Estimated Dynamic Optimization (EDO) Model

Gary Young (editor)

June 26, 2016

Contents

1 EDO model packages **5**

Appendices **7**

A Original Files **9**
A.1 Dynare_edo_mod . . . 9
A.1.1 var . . . 11
A.1.2 varexo . . . 13
A.1.3 parameters . . . 14
A.1.4 est params . . . 18
A.1.5 calib params . . . 19
A.1.6 free params . . . 20
A.1.7 Model . . . 21
A.1.8 prelim . . . 22
A.1.9 labor . . . 23
A.1.10 identities . . . 24
A.1.11 hours . . . 24
A.1.12 durables . . . 24
A.1.13 housing . . . 25
A.1.14 measure . . . 26
A.1.15 varobs . . . 26
A.1.16 shocks . . . 27
A.1.17 est params . . . 28
A.1.18 stderr . . . 29
A.2 Dynare_edo_steadystate.m . . . 30
A.2.1 values . . . 31
A.2.2 trends . . . 34
A.2.3 result return . . . 35
A.3 linearized.mod . . . 38
A.3.1 var . . . 40
A.3.2 varexo . . . 42
A.3.3 parameters . . . 43
A.3.4 est parms . . . 47
A.3.5 calib params . . . 48

A.3.6 free params 49
A.3.7 calib ME 50
A.3.8 model block 51
A.3.9 labor 52
A.3.10 identities 53
A.3.11 hours 53
A.3.12 data itentities 53
A.3.13 housing 54
A.3.14 measure 55
A.3.15 varobs 55
A.3.16 shocks 56
A.3.17 est params 58
A.3.18 stderr 59
A.4 linearized_steadystate.m 60
A.4.1 values 61
A.4.2 trends 64
A.4.3 result return 65
A.5 readme.txt 68

B Notes, Bibliography and Indexes **69**
B.1 Chunks 69
B.2 Index 70

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".

Appendices

Appendix A

Original Files

A.1 Dynare_edo_mod

9 ⟨*srcedo/Dynare.edo.mod* 9⟩≡

```
⟨edo var 11⟩


⟨edo varexo 13⟩


⟨edo parameters 14⟩


//estimated_params;
⟨edo estimated_params 18⟩
//end_estimated_params;

//calibrated_params;
⟨edo calibrated_params 19⟩
//end_calibrated_params;

//free_params;
⟨edo free_params 20⟩
//end_free_params;

//calibrated ME


⟨edo model 21⟩
```

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

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

This code is written to file `srcedo/Dynare.edo.mod`.
Defines:
`jacobian_flag`, used in chunk 38.
`nonlin`, used in chunk 38.
`options_`, used in chunk 38.
`order`, used in chunk 38.
`stoch_simul`, used in chunk 38.

A.1.1 Dynare EDO Var

11 ⟨*edo var* 11⟩≡ (9)

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

Defines:
- AH, used in chunks 24b, 26a, 53b, and 55a.
- AH_obs, used in chunks 26, 27, 55, and 56.
- betas, used in chunks 22, 25, 51, and 54.
- DIFFNORMGDP, used in chunks 23 and 52.
- DIFFREALEC, used in chunks 23, 26a, 52, and 55a.
- DIFFREALEC_obs, used in chunks 26, 27, 55, and 56.
- DIFFREALECD, used in chunks 25, 26a, 54, and 55a.
- DIFFREALECD_obs, used in chunks 26, 27, 55, and 56.
- DIFFREALECH, used in chunks 25, 26a, 54, and 55a.
- DIFFREALECH_obs, used in chunks 26, 27, 55, and 56.
- DIFFREALEIK, used in chunks 23, 26a, 52, and 55a.
- DIFFREALEIK_obs, used in chunks 26, 27, 55, and 56.
- DIFFREALGDP, used in chunks 23, 24b, 26a, 52, 53b, and 55a.
- DIFFREALGDP_obs, used in chunks 26, 27, 55, and 56.
- DIFFREALW, used in chunks 24a, 26a, 53a, and 55a.
- DIFFREALW_obs, used in chunks 26, 27, 55, and 56.
- EC, used in chunks 22, 23, 51, and 52.
- ECD, used in chunks 23–25 and 52–54.
- ECH, used in chunks 23, 25, 52, and 54.
- EFFECD, used in chunks 24c, 25, 53c, and 54.
- EFFECH, used in chunks 25 and 54.
- EFFK, used in chunks 22, 23, 25, 51, 52, and 54.
- EIK, used in chunks 22, 23, 51, and 52.
- empC, used in chunks 23 and 52.
- empK, used in chunks 23 and 52.
- empSC, used in chunks 23 and 52.
- empSK, used in chunks 23 and 52.
- GAP, used in chunks 24b and 53b.
- HC, used in chunks 22–24 and 51–53.
- HG, used in chunks 23, 25, 52, and 54.
- HK, used in chunks 22–24 and 51–53.
- HrC, used in chunks 23 and 52.
- HrK, used in chunks 23 and 52.
- HrSC, used in chunks 23 and 52.
- HrSK, used in chunks 23 and 52.
- HSC, used in chunks 23 and 52.
- HSK, used in chunks 23 and 52.
- INFC, used in chunks 22–24 and 51–53.
- INFC10, used in chunks 24b and 53b.
- INFCNA, used in chunks 22, 24b, 26a, 51, 53b, and 55a.
- INFCNA_obs, used in chunks 26, 27, 55, and 56.
- INFCOR, used in chunks 24b, 26a, 53b, and 55a.
- INFCOR_obs, used in chunks 26, 27, 55, and 56.
- INFGDP, used in chunks 24b and 53b.
- INFK, used in chunks 22–24, 26a, 51–53, and 55a.
- INFK_obs, used in chunks 26, 27, 55, and 56.
- INFWC, used in chunks 23, 24a, 52, and 53a.
- INFWK, used in chunks 23, 24a, 52, and 53a.
- KC, used in chunks 22, 23, 51, and 52.
- KCH, used in chunks 25 and 54.
- KD, used in chunks 24c, 25, 53c, and 54.

KK, used in chunks 22, 23, 51, and 52.
L, used in chunks 21–25 and 50–54.
LAGKCH, used in chunks 25 and 54.
LAGKD, used in chunks 24c, 25, 53c, and 54.
Lpref, used in chunks 23, 25, 52, and 54.
MCC, used in chunks 22 and 51.
MCK, used in chunks 22 and 51.
MUC, used in chunks 22–25 and 51–54.
MUK, used in chunks 22–25 and 51–54.
MUZK, used in chunks 25 and 54.
MUZM, used in chunks 25 and 54.
NORMINFGDP, used in chunks 23 and 52.
PFGAP, used in chunks 22, 24b, 51, and 53b.
PKB, used in chunks 22–24 and 51–53.
QCD, used in chunks 24c and 53c.
QCH, used in chunks 25 and 54.
QK, used in chunks 22, 23, 25, 51, 52, and 54.
R, used in chunks 22, 25, 26a, 51, 54, and 55a.
R_obs, used in chunks 26 and 55.
RC, used in chunks 22, 25, 51, and 54.
RCD, used in chunks 24c and 53c.
RCH, used in chunks 25 and 54.
RK, used in chunks 22, 25, 51, and 54.
RL1, used in chunks 25 and 54.
RL2, used in chunks 25 and 54.
RL3, used in chunks 25 and 54.
RL4, used in chunks 25 and 54.
RL5, used in chunks 25 and 54.
RL6, used in chunks 25 and 54.
RL7, used in chunks 25 and 54.
RT2, used in chunks 25, 26a, 54, and 55a.
RT2_obs, used in chunks 26, 27, 55, and 56.
STAR, used in chunks 25 and 54.
UC, used in chunks 22, 24b, 25, 51, 53b, and 54.
UHC, used in chunks 23 and 52.
UHK, used in chunks 23 and 52.
UHSC, used in chunks 23 and 52.
UHSK, used in chunks 23 and 52.
UK, used in chunks 22, 24b, 25, 51, 53b, and 54.
unemp, used in chunks 23, 26a, 52, and 55a.
unemp_obs, used in chunks 26, 27, 55, and 56.
WC, used in chunks 22, 23, 51, and 52.
WK, used in chunks 22, 23, 51, and 52.
XiL, used in chunks 23, 25, 52, and 54.
YC, used in chunks 22–24 and 51–53.
YK, used in chunks 22–24 and 51–53.

Uses var 40.

A.1.2 Dynare EDO VarExo

13 ⟨*edo varexo* 13⟩≡ (9)

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

Defines:

`eB`, used in chunks 25, 27, 29, 54, 56, and 59.
`eEFFECD`, used in chunks 25, 27, 29, 54, 56, and 59.
`eEFFECH`, used in chunks 25, 27, 29, 54, 56, and 59.
`eEFFK`, used in chunks 25, 27, 29, 54, 56, and 59.
`eHG`, used in chunks 25, 27, 29, 54, 56, and 59.
`eLpref`, used in chunks 25, 27, 29, 54, 56, and 59.
`eMUZK`, used in chunks 25, 27, 29, 54, 56, and 59.
`eMUZM`, used in chunks 25, 27, 29, 54, 56, and 59.
`ePMKC`, used in chunks 22, 27, 29, 51, 56, and 59.
`ePMKK`, used in chunks 22, 27, 29, 51, 56, and 59.
`eR`, used in chunks 22, 27, 29, 51, 56, and 59.
`eSTAR`, used in chunks 25, 27, 29, 54, 56, and 59.
`eXiL`, used in chunks 23, 25, 27, 29, 52, 54, 56, and 59.

A.1.3 Dynare EDO Parameters

14 ⟨*edo parameters* 14⟩≡ (9)

```
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_
rho_EFFECD rho_HG rho_EFFECH tp2 ONE MUZMSS MUZKSS  r_dinf rpr phi_u rho_MUZK rho_MUZ
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_
icoef mu_ betarl MUZCSS RCSS RKSS WCSS WKSS YCSS YKSS MCCSS MCKSS KCSS KKSS LSS HCSS
MUCSS MUKSS AHSS ECDSS KCDSS QCDSS RCDSS ECHSS KCHSS QCHSS RCHSS UKSS UCSS USS MUKSSK
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_
sig_HG sig_XiL sig_lpref sig_R sig_MUZK sig_MUZM sig_PMKC sig_PMKK sig_EFFECH  sig_EF
HSKSS HSCSS HrCSS HrKSS A_HC sigman sigmah A_HK xsi_NC xsi_HrC xsi_NK xsi_HrK rho_XiL
empCSS empKSS HrSKSS HrSCSS empSCSS empSKSS UHCSS UHKSS UHSCSS UHSKSS unempSS DIFFREA
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 20, 23, 31, 49, 52, and 61.
- `A_HK`, used in chunks 20, 23, 31, 49, 52, and 61.
- `a_ks`, used in chunks 19 and 48.
- `AA`, used in chunks 31 and 61.
- `AHSS`, used in chunks 24a, 26a, 31, 35, 53, and 61.
- `alpha_`, used in chunks 19, 22, 24b, 25, 31, 48, 51, 53b, 54, and 61.
- `beta_`, used in chunks 22–25, 31, 35, 51–54, and 61.
- `beta_0`, used in chunks 31 and 61.
- `beta_2`, used in chunks 31 and 61.
- `betarl`, used in chunks 19, 24b, 48, and 53b.
- `DD`, used in chunks 31 and 61.
- `delta_`, used in chunks 19, 22, 31, 48, 51, and 61.
- `delta_cd`, used in chunks 19, 24c, 31, 48, 53c, and 61.
- `delta_ch`, used in chunks 19, 25, 31, 48, 54, and 61.
- `DIFFREALECDSS`, used in chunks 26a, 31, 35, 54, and 61.
- `DIFFREALECDSS_obs`, used in chunks 26a, 34, 35, 55a, 64, and 65.
- `DIFFREALECHSS`, used in chunks 26a, 31, 35, 54, and 61.
- `DIFFREALECHSS_obs`, used in chunks 26a, 34, 35, 55a, 64, and 65.
- `DIFFREALECSS`, used in chunks 26a, 31, 35, 52, and 61.
- `DIFFREALECSS_obs`, used in chunks 26a, 34, 35, 55a, 64, and 65.
- `DIFFREALEIKSS`, used in chunks 26a, 31, 35, 52, and 61.
- `DIFFREALEIKSS_obs`, used in chunks 26a, 34, 35, 55a, 64, and 65.
- `DIFFREALGDPSS`, used in chunks 26a, 31, 35, 52, 53b, and 61.
- `DIFFREALGDPSS_obs`, used in chunks 26a, 34, 35, 55a, 64, and 65.
- `DIFFREALWSS`, used in chunks 26a, 31, 35, 53a, and 61.
- `DIFFREALWSS_obs`, used in chunks 26a, 34, 35, 55a, 64, and 65.
- `ECDSS`, used in chunks 24c, 31, 35, 52–54, and 61.
- `ECHSS`, used in chunks 25, 31, 35, 52, 54, and 61.
- `ECSS`, used in chunks 31, 35, 51, 52, and 61.
- `EIKSS`, used in chunks 23, 31, 35, 51, 52, and 61.
- `empCSS`, used in chunks 31, 35, 52, and 61.
- `empKSS`, used in chunks 31, 35, 52, and 61.
- `empSCSS`, used in chunks 31, 35, 52, and 61.
- `empSKSS`, used in chunks 31, 35, 52, and 61.

eta_cd, used in chunks 20, 24c, 31, 49, 53c, and 61.
eta_cd_eta_cnn, used in chunks 31 and 61.
eta_ch, used in chunks 20, 25, 31, 49, 54, and 61.
eta_ch_eta_cnn, used in chunks 31 and 61.
eta_cnn, used in chunks 20, 22, 31, 49, 51, and 61.
g_y, used in chunks 19 and 48.
gam_ech, used in chunks 19, 25, 48, and 54.
gam_h, used in chunks 19, 23, 48, and 52.
gam_ic, used in chunks 19, 23, 48, and 52.
gam_icd, used in chunks 19, 24c, 48, and 53c.
gam_pc, used in chunks 18, 22, 28, 47, 51, and 58.
gam_wc, used in chunks 18, 23, 28, 47, 52, and 58.
h, used in chunks 18, 22, 28, 31, 47, 51, 58, and 61.
h_cd, used in chunks 19, 24c, 31, 48, 53c, and 61.
h_ch, used in chunks 19, 25, 31, 48, 54, and 61.
hc_hk, used in chunks 31 and 61.
HCSS, used in chunks 23, 24, 31, 35, 51–53, and 61.
HKSS, used in chunks 23, 24, 31, 35, 51–53, and 61.
HrCSS, used in chunks 31, 35, 52, and 61.
HrKSS, used in chunks 31, 35, 52, and 61.
HrSCSS, used in chunks 31, 35, 52, and 61.
HrSKSS, used in chunks 31, 35, 52, and 61.
HSCSS, used in chunks 23, 31, 35, 52, and 61.
HSKSS, used in chunks 23, 31, 35, 52, and 61.
HSS, used in chunks 31 and 61.
icoef, used in chunks 19 and 48.
IMPHSSS, used in chunks 31 and 61.
INFC10SS, used in chunks 31, 35, 53b, and 61.
INFCNASS, used in chunks 22, 26a, 31, 35, 51, 53b, and 61.
INFCNASS_obs, used in chunks 26a, 34, 35, 55a, 64, and 65.
INFCORSS, used in chunks 26a, 31, 35, 53b, and 61.
INFCORSS_obs, used in chunks 26a, 34, 35, 55a, 64, and 65.
INFCSS, used in chunks 22, 31, 34, 35, 51–53, 61, and 64.
INFGDPSS, used in chunks 31, 35, 53b, and 61.
INFKSS, used in chunks 22, 26a, 31, 34, 35, 51–53, 61, and 64.
INFKSS_obs, used in chunks 26a, 34, 35, 55a, 64, and 65.
INFWCSS, used in chunks 23, 31, 35, 52, 53a, and 61.
INFWKSS, used in chunks 23, 31, 35, 52, 53a, and 61.
KCDSS, used in chunks 31, 35, 53c, 54, and 61.
KCHSS, used in chunks 31, 35, 54, and 61.
KCSS, used in chunks 22, 31, 35, 51, 52, and 61.
KKSS, used in chunks 22, 31, 35, 51, 52, and 61.
LSS, used in chunks 31, 35, 50–54, and 61.
MCCSS, used in chunks 31, 35, 51, and 61.
MCKSS, used in chunks 31, 35, 51, and 61.
mu_, used in chunks 20, 22, 25, 31, 49, 51, 54, and 61.
MUCSS, used in chunks 31, 34, 35, 51–54, 61, and 64.
MUCSShabit, used in chunks 31 and 61.
MUKSS, used in chunks 31, 34, 35, 51–54, 61, and 64.
MUKSShabit, used in chunks 31 and 61.
MUZCSS, used in chunks 31 and 61.
MUZKSS, used in chunks 19, 25, 31, 35, 48, 54, and 61.
MUZMSS, used in chunks 19, 25, 31, 35, 48, 54, and 61.
ONE, used in chunks 19, 31, 35, 48, 51–54, and 61.
pbeta, used in chunks 19, 31, 48, and 61.
phi_cd, used in chunks 18, 24c, 28, 47, 53c, and 58.
phi_ech, used in chunks 18, 25, 28, 47, 54, and 58.

phi_H, used in chunks 18, 23, 28, 47, 52, and 58.
phi_ic, used in chunks 18, 23, 28, 47, 52, and 58.
phi_pc, used in chunks 18, 22, 28, 47, 51, and 58.
phi_u, used in chunks 19, 22, 25, 48, 51, and 54.
phi_wc, used in chunks 18, 23, 28, 47, 52, and 58.
PKBSS, used in chunks 31, 35, 51–53, and 61.
PYSS, used in chunks 31 and 61.
QCDSS, used in chunks 31, 35, 53c, and 61.
QCHSS, used in chunks 31, 35, 54, and 61.
QKSS, used in chunks 31, 35, 51, 52, 54, and 61.
r_dinf, used in chunks 19, 22, 48, and 51.
r_dy, used in chunks 19 and 48.
r_inf, used in chunks 18, 22, 28, 47, 51, and 58.
r_y, used in chunks 18, 22, 28, 47, 51, and 58.
RCDSS, used in chunks 31, 35, 53c, and 61.
RCHSS, used in chunks 31, 35, 54, and 61.
RCSS, used in chunks 31, 35, 51, 54, and 61.
rho_B, used in chunks 18, 25, 28, 47, 54, and 58.
rho_EFFECD, used in chunks 18, 25, 28, 47, 54, and 58.
rho_EFFECH, used in chunks 18, 25, 28, 47, 54, and 58.
rho_EFFK, used in chunks 18, 25, 28, 47, 54, and 58.
rho_HG, used in chunks 18, 25, 28, 47, 54, and 58.
rho_lpref, used in chunks 18, 25, 28, 47, 54, and 58.
rho_MUZK, used in chunks 19 and 48.
rho_MUZM, used in chunks 19 and 48.
rho_R, used in chunks 18, 22, 28, 47, 51, and 58.
rho_STAR, used in chunks 18, 25, 28, 47, 54, and 58.
rho_XiL, used in chunks 18, 25, 28, 47, 54, and 58.
RKSS, used in chunks 31, 35, 51, 54, and 61.
RL1SS, used in chunks 31, 35, 54, and 61.
RL2SS, used in chunks 31, 35, 54, and 61.
RL3SS, used in chunks 31, 35, 54, and 61.
RL4SS, used in chunks 31, 35, 54, and 61.
RL5SS, used in chunks 31, 35, 54, and 61.
RL6SS, used in chunks 31, 35, 54, and 61.
RL7SS, used in chunks 31, 35, 54, and 61.
Rnr, used in chunks 31 and 61.
rpr, used in chunks 19, 22, 31, 48, 51, and 61.
RR, used in chunks 31 and 61.
RSS, used in chunks 22, 26a, 31, 34, 35, 51, 54, 61, and 64.
RSS_obs, used in chunks 26a, 34, 35, 55a, 64, and 65.
RT2SS, used in chunks 26a, 31, 34, 35, 54, 61, and 64.
RT2SS_obs, used in chunks 26a, 34, 35, 55a, 64, and 65.
s_AS, used in chunks 19, 31, 48, and 61.
s_c_ech, used in chunks 31 and 61.
s_ecdc, used in chunks 20, 24b, 31, 34, 49, 53b, 61, and 64.
s_k, used in chunks 20, 23, 24b, 31, 34, 49, 52, 53b, 61, and 64.
s_k_ecd, used in chunks 31 and 61.
s_k_eik, used in chunks 31 and 61.
s_yc, used in chunks 31 and 61.
sig_B, used in chunks 18, 27, 47, and 56.
sig_EFFECD, used in chunks 18, 27, 47, and 56.
sig_EFFECH, used in chunks 18, 27, 47, and 56.
sig_EFFK, used in chunks 18, 27, 47, and 56.
sig_HG, used in chunks 18, 27, 47, and 56.
sig_lpref, used in chunks 18, 27, 47, and 56.
sig_MUZK, used in chunks 18, 27, 47, and 56.

sig_MUZM, used in chunks 18, 27, 47, and 56.
sig_PMKC, used in chunks 18, 27, 47, and 56.
sig_PMKK, used in chunks 18, 27, 47, and 56.
sig_R, used in chunks 18, 27, 47, and 56.
sig_STAR, used in chunks 18, 27, 47, and 56.
sig_XiL, used in chunks 18, 27, 47, and 56.
sigmah, used in chunks 18, 23, 28, 31, 47, 52, 58, and 61.
sigman, used in chunks 18, 23, 28, 31, 47, 52, 58, and 61.
theta_c, used in chunks 19, 22, 31, 48, 51, and 61.
theta_k, used in chunks 19, 22, 31, 48, 51, and 61.
theta_wc, used in chunks 20, 23, 31, 49, 52, and 61.
theta_wk, used in chunks 20, 23, 31, 49, 52, and 61.
tp2, used in chunks 18, 25, 28, 31, 47, 54, 58, and 61.
UCSS, used in chunks 31 and 61.
UHCSS, used in chunks 31, 35, 52, and 61.
UHKSS, used in chunks 31, 35, 52, and 61.
UHSCSS, used in chunks 31, 35, 52, and 61.
UHSKSS, used in chunks 31, 35, 52, and 61.
UKSS, used in chunks 31 and 61.
unempSS, used in chunks 19, 26a, 31, 34, 35, 48, 52, 61, and 64.
unempSS_obs, used in chunks 26a, 34, 35, 55a, 64, and 65.
USS, used in chunks 24b, 31, 35, 51, 53b, 54, and 61.
WCSS, used in chunks 31, 35, 51, 52, and 61.
WKSS, used in chunks 31, 35, 51, 52, and 61.
xsi_HrC, used in chunks 20, 23, 31, 49, 52, and 61.
xsi_HrK, used in chunks 20, 23, 31, 49, 52, and 61.
xsi_NC, used in chunks 20, 23, 31, 49, 52, and 61.
xsi_NK, used in chunks 20, 23, 31, 49, 52, and 61.
ycbi, used in chunks 31 and 61.
ycbi_ykb, used in chunks 31 and 61.
YCSS, used in chunks 22–24, 31, 35, 51–53, and 61.
ykb, used in chunks 31 and 61.
YKSS, used in chunks 22–24, 31, 35, 51–53, and 61.
YYSS, used in chunks 31 and 61.

A.1.4 Dynare EDO Estimated Params

18 ⟨*edo estimated_params* 18⟩≡ (9)

```
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 14 43, gam_wc 14 43, h 14 43, phi_cd 14 43, phi_ech 14 43, phi_H 14 43, phi_ic 14 43, phi_pc 14 43, phi_wc 14 43, r_inf 14 43, r_y 14 43, rho_B 14 43, rho_EFFECD 14 43, rho_EFFECH 14 43, rho_EFFK 14 43, rho_HG 14 43, rho_lpref 14 43, rho_R 14 43, rho_STAR 14 43, rho_XiL 14 43, sig_B 14 43, sig_EFFECD 14 43, sig_EFFECH 14 43, sig_EFFK 14 43, sig_HG 14 43, sig_lpref 14 43, sig_MUZK 14 43, sig_MUZM 14 43, sig_PMKC 14 43, sig_PMKK 14 43, sig_R 14 43, sig_STAR 14 43, sig_XiL 14 43, sigmah 14 43, sigman 14 43, and tp2 14 43.

A.1.5 Dynare EDO Calibrated Params

19 ⟨*edo calibrated_params* 19⟩≡ (9)

```
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` 14 43, `alpha_` 14 43, `betarl` 14 43, `delta_` 14 43, `delta_cd` 14 43, `delta_ch` 14 43, `g_y` 14 43, `gam_ech` 14 43, `gam_h` 14 43, `gam_ic` 14 43, `gam_icd` 14 43, `h_cd` 14 43, `h_ch` 14 43, `icoef` 14 43, `MUZKSS` 14 43, `MUZMSS` 14 43, `ONE` 14 43, `pbeta` 14 43, `phi_u` 14 43, `r_dinf` 14 43, `r_dy` 14 43, `rho_MUZK` 14 43, `rho_MUZM` 14 43, `rpr` 14 43, `s_AS` 14 43, `theta_c` 14 43, `theta_k` 14 43, and `unempSS` 14 43.

A.1.6 Dynare EDO Free Params

20 ⟨*edo free_params* 20⟩≡ (9)

```
//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 14 43, A_HK 14 43, eta_cd 14 43, eta_ch 14 43, eta_cnn 14 43, mu_ 14 43, s_ecdc 14 43, s_k 14 43, theta_wc 14 43, theta_wk 14 43, xsi_HrC 14 43, xsi_HrK 14 43, xsi_NC 14 43, and xsi_NK 14 43.

A.1.7 Dynare EDO Model

21 ⟨*edo model* 21⟩≡ (9)

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

model;
⟨edo model prelim 22⟩

// XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
// labor block
// TOTAL LABOR INPUT (called "L" in the paper, I kept the "H" notation of the original EDO prg)
⟨edo model labor 23⟩

// Identities
⟨edo model identities 24a⟩


// XXXXXXXXXXXXXXXXXXXXXX
// Aggregate hours equals agg hours in each sector
⟨edo model hours 24b⟩

// See Section 8: Data Identities

// new equations
// Durable Block

⟨edo model durables 24c⟩

// Housing Block
⟨edo model housing 25⟩

//measurement_equations;
⟨edo model measurement 26a⟩
//end_measurement_equations;
end;

⟨edo model varobs 26b⟩

shocks;
⟨edo model shocks 27⟩
end;

steady;
```

```
  estimated_params;
  ⟨edo model estimated_params 28⟩

  ⟨edo model stderr 29⟩
  end;
```
Uses L 11 40.

A.1.8 Dynare EDO Prelim

22 ⟨*edo model prelim* 22⟩≡ (21)

```
  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)*INF
  MCK*YK*theta_k/PKB-(theta_k-1)*YK-100*phi_pc*(INFK-gam_pc*INFK(-1)-(1-gam_pc)*INFKSS)
  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)
  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/
```
Uses alpha_ 14 43, beta_ 14 43, betas 11 40, delta_ 14 43, EC 11 40, EFFK 11 40, EIK 11 40, ePMKC 13 42, ePMKK 13 42, eR 13 42, eta_cnn 14 43, gam_pc 14 43, h 14 43, HC 11 40, HK 11 40, INFC 11 40, INFCNA 11 40, INFCNASS 14 43, INFCSS 14 43, INFK 11 40, INFKSS 14 43, KC 11 40, KCSS 14 43, KK 11 40, KKSS 14 43, L 11 40, MCC 11 40, MCK 11 40, mu_ 14 43, MUC 11 40, MUK 11 40, PFGAP 11 40, phi_pc 14 43, phi_u 14 43, PKB 11 40, QK 11 40, R 11 40, r_dinf 14 43, r_inf 14 43, r_y 14 43, RC 11 40, rho_R 14 43, RK 11 40, rpr 14 43, RSS 14 43, theta_c 14 43, theta_k 14 43, UC 11 40, UK 11 40, WC 11 40, WK 11 40, YC 11 40, YCSS 14 43, YK 11 40, and YKSS 14 43.

A.1.9 Dynare EDO Labor

23 ⟨*edo model labor* 23⟩≡ (21)

```
-100+UHC*theta_wc-(theta_wc-1)*WC-100*phi_wc*(INFWC-gam_wc*INFWC(-1)-(1-gam_wc)*INFWCSS)*INFWC*
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)*INFWKSS)*INFWK*
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+sigmah)))=
UHSC*L*Lpref-A_HC*((1+sigman)/(1+sigman/(1+sigmah)))*(HSC)^(-1+(1+sigman)/(1+sigman/(1+sigmah))
UHK*L*Lpref-A_HK*((1+sigman)/(1+sigman/(1+sigmah)))*(HK)^(-1+(1+sigman)/(1+sigman/(1+sigmah)))=
UHSK*L*Lpref-A_HK*((1+sigman)/(1+sigman/(1+sigmah)))*(HSK)^(-1+(1+sigman)/(1+sigman/(1+sigmah))
empC-((1+sigmah)/sigmah*xsi_NC/xsi_HrC)^(-1/(1+sigmah+sigman))*HC^(1/(1+sigman/(1+sigmah)))=0;
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+sigmah)))=0;
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+sigmah)))=0
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+sigmah)))=0
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_*(1/EFFK)*
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 14 43, A_HK 14 43, beta_ 14 43, DIFFNORMGDP 11 40, DIFFREALEC 11 40, DIFFREALEIK 11 40, DIFFREALGDP 11 40, EC 11 40, ECD 11 40, ECH 11 40, EFFK 11 40, EIK 11 40, EIKSS 14 43, empC 11 40, empK 11 40, empSC 11 40, empSK 11 40, eXiL 13 42, gam_h 14 43, gam_ic 14 43, gam_wc 14 43, HC 11 40, HCSS 14 43, HG 11 40, HK 11 40, HKSS 14 43, HrC 11 40, HrK 11 40, HrSC 11 40, HrSK 11 40, HSC 11 40, HSCSS 14 43, HSK 11 40, HSKSS 14 43, INFC 11 40, INFK 11 40, INFWC 11 40, INFWCSS 14 43, INFWK 11 40, INFWKSS 14 43, KC 11 40, KK 11 40, L 11 40, Lpref 11 40, MUC 11 40, MUK 11 40, NORMINFGDP 11 40, phi_H 14 43, phi_ic 14 43, phi_wc 14 43, PKB 11 40, QK 11 40, s_k 14 43, sigmah 14 43, sigman 14 43, theta_wc 14 43, theta_wk 14 43, UHC 11 40, UHK 11 40, UHSC 11 40, UHSK 11 40, unemp 11 40, WC 11 40, WK 11 40, XiL 11 40, xsi_HrC 14 43, xsi_HrK 14 43, xsi_NC 14 43, xsi_NK 14 43, YC 11 40, YCSS 14 43, YK 11 40, and YKSS 14 43.

A.1.10 Dynare EDO Identities

24a ⟨*edo model identities* 24a⟩≡ (21)

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

Uses AHSS 14 43, DIFFREALW 11 40, HCSS 14 43, HKSS 14 43, INFC 11 40, INFWC 11 40, and INFWK 11 40.

A.1.11 Dynare EDO Hours

24b ⟨*edo model hours* 24b⟩≡ (21)

```
AH-HC-HK=0;
ln(INFGDP)-ln(INFC)-ln(YC*MUC/YC(-1))+ln(DIFFREALGDP)-ln((1+PKB*YK/YC)/(1+PKB(-1)*YK(
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)
ln(INFC10)-betarl*ln(INFC10(+1))-(1-betarl)*ln(INFCOR)=0;
```

Uses AH 11 40, alpha_ 14 43, betarl 14 43, DIFFREALGDP 11 40, GAP 11 40, HC 11 40, HCSS 14 43, HK 11 40, HKSS 14 43, INFC 11 40, INFC10 11 40, INFCNA 11 40, INFCOR 11 40, INFGDP 11 40, INFK 11 40, MUC 11 40, PFGAP 11 40, PKB 11 40, s_ecdc 14 43, s_k 14 43, UC 11 40, UK 11 40, USS 14 43, YC 11 40, YCSS 14 43, YK 11 40, and YKSS 14 43.

A.1.12 Dynare EDO Durables

24c ⟨*edo model durables* 24c⟩≡ (21)

```
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(
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/
```

Uses beta_ 14 43, delta_cd 14 43, ECD 11 40, ECDSS 14 43, EFFECD 11 40, eta_cd 14 43, gam_icd 14 43, h_cd 14 43, KD 11 40, L 11 40, LAGKD 11 40, MUK 11 40, phi_cd 14 43, PKB 11 40, QCD 11 40, and RCD 11 40.

A.1.13 Dynare EDO Housing

25 ⟨*edo model housing* 25⟩≡ (21)

```
L*RCH-eta_ch/(KCH(-1)/MUC-h_ch*LAGKCH(-1)/(MUC*MUC(-1)))+beta_*eta_ch*h_ch/(KCH-h_ch*KCH(-1)/MU
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/EFFECH)*10
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(RL6) + ln(R
```

Uses alpha_ 14 43, beta_ 14 43, betas 11 40, delta_ch 14 43, DIFFREALECD 11 40, DIFFREALECH 11 40, eB 13 42, ECD 11 40, ECH 11 40, ECHSS 14 43, eEFFECD 13 42, eEFFECH 13 42, eEFFK 13 42, EFFECD 11 40, EFFECH 11 40, EFFK 11 40, eHG 13 42, eLpref 13 42, eMUZK 13 42, eMUZM 13 42, eSTAR 13 42, eta_ch 14 43, eXiL 13 42, gam_ech 14 43, h_ch 14 43, HG 11 40, KCH 11 40, KD 11 40, L 11 40, LAGKCH 11 40, LAGKD 11 40, Lpref 11 40, mu_ 14 43, MUC 11 40, MUK 11 40, MUZK 11 40, MUZKSS 14 43, MUZM 11 40, MUZMSS 14 43, phi_ech 14 43, phi_u 14 43, QCH 11 40, QK 11 40, R 11 40, RC 11 40, RCH 11 40, rho_B 14 43, rho_EFFECD 14 43, rho_EFFECH 14 43, rho_EFFK 14 43, rho_HG 14 43, rho_lpref 14 43, rho_STAR 14 43, rho_XiL 14 43, RK 11 40, RL1 11 40, RL2 11 40, RL3 11 40, RL4 11 40, RL5 11 40, RL6 11 40, RL7 11 40, RT2 11 40, STAR 11 40, tp2 14 43, UC 11 40, UK 11 40, and XiL 11 40.

A.1.14 Dynare EDO Model Measurement

26a ⟨*edo model measurement* 26a⟩≡ (21)

```
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 11 40, AH_obs 11 40, AHSS 14 43, DIFFREALEC 11 40, DIFFREALEC_obs 11 40, DIFFREALECD 11 40, DIFFREALECD_obs 11 40, DIFFREALECDSS 14 43, DIFFREALECDSS_obs 14 43, DIFFREALECH 11 40, DIFFREALECH_obs 11 40, DIFFREALECHSS 14 43, DIFFREALECHSS_obs 14 43, DIFFREALECSS 14 43, DIFFREALECSS_obs 14 43, DIFFREALEIK 11 40, DIFFREALEIK_obs 11 40, DIFFREALEIKSS 14 43, DIFFREALEIKSS_obs 14 43, DIFFREALGDP 11 40, DIFFREALGDP_obs 11 40, DIFFREALGDPSS 14 43, DIFFREALGDPSS_obs 14 43, DIFFREALW 11 40, DIFFREALW_obs 11 40, DIFFREALWSS 14 43, DIFFREALWSS_obs 14 43, INFCNA 11 40, INFCNA_obs 11 40, INFCNASS 14 43, INFCNASS_obs 14 43, INFCOR 11 40, INFCOR_obs 11 40, INFCORSS 14 43, INFCORSS_obs 14 43, INFK 11 40, INFK_obs 11 40, INFKSS 14 43, INFKSS_obs 14 43, R 11 40, R_obs 11 40, RSS 14 43, RSS_obs 14 43, RT2 11 40, RT2_obs 11 40, RT2SS 14 43, RT2SS_obs 14 43, unemp 11 40, unemp_obs 11 40, unempSS 14 43, and unempSS_obs 14 43.

A.1.15 Dynare EDO Model VarObs

26b ⟨*edo model varobs* 26b⟩≡ (21)

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

Uses AH_obs 11 40, DIFFREALEC_obs 11 40, DIFFREALECD_obs 11 40, DIFFREALECH_obs 11 40, DIFFREALEIK_obs 11 40, DIFFREALGDP_obs 11 40, DIFFREALW_obs 11 40, INFCNA_obs 11 40, INFCOR_obs 11 40, INFK_obs 11 40, R_obs 11 40, RT2_obs 11 40, and unemp_obs 11 40.

A.1.16 Dynare EDO Shocks

27 ⟨*edo model shocks* 27⟩≡ (21)

```
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 11 40, DIFFREALEC_obs 11 40, DIFFREALECD_obs 11 40, DIFFREALECH_obs 11 40, DIFFREALEIK_obs 11 40, DIFFREALGDP_obs 11 40, DIFFREALW_obs 11 40, eB 13 42, eEFFECD 13 42, eEFFECH 13 42, eEFFK 13 42, eHG 13 42, eLpref 13 42, eMUZK 13 42, eMUZM 13 42, ePMKC 13 42, ePMKK 13 42, eR 13 42, eSTAR 13 42, eXiL 13 42, INFCNA_obs 11 40, INFCOR_obs 11 40, INFK_obs 11 40, RT2_obs 11 40, sig_B 14 43, sig_EFFECD 14 43, sig_EFFECH 14 43, sig_EFFK 14 43, sig_HG 14 43, sig_lpref 14 43, sig_MUZK 14 43, sig_MUZM 14 43, sig_PMKC 14 43, sig_PMKK 14 43, sig_R 14 43, sig_STAR 14 43, sig_XiL 14 43, unemp_obs 11 40, and var 40.

A.1.17 Dynare EDO Model Estimated Params

28 ⟨*edo model estimated_params* 28⟩≡ (21)

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

Uses gam_pc 14 43, gam_wc 14 43, h 14 43, phi_cd 14 43, phi_ech 14 43, phi_H 14 43, phi_ic 14 43, phi_pc 14 43, phi_wc 14 43, r_inf 14 43, r_y 14 43, rho_B 14 43, rho_EFFECD 14 43, rho_EFFECH 14 43, rho_EFFK 14 43, rho_HG 14 43, rho_lpref 14 43, rho_R 14 43, rho_STAR 14 43, rho_XiL 14 43, sigmah 14 43, sigman 14 43, and tp2 14 43.

A.1.18 Dynare EDO Model Stderr

29 ⟨*edo model stderr* 29⟩≡ (21)

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

Uses `eB` 13 42, `eEFFECD` 13 42, `eEFFECH` 13 42, `eEFFK` 13 42, `eHG` 13 42, `eLpref` 13 42, `eMUZK` 13 42, `eMUZM` 13 42, `ePMKC` 13 42, `ePMKK` 13 42, `eR` 13 42, `eSTAR` 13 42, and `eXiL` 13 42.

A.2 Dynare_edo_steadystate.m

30 ⟨*srcedo/Dynare.edo.steadystate.m* 30⟩≡

```
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;

⟨edo steady state values 31⟩

%end_steady_state;

%trends;

⟨edo steady state trends 34⟩

%end_trends;


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


⟨edo steady state result return 35⟩
```

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

A.2.1 EDO Steady State Values

31 ⟨*edo steady state values* 31⟩≡ (30)

```
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
```

```
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 14 43, A_HK 14 43, AA 14 43, AHSS 14 43, alpha_ 14 43, beta_ 14 43, beta_0 14 43, beta_2 14 43, DD 14 43, delta_ 14 43, delta_cd 14 43, delta_ch 14 43, DIFFREALECDSS 14 43, DIFFREALECHSS 14 43, DIFFREALECSS 14 43, DIFFREALEIKSS 14 43, DIFFREALGDPSS 14 43, DIFFREALWSS 14 43, ECDSS 14 43, ECHSS 14 43, ECSS 14 43, EIKSS 14 43, empCSS 14 43, empKSS 14 43, empSCSS 14 43, empSKSS 14 43, eta_cd 14 43, eta_cd_eta_cnn 14 43, eta_ch 14 43, eta_ch_eta_cnn 14 43, eta_cnn 14 43, h 14 43, h_cd 14 43, h_ch 14 43, hc_hk 14 43, HCSS 14 43, HKSS 14 43, HrCSS 14 43, HrKSS 14 43, HrSCSS 14 43, HrSKSS 14 43, HSCSS 14 43, HSKSS 14 43, HSS 14 43, IMPHSSS 14 43, INFC10SS 14 43, INFCNASS 14 43, INFCORSS 14 43, INFCSS 14 43, INFGDPSS 14 43, INFKSS 14 43, INFWCSS 14 43, INFWKSS 14 43, KCDSS 14 43, KCHSS 14 43, KCSS 14 43, KKSS 14 43, LSS 14 43, MCCSS 14 43, MCKSS 14 43, mu_ 14 43, MUCSS 14 43, MUCSShabit 14 43, MUKSS 14 43, MUKSShabit 14 43, MUZCSS 14 43, MUZKSS 14 43, MUZMSS 14 43, ONE 14 43, pbeta 14 43, PKBSS 14 43, PYSS 14 43, QCDSS 14 43, QCHSS 14 43, QKSS 14 43, RCDSS 14 43, RCHSS 14 43, RCSS 14 43, RKSS 14 43, RL1SS 14 43, RL2SS 14 43, RL3SS 14 43, RL4SS 14 43, RL5SS 14 43, RL6SS 14 43, RL7SS 14 43, Rnr 14 43, rpr 14 43, RR 14 43, RSS 14 43, RT2SS 14 43, s_AS 14 43, s_c_ech 14 43, s_ecdc 14 43, s_k 14 43, s_k_ecd 14 43, s_k_eik 14 43, s_yc 14 43, sigmah 14 43, sigman 14 43, theta_c 14 43, theta_k 14 43, theta_wc 14 43, theta_wk 14 43, tp2 14 43, UCSS 14 43, UHCSS 14 43, UHKSS 14 43, UHSCSS 14 43, UHSKSS 14 43, UKSS 14 43, unempSS 14 43, USS 14 43, WCSS 14 43, WKSS 14 43, xsi_HrC 14 43, xsi_HrK 14 43, xsi_NC 14 43, xsi_NK 14 43, ycbi 14 43, ycbi_ykb 14 43, YCSS 14 43, ykb 14 43, YKSS 14 43, and YYSS 14 43.

A.2.2 EDO Steady State Trends

34 ⟨*edo steady state trends* 34⟩≡ (30)

```
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 14 43, DIFFREALECHSS_obs 14 43, DIFFREALECSS_obs 14 43, DIFFREALEIKSS_obs 14 43, DIFFREALGDPSS_obs 14 43, DIFFREALWSS_obs 14 43, INFCNASS_obs 14 43, INFCORSS_obs 14 43, INFCSS 14 43, INFKSS 14 43, INFKSS_obs 14 43, MUCSS 14 43, MUKSS 14 43, RSS 14 43, RSS_obs 14 43, RT2SS 14 43, RT2SS_obs 14 43, s_ecdc 14 43, s_k 14 43, unempSS 14 43, and unempSS_obs 14 43.

A.2.3 EDO Steady State Result Return

35 ⟨*edo steady state result return* 35⟩≡ (30)

```
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 14 43, beta_ 14 43, DIFFREALECDSS 14 43, DIFFREALECDSS_obs 14 43, DIFFREALECHSS 14 43, DIFFREALECHSS_obs 14 43, DIFFREALECSS 14 43, DIFFREALECSS_obs 14 43, DIFFREALEIKSS 14 43, DIFFREALEIKSS_obs 14 43, DIFFREALGDPSS 14 43, DIFFREALGDPSS_obs 14 43, DIFFREALWSS 14 43, DIFFREALWSS_obs 14 43, ECDSS 14 43, ECHSS 14 43, ECSS 14 43, EIKSS 14 43, empCSS 14 43, empKSS 14 43, empSCSS 14 43, empSKSS 14 43, HCSS 14 43, HKSS 14 43, HrCSS 14 43, HrKSS 14 43, HrSCSS 14 43, HrSKSS 14 43, HSCSS 14 43, HSKSS 14 43, INFC10SS 14 43, INFCNASS 14 43, INFCNASS_obs 14 43, INFCORSS 14 43, INFCORSS_obs 14 43, INFCSS 14 43, INFGDPSS 14 43, INFKSS 14 43, INFKSS_obs 14 43, INFWCSS 14 43, INFWKSS 14 43, KCDSS 14 43, KCHSS 14 43, KCSS 14 43, KKSS 14 43, LSS 14 43, MCCSS 14 43, MCKSS 14 43, MUCSS 14 43, MUKSS 14 43, MUZKSS 14 43, MUZMSS 14 43, ONE 14 43, PKBSS 14 43, QCDSS 14 43, QCHSS 14 43, QKSS 14 43, RCDSS 14 43, RCHSS 14 43, RCSS 14 43, RKSS 14 43, RL1SS 14 43, RL2SS 14 43, RL3SS 14 43, RL4SS 14 43, RL5SS 14 43, RL6SS 14 43, RL7SS 14 43, RSS 14 43, RSS_obs 14 43, RT2SS 14 43, RT2SS_obs 14 43, UHCSS 14 43, UHKSS 14 43, UHSCSS 14 43, UHSKSS 14 43, unempSS 14 43, unempSS_obs 14 43, USS 14 43, WCSS 14 43, WKSS 14 43, YCSS 14 43, and YKSS 14 43.

A.3 linearized.mod

38 ⟨*srcedo/linearized.mod* 38⟩≡

```
⟨linearized var 40⟩


⟨linearized varexo 42⟩


⟨linearized parameters 43⟩


//estimated_params;
⟨linearized estimated params 47⟩
//end_estimated_params;

//calibrated_params;
⟨linearized calibrated params 48⟩
//end_calibrated_params;

//free_params;
⟨linearized free params 49⟩
//end_free_params;

⟨linearized calibrated ME 50⟩

⟨linearized varobs 55b⟩

⟨linearized shocks 56⟩
end;

steady;

estimated_params;
⟨linearized steady estimated params 58⟩

⟨linearized stderr 59⟩
end;


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

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

This code is written to file `srcedo/linearized.mod`.
Uses `jacobian_flag` 9, `nonlin` 9, `options_` 9, `order` 9, and `stoch_simul` 9.

A.3.1 Linearized Var

40 ⟨*linearized var* 40⟩≡ (38)

```
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 DIFF
```

Defines:
- AH, used in chunks 24b, 26a, 53b, and 55a.
- AH_obs, used in chunks 26, 27, 55, and 56.
- betas, used in chunks 22, 25, 51, and 54.
- DIFFNORMGDP, used in chunks 23 and 52.
- DIFFREALEC, used in chunks 23, 26a, 52, and 55a.
- DIFFREALEC_obs, used in chunks 26, 27, 55, and 56.
- DIFFREALECD, used in chunks 25, 26a, 54, and 55a.
- DIFFREALECD_obs, used in chunks 26, 27, 55, and 56.
- DIFFREALECH, used in chunks 25, 26a, 54, and 55a.
- DIFFREALECH_obs, used in chunks 26, 27, 55, and 56.
- DIFFREALEIK, used in chunks 23, 26a, 52, and 55a.
- DIFFREALEIK_obs, used in chunks 26, 27, 55, and 56.
- DIFFREALGDP, used in chunks 23, 24b, 26a, 52, 53b, and 55a.
- DIFFREALGDP_obs, used in chunks 26, 27, 55, and 56.
- DIFFREALW, used in chunks 24a, 26a, 53a, and 55a.
- DIFFREALW_obs, used in chunks 26, 27, 55, and 56.
- EC, used in chunks 22, 23, 51, and 52.
- ECD, used in chunks 23–25 and 52–54.
- ECH, used in chunks 23, 25, 52, and 54.
- EFFECD, used in chunks 24c, 25, 53c, and 54.
- EFFECH, used in chunks 25 and 54.
- EFFK, used in chunks 22, 23, 25, 51, 52, and 54.
- EIK, used in chunks 22, 23, 51, and 52.
- empC, used in chunks 23 and 52.
- empK, used in chunks 23 and 52.
- empSC, used in chunks 23 and 52.
- empSK, used in chunks 23 and 52.
- GAP, used in chunks 24b and 53b.
- HC, used in chunks 22–24 and 51–53.
- HG, used in chunks 23, 25, 52, and 54.
- HK, used in chunks 22–24 and 51–53.
- HrC, used in chunks 23 and 52.
- HrK, used in chunks 23 and 52.
- HrSC, used in chunks 23 and 52.
- HrSK, used in chunks 23 and 52.
- HSC, used in chunks 23 and 52.
- HSK, used in chunks 23 and 52.
- INFC, used in chunks 22–24 and 51–53.
- INFC10, used in chunks 24b and 53b.
- INFCNA, used in chunks 22, 24b, 26a, 51, 53b, and 55a.
- INFCNA_obs, used in chunks 26, 27, 55, and 56.
- INFCOR, used in chunks 24b, 26a, 53b, and 55a.
- INFCOR_obs, used in chunks 26, 27, 55, and 56.
- INFGDP, used in chunks 24b and 53b.
- INFK, used in chunks 22–24, 26a, 51–53, and 55a.
- INFK_obs, used in chunks 26, 27, 55, and 56.
- INFWC, used in chunks 23, 24a, 52, and 53a.
- INFWK, used in chunks 23, 24a, 52, and 53a.
- KC, used in chunks 22, 23, 51, and 52.
- KCH, used in chunks 25 and 54.
- KD, used in chunks 24c, 25, 53c, and 54.

KK, used in chunks 22, 23, 51, and 52.
L, used in chunks 21–25 and 50–54.
LAGKCH, used in chunks 25 and 54.
LAGKD, used in chunks 24c, 25, 53c, and 54.
Lpref, used in chunks 23, 25, 52, and 54.
MCC, used in chunks 22 and 51.
MCK, used in chunks 22 and 51.
MUC, used in chunks 22–25 and 51–54.
MUK, used in chunks 22–25 and 51–54.
MUZK, used in chunks 25 and 54.
MUZM, used in chunks 25 and 54.
NORMINFGDP, used in chunks 23 and 52.
PFGAP, used in chunks 22, 24b, 51, and 53b.
PKB, used in chunks 22–24 and 51–53.
QCD, used in chunks 24c and 53c.
QCH, used in chunks 25 and 54.
QK, used in chunks 22, 23, 25, 51, 52, and 54.
R, used in chunks 22, 25, 26a, 51, 54, and 55a.
R_obs, used in chunks 26 and 55.
RC, used in chunks 22, 25, 51, and 54.
RCD, used in chunks 24c and 53c.
RCH, used in chunks 25 and 54.
RK, used in chunks 22, 25, 51, and 54.
RL1, used in chunks 25 and 54.
RL2, used in chunks 25 and 54.
RL3, used in chunks 25 and 54.
RL4, used in chunks 25 and 54.
RL5, used in chunks 25 and 54.
RL6, used in chunks 25 and 54.
RL7, used in chunks 25 and 54.
RT2, used in chunks 25, 26a, 54, and 55a.
RT2_obs, used in chunks 26, 27, 55, and 56.
STAR, used in chunks 25 and 54.
UC, used in chunks 22, 24b, 25, 51, 53b, and 54.
UHC, used in chunks 23 and 52.
UHK, used in chunks 23 and 52.
UHSC, used in chunks 23 and 52.
UHSK, used in chunks 23 and 52.
UK, used in chunks 22, 24b, 25, 51, 53b, and 54.
unemp, used in chunks 23, 26a, 52, and 55a.
unemp_obs, used in chunks 26, 27, 55, and 56.
var, used in chunks 11, 27, and 56.
WC, used in chunks 22, 23, 51, and 52.
WK, used in chunks 22, 23, 51, and 52.
XiL, used in chunks 23, 25, 52, and 54.
YC, used in chunks 22–24 and 51–53.
YK, used in chunks 22–24 and 51–53.

A.3.2 Linearized Varexo

42 ⟨*linearized varexo* 42⟩≡ (38)

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

Defines:
- `eB`, used in chunks 25, 27, 29, 54, 56, and 59.
- `eEFFECD`, used in chunks 25, 27, 29, 54, 56, and 59.
- `eEFFECH`, used in chunks 25, 27, 29, 54, 56, and 59.
- `eEFFK`, used in chunks 25, 27, 29, 54, 56, and 59.
- `eHG`, used in chunks 25, 27, 29, 54, 56, and 59.
- `eLpref`, used in chunks 25, 27, 29, 54, 56, and 59.
- `eMUZK`, used in chunks 25, 27, 29, 54, 56, and 59.
- `eMUZM`, used in chunks 25, 27, 29, 54, 56, and 59.
- `ePMKC`, used in chunks 22, 27, 29, 51, 56, and 59.
- `ePMKK`, used in chunks 22, 27, 29, 51, 56, and 59.
- `eR`, used in chunks 22, 27, 29, 51, 56, and 59.
- `eSTAR`, used in chunks 25, 27, 29, 54, 56, and 59.
- `eXiL`, used in chunks 23, 25, 27, 29, 52, 54, 56, and 59.

A.3.3 Linearized Parameters

43 ⟨*linearized parameters* 43⟩≡ (38)

```
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 20, 23, 31, 49, 52, and 61.
- A_HK, used in chunks 20, 23, 31, 49, 52, and 61.
- a_ks, used in chunks 19 and 48.
- AA, used in chunks 31 and 61.
- AHSS, used in chunks 24a, 26a, 31, 35, 53, and 61.
- alpha_, used in chunks 19, 22, 24b, 25, 31, 48, 51, 53b, 54, and 61.
- beta_, used in chunks 22–25, 31, 35, 51–54, and 61.
- beta_0, used in chunks 31 and 61.
- beta_2, used in chunks 31 and 61.
- betarl, used in chunks 19, 24b, 48, and 53b.
- DD, used in chunks 31 and 61.
- delta_, used in chunks 19, 22, 31, 48, 51, and 61.
- delta_cd, used in chunks 19, 24c, 31, 48, 53c, and 61.
- delta_ch, used in chunks 19, 25, 31, 48, 54, and 61.
- DIFFREALECDSS, used in chunks 26a, 31, 35, 54, and 61.
- DIFFREALECDSS_obs, used in chunks 26a, 34, 35, 55a, 64, and 65.
- DIFFREALECHSS, used in chunks 26a, 31, 35, 54, and 61.
- DIFFREALECHSS_obs, used in chunks 26a, 34, 35, 55a, 64, and 65.
- DIFFREALECSS, used in chunks 26a, 31, 35, 52, and 61.
- DIFFREALECSS_obs, used in chunks 26a, 34, 35, 55a, 64, and 65.
- DIFFREALEIKSS, used in chunks 26a, 31, 35, 52, and 61.
- DIFFREALEIKSS_obs, used in chunks 26a, 34, 35, 55a, 64, and 65.
- DIFFREALGDPSS, used in chunks 26a, 31, 35, 52, 53b, and 61.
- DIFFREALGDPSS_obs, used in chunks 26a, 34, 35, 55a, 64, and 65.
- DIFFREALWSS, used in chunks 26a, 31, 35, 53a, and 61.
- DIFFREALWSS_obs, used in chunks 26a, 34, 35, 55a, 64, and 65.
- ECDSS, used in chunks 24c, 31, 35, 52–54, and 61.
- ECHSS, used in chunks 25, 31, 35, 52, 54, and 61.
- ECSS, used in chunks 31, 35, 51, 52, and 61.
- EIKSS, used in chunks 23, 31, 35, 51, 52, and 61.
- empCSS, used in chunks 31, 35, 52, and 61.
- empKSS, used in chunks 31, 35, 52, and 61.
- empSCSS, used in chunks 31, 35, 52, and 61.
- empSKSS, used in chunks 31, 35, 52, and 61.

eta_cd, used in chunks 20, 24c, 31, 49, 53c, and 61.
eta_cd_eta_cnn, used in chunks 31 and 61.
eta_ch, used in chunks 20, 25, 31, 49, 54, and 61.
eta_ch_eta_cnn, used in chunks 31 and 61.
eta_cnn, used in chunks 20, 22, 31, 49, 51, and 61.
g_y, used in chunks 19 and 48.
gam_ech, used in chunks 19, 25, 48, and 54.
gam_h, used in chunks 19, 23, 48, and 52.
gam_ic, used in chunks 19, 23, 48, and 52.
gam_icd, used in chunks 19, 24c, 48, and 53c.
gam_pc, used in chunks 18, 22, 28, 47, 51, and 58.
gam_wc, used in chunks 18, 23, 28, 47, 52, and 58.
h, used in chunks 18, 22, 28, 31, 47, 51, 58, and 61.
h_cd, used in chunks 19, 24c, 31, 48, 53c, and 61.
h_ch, used in chunks 19, 25, 31, 48, 54, and 61.
hc_hk, used in chunks 31 and 61.
HCSS, used in chunks 23, 24, 31, 35, 51–53, and 61.
HKSS, used in chunks 23, 24, 31, 35, 51–53, and 61.
HrCSS, used in chunks 31, 35, 52, and 61.
HrKSS, used in chunks 31, 35, 52, and 61.
HrSCSS, used in chunks 31, 35, 52, and 61.
HrSKSS, used in chunks 31, 35, 52, and 61.
HSCSS, used in chunks 23, 31, 35, 52, and 61.
HSKSS, used in chunks 23, 31, 35, 52, and 61.
HSS, used in chunks 31 and 61.
icoef, used in chunks 19 and 48.
IMPHSSS, used in chunks 31 and 61.
INFC10SS, used in chunks 31, 35, 53b, and 61.
INFCNASS, used in chunks 22, 26a, 31, 35, 51, 53b, and 61.
INFCNASS_obs, used in chunks 26a, 34, 35, 55a, 64, and 65.
INFCORSS, used in chunks 26a, 31, 35, 53b, and 61.
INFCORSS_obs, used in chunks 26a, 34, 35, 55a, 64, and 65.
INFCSS, used in chunks 22, 31, 34, 35, 51–53, 61, and 64.
INFGDPSS, used in chunks 31, 35, 53b, and 61.
INFKSS, used in chunks 22, 26a, 31, 34, 35, 51–53, 61, and 64.
INFKSS_obs, used in chunks 26a, 34, 35, 55a, 64, and 65.
INFWCSS, used in chunks 23, 31, 35, 52, 53a, and 61.
INFWKSS, used in chunks 23, 31, 35, 52, 53a, and 61.
KCDSS, used in chunks 31, 35, 53c, 54, and 61.
KCHSS, used in chunks 31, 35, 54, and 61.
KCSS, used in chunks 22, 31, 35, 51, 52, and 61.
KKSS, used in chunks 22, 31, 35, 51, 52, and 61.
LSS, used in chunks 31, 35, 50–54, and 61.
MCCSS, used in chunks 31, 35, 51, and 61.
MCKSS, used in chunks 31, 35, 51, and 61.
mu_, used in chunks 20, 22, 25, 31, 49, 51, 54, and 61.
MUCSS, used in chunks 31, 34, 35, 51–54, 61, and 64.
MUCSShabit, used in chunks 31 and 61.
MUKSS, used in chunks 31, 34, 35, 51–54, 61, and 64.
MUKSShabit, used in chunks 31 and 61.
MUZCSS, used in chunks 31 and 61.
MUZKSS, used in chunks 19, 25, 31, 35, 48, 54, and 61.
MUZMSS, used in chunks 19, 25, 31, 35, 48, 54, and 61.
ONE, used in chunks 19, 31, 35, 48, 51–54, and 61.
pbeta, used in chunks 19, 31, 48, and 61.
phi_cd, used in chunks 18, 24c, 28, 47, 53c, and 58.
phi_ech, used in chunks 18, 25, 28, 47, 54, and 58.

phi_H, used in chunks 18, 23, 28, 47, 52, and 58.
phi_ic, used in chunks 18, 23, 28, 47, 52, and 58.
phi_pc, used in chunks 18, 22, 28, 47, 51, and 58.
phi_u, used in chunks 19, 22, 25, 48, 51, and 54.
phi_wc, used in chunks 18, 23, 28, 47, 52, and 58.
PKBSS, used in chunks 31, 35, 51–53, and 61.
PYSS, used in chunks 31 and 61.
QCDSS, used in chunks 31, 35, 53c, and 61.
QCHSS, used in chunks 31, 35, 54, and 61.
QKSS, used in chunks 31, 35, 51, 52, 54, and 61.
r_dinf, used in chunks 19, 22, 48, and 51.
r_dy, used in chunks 19 and 48.
r_inf, used in chunks 18, 22, 28, 47, 51, and 58.
r_y, used in chunks 18, 22, 28, 47, 51, and 58.
RCDSS, used in chunks 31, 35, 53c, and 61.
RCHSS, used in chunks 31, 35, 54, and 61.
RCSS, used in chunks 31, 35, 51, 54, and 61.
rho_B, used in chunks 18, 25, 28, 47, 54, and 58.
rho_EFFECD, used in chunks 18, 25, 28, 47, 54, and 58.
rho_EFFECH, used in chunks 18, 25, 28, 47, 54, and 58.
rho_EFFK, used in chunks 18, 25, 28, 47, 54, and 58.
rho_HG, used in chunks 18, 25, 28, 47, 54, and 58.
rho_lpref, used in chunks 18, 25, 28, 47, 54, and 58.
rho_MUZK, used in chunks 19 and 48.
rho_MUZM, used in chunks 19 and 48.
rho_R, used in chunks 18, 22, 28, 47, 51, and 58.
rho_STAR, used in chunks 18, 25, 28, 47, 54, and 58.
rho_XiL, used in chunks 18, 25, 28, 47, 54, and 58.
RKSS, used in chunks 31, 35, 51, 54, and 61.
RL1SS, used in chunks 31, 35, 54, and 61.
RL2SS, used in chunks 31, 35, 54, and 61.
RL3SS, used in chunks 31, 35, 54, and 61.
RL4SS, used in chunks 31, 35, 54, and 61.
RL5SS, used in chunks 31, 35, 54, and 61.
RL6SS, used in chunks 31, 35, 54, and 61.
RL7SS, used in chunks 31, 35, 54, and 61.
Rnr, used in chunks 31 and 61.
rpr, used in chunks 19, 22, 31, 48, 51, and 61.
RR, used in chunks 31 and 61.
RSS, used in chunks 22, 26a, 31, 34, 35, 51, 54, 61, and 64.
RSS_obs, used in chunks 26a, 34, 35, 55a, 64, and 65.
RT2SS, used in chunks 26a, 31, 34, 35, 54, 61, and 64.
RT2SS_obs, used in chunks 26a, 34, 35, 55a, 64, and 65.
s_AS, used in chunks 19, 31, 48, and 61.
s_c_ech, used in chunks 31 and 61.
s_ecdc, used in chunks 20, 24b, 31, 34, 49, 53b, 61, and 64.
s_k, used in chunks 20, 23, 24b, 31, 34, 49, 52, 53b, 61, and 64.
s_k_ecd, used in chunks 31 and 61.
s_k_eik, used in chunks 31 and 61.
s_yc, used in chunks 31 and 61.
sig_B, used in chunks 18, 27, 47, and 56.
sig_EFFECD, used in chunks 18, 27, 47, and 56.
sig_EFFECH, used in chunks 18, 27, 47, and 56.
sig_EFFK, used in chunks 18, 27, 47, and 56.
sig_HG, used in chunks 18, 27, 47, and 56.
sig_lpref, used in chunks 18, 27, 47, and 56.
sig_MUZK, used in chunks 18, 27, 47, and 56.

sig_MUZM, used in chunks 18, 27, 47, and 56.
sig_PMKC, used in chunks 18, 27, 47, and 56.
sig_PMKK, used in chunks 18, 27, 47, and 56.
sig_R, used in chunks 18, 27, 47, and 56.
sig_STAR, used in chunks 18, 27, 47, and 56.
sig_XiL, used in chunks 18, 27, 47, and 56.
sigmah, used in chunks 18, 23, 28, 31, 47, 52, 58, and 61.
sigman, used in chunks 18, 23, 28, 31, 47, 52, 58, and 61.
theta_c, used in chunks 19, 22, 31, 48, 51, and 61.
theta_k, used in chunks 19, 22, 31, 48, 51, and 61.
theta_wc, used in chunks 20, 23, 31, 49, 52, and 61.
theta_wk, used in chunks 20, 23, 31, 49, 52, and 61.
tp2, used in chunks 18, 25, 28, 31, 47, 54, 58, and 61.
UCSS, used in chunks 31 and 61.
UHCSS, used in chunks 31, 35, 52, and 61.
UHKSS, used in chunks 31, 35, 52, and 61.
UHSCSS, used in chunks 31, 35, 52, and 61.
UHSKSS, used in chunks 31, 35, 52, and 61.
UKSS, used in chunks 31 and 61.
unempSS, used in chunks 19, 26a, 31, 34, 35, 48, 52, 61, and 64.
unempSS_obs, used in chunks 26a, 34, 35, 55a, 64, and 65.
USS, used in chunks 24b, 31, 35, 51, 53b, 54, and 61.
WCSS, used in chunks 31, 35, 51, 52, and 61.
WKSS, used in chunks 31, 35, 51, 52, and 61.
xsi_HrC, used in chunks 20, 23, 31, 49, 52, and 61.
xsi_HrK, used in chunks 20, 23, 31, 49, 52, and 61.
xsi_NC, used in chunks 20, 23, 31, 49, 52, and 61.
xsi_NK, used in chunks 20, 23, 31, 49, 52, and 61.
ycbi, used in chunks 31 and 61.
ycbi_ykb, used in chunks 31 and 61.
YCSS, used in chunks 22–24, 31, 35, 51–53, and 61.
ykb, used in chunks 31 and 61.
YKSS, used in chunks 22–24, 31, 35, 51–53, and 61.
YYSS, used in chunks 31 and 61.

A.3.4 Linearized Estimated Params

47 ⟨*linearized estimated params* 47⟩≡ (38)

```
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 14 43, gam_wc 14 43, h 14 43, phi_cd 14 43, phi_ech 14 43, phi_H 14 43, phi_ic 14 43, phi_pc 14 43, phi_wc 14 43, r_inf 14 43, r_y 14 43, rho_B 14 43, rho_EFFECD 14 43, rho_EFFECH 14 43, rho_EFFK 14 43, rho_HG 14 43, rho_lpref 14 43, rho_R 14 43, rho_STAR 14 43, rho_XiL 14 43, sig_B 14 43, sig_EFFECD 14 43, sig_EFFECH 14 43, sig_EFFK 14 43, sig_HG 14 43, sig_lpref 14 43, sig_MUZK 14 43, sig_MUZM 14 43, sig_PMKC 14 43, sig_PMKK 14 43, sig_R 14 43, sig_STAR 14 43, sig_XiL 14 43, sigmah 14 43, sigman 14 43, and tp2 14 43.

A.3.5 Linearized Calibrated params

48 ⟨*linearized calibrated params* 48⟩≡ (38)

```
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` 14 43, `alpha_` 14 43, `betarl` 14 43, `delta_` 14 43, `delta_cd` 14 43, `delta_ch` 14 43, `g_y` 14 43, `gam_ech` 14 43, `gam_h` 14 43, `gam_ic` 14 43, `gam_icd` 14 43, `h_cd` 14 43, `h_ch` 14 43, `icoef` 14 43, `MUZKSS` 14 43, `MUZMSS` 14 43, `ONE` 14 43, `pbeta` 14 43, `phi_u` 14 43, `r_dinf` 14 43, `r_dy` 14 43, `rho_MUZK` 14 43, `rho_MUZM` 14 43, `rpr` 14 43, `s_AS` 14 43, `theta_c` 14 43, `theta_k` 14 43, and `unempSS` 14 43.

A.3.6 Linearized Free Params

49 ⟨*linearized free params* 49⟩≡ (38)

```
//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` 14 43, `A_HK` 14 43, `eta_cd` 14 43, `eta_ch` 14 43, `eta_cnn` 14 43, `mu_` 14 43, `s_ecdc` 14 43, `s_k` 14 43, `theta_wc` 14 43, `theta_wk` 14 43, `xsi_HrC` 14 43, `xsi_HrK` 14 43, `xsi_NC` 14 43, and `xsi_NK` 14 43.

A.3.7 Linearized Calibrated Measured Equations

50 ⟨*linearized calibrated ME* 50⟩≡ (38)

```
//calibrated ME


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

model;
⟨linearized cme model block 51⟩

// XXXXXXXXXXXXXXXXXXXXXXXXXXXX
// labor block
// TOTAL LABOR INPUT (called "(LSS*exp(L))" in the paper, I kept the "H" notation of
⟨linearized cme model labor 52⟩

// Identities
⟨linearized cme identities 53a⟩


// XXXXXXXXXXXXXXXXXXXXX
// Aggregate hours equals agg hours in each sector
⟨linearized cme hours 53b⟩

// See Section 8: Data Identities

// new equations
// Durable Block

⟨linearized cme data identities 53c⟩

// Housing Block
⟨linearized cme housing 54⟩

//measurement_equations;
⟨linearized cme measurement 55a⟩

//end_measurement_equations;

end;
```

Uses `L` 11 40 and `LSS` 14 43.

A.3.8 Linearized Calibrated ME Model BLock

51 ⟨*linearized cme model block* 51⟩≡ (50)

```
(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_ 14 43, beta_ 14 43, betas 11 40, delta_ 14 43, EC 11 40, ECSS 14 43, EFFK 11 40, EIK 11 40, EIKSS 14 43, ePMKC 13 42, ePMKK 13 42, eR 13 42, eta_cnn 14 43, gam_pc 14 43, h 14 43, HC 11 40, HCSS 14 43, HK 11 40, HKSS 14 43, INFC 11 40, INFCNA 11 40, INFCNASS 14 43, INFCSS 14 43, INFK 11 40, INFKSS 14 43, KC 11 40, KCSS 14 43, KK 11 40, KKSS 14 43, L 11 40, LSS 14 43, MCC 11 40, MCCSS 14 43, MCK 11 40, MCKSS 14 43, mu_ 14 43, MUC 11 40, MUCSS 14 43, MUK 11 40, MUKSS 14 43, ONE 14 43, PFGAP 11 40, phi_pc 14 43, phi_u 14 43, PKB 11 40, PKBSS 14 43, QK 11 40, QKSS 14 43, R 11 40, r_dinf 14 43, r_inf 14 43, r_y 14 43, RC 11 40, RCSS 14 43, rho_R 14 43, RK 11 40, RKSS 14 43, rpr 14 43, RSS 14 43, theta_c 14 43, theta_k 14 43, UC 11 40, UK 11 40, USS 14 43, WC 11 40, WCSS 14 43, WK 11 40, WKSS 14 43, YC 11 40, YCSS 14 43, YK 11 40, and YKSS 14 43.

A.3.9 Linearized Calibrated ME Labor BLock

52 ⟨*linearized cme model labor* 52⟩≡ (50)

```
-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 14 43, A_HK 14 43, beta_ 14 43, DIFFNORMGDP 11 40, DIFFREALEC 11 40, DIFFREALECSS 14 43, DIFFREALEIK 11 40, DIFFREALEIKSS 14 43, DIFFREALGDP 11 40, DIFFREALGDPSS 14 43, EC 11 40, ECD 11 40, ECDSS 14 43, ECH 11 40, ECHSS 14 43, ECSS 14 43, EFFK 11 40, EIK 11 40, EIKSS 14 43, empC 11 40, empCSS 14 43, empK 11 40, empKSS 14 43, empSC 11 40, empSCSS 14 43, empSK 11 40, empSKSS 14 43, eXiL 13 42, gam_h 14 43, gam_ic 14 43, gam_wc 14 43, HC 11 40, HCSS 14 43, HG 11 40, HK 11 40, HKSS 14 43, HrC 11 40, HrCSS 14 43, HrK 11 40, HrKSS 14 43, HrSC 11 40, HrSCSS 14 43, HrSK 11 40, HrSKSS 14 43, HSC 11 40, HSCSS 14 43, HSK 11 40, HSKSS 14 43, INFC 11 40, INFCSS 14 43, INFK 11 40, INFKSS 14 43, INFWC 11 40, INFWCSS 14 43, INFWK 11 40, INFWKSS 14 43, KC 11 40, KCSS 14 43, KK 11 40, KKSS 14 43, L 11 40, Lpref 11 40, LSS 14 43, MUC 11 40, MUCSS 14 43, MUK 11 40, MUKSS 14 43, NORMINFGDP 11 40, ONE 14 43, phi_H 14 43, phi_ic 14 43, phi_wc 14 43, PKB 11 40, PKBSS 14 43, QK 11 40, QKSS 14 43, s_k 14 43, sigmah 14 43, sigman 14 43, theta_wc 14 43, theta_wk 14 43, UHC 11 40, UHCSS 14 43, UHK 11 40, UHKSS 14 43, UHSC 11 40, UHSCSS 14 43, UHSK 11 40, UHSKSS 14 43, unemp 11 40, unempSS 14 43, WC 11 40, WCSS 14 43, WK 11 40, WKSS 14 43, XiL 11 40, xsi_HrC 14 43, xsi_HrK 14 43, xsi_NC 14 43, xsi_NK 14 43, YC 11 40, YCSS 14 43, YK 11 40, and YKSS 14 43.

A.3.10 Linearized Calibrated ME Identities

53a ⟨*linearized cme identities* 53a⟩≡ (50)

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

Uses AHSS 14 43, DIFFREALW 11 40, DIFFREALWSS 14 43, HCSS 14 43, HKSS 14 43, INFC 11 40, INFCSS 14 43, INFWC 11 40, INFWCSS 14 43, INFWK 11 40, and INFWKSS 14 43.

A.3.11 Linearized Calibrated ME Hours

53b ⟨*linearized cme hours* 53b⟩≡ (50)

```
(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 11 40, AHSS 14 43, alpha_ 14 43, betarl 14 43, DIFFREALGDP 11 40, DIFFREALGDPSS 14 43, GAP 11 40, HC 11 40, HCSS 14 43, HK 11 40, HKSS 14 43, INFC 11 40, INFC10 11 40, INFC10SS 14 43, INFCNA 11 40, INFCNASS 14 43, INFCOR 11 40, INFCORSS 14 43, INFCSS 14 43, INFGDP 11 40, INFGDPSS 14 43, INFK 11 40, INFKSS 14 43, MUC 11 40, MUCSS 14 43, ONE 14 43, PFGAP 11 40, PKB 11 40, PKBSS 14 43, s_ecdc 14 43, s_k 14 43, UC 11 40, UK 11 40, USS 14 43, YC 11 40, YCSS 14 43, YK 11 40, and YKSS 14 43.

A.3.12 Linearized Calibrated ME Data Itentities

53c ⟨*linearized cme data identities* 53c⟩≡ (50)

```
(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_ 14 43, delta_cd 14 43, ECD 11 40, ECDSS 14 43, EFFECD 11 40, eta_cd 14 43, gam_icd 14 43, h_cd 14 43, KCDSS 14 43, KD 11 40, L 11 40, LAGKD 11 40, LSS 14 43, MUK 11 40, MUKSS 14 43, ONE 14 43, phi_cd 14 43, PKB 11 40, PKBSS 14 43, QCD 11 40, QCDSS 14 43, RCD 11 40, and RCDSS 14 43.

A.3.13 Linearized Calibrated ME Housing

54 ⟨*linearized cme housing* 54⟩≡ (50)

```
(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*exp
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-g
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((EC
ln((DIFFREALECDSS*exp(DIFFREALECD)))-ln((MUKSS*exp(MUK)))-ln((ECDSS*exp(ECD)))+ln((EC
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((RL2
```

Uses `alpha_` 14 43, `beta_` 14 43, `betas` 11 40, `delta_ch` 14 43, `DIFFREALECD` 11 40, `DIFFREALECDSS` 14 43, `DIFFREALECH` 11 40, `DIFFREALECHSS` 14 43, `eB` 13 42, `ECD` 11 40, `ECDSS` 14 43, `ECH` 11 40, `ECHSS` 14 43, `eEFFECD` 13 42, `eEFFECH` 13 42, `eEFFK` 13 42, `EFFECD` 11 40, `EFFECH` 11 40, `EFFK` 11 40, `eHG` 13 42, `eLpref` 13 42, `eMUZK` 13 42, `eMUZM` 13 42, `eSTAR` 13 42, `eta_ch` 14 43, `eXiL` 13 42, `gam_ech` 14 43, `h_ch` 14 43, `HG` 11 40, `KCDSS` 14 43, `KCH` 11 40, `KCHSS` 14 43, `KD` 11 40, `L` 11 40, `LAGKCH` 11 40, `LAGKD` 11 40, `Lpref` 11 40, `LSS` 14 43, `mu_` 14 43, `MUC` 11 40, `MUCSS` 14 43, `MUK` 11 40, `MUKSS` 14 43, `MUZK` 11 40, `MUZKSS` 14 43, `MUZM` 11 40, `MUZMSS` 14 43, `ONE` 14 43, `phi_ech` 14 43, `phi_u` 14 43, `QCH` 11 40, `QCHSS` 14 43, `QK` 11 40, `QKSS` 14 43, `R` 11 40, `RC` 11 40, `RCH` 11 40, `RCHSS` 14 43, `RCSS` 14 43, `rho_B` 14 43, `rho_EFFECD` 14 43, `rho_EFFECH` 14 43, `rho_EFFK` 14 43, `rho_HG` 14 43, `rho_lpref` 14 43, `rho_STAR` 14 43, `rho_XiL` 14 43, `RK` 11 40, `RKSS` 14 43, `RL1` 11 40, `RL1SS` 14 43, `RL2` 11 40, `RL2SS` 14 43, `RL3` 11 40, `RL3SS` 14 43, `RL4` 11 40, `RL4SS` 14 43, `RL5` 11 40, `RL5SS` 14 43, `RL6` 11 40, `RL6SS` 14 43, `RL7` 11 40, `RL7SS` 14 43, `RSS` 14 43, `RT2` 11 40, `RT2SS` 14 43, `STAR` 11 40, `tp2` 14 43, `UC` 11 40, `UK` 11 40, `USS` 14 43, and `XiL` 11 40.

A.3.14 Linearized Calibrated ME Measurement

55a ⟨*linearized cme measurement* 55a⟩≡ (50)

```
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 11 40, AH_obs 11 40, DIFFREALEC 11 40, DIFFREALEC_obs 11 40, DIFFREALECD 11 40, DIFFREALECD_obs 11 40, DIFFREALECDSS_obs 14 43, DIFFREALECH 11 40, DIFFREALECH_obs 11 40, DIFFREALECHSS_obs 14 43, DIFFREALECSS_obs 14 43, DIFFREALEIK 11 40, DIFFREALEIK_obs 11 40, DIFFREALEIKSS_obs 14 43, DIFFREALGDP 11 40, DIFFREALGDP_obs 11 40, DIFFREALGDPSS_obs 14 43, DIFFREALW 11 40, DIFFREALW_obs 11 40, DIFFREALWSS_obs 14 43, INFCNA 11 40, INFCNA_obs 11 40, INFCNASS_obs 14 43, INFCOR 11 40, INFCOR_obs 11 40, INFCORSS_obs 14 43, INFK 11 40, INFK_obs 11 40, INFKSS_obs 14 43, R 11 40, R_obs 11 40, RSS_obs 14 43, RT2 11 40, RT2_obs 11 40, RT2SS_obs 14 43, unemp 11 40, unemp_obs 11 40, and unempSS_obs 14 43.

A.3.15 Linearized Varobs

55b ⟨*linearized varobs* 55b⟩≡ (38)

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

Uses AH_obs 11 40, DIFFREALEC_obs 11 40, DIFFREALECD_obs 11 40, DIFFREALECH_obs 11 40, DIFFREALEIK_obs 11 40, DIFFREALGDP_obs 11 40, DIFFREALW_obs 11 40, INFCNA_obs 11 40, INFCOR_obs 11 40, INFK_obs 11 40, R_obs 11 40, RT2_obs 11 40, and unemp_obs 11 40.

A.3.16 Linearized Shocks

56 ⟨*linearized shocks* 56⟩≡ (38)

```
shocks;
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 11 40, DIFFREALEC_obs 11 40, DIFFREALECD_obs 11 40, DIFFREALECH_obs 11 40, DIFFREALEIK_obs 11 40, DIFFREALGDP_obs 11 40, DIFFREALW_obs 11 40, eB 13 42, eEFFECD 13 42, eEFFECH 13 42, eEFFK 13 42, eHG 13 42, eLpref 13 42, eMUZK 13 42, eMUZM 13 42, ePMKC 13 42, ePMKK 13 42, eR 13 42, eSTAR 13 42, eXiL 13 42, INFCNA_obs 11 40, INFCOR_obs 11 40, INFK_obs 11 40, RT2_obs 11 40, sig_B 14 43, sig_EFFECD 14 43, sig_EFFECH 14 43, sig_EFFK 14 43, sig_HG 14 43, sig_lpref 14 43, sig_MUZK 14 43, sig_MUZM 14 43, sig_PMKC 14 43, sig_PMKK 14 43, sig_R 14 43, sig_STAR 14 43, sig_XiL 14 43, unemp_obs 11 40, and var 40.

A.3.17 Linearized Estimated params

58 ⟨*linearized steady estimated params* 58⟩≡ (38)

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

Uses gam_pc 14 43, gam_wc 14 43, h 14 43, phi_cd 14 43, phi_ech 14 43, phi_H 14 43, phi_ic 14 43, phi_pc 14 43, phi_wc 14 43, r_inf 14 43, r_y 14 43, rho_B 14 43, rho_EFFECD 14 43, rho_EFFECH 14 43, rho_EFFK 14 43, rho_HG 14 43, rho_lpref 14 43, rho_R 14 43, rho_STAR 14 43, rho_XiL 14 43, sigmah 14 43, sigman 14 43, and tp2 14 43.

A.3.18 Linearized Stderr

59 ⟨*linearized stderr* 59⟩≡ (38)

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

Uses `eB` 13 42, `eEFFECD` 13 42, `eEFFECH` 13 42, `eEFFK` 13 42, `eHG` 13 42, `eLpref` 13 42, `eMUZK` 13 42, `eMUZM` 13 42, `ePMKC` 13 42, `ePMKK` 13 42, `eR` 13 42, `eSTAR` 13 42, and `eXiL` 13 42.

A.4 linearized_steadystate.m

60 ⟨*srcedo/linearized.steadystate.m* 60⟩≡

```
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;

⟨linearized steady state values 61⟩

%end_steady_state;

%trends;

⟨linearized steady state trends 64⟩

%end_trends;


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

⟨linearized steady state result return 65⟩
```

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

A.4.1 Linearized Steady State Values

61 ⟨*linearized steady state values* 61⟩≡ (60)

```
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+sigman
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+sigman
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+sigman
xsi_NK=A_HK/((1/(1+sigman)+1/sigmah)*(HKSS^sigman/HrKSS^(1+sigman+sigmah))^((1+sigman
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
```

```
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 14 43, A_HK 14 43, AA 14 43, AHSS 14 43, alpha_ 14 43, beta_ 14 43, beta_0 14 43, beta_2 14 43, DD 14 43, delta_ 14 43, delta_cd 14 43, delta_ch 14 43, DIFFREALECDSS 14 43, DIFFREALECHSS 14 43, DIFFREALECSS 14 43, DIFFREALEIKSS 14 43, DIFFREALGDPSS 14 43, DIFFREALWSS 14 43, ECDSS 14 43, ECHSS 14 43, ECSS 14 43, EIKSS 14 43, empCSS 14 43, empKSS 14 43, empSCSS 14 43, empSKSS 14 43, eta_cd 14 43, eta_cd_eta_cnn 14 43, eta_ch 14 43, eta_ch_eta_cnn 14 43, eta_cnn 14 43, h 14 43, h_cd 14 43, h_ch 14 43, hc_hk 14 43, HCSS 14 43, HKSS 14 43, HrCSS 14 43, HrKSS 14 43, HrSCSS 14 43, HrSKSS 14 43, HSCSS 14 43, HSKSS 14 43, HSS 14 43, IMPHSSS 14 43, INFC10SS 14 43, INFCNASS 14 43, INFCORSS 14 43, INFCSS 14 43, INFGDPSS 14 43, INFKSS 14 43, INFWCSS 14 43, INFWKSS 14 43, KCDSS 14 43, KCHSS 14 43, KCSS 14 43, KKSS 14 43, LSS 14 43, MCCSS 14 43, MCKSS 14 43, mu_ 14 43, MUCSS 14 43, MUCSShabit 14 43, MUKSS 14 43, MUKSShabit 14 43, MUZCSS 14 43, MUZKSS 14 43, MUZMSS 14 43, ONE 14 43, pbeta 14 43, PKBSS 14 43, PYSS 14 43, QCDSS 14 43, QCHSS 14 43, QKSS 14 43, RCDSS 14 43, RCHSS 14 43, RCSS 14 43, RKSS 14 43, RL1SS 14 43, RL2SS 14 43, RL3SS 14 43, RL4SS 14 43, RL5SS 14 43, RL6SS 14 43, RL7SS 14 43, Rnr 14 43, rpr 14 43, RR 14 43, RSS 14 43, RT2SS 14 43, s_AS 14 43, s_c_ech 14 43, s_ecdc 14 43, s_k 14 43, s_k_ecd 14 43, s_k_eik 14 43, s_yc 14 43, sigmah 14 43, sigman 14 43, theta_c 14 43, theta_k 14 43, theta_wc 14 43, theta_wk 14 43, tp2 14 43, UCSS 14 43, UHCSS 14 43, UHKSS 14 43, UHSCSS 14 43, UHSKSS 14 43, UKSS 14 43, unempSS 14 43, USS 14 43, WCSS 14 43, WKSS 14 43, xsi_HrC 14 43, xsi_HrK 14 43, xsi_NC 14 43, xsi_NK 14 43, ycbi 14 43, ycbi_ykb 14 43, YCSS 14 43, ykb 14 43, YKSS 14 43, and YYSS 14 43.

A.4.2 Linearized Steady State Trends

64 ⟨*linearized steady state trends* 64⟩≡ (60)

```
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 14 43, DIFFREALECHSS_obs 14 43, DIFFREALECSS_obs 14 43, DIFFREALEIKSS_obs 14 43, DIFFREALGDPSS_obs 14 43, DIFFREALWSS_obs 14 43, INFCNASS_obs 14 43, INFCORSS_obs 14 43, INFCSS 14 43, INFKSS 14 43, INFKSS_obs 14 43, MUCSS 14 43, MUKSS 14 43, RSS 14 43, RSS_obs 14 43, RT2SS 14 43, RT2SS_obs 14 43, s_ecdc 14 43, s_k 14 43, unempSS 14 43, and unempSS_obs 14 43.

A.4.3 Linearized Steady State Result Return

65 ⟨*linearized steady state result return* 65⟩≡ (60)

```
  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 14 43, DIFFREALECHSS_obs 14 43, DIFFREALECSS_obs 14 43, DIFFREALEIKSS_obs 14 43, DIFFREALGDPSS_obs 14 43, DIFFREALWSS_obs 14 43, INFCNASS_obs 14 43, INFCORSS_obs 14 43, INFKSS_obs 14 43, RSS_obs 14 43, RT2SS_obs 14 43, and unempSS_obs 14 43.

A.5 readme.txt

68 ⟨*srcedo/readme.txt* 68⟩≡

```
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 th
   "addpath path/to/dynare").
5) Run the command "dynare linearized" or "dynare Dynare_edo" from the Matlab/Octave


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

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

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

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

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

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

⟨*edo calibrated_params* 19⟩
⟨*edo estimated_params* 18⟩
⟨*edo free_params* 20⟩
⟨*edo model* 21⟩
⟨*edo model durables* 24c⟩
⟨*edo model estimated_params* 28⟩
⟨*edo model hours* 24b⟩
⟨*edo model housing* 25⟩
⟨*edo model identities* 24a⟩
⟨*edo model labor* 23⟩
⟨*edo model measurement* 26a⟩
⟨*edo model prelim* 22⟩
⟨*edo model shocks* 27⟩
⟨*edo model stderr* 29⟩
⟨*edo model varobs* 26b⟩
⟨*edo parameters* 14⟩
⟨*edo steady state result return* 35⟩
⟨*edo steady state trends* 34⟩
⟨*edo steady state values* 31⟩
⟨*edo var* 11⟩
⟨*edo varexo* 13⟩
⟨*linearized calibrated ME* 50⟩
⟨*linearized calibrated params* 48⟩
⟨*linearized cme data identities* 53c⟩
⟨*linearized cme hours* 53b⟩
⟨*linearized cme housing* 54⟩

⟨*linearized cme identities* 53a⟩
⟨*linearized cme measurement* 55a⟩
⟨*linearized cme model block* 51⟩
⟨*linearized cme model labor* 52⟩
⟨*linearized estimated params* 47⟩
⟨*linearized free params* 49⟩
⟨*linearized parameters* 43⟩
⟨*linearized shocks* 56⟩
⟨*linearized stderr* 59⟩
⟨*linearized steady estimated params* 58⟩
⟨*linearized steady state result return* 65⟩
⟨*linearized steady state trends* 64⟩
⟨*linearized steady state values* 61⟩
⟨*linearized var* 40⟩
⟨*linearized varexo* 42⟩
⟨*linearized varobs* 55b⟩
⟨*srcedo/Dynare.edo.mod* 9⟩
⟨*srcedo/Dynare.edo.steadystate.m* 30⟩
⟨*srcedo/linearized.mod* 38⟩
⟨*srcedo/linearized.steadystate.m* 60⟩
⟨*srcedo/readme.txt* 68⟩

B.2 Index

`A_HC`: 14, 20, 23, 31, 43, 49, 52, 61
`A_HK`: 14, 20, 23, 31, 43, 49, 52, 61
`a_ks`: 14, 19, 43, 48
`AA`: 14, 31, 43, 61
`AH`: 11, 24b, 26a, 40, 53b, 55a
`AH_obs`: 11, 26a, 26b, 27, 40, 55a, 55b, 56
`AHSS`: 14, 24a, 26a, 31, 35, 43, 53a, 53b, 61
`alpha_`: 14, 19, 22, 24b, 25, 31, 43, 48, 51, 53b, 54, 61
`beta_`: 14, 22, 23, 24c, 25, 31, 35, 43, 51, 52, 53c, 54, 61
`beta_0`: 14, 31, 43, 61
`beta_2`: 14, 31, 43, 61
`betarl`: 14, 19, 24b, 43, 48, 53b
`betas`: 11, 22, 25, 40, 51, 54
`DD`: 14, 31, 43, 61
`delta_`: 14, 19, 22, 31, 43, 48, 51, 61
`delta_cd`: 14, 19, 24c, 31, 43, 48, 53c, 61
`delta_ch`: 14, 19, 25, 31, 43, 48, 54, 61
`DIFFNORMGDP`: 11, 23, 40, 52
`DIFFREALEC`: 11, 23, 26a, 40, 52, 55a
`DIFFREALEC_obs`: 11, 26a, 26b, 27, 40, 55a, 55b, 56
`DIFFREALECD`: 11, 25, 26a, 40, 54, 55a

DIFFREALECD_obs: 11, 26a, 26b, 27, 40, 55a, 55b, 56
DIFFREALECDSS: 14, 26a, 31, 35, 43, 54, 61
DIFFREALECDSS_obs: 14, 26a, 34, 35, 43, 55a, 64, 65
DIFFREALECH: 11, 25, 26a, 40, 54, 55a
DIFFREALECH_obs: 11, 26a, 26b, 27, 40, 55a, 55b, 56
DIFFREALECHSS: 14, 26a, 31, 35, 43, 54, 61
DIFFREALECHSS_obs: 14, 26a, 34, 35, 43, 55a, 64, 65
DIFFREALECSS: 14, 26a, 31, 35, 43, 52, 61
DIFFREALECSS_obs: 14, 26a, 34, 35, 43, 55a, 64, 65
DIFFREALEIK: 11, 23, 26a, 40, 52, 55a
DIFFREALEIK_obs: 11, 26a, 26b, 27, 40, 55a, 55b, 56
DIFFREALEIKSS: 14, 26a, 31, 35, 43, 52, 61
DIFFREALEIKSS_obs: 14, 26a, 34, 35, 43, 55a, 64, 65
DIFFREALGDP: 11, 23, 24b, 26a, 40, 52, 53b, 55a
DIFFREALGDP_obs: 11, 26a, 26b, 27, 40, 55a, 55b, 56
DIFFREALGDPSS: 14, 26a, 31, 35, 43, 52, 53b, 61
DIFFREALGDPSS_obs: 14, 26a, 34, 35, 43, 55a, 64, 65
DIFFREALW: 11, 24a, 26a, 40, 53a, 55a
DIFFREALW_obs: 11, 26a, 26b, 27, 40, 55a, 55b, 56
DIFFREALWSS: 14, 26a, 31, 35, 43, 53a, 61
DIFFREALWSS_obs: 14, 26a, 34, 35, 43, 55a, 64, 65
eB: 13, 25, 27, 29, 42, 54, 56, 59
EC: 11, 22, 23, 40, 51, 52
ECD: 11, 23, 24c, 25, 40, 52, 53c, 54
ECDSS: 14, 24c, 31, 35, 43, 52, 53c, 54, 61
ECH: 11, 23, 25, 40, 52, 54
ECHSS: 14, 25, 31, 35, 43, 52, 54, 61
ECSS: 14, 31, 35, 43, 51, 52, 61
eEFFECD: 13, 25, 27, 29, 42, 54, 56, 59
eEFFECH: 13, 25, 27, 29, 42, 54, 56, 59
eEFFK: 13, 25, 27, 29, 42, 54, 56, 59
EFFECD: 11, 24c, 25, 40, 53c, 54
EFFECH: 11, 25, 40, 54
EFFK: 11, 22, 23, 25, 40, 51, 52, 54
eHG: 13, 25, 27, 29, 42, 54, 56, 59
EIK: 11, 22, 23, 40, 51, 52
EIKSS: 14, 23, 31, 35, 43, 51, 52, 61
eLpref: 13, 25, 27, 29, 42, 54, 56, 59
empC: 11, 23, 40, 52
empCSS: 14, 31, 35, 43, 52, 61
empK: 11, 23, 40, 52
empKSS: 14, 31, 35, 43, 52, 61
empSC: 11, 23, 40, 52
empSCSS: 14, 31, 35, 43, 52, 61
empSK: 11, 23, 40, 52
empSKSS: 14, 31, 35, 43, 52, 61

eMUZK: 13, 25, 27, 29, 42, 54, 56, 59
eMUZM: 13, 25, 27, 29, 42, 54, 56, 59
ePMKC: 13, 22, 27, 29, 42, 51, 56, 59
ePMKK: 13, 22, 27, 29, 42, 51, 56, 59
eR: 13, 22, 27, 29, 42, 51, 56, 59
eSTAR: 13, 25, 27, 29, 42, 54, 56, 59
eta_cd: 14, 20, 24c, 31, 43, 49, 53c, 61
eta_cd_eta_cnn: 14, 31, 43, 61
eta_ch: 14, 20, 25, 31, 43, 49, 54, 61
eta_ch_eta_cnn: 14, 31, 43, 61
eta_cnn: 14, 20, 22, 31, 43, 49, 51, 61
eXiL: 13, 23, 25, 27, 29, 42, 52, 54, 56, 59
g_y: 14, 19, 43, 48
gam_ech: 14, 19, 25, 43, 48, 54
gam_h: 14, 19, 23, 43, 48, 52
gam_ic: 14, 19, 23, 43, 48, 52
gam_icd: 14, 19, 24c, 43, 48, 53c
gam_pc: 14, 18, 22, 28, 43, 47, 51, 58
gam_wc: 14, 18, 23, 28, 43, 47, 52, 58
GAP: 11, 24b, 40, 53b
h: 14, 18, 22, 28, 31, 43, 47, 51, 58, 61
h_cd: 14, 19, 24c, 31, 43, 48, 53c, 61
h_ch: 14, 19, 25, 31, 43, 48, 54, 61
HC: 11, 22, 23, 24b, 40, 51, 52, 53b
hc_hk: 14, 31, 43, 61
HCSS: 14, 23, 24a, 24b, 31, 35, 43, 51, 52, 53a, 53b, 61
HG: 11, 23, 25, 40, 52, 54
HK: 11, 22, 23, 24b, 40, 51, 52, 53b
HKSS: 14, 23, 24a, 24b, 31, 35, 43, 51, 52, 53a, 53b, 61
HrC: 11, 23, 40, 52
HrCSS: 14, 31, 35, 43, 52, 61
HrK: 11, 23, 40, 52
HrKSS: 14, 31, 35, 43, 52, 61
HrSC: 11, 23, 40, 52
HrSCSS: 14, 31, 35, 43, 52, 61
HrSK: 11, 23, 40, 52
HrSKSS: 14, 31, 35, 43, 52, 61
HSC: 11, 23, 40, 52
HSCSS: 14, 23, 31, 35, 43, 52, 61
HSK: 11, 23, 40, 52
HSKSS: 14, 23, 31, 35, 43, 52, 61
HSS: 14, 31, 43, 61
icoef: 14, 19, 43, 48
IMPHSSS: 14, 31, 43, 61
INFC: 11, 22, 23, 24a, 24b, 40, 51, 52, 53a, 53b
INFC10: 11, 24b, 40, 53b

INFC10SS: _14_, 31, 35, _43_, 53b, 61
INFCNA: _11_, 22, 24b, 26a, _40_, 51, 53b, 55a
INFCNA_obs: _11_, 26a, 26b, 27, _40_, 55a, 55b, 56
INFCNASS: _14_, 22, 26a, 31, 35, _43_, 51, 53b, 61
INFCNASS_obs: _14_, 26a, 34, 35, _43_, 55a, 64, 65
INFCOR: _11_, 24b, 26a, _40_, 53b, 55a
INFCOR_obs: _11_, 26a, 26b, 27, _40_, 55a, 55b, 56
INFCORSS: _14_, 26a, 31, 35, _43_, 53b, 61
INFCORSS_obs: _14_, 26a, 34, 35, _43_, 55a, 64, 65
INFCSS: _14_, 22, 31, 34, 35, _43_, 51, 52, 53a, 53b, 61, 64
INFGDP: _11_, 24b, _40_, 53b
INFGDPSS: _14_, 31, 35, _43_, 53b, 61
INFK: _11_, 22, 23, 24b, 26a, _40_, 51, 52, 53b, 55a
INFK_obs: _11_, 26a, 26b, 27, _40_, 55a, 55b, 56
INFKSS: _14_, 22, 26a, 31, 34, 35, _43_, 51, 52, 53b, 61, 64
INFKSS_obs: _14_, 26a, 34, 35, _43_, 55a, 64, 65
INFWC: _11_, 23, 24a, _40_, 52, 53a
INFWCSS: _14_, 23, 31, 35, _43_, 52, 53a, 61
INFWK: _11_, 23, 24a, _40_, 52, 53a
INFWKSS: _14_, 23, 31, 35, _43_, 52, 53a, 61
jacobian_flag: _9_, 38
KC: _11_, 22, 23, _40_, 51, 52
KCDSS: _14_, 31, 35, _43_, 53c, 54, 61
KCH: _11_, 25, _40_, 54
KCHSS: _14_, 31, 35, _43_, 54, 61
KCSS: _14_, 22, 31, 35, _43_, 51, 52, 61
KD: _11_, 24c, 25, _40_, 53c, 54
KK: _11_, 22, 23, _40_, 51, 52
KKSS: _14_, 22, 31, 35, _43_, 51, 52, 61
L: _11_, 21, 22, 23, 24c, 25, _40_, 50, 51, 52, 53c, 54
LAGKCH: _11_, 25, _40_, 54
LAGKD: _11_, 24c, 25, _40_, 53c, 54
Lpref: _11_, 23, 25, _40_, 52, 54
LSS: _14_, 31, 35, _43_, 50, 51, 52, 53c, 54, 61
MCC: _11_, 22, _40_, 51
MCCSS: _14_, 31, 35, _43_, 51, 61
MCK: _11_, 22, _40_, 51
MCKSS: _14_, 31, 35, _43_, 51, 61
mu_: _14_, 20, 22, 25, 31, _43_, 49, 51, 54, 61
MUC: _11_, 22, 23, 24b, 25, _40_, 51, 52, 53b, 54
MUCSS: _14_, 31, 34, 35, _43_, 51, 52, 53b, 54, 61, 64
MUCSShabit: _14_, 31, _43_, 61
MUK: _11_, 22, 23, 24c, 25, _40_, 51, 52, 53c, 54
MUKSS: _14_, 31, 34, 35, _43_, 51, 52, 53c, 54, 61, 64
MUKSShabit: _14_, 31, _43_, 61
MUZCSS: _14_, 31, _43_, 61

MUZK: 11, 25, 40, 54
MUZKSS: 14, 19, 25, 31, 35, 43, 48, 54, 61
MUZM: 11, 25, 40, 54
MUZMSS: 14, 19, 25, 31, 35, 43, 48, 54, 61
nonlin: 9, 38
NORMINFGDP: 11, 23, 40, 52
ONE: 14, 19, 31, 35, 43, 48, 51, 52, 53b, 53c, 54, 61
options_: 9, 38
order: 9, 38
pbeta: 14, 19, 31, 43, 48, 61
PFGAP: 11, 22, 24b, 40, 51, 53b
phi_cd: 14, 18, 24c, 28, 43, 47, 53c, 58
phi_ech: 14, 18, 25, 28, 43, 47, 54, 58
phi_H: 14, 18, 23, 28, 43, 47, 52, 58
phi_ic: 14, 18, 23, 28, 43, 47, 52, 58
phi_pc: 14, 18, 22, 28, 43, 47, 51, 58
phi_u: 14, 19, 22, 25, 43, 48, 51, 54
phi_wc: 14, 18, 23, 28, 43, 47, 52, 58
PKB: 11, 22, 23, 24b, 24c, 40, 51, 52, 53b, 53c
PKBSS: 14, 31, 35, 43, 51, 52, 53b, 53c, 61
PYSS: 14, 31, 43, 61
QCD: 11, 24c, 40, 53c
QCDSS: 14, 31, 35, 43, 53c, 61
QCH: 11, 25, 40, 54
QCHSS: 14, 31, 35, 43, 54, 61
QK: 11, 22, 23, 25, 40, 51, 52, 54
QKSS: 14, 31, 35, 43, 51, 52, 54, 61
R: 11, 22, 25, 26a, 40, 51, 54, 55a
r_dinf: 14, 19, 22, 43, 48, 51
r_dy: 14, 19, 43, 48
r_inf: 14, 18, 22, 28, 43, 47, 51, 58
R_obs: 11, 26a, 26b, 40, 55a, 55b
r_y: 14, 18, 22, 28, 43, 47, 51, 58
RC: 11, 22, 25, 40, 51, 54
RCD: 11, 24c, 40, 53c
RCDSS: 14, 31, 35, 43, 53c, 61
RCH: 11, 25, 40, 54
RCHSS: 14, 31, 35, 43, 54, 61
RCSS: 14, 31, 35, 43, 51, 54, 61
rho_B: 14, 18, 25, 28, 43, 47, 54, 58
rho_EFFECD: 14, 18, 25, 28, 43, 47, 54, 58
rho_EFFECH: 14, 18, 25, 28, 43, 47, 54, 58
rho_EFFK: 14, 18, 25, 28, 43, 47, 54, 58
rho_HG: 14, 18, 25, 28, 43, 47, 54, 58
rho_lpref: 14, 18, 25, 28, 43, 47, 54, 58
rho_MUZK: 14, 19, 43, 48

rho_MUZM: 14, 19, 43, 48
rho_R: 14, 18, 22, 28, 43, 47, 51, 58
rho_STAR: 14, 18, 25, 28, 43, 47, 54, 58
rho_XiL: 14, 18, 25, 28, 43, 47, 54, 58
RK: 11, 22, 25, 40, 51, 54
RKSS: 14, 31, 35, 43, 51, 54, 61
RL1: 11, 25, 40, 54
RL1SS: 14, 31, 35, 43, 54, 61
RL2: 11, 25, 40, 54
RL2SS: 14, 31, 35, 43, 54, 61
RL3: 11, 25, 40, 54
RL3SS: 14, 31, 35, 43, 54, 61
RL4: 11, 25, 40, 54
RL4SS: 14, 31, 35, 43, 54, 61
RL5: 11, 25, 40, 54
RL5SS: 14, 31, 35, 43, 54, 61
RL6: 11, 25, 40, 54
RL6SS: 14, 31, 35, 43, 54, 61
RL7: 11, 25, 40, 54
RL7SS: 14, 31, 35, 43, 54, 61
Rnr: 14, 31, 43, 61
rpr: 14, 19, 22, 31, 43, 48, 51, 61
RR: 14, 31, 43, 61
RSS: 14, 22, 26a, 31, 34, 35, 43, 51, 54, 61, 64
RSS_obs: 14, 26a, 34, 35, 43, 55a, 64, 65
RT2: 11, 25, 26a, 40, 54, 55a
RT2_obs: 11, 26a, 26b, 27, 40, 55a, 55b, 56
RT2SS: 14, 26a, 31, 34, 35, 43, 54, 61, 64
RT2SS_obs: 14, 26a, 34, 35, 43, 55a, 64, 65
s_AS: 14, 19, 31, 43, 48, 61
s_c_ech: 14, 31, 43, 61
s_ecdc: 14, 20, 24b, 31, 34, 43, 49, 53b, 61, 64
s_k: 14, 20, 23, 24b, 31, 34, 43, 49, 52, 53b, 61, 64
s_k_ecd: 14, 31, 43, 61
s_k_eik: 14, 31, 43, 61
s_yc: 14, 31, 43, 61
sig_B: 14, 18, 27, 43, 47, 56
sig_EFFECD: 14, 18, 27, 43, 47, 56
sig_EFFECH: 14, 18, 27, 43, 47, 56
sig_EFFK: 14, 18, 27, 43, 47, 56
sig_HG: 14, 18, 27, 43, 47, 56
sig_lpref: 14, 18, 27, 43, 47, 56
sig_MUZK: 14, 18, 27, 43, 47, 56
sig_MUZM: 14, 18, 27, 43, 47, 56
sig_PMKC: 14, 18, 27, 43, 47, 56
sig_PMKK: 14, 18, 27, 43, 47, 56

sig_R: 14, 18, 27, 43, 47, 56
sig_STAR: 14, 18, 27, 43, 47, 56
sig_XiL: 14, 18, 27, 43, 47, 56
sigmah: 14, 18, 23, 28, 31, 43, 47, 52, 58, 61
sigman: 14, 18, 23, 28, 31, 43, 47, 52, 58, 61
STAR: 11, 25, 40, 54
stoch_simul: 9, 38
theta_c: 14, 19, 22, 31, 43, 48, 51, 61
theta_k: 14, 19, 22, 31, 43, 48, 51, 61
theta_wc: 14, 20, 23, 31, 43, 49, 52, 61
theta_wk: 14, 20, 23, 31, 43, 49, 52, 61
tp2: 14, 18, 25, 28, 31, 43, 47, 54, 58, 61
UC: 11, 22, 24b, 25, 40, 51, 53b, 54
UCSS: 14, 31, 43, 61
UHC: 11, 23, 40, 52
UHCSS: 14, 31, 35, 43, 52, 61
UHK: 11, 23, 40, 52
UHKSS: 14, 31, 35, 43, 52, 61
UHSC: 11, 23, 40, 52
UHSCSS: 14, 31, 35, 43, 52, 61
UHSK: 11, 23, 40, 52
UHSKSS: 14, 31, 35, 43, 52, 61
UK: 11, 22, 24b, 25, 40, 51, 53b, 54
UKSS: 14, 31, 43, 61
unemp: 11, 23, 26a, 40, 52, 55a
unemp_obs: 11, 26a, 26b, 27, 40, 55a, 55b, 56
unempSS: 14, 19, 26a, 31, 34, 35, 43, 48, 52, 61, 64
unempSS_obs: 14, 26a, 34, 35, 43, 55a, 64, 65
unlinearized_edo_steadystate: 30
USS: 14, 24b, 31, 35, 43, 51, 53b, 54, 61
var: 11, 27, 40, 56
WC: 11, 22, 23, 40, 51, 52
WCSS: 14, 31, 35, 43, 51, 52, 61
WK: 11, 22, 23, 40, 51, 52
WKSS: 14, 31, 35, 43, 51, 52, 61
XiL: 11, 23, 25, 40, 52, 54
xsi_HrC: 14, 20, 23, 31, 43, 49, 52, 61
xsi_HrK: 14, 20, 23, 31, 43, 49, 52, 61
xsi_NC: 14, 20, 23, 31, 43, 49, 52, 61
xsi_NK: 14, 20, 23, 31, 43, 49, 52, 61
YC: 11, 22, 23, 24b, 40, 51, 52, 53b
ycbi: 14, 31, 43, 61
ycbi_ykb: 14, 31, 43, 61
YCSS: 14, 22, 23, 24b, 31, 35, 43, 51, 52, 53b, 61
YK: 11, 22, 23, 24b, 40, 51, 52, 53b
ykb: 14, 31, 43, 61

`YKSS`: <u>14</u>, 22, 23, 24b, 31, 35, <u>43</u>, 51, 52, 53b, 61
`YYSS`: <u>14</u>, 31, <u>43</u>, 61