A how-to guide to risk-adjusted linearizations

Based on P. Lopez, D. Lopez-Salido, and F. Vazquez-Grande "Entropy-based approximations of DSGE models: A unified theory of risk-adjusted linearizations"

Pierlauro Lopez   Francisco Vazquez-Grande

Support slides to the code genaffine v1.0

August 17, 2018

Risk-adjusted linearizations

I will show you an ad-hoc approximation algorithm from the finance literature that produces linearizations that include risk adjustments and that can handle non-Gaussian shocks $\Rightarrow$ Affine approximation.

Not in this presentation (but in the paper!): A formal justification.
$\Rightarrow$ equivalence with a linear perturbation around the risky steady state
- $\Rightarrow$ closed-form formulas for equilibrium coefficients
- $\Rightarrow$ counterpart to Blanchard-Kahn conditions

Deterministic SS: ex-post shocks are 0 and agents know that ex ante.
Risky SS (Juillard, 2011; Coeurdacier et al., 2011): ex-post shocks are 0 but agents don't know that ex ante.

Campbell-Cochrane (1999): Pricing a one-period bond

Suppose the economy is described by:

$$
\begin{aligned}
0 &= \ln E_t \exp(m_{t+1} + r_t) \\
m_{t+1} &= \ln(\beta) - \gamma \Delta c_{t+1} - \gamma \Delta s_{t+1} \\
s_{t+1} &= \phi s_t + \lambda(s_t)\sigma\varepsilon_{t+1} \\
c_{t+1} &= c_t + \sigma\varepsilon_{t+1}
\end{aligned}
$$

with $\varepsilon_t \sim Niid(0, 1)$.

Campbell-Cochrane (1999): Pricing a one-period bond

Suppose the economy is described by:

$$
\begin{aligned}
0 &= E_t m_{t+1} + r_t + \frac{1}{2} var_t(m_{t+1}) \\
m_{t+1} &= \ln(\beta) - \gamma \Delta c_{t+1} - \gamma \Delta s_{t+1} \\
s_{t+1} &= \phi s_t + \lambda(s_t)\sigma\varepsilon_{t+1} \\
c_{t+1} &= c_t + \sigma\varepsilon_{t+1}
\end{aligned}
$$

with $\varepsilon_t \sim Niid(0,1)$.

Campbell-Cochrane (1999): Pricing a one-period bond

Suppose the economy is described by:

$$
\begin{aligned}
0 &= \ln\beta - \gamma E_t \Delta c_{t+1} - \gamma E_t \Delta s_{t+1} + r_t + \frac{\gamma^2}{2} var_t(\Delta c_{t+1} + \Delta s_{t+1}) \\
m_{t+1} &= \ln(\beta) - \gamma \Delta c_{t+1} - \gamma \Delta s_{t+1} \\
s_{t+1} &= \phi s_t + \lambda(s_t)\sigma\varepsilon_{t+1} \\
c_{t+1} &= c_t + \sigma\varepsilon_{t+1}
\end{aligned}
$$

with $\varepsilon_t \sim Niid(0,1)$.

Campbell-Cochrane (1999): Pricing a one-period bond

Suppose the economy is described by:

$$\begin{aligned}
0 &= \ln\beta - \gamma E_t \Delta c_{t+1} - \gamma E_t \Delta s_{t+1} + r_t + \frac{\gamma^2 [1+\lambda(s_t)]^2 \sigma^2}{2} \\
m_{t+1} &= \ln(\beta) - \gamma \Delta c_{t+1} - \gamma \Delta s_{t+1} \\
s_{t+1} &= \phi s_t + \lambda(s_t) \sigma \varepsilon_{t+1} \\
c_{t+1} &= c_t + \sigma \varepsilon_{t+1}
\end{aligned}$$

with $\varepsilon_t \sim Niid(0,1)$.

Campbell-Cochrane (1999): Pricing a one-period bond

$$0 = \ln\beta - \gamma E_t \Delta c_{t+1} - \gamma E_t \Delta s_{t+1} + r_t + \frac{\gamma^2[1+\lambda(s_t)]^2\sigma^2}{2}$$

Campbell-Cochrane (1999): Pricing a one-period bond

We conjecture $r_t = r + \psi_{rs} s_t$:

$$\begin{aligned} 0 &= \ln \beta - \gamma E_t \Delta c_{t+1} - \gamma E_t \Delta s_{t+1} + r_t + \frac{\gamma^2 [1 + \lambda(s_t)]^2 \sigma^2}{2} \\ &= \ln \beta + \gamma(1 - \phi) s_t + r + \psi_{rs} s_t + \frac{\gamma^2 [1 + \lambda(s_t)]^2 \sigma^2}{2} \end{aligned}$$

Campbell-Cochrane (1999): Pricing a one-period bond

We conjecture $r_t = r + \psi_{rs} s_t$:

$$
\begin{aligned}
0 = & \ln \beta - \gamma E_t \Delta c_{t+1} - \gamma E_t \Delta s_{t+1} + r_t + \frac{\gamma^2 [1 + \lambda(s_t)]^2 \sigma^2}{2} \\
= & \ln \beta + \gamma(1 - \phi) s_t + r + \psi_{rs} s_t + \frac{\gamma^2 [1 + \lambda(s_t)]^2 \sigma^2}{2} \\
\approx & \ln \beta + \frac{\gamma^2 [1 + \lambda(0)]^2 \sigma^2}{2} + r + \gamma(1 - \phi) s_t + \psi_{rs} s_t + \frac{\gamma^2 [1 + \lambda(0)] \lambda'(0) \sigma^2}{2} s_t
\end{aligned}
$$

Campbell-Cochrane (1999): Pricing a one-period bond

We conjecture $r_t = r + \psi_{rs} s_t$:

$$\begin{aligned}
0 &= \ln\beta - \gamma E_t \Delta c_{t+1} - \gamma E_t \Delta s_{t+1} + r_t + \frac{\gamma^2[1+\lambda(s_t)]^2\sigma^2}{2} \\
&= \ln\beta + \gamma(1-\phi)s_t + r + \psi_{rs} s_t + \frac{\gamma^2[1+\lambda(s_t)]^2\sigma^2}{2} \\
&\approx \ln\beta + \frac{\gamma^2[1+\lambda(0)]^2\sigma^2}{2} + r + \gamma(1-\phi)s_t + \psi_{rs} s_t + \frac{\gamma^2[1+\lambda(0)]\lambda'(0)\sigma^2}{2} s_t
\end{aligned}$$

and solve for unknown coefficients r and ψ_{rs}:

$$r = -\ln\beta - \underbrace{\frac{\gamma^2[1+\lambda(0)]^2\sigma^2}{2}}_{\text{precautionary saving}}, \quad \psi_{rs} = -\underbrace{\gamma(1-\phi)}_{\substack{\text{intertemporal}\\ \text{substitution}}} - \underbrace{\gamma[1+\lambda(0)]\lambda'(0)\sigma^2}_{\text{precautionary saving}}$$

Campbell-Cochrane (1999): Pricing a one-period bond

We conjecture $r_t = r + \psi_{rs} s_t$:

$$\begin{aligned} 0 &= \ln\beta - \gamma E_t \Delta c_{t+1} - \gamma E_t \Delta s_{t+1} + r_t + \frac{\gamma^2[1+\lambda(s_t)]^2\sigma^2}{2} \\ &= \ln\beta + \gamma(1-\phi)s_t + r + \psi_{rs}s_t + \frac{\gamma^2[1+\lambda(s_t)]^2\sigma^2}{2} \\ &\approx \ln\beta + \frac{\gamma^2[1+\lambda(0)]^2\sigma^2}{2} + r + \gamma(1-\phi)s_t + \psi_{rs}s_t + \frac{\gamma^2[1+\lambda(0)]\lambda'(0)\sigma^2}{2}s_t \end{aligned}$$

and solve for unknown coefficients r and ψ_{rs}:

$$r = -\ln\beta - \underbrace{\frac{\gamma^2[1+\lambda(0)]^2\sigma^2}{2}}_{\text{precautionary saving}}, \quad \psi_{rs} = -\underbrace{\gamma(1-\phi)}_{\substack{\text{intertemporal}\\ \text{substitution}}} - \underbrace{\gamma[1+\lambda(0)]\lambda'(0)\sigma^2}_{\text{precautionary saving}}$$

Let's generalize this line of reasoning...

Let the dynamic system:

$$
\begin{aligned}
0 &= \ln E_t \exp[h(y_t, z_t) + f_3 y_{t+1} + f_4 z_{t+1}] \\
z_{t+1} &= g(y_t, z_t) + \lambda(z_t)(E_{t+1} - E_t) y_{t+1} + \sigma(z_t)\varepsilon_{t+1}
\end{aligned}
$$

where $\varepsilon_t \sim MDS$, with conditional cumulant generating function

$$
\kappa[\alpha(z_t); z_t] \doteq \ln E_t \exp[\alpha(z_t)' \varepsilon_{t+1}]
$$

Let's generalize this line of reasoning...

Let the dynamic system:

$$
\begin{aligned}
0 &= h(y_t, z_t) + f_3 E_t y_{t+1} + f_4 E_t z_{t+1} + \mathcal{V}_t[\exp(f_3 y_{t+1} + f_4 z_{t+1})] \\
z_{t+1} &= g(y_t, z_t) + \lambda(z_t)(E_{t+1} - E_t) y_{t+1} + \sigma(z_t)\varepsilon_{t+1}
\end{aligned}
$$

where $\varepsilon_t \sim MDS$, with conditional cumulant generating function

$$
\kappa[\alpha(z_t); z_t] \doteq \ln E_t \exp[\alpha(z_t)' \varepsilon_{t+1}]
$$

Conditional entropy of a r.v. $\mathcal{V}_t[\exp(x_{t+1})] \doteq \ln E_t \exp(x_{t+1}) - E_t x_{t+1}$.

Let's generalize this line of reasoning...

Let the dynamic system:

$$\begin{aligned} 0 &= h(y_t, z_t) + f_3 E_t y_{t+1} + f_4 E_t z_{t+1} + \mathcal{V}_t[\exp(f_3 y_{t+1} + f_4 z_{t+1})] \\ z_{t+1} &= g(y_t, z_t) + \lambda(z_t)(E_{t+1} - E_t) y_{t+1} + \sigma(z_t)\varepsilon_{t+1} \end{aligned}$$

where $\varepsilon_t \sim MDS$, with conditional cumulant generating function

$$\kappa[\alpha(z_t); z_t] \doteq \ln E_t \exp[\alpha(z_t)' \varepsilon_{t+1}]$$

Conditional entropy of a r.v. $\mathcal{V}_t[\exp(x_{t+1})] \doteq \ln E_t \exp(x_{t+1}) - E_t x_{t+1}$.
We are looking for an affine solution with unknowns $[\widetilde{y}; \widetilde{z}; \widetilde{\Psi}]$:

$$y_t = \widetilde{y} + \widetilde{\Psi}(z_t - \widetilde{z})$$

which implies:

$$\begin{aligned} z_{t+1} &= g(y_t, z_t) + [I_{n_z} - \lambda(z_t)\widetilde{\Psi}]^{-1} \sigma(z_t)\varepsilon_{t+1} \\ \widetilde{\mathcal{V}}(z_t) &\doteq \mathcal{V}_t[\exp((f_3 \widetilde{\Psi} + f_4) z_{t+1})] \end{aligned}$$

Let's generalize this line of reasoning...

Let the dynamic system:

$$0 = h(y_t, z_t) + f_3 E_t y_{t+1} + f_4 E_t z_{t+1} + \mathcal{V}_t[\exp(f_3 y_{t+1} + f_4 z_{t+1})]$$
$$z_{t+1} = g(y_t, z_t) + \lambda(z_t)(E_{t+1} - E_t) y_{t+1} + \sigma(z_t)\varepsilon_{t+1}$$

where $\varepsilon_t \sim MDS$, with conditional cumulant generating function

$$\kappa[\alpha(z_t); z_t] \doteq \ln E_t \exp[\alpha(z_t)'\varepsilon_{t+1}]$$

Conditional entropy of a r.v. $\mathcal{V}_t[\exp(x_{t+1})] \doteq \ln E_t \exp(x_{t+1}) - E_t x_{t+1}$.
We are looking for an affine solution with unknowns $[\widetilde{y}; \widetilde{z}; \widetilde{\Psi}]$:

$$y_t = \widetilde{y} + \widetilde{\Psi}(z_t - \widetilde{z})$$

which implies:

$$z_{t+1} = g(y_t, z_t) + [I_{n_z} - \lambda(z_t)\widetilde{\Psi}]^{-1}\sigma(z_t)\varepsilon_{t+1}$$
$$\widetilde{\mathcal{V}}(z_t) \doteq \mathcal{V}_t[\exp((f_3\widetilde{\Psi} + f_4) z_{t+1})] = \ln E_t e^{(f_3\widetilde{\Psi}+f_4)[I_{n_z}-\lambda(z_t)\widetilde{\Psi}]^{-1}\sigma(z_t)\varepsilon_{t+1}}$$

Let's generalize this line of reasoning...

Let the dynamic system:

$$0 = h(y_t, z_t) + f_3 E_t y_{t+1} + f_4 E_t z_{t+1} + \widetilde{\mathcal{V}}(z_t)$$
$$z_{t+1} = g(y_t, z_t) + \lambda(z_t)(E_{t+1} - E_t) y_{t+1} + \sigma(z_t)\varepsilon_{t+1}$$

where $\varepsilon_t \sim MDS$, with conditional cumulant generating function

$$\kappa[\alpha(z_t); z_t] \doteq \ln E_t \exp[\alpha(z_t)' \varepsilon_{t+1}]$$

Conditional entropy of a r.v. $\mathcal{V}_t[\exp(x_{t+1})] \doteq \ln E_t \exp(x_{t+1}) - E_t x_{t+1}$.
We are looking for an affine solution with unknowns $[\widetilde{y}; \widetilde{z}; \widetilde{\Psi}]$:

$$y_t = \widetilde{y} + \widetilde{\Psi}(z_t - \widetilde{z})$$

which implies:

$$z_{t+1} = g(y_t, z_t) + [I_{n_z} - \lambda(z_t)\widetilde{\Psi}]^{-1} \sigma(z_t)\varepsilon_{t+1}$$
$$\widetilde{\mathcal{V}}(z_t) \doteq \kappa[(f_3 \widetilde{\Psi} + f_4)[I_{n_z} - \lambda(z_t)\widetilde{\Psi}]^{-1} \sigma(z_t); z_t]$$

Let's generalize this line of reasoning...

The unknowns $[\widetilde{y}, \widetilde{z}, \widetilde{\Psi}]$ solve the system of equations:

$$\begin{aligned}
\widetilde{z} &= g(\widetilde{y}, \widetilde{z}) \\
0 &= h(\widetilde{y}, \widetilde{z}) + f_3 \widetilde{y} + f_4 \widetilde{z} + \widetilde{\mathcal{V}}(\widetilde{z}) \\
0 &= \widetilde{h}_1 \widetilde{\Psi} + \widetilde{h}_2 + (f_3 \widetilde{\Psi} + f_4)(\widetilde{g}_1 \widetilde{\Psi} + \widetilde{g}_2) + \widetilde{\mathcal{V}}_1(\widetilde{z})
\end{aligned}$$

with $\widetilde{\mathcal{V}}(z_t) \doteq \kappa \left[(f_3 \widetilde{\Psi} + f_4)[I_{n_z} - \lambda(z_t)\widetilde{\Psi}]^{-1} \sigma(z_t); z_t \right]$.

A companion code: `genaffine.m`

Matlab code based on Symbolic Math Toolbox.

User friendly? Yes. Non-Gaussianities (e.g., disaster risk) are easy to include via MGF.

Dynare-integrable? Probably.

Optimizable? Definitely, and would bypass Symbolic Math Toolbox.

Before going to the code, one caveat and one more application to see at least one big example in which adjusting for risk matters quantitatively...

Loose ends

How restrictive is the form $0 = \ln E_t \exp[h(y_t, z_t) + f_3 y_{t+1} + f_4 z_{t+1}]$?

- Argument of expectation a.s.-strictly positive.
  $\Rightarrow$ Can be restrictive, although it can often be satisfied by splitting the argument into strictly positive components.
- Linearity in y_{t+1} and z_{t+1}.
  $\Rightarrow$ Not restrictive (can always define variables appropriately).
  $\Rightarrow$ But there are infinite ways to write forward-looking difference equations that way!
  (Which one should we pick? One whose associated affine approximation minimizes FLDE's Euler equation error.)

Campbell-Cochrane (1999): Pricing wealth

From $1 = E_t M_{t+1} \frac{P_{t+1}+C_{t+1}}{P_t}$, log price-consumption ratio of wealth solves:

$$
\begin{aligned}
e^{pc_t} &= E_t e^{m_{t+1}+\Delta c_{t+1}} + E_t e^{m_{t+1}+\Delta c_{t+1}+pc_{t+1}} \\
&= \sum_{n=1}^{N} E_t e^{m_{t,t+n}+c_{t+n}-c_t} + E_t e^{m_{t,t+N}+c_{t+N}-c_t+pc_{t+N}}
\end{aligned}
$$

Campbell-Cochrane (1999): Pricing wealth

From $1 = E_t M_{t+1} \frac{P_{t+1}+C_{t+1}}{P_t}$, log price-consumption ratio of wealth solves:

$$\begin{aligned} e^{pc_t} &= E_t e^{m_{t+1}+\Delta c_{t+1}} + E_t e^{m_{t+1}+\Delta c_{t+1}+pc_{t+1}} \\ &= \sum_{n=1}^{N} E_t e^{m_{t,t+n}+c_{t+n}-c_t} + E_t e^{m_{t,t+N}+c_{t+N}-c_t+pc_{t+N}} \end{aligned}$$

$\Rightarrow$ For a given $N > 0$, we must solve the system of $2N+1$ equations:

$$pc_t = \ln\left(\sum_{n=1}^{N} e^{pc_t^{(n)}} + e^{rc_t^{(N)}}\right)$$

$$pc_t^{(n)} = \ln E_t e^{m_{t+1}+\Delta c_{t+1}+pc_{t+1}^{(n-1)}}, \qquad pc_t^{(0)} = 0$$

$$rc_t^{(n)} = \ln E_t e^{m_{t+1}+\Delta c_{t+1}+rc_{t+1}^{(n-1)}} \qquad rc_t^{(0)} = pc_t$$

Note: transversality condition implies $rc_t^{(N)}, pc_t^{(N)} \to 0$ as $N \to \infty$.

Campbell-Cochrane (1999): Pricing wealth

The approximate solution is:

$$pc_t^{(n)} = \widetilde{pc}^{(n)} + \widetilde{\psi}^{(n)}\widehat{s}_t$$

$$rc_t^{(n)} = \widetilde{rc}^{(n)} + \widetilde{\varphi}^{(n)}\widehat{s}_t$$

$$pc_t = \ln\left(\sum_{n=1}^{N} e^{\widetilde{pc}^{(n)}+\widetilde{\psi}^{(n)}\widehat{s}_t} + e^{\widetilde{rc}^{(N)}+\widetilde{\varphi}^{(N)}\widehat{s}_t}\right) \qquad \text{(quasi-affine)}$$

$$\approx \ln\left(\sum_{n=1}^{N} e^{\widetilde{pc}^{(n)}} + e^{\widetilde{rc}^{(N)}}\right) + \frac{\sum_{n=1}^{\infty} e^{\widetilde{pc}^{(n)}}\widetilde{\psi}^{(n)} + e^{\widetilde{rc}^{(N)}}\widetilde{\varphi}^{(N)}}{\sum_{n=1}^{N} e^{\widetilde{pc}^{(n)}} + e^{\widetilde{rc}^{(N)}}}\widehat{s}_t \qquad \text{(affine)}$$

Campbell-Cochrane (1999): Pricing wealth

The approximate solution is:

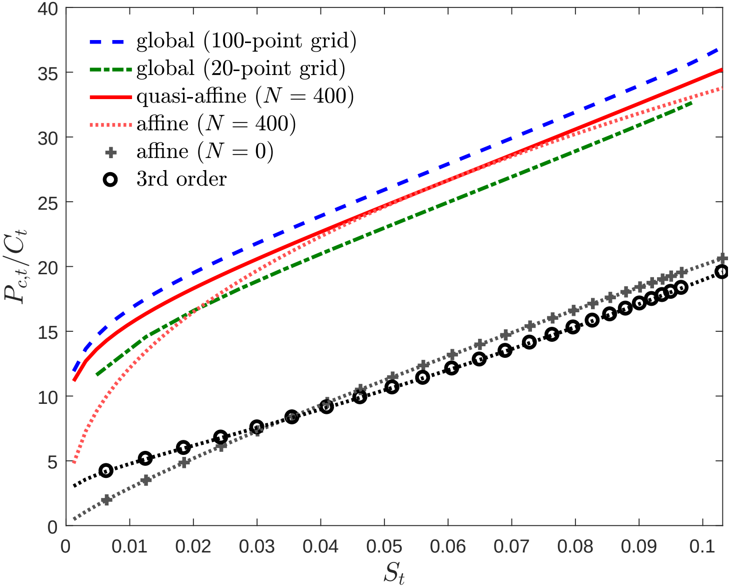
$$pc_t^{(n)} = \widetilde{pc}^{(n)} + \widetilde{\psi}^{(n)}\widehat{s}_t$$

$$rc_t^{(n)} = \widetilde{rc}^{(n)} + \widetilde{\varphi}^{(n)}\widehat{s}_t$$

$$pc_t = \ln\left(\sum_{n=1}^{N} e^{\widetilde{pc}^{(n)}+\widetilde{\psi}^{(n)}\widehat{s}_t} + e^{\widetilde{rc}^{(N)}+\widetilde{\varphi}^{(N)}\widehat{s}_t}\right) \qquad \text{(quasi-affine)}$$

$$\approx \ln\left(\sum_{n=1}^{N} e^{\widetilde{pc}^{(n)}} + e^{\widetilde{rc}^{(N)}}\right) + \frac{\sum_{n=1}^{\infty} e^{\widetilde{pc}^{(n)}}\widetilde{\psi}^{(n)} + e^{\widetilde{rc}^{(N)}}\widetilde{\varphi}^{(N)}}{\sum_{n=1}^{N} e^{\widetilde{pc}^{(n)}} + e^{\widetilde{rc}^{(N)}}}\widehat{s}_t \qquad \text{(affine)}$$

$\Rightarrow$ If strips are nearly linear, then quasi-affine solution is nearly global!


global (100-point grid)
global (20-point grid)
quasi-affine ($N = 400$)
affine ($N = 400$)
affine ($N = 0$)
3rd order
$P_{c,t}/C_t$
S_t
0
5
10
15
20
25
30
35
40
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1

Companion code: genaffine.m (cont'd)

```
%% variables

% Enter name of state variables
syms z_name1 z_name2 ;
zt = [z_name1; z_name2] ;
% Enter name of jump variables
syms y_name1 y_name2 ;
% Optional:  Enter numbers of strips (if used)
N = ;
pd_d = sym('pd_d_%d', [1 N]).';
rd_d = sym('rd_d_%d', [1 N]).';
yt = [y_name1; y_name2 ; pd_d; rd_d];
% Enter name of exogenous shocks
syms eps_1
epst = [eps_1] ;
```

```
%% variables

% Enter name of state variables
syms hats epsa ;
zt = [hats; epsa] ;
% Enter name of jump variables
syms rf pc
disp('Select number of strips')
N = input(''); % number of strips (if used)
if N > 0
    pd_c = sym('pd_c_%d', [1 N]).';
    rd_c = sym('rd_c_%d', [1 N]).';
    yt = [rf; pc; pd_c; rd_c];
else
    syms wc
    yt = [rf; pc; wc];
end
% Enter name of exogenous shocks
syms eps_a
epst = [eps_a] ;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    run fct_setup_step1 ;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Companion code: genaffine.m (cont'd)

```
%% parameters

% Enter name of parameters
syms par1 par2
MODEL.parameters.params = [ par1; par2 ] ;

%% calibration
% Enter values of parameters
par1 = ;
par2 = ;
```

```
%% parameters

% Enter name of parameters
syms beta ga sigmaa gamma rhos S
MODEL.parameters.params = [ beta; ga; sigmaa; gamma; rhos; S ] ;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    run fct_setup_step2 ;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Enter values of parameters
freq = 4;
ga = .0189/freq;
sigmaa = .015/sqrt(freq);
gamma = 2;
rhos = .875^(1/freq);
beta = exp(-.0094/freq+gamma*ga-0.5*gamma*(1-rhos));
xi = 0;
S = sqrt(gamma*sigmaa^2/(1-rhos-xi/gamma));
```

Companion code: genaffine.m (cont'd)

```
%% System

% EXPECTATIONAL EQUATIONS
% Enter row by row function f(y_t, z_t, y_{t+1}, z_{t+1}) = hh(y_t, z_t) + f_3 y_{t+1} + f_4 z_{t+1} in
FOCs:  ln(E_t exp[f(y_t, z_t, y_{t+1}, z_{t+1})]) = 0

% Enter h(y_t, z_t)
hh1 = ;
hh2 = ;
for i=1:N
   stripstart = ; stripclasses = ; % if strips are included,
indicate number of row of h at which Euler equations for strips start
and how many classes of strips there are
   eval(['hh' num2str(4+i) ' = ...  - pd_a(' num2str(i) ');']);
   eval(['hh' num2str(4+N+i) ' = ...  - rd_a(' num2str(i) ');']);
end
```

Companion code: genaffine.m (cont'd)

```
%% System

% EXPECTATIONAL EQUATIONS
% Enter row by row function f(y_t, z_t, y_{t+1}, z_{t+1}) = hh(y_t, z_t) + f_3 y_{t+1} + f_4 z_{t+1} in
FOCs:   ln(E_t exp[f(y_t, z_t, y_{t+1}, z_{t+1})]) = 0

% Enter f_3 y_t + f_4 z_t (note timing:  t not t+1)
ff1 = ;
ff2 = ;
eval(['ff' num2str(4+1) ' = ...;']);
eval(['ff' num2str(4+N+1) ' = ...  + vau;']);
for i=2:N
   eval(['ff' num2str(4+i) ' =...  + pd_a(' num2str(i-1) ');']);
   eval(['ff' num2str(4+N+i) ' = ...  + rd_a(' num2str(i-1)
');']);
end
```

Companion code: genaffine.m (cont'd)

```
%% System

% LAW OF MOTION OF STATE VARIABLES
% Enter row by row law of motion of state variables
```

$z_{t+1} = gg(y_t, z_t) + \lambda(z_t)(E_{t+1} - E_t)y_{t+1} + \sigma(z_t)\varepsilon_{t+1}$

```
gg1 = ;
gg2 = ;
```

% Fill in nonzero elements of $\lambda(z_t)$

```
lambdaz(1,1) = ;
```

% Fill in nonzero elements of $\sigma(z_t)$

```
sigmaz(1,1) = ;

% Enter cumulant generating function of shocks
CCDF = @(uu) .5*diag(uu*uu.')  ;
```

```
%% System

% EXPECTATIONAL EQUATIONS
% Enter row by row function f(y_t,z_t,y_{t+1},z_{t+1})=hh(y_t,z_t)+f_3y_{t+1}+f_4z_{t+1} in FOCs:
% Enter h(y_t,z_t)
hh1 = log(beta*exp(-gamma*ga)) + gamma*hats + rf ;
hh2 = sum(exp(pd_c(1:N))) + exp(rd_c(N)) - exp(pc) ;
stripstart = 3; stripclasses = 2; % indicate number of row at which strips start and how many cla
for i=1:N
    eval(['hh' num2str(2+i)   ' = log(beta*exp((1-gamma)*ga)) + gamma*hats - pd_c(' num2str(i) ')
    eval(['hh' num2str(2+N+i) ' = log(beta*exp((1-gamma)*ga)) + gamma*hats - rd_c(' num2str(i) ')
end
% Enter f_3y_t+f_4z_t (note timing: t not t+1)
ff1 = -gamma*hats -gamma*sigmaa*epsa;
ff2 = 0;
eval(['ff' num2str(2+1)        ' = -gamma*hats + (1-gamma)*sigmaa*epsa;']) ;
eval(['ff' num2str(2+N+1)      ' = -gamma*hats + (1-gamma)*sigmaa*epsa + pc;']) ;
for i=2:N
    eval(['ff' num2str(2+i)   ' = -gamma*hats + (1-gamma)*sigmaa*epsa + pd_c(' num2str(i-1) ');']
    eval(['ff' num2str(2+N+i) ' = -gamma*hats + (1-gamma)*sigmaa*epsa + rd_c(' num2str(i-1) ');']
end
```

```
% LAW OF MOTION OF STATE VARIABLES
% Enter row by row law of motion of state variables z_{t+1}=gg(y_t,z_t)+lambda(z_t)(E_{t+
gg1 = rhos*hats ;
gg2 = 0 ;
% Fill in nonzero elements of sigma(z_t)
sigmaz(1,1) = (1/S*sqrt(1-2*hats)-1)*sigmaa ;
sigmaz(2,1) = 1 ;

% Enter cumulant generating function of shocks
CCDF = @(uu) .5*diag(uu*uu.') ;
```

How to solve the nonlinear matrix equation: Method 1

Solution $[\widetilde{y}, \widetilde{z}, \widetilde{\Psi}]$ to system:

$$0 = f(\widetilde{y}, \widetilde{z}) + f_3 \widetilde{y} + f_4 \widetilde{z} + \widetilde{\mathcal{V}}(\widetilde{z}), \quad \widetilde{z} = g(\widetilde{y}, \widetilde{z})$$
$$0 = \widetilde{f_1} \widetilde{\Psi} + \widetilde{f_2} + (f_3 \widetilde{\Psi} + f_4)(\widetilde{g}_1 \widetilde{\Psi} + \widetilde{g}_2) + \widetilde{\mathcal{V}}_1(\widetilde{z})$$

$\widetilde{\mathcal{V}}(z_t) = \kappa[(f_3 \widetilde{\Psi} + f_4)\widetilde{\sigma}_z(z_t); z_t]$, with $\widetilde{\sigma}_z(z_t) \doteq [I_{n_z} - \lambda(z_t)\widetilde{\Psi}]^{-1} \sigma(z_t)$.

How to solve the nonlinear matrix equation: Method 1

Solution $[\widetilde{y}, \widetilde{z}, \widetilde{\Psi}]$ to system:

$$0 = f(\widetilde{y}, \widetilde{z}) + f_3 \widetilde{y} + f_4 \widetilde{z} + q \widetilde{\mathcal{V}}(\widetilde{z}), \quad \widetilde{z} = g(\widetilde{y}, \widetilde{z})$$
$$0 = \widetilde{f}_1 \widetilde{\Psi} + \widetilde{f}_2 + (f_3 \widetilde{\Psi} + f_4)(\widetilde{g}_1 \widetilde{\Psi} + \widetilde{g}_2) + q \widetilde{\mathcal{V}}_1(\widetilde{z})$$

$\widetilde{\mathcal{V}}(z_t) = \kappa[(f_3 \widetilde{\Psi} + f_4)\widetilde{\sigma}_z(z_t); z_t]$, with $\widetilde{\sigma}_z(z_t) \doteq [I_{n_z} - \lambda(z_t)\widetilde{\Psi}]^{-1}\sigma(z_t)$.

$q = 0$: linear approximation around the deterministic steady state
$q = 1$: linear approximation around the risky steady state

A simple continuation algorithm Start from the standard (i.e., saddle-path stable) linear solution at $q = 0$ and proceed sequentially until $q = 1$ with the outcome of the previous step as the initial guess for the next.

How to solve the nonlinear matrix equation: Method 1bis

What if the deterministic steady state is not well-defined?
(E.g., non-closed small open economies.)

$\Rightarrow$ Solve the whole system simultaneously at some $q > 0$. If you start at $q < 1$, then run either the continuation method 1 or the iterative method 2 until $q = 1$.

How to solve the nonlinear matrix equation: Method 2

A simple iterative algorithm Find the deterministic steady state $[\bar{y}, \bar{z}]$:

$$0 = f(\bar{y}, \bar{z}) + f_3 \bar{y} + f_4 \bar{z}, \quad \bar{z} = g(\bar{y}, \bar{z})$$

By a QZ decomposition find the deterministic steady state slope $\bar{\Psi}$:

$$0 = f_1(\bar{y}, \bar{z})\bar{\Psi} + f_2(\bar{y}, \bar{z}) + (f_3\bar{\Psi} + f_4)[g_1(\bar{y}, \bar{z})\bar{\Psi} + g_2(\bar{y}, \bar{z})]$$

Set $y_0 = \bar{y}$, $z_0 = \bar{z}$, $\Psi_0 = \bar{\Psi}$, and run to convergence the algorithm:
(1) Find $[\widetilde{y}_n, \widetilde{z}_n]$ in:

$$\begin{aligned} 0 &= f(\widetilde{y}_n, \widetilde{z}_n) + f_3 \widetilde{y}_n + f_4 \widetilde{z}_n + \widetilde{\mathcal{V}}(\widetilde{z}_{n-1}, \Psi_{n-1}) \\ \widetilde{z}_n &= g(\widetilde{y}_n, \widetilde{z}_n) \end{aligned}$$

using $\widetilde{\mathcal{V}}(z_t, \Psi) = \kappa[(f_3\Psi + f_4)[I_{n_z} - \lambda(z_t)\Psi]^{-1}\sigma(z_t); z_t]$.
(2) By a QZ decomposition find Ψ_n in:

$$0 = f_1(\widetilde{y}_n, \widetilde{z}_n)\Psi_n + f_2(\widetilde{y}_n, \widetilde{z}_n) + (f_3\Psi_n + f_4)[g_1(\widetilde{y}_n, \widetilde{z}_n)\Psi_n + g_2(\widetilde{y}_n, \widetilde{z}_n)] + \widetilde{\mathcal{V}}_1(\widetilde{z}_{n-1}, \Psi_{n-1})$$

Appendix: Campbell-Cochrane (1999): Pricing wealth

$$pc_t^{(n)} = \ln E_t e^{m_{t+1} + \Delta c_{t+1} + pc_{t+1}^{(n-1)}}$$

Step 1. Split into certainty-equivalent and entropy terms:

$$\begin{aligned} 0 = & \ln(\beta) + (1-\gamma) E_t \Delta c_{t+1} - \gamma E_t \Delta s_{t+1} + E_t pc_{t+1}^{(n-1)} - pc_t^{(n)} \\ & + \mathcal{V}_t \left(e^{(1-\gamma)\Delta c_{t+1} - \gamma \Delta s_{t+1} + pc_{t+1}^{(n-1)}} \right) \end{aligned}$$

Appendix: Campbell-Cochrane (1999): Pricing wealth

$$pc_t^{(n)} = \ln E_t e^{m_{t+1} + \Delta c_{t+1} + pc_{t+1}^{(n-1)}}$$

Step 1. Split into certainty-equivalent and entropy terms:

$$\begin{aligned} 0 = \ln(\beta) + (1-\gamma) E_t \Delta c_{t+1} - \gamma E_t \Delta s_{t+1} + E_t pc_{t+1}^{(n-1)} - pc_t^{(n)} \\ + \mathcal{V}_t \left(e^{(1-\gamma)\Delta c_{t+1} - \gamma \Delta s_{t+1} + pc_{t+1}^{(n-1)}} \right) \end{aligned}$$

Step 2. Guess $pc_t^{(n)} = \widetilde{pc}^{(n)} + \widetilde{\psi}^{(n)} \widehat{s}_t$, hence:

$$\mathcal{V}_t \left(e^{(1-\gamma)\Delta c_{t+1} - \gamma \Delta s_{t+1} + pc_{t+1}^{(n-1)}} \right) = \left(1 - \gamma [1 + \Lambda(\widehat{s}_t)] + \widetilde{\psi}^{(n-1)} \Lambda(\widehat{s}_t) \right)^2 \frac{\sigma^2}{2}$$

Appendix: Campbell-Cochrane (1999): Pricing wealth

$$pc_t^{(n)} = \ln E_t e^{m_{t+1} + \Delta c_{t+1} + pc_{t+1}^{(n-1)}}$$

Step 3. Linearize:

$$\begin{aligned} 0 &= \ln(\beta) + (1-\gamma) E_t \Delta c_{t+1} - \gamma E_t \Delta s_{t+1} + E_t pc_{t+1}^{(n-1)} - pc_t^{(n)} \\ &\quad + \left(1 - \gamma[1 + \Lambda(\widehat{s}_t)] + \widetilde{\psi}^{(n)} \Lambda(\widehat{s}_t)\right)^2 \frac{\sigma^2}{2} \\ &\approx \widetilde{pc}^{(n-1)} - \widetilde{pc}^{(n)} + \ln(\beta e^{(1-\gamma)\mu}) + \left(1 - \gamma[1+\Lambda(0)] + \widetilde{\psi}^{(n-1)} \Lambda(0)\right)^2 \frac{\sigma^2}{2} \\ &\quad + \widetilde{\psi}^{(n-1)} \phi \widehat{s}_t - \widetilde{\psi}^{(n)} \widehat{s}_t + \gamma(1-\phi)\widehat{s}_t \\ &\quad + \left(1 - \gamma[1+\Lambda(0)] + \widetilde{\psi}^{(n-1)} \Lambda(0)\right) (\widetilde{\psi}^{(n-1)} - \gamma) \Lambda_1(0) \sigma^2 \widehat{s}_t \end{aligned}$$

Appendix: Campbell-Cochrane (1999): Pricing wealth

$$pc_t^{(n)} = \ln E_t e^{m_{t+1} + \Delta c_{t+1} + pc_{t+1}^{(n-1)}}$$

Step 4. Identify:

$$\widetilde{pc}^{(n)} = \widetilde{pc}^{(n-1)} + \ln(\beta e^{(1-\gamma)\mu}) + \left(1 - \gamma[1 + \Lambda(0)] + \widetilde{\psi}^{(n-1)}\Lambda(0)\right)^2 \frac{\sigma^2}{2}$$

$$\widetilde{\psi}^{(n)} = \widetilde{\psi}^{(n-1)}\phi + \gamma(1-\phi) + \left(1 - \gamma[1 + \Lambda(0)] + \widetilde{\psi}^{(n-1)}\Lambda(0)\right)(\widetilde{\psi}^{(n-1)} - \gamma)\Lambda_1(0)\sigma^2$$

with boundary condition $\widetilde{pc}^{(0)} = \widetilde{\psi}^{(0)} = 0$.

Appendix: Campbell-Cochrane (1999): Pricing wealth

$$rc_t^{(n)} = \ln E_t e^{m_{t+1}+\Delta c_{t+1}+rc_{t+1}^{(n-1)}}$$

Analogously, $rc_t^{(n)} = \widetilde{rc}^{(n)} + \widetilde{\varphi}^{(n)}\widehat{s}_t$ where

$$\widetilde{rc}^{(n)} = \widetilde{rc}^{(n-1)} + \ln(\beta e^{(1-\gamma)\mu}) + \left(1-\gamma[1+\Lambda(0)] + \widetilde{\varphi}^{(n-1)}\Lambda(0)\right)^2 \frac{\sigma^2}{2}$$

$$\widetilde{\varphi}^{(n)} = \widetilde{\varphi}^{(n-1)}\phi + \gamma(1-\phi) + \left(1-\gamma[1+\Lambda(0)] + \widetilde{\varphi}^{(n-1)}\Lambda(0)\right)(\widetilde{\varphi}^{(n-1)} - \gamma)\Lambda_1(0)\sigma^2$$

with $\widetilde{rc}^{(0)} = \ln\left(\sum_{n=1}^{N} e^{\widetilde{pc}^{(n)}} + e^{\widetilde{rc}^{(N)}}\right)$ and $\widetilde{\varphi}^{(0)} = \frac{\sum_{n=1}^{\infty} e^{\widetilde{pc}^{(n)}}\widetilde{\psi}^{(n)} + e^{\widetilde{rc}^{(N)}}\widetilde{\varphi}^{(N)}}{\sum_{n=1}^{N} e^{\widetilde{pc}^{(n)}} + e^{\widetilde{rc}^{(N)}}}$.

Appendix: Conventional third-order perturbations

Compare $\widetilde{pc}^{(n)}$ and $\widetilde{\psi}^{(n)}$ to a conventional third-order perturbation:

$$pc_t = \ln\left(\sum_{n=1}^{\infty} e^{pc_t^{(n)}}\right), \quad pc_t^{(n)} = n\ln(\beta e^{(1-\gamma)\mu}) + \bar{\psi}_1^{(n)}\widehat{s}_t + \bar{\psi}_2^{(n)} + \bar{\psi}_3^{(n)}\widehat{s}_t$$

where:

$$\bar{\psi}_1^{(n)} = \gamma(1-\phi^n)$$

$$\bar{\psi}_2^{(n)} = \bar{\psi}_2^{(n-1)} + \left(1-\gamma[1+\Lambda(0)] + \bar{\psi}_1^{(n-1)}\Lambda(0)\right)^2 \frac{\sigma^2}{2}$$

$$\bar{\psi}_3^{(n)} = \bar{\psi}_3^{(n-1)}\phi + \left(1-\gamma[1+\Lambda(0)] + \bar{\psi}_1^{(n-1)}\Lambda(0)\right)(\bar{\psi}_1^{(n-1)} - \gamma)\Lambda_1(0)\sigma^2$$

with boundary conditions $\bar{\psi}_2^{(0)} = \bar{\psi}_3^{(0)} = 0$.

$$\widetilde{pc}^{(n)} = \widetilde{pc}^{(n-1)} + \ln(\beta e^{(1-\gamma)\mu}) + \left(1-\gamma[1+\Lambda(0)] + \widetilde{\psi}^{(n-1)}\Lambda(0)\right)^2 \frac{\sigma^2}{2}$$

$$\widetilde{\psi}^{(n)} = \widetilde{\psi}^{(n-1)}\phi + \gamma(1-\phi) + \left(1-\gamma[1+\Lambda(0)] + \widetilde{\psi}^{(n-1)}\Lambda(0)\right)(\widetilde{\psi}^{(n-1)} - \gamma)\Lambda_1(0)\sigma^2$$

Appendix: Uniqueness

Is the solution $[\widetilde{y}, \widetilde{z}, \widetilde{\Psi}]$ unique? Are dynamics locally unique?

The matrix equation is nonlinear, so there can be multiple solutions. Each solution has locally unique dynamics if BK-like conditions are met.

For given matrix $\widetilde{\Psi}$, the point $(\widetilde{y}, \widetilde{z})$ is a saddle point of the approximate system if and only if the generalized eigenvalues

$$\alpha(\Gamma, \Xi) \doteq \{\alpha \in \mathbb{C} : \det(\Gamma\alpha - \Xi) = 0\} = \{\alpha_i, i = 1, ..., n_y + n_z\}$$

of the square matrices:

$$\Gamma \doteq \begin{bmatrix} f_4 & f_3 \\ I_{n_z} & 0 \end{bmatrix} \qquad \Xi \doteq \begin{bmatrix} -h_2(\widetilde{y}, \widetilde{z}) - \widetilde{\mathcal{V}}_1(\widetilde{z}) & -h_1(\widetilde{y}, \widetilde{z}) \\ g_2(\widetilde{y}, \widetilde{z}) & g_1(\widetilde{y}, \widetilde{z}) \end{bmatrix}$$

are such that there are n_z generalized eigenvalues with modulus within the unit circle and n_y with modulus larger than unity.