# Stochastic simulations with DYNARE. A practical guide.

Fabrice Collard (GREMAQ, University of Toulouse)
Adapted for Dynare 4.x
by Michel Juillard and Sébastien Villemot (CEPREMAP)

First draft: February 2001 This draft: December 2009.

This document describes a model involving both endogenous and exogenous state variable. We first describe the theoretical model, before showing how the perturbation method is implemented in DYNARE.

## 1 A theoretical model

We consider an economy that consists of a large number of dynastic households and a large number of firms. Firms are producing a homogeneous final product that can be either consumed or invested by means of capital and labor services. Firms own their capital stock and hire labor supplied by the households. Households own the firms. In each and every period three perfectly competitive markets open — the markets for consumption goods, labor services, and financial capital in the form of firms' shares. Household preferences are characterized by the lifetime utility function:

$$E_t \sum_{\tau=t}^{\infty} \beta^{\star\tau-t} \left( \log(c_t) - \theta \frac{h_t^{1+\psi}}{1+\psi} \right) \tag{1}$$

where $0 < \beta^\star < 1$ is a constant discount factor, $c_t$ is consumption in period $t$, $h_t$ is the fraction of total available time devoted to productive activity in period $t$, $\theta > 0$ and $\psi \geqslant 0$. We assume that there exists a central planner that determines hours, consumption and capital accumulation maximizing the household's utility function subject to the following budget constraint

$$c_t + i_t = y_t \tag{2}$$

where $i_t$ is investment and $y_t$ is output. Investment is used to form physical capital, which accumulates in the standard form as:

$$k_{t+1} = \exp(b_t) i_t + (1-\delta) k_t \text{ with } 0 < \delta < 1 \tag{3}$$

where $\delta$ is the constant physical depreciation rate. $b_t$ is a shock affecting incorporated technological progress, which properties will be defined later.

Output is produced by means of capital and labor services, relying on a constant returns to scale technology represented by the following Cobb–Douglas production function:

$$y_t = \exp(a_t) k_t^{\alpha} h_t^{1-\alpha} \text{ with } 0 < \alpha < 1 \tag{4}$$

$a_t$ represents a stochastic shock to technology or Solow residual. We assume that the shocks to technology are distributed with zero mean, but display both persistence across time and correlation in the current period. Let us consider the joint process $(a_t, b_t)$ defined as

$$\begin{pmatrix} a_t \\ b_t \end{pmatrix} = \begin{pmatrix} \rho & \tau \\ \tau & \rho \end{pmatrix} \begin{pmatrix} a_{t-1} \\ b_{t-1} \end{pmatrix} + \begin{pmatrix} \varepsilon_t \\ \nu_t \end{pmatrix} \tag{5}$$

where $|\rho + \tau| < 1$ and $|\rho - \tau| < 1$ for sake of stationarity and

$$\begin{aligned}
E(\varepsilon_t) &= 0, \\
E(\nu_t) &= 0, \\
E(\varepsilon_t \varepsilon_s) &= \begin{cases} \sigma_\varepsilon^2 & \text{if} \quad t = s \\ 0 & \text{if} \quad t \neq s \end{cases}, \\
E(\nu_t \nu_s) &= \begin{cases} \sigma_\nu^2 & \text{if} \quad t = s \\ 0 & \text{if} \quad t \neq s \end{cases}, \\
E(\varepsilon_t \nu_s) &= \begin{cases} \varphi \sigma_\varepsilon \sigma_\nu & \text{if} \quad t = s \\ 0 & \text{if} \quad t \neq s \end{cases}.
\end{aligned}$$

## 2 Dynamic Equilibrium

The dynamic equilibrium of this economy follows from the first order conditions for optimality:

$$\begin{aligned}
& c_t \theta h_t^{1+\psi} = (1-\alpha) y_t \\
& \beta E_t \left[ \left( \frac{\exp(b_t) c_t}{\exp(b_{t+1}) c_{t+1}} \right) \left( \exp(b_{t+1}) \alpha \frac{y_{t+1}}{k_{t+1}} + 1 - \delta \right) \right] = 1 \\
& y_t = \exp(a_t) k_t^{\alpha} h_t^{1-\alpha} \\
& k_{t+1} = \exp(b_t)(y_t - c_t) + (1-\delta) k_t \\
& a_t = \rho a_{t-1} + \tau b_{t-1} + \varepsilon_t \\
& b_t = \tau a_{t-1} + \rho b_{t-1} + \nu_t
\end{aligned}$$

## 3 The DYNARE code

The DYNARE code is straightforward to write, as the equilibrium is written in the natural way. The whole code is reported at the end of the section. Before that we proceed step by step.

**Preamble** The preamble consists of the some declarations to setup the endogenous and exogenous variables, the parameters and assign values to these parameters.

1. `var y, c, k, h, a, b;` specifies the endogenous variables in the model since we have output (`y`), consumption (`c`), capital (`k`), hours (`h`) and the two shocks (`a, b`).

2. `varexo e, u;` specifies the exogenous variables in the model — namely the innovations of the shocks, since we have the innovation of the non–incorporated shock (`e`), and the innovation of the incorporated shock (`u`).

3. `parameters list;` specifies the list of parameters of the model. In the case we are studying:

   ```
   parameters beta, alpha, delta, theta, psi, rho, tau;
   ```

   | | |
   |---|---|
   | `beta` | discount factor |
   | `alpha` | capital elasticity in the production function |
   | `delta` | depreciation rate |
   | `theta` | disutility of labor parameter |
   | `psi` | labor supply elasticity |
   | `rho` | persistence |
   | `tau` | cross–persistence |

4. Assignment of parameter values. This is done the standard way in MATLAB. For example, we write

   ```
   alpha = 0.36;
   rho   = 0.95;
   tau   = 0.025;
   beta  = 0.99;
   delta = 0.025;
   psi   = 0;
   theta = 2.95;
   ```

5. Note that $\varphi$, the conditional correlation of the shocks, is not, strickly speaking, a parameter of the recursive equations and doesn't need to be listed in the `parameters` instruction. It may however be convenient to express it as a parameter in the expression of the variance–covariance matrix of the shocks (see below) and one may simply write:

   ```
   phi = 0.1;
   ```

**Declaration of the model:** This step is done in a straightforward way. It starts with the instruction `model;` and ends with `end;`, in between all equilibrium conditions are written exactly the way we write it "by hand". However, there is a

simple rule that should be kept in mind when the model is written. Let us consider a variable $x$:

- If $x$ is decided in period $t$ then we simply write `x`.
- When the variable is decided in $t-1$, such as the capital stock in our simple model, we write `x(-1)`.
- Finally, when a variable is decided in the next period, $t+1$, such as consumption in the Euler equation, we write `x(+1)`.

Hence the required code to declare our model in DYNARE will be:

```
model;
c*theta*h^(1+psi)=(1-alpha)*y;
k = beta*(((exp(b)*c)/(exp(b(+1))*c(+1)))*
    (exp(b(+1))*alpha*y(+1)+(1-delta)*k));
y = exp(a)*(k(-1)^alpha)*(h^(1-alpha));
k = exp(b)*(y-c)+(1-delta)*k(-1);
a = rho*a(-1)+tau*b(-1) + e;
b = tau*a(-1)+rho*b(-1) + u;
end;
```

Assume now that we want to take a Taylor series expansion in logs rather than in level, we just rewrite the model as

```
model;
exp(c)*theta*exp(h)^(1+psi)=(1-alpha)*exp(y);
exp(k) = beta*(((exp(b)*exp(c))/(exp(b(+1))*exp(c(+1))))
         *(exp(b(+1))*alpha*exp(y(+1))+(1-delta)*exp(k)));
exp(y) = exp(a)*(exp(k(-1))^alpha)*(exp(h)^(1-alpha));
exp(k) = exp(b)*(exp(y)-exp(c))+(1-delta)*exp(k(-1));
a = rho*a(-1)+tau*b(-1) + e;
b = tau*a(-1)+rho*b(-1) + u;
end;
```

so that the level of consumption is actually given by `exp(c)`.

**Solving the model**

1. Now we need to provide numerical initial conditions for the computation of the deterministic steady state. This is done with the sequence between `initval;` and `end;`. Each variable, endogenous or exogenous, should be initialized. In our example, we give the exact values of the deterministic equilibrium in absence of shocks. This takes the form

```
initval;
y = 1.08068253095672;
c = 0.80359242014163;
h = 0.29175631001732;
k = 11.08360443260358;
a = 0;
b = 0;
e = 0;
u = 0;
end;
```

Alternatively, we could provide only approximated values. DYNARE would then automatically compute the exact values.

2. We then specify the innovations and their matrix of variance–covariance. This is done using a `shocks;` and `end;` block and declare only the nonzero elements of the covariance matrix:

```
shocks;
var e; stderr 0.009;
var u; stderr 0.009;
var e, u = phi*0.009*0.009;
end;
```

Note that it is possible to shut down a shock by assigning it a zero variance.

3. The model is then solved and simulated using the `stoch_simul;` command. By default, the coefficients of the approximated decision rules are reported as well as the moments of the variables and impulse response functions for each exogenous shocks are ploted. In addition, the following options are available (out of many others):

   - `ar` = Integer: 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`: Doesn't print the correlation matrix (default = PRINT)
   - `drop` = Integer: Number of points dropped at the beginning of simulation before computing the summary statistics (default = 100)
   - `irf` = Integer: Number of periods on which to compute the IRFs (default = 40)

- `nofunctions`: Doesn't print the coefficients of the approximated solution
- `nomoments`: Doesn't print moments of the endogenous variables
- `order` = [1,2,3]: Order of Taylor approximation (default = 2)
- `replic` = Integer: Number of simulated series used to compute the IRFs (default = 1, if order = 1, and 50 otherwise)

In our first example, we use simply:

```
stoch_simul;
```

DYNARE will compute theoretical moments of variables.

In our second example, we use:

```
stoch_simul(periods=2000, drop=200);
```

DYNARE will compute simulated moments of variables. The simulated trajectories are returned in MATLAB vectors named as the variables (be careful not to use MATLAB reserved names such as INV for your variables ...).

# A DYNARE code for the model in level

Here is the model file for the model in level. It can be found in file `example1.mod`.

```
var y, c, k, a, h, b;
varexo e, u;

parameters beta, rho, alpha, delta, theta, psi, tau;

alpha = 0.36;
rho   = 0.95;
tau   = 0.025;
beta  = 0.99;
delta = 0.025;
psi   = 0;
theta = 2.95;

phi   = 0.1;

model;
c*theta*h^(1+psi)=(1-alpha)*y;
k = beta*(((exp(b)*c)/(exp(b(+1))*c(+1)))
    *(exp(b(+1))*alpha*y(+1)+(1-delta)*k));
```

```
y = exp(a)*(k(-1)^alpha)*(h^(1-alpha));
k = exp(b)*(y-c)+(1-delta)*k(-1);
a = rho*a(-1)+tau*b(-1) + e;
b = tau*a(-1)+rho*b(-1) + u;
end;

initval;
y = 1.08068253095672;
c = 0.80359242014163;
h = 0.29175631001732;
k = 11.08360443260358;
a = 0;
b = 0;
e = 0;
u = 0;
end;

shocks;
var e; stderr 0.009;
var u; stderr 0.009;
var e, u = phi*0.009*0.009;
end;

stoch_simul;
```

## B DYNARE code for the model in logs

Here is the model file for the model in logs. In this case, `initval` only contains guessed values and `steady` is used to compute and display the exact value of the deterministic equilibrium. The shocks are supposed to be uncorrelated. The model file can be found in `example2.mod`.

```
var y, c, k, a, h, b;
varexo e, u;

parameters beta, rho, alpha, delta, theta, psi, tau;

alpha = 0.36;
rho   = 0.95;
tau   = 0.025;
beta  = 0.99;
delta = 0.025;
psi   = 0;
theta = 2.95;
```

```
model;
exp(c)*theta*exp(h)^(1+psi)=(1-alpha)*exp(y);
exp(k) = beta*(((exp(b)*exp(c))/(exp(b(+1))*exp(c(+1))))
         *(exp(b(+1))*alpha*exp(y(+1))+(1-delta)*exp(k)));
exp(y) = exp(a)*(exp(k(-1))^alpha)*(exp(h)^(1-alpha));
exp(k) = exp(b)*(exp(y)-exp(c))+(1-delta)*exp(k(-1));
a = rho*a(-1)+tau*b(-1) + e;
b = tau*a(-1)+rho*b(-1) + u;
end;

initval;
y = 0.1;
c = -0.2;
h = -1.2;
k =  2.4;
a = 0;
b = 0;
e = 0;
u = 0;
end;

steady;

shocks;
var e = 0.009^2;
var u = 0.009^2;
end;

stoch_simul(periods=2000, drop=200);
```