∫ IRIS Macroeconomic Modeling Tutorials

READ AND SOLVE MODEL

by Jaromir Benes

January 19, 2016

Summary

Create a model object by loading the model file `Simple_SPBC.model`, assign parameters to the model object, find its steady state, and compute the first-order accurate solution. The model object is then saved to a mat file, and ready for further experiments.

CONTENTS

1 CLEAR WORKSPACE

Clear workspace, close all graphics figures, clear command window, and check the IRIS version.

```
clear;
close all;
clc;
irisrequired 20140315;
```

2 READ MODEL FILE AND CREATE MODEL OBJECT

The function `model` reads the model file and translates it into a model object, called here `m`. Model objects are complex structures that carry all the information needed about the model, and can be manipulated by calling some of the IRIS functions.

```
m = model('simple_SPBC.model');
m2 = model('simple_SPBC.model','symbdiff=',false);
```

3 CALIBRATE MODEL PARAMETERS

Assign parameters using the simplest possibly syntax: `m.XXX = YYY;` where `XXX` is the name of a parameters, and `YYY` is its value. Use the function `get` [1] to retrieve a database with currently assigned parameter values.

```
m.alpha = 1.03^(1/4);
m.beta = 0.985^(1/4);
m.gamma = 0.60;
m.delta = 0.03;
m.pi = 1.025^(1/4);
m.eta = 6;
m.k = 10;
m.psi = 0.25;

m.chi = 0.85;
m.xiw = 60;
m.xip = 300;
m.rhoa = 0.90;

m.rhor = 0.85;
m.kappap = 3.5;
m.kappan = 0;
```

```
53
54 m.Short_ = 0;
55 m.Infl_ = 0;
56 m.Growth_ = 0;
57 m.Wage_ = 0;
58
59 m.std_Mp = 0;
60 m.std_Mw = 0;
61 m.std_Ea = 0.001;
62
63 disp('Get a parameter database from the model object');
64 get(m,'parameters') 1
```

4 COMPUTE STEADY STATE

Compute and numerically check the steady-state values for all model variables. The option 'growth=' true 2 says this is a non-stationary BGP model where variables can grow at a constant rate over time (the steady-state solution is modified to handle such models). The option 'blocks=' true 3 allows to explore the steady-state structure of the model, and makes the numerical solution more efficient by splitting the system of steady-state equations into smaller recusrive blocks clusters.

```
77 m = sstate(m,'growth=',true,'blocks=',true); 2 3
78 chksstate(m);
79
80 m2 = assign(m2,m);
81 m2 = sstate(m2,'growth=',true,'blocks=',true); 2 3
82 chksstate(m2);
```

5 COMPUTE FIRST ORDER SOLUTION

```
86 m = solve(m);
87
88 disp('Solved model')
89 m %#ok<NOPTS>
```

6 SAVE MODEL OBJECT

Save the solved model object to a mat-file (binary file) for future use.

```
95 save read_model.mat m;
```

7 HELP ON IRIS FUNCTIONS USED IN THIS FILE

Use either help to display help in the command window, or idoc to display help in an HTML browser window.

```
help model/model
help model/subsasgn
help model/assign
help model/sstate
help model/chksstate
help model/solve
```