

# FRBNY DSGE Model (Version 990.2)

MATLAB code to estimate and forecast the model discussed in the Liberty Street Economics blog post "The FRBNY DSGE Model Forecast".

## Running the Code

### Running with Default Settings

All you need to do is run the file `Main.m`. This script will run the entire set of code, calling

- `set_paths.m`: Set default directories for input and output; add code subfolders to MATLAB path.
- `spec_990.m`: Set model specifications and important flags for estimation and forecasting.
- `gibb_est_ant.m`: Find posterior mode and sample from posterior distribution.
- `forecast_parallel_est_ant.m`: Forecast observables; can run in parallel.
- `forplot.m`: Load forecasts into data structures to prepare for plotting.
- `plotPresentation.m`: Plot forecasts.

### Running with Modified Settings

If you would like to change defaults for estimation and forecasts, see `spec_990.m`. There, you can modify

- **Estimation Parameters**
  - `reoptimize`: Whether to re-optimize and find the mode or use saved mode.
  - `CH`: Whether to re-compute the hessian or use saved.
  - `nsim`: The number of posterior draws per block.
  - `nblocks`: The number of blocks.
  - `nburn`: Size of the burn-in.
  - `jstep`: From the blocks, the forecasting code will only use every `jstep`-th element.
- **Forecast Parameters**
  - `zerobound`: Whether to incorporate anticipated policy shocks.
  - `peachflag`: Whether to condition time T+1 forecasts of observables on user-provided forecasts of observables (treating the information supplied by the user as data).
  - `distr`: Flag to specify whether to parallelize the forecast procedure.
  - `nMaxWorkers`: Number of workers to use in parallel forecast procedure.

## Directory Structure

In the main folder, there exist the following directories to house code components:

- `data/`: Input data
- `dsgesolv/`: Solving the model; includes `gensys.m` code.

- `estimation/`: Mode-finding and posterior sampling.
- `figures/`: For output including parameter moment-tables and TeX tables.
- `forecast/`: Forecasting programs.
- `graphs/`: For graphs of forecasts.
- `initializaization/`: Loading data, defining model structure, setting up important model flags.
- `kalman/`: Kalman filtering and smoothing.
- `plotting`: Loading forecast distributions and generating/saving plots.
- `save/`: Input mode and output data generated by the code (output mode, posterior draws, forecasts).
- `toolbox/`: Supporting programs.

## Program Details

This section describes important programs in greater detail. If the user is interested only in running the default model and reproducing the forecast results, this section can be ignored.

This section focuses on what the code does and why, while the code itself (including comments) provides detailed information regarding *how* these basic procedures are implemented.

### Estimation

**Main Program:** `estimation/gibb_est_ant.m`

**Purpose:** Finds modal parameter estimates and samples from posterior distribution.

#### Main Steps

- *Initialization*: Read in transform raw data from `data/`. Load files from `initialization/` related to model specification, parameter priors, parameter restrictions.
- *Find Mode*: The main program will call the `csminwel.m` optimization routine to find modal parameter estimates. Can optionally start estimation from a starting parameter vector by specifying `data/mode_in`.
- *Sample from Posterior*: Posterior sampling begins from the computed mode, first computing the Hessian matrix to scale the proposal distribution in the Metropolis Hastings algorithm. Settings for the number of sampling blocks and the size of those blocks can be specified in `spec_990.m`.

*Remark:* In addition to saving each draw of the parameter vector, the estimation program also saves the resulting posterior value and transition equation matrices implied by each draw of the parameter vector. This is to save time in the forecasting step since that code can avoid recomputing those matrices. In addition, to save space, all files in `save/` are binary files.

### Forecasting

**Main Program:** `forecast/forecast_parallel_est_ant.m`

**Purpose:** Compute forecast distribution for the observables, sampling from the full posterior distribution of parameters and sampling exogenous shocks.

### Main Steps

- *Load Draws:* Load in posterior distribution blocks that are output from the estimation stage.
- *Filter and Smooth:* Pass matrices defining the state transition equation into `forecastFcn_est_ant.m`, which will filter and smooth the states over the history.
- *Forecast:* Compute forecasts using `getForecast.m`, which takes matrices corresponding to a posterior draw and uses them to iterate on the time T state vector to obtain forecasts, adding in draws of exogenous shocks as well.

*Remark:* The code can be run in parallel or sequentially. Running in parallel requires the MATLAB parallel toolbox.

## Plotting

**Main Programs:** `plotting/forplot.m`, `plotting/plotPresentation.m`

**Purpose:** Generate plots of observables.

### Main Steps

- *Load Plots:* The program `forplot.m` will load output from the forecast program into the workspace, computing entries in the data structures `Means` and `Bands`.
- *Plot:* The program `plotPresentation.m` will plot the forecasts and shock decompositions, saving the graphs in the `graphs/` folder.

*Remark:* Each time `forplot.m` is called, it will attempt to recompute the forecast means and bands, which takes some time. However, these are saved in `save/` after the very first call to `forplot.m`. To use these saved Means and Bands structures and avoid recomputing, set `useSavedMB=1` before running `forplot.m`.

## Final Notes on MATLAB Versions and Toolboxes

In certain functions implemented in this program are Toolbox functions provided by Mathworks. If you are receiving errors in running these programs due to undefined functions, it is likely because you do not yet have access to these Toolboxes. For example, to run the forecasts in parallel (the default setting), you will need the parallel toolbox. Also, `dlyap.m`, a function to solve discrete-time Lyapunov equations, comes from the Control System Toolbox.

These programs are meant to be run in Matlab09a. While we have not attempted to run these programs using a more recent version of Matlab, it is possible that some of the Matlab-defined functions are not identical and thus may yield nonidentical results.

# DISCLAIMER

Copyright Federal Reserve Bank of New York. You may reproduce, use, modify, make derivative works of, and distribute and this code in whole or in part so long as you keep this notice in the documentation associated with any distributed works. Neither the name of the Federal Reserve Bank of New York (FRBNY) nor the names of any of the authors may be used to endorse or promote works derived from this code without prior written permission. Portions of the code attributed to third parties are subject to applicable third party licenses and rights. By your use of this code you accept this license and any applicable third party license.

THIS CODE IS PROVIDED ON AN "AS IS" BASIS, WITHOUT ANY WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, EXCEPT TO THE EXTENT THAT THESE DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. FRBNY IS NOT, UNDER ANY CIRCUMSTANCES, LIABLE TO YOU FOR DAMAGES OF ANY KIND ARISING OUT OF OR IN CONNECTION WITH USE OF OR INABILITY TO USE THE CODE, INCLUDING, BUT NOT LIMITED TO DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, PUNITIVE, SPECIAL OR EXEMPLARY DAMAGES, WHETHER BASED ON BREACH OF CONTRACT, BREACH OF WARRANTY, TORT OR OTHER LEGAL OR EQUITABLE THEORY, EVEN IF FRBNY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR LOSS AND REGARDLESS OF WHETHER SUCH DAMAGES OR LOSS IS FORESEEABLE.