# GLOBAL MACROECONOMICS
## Solving Quantitative Macroeconomic Models

*MS.c. in International Economics*


*Valerio Dionisi*

*Department of Economics, Management and Statistics*
*University of Milano-Bicocca*


A.Y. 2023/2024

# The course

- lectures aim at teaching how to solve and simulate quantitative macroeconomic models designed to explain business cycles and economic fluctuations
  - these models highlight the role of forward looking expectations and uncertainty in macroeconomics;
  - all the models are firstly introduced in their theoretical framework, to then learn how to simulate them in `dynare`.

- if this is your first approach to coding, it may be an *highway to hell*:
  - *it's a long way* ... you may do not know how to begin, how to proceed along, and finally how to conclude;
  - each line may be a *shot in the dark*: codes per se are not difficult, but you may get lost in ordering and give a choral sense.

➪ I will try to guide you in interpreting and shaping the `dynare` codes
  - first with a theoretical introduction behind what `dynare` does;
  - and then jumping to the definition and the ordering of the different commands;
  - so that, for you, *hell ain't a bad place to be* ... but now *hells bells*!

# Schedule

- the ***plan*** of the lectures will be as follows:
  1. *recursive methods* and introduction to `dynare`;
  2. *deterministic simulation*: solution of the Solow (1956) model, and simulation of the Ramsey-Cass-Koopmans (RCK) economy;
  3. *Real Business Cycle* (RBC) models, and introduction to *log*-linearization;
  4. baseline *New Keynesian* (NK) models;
  5. optimal monetary policy (*discretion* vs. *commitment*) in NK models;
  6. topics in *Bayesian estimation*.

⚠ codes to each topic will be provided during lectures and indicated in the slides

- in class I will show even harder codes for your level. If some are interested, these will be provided only under request

# Overview

*"solving a standard quantitative macroecnomic model is just a big rootfinding problem"*

- finding the *root* of a function means to find $x$ such that $f(x) = 0$

$\longleftrightarrow$ a typical problem is $f(x) = g(x)$, so that the *root* is simply $f(x) - g(x) = h(x)$

- economic problems are:
  1. ***univariate***, solution to a fixed-point equation;
  2. ***multivariate***, solution to a system of equilibrium conditions.

- roadmap of *rootfinding methods* (not covered in this course)
  - univariate case $\rightarrow$ BISECTION, shrink the interval until finding the *root* (`fzero`);
  - for the multivariate case:
    - ★ NEWTON'S METHOD, iterate $n > 1$ times to find the zero of the Taylor-approximated system until $|\, f(\mathbf{x}_n) < \epsilon\, |$ (command `fsolve`);
    - ★ FUNCTION ITERATION, transform rootfinding into a *fixed-point* problem, $\mathbf{x} = \mathbf{x} + f(\mathbf{x})$
  - COMPLEMENTARITY PROBLEM, when $max\ \ f(x)\ \ s.t.\ \ \{x > a\, ;\, x < b\}$

☆ *important*: the outcome of an optimization problem is a rootfinding problem!

# Outline

# Table of Contents

# Introduction

- in macroeconomics, a quintessential model is just a set of nonlinear equations

$$E_t\Big[f\left(x_t, x_{t+1}, u_t, u_{t+1}\right)\Big] = 0, \forall t$$

with $x_t = \begin{pmatrix} x_t^c \\ x_t^s \end{pmatrix}$ and

  - $x_t^c$: $m \times 1$ vector of endogenous (*predetermined*) control variables;
  - $x_t^s$: $n \times 1$ vector of endogenous (*forward*, *jump*) state variables;
  - $u_t$: $k \times 1$ vector of exogenous state variables.

☆ ***Goal***: find the path $\{x_t\}_{t=0}^{\infty}$ satisfying model equations given initial conditions $x_{t=0-1}^s$ and exogenous shocks $\{u_t\}_{t=0}^{\infty}$

# Recursive methods

- main curse of modern quantitative macroeconomics models: *high dimensionality*
  - many variables over many periods and many *histories* of exogenous variables;
  - cannot solve infinitely many unknowns . . . use *recursion* idea.

➪ ***Recursive methods*** $\longleftrightarrow$ few variables are sufficient to summarize past history

$$x_t = g\left(x_{t-1}^s, u_t\right)$$

  - solving a model means finding $g(\cdot)$, called the *policy function*

- high dimensionality of $g(\cdot)$ requires some form of *approximation*
  - ***local methods***: approximate equilibrium objects locally around some focus point
    - ★ *linearization*, *LQ methods*, ***perturbation*** *methods*
  - ***global methods***: approximate global properties of equilibrium objects
    - ★ *discretization* (grids), ***projection*** *methods* (splines, integral approximation (Chebyshev, . . . ) )

# What is *perturbation*?

- *key idea*
  - rewrite the model in terms of a *perturbation parameter*, $\sigma \geq 0$

$$E_t\left[f\left(x_t, x_{t+1}, u_t, u_{t+1}\,;\,\sigma\right)\right] = 0, \forall t$$

  - Taylor expansion in terms of the variables and the perturbation parameter
  - solve the approximated model using a dedicated approach

- thus, the model solution is as a function of $\sigma$

$$x_t = g\left(x^s_{t-1}, u_t\,;\,\sigma\right)$$

- usually, the perturbation parameter arises from

$$u_{t+1} = \mathcal{B}\left(u_t\right) + \sigma \underset{k\times k}{\mho} e_t$$

where $k \times k$ is the dimension of the matrix if $(x_t, u_t)$ are vectors of variables

# Steps

- *Step 0*: find the system of equations characterizing the model, *i.e.*,

$$E_t\left[f\left(x_t, x_{t+1}, u_t, u_{t+1}\,;\,\sigma\right)\right] = 0,\ \forall t$$

- *Step 1*: solve the model with no uncertainty[1] $(\sigma = 0)$, *i.e.*, the steady-state system

$$f\left(x, x, u, u\right) = 0$$

- *Step 2*: find an approximation of $g(\cdot)$ around the above steady state
  - *i.e.*, approximate the policy functions

$$g\left(x^s_{t-1}, u_t\,;\,\sigma\right)$$

    around

$$g\left(x^s, u\,;\,\sigma\right)$$

  - how? $j^{th}$-order Taylor series expansion, or *pruning*[2]

- *Step 3*: resolution $\Big\{$Blanchard and Kahn (1980), Uhlig (1999), Sims (2002), ...$\Big\}$

[1] This can typically be done with *rootfinding* (*e.g.*, Newton method). Be aware of multiple steady states (select only the best).
[2] Used if higher-order perturbations are worse than linear, *i.e.*, when $j > 1$ leads to different shape than true policy function.

# Table of Contents

# What is `dynare`?

- `dynare` is a collection of `MatLab` functions
  - to solve and simulate nonlinear models with forward-looking variables under rational expectations
- freely available on `www.dynare.org`
  - online you can also find reference manuals, some tutorials and an active forum
- what can `dynare` do?
  - solve deterministic models nonlinearly;
  - solve stochastic models (under local approximations);
  - solve for policy functions;
  - estimate DSGE models: maximum likelihood, Bayesian methods.
- *note*: `dynare` can solve only *equality* constraints
  - with *inequality* constraints, *rootfinding* methods for *complementarity problems*

# How does `dynare` solve for the *policy rules*?

- `dynare` uses perturbation methods
  - there are other numerical methods, such as *value* or *policy function iterations*
- how does *perturbation* work?
  - start from a steady state;
  - take a Taylor expansion of the policy rules around the steady state;
  - find the coefficients of the Taylor expansion sequentially.
    - *e.g.*, for a second-order expansion, solve for the linear terms first and the quadratic then
- *caveats*:
  - perturbation requires differentiability;
  - it's an approximation around a steady state.

# `dynare`: Basics

- main unit in `dynare` is the `<name>.mod` file. In `MatLab`:
  - execute `dynare` with the command `dynare <name>.mod`
  - modify `dynare` with the command `edit <name>.mod`

$\rightarrow$ it is the file in which you write down your stochastic model

- the `<name>.mod` file consists of different blocks
  - *variables and parameters* block
  - *model* block
  - *steady state* block
  - *shocks* block
  - *solution* (or *estimation*) block

# Preliminary blocks

## Variables and Parameters block

- `var` *list endogenous variable names*;
- `varexo` *list exogenous variables (shocks)*;
- `parameters` *list parameter names*;
  - then, *list parameter values*.

- parameter values can also be loaded from an external file, using the syntax `load <filename;>.mat` and `set_param_value ('parametername', parametername)`

## Model block

- starts with `model;`
- ends with `end;`
- in between, type equations ending with ";"
  - `x(-1)` for predetermined variables
  - `x(+1)` for expectations
  - you can type an equation over several lines if you do not end it with ";"

- if the model is (*log*-)linear, type `model(linear);`

# Steady state block



- `dynare` linearizes around the deterministic steady state
  - this steady state needs to be calculated
- two options:
  - let `dynare` calculate the steady state numerically . . .
  - . . . or calculate the steady state with paper and pencil and tell `dynare` what it is.
- calculating the steady state is a nonlinear problem
  - it might be difficult for the computer

## Steady state block 2/2

- *Option 1*: numeric solution

### Steady State block, option 1

- start with `initval;`
- ends with `end;`
- in between, add initial values (even previously defined) for all variables.

  - if a variable does not appear, `dynare` assumes it is zero in steady state;
  - if the model is elaborate, take initial values from a simpler model[3].

- *Option 2*: paper and pencil for analytical solutions

### Steady State block, option 2

- start with `steady_state_model;`
- ends with `end;`
- in between, add equations where variables are function of parameters only.

  - or separate `<name>_steadystate.m` file with the same name as the `<name>.mod` file
    - inside, provide `dynare` with the steady state values (based on parameters)

[3] This concept is closely related to a more elaborate way on finding the steady state called *homotopy*.

# Simulation blocks

## Shocks block

- starts with `shocks;`
- ends with `end;`
- in between, declare shock standard deviations (as numerical values or by referring to a parameter name)

`var` *shock name*; `stderr` *standard deviation value*;

- note how the shock's value is expressed as being a *standard deviation* of the given innovation in that particular variable

## Solution (or Estimation) block

- at the bottom of the `.mod` file add

```
stoch_simul(order=1,IRF=20);
```

- inside the brackets, many more options can be specified (mostly related to which output you want `dynare` to report);
- after the brackets you can list individual variables, if you want output for these variables only;
- otherwise, `dynare` will do it for all variables;
- `stoch_simul()` holds only for stochastic models. See below for deterministic models.

# Time convention

*pay attention to the timing of the variables!*

- in `dynare`, the timing of each variable reflects when that variable is decided
  - *example*: in most models, capital stock is subject to a law of motion. This implies that such capital is not decided at $t$, but rather at $t-1$
    - ★ in jargon, it is a *predetermined* variable;
    - ★ that is why in `dynare`, the capital stock in the production function is written at $t-1$.

- time indices are given in parenthesis: $x_{t+1} \rightarrow$ `x(+1)`, $x_t \rightarrow$ `x`, $x_{t-1} \rightarrow$ `x(-1)`

⚠ why is it important to struggle with time convention?

  - you should know that Blanchard and Khan (1980) conditions are met only if *the number of non-predetermined variables equals the number of eigenvalues greater than one* . . .
    - ★ . . . and if this condition is not met, you will get a warning!

- note that if a variable is predetemined, it is not true that this should always be written at $t-1$ throughout the code
  1. consumption may be forward-looking in the Euler, but may also have lag(s) if habit formation is featured in the model;
  2. in a resource constraint, capital on the left-hand-side is at $t$, but that on the right-hand-side is at $t-1$ since capital belongs both from the law of motion and the production function.

# Running the `.mod` file

- in `MatLab`, make sure you are in the right directory, then just type:

  `dynare <name>.mod`

  to run the simulation, or type `edit <name>.mod` to modify it

- depending on the settings inside the `stoch_simul` command, `dynare` generates:
  - policy rules;
  - implied moments (means, standard deviations, correlations);
  - impulse response functions, ...

- `dynare` writes these results to the screen, but also stores them in additional files
  - `oo_.dr.ys` is a $(m+n) \times 1$ vector of steady state values;
  - `oo_.dr.ghx` is a $(m+n) \times n$ matrix of $x_t$ coefficients;
  - `oo_.dr.ghu` is a $(m+n) \times k$ matrix of $u_t$ coefficients;
  - `oo_.irfs` reports IRFs for each variable.

- note:
  - `dynare` reports policy functions in deviations from steady state values;
  - the constant gives the steady state value of a given variable (column).

# Other resources

- on the `dynare` website https://www.dynare.org/resources/ you can get . . .
  - forum;
  - quick start/tutorial;
  - manual;
  - or `dynare` implementations of published models at:
    - https://www.macromodelbase.com/, directly in its `download` section;
    - https://github.com/johannespfeifer/dsge_mod;
    - https://forum.dynare.org/t/practicing-dynare-2019-updated-version/13389.

. . . and hundreds of replication codes available; just learn by using it!

☞ ***concluding remarks***. Simulating in `dynare` is:

- easy and flexible, but be careful with its idiosyncratic behaviour!
- great for prototyping (*local* solutions), but may need more accurate (*global*) solutions.

# Table of Contents

# Characteristics of deterministic models

- ***deterministic simulation***: the occurrence of all future shocks is known exactly at the time of computing the model's solution
- characteristics:
  1. full information, perfect foresight and no uncertainty around shocks;
  2. shocks can hit the economy today or at any time in the future, and these are perfectly expected;
  3. the solution does not require linearization $\rightarrow$ find the exact paths of endogenous variables.
- this solution method can be useful when the economy is far away from steady state (*i.e.*, when linearization offers a poor approximation)

# Deterministic models in `dynare`

- the system is supposed to be in a state of equilibrium before a *shock*-period
  - *i.e.*, before the news of a contemporaneous (or future) shock is learned by the agents
- the purpose of the simulation is to describe the reaction in reaction to (or in anticipation of) the shock ...

  ... until the system returns to the old state of equilibrium, or eventually converges to a new one
  - in most models, this return to equilibrium is only an asymptotic phenomenon, which one must approximate by an horizon of simulation far enough in the future
- another exercise for which `dynare` is well suited is to study the *transition path* to a new equilibrium following a permanent shock

# The Solow model

- the classical Solow (1956) model in unit of effective labor is made as follows

$$y_t = A_t\, k_t^{\alpha}$$

$$k_{t+1} = i_t + (1 - g - n - \delta)\, k_t$$

$$y_t = c_t + i_t$$

$$c_t = (1 - s) y_t$$

with $g, n > 0$ known, and parameters $\{\alpha, \delta, s\} \in (0, 1)$ taken as given

- how to simulate this model in `dynare`?
  - simulate both *transitory* and *permanent changes* in TFP $(A_t)$
  - *file*: `Solow_deterministic.mod`

... but firstly it is necessary to have the *steady-state version* of the model

- a situation in which $\bar{x} \equiv \Delta x_t = 0$

# Steady-state system of equations

① drop time notation to the system of equilibrium conditions …

$$\bar{y} = \overline{A}\,\bar{k}^{\alpha} \quad with \quad \overline{A} = 1$$
$$\bar{k} = \bar{i} + (1 - g - n - \delta)\,\bar{k}$$
$$\bar{y} = \bar{c} + \bar{i}$$
$$\bar{c} = (1 - s)\bar{y}$$

② … and solve the resulting equations such that the endogenous variables are functions of parameters only. Given $\overline{A} = 1$:

$$\bar{y} = \overline{A}\left(\frac{\overline{A}s}{(\delta + g + n)}\right)^{\frac{\alpha}{(1-\alpha)}}$$

$$\bar{k} = \left(\frac{\overline{A}s}{(\delta + g + n)}\right)^{\frac{1}{(1-\alpha)}}$$

$$\bar{i} = s\overline{A}\left(\frac{\overline{A}s}{(\delta + g + n)}\right)^{\frac{\alpha}{(1-\alpha)}}$$

$$\bar{c} = (1 - s)\overline{A}\left(\frac{\overline{A}s}{(\delta + g + n)}\right)^{\frac{\alpha}{(1-\alpha)}}$$

# *Step 1*: Variables and parameters block

- specify variables (endogenous `var` and exogenous `varexo`) and `parameters`

```
%% variables and parameters
var c i y k exp_c exp_i exp_y exp_k;                        % endogenous
varexo A;                                                   % exogenous
parameters alpha delta s g n css iss yss kss Ass gss nss; % parameters

alpha = 0.33;  % capital share
delta = 0.025; % capital depreciation
s     = 0.2;   % savings rate
g     = 0.1;   % technology growth rate
n     = 0.2;   % population growth rate

Ass = 1;       % technology, steady state

css = (1 - s)*Ass*((Ass*s)/(delta + gss + nss))^(alpha/(1 - alpha));
iss = s*Ass*(Ass*s/(delta + gss + nss))^(alpha/(1 - alpha));
yss = Ass*(Ass*s/(delta + gss + nss))^(alpha/(1 - alpha));
kss = (Ass*s/(delta + gss + nss))^(1/(1 - alpha));
```

  - *note*: steady state values can be determined as parameters (as above), or directly in the *steady state block* (as in the RBC section)

## *Step 2*: Model block

- specify the equilibrium conditions

```
%% model equations
model;

y = A*k(-1)^(alpha);                 % production function. If the
                                     % capital law of motion is at
                                     % time t, then set k(-1) here

k = i + (1 - g - n - delta)*k(-1);  % law of motion for capital

y = c + i;                           % market clearing condition

c = (1 - s)*y;                       % consumption function

% exp values (to capture percentage changes)
exp_y = exp(y);                      % exp output
exp_c = exp(c);                      % exp consumption
exp_i = exp(i);                      % exp investment
exp_k = exp(k(+1));                  % exp capital

end;
```

  - *note*: `exp`-values are only for simulation purposes (compute impulse response function as percent deviation from steady-state); can avoid to use it (remember to remove their declaration in `var`)

# *Step 3*: Steady state block

- provide initial values for the steady state (model system at the steady state)

```
%% steady state conditions (with values defined earlier)
initval;

A = Ass;          % technology, initial value
c = css;          % consumption, initial value
i = iss;          % investment, initial value
y = yss;          % output, initial value
k = kss;          % capital, initial value

% exp values (to capture percentage changes)
exp_y = exp(y);   % exp output
exp_c = exp(c);   % exp consumption
exp_i = exp(i);   % exp investment
exp_k = exp(k);   % exp predetermined capital

end;

steady;
check;
```

  - command `steady` solves for the steady state values of the model, and forces all initial values to the (exact) steady states
  - command `check` controls the consistency of the specified steady state values with that implied by the model

# *Step 4*: Shock block

- the command `shocks` defines the type of shock to be simulated

```
%% transitory shock
shocks;
var A;
periods 1:4;  % one year shock (remember: quarterly data!)
values 1.01;  % 0.01 shock in A. Remember that initial value is A = 1
end;

perfect_foresight_setup(periods = 25);
perfect_foresight_solver;

%% permanent shock
initval;  % initial value declaration
A = 1;
end;
steady;   % steady states at the initial value

endval;   % final value declaration (permanent value of the shock)
A = 1.01;
end;
steady;   % steady states at the final value

perfect_foresight_setup(periods = 25);
perfect_foresight_solver;
```

  - the command `perfect_foresight_setup(periods=25)` computes:
    - deterministic impulse response functions (IRFs, 25 periods) under perfect foresight, and computation occurs through the `perfect_foresight_solver` command.
  - under perfect foresight, the permanent shock requires initial and final values

# Shock block in *plain* english for deterministic models

- ***temporary shock***
  - under a temporary shock, the model eventually comes back to its steady state;
  - codes:
    - ★ must start with `shocks;` and must finish with `end;`
    - ★ in between, set the duration (command `periods`) and level (command `values`) of the shock specified in `var`
  - `dynare` would directly compute the new steady state
    - ★ possible to enter future periods (*e.g.*, `periods 5:10`), to study the anticipatory behavior in response to future shocks

- ***permanent shock***
  - not specify `shocks` but simply define the initial and final (steady state) values of the variable of interest ...

    ... and let `dynare` compute the transition path between the two specified values.
  - define two different steady states:
    - ★ the first must start with `initval;` and must finish with `end;`, and has to report the initial value of the shock;
    - ★ the second must start with `endval;` and must finish with `end;`, and has to report the final value of the shock.

# More versions

- in addition to the above codes, permanent change to TFP occurring in period 5, and permanent changes to $g$ and $n$ $\longleftrightarrow$ *file*: `Solow_deterministic.mod`

- deterministic Solow model with shocks to growth rates of capital given permanent shocks to $g$ and $n$ $\longleftrightarrow$ *file*: `Solow_growthrates.mod`

# The Ramsey-Cass-Koopmans model 1/2

- time is *discrete* (to be consistent with the timing set-up structure of `dynare`)

- ***production side***:
  - perfectly competitive Cobb-Douglas production function

$$Y_t = K_t^{\alpha} (A_t L_t)^{1-\alpha}$$

  - one can define that
    - labour hours (the same as population) increases exogenously at a constant rate, $\Delta L = n$;
    - labor productivity grows at the exogenous rate $\Delta A = g$.
  - aggregate capital accumulates according to $\Delta K = \underbrace{Y_t - C_t}_{I_t} + \delta K_t$

- in *intensive form*:
  - production function, $y_t = k_t^{\alpha}$;
  - capital law of motion,

$$\frac{\Delta K}{A_t L_t} = y_t - c_t - \delta k_t \rightarrow k_{t+1} + (g+n)\, k_t = y_t - c_t - \delta k_t \Rightarrow k_{t+1} = y_t - c_t - (g+n+\delta)\, k_t$$

- a *steady-state* occurs when $\Delta k = 0$. When is it the case?

# The Ramsey-Cass-Koopmans model 2/2

- the best possible steady-state is the one maximizing $c_t$ $\rightarrow$ *golden rule* (Phelps, 1961)

- social planner problem:
  - the optimization problem is

$$\max_{c_t} \sum_{t=0}^{\infty} \beta^t u(c_t)$$
$$s.t. \quad k_{t+1} = y_t - c_t - (g + n + \delta) k_t$$

whose solution is the well-known (deterministic) *Euler equation*

$$\frac{u'(c_{t+1})}{u'(c_t)} = \beta \left[ f'(k_{t+1}) + (1 - \delta) \right]$$

- equilibrium of the economy is simply

$$k_{t+1} = y_t - c_t - (g + n + \delta) k_t$$
$$\frac{u'(c_{t+1})}{u'(c_t)} = \beta \left[ f'(k_{t+1}) + (1 - \delta) \right]$$

- the behavior of the economy is analyzed by the mean of a *phase diagram*

# Phase diagram



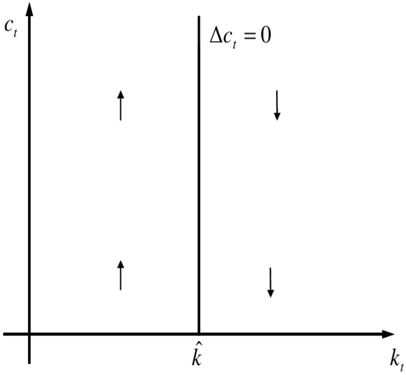



Figure 3.1: Consumption

when $\Delta c = 0$ the *Euler* becomes $1 = \beta \left[ f'\left(\widehat{k}\right) + (1-\delta) \right]$,
with $\widehat{k}$ being the value of $k_{t+1}$ that sustains $\Delta c = 0$

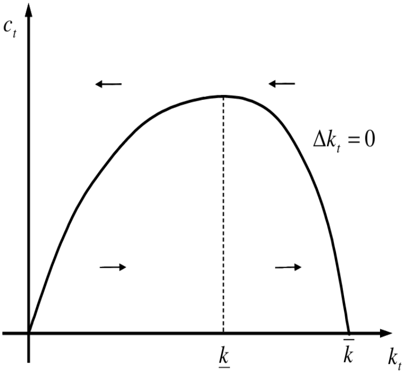



Figure 3.2: Capital

when $\Delta k = 0$ capital equation turns to be $\widehat{c} = f(k_t) - \delta k$,
with $\widehat{c}$ being the value of $c_t$ that sustains $\Delta k = 0$

# Phase diagram interpretation



- ***consumption***, with $\widehat{k}$ being the value of $k$ such that $\Delta c = 0$
  - $k_{t+1} > \widehat{k}$, then $c_t$ decreases over time $\rightarrow$ the *marginal product of capital* (MPK) is lower, and thus lower is the interest rate, $r = f'(k)$. Hence, consumers will not save thus consuming more today;
  - $k_{t+1} < \widehat{k}$, then $c_t$ increases over time $\rightarrow$ the MPK is higher, and thus higher is the interest rate. Consumers will save to consume more tomorrow, therefore consuming less today.
- ***capital***, with $\widehat{c}$ being the value of $k$ such that $\Delta k = 0$
  - $c_t > \widehat{c}$, then $k_t$ decreases over time $\rightarrow$ too much consumption, so that the existing capital is not as much to fulfill consumption;
  - $c_t < \widehat{c}$, then $k_t$ increases over time $\rightarrow$ consumers will save more to consume more, and so capital increases over time.

# Phase diagram



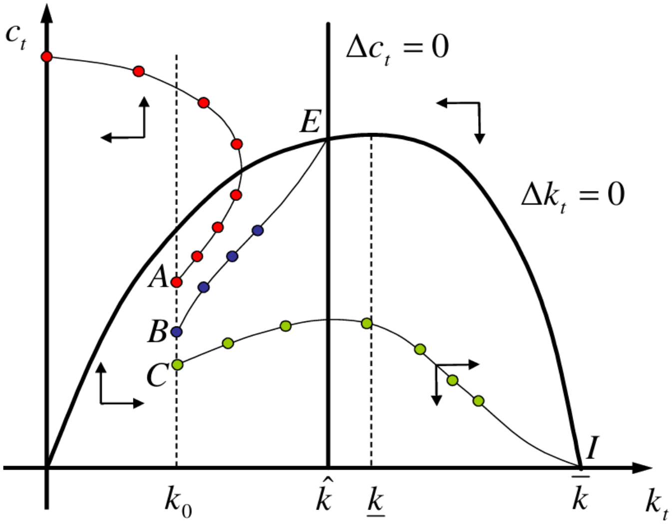



Figure 3.3: Balanced growth path (BGP)

# Phase diagram interpretation *ii/ii*

- ***balanced growth path*** (BGP), given a situation in which $k_0 < \widehat{k}$
  - point *A*. If the economy starts at a capital level lower than $\widehat{k}$, say $k_0$, but with a consumption level higher than $c^*$, say $c_A$, then the economy will never converge to the steady state since capital is not accumulated over time. *Euler* not fulfilled;
  - point *C*. If the economy starts at a capital level lower than $\widehat{k}$, say $k_0$, and with a consumption level lower than $c^*$ say $c_B$, then the economy will never converge to the steady state since the accumulated capital is too much that it becomes unproductive. *Transversality condition* not satisfied;
  - point *B*. There exists only one path, $\left\{c^* = c_B, k_0\right\}$ such that the economy converges to its steady state. This roadway is called the *saddle path*, and it is represented as the ride going from $B$ to $E$.
- the same applies when $k_0 > \widehat{k}$.

# The Ramsey-Cass-Koopmans model in `dynare`

- in aggregate (not intensive) form, the above model featuress:
  - production sector as in Solow (1956) model

$$Y_t = K_t^{\alpha} \left(A_t L_t\right)^{1-\alpha}$$

$$K_{t+1} = I_t + \left(1-\delta\right) K_t$$

$$Y_t = C_t + I_t$$

  - household *log*-utility maximizing behaviour

$$\frac{1}{C_t} = \beta \left[ f_Y' \left(K\right) + \left(1-\delta\right) \right] \frac{1}{C_{t+1}}$$

- shocks in
  - labour population, $L$;
  - labour productivity, $A$;
  - ... to analyze the transition over the balanced growth path (BGP).
- *file*: `RamseyCassKoopmans_deterministic.mod`

# Table of Contents

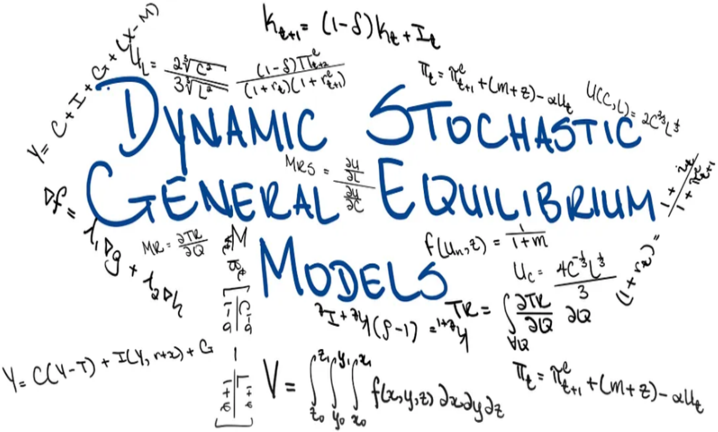
# Characteristics of stochastic models

- ***stochastic simulation***: only the distribution of future shocks is known
- characteristics:
  1. full information, perfect foresight and uncertainty around shocks;
  2. shocks can hit the economy today or at any time in the future with surprise, but thereafter their expected value is zero;
  3. the solution may require linearization (*i.e.*, use of Taylor approximations around a steady state) $\rightarrow$ *certainty equivalence* property.[4]
- expected future shocks, or permanent changes in the exogenous variables cannot be handled due to the use of Taylor approximations around a steady state

basically, the difference between *deterministic* and a *stochastic* model relies on the ***nature*** of the shock

- in the first, clear cause-and-effect of the shock $\rightarrow$ if ran several times, identical results;
- in the second, uncertainty around the shock $\rightarrow$ random evolution if ran several times;
- hence, *a deterministic model is a special kind of stochastic model with zero randomness*.

[4] When these models are linearized to the first order, agents behave as if future shocks where equal to zero since $E[u] = 0$.


Dynamic Stochastic
General Equilibrium
Models

# Real Business Cycle (RBC) models

- up to the 1970s, dominant *Keynesian* approach: postulating theorems without giving them rigorous theoretical explanations (wrong?)

- *Lucas's critique* (Lucas 1976, 1987): macroeconomists should build models where the agents' behaviour is invariant with respect to policy. Methodological revolution:

  ① *research agenda*: business cycle fluctuations as the central object;

  ② *theory and method*: macroeconomics should be built on (choice-theoretical) microfoundations;

  ③ *disequilibrium to equilibrium*: dynamic optimization and market clearing conditions.

  ☎ the *Rational Expectations* (RE) *Revolution*[5] of the 1970s as the logical outcome

    → agents can make "mistakes", but cannot make *systematic* mistakes

- Kydland and Prescott (1982)[6] $\longrightarrow$ combination of *growth* and *cycle*
  - integrated design to study economic (*business-cycle*) fluctuations in a growing economy;
  - it depicts an economy without money, thus belonging to the class of RBC models.

[5] Scholars as Robert Lucas, Thomas J. Sargent, and Neil Wallace, based on the previous analysis of Muth (1960).

[6] Awarded with the Nobel Price in economics 2004 for their contribution to the theory of business cycles and economic policy.

# Baseline RBC model: *Households*

- preferences

$$U = E_0 \sum_{t=1}^{\infty} \left(\frac{1}{1+\rho}\right)^{t-1} \left(log\, C_t - \frac{L_t^{1+\gamma}}{1+\gamma}\right)$$

- optimality conditions
  - Euler equation

$$\frac{1}{C_t} = \left(\frac{1}{1+\rho}\right) E_t \left[\frac{1 + r_{t+1} - \delta}{C_{t+1}}\right]$$

  - labour supply

$$L_t^{\gamma} = \frac{w_t}{C_t}$$

# Baseline RBC model: *Technology*

- Cobb-Douglas production function

$$Y_t = A_t\, K_t^{\alpha} \left[ (1+g)^t\, L_t \right]^{1-\alpha}$$

where labour-augmenting productivity grows at $g$

- $A_t$ is a technological shock that follows a *stationary* AR(1) process

$$log\, A_t = \rho_A\, log\, A_{t-1} + e_t \tag{1}$$

where:

  - $e_t$ is *iid* normally distributed with mean zero and standard deviation $\sigma$;
  - $\rho_A \in (0, 1)$ captures persistence.

- profit maximization

$$\begin{aligned} r_t &= \alpha\, A_t\, K_t^{\alpha-1} \left[ (1+g)^t\, L_t \right]^{1-\alpha} \\ w_t &= (1-\alpha)\, A_t\, K_t^{\alpha} \left[ (1+g)^t \right]^{1-\alpha} L_t^{-\alpha} \end{aligned}$$

# Baseline RBC model: *Equilibrium*

- the previous model is *non-stationary*
  - converges to a steady state in the normalized (*de-trended*) variables:

$$\widehat{w}_t = \frac{w_t}{(1+g)^t}; \quad \widehat{C}_t = \frac{C_t}{(1+g)^t}; \quad \widehat{K}_t = \frac{K_t}{(1+g)^t}$$

- equilibrium conditions
  - F.O.C.s of the (*stationary*) model:

$$\frac{1}{\widehat{C}_t} = \left(\frac{1}{1+\rho}\right) E_t \left[\frac{1 + r_{t+1} - \delta}{(1+g)\widehat{C}_{t+1}}\right] \tag{2}$$

$$L_t^{\gamma} = \frac{\widehat{w}_t}{\widehat{C}_t} \tag{3}$$

$$r_t = \alpha A_t \left(\frac{\widehat{K}_t}{1+g}\right)^{\alpha-1} L_t^{1-\alpha} \tag{4}$$

$$\widehat{w}_t = (1-\alpha) A_t \left(\frac{\widehat{K}_t}{1+g}\right)^{\alpha} L_t^{-\alpha} \tag{5}$$

  - resource constraint:

$$\widehat{K}_t + \widehat{C}_t = \frac{\widehat{K}_t}{(1+g)}(1+\delta) + A_t \left(\frac{\widehat{K}_t}{1+g}\right)^{\alpha} L_t^{1-\alpha} \tag{6}$$

# Baseline RBC model: *Steady state*

- eqs. (1)-(6) make up the non-linear (*de-trended*) equilibrium system

  $\rightarrow$ 6 equations in 6 *endogenous* variables $\left(\widehat{C}_t, A_t, \widehat{K}_t, \widehat{L}_t, \widehat{w}_t, r_t\right)$

- to simulate such model in `dynare`, need to specify[7] *steady state conditions*

$$
\begin{aligned}
A &= 1 \\
r &= (1+g)(1+\rho)+\delta-1 \\
L &= \left(\frac{1-\alpha}{\frac{r}{\alpha}-\delta-g}\right)\frac{r}{\alpha} \\
K &= (1+g)\left(\frac{r}{\alpha}\right)^{\frac{1}{\alpha-1}} L \\
C &= (1-\delta)\frac{K}{1+g}+\left(\frac{K}{1+g}\right)^{\alpha} L^{1-\alpha}-K \\
w &= C;
\end{aligned}
$$

➪ equations should display each variable as being ***constant*** (*i.e.*, as a function of only parameters and/or other steady-state variables)

[7] *Hint*: write a one-period non-liner system (*i.e.*, drop time subscript) and solve it.

# *Step 1*: Variables and parameters block

*file*: `RBC_baseline.mod`

- specify variables (endogenous `var` and exogenous `varexo`) and parameters

```
%% variables and parameters
var C K L w r A;                              % endogenous
varexo e;                                     % exogenous
parameters alpha delta gamma rho rho_A g;     % parameters

alpha = 0.33;  % capital share
delta = 0.1;   % depreciation rate
gamma = 0;     % inverse of Frisch elasticity
rho   = 0.03;  % discount rate
rho_A = 0.97;  % AR(1) coefficient
g     = 0.015; % labor-augmenting productivity growth rate
```

  - *note*: annual data

# *Step 2*: Model block

- specify the equilibrium conditions

```
%% model equations
model;

1/C = 1/(1+rho)*(1/(C(+1)*(1+g)))*(r(+1)+1-delta); % Euler

L^gamma = w/C;                                     % labour supply

r = alpha*A*(K(-1)/(1+g))^(alpha-1)*L^(1-alpha);   % rental rate

w = (1-alpha)*A*(K(-1)/(1+g))^alpha*L^(-alpha);    % wage rate

K+C = (K(-1)/(1+g))*(1-delta)+A*(K(-1)/
       (1+g))^alpha*L^(1-alpha);                   % market clearing

log(A) = rho_A*log(A(-1))+e;                       % technology

end;
```

  - why does it appear $K$ in two periods in the resource constraint? see timing slide

# *Step 3*: Steady state block

- provide analytical solutions for the steady state ($\gamma = 0$ imposed to simplify)

```
%% steady state conditions
steady_state_model;

A = 1;                                                  % technology
r = (1+g)*(1+rho)+delta-1;                              % rental rate
L = ((1-alpha)/(r/alpha-delta-g))*r/alpha;              % labour
K = (1+g)*(r/alpha)^(1/(alpha-1))*L;                    % capital
C = (1-delta)*K/(1+g)+(K/(1+g))^alpha*L^(1-alpha)-K; % resource
    constraint
w = C;                                                  % wage rate

end;

steady;
check;
```

  - command `steady` solves for the steady state values of the model
  - command `check` controls the consistency of the specified steady state values with that implied by the model

# *Step 4*: Shock block

- the command `shocks` defines the type of shock to be simulated

```
%% shock
shocks;
var e; stderr 0.01; % technology innovation in AR(1)
end;

check;

stoch_simul(order=1,irf=100)
```

  - the command `stoch_simul(order=1,irf=100)` computes:
    - a first order Taylor expansion around the steady state (the default order is 2);
    - impulse response functions (IRFs, 100 periods);
    - various descriptive statistics (moments, variance decomposition, correlation and autocorrelation coefficients).

# Permanent shock in *plain* english for deterministic vs. stochastic models

- ***deterministic model***
  - agents in the economy perfectly know when a shock would occur
    - no uncertainty and zero randomness around shocks
  - codes:
    - define initial (`initval;`) value of the shock (likely its steady state level);
    - them, define terminal (`endval;`) value of the shock (the value at which the economy should be simulated again);
    - see the convergence behaviour of the variables after the terminal condition is in place.

- ***stochastic model***
  - agents in the economy know that a shock will occur, but its occurrence is unknown
    - randomness around shocks would surprise agents when a shock is in place . . .

      . . . but future expectations around shocks are set to zero.
  - codes:
    - define `shocks;` block that shoudl finish with `end;`
    - in between, define the exogenous variable(s) on which the shock is computed with `var`; then, just impose the value of the shock (either with variance, `variance`, level, or in standard deviation, `stderr`);
    - see the convergence behaviour of the variables after the shock(s) hits.

# *log*-linearization

- so far, models are expolited in their *non-linear* form
  - difficult to analyze qualitatively and quantitatively bigger models
- *strategy*: approximate the equilibrium *non-linear* equations with *log*-linear ones
  - use a $n^{th}$-order Taylor expansion around the steady state to replace the equations with approximations, which are *linear in the log-deviations of the variables*;
  - there are many ways to *log*-linearize a set of equilibrium conditions.
- in what follows, I first briefly revise *log*-linearization . . .

  . . . to then turn to analyze the *log*-linearized version of an extended RBC model slightly different with that shown before
  - *file*: `RBC_loglin.mod` *(codes under request)*

# *log*-linearization: Basics

- *log*-linearization is a *first-order Taylor expansion*, expressed in *percentage terms* rather than in levels differences
  - in economics, units are not always well defined or consistent; think them in terms of percentage deviations from reference values, most likely its steady-state

✪ ***Taylor's theorem***: a differentiable function can be expressed as a power series, $n$, around a particular point, $x^*$, which belongs to the set of possible values:

$$
\begin{aligned}
f(x) &= f(x^*) + \sum_{n=1}^{\infty} \frac{f^{(n)} x^*}{n!} (x - x^*)^n \\
&= f(x^*) + \frac{f'(x^*)}{1!}(x - x^*) + \frac{f''(x^*)}{2!}(x - x^*)^2 + \frac{f^{(3)}(x^*)}{3!}(x - x^*)^3 + \dots
\end{aligned}
$$

- long, but safer, procedure (shorter procedure not covered):
  1. take *logs* on both sides of an equation;
  2. compute first-oder Taylor series of all the variables around their own steady state;
  3. define $\tilde{x}_t = \frac{x_t - x^*}{x^*}$ the *log*-deviation of variable $x$ around its reference point $x^*$;
  4. rewrite the whole equation(s) so as to isolate $\tilde{x}_t$.

# Extended RBC model: *Households*

- maximization problem

$$\max_{C_t,N_t,B_{t+1},K_{t+1}} \mathcal{U} = E_0 \sum_{t=0}^{\infty} \beta^t \left[ u\left(C_t\right) - f\left(1-N_t\right) \right]$$

$$s.t. \quad C_t + \underbrace{K_{t+1} + (1-\delta)\, K_t}_{I_t} + B_{t+1} = w_t N_t + \left(1 + r_{t-1}\right) B_t + R_t K_t + \Pi_t$$

- optimality conditions

$$\begin{aligned}
\frac{f'(1-N_t)}{u'(C_t)} &= w_t && \textit{labour supply} \\
u'(C_t) &= \beta E_t \left[ u'(C_{t+1}) \right] (1 + r_t) && \textit{Euler for bonds} \\
u'(C_t) &= \beta E_t \left\{ u'(C_{t+1}) \left[ R_{t+1} + (1-\delta) \right] \right\} && \textit{Euler for capital}
\end{aligned}$$

# Extended RBC model: *Firms*

- maximization problem

$$\max_{K_t, N_t, B_{t+1}} \mathcal{V} = E_0 \sum_{t=0}^{\infty} M^t \left[ \underbrace{A_t f\left(K_t, N_t\right)}_{Y_t} - \Big( w_t N_t + R_t K_t + B_{t+1} - (1 + r_{t-1}) B_t \Big) \right]$$

where $M^t = \frac{E_0 u'(C_t)}{u'(C_0)} \beta^t$ is the firm's stochastic discount factor, and technology evolves as a stochastic AR(1) process

$$log\left(A_t\right) = \rho_A log\left(A_{t-1}\right) + \epsilon_t$$

- optimality conditions

$$\begin{aligned} w_t &= A_t f_N\left(K_t, N_t\right) && \textit{labour demand} \\ R_t &= A_t f_K\left(K_t, N_t\right) && \textit{capital rental price} \\ u'(C_t) &= \beta E_t\Big[u'(C_{t+1})\Big](1 + r_t) && \textit{Euler for bonds} \end{aligned}$$

# Extended RBC model: *Equilibrium conditions*

- assuming functional forms as $u(C_t) = log(C_t)$ and $f\,(1 - N_t) = \theta\, log(1 - N_t)$, and Cobb-Douglas production function, the equilibrium system turns to be

$$
\begin{aligned}
w_t &= \theta \frac{C_t}{1 - N_t} \\
\frac{1}{C_t} &= \beta E_t \left[ \frac{1}{C_{t+1}} \right] (1 + r_t) \\
\frac{1}{C_t} &= \beta E_t \left\{ \frac{1}{C_{t+1}} \left[ R_{t+1} + (1 - \delta) \right] \right\} \\
K_{t+1} &= I_t + (1 - \delta) K_t \\
w_t &= (1 - \alpha) A_t K_t^{\alpha} N_t^{-\alpha} \\
r_t &= \alpha A_t K_t^{\alpha - 1} N_t^{1 - \alpha} \\
Y_t &= C_t + I_t \\
Y_t &= A_t f\,(K_t, N_t) \\
log\,(A_t) &= \rho_A log\,(A_{t-1}) + \epsilon_t
\end{aligned}
\tag{7}
$$

➪ 9 equations in 9 *endogenous* variables

# Extended RBC model: *Steady state*

- bring the above system such as $\overline{x} \equiv \Delta x_t = x_{t+1} - x_t = 0$, that is

$$\begin{aligned}
\overline{R} &= \rho + \delta = \frac{1}{\beta} + 1 - \delta \\
\frac{\overline{K}}{\overline{N}} &= \left[ \frac{\alpha \overline{A}}{\rho + \delta} \right]^{\frac{1}{1-\alpha}} \\
\overline{w} &= (1-\alpha)\overline{A} \left( \frac{\overline{K}}{\overline{N}} \right)^{\alpha} \\
\frac{\overline{C}}{\overline{N}} &= \overline{A} \left( \frac{\overline{K}}{\overline{N}} \right)^{\alpha} - \delta \left( \frac{\overline{K}}{\overline{N}} \right) \\
\overline{N} &= \frac{1-\alpha}{1 - \alpha + \theta \left( 1 - \frac{\alpha\delta}{\rho+\delta} \right)} \\
\frac{\overline{Y}}{\overline{N}} &= \overline{A} \left( \frac{\overline{K}}{\overline{N}} \right)^{\alpha} \\
\overline{A} &= 1
\end{aligned} \tag{8}$$

# Extended RBC model: *log*-linearized system

- finally, express the model equilibrium conditions in terms of *log-deviations from its deterministic steady state*

$$
\begin{aligned}
\tilde{w}_t &= \tilde{C}_t + \psi \tilde{N}_t, \quad \textit{with} \quad \psi = \frac{\overline{N}}{1-\overline{N}} \\
E_t\left[\tilde{C}_{t+1}\right] - \tilde{C}_t &= \frac{1}{1+\rho}\tilde{r}_t \\
\tilde{r}_t &= \overline{R}\tilde{R}_{t+1} \\
\tilde{K}_{t+1} &= \frac{1}{\overline{K}}\tilde{I}_t + \tilde{K}_t \\
\tilde{w}_t &= \tilde{A}_t + \alpha\tilde{K}_t - \alpha\tilde{N}_t \\
\tilde{R}_t &= \tilde{A}_t - (1-\alpha)\tilde{K}_t + (1-\alpha)\tilde{N}_t \\
\tilde{Y}_t &= \frac{\overline{C}}{\overline{Y}}\tilde{C}_t + \frac{\overline{I}}{\overline{Y}}\tilde{I}_t \\
\tilde{Y}_t &= \tilde{A}_t + \alpha\tilde{K}_t + (1-\alpha)\tilde{N}_t \\
\tilde{A}_t &= \gamma_a\tilde{A}_{t-1} + \epsilon_t
\end{aligned}
\tag{9}
$$

➪ *note*: to simulate the model in `dynare`, just plug this system and calibrate only the steady state values (not in the steady state block) appearing throughout

# Table of Contents

# Towards New Keynesian (NK) models

- economic analysis so far was based on models where markets are perfectly competitive and prices adjust instantaneously to clear markets
  - these analytical features are referred to as the *neoclassical* framework

- main departures of the New Keynesian (NK) from Real Business Cycle (RBC)
  1. goods markets are *imperfectly competitive* $\longrightarrow$ firms are price setters, thus having *market power* in the way they set prices
     - ★ each firm is assumed to hold a slight degree of monopoly power in the production of a differentiated variety
  2. *sticky prices* $\longrightarrow$ nominal goods prices do not fully adjust instantaneously

- regarding monetary policy, NK models are well suitable for its
  - analysis of its *systematic* component. Monetary policy:
    - ★ can be a source of shocks and surprises for the economy (*unanticipated* component);
    - ★ can follow a rule of behaviour to (endogenously) respond to the state of the economy (*systematic* component).
  - *normative* implications. What should monetary policy do?
    - ★ central hypothesis: minimize total loss after an unexpected shock in the economy

# Baseline *3 equations* NK model: *Set-up*

- recall the *log-linear* 3 eqs. DSGE-NK model:
  - equilibrium conditions

$$\begin{aligned}
\tilde{y}_t &= E_t\left[\tilde{y}_{t+1}\right] - \frac{1}{\sigma}\left(\tilde{i}_t - \tilde{r}^n - \mathbb{E}_t\left[\tilde{\pi}_{t+1}\right] - s_t^{IS}\right) && \textbf{\textit{NK-IS}} \\
\tilde{\pi}_t &= \kappa\,\tilde{y}_t + \beta E_t\left[\tilde{\pi}_{t+1}\right] + s_t^{\pi} && \textbf{\textit{NK-PC}} \\
\tilde{i}_t - \tilde{r}^n &= \phi_\pi \tilde{\pi}_t + \phi_y \tilde{y}_t + s_t^{MP} && \textbf{\textit{Taylor rule}}
\end{aligned}$$

    where:

    - ★ $\tilde{y}_t \equiv y_t - y_t^n$ is the "output gap ";
    - ★ $\beta$ is the discount factor;
    - ★ $\frac{1}{\sigma}$ is the *Inter-temporal Elasticity of Substitution* (IES);
    - ★ $s_t$ are related shocks.

  - deviations from the zero-inflation steady state:

$$\tilde{\pi}_t = 0, \quad \tilde{y}_t = 0, \quad \tilde{i}_t - \tilde{r}^n$$

# *Aside*: steps to compute the New Keynesian-Phillips Curve (NK-PC)

✄ given the F.O.C.s and their steady-state, next step is to *log*-linearize the system. I now review the basic (crazy) steps to obtain the ***NK-PC*** (labelled as $\pi_t$ above):

① write the ***optimal price*** $P_t^*$ in real term and substitute-in the stochastic discount factor;

② write $\frac{P_t^*}{P_t} = \frac{\epsilon}{\epsilon - 1} \frac{f_{nom}(\cdot)}{f_{den}(\cdot)}$ as $p_t^* = \frac{\epsilon}{\epsilon - 1} \frac{B_t}{C_t}$ and *log*-linearize it in such form to obtain $\mathcal{P}_t$;

③ *quasi-differentiation* of both the numerator and denominator:

- ★ start writing each summation, and it will appear (a sort of) recursive pattern;
- ★ due to its recursive nature, express the summation in just two periods.

thus obtaining two equations, $B_{t\,|\,t+1}$ and $C_{t\,|\,t+1}$. Compute their steady state $\left[\overline{\mathcal{B}}\,,\,\overline{\mathcal{C}}\right]$;

④ *log*-linearize $\left[B_{t\,|\,t+1}\,,\,C_{t\,|\,t+1}\right]$ around the steady-state, $\overline{\pi} = 1$, and obtain $\left[\mathcal{B}_t\,,\,\mathcal{C}_t\right]$;

⑤ plug $\mathcal{B}_t$ and $\mathcal{C}_t$ in $\mathcal{P}_t$ and solve to obtain $\mathcal{P}_t\left(\mathcal{B}_t\,,\,\mathcal{C}_t\right)$;

⑥ divide each element of the ***aggregate price index*** by $P_t$ and *log*-linearize it around the (zero inflation) steady state, $\overline{\pi} = 1$, thus obtaining $\mathcal{P}_t\left(\pi_t\right)$;

⑦ equate $\mathcal{P}_t\left(\mathcal{B}_t\,,\,\mathcal{C}_t\right) = \mathcal{P}_t\left(\pi_t\right)$ and solve for $\pi_t^{MC}$ $\Rightarrow$ ***NK-PC in terms of marginal costs***

⑧ characterize marginal costs in *log*-linear deviation from the output gap, $\mathcal{MC}_t$;

- ★ *log*-linearize marginal costs in both sticky and flexible prices, to then take their difference;

⑨ use $\mathcal{MC}_t$ in $\pi_t^{MC}$ to obtain $\pi_t$ $\Longrightarrow$ ***NK-PC in terms of output gap***

# Baseline *3 equations* NK model: *Shocks*

- slope of the Phillips curve
  - recall:

$$\kappa = \frac{(1-\theta)(1-\theta\beta)}{\theta}(\sigma+\phi);$$

    where:
    - $\theta$ is the fraction of fixed prices;
    - $\phi$ is the labor supply elasticity.

- shocks can be persistent
  - assume *autoregressive processes of order 1*, AR(1):

$$\begin{aligned} s_t^{IS} &= \rho_{IS}\, s_{t-1}^{IS} + e_t^{IS} \\ s_t^{\pi} &= \rho_{\pi}\, s_{t-1}^{\pi} + e_t^{\pi} \\ s_t^{MP} &= \rho_{MP}\, s_{t-1}^{MP} + e_t^{MP} \end{aligned}$$

    where:
    - $e_t^x$ are *iid* normally distributed with mean zero;
    - $\rho_x$ measures *persistence*.

# *Step 1*: Variables and parameters block

*file*: NK_3sim.mod

- notation:
  - D denotes *demand* (Euler);
  - S denotes *supply* (NK-PC);
  - R denotes *monetary policy* (Taylor rule).

- specify variables (endogenous var and exogenous varexo) and parameters

```
%% variables and parameters
var y pi r s_D s_S s_r;                                     % endogenous
varexo e_D e_S e_R;                                         % exogenous
parameters beta sigma phi chi phi_pi phi_y lambda
           rho_D rho_S rho_R kappa;                         % parameters

beta   = 0.99;
sigma  = 1;
phi    = 1;
theta  = 3/4
phi_pi = 1.5;
phi_y  = 0.5/4;
kappa  = ((1-theta)*(1-theta*beta)/theta)*(sigma+phi);
rho_D  = 0.9;
rho_S  = 0.9;
rho_R  = 0.3
```

# *Step 2*: Model block

- specify the equilibrium conditions

```
%% model equations
model(linear);

y = y(+1) - 1/sigma*(r-pi(+1) - s_D); % NK-IS curve

pi = beta*pi(+1) + kappa*y + s_S;     % NK-PC curve

r = phi_pi*pi + phi_y*y + s_r;        % Taylor rule

s_D = rho_D*s_D(-1)+e_D;              % demand (IS) shock

s_S = rho_S*s_S(-1)+e_S;              % supply (inflation) shock

s_r = rho_R*s_r(-1)+e_R;              % monetary policy shock

end;
```

- *Step 3*: Steady state block ⟶ no need to specify steady-state values (already used in the *log*-linearization process)

# *Step 4*: Shock block

- the command **shocks** defines the type of shock to be simulated

```
%% shock
shocks;

var e_D; stderr 1; % demand shock
var e_S; stderr 1; % supply shock
var e_R; stderr 1; % monetary policy shock

end;

check;

stoch_simul(order=1,irf=20) y r pi;
```

  - the command `stoch_simul(order=1,IRF=20)` computes:
    - a first order expansion around the steady state ...
    - impulse response functions (IRFs, 20 periods) ...
    - and various descriptive statistics (moments, variance decomposition, correlation and autocorrelation coefficients) ...
    - ... only for $\tilde{y}_t$, $\tilde{i}_t$ and $\tilde{\pi}_t$ variables.

# More equations

- the previous model shows the dynamics of inflation, output and the nominal interest rate only
  - what about other variables such as hours worked?

- these variables can be backed-out from the equilibrium conditions ...

  ... or solve a *log*-inearized system that includes these variables
  - *note*: the extended system can be reduced to the 3-equations model

*file*: `NK_5sim.mod`

# A *5 equations* NK model: *Set-up*

- *log*-linearized equilibrium conditions
  - remove expectations to simplify notation

$$
\begin{aligned}
\tilde{c}_t &= \tilde{c}_{t+1} - \sigma^{-1}\left(\tilde{i}_t - \tilde{\pi}_{t+1}\right) && \textit{\textbf{NK-IS}} \\
\varphi \tilde{h}_t &= -\sigma \tilde{c}_t + \tilde{w}_t - \tilde{p}_t && \textit{\textbf{labor supply}} \\
\tilde{\pi}_t &= \beta \tilde{\pi}_{t+1} + \tfrac{(1-\theta)(1-\theta\beta)}{\theta}\left(\tilde{w}_t - \tilde{a}_t - \tilde{p}_t\right) && \textit{\textbf{pricing + price index (NK-PC)}} \\
\tilde{i}_t &= \phi_\pi \tilde{\pi}_t + s_t^{MP} && \textit{\textbf{simplified MP rule}} \\
\tilde{c}_t &= \tilde{y}_t = \tilde{a}_t + \tilde{h}_t && \textit{\textbf{production and market clearing}}
\end{aligned}
$$

- to simplify notation
  - define $\tilde{\mu}_t = \tilde{p}_t - \tilde{w}_t + \tilde{a}_t$ as the average markup over marginal costs;
  - $\tilde{\mu}_t$ varies over time because some prices are fixed $\rightarrow$ it follows the dynamics of $\tilde{a}_t - \tilde{w}_t$.

# More versions

- extended baseline NK model with marginal costs rather than the output gap
  $\longleftrightarrow$ *file*: `Valerio_NK_Basic_loglinear_MargCost.mod` (*codes under request*)

- model indeterminacy and Blanchard and Khan (1980) conditions analysis
  $\longleftrightarrow$ *file*: `Valerio_NK_Basic_loglinear_BKcond.mod` (*codes under request*)

- version of the baseline NK model featuring physical capital accumulation
  $\longleftrightarrow$ *file*: `Valerio_NK_Benchmark_loglinear.mod` (*codes under request*)

# Optimal monetary policy

- how monetary policy should be conducted?
  - how the short-term nominal interest rate should be set in order to maximize total welfare at the steady state and, especially, in response to shocks

- *forward-looking* nature of NK models

$\longleftrightarrow$ substantial differences between ***discretion*** (period-by-period optimization) and ***commitment*** (policy path)

  - when the two are different, optimal monetary policy under commitment is said to be *time-inconsistent* (Kydland and Prescott, 1977)

➪ the criterion for an optimal monetary policy is to minimize the loss function[8]

$$\min_{\{\pi_t\,,\,x_t\}_{t=0}^{\infty}} E_0 \sum_{t=0}^{\infty} \beta^t \left(\pi_t^2 + \alpha_x x_t^2\right) \quad s.t. \quad \textbf{\textit{NK-PC}}$$

after the occurrence of a given shock(s), with $x_t$ identifying the output gap

[8] The ***NK-IS*** is not binding since the interest rate does not enter in the minimization problem. As a consequence, monetary authority can change the interest rate up to the desired level without causing any loss (unrealistic).

## Commitment vs. discretion



***Discretion***. Expectations of future variables taken as given (labelled as $Z_t$ and $z_t$)

$$\min_{x_t} \left[ \pi_t^2 + \alpha_x x_t^2 + Z_t \right] \quad s.t. \quad \pi_t = \kappa x_t + z_t$$

- sequence of single-period optimizations
  - → *targeting rule*: $x_t = -\frac{\kappa}{\alpha_x} \pi_t$
  - "lean against the wind" policy prescription: If inflationary pressures arise, contract output gap $(x_t < 0)$ to bring inflation back to steady state;
  - *trade-off* between inflation $(\pi_t)$ and output gap $(x_t)$ stabilization.

- problems:
  - interest rate according to a shock in the output gap rather than to inflation;
  - Taylor principle not satisfied: no reaction to inflation!
    - ★ in fact, by solving the above problem, substituting first order conditions in both NK-PC and NK-IS, one can write that the optimal path of the nominal interest rate is given by

$$i_t = \frac{\sigma \kappa}{\alpha_x + \kappa^2} u_t$$

called the *instrument rule*, with $u_t$ being some shock

***Commitment***. Monetary authority is able to affect expectations, and agents know and credibly believe this. It chooses $\{\pi_t\,,\,x_t\}_{t=0}^{\infty}$ that minimize the loss function

$$\mathcal{L}_{\{\pi_t\,,x_t\,,\lambda_t\}_{t=0}^{\infty}} = E_0 \sum_{t=0}^{\infty} \beta^t \left[ \pi_t^2 + \alpha_x x_t^2 - \lambda_t \left( \pi_t - \kappa x_t - \beta \pi_{t+1} - s_t^{\pi} \right) \right]$$

- inflation and output gap exhibit *history dependence* (***endogenous inertial behavior***)
  - inertia affects inflation expectations and improves inflation-output gap trade-off

- *pros* and *cons*:
  - generally, policy inertia improves predictability of future policy actions;
  - *credibility* problem of commitment solution;
  - *time-inconsistency* of optimal policy: commitment plan at time $t$ is optimal at time $t$ but not at time $t+1$;
    - if the policymaker cannot commit, *i.e.*, affect expectations, it doesn't reap the gains in terms of better stabilization performance;
    - desirability of institutional frameworks that secure and enhance credible policy commitments.

# Commitment vs. discretion in dynare

## Discretion

- set the `linear` option of the model block, and define the monetary instrument;
- steady state checks (`steady;` and `check;`), and define the shock(s);
- define the welfare loss function: `planner_objective pi^2 + alpha_x*y_gap^2;`
- compute an approximation of the optimal policy under discretion and trigger optimal policy responses by adding the command `discretionary_policy();`

  - limited to quadratic loss function;
  - in `discretionary_policy();` may add several options, *e.g.*, a way of writing is `discretionary_policy(instruments=(i),irf=20,planner_discount=beta, discretionary_tol=1e-12)) <variables for IRFs>;` which denotes, respectively:
    - instrument (*i.e.*, a variable) for the computation of the steady state under optimal policy;
    - discount factor of the monetary authority;
    - tolerance level used to assess convergence of the solution algorithm.

## Committment

- almost identical to the ones for discretion;
- instead of `discretionary_policy();` use `ramsey_policy();`

  - not limited to quadratic objectives; may use any arbitrary nonlinear expression;
  - avoid to include `steady;` since, under commitment, there is no steady state;

# Analysis

- to implement the *discretion vs. commitment* analysis, I will rely on two different types of monetary authority instruments:
  1. the ***Fisher equation***, so that the instrument is the *real interest rate*,

  $$r_t = i_t - E_t\left[\pi_{t+1}\right]$$

  2. the ***Taylor rule***, so that the instrument is the monetary authority *response of nominal interest rate to inflation and output gap* (below takes into account also lagged interest rate),

  $$i_t = \phi_1 i_{t-1} + (1-\phi_1)\left(\phi_2 \pi_t + \phi_3 x_t\right) + \nu$$

  where $x_t \equiv y_t - y_t^n$ is the output gap.

- optimal monetary policy in a New Keynesian (NK) model with 3 equations:
  - discretion $\longleftrightarrow$ *files*: `NK_discretion_fisher.mod` and `NK_discretion_taylor.mod`
  - commitment $\longleftrightarrow$ *files*: `NK_commitment_fisher.mod` and `NK_commitment_taylor.mod`

## Table of Contents

# Main issue

- DSGE models are often seen as abstractions of actual economies
  - in the examples shown so fare, all the parameters were calibrated with some values *believed* to be true
  - they may hold in some cases. What about other scenarios?
    - ★ some parameters may not be present in any other models, thus impossible to be calibrated ...
    - ★ existing parameters may be only for US, thus not holding for EU or other areas ...

  → these models should feature an *internal* calibration of (most of) the parameters

- up to late 1990s, optimization techniques[9] implemented to calibrate DSGE models
  - assumption of a *true* model binds the identification to model misspecification
  - can any parameters of a DSGE model be identified when such model is misspecified?

- bayesian techniques avoid to assume there exists a true DSGE model because of the *likelihood principle* (LP)
  - all its evidence is in its *likelihood conditional on the data* (see Berger and Wolpert, 1988)

  ➪ thus, bayesian likelihood-based evaluation is consistent with the view that there is no true DSGE model

[9] These are *maximum likelihood* (ML), *generalized method of moments* (GMM), and *indirect inference* (II).

# Turning a DSGE model into a *bayesian* model

- ***Bayesian estimation*** $\rightarrow$ tool to estimate (DSGE) models using real data series

- a bayesian model consists of a joint distribution of data $Y$ and parameters $\Theta$
  - *priors*, $p(\Theta)$, a priori beliefs about the parameter vector $\Theta$;
  - *likelihood function*, $p(Y \mid \Theta)$, to update *priors* in view of the sample information (*i.e.*, the data);
  - *posteriors*, $p(\Theta \mid Y)$, state of knowledge about the parameter vector after the updating;

$$p(\Theta \mid Y) \propto p(Y \mid \Theta)\, p(\Theta)$$

☞ *Bayes theorem* provides the formal link between these three elements.

- *steps*:
  1. write the model in *reduced form* state equation in its predetermined variables;
  2. *state space representation* (transition equation + measurement equation);
  3. *Kalman filter* to evaluate the likelihood function, $p(Y \mid \Theta)$;
  4. combine the prior information on the parameters, $p(\Theta)$, with the likelihood of the data;
  5. *Metropolis-Hastings algorithm* to obtain the posteriors, $p(\Theta \mid Y)$.

# State-space representation

- compact summary of the model, and mapping between the model and the data
  - recall that a model is just a set of nonlinear equations $E_t f\left(x_t, x_{t+1}, u_t, u_{t+1}\,;\,\sigma\right) = 0, \forall t$
- its *canonical linear rational expectations* (RE) *form* (see Sims 2002) is

$$\Gamma_0 s_t = \Gamma_1 s_{t-1} + \Psi u_u + \Pi \eta_t$$

with $s_t = \{x_t, E\left[x_t^c\right]\}$ and $\eta_t$ is one-step ahead RE forecast errors

- the *state-space representation* of the model is

$$s_t = \Phi_1(\Theta)\, s_{t-1} + \Phi_u(\Theta)\, \sigma_t \qquad \textit{transition equation}$$

$$y_t = \Psi_0(\Theta) + \Psi_1(\Theta)t + \Psi_2(\Theta)\, s_t + \epsilon_t \qquad \textit{measurement equation}$$

with $\Psi$ and $\Phi$ being matrix of coefficients functions of the structural parameters, and $\epsilon_t$ the measurement error

➪ the state-space representation provides a *joint density for the observations and latent states given the parameters*

# A practical example

- once that a model has been *log*-linearized, its equilibrium conditions can be cast as an expectational stochastic difference equation

$$E_t f\left(\tilde{x}_t^c, \tilde{x}_{t-1}^c, \tilde{x}_t^s, \tilde{x}_{t-1}^s, \tilde{u}_t, \tilde{u}_{t-1}\ ;\ \sigma\right) = 0, \quad \forall t$$

  where $\tilde{x}_t^c$, $\tilde{x}_t^s$ and $\tilde{u}_t$ are vectors of *log*-linear control, state and exogenous variables, and $\sigma$ is a vector of exogenous structural innovations in $\tilde{u}_t$

- the solution of this DSGE model can be written in *state-space* form as
  - *transition* equation, showing the decision rules of the state variables

$$\tilde{x}_t^s = \Phi_1(\Theta)\, \tilde{x}_{t-1}^s + \Phi_u(\Theta)\, \sigma_t$$

  - *measurement* equation, giving a mapping from the state to the control variables

$$\tilde{x}_t^c = \Psi_0(\Theta) + \Psi_1(\Theta)t + \Psi_2(\Theta)\, \tilde{x}_t^s + \epsilon_t$$

## *Filtering* and *posteriors*

- the *state-space form* is needed to compute the *likelihood function*
  - the *measurement equation* links the state variables to ***observable*** variables[10]

- since DSGE model is *log*-linearized and the errors are Gaussian, apply *Kalman filter* (a linear prediction error algorithm) to estimate the likelihood function
  - derive the *log*-likelihood given by $log\, p\left(Y \mid \theta\right) \ldots$
  - so that the *log*-posterior Kernel is nothing but $log\, K\left(\theta \mid Y\right) = log\, p\left(Y \mid \theta\right) + log\, p\left(\theta\right)$

- then, to find the mode of the posterior distribution, $log\, K\left(\theta \mid Y\right)$ function has to be maximized with respect to the parameter vector $\theta$

- finally, to compute the *posterior distribution* of the parameters, implement the *Metropolis-Hastings algorithm* (a rejection algorithm) that draws new parameters from that posterior distribution

[10] These are the variables for which data are used in the estimation procedure.

## Bayesian estimation in `dynare`



- fortunately, bayesian estimation in `dynare` is very easy compared to the theoretical background behind it

- after having simulated the model under arbitrary calibrated parameters[11]:
  - set the values of the *priors*;
  - declare the *observable* variables, *i.e.*, those variables for which data are available in the estimation;
  - estimate the model using bayesian techniques.

❊ all the above steps can be done in few lines

*file*: `Bayesian_Smets_Wouters_2007.mod`
*data*: `bayesian_SM_usmodel_data.xls`

*file*: `Bayesian_Albonico_Kalyviti_Pappa_2014.mod`
*data*: `bayesian_AKP_maint_data_itot.xls`

(*all under request*)

[11] If you cannot get your model to run when you get to pick the parameters, estimation typically won't help either! When everything is up and running, you can still uncomment the `stoch_simul()` line below and directly go to estimation.

### Priors

- starts with `estimated_params;`
- ends with `end;`
- in between, type *parameter name, [, initial value [, lower bound, upper bound ] ], shape, mean, std. err.*;

  - in most cases, it is only necessary *parameter name, shape, mean, std. err.*, (*e.g.*, `alpha, NORMAL_PDF, 0.7, 0.05;`)

### Observables

- `varobs` *list observables variables name*

  - in order to avoid stochastic singularity, need to have at least as many shocks or measurement errors as observed variables

# Bayesian estimation in dynare

## Estimation

- `identification;`
- `dynare_sensitivity;`
- `estimation(optim=('MaxIter',200),datafile=maint_data_itot, mode_compute=4,first_obs=1,presample=4,lik_init=2,prefilter=0, plot_priors=1,mh_conf_sig=0.95, mh_replic=2500,mh_nblocks=2,mh_jscale=0.20,mh_drop=0.2,bayesian_irf, irf=40,moments_varendo,filtered_vars,smoother)`
- *list variables to plot IRFS;*

- `identification(...)` aims at identifying the parameters whose statistical properties do not affect the parameters' updating;
- `dynare_sensitivity` performs a global sensitivity analysis prior the estimation;
- to estimate the model, add the command `estimation(...)` with all the options you need (references at https://archives.dynare.org/manual/Estimation.html).

# Interpreting the output of `dynare` 1/4

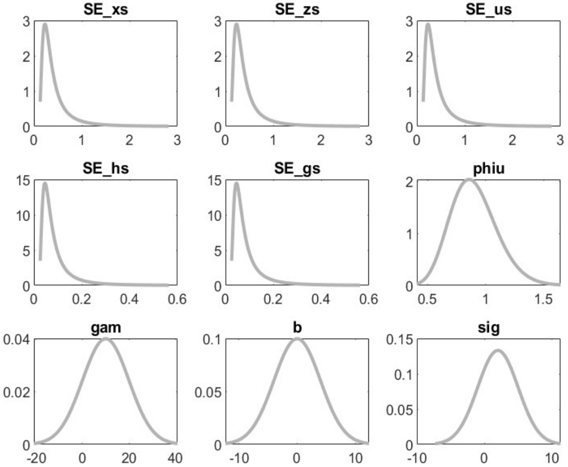
- in the section `ESTIMATION RESULTS`, it is possible to get a summary of the estimation of the parameter values . . .

```
ESTIMATION RESULTS

Log data density is 395.060040.

parameters
        prior mean   post. mean        95% HPD interval     prior        pstdev

phiu         0.900       1.4773      1.1745       1.8285     gamm        0.2000
gam         10.000       3.5222      0.4839       7.3067     norm       10.0000
```

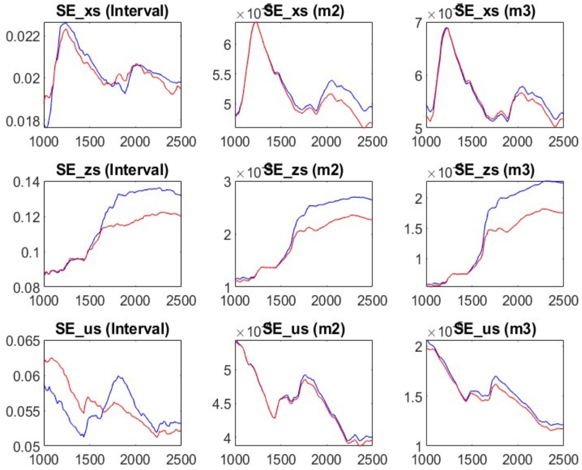
- . . . and the same results for the shocks

## Interpreting the output of `dynare` 2/4

- graphical representation of the *priors* for each parameter of interest

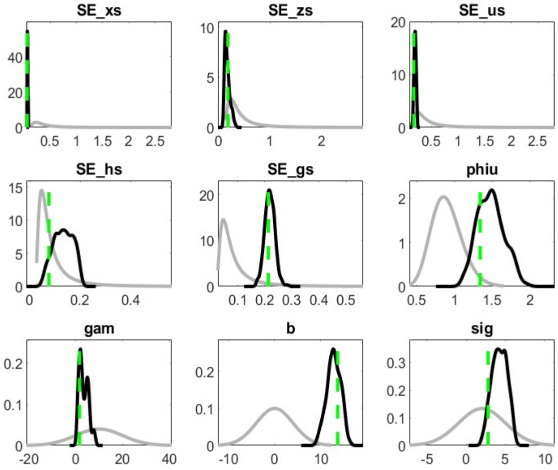


- diagnostics graph for convergence in the estimation



# Interpreting the output of `dynare`

- comparison between the posterior (*black*) and the prior (*grey*) distributions




  - the distributions should not be excessively different;
  - the posterior distributions should be close to normal, or at least not display a shape that is clearly non-normal;
  - the *green* mode should not be too far away from the mode of the posterior distribution.

# Bayesian impluse respons functions (IRFs)

- then, `dynare` will report the IRFs implied by the new calibrated values

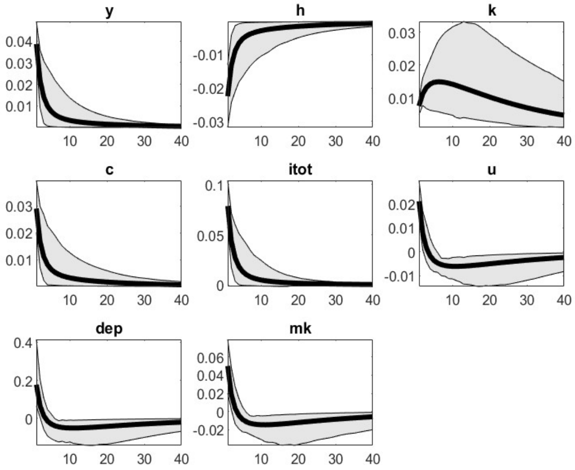



Figure 6.1: Productivity shock

# Historical decomposition

- finally, `dynare` will decompose the historical deviations
  - of the endogenous variables from their respective steady state values . . .

    . . . into the contribution coming from the various shocks the model features

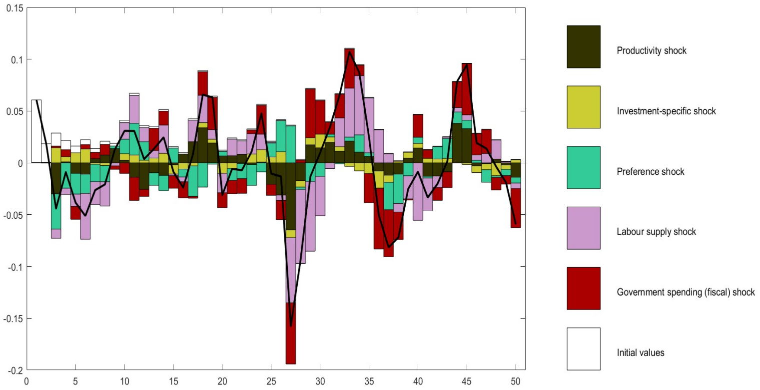



Figure 6.2: Decomposition of output