

```

(***** Linear system of rational expectations equations +
expectational errors. Note that Ex_t means E_t x_{t+1}. *****)
Omegah = (lambda_star - beta (1 + lambda_a) gamma_h) (lambda_star - gamma_h);
Omegae = (lambda_star - beta gamma_e) (lambda_star - gamma_e);
Delta_log_lambda_start = (1 - (1 - phi) alpha)^-1 (lambda_z_t + nu_z_t - nu_z_{t-1} + (1 - phi) alpha (lambda_q_t + nu_q_t - nu_q_{t-1})); (* Delta_log lambda_{*,t} *)
Delta_log_lambda_k_t = (1 - (1 - phi) alpha)^-1 (lambda_z_t + nu_z_t - nu_z_{t-1} + lambda_q_t + nu_q_t - nu_q_{t-1}); (* Delta_log lambda_{k,t} *)
E_Delta_log_lambda_start = (1 - (1 - phi) alpha)^-1 (rho_lambda_z lambda_z_t + (rho_nu_z - 1) nu_z_t + (1 - phi) alpha (rho_lambda_q lambda_q_t + (rho_nu_q - 1) nu_q_t));
(* E_t Delta_log lambda_{*,t+1} *)
E_Delta_log_lambda_k_t = (1 - (1 - phi) alpha)^-1 (rho_lambda_z lambda_z_t + (rho_nu_z - 1) nu_z_t + rho_lambda_q lambda_q_t + (rho_nu_q - 1) nu_q_t); (* E_t Delta_log lambda_{k,t+1} *)
REequations = {
(* ----- Household equations ----- *)
(* 1. Marginal utility of consumption. *)
Omegah mu_h_t == - (lambda_star^2 + beta (1 + lambda_a) gamma_h^2) ch_t + lambda_star gamma_h (ch_{t-1} - Delta_log_lambda_start) +
beta (1 + lambda_a) gamma_h lambda_star (Ech_t + E_Delta_log_lambda_start) - beta gamma_h lambda_a (lambda_star - gamma_h) rho_lambda_a lambda_a_t,
(* 2. Residential housing decision *)
ql_t + mu_h_t == (lambda_a / (1 + lambda_a)) rho_lambda_a lambda_a_t + beta (1 + lambda_a) (Eq1_t + E mu_h_t) + (1 - beta (1 + lambda_a)) (rho_phi phi_t - Lh_t),
(* 3. Labor supply *)
w_t + mu_h_t == psi_t + eta n_t,
(* 4. Saving decision *)
mu_h_t - R_t == (lambda_a / (1 + lambda_a)) rho_lambda_a lambda_a_t + E mu_h_t - E_Delta_log_lambda_start,
(* ----- Entrepreneur equations ----- *)
(* 5. Marginal utility of consumption. *)
Omegae mu_e_t ==
- (lambda_star^2 + beta gamma_e^2) ce_t + lambda_star gamma_e (ce_{t-1} - Delta_log_lambda_start) + beta gamma_e lambda_star (Ece_t + E_Delta_log_lambda_start),
(* 6. Commercial housing decision *)
ql_t + (1 - (1 - Iq1) theta) lambda_star mu_boe mu_e_t == Iq1 theta lambda_star mu_boe (theta_t + mu_b_t) + (1 - theta lambda_star mu_boe) E mu_e_t +
(1 - beta - beta theta lambda_a) (EY_t - Le_t) + (beta + Iq1 theta lambda_star mu_boe) Eq1_t + Iq1 theta lambda_star mu_boe E_Delta_log_lambda_start,
(* 7. Labor demand *)
w_t == Y_t - n_t,
(* 8. Investment decision *)
qk_t == lambda_k^2 Omega (i_t - i_{t-1} + Delta_log_lambda_k_t) - beta lambda_k^2 Omega (Ei_t - i_t + E_Delta_log_lambda_k_t),
(* 9. Capital decision *)
qk_t + (1 - (1 - Iqk) theta) theta lambda_q^-1 mu_boe mu_e_t ==
(1 - theta theta lambda_q^-1 mu_boe) E mu_e_t + alpha beta (1 - phi) Y_k (EY_t - k_t) + beta (1 - delta) lambda_k^-1 (Eqk_t - E_Delta_log_lambda_k_t) +
Iqk theta theta lambda_q^-1 mu_boe (mu_b_t + Eqk_t + (1 - rho_nu_q) nu_q_t - rho_lambda_q lambda_q_t + theta_t),
(* 10. Demand for loans or borrowing demand *)
R_t == mu_e_t - (lambda_a / (1 + lambda_a)) mu_b_t - (1 / (1 + lambda_a)) (E mu_e_t - E_Delta_log_lambda_start),
(* 11. Production function *)
Y_t + (1 - phi) alpha Delta_log_lambda_k_t == alpha phi Le_{t-1} + alpha (1 - phi) k_{t-1} + (1 - alpha) n_t,
(* 12. Law of motion for capital accumulation *)
k_t == (1 - delta) lambda_k^-1 k_{t-1} + i_k i_t - (1 - delta) lambda_k^-1 Delta_log_lambda_k_t,
(* 13. Budget (flow of funds) constraint *)
alpha Y_t + b_y R^-1 (b_t - R_t) ==
ce_y ce_t + i_y i_t + qlLeOY (Le_t - Le_{t-1}) + b_y lambda_star^-1 (b_{t-1} - Delta_log_lambda_start),
(* 14. Borrowing constraint *)

```

```

b_t == Iq1_ theta + Iq1_ theta lambda_star qlLeOB ( Eq1_t + Le_t + EDelta log lambda_start ) +
      Iqk_ (1 - theta lambda_star qlLeOB) ( Eqk_t + k_t + (1 - rho_vq) vq_t - rho_lambda_q lambda_q_t ),

(* ----- Aggregation equations ----- *)
(* 15. Resource constraint *)
y_t == ch_y ch_t + ce_y ce_t + i_y i_t,
(* 16. Housing resource constraint *)
0 == L_h Lh_t + L_e Le_t,

(* ----- Exogeneous processes ----- *)
(* We use gphi instead of phi,
because phi is an expression and will be automatically substituted in. *)
la_t == rho_lambda_a la_{t-1} + epsilon_lambda_a, (* 17. Household patience shock *)
lz_t == rho_lambda_z lz_{t-1} + epsilon_lambda_z, (* 18. Neutral tech trend growth shock *)
vz_t == rho_lambda_vz vz_{t-1} + epsilon_lambda_vz, (* 19. Neutral tech level shock *)
lq_t == rho_lambda_q lq_{t-1} + epsilon_lambda_q, (* 20. Biased tech trend growth shock *)
vq_t == rho_lambda_vq vq_{t-1} + epsilon_lambda_vq, (* 21. Biased tech level shock *)
gphi_t == rho_phi gphi_{t-1} + epsilon_phi, (* 22. Household housing intratemporal (MRS) shock *)
gpsi_t == rho_psi gpsi_{t-1} + epsilon_psi, (* 23. Household labor intratemporal (MRS) shock *)
gtheta_t == rho_theta gtheta_{t-1} + epsilon_theta, (* 24. (Borrowing) collateral shock *)

(* ----- Expectational errors ----- *)
(* Note that eta has been already used in REequations. *)
ch_t == Ech_{t-1} + eta_ch_t, (* 25. Household consumption expectation. *)
ce_t == Ece_{t-1} + eta_ce_t, (* 26. Entrepreneur consumption expectation. *)
mh_t == Emh_{t-1} + eta_mh_t, (* 27. Household marginal utility expectation. *)
me_t == Eme_{t-1} + eta_me_t, (* 28. Entrepreneur marginal utility expectation. *)
ql_t == Eq1_{t-1} + eta_ql_t, (* 29. Housing price expectation. *)
yt == Eyt_{t-1} + eta_yt, (* 30. Output expectation. *)
it == Eit_{t-1} + eta_it, (* 31. Investment expectation. *)
qk_t == Eqk_{t-1} + eta_qk_t (* 32. Tobin's q expectation. *)
};

(* Variables *)
x0 = {ql_t, ch_t, ce_t, i_t, n_t, R_t, w_t, Y_t, Lh_t, Le_t, k_t, b_t, mh_t, me_t, qk_t, mb_t, la_t,
      lz_t, vz_t, lq_t, vq_t, gphi_t, gpsi_t, gtheta_t, Ech_t, Ece_t, Emh_t, Eme_t, Eq1_t, Eyt, Eit, Eqk_t};
Length[REequations]
Length[x0]
(* Save forever the following forever for getting rid of hats. *)
(*
REequationsWithNoOverHat=REequations/.OverHat[z_]>=>z;
Length[REequationsWithNoOverHat];
x0hat = {c_t, i_t, q_t, l_t, k_t, qk_t, uc_t, z_t, a_t, x_t, Ec_t, Eu_c_t, Ei_t, Eqk_t};
x0 = x0hat/.OverHat[z_]>=>z;
*)

(*
(* Important trick here:*)
epsilon_t = {epsilon_z_t, epsilon_q_t, epsilon_a_t, epsilon_x_t};
epsilon = epsilon_t/.Subscript[z_, Times[q_, t]]>=>Subscript[ToString[z]<>ToString[q], t];

(* The following is yet not working. See

```

```

ComplicatedSubscripts2StringFromMark.nb in /FromMarkFisher. *)
 $\eta_t = \{\eta_{ct}, \eta_{uct}, \eta_{it}, \eta_{yt}, \eta_{qkt}\};$ 
 $\eta = \eta_t /. \text{Subscript}[z_, q\_t] \rightarrow \text{Subscript}[\text{ToString}[z] \langle \rangle \text{ToString}[q], t];$ 
*)

(* Shocks *)
 $\varepsilon_0 = \{\varepsilon \lambda a_t, \varepsilon \lambda z_t, \varepsilon v z_t, \varepsilon \lambda q_t, \varepsilon v q_t, \varepsilon \varphi_t, \varepsilon \psi_t, \varepsilon \theta_t\};$ 
(* Expectational terms *)
 $\eta_0 = \{\eta ch_t, \eta ce_t, \eta \mu h_t, \eta \mu e_t, \eta q l_t, \eta y_t, \eta i_t, \eta q k_t\};$ 
(* because  $\eta$  has been already used in REequations. *)

```