Solving DSGE Models with Dynare

Fabio Canova
EUI and CEPR
September 2014

Outline

- Installation

- The syntax

- Some examples (level vs. logs, first vs. second order)

- Some tips

- Computing statistics.

- Checking accuracy of the approximation.

- Computing forecasts.

1 TO instal and run Dynare

To install it:

1. Go to the Dynare website: http://www.dynare.org .

2. In the Main Menu go to Download.

3. Choose Dynare for Matlab/Octave [Windows].

4. Download version 4.4.0 (all the programs should work with it) and save it in Dynare directory in the C drive. Unpack the zip file in this directory.

5. Start Matlab. Click in the File menu on Set Path.

6. Click on the button Add with Subfolders. Now select the Dynare directory. Click on Save and close the dialog window.

To run it:

1. In the Matlab main window change the directory to the one in which you have stored your Dynare program.

2. To run the program myfile.mod type the command: Dynare myfile.mod

1.1 What is Dynare?

- It is an interface to Matlab/Octave (also to C++).

- It takes a user-supply the file (which looks very much like what you write on a piece of paper), transforms it into a series of Matlab files and runs it.

- Dynare is a collection of routines, written by various people (economists) and some connecting programs, written by computer programmers.

2 Perturbation methods: a review

We want to obtain an approximate policy function that satisfies the first order conditions.

Let ϵ_t be a $n \times 1$ vector of state (exogenous and predetermined) variables and y_t a $m \times 1$ vector of endogenous variables. The first order (linear) approximation is

$$y_t - \bar{y} = (\epsilon_t - \bar{\epsilon})' a \tag{1}$$

where a bar indicates steady state values. For a simple consumption/saving model (1) is

$$c_t - \bar{c} = a_{c,k}(k_{t-1} - \bar{k}) + a_{c,z}(z_t - \bar{z}) \tag{2}$$

where k_t is the capital stock and z_t a technology shock.

The FOCs of a DSGE model are of the form:

$$E_t f(y, \epsilon, \theta) = 0 \tag{3}$$

Let $y = h(\epsilon, \theta)$ be the unknown policy function.

First order (perturbation) method: Find the coefficients of the linear approximation to the $h(\epsilon, \theta)$ function, i.e. $h(\epsilon, \theta) = h_0(\theta) + h_1(\theta)(\epsilon - \overline{\epsilon})$.

Higher order (perturbation) method: Find the coefficients of the higher order approximation to the $h(\epsilon, \theta)$ function. i.e.$h(\epsilon, \theta) = h_0(\theta) + h_1(\theta)(\epsilon - \overline{\epsilon}) + h_2(\theta)(\epsilon - \overline{\epsilon})^2 + h_3(\theta)(\epsilon - \overline{\epsilon})^3 + \ldots$.

3 A simple growth model in Dynare

Level model

$$\max_{\{c_t,k_t\}} E_t \sum_{t=0}^{\infty} \beta^t (\frac{c_t^{1-\phi} - 1}{1-\phi}) \tag{4}$$

subject to

$$\begin{aligned} c_t + k_t &= z_t k_{t-1}^{\alpha} + (1-\delta)k_{t-1} &(5)\\ z_t &= \rho z_{t-1} + \epsilon_t, \quad \epsilon_t \sim (1, \sigma_e^2) &(6) \end{aligned}$$

FOCs:

$$\begin{aligned} c_t^{-\phi} &= \beta E_t[c_{t+1}^{-\phi}(\alpha z_{t+1} k_t^{\alpha-1} + (1-\delta))] &(7)\\ c_t + k_t &= z_t k_{t-1}^{\alpha} + (1-\delta)k_{t-1} &(8)\\ z_t &= \rho z_{t-1} + \epsilon_t &(9) \end{aligned}$$

Log model

$$\max_{\{c_t,k_t\}} E_t \sum_{t=0}^{\infty} \beta_t (\frac{c_t^{1-\phi} - 1}{1-\phi}) \tag{10}$$

subject to

$$c_t + k_t = exp(\tilde{z}_t) k_{t-1}^{\alpha} + (1-\delta) k_{t-1} \tag{11}$$

$$\tilde{z}_t = \rho \tilde{z}_{t-1} + \epsilon_t, \qquad \epsilon_t \sim (0, \tilde{\sigma}_e^2) \tag{12}$$

$\tilde{z}_t$ is the log of TFP. Note that the constraints of the two problems are identical; they are simply written in a different way.

FOCs:

$$
\begin{aligned}
c_t^{-\phi} &= \beta E_t c_{t+1}^{-\phi}[\alpha exp(\tilde{z}_{t+1}) k_t^{\alpha-1} + (1-\delta)] && (13)\\
c_t + k_t &= exp(\tilde{z}_t) k_{t-1}^{\alpha} + (1-\delta) k_{t-1} && (14)\\
\tilde{z}_t &= \rho \tilde{z}_{t-1} + \epsilon_t, && (15)
\end{aligned}
$$

Transform the FOCs so that also c_t, k_t are measured in log

$$
\begin{aligned}
(exp(\tilde{c}_t))^{-\phi} &= \beta E_t[(exp(\tilde{c}_{t+1}))^{-\phi}(\alpha exp(\tilde{z}_{t+1})(exp(\tilde{k}_t))^{\alpha-1} + (1-\delta))] && (16)\\
exp(\tilde{c}_t) + exp(\tilde{k}_t) &= exp(\tilde{z}_t)(exp(\tilde{k}_{t-1}))^{\alpha} + (1-\delta) exp(\tilde{k}_{t-1}) && (17)\\
\tilde{z}_t &= \rho \tilde{z}_{t-1} + \epsilon_t && (18)
\end{aligned}
$$

Note $exp(\tilde{c}_t) = c_t$ and $exp(\tilde{k}_t) = k_t$ if $\tilde{c} = \log c$ and $\tilde{k} = \log k$.

Then rewrite equations as:

$$\begin{aligned} exp(-\phi\tilde{c}_t) &= \beta E_t[exp(-\phi\tilde{c}_{t+1})(\alpha exp(\tilde{z}_{t+1}+(\alpha-1)\tilde{k}_t)+(1-\delta))] && (19)\\ exp(\tilde{c}_t)+exp(\tilde{k}_t) &= exp(\tilde{z}_t+\alpha\tilde{k}_{t-1})+(1-\delta)exp(\tilde{k}_{t-1}) && (20)\\ \tilde{z}_t &= \rho\tilde{z}_{t-1}+\epsilon_t && (21) \end{aligned}$$

(7)-(9) are the optimality conditions for the level of (c_t, k_t, z_t); (19)-(21) are the optimality conditions for $(\tilde{c}_t, \tilde{k}_t, \tilde{z}_t)$.

-Equations are slightly different.

- States (exogenous and predetermined) $x_t = [k_{t-1}, z_t]$.

- Controls $y_t = [c_t, k_t, z_t]$ (some redundancy here).

- First order approximate solution will be of the form

$$c_t = \bar{c} + a_{c,k}(k_{t-1} - \bar{k}) + a_{c,z}(z_t - \bar{z}) \quad (22)$$

$$k_t = \bar{k} + a_{k,k}(k_{t-1} - \bar{k}) + a_{k,z}(z_t - \bar{z}) \quad (23)$$

$$z_t = \bar{z} + \rho(z_{t-1} - \bar{z}) + \epsilon_t \quad (24)$$

Depending on how you write the model, $x_t = [c_t, k_t, z_t]$ could be the level or the (natural) log of the variables.

Typically $\bar{z} = 0$. How do we write these two models in Dynare?

3.1 Preliminaries

i) All variables known at time t must be dated $t-1$ (that is, k_t is a choice variables, k_{t-1} is a predetermined variable). Can change this timing convention using the "predetermined_variables" command.

ii) Dynare looks for predetermined and exogenous variables. Is a technology shock z_t a predetermined or an exogenous variable? Dynare artificially splits z_t into a predetermined component and an exogenous component. Thus, policy rules are:

$$c_t = \bar{c} + a_{c,k}(k_{t-1} - \bar{k}) + a_{c,z}\rho(z_{t-1} - \bar{z}) + a_{c,z}\epsilon_t \quad (25)$$

$$k_t = \bar{k} + a_{k,k}(k_{t-1} - \bar{k}) + a_{k,z}\rho(z_{t-1} - \bar{z}) + a_{k,z}\epsilon_t \quad (26)$$

$$z_t = \bar{z} + \rho(z_{t-1} - \bar{z}) + \epsilon_t \quad (27)$$

iii) Conditional expectations are omitted. Future variables are denoted by $y(+1), y(+2)$, etc.; lagged variables by $y(-1), y(-2)$, etc.

iv) For all variables dated at $t \pm k, \ k > 1$, Dynare creates a fake star variable e.g. $y^*_{t+1} = y_{t+2}$,substitutes y^*_{t+1} wherever y_{t+2} appears and adds an identity to the system of equations.

v) First and second order solutions look like:

$$
\begin{aligned}
y_t &= \bar{y} + A(y_{t-1} - \bar{y}) + B\epsilon_t && (28) \\
y_t &= \bar{y} + 0.5\Delta^2 + A(y_{t-1} - \bar{y}) + B\epsilon_t + 0.5C[(y_{t-1} - \bar{y}) \otimes (y_{t-1} - \bar{y})] \\
&+ 0.5D(\epsilon_t \otimes \epsilon_t) + F[(y_{t-1} - \bar{y}) \otimes \epsilon_t] && (29)
\end{aligned}
$$

where Δ^2 is the variance covariance matrix of the innovations in the shocks and A, B, C, D, F are matrices.

- The output is different from those of Klein, Uhlig or Sims. Careful!!!!

- Main difference between first and second order solution is that in the latter the standard deviation of the shocks Δ has effects on the (level) of y_t.

vi) Dynare automatically produces a series of MATLAB files. The files programname.m, programname_static.m, and programname_dynamic.m produce main program and the static and the dynamics equations of the model.

Once you have these files you do not need Dynare any longer. You can run them in MATLAB if you wish.

vii) The output is visible on the screen and saved in the file programname.log. The variables and matrices created solving the model are stored in the file programname_result.mat (a matlab storage file).

viii) Figures is saved in *.eps, *.fig, and *.pdf formats.

3.2 Dynare programming blocks

- Labelling block:

i) "var" are the endogenous variables

ii) "varexo" are the exogenous variables

iii) "predeterminated_variables" are the predeterminated variables

iv) "parameters" are the parameters of the model.

- Parameter block: gives the values of the parameters.

• Model block: starts with "model" and ends with "end". It contains the equations of the model. If the model has been already (log-)linearized by hand use the option "linear" ("loglinear") in the model command

The following two set of statements are equivalent:

var y, k, i;	var y, k, i;
	predetermined_variables k;
parameters alpha, delta;	parameters alpha, delta;
alpha= 0.36;	alpha= 0.36;
delta=0.025;	delta=0.025;
model;	model;
y=k(-1)^alpha;	y=k^alpha;
k= i +(1-delta) k(-1);	k(+1)= i +(1-delta) k;
end;	end;

• Initialization block: starts with "initval" and ends with "end" . It sets the initial conditions for the variables. It is important to choose these well because they may influence the calculation of the steady states (which is a non-linear problem).

```
initval;
k=1.5;
y=1.3;
i=1.0;
end;
```

In some cases, rather than initial conditions, you want to fix terminal conditions. For terminal conditions use the "endval" command.

How do you choose initial conditions? Use steady states or the steady solution to simplified versions of the model (if the model is very complicated).

Example 3.1 *When $\delta = \phi = 1.0$ (full capital depreciation, log utility). The analytic solution to the log model is:*

$$\tilde{k}_t = ln(\alpha\beta) + \alpha\tilde{k}_{t-1} + \tilde{z}_t \qquad (30)$$

$$\tilde{c}_t = ln(1-\alpha\beta) + \alpha\tilde{k}_{t-1} + \tilde{z}_t \qquad (31)$$

- Policy rules are linear in the log of the variables

- Consumption and investment are constant fractions of output.

Steady States

Setting $\tilde{k}_t = \tilde{k}_{t-1} = k_s$ *and* $\tilde{z}_t = 0$ *we have*

$$\tilde{k}_s = \frac{ln(\alpha\beta)}{1-\alpha} \tag{32}$$

$$\tilde{c}_s = ln(1-\alpha\beta) + \alpha\frac{ln(\alpha\beta)}{1-\alpha} \tag{33}$$

If $\alpha = 0.36, \beta = 0.99$, *then* $\tilde{k}_s = -1.612, \tilde{c}_s = -1.021$. *These could be used as initial values in the computations of the steady states of the general model.*

- Shock block: starts with "shocks" and ends with "end". It defines what variables are the stochastic shocks, their standard deviation and, potentially, the correlation between shocks.

```
shocks;
var e1;
stderr sig1;
var e2;
stderr sig2;
corr e1, e2 = 0.8
end;
```

• The steady state block: it contains the commands to calculate the steady states and to check for stability of the solution.

steady(options);

The options of steady are:

1) solve_algo=number; it select which algorithm is used to solve the system of non-linear equations; 0 (fsolve.m); 1 (dynare own nonlinear solver); 2 (default: recursive block splitting) 3 (Sims' solver). Try other options only if these fail (these use homotophy methods and Sparse Gaussian elimination).

2) homotomy_mode= number; homotomy_steps=number.

This last option should used when other algorithms fail (see Dynare manual for details)

- To avoid time consuming repetitions steady states can be loaded into the program with a file.

- The file must have the same name as the main file and the extension "_steadystate.m'. e.g. if the program is called rbc.mod the steady state can be loaded into the program using the file rbc_steadystate.m file check;

- It examine the conditions for existence of a solution to the system.

- Necessary condition: number of eigenvalues greater than one equal to the number of forward looking variables.

- There is a stronger rank condition (see dynare manual) that can also be checked.

- Solution block: it contains the commands to solve the model

First order solution : stoch_simul(order=1,nocorr,nomoments,IRF=0)

Second order solution: stoch_simul(order=2,nocorr,nomoments,IRF=0)

Third order solution: stoch_simul(order=3,nocorr,nomoments,IRF=0)

- Properties block: to calculate interesting statistics using the solution.

Some options with stoch_simul (default)

order: order of Taylor approximation (1, 2, 3)

k_order_solver: use a C++ solver (more complicated, check manual).

aim_solver: triggers the use of the nonlinear perfect foresight solver used in the AIM program

HP_filter=number: sets the smoothing parameter of the HP filter (128000 for monthly, 1600 for quarterly, 6.25 for annual data).

ar = number: sets the number of autocorrelations to be computed (5) irf=number: sets the number of responses to be computed; if number=0 suppresses plotting of impulse response functions.

relative_irf: computes normalized impulse responses.

periods: specifies the number of periods to use in simulations.

Note: periods triggers the computation of moments and correlation using **simulated data** rather than the population solution.

nocorr: correlation matrix is not printed

nofunctions: coefficients of approximated solution not printed

nomoments: moments of endogenous variables not printed

noprint: suppresses all printing (useful for loops)

Example: stoch_simul(order=1, irf=60) y k
computes impulse responses for y and k only, for 60 periods, using a first order approximation.

3.3 Where is the solution output stored?

- The output of steady is stored into oo_.steady_state. Endogenous variables are ordered as you have declared them.

- Output of check is stored in: oo_.dr.eigeval.

- Coefficients of the decision rules are stored in oo_.dr.xxx where xxx is

i) ys: steady state values

ii) ghx, ghu: the matrices A, B.

iii) ghxx, ghuu, ghxu : the matrices C, D, F

iv) ghs2 : the matrix Δ^2.

Summary statistics are stored in oo_.yyy where yyy is mean (mean values), var (variances) autocorr (autocorrelations), irf (impulse responses, with the convention variable_shock).

Example: oo_.irfs.gnp_ea contains the responses of gnp to a shock in ea.

A Dynare program for the level model

```
var c, k, z;
varexo e;
parameters beta, rho, alpha, phi, delta, sig;

alpha= 0.36; beta =0.99;
rho=0.95; phi=1.0;
delta=0.025; sig=0.007;

model;
c^(-phi)=beta*c(+1) ^(-phi)*(alpha*z(+1)*k^(alpha-1)+(1-delta))
c+k=z*k(-1)^alpha+(1-delta)*k(-1);
z=rho*z(-1)+e;
end;
```

```
initval;
k=6.35; c=1.31; z=0;
end;

shocks;
var e;
stderr sig;
end;

steady;
check;

stoch_simul(order=1,nocorr, nomoments, IRF=0);
```

Dynare equations for the log model

```
model;
exp(-phi*lc)=beta*exp(-phi*lc(+1))*(alpha*exp(lz(+1)+(alpha-1)*lk))
+1-delta);
exp(lc)+exp(lk)=exp(lz+alpha*lk(-1))+(1–delta)*exp(lk(-1);
lz=rho*lz(-1)+e;
end;
```

where lc= log consumption, lk = log capital, lz = log TFP.

Example 3.2 *(Behind the scenes: solving a RBC model)*

Social planner problem:

$$\max_{\{c_t, K_t, N_t\}} E_0 \sum_t \beta^t (\log c_t + \log(1 - N_t)) \tag{34}$$

subject to

$$c_t + K_t = e^{\zeta_t} K_{t-1}^{\alpha} N_t^{1-\alpha} + (1 - \delta) K_{t-1} \tag{35}$$

$$\zeta_t = \rho \zeta_{t-1} + e_t \tag{36}$$

Lagrangian:

$$\max_{\{c_t,K_t,N_t\}} E_0 \sum_t \beta^t [\log c_t + \log(1 - N_t) \quad - $$
$$\lambda_t(c_t + K_t - e^{\zeta_t} K_{t-1}^{\alpha} N_t^{1-\alpha} - (1-\delta)K_{t-1})] \tag{37}$$

The first order conditions (with respect to c_t, N_t, K_t) and the resource constraints are

$$\frac{1}{c_t} = \lambda_t \tag{38}$$
$$\frac{1}{1-N_t} = \lambda_t(1-\alpha)e^{\zeta_t} K_{t-1}^{\alpha} N_t^{-\alpha} \tag{39}$$
$$\lambda_t = \beta E_t \lambda_{t+1}[(\alpha e^{\zeta_{t+1}} K_t^{\alpha-1} N_{t+1}^{1-\alpha} + (1-\delta)] \tag{40}$$
$$c_t + K_t = e^{\zeta_t} K_{t-1}^{\alpha} N_t^{1-\alpha} + (1-\delta)K_{t-1} \tag{41}$$

Eliminating λ we have

$$\begin{aligned}
\frac{c_t}{1 - N_t} &= (1-\alpha) e^{\zeta_t} K_{t-1}^{\alpha} N_t^{-\alpha} && (42) \\
\frac{1}{c_t} &= \beta E_t \frac{1}{c_{t+1}} [\alpha e^{\zeta_{t+1}} K_t^{\alpha-1} N_{t+1}^{1-\alpha} + (1-\delta)] && (43) \\
c_t + K_t &= e^{\zeta_t} K_{t-1}^{\alpha} N_t^{1-\alpha} + (1-\delta) K_{t-1} && (44)
\end{aligned}$$

- *Three equations in three unknowns (c_t, K_t, N_t). These equations are nonlinear and involve expectations. Compute an approximate solution.*

Strategy

1) Find the steady states.

2) Take a first order expansion of the optimality conditions evaluated at steady state values.

3) Solve the approximation using the " guess and verify" method.

4) Compute second order approximation.

5) General variable transformation.

1) Computation of the steady states.

Assuming $\sigma_e = 0$ *and eliminating time subscripts*

$$\frac{c}{1-N} = (1-\alpha)K^{\alpha}N^{-\alpha} \tag{45}$$

$$1 = \beta[(\alpha K^{\alpha-1}N^{1-\alpha} + (1-\delta)] \tag{46}$$

$$c + \delta K = K^{\alpha}N^{1-\alpha} \tag{47}$$

The solution is:

- $K^s = z_1/(z_2 + z_3 z_1)$.

- $N^s = z_3 K^s$.

- $c^s = z_2 K^s$.

where $z_3 = (\frac{1}{\alpha}(\frac{1}{\beta} - 1 + \delta))^{1/(1-\alpha)}$, $z_2 = z_3^{1-\alpha} - \delta$, $z_1 = (1-\alpha) z_3^{-\alpha}$.

- *We can compute output and prices using the production function* $y^s = (K^s)^\alpha (N^s)^{1-\alpha}$ *and the FOC of the firm's problem* $w^s = y^s/N^s$ *and* $r^s = y^s/K^s$.

2) Linearize the optimality conditions and the low of motion of the shocks around (c^s, N^s, K^s).

$$
\begin{aligned}
(c_t - c^s) &= m_5\zeta_t + \frac{\alpha c^s}{K^s}(K_{t-1} - K^s) + m_6(N_t - N^s) && (48)\\
m_1(c_t - c^s) &= E_t[m_1(c_{t+1} - c^s) + m_2\zeta_{t+1} + m_3(K_t - K^s) + m_4(N_{t+1} - N^s)] && (49)\\
(c_t - c^s) &= -(K_t - K^s) + m_7\zeta_t + m_8(K_{t-1} - K^s) + m_9(N_t - N^s) && (50)\\
\zeta_t &= \rho\zeta_{t-1} + e_t && (51)
\end{aligned}
$$

where $m_1 = -(1/c^s), m_2 = \alpha(1-\alpha)\beta(y^s/K^s), m_3 = \alpha(\alpha-1)\beta(y^s/(K^s)^2),$

$m_4 = m_2/N, m_5 = (1-\alpha)c, m_6 = -(\alpha/N^s + 1/(1-N^s))c,$

$m_7 = (1-\alpha)y^s, m_8 = \alpha(y^s/K^s) + (1-\delta), m_9 = (y^s/N^s)(1-\alpha).$

Solving for $(c_t - c^s)$ from(48) we rewrite the system as:

$$A(K_t - K^s) + B(K_{t-1} - K^s) + C(N_t - N^s) + D\zeta_t = 0 \quad (52)$$

$$E_t[G(K_t - K^s) + H(K_{t-1} - K^s) + J(N_{t+1} - N^s) + I(N_t - N^s) + L\zeta_{t+1} + M\zeta_t] = 0 \quad (53)$$

$$E_t\zeta_{t+1} = N\zeta_t \quad (54)$$

Two equations in $(K_t - K^s)$ and $(N_t - N^s)$; both driven by ζ_t.

The matrices $A, B, \ldots, M, N$ are function of the parameters of the model.

3) Guess policy functions (solutions) of the form

$$(K_t - K^s) = P_1(K_{t-1} - K^s) + P_2\zeta_t$$

$$(N_t - N^s) = R_1(K_{t-1} - K^s) + R_2\zeta_t$$

Plugging these guesses in (52)-(53) and eliminating expectations we have

$$\begin{aligned} A(P_1(K_{t-1} - K^s) + P_2\zeta_t) + B(K_{t-1} - K^s) &+ \\ C(R_1(K_{t-1} - K^s) + R_2\zeta_t) + D\zeta_t &= 0 \qquad (55) \\ G(P_1(K_{t-1} - K^s) + P_2\zeta_t) + H(K_{t-1} - K^s) &+ \\ J(R_1(P_1(K_{t-1} - K^s) + P_2\zeta_t) + R_2N\zeta_t) &+ \\ I(R_1(K_{t-1} - K^s) + R_2\zeta_t)(LN + M)\zeta_t] &= 0 \qquad (56) \end{aligned}$$

These equations must hold for any $(K_{t-1} - K), \zeta_t$. *Therefore:*

$$AP_1 + B + CR_1 = 0 \tag{57}$$

$$GP_1 + H + JR_1P_1 + IR_1 = 0 \tag{58}$$

$$AP_2 + CR_2 + D = 0 \tag{59}$$

$$(G + JR_1)P_2 + JR_2N + IR_2 + LN + M = 0 \tag{60}$$

- *Four equations, four unknowns* P_1, P_2, R_1, R_2.

From (57), we have $R_1 = -\frac{1}{C}(AP_1 + B)$. *Using it in (58), we have*

$$P_1^2 + \left(\frac{B}{A} + \frac{I}{J} + \frac{GC}{JA}\right)P_1 + \frac{IB - HC}{JA} = 0 \qquad (61)$$

This is a quadratic equation in P_1*: there are two solutions.*

$$P_1 = -0.5\left(-\frac{B}{A} - \frac{I}{J} + \frac{GC}{JA} \pm \left(\left(\frac{B}{A} + \frac{I}{J} + \frac{GC}{JA}\right)^2 - 4\frac{IB - HC}{JA}\right)^{0.5}\right) \quad (62)$$

One value gives you a stable solution; one an unstable one. Pick the stable solution, plug it into the solution for R_1. Also from (59)-(60):

$$P_2 = \frac{-D(JN + I) + CLN + CM}{AJN + AI - CG - CJR_1} \quad (63)$$

$$R_2 = \frac{-ALN - AM + DG + DJR_1}{AJN + AI - CG - CJR_1} \quad (64)$$

• In general, the solution for P_1 can not be found analytically since P_1 is a matrix.

• In this case P_1 is found by solving a generalized eigenvalue problem (QZ decompostion), see e.g. Uhlig, 1999 or Klein, 2001.

General form of the solution (X=states, Y= controls, Z= shocks)

$$X_t = P_1 X_{t-1} + P_2 Z_t \quad (65)$$

$$Y_t = R_1 X_{t-1} + R_2 Z_t \quad (66)$$

3.1) If want a log-linear solution (rather than a linear one), use:

$$\begin{aligned} c^s * (c_t - c^s)/c^s &= m_5\zeta_t + (\alpha c^s)(K_{t-1} - K^s)/K^s + (m_6 * N)(N_t - N^s)/N^s \quad (67) \\ (c_t - c^s)/c^s &= E_t[(c_{t+1} - c^s)/c^s + m_2\zeta_{t+1} + (m_3 * K^s)(K_t - K^s)/K^s \\ &+ (m_4 * N^s)(N_{t+1} - N^s)/N^s] \quad (68) \\ c^s * (c_t - c^s)/c^s &= -K^s * (K_t^s - K^s)/K^s + m_7\zeta_t + (m_8 * K^s)(K_{t-1} - K^s)/K^s \\ &+ (m_9 * N^s)(N_t - N^s)/N^s \quad (69) \end{aligned}$$

and solve for $\hat{c}_t = (c_t - c^s)/c^s, \hat{N}_t = (N_t - N^s)/N^s, \hat{K}_t = (K_t - K^s)/K^s$.

The solution procedure is unchanged.

4) To find a second order approximation, guess policy functions (solutions) of the form

$$(K_t - K^s) = P_1(K_{t-1} - K^s) + P_2\zeta_t + P_3(K_{t-1} - K^s)^2 + P_4(K_{t-1} - K^s)\zeta_t + P_5\zeta_t^2 + P_6\sigma^2$$

$$(N_t - N^s) = R_1(K_{t-1} - K^s) + R_2\zeta_t R_3(K_{t-1} - K^s)^2 + R_4(K_{t-1} - K^s)\zeta_t + R_5\zeta_t^2 R_6\boldsymbol{\Sigma}^2$$

- Plug these guesses in (52)-(53) and eliminate expectations.

- Solve sequentially (i.e. first order approximation first and then once P_1, P_2, R_1, R_2 are found, solve for $P_3, P_4, P_5, P_6, R_3, R_4, R_5, R_6$).

- For details see Schmitt-Grohe and Uribe (2004).

5) General variable transformations. Look for solutions of the form:

$$K_t^{\gamma} - K_0^{\gamma} = \psi_1(K_{t-1}^{\nu} - K_0^{\nu}) + \psi_2\zeta_t \tag{70}$$
$$N_t^{\eta} - N_0^{\eta} = \psi_3(K_{t-1}^{\nu} - K_0^{\nu}) + \psi_4\zeta_t \tag{71}$$

where K_0, N_0, c_0 *are pivotal points.*

- If $\mu, \gamma, \eta, \nu = 1$ *linear approximation. If* $\mu, \gamma, \eta, \nu \to 0$ *log-linear approximation*

- How do we choose μ, γ, η, ν*? Use a grid. Find the values that minimize the Mean square of the error in the Euler equation*

Generally $\gamma \approx \nu$ *is optimal in this model. Thus, a linear policy function for* K_t *is close to the best.*

3.4 Running Dynare

- Save the file with the code with extension *.mod (e.g. RBC.mod)

- After you have installed dynare, and made sure to have added the dynare path to matlab, in the matlab window type:

dynare RBC.mod

- This runs the program and creates a bunch of matlab files you can use to check what the program is doing and use later on. The Matlab file that Dynare creates will have the same name as you original file but with the extension *.m (i.e. rbc.m).

3.5 Making sense of dynare output

Suppose $\alpha = 0.36, \beta = 0.99, \delta = \phi = 1.0$. Then the log model delivers:

POLICY AND TRANSITION FUNCTIONS

	k	z	c
constant	-1.612034	0	-1.021010
k(-1)	0.36	0	0.36
z(-1)	0.95	0.95	0.95
e	1	1	1

- Recall that the policy functions that dynare generates are of the form

$$\tilde{k}_t = ln(\alpha\beta) + \alpha(\tilde{k}_{t-1} - ln(\alpha\beta)) + \rho(\tilde{z}_{t-1} - \bar{z}) + \epsilon \quad (72)$$

$$\tilde{c}_t = ln(1-\alpha\beta) + \alpha(\tilde{k}_{t-1} - ln(\alpha\beta)) + \rho(\tilde{z}_{t-1} - \bar{z}) + \epsilon \quad (73)$$

$$\tilde{z}_t = \bar{z} + \rho(\tilde{z}_{t-1} - \bar{z}) + \epsilon_t \quad (74)$$

- "Constant" is the steady state of the variables.

- $k(-1)$ is really $(\tilde{k}(-1) - \bar{\tilde{k}})$ and $z(-1)$ is really $(\tilde{z}(-1) - \bar{\tilde{z}})$ (note that $\bar{\epsilon} = 0$ by assumption)

- Not the best way to write solution since what matters is $\tilde{z}_t$.

3.6 Fine tuning

- In the level and the log solutions, both the equations for consumption and investment are linearized.

- Makes life easy, especially in large systems, but creates unneeded approximation errors.

- Alternative: use the budget constraint to solve for c_t from the Euler equation and approximate only the decision rule for k_t. Decision rule for c_t is found (nonlinearly) from the budget constraint.

FOC

$$c_t^{-\phi} = \beta E_t[c_{t+1}^{-\phi}(\alpha exp(\tilde{z}_{t+1})k_t^{\alpha-1} + (1-\delta))] \quad (75)$$

$$c_t + k_t = exp(\tilde{z}_t)k_{t-1}^{\alpha} + (1-\delta)k_{t-1} \quad (76)$$

$$\tilde{z}_t = \rho\tilde{z}_{t-1} + \epsilon_t \quad (77)$$

New FOC (solving c_t from (76) and plug it into (75)).

$$(exp(\tilde{z}_t)k_{t-1}^{\alpha} + (1-\delta)k_{t-1} - k_t)^{-\phi} = \quad (78)$$

$$\beta E_t[(exp(\tilde{z}_{t+1})k_t^{\alpha} + (1-\delta)k_t - k_{t+1})^{-\phi}(\alpha exp(\tilde{z}_{t+1})k_t^{\alpha-1} + (1-\delta))]$$

$$\tilde{z}_t - \rho\tilde{z}_{t-1} - \epsilon_t = 0 \quad (79)$$

4 Two exercises

Exercise 4.1 *(rosen.mod) Write a Dynare code to solve the following four equations (non-optimizing) model*

$$\begin{aligned} s_t &= a_0 + a_1 P_t + e^s_t && (80) \\ N_t &= (1-\delta)N_{t-1} + s_{t-k} && (81) \\ N_t &= d_0 - d_1 W_t + e^d_t && (82) \\ P_t &= (1-\delta)\beta P_{t+1} + \beta^4 W_{t+4} && (83) \end{aligned}$$

The first equation is the flow supply of new engineers,; the second time to school for engineers; the third the demand for engineers and the last the present value of wages of an engineer.

The parameters are $(a_0, a_1, \delta, d_0, d_1, \beta)$*. The endogenous variables are* $(s_t, N_t,\ P_t, W_t)$ *and the exogenous variables are* (e^s_t, e^d_t)*.*

Pick (reasonable) values for the parameters, for the standard deviation of the shocks and choose initial conditions. Simulate 500 data points (and save them) and compute impulse responses to the two shocks.

Exercise 4.2 *(growth_wls_log.mod) In a basic RBC model agents maximize:*

$$\max E_t \sum_t \beta^t \left(\frac{c_t^{1-\phi} - 1}{1-\phi} + \frac{h_t^{1+1/\nu}}{1+1/\nu}\right) \tag{84}$$

subject to

$$c_t + k_t = exp(\tilde{z}_t) k_{t-1}^{\alpha} h_t^{1-\alpha} + (1-\delta) k_{t-1} \tag{85}$$

where $\ln z_t \equiv \tilde{z}_t = \rho \tilde{z}_{t.-1} + \epsilon_t, \epsilon_t \sim (0, \sigma_e^2)$.

Note when $\nu = 0$*, utility does not depend on* h_t*; when* $\nu = \infty$*, utility is linear in* h_t *(Hansen utility).*

Calculate the FOC conditions, the steady states and write a Dynare code for solving the model using as the endogenous variables are $(lnc_t, lnk_t, lnh_t, lnz_t)$*, as the exogenous innovation is* e_t *and as parameters* $(\beta, \rho, \alpha, \phi, \delta, \nu, \sigma_e)$*.*

Pick reasonable values for the parameters and the initial conditions and solve the model using a linear and a quadratic approximation. Compare the decision rules.

5 Some Tips

• Solution programs (e.g. Uhlig, Klein, sims) not very transparent. May as well use Dynare - can input FOC directly without(log-)linearizing by hand.

• Better to find the decision rules for states and then use the model equations to compute other variables, e.g. $i_t = k_t - (1-\delta)k_{t-1}$ or $y_t = z_t k_t^\alpha$, once you have simulated k_t and z_t. Approximation error is smaller.

• Dynare can be nested into Matlab programs. Careful because Dynare clear the memory before starting (so if there are some matlab calculations before, the results will be cancelled). To avoid this use

dynare rbc.mod noclearall

• Matlab commands can be inserted into Dynare files

Example 5.1 *load ypr.dat;*

```
[nobs,nvar] = size(ypr);
y = ypr(:,1);
p = ypr(:,2);
r = ypr(:,3);

// detrend y, remove linear trend
tr = (1:nobs)';
cr = ones(nobs,1);
yt = ols(y,[cr tr]);
x = y-yt.beta(1)*tr;
// statistics on the time series
stats_r = [mean(r) std(r) min(r) max(r)];
```

```
inv.cnames = strvcat('mean','std','min','max');
inv.rnames = strvcat('series','r');
disp('*** Summary Statistics of time series ***')
disp('—————————————————————- ')
mprint([stats_r],inv)

var r x p g u eer;
varexo eg eu er;
parameters phi betta nu psir psix psip rhog rhou sigg sigu sigr;
⋮
```

Repetitive work

• Dynare is not setup to do repeated experiments with different parameter values to check, e.g., how certain responses change with the parameters.

- From version 4.2 there is a possibility of running the same model for a number of countries (if needed) but it is a bit cumbersome and does not cover all the cases of interest.

- Essentially you would have to run the program many times if you need to do sensitivity analysis. One alternative is to nest the dynare program inside a MATLAB loop. An example of how this is done is the following:

Example 5.2 *(examining the sensitivity of the results to changes in the persistence of the technology shocks)*

```
for n = 1:n_draws;
var c, k, lnz;
varexo e;

low_rho = 0.90; up_rho = 0.99;
rho_ = low_rho + rand.*(up_rho-low_rho);

parameters beta, rho, alpha, phi, delta, sig;

alpha = 0.36; beta = 0.99; delta = 0.10; sig = 0.01; phi=3.0 ; rho=rho_;
⋮
end;
```

Summary statistics

• Dynare produces a lot of output. If you just care about the decision rules use the options: nocorr, nomoments, IRF=0 in stoch_simul command.

• Dynare is set up to produce conditional and unconditional moments of the data, as generated by the decision rules. These moments are computed analytically, unless you specify the HP_filter option in stoch_simul.

In particular it will produce the following unconditional moments:

- Mean and standard deviation of the endogenous variables.

- Contemporaneous correlation of the endogenous variables.

- Autocorrelation function (ACF) of the endogenous variables.

If you want the ACF of a variable like output which is not solved for in the model you will have to compute it by hand. The alternative is to include another equation in the model with the production function to compute output moments directly

Dynare also produce, if requested, impulse responses to each of the shocks and the variance decomposition. The current version is not yet set to do historical decompositions (which would allow you to see, for example, if the current level of consumption in difference form the steady state is due to one shock or another).

- Impulse responses are typically calculated as the difference of two paths, one where $\epsilon_t = 0, \forall t$, one where $\epsilon_1 = 1, \epsilon_t = 0, \forall t > 1$. This is OK for first order approximations. It is not the right way to compute impulse responses in higher order approximations.

- There an option in stoch_simul command called "replication=number" which computes impulse responses correctly for higher order approximations.

5.1 Computing impulse responses

- In first order approximation, the initial condition, the sequence of shocks, their sign and size does not matter. Thus

i) set the initial condition at the steady state,

ii) set the shocks to zero in one run,

iii) set the shocks to zero except the first period (normalized to 1) in the second run.

iv) take the difference between the path of the variables of interest generated in iii) and ii).

When you consider a second order approximation responses must be computed as follows. Dynare gives you the decision rules (say for k_t, z_t).

Part I

- Draw n initial conditions for k_0 and z_0.

- Draw a m sequences of shocks $\epsilon_{t=0}^{T}$ for each k_0, z_0

- Construct paths for z_t, k_t for each k_0, z_0 and each ϵ_t sequence using the decision rules.

- Average the paths for z_t, k_t

Part II

- Draw n initial conditions for k_0 and z_0.

- Set $\epsilon_{t=0}^{T} = 0$ for each k_0, z_0.

- Construct paths for z_t, k_t for each k_0, z_0.

- Average the paths for z_t, k_t.

Take the difference in the paths for z_t, k_t you have in part I and part II

- This procedure is valid for both first and second order approximations.

- Dynare takes the initial conditions as fixed (I guess at the steady state, if the steady command precedes stoch_simul). Thus the replication=number option refers to the number of paths for ϵ_t generated in part I.

- Impulse responses computed with second order approximations are tricky. Paths may explode if the ϵ_t sequence contains, by chance, large numbers (non-negligible probability). Typically you either you throw away exploding paths or you use a simulation procedure called "pruning".

- Pruning simulates components of high order approximation by steps, i.e. first simulates first order terms, then second order, etc. If we put enough restrictions on the simulated realization of first order terms, then second order terms are not exploding (see Kim, Kim, Schaumburg and Sims (2008)). Dynare uses pruning to compute impulse responses.

5.2 Variance decompositions

- The variance decomposition is computed with the option conditional_variance_decomposition=number of stoch_simul. You need to specify the horizon where you want the decomposition via ´´number´´.

- What this option produces is var($y_{t+j}|I_t$) where I_t is the information set available at t. When j is large, the variance decomposition is unconditional.

- If you are interested in the variance decomposition at different horizons you can specify the option as conditional_variance_decomposition=[number1, number2].

6 Accuracy of the solution

- There are situations when second order approximations are needed (e.g. welfare calculations).

- In other cases, one has to decide which approximation to take and whether level or log approximations should be used.

- Check 1: Run dynare for both approximations. Simulate a path for the endogenous variables. Are the paths similar? Are they different? In what are they different?

- Check 2: Run formal accuracy tests (Den Haan and Marcet (1994) or Judd (2004)).

Informal approach

- Understand properties of model/ algorithm.

- Change parameter values, see how output changes.

- Change approximation methods. Solve all the variables inside Dynare. Compare with solution obtained solving only for the states.

- **Needs a lot of time. But it is worthy - especially if you want to calibrate or later estimate the model**.

Basic idea of accuracy tests

Theory $E(f(x_t, x_{t-1}, y_t, y_{t+1})|I_t) = 0$.
Then $E(f(x_t, x_{t-1}, y_t, y_{t+1})h(z_t)'|I_t) = 0$ where $z_t \in I_t$ and $h(z)$ a continuous function of z_t.

Test: $f(x_t, x_{t-1}, y_t, y_{t+1})$ should be white noise, i.e. for any simulated path for (z_t, x_t, y_t) correlation between $f(x_t, x_{t-1}, y_t, y_{t+1})$ and anything in the info set should be zero.

This test should be tried for many drawing of shocks - the path of z_t, x_t, y_t depends on ϵ_t.

Example 6.1 *Euler equation with CRRA utility:*

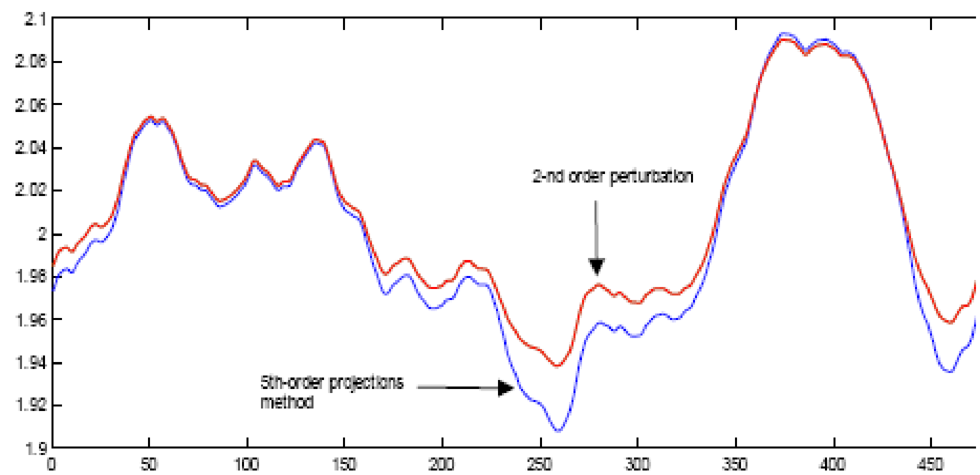
$$c_t^{-\phi} = \beta E_t[c_{t+1}^{-\phi}(\alpha z_t k_t^{\alpha-1} + (1-\delta))]$$

Euler error

$$u_t = c_t^{-\phi} - \beta c_{t+1}^{-\phi}(\alpha z_t k_t^{\alpha-1} + (1-\delta))$$

Then, for example, $\sum_t \frac{1}{T} u_t c_{t-1}$ *should be closed to zero using simulated paths for* (c_t, k_t, z_t)*.*

Formally, you can compare $Q = (\frac{1}{T}\sum_t u_t c_{t-1})^2$ to a $\chi^2(1)$



- Paths very similar.

- Main different is magnitude of recessions.

Alternatives

- Euler equation errors approach: compute $E_t f(x_t, x_{t-1}, y_t, y_{t+1})h(z_t)'$ numerically using (many) simulated paths

If you repeat experiment for many ϵ_t could construct a distribution of Q and compare it with a χ^2 distribution using the Kolmogorov-Smirnov statistic.

- Welfare based approximations: compute welfare assuming $k_t = k_s$ and the optimal policy. Careful: the numbers here are typically small.

7 Forecasting

- You can do two types of forecasting in Dynare

- Unconditional forecasting, with the command forecast.

The options are the horizon of the forecasts(periods=number); and the confidence intervals of the forecasts (conf_sig=number).

The command forecast must follow stoch_simul command.

- **If the steady command precedes stoch_simul command then the initial conditions of the forecasts are the steady state. If the steady command is not used, the initial conditions of the forecast are the initval you use.**

Note that the two need not to produce the same forecasts since in the second case, you may be simply tracing out the dynamics out of the steady states.

- Forecasts are stored in oo_.forecast.xxx.variablename. xxx is the Mean, HPDinf, HPDsup, where the last two are the upper and lower limits given in conf_sig option.

- Conditional forecasts can be computed with three sequential commands: conditional_forecast_paths; conditional_forecasts; plot_conditional_forecast.

- Conditional_forecast_paths specify the variables which are constrained, the constrained value and the number of periods the variable is constrained.

Example 7.1 *var y k c;*
varexo e u;
⋮
stoch_simul;
conditional_forecast_paths;
var y;
periods 1:3 ;
values 2;
end;

The conditional_forecast command computes the conditional forecasts. Options are:

- parameter_set: tells dynare which set of parameters can be used. Here we will use parameter_set=calibration (This option is working only in the versions above 4.2.3);

- controlled_varexo tells dynare which exogenous variable needs to change to insure that the path for the endogenous variable is the required one (in the above example it could be either e, or u, or both);

- replic controls how many Monte Carlo replications will be computed (default=5000); the other two options (period=number, conf_sig=number) are the same as in the forecast command.

- The results of the conditional forecasting exercise are not automatically plotted. If you want to see them use the command plot_conditional_forecast. You can control how many periods you want to plot using the option periods = number.

Example 7.2 *conditional_forecast(parameter_set = calibration, controlled_varexo = (e), replic = 3000, conf_sig=0.95, periods=40);*

plot_conditional_forecast(periods = 10) e c;

Alternatives to Dynare:

- IRIS (J. Benes, IMF); http://www.iris-toolbox.com/

- YADA (A. Warne, ECB); http://www.texlips.net/yada/

Solution to exercise 4.2

FOC

$$c_t^{-\phi} = \beta E_t[c_{t+1}^{-\phi}(\alpha exp(\tilde{z}_{t+1})(\frac{k_t}{h_{t+1}})^{\alpha-1} + (1-\delta))] \tag{86}$$

$$h_t^{1/\nu} = c_t^{-\phi}(1-\alpha)exp(\tilde{z}_t)(\frac{k_{t-1}}{h_t})^{\alpha} \tag{87}$$

$$c_t + k_t = exp(\tilde{z}_t)k_{t-1}^{\alpha}h_t^{1-\alpha} + (1-\delta)k_{t-1} \tag{88}$$

$$\tilde{z}_t = \rho\tilde{z}_{t-1} + \epsilon_t \tag{89}$$

In the steady states ($\bar{z} = 0$) the FOC are

$$\bar{c}^{-\phi} = \beta\bar{c}^{-\phi}(\alpha(\frac{\bar{k}}{\bar{h}})^{\alpha-1} + 1 - \delta) \tag{90}$$

$$\bar{h}^{1/\nu} = \bar{c}^{-\phi}(1-\alpha)(\frac{\bar{k}}{\bar{h}})^{\alpha} \tag{91}$$

$$\bar{c} + \bar{k} = \bar{k}^{\alpha}\bar{h}^{1-\alpha} + (1-\delta)\bar{k} \tag{92}$$

From the first equation we have a solution for $\frac{\bar{k}}{\bar{h}}$ as a function of the parameters.

The other two equations determine $\bar{c}$ and $\bar{h}$

$$\bar{c} = \bar{h}(\frac{\bar{k}}{\bar{h}})^{\alpha} - \delta\bar{k}) \quad (93)$$

$$\bar{h}^{1/\nu} = (1-\alpha)(\bar{c})^{-\phi}(\frac{\bar{k}}{\bar{h}})^{\alpha} \quad (94)$$

The equations to put into Dynare are:

$$\begin{aligned}
\exp(-\phi lnc_t) &= \beta(\exp(-\phi lnc_{t+1})) * (\exp(lnz_{t+1})\alpha \\
&* \exp((\alpha-1)(lnk_t - lnh_{t+1})) + 1 - \delta) && (95) \\
\exp(lnh_t/\nu) &= \exp(-\phi lnc_t)(1-\alpha) * \exp(lnz_t) + \alpha(lnk_{t-1} - lnh_t) && (96) \\
\exp(lnc_t) + \exp(lnk_t) &= \exp(lnz_t + \alpha lnk_{t-1} + (1-\alpha)lnh_t) \\
&+ (1-\delta)exp(lnk_{t-1}) && (97) \\
lnz_t &= \rho lnz_{t-1} + e && (98)
\end{aligned}$$

where the endogenous variables are $lnc_t, lnk_t, lnh_t, lnz_t$, the exogenous innovation is e_t and the parameters $\beta, \rho, \alpha, \phi, \delta, \nu, sig_e$.