Model File

Generated by Python Framework

September 21, 2022

1 Model Information

name: *Eichenbaum, Rebelo and Trabandt Model with Resistant Virus Strain.*
file: */home/alexei/work/Platform/examples/models/COVID19/ert1_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, i1 = 0.0, i2 = 0.0, iF = 0.0, iF1 = 0.0, iF2 = 0.0, k = 178551.2, kF = 178551.2, lambtilde = 0.0, lambtildeF = 0.0, lami = 15206.4, lamiF = 15206.4, lamr = 15292.0, lamrF = 15292.0, lams = 15292.0, lamsF = 15292.0, lamtau = -85.6, lamtauF = -85.6, 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, r1 = 0.0, r2 = 0.0, rF = 0.0, rF1 = 0.0, rF2 = 0.0, rk = 0.0, rkF = 0.0, rr = 1.0, rrF = 1.0, s = 1.0, s1 = 1.0, s2 = 1.0, sF = 1.0, sF1 = 1.0, sF2 = 1.0, tau = 0.0, tau1 = 0.0, tau2 = 0.0, tauF = 0.0, tauF1 = 0.0, tauF2 = 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 = 1.00e-04, delta = 1.15e-03, eta = 0.19, g_ss = 2.12e+02, gam = 1.35, i_ini = 1.00e-04, inc_target = 1.12e+03, lockdown_policy = 0.00, mult = 1.25, mult2 = 4.00, n_target = 28.00, pi1 = 2.00e-07, pi2 = 2.00e-04, pi3 = 0.50, pid = 2.80e-03, pie_ss = 1.00, pir = 0.50, rpi = 1.50, rr_ss = 1.00, rx = 9.62e-03, sigma = 0.90, theta = 1.01e-03, theta_lockdown = 0.00, vaccination_policy = 0.00, vaccination_rate = 2.00e-04, virus_resistant_strain = 0.00, virus_variant_start = 62.00, xi = 0.98, xi_flex = 0.00

1.3 Shocks

ed, ei1, ei2

1.4 Equations

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

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

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

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

5 : y=c+x+g_ss

6 : n = s1(-1)*ns+i1(-1)*ni+r1(-1)*nr

7 : c = s1(-1)*cs+i1(-1)*ci+r1(-1)*cr

8 : tau1 = (pi1*s1(-1)*cs*i1(-1)*ci + pi2*s1(-1)*ns*i1(-1)*ni + pi3*s1(-1)*i1(-1))*(1-theta_lockdown*lockdown_policy)^2

9 : tau2 = (pi1*s2(-1)*cs*i2(-1)*ci + pi2*s2(-1)*ns*i2(-1)*ni + mult*pi3*s2(-1)*i2(-1))*virus_resistant_strain*(1-theta_lockdown*lockdown_policy)^2

10 : tau = tau1 + tau2

11 : s1 = s1(-1)-tau1-v

12 : s2 = s2(-1) - tau2

13 : s = IfThenElse(s(-1)-tau-v,s(-1)-tau-v,0)

14 : i1 = i1(-1) + tau1 - (pir+pid)*i1(-1) + ei1

15 : i2 = i2(-1) + (tau2 - (pir+pid/mult2)*i2(-1))*virus_resistant_strain + ei2

16 : i = i1 + i2

17 : r1 = r1(-1) + pir*i1(-1) + v

18 : r2 = r2(-1) + pir*i2(-1)

19 : r = r1 + r2

20 : v = vaccination_rate*s1(-1)

21 : dd = dd(-1) + pid*i1(-1) + pid/mult2*i2(-1) + ed

22 : pop = pop(-1) - pid*i1(-1) - pid/mult2*i2(-1)

23 : 1/cs=lambtilde-lamtau*pi1*i1(-1)*ci

24 : 1/ci=lambtilde

25 : 1/cr=lambtilde

26 : theta*ns=(lambtilde*w+lamtau*pi2*i1(-1)*ni) *(1-theta_lockdown*lockdown_policy)

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

28 : theta*nr=lambtilde*w

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

30 : lami=lamtau+lams

31 : log(cs(+1)) - theta/2*(ns(+1))^2 + lamtau(+1)*(pi1*cs(+1)*i1*ci(+1)+pi2*ns(+1)*i1*ni(+1)+pi3*i1)
+ lambtilde(+1)*(w(+1)*ns(+1)-cs(+1)) - lams/betta+lams(+1)

32 : 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

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

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

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

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

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

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

39 : 1/pbreve=(1-xi)*((1-xi*pie^(1/(gam-1)))/(1-xi))^gam + xi*pie^(gam/(gam-
1))/pbreve(-1)

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

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

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

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

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

45 : yF=cF+xF+g_ss

46 : nF=sF1(-1)*nsF+iF1(-1)*niF+rF1(-1)*nrF

47 : cF=sF1(-1)*csF+iF1(-1)*ciF+rF1(-1)*crF

48 : tauF1 = (pi1*sF1(-1)*csF*iF1(-1)*ciF + pi2*sF1(-1)*nsF*iF1(-1)*niF +
pi3*sF1(-1)*iF1(-1))*(1-theta_lockdown*lockdown_policy)^2

49 : tauF2 = (pi1*sF2(-1)*csF*iF2(-1)*ciF + pi2*sF2(-1)*nsF*iF2(-1)*niF +
mult*pi3*sF2(-1)*iF2(-1))*virus_resistant_strain*(1-theta_lockdown*lockdown_policy)^2

50 : tauF = tauF1 + tauF2

51 : sF1 = sF1(-1) - tauF1 - vF

52 : sF2 = sF2(-1) - tauF2

53 : sF = IfThenElse(sF(-1)-tauF-vF,sF(-1)-tauF-vF,0)

54 : iF1 = iF1(-1) + tauF1 - (pir+pid)*iF1(-1) + ei1

55 : iF2 = iF2(-1) + (tauF2 - (pir+pid/mult2)*iF2(-1))*virus_resistant_strain
+ ei2

56 : iF = iF1 + iF2

57 : rF1 = rF1(-1) + pir*iF1(-1) + vF

58 : rF2 = rF2(-1) + pir*iF2(-1)

59 : rF = rF1 + rF2

60 : vF = vaccination_rate*sF1(-1)

61 : ddF = ddF(-1) + pid*iF1(-1) + pid/mult2*iF2(-1) + ed

62 : popF = popF(-1) - pid*iF1(-1) - pid/mult2*iF2(-1)

63 : 1/csF=lambtildeF-lamtauF*pi1*iF1(-1)*ciF

64 : 1/ciF=lambtildeF

65 : 1/crF=lambtildeF

66 : theta*nsF=(lambtildeF*wF+lamtauF*pi2*iF1(-1)*niF) *(1-theta_lockdown*lockdown_policy)

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

68 : theta*nrF=lambtildeF*wF

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

70 : lamiF=lamtauF+lamsF

71 : log(csF(+1)) - theta/2*(nsF(+1))^2 + lamtauF(+1)*(pi1*csF(+1)*iF1*ciF(+1)+pi2*nsF(+1)*iF1*niF(+1
+ lambtildeF(+1)*(wF(+1)*nsF(+1)-csF(+1)) -lamsF/betta+lamsF(+1)

72 : 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

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

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

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

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

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

78 : KfF=FF*((1-xi_flex*pieF^(1/(gam-1))) / (1-xi_flex))^(-(gam-1))

79 : 1/pbreveF=(1-xi_flex)*((1-xi_flex*pieF^(1/(gam-1)))/(1-xi_flex))^gam
+ xi_flex*pieF^(gam/(gam-1))/pbreveF(-1)

80 : RbF=Rb_ss+rpi*log(pieF/pie_ss)