Final Seminar

Antoine Godin

26 October 2016

Objective

The goal of this final seminar is to get you ready for the final coursework to be submitted in 3 weeks. We will briefly discuss the different ways to calibrate/estimate a model and then try the various solutions to a (extremely) simple model: PCEX.

Steady/stationary state calibration

Assuming you want to start with a model at the steady/stationary state, one way of calibrating these model is by doing "forward deduction":

1. Find in the litterature (SFC or other) the values for the parameters you have in your model
2. Run the model, hoping it will converge towards the steady state
3. Fiddle as needed with some parameters such that the steady state meets your requirements

Steady/stationary state calibration

Assuming you want to start with a model at the steady/stationary state, one way of calibrating these model is by doing "backward induction":

1. First you need to change your model equation so that it represent a steady state model
    a. All lags are removed
    b. All expectation are equal to the corresponding steady state values
    c. Some equation have to be changed completely: i.e. assuming a change in wealth function ($V = V_{-1} + YD - C$), it becomes ($C = YD$) but then V is undefined. This implies that the consumption function ($C = \alpha_1 YD^e + \alpha 2 V_{-1}$) has to change as well so as to show the relationship between wealth and income ($V = \frac{1-\alpha_1}{\alpha_2} YD$)
2. Endogenise the parameters and exogenise some stock-flow norms or accepted values for specific indicators
    a. Endogenising the parameters imply that instead of having the wealth to income function ($V = \frac{1-\alpha_1}{\alpha_2} YD$), we now specify α_2 as being determined by this equation: $\alpha_2 = (1 - \alpha 1)\frac{YD}{V}$.
    b. Exogenising the stock-flow norms or accepeted indicators implies fixing debt-to-gdp ratios or unemployment rates to realistic values
3. Once you have enough equations for the number of parameters, and enough constraints (i.e. fixed values for stock-flow norms), the model should be determined and it is possible to solve the system and obtain both the parameters and initial values for the specific steady state. Note that you can use the PK-SFC package for that.

Out-of-Steady State calibration/estimation: massaged data

one of the problem with SFC models is that they explicitely constrain the data into a specific data structure. I.e. the balance sheets and other transaction flow matrices are sparse while it is not the case in reality. This means that you will have to massage the data so that it meets your requirement. There is a lot of subjectivity happening there. A pseudo-algorithm for this part would look like this.

1. Individuate the time series you want to be exactly like the real world (e.g. GDP, unemployment, Household wealth)
2. Individuate the time series you are ok to play the buffer: these will have to be constructed from the accounting structure of the model. E.g. if you have households having two type of assets: money and bonds and you want households wealth and bond holding to be based on data, then money will have to be constructed as wealth-bonds.
3. Once you have a stock-flow consistent data set: run regressions to compute the parameters
   a. I personally use OLS or dynamic OLS equation per equation
   b. You could use 2-stage LS or minimum likelihood to estimate the system as a whole
   c. Gennaro Zezza and others use co-integration methods
4. Initial conditions are determined by the data you have.
5. You need to say what you will do with residuals (coming out of your regressions): set them a 0, have them fixed at a non-zero value, have them decreasing through time?

Pure estimation

If you want to do a pure empirical model, then you have to start from the data and construct the model around it. You will need to look at how disaggregated the data is and what you can afford to do. A pseudo algorithm would look like this:

1. Aggregate/Disaggregate existing data sets such as it meets with the model structure you think is relavant *or* modify the model structure you want to have such that it fits with the data you have
2. Once you are ahhpy with the data and the model structure, run regressions to compute the parameters
   a. I personally use OLS or dynamic OLS equation per equation
   b. You could use 2-stage LS or minimum likelihood to estimate the system as a whole
   c. Gennaro Zezza and others use co-integration methods
3. Initial conditions are determined by the data you have.
4. You need to say what you will do with residuals (coming out of your regressions): set them a 0, have them fixed at a non-zero value, have them decreasing through time?

Exercise

We will try to apply the middle two techniques (Backward induction and massaged data) to model PCEX to the UK economy. You have the source code for the model in this folder, it is called PCEX_forCalibration. In order to speed up the process, I will show you the R code to for each case.

Backward induction

```r
library(PKSFC)

# Loading the Steady State version of the model: you need to write down that file.
# Bear in mind that you dont need to have more than two periods in the steady state
modelSS<-sfc.model("PCEXSS.txt")

# Solving the steady state
dataSS<-simulate(modelSS)

#Getting the value for all parameters and steady state values
calib<-as.data.frame(t(dataSS$baseline[2,]))

#What follows is a bit tricky but not too complicated to understand. What we will do is
```

```r
# to create a temporary file called calibration.txt based on the values obtained via the
# steady state model. This file will then be merged into the PCEX_forCalibration.txt file
# so as to dynamically create a complete model file (i.e. equations + calibration) which
# can then be simulated

#Creation of the calibration.txt file
sink("calibration.txt")
#######  PARAMETERS
cat(paste("alpha1=",calib$alpha1,"\n"))
cat(paste("alpha2=",calib$alpha2,"\n"))
cat(paste("lambda0=",calib$lambda0,"\n"))
cat(paste("lambda1=",calib$lambda1,"\n"))
cat(paste("lambda2=",calib$lambda2,"\n"))
cat(paste("theta=",calib$theta,"\n"))
#######  EXOGENOUS
cat(paste("g=",calib$g,"\n"))
cat(paste("r_bar=",calib$r_bar,"\n"))
#######  Starting values for stocks
cat(paste("b_cb=",calib$b_cb,"\n"))
cat(paste("b_h=",calib$b_h,"\n"))
cat(paste("b_s=",calib$b_s,"\n"))
cat(paste("h_h=",calib$h_h,"\n"))
cat(paste("h_s=",calib$h_s,"\n"))
cat(paste("r=",calib$r,"\n"))
cat(paste("v=",calib$v,"\n"))
cat(paste("yd=",caliby$d,"\n"))
sink()

# Now we will read the calibration file
calibrationLines<-readLines("calibration.txt")

#Next step is to read the model without calibration
modelLines<-readLines("PCEX_forCalibration")

#We need to determine where to add the calibration lines, this is done where
# "{CALIBRATION}" has been inserted in the PCEX_forCalibration.txt file
indexCalib<-grep("{CALIBRATION}",modelLines)

#Now we can merge the two files into a complete model
totModel<-c(modelLines[1:(indexCalib-1)],calibrationLines,
                        modelLines[(indexCalib+1):length(modelLines)])

#Write down the model into a txt file
writeLines(totModel,"PCEX_Auto.sfc")

#And load the model. The model is now ready to be simulated
WholeModel<-sfc.model("PCEX_Auto.sfc")
```

Massaged data

```r
library(PKSFC)
```

```r
#1. fetch all the relevant data sets for your case, using pdfetch
#getting the data in a raw format
NF_TR_raw = pdfetch_EUROSTAT("nasa_10_nf_tr", UNIT="CP_MNAC", NA_ITEM=c("..."),
                                           SECTOR=c("..."), DIRECT = "...")
#Transforming the data into a data.frame (easy to access and to view)
NF_TR<-as.data.frame(NF_TR_raw)

#2. Construct the various massaged time series
NF_TR$MASSAGED<-NF_TR[,"..."]+NF_TR[,"..."]-NF_TR[,"..."]

#3. Run the regression
model.cons<-lm(CONS[-1]~YD[-19]+V[-19]-1,data=NF_TR)
calib$alpha1<-coef(model.cons)[1]
calib$alpha2<-coef(model.cons)[2]

...

#What follows is a bit tricky but not too complicated to understand. What we will do is
# to create a temporary file called calibration.txt based on the values obtained via the
# steady state model. This file will then be merged into the PCEX_forCalibration.txt file
# so as to dynamically create a complete model file (i.e. equations + calibration) which
# can then be simulated

#Creation of the calibration.txt file
sink("calibration.txt")
#######  PARAMETERS
cat(paste("alpha1=",calib$alpha1,"\n"))
cat(paste("alpha2=",calib$alpha2,"\n"))
cat(paste("lambda0=",calib$lambda0,"\n"))
cat(paste("lambda1=",calib$lambda1,"\n"))
cat(paste("lambda2=",calib$lambda2,"\n"))
cat(paste("theta=",calib$theta,"\n"))
#######  EXOGENOUS
cat(paste("g=",calib$g,"\n"))
cat(paste("r_bar=",calib$r_bar,"\n"))
#######  Starting values for stocks
cat(paste("b_cb=",calib$b_cb,"\n"))
cat(paste("b_h=",calib$b_h,"\n"))
cat(paste("b_s=",calib$b_s,"\n"))
cat(paste("h_h=",calib$h_h,"\n"))
cat(paste("h_s=",calib$h_s,"\n"))
cat(paste("r=",calib$r,"\n"))
cat(paste("v=",calib$v,"\n"))
cat(paste("yd=",caliby$d,"\n"))
sink()

# Now we will read the calibration file
calibrationLines<-readLines("calibration.txt")

#Next step is to read the model without calibration
modelLines<-readLines("PCEX_forCalibration")

#We need to determine where to add the calibration lines, this is done where
```

```r
# "{CALIBRATION}" has been inserted in the PCEX_forCalibration.txt file
indexCalib<-grep("{CALIBRATION}",modelLines)

#Now we can merge the two files into a complete model
totModel<-c(modelLines[1:(indexCalib-1)],calibrationLines,
                    modelLines[(indexCalib+1):length(modelLines)])

#Write down the model into a txt file
writeLines(totModel,"PCEX_Auto.sfc")

#And load the model. The model is now ready to be simulated
WholeModel<-sfc.model("PCEX_Auto.sfc")
```