

Shock Elasticities in Dynare

User's Manual

Toolbox version 0.2*

Jaroslav Borovička
University of Chicago
Federal Reserve Bank of Chicago
borovicka@uchicago.edu

Lars Peter Hansen
University of Chicago
National Bureau of Economic Research
lhansen@uchicago.edu

December 8, 2011

Abstract

We provide a ready-to-use set of functions for Dynare/Dynare++ that calculates approximate sensitivities of asset prices and payoffs to structural shocks in DSGE models. The method is based on a quadratic approximation of *shock elasticities* introduced in [Hansen and Scheinkman \(2010\)](#), [Borovička, Hansen, Hendricks, and Scheinkman \(2011\)](#), and [Borovička and Hansen \(2011\)](#) that can be fully embedded in Dynare/Dynare++.

*We thank Ondra Kamenik, Michel Juillard and Junghoon Lee for useful discussions. All errors remain our own. The views expressed herein are those of the authors and not necessarily those of the Federal Reserve Bank of Chicago or the Federal Reserve System.

Contents

1	Introduction	1
2	Quick-start guide	1
3	Installation	2
4	Implementation	3
4.1	Setting up the model file	3
4.1.1	An example	3
4.2	Running the elasticity calculation	4
4.3	Parameter file	5
4.4	Plotting graphs	6
4.4.1	Plotting graphs manually	7
4.5	Output file	7
5	What the elasticity toolbox does	9
5.1	Dynamics	9
5.2	Shock elasticities	10
5.3	Entropy decomposition	10
6	Function reference	10
6.1	Functions	11
6.1.1	dynare_elasticity	11
6.1.2	plot_elast_graphs	12
6.2	Parameter file	13

1 Introduction

The elasticity toolbox implements the calculation of shock-price and shock-exposure elasticities constructed in [Borovička and Hansen \(2011\)](#). The toolbox integrates seamlessly with Dynare and Dynare++, and generates the shock elasticities from the second-order approximation of the model solution.

The shock elasticities measure the sensitivity of expected returns and expected cash flows to the exposure to structural shocks. The economic theory underlying the construction in continuous time can be found in [Hansen and Scheinkman \(2010\)](#) and [Borovička, Hansen, Hendricks, and Scheinkman \(2011\)](#). [Borovička and Hansen \(2011\)](#) construct discrete-time counterparts and specifically focus on the second-order approximation utilized in the elasticity toolbox.

The underlying idea is to develop *valuation accounting methods* that decompose expected returns associated with cash-flow streams into the contributions of individual structural shocks across different maturities of the payoffs from the cash-flow stream. Consider a single cash flow (e.g., a zero coupon bond, a consumption strip in the language of [Lettau and Wachter \(2007\)](#), or a dividend strip in [Binsbergen, Brandt, and Koijen \(2011\)](#)) maturing t periods ahead. The shock-exposure elasticity is a function of the current state and the maturity of the cash flow t that measures how the expected cash flow is exposed to a shock today. Similarly, the shock-price elasticity captures how the expected return on this cash-flow changes in response to this shock. [Borovička, Hansen, Hendricks, and Scheinkman \(2011\)](#) and [Borovička and Hansen \(2011\)](#) show how to decompose expected returns into the shock elasticities associated with shocks at different horizons. Consequently, shock elasticities can be viewed as fundamental building blocks of asset pricing moments.

The toolbox also constructs decompositions of entropy measures analyzed in [Alvarez and Jermann \(2005\)](#) and [Backus, Chernov, and Zin \(2011\)](#). It turns out that shock elasticities and entropy decompositions share a common mathematical structure which we exploit.

In conjunction with Dynare, the construction of shock elasticities yields several other benefits. First, shock-exposure elasticities can be interpreted as nonlinear versions of impulse response functions. Dynare and Dynare++ construct impulse responses in a nonlinear model by simulating it. The shock-exposure elasticities constructed using the elasticity toolbox are analytical, and thus they can be evaluated at an arbitrary point of the state space without a need to resimulate the model.

Second, the elasticity toolbox produces a term structure of elasticities across many horizons without the need to specify large numbers of variables in Dynare. Every cash-flow stream or stochastic discount factor is represented by a single variable that defines the one-period increment in the logarithm of that object. The elasticity toolbox then builds up the term structure iteratively from the one-period increments.

This version of the elasticity toolbox was tested with Dynare version 4.2.1, and Dynare++ version 1.3.7.

2 Quick-start guide

This section provides a quick recipe for a reasonably experienced Dynare user on how to start using the elasticity toolbox as quickly as possible.

1. Download the Matlab files for the elasticity toolbox.¹ Unpack them into a folder that is on Matlab's path.
2. Include stochastic discount factor (prefixed `sdf_`) and cash flow (prefixed `cf_`) variables into your Dynare model file (`modelname.mod`). These variables have the form of one-period logarithmic growth rates. The extended model file may, for example, include the one-period increment in the logarithm of the stochastic discount factor of a recursive-utility representative household, and the increments in the logarithms of the cash flows associated with aggregate consumption and investment (see the example file `example_RBC.mod` in the `examples` directory). In the code below, `c` stands for consumption, `v` for the continuation value of a representative household, `i` for investment, and `da` for the permanent shock which is used to scale model variables in order to obtain a stationary solution.

¹The current version is available at <http://home.uchicago.edu/~borovicka/software.html>.

```

...
var sdf_household cf_consumption cf_investment;
...
model;
...
sdf_household = log(BETA) - GAMMA*(c - c(-1) + da) + (RHO-GAMMA)*(v+da - ev(-1));
cf_consumption = c - c(-1) + da;
cf_investment = i - i(-1) + da;
...
end;
...

```

3. Run Dynare or Dynare++ on the model file. Preferably, calculate the second-order approximation. By default, Dynare stores the results in the file `modelname_results.mat`, and Dynare++ in `modelname.mat`. In the following, the latter is used.
4. Adjust the model parameters, using one of the three following options. **In every case, you should set the model frequency** in `model_periodlength` and `model_periodname` properly to calculate the right scaling of the shock elasticities. Shock elasticities and entropy decompositions are always annualized.
 - (a) Edit directly the default file `default_elast_parameters.m`.
 - (b) Copy the default model parameter file to `modelname_elast_parameters.m`, and edit that.
 - (c) Name the parameter file arbitrarily, and call it using a second parameter when running the elasticity calculation.

The code attempts to load every parameter file from the list above, in the given order, so that latter parameter settings overwrite the prior ones.

5. Run the elasticity calculation. File names are entered without extensions. The second parameter can be omitted. The code below opens the file `modelname.mat` with the model solution generated by Dynare, reads parameter values from the three parameter files described above (the last being `param_filename.m`), and produces shock elasticities which are stored in the `elast` structure and in the file `modelname_elasticity.mat`.

```

elast = dynare_elasticity('modelname', 'param_filename');

```

6. Shock-price and shock-exposure elasticity plots are produced by default. To inspect the results, check the `elast` structure, or the file `modelname_elasticity.mat`.

The following sections in the manual describe the implementation in more detail.

3 Installation

1. Download the latest version of the elasticity toolbox.²
2. Unpack the zip file into your preferred directory, say `c:\elast_toolbox\`.
3. Add the directory to your Matlab. In Matlab, either use the menu File → Set path..., or type:

```

addpath('c:/elast_toolbox/');

```

4. Test the toolbox. Go to the `examples` directory (`c:/elast_toolbox/examples/`), and run the following code. This should produce graphs with shock-price and shock-exposure elasticities for the two shocks in the model (in this example, no output file is generated but the graphs will be stored by default in `c:/elast_toolbox/examples/graphs/`).

²<http://home.uchicago.edu/~borovicka/software.html>.

```

dynare('example_RBC.mod');
dynare_elasticity;

```

4 Implementation

4.1 Setting up the model file

The fundamental building blocks of the elasticity calculations are increments in additive functionals — logarithms of cash flows and stochastic discount factors. These need to be defined in the Dynare model file. The name of a variable that denotes the increment in the logarithm of the cash flow must contain the `cf_` prefix, while a variable that denotes the increment in the logarithm of the stochastic discount factor must contain the `sdf_` prefix.

The elasticity toolbox processes the model solution calculated in Dynare or Dynare++, and extracts variables prefixed with `cf_` and `sdf_`. Shock-exposure elasticities are calculated for every cash-flow variable, while shock-price elasticities are calculated for every combination of a stochastic discount factor variable and a cash-flow variable.

4.1.1 An example

Consider a prototypical real business cycle model, driven by permanent technology shocks with a small predictable component in the growth rate of technology. The model file `example_RBC.mod` provided with the toolbox defines the Dynare version of the model. The problem is to maximize the continuation value of a representative household given by the recursion

$$V_t = \left[(1 - \beta) C_t^{1-\rho} + \beta E_t \left[V_{t+1}^{1-\gamma} \right]^{\frac{1-\rho}{1-\gamma}} \right]^{\frac{1}{1-\rho}}$$

where γ represents the risk aversion toward intratemporal gambles, while ρ is the intertemporal elasticity of substitution, subject to the feasibility constraint

$$Y_t \equiv A_t^{1-\alpha} K_{t-1}^\alpha = C_t + I_t$$

and the law of motion for capital

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

The level of technology A is specified as

$$da_{t+1} = \log A_{t+1} - \log A_t = \mu + x_t + \sigma_a \varepsilon_{a,t+1}$$

where x is a stationary process

$$x_{t+1} = \rho x_t + \sigma_x \varepsilon_{x,t+1}.$$

Notice that the shocks in the model have permanent components, which makes the model nonstationary. In the resulting Dynare code, variables will be scaled by the level of technology.

The stochastic discount factor of the representative household is given by

$$\frac{S_{t+1}}{S_t} = \beta \left(\frac{C_{t+1}}{C_t} \right)^{-\rho} \left(\frac{V_{t+1}}{E_t \left[V_{t+1}^{1-\gamma} \right]^{\frac{1}{1-\gamma}}} \right)^{\rho-\gamma}$$

and the return on capital is

$$R_t = \alpha \left(\frac{K_{t-1}}{A_t} \right)^{\alpha-1} + 1 - \delta,$$

which allows to write the Euler equation as

$$1 = E_t \left[\frac{S_{t+1}}{S_t} R_{t+1} \right].$$

To produce a stationary representation of the model, all endogenous variables Z should be scaled by A . If we are interested in a logarithmic expansion, we can define logarithms of the scaled quantities as

$$z_t = \log \frac{Z_t}{A_t}.$$

Increments of variables, however, are stationary even in the nonstationary version of the model. We can take advantage of this fact and define the stochastic discount factor functional and two cash-flow functionals as

$$\begin{aligned} \text{sdf_household}_t &= \log S_t - \log S_{t-1} = \log \beta - \rho(c_t - c_{t-1} + da_t) + \\ &\quad + (\rho - \gamma) \left(v_t + da_t - \frac{1}{1 - \gamma} \log E_{t-1} [\exp((1 - \gamma)(v_t + da_t))] \right) \\ \text{cf_consumption}_t &= \log C_t - \log C_{t-1} = c_t - c_{t-1} + da_t \\ \text{cf_investment}_t &= \log I_t - \log I_{t-1} = i_t - i_{t-1} + da_t \end{aligned}$$

Why is this advantageous? Impulse response functions constructed by Dynare or Dynare++ correspond to responses of the scaled versions of the variables, and thus need to be readjusted by adding back the response of the shock. On the other hand, the shock elasticities constructed by the elasticity toolbox represent responses of the original, unscaled variables. In this way, the elasticity toolbox can directly produce the correct shock elasticities with respect to permanent shocks, which are often important in asset pricing applications. Naturally, the stationary counterparts of the shock elasticities can be produced as well by defining stationary versions of the stochastic discount factor and cash flows.

The following code is a fragment from the model file `example_RBC.mod`, available in the elasticity toolbox.

```
...
var sdf_household cf_consumption cf_investment;
...
model;
...
sdf_household = log(BETA) - GAMMA*(c - c(-1) + da) + (RHO-GAMMA)*(v+da - ev(-1));
cf_consumption = c - c(-1) + da;
cf_investment = i - i(-1) + da;
...
end;
...
```

4.2 Running the elasticity calculation

The core function providing the user interface is `dynare_elasticity`. This function retrieves the solution of a model calculated by Dynare or Dynare++, processes it, and calculates the shock-price and shock-exposure elasticities for objects specified in the model file.

Important: In order to generate correctly annualized shock elasticities, you need to **specify the correct frequency** at which the parameter values are defined (the default is **quarterly**). The frequency can be set by modifying the parameter file. Annualization implies that shock elasticities are divided by the square root of period length, and entropies by the period length. See Section 4.3 for details.

If you use Dynare++, you would typically run (for a generic model filename `modelname.mod`):

```
system('dynare++ modelname.mod'); // or run dynare++ from the command line
elast = dynare_elasticity('modelname'); // Dynare++ produces an output file called modelname.mat
```

Remember you need to set the path to your Dynare++ executable file properly. The elasticity toolbox processes the output from Dynare++, plots the elasticity graphs (unless switched off in the parameter file, see Section 4.3), and returns the `elast` structure that contains the results (see Section 4.5). The `elast` structure is also stored in the file `modelname_elasticity.mat`.

If you use Dynare, you can use two alternatives. If you invoke the elasticity calculation right after running Dynare, the elasticity toolbox can read the variables directly from memory. For example:

```
dynare('modelname.mod');
elast = dynare_elasticity;
```

In this way, the results are not stored in a file (the toolbox does not know the filename) but you can store the `elast` structure manually into a file for later use.

You can also provide a filename as an input parameter. Dynare typically stores the output in a Matlab data file named `modelname_results.mat`. If you wish to load the Dynare model solution later, you can use:

```
dynare('modelname.mod');

% some code here
system('ren modelname_results.mat modelname.mat'); // or use mv on Unix machines
elast = dynare_elasticity('modelname');
```

Renaming the file with the Dynare model solution is not necessary. The only consequence of not changing the file name is that `modelname` will subsequently be replaced by `modelname_results`, and the elasticities will be stored in `modelname_results_elasticity.mat`.

The elasticity toolbox automatically detects if the supplied file with the model solution was produced by Dynare (it searches for variables named `oo_`, `M_`, and `options_`), or by Dynare++ (any other case). There is no extensive checking that the files have the right structure.

The structure of the stored output is described in Section 4.5.

4.3 Parameter file

All parameters that determine what is calculated and displayed are stored in the parameter file. The parameter file contains essential parameter settings that include model frequency, choice of elasticity calculations, and model adjustments.

The parameter file is a piece of Matlab code that defines fields in the `param` structure. Default parameter values are stored in the file `default_elast_parameters.m`. This file also contains a detailed description of all settings. Further details are provided in Section 6.2.

For every model with Dynare solution stored in, say, `modelname.mat`, Dynare additionally searches for the file `modelname_elast_parameters.m`. This file should contain any user-specific settings. One way to produce it is to copy the default parameter file and modify the parameters. **At least the model frequency must be set correctly** (the default is quarterly), so you will always need to set something up manually.

Finally, you can also provide a user-specific parameter filename as a second argument when running `dynare_elasticity`. You typically want to do this if you have multiple variants of the model file but want to apply the same user-specific setting to all of them, or, alternatively, one model file that should be evaluated using different settings.

The following code requests to retrieve parameters from `param_filename.m`.

```
dynare_elasticity('modelname','param_filename');
```

The elasticity toolbox loads all files sequentially, using values defined in later files to override values from previous files. To summarize, the sequence is:

1. `default_elast_parameters.m`.
2. `modelname_elast_parameters.m`
3. `param_filename.m`, where `param_filename` is the second (non-mandatory) argument specified when calling the `dynare_elasticity` function.

4.4 Plotting graphs

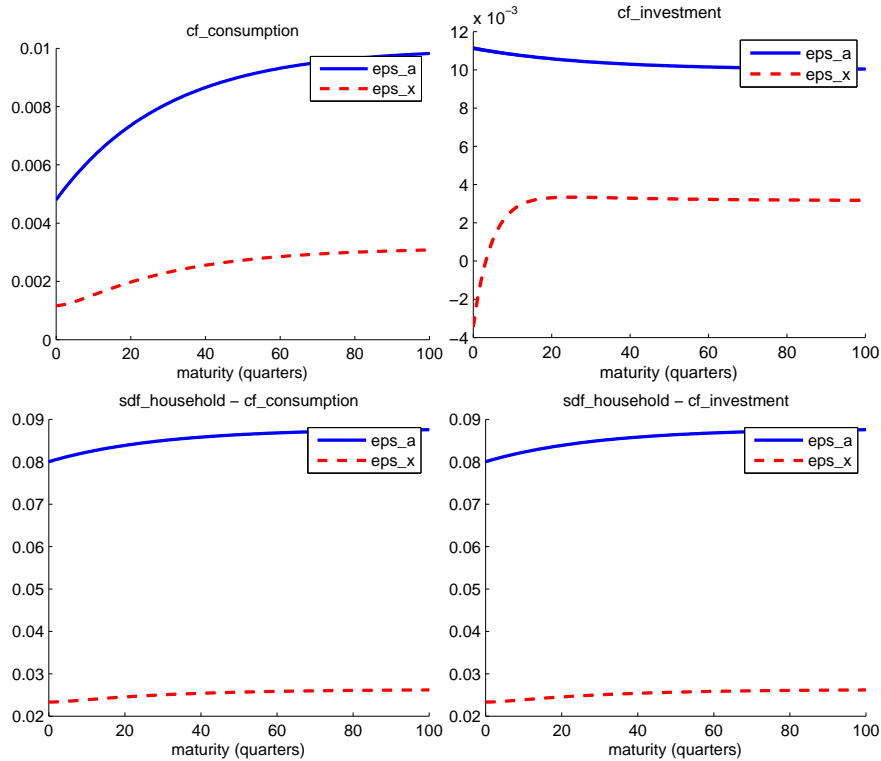


Figure 1: Shock-exposure elasticities (top row) and shock-price elasticities (bottom row) for the model defined in `example_RBC.mod`.

The code plots the shock elasticities by default. Shock elasticities are state-dependent (if the model is solved to the second order), and thus an appropriate evaluation point needs to be chosen. By default, the code plots the shock elasticity at the deterministic steady state, and simulates the stationary distribution of the shock elasticities to plot quartiles of the distribution. Other options (e.g., evaluation at the fixed point of the decision rule, or other quantiles), and more settings for plotting can be chosen in the parameter file.

If you change plotting parameters, the model does not need to be resolved, it is sufficient to call the `plot_elast_graphs` function with the elasticity results file as a parameter:

```
plot_elast_results('modelname_elasticity');
```

Figure 1 plots the shock-exposure and shock-price elasticities for the `example_RBC` model. Each panel shows the shock elasticities for one particular cash flow (in the case of shock-exposure elasticities in the top row), or one particular combination of a cash flow and a stochastic discount factor (in the case of shock-price elasticities in the bottom row). There are two curves in the graph, each corresponding to one shock in the model. Although the code plots quantiles of the distribution of the shock elasticities, the state dependence of the shock elasticities in this model is extremely limited, so that the whole distribution appears as a single line.

In specific cases, it may be desirable to call the plotting function with a different set of parameters. In that case, you can supply `plot_elast_results` with a different parameter file name as a second parameter.

```
plot_elast_results('modelname_elasticity', 'param_filename');
```

This will call the parameter file `param_filename.m`, and update parameters (obviously, it only make sense to update parameters for plotting). In the parameter file, you can, among other things, set:

1. Points of the state space (deterministic steady state, fixed point of the nonlinear rule) at which the elasticities should be evaluated.
2. Quantiles of the simulated distributions of elasticities.
3. Horizons, grouping, and other layout features.

4.4.1 Plotting graphs manually

Graphs can be plotted manually as well, using the output stored in the `elast` structure (see Section 4.5 for how the elasticities are stored). The following Matlab code fragment plots the shock-price elasticity of cash flow k priced using stochastic discount factor j for shock m , evaluated at the fixed point of the decision rule:

```
load('modelname_elasticity.mat'); % results stored in elast
j = 1; % stochastic discount factor index
k = 1; % cash flow index
m = 1; % shock index
disp(sprintf('SDF: %s, cash flow: %s, shock: %s', ...
    elast.Eprice(j,k).sdf_name, elast.Eprice(j,k).cf_name, elast.model.shock_names(m)));
figure;
plot(squeeze(elast.Eprice(j,k).XHBAR(m, :, :)));
```

Similarly, you can plot elasticities corresponding to quantiles of the elasticity distribution, specified in `param.simul_quantiles` in the parameter file. For instance:

```
plot(squeeze(elast.Eprice(j,k).simul_quantiles(m,n, :)));
```

plots the n -th of the list of quantiles in `param.simul_quantiles` (j, k, m are as previously).

4.5 Output file

Shock elasticities are stored in a data structure named `elast`, and are returned in two ways.

```
elast = dynare_elasticity('modelname');
```

1. As a return argument of the `dynare_elasticity` function (if specified).
2. In a file named `modelname_elasticity.mat`. If no model name is specified (when `dynare_elasticity` is called immediately after solving the model using Dynare, and retrieves the model solution from the memory), then no output file is saved. The output file contains the same `elast` structure as the return argument of the function.

The `elast` structure contains the following fields:

- `param` - The list of all parameters used in the model. See the file `default_elast_parameters.m` and Section 6.2 for detailed explanations.
 - `model_original` - The model solution, generated by Dynare/Dynare++. The notation corresponds to Dynare++.
- `X` - the dynamics of the law of motion for the economy

$$\begin{aligned} Z_{t+1} &= A_0 + A_1 [X'_t W'_{t+1}]' + A_2 [X'_t W'_{t+1}]' \otimes [X'_t W'_{t+1}]' = \\ &= A_0 + A_x X_t + A_w W_{t+1} + A_{xx} (X_t \otimes X_t) + 2A_{xw} (X_t \otimes W_{t+1}) + A_{ww} (W_{t+1} \otimes W_{t+1}) \end{aligned} \quad (1)$$

where Z_{t+1} contains both state variables (at indices given by `state_vars`), and other observed variables.

The `X` field also contains specific values of `X` (expansion points, steady state, etc.), see the description of `elast.X` below. The covariance matrix `SIGMA2` must be an identity matrix.

- `expansion-order` - order of approximation
 - `state-vars` - vector of indices of the variable vector X that contain state variables
 - `var_names` - names of all variables in X . `var_names_tex` contains the names of variables in \TeX format, if model is solved by Dynare.
 - `shock_names` - names of all shocks in W . `shock_names_tex` contains the names of variables in \TeX format, if model is solved by Dynare.
- `model` - The model solution to the second order, in recursive-linear form generated by the elasticity toolbox. This specification of the model dynamics doubles the number of state variables. The structure of the data is the same as for `model`.
 - `X` - the dynamics of the law of motion for the economy, reduced to only state variables. The structure contains both the matrices for the Dynare++ notation (1), as well as the matrices for the notation (3) used in [Borovička and Hansen \(2011\)](#).
 - `A..` - coefficient matrices for the notation (1)
 - `THETA.`, `LAMBDA.` - coefficient matrices for the notation (3)
 - `XSTEADY` - deterministic steady state of the model
 - `XEXP` - the point around which the expansion is expressed (this can be different from the deterministic steady state in Dynare++ when centralization of the law of motion is used (which is the default))
 - `XMEAN` - mean of the stationary distribution obtained from Dynare. Not recommended to be used. Full higher-order approximations typically do not have stationary distributions because the dynamics are only locally stable. In ill-behaving models, `XMEAN` does not need to provide meaningful information.
 - `XHSTEADY` - deterministic steady state, as a deviation from `XEXP`.
 - `XHMEAN` - mean of the stationary distribution, as a deviation from `XEXP`.
 - `XHBAR` - ‘fix point’ of the deterministic rule, as a deviation from `XEXP`. This fix point is obtained by iterating on (1) while setting all shocks W_t to zero. For a first-order expansion, `XHBAR` is zero by construction. `XHBAR` will also be zero when the model is solved using Dynare++ to the second order with the `--centralize` option activated (default setting) because the centralization is deliberately chosen so as to zero out `XHBAR`.
 - `SIGMA` - a matrix with shock volatilities on the diagonal. Must be an identity matrix.
 - `Ecf` - vector of cash flow specifications. The length of the vector corresponds to the number of `cf_` variables in the model file. Every cash flow is specified as a function $H(x, w)$ that represents the one-period increments in the logarithm of the cash-flow

$$\begin{aligned}
Y_{t+1} - Y_t = H(X_t, W_{t+1}) = & \Gamma_0 + \Gamma_1 X_{1,t} + \Gamma_2 X_{2,t} + \Gamma_3 (X_{1,t} \otimes X_{1,t}) \\
& + \Psi_0 W_{t+1} + \Psi_1 (X_{1,t} \otimes W_{t+1}) + \Psi_2 (W_{t+1} \otimes W_{t+1}).
\end{aligned} \tag{2}$$

The coefficient matrices for one cash flow correspond to one observational equation from the law of motion (1).

- `GAMMA.`, `PSI.` - coefficient matrices (row vectors).
- `Esd` - vector of stochastic discount factor specifications. See `Ecf` for details.
 - `Eexposure` - a tensor containing the shock-exposure elasticities. The shock-exposure elasticity of cash-flow k with respect to shock m at horizon t is stored in `Eexposure(k)` and takes the linear form

$$\varepsilon_{km}(x, t) = \Gamma_0(m, 1, t) + \Gamma_1(m, :, t) x_{1,}$$

- `XSTEADY` - shock-exposure elasticity evaluated at the deterministic steady state
 - `XEXP` - shock-exposure elasticity evaluated at the point around which the expansion is expressed
 - `XBAR` - shock-exposure elasticity evaluated at the ‘fix point’ of the deterministic rule
 - `XMEAN` - shock-exposure elasticity evaluated at the mean of the distribution of X (calculated by Dynare or Dynare++)
 - `simul_quantiles` - shock-exposure elasticity evaluated at the quantiles of the distribution of elasticities. `simul_quantiles(m,n,t)` is the elasticity with respect to shock m , at quantile n and horizon t .
 - `cf_name` - name of the cash flow variable for which the shock-exposure elasticity is calculated
 - `cf_index` - index of the cash flow variable in the vector of variables
 - `name` - name of the shock-exposure elasticity function used for graph titles, etc.
 - `sdf_name`, `sdf_index` - unused
- `Eprice` - matrix of shock-price elasticities. `Eprice(j,k)` contains the shock-price elasticity of cash flow k under stochastic discount factor j . For details, see `Eexposure`. The only additional fields defined are:
 - `sdf_name` - name of the stochastic discount factor variable for which the shock-exposure elasticity is calculated
 - `sdf_index` - index of the stochastic discount factor variable in the vector of variables
 - `Entropy_sdf`, `Entropy_cf` - (**experimental, needs more testing**) entropy decomposition for multiplicative functionals representing the stochastic discount factors and cash flows, respectively. These fields have the same structure as `Eexposure` or `Eprice`. The only exception that the first dimension in the `GAMMA`. and `X...` matrices does not index the shock but has only one element which represents the entropy. By default, entropies are not plotted (see Section 6.2 for details).

5 What the elasticity toolbox does

The construction of the shock elasticities performed by the toolbox is based on theory developed in Hansen and Scheinkman (2010) and Borovička, Hansen, Hendricks, and Scheinkman (2011) for the continuous-time case, and in Borovička and Hansen (2011) for the discrete-time case.

5.1 Dynamics

Borovička and Hansen (2011) consider the following triangular (recursively-linear) state vector system:

$$\begin{aligned}
X_{1,t+1} &= \Theta_{10} + \Theta_{11}X_{1,t} + \Lambda_{10}W_{t+1} \\
X_{2,t+1} &= \Theta_{20} + \Theta_{21}X_{1,t} + \Theta_{22}X_{2,t} + \Theta_{23}(X_{1,t} \otimes X_{1,t}) \\
&\quad + \Lambda_{20}W_{t+1} + \Lambda_{21}(X_{1,t} \otimes W_{t+1}) + \Lambda_{22}(W_{t+1} \otimes W_{t+1}).
\end{aligned} \tag{3}$$

where the iid shock vector satisfies $W \sim N(0, I)$. Such a system allows for stochastic volatility, and we restrict the matrices Θ_{11} and Θ_{22} to have stable eigenvalues.

Cash flows and stochastic discount factors are constructed as multiplicative functionals. A multiplicative functional is a process $M = \exp(Y)$ where Y is a so-called additive functional. Increments of additive functionals are modeled as stationary functions of the state vector and the shock vector

$$\begin{aligned}
Y_{t+1} - Y_t &= \Gamma_0 + \Gamma_1X_{1,t} + \Gamma_2X_{2,t} + \Gamma_3(X_{1,t} \otimes X_{1,t}) \\
&\quad + \Psi_0W_{t+1} + \Psi_1(X_{1,t} \otimes W_{t+1}) + \Psi_2(W_{t+1} \otimes W_{t+1}).
\end{aligned} \tag{4}$$

Examples of such increments include the increment in the logarithm of the consumption process, $\log C_{t+1} - \log C_t$, or the increment in the logarithm of the stochastic discount factor, $\log S_{t+1} - \log S_t$.

It turns out that equilibrium models solved by perturbation methods used in [Holmes \(1995\)](#) or [Lombardo \(2010\)](#) satisfy the functional form (3).

5.2 Shock elasticities

Shock elasticities capture the sensitivity of multiplicative functionals with respect to small perturbations. Consider a perturbation

$$\log H_1(r) = -\frac{r^2}{2} |\alpha_h(X_0)|^2 + r\alpha_h(X_0) \cdot W_1$$

where r is a scalar parameter, α_h is the direction of the exposure, and we impose the scaling $E \left[|\alpha_h(X_0)|^2 \right] = 1$. A shock elasticity function for process M is defined as

$$\varepsilon(x, t) = \frac{d}{dr} \log E[M_t H_1(r) | X_0 = x] \Big|_{r=0} = \alpha_h(x) \cdot \frac{E[M_t W_1 | X_0 = x]}{E[M_t | X_0 = x]}.$$

[Borovička and Hansen \(2011\)](#) show how to construct the shock elasticities in closed form for the dynamics (3–4). These formulas are implemented in the elasticity toolbox.

Two types of shock elasticities are of special interest. Given a cash-flow process G , a *shock-exposure elasticity*

$$\varepsilon_g(x, t) = \frac{d}{dr} \log E[G_t H_1(r) | X_0 = x] \Big|_{r=0}$$

represents the sensitivity of the expected cash flow G to the shock W in direction α_h . Additionally, given a stochastic discount factor process S , a *shock-price elasticity*

$$\varepsilon_p(x, t) = \frac{d}{dr} \log E[G_t H_1(r) | X_0 = x] \Big|_{r=0} - \frac{d}{dr} \log E[S_t G_t H_1(r) | X_0 = x] \Big|_{r=0}$$

captures the sensitivity of the expected return on the cash flow G to the shock W in direction α_h .

5.3 Entropy decomposition

[Alvarez and Jermann \(2005\)](#), [Backus, Chernov, and Zin \(2011\)](#) and others analyze entropy measures. The t -period entropy of a multiplicative functional M is

$$\frac{1}{t} [\log E[M_t | X_0 = x] - E[\log M_t | X_0 = x]].$$

[Borovička and Hansen \(2011\)](#) show that one can decompose the entropy into one-period contributions by utilizing conditioning information. Define the one-period contribution as

$$\zeta_m(x, t) \equiv \log E[M_t | X_0 = x] - E[\log E[M_t | W_1, X_0] | X_0 = x]$$

Then

$$\frac{1}{t} [\log E[M_t | X_0 = x] - E[\log M_t | X_0 = x]] = \frac{1}{t} \sum_{j=1}^t E[\zeta_m(X_{t-j}, j) | X_0 = x].$$

The function $\zeta_m(x, t)$ shares the functional form that fits the dynamics (3)–(4), and its computation can therefore be implemented in this toolbox as well.

6 Function reference

This section describes the functions that provide the user interface, and the structure of the parameter file.

6.1 Functions

6.1.1 dynare_elasticity

Syntax: `[elast =] dynare_elasticity[(modelname[, param_filename])]`

Calculates shock-price and shock-exposure elasticities, from a model solved using Dynare or Dynare++.

The function reads the model solution generated by Dynare either from memory or from a mat file specified by `modelname`, and constructs shock-price and shock-exposure elasticities. The function plots graphs for the shock elasticities (can be controlled in the parameter file), and stores results in the return argument `elast`, and in an output file `modelname_elast_parameters.mat`.

Make sure to set the `model_periodlength` parameter correctly in the parameter file in order to produce the correct annualized magnitudes of the elasticities.

Input: If any input is omitted, it is assumed that `dynare_elasticity` is called immediately after solving the model using Dynare (`dynare('modelname')`), and variables `M_` and `oo_` are stored in memory.

- `modelname` - mat file with the model solution (without the extension). If the mat file contains the `oo_` variable, it is treated as a data file produced by Dynare, otherwise it is treated as a datafile produced by Dynare++.
- `param_filename` - Matlab code file (without extension) with parameters that overrides the default parameterization of the elasticity calculation. See example below.

Output: The return argument can be omitted.

- `elast` - structure containing the results. See Section 4.5 for details.

Examples: Start in the `examples` directory, and make sure that the dynare elasticity toolbox folder (containing the function `dynare_elasticity.m`) is on your path.

1. Solve the model `example_RBC.mod` using Dynare, calculate shock elasticities from the model solution stored in memory, and store the elasticities in the `elast` structure. The specification parameters will only be read from `default_elast_parameters.m`, located in the main toolbox directory.

```
dynare('example_RBC.mod');
elast = dynare_elasticity;
```

2. Solve the model `example_RBC.mod` using Dynare, rename the model solution file, calculate shock elasticities from the model solution file, and store the elasticities in the `elast` structure. Shock elasticities will also be stored in the file `example_RBC_elasticity.mat`. The specification parameters will be read from `default_elast_parameters.m` (located in the main toolbox directory), and subsequently from `example_RBC_elast_parameters.m` (if it exists in the directory with the model solution file).

```
dynare('example_RBC.mod');
system('ren example_RBC_results.mat example_RBC.mat');
% on Unix machines, use mv instead of ren
elast = dynare_elasticity('example_RBC');
```

3. Solve the model `example_RBC_plusplus.mod` using Dynare++, calculate shock elasticities from the model solution file, and store the shock elasticities in the `elast` structure. Shock elasticities will also be stored in the file `example_RBC_plusplus_elasticity.mat`. The specification parameters will be read from `default_elast_parameters.m` (located in the main toolbox directory), and subsequently from `example_RBC_plusplus_elast_parameters.m` (if it exists in the directory with the model solution file).

```
system('dynare++ example_RBC_plusplus.mod');
% Dynare++ must be on your system path. Alternatively, run Dynare++ from the command line.
elast = dynare_elasticity('example_RBC_plusplus');
```

4. Calculate shock elasticities from a model solution stored in `example_RBC.mat`, using a parameter file name `myparam.m` which overrides parameter values loaded from files described in previous examples. The parameter files are searched in the following order:
 - (a) `default_elast_parameters.m` - stored in the toolbox directory, searched automatically
 - (b) `modelname_elast_parameters.m` - stored in the directory where the model solution file is. In this example, the toolbox would search for `examples/example_RBC_elast_parameters.m`. This file does not need to exist.
 - (c) `myparamfile.m` - if it is specified as the second argument when running `dynare_elasticity`. This file should be stored in the same directory as the model solution file (in this example, `examples/myparamfile.m`).

```
elast = dynare_elasticity('example_RBC', 'myparamfile');
```

6.1.2 plot_elast_graphs

Syntax: `[fighandles =] plot_elast_graphs(model[, param_filename])`

Plots graphs for previously calculated elasticities.

The function can be called to re-plot shock elasticities without the need to resolve the model. The second argument `param_filename` can be used to provide a different parameter file with alternative graph specifications. See Section 6.2 for the description of the graph plotting parameters.

Input:

- `model` - This argument can either be a structure containing the calculated elasticities (see the `elast` structure returned by `dynare_elasticity`), or the name of a file containing this structure.
- `param_filename` - filename with parameters that overrides the default parameterization for the graph plotting.

Output: The return argument can be omitted.

- `fighandles` - handles for the plotted figures, for easier referencing in automated processing.

Examples:

1. Load shock elasticities from `example_RBC_elasticity.mat`, and use the file `mygraphparams.m` as the parameter file to override selected graph plotting parameters.

```
fighandles = plot_elast_graphs('example_RBC_elasticity', 'mygraphparams');
```

2. Load shock elasticities from the `elast` structure obtained previously using `dynare_elasticity`. Use the parameter file `mygraphparams.m` to override selected graph plotting parameters.

```
fighandles = plot_elast_graphs(elast, 'mygraphparams');
```

6.2 Parameter file

The parameter file controls the behavior of the shock elasticity toolbox. This is a regular Matlab code file that is called as a piece of code, so an incorrect syntax will result in an error. **Make sure to set the `model_periodlength` parameter correctly in the parameter file in order to produce the correct annualized magnitudes of the elasticities.**

When calling `dynare_elasticity('modelname', 'param_filename')`, three different parameter files are searched, in the following order:

- (a) `default_elast_parameters.m` - stored in the toolbox directory, searched automatically. Although this file can be edited directly, it may be more desirable to provide model-specific parameter values using the following parameter files.
- (b) `modelname_elast_parameters.m` - stored in the directory where the model solution file is. This file does not need to exist.
- (c) `param_filename.m` - if it is specified as the second argument when running `dynare_elasticity`. This file should be stored in the same directory as the model solution file.

The latter files may contain only a subset of parameters which should be overwritten.

The graph-plotting function `plot_elast_graphs` may be supplied with another parameter file that contains graph-specific parameter values.

Model-related parameters

- `model_periodlength` - period length in the model. Required to be set appropriately in order to produce the correct annualized magnitudes of shock elasticities.
- `model_periodname` - textual information about the period length in the model
- `model_T` - maximum horizon for which to calculate the shock elasticities
- `model_recursive` - type of recursive approximation method to be applied
 - = 1 - Split the dynamics of the state vector X into first- and second-order dynamics and construct the dynamics for $X_{1.}$ and $X_{2.}$. (respects separation of first- and second-order terms) [default].
 - = 2 - Split the dynamics of X into first- and second-order dynamics and construct the dynamics for X_1 and $X_{1.} + X_{2.}$. (the $X_{1.} + X_{2.}$ variable is easier to interpret as X).

Distribution simulation parameters

- `simul_distribution_run` - (0 or 1) determines whether simulated quantiles of the shock elasticities should be calculated [default = 1]
- `simul_paths` - number of paths simulated in parallel
- `simul_dropperiods` - number of initial periods to drop, in order to mitigate the impact of initial conditions
- `simul_record_periods` - number of periods for the calculation of the simulated stationary distribution. The total number of data points is `simul_paths` \times `simul_recordperiods`.
- `simul_calc_shockpriceelast` - (0 or 1) determines whether quantiles of the shock-price elasticities should be calculated [default = 1]
- `simul_calc_shockexposureelast` - (0 or 1) determines whether quantiles of the shock-exposure elasticities should be calculated [default = 1]

- `simul_calc_entropy_sdf`, `simul_calc_entropy_cf` - (0 or 1) determines whether quantiles of the entropy decomposition for stochastic discount factors and cash flows, respectively, are calculated [default = 0] (**needs more testing!**)
- `simul_quantiles` - vector of quantiles on the (0,1) interval for which the shock elasticities should be recorded
- `simul_zerooutthreshold` - numerical threshold for resetting escaping paths in the simulations. This may only be an issue for models which are not in the recursive-linear form [default = 0, i.e., no threshold].

Plotting parameters

- `plot_graphs` - (0 or 1) determines whether graphs will be plotted [default = 1]. Additional parameters determine desired subsets of shock elasticity types and entropy decompositions to be plotted.
- `plot_periodlength` - length of period for the horizontal axis of the graphs. This does not influence the scaling of the shock elasticity functions which are always annualized.
- `plot_periodname` - textual information that will be used as a descriptor for the horizontal axis.
- `plot_horizon` - maximum horizon plotted.
- `plot_elastscale` - scaling of the elasticity unit [default = 1, annualized shock elasticity]
- `plot_elastunit` - text description of the shock elasticity unit, displayed as the label on the vertical axis
- `plot_doxsteady`, `plot_doxexp`, `plot_doxbar`, `plot_doxmean` - (0 or 1) plot shock elasticities for specific values of the state vector X . For the description of the individual points (`XSTEADY`, `XEXP`, `XBAR`, `XMEAN`), see the description of X in the output file in Section 4.5.
- `plot_doquantiles` - (0 or 1) plot shock elasticities at individual quantiles [default = 1]
- `plot_shock_indices` - plot only shock elasticities corresponding to selected shock indices. By default, the vector is empty, and all elasticities corresponding to all shocks are plotted. Enumerate the shock indices if only a subset of shock should be plotted [default = []].
- `plot_savedirectory` - subdirectory in which graphs will be stored [default = `graphs`]
- `plot_saveeps`, `plot_savefig`, `plot_savepng` - (0 or 1) save graphs in the specified formats (jpg, Matlab figure, png).
- `plot_filename_spec` - by default, all graph file names have a common prefix, corresponding to the name of the model file. `plot_filename_spec` can specify a second prefix used after the model file name.
- `plot_graphtitle` - (0 or 1) determines whether to plot graph titles [default = 1]
- `plot_titlenote` - textual information that will appear at the end of each graph title (e.g., to distinguish different versions of the graphs)
- `plot_graphlegend` - (0 or 1) determines whether to plot the legend into each of the graphs (the legend contains entries for each plotted shock) [default = 1]
- `plot_lineorder` - plot line type specification
- `plot_linewidth` - line width for plotting the shock elasticities at the individual points (`XSTEADY`, `XEXP`, `XBAR`, `XMEAN`). Quantiles are always plotted with width = 1.
- `plot_horiz_scaling` - horizontal size of the graph, in pixels

- `plot_vert_scaling` - vertical size of the graph, in pixels
- `plot_figure_rows`, `plot_figure_columns` - specifies the number of subplots plotted in each graph. By default, both values are equal to 1, which implies that each cash flow (for shock-exposure elasticities) and each combination of a cash flow and a stochastic discount factor (for shock-price elasticities) is plotted into a new graph window (the filename for the graph will contain the name of the stochastic discount factor and/or cash flow).

When there are many cash flows, this may open too many windows. These two options allow to control the number of subplots in each graph window for more concise plotting (in this case, the filenames will only contain an index number). It may be desirable to change the size of the graph window (above) when using multiple subplots in a single window.

Specific parameters

- `env_displayverbose` - (0 or 1) produce much more text output [default = 0].
- `env_displaysilent` - (0 or 1) produce no text output, overrides `env_displayverbose` [default = 0].

References

- Alvarez, Fernando, and Urban J. Jermann, 2005. “Using Asset Prices to Measure the Persistence of the Marginal Utility of Wealth.” *Econometrica*. 73 (6): 1977–2016.
- Backus, David K., Mikhail Chernov, and Stanley E. Zin, 2011. “Sources of Entropy in Representative Agent Models.” NBER Working paper W17219.
- Binsbergen, Jules H. van, Michael W. Brandt, and Ralph S. J. Koijen, 2011. “On the Timing and Pricing of Dividends.” Forthcoming in *American Economic Review*.
- Borovička, Jaroslav, and Lars Peter Hansen, 2011. “Examining Macroeconomic Models through the Lens of Asset Pricing.” University of Chicago.
- Borovička, Jaroslav, Lars Peter Hansen, Mark Hendricks, and José A. Scheinkman, 2011. “Risk-Price Dynamics.” *Journal of Financial Econometrics*. 9 (1): 3–65.
- Hansen, Lars Peter, and José A. Scheinkman, 2010. “Pricing Growth-Rate Risk.” Forthcoming in *Finance and Stochastics*.
- Holmes, Mark H., 1995. *Introduction to Perturbation Methods*, Springer.
- Lettau, Martin, and Jessica A. Wachter, 2007. “Why Is Long-Horizon Equity Less Risky? A Duration-Based Explanation of the Value Premium.” *The Journal of Finance*. 62 (1): 55–92.
- Lombardo, Giovanni, 2010. “On Approximating DSGE Models by Series Expansions.” ECB Working paper No. 1264.