RL4SS=RSS;
RL5SS=RSS;
RL6SS=RSS;
RL7SS=RSS;
DIFFREALECSS =exp( log(MUCSS));
DIFFREALEIKSS =exp( log(MUKSS));
DIFFREALECDSS =exp( log(MUKSS));
DIFFREALECHSS =exp( log(MUCSS));
DIFFREALWSS =exp( log(MUCSS) );
DIFFREALGDPSS =exp( (1-s_k)*log(MUCSS)+(s_k)*log(MUKSS));
```

Uses A_HC 37, A_HK 37, AA 37, AHSS 37, alpha_ 37, beta_ 37, beta_0 37, beta_2 37, DD 37, delta_ 37, delta_cd 37, delta_ch 37, DIFFREALECDSS 37, DIFFREALECHSS 37, DIFFREALECSS 37, DIFFREALEIKSS 37, DIFFREALGDPSS 37, DIFFREALWSS 37, ECDSS 37, ECHSS 37, ECSS 37, EIKSS 37, empCSS 37, empKSS 37, empSCSS 37, empSKSS 37, eta_cd 37, eta_cd_eta_cnn 37, eta_ch 37, eta_ch_eta_cnn 37, eta_cnn 37, h 37, h_cd 37, h_ch 37, hc_hk 37, HCSS 37, HKSS 37, HrCSS 37, HrKSS 37, HrSCSS 37, HrSKSS 37, HSCSS 37, HSKSS 37, HSS 37, IMPHSSS 37, INFC10SS 37, INFCNASS 37, INFCORSS 37, INFCSS 37, INFGDPSS 37, INFKSS 37, INFWCSS 37, INFWKSS 37, KCDSS 37, KCHSS 37, KCSS 37, KKSS 37, LSS 37, MCCSS 37, MCKSS 37, mu_ 37, MUCSS 37, MUCSShabit 37, MUKSS 37, MUKSShabit 37, MUZCSS 37, MUZKSS 37, MUZMSS 37, ONE 37, pbeta 37, PKBSS 37, PYSS 37, QCDSS 37, QCHSS 37, QKSS 37, RCDSS 37, RCHSS 37, RCSS 37, RKSS 37, RL1SS 37, RL2SS 37, RL3SS 37, RL4SS 37, RL5SS 37, RL6SS 37, RL7SS 37, Rnr 37, rpr 37, RR 37, RSS 37, RT2SS 37, s_AS 37, s_c_ech 37, s_ecdc 37, s_k 37, s_k_ecd 37, s_k_eik 37, s_yc 37, sigmah 37, sigman 37, theta_c 37, theta_k 37, theta_wc 37, theta_wk 37, tp2 37, UCSS 37, UHCSS 37, UHKSS 37, UHSCSS 37, UHSKSS 37, UKSS 37, unempSS 37, USS 37, WCSS 37, WKSS 37, xsi_HrC 37, xsi_HrK 37, xsi_NC 37, xsi_NK 37, ycbi 37, ycbi_ykb 37, YCSS 37, ykb 37, YKSS 37, and YYSS 37.

A.6.2 Common Steady State Trends

50 ⟨*common steady state trends* 50⟩≡ (18 28)

```
DIFFREALGDPSS_obs=(1-s_k)*log(MUCSS)*100+(s_k)*log(MUKSS)*100;
DIFFREALECSS_obs=log(MUCSS)*100;
DIFFREALEIKSS_obs=log(MUKSS)*100;
DIFFREALECDSS_obs=log(MUKSS)*100;
DIFFREALECHSS_obs=log(MUCSS)*100;
DIFFREALWSS_obs=log(MUCSS)*100;
INFCNASS_obs=(1-s_ecdc)*log(INFCSS)*100+s_ecdc*log(INFKSS)*100;
INFCORSS_obs=(1-s_ecdc)*log(INFCSS)*100+s_ecdc*log(INFKSS)*100;
INFKSS_obs=log(INFCSS)*100-log(MUKSS)*100+log(MUCSS)*100;
RSS_obs=log(RSS)*100;
RT2SS_obs=log(RT2SS)*100;
unempSS_obs=100*log(unempSS);
```

Uses DIFFREALECDSS_obs 37, DIFFREALECHSS_obs 37, DIFFREALECSS_obs 37, DIFFREALEIKSS_obs 37, DIFFREALGDPSS_obs 37, DIFFREALWSS_obs 37, INFCNASS_obs 37, INFCORSS_obs 37, INFCSS 37, INFKSS 37, INFKSS_obs 37, MUCSS 37, MUKSS 37, RSS 37, RSS_obs 37, RT2SS 37, RT2SS_obs 37, s_ecdc 37, s_k 37, unempSS 37, and unempSS_obs 37.

A.7 readme.txt

51 ⟨*srcedo/readme.txt* 51⟩≡

```
How to run the model:
====================

In Matlab/Octave:

1) Download Dynare Version 4 from the Dynare website: http://www.dynare.org/
2) Download the EDO files in a folder you choose.
3) Start Matlab/Octave and change the current directory to the folder in step 2.
4) Link in Matlab/Octave the Dynare folder in the menu under file/Set Path (or use the command
   "addpath path/to/dynare").
5) Run the command "dynare linearized" or "dynare Dynare_edo" from the Matlab/Octave command li


Content of the EDO folder:
========================

Dynare_edo.mod: Dynare model file containing the latest estimated parameters and nonlinear mode

Dynare_edo_steadystate.mod: Dynare steady-state file computes the steady state of the model var

linearized.mod: Dynare model file containing the latest estimated parameters and nonlinear mode

linearized_steadystate.mod: Dynare steady-state file computes the steady state of the model var

readme.txt: The file you are currently reading.
```

This code is written to file `srcedo/readme.txt`.

Appendix B

Notes, Bibliography and Indexes

B.1 Chunks

⟨*common calibrated params* 42⟩
⟨*common estimated params* 41⟩
⟨*common free params* 43a⟩
⟨*common parameters* 37⟩
⟨*common setup* 32⟩
⟨*common shocks* 44⟩
⟨*common stderr* 46⟩
⟨*common steady estimated params* 45⟩
⟨*common steady state trends* 50⟩
⟨*common steady state values* 47⟩
⟨*common stoch sim* 33⟩
⟨*common var* 34⟩
⟨*common varexo* 36⟩
⟨*common varobs* 43b⟩
⟨*edo model* 12⟩
⟨*edo model durables* 15b⟩
⟨*edo model hours* 15a⟩
⟨*edo model housing* 16⟩
⟨*edo model identities* 14b⟩
⟨*edo model labor* 14a⟩
⟨*edo model measurement* 17⟩
⟨*edo model prelim* 13⟩
⟨*edo steady state result return* 19⟩
⟨*linearized model* 22⟩
⟨*linearized model durables* 25c⟩
⟨*linearized model hours* 25b⟩

⟨*linearized model housing* 26⟩
⟨*linearized model identities* 25a⟩
⟨*linearized model labor* 24⟩
⟨*linearized model measurement* 27⟩
⟨*linearized model prelim* 23⟩
⟨*linearized steady state result return* 29⟩
⟨*srcedo/Dynare.edo.mod* 11⟩
⟨*srcedo/Dynare.edo.steadystate.m* 18⟩
⟨*srcedo/linearized.mod* 21⟩
⟨*srcedo/linearized.steadystate.m* 28⟩
⟨*srcedo/readme.txt* 51⟩

B.2 Index

`A_HC`: 14a, 24, <u>37</u>, 43a, 47
`A_HK`: 14a, 24, <u>37</u>, 43a, 47
`a_ks`: <u>37</u>, 42
`AA`: <u>37</u>, 47
`AH`: 15a, 17, 25b, 27, <u>34</u>
`AH_obs`: 17, 27, <u>34</u>, 43b, 44
`AHSS`: 14b, 17, 19, 25a, 25b, <u>37</u>, 47
`alpha_`: 13, 15a, 16, 23, 25b, 26, <u>37</u>, 42, 47
`beta_`: 13, 14a, 15b, 16, 19, 23, 24, 25c, 26, <u>37</u>, 47
`beta_0`: <u>37</u>, 47
`beta_2`: <u>37</u>, 47
`betarl`: 15a, 25b, <u>37</u>, 42
`betas`: 13, 16, 23, 26, <u>34</u>
`DD`: <u>37</u>, 47
`delta_`: 13, 23, <u>37</u>, 42, 47
`delta_cd`: 15b, 25c, <u>37</u>, 42, 47
`delta_ch`: 16, 26, <u>37</u>, 42, 47
`DIFFNORMGDP`: 14a, 24, <u>34</u>
`DIFFREALEC`: 14a, 17, 24, 27, <u>34</u>
`DIFFREALEC_obs`: 17, 27, <u>34</u>, 43b, 44
`DIFFREALECD`: 16, 17, 26, 27, <u>34</u>
`DIFFREALECD_obs`: 17, 27, <u>34</u>, 43b, 44
`DIFFREALECDSS`: 17, 19, 26, <u>37</u>, 47
`DIFFREALECDSS_obs`: 17, 19, 27, 29, <u>37</u>, 50
`DIFFREALECH`: 16, 17, 26, 27, <u>34</u>
`DIFFREALECH_obs`: 17, 27, <u>34</u>, 43b, 44
`DIFFREALECHSS`: 17, 19, 26, <u>37</u>, 47
`DIFFREALECHSS_obs`: 17, 19, 27, 29, <u>37</u>, 50
`DIFFREALECSS`: 17, 19, 24, <u>37</u>, 47
`DIFFREALECSS_obs`: 17, 19, 27, 29, <u>37</u>, 50
`DIFFREALEIK`: 14a, 17, 24, 27, <u>34</u>

DIFFREALEIK_obs: 17, 27, 34, 43b, 44
DIFFREALEIKSS: 17, 19, 24, 37, 47
DIFFREALEIKSS_obs: 17, 19, 27, 29, 37, 50
DIFFREALGDP: 14a, 15a, 17, 24, 25b, 27, 34
DIFFREALGDP_obs: 17, 27, 34, 43b, 44
DIFFREALGDPSS: 17, 19, 24, 25b, 37, 47
DIFFREALGDPSS_obs: 17, 19, 27, 29, 37, 50
DIFFREALW: 14b, 17, 25a, 27, 34
DIFFREALW_obs: 17, 27, 34, 43b, 44
DIFFREALWSS: 17, 19, 25a, 37, 47
DIFFREALWSS_obs: 17, 19, 27, 29, 37, 50
eB: 16, 26, 36, 44, 46
EC: 13, 14a, 23, 24, 34
ECD: 14a, 15b, 16, 24, 25c, 26, 34
ECDSS: 15b, 19, 24, 25c, 26, 37, 47
ECH: 14a, 16, 24, 26, 34
ECHSS: 16, 19, 24, 26, 37, 47
ECSS: 19, 23, 24, 37, 47
eEFFECD: 16, 26, 36, 44, 46
eEFFECH: 16, 26, 36, 44, 46
eEFFK: 16, 26, 36, 44, 46
EFFECD: 15b, 16, 25c, 26, 34
EFFECH: 16, 26, 34
EFFK: 13, 14a, 16, 23, 24, 26, 34
eHG: 16, 26, 36, 44, 46
EIK: 13, 14a, 23, 24, 34
EIKSS: 14a, 19, 23, 24, 37, 47
eLpref: 16, 26, 36, 44, 46
empC: 14a, 24, 34
empCSS: 19, 24, 37, 47
empK: 14a, 24, 34
empKSS: 19, 24, 37, 47
empSC: 14a, 24, 34
empSCSS: 19, 24, 37, 47
empSK: 14a, 24, 34
empSKSS: 19, 24, 37, 47
eMUZK: 16, 26, 36, 44, 46
eMUZM: 16, 26, 36, 44, 46
ePMKC: 13, 23, 36, 44, 46
ePMKK: 13, 23, 36, 44, 46
eR: 13, 23, 36, 44, 46
eSTAR: 16, 26, 36, 44, 46
eta_cd: 15b, 25c, 37, 43a, 47
eta_cd_eta_cnn: 37, 47
eta_ch: 16, 26, 37, 43a, 47
eta_ch_eta_cnn: 37, 47

eta_cnn: 13, 23, 37, 43a, 47
eXiL: 14a, 16, 24, 26, 36, 44, 46
g_y: 37, 42
gam_ech: 16, 26, 37, 42
gam_h: 14a, 24, 37, 42
gam_ic: 14a, 24, 37, 42
gam_icd: 15b, 25c, 37, 42
gam_pc: 13, 23, 37, 41, 45
gam_wc: 14a, 24, 37, 41, 45
GAP: 15a, 25b, 34
h: 13, 23, 37, 41, 45, 47
h_cd: 15b, 25c, 37, 42, 47
h_ch: 16, 26, 37, 42, 47
HC: 13, 14a, 15a, 23, 24, 25b, 34
hc_hk: 37, 47
HCSS: 14a, 14b, 15a, 19, 23, 24, 25a, 25b, 37, 47
HG: 14a, 16, 24, 26, 34
HK: 13, 14a, 15a, 23, 24, 25b, 34
HKSS: 14a, 14b, 15a, 19, 23, 24, 25a, 25b, 37, 47
HrC: 14a, 24, 34
HrCSS: 19, 24, 37, 47
HrK: 14a, 24, 34
HrKSS: 19, 24, 37, 47
HrSC: 14a, 24, 34
HrSCSS: 19, 24, 37, 47
HrSK: 14a, 24, 34
HrSKSS: 19, 24, 37, 47
HSC: 14a, 24, 34
HSCSS: 14a, 19, 24, 37, 47
HSK: 14a, 24, 34
HSKSS: 14a, 19, 24, 37, 47
HSS: 37, 47
icoef: 37, 42
IMPHSSS: 37, 47
INFC: 13, 14a, 14b, 15a, 23, 24, 25a, 25b, 34
INFC10: 15a, 25b, 34
INFC10SS: 19, 25b, 37, 47
INFCNA: 13, 15a, 17, 23, 25b, 27, 34
INFCNA_obs: 17, 27, 34, 43b, 44
INFCNASS: 13, 17, 19, 23, 25b, 37, 47
INFCNASS_obs: 17, 19, 27, 29, 37, 50
INFCOR: 15a, 17, 25b, 27, 34
INFCOR_obs: 17, 27, 34, 43b, 44
INFCORSS: 17, 19, 25b, 37, 47
INFCORSS_obs: 17, 19, 27, 29, 37, 50
INFCSS: 13, 19, 23, 24, 25a, 25b, 37, 47, 50

INFGDP: 15a, 25b, 34
INFGDPSS: 19, 25b, 37, 47
INFK: 13, 14a, 15a, 17, 23, 24, 25b, 27, 34
INFK_obs: 17, 27, 34, 43b, 44
INFKSS: 13, 17, 19, 23, 24, 25b, 37, 47, 50
INFKSS_obs: 17, 19, 27, 29, 37, 50
INFWC: 14a, 14b, 24, 25a, 34
INFWCSS: 14a, 19, 24, 25a, 37, 47
INFWK: 14a, 14b, 24, 25a, 34
INFWKSS: 14a, 19, 24, 25a, 37, 47
jacobian_flag: 33
KC: 13, 14a, 23, 24, 34
KCDSS: 19, 25c, 26, 37, 47
KCH: 16, 26, 34
KCHSS: 19, 26, 37, 47
KCSS: 13, 19, 23, 24, 37, 47
KD: 15b, 16, 25c, 26, 34
KK: 13, 14a, 23, 24, 34
KKSS: 13, 19, 23, 24, 37, 47
L: 12, 13, 14a, 15b, 16, 22, 23, 24, 25c, 26, 34
LAGKCH: 16, 26, 34
LAGKD: 15b, 16, 25c, 26, 34
Lpref: 14a, 16, 24, 26, 34
LSS: 19, 22, 23, 24, 25c, 26, 37, 47
MCC: 13, 23, 34
MCCSS: 19, 23, 37, 47
MCK: 13, 23, 34
MCKSS: 19, 23, 37, 47
mu_: 13, 16, 23, 26, 37, 43a, 47
MUC: 13, 14a, 15a, 16, 23, 24, 25b, 26, 34
MUCSS: 19, 23, 24, 25b, 26, 37, 47, 50
MUCSShabit: 37, 47
MUK: 13, 14a, 15b, 16, 23, 24, 25c, 26, 34
MUKSS: 19, 23, 24, 25c, 26, 37, 47, 50
MUKSShabit: 37, 47
MUZCSS: 37, 47
MUZK: 16, 26, 34
MUZKSS: 16, 19, 26, 37, 42, 47
MUZM: 16, 26, 34
MUZMSS: 16, 19, 26, 37, 42, 47
nonlin: 33
NORMINFGDP: 14a, 24, 34
ONE: 19, 23, 24, 25b, 25c, 26, 37, 42, 47
options_: 33
order: 33
pbeta: 37, 42, 47

PFGAP: 13, 15a, 23, 25b, 34
phi_cd: 15b, 25c, 37, 41, 45
phi_ech: 16, 26, 37, 41, 45
phi_H: 14a, 24, 37, 41, 45
phi_ic: 14a, 24, 37, 41, 45
phi_pc: 13, 23, 37, 41, 45
phi_u: 13, 16, 23, 26, 37, 42
phi_wc: 14a, 24, 37, 41, 45
PKB: 13, 14a, 15a, 15b, 23, 24, 25b, 25c, 34
PKBSS: 19, 23, 24, 25b, 25c, 37, 47
PYSS: 37, 47
QCD: 15b, 25c, 34
QCDSS: 19, 25c, 37, 47
QCH: 16, 26, 34
QCHSS: 19, 26, 37, 47
QK: 13, 14a, 16, 23, 24, 26, 34
QKSS: 19, 23, 24, 26, 37, 47
R: 13, 16, 17, 23, 26, 27, 34
r_dinf: 13, 23, 37, 42
r_dy: 37, 42
r_inf: 13, 23, 37, 41, 45
R_obs: 17, 27, 34, 43b
r_y: 13, 23, 37, 41, 45
RC: 13, 16, 23, 26, 34
RCD: 15b, 25c, 34
RCDSS: 19, 25c, 37, 47
RCH: 16, 26, 34
RCHSS: 19, 26, 37, 47
RCSS: 19, 23, 26, 37, 47
rho_B: 16, 26, 37, 41, 45
rho_EFFECD: 16, 26, 37, 41, 45
rho_EFFECH: 16, 26, 37, 41, 45
rho_EFFK: 16, 26, 37, 41, 45
rho_HG: 16, 26, 37, 41, 45
rho_lpref: 16, 26, 37, 41, 45
rho_MUZK: 37, 42
rho_MUZM: 37, 42
rho_R: 13, 23, 37, 41, 45
rho_STAR: 16, 26, 37, 41, 45
rho_XiL: 16, 26, 37, 41, 45
RK: 13, 16, 23, 26, 34
RKSS: 19, 23, 26, 37, 47
RL1: 16, 26, 34
RL1SS: 19, 26, 37, 47
RL2: 16, 26, 34
RL2SS: 19, 26, 37, 47

RL3: 16, 26, 34
RL3SS: 19, 26, 37, 47
RL4: 16, 26, 34
RL4SS: 19, 26, 37, 47
RL5: 16, 26, 34
RL5SS: 19, 26, 37, 47
RL6: 16, 26, 34
RL6SS: 19, 26, 37, 47
RL7: 16, 26, 34
RL7SS: 19, 26, 37, 47
Rnr: 37, 47
rpr: 13, 23, 37, 42, 47
RR: 37, 47
RSS: 13, 17, 19, 23, 26, 37, 47, 50
RSS_obs: 17, 19, 27, 29, 37, 50
RT2: 16, 17, 26, 27, 34
RT2_obs: 17, 27, 34, 43b, 44
RT2SS: 17, 19, 26, 37, 47, 50
RT2SS_obs: 17, 19, 27, 29, 37, 50
s_AS: 37, 42, 47
s_c_ech: 37, 47
s_ecdc: 15a, 25b, 37, 43a, 47, 50
s_k: 14a, 15a, 24, 25b, 37, 43a, 47, 50
s_k_ecd: 37, 47
s_k_eik: 37, 47
s_yc: 37, 47
sig_B: 37, 41, 44
sig_EFFECD: 37, 41, 44
sig_EFFECH: 37, 41, 44
sig_EFFK: 37, 41, 44
sig_HG: 37, 41, 44
sig_lpref: 37, 41, 44
sig_MUZK: 37, 41, 44
sig_MUZM: 37, 41, 44
sig_PMKC: 37, 41, 44
sig_PMKK: 37, 41, 44
sig_R: 37, 41, 44
sig_STAR: 37, 41, 44
sig_XiL: 37, 41, 44
sigmah: 14a, 24, 37, 41, 45, 47
sigman: 14a, 24, 37, 41, 45, 47
STAR: 16, 26, 34
stoch_simul: 33
theta_c: 13, 23, 37, 42, 47
theta_k: 13, 23, 37, 42, 47
theta_wc: 14a, 24, 37, 43a, 47

theta_wk: 14a, 24, 37, 43a, 47
tp2: 16, 26, 37, 41, 45, 47
UC: 13, 15a, 16, 23, 25b, 26, 34
UCSS: 37, 47
UHC: 14a, 24, 34
UHCSS: 19, 24, 37, 47
UHK: 14a, 24, 34
UHKSS: 19, 24, 37, 47
UHSC: 14a, 24, 34
UHSCSS: 19, 24, 37, 47
UHSK: 14a, 24, 34
UHSKSS: 19, 24, 37, 47
UK: 13, 15a, 16, 23, 25b, 26, 34
UKSS: 37, 47
unemp: 14a, 17, 24, 27, 34
unemp_obs: 17, 27, 34, 43b, 44
unempSS: 17, 19, 24, 37, 42, 47, 50
unempSS_obs: 17, 19, 27, 29, 37, 50
unlinearized_edo_steadystate: 18
USS: 15a, 19, 23, 25b, 26, 37, 47
WC: 13, 14a, 23, 24, 34
WCSS: 19, 23, 24, 37, 47
WK: 13, 14a, 23, 24, 34
WKSS: 19, 23, 24, 37, 47
XiL: 14a, 16, 24, 26, 34
xsi_HrC: 14a, 24, 37, 43a, 47
xsi_HrK: 14a, 24, 37, 43a, 47
xsi_NC: 14a, 24, 37, 43a, 47
xsi_NK: 14a, 24, 37, 43a, 47
YC: 13, 14a, 15a, 23, 24, 25b, 34
ycbi: 37, 47
ycbi_ykb: 37, 47
YCSS: 13, 14a, 15a, 19, 23, 24, 25b, 37, 47
YK: 13, 14a, 15a, 23, 24, 25b, 34
ykb: 37, 47
YKSS: 13, 14a, 15a, 19, 23, 24, 25b, 37, 47
YYSS: 37, 47