Adding Trends

###Setting up the environment

This tutorial will show you how to use exogenous trends into an SFC model, using the model SIMEX as an example.

Before doing any modelling, we need to load the package in the R environment.

```
library(PKSFC)
```

```
## Loading required package: expm
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'expm'
```

```
## The following object is masked from 'package:Matrix':
##
##     expm
```

```
## Loading required package: igraph
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```
## Loading required package: networkD3
```

The, you need to download the attached 'SIMEX.txt' file and save it in the folder of your choice. Make sure to set the working directory where you saved the downloaded file. In command line this looks like this but if you use RStudio, you can use the graphical interface as well (Session>Set Working Directory>Choose Directory)

```
setwd("pathToYourDirectory")
```

###Loading the model

The first thig to do is to load the model.

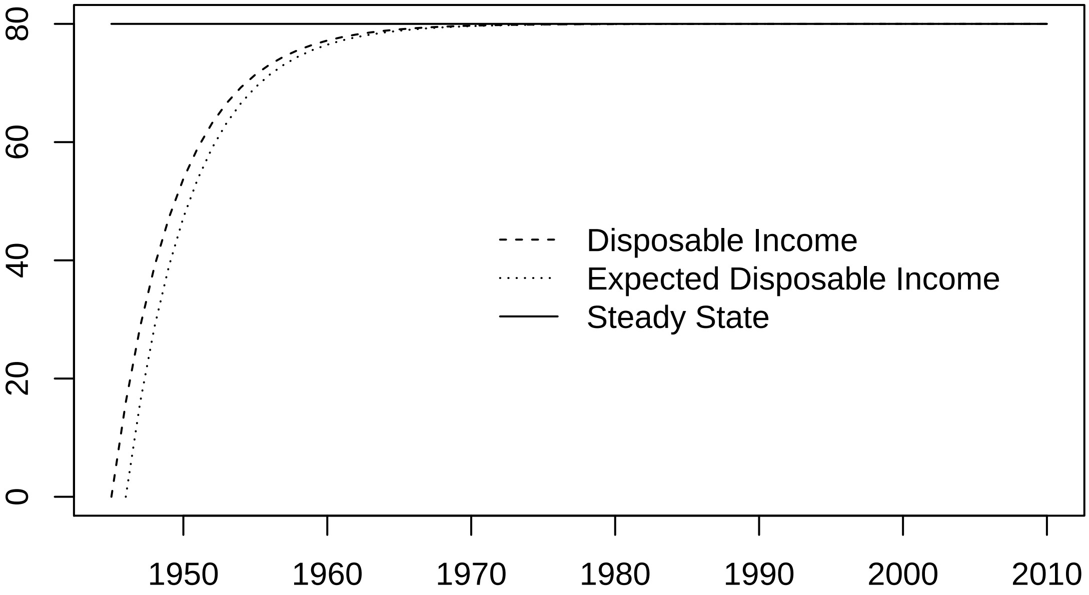
```
simex<-sfc.model("SIMEX.txt",modelName="SIMplest model")
```

We are now ready to simulate the model

```
datasimex<-simulate(simex)
```

Let's plot some results

```
plot(simex$time,datasimex$baseline[,"Yd"],type="l",xlab="",ylab="",lty=2)
lines(simex$time,datasimex$baseline[,"Yd_e"],lty=3)
lines(simex$time,vector(length=length(simex$time))+datasimex$baseline["2010","Yd"])
legend(x=1970,y=50,legend=c("Disposable Income","Expected Disposable Income","Steady State"),
                                                lty=c(2,3,1),bty="n")
```

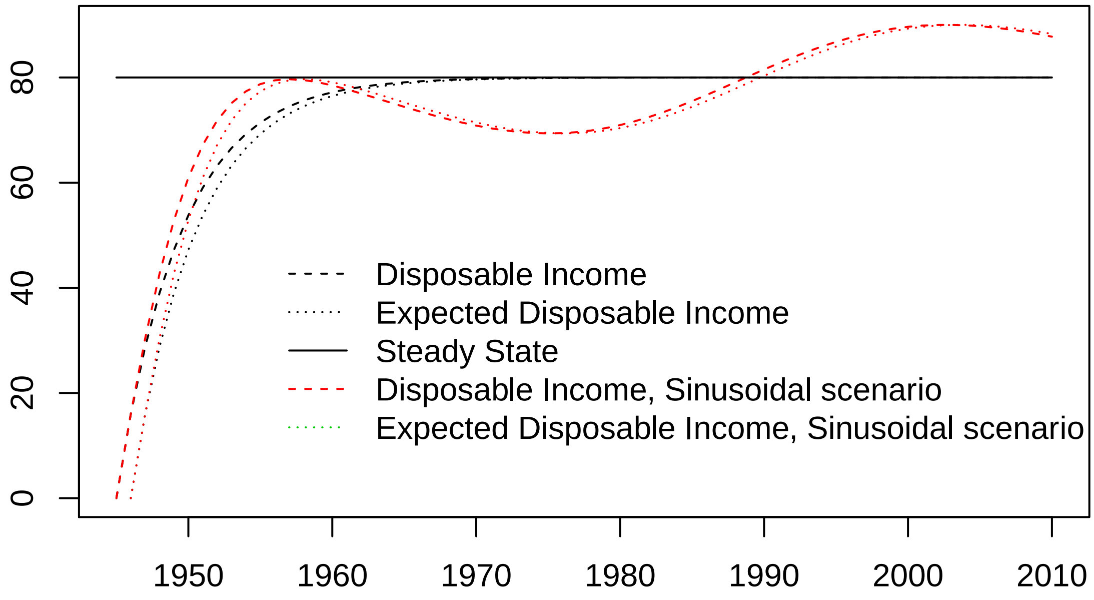


Let's imagine now, that instead of using a constant value for the propensities to consume you want to introduce a moving value, say a sinusoidal around their initial value. We can generate a new scenario with trended values using the option `trends` as follows:

```
vars= c("alpha1","alpha2")
trend = matrix(0,nrow=66,ncol=2)
trend[,1] = 0.6+0.1*sin(0.1*seq(1:66))
trend[,2] = 0.4+0.1*cos(0.1*seq(1:66))
simex <- sfc.addScenarioTrend(simex,vars=list(vars),trends=list(trend))
```

We can then simulate the baseline and the scenario and look at the corresponding results:

```
datasimex = simulate(simex)
matplot(simex$time,cbind(datasimex$baseline[,c("Yd","Yd_e")],datasimex$scenario_1[,c("Yd","Yd_e")]),typ
lines(simex$time,vector(length=length(simex$time))+datasimex$baseline["2010","Yd"])
legend(x=1955,y=50,legend=c("Disposable Income","Expected Disposable Income","Steady State","Disposable
```



Let's say now that you don't quite know the exact value of the tax rate and want to run Monte Carlo simulations around the mean value adding a certain stochasticity around it. In order to do so, we are going to change the `baselineMat` argument of the model and change it for each Monte Carlo simulation. The `baselineMat` is the matrix that contains the values for each parameter and each variable and for each period of simulation. In a way, it is the matrix that will be filled as the simulations run. These are the first 10 lines for the `simex` model:

```r
library(knitr)
kable(simex$baselineMat[1:10,c("alpha1","alpha2","G_d","theta")])
```

	alpha1	alpha2	G_d	theta
1945	0.6	0.4	20	0.2
1946	0.6	0.4	20	0.2
1947	0.6	0.4	20	0.2
1948	0.6	0.4	20	0.2
1949	0.6	0.4	20	0.2
1950	0.6	0.4	20	0.2
1951	0.6	0.4	20	0.2
1952	0.6	0.4	20	0.2
1953	0.6	0.4	20	0.2
1954	0.6	0.4	20	0.2

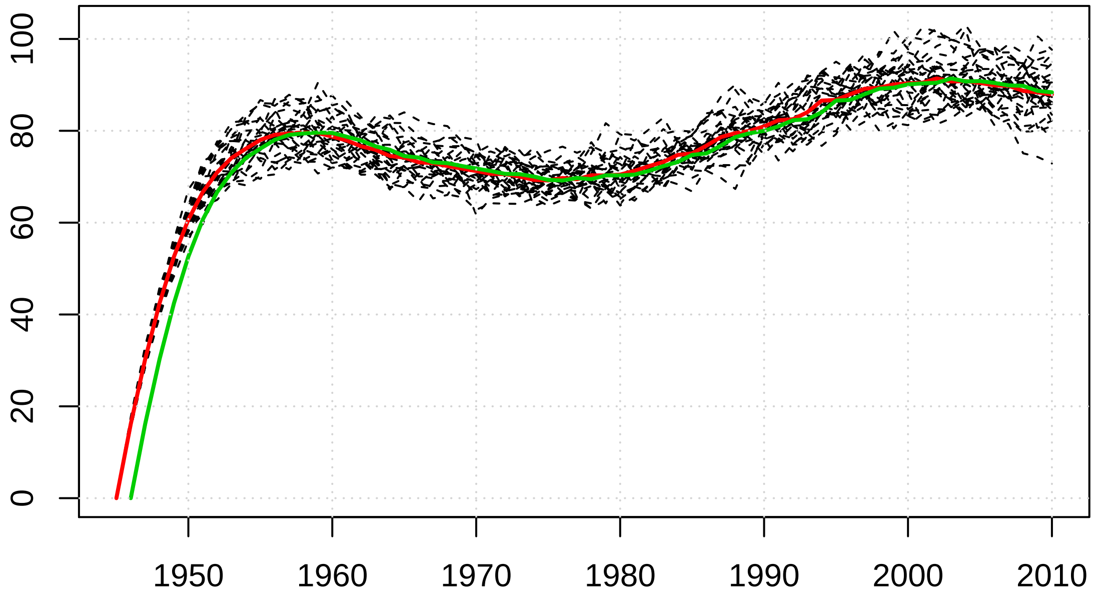
We can see that in the baseline model, the value of the parameters are constant. Now we can change this by loading the model with a trend. This is done like this

```
simex<-sfc.model("SIMEX.txt",modelName="SIMplest model", trends=trend, trendsVars=vars)
kable(simex$baselineMat[1:10,c("alpha1","alpha2","G_d","theta")])
```

	alpha1	alpha2	G_d	theta
1945	0.6099833	0.4995004	20	0.2
1946	0.6198669	0.4980067	20	0.2
1947	0.6295520	0.4955336	20	0.2
1948	0.6389418	0.4921061	20	0.2
1949	0.6479426	0.4877583	20	0.2
1950	0.6564642	0.4825336	20	0.2
1951	0.6644218	0.4764842	20	0.2
1952	0.6717356	0.4696707	20	0.2
1953	0.6783327	0.4621610	20	0.2
1954	0.6841471	0.4540302	20	0.2

We can see that the first ten values are now changing through time. Let's get back to our exercise and try to change the value of the tax rate (`theta` parameter). We are going to record only the value for disposable income (`Yd`) and expected disposable income (`Yd_e`) for simplicity. We are going to use a normal distribution with mean equal to 0.2 (the standard value of the tax rate in SIMEX) with a standard deviation of 0.025. To extract random value out of a normal distribution, we use the function `rnorm(n,mean=a, sd=b)` where `n` is the number of extractions, `a` is the mean and `b` is the standard deviation.

```
simex<-sfc.model("SIMEX.txt",modelName="SIMplest model", trends=trend, trendsVars=vars)
MC=30
Yes=matrix(NA,nrow=length(simex$time),ncol=MC)
Yds=matrix(NA,nrow=length(simex$time),ncol=MC)
for(i in 1:MC){
    simex$baselineMat[,"theta"]=rnorm(length(simex$time),mean=0.2,sd=0.025)
    datasimex=simulate(simex)
    Yes[,i]=datasimex$baseline[,"Yd_e"]
    Yds[,i]=datasimex$baseline[,"Yd"]
}
matplot(simex$time,Yds,type='l',lty=2,col=1,xlab="",ylab="")
lines(simex$time,rowMeans(Yds),lwd=2,col=2)
lines(simex$time,rowMeans(Yes),lwd=2,col=3)
grid()
```



The black dashed lines are the values of disposable income for each of the Monte Carlo simulation while the red solid one is the average of disposable income over all the simulation and the green solid one is the average of expected disposable income.