

# Estimating/Calibrating a Structural Model

Trevor Gallen

Econ 64200

Fall 2022

## Deliverables

- You should have a word/L<sup>A</sup>T<sub>E</sub>X document that has three sections:
  1. Discusses the model and answers the questions I pose throughout.
  2. Contains the tables and figures you will produce.
  3. Contains a discussion of your programming choices if you had to make any.
- You should have a Matlab file or set of files (zipped) that contain **all** your programs and raw data. There should be a file called “Main.M” that produces everything I need in one click.

## 1 Model

Take the model from Homework 2, repeated here for convenience, with slightly different parts, highlighted in **red**: Households have Stone-Geary utility over consumption  $c_t$ :

$$u(c_t, L_t) = \sqrt{(c_t - 1)}$$

Their income each period is the sum of permanent income  $P_t$  and a transitory shock  $\epsilon_t$ :

$$Y_t = P_t + \epsilon_t$$

Where:

$$\epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2)$$

Permanent income is a random walk:

$$P_t = P_{t-1} + \zeta_t$$

Where:

$$\zeta_t \sim \mathcal{N}(0, \sigma_\zeta^2)$$

Households face the budget constraint:

$$Y_t + (1 + r)s_{t-1} = c_t + s_t$$

And the borrowing constraint:  $s_t \geq \bar{s}$ .

Households maximize the net present value of utility, discounted at a rate  $0 < \beta < 1$ . Assume that households live until period  $T$ .

Now, let the following numerical assumptions hold:

Table 1: Calibration		
Concept	Parameter	Value
Lifespan	$T$	45
Discount factor	$\beta$	???
Borrowing constraint	$\bar{s}$	-0.2
Initial permanent income:	$P_0$	5
Variance of permanent income shock:	$\sigma_\zeta^2$	0.02
Variance of transitory income shock:	$\sigma_\epsilon^2$	0.04
Interest rates:	$r_t$	0.05
Initial savings:	$s_0$	$\sim \mathcal{N}(0, ???)$

Table 1: Note that  $s_0$  is truncated normal at  $\bar{s}$

**Question 1:** Your goal is to calibrate this structural model by finding  $\beta$  and  $\sigma_\zeta^2$  such that, in expectation:

$$mean(\log(c_{30}^{sim})/\log(c_{29}^{sim})) = 1.001788$$

$$Var((s_2^{sim})) = 0.00409$$

To do so, you need to have an estimating “outer loop” and a value-function solving and simulating “inner loop.” Homework 2 created a file that 1) solved household models and (2) simulated households *given*  $\beta$  and  $\sigma_{s_0}^2$ . Your goal is to now (1) create a “wrapper” function that *takes*  $\beta$  and  $\sigma_{s_0}^2$  and spits out (for instance) the squared sum of difference between your simulated result and your target result, and chooses  $\beta$  and  $\sigma_{s_0}^2$  to minimize that sum of squared errors.

This is the core of simulated method of moments/“calibration.”

### Suggested Solutions:

This homework had a few traps! Broadly, the goal was to introduce you to how to estimate a structural model via simulated method of moments (our minimum distance estimator.)

- Write down code to solve for an agent’s policy function  $g$  as a function of parameters to be estimated  $\theta$
- Simulate agents
- Find sum of errors of simulated vs data moments
- Change  $\theta$  until this is minimized (or zero)

However, there were a few issues you had to think about:

- Any errors in your solution (due to a sparse grid for instance) would yield errors in the simulation, and thus in the estimator
- Any errors in your simulation (due to too-few observations for instance) would yield errors in the simulation’s results, and thus in the estimator
- Massive errors would be caused by not setting a common seed for the random number generator across function calls
- Any of the above three would cause a derivative-based solver to have a hard time (can’t take (accurate) derivatives when your surface is changing). Derivatives would be all noise, convergence is extremely unlikely.
- Depending on how you did the simulation, you could end up with a state variable outside the bounds you set up. This is *particularly* a problem when we’re changing the variance of shocks! Increasing variance
- It is possible not to be able to fit a structural model even if you have as many moments as parameters (this was the case here). (  $\#$  moments =  $\#$  parameters is not enough!)
- Flat regions of error as a function of parameters will cause high standard errors (think Fisher Information matrix!)

Each iteration took a while, as will happen with a lot of dynamic structural estimation! To think about why, imagine that you might be iterating many many nested loops!

- Outermost loop: search over parameters (estimation)

- Middle loop: given parameters (outermost loop) & solutions (inner loop) search to find equilibrium between agents (CGE-style), such as wages clearing markets, or agents beliefs about other agents (Krusell-Smith style)
- Inner loop: given parameters (outermost) and wages/equilibrium objects (middle loop), find agent's solution

In this case we just had an outermost loop and an inner loop. I used Matlab's surrogate optimization, but you could have used patternsearch or any of the minimizers. *However*, derivative-based minimizers were subject to error, as discussed above.

When I did, I found:  $\beta = 0.8904$ , and  $\sigma_{s_0}^2 = 0.36511$ , which isn't too far from the original  $\beta = 0.9$  that I used to generate the moments, but very far from the and  $\sigma_{s_0}^2 = 0.1$ . Why? Looking at the sampled simulated moment errors with respect to the parameter inputs (see Figure 1, you see that we are very "flat" around the minimum— $\sigma_{s_0}^2$  doesn't map very well into our moments (they were poorly picked!). If you think about what the Fisher Information matrix would say about the standard errors implied in Figure 1, you'd see that the second derivative seems very low (with low off-diagonals), so the inverse of that matrix is very large—large standard errors on both. Put another way, if you had simulated this differently (different seed), you might have gotten almost anywhere on the flat region. This problem was *not* well-identified, primarily due to the second moment.

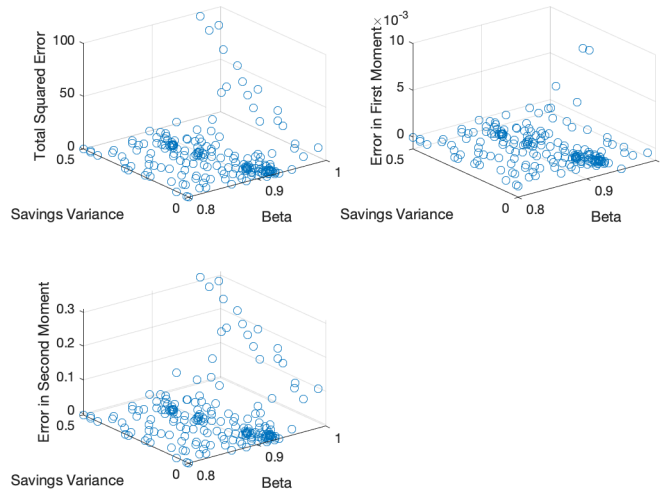


Figure 1: Total squared error, error of moment #1 (growth) and moment #2 (savings variance).