DGE–CRED Practice Session 2: Implementation of a Model in Dynare

Andrej Drygalla, Katja Heinisch and Christoph Schult* | August 2020
Halle Institute for Economic Research

Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) GmbH

IWH Halle Institute for Economic Research
Member of the Leibniz Association

On behalf of:
Federal Ministry for the Environment, Nature Conservation and Nuclear Safety
of the Federal Republic of Germany

* Research assistance by Yoshiki Wiskamp is greatly acknowledged.

Outline

1. Task 1: Derive the equations
2. Task 2: Find the steady state
3. Task 3: Declare the model variables and parameters
4. Task 4: Declare the model equations
5. Task 5: Implement the steady state routine
6. Task 6: Simulation

Outline

1. Task 1: Derive the equations

Task 1: Derive the equations for the Neoclassical Growth Model

- Households maximize lifetime utility subject to their budget constraint

$$\max_{\{c_t,\, k_{t+1}\}_{t=1}^{\infty}} = \sum_{t=1}^{\infty} \beta^{\,t-1} \, \frac{c_t^{1-\sigma}}{1-\sigma}$$

$$\text{s.t.} \quad c_t + k_{t+1} = A_t \, k_t^{\alpha} + (1-\delta)\, k_t$$

Solution Task 1: Derive the equations for the Neoclassical Growth Model

- the Lagrangian of the problem is

$$\max_{\{c_t, k_{t+1}, \lambda_t\}_{t=1}^{\infty}} \mathcal{L}_t = \sum_{t=1}^{\infty} \beta^{t-1} \left\{ \frac{c_t^{1-\sigma}}{1-\sigma} + \lambda_t \left(A_t k_t^{\alpha} - c_t - k_{t+1} + (1-\delta) k_t \right) \right\}$$

- first order conditions are

$$\begin{aligned} \lambda_t &= c_t^{-\sigma} \\ \lambda_t &= \beta \lambda_{t+1} (\alpha A_{t+1} k_{t+1}^{\alpha-1} + 1 - \delta) \\ c_t + k_{t+1} &= A_t k_t^{\alpha} + (1-\delta) k_t \end{aligned}$$

Outline

2. Task 2: Find the steady state

Task 2: Find the steady state of the model for $A = 1$

$$\lambda_t = c_t^{-\sigma} \tag{1}$$

$$\lambda_t = \beta\, \lambda_{t+1}(\alpha\, A_{t+1}\, k_{t+1}^{\alpha-1} + 1 - \delta) \tag{2}$$

$$c_t + k_{t+1} = A_t\, k_t^{\alpha} + (1-\delta)\, k_t \tag{3}$$

Solution Task 2: Find the steady state of the model for $A = 1$

$$A = 1 \tag{4}$$

$$k = \left\{ \frac{1}{\alpha A} \left(\frac{1}{\beta} + \delta - 1 \right) \right\}^{\frac{1}{\alpha - 1}} \tag{5}$$

$$c = A k^{\alpha} - \delta k \tag{6}$$

$$\lambda = c^{-\sigma} \tag{7}$$

Outline

3. Task 3: Declare the model variables and parameters

Task 3: Declare the model variables and parameters in Dynare and assign values to the parameters

- variables: c, k, A, λ
- parameters: β, δ, α, σ

Solution Task 3: Declare the model variables and parameters in Dynare

```
var c k lamb;
varexo A;
parameters alpha_p beta_p sigma_p delta_p;

alpha_p = 0.5;
beta_p = 0.95;
sigma_p = 0.5;
delta_p = 0.02;
```

Outline

4. Task 4: Declare the model equations

Task 4: Declare the model equations in Dynare.

$$\lambda_t = c_t^{-\sigma} \tag{8}$$

$$\lambda_t = \beta\, \lambda_{t+1}(\alpha\, A_{t+1}\, k_{t+1}^{\alpha-1} + 1 - \delta) \tag{9}$$

$$c_t + k_{t+1} = A_t\, k_t^{\alpha} + (1-\delta)\, k_t \tag{10}$$

Solution Task 4: Declare the model equations in Dynare.

```
model;
lamb = c^(-sigma_p)
c + k = A*k(-1)^alpha_p + (1-delta_p)*k(-1);
lamb = beta_p*lamb(+1)*(alpha_p*A(+1)*k^(alpha_p-1) + 1 - delta_p);
end;
```

Outline

5. Task 5: Implement the steady state routine

Task 5: Implement the steady state routine in a steady state file.

- You need to create a steady state file called `ModFileName_steady_state.m`.
- Use the template `SteadyStateTemplate.m` file.

Solution Task 5: Implement the steady state routine in a steady state file.

```
function [ys,params,check] = NGM_steadystate(ys,exo,M_,options_)
....
%% Step 2: Determine the steady state

k = ((1-beta_p*(1-delta_p))/(beta_p*alpha_p*A))^(1/(alpha_p-1));
c = A * k^alpha_p-delta_p*k;
lamb = c^(-sigma_p);
% ... steady state is now determined.
...

end
```

Outline

6. Task 6: Simulation

Task 6: Simulate a permanent increase in productivity by 10%

1. Define the initial and terminal steady state.
2. Plot the trajectories of the endogenous variables.

Solution Task 6: Simulate a permanent increase in productivity by 10%

Define the initial and terminal steady state.

```
// Section: Perfect Foresight Setup in Dynare
initval;
A = 1;
end;
steady;
check;

endval;
A = 1.1;
end;
steady;
check;

// Note: Deactivate this section if only the steady state should be computed.
// Conduct deterministic simulation using perfect foresight:
perfect_foresight_setup(periods=100);
perfect_foresight_solver;
```

Solution Task 6: Simulate a permanent increase in productivity by 10%.

Plot the trajectories of the endogenous variables.

```
// Optional: plot graphs
rplot c;
rplot k;
```