Supplementary Material for Survival Pessimism and the Demand for Annuities

Cormac O'Dea* David Sturrock†

March 8, 2021

This is the readme associated with the replication files for O'Dea, C. & Sturrock, D. "Survival Pessimism and the Demand for Annuities". Section 1 gives an overview of the data we use and how to apply for access to it. Section 2 gives details of the code, which is uploaded along with this Supplementary Material.

1 Data Access

The data used in this paper are:

1. The publicly available version of the English Longitudinal Study of Ageing (ELSA), specifically:
   (a) ELSA core data, waves 1 to 7
   (b) ELSA IFS derived variables, waves 1 to 7
   (c) ELSA financial derived variables, waves 1 to 7
   (d) ELSA pension wealth variables, waves 2 to 5
   (e) ELSA index file, waves 0 to 5
2. ELSA datasets with restrictions around access, specifically:
   (a) ELSA linked mortality data
   (b) ELSA date of birth

*cormac.odeayale.edu
†david.sturrockifs.org.uk

(c) ELSA interview date, waves 1 to 7

(d) ELSA household grid, waves 1 to 7

3. UK Office for National Statistics (ONS) 2014-based Life Tables for England and Wales

All of the datasets listed under point 1 above can be downloaded from the UK Data Service by creating a user account and registering a project using this data: `https://beta.ukdataservice.ac.uk/datacatalogue/series/series?id=200011`. The zipped file downloaded will contain each of the datasets listed, for each of the waves it is available.

The data listed under point 2 can be accessed only with special permission. Those wishing to access the data listed under point 2 must make a request to the NatCen Social Research Data Release Panel. Please contact the ELSA Data Manager at elsadata@natcen.ac.uk for details of how to do so. We would be very happy to engage with any interested authors who wish to try and use the datasets listed above. Please feel free to email us to start a conversation about this.

The ONS life tables can be downloaded from
`https://www.ons.gov.uk/peoplepopulationandcommunity/birthsdeathsandmarriages/lifeexpectancies/datasets/lifetablesprincipalprojectionenglandandwales`.
We have included the downloaded version of this data in the replication files.

2 Code

This section gives detail on the code used to create our results. Section 2.1 sets out the preparation and analysis of the ELSA data and creation of the ELSA-based inputs for the dynamic model. Section 2.2 sets out the code for the dynamic model. Section 2.3 sets out the code used to analyse the model outputs and create the results figures.

2.1 ELSA data preparation and analysis

The descriptive analysis of the ELSA data and its prepration for use in the dynamic model are conducted in Stata. This subsection describes the steps of this data preparation and analysis, which (i) produces the results in Sections 2 and 3 of the paper (ii) produces figures 1 to 5 in the paper, and (iii) produces the ELSA-based inputs for the dynamic model.

To replicate the analysis, one can create a folder structure as included the in "Stata" folder of these replication files and described in "Master.do" and then run the dofiles below in the order specified in the file "Master.do". We now describe the contents of each .do file.

1. `00 CombinedMortalityRecord`: cleans and prepares the linked mortality data
2. `01 ConstructHealthVars`: creates a cleaned set of variables respondents' health outcomes. Note that this dofile calls dofiles in the folder "HealthDynamics".
3. `02 Basedata`: compiles the variables from the various ELSA datasets that are used in the analysis and prepares them as required
4. `03 Fitted ELSA survival curves`: creates the 'scaled' survival curves using ELSA mortality data
5. `04 Descriptives`: conducts the descriptive analysis of sections 2 and 3
6. `05a Make parameters`: prepares the parameters for the dynamic model
7. `05.1 calc state pension age`: this file is called by the preceding file. It calculates the exact date that an individual becomes eligible for the state pension, given their exact date of birth
8. `05b Make parameters unscaled`: prepares a version of the parameters for the dynamic model using the 'unscaled' version of the ONS life tables

2.1.1 Dependencies

Running the files called by "01 ConstructHealthVars.do" requires that two .ado files are added to the user's ado path: "imputesk.ado" and "tabgenlabel.ado". These ado files are in the folder *"HealthDynamics/Health adofiles"*.

2.1.2 Outputs

All of the outputs from the above files which are used in the paper are saved in the folder called "Outputs". Each Figure X is saved as "FigX.pdf". Some numbers quoted in the text are saved in the excel spreadsheet "Section 2.xls". The inputs for the dynamic model are saved in the folder "Fortran inputs".

2.2 Dynamic Model

2.2.1 Code Structure

The model is solved using Fortran, supported by the Numerical Algorithms Group libraries. The code uses NAG Mark 26 libraries. The Fortran Code contains 8 Fortran `.f90` files, each containing a single

module, in turn containing several subroutines and functions. These are:

1. `globalvalues.f90`: sets globals (described below)
2. `main.f90`: main files
3. `precisions.f90`: set the kind types
4. `routines_generic.f90`: A set of routines, which are generic in that they do not need `globalvalues.` but they do need `precicions.f90`
5. `routines_specific.f90`: A set of routines, which do need `globalvalues.f90` in addition to `precisions.f90`
6. `types_generic.f90`: A set of derived types, which are generic in that they do not need `globalvalues.f90` but they do need `precisions.f90`
7. `solveVSL.f90`: Solves the model by value function iteration
8. `simulate.f90`: Simulates behavior

2.2.2 Dependencies

NAG library Some code relies on the (commerical) numerical library sold by the Numerical Algorithms Group. The commands in the attached code are those from the library's Mark 26 (the precise implementation we use is FLW6I26DE but any Mark 26 version of the library should work).

OpenMP The code works in parallel using OpenMP, with the relevant instruction at lines 41 and 96 in `solveVSL.f90`. However, if a user does not flag OpenMP to be on, the code will run and will not be affected, other than the fact that it will be substantially slower.

2.2.3 Setting model parameters and reading data in

Data is fed to the Fortran program through three channels. The first is through globals declared in file `globalvalues.f90`. The second is through text files that are read in that set parameters (e.g. the discount factor, the coefficient of relative risk aversion). We describe each of these in turn.

Globals File `globalvalues.f90` sets some globals. Most importantly:

1. `numParamsCombGlob` sets the number of parameter combinations the model is solved for. In total 411 different combinations of solved for (most of these are the various pairwise combinations of β and γ in Figure 6 and 7). We describe the specification of each of these below.
2. `numPeeps` sets the number of people for which each specification is solved
3. `assGlob`, `annPointsGlob` set the number of grid point respectively for each of assets and annuities
4. `annAdminLoadGlob` and `rGlob` sets the administrative load and interest rate respectively

Parameterization The model is solved for 411 distinct parameter combinations. These are read in through the 411-row text file `inputParamsMain.txt` Each row contains 8 columns. These are, respectively:

1. β
2. γ
3. An indicator for whether objective or subjective survival probabilities should be used in the lifecycle problem
4. An indicator for whether the administrative load is turned on or off
5. An indicator for whether housing income is considered or not (in the paper it is not, the appendix contains results where it is)
6. An indicator for whether the annuity is turned on or off (this is turned on everywhere in the paper so is 1 everywhere in this text file)
7. An indicator for whether the scaled or unscaled survival probabilities are used (scaled in the paper, unscaled reported in the appendix)
8. An indicator for whether annuity purchase is binary or continuous (continuous in the paper, binary results reported in the appendix)

The text file `inputParamsMain.txt` is included in our replication file, but it can also be generated using a `Matlab` also included. This `Matlab` file, which in turn, relies on a user-written command written by David Fass. This is not included in the replication files, but is available for download[1]. Note this file does not need to be used unless a researcher wants to solve the model for parameter combinations we don't consider in the paper.

These parameters and parameter switches are read in to the Fortran code by subroutine `readinparams` in module `routines_specific.f90`.

Individual Level Data Finally, the Fortran code reads in data at the individual level. These are `annInputs.txt` and `annInputsUnscaled.txt`. These contain one row for each of our sample members. The columns contain, in order:

1. k in the Weibull formula
2. λ in the Weibull
3. The sample member's age
4. The annuity rate they would face
5. An unique identifier
6. Their state pension entitlements
7. Their asset level
8. For 51 columns, their (objective) survival hazards at each age, which differ in scaled and unscaled versions
9. In the final column, imputed housing consumption for the version of the model in the appendix where we consider housing

This data is read in to the Fortran code by subroutine `readinparams` in module `routines_specific.f9`

[1] `https://www.mathworks.com/matlabcentral/fileexchange/5475-cartprod-cartesian-product-of-multiple-sets`

2.3 Code for Analysis of the Model Output

The Fortran Code produces output that is deposited in a directory set in the subroutine `getpaths` in module `globalvalues`. For this to work, inside that directory there will need to be numbered subdirectories ranging from 1 to 411 where each of the 411 parameter combinations will be deposited. The relevant file is `annPropmean.txt`

The outputs deposited in these folders are then analysed by the following Stata dofiles. These are called by the same "Master.do" file that was described in section 2.1. We now describe what these files do, in turn. To replicate this analysis, users should set the global variable "simMasterPath" to be file path for the folder containing the 411 folders described above (as set out in "Master.do").

1. `06 Gather fortran outputs.do`: compiles the model outputs
2. `07 Analyse fortran outputs.do`: analyses the model outputs and creates the results figures (figures 6, 7 and 9) and table 5.
3. `08 Appendix.do`: conducts the additional analysis included in the paper's appendix, including tables 1, 2, 3 and 4 and figure 8.

2.3.1 Outputs

All of the outputs from the above files which are used in the paper are saved in the folder called "Outputs". Figures 6, 7 and 9 are saved in the excel spreadsheets named "Fig6a", "Fig6b",..., "Fig9b". Each of these spreadsheets includes a tab with the exported results and a tab which creates the figure. Table 1 is exported to the excel file "Appendix.xls". Tables 2, 3, 4 and 5 are exported as .tex files.