The Science and Art of DSGE Modelling
A Foundations Course

Calibration and Use of the External Steady State

Paul Levine

September 7, 2020

The Steady State in Dynare

To recall, this is the most difficult part of the set-up!! Recall four approaches:

- Use Initial Guesses
- Solve for n variables in n unknowns using fsolve.
- Exploit the recursive structure to solve for $m << n$ variables in Dynare
- Obtain steady state analytically (Day 1).

Now we will focus on the use of **fsolve** in an external steady state (third bullet point).

Analytical Steady State in the .mod file

- Dynare can solve models without setting up the steady state in an external file and by guessing initial values.
- For medium and large size models this approach doesn't work!!
- We have seen see how to work out the recursive analytical steady state for both the RBC and the NK models.
- For richer models (for example those with financial frictions) a recursive analytical steady state is not possible and it must be solved using fsolve.
- This part of the course shows you how to do this.

Numerical Solution of the Steady State

- If you want to solve non-linear system of equations you can use the Matlab built in function *fsolve*.
- Consider you want to numerically find the root of the function $y = x^2 + 3x - 10$ which means that you want to find the value of x such that y is zero. (you basically want to solve $0 = x^2 + 3x - 10$).
- The command is $[xsoln, fval] = fsolve(@(x)x^2 + 3 * x - 10, x0)$
- *xsoln* is the solution returned for x, *fval* the function value y at *xsoln* and x0 is the initial guess . If solving the equation was successful, fval should be 0.

Numerical Solution of the Steady State

- Looking at the input arguments, the *@(x)* is called a function handle. It indicates for which variable in the following equation it should be solved.

- The second argument, the x0 is the starting value for the algorithm. fsolve is only able to find local solutions and the starting value matters.

- Since $x^2 + 3 * x - 10 = (x - 2)(x + 5)$ there are two solutions x=2, -5. Which one is arrived at depends on the choice of x0.

- You can also provide your own functions to Matlab's optimization routines. We do this in what follows.

Numerical Optimization

- We only briefly introduce numerical optimization techniques in Matlab.

- A good starting point is often the Matlab help, which provides good guidance on which algorithm/solver to use for the problem at hand. In general, functions differ whether your problem i) is nonlinear, ii) differentiable, iii) is univariate or multivariate, iv) involves solving or minimizing a function, and/or iv) features constraints.

- Important solvers for economic applications are i) *fminbnd*, a fast Matlab routine that finds the minimum of single-variable continuous function on a fixed interval and is thus often used to solve for the optimal labor supply, and ii) Chris Sims *csminwel.m*, a popular solver used in the Estimation - see Day 3.

- Also the symbolic toolbox can be used to perform symbolic math and calculus like you do in Maple or Mathematica.

Calling an external steady state

- Now we explain how to code and use an external steady state file to accompany the .mod file.
- The steady states for the RBC and NK-SW models are solved using matlab programs *MODFILENAME_steadystate.m* which calls *fun_RBC.m*.
- In order for Dynare to understand that it must look for the external file solving for the steady state the m-file should have exactly the same name as the mod file followed by *_steadystate.m*.
- The file *MODFILENAME_steadystate.m* uses the matrix *M_* in which Dynare stores the value of the parameters used in the mod file and uses the Matlab function **fsolve** to find the roots of the equations set up in *fun_RBC.m*.
- The file *MODFILENAME_steadystate.m* then calculate the steady state values of all the endogenous variables in the model and stores them in the array *ys* that is the name that Dynare uses for the vector of the steady state values of the endogenous variables in the mod file.

Calling an external steady state

- The codes that use this approach can be found in the folder **External_ss**. **RBC_1.mod** solves the steady state with an external steady-state matlab file simply repeating the analytical steady state used before in the set-up without an external steady state.

- Then **RBC_2.mod** treats hours as a variable to be solved using fsolve and the function file, *fun_RBC_2.m*.
- This is similar to the steady state for the NK-SW model seen in Days 2 and 3.

Example: RBC model SS solving for hours

- The steady state of the RBC model solved by considering hours as unknown ($H = \frac{e^x}{1+e^x}$, constraints $H \in [0,1]$)

$$
\begin{aligned}
R = \frac{1}{\beta} \quad &\Rightarrow \quad R; \quad \frac{K(R-1+\delta)}{Y} = 1-\alpha \Rightarrow \frac{K}{Y} \\
Y \quad &= \quad (AH)^{\alpha} K^{1-\alpha} = (AH)(K/Y)^{\frac{1-\alpha}{\alpha}} \Rightarrow Y \\
K \quad &= \quad \frac{K}{Y} Y; \quad I = \delta K \\
G \quad &= \quad \frac{G}{Y} Y \\
W \quad &= \quad \alpha \frac{Y}{H} \\
C \quad &= \quad W \frac{(1-\varrho)(1-H)}{\varrho H}
\end{aligned}
$$

- And then use fsolve to find H from the resource constraint $Y - C - I - G = 0$ (using a matlab function **fun_RBC.m**).

Calbration of the RBC Model in Dynare

The subfolder **External_ ss** contains modifications of the main RBC code used for various stages of the calibrations:

- **RBC_1.mod** : is the previous set-up with an external steady state replacing steady_state_model in the mod file;
- **RBC_2.mod** : calibrates hours worked (*hobs*) and treats ϱ as a constant endogenous variable. The value of ϱ consistent with *hobs* is then found using fsolve which calls **fun_RBC_2.m**.
- **RBC_3.mod** : calibrates hours worked and investment as a share of GDP (*hobs*, *iobs*) and treats ϱ and δ as constant endogenous variables. The values of ϱ and δ consistent with *hobs* and *iobs* are then found using fsolve calls **fun_RBC_3.m**.
- **RBC_4.mod** : calibrates hours worked, the investment share and the interest rate (*hobs*, *iobs*, *Rss*) and treats ϱ, δ and β as constant endogenous variables. The values of ϱ, δ and β consistent with *hobs*, *iobs* and *Rss* are then found using fsolve calls **fun_RBC_4.m**.
- **Exercise** Rework the examples above using the RBC model with external habit from the exercise.