

=====

OVERVIEW

This program reads a Dynare model file, generates the first-order conditions of the Ramsey policymaker, and writes out a new Dynare model and an M-file that provides a linear system for obtaining the numerical steady-state of the Lagrange multipliers.

(c) Andrew T. Levin 2005, 2006

Note: This program (and associated subroutines) may be used freely subject to acknowledgement of the source and citation of either or both of the following papers (for which these routines were created):

Levin, A., Lopez-Salido, J.D., 2004. "Optimal Monetary Policy with Endogenous Capital Accumulation", manuscript, Federal Reserve Board.

Levin, A., Onatski, A., Williams, J., Williams, N., 2005. "Monetary Policy under Uncertainty in Microfounded Macroeconometric Models." In: NBER Macroeconomics Annual 2005, Gertler, M., Rogoff, K., eds. Cambridge, MA: MIT Press.

=====

INSTRUCTIONS

To run this program, just do the following:

1. Create a plain-text file (henceforth denoted by xxx.mod) containing your model in dynare format. (For a sample model, see the file BestSW.mod that is included in this package.) In addition to the standard dynare syntax rules, you'll need to add comment lines to indicate which groups of variables are endogenous vs. exogenous (that is, invariant to policy); this distinction is needed because we only want to differentiate the Lagrangian with respect to the endogenous variables. Also, you need comments to indicate which equations are simple policy rules (one for each policy instrument). Note that in the new model file (henceforth denoted by yyy.mod) created by get_ramsey, these simple rules are replaced by the set of first-order conditions that represent the Ramsey policy.

2. In matlab, set the two string variables specifying the name of the input model file that you've just created (xxx) and the name of the new Ramsey model file (yyy), and then execute get_ramsey.m (which takes no explicit arguments):

```
infilename = 'xxx';
outfilename = 'yyy';
get_ramsey;
```

3. If the model isn't very big and the steady states of the lagrange multipliers can be readily computed by the dynare nonlinear solver (executed by the "steady" command), then you can proceed by simply adding parameter values and other dynare statements into yyy.mod, and then feed this file to the dynare processor.

4. Alternatively, you can use the following approach to compute a good set of initial guesses for the Lagrange multipliers and then evaluate the Ramsey policy for a given set of parameters. In this case, do *not* add any parameter values or other dynare commands to the xxx.mod or yyy.mod files. Instead, specify all of the parameter values in a matlab file (henceforth denoted as parms.m), and put any subsequent dynare commands into a separate matlab file that can be executed after dynare has successfully computed the steady state of the ramsey model.

```
dynparse xxx;      (use the dynare parser to process xxx.mod)
dynparse yyy;     (use the dynare parser to process yyy.mod)
parms;            (specify parameters of model xxx)
```

```
xxx;          (parsed version of the original model)
steady;      (get the steady state of model xxx)
copy_steadystate; (creates the variables lmult*_SS)
yyy_lmss.m;  (generates initial guesses for lagrange multipliers)
yyy.m;       (parsed version of the Ramsey model)
set_steadystate; (copies steady state values into dynare vector ys_)
steady;      (get steady state of model yyy)
...          (now give other dynare commands as desired)
```

5. Finally, note that the program `calc_welfframsey.m` can be used to compute conditional expected welfare at the deterministic steady state.

```
=====
UPDATE HISTORY
10 August 2005: first public release
17 September 2005: new version allows for multiple policy instruments
                   also much faster for large models (now takes about
                   one second per endogenous variable)
23 March 2006: facilitates computation of steady states of Lagrange multipliers
                (especially useful when Ramsey steady-state inflation rate
                is not known a priori)
12 August 2006: clarified instructions for use
=====
```