Demand for Disability Insurance

Evidence from Germany

Sebastian Seitz

University of Mannheim

January 27, 2019

Introduction - Project

- Estimation of HH life-cycle model:
  - Choice: Labor supply, savings, purchase private disability insurance, and (if sick) claim (private/ public) DI
- Research Question: Crowd - out of private DI:
  - Welfare improving policies
  - Response heterogeneity
- Data:
  - G-SOEP
  - Income and Consumption Survey (2013)
  - Private Company Data

Introduction - Project

- Estimation Procedure: Method of Simulated Moments:
  Minimize squared distance between data and simulated moments
- Basically GMM with identity weighting matrix:

$$\min \hat{Z} = (M_{Data} - M_{Sim})'(M_{Data} - M_{Sim}) \tag{1}$$

  where M is a certain moment, e.g. mean labor force participation
- Goal: Recover structural parameters for counterfactual exercise

Introduction - Project

Estimation in three steps:

1. Pre-set some values from literature
2. Estimate health and income processes outside the model
3. Estimate remaining structural parameters using MSM

Technicalities

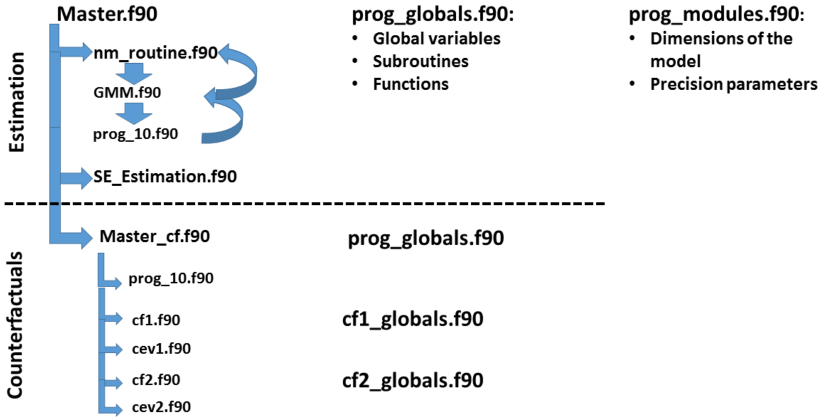
- No analytical solution exists $\rightarrow$ Computation
- I use:
  - My laptop (enough prowness)
  - Self - written Fortran code
  - Intel Visual Studios

Fortran

Fortran:

- Formula Translation = Fortran
- High - performance language
- Compiled imperative language $\rightarrow$ Needs a compiler to run
- Advantages over MatLab or R:
  - MatLab and R are not compiled but interpreted languages $\rightarrow$ need interpreter to run
  - Line-by-Line versus compiling into program $\rightarrow$ Easier to spot mistakes
  - Significantly faster
- Disadvantages:
  - Only user-written packages $\rightarrow$ Adapt this code to your problem
  - No "output" window like MatLab or R
  - Debugging can take a while

My Code - Structure



My Code - Algorithms

- Nelder - Mead to minimize GMM function
- Brent's method to minimize negative utility function
- Probabilities via Tauchen Method (for non-stationary processes)
- Grids: Equal spaced
- Interpolation: Spline

My Code - Technical Details

- Computationally intensive to solve my model
  - NM - algorithm is derivative free, but slow
  - Large state space
  - "French Correction" (French, 2005) → Solves selection issues but increases computational time French Correction
- Solutions:
  - Apply parallelization (openMP):
    - for solving the model
    - for interpolation of functions
  - NM is fast on first 100 - 200 steps so re-initialize it after some time

My Code - Testing

Important!

- Check whether written subroutines and functions work
- Two examples:
  1. Derivatives $\rightarrow$ Analytical
  2. OLS $\rightarrow$ Use R

Example: Derivative subroutine

```
subroutine num_der(func, x, error, Grad)
!Define the parameters:
⋮
! Variables needed in the procedure:
⋮
real(ndp):: h_int( 1: size(x,1))
!Define the "optimal" h: h ∼ ϵ^(1/3) ∗ x
h(:) = error**(1.0d0/3.0d0) * x(:)
!Compute the elements of the Jacobian:
do jj = 1, size(x,1)
    h_int(:) = 0.0d0
    h_int(jj) = h(jj)
    Grad(1,jj) = (func(x + h_int) -&
    func(x - h_int)) / (2*h(jj))
enddo
```

My Code - Testing

OLS:

- See Fortran Code
- Idea:
  - Randomly generate some independent variables and an error term
  - Pick some coefficients
  - Compute the dependent variables
  - Run a regression
  - Check:
    1. Are hand picked coefficients and estimated ones similar?
    2. Are the results for coefficients and SE identical in your code and R?

My Code - Results and Outlook

- Results:
  - Estimation works ▸ Parameters
  - Standard Error Estimation works
  - Counterfactuals work
- Outlook
  - Consumption Equivalent Variation missing
  - Adopt code for married couples
  - Increase number of simulations $\rightarrow$ Issue: Stack Overflow
  - Maybe open MPI implementation?

APPENDIX

French Correction

For selection into private DI

1. Feed in the (potentially) biased regression coefficients from data
2. Run model and estimate same regression for simulated data for:
   - All individuals
   - Those purchasing private DI (selected sample)
3. Compute the difference between the coefficients from 2.)
4. Correct the initial coefficients based on this difference
5. Repeat until convergence

Back

Results - Parameter Estimation

Parameter	Estimation Result
γ	1.85625973375625
κ	0.622325041885433
θ	0.169797166835680
φ	0.382637889510906

Results - Parameter Estimation

Description	Data Moment	Simulated Moment
Mean private DI	48.90139	44.9
Mean private DI (25th)	26.04798	22.13
Mean private DI (50th)	53.36748	40.13
Mean private DI (75th)	56.80155	49.27
Mean private DI (100th)	63.19934	68.04
LFP (35 - 39)	96.20996 9	95.99
LFP (40 - 44)	92.49184	93.58
LFP (50 - 54)	89.08864	89.71
LFP (55 - 59)	74.88556	86.48
NP (35-39)	11.07029	40.5
NP (45-49)	49.55922	46.60
NP (55-59)	55.12941	49.41
NP (60-64)	69.38978	66.05

Back