Model File

Generated by Python Framework

June 6, 2022

1 Model Information

name: *Eichenbaum, Rebelo and Trabandt model. Application to forecasting Covid-19 economic impact.*
file: */home/alexei/work/Platform/examples/models/COVID19/ert_model.yaml*

1.1 Endogenous Variables Values

F = 78.5, FF = 1.6, Kf = 78.5, KfF = 1.6, Rb = 1.0, RbF = 1.0, c = 697.4, cF = 697.4, ci = 697.4, ciF = 697.4, cr = 697.4, crF = 697.4, cs = 697.4, csF = 697.4, dd = 0.0, ddF = 0.0, i = 0.0, iF = 0.0, k = 178551.2, kF = 178551.2, lambtilde = 0.0, lambtildeF = 0.0, lami = 15215.6, lamiF = 15215.6, lamr = 15292.0, lamrF = 15292.0, lams = 15292.0, lamsF = 15292.0, lamtau = -76.4, lamtauF = -76.4, mc = 0.7, mcF = 0.7, n = 28.0, nF = 28.0, ni = 28.0, niF = 28.0, nr = 28.0, nrF = 28.0, ns = 28.0, nsF = 28.0, pbreve = 1.0, pbreveF = 1.0, pie = 1.0, pieF = 1.0, pop = 1.0, popF = 1.0, r = 0.0, rF = 0.0, rk = 0.0, rkF = 0.0, rr = 1.0, rrF = 1.0, s = 1.0, sF = 1.0, tau = 0.0, tauF = 0.0, v = 0.0, vF = 0.0, w = 19.7, wF = 19.7, x = 206.0, xF = 206.0, y = 1115.4, yF = 1115.4

1.2 Parameters

A = 2.15, Rb_ss = 1.00, alfa = 0.67, betta = 1.00, d_ini = 3.00e-04, delta = 1.15e-03, eta = 0.19, g_ss = 2.12e+02, gam = 1.35, i_ini = 2.00e-03, inc_target = 1.12e+03, lockdown_policy = 0.00, n_target = 28.00, pi1 = 2.00e-07, pi2 = 2.00e-04, pi3 = 0.50, pid = 2.50e-03, pie_ss = 1.00, pir = 0.50, rpi = 1.50, rr_ss = 1.00, rx = 9.62e-03, theta = 1.01e-03, theta_lockdown = 0.00, vaccination_policy = 0.00, vaccination_rate = 0.02, virus_resistant_strain = 0.00, virus_variant_start = 52.00, xi = 0.98, xi_flex = 0.00

1.3 Shocks

ed, ei

1.4 Equations

1
1 : y=pbreve*A*k(-1)^(1-alfa)*n^alfa

2
2 : mc=1/(A*alfa^alfa*(1-alfa)^(1-alfa))*w^alfa*rk^(1-alfa)

3
3 : w=mc*alfa*A*n^(alfa-1)*k(-1)^(1-alfa)

4
4 : k=x+(1-delta)*k(-1)

5
5 : y=c+x+g_ss

6
6 : n = s(-1)*ns+i(-1)*ni+r(-1)*nr

7
7 : c = s(-1)*cs+i(-1)*ci+r(-1)*cr

8
8 : tau = (pi1*s(-1)*cs*i(-1)*ci + pi2*s(-1)*ns*i(-1)*ni + pi3*s(-1)*i(-1))*(1-theta_lockdown*lockdown_policy)^2

9
9 : v = vaccination_rate*vaccination_policy*s(-1)

10
10 : s = s(-1) - tau - v

11
11 : i = i(-1) + tau - (pir+pid)*i(-1) + ei

12
12 : r = r(-1) + pir*i(-1) + v

13
13 : dd = dd(-1) + pid*i(-1) + ed

14
14 : pop = pop(-1) - pid*i(-1)

15

15 : 1/cs=lambtilde-lamtau*pi1*i(-1)*ci

16
16 : 1/ci=lambtilde

17
17 : 1/cr=lambtilde

18
18 : theta*ns=(lambtilde*w+lamtau*pi2*i(-1)*ni) *(1-theta_lockdown*lockdown_policy)

19
19 : theta*ni=lambtilde*w *(1-theta_lockdown*lockdown_policy)

20
20 : theta*nr=lambtilde*w

21
21 : lambtilde=betta*(rk(+1)+(1-delta))*lambtilde(+1)

22
22 : lami=lamtau+lams

23
23 : log(cs(+1)) - theta/2*(ns(+1))^2 + lamtau(+1)*(pi1*cs(+1)*i*ci(+1)+pi2*ns(+1)*i*ni(+1)+pi3*i)
+ lambtilde(+1)*(w(+1)*ns(+1)-cs(+1)) - lams/betta+lams(+1)

24
24 : log(ci(+1)) - theta/2*(ni(+1))^2 + lambtilde(+1)*(w(+1)*ni(+1)-ci(+1)
) - lami/betta+lami(+1)*(1-pir-pid)+lamr(+1)*pir

25
25 : log(cr(+1))-theta/2*(nr(+1))^2 + lambtilde(+1)*(w(+1)*nr(+1)-cr(+1)
) - lamr/betta+lamr(+1)

26
26 : lambtilde=betta*Rb/(pie(+1))*lambtilde(+1)

27
27 : rr=Rb/(pie(+1))

28
28 : Kf=gam*mc*lambtilde*y+betta*xi*(pie(+1))^(gam/(gam-1))*Kf(+1)

29
29 : F=lambtilde*y+betta*xi*(pie(+1))^(1/(gam-1))*F(+1)

30

30 : Kf=F*((1-xi*pie^(1/(gam-1))) / (1-xi))^(-(gam-1))

31

31 : 1/pbreve=(1-xi)*((1-xi*pie^(1/(gam-1)))/(1-xi))^gam + xi*pie^(gam/(gam-1))/pbreve(-1)

32

32 : Rb=rr_ss+rpi*log(pie/pie_ss)+rx*log(y/yF)

33

33 : yF=pbreveF*A*kF(-1)^(1-alfa)*nF^alfa

34

34 : mcF=1/(A*alfa^alfa*(1-alfa)^(1-alfa))*wF^alfa*rkF^(1-alfa)

35

35 : wF=mcF*alfa*A*nF^(alfa-1)*kF(-1)^(1-alfa)

36

36 : kF=xF+(1-delta)*kF(-1)

37

37 : yF=cF+xF+g_ss

38

38 : nF=sF(-1)*nsF+iF(-1)*niF+rF(-1)*nrF

39

39 : cF=sF(-1)*csF+iF(-1)*ciF+rF(-1)*crF

40

40 : tauF = (pi1*sF(-1)*csF*iF(-1)*ciF + pi2*sF(-1)*nsF*iF(-1)*niF + pi3*sF(-1)*iF(-1))*(1-theta_lockdown*lockdown_policy)^2

41

41 : vF = vaccination_rate*vaccination_policy*sF(-1)

42

42 : sF = sF(-1) - tauF - vF

43

43 : iF = iF(-1) + tauF - (pir+pid)*iF(-1) + ei

44

44 : rF = rF(-1) + pir*iF(-1) + vF

45

45 : ddF = ddF(-1) + pid*iF(-1) + ed

46

46 : popF=popF(-1) - pid*iF(-1)

47

47 : 1/csF=lambtildeF-lamtauF*pi1*iF(-1)*ciF

48

48 : 1/ciF=lambtildeF

49

49 : 1/crF=lambtildeF

50

50 : theta*nsF=(lambtildeF*wF+lamtauF*pi2*iF(-1)*niF) *(1-theta_lockdown*lockdown_policy)

51

51 : theta*niF=lambtildeF*wF *(1-theta_lockdown*lockdown_policy)

52

52 : theta*nrF=lambtildeF*wF

53

53 : lambtildeF=betta*(rkF(+1)+(1-delta))*lambtildeF(+1)

54

54 : lamiF=lamtauF+lamsF

55

55 : log(csF(+1)) - theta/2*(nsF(+1))^2 + lamtauF(+1)*(pi1*csF(+1)*iF*ciF(+1)+pi2*nsF(+1)*iF*niF(+1)-
+ lambtildeF(+1)*(wF(+1)*nsF(+1)-csF(+1)) -lamsF/betta+lamsF(+1)

56

56 : log(ciF(+1)) - theta/2*(niF(+1))^2 + lambtildeF(+1)*(wF(+1)*niF(+1)-
ciF(+1)) - lamiF/betta + lamiF*(1-pir-pid) + lamrF(+1)*pir

57

57 : log(crF(+1)) - theta/2*(nrF(+1))^2 + lambtildeF(+1)*(wF(+1)*nrF(+1)-
crF(+1)) - lamrF/betta + lamrF(+1)

58

58 : lambtildeF=betta*RbF/(pieF(+1))*lambtildeF(+1)

59

59 : rrF=RbF/(pieF(+1))

60

60 : KfF=gam*mcF*lambtildeF*yF+betta*xi__flex*(pieF(+1))^(gam/(gam-1))*KfF(+1)

61

61 : FF=lambtildeF*yF+betta*xi__flex*(pieF(+1))^(1/(gam-1))*FF(+1)

62

62 : KfF=FF*((1-xi__flex*pieF^(1/(gam-1))) / (1-xi__flex))^(-(gam-1))

63

63 : 1/pbreveF=(1-xi__flex)*((1-xi__flex*pieF^(1/(gam-1)))/(1-xi__flex))^gam
+ xi__flex*pieF^(gam/(gam-1))/pbreveF(-1)

64

64 : RbF=Rb__ss+rpi*log(pieF/pie__ss)