MathWorks
FINANCE CONFERENCE 2023

Dynare: Macroeconomic Modelling for All

October 11-12 | Online

Sébastien Villemot, CEPREMAP

Dynare: Macroeconomic Modeling for All

Sébastien Villemot

Dynare

MathWorks Finance Conference
October 12, 2023

Dynare in a nutshell

- MATLAB-based software for solving, simulating, and estimating macroeconomic models
- leading position in both policy institutions and academia
- originally for Dynamic Stochastic General Equilibrium models (DSGE), but also works for other theoretical paradigms
- user-friendly
- free/libre and open-source software
- community-driven

What Dynare brings to the community

- Easy description of macroeconomic models and operations on it, through a domain-specific language (DSL)
- Better focus on economic modeling task, without the need of specialized knowledge in applied mathematics and computer programming
- Less risk of coding mistakes, thanks to its proven and widely-tested implementation
- Research reproducibility, due to its open-source nature
- Means of communication of models among researchers and practictioners

Macroeconomic models

- Variables
  - endogenous: GDP, consumption, prices, unemployment, capital stock...
  - exogenous: determined outside of the model; have probability distribution
- Equations that determine endogenous variables
  - behaviour of agents (consumers, firms, government)
  - production technology
  - accounting and market-clearing
- Parameters
- Dynamics: contemporaneous variables can be influenced by
  - contemporaneous exogenous shocks
  - past variables
  - ...and future variables (psychological expectations)
- Steady state: point of the state space where agents choose to remain in the absence of shock (exogenous variables are at their mean)

The problem of expectations

- How to model expectations?
- Rational expectations
  - agents are intelligent (they even have unlimited computational power)
  - agents know the model (equations, parameters, probability distributions of exogenous)
  - agents know the value of all past variables and contemporaneous shocks
  - all of this is common knowledge among agents
- Technically, assuming rational expectations means solving a fixed-point nonlinear functional equation over the state space
- Rational expectations are the benchmark for economists, but there are alternatives:
  - bounded rationality (learning process, rule-based decision-making. . .)
  - partial information

Solving a model under rational expectations

- Challenging fixed-point nonlinear functional equation problem
- Perturbation approach
  - compute Taylor expansion of the solution around the steady state
  - fast and robust, but accuracy suffers for large shocks and strong nonlinearities
  - extension: piecewise-linear solution (OccBin)
- Perfect foresight solution
  - approximation in the dimension of expectations: agents know the future value of shocks
  - provides exact nonlinear solution, at moderate computational cost
- Hybrid techniques: perfect foresight with expectation errors, (stochastic) extended path
  - build on several runs of perfect foresight algorithm
  - closer to rational expectations, incorporate some uncertainty
  - same good numerical properties as perfect foresight, but computationally more costly
- Global methods
  - functional approximation over a bounded domain
  - costly and fragile, not (yet) in Dynare

Classes of models

- Structural models (mostly Dynamic Stochastic General Equilibrium models, DSGE)
  - Real business cycle models (RBC)
  - New Keynesian models (NK)
  - Overlapping generations models (OLG)
  - Heterogenous agents new Keynesian models (HANK): limited support (simplified heterogeneity, or external toolkit by Thomas Winberry)
- Semi-structural models
  - Behavioural equations, but not necessarily derived from optimization
  - e.g. FRB/US or ECB/BASE
- Statistical models
  - Timeseries support (interface to X13-ARIMA-SEATS)
  - Estimation of VAR models possible, but inefficient
  - Better coverage in some alternatives like the BEAR toolbox

The Dynare language

Example: the neoclassical growth model

```
var c k;
varexo x;

parameters alph gam delt bet aa;
alph=0.5; gam=0.5; delt=0.02; bet=0.05; aa=0.5;

model;
  c + k = aa*(1+x)*k(-1)^alph + (1-delt)*k(-1);
  c^(-gam) = (1+bet)^(-1)*(aa*alph*(1+x(+1))*k^(alph-1) + 1 - delt)*c(+1)^(-gam);
end;

initval;
  k = 10; c = 1;
end;
steady;
check;

shocks;
  var x = 0.1;
end;
stoch_simul(order=2);
```

Computational tasks available

- Steady state computation
- Model solution and dynamic simulation
- Estimation
  - full information, Bayesian
  - full information, frequentist
  - limited information, method of moments (generalized or simulated)
- Forecasting
- Shock decomposition
- Sensitivity and identification analysis
- Optimal policy
  - under commitment (Ramsey problem)
  - under discretion
  - optimal simple rules

Mathematical and computational techniques employed

- Multivariate nonlinear solving and optimization
- Matrix factorizations
- Local functional approximation
- Filtering and smoothing
- MCMC techniques for Bayesian estimation
- Graph algorithms
- Optimal control
- Symbolic algebra
- Just-in-time bytecode compilation
- Divide-and-conquer methods (homotopy, block decomposition)

Architecture of Dynare

- Preprocessor
  - Standalone executable, written in C++ (∼22% of source code)
  - Parses the Dynare language
  - Symbolic algebra manipulations (e.g. derivatives, block decomposition)
  - Outputs M files (or MEX files) embedding the model and the requested operations
- M files
  - Represent ∼61% of source code
  - For most high-level computational tasks
- MEX files
  - In modern Fortran or C++ (∼16% of source code)
  - For accelerating costly inner tasks
  - Specialized mathematical problems
  - Stacked model Jacobian construction
  - Parallelized law-of-motion of particles in particle filtering
  - Machine or bytecode representation of model equations and derivatives
- Testsuite: unit and integration tests

Leveraging the MATLAB platform

- First version of Dynare (1994) was in Gauss; switch to MATLAB occurred around 2000, because of its popularity in academia
- Intuitive and convenient syntax for scientific computing
- Easier to learn by economists than general programming languages
- Platform available in most policy institutions
- Good performance, improving over time
- Several environments
  - on the desktop
  - on servers (MATLAB Production Server)
  - in the web broser (MATLAB Online)

An open-source development model

- License: GNU General Public License (version 3)
- Development on our public GitLab instance
- Bug reports through GitLab issues or the forum
- Merge requests possible from anyone
- Continuous Integration (CI) for quality assurance
- Cooperation with other open-source projects (GNU Octave, OpenBLAS, SuiteSparse, Debian, Homebrew. . .)
- Recently accepted as a NumFOCUS project

The community

- Core developers
- Advisory committee
- Institutional sponsors and users
  - central banks
  - ministries of economy and finance
  - international organizations
- Academic users
  - professors and researchers
  - PhD and master students
- Online forum for help and discussions
- Annual events
  - Academic conference
  - Summer school (for beginners)
  - Advanced users workshop

Future projects

- Performance improvements on (very) large models
- Heterogenous agents new Keynesian (HANK) models
- Global solution methods
- Sequential Monte-Carlo (SMC) sampler
- Better solver for mixed-complementarity problems (MCP)
- More interactive model building (à la TROLL)
- Graphical user interfaces