INTRODUCTION TO DYNARE

ICE 2005

Michel Juillard

CEPREMAP, Paris Sciences Economics, University Paris 8

Acknowledgments

DYNARE started at CEPREMAP in 1994.

DYNARE development: S. Adjemian, O. Kamenik

Built on work of: R. Boucekkine, F. Collard, J.P. Laffargue, M. Ratto, F. Schorfheide, C. Sims, R. Wouters

Public domain software: cygwin, gnumex, lapack, styxbox, asamin, asa.

DYNARE

1. computes the steady state of the model
2. computes the solution of deterministic models (arbitrary accuracy)
3. computes first and second order approximation to solution of stochastic models
4. estimates (maximum likelihood or Bayesian approach) parameters of DSGE models

Solution of deterministic models

- based on work of Laffargue, Boucekkine and myself
- approximation: impose return to equilibrium in finite time instead of asymptotically
- computes the trajectory of the variables numerically
- uses a Newton–type method
- usefull to study full implications of non–linearities

An example

The effect of a change in tax rate in a model with monopolistic competition (adapted from Hairault, Langot and Portier, 2001)

$$
\begin{aligned}
W_t &= \ln c_t + \eta \ln(1-h_t) + \beta W_{t+1} \\
c_t + i_t &= \bar{A} k_{t-1}^{\alpha} h_t^{1-\alpha} \\
i_t &= k_t - (1-\delta) k_{t-1} \\
\frac{1}{c_t} &= \beta E_t \left\{ \frac{1}{c_{t+1}} (z_{t+1} + 1 - \delta) \right\} \\
\frac{\eta}{1-h_t} &= \frac{w_t}{c_t} \\
\alpha \left(\frac{k_{t-1}}{h_t} \right)^{\alpha-1} &= (1+\mu)(1+\tau_t) z_t \\
(1-\alpha) \left(\frac{k_{t-1}}{h_t} \right)^{\alpha} &= (1+\mu)(1+\tau_t) w_t
\end{aligned}
$$

DYNARE implementation – Preambule

```
var Welf w c h i k z;
varexo tau;

parameters beta delta alpha mu eta rho Abar;
delta = 0.025;
eta = 2;
mu = 0.1;
alpha = 0.36;
rho = 0.95;
beta = 0.988;
Abar = 1;
```

DYNARE implementation – Model

```
model;
Welf = log(c)+eta*log(1-h)+beta*Welf(+1);
c+i = Abar*k(-1)^alpha*h^(1-alpha);
i = k - (1-delta)*k(-1);
1/c = beta*(1/c(+1))*(z(+1)+1-delta);
eta/(1-h) = w/c;
alpha*(k(-1)/h)^(alpha-1) = (1+mu)*(1+tau)*z;
(1-alpha)*(k(-1)/h)^alpha = (1+mu)*(1+tau)*w;
end;
```

DYNARE implementation – Initialization

```
initval;
Welf = -100;
w = 0.5;
c = 0.6;
h = 0.3;
i = 0.4;
k = 3;
z = 0.1;
tau = 0;
end;

steady;
```

DYNARE implementation – Initialization

```
endval;
Welf = -100;
w = 0.5;
c = 0.6;
h = 0.3;
i = 0.4;
k = 3;
z = 0.1;
tau = -mu/(1+mu);
end;

steady;
```

DYNARE implementation – Computation

```
simul(periods=300);

dsample 0 50;
rplot Welf;
rplot k;
rplot c;
rplot h;
```

Stochastic models: First order approximati

In a a stochastic framework, the unknowns are the decision functions.
For a large class of DSGE models, DYNARE computes approximated decision rules and transition equations of the form

$$y_t = \bar{y} + A\hat{y}_{t-1} + Bu_t$$

with $\hat{y}_t = y_t - \bar{y}$.
Method proposed by Klein (2000) and Sims (2002).
DYNARE computes also theoretical moments and IRFs.

Example

In the previous example, one introduces stochastic productivity according to

$$\ln A_t = (1-\rho) * \ln \bar{A} + \rho * \ln A_{t-1} + e_t$$

and one considers the case of no tax.
New instructions:

```
shocks;
var e; stderr 0.072;
end;

stoch_simul(order=1) Welf h c i w z;
```

Second order approximation

Two features:

- decision rules and transition functions are 2nd order polynomials
- departure from certainty equivalence: the variance of future shocks matters

Decision rules and transition equations of the form

$$y_t = \bar{y} + A\hat{y}_{t-1} + Bu_t + 0.5\left(\hat{y}'_{t-1}C\hat{y}_{t-1} + u'_t D u_t\right) + \hat{y}'_{t-1}Fu_t + \Delta\left(\Sigma_u\right)$$

Method suggested by K. Judd, developped by C. Sims (2002), S. Schmitt-Grohe and M. Uribe (2003), F. Collard and M. Juillard (2000).

A k-order Perturbation Approach to Solve Complete Market RBC Models

July 2004

prepared for the conference “Computational Methods for Dynamic Stochastic Economic Models”, SITE 2004.

Michel Juillard
CEPREMAP and University Paris 8

Solving DSGE models

Let's consider the following model:

$$E_t\left(f\left(y_{t-1}, y_t, y_{t+1}, u_t\right)\right) = 0$$

with

$$u_t = \sigma\varepsilon_t, \qquad E(\varepsilon_t) = 0, \qquad E\left([\varepsilon_t]^{\beta_1}\ldots[\varepsilon_t]^{\beta_k}\right) = [\Sigma]^{\beta_1\ldots\beta_k}$$

The solution takes the form:

$$y_t = g(y_{t-1}, u_t, \sigma)$$

The perturbation method

- Computes a Taylor expansion for $g()$ from the coefficients of the Taylor expansion of $f()$.
- The Taylor expansion is generaly computed around the deterministic equilibrium of the model:

$$f(\bar{y}, \bar{y}, \bar{y}, 0, 0) = 0$$

The state variables

The state variables are y_{t-1}, and u_t.
Then,

$$\begin{aligned} y_{t+1} &= g(y_t, u_{t+1}, \sigma) \\ &= g(g(y_{t-1}, u_t, \sigma), u_{t+1}, \sigma) \end{aligned}$$

and

$$\begin{aligned} F(y_{t-1}, u_t, \sigma, u_{t+1}) &= f\Big(y_{t-1}, g(y_{t-1}, u_t, \sigma), \\ &\qquad g(g(y_{t-1}, u_t, \sigma), u_{t+1}, \sigma), u_t\Big) \\ \bar{F}(y_{t-1}, u_t, \sigma) &= E_t\Big(F(y_{t-1}, u_t, \sigma, u_{t+1})\Big) \\ &= 0 \end{aligned}$$

The first order approximation

First order Taylor expansion of the structural model:

$$
\begin{aligned}
\bar{F}^{(1)}(y_{t-1}, u_t, \sigma) &= E_t\Big\{ f(\bar{y}, \bar{y}, \bar{y}, 0, 0) + f_{y_-}\hat{y} \\
&+ f_y(g_y\hat{y} + g_u u + g_\sigma \sigma) \\
&+ f_{y_+} g_y (g_y \hat{y} + g_u u + g_\sigma \sigma) \\
&+ \boxed{f_{y_+} g_u u'} + f_{y_+} g_\sigma \sigma + f_u u \Big\} \\
&= 0
\end{aligned}
$$

where $\hat{y} = y_{t-1} - \bar{y}$, $u = u_t$, $u' = u_{t+1}$. The partial derivatives are taken at the deterministic equilibrium and aren't stochastic.

for this to hold . . .

$$\begin{aligned}\left(f_{y_-} + f_y g_y + \boxed{f_{y_+} g_y g_y}\right)\hat{y} &= 0\\ (f_y g_u + f_{y_+} g_y g_u + f_u)\, u &= 0\\ (f_y g_\sigma + f_{y_+} g_y g_\sigma + f_{y_+} g_\sigma)\, \sigma &= 0\end{aligned}$$

k order approximation

Let's write

$$
\begin{aligned}
s &= [y_{t-1}, u_t]' \\
\bar{s} &= [\bar{y}, 0]' \\
\hat{s} &= s - \bar{s} \\
u &= u_t \\
u' &= u_{t+1}
\end{aligned}
$$

Tensor notation

$$\frac{\partial^j F^i}{\partial s_{\alpha_1} \ldots \partial s_{\alpha_j}} = \left[F^i_{s^j}\right]_{\alpha_1 \ldots \alpha_j}$$

and

$$\sum_{\alpha_1}^{n} \cdots \sum_{\alpha_j}^{n} \frac{\partial^j F^i}{\partial s_{\alpha_1} \ldots \partial s_{\alpha_j}} \hat{s}_{\alpha_1} \ldots \hat{s}_{\alpha_j} = \left[F^i_{s^j}\right]_{\alpha_1 \ldots \alpha_j} [\hat{s}]^{\alpha_1} \ldots [\hat{s}]^{\alpha_j}$$

Taylor expansion of the model

$$
\begin{aligned}
F_i^{(p)}(s,\sigma,u') \\
= \quad & F_i(\bar{s},0,0) + \sum_{j=1}^{p} \frac{1}{j!} \Big(\left[F^i_{s^j}\right]_{\alpha_1\ldots\alpha_j} [\hat{s}]^{\alpha_1} \ldots [\hat{s}]^{\alpha_j} \\
& + \sum_{k=1}^{j-1} \left[F^i_{s^k\sigma^{j-k}}\right]_{\alpha_1\ldots\alpha_k} [\hat{s}]^{\alpha_1} \ldots [\hat{s}]^{\alpha_k} \sigma^{j-k} \\
& + \sum_{k=1}^{j-1} \left[F^i_{s^k u'^{j-k}}\right]_{\alpha_1\ldots\alpha_k\beta_1\ldots\beta_{j-k}} [\hat{s}]^{\alpha_1} \ldots [\hat{s}]^{\alpha_k} [u']^{\beta_1} \ldots [u']^{\beta_{j-k}} \\
& + \sum_{k=1}^{j-1} \left[F^i_{u'^k\sigma^{j-k}}\right]_{\beta_1\ldots\beta_k} [u']^{\beta_1} \ldots [u']^{\beta_k} \sigma^{j-k} \\
& + \sum_{k=1}^{j-2} \sum_{m=k+1}^{j-1} \left[F^i_{s^k u'^{m-k}\sigma^{j-m}}\right]_{\alpha_1\ldots\alpha_k\beta_1\ldots\beta_{m-k}} [\hat{s}]^{\alpha_1} \ldots [\hat{s}]^{\alpha_k} [u']^{\beta_1} \ldots [u']^{\beta_{m-k}} \sigma^{j-m} \\
& + \left[F^i_{\sigma^j}\right] \sigma^j + \left[F^i_{u'^j}\right]_{\beta_1\ldots\beta_j} [u']^{\beta_1} \ldots [u']^{\beta_j} \Big)
\end{aligned}
$$

Reminding

$$\begin{aligned} u &= \sigma\varepsilon \\ E\left([u]^{\beta_1}\dots[u]^{\beta_k}\right) &= \sigma^k\,[\Sigma]^{\beta_1\dots\beta_k} \end{aligned}$$

Taking the expectation

$$
\begin{aligned}
& E\left(F_i^{(p)}(s,\sigma)\right) \\
= \quad & F_i(\bar{s},0,0) + \sum_{j=1}^{p} \frac{1}{j!} \Big(\left[F^i_{s^j}\right]_{\alpha_1\ldots\alpha_j} [\hat{s}]^{\alpha_1} \ldots [\hat{s}]^{\alpha_j} \\
& + \sum_{k=1}^{j-1} \left[F^i_{s^k\sigma^{j-k}}\right]_{\alpha_1\ldots\alpha_k} [\hat{s}]^{\alpha_1} \ldots [\hat{s}]^{\alpha_k}\, \sigma^{j-k} \\
& + \sum_{k=1}^{j-1} \left[F^i_{s^k u'^{j-k}}\right]_{\alpha_1\ldots\alpha_k\beta_1\ldots\beta_{j-k}} [\hat{s}]^{\alpha_1} \ldots [\hat{s}]^{\alpha_k} [\Sigma]^{\beta_1\ldots\beta_{j-k}}\, \sigma^{j-k} \\
& + \sum_{k=1}^{j-1} \left[F^i_{u'^k\sigma^{j-k}}\right]_{\beta_1\ldots\beta_k} [\Sigma]^{\beta_1\ldots\beta_k}\, \sigma^{j} \\
& + \sum_{k=1}^{j-2} \sum_{m=k+1}^{j-1} \left[F^i_{s^k u'^{m-k}\sigma^{j-m}}\right]_{\alpha_1\ldots\alpha_k\beta_1\ldots\beta_m} [\hat{s}]^{\alpha_1} \ldots [\hat{s}]^{\alpha_k} [\Sigma]^{\beta_1\ldots\beta_{m-k}}\, \sigma^{j-k} \\
& + \left[F^i_{\sigma^j}\right] \sigma^j + \left[F^i_{u'^j}\right]_{\beta_1\ldots\beta_j} [\Sigma]^{\beta_1\ldots\beta_j}\, \sigma^j \Big) \\
= \quad & 0.
\end{aligned}
$$

Constraints on the partial derivatives

$$
\begin{array}{lll}
(1) & \left[F^i_{s^j}\right]_{\alpha_1\ldots\alpha_j} & = \quad 0 \\
(2) & \left[F^i_{s^k\sigma^{j-k}}\right]_{\alpha_1\ldots\alpha_k} + \left[F^i_{s^k u'^{j-k}}\right]_{\alpha_1\ldots\alpha_k\beta_1\ldots\beta_{j-k}} [\Sigma]^{\beta_1\ldots\beta_{j-k}} & \\
 & + \sum_{m=k+1}^{j-1} \left[F^i_{s^k u'^{m-k}\sigma^{j-m}}\right]_{\alpha_1\ldots\alpha_k\beta_1\ldots\beta_m} [\Sigma]^{\beta_1\ldots\beta_{m-k}} & = \quad 0 \\
(3) & \left[F^i_{\sigma^j}\right] + \left[F^i_{u'^j}\right]_{\beta_1\ldots\beta_j} [\Sigma]^{\beta_1\ldots\beta_j} & \\
 & + \sum_{k=1}^{j-1} \left[F^i_{u'^k\sigma^{j-k}}\right]_{\beta_1\ldots\beta_k} [\Sigma]^{\beta_1\ldots\beta_k} & = \quad 0
\end{array}
$$

$$j = 1, \ldots, p \qquad k = 1, \ldots, j-1$$

F as a composition of functions

Let's define z as

$$z = \begin{bmatrix} y_{t-1} \\ y_t \\ y_{t+1} \\ u_t \end{bmatrix} = z(y, u, \sigma, u') = \begin{bmatrix} y \\ g(y, u, \sigma) \\ g(g(y, u, \sigma), u', \sigma) \\ u \end{bmatrix}$$

and

$$F(y, u, \sigma, u') = f\left(z(y, u, \sigma, u')\right)$$

kth order derivatives of a composition

Faa Di Bruno formula: if $y = f(z(s))$, then

$$\left[f^i_{s^j}\right]_{\alpha_1 \ldots \alpha_j} = \sum_{l=1}^{j} \left[f^i_{z^l}\right]_{\beta_1 \ldots \beta_l} \sum_{c \subset \mathcal{M}_{l,j}} \prod_{m=1}^{l} \left[z_{s^{\mathcal{N}(c_m)}}\right]^{\beta_m}_{\alpha_{c_m}}$$

where $\mathcal{M}_{l,j}$ is the set of all partitions of the set of j indices with l classes and $\mathcal{N}(c_m)$ is the cardinality of class c_m.
Note that $\mathcal{M}_{1,j} = \{1, \ldots, j\}$ and $\mathcal{M}_{j,j} = \{\{1\}, \{2\}, \ldots, \{j\}\}$.
Good news: The highest order derivatives appear only once and are multiplied by first order derivatives.

Recovering g_{y^j}

Back into matrix notation. The partial derivatives unfold along the columns.

$$F_{y^j} = f_{y+}\left(g_{y^j} \bigotimes_{k=1}^{j} g_y + g_y g_{y^j} \right) + f_y g_{y^j} + D = 0$$

where D is a term depending on partial derivatives of $g()$ of order lower than j and therefor already computed. This requires solving the generalized Sylvester equation

$$(f_{y+}g_y + f_y)\, g_{y^j} + f_{y+} g_{y^j} \bigotimes_{k=1}^{j} g_y = -D,$$

using Kamenik's algorithm.

Recovering other terms in g_{s^j}

$$F_{s^j} = f_{y+} \left(g_{y^j} \bigotimes_{k=1}^{j} g_s + g_y g_{s^j} \right) + f_y g_{s^j} + D = 0$$

where D is a term depending on partial derivatives of $g()$ of order lower than j and therefor already computed. This requires solving the linear system

$$(f_{y+} g_y + f_y)\, g_{s^j} = -D - f_{y+} g_{y^j} \bigotimes_{k=1}^{j} g_s ,$$

Recovering $g_{y^k\sigma^{j-k}}$

Must be solved in decreasing order of k.

$$F_{y^k\sigma^{j-k}} = f_{y+}\left(g_{y^k\sigma^{j-k}} \bigotimes_{\ell=1}^{k} g_y + g_y g_{y^k\sigma^j} \right) + f_y g_{y^j} + D + E = 0$$

where D is a term not depending on $g_{y^k\sigma^{j-k}}$, but on $g_{y^r\sigma^{j-r}}$, for $r > k$ and

$$\begin{aligned}[E]^i_{\alpha_1\ldots\alpha_k} &= \left[F^i_{s^k u'^{j-k}}\right]_{\alpha_1\ldots\alpha_k\beta_1\ldots\beta_{j-k}} [\Sigma]^{\beta_1\ldots\beta_{j-k}} \\ &+ \sum_{m=k+1}^{j-1} \left[F^i_{s^k u'^{m-k}\sigma^{j-m}}\right]_{\alpha_1\ldots\alpha_k\beta_1\ldots\beta_m} [\Sigma]^{\beta_1\ldots\beta_{m-k}}\end{aligned}$$

Recovering $g_{y^k \sigma^{j-k}}$ (continued)

This requires solving the generalized Sylvester equation

$$\left(f_{y_+} g_y + f_y\right) g_{y^k \sigma^{j-k}} + f_{y_+} g_{y^k \sigma^{j-k}} \bigotimes_{\ell=1}^{k} g_y = -D - E.$$

Recovering g_{σ^j}

$$F_{\sigma^j} = f_{y+} g_y g_{\sigma^j} + f_{y+} g_{\sigma^j} + f_y g_{\sigma^j} + D + E = 0$$

where D is a term not depending on g_{σ^j} and

$$[E]^i_{\beta_1 \ldots \beta_j} = \left[F^i_{u'^j}\right]_{\beta_1 \ldots \beta_j} [\Sigma]^{\beta_1 \ldots \beta_j} + \sum_{k=1}^{j-1} \left[F^i_{u'^k \sigma^{j-k}}\right]_{\beta_1 \ldots \beta_k} [\Sigma]^{\beta_1 \ldots \beta_k}$$

This requires solving the linear system

$$\left(f_{y+} g_y + f_{y+} + f_y\right) g_{\sigma^j} = -D - E$$