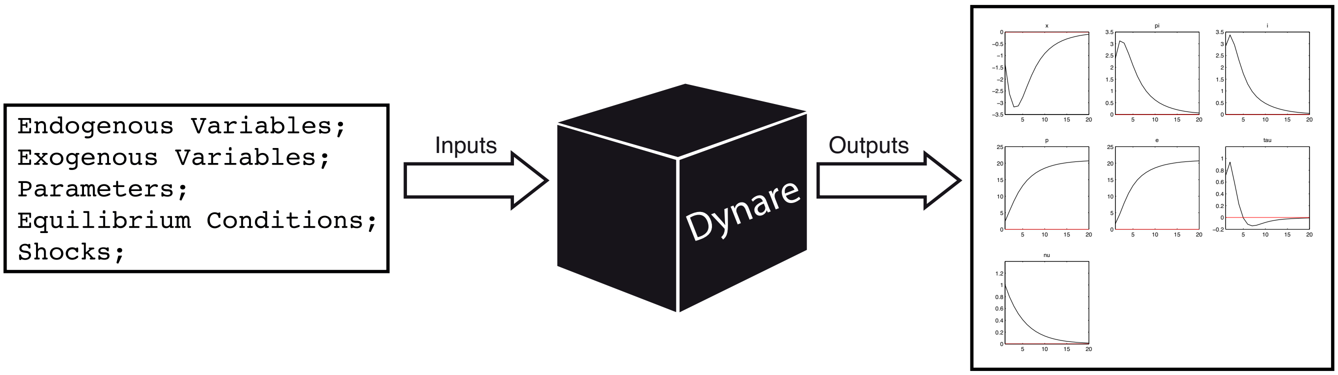
Introduction to Dynare 4.3



Getting started

To run `Dynare`, one has the choice of using `GNU Octave` or `Mathworks MATLAB`. Since MATLAB is already available on your computers we choose the latter. Therefore, you should now

- Download the latest Dynare Windows installer at www.dynare.org (which will be `Dynare 4.3.x`)
- Double-click on the Dynare installer and follow the instructions.
- Run MATLAB
- Configure MATLAB for Dynare (using MATLAB's command window):
  - Type: `addpath c:\dynare\4.3.x\matlab`
    (You will have to do this *every time* you start MATLAB, as at least older versions of MATLAB will not remember the setting.)

Now MATLAB is prepared for the use of Dynare.

Running and Editing a Dynare *.mod-File

In order to give instructions to Dynare, the user has to write a *model file* whose filename extension has to be `.mod` (instead of just `.m`). This file contains the description of the model and the computing tasks required by the user. Once the model file is written, Dynare is invoked using the `dynare`-command at the MATLAB prompt (with the *filename* of the `.mod` given as argument).

- Create a working directory that will hold your Dynare-files. For example, `c:\dynare\work`.
- Copy the example model file from `c:\dynare\4.3.x\examples\example1.mod` to your working directory. (Note that dynare-files use the extension `.mod` instead of just `.m`)
- At the MATLAB command window type the following to change the working directory of MATLAB: `cd c:\dynare\work`
- Now type the following to run the example model file: `dynare example1`
  (without any extension!)
- If you want to edit your `.mod`-file type the following: `edit example1.mod`

Solving/Simulating the Hybrid New Keynesian Model with Dynare - A Code Example

(in response to a temporary cost-push shock)

Preamble

The preamble consists of the declarations to setup the endogenous variables, the exogenous variables as well as the parameters and assigns values to these parameters.

1. First of all the *endogenous* variables of the model like output gap (x), inflation (π), nominal interest rate (i) etc. must be specified:

```
var x pi i r nu;
```

2. After the endogenous variables the *exogenous* variables (i.e. the shock ε_t) of the model must be specified:

```
varexo eps;
```

3. Now follows the list of parameters:

```
parameters phi sig gam_f gam_b gam_x del_pi del_x AR_par;
```

4. The assignment of parameter values is left. This is done the standard way in MATLAB:

```
phi    = 0.5;
sig    = 1;
gam_f  = 0.5;
gam_b  = 0.5;
gam_x  = 0.2;
del_pi = 1.5;
del_x  = 0.5;
AR_par = 0.8;
```

A Dynare model file contains a list of commands and of blocks. Each command and each element of a block is terminated by a semicolon (`;`). Blocks are terminated by `end;`. This terminations are mandatory in Dynare. To forget one or more of these terminations is one of the most common reasons for issuing error messages.

Declaration of the Model

This step starts with the instruction

```
model(linear);
```

or just with `model;` which declares the model as being non-linear.

Now all equilibrium conditions of the model are written exactly the way you would write them on a piece of paper. However, there are very simple rules to follow concerning variables:

- for variables decided in t (like x_t) write `x`
- for variables decided in $t-1$ (like x_{t-1}) write `x(-1)`
- for variables decided in $t+1$ (like x_{t+1}) write `x(+1)`

```
x  = (1-phi)*x(+1) + phi*x(-1) - 1/sig*(i-pi(+1));
pi = gam_f*pi(+1) + gam_b*pi(-1) + gam_x*x + nu;
i  = del_pi*pi + del_x*x;
r  = i - pi(+1);
nu = AR_par*nu(-1) + eps;
```

Note that there must be **as many equations as there are endogenous variables** in the model, except when computing the unconstrained optimal policy with `ramsey_policy` or `discretionary_policy`[1].
The declaration (or the block) of the model ends with

```
end;
```

[1] Dynare has tools to compute optimal policies for various types of objectives. You can either solve for *optimal policy under commitment* with “ramsey_policy”, for *optimal policy under discretion* with “discretionary_policy” or for *optimal simple rule* with “osr”.

Solving the Model

A typical experiment is to study the effects of a *temporary shock* after which the system goes back to the original equilibrium (if the model is stable...). A temporary shock is a temporary change of one or several exogenous variables of the model. Temporary shocks are specified within the next block between the commands `shocks;` and `end;`:

```
shocks;
var eps = 1;
end;
```

Another typical experiment in a deterministic context is to study the transition of one equilibrium position to another, it is equivalent to analyze the consequences of a *permanent shock* and this in done in Dynare through the proper use of `initval` and `endval`.

To get Dynare to compute the steady state values just add:

```
steady;
```

To get informations about the eigenvalues and, thus, about the stability of the model (i.e. if its specification satisfies the Blanchard-Kahn condition) just add:

```
check;
```

The model is then solved and simulated using the `stoch_simul`-command for 20 periods

```
stoch_simul(irf = 20);
```

which solves a stochastic (i.e. rational expectations) model, using perturbation techniques. More precisely, `stoch_simul;` computes Taylor approximations of the decision and transition functions for the model around the steady-state. Using them, it computes impulse response functions as the difference between the trajectory of a variable following a shock at the beginning of period 1 and its steady state value as well as various descriptive statistics (moments, variance decomposition, correlation and autocorrelation coefficients).

By default, the

- coefficients of the approximated decision rules as well as
- moments of the variables are reported and
- impulse response functions for each exogenous shock are plotted.

Note that while running the dynare code it stores the values of the endogenous variables of each period in corresponding vectors. Their names with the length of `irf = 20` have the following structure:

```
nameOfEndogenousVariable_ShockToWhichItResponses
```

which just means that since all of our endogenous variables response to the shock ε (`epsilon`) dynare names them `x_epsilon`, `pi_epsilon`, `i_epsilon` and so on (see figure 1). This is very convenient because the default dynare plots aren't very pretty so that we have the opportunity to plot the impulse-response-functions the way we want.

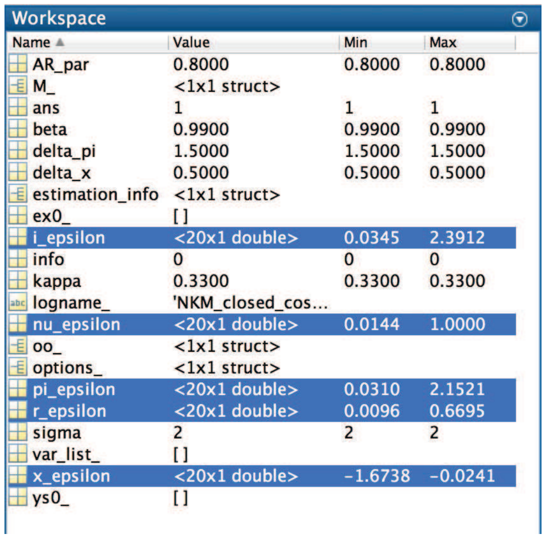


Figure 1: MATLAB's workspace after the simulation with dynare.

Most Dynare commands have arguments and several accept options, indicated in parentheses after the command keyword. Several options are separated by commas:

- `ar = integer`: Specifies the order of autocorrelation coefficients to compute and to print (default = `5`)
- `periods = integer`: If different from zero, the model will be simulated and *empirical* moments will be computed instead of *theoretical* moments. The value of the option specifies the number of periods to use in the simulations (default = `0`)
- `nocorr`: Disables the printing of the correlation matrix (default = `print`)
- `drop = integer`: Specifies the number of points dropped at the beginning of the simulation before computing the summary statistics (default = `100`)
- `irf = integer`: Specifies the number of periods on which to compute the impulse response functions (default = `40`)
- `graph_format = format`: Specifies the file format for graphs saved to disk. Possible values are `eps` (the default), `pdf` and `fig`
- `qz_criterium = double`: Value used to split stable from unstable eigenvalues in reordering the Generalized Schur decomposition used for solving 1st order problems. Default: `1.000001`
- `nofunctions`: Disables the printing of the coefficients of the approximated solution
- `nomoments`: Disables the printing of the moments of the endogenous variables
- `order = [order]`: Specifies the order of the Taylor approximation. Possible values: `1`,`2`,`3` (default = `2`)
- `replic = integer`: Specifies the number of simulated series used to compute the impulse response functions (default = `1`, if order = 1, and = `50` otherwise)

The Output

At first, Dynare starts to run the code and to generate results.

```
Configuring Dynare ...
[mex] Generalized QZ.
[mex] Sylvester equation solution.
[mex] Kronecker products.
[mex] Sparse kronecker products.
[mex] Local state space iteration (second order).
```

```
[mex] Bytecode evaluation.
[mex] k-order perturbation solver.
[mex] k-order solution simulation.
[mex] Quasi Monte-Carlo sequence (Sobol).

Starting Dynare (version 4.3.0).
Starting preprocessing of the model file ...
Found 5 equation(s).
Evaluating expressions...done
Computing static model derivatives:
 - order 1
Computing dynamic model derivatives:
 - order 1
 - order 2
Processing outputs ...done
Preprocessing completed.
Starting MATLAB/Octave computing.
```

Due to the `steady;`-command the steady-states of the endogenous variables are computed. Since the a temporary cost-push shock in a stables model is considered all values are equal to zero.

```
STEADY-STATE RESULTS:

x    0
pi   0
i    0
r    0
nu   0
```

Since we have used `check;` the eigenvalues of the system and a confirmation of the Blanchard-Kahn condition is displayed. Note that the eigenvalues are given in ascending order.

```
EIGENVALUES:

Modulus             Real          Imaginary
0.4472              0.4              0.2
0.4472              0.4             -0.2
0.8                 0.8              0
1.382               1.382            0
3.618               3.618            0

There are 2 eigenvalue(s) larger than 1 in modulus
for 2 forward-looking variable(s)

The rank condition is verified.
```

The model summary just counts the various variable types of the model:

```
MODEL SUMMARY

Number of variables:          5
Number of stochastic shocks: 1
Number of state variables:   3
Number of jumpers:           2
Number of static variables:  2
```

We introduced only one shock, therefore, a scalar instead of a matrix is displayed:

```
MATRIX OF COVARIANCE OF EXOGENOUS SHOCKS
```

```
Variables      eps
eps         1.000000
```

Finally, various descriptive statistics are displayed:

```
POLICY AND TRANSITION FUNCTIONS

                x            pi            i             r            nu
nu(-1)     -0.878049     1.658537     2.048780     -0.019512     0.800000
x(-1)       0.300000     0.100000     0.300000      0.220000            0
pi(-1)     -0.500000     0.500000     0.500000      0.300000            0
eps        -1.097561     2.073171     2.560976     -0.024390     1.000000


THEORETICAL MOMENTS

VARIABLE        MEAN    STD. DEV.   VARIANCE
x              0.0000     5.8269    33.9528
pi             0.0000     5.2389    27.4463
i              0.0000     5.1268    26.2844
r              0.0000     0.4695     0.2204
nu             0.0000     1.6667     2.7778


MATRIX OF CORRELATIONS

Variables       x       pi        i        r       nu
x            1.0000 -0.9600 -0.9032 -0.6135 -0.8720
pi          -0.9600  1.0000  0.9873  0.6824  0.9699
i           -0.9032  0.9873  1.0000  0.6973  0.9911
r           -0.6135  0.6824  0.6973  1.0000  0.6061
nu          -0.8720  0.9699  0.9911  0.6061  1.0000


COEFFICIENTS OF AUTOCORRELATION

Order      1       2       3       4       5
x        0.9506  0.8330  0.6943  0.5633  0.4513
pi       0.9050  0.7528  0.6043  0.4793  0.3797
i        0.8566  0.6676  0.5086  0.3901  0.3037
r        0.5672  0.1950 -0.0045 -0.0802 -0.0934
nu       0.8000  0.6400  0.5120  0.4096  0.3277

Total computing time : 0h00m05s
```

Dynare Exercise 1
Solving the New Keynesian Model for a Small Open Economy

(in response to a temporary cost-push shock)

You are asked to write a `*.mod`-file containing the missing Dynare-Code which simulates the New Keynesian Model for a small open economy in response to a temporary cost-push shock and, thus, produces the impulse response functions[2] shown below given the following parameter set:

$$\sigma = 2, c = 0.2, \beta = 0.99, \kappa = 0.3433, \delta_\pi = 1.5, \delta_x = 0.125, \varphi^\nu = 0.8$$

<u>Hint:</u> You will have to specify 9 equations (including the shock process) to end up with a closed equation system.

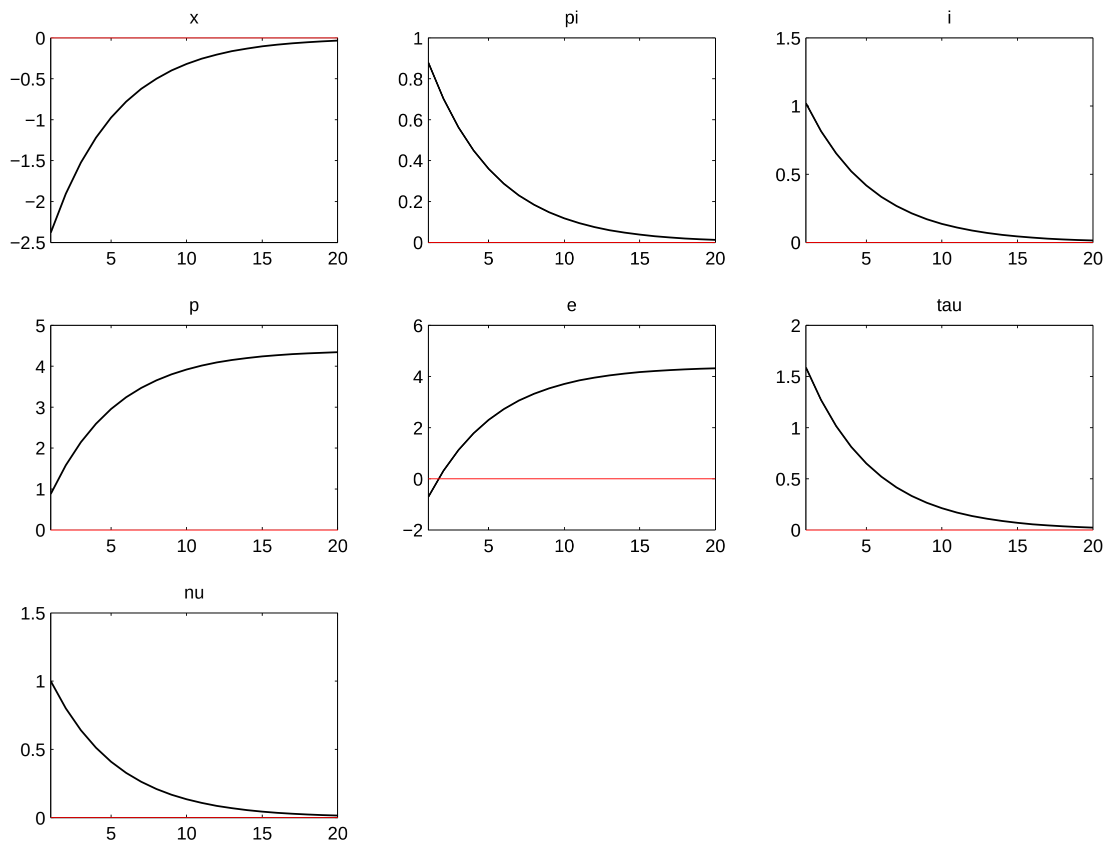


Figure 2: Adjustment time paths in response to a temporary cost-push shock shock with $\varphi^\nu = 0.8$.

[2]The IRFs are computed for the first 20 periods.

Dynare Exercise 2
Solving the Hybrid New Keynesian Model for a Small Open Economy

(in response to a temporary cost-push shock)

You are asked to write a `*.mod`-file containing the missing Dynare-Code which simulates the Hybrid New Keynesian Model for a small open economy in response to a temporary cost-push shock and, thus, produces the impulse response functions[3] shown below given the following parameter set:

$$\phi = 0.5, \sigma = 2, c = 0.2, \lambda = 0.05, \gamma_f = 0.5, \gamma_b = 0.5, \gamma_x = 0.2, \delta_\pi = 1.5, \delta_x = 0.5, \varphi^\nu = 0.8$$

Hint: You will have to specify 9 equations (including the shock process) to end up with a closed equation system.

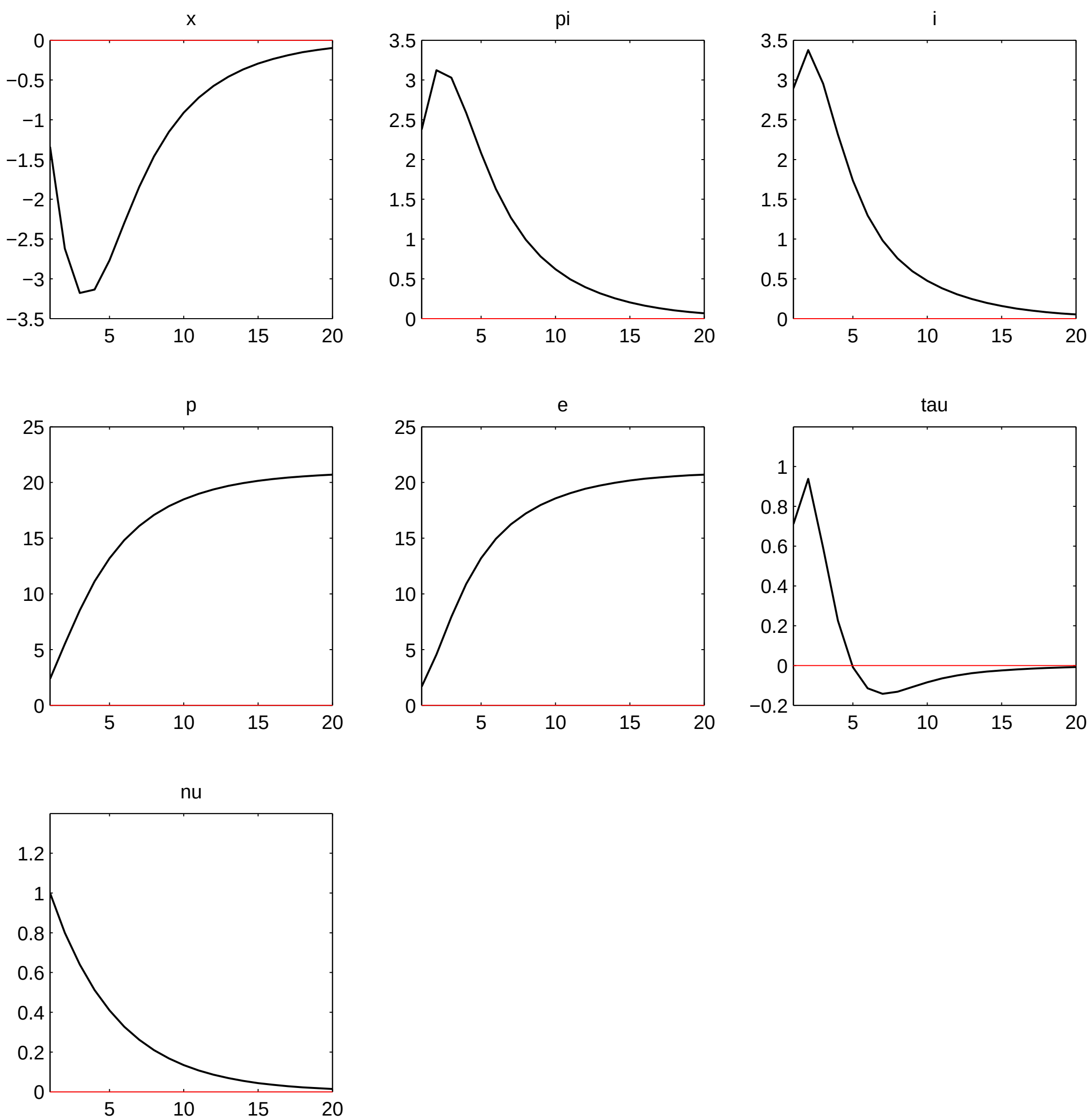


Figure 3: Adjustment time paths in response to a temporary cost-push shock shock with $\varphi^\nu = 0.8$.

[3] The IRFs are computed for the first 20 periods.