Learning Guide for "The BIS Multisector Model: A Multi-Country Environment for Macroeconomic Analysis" *

Matthias Burgert[§] Giulio Cornelli[†] Burcu Erik[†] Benoit Mojon[†]
Daniel Rees[†] Matthias Rottner[†,‡]

October 9, 2025

1 Introduction

This accompanying material presents a learning guide for the toolbox of the BIS Multisector Model (Burgert et al., 2025). The model is a dynamic stochastic general equilibrium (DSGE) framework designed to analyze macroeconomic dynamics within a multi-sector production network readily applicable to the input-output tables of more than 80 countries. The studied shocks can be either temporary or permanent shocks to account for transition dynamics. The toolbox also features alternative expectation formation mechanisms.

This guide provides practical examples to demonstrate how to use the toolbox accompanying the BIS-MS model. First, we discuss how to set up the toolbox, which requires Matlab, Matlab Optimization Toolbox and Dynare version 5.5. Then, it guides the user through the capabilities of the toolbox by using two examples: the simulation of a temporary shock and the simulation of a structural change in the economy.

*The views in this paper do not necessarily reflect the views of the Bank for International Settlements (BIS), the Deutsche Bundesbank or the Swiss National Bank (SNB).

Emails: matthias.burgert@snb.ch, giulio.cornelli@bis.org, burcu.erik@bis.org, benoit.mojon@bis.org, daniel.rees@bis.org, and matthias.rottner@bis.org.

[†]Bank for International Settlements

[‡]Deutsche Bundesbank

[§]Swiss National Bank

1.1 Installing the toolbox

After downloading, the user will find a folder named "BIS_Multisector_Model", which contains the following files and sub-folders:

- The master codes to run the simulations (*run_temporary_shock.m* and *run_permanent_shock.m*) and *bis_multisector_model.mod*, which estimates the model.
- A sub-folder called "utility" that contains *model_run.m*, which sets up the inputs and the Dynare file, along with all the relevant functions and routines called by the model, including all the necessary equations.[1]
- A sub-folder called "input" that contains raw data and functions used to process the raw data into standardised form.

1.2 Multi-Country Environment

The BIS-MS model utilises three sources of I-O data: (i) the OECD Inter-Country Input-Output (ICIO) tables (2023 release),[2] (ii) the Asian Development Bank (ADB) Multi-Regional Input-Output (MRIO) tables (2022 release), and (iii) the US Bureau of Economic Analysis (BEA) "Use of Commodities by Industry" summary (retrieved in 2023). The OECD ICIO tables provide data for 76 economies, while the ADB MRIO tables cover 62 economies; both also include data for the "rest of the world". Although there is some overlap between the datasets, their combined use allows us to cover a total of 84 economies (Euro Area is obtained by summing up data for individual member countries). A complete list of the economies covered by both sources is provided in Table 1.

[1]Among these, we include the function *dyn_to_str.m*, developed by Callum Jones, which is used to rebuild the structural matrices from the Dynare output. It also computes the reduced form matrices from the extracted structural matrices. For details, please see https://callumjones.github.io/files/dynare_str.pdf.

[2]Yamano, N. et al. (2023), "Development of the OECD Inter Country Input-Output Database 2023," OECD Science, Technology and Industry Working Papers, No. 2023/08, OECD Publishing, Paris, https://doi.org/10.1787/5a5d0665-en.

Table 1: List of economies by source

ADB MRIO		OECD ICIO	
-	KG: Kyrgyz Republic	AR: Argentina	-
AT: Austria	KH: Cambodia	AT: Austria	KH: Cambodia
AU: Australia	KR: Korea	AU: Australia	KR: Korea
BD: Bangladesh	KZ: Kazakhstan	BD: Bangladesh	KZ: Kazakhstan
BE: Belgium	LA: Laos	BE: Belgium	LA: Laos
BG: Bulgaria	LK: Sri Lanka	BG: Bulgaria	-
BN: Brunei	LT: Lithuania	BN: Brunei	LT: Lithuania
BR: Brazil	LU: Luxembourg	BR: Brazil	LU: Luxembourg
BT: Bhutan	LV: Latvia	-	LV: Latvia
-	-	BY: Belarus	MA: Morocco
CA: Canada	-	CA: Canada	MM: Myanmar
CH: Switzerland	MN: Mongolia	CH: Switzerland	-
-	MT: Malta	CI: Côte d'Ivoire	MT: Malta
-	MV: Maldives	CL: Chile	-
-	MX: Mexico	CM: Cameroon	MX: Mexico
CN: China	MY: Malaysia	CN: China	MY: Malaysia
-	-	CO: Colombia	NG: Nigeria
-	NL: Netherlands	CR: Costa Rica	NL: Netherlands
CY: Cyprus	NO: Norway	CY: Cyprus	NO: Norway
CZ: Czechia	NP: Nepal	CZ: Czechia	-
DE: Germany	-	DE: Germany	NZ: New Zealand
DK: Denmark	-	DK: Denmark	PE: Peru
EA: Euro area	PH: Philippines	EA: Euro area	PH: Philippines
EE: Estonia	PK: Pakistan	EE: Estonia	PK: Pakistan
-	PL: Poland	EG: Egypt	PL: Poland
ES: Spain	PT: Portugal	ES: Spain	PT: Portugal
FI: Finland	RO: Romania	FI: Finland	RO: Romania
FJ: Fiji	RU: Russia	-	RU: Russia
FR: France	-	FR: France	SA: Saudi Arabia
GB: United Kingdom	SE: Sweden	GB: United Kingdom	SE: Sweden
GR: Greece	SG: Singapore	GR: Greece	SG: Singapore
HK: Hong Kong	SI: Slovenia	HK: Hong Kong	SI: Slovenia
HR: Croatia	SK: Slovakia	HR: Croatia	SK: Slovakia
HU: Hungary	-	HU: Hungary	SN: Senegal
ID: Indonesia	TH: Thailand	ID: Indonesia	TH: Thailand
IE: Ireland	-	IE: Ireland	TN: Tunisia
-	TR: Türkiye	IL: Israel	TR: Türkiye
IN: India	TW: Chinese Taipei	IN: India	TW: Chinese Taipei
-	-	IS: Iceland	UA: Ukraine
IT: Italy	US: United States	IT: Italy	US: United States
-	VN: Vietnam	JO: Jordan	VN: Vietnam
JP: Japan	-	JP: Japan	ZA: South Africa

When running the scripts, *run_temporary_shock.m* and *run_permanent_shock.m*, both located in the main folder, the users are prompted to specify the country code and preferred data source. The prompts include a list of available options, the required format, and examples to guide the user. In turn, the function *input/progs/f_main_-prepare_data_auto.m* retrieves and processes raw data from the selected source and saves the standardised data in the "input" sub-folder (eg, *US_io_results_oecd.mat*).

```
Command Window
-------------------

List of available economies by source:
  - Asian Development Bank Multi-Regional Input-Output tables (ADB MRIO):
    AT, AU, BD, BE, BG, BN, BR, BT, CA, CH, CN, CY, CZ, DE, DK, EA, EE, ES, FI, FJ, FR, GB, GR, HK, HR, HU,
    ID, IE, IN, IT, JP, KG, KH, KR, KZ, LA, LK, LT, LU, LV, MN, MT, MV, MX, MY, NL, NO, NP, PH, PK, PL, PT,
    RO, RU, SE, SG, SI, SK, TH, TR, TW, US, VN
  - Organisation for Economic Co-operation and Development Inter-Country Input-Output tables (OECD ICIO):
    AR, AT, AU, BD, BE, BG, BN, BR, BY, CA, CH, CI, CL, CM, CN, CO, CR, CY, CZ, DE, DK, EA, EE, EG, ES, FI,
    FR, GB, GR, HK, HR, HU, ID, IE, IL, IN, IS, IT, JO, JP, KH, KR, KZ, LA, LT, LU, LV, MA, MM, MT, MX, MY,
    NG, NL, NO, NZ, PE, PH, PK, PL, PT, RO, RU, SA, SE, SG, SI, SK, SN, TH, TN, TR, TW, UA, US, VN, ZA
  - United States Bureau of Economic Analysis Input-Output table (US BEA IO):
    US
* Note: EA for the euro area.
USER INPUT NEEDED: Enter the two-letter ISO code of the economy of your choice (eg, US) and press `Enter'.
See above for the list of available country codes.
> US
```

```
Command Window
-------------------

List of available sources for US:
  adb  : Asian Development Bank Multi-Regional Input-Output tables (ADB MRIO)
  bea  : United States Bureau of Economic Analysis Input-Output table (US BEA IO)
  oecd : Organisation for Economic Co-operation and Development Inter-Country Input-Output tables (OECD ICIO)
USER INPUT NEEDED: Enter the abbreviation for the data source of your choice (eg, oecd) and press `Enter'.
See above for the list of available abbreviations.
> oecd
```

1.3 Simulating a temporary shock

In this section, we simulate a temporary shock to the total factor productivity (TFP) of the energy sector, with spillovers to the manufacturing sector (ie the shock eps_mm as used in Section 4 in the main paper), under different monetary policy rules for the United States. Specifically, the shock will last for four quarters and will target a change of approximately one-quarter of the corresponding steady-state value.

The main script for running the temporary shock is the *run_temporary_shock.m*, located in the main folder. When running the script, the users are first prompted to provide the Dynare path on their local drives if it is not already in the Matlab search path. Next, they are asked to specify country code and preferred data source as explained in Section 1.2.

To modify the properties of the shock, the user should adjust the *run_temporary_shock.m* script before running it. Lines 47 and 50 are used to specify the horizon for the resulting impulse response functions (IRFs) and the number of shocks to be simulated. In the subsequent block of code, the user can define the shock and the variable that the shock will target (lines 53 and 57). Information on the specific shocks and the variable names can be found in the MATLAB script *utility/variables_exo.m* and *utility/model_2_parameters.m*, respectively.

```
% Set the horizon of the IRFs and the number of shocks
horizon = 20;
% If numshk is larger than 1, selected_shock and selected_target must be of
% the same size as the number of shocks
numshk = 1;

% Select the shocks
selected_shocks = {'eps_mm'};
exovars1 = selected_shocks;

% Select the target
selected_targets = {'gam_2'};
targetvar1 = selected_targets;
```

Similarly, lines 61, 65 and 69 indicate the number of periods for which the temporary shock is active (ie 4 in this example), the magnitude of the shock (25% change relative to the steady state value) and the number of quarters considered for the AIT rule, in case this is the monetary policy rule selected at any point of the exercise.

```
60  % Specify the number of periods for which the shock is active
61  number_of_periods = 4;
62
63  % Specify the size of the change in the target variable
64  % 1 stands for doubling the steady state value
65  target_change_size = repmat(.25,length(targetvar1),number_of_periods);
66
67  % Specify the number of quarters for AIT rule
68  % n_ait_parameter = 1 >>> IT
69  n_ait_parameter = 8;
```

After completing this preamble, we are ready to derive the baseline results which are estimated under a standard headline inflation targeting regime.

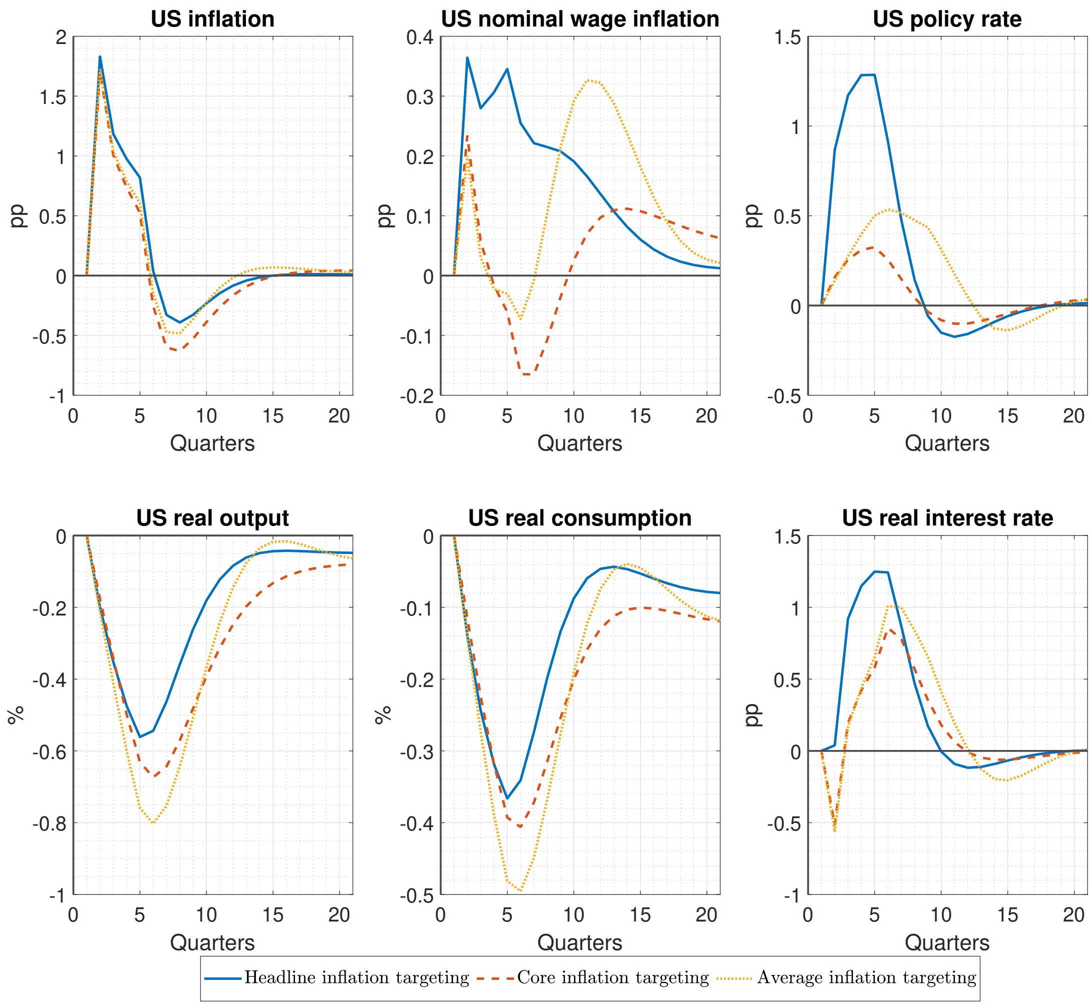
The next block of code (starting at line 99) is used to simulate the same shock under different monetary policy rules. The struct *exercises* contains all the different specifications that the users wants to simulate, where each "*id_*", in our example, corresponds to a specific monetary policy rule. Specifically, the struct elements for *id_2* correspond to a scenario in which the monetary policy rule targets core inflation instead of headline inflation, as in the baseline scenario. The field *names* specifies the names of the parameters for which we want to change the associated value. As mentioned previously, the names of specific parameters can be found in the Matlab script *utility/model_2_parameters.m*. The field *names_legend* defines the legend labels that will appear in the final graph showing the IRFs produced by the code. The field *values* specifies the new values for the parameters of interest. Finally, the field *targetvar1* defines the target variable for the shock being simulated.

```
% Define the exercises
% Core
exercises = struct();
exercises.id_2 = struct();
exercises.id_2.id = 2; % Must be at least 2 because baseline exercise is 1
% Set the names of the parameters to change
exercises.id_2.names =        {'phi_pie';'phi_pie_core'};
exercises.id_2.names_legend = {'\phi_{\pi}';'\phi_{core}'}; % LaTeX format
% Set the new values of the parameters to change
exercises.id_2.values = [0;1.5];
% Set the target variable
exercises.id_2.targetvar1 = extractAfter(selected_shocks{1},'eps_');
```

By changing the values associated with these fields, the user can specify as many scenarios as desired. For instance, if in addition to scenario *id_2* the user would like to add a simulation in which monetary policy follows an Average Inflation Targeting (AIT) rule, all we have to do is to define a new scenario *id_3* and update the relevant values in the fields *names* (ie *phi_pie* and *phi_pie_bar*), *names_legend* (ie ϕ_π and $\phi_{\overline{\pi}}$) and *values* (ie 0 and 1.5).

The loop in lines 123-144 executes the various simulations and stores the corresponding results. The final section of the code includes a few lines to plot the derived impulse response functions and facilitate the comparison of the results (Figure 1).

Figure 1: Temporary energy price shock under alternative monetary policy rules

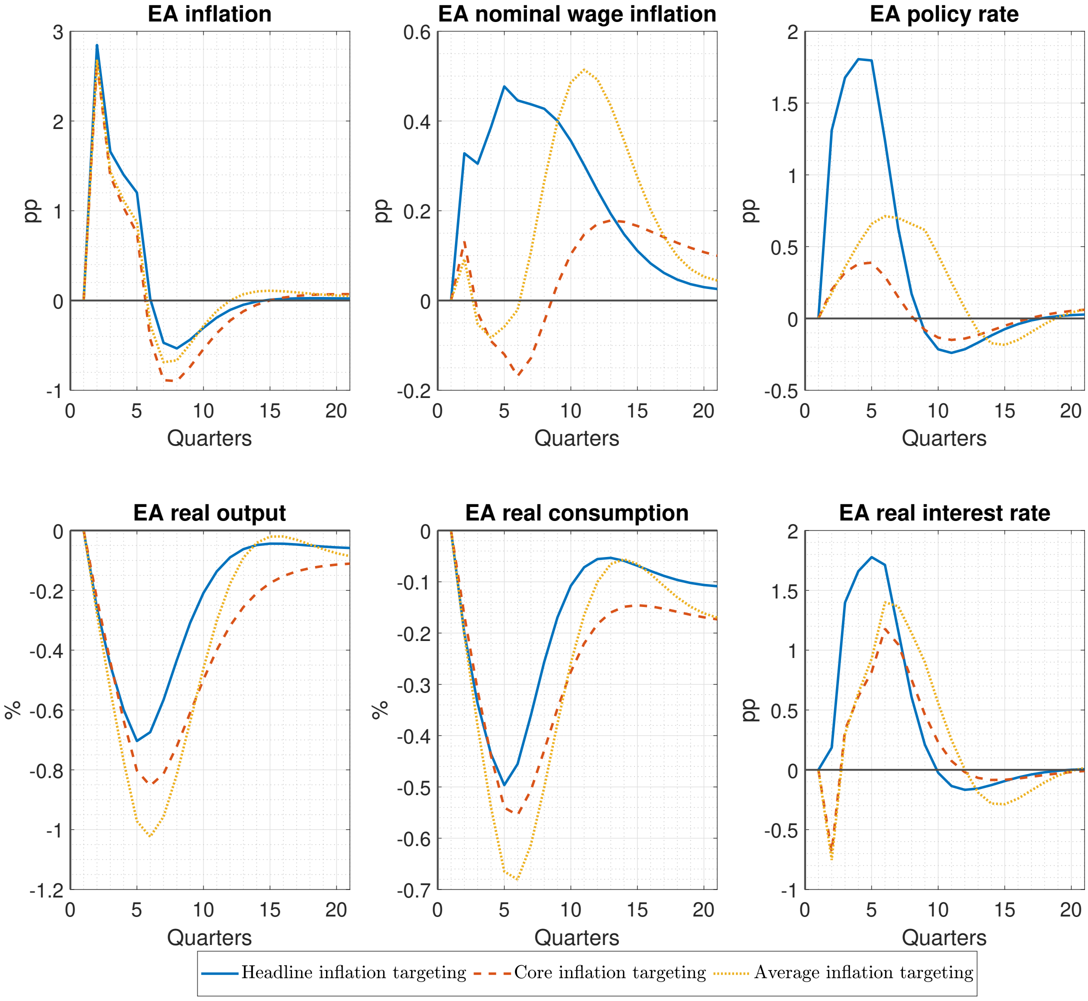


The results are stored in automatically generated country-specific sub-folders within "output\temporary_shock" and "graphs\temporary_shock" named according to the two-digit country code of the selected country.

1.3.1 Exercise 1: simulating the temporary shock for a different country

In the exercise presented in the previous section we derived the impulse response functions for the US under a temporary shock to the total factor productivity (TFP) of the energy sector, with spillovers to the manufacturing sector. To run the same exercise for a different country, the user just needs to re-run the code and enter the two-digit ISO code of the country of their choice when prompted. The results for EA are reported in Figure 2 as an example.

Figure 2: Temporary energy price shock under alternative monetary policy rules



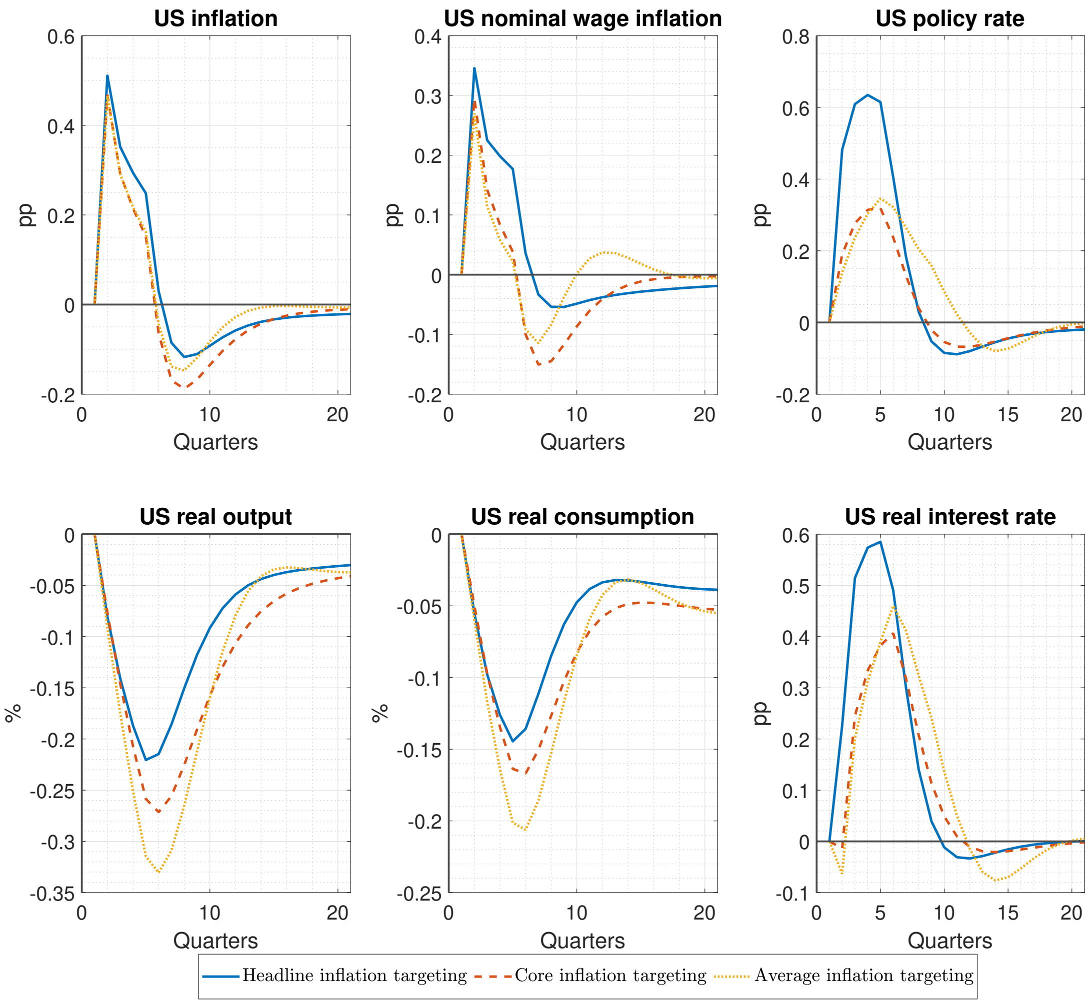
1.3.2 Exercise 2: simulating a different temporary shock for the US

The toolbox allows users to simulate different shocks. For instance, to simulate a temporary shock to the total factor productivity (TFP) of the energy sector, without spillovers to the manufacturing sector, the user just needs to modify the shock specified in line 53 of the *run_temporary_shock.m* code. Specifically, the user should change *selected_shocks* from `'eps_mm'` to `'eps_a_2'` and then re-run the entire code to estimate the results.

```
52    % Select the shocks
53    selected_shocks = {'eps_a_2'};
54    exovars1 = selected_shocks;
```

The results are reported in Figure 3.

Figure 3: Temporary shock to the total factor productivity of the energy sector under alternative monetary policy rules



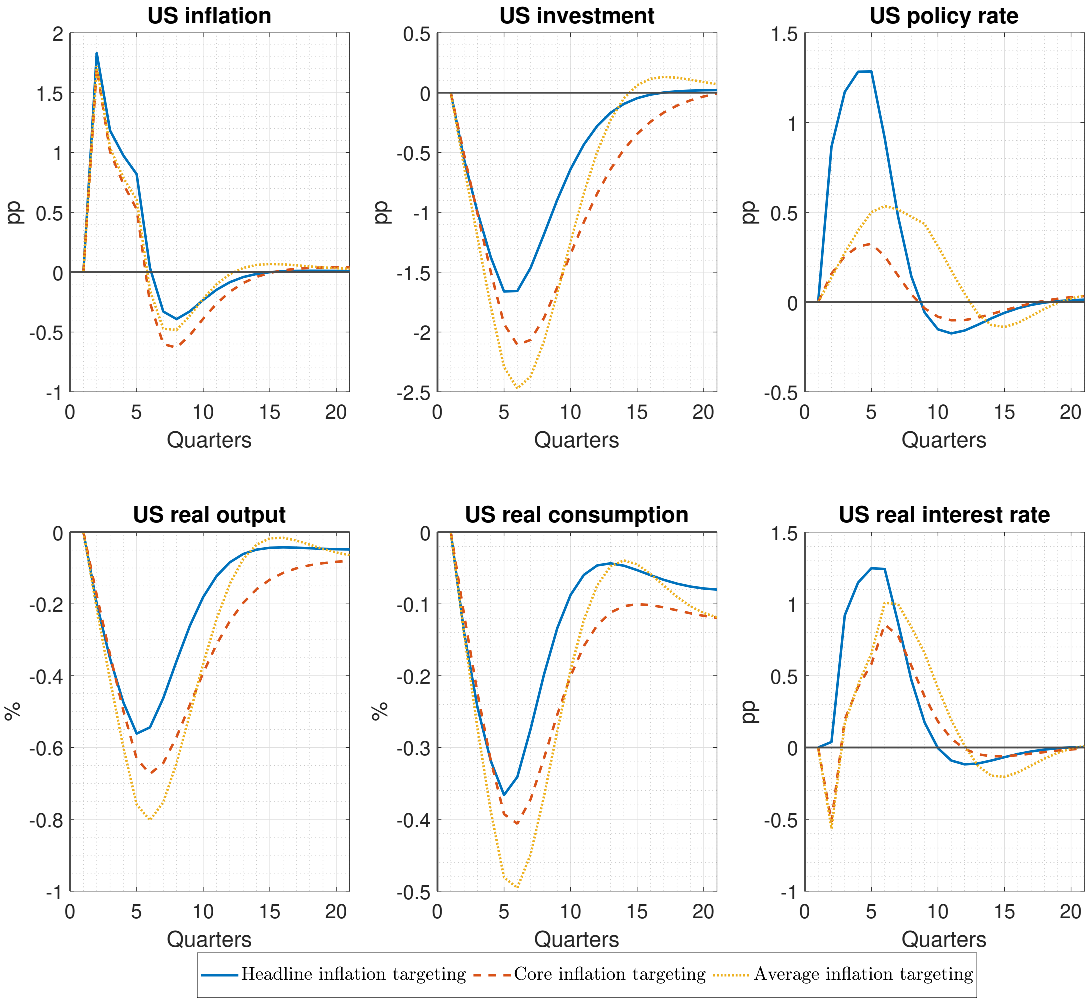
1.3.3 Exercise 3: plotting a different set of variables

The toolbox allows to plot the results of the simulations for different variables. For instance, to replace the chart showing the impulse response function for nominal wage inflation with the results for investment, the user only needs to update the corresponding entries in the cell vectors *vcell, annualisation_cell, titlecell* and *obs_unit* in lines 149-153 of the *run_temporary_shock.m* code. Specifically, the user needs to change the second entry in the cell vector *vcell* from 'pie_W' to 'invest' and the second entry in the cell vector *titlecell* from "Nominal wage inflation" to "Investment". After making these changes, the user must re-run the entire code. In this specific example the corresponding entries in the cell vectors *annualisation_cell* and *obs_units* don't require any adjustment. However, if the user wishes to plot the results for a variable with a different annualisation coefficient and/or unit, these entries must also be adjusted accordingly.

```
%% Variable to plot IRFs for
vcell = {'pie','invest','r','y_va','c','rr'};
annualisation_cell = {'400','100','400','100','100','400'};
titlecell = {'Inflation','Investment','Policy rate','Real output',...
    'Real consumption','Real interest rate'};
obs_units = {'pp','pp','pp','%','%','pp'};
```

The results are reported in Figure 4.

Figure 4: Temporary energy price shock under alternative monetary policy rules



1.4 Simulating a structural change

In this section, we simulate a sustained 25% increase in carbon-intensive energy prices (via parameter *tau_c_*), gradually implemented over a 10-year horizon, under different monetary policy rules for the United States. Specifically, the shock will hit in period 10 and will phase out over 40 periods. The toolbox allows for the simulation of this exercise under different expectation rules: perfect anticipation of the shock (ie perfect foresight), fully unanticipated shocks (ie abrupt), staggered anticipation (ie stag), and stepwise anticipation (ie step).

The main script for running the structural change (permanent shock) is the *run_permanent_shock.m* in the main folder. Similar to when running the simulation for the temporary shock, users are prompted to provide the Dynare path (if it is not already in the Matlab search path), country code and preferred data source when they run the script.

To modify the properties of the shock, the user should adjust the *run_permanent_shock.m* script before running it. *horizon* in line 48 specifies the desired number of periods over which the impulse response functions (IRFs) will be calculated. *target_change_size* in line 69 specifies the magnitude of the shock (25% change relative to the steady state value). In the subsequent block of code, the user can define the variable that the shock will target (*selected_targets*). Information on the specific shocks and the variable names can be found in the *utility/variables_exo.m* and *utility/model_2_parameters.m* scripts, respectively.

```
47    % Set the horizon of the IRFs and the number of shocks
48    horizon = 100;
49    % If numshk is larger than 1, selected_shock and selected_target must be of
50    % the same size as the number of shocks
51    numshk = 1;
```

```
60    % Select the target
61    selected_targets = {'gam_2'}; % Commodity price is target
62    for selected_target = 1:length(selected_targets)
63        eval(strcat('targetvar',num2str(selected_target), " = '", ...
64            selected_targets{selected_target},"';"));
65    end
```

```
67    % Specify the size of the change in the target variable
68    % 1 stands for doubling the steady state value
69    target_change_size = .25;
```

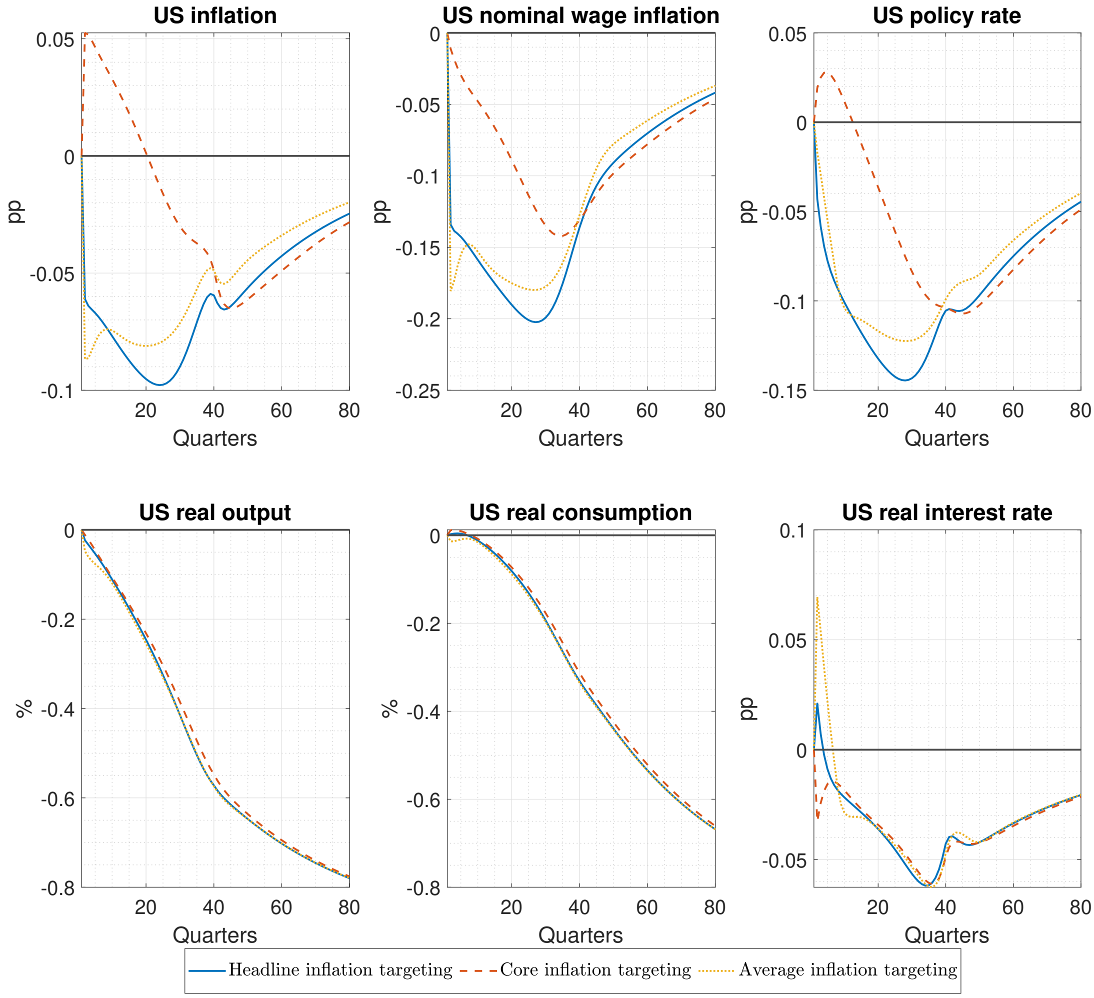
The next block of code is used to simulate the same shock under different monetary policy rules or expectation rules. The struct *exercises* contains all the different specifications that the user wants to simulate, where each "*id_*", in our example, corresponds to a specific combination monetary policy rule-expectation rule. Specifically, the struct elements for *id_2* correspond to a scenario in which the results are estimated under a standard headline inflation targeting regime (Taylor, 1993). The field *names* specifies the names of the parameters for which we want to change the associated value. As mentioned previously, the names of specific parameters can be found in the Matlab script *utility/model_2_parameters.m*. The field *names_legend* defines the legend labels that can be used in the final graph showing the IRFs produced by the code. The field *values* specifies the new values for the parameters of interest. The field *foresight* defines the expectation rule under which the scenario is simulated. The rules currently supported by the toolbox are perfect foresight (*pf*), no anticipation (*abrupt*), staggered anticipation (*stag*) and stepwise anticipation (*step*). Finally, the field *name_save* specifies the name under which the results are saved.

```
96   % Specify the exercises that you want to run
97   exercises = struct();
98   exercises.id_2 = struct();
99   exercises.id_2.id = 2; % Must be at least 2 because baseline exercise is 1
100  % Set the names of the parameters to change
101  exercises.id_2.names =        {'c';'phi_pie'};
102  exercises.id_2.names_legend = {'\c';'\phi_{\pi}'}; % LaTeX format
103  % Set the new values of the parameters to change
104  exercises.id_2.values = [0.1;1.5];
105  % Specify if perfect foresight or non-perfect foresight
106  exercises.id_2.foresight = 'pf'; % Perfect foresight = pf; ...
107  % non perfect foresight = abrupt; stepwise = step; staggered expectations = stag
108
109  % Choose the parameter
110  exercises.id_2.permanent_shock = 'tau_c_';
111  % Choose the parameter value to pass
112  exercises.id_2.permanent_shock_value = permanent_shock_value;
113  % Set the target variable
114  exercises.id_2.name_save = '_tau_c_01_it';
```

By changing the values associated with these fields, the user can specify as many scenarios as desired. For instance, if in addition to scenario *id_2* the user would like to add a simulation in which monetary policy targets core inflation instead of headline inflation, all we have to do is to define a new scenario *id_3* and update the relevant values in the fields *names* (ie *phi_pie* and *phi_pie_core*), *names_legend* (ie ϕ_π and ϕ_{core}) and *values* (ie 0 and 1.5).

The loop from lines 140 − 303 executes the various simulations and stores the corresponding results. The final section of the code includes a few lines to plot the derived impulse response functions and facilitate the comparison of the results (Figure 5).

Figure 5: Structural change under alternative monetary policy rules



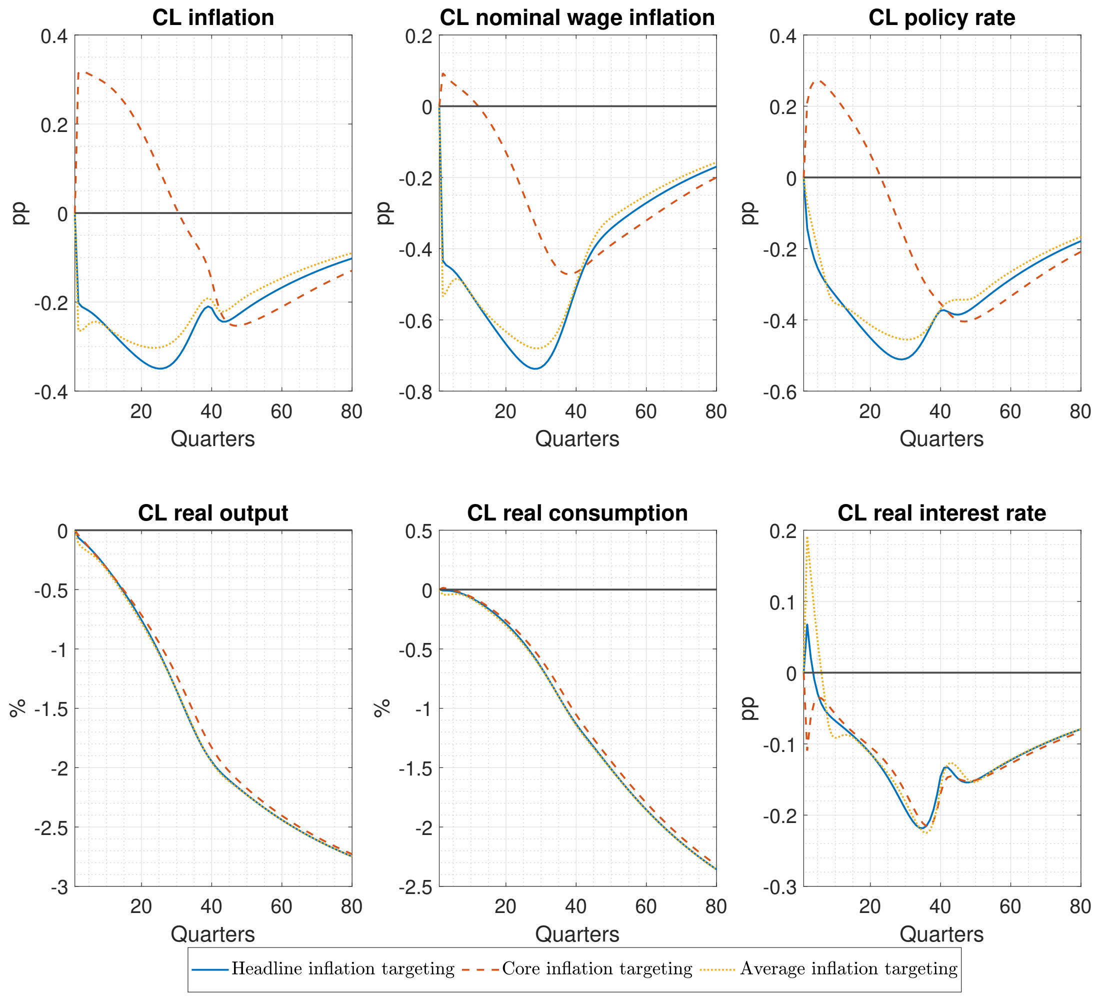
The results are stored in automatically generated country-specific sub-folders within "output\permanent_shock" and "graphs\permanent_shock" named according to the two-digit country code of the selected country.

1.4.1 Exercise 1: simulating the structural shock for a different country

In the exercise presented in the previous section we derived the impulse response functions for the US under a sustained 25% increase in carbon-intensive energy prices (via parameter *tau_c_*), gradually implemented over a 10-year horizon.

To run the same exercise for a different country, the user just needs to re-run the code and enter the two-digit ISO code of the country of their choice when prompted. The results for CL are reported in Figure 6 as an example.

Figure 6: Structural change under alternative monetary policy rules



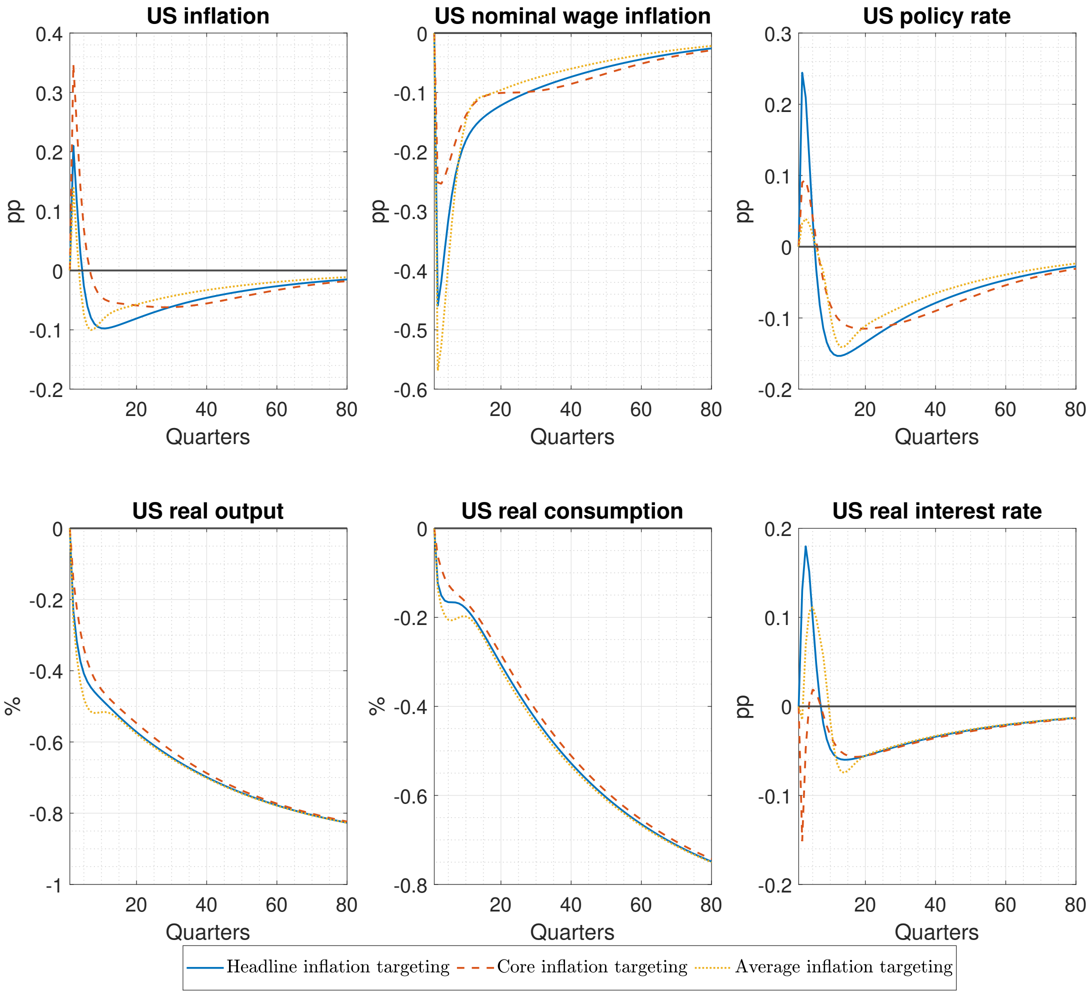
1.4.2 Exercise 2: simulating a structural shock for the US under a different expectation rule

The toolbox allows to simulate a structural shock under different expectation rules. For instance, to simulate an unanticipated sustained 25% increase in carbon-intensive energy prices (via parameter *tau_c_*), gradually implemented over a 10-year horizon, the user just needs to change the field *foresight* in the struct *exercises* in lines 106, 122 and 133 of the *run_permanent_shock.m* code. Specifically, to estimate the results, the user need change the value of the field *foresight* from 'pf', which stands for perfect foresight (ie perfectly anticipated), to 'abrupt' and re-run the whole code.

```
% Specify the exercises that you want to run
exercises = struct();
exercises.id_2 = struct();
exercises.id_2.id = 2; % Must be at least 2 because baseline exercise is 1
% Set the names of the parameters to change
exercises.id_2.names =        {'c';'phi_pie'};
exercises.id_2.names_legend = {'\c';'\phi_{\pi}'}; % LaTeX format
% Set the new values of the parameters to change
exercises.id_2.values = [0.1;1.5];
% Specify if perfect foresight or non-perfect foresight
exercises.id_2.foresight = 'abrupt'; % Perfect foresight = pf; ...
% non perfect foresight = abrupt; stepwise = step; staggered expectations = stag

% Choose the parameter
exercises.id_2.permanent_shock = 'tau_c_';
% Choose the parameter value to pass
exercises.id_2.permanent_shock_value = permanent_shock_value;
% Set the target variable
exercises.id_2.name_save = '_tau_c_01_it';

% Exercise 3
exercises.id_3 = struct();
exercises.id_3.id = 3;
exercises.id_3.names =        {'c';'phi_pie';'phi_pie_core'};
exercises.id_3.names_legend = {'\c';'\phi_{\pi}';'\phi_{core}'}; % LaTeX format
exercises.id_3.values = [0.1;0;1.5];
exercises.id_3.foresight = 'abrupt';
exercises.id_3.permanent_shock = 'tau_c_';
exercises.id_3.permanent_shock_value =  permanent_shock_value;
exercises.id_3.name_save = '_tau_c_01_core';

% Exercise 4
exercises.id_4 = struct();
exercises.id_4.id = 4;
exercises.id_4.names =        {'c';'phi_pie';'phi_pie_bar'};
exercises.id_4.names_legend = {'\c';'\phi_{\pi}';'\phi_{\bar{\pi}}'}; % LaTeX format
exercises.id_4.values = [0.1;0;1.5];
exercises.id_4.foresight = 'abrupt';
exercises.id_4.permanent_shock = 'tau_c_';
exercises.id_4.permanent_shock_value = permanent_shock_value;
exercises.id_4.name_save = '_tau_c_01_ait';
```

The results are reported in Figure 7.

Figure 7: Structural change under alternative monetary policy rules



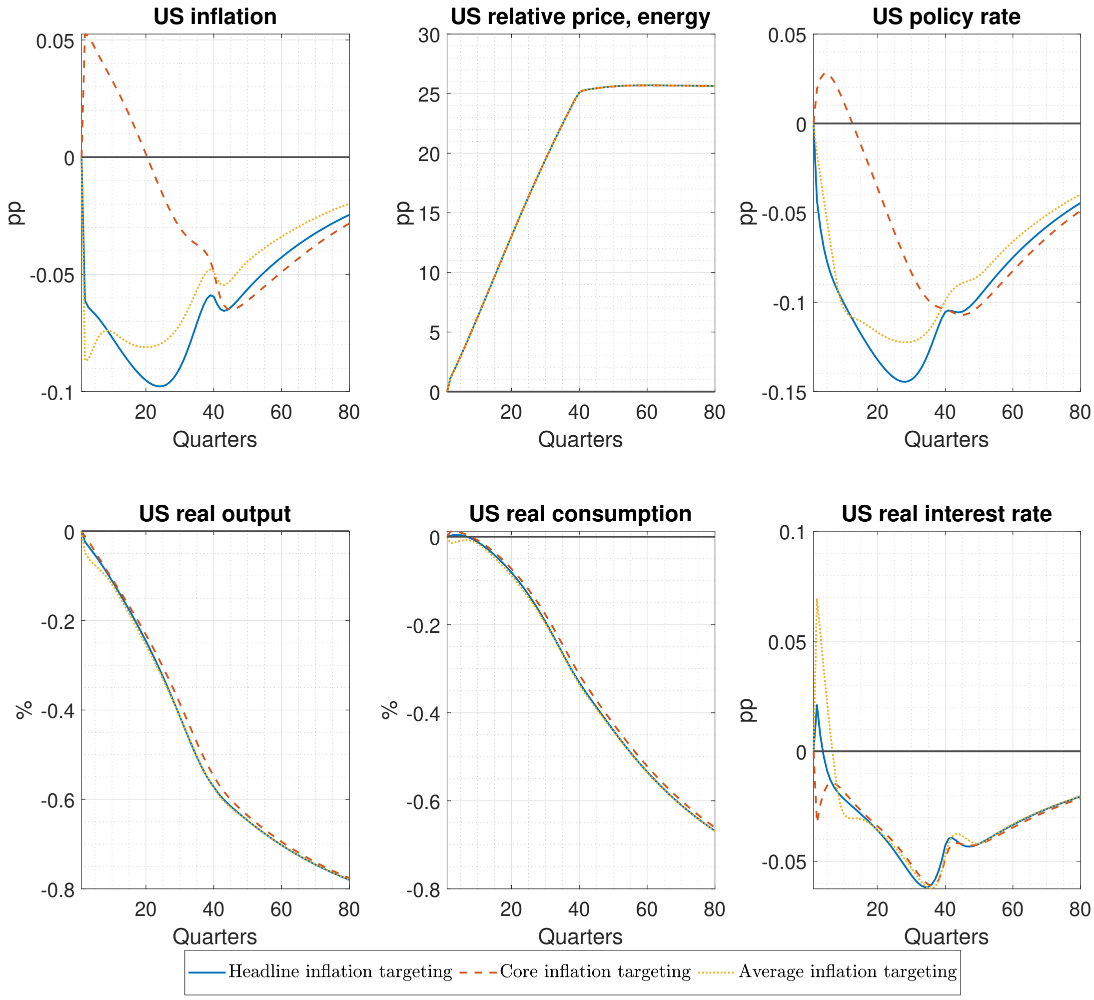
1.4.3 Exercise 3: plotting a different set of variables

The toolbox allows to plot the results of the simulations for different variables. For instance, to replace the chart showing the impulse response function for nominal wage inflation with the results for the relative price in the energy sector, the user only needs to update the corresponding entries in the cell vectors *vcell, annualisation_cell, titlecell* and *obs_unit* in lines 311 – 315 of the *run_permanent_shock.m* code. Specifically, the user needs change the second entry in the cell vector *vcell* from 'pie_W' to 'gam_2' and the second entry in the cell vector *titlecell* from "Nominal wage inflation" to "Relative price, energy". After making these changes, the user must re-run the entire code. In this specific example the corresponding entries in the cell vectors *annualisation_cell* and *obs_units* don't require any adjustment. However, if the user wishes to plot the results for a variable with a different annualisation coefficient and/or unit, these entries must also be adjusted accordingly.

```
%% Variable to plot IRFs for
vcell = {'pie','gam_2','r','y_va','c','rr'}; % b
annualisation_cell = {'400','100','400','100','100','400'};
titlecell = {'Inflation','Relative price, energy','Policy rate','Real output',...
    'Real consumption','Real interest rate'};
obs_units = {'pp','pp','pp','%','%','pp'};
```

The results are reported in Figure 8.

Figure 8: Structural change under alternative monetary policy rules



References

Burgert, Matthias, Giulio Cornelli, Burcu Erik, Benoit Mojon, Daniel Rees, and Matthias Rottner, *The BIS Multisector Model: A Multi-Country Environment for Macroeconomic Analysis*, BIS Working Papers 1297, 2025.

Taylor, John B, "Discretion versus policy rules in practice," in "Carnegie-Rochester conference series on public policy," Vol. 39 Elsevier 1993, pp. 195–214.