The past, present and future of Dynare

The Better Policy Project seminar

Michel Juillard[1]

March 16, 2022

[1] michel.juillard@mjui.fr, Banque de France. The views expressed herein are ours and do not necessarily represent the views of Banque de France.

The Dynare Team

Currently the development team of Dynare is composed of:

- Stéphane Adjemian (Le Mans Université, Gains)
- Michel Juillard (Banque de France)
- Sumudu Kankanamge (Toulouse School of Economics and CEPREMAP)
- Frédéric Karamé (Le Mans Université, Gains and CEPREMAP)
- Junior Maih (Norges Bank)
- Willi Mutschler (University of Tübingen)
- Johannes Pfeifer (Universität der Bundeswehr, München)
- Marco Ratto (European Commission, Joint Research Centre - JRC)
- Normann Rion (CY Cergy Paris Université and CEPREMAP)
- Sébastien Villemot (CEPREMAP)

Outline

HISTORY

Dynare beginning

- 1994: A front end for for perfect foresight models based on Laffargue Boucekkine algorithm written in Gauss (see Laffargue 1990, Boucekkine 1995)
- Parser written in Gauss
- Benchmark case: Multimod
- 1995: first visit to IMF
- Collaboration with Troll on *New Stack* and *lkroots*
- Papers: M. Juillard 1996, Michel Juillard and Laxton 1996, Boucekkine, Michel Juillard, and Malgrange 1997, M. Juillard et al. 1998

Second order local approximation

- 1998: CEPREMAP workshop on solution of stochastic models
- First and second order approximation (see Söderlind 1999, Klein 2000)
- Confusion with risky steady state (see Schmitt–Grohe and Uribe 2002)
- Switch to Matlab
- Parser written
- Second order more important for asset prices than for business cycle
- Departure from certainty equivalence more important than increased accuracy
- Papers: Collard and M. Juillard 2001a, Collard and M. Juillard 2001b and Villemot 2011

Bayesian estimation

- Based on Schorfheide 2000 and Smets and Wouters 2002
- Bank of Finland Modeling Workshop for Central Banker Modelers, 2002, organized by Doug Laxton and Chris-Marie Rasi
- Financing for adding Bayesian estimation to Dynare
- Kalman filter to compute likelihood
- Metropolis-Hastings algorithm to simulate the posterior distribution
- Paper: Adjemian, Darracq Pariès, and Moyen 2007

Dynare++

- No reason to stop at second order
- Second order modifies the constant of the approximated solution
- Third order the slope
- Fourth order the quadratic terms ...
- Written as a stand-alone C++ program because of large computations (see Kamenik 2005)
- Dynare++ released in 2004
- 3rd order approximation integrated to Dynare-Matlab in 2010
- *k*-order integrated in Dynare-Matlab v.4.6.0 in 2020
- How to simulate with higher-order approximated solution? Full recursion versus pruning.

Consolidation

- Rewriting preprocessor in C++
- Global sensitivity toolbox (thanks to JRC, Ratto 2008)
- Diffuse Kalman filter Durbin and Koopman 2012
- Optimal policy along the lines of Christiano, Motto, and Rostagno 2007, Svensson 2010 and Dennis 2007
- Block decomposition of large models (see Mihoubi 2011)
- Macro-processor
- Octave version
- Version 4.0, 2008
- Dynare Forum

New horizons

- Parallel computation of MCMC chains
- Automatic stationarization
- Stochastic extended path
- Particle filter for estimation of nonlinear models
- Identification analysis
- Conditional forecasts
- MS-BVAR
- More efficient algorithms (Lyapunov equation, Sylvester equation, cyclic reduction algorithm for quadratic matrix equations, written in C++ and in Fortran)
- Integration of OCCBIN (Luca Guerrieri and Matteo Iacoviello 2014)
- Version 5.0, January 2022.

Collaborations

- IMF
- Norges Bank
- Joint Research Center of the European Commission (JRC-Ispra): Global sensitivity, identification analysis
- Federal Reserve Bank of Atlanta: MS-BVAR
- ECB (ECB-Base)
- BdF (FR-BDF)

Dynare Diffusion

Version	Release	Downloads
4.1.0	2009	5924
4.1.1	2010	2553
4.1.2	2010	12795
4.1.3	2010	11150
4.2.0	2011	8522
4.2.1	2011	12780
4.2.2	2011	5866
4.2.4	2011	11165
4.2.5	2012	6642
4.3.0	2012	5229
4.3.1	2013	18007
4.3.3	2013	8699
4.4.0	2014	4531
4.4.1	2014	4115
4.4.2	2014	43115
4.4.3	2014	259697
4.5.0	2017	10265
4.5.1	2017	11829
4.5.2	2017	460
4.5.3	2017	23329
4.5.4	2018	14214
4.5.5	2018	4172
4.5.6	2018	17439
4.5.7	2019	29693
4.6.0	2020	2972
4.6.1	2020	20174
4.6.2	2020	10486
4.6.3	2020	14495
4.6.4	2021	32008
5.0	2022	7508

Notes:
Downloads are for all architectures.
Data are missing for year 2013.
Results depend on the length of time a given version remained the most recent one.

STRENGTHS and WEAKNESSES

Strong points (I)

- A simple and intuitive modeling language (see Macroeconomic Model Data Base, model collections such as Johannes Pfeifer at `https://github.com/JohannesPfeifer/DSGE_mod` and many more)
- Help for replication
- A very efficient preprocessor (including symbolic derivation)
- Fast algorithms

Strong points (II)

- A unique environment for several connected tasks:
  - Solution of the model
  - Simulation
  - Estimation (Bayesian, ML, methods of moments)
  - Forecasting
  - Optimal policy
  - Occasionally binding constraints
  - Markov-Switching VAR
  - Simple time series operations
  - Reporting results
- A lively community (forum, Summer school, Dynare conferences)
- Fast correction of bugs
- Has been adopted for teaching

Some birth defects

- Too much concern about memory footprint
  - selecting variables present with leads or lags,
  - in perfect foresight algorithm: triangularizing the Jacobian matrix of the stacked system period by period, instead of using sparse matrix code (corrected since).
  - partly due to hardware limitations in mid-1990
- Ignoring time series nature of data (partly corrected with dseries)
- Absence of block for calibrating parameters
- Absence of a documented API for the main numerical functions.
- Difficulties in handling contributions

New directions

- Estimation of models with zero lower bound
- Support for heterogeneous agents models
- Global solution methods
- Limited rationality

DYNARE-JULIA

Rewriting Dynare

- More than 20 years of development layers
- Two language problem: fast code must be written in C, C++ or Fortran
- Matlab doesn't have an ecosystem or a community of open source developers
- Matlab price tag is a deterrent for many users and Octave is too slow
- Why not Python?

Why Julia

- meant for scientific computing
- JIT compiling
- solves the two language problem
- user defined types, no classes, multiple dispatch
- rich collection of existing packages
- lively community
- efficient package management
- powerful metaprogramming

Dynare-Julia

- Different packages for numerical algorithms and for interpretation of Dynare modeling language
- Re-using Dynare C++ preprocessor with special output for Julia
- Insist on modularity
- Homogeneous coding style
- Leveraging existing Julia packages (NLsolve, Plots, PrettyTables)
- `https://github.com/DynareJulia`

Existing packages

- Dynare.jl
- ExtendedDates.jl
- FastLapackInterface.jl
- GeneralizedSylvesterSolver.jl
- KalmanFilterTools.jl
- KroneckerTools.jl
- LinearRationalExpectations.jl
- PolynomialMatrixEquations.jl
- QuasiUpperTriangular.jl
- TimeDataFrame.jl

Dynare.jl

- Deals with Dynare instructions
- The preprocessor is the same executable as for Dynare Matlab/Octave
- The state of computations is kept in `context` as structure of `Context` type

```
struct Context
    symboltable::SymbolTable
    models::Vector{Model}
    modelfileinfo::ModFileInfo
    results::Results
    work::Work
end
```

- Instructions are extracted from

  `./<MODFILENAME>/model/json/modfile.json`

  that is written by the C++ preprocessor

LinearRationalExpectations.jl

- Solves model of the form

$$E_t\{Ay_{t+1} + By_t + Cy_{t-1} + Du_t + e\} = 0$$

- The deterministic steady state of the model is defined as

$$\bar{y} = -(I - A - B - C)^{-1}e$$

- The solution takes the form

$$y_t - \bar{y} = G_y(y_{t-1} - \bar{y}) + G_u u_t$$

- G_y is the solution of the polynomial matrix equation

$$AG_yG_y + BG_y + C = 0$$

and

$$G_u = -(AG_y + B)^{-1}Du_t$$

PolynomialMatrixEquations.jl

Two different algorithms are provided:

- generalized Schur decomposition (Klein 2000; Sims 2002; Söderlind 1999)
- cyclic reduction (Bini, Iannazzo, and Meini 2012)
- the package provides functions to reduce the problem size by eliminating static variables.

QuasiTriangular.jl

- Multiplication and linear system solution for upper quasi triangular matrices
- Useful for handling real generalized Schur decomposition

KalmanFilterTools.jl

- Kalman filter
- Monitored Kalman filter
- Fast Kalman filter (Herbst 2012)
- Kalman smoother
- Diffuse filter/smoother (Durbin and Koopman 2012)
- Univariate filter/smoother

TimeDataFrames and ExtendedDates

- A data frame where the first column represents dates
- Extension of dates in Julia to Year, Semester, Quarter, Month, Week, Day and Undated
- Each frequency is represented by an integer starting at some epoch (January 1st of year 1970)

FastLapackInterface.jl

- Pre-allocate workspace for some Lapack function
- LU decomposition and solution of linear systems
- QR decomposition
- Schur decomposition

KroneckerTools.jl

- Chains of Kronecker products

$$
\begin{aligned}
c &= (Ip \otimes A \otimes Iq) * b \\
c &= (Ip \otimes AT \otimes Iq) * b \\
c &= (A \otimes A \otimes ... \otimes A) * b \\
d &= (AT \otimes AT \otimes ... \otimes AT \otimes B) * c \\
C &= A * (B \otimes B \otimes .... \otimes B) \\
D &= A * B * (C \otimes C \otimes .... \otimes C) \\
D &= AT * B * (C \otimes C \otimes .... \otimes C) \\
E &= AB(C \otimes D \otimes ... \otimes D)
\end{aligned}
$$

- Implements algorithms in Kamenik 2005

GeneralizedSylvesterSolver.jl

solves

$$ax + bx(c \otimes c \otimes ... \otimes c) = d$$

by using

$$(I + c^T \otimes c^T \otimes ... \otimes c^T \otimes b)x = d$$

Currently implemented

- `calib_smoother`
- check
- deterministic_trends
- histval
- initval
- perfect_foresight_setup (only some options)
- perfect_foresight_solver (only some options)
- planner_objective
- ramsey_model
- shocks
- steady
- stoch_simul (only order=1, no moments)

Installation and usage

- Installation

```
>> using Pkg
>> pkg"add Dynare"
```

- Usage

```
>> using Dynare
>> context = @dynare "example1.mod";
```

The results are in the context structure.

Dynare-Julia roadmap

1. Estimation of linear(-ized) models
2. Address *time to first result* problem
3. Mixed complementarity problems
4. Forecasting
5. Making existing packages production grade (tests, documentation, benchmarking)
6. Dynare-Julia documentation
7. *k*-order approximation
8. Multiple models
9. Linear models with occasionally binding constraints
10. Nonlinear estimation - particle filters
11. Nonlinear estimation - simulated methods of moments
12. Markov-Switching DSGE models
13. Global sensitivity analysis
14. Global methods of approximation
15. Limited rationality

Conclusion

- An ambitious project
- Will succeed only as a collaborative effort

References I

- Adjemian, Stéphane, Matthieu Darracq Pariès, and Stéphane Moyen (Aug. 2007). *Optimal monetary policy in an estimated DSGE for the euro area*. en. Tech. rep. 803. Publication Title: Working Paper Series. European Central Bank.
- Bini, Dario A., Bruno Iannazzo, and Beatrice Meini (2012). *Numerical solution of algebraic Riccati equations*. Philadelphia, PA: SIAM.
- Boucekkine, Raouf (1995). "An alternative methodology for solving nonlinear forward-looking models". In: *Journal of Economic Dynamics and Control* 19.4, pp. 711–734.
- Boucekkine, Raouf, Michel Juillard, and Pierre Malgrange (1997). "Precision Performances of Terminal Conditions for Short Time Horizons Forward-Looking Systems". en. In: *Computational Economics* 10.2. Publisher: Springer & Society for Computational Economics, pp. 169–186.

References II

- Christiano, Lawrence J., Roberto Motto, and Massimo Rostagno (2007). *Notes on ramsey-optimal monetary policy*. Tech. rep. mimeo, Northwestern University. URL http://faculty. wcas. northwestern. edu/ lchrist/d16/d1606/ramsey. pdf.
- Collard, F. and M. Juillard (2001a). "A Higher–Order Taylor Expansion Approach to Simulation of Stochastic Forward–Looking Models with an Application to a Non–Linear Phillips Curve". In: *Computational Economics* 17, pp. 125–139.
- — (2001b). "Accuracy of stochastic perturbation methods: The case of asset pricing models". In: *Journal of Economic Dynamics and Control* 25, pp. 979–999.
- Dennis, Richard (2007). "Optimal Policy In Rational Expectations Models: New Solution Algorithms". In: *Macroeconomic Dynamics* 11.01, pp. 31–55.

References III

- Durbin, James and Siem Jan Koopman (2012). *Time Series Analysis by State Space Methods*. en. Publication Title: OUP Catalogue. Oxford University Press.
- Herbst, Edward (2012). *Using the "Chandrasekhar Recursions" for likelihood evaluation of DSGE models*. en. Tech. rep. 2012-35. Board of Governors of the Federal Reserve System (U.S.)
- Juillard, M. (1996). *DYNARE: A program for the resolution and simulation od dynamic models with forward variables through the use of a relaxation algorithm*. Working Paper 9602. CEPREMAP.
- Juillard, M. et al. (1998). "An algorithm competition: first–order iterations versus Newton–based techniques". In: *Journal of Economic Dynamic and Control* 22, pp. 1291–1318.
- Juillard, Michel and Douglas Laxton (Sept. 1996). *A Robust and Efficient Method for Solving Nonlinear Rational Expectations Models*. en. Tech. rep. 96/106. Publication Title: IMF Working Papers. International Monetary Fund.

References IV

- Kamenik, Ondra (2005). "Solving SDGE models: A new algorithm for the Sylvester equation". In: *Computational Economics* 25, pp. 167–187.
- Klein, P. (2000). "Using the generalized Schur form to solve a multivariate linear rational expectations model". In: *Journal of Economic Dynamics and Control* 24, pp. 1405–1423.
- Laffargue, Jean-Pierre (1990). "Résolution d'un modèle macroéconomique avec anticipations rationnelles". In: *Annales d'Economie et de Statistique* 17, pp. 97–119.
- Luca Guerrieri and Matteo Iacoviello (2014). *OccBin: A toolkit for solving dynamic models with occasionally binding constraints easily.* Tech. rep.
- Mihoubi, Ferhat (2011). "Solving and estimating stochastic models with block decomposition". en.
- Ratto, Marco (Mar. 2008). "Analysing DSGE Models with Global Sensitivity Analysis". en. In: *Computational Economics* 31.2, pp. 115–139.

References V

- Schmitt–Grohe, S. and M. Uribe (2002). *Solving Dynamic General Equilibrium Models Using a Second-Order Approximation to the Policy Function*. Working Paper. Rutgers University.
- Schorfheide, Frank (2000). "Loss function-based evaluation of DSGE models". en. In: *Journal of Applied Econometrics* 15.6. Publisher: John Wiley & Sons, Ltd., pp. 645–670.
- Sims, C. (2002). "Solving rational expectations models". In: *Computational Economics* 20, pp. 1–20.
- Smets, Frank and Raf Wouters (2002). *An estimated dynamic stochastic general equilibrium model of the euro area*. Working Paper Research 35. National Bank of Belgium.
- Söderlind, P. (1999). "Solution and estimation of RE macromodels with optimal policy". In: *European Economic Review* 43, pp. 813–823.
- Svensson, Lars EO (2010). *Optimization under Commitment and Discretion, the Recursive Saddlepoint Method, and Targeting Rules and Instrument Rules: Lecture Notes*.

References VI


Villemot, Sébastien (Apr. 2011). *Solving rational expectations models at first order: what Dynare does*. en. Tech. rep. 2. Publication Title: Dynare Working Papers. CEPREMAP.