pictures/BackgroundGIZ.jpg

Dynamic General Equilibrium Model for Climate Resilient Economic Development

Andrej Drygalla and Christoph Schult | June 2020
Halle Institute for Economic Research

giz Deutsche Gesellschaft
für Internationale
Zusammenarbeit (GIZ) GmbH

pictures/IWH_Logo_RGB_DE_Grossformat.png

pictures/seperationbar.png

Outline

1. Introduction to Dynare
2. DGE-CRED Model
3. Model Simulation and Calibration

Outline

1. Introduction to Dynare
   - What is Dynare?
   - Neoclassical Growth Model
   - Setting up a Model File
   - Deterministic Simulation
   - Perfect Foresight Setup
   - Steady State
   - Remarks
   - Examples
   - Newton Method
   - Macro Processor

1.1 What is Dynare?

- Dynare is an open-source program for dynamic general equilibrium modeling,
- mainly a collection of different functions written for Matlab,
- preprocessor translates mod files into matlab code.

1.2 Neoclassical Growth Model (1)

- In order to illustrate the basic structure of Dynare, this section considers the following neoclassical growth model:

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

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

- The resulting first order conditions are:

$$c_t^{-\sigma} = \beta c_{t+1}^{-\sigma} (\alpha A_{t+1} k_t^{\alpha - 1} + 1 - \delta)$$

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

pictures/sepera

1.2 Neoclassical Growth Model (2)

- The steady state can be obtained analytically and is characterized by the following two equations:

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

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

- The neoclassical growth model considered in these slides, as well as the DGE-CRED model are deterministic models. There are no stochastic elements requiring expectancy terms or probability distributions. Therefore, the remained of this introduction to Dynare focuses on this class of models.

Outline

- What is Dynare?
- Neoclassical Growth Model
- Setting up a Model File
  - Model File
  - Variables and Parameters
  - The `model` block
- Deterministic Simulation
- Perfect Foresight Setup
- Steady State
- Remarks
- Examples
- Newton Method
- Macro Processor

1.3.1 Model File

- A model file (filename.mod) contains commands and blocks. Each command and each element of a block is terminated by a semicolon (;). Blocks are terminated by `end;` .
- The model file used for this introduction to Dynare is entername.mod and it is recommendable to take a look at that file while proceeding with these slides.
- Code lines within the mode file can be enabled using %, // or /* ... */.
- In order to run a model file (i) the Dynare path has to be added to the search path of MATLAB and (ii) Dynare executed to run the model in MATLAB:

```
addpath('C:\dynare\4.6.1\matlab')
dynare filename
```

1.3.2 Variables and Parameters (1)

- At the beginning of each model file, the endogenous (`var`) and exogenous (`varexo`) variables as well as parameters (`parameters`) have to be defined:

```
var = c k;
varexo = A;
parameters alpha_p beta_p gamma_p delta_p;
```

- Then, the parameter values have to be assigned:

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

1.3.2 Variables and Parameters (2)

- Optionally, a LaTeX name and long name can be assigned to a variable using the convention:

  ```
  variable_name $tex_name$ (long_name = ’quoted_string’).
  ```

  ```
    var
    k $k$ (long_name = ’capital’),
    c $c$ (long_name = ’consumption’),
    ;
    varexo
    A $A$ (long_name = ’technology’),
    ;
  ```

1.3.2 Variables and Parameters (3)

- There are some restrictions to kept in mind, when choosing variable and parameter names in Dynare. First, variables and parameters must not have the same name as Dynare commands or build-in functions. Second, minimize interference with MATLAB or Octave functions. In particular, do not use correctly-spelled greek letters like `beta`, because there are MATLAB functions of the same name.
- Note that by convention in Dynare, the time indice of a variable reflects when this variable is decided. The typical example is for capital stock: since the capital stock entering the production function in the current period is decided in the previous period, the capital stock becomes `k(-1)`, and the law of motion of capital must be written: `k = i + (1-delta)*k(-1)`.[1]

pictures/sepera

[1] This convention can be modified using the `predetermined_variables` setting.

1.3.3 The `model` block

- A model is declared inside a `model` block. In general, there must be as many equations as there are endogenous variables in the model. A great advantage of using Dynare is that the equations can be written almost as on paper.

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

- Now, the model is set up and one can begin with the deterministic simulation.

pictures/sepera

1.4 Deterministic Simulations in Dynare (1)

- The deterministic simulation builds up on the concept of perfect foresight, in which agents have full knowledge and perfectly anticipate all future shocks.
- Concretely, at period 1:
  - agents learn the value of all future shocks;
  - since there is shared knowledge of the model and of future shocks, agents can compute their optimal plans for all future periods;
  - optimal plans are not adjusted in periods 2 and later

    $\Rightarrow$ the model behaves as if it were deterministic.
- Cost of this approach: the effect of future uncertainty is not taken into account.
- Advantage: numerical solutions can be computed exactly (up to rounding errors) and nonlinearities are fully taken into account.

1.4 Deterministic Simulations in Dynare (2)

- The general problem in the deterministic, perfect foresight, case can be expressed as:

  $$f(y_{t+1}, y_t, y_{t-1}, u_t) = 0,$$

  where y is the vector of endogenous variables and u the vector of exogenous shocks.
- Identification rule: there must be as many equations in $f(...)$ as there are endogneous variables in y.

1.4 Deterministic Simulations in Dynare (3)

- Accordingly, the general perfect foresight problem for the neoclassical growth model is:

$$y_t = \begin{pmatrix} c_t \\ k_t \end{pmatrix}$$

$$u_t = A_t$$

$$f(y_{t+1}, y_t, y_{t-1}, u_t) = \begin{pmatrix} c_t^{-\sigma} - \beta c_{t+1}^{-\sigma}(\alpha A_{t+1} k_t^{\alpha-1} + 1 - \delta) \\ c_t + k_t - A_t k_{t-1}^{\alpha} - (1-\delta) k_{t-1} \end{pmatrix}$$

1.4 Deterministic Simulations in Dynare (4)

- The aim of a deterministic simulation is to examine the trajectories of the model variables over time $t = 1, ..., T$.
- Consequently, the stacked system for a perfect foresight simulation over T periods becomes a two-boundary value problem:

$$\begin{cases} f(y_2, y_1, y_0, u_1) = 0 \\ f(y_3, y_2, y_1, u_2) = 0 \\ \vdots \\ f(y_{T+1}, y_T, y_{T-1}, u_T) = 0 \end{cases},$$

where y_0 and y_{T+1} as well as $u_1, ..., u_T$ are given.

- Dynare uses a Newton-type methode to solve this stacked system.

1.4 Deterministic Simulations in Dynare (5)

- The previously described technique numerically computes trajectories for given shocks over a *finite* number of periods.
- If one is rather interested in solving an infinite-horizon problem, the easiest way is to approximate the solution by a finite-horizon problem (large T). The drawback of this approach is that the solution is specific to a given sequence of shocks, and not generic.
- In case there is more than one lead or lag, Dynare automatically transforms the model in the form with one lead and one lag using auxiliary variables. For example, if there is a variable with two leads x_{t+2}:
  - create a new auxiliary variable a
  - replace all occurrences of x_{t+2} by a_{t+1}
  - add a new equation: $a_t = x_{t+1}$

Outline

- What is Dynare?
- Neoclassical Growth Model
- Setting up a Model File
- Deterministic Simulation
- Perfect Foresight Setup
  - Getting stated with perfect foresight
  - The `initval` block
  - The `endval` block
  - The `histval` block
  - Shocks on exogenous variables
- Steady State
- Remarks
- Examples
- Newton Method
- Macro Processor

1.5.1 Getting stated with perfect foresight

- In order to perform the deterministic simulation, one has specify (i) the constraints of the stacked system y_0, y_{T+1} and $u_1, \dots, u_T$ and (ii) provide an initial guess $y_1, \dots, y_T$ for the Newton algorithm.
- The path for the endogenous and exogenous variables are stored in two MATLAB/Octave matrices:
  - `oo_.endo_simul = (`$y_0 \;\; y_1 \;\; \dots y_T \;\; y_{T+1}$`)`
  - `oo_.exo_simul' = (`$y_0 \;\; y_1 \;\; \dots y_T \;\; y_{T+1}$`)`
- The `perfect_foresight_setup` initializes those matrices, given the `shocks`, `initval`, `endval` and `histval` blocks.
- Then, the `perfect_foresight_solver` replaces $y_1, \dots, y_T$ by the solution.

pictures/sepera

1.5.2 The `initval` block (1)

- In presence of the `initval` and absence of other blocks, both the `oo_.endo_simul` and `oo_.exo_simul` variables storing the endogenous and exogenous variables will be filled with the values provided by this block. If there are no other blocks present, it will therefore provide the initial and terminal conditions for all the endogenous and exogenous variables, because it will also fill the last column/row of these matrices. For the intermediate simulation periods it thereby provides the starting values for the solver.
- It is important to be aware that if some variables, endogenous or exogenous, are not mentioned in the `initval` block, a zero value is assumed.

pictures/sepera

1.5.2 The `initval` block (2)

- Example: Let us assume that we want to set $c_0 = c_{T+1} = 4$ and $k_0 = k_{T+1} = 20$ in the neocalssical growth model. Furthermore, we assume that TFP is constant at $A_t = 1$.

```
initval;
c = 4;
k = 20;
A = 1;
end;
```

- In order to run the deterministic simulation for $T = 100$ periods, enter:

```
perfect_foresight_setup(periods=100);
perfect_foresight_solver;
```

pictures/sepera

1.5.2 The `initval` block (3)

- The following trajectories are obtained:

- Comments:
  - As consumption is a forward looking variable in this model, its initialization c_0 does not affect the trajectory. However, its terminal value does.
  - The opposite holds for capital.

1.5.3 The `endval` block (1)

- In the absence of an `initval` block, the `endval` block fills both `oo_.endo_simul` and `oo_.exo_simul`. In this case it, therefore, has the same effect as if only an `initval` block was present.
- However, if an `initval` and `endval` block are both present, the former assigns the initial conditions in $t = 0$ while the latter provides the terminal conditions in $t = T + 1$ as well as the initial guess for the perfect foresight solver.
- Example: Let us assume that we want to set $c_0 = 4$, $c_{T+1} = 6$, $k_0 = 20$, $k_{T+1} = 30$ and $A_t = 1$.

1.5.3 The `endval` block (2)

- Leading the following code:

```
initval;
c = 4;
k = 20;
A = 1;
end;

endval;
c = 6;
k = 30;
A = 1;
end;
```

1.5.3 The `endval` block (3)

- The following trajectories are obtained:
- Comment: As in the previous example, consumption is forward looking, while capital is backward looking.

pictures/sepera

1.5.4 The `histval` block (1)

- The usage of the `histval` block is particularly interesting, if there are variables with more than one lead or lag.
- In a determinsitic simulation the `histval` block must be combined with an `initval` block. A `histval` and `endval` block cannot be combined.
- The `histval` block assigns the initial condition, while the `initval` block provides the terminal condition and inital guess for the perfect foresight solver.

1.5.4 The `histval` block (2)

- The previous example can, therefore, also be implemented by:

```
histval;
c(0) = 4;
k(0) = 20;
A(0) = 1;
end;

initval;
c = 6;
k = 30;
A = 1;
end;
```

pictures/sepera

1.5.5 Shocks on exogenous variables (1)

- For deterministic simulations, the `shocks` block specifies temporary changes in the value of exogenous variables. For permanent shocks, use an `endval` block.
- It is possible to specify shocks which last several periods and which can vary over time. The `periods` keyword accepts a list of several dates or date ranges, which must be matched by as many shock values in the `values` keyword.
- Note that a range in the `periods` keyword can be matched by only one value in the `values` keyword. If values represents a scalar, the same value applies to the whole range. If values represents a vector, it must have as many elements as there are periods in the range.

1.5.5 Shocks on exogenous variables (2)

- The `shock` block has the following structure:

```
shocks;
var ... ;
periods ... ;
values ... ;
end;
```

- Examples using the `shock` block will be examined at the end of this section.

1.6 Steady State

- A steady state, $\bar{y}$, for the model satisfies:

$$f(\bar{y}, \bar{y}, \bar{y}, \bar{u}) = 0.$$

- Note that a steady state is conditional to:
  - The steady state values of exogenous variables $\bar{u}$.
  - The values of parameters (implicit in the above definition).
- Even for a given set of exogenous and parameter values, some (nonlinear) models have several steady states.
- There are three approaches in Dynare to calculate the steady state.
- The steady state values are stored in the MATLAB/Octave matrix `oo_.steady_state`.

pictures/sepera

1.6 Steady State: Approach 1

- Idea: Provide an initial guess for the steady state in the `initval` block and then conduct the steady state calculation using `steady`.

```
initval;
...
end;
steady;
```

- The obtained steady state values are then written into `oo_.endo_simul` and `oo_.exo_simul`, unless instructed otherwise. An `initval` block followed by `stedy`, therefore, has the same effect as an `initval` block containing the steady state values.

pictures/sepera

1.6 Steady State: Approach 2

- Idea: Use a `steady_state_model` block, in which the steady state values are calculated.

  ```
  steady_state_model;
  ...
  end;
  ```

- The obtained steady state values are then written into `oo_.endo_simul` and `oo_.exo_simul`, unless instructed otherwise.
- In cases where the steady state can be solved analytically, using a `steady_state_model` block is a suitable approach.

1.6 Steady State: Approach 3

- Idea: Use an explicit steady state file, which is an external MATLAB-file that must conform with a certain structure and naming convention: `NAMEofMODfile_steadystate.m`.
- In this steady state file, you must provide the exact steady state values as in the case of the `steady_state_model` block.
- Advantage: flexibility, can call build-in MATLAB functions, allows for changing parameters to take parameter dependencies into account without resorting to model-local variables.
- Drawback: the additional flexibility offered by a steady state file increases the scope for errors.
- Note: A steady state file is used in the DGE-CRED model.

1.7 Remarks on the usage of perfect foresight

- Because of the various functions of the `initval`, `endval` and `histval` blocks, it is stronglyrecommended to check the constructed `oo_.endo_simul` and `oo_.exo_simul` variables after running `perfect_foresight_setup` and before running `perfect_foresight_solver` to see whether the desired outcome has been achieved.
- Instead of using `perfect_foresight_setup` and `perfect_foresight_solver` the command `simul(periods = T)` can also be used.
- The following slides cover some examples based on the neoclassical growth model. It is recommendable to take a look at the file entername.mod while proceeding with these examples.

pictures/sepera

1.8 Example No. 1

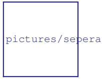
- Scenario 1: Return to equilibrium starting from $k_0 = 0.5\bar{k}$.

```
...
steady;

ik = varlist_indices(’k’,M_.endo_names);
kstar = 00_.steady_state(ik);

histval;
k(0) = kstar/2;
end;
...
```

1.8 Example No. 2

- Scenario 2: The economy starts from the steady state. There is an unexpected positive productivity shock at the beginning of period 1: $A_1 = 1.1$.

```
...
steady;

shocks;
var A;
periods 1;
values 1.1;
end;
...
```



1.8 Example No. 3

- Scenario 3: The economy starts from the steady state. There is a sequence of shocks to A_t: 10% in period 5 and an additional 5% during the 4 following periods.

```
...
steady;

shocks;
var A;
periods 4, 5:8;
values 1.1, 1.05;
end;
...
```

1.8 Example No. 4

- Scenario 4: The economy starts from the initial steady state. In period 1, TFP increases by 10% permanently.

```
...
initval;
A = 1;
end;
steady;

endval;
A = 1.1;
end;
steady;
...
```

pictures/sepera

1.8 Example No. 5

- Scenario 5: The economy starts from the initial steady state. In period 6, TFP increases by 10% permanently.
  - Same `initval` and `endval` blocks as in Scenario 4.
  - A shock block is used to maintain TFP at its initial level during periods 1-5.

```
...
shocks;
var A;
periods 1:5;
values 1;
end;
...
```

1.9 Newton Method (1)

- The following slides aim to provide an intuition for the Newton method used by the perfect foresight solver in Dynare.
- Start from an initial guess $Y^{(0)}$, where $Y = [y'_1 \quad y'_2 \dots y'_T]'$.
- Iterate: Updated solutions $Y^{(k+1)}$ are obtained by solving a linear system

$$F(Y^k) + \left[\frac{\partial F}{\partial Y}\right](Y^{(k+1)} - Y^{(k)}) = 0.$$

1.9 Newton Method (2)

- Terminal condition for the solver:

$$\|Y^{(k+1)} - Y^{(k)}\| < \epsilon_Y$$

or

$$\|F(Y^{(k)})\| < \epsilon_F$$

- Convergence may never happen if function is ill-behaved or initial guess $Y^{(0)}$ is too far from solution.

  $\Rightarrow$ to avoid an infinite loop, abort after a given number of iterations

1.9 Newton Method (3)

- The following options to the `perfect_foresight_solver` can be used to control the Newton algorithm:
  - `maxit`: Maximum number of iterations before aborting.
  - `maxit`: Maximum number of iterations before aborting (default: 50).
  - `tolf`: Convergence criterion based on function value (ϵ_F) (default: 10^{-5}
  - `tolx`: Convergence criterion based on change in function argument (ϵ_Y) (default: 10^{-5}).
  - `stack_solver_algo`: select between the different flavors of Newton algorithms.

Outline

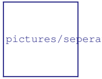
- What is Dynare?
- Neoclassical Growth Model
- Setting up a Model File
- Deterministic Simulation
- Perfect Foresight Setup
- Steady State
- Remarks
- Examples
- Newton Method
- Macro Processor
    - Macro-processing language
    - Introduction to the Syntax
    - Common use
    - Example ?

1.10.1 Macro-processing language (1)

- So far, we have ecountered the **Dynare language** (used in .mod files), which is well suited for many economic models.
  - The Dynare language is a markup language for MATLAB/Octave that defines models,
  - but itself lacks a programmatic element.
- The **Dynare macro language** adds a programmatic element to Dynare.
  - It allows to replicate blocks of equations through loops (`for` structure), conditionally executing some code (`if/then/else` structure), writing indexed sums or products inside equations, etc.
  - It is used to speed model development,
  - and useful in various situations. Examples: Multi-country models, creation of modular .mod files, variable flipping, conditional inclusing of equations, etc.

1.10.1 Macro-processing language (2)

- The Dynare macro language provides a new set of **macro commands** that can be used in .mod files.
- Technically, this macro language is totally independent of the basic Dynare language, and is processed by a seperate component of the Dynare preprocessor.
- This macro processor transforms a .mod file with macros into a .mod file without macros (doing text expansion/inclusions) and then feeds it to the Dynare parser.
- The key point is to understand that the macro processor only does text substitution (like the C preprocessor or the PHP langage).



1.10.1 Macro-processing language (3)

- The flowchart below illustrates the relation between the Dynare macro language, macro processor, Dynare language and MATLAB/Octave.

pictures/sepera

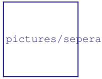
1.10.2 Introduction to the Syntax (1)

- The macro-processor is invoked by placing *macro directives* in the .mod file.
- Directives begin with: @#
- The main directives are:
  - `@#include`: file inclusion
  - `@#define`: definition of a macro processor variable
  - `@#if/@#ifdef/@#ifndef/@#ifndef/@#else/@#endif`: conditional statements
  - `@#for/@#endfor`: loop statements
- Most directives fit on one line. If needed however, two backslashes at the end of a line indicate that the directive is continued on the next line.

pictures/sepera

1.10.2 Introduction to the Syntax (2)

- The macro processor has its own list of variables which are different than model variables and MATLAB/Octave variables.
- There are 4 types of macro-variables:
  - integer
  - string (declared between *double* quotes)
  - integer array
  - string array
- Note that there is no boolean type:
  - false is represented by integer zero
  - true is any non-zero integer
- Further note that, as the macro-processor cannot handle non-integer real numbers, integer division results in the quotient with the fractional part truncated (hence, $5/3 = 3/3 = 1$).



1.10.2 Introduction to the Syntax (3)

- For a more comprehensive explanation of the syntax of the macro language, also containing illustrative examples, it is recommended to consider two following sources:
  - S. Villemot and H. Bastani: The Dynare Macro Processor - Dynare Summer School 2019.
  - Chapter 4.24 in Dynare: Reference Manual Version 4.

1.10.3 Common use (1)

- Add content.

pictures/sepera

Outline

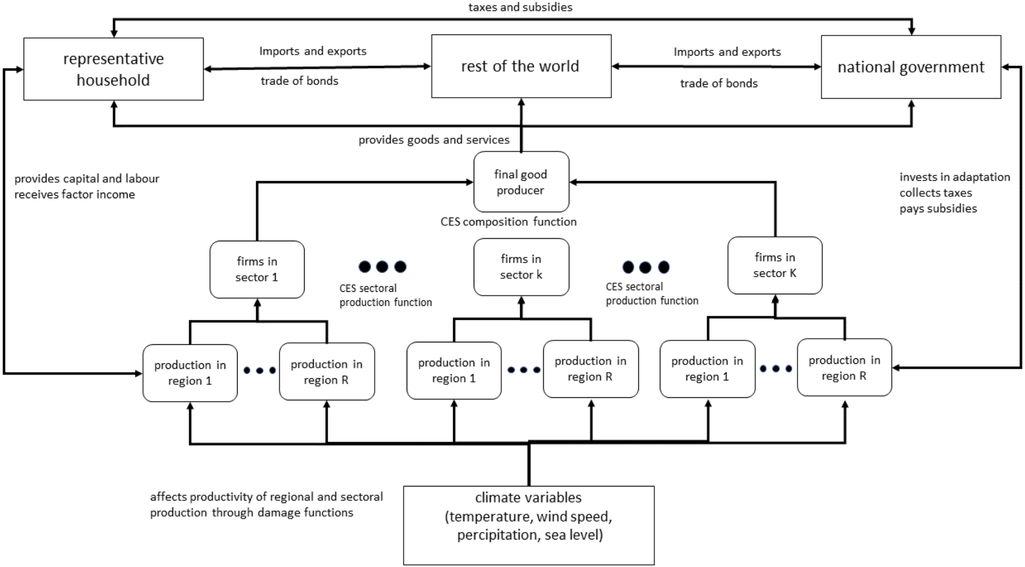
2. DGE-CRED Model
   - Introduction
   - Demand
   - Production

pictures/sepera

Introduction

- A dynamic general equilibrium model with optimizing agents
- We differentiate between regions and economic activities.
- Our model is implemented in the open source environment Dynare and can be run using Matlab or Octave.
  - Sectors in the model correspond to economic activities and the classification by the General Statistical Office (GSO).
  - Regions are based on the statistical regions.
- We extend the approach by Nordhaus 1993 to model the impact of climate change through damage functions.

Model Structure



Households

- representative households *h* providing labour *N* and capital *K* to domestic firms *f*
- maximize discounted utility over an infinite horizon by choosing consumption $C_t(h)$, capital $K_{k,r,t+1}(h)$, investments $I_{k,r,t}(h)$, labour $N_{k,r,t}(h)$ and foreign net wealth B_{t+1}
- the optimization problem of the representative household is

$$\max_{C_t(h),\, K_{k,r,t+1}(h),\, I_{k,r,t}(h),,\, N_{k,r,t}(h),\, B_{t+1}} \sum_{t=0}^{\infty} \beta^t \left(\frac{C_t(h)^{1-\sigma^C}}{1-\sigma^C} - \sum_{k=1}^{K} \sum_{r=1}^{R} A^N_{k,r,t}\, \phi^L_{k,r} \frac{N_{k,r,t}(h)^{1+\sigma^L}}{1+\sigma^L} \right)$$

$$\text{s.t.} P_t\, C_t(h)\,(1+\tau^C) + \sum_{k=1}^{K} \sum_{r=1}^{R} P_{k,r,t} I_{k,r,t}(h) + B_{t+1}(h) =$$

$$\sum_{k=1}^{K} \sum_{r=1}^{R} (1-\tau^N)\, W_{k,r,t} N_{k,r,t}(h) + \sum_{k=1}^{K} \sum_{r=1}^{R} P_{k,r,t}\, r_{k,r,t}\,(1-\tau^K)\, K_{k,r,t}(h) + S^f_t\, \phi^B_t\,(1+r^f_t)\, B_t(h)$$

Households Lagrangian

- We set-up the Lagrangian for the optimization problem to derive the first order conditions.

$$
\begin{aligned}
&\sum_{t=0}^{\infty} \beta^t \Bigg[\Bigg(\frac{C_t(h)^{1-\sigma^C}}{1-\sigma^C} - \sum_{k=1}^{K}\sum_{r=1}^{R} A^N_{k,r,t}\,\phi^L_{k,r}\,\frac{N_{k,r,t}(h)^{1+\sigma^L}}{1+\sigma^L} \Bigg) \\
&-\lambda_t(h)\Big(P_t\,C_t(h)\,(1+\tau^C) + \sum_{k=1}^{K}\sum_{r=1}^{R} P_{k,r,t} I_{k,r,t}(h) + B_{t+1}(h) - \sum_{k=1}^{K}\sum_{r=1}^{R}(1-\tau^N)\,W_{k,r,t} N_{k,r,t}(h) \\
&-\sum_{k=1}^{K}\sum_{r=1}^{R} P_{k,r,t}\,r_{k,r,t}\,(1-\tau^K)\,K_{k,r,t}(h) - S^f_t\,\phi^B_t\,(1+r^f_t)\,B_t(h)\Big) \\
&-\sum_{k=1}^{K}\sum_{r=1}^{R} \lambda_t(h)\omega^I_{k,r,t}(h)\left\{K_{k,r,t+1} - (1-\delta-D^K_{k,r,t})\,K_{k,r,t} - I_{k,r,t}\,\Gamma\left(\frac{I_{k,r,t}}{I_{k,r,t-1}}\right)\right\}\Bigg].
\end{aligned}
$$

pictures/sepera

Households First Order Conditions - Intratemporal

- Marginal utility of consumption

$$\lambda_t = \frac{C_t(h)^{-\sigma^C}}{P_t\,(1+\tau^C)}$$

- Labour supply curve

$$\phi^L_{k,r}\, A^N_{k,r,t}\, N_{k,r,t}(h)^{\sigma^L} = \lambda_t(h)\, W_{k,r,t}\,(1-\tau^N)$$

Households First Order Conditions - Intertemporal

- Euler equation for foreign bonds

$$\lambda_{t+1}\,\beta\,S^f_{t+1}\,\phi^B_{t+1}\left(1 + {r^f}_{t+1}\right) = \lambda_t$$

- Euler equation for capital

$$\lambda_{t+1}(h)\,\beta\,\left(P_{k,r,t+1}\,r_{k,r,t+1}\,(1-\tau^K) + (1-\delta-D^K_{k,r,t+1})\,\omega^I_{k,r,t+1}\right) = \lambda_t(h)\,\omega^I_{k,r,t}.$$

- Euler equation for investment

$$P_{k,r,t}\,\lambda_t(h) = \lambda_t(h)\,\omega^I_{k,r,t}\left(\Gamma(\frac{I_{k,r,t}}{I_{k,r,t-1}}) + \frac{\partial\Gamma(\frac{I_{k,r,t}}{I_{k,r,t-1}})}{\partial(\frac{I_{k,r,t}}{I_{k,r,t-1}})}\,\frac{I_{k,r,t}}{I_{k,r,t-1}}\right) - \beta\lambda_{t+1}(h)\,\omega^I_{k,r,t+1}\,\frac{\partial\Gamma(\frac{I_{k,r,t+1}}{I_{k,r,t}})}{\partial(\frac{I_{k,r,t+1}}{I_{k,r,t}})}\left(\frac{I_{k,r,t+1}}{I_{k,r,t}}\right)^2$$

- Investment adjustment cost

$$\Gamma(\frac{I_{k,r,t}}{I_{k,r,t-1}}) = 3 - exp\left\{\sqrt{\phi^K/2}\left(\frac{I_{k,r,t}}{I_{k,r,t-1}} - 1\right\}\right) - exp\left\{-\sqrt{\phi^K/2}\left(\frac{I_{k,r,t}}{I_{k,r,t-1}} - 1\right)\right\}$$

Households First Order Conditions - Intertemporal

- Euler equation for capital

$$\lambda_{t+1}(h)\,\beta\,\left(P_{k,r,t+1}\,r_{k,r,t+1}\,(1-\tau^{K})+(1-\delta-D^{K}_{k,r,t+1})\,\omega^{I}_{k,r,t+1}\right)=\lambda_{t}(h)\,\omega^{I}_{k,r,t}.$$

- Euler equation for investment

$$P_{k,r,t}\,\lambda_{t}(h)=\lambda_{t}(h)\,\omega^{I}_{k,r,t}\left(\Gamma(\frac{I_{k,r,t}}{I_{k,r,t-1}})+\frac{\partial\Gamma(\frac{I_{k,r,t}}{I_{k,r,t-1}})}{\partial(\frac{I_{k,r,t}}{I_{k,r,t-1}})}\,\frac{I_{k,r,t}}{I_{k,r,t-1}}\right)-\beta\lambda_{t+1}(h)\,\omega^{I}_{k,r,t+1}\,\frac{\partial\Gamma(\frac{I_{k,r,t+1}}{I_{k,r,t}})}{\partial(\frac{I_{k,r,t+1}}{I_{k,r,t}})}\,\left(\frac{I_{k,r,t+1}}{I_{k,r,t}}\right)^{2}$$

Rest of the world

- Euler equation foreign bonds

$$\lambda_{t+1}\,\beta\,S^f_{t+1}\,\phi^B_{t+1}\left(1 + {r^f}_{t+1}\right) = \lambda_t$$

- Effective exchange rate S^f and the world interest rate r^f.
- The required interest rate is above the world interest rate if the foreign debt ($B_{t+1} < 0$)/ foreign claims ($B_{t+1} > 0$) relative to GDP increases/decreases and future net exports relative to GDP will decrease.

$$\phi^B_{t+1} = exp\left(-\phi^B\,(S^f_{t+1}\,r^f_{t+1}\,\frac{B_{t+1}}{Y_{t+1}} + \frac{NX_{t+1}}{Y_{t+1}})\right)$$

pictures/sepera

Government Budget Constraint

- We are interested in different policy measures taken by the government to adapt to a new climate regime.
- Government behaviour is not a result of an optimization problem.

$$\begin{aligned} G_t + \sum_k^K \sum_r^R G^A_{k,r,t} + B^G_{t+1} = & \sum_k^K \sum_r^R \left\{ (\tau^K + \tau^K_{r,k,t})\, P_{k,r,t}\, r_{k,r,t}\, K_{k,r,t} + (\tau^N + \tau^N_{k,r,t})\, W_{k,r,t}\, N_{k,r,t}\, Pop_t \right\} \\ & + (1 + r^f_t)\, S^f_t \phi^B_t\, B^G_t \end{aligned}$$

pictures/sepera

Government Policy Instruments

- Governments can invest into adaptation capital stocks

$$K^{A,z}_{k,r,t+1} = \eta^{A,z}_{k,r,t}$$

- Evolution of adaptation capital stocks

$$K^{A,z}_{k,r,t+1} = \left(1 - \delta_{K^{A,z},k,r}\right) K^{A,z}_{k,r,t} + G^{A,z}_{k,r,t}$$

- Tax on capital expenditures paid by firms

$$\tau^{K}_{k,r,t} = \tau^{K}_{k,r,0} + \eta^{\tau^{K}}_{k,r,t}$$

- Tax rate on wage bill paid by firms

$$\tau^{N}_{k,r,t} = \tau^{N}_{k,r,0} + \eta^{\tau^{N}}_{k,r,t}$$

Resource constraint

- Households and government use domestic final goods Y_t produced by firms for consumption, investment and for exports X_t and can also use imports M_t for consumption and investment

$$Y_t = C_t + I_t + G_t + \underbrace{X_t - M_t}_{NX_t} \tag{1}$$

- The aggregation of the budget constraints of the representative households also states that positive net exports are used to increase net financial wealth to the rest of the world.

$$NX_t = B_{t+1} - (1 + r_t^f) S_t^f \phi_t^B B_t \tag{2}$$

Composite Production Function

- Households and government use domestic final goods Y_t produced by firms for consumption, investment and for exports X_t and can also use imports M_t for consumption and investment

$$Y_t = C_t + I_t + G_t + \underbrace{X_t - M_t}_{NX_t} \tag{3}$$

- The aggregation of the budget constraints of the representative households also states that positive net exports are used to increase net financial wealth to the rest of the world.

$$NX_t = B_{t+1} - (1 + r_t^f) S_t^f \phi_t^B B_t \tag{4}$$

pictures/sepera

Outline

3. Model Simulation and Calibration

pictures/sepera

References I

Nordhaus, William D (1993). “Optimal greenhouse-gas reductions and tax policy in the “DICE” model”. In: American Economic Review 83.2, pp. 313–317.