```
var cp rr b q h infl X R Y c hp sdf lam L vp w inflp z1 z2 lR lY lpi lq llam;

varexo eR;

parameters beta gamma nu j m eta Xs theta gammae kappa qhY bY cY cpY hH rY rpi rR sR
iota epsi rrb As Ys Hs hs hps bs cs cps Ls rrb ws lams qs;

load iaco_param_base_nonlin;
set_param_value('beta',beta);
set_param_value('gamma',gamma);
set_param_value('nu',nu);
set_param_value('j',j);
set_param_value('m',m);
set_param_value('eta',eta);
set_param_value('Xs',Xs);
set_param_value('theta',theta);
set_param_value('gammae',gammae);
set_param_value('kappa',kappa);
set_param_value('qhY',qhY);
set_param_value('bY',bY);
set_param_value('cY',cY);
set_param_value('cpY',cpY);
set_param_value('hH',hH);
set_param_value('rY',rY);
set_param_value('rpi',rpi);
set_param_value('rR',rR);
set_param_value('sR',sR);
set_param_value('iota',iota);
set_param_value('As',As);
set_param_value('Ys',Ys);
set_param_value('Hs',Hs);
set_param_value('hs',hs);
set_param_value('qs',qs);
set_param_value('lams',lams);
set_param_value('hps',hps);
set_param_value('bs',bs);
set_param_value('cs',cs);
set_param_value('cps',cps);
set_param_value('Ls',Ls);
set_param_value('epsi',epsi);
set_param_value('rrb',rrb);
set_param_value('ws',ws);
set_param_value('lams',lams);

model;

% (1) Housing Euler equation patient
q/cp = j/hp + beta*q(+1)/cp(+1);

% (2) Labor supply patient
L^(eta - 1) = w/cp;

% (3) Bond Euler equation patient
1/cp = beta*(1/cp(+1))*R/infl(+1);

% (4) Labor demand
(1-nu)*As*h(-1)^(nu)*L^(-nu) = X*w;

% (5) Euler equation housing entrepreneur
q/c = (gamma/c(+1))*(nu*As*h^(nu -1)*L(+1)^(1-nu)/(X(+1)) + q(+1)) +
m*lam*q(+1)*infl(+1);

% (6) Euler equation bonds entrepreneur
1/c = gamma*(1/c(+1))*R/infl(+1) + lam*R;

% (7) Borrowing constraint
b = m*(q(+1)*h*infl(+1)/R);

% (8) z1
```

```
z1 = Y/X + theta*sdf(+1)*infl(+1)^(epsi)*z1(+1);

% (9) z2
z2 = Y + theta*sdf(+1)*infl(+1)^(epsi-1)*z2(+1);

% (10) Reset inflation
inflp = (epsi/(epsi-1))*z1/z2;

% (11) Taylor rule
R = (rrb)^(1-rR)*R(-1)^(rR)*(infl(-1)^(1+rpi)*(Y(-1)/Ys)^rY)^(1-rR)*exp(sR*eR);

% (12) Price evolution
1 = theta*infl^(epsi-1) + (1-theta)*inflp^(1-epsi);

% (13) Production function
Y*vp = As*h(-1)^(nu)*L^(1-nu);

% (14) Price dispersion
vp = (1-theta)*inflp^(-epsi) + theta*infl^(epsi)*vp(-1);

% (15) Resource constraint
c + cp = Y;

% (16) Housin equilibrium
h + hp = Hs;

% (17) Budget constraint for entrepreneur
b = c + q*(h - h(-1)) + R(-1)*b(-1)/infl + w*L - Y*vp/X;

% (18) SDF
sdf = beta*cp(-1)/cp;

% (19) Real rate
rr = R/infl(+1);

% (20) Log nominal rate
lR = log(R);

% (21) Log output
lY = log(Y);

% (22) Net inflation
lpi = log(infl);

% (23) Log house price
lq = log(q);

% (24) log multiplier
llam = log(lam);

end;

initval;
R = rrb;
Y = Ys;
c = cs;
cp = cps;
h = hs;
hp = hps;
b = bs;
infl = 1;
inflp = 1;
vp = 1;
L = Ls;
sdf = beta;
z1 = (Ys/Xs)/(1-beta*theta);
z2 = Ys/(1-beta*theta);
q = qs;
X = Xs;
```

```
lam = lams;
R = rrb;
w = ws;
lY = 0;
lR = 0;
lpi = 0;
lq = log(qs);
llam = log(lams);
end;

steady;

shocks;
var eR = 1;
end;

steady;

stoch_simul(order=1,irf=40,ar=0,nocorr,nograph);
```