DGE–CRED Practice Session 3: Macro Processor

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: Replace the Lagrange multiplier
2. Task 2: Add climate variables
3. Task 3: Add damage function
4. Task 4: Include climate related damages
5. Task 5: Modify the steady state
6. Task 6: Simulate

Outline

1. Task 1: Replace the Lagrange multiplier

Task 1: Replace the Lagrange multiplier using local variables (#)

- Use the mod file from the previous practical session.
- Modify the `var` and `model` block accordingly.
- Is it necessary to change the steady state file?

Solution Task 1: Replace the Lagrange Multiplier using local variables (#)

```
#lamb = c^(-sigma_p);
#lambp1 = c(+1)^(-sigma_p);
c + k = A*k(-1)^alpha_p + (1-delta_p)*k(-1);
lamb = beta_p*lambp1*(alpha_p*A(+1)*k^(alpha_p-1) + 1 - delta_p);
```

Outline

2. Task 2: Add climate variables

Task 2: Use the Macro language to add an arbitrary number of climate variables

- Call the exogenous variables `CV_1` ... `CV_N`.
- Define an integer specifying the number of variables `ClimateVariables`.

Solution Task 2: Use the Macro language to add an arbitrary number of climate variables

```
@# define ClimateVariable = 2

varexo A
@# if ClimateVariable > 0
   @# for z in 1:ClimateVariable
      CV_@{z}
   @# endfor
@# endif
;
```

Outline

3. Task 3: Add damage function

Task 3: Use the Macro language to add climate related damages

- Define an array of climate coefficients called `ClimateCoefficient`.
- Add a damage function variable to the model variables called `D`, if there exists at least one climate variable.
- Add damage function coefficients as parameters called `theta_n_p`, if there exists at least one climate variable.
- Assign values to the damage function coefficients stored in `ClimateCoefficient`, if there exists at least one climate variable.

Solution Task 3: Use the Macro language to add climate related damages

```
@# define ClimateCoefficient = [0.1, 0]

var c k
@# if ClimateVariable > 0
D
@# endif
;

parameters alpha_p beta_p sigma_p delta_p inbCV_p
@# if ClimateVariable > 0
   @# for z in 1:ClimateVariable
      theta_@{z}_p
   @# endfor
@# endif
    ;
```

Outline

4. Task 4: Include climate related damages

Task 4: Modify the model block to include climate related damages using the Macro language

$$
\begin{aligned}
D_t &= exp(-\sum_{n=1}^{N} \theta_n CV_n), && (1)\\
c_t + k_{t+1} &= A_t \left(1 - D_t\right) k_t^{\alpha} + \left(1 - \delta\right) k_{t-1}, && (2)\\
\lambda_t &= \beta \, \lambda_{t+1} \left(\alpha A_{t+1} \left(1 - D_{t+1}\right) k_{t+1}^{\alpha} + \left(1 - \delta\right)\right. && (3)\\
& && (4)
\end{aligned}
$$

Solution Task 4: Modify the model block to include climate related damages using the Macro language

```
model;
#lamb = c^(-sigma_p);
#lambp1 = c(+1)^(-sigma_p);
@# if ClimateVariable > 0
   D = 1 - exp(-(
   @# for z in 1:ClimateVariable
      + CV_@{z} * theta_@{z}_p
   @# endfor
   ));
@# endif
c + k = A*(1 - D)*k(-1)^alpha_p + (1-delta_p)*k(-1);
lamb = beta_p*lambp1*(alpha_p*A(+1)*(1 - D(+1))*k^(alpha_p-1) + 1 - delta_p);
end;
```

Outline

5. Task 5: Modify the steady state

Task 5: Modify the steady state file to calculate a new long-run equilibrium

- You need to define damages `D` as a function of `CV_n`.
- Use the command `eval(['CV_' num2str(icocv)])` to define the damages.
- Further use a temporary expression to sum up all single components of the damage functions.

```
temp = 0;
if inbCV_p > 0
   for icocv = 1:inbCV_p
       temp = temp + ...;
   end
end
```

Solution Task 5: Modify the steady state file to calculate a new long-run equilibrium

```
temp = 0;
if inbCV_p > 0
    for icocv = 1:inbCV_p
       temp = temp + eval(['CV_' num2str(icocv)]) * eval(['theta_' num2str(icocv) '_p']);
    end
end
D = 1 - exp(-temp);
k = ((1-beta_p*(1-delta_p))/(beta_p*alpha_p * (1 - D) *A))^(1/(alpha_p-1));
c = (1 - D) * A*k^alpha_p-delta_p*k;
lamb = c^(-sigma_p);
```

Outline

6. Task 6: Simulate

Task 6: Simulate the effect of an increase in temperature by one degree Celsius after 100 periods

- The first climate variable represents temperature.
- A one degree increase in temperature leads to a 10% reduction in total factor productivity. Specify the coefficient accordingly.
- Assume that temperature follows a step function and increases by 0.1 degree at the end of every decade. In order to specify the evolution of temperature one can use the block `shocks` described in the manual of dynare.

Solution Task 6: Simulate the effect of an increase in temperature by one degree Celsius after 100 periods

```
initval;
A = 1;
CV_1 = 0;
end;
steady;
check;

endval;
A = 1;
CV_1 = 1;
end;
steady;
check;
```

Solution Task 6: Simulate the effect of an increase in temperature by one degree Celsius after 100 periods. (cont.)

```
shocks;
var CV_1;
periods 1:9 10:19 20:29 30:39 40:49 50:59 60:69 70:79 80:89 90:99;
values 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9;
end;
// Conduct deterministic simulation using perfect foresight:

perfect_foresight_setup(periods=200);
perfect_foresight_solver;
```