Instructions for code accompanying "The Decision to Move House and Aggregate Housing-Market Dynamics"

L. Rachel Ngai and Kevin D. Sheedy
London School of Economics

December 2019

Prerequisites

The data are provided as *Excel* spreadsheets and the code is written for use in *MATLAB*.

Contents

Data files

Sources and construction are described within the file `Data.xlsx`.

- `DurationDistribution.xlsx`: AHS data on the numbers of homeowners by the year they moved in.
- `HousingStock.xlsx`: AHS data on the housing stock.
- `Data.xlsx`: Full set of data with details of sources and construction.
- `NotRented.xlsx`: AHS data on the fraction of homes that are not rented.
- `RealHPI.xlsx`: FHFA data on house prices (converted to real terms using PCE price index).
- `SalesInventory.xlsx`: NAR data on housing sales and inventories.

Main code

- `calibration.m`: Implements the calibration procedure described in Appendix A.11 and saves the model parameters.
- `compstat.m`: Performs comparative statics exercises using the model with the calibrated parameters.
- `dataanalysis.m`: Implements the empirical work in Section 2 of the paper.
- `onetypecalibration.m`: Implements the calibration procedure for the one-type model described in Appendix A.12 and saves the model parameters.
- `overshooting.m`: Computes the maximum transitional effects from the impulse response functions and ratios of these maximums to the long-run effects.
- `plothazardfunction.m`: Plots empirical and model-implied hazard functions with the calibrated parameters.
- `plotimpulseresponses.m`: Plots impulse response functions using the calibrated parameters.

Subroutines

- `calibcrit.m`: Return criterion used in calibration procedure.
- `cfsales.m`: Compute counterfactual transactions series following the procedure described in Section 2 of the paper.
- `empiricalhazard.m`: Computes the empirical hazard function from data on the numbers of homeowners by year they moved in.
- `findlambda.m`: Solve for λ parameter in calibration procedure.
- `hazard.m`: Computes the hazard function implied by the model for given parameters.
- `hazdist.m`: Computes the distance between the empirical and model-implied hazard functions.
- `modelcrit.m`: Returns criterion used to solve for the transaction threshold in the model.
- `onetypecalibcrit.m`: Return criterion used in calibration procedure for the one-type model.

- `onetypecalibfunc.m`: Return calibrated parameters using the procedure for the one-type model.
- `paramcalib.m`: Computes parameters matching calibration targets conditional on knowing the probability distributions of the shocks to match quality.
- `seasonadj.m`: Performs seasonal adjustment of data.
- `solvemodel.m`: Solves the model for given parameters and computes predictions of the model.
- `survival.m`: Computes the survival function implied by the model for given parameters.

Parameters

- `paramendog2.mat`: Calibrated parameters of model with endogenous moving and two types (Model I in Table 2).
- `paramendog1.mat`: Calibrated parameters of model with endogenous moving and one type (Model III in Table 2).
- `paramexog2.mat`: Calibrated parameters of model with exogenous moving and two types (Model II in Table 2).

Replication

The numerical results in the paper and appendix can be replicated as follows:

- Figure 1: Call `dataanalysis.m`; data from graphs are plotted and returned in variables `fig1a` and `fig1b`.
- Figure 2: Call `dataanalysis.m`; data from graph is plotted and returned in variable `fig2`.
- Figure 3: Call `dataanalysis.m`; data from graph is plotted and returned in variable `fig3`.
- Figure 4: Call `dataanalysis.m`; data from graph is plotted and returned in variable `fig4`.
- Figure 5: Direct plot of the data in *DataFig5* from file `Data.xlsx`.
- Figure 9: Call `plothazardfunction.m`; graph is plotted and hazard functions in the model and data are returned in variables `haz` and `hazemp` respectively.

- Table 1: Calculations are described in the text of the paper.
- Table 2: Each column generated as below with results returned in variable `paramvec`.
  - Column 'Model I': Call `calibration.m` setting `q=2` and `exog=false`.
  - Column 'Model II': Call `calibration.m` setting `q=2` and `exog=true`.
  - Column 'Model III': Call `onetypecalibration.m`.
- Table 3, by rows:
  - Call `compstat.m` for the model simulations; results are returned by row in variables `resp{1}`, `resp{2}`, `resp{3}`, `resp{4}`, and `resp{5}`.
  - Call `dataanalysis.m` for the data row; results are returned in variables `chgsalesadjstock9503`, `chglistingsadjstock9503`, `chgsalesrate9503`, `chglistingrate9503`, `chgfracforsale9503`, and `chgprice9503`.
- Table 4, by rows:
  - 1st row: Call `compstat.m` with the line `load('paramendog2.mat');` results are returned in variable `resp{4}`.
  - 2nd-4th rows: Call `compstat.m` with the line `load('paramexog2.mat');` results are returned by row in variables `resp{4}`, `resp{12}`, and `resp{13}`.
  - Data row: See Table 3.
- Figure 10: Call `plotimpulseresponses.m`; results from graphs are plotted and returned in variable `impresp`.
- Table 5: Call `overshooting.m`; results are returned in variables `maximpresp` and `maximprespratio`. See Table 3 for data row.
- Table 6, by rows:
  - 1st row: Call `compstat.m` with the line `load('paramendog2.mat');` results are returned in variable `resp{4}`.
  - 2nd row: Call `compstat.m` with the line `load('paramexog2.mat');` results are returned in variable `resp{4}`.
  - Data row: See Table 3.
- Table 7, by columns:
  - Model columns: Call `compstat.m`; results are returned by column in variables `resp{6}` and `resp{7}`.

  - Data columns: Call `dataanalysis.m`; results are returned in variables `chgsalesadjstock9506` and `chgprice9506` for the first data column, and `chgsalesadjstock0709` and `chgprice9506` for the second data column.

- Figure 11: Call `dataanalysis.m` setting `notrentedadjustment=true`; data from graph is plotted and returned in variable `fig2`.

- Table 8: Call `compstat.m` for the model simulations; results are returned by row in variables `resp{8}`, `resp{9}`, and `resp{14}`. See Table 3 for data row.

- Table 9: Call `compstat.m` for the model simulations; results are returned by column in variables `resp{10}` and `resp{11}`. See Table 7 for data columns.

- Table 10: Call `calibration.m` for each change to the calibration targets; this saves the new model parameters in `paramendog2.mat`. Then call `compstat.m`; results are returned in variable `resp{4}`. See Table 3 for the baseline and data rows.