DYNARE for Macroeconomic Analysis

CAMA Lecture

Philip Liu

philip.liu@anu.edu.au

Centre for Applied Macroeconomic Analysis (CAMA)
Research School of Pacific and Asian Studies
The Australian National University

March 28, 2006

Outline

- what is DYNARE and some features
- the general model
- DYNARE syntax file
- solution of deterministic models
- solution of stochastic models
- estimation
- summary

What is DYNARE?

- DYNARE: A suite of programs for the simulation and estimation of rational expectation models
- Developed by a group of leading applied DSGE researchers headed by Michel Juillard since 1994
- Platforms: **MATLAB**, Scilab and Gauss
- Collection of over 200 functions (MATLAB)
- Widely used by central banks, IMF, academics research and in teaching graduate students
- More information: `http://www.cepremap.cnrs.fr/dynare/`
- Online help/discussion forum available

Features

- Computes the steady state of the D(S)GE model
- Computes the solution of deterministic models
- Computes the first and second order approximation of linear/non-linear stochastic models
- Estimates the parameters of DSGE models using MLE or Bayesian methods
- Computes optimal policy for LQ economies
- Simple regression tool
- Useful checking tool
- No or little programming skills required (That maybe a lie)

The general model

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

- y: vector of endogenous (state and jump) variables
- v: vector of exogenous shocks
- θ: vector of model parameters
- $f(.)$: linear or non-linear function

- Steady state: $\bar{y} = \{y : [f(y_{t+1}, y_t, y_{t-1}, v_t; \theta)] = 0, t \rightarrow \infty\}$
- Solution: $y_t = g(y_{t-1}, v_t; \psi)$
- Compute statistics of interest: $z_t = h(v_t; \psi, y_0, \bar{y})$

An Example: Cho and Moreno (JMCB)

- IS: Euler equation with habit, Fuhrer (2000)

$$y_t = \mu E_t y_{t+1} + (1-\mu) y_{t-1} - \phi(r_t - E_t \pi_{t+1}) + \varepsilon_{IS,t}$$

- Phillips Curve: Sticky price, Calvo (1983)

$$\pi_t = \delta E_t \pi_{t+1} + (1-\delta)\pi_{t-1} + \lambda y_t + \varepsilon_{AS,t}$$

- Taylor rule: Clarida, Gali and Gertler (2000)

$$r_t = \rho r_{t-1} + (1-\rho)[\beta E_t \pi_{t+1} + \gamma y_t] + \varepsilon_{MP,t}$$

- Matrix form: $AX_t = BE_t X_{t+1} + CX_{t-1} + \varepsilon_t,\ \varepsilon_t \sim (0, \Omega)$
- Solution VAR(1): $X_{t+1} = \Gamma_1 X_t + \Gamma_2 \varepsilon_{t+1}$

*.mod VS *.m VS *.mat

- *.mod are DYNARE instruction (Syntax) file containing:
  - variables and parameters declarations
  - model declaration
  - shocks, initial and terminal conditions
  - tasks to be performed, eg: steady, check, forecast, simul, stoch_simul, estimation, olr
  - To run in MATLAB, type:

    ```
    dynare filename.mod
    ```

- *.m are MATLAB program (function) files:

  ```
  function [output]=fcn_name(input)
  ```

  DYNARE has a collection of over 200 m-files
- *.mat are matrices produced by MATLAB

.mod file syntax: basic

```
// # declaration of endogenous variables
var inf, y, r;

// # declaration of exogenous variables
varexo eas, eis, emp;

// # declaration of paramaters
parameters del,lam,mu,phi,roe,bet,gam,sigas,sigis,sigmp;

//parameter values
del=.5586; lam=.0011; mu=.4859; phi=.0045; roe=.8458; bet=1.6409;
gam=.6038; sigas=.4585; sigis=.3734; sigmp=.7327;

// # specification of the model equations
model(linear);
inf-del*inf(+1)-(1-del)*inf(-1)-lam*y-eas;
y-mu*y(+1)-(1-mu)*y(-1)+phi*(r-inf(+1))-eis;
r-roe*r(-1)-(1-roe)*(bet*inf(+1)+gam*y)-emp;
end;
```

.mod file syntax: simulations

```
// # initial and terminal conditions
initval; y=2; r=0; inf=0; end;

endval; y=0; r=0; inf=0; end;

// # specification of shocks
shocks; var eas= sigas^2; var eis=sigis^2; var emp=sigmp^2; end;

// # Tasks
check; simul[(periods=INTERGER)];
// or
stoch_simul[(OPTIONS...)]
    inf y r;
```

.mod file syntax: estimation

```
// or estimation MLE setup
estimated_params;
del, 0.5; //0, 1;
lam, 0.1; // -10, 10;
mu,  0.5; // 0, 1;
phi, 0.1; // -10, 10;
roe, 0.5; // 0, 1;
bet, 1.5; // 1, 10;
gam, 0.5; // 0, 10;

stderr eas, 0.5, 0,100;
stderr eis, 0.5,0,100;
stderr emp, 0.5,0,100;
end;

// observed variables
varobs inf y r;

// There are a lot of other options for estimation
estimation(datafile=cmdata,prefilter=1);
```

Solution of deterministic models

- The model: $AX_t = BX_{t+1} + CX_{t-1}$
- perfect foresight, use ‘initval’ and ‘enval’ to specify shocks
- approximation: impose return to terminal condition in finite time
- `simul(periods=30)`: computes the path of variables over a 30 period horizon by solving all the equations for every period
- Results:
    - eigenvalues of the system, no of eigenvalues >1 should be exactly the same as no of jump variables to ensure unique solution
    - the simulated variables are stored in y_.mat

DYNARE demonstration

- installing DYNARE in MATLAB
- setting up the .mod file
- initializing simulations
- perform simple deterministic simulation of Cho and Moreno model

Take a break now!

Recap and what's next

- what we have talked about
  - what is DYNARE and overview
  - presented the general model DYNARE works with and an example
  - DYNARE syntax file, `mod` vs `m` vs `mat` files
  - solution of deterministic models
  - brief illustration
- what's next
  - solution to stochastic models for linear and nonlinear
  - maximum likelihood estimation
  - Bayesian estimation

Solution: linear stoch. models

- The linear model:

  $AX_t = BE_tX_{t+1} + CX_{t-1} + \varepsilon_t, \ \varepsilon_t \sim (0, \Omega)$

- agents decisions are subject to stochastic shocks but the process of the shocks are known
- agents has rational expectations
- DYNARE uses solution algorithms proposed by Klein (2000) and Sims (2002) - generalized Schur decomposition
- Rewrite model: $A_0E_tY_{t+1} = A_1Y_t + v_{t+1}$, where

  $$Y_t = \begin{bmatrix} X_{t-1} \\ X_t \end{bmatrix}$$

Idea of the algorithm

- perform the generalized Schur (QZ) decomposition of the system
- check for the BK condition
- transform the system into a triangular one and isolate the unstable block
- solve unstable (2nd) block forward
- using the solution of the 2nd block and solve the stable block (1st) backward
- the solution of the system can be written as an VAR(1) $X_t = \Gamma_1 X_{t-1} + \Gamma_2 \varepsilon_t$

Solution: nonlinear models

- The nonlinear model: $E_t \left\{ f(y_{t+1}, y_t, y_{t-1}, v_t; \theta) \right\} = 0$
- 2nd order perturbation methods, Schmitt-Grohe and Uribe (2003)
- departure from certainty equivalence: the variance of shocks matters
- solution is in terms of 2nd order polynomials with variances and cross products
- version 4.0 calculates the derivatives analytically rather than relying on numerical ones every time
- $y_t = Ay_{t-1} + Bv_t + \frac{1}{2}[y_{t-1}' C y_{t-1} + v_t' D v_t] + y_{t-1}' E v_t + \Delta \Sigma_v$

Decision rule coefficients

- $y_t = y^s + Ay_{t-1} + Bv_t + \frac{1}{2}[y'_{t-1}Cy_{t-1} + v'_tDv_t] + y'_{t-1}Ev_t + \Delta\Sigma_v$
- y_t and v_t are both in alphabetical order
  - y^s : `dr_.ys` steady state values
  - Δ : `dr_.ghs2` coef. variance of shocks
  - A : `dr_.ghx` coef. on state variables
  - B : `dr_.ghu` coef. on exogenous variables
  - C : `dr_.ghxx` coef. $\otimes$ of state variables
  - D : `dr_.ghuu` coef. $\otimes$ of exogenous variables
  - E : `dr_.ghxu` coef. cross product of the state and exogenous variables

Stochastic simulations

- you can do lot with `stoch_simul`
- autocorrelation coefficients of (endo): `ar=INTEGER`, `oo.autocorr`
- IRF of shocks: `irf=INTEGER`, VARIABLE_SHOCK
- stochastic simulation of y_t: `periods=INTEGER`
  - simulated variables: VARIABLE
  - mean of endo: `oo.mean`
  - variance of endo: `oo.var`
  - cross correlations

Estimation: MLE

estimates the structural (“deep”) parameters of the model:

1. calculates the steady state
2. linear approximation
3. compute the solution of the linearized model, ie:
   $X_t = \Gamma_1 X_{t-1} + \Gamma_2 \varepsilon_t$
4. construct the Kalman filter system
   State: $X_t = \Gamma_1 X_{t-1} + \Gamma_2 \varepsilon_t, \varepsilon_t \sim N(0, \Omega)$
   Measurement: $Y_t = G X_t + H \mu_t, \mu_t \sim N(0, \Sigma)$
5. evaluate the $\ell(Y^T|\Theta)$ via the Kalman filter
6. max $\ell(Y^T|\Theta, \mathbb{B})$ wrt Θ using Sims csminwel
7. evaluate the 2nd derivative to calculate var-cov(Θ)

Estimation: MLE cont.

- the estimation results are stored in the structure array `oo`
- Cho and Moreno example, US data 1980Q4-2000Q1:

parameters	Estimate	s.d.	t-stat	CM(04)	Estimate(HP)
del	0.5130	0.0168	30.5124	0.5586	0.5056
lam (PC)	0.0047	0.0039	1.1897	0.0011	0.0118ˆ
mu	0.4937	0.0225	21.9797	0.4859	0.5149
phi (IS)	0.0029	0.0039	0.7600	0.0045	0.0018ˆ
roe	0.8502	0.0404	21.0662	0.8458	0.8660
bet	1.3484	0.3492	3.8611	1.6409	1.4059
gam	0.5320	0.3385	1.5716	0.6038	1.2563ˆ
std. dev.	Estimate	s.d.	t-stat		
eas	0.5092	0.0435	11.7167	0.4585	0.5135
eis	0.3605	0.0302	11.9219	0.3734	0.3357
emp	0.7443	0.0599	12.4219	0.7327	0.7309

Estimation cont.

- the other estimation option is using Bayesian methods
  - need to specify the priors
  - the likelihood is evaluated in the same way
  - use MCMC algorithm to simulate the posterior density
  - however choice of prior is important for the result
- Cho and Moreno example:

para	p_mean	p_std	shape	ps_mean	mean	conf.	mle
del	0.500	0.2000	beta	0.4196	0.2111	0.5440	0.5130
lam	0.010	0.0100	norm	0.0081	0.0003	0.0200	0.0047
mu	0.500	0.2000	beta	0.4586	0.3934	0.5158	0.4937
phi	0.010	0.0100	norm	0.0094	0.0007	0.0180	0.0029
roe	0.500	0.2000	beta	0.8380	0.7824	0.8975	0.8502
bet	1.500	0.5000	norm	1.4381	1.0280	1.9021	1.3484
gam	0.500	0.4000	norm	0.5603	0.1541	0.9751	0.5320

Other DYNARE tools

- compute optimal policies for quadratic
  - `olr` computes the optimal policy under commitment for LQ economies
  - `osr` computes the optimal simple rule for LQ economies
- checking tool
  - the `check` command is very useful, displays eigenvalues and check BK condition
  - if you are programming things up you self, useful to check against results before proceeding
- simple regression tool

What DYNARE can do

- specify a DSGE in linear or non-linear form, no need to write out state space matrices
- solves the model for its decision rule (with linear approximation)
- simulate (both deterministic and stochastic) the solution of the model to produce various statistics of interest: moments, IRF, forecasts, etc
- estimate the model's parameters using MLE or Bayesian methods
- compute optimal policies
- built in conditioning statements

What DYNARE CAN'T do

- write your thesis
- solve all your problems without thinking about the problem
- relatively difficult to modify the DYNARE code to produce "non-standard" calculations
- the documentation is still relatively poor at this stage with little explanation on which routines does what
- relatively difficult to detect errors