Dynare++

Ondra Kamenik

Czech National Bank

Dynare++

- Perturbation method to DSGE models having the form

$$E_t[f(y_{t+1}, y_t, y_{t-1}, u_t) = 0],$$

  where y are endogenous variables, u are exogenous variables (shocks)
- No limit to the order of approximation
- Completely standalone
- Input: text file similar to Matlab Dynare
- Output: Matlab MAT-4 file and journal file

Dynare++ Calculations

1. Parses the model file and forms symbolic derivatives
2. Solves deterministic steady state
3. Solves k-order approximation about the deterministic steady state
4. Optional: invokes a multi-step algorithm to get an approximation about the solution's fix point
5. Calculates the fix point
6. Optional: calculates approximation error on ellipse, along simulation path or along shocks
7. Simulates to get an ergodic distribution
8. Optional: simulates impulse responses

Input File

The input file is very similar to Matlab Dynare model file:

- Preamble — declaration of endogenous, exogenous variables and parameters
- Parameter values (might use expressions dependent on other parameters)
- Model section
  - Only lags (-1) and leads (+1) are allowed
  - Be aware of the timing convention!
- Initial values for non-linear solver (can use expressions dependent on other variables and parameters)
- Variance-covariance matrix of the shocks
- Order statement

Output

Contents of Matlab MAT file:

- Derivatives of the decision rule
- Deterministic steady state and fix point(s)
- Impulse response functions
- Ergodic distribution mean and covariance

Contents of journal file:

- Resource usage: time, memory and processors
- Debugging information

Why should I trust it?

- Do not trust it
- I solved a few models with closed form by hand and compared
- I compared against *perturbationAIM* Mathematica binary by G. Anderson et al. (FED)
- Tensor library tested for Faa Di Bruno on nested multidimensional polynomials
- K-order module tested by evaluation of equation residuals
- There is never enough testing

Documentation

Everything available on Dynare site:

- Dynare++ Tutorial
- This presentation
- Documentation to the code describes Dynare++ algorithms
  - Tensor library
  - K-order module
  - Numerical integration module

Dynare++ Modules

- Parser and symbolic derivator and evaluator
- Non-linear solver employing a line search between Cauchy and Newton directions
- Multidimensional tensor library
- Linear algebra and Sylvester module
- k-order module
- Numerical integration module

Tensor Library

- Aim: evaluate multidimensional Faa Di Bruno formula

$$\left[\frac{\partial^k}{\partial x^k} f(z(x))\right]^{\gamma}_{\alpha_1\ldots\alpha_k} = \sum_{l=1}^{k} [f_{z^l}]^{\gamma}_{\beta_1\ldots\beta_l} \sum_{c\in M_{l,k}} \prod_{m=1}^{l} [z_{x^{|c_m|}}]^{\beta_m}_{\alpha(c_m)}$$

- Aim: evaluate tensor polynomials

$$[result]^{\gamma} = \sum_{l=0}^{k} [g_{x^l}]^{\gamma}_{\alpha_1\ldots\alpha_l} \prod_{m=1}^{l} [x]^{\alpha_m}$$

- Faa Di Bruno done by matrix–matrix products and permutation of columns
- Adaptive tensor splitting to fit available memory

Linear Algebra and Sylvester Module

- BLAS interface allows using of hardware tuned linear algebra libraries
- LAPACK interface
- Sylvester module solves

$$AX + BX(C \otimes \ldots \otimes C) = D$$

K-order Module

- First order at deterministic steady state
- Higher orders at deterministic steady state
- Multi-step algorithm to approximate solution at the fix point
- Decision rule simulation (for ergodic distribution and impulse responses)

Numerical Integration Module

For an approximation g the module evaluates the residual

$$\int_{u'} f(g(g(y,u),u'),g(y,u),y,u)\mathrm{d}u'$$

Available options for y and u:

- y, u along some simulation path
- y is an ellipse centered at the fix point
- u varies, y is the fix point

Available algorithms

- Product quadrature (number of shocks < 4)
- Non-nested Smolyak quadrature (shocks ≥ 4)
- (Quasi) Monte–Carlo