

Costain, Nakov and Petit (2021) - Replication codes

This repository includes all the codes required to replicate the results in the paper "**Flattening of the Phillips curve with state-dependent prices and wages**", by with James Costain (Banco de España), Anton Nakov (ECB and CEPR) and Borja Petit (CUNEF).

The model is solved it two steps. First, we use Fortran to solve for the steady state and to linearize the dynamic system. Then, we use Matlab to implement the Kelin's complex QZ decomposition solution method and to compute the Impulse Response Functions.

Structure of the codes

It is very important that you keep the same structure of folders and subfolders

List of folders

Folder	Description
../compiledfiles/	This folder contains the <code>.mod</code> files produced at compilation
../figures/	This folder contains the Matlab codes that generate the figures of the paper
../matlab/	This folder contains a number of Matlab codes that are used to implement the Kelin's decomposition, and some other auxiliary functions
../tables/	This folder contains a set of Matlab codes that produce the tables shown in the paper
../textfiles/	This folder contains text files with data and the solution to the model

List of files

Root	File	Description
../	dynamics.f90	This code generates computes the Jacobian of the dynamic system and stores it in <code>../textfiles/_dyn/Vxy_dyn.txt</code> where <code>x</code> and <code>y</code> refer to the specific version solved (see below)
	main.f90	This code contains the execution of the program
	parameters.f90	This code contain the parameters of the model as well as some general-porpose functions
	solution.f90	This code solves the firms' and workers' problems and compute the invariant distribution of prices and wages
	toolkit.f90	This code contains a set of general-porpose functions and subroutines, including optimization routines used in the code
../figures/	fig_2.m	Generate figure 2 of the paper

	fig_3.m	Generate figure 3 of the paper
	fig_4.m	Generate figure 4 of the paper
	fig_5.m	Generate figure 5 of the paper
	fig_6.m	Generate figure 6 of the paper
	fig_7.m	Generate figure 7 and table 8 of the paper
	fig_8.m	Generate figure 8 of the paper
../matlab/	solve_dyn.m	This code reads the .txt files produced with Fortran and implements the Klein QZ decomposition to generate the IRFs
	extract_dyn.m	Given a Jacobian matrix, this code fill the matrices to accomodate the problem to the Kelin's QZ decomposition algorithm
	extract_ss.m	Takes a vector with all the results from the steady-state and fill vectors and matrices
	kleinsolve.m	This code implements the Klein QZ decomposition algorithm
	parameters.m	Defines some parameters that are used in other codes
	plot_irf.m	Generate the figure with IRFs
../tables/	table_4.m	Generate Table 4 of the paper
	table_5.m	Generate Table 6 of the paper
	table_7.m	Generate Table 7 of the paper
../textfiles/	calibprams.txt	Values of the calibrated parameters
	data_pc.mat	Data used in section 4.2.3
	data_pdfprices.txt	Empirical histogram of price changes
	data_pdfwages.txt	Empirical histogram of wage changes

Compilation command

To compile the Fortran codes run the following command:

```
cd [set your own working directory]
gfortran [optional: compilation flags] -J $(pwd)/compiledfiles
toolkit.f90 parameters.f90 solution.f90 dynamics.f90 main.f90 -o lpw
```

You tend to use the following compilation flags:

```
-fopenmp -O3 -ffixed-line-length-150 -fmax-stack-var-size=1000000
```

Model versions

The different versions of the model, combining different noise parameters and inflation rates, are named according to V_{xy} where x refers to the noise parameters and y to the inflation rate. In particular:

Adjustment cost / Inflation rate	2%	-1%	0%	4%	8%	-2%	1%
Baseline	V10	V11	V12	V13	V14	V15	V16
Semi-flexible prices and sticky wages	V20	V21	V22	V23	V24	V25	V26
Flexible prices and sticky wages	V30	V31	V32	V33	V34	V35	V36
Sticky prices and semi-flexible wages	V40	V41	V42	V43	V44	V45	V46
Sticky prices and flexible wages	V50	V51	V52	V53	V54	V55	V56
Flexible prices and flexible wages	V60	V61	V62	V63	V64	V65	V66

This table includes all the versions that can be computed, which are few more than the ones shown in the paper. In particular, the versions in the second and fourth rows are not in the paper.

Beyond this, we allow the inflation rate to take two different extra values to compute the Phillips curve slope presented in the paper. In particular:

Inflation rate	
Inflation rate of 4.63% (US, 1980-2000)	V17
Inflation rate of 2.01% (US, 2000-2020)	V18

Running the codes

The code offers 7 possibilities:

1. Solve only the steady state

The user is asked to specify which version to solve

2. Solve the steady state and the dynamics

The user is asked to specify which version to solve

3. Calibrate the parameters of the model

The user is asked which algorithm to use for calibration: Nelder-Mead or a Newton-based algorithm

4. Solve for different noise parameters

The code solves the steady-state and the dynamics for all the combinations of noise parameters and the baseline inflation rate.

Version V10, V20, V30, V40, V50, and V60

5. Solve for different inflation rates

The code solves the steady-state and the dynamics for all possible inflation rates, and the baseline level of noise parameters.

Versions V10, V11, V12, V13, V14, V15 and V16.

6. Solve for all possible cases

The code solves the steady-state and the dynamics for all possible combinations of noise parameters and inflation rates.

7. Solve for pre and post inflation rates

The code solves the steady-state and the dynamics for an inflation rate of 4.63% (US, 1980-2000) and 2.01% (US, 2000-2020).

Versions V17 and V18