ECON501 2018

lab08

OUTLINE

1. **Octave 3.8.2**
2. The Neoclassical growth model
3. Solve F.O.C
4. dynare template
5. parameters
6. model

SOFTWARE

1. Octave 3.8.2 works with Dynare

http://www.dynare.org/documentation-and-support/quick-start

https://web.archive.org/web/20150402161713/http:/mxeoctave.osuv.de/

2. Watch video

https://www.youtube.com/watch?v=ZH-BRQaAaBU

https://www.youtube.com/watch?v=P9VYrKMAYrE

https://www.youtube.com/watch?v=RtGW-ZogMqg

TUTORIAL

1. Intro

http://fabcol.free.fr/dynare/pdf/tr_dynare.pdf ar1.mod

2. Basic

http://www3.nd.edu/~esims1/using_dynare_sp17.pdf ramsey.mod

3. Practice

http://www.sfu.ca/~kkasa/Sargent_Dynare.pdf

4. Models

http://fabcol.free.fr/dynare/

https://github.com/JohannesPfeifer/DSGE_mod

THE NEOCLASSICAL MODEL WITH FIXED LABOR

Consider a simple stochastic neoclassical model with fixed labor as the laboratory. The planner's problem can be written:

$$\max \qquad E_0 \sum_{t=0}^{\infty} \beta^t \frac{c_t^{1-\sigma} - 1}{1-\sigma}$$

s.t.

$$k_{t+1} = a_t k_t^{\alpha} - c_t + (1-\delta) k_t$$

TFP is assumed to follow a mean zero AR(1) in the log:

$$\ln a_t = \rho \ln a_{t-1} + \varepsilon_t$$

The first order conditions for this problem can be characterized with three non-linear difference equations and a transversality condition:

$$\begin{aligned}
c_t^{-\sigma} &= \beta E_t c_{t+1}^{-\sigma}(\alpha a_{t+1} k_{t+1}^{\alpha-1} + (1-\delta)) \\
k_{t+1} &= a_t k_t^{\alpha} - c_t + (1-\delta) k_t \\
\ln a_t &= \rho \ln a_{t-1} + \varepsilon_t \\
\lim_{t\to\infty} \beta^t c_t^{-\sigma} k_{t+1} &= 0
\end{aligned}$$

TRY TO IMPLEMENT COMPUTATION IN DYNARE

See github

PARAMETERS

```
parameters alpha beta delta rho sigma sigmae;

beta     = 0.99;  % discount factor

delta     = 0.025;   % appreciation rate

alpha     = 1/3;     % share of capital in production function

sigma     = 1;       % intertemporal preference parameter

rho     = 0.95;    % coefficient for AR(1) stochastic process of technology

sigmae = 0.01;  % standard error of technology shock
```

MODEL;

```
model;
exp(c)^(-sigma) = beta*exp(c(+1))^(-sigma)*(alpha*exp(z(+1))*exp(k)^(alpha-1) + (1-delta));
exp(y) = exp(z)*exp(k(-1))^(alpha);
exp(k) = exp(z)*exp(k(-1))^(alpha) - exp(c) + (1-delta)*exp(k(-1));
z = rho*z(-1) + e;
exp(y) = exp(c) + exp(I);
exp(c)^(-sigma) = beta*exp(c(+1))^(-sigma)*(1+r);
exp(R) = alpha*exp(z)*exp(k(-1))^(alpha-1);
exp(w) = (1-alpha)*exp(z)*exp(k(-1))^(alpha);
end;
```