Advanced Macroeconomics

Lecture 11: real business cycles, part three

Chris Edmond

1st Semester 2019

This class

- Solving dynamic models in Dynare
  - stochastic growth example revisited
  - working with the results

The problem to be solved

- Same stochastic growth model as in Lecture 9, planner maximizes

$$\mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t \frac{c_t^{1-\sigma}}{1-\sigma}$$

subject to resource constraints

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

given productivity

$$\log z_{t+1} = \phi \log z_t + \varepsilon_{t+1}, \qquad \varepsilon_{t+1} \sim \text{IID } N(0, \sigma_{\varepsilon}^2)$$

for some given parameters $\alpha, \beta, \delta, \phi, \sigma, \sigma_{\varepsilon}$ (here normalized $\bar{z} = 1$)

Dynare model file

The structure of a Dynare model file (`.mod`)

(1) Declare endogenous variables

(2) Declare exogenous variables

(3) Declare parameters

(4) Declare the model equations

(5) Ask Dynare to solve for the steady state

(6) Ask Dynare to solve for the the dynamics

(1) Declare endogenous variables

- From `stochastic_growth_dynare.mod` in LMS

```
// (1) declare endogenous variables

var         c, k, z;
```

- Note productivity z_t is treated as endogenous

(2) Declare exogenous variables

- It is the innovations ε_t that are fundamentally exogenous

```
// (2) declare exogenous variables (shocks)

varexo      e;
```

(3) Declare parameters

- List of parameter names

```
// (3) declare parameters

parameters alpha, beta, delta, sigma, phi, sigmaeps;
```

- Set parameter values

```
alpha    = 0.30;
beta     = 0.95;
delta    = 0.05;
phi      = 0.95;
sigma    = 1.00;
sigmaeps = 0.01;
```

Dynare notation for model equations

Resource constraint

- In usual notation

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

- In Dynare notation, *supposing we want an approximation in logs*

```
exp(c) + exp(k)
= exp(z)*(exp(k(-1))^alpha)+(1-delta)*exp(k(-1));
```

(otherwise approximation will be in levels)

- Variables chosen at t have no time argument

  Variables chosen at $t-1$ have '-1' argument

  Variables chosen at $t+1$ have '$+1$' argument

Dynare notation for model equations

Consumption Euler equation

- In usual notation, using $u'(c) = c^{-\sigma}$ and $f'(k) = \alpha k^{\alpha-1}$ etc

$$c_t^{-\sigma} = \beta \mathbb{E}_t \left\{ c_{t+1}^{-\sigma} \left[z_{t+1} \alpha k_{t+1}^{\alpha-1} + 1 - \delta \right] \right\}$$

- In Dynare notation, supposing we want an approximation in logs

```
exp(c)^(-sigma)
= beta*(exp(c(+1))^(-sigma))
*(alpha*exp(z(+1))*(exp(k)^(alpha-1))+1-delta);
```

Dynare notation for model equations

Law of motion for productivity

- In usual notation

$$\log z_{t+1} = \phi \log z_t + \varepsilon_{t+1}$$

- In Dynare notation, supposing we want an approximation in logs

```
z = phi*z(-1) + e;
```

(4) Declare the model equations

- Start block with '**model**,' then list equations, then '**end**'

```
model;

// resource constraint
exp(c) + exp(k)
= exp(z)*(exp(k(-1))^alpha)+(1-delta)*exp(k(-1));

// consumption Euler equation
exp(c)^(-sigma)
= beta*(exp(c(+1))^(-sigma))
*(alpha*exp(z(+1))*(exp(k)^(alpha-1))+1-delta);

// law of motion productivity
z = phi*z(-1) + e;

end;
```

(5) Solve for the steady state

- Solve for steady state numerically (system of nonlinear equations)

- Start block with '**initval**,' then list guess, then '**end**'

```
// (5) solve the steady state
initval;
c = 0.75;
k = 3;
z = 1;
e = 0;
end;

steady;
```

- Can enter exact solution here if you have one

- Given the model equations we've declared, steady state *in logs*

Set the shocks

- Set the variance/covariance structure of shocks

```
// specify variance of shocks

shocks;
var e = 100*sigmaeps^2;
end;
```

- Only one shock here so this is simple

(6) Solve for the dynamics

- Solve for coefficients, obtain moments, plot impulse responses etc

```
// (6) solve the dynamics
stoch_simul(order=1,irf=50);
```

- Lots of options available (see user guide for details)

Dynare output

```
STEADY-STATE RESULTS

c     0.301697
k     1.53234
z     0
```

Given the model equations we've declared, steady state in logs

Dynare output

```
POLICY AND TRANSITION FUNCTIONS
                      c                   k                  z
Constant              0.301697            1.532337                  0
k(-1)                 0.555042            0.890501                  0
z(-1)                 0.493721            0.180782           0.950000
e                     0.519706            0.190296           1.000000
```

Solutions of the form

$$\begin{aligned}\log c_t &= \log \bar{c} + \psi_{ck} \log k_{t-1} + \psi_{cz} \log z_t \\ &= \log \bar{c} + \psi_{ck} \log k_{t-1} + (\psi_{cz}\phi) \log z_{t-1} + \psi_{cz}\varepsilon_t\end{aligned}$$

and similarly

$$\log k_t = \log \bar{k} + \psi_{kk} \log k_{t-1} + (\psi_{kz}\phi) \log z_{t-1} + \psi_{kz}\varepsilon_t$$

- Recall from Lecture 9 where we did things ‘by hand’

$$\begin{pmatrix} \psi_{kk} & \psi_{kz} \\ \psi_{ck} & \psi_{cz} \end{pmatrix} = \begin{pmatrix} 0.89 & 0.19 \\ 0.56 & 0.52 \end{pmatrix}$$

- So we are getting the same answers.

Dynare output

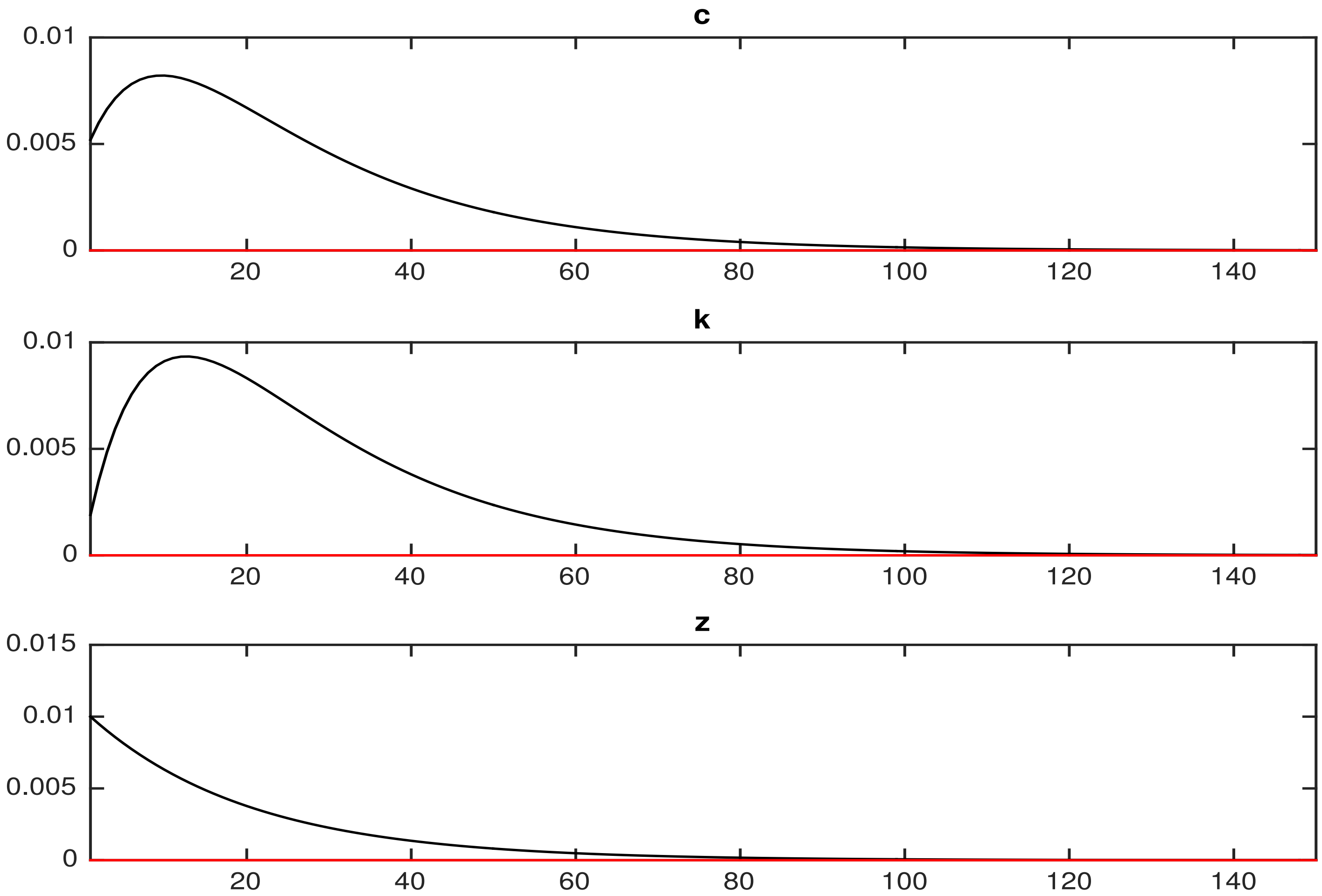
```
THEORETICAL MOMENTS

VARIABLE        MEAN       STD. DEV.    VARIANCE
c               0.3017      0.0404      0.0016
k               1.5323      0.0464      0.0022
z               0.0000      0.0320      0.0010
```

Dynare output

MATRIX OF CORRELATIONS

Variables	c	k	z
c	1.0000	0.9862	0.9279
k	0.9862	1.0000	0.8533
z	0.9279	0.8533	1.0000

Dynare output



Impulse responses of log consumption and log capital to 1 standard deviation shock to log productivity.

Matlab workspace

Name	Size	Bytes	Class	Attributes
M_	1x1	8280	struct	global
alpha	1x1	8	double	
beta	1x1	8	double	
c_e	150x1	1200	double	
delta	1x1	8	double	
estimation_info	1x1	33616	struct	global
ex0_	0x0	0	double	global
info	1x1	8	double	
k_e	150x1	1200	double	
oo_	1x1	10768	struct	global
options_	1x1	74358	struct	global
phi	1x1	8	double	
sigma	1x1	8	double	
sigmaeps	1x1	8	double	
var_list_	0x0	0	double	
ys0_	0x0	0	double	global
z_e	150x1	1200	double	

Matlab workspace

- Parameters that we supplied

```
alpha, beta, delta, phi, sigma, sigmaeps
```

- Impulse responses of each endogenous variable to each shock

```
c_e, k_e, z_e
```

- Specialized Matlab *structures* that store key results

```
M_, options_, oo_, ...
```

(these are also saved in `stochastic_growth_dynare_results.mat` created by running Dynare on `stochastic_growth_dynare.mod`)

Structures

- Structures created by Dynare

  `M_`, information about the model

  `options_`, options used during computation

  `oo_`, results of the computation

- Get structure entries by '`structure_.entry`', for example

  ```
  oo_.steady_state
  ```

  recovers the steady state values given on slide 15 above

- Structure entries can be relatively complicated objects

Results of the computation structure

- For example, the entries of `oo_` are

```
         exo_simul: [3x1 double]
        endo_simul: []
                dr: [1x1 struct]
  exo_steady_state: 0
exo_det_steady_state: []
     exo_det_simul: []
      steady_state: [3x1 double]
           gamma_y: {6x1 cell}
              mean: [3x1 double]
               var: [3x3 double]
          autocorr: {1x5 cell}
              irfs: [1x1 struct]
```

- Entry `oo_.dr` is another structure . . .

Decision rules structure

... and the entries of `oo_.dr` (for 'decision rules') are

```
                      order_var: [3x1 double]
                  inv_order_var: [3x1 double]
                         kstate: [4x4 double]
transition_auxiliary_variables: []
                             ys: [3x1 double]
                            ghx: [3x2 double]
                            ghu: [3x1 double]
                      state_var: [2 3]
                         eigval: [4x1 double]
                             gx: [2x2 double]
                             Gy: [2x2 double]
```

Decision rules structure

- And finally `oo_.dr.ghx` is the `3x2` matrix

```
0.8905    0.1808
     0    0.9500
0.5550    0.4937
```

while `oo_.dr.ghu` is the `3x1` matrix

```
0.1903
1.0000
0.5197
```

- These are the coefficients from the solution given on slide 16 above. We can use these and other Dynare output for our analysis

Embed in Matlab `.m` file

- Better to run the Dynare `.mod` file from a Matlab `.m` file
- From `start_dynare_example.m` in LMS

```
close all;

%%%%% run dynare on .mod file
%     suppress log file and warnings

dynare stochastic_growth_dynare.mod nowarn nolog
```

- Can then use Dynare output for Matlab plots, simulations etc

Embed in Matlab `.m` file

- Tricky bookkeeping to extract coefficients (sigh)

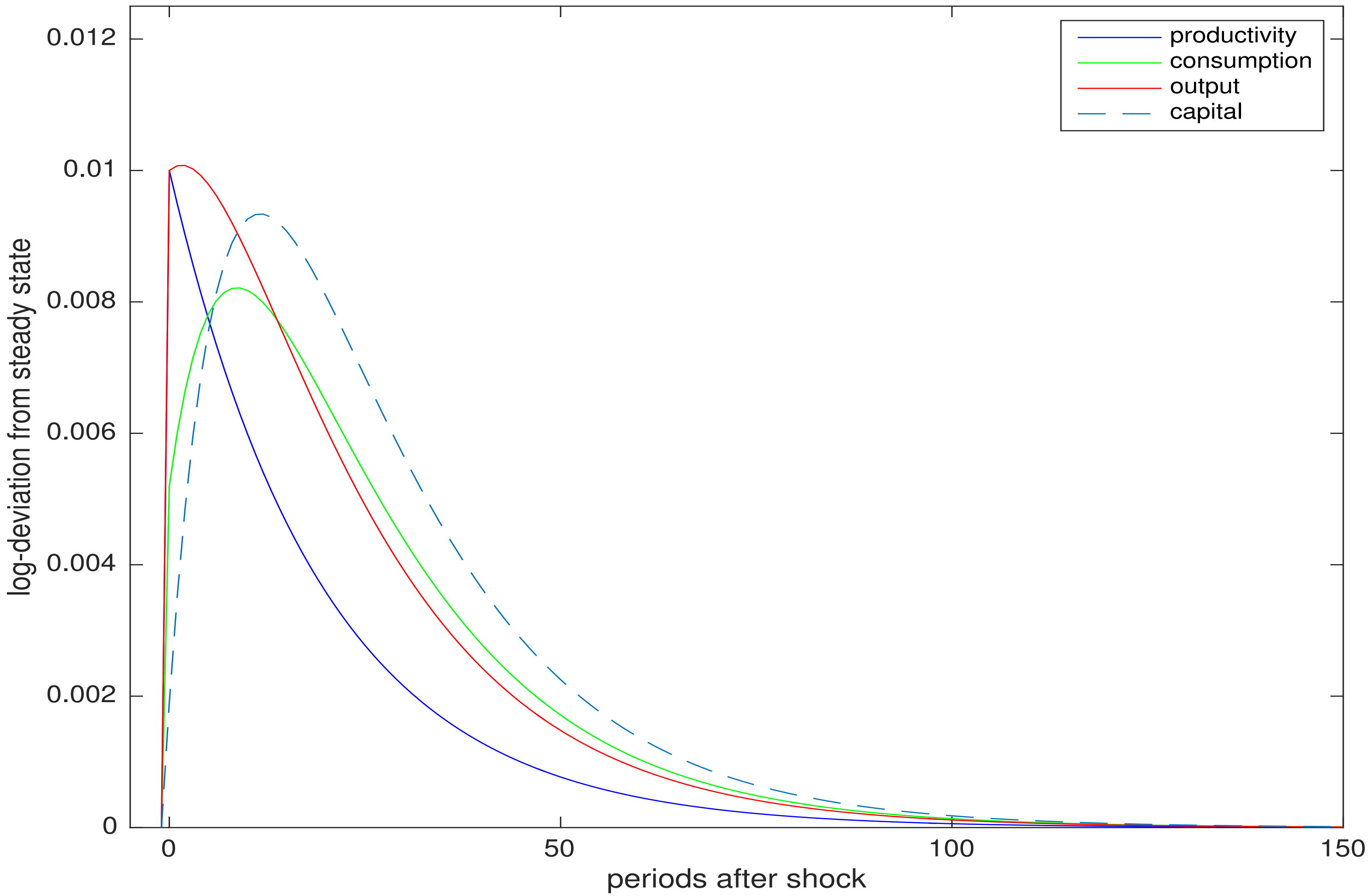
```
%%%%% recover coefficients from dynare structures
%%%%% my ordering

n_c = 1; n_k = 2; n_z = 3;

%%%%% matrices for state-space representation
% s=As(-1) + Be, x=C*s(-1) + De, s = (k,z), x=c etc

A = [oo_.dr.ghx(oo_.dr.inv_order_var(n_k),:);
     oo_.dr.ghx(oo_.dr.inv_order_var(n_z),:)];

B = [oo_.dr.ghu(oo_.dr.inv_order_var(n_k),:);
     oo_.dr.ghu(oo_.dr.inv_order_var(n_z),:)];

C = [oo_.dr.ghx(oo_.dr.inv_order_var(n_c),:)];

D = [oo_.dr.ghu(oo_.dr.inv_order_var(n_c),:)];
```

Embed in Matlab `.m` file

- Can then use coefficients in usual way

```
%%%%% in lecture notation

psi_kk = A(1,1);
psi_kz = B(1); % = A(1,2)/A(2,2)

psi_ck = C(1);
psi_cz = D(1);
```

Constructed from Dynare output



Next class

- Permanent shocks in the RBC model
  - random walks, stochastic trends etc