Chapter 2: Overview of GROCER

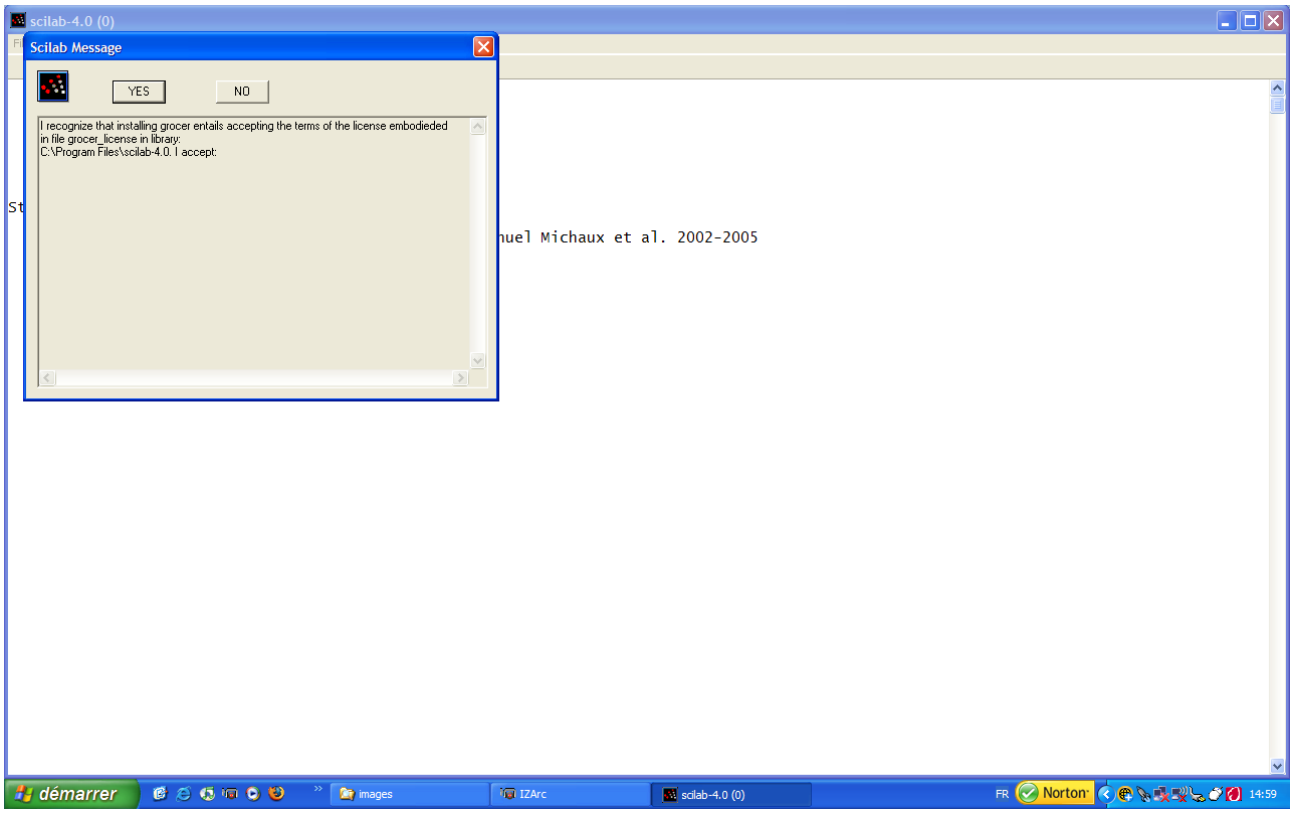
This chapter presents the main characteristics and the principles of GROCER. The next chapters give a detailed description of GROCER programs and functions,

A first part describes how to install grocer. A second part provides a short description of GROCER main characteristics and how they compare to the usual econometric products. The third part describes briefly the content of the following chapters. The fourth part is the most important and even essential. It explains the main principles which govern GROCER and its reading is thus necessary to use correctly and efficiently GROCER. Part five presents an extensive example. Part six exposes what are the developments planned for the near future. Part seven deals with bugs and contributions and part eight says how to cite GROCER if you want to do so in your own work.

1. How to install GROCER

To install GROCER, you must download, at http://www-rocq.inria.fr/scilab/contributions.html or at http://dubois.ensae.net/grocer.html, the zip file called grocer_1.2_for_sci: this version can work with Scilab 3.0, Scilab 3.1.1 and Scilab 4.0. Note however that, because of the incompatibility between grocer and Scilab 3.0 version of the new graphic mode, grocer under Scilab 3.0 uses the old graphic mode. Unzip the file in your Scilab directory. This operation creates the .sci files associated with GROCER under the Scilab directory “macros/grocer” and the .html help files under the Scilab library “man/groc_man”. It also replaces the scilab.star file in your Scilab directory: this makes GROCER toolbox ready to use, but erases all modifications you could have made to this file. If you want to avoid erasing your own modifications, then delete the file scilab.star in the zip file and adapt your scilab.star file to incorporate the instructions pertaining to grocer that are signalled as such in scilab.star file that comes with your grocer zip file.

The first time you run Scilab after having unzipped GROCER, then you obtain the following screen:



If you do not obtain this screen, then it is probably because:
- either you have unzipped the file in another directory than Scilab one...
- or you have unzipped the file inScilab directory but not kept the folder name of each file (for instance c:/Program Files/scilab-4.0macros/grocer/ols.sci for the ols function if you have used the default installation procedure for scilab under Windows). Click the corresponding option in your zip programm so that these are kept.

If you obtain this screen, then read the license in the file grocer_license.txt in Scilab folder and if you accept the terms of this license, click on YES

Grocer begins to load. The operation takes a few seconds (but less than a minute on sufficiently recent computers).

When it is ended, the screen presents as follows:

```
                    scilab-4.0

          Copyright (c) 1989-2006
      Consortium Scilab (INRIA, ENPC)


Startup execution:
  loading initial environment
```

```
   installing Grocer 1.2 / Copyright Éric Dubois, Emmanuel Michaux et al.
2002-2006
  be patient...

Please, we would greatly appreciate if you could send us an e-mail at
grocer.toolbox(at)free.fr to inform us that you have installed grocer[1]


-->
```

It is asked to you to send an e-mail: it is not compulsory, but we would greatly appreciate it to support our efforts at developping grocer.

3) At the end of the operation, the file init.dat in the directory macros/grocer is modified. This file contains a flag grocer_initialized that is turned from %f to %t. So, the next time you will run scilab, then the following will appear on screen:

```
                    ____________________________________________
                                    scilab-4.0

                            Copyright (c) 1989-2006
                        Consortium Scilab (INRIA, ENPC)
                    ____________________________________________


Startup execution:
  loading initial environment
  loading Grocer 1.2 / Copyright Éric Dubois, Emmanuel Michaux et al. 2002-
2006

-->
```

.bin files are stored in Scilab library “macros/grocer”, but .sci are stored by theme in sub-libraries. So if you want to know how a function has been programmed, look at the .sci files in the corresponding sub-library. .html files are stored by chapter, each chapter being the counterpart of a chapter in this manual.

2. GROCER main characteristics or why to use or not use GROCER

What are GROCER characteristics that can lead you to use it as an econometric toolbox?

•first, it is totally free and opensource: the toolbox itself is provided for free with its sources; Scilab, the language it is written in, is also free and opensource; and Scilab works on LINUX as well as on Windows...

•second, it is portable: it works on the major existing O/S (Linux, Windows 95 to XP, Mac O/S); and since it is free, you can exchange programs with anyone that has an Internet

[1]To limit spam, the required symbol @ is replaced by (at): do not forget to do the reverse substitution.

connection; if your employer changes, you can continue using all tools you have developed in your previous job, whatever econometric software your employer uses;

•third, it contains most basic econometric methods that other econometric software have (see below); it even contains a package that mimics David Hendry's PC-gets commercial software and that no other software today embodies; and, unlike Gauss or matlab, it can manipulate explicitly time series;

•fourth, since Scilab is a matrix oriented application very much like Matlab or Gauss, it is easy to add your own programs, or even to adapt existing Matlab or Gauss programs. A great part of GROCER has been built that way.

•Fifth, although GROCER does not contain any simulation package, M. Julliard has adapted his famous package Dynare, originally written in Gauss, to Scilab; Dynare and GROCER are not yet linked[2], but with a little effort, you should be able to use GROCER to estimate equations of a model that you simulate thereafter with Dynare.

What are GROCER characteristics that can lead you not to use it as an econometric toolbox?

•first, it is not a click and button toolbox: the entry cost in GROCER is therefore greater that in click an button products, such as E-Views or Easyreg;

•second, GROCER execution speed is not as high as with some other products; if you have a very big database or to deal with problems that require time-consuming computations, then you should not choose GROCER; for most current uses, the time factor should however not be crucial;

•third, if you can not stand the few constraints it imposes to you in day-to-day use (see part 4 in this chapter).

To sum up:

•if money is not a problem for you and if you do not bother about portability:

- ✓for your estimation needs, E-views or PC-give are powerful enough and more user-friendly than GROCER;
- ✓for your simulation needs, TROLL is without any contest the best product on the market; its estimation capabilities are also respectable (particularly since the long awaited unit root and cointegration functions are available); it is however no more user-friendly and less flexible than Scilab and GROCER; and it is rather expensive;
- ✓if you need a matrix oriented language to develop your own econometric procedures, then Gauss is certainly the best, although Scilab is a valuable alternative.

•if your financial needs are very limited or portability is essential for you:

- ✓for your estimation needs, if you do not want to program your own econometric procedures and if you cannot give up using the clik-and-button device, then EASYREG should answer your needs;
- ✓for your estimation needs, if you want to be able to program your own programs or to adapt existing ones, and if you can dispense with the clik-and-button device, then Scilab and GROCER are certainly for you! And for your simulation needs you should use Dynare for Scilab.

[2] A proper interface between Dynare and GROCER is planned for further releases.

3. The content of the manual.

The next chapter deals with time series. It describes how time series are implemented in GROCER and how to manipulate them: unless you are ready to work only with vectors and matrices, it is indispensable to read this chapter before starting with GROCER.

Chapter 4 presents a number of basic functions. The reading of this chapter is highly recommended if you want to create your own programs. It is also necessary if you want to import series from EXCEL or export series to EXCEL.

Chapter 5 is devoted to optimisation. Its reading is recommended if you want to make your own likelihood maximisation programs.

Chapter 6 deals with single regression estimation: ordinary least squares, estimation of autocorrelated models, instrumental variables estimation, non linear least squares, logit, probit and tobit estimations, robust estimation, least absolute deviation regression, ridge regression, Theil-Goldbegrer. This chapter contains also a number of generalities about regressions (on the setting of time bounds for instance): it is thus also highly recommended to read this chapter.

Chapter 7 deals mainly with multicollinearity diagnostics (condition index Belsley, Kuh and Welsh variance decomposition), outlier diagnostics (max hat, dfbetas, studentized residuals and dffits) specification tests (autocorrelation, heteroskedasticity, normality, ARCH, RESET tests) and forecasting accuracy tests.

Chapter 8 presents unit root (Dickey-Fuller, augmented or not, Elliott, Rothenberg and Stock, Phillips and Perron, Schmidt-Phillips, Kwiatkowski, Phillips, Schmidt and Shin) and cointegration tests (CADF, Johansen).

Chapter 9 is devoted to multiple equation regressions: Zellner seemingly unrelated regressions, Two and Three Stage Least Squares, VAR models.

Chapter 10 deals with ARMA and VARMA estimations.

Chapter 11 deals with GARCH estimations.

Chapter 12 presents the Kalman filter and the application that has been implemented in GROCER: the time-varying parameters model.

Chapter 13 is devoted to the automatic regression package: an applied econometrician should save much time in his applied work with this package and should find worthwhile spending a little time reading this chapter.

Chapter 14 presents some distribution tools implemented in GROCER that were not at first implemented in Scilab: cumulative distribution function, p-value or number of degrees of

freedom for the hypergeometric, the lognormal and the logistic laws; random generator for these 3 laws and for a variety of other laws (beta, binomial, normal, chi-square, Fisher, gamma, Poisson, Student).

Chapter 15 presents the tools for business cycle analysis: standard filters (HP, Baxter-King and Christiano and Fitzgerald); spectral analysis; the Bry-Boscahn-Harding-Pagan datation method.

Chapter 16 presents the calculation of contributions, a tool that allows to split the evolution of an endogenous variable into the contributions of the values, past and present, of its exogenous determinants.

Chapter 17 presents the panel data estimation methods available in grocer.

Chapter 18 presents the tools implemented to allow the disaggregation of time series into higher frequency ones.

Chapter 19 presents the estimation of Markov-switching models.

Chapter 20 presents the Bayesian Model Averaging method.

Lastly, chapter 21 deals with graphs and printings of results: this chapter contains a presentation of Scilab graphing capabilities and the graph functions that have been developed in GROCER, particularly in order to graph time series; it explains also how to use the subroutines developed to print regression results.

4. GROCER main principles

To build GROCER, we have applied a series of principles which seemed convenient to us and which impact strongly its day-to-day use: so, it seems to us necessary to expose them to you before you begin working with GROCER. You can disagree with some of these principles, find them ineffective, dangerous or tedious. In some cases, you can overturn the corresponding principles (case of the third principle for instance) or overrule them by the mean of a little programming (case of the fifth principle). For some other principles, overrule them would be a daunting task: if they do not appeal to you, then you should better not use GROCER.

4.1. First principle: every task is performed by the mean of commands in Scilab command window

As already mentioned, GROCER, as Scilab, is not a click and button software. So, every task must be performed by typing a command, which refers to a GROCER function or a Scilab command. For instance, if you want to perform an ordinary least squares regression involving variables such as conso, inc and cte, you will not have to click on the names of the variables conso, inc and cte, but use the GROCER function <u>ols</u> and type:

```
--> ols('conso','inc','cte')
```
[3]

Similarly, if you want to multiply the (mxn) matrix A by the (nxp) matrix B, you will use the Scilab operator "*" and type

```
--> A*B
```

4.2. Second principle: (almost) every econometric function can work with vectors or time series

Econometrics deals basically with vectors and matrices. For instance, an ols regression involves the famous textbook formula $(X'X)^{-1} X'Y$, where X is a (Nxp) matrix and Y a (Nx1) vector. In GROCER, every econometric function can work with matrices and vectors as in the textbook.

If X and Y are respectively a -Scilab- (Nxp) matrix and a -Scilab- (Nx1) vector, then performing ols involves the following command:

```
--> ols(Y,X)
```

When working with macroeconomic data, it is however more convenient to work not with vectors, but with time series, that is vectors associated with the dates to which the values of this vector refers. So in GROCER, as in many other econometric software, you can apply most econometric functions to time series.

If X and Y are now time series (say, annual data from 1970 to 2002), then you can perform an ols regression exactly in the same way than with matrices:

```
--> ols(Y,X)
```

There is a well-known specificity of econometrics with time series (ts): you can, and sometimes you have to (if the ts do not cover exactly the same period), specify the estimation bounds. This is done in grocer with the command bounds(b1,b2) where b1 and b2 are the chosen estimation bounds.

Lastly, for almost every econometric function that can deal with vectors or time series, there is a more basic one (and a little bit more efficient) which works only with matrices and which does not provide all the options available in the more general one: these basic functions are not useful for an applied econometric purpose, but they can be useful if you want to write your own econometric functions. For instance, the function var, which implements VAR estimation, has a derivative called var1. Ols has even 3 derivatives: ols0, which calculates only the estimated coefficients; ols1, which provides also the t-stats, their p-values, the residuals, the var-covar matrix, SSR, SER; and ols2, which provides complete estimation, but without printings. Contrary to ols, all three functions ols0, ols1 and ols2 work only with matrices.

[3] Recall from chapter 1 that a character "-->" at the beginning of a line will indicate the beginning of a Scilab command, as it appears in Scilab command window.

4.3. Third principle: for time series, bounds are not necessary

If you do not want to specify your regression bounds, then you are not obliged to do it. GROCER will then choose the bounds corresponding to the maximum range for which the estimation can be performed. This feature is for very impatient users, but wewill recommend using it with care and, if you are risk adverse and not too impatient, to specify explicitly your bounds every time you perform a regression.

4.4. Fourth principle: if you want to keep the trace of your variable names, you can and have then to enter the names of your variables between quotes

As in many other languages, variables names in Scilab are lost when passed to a function. For instance, if y, x and z are three ts, and if you type the following command:

```
-->ols(y,x,z)
```

then variable y will be called by default (for instance, for the printing of the regression results) "endogenous", variable x "exogenous #1" and variable z "exogenous #2".

You can however keep trace of the name of your variables: to that end, you just have to enter the names of your variables between quotes. With the previous example, if you type the command:

```
-->ols('y','x','z')
```

then variable y will be called "y", variable x "x" and variable z "z".

You can even mix the two representations. If you type:

```
-->ols('y','x', z)
```

then variable y will be called "y", variable x "x" and variable z "exogenous #2".

4.5. Fifth principle: in GROCER functions, all the variables are local ones, except the bounds

In Scilab functions, a variable which is passed as an input or defined in the text of the function has a local statute. All other variables, that is all variables which are used in the function but not defined before in it, or not passed as an input, are considered as global, that is scilab looks in the calling environment (another function calling the function at stake or the environment in the open session) if a variable with the same name exists[4]. Global variables are of very dangerous use, so GROCER avoids largely using them. There are two exceptions.

The first one concerns the variables which are passed by their names: to transform the name of the variables, which is the argument passed to the function, into the variable itself, GROCER econometric functions make use of the Scilab command 'execstr', which meets the name of the variable and searches it. Since the variable itself has not been passed to the function (but a string which represents its name), the function searches it in the environment.

The second one concerns the bounds: the commands bounds creates a variable named

[4] Scilab allows one to declare a variable as global. In that case, a variable that is modified inside a function will also be modified for all future uses. If you do not declare it as global, then it can be modified within a function, but this modification will not be passed to the environment.

grocer_boundsvar in the environment where it is called. The call to an econometric function with time series leads the function to look for a variable grocer_boundsvar in the environment and if such a variable exists, to perform the regression with the corresponding bounds.

This choice is not without dangers: once bounds are created they remain valid in the environment and only in that environment: if you create bounds in a function, these bounds will not be created in Scilab environment; thus, a regression performed after the execution of that function will not use the bounds chosen in that function, but the ones existing (if they exist) in Scilab environment.

Although a little bit dangerous, this choice has been made for the sake of convenience. It is the same choice that have made Troll developers. The other choice would have been to make the user enter the bounds in every econometric function along with the names of the variables. This choice can still be made, with some programming effort. Do not hesitate to signal us if you think this effort is worthwhile.

4.6. Sixth principle: do not give names beginning with "grocer_"

As with every software, it is recommended not to give to your variables reserved names such as the names of existing functions (do not call a variable ols or optim, for instance, which is the name of a Scilab function!).

But it is also highly recommended not to give names beginning with "grocer_". Since GROCER functions can call global variables, it is fundamental that a local variable created by a function has not the same name as a global variable which will be used after: the function would in that case use the local variable instead of the global one which the user intended to be used. To avoid that risk, such local variables begin systematically with the prefix "grocer_". This is the reason why it is asked to you not to use to name variables with such a prefix.

4.7. Seventh principle: the results of a regression can be stored in a tlist

All the econometric functions print the main corresponding results on the screen. But you can also store the results in a tlist to retrieve them after during a session or even store them on disk. For instance, if you want to store the results of the ols regression of y on x and z, then type:

```
-->myresults=ols('y','x','z')
```

It is even possible to perform a regression without printing the results and print the results when you want. This is done by specifying 'noprint' in the function and using then printing module associated with the econometric function. For instance:

```
--> myresults=ols('y','x','z','noprint')
--> a=0;
--> prtruniv(myresults)
```

This sequence give the same results on the screen than the previous one (but it has also created a constant variable a, equal to 0).

4.8. Eighth principle: demos, that is functions suffixed by "_d", generally exist

Most functions existing in GROCER are presented in this manual with examples. You can also find demos, functions with a suffix "_d" after the name of the function (a convention taken from James LeSage; note that Scilab demos are simply suffixed by d, without the underscore; James Lesage convention seemed to us much more appealing, this is why we have adopted it at the detriment of consistency with Scilab).

5. A demo session

Once you have installed GROCER, you should have the following screen when starting Scilab:

```
_____________________________________________
                  scilab-4.0

          Copyright (c) 1989-2006
      Consortium Scilab (INRIA, ENPC)
_____________________________________________


Startup execution:
  loading initial environment
  loading Grocer 1.2 / Copyright Éric Dubois, Emmanuel Michaux et al. 2002-
2006

-->
```

A GROCER session could take the following form:

```
-->impexc2bd('SCI/macros/grocer/db/bdhenderic.csv',';','SCI/mylibrary/
bdhenderic.dat')
```

// import the Excel Csv file that contains the data taken from a Hendry and Ericsson (1991) // paper[5] into a Scilab database (the library mylibrary in your Scilab working library must // already exist; check also that the csv separator of my Excel version is ';', which you can // check by editing the file with a text editor; when you save your own Excel files in a csv // format, the separator can be different, ',' for example; in that case second argument of // impexc2bd should be changed, to ',' for example). See chapter 4.

```
-->load('SCI/mylibrary/bdhenderic.dat')
```

// Scilab command to load a database in your working space.

```
-->dblist('SCI/mylibrary/bdhenderic.dat')
 ans  =

!rnet  !
!      !
!lm1   !
```

[5] D.F Hendry et N.R Ericsson (1991): "Modeling the demand for narrow money in the United Kingdom and the United States", European Economic Review, p833-886.

```
!      !
!lp    !
!      !
!ly    !
```

// Display the content of the database. See chapter 4 for a number of basic functions such as
// dblist.

```
-->prtts('ly+lp-lm1','rnet')
dates  ly+lp-lm1 rnet
dates  ly+lp-lm1 rnet
1963q1 0.1334977 0.0431333
1963q2 0.1553029 0.0439667
1963q3 0.1613555 0.0420667
1963q4 0.1656651 0.0435333
1964q1 0.1724272 0.0489667
1964q2 0.1937889 0.05
1964q3 0.1892256 0.0507
1964q4 0.2288966 0.0679333
1965q1 0.2161062 0.0752
1965q2 0.220251  0.0677
1965q3 0.2444199 0.0643667
1965q4 0.2373356 0.0625333
1966q1 0.2373999 0.0624333
1966q2 0.2621366 0.0633667
1966q3 0.2630992 0.0748
1966q4 0.2896196 0.0730333
1967q1 0.289349  0.0636333
1967q2 0.2909229 0.0564667
1967q3 0.2679329 0.0558333
1967q4 0.2725836 0.0727
1968q1 0.324427  0.0801
1968q2 0.3127718 0.0824667
1968q3 0.3446502 0.0768
1968q4 0.3526328 0.0744
1969q1 0.3716218 0.0849333
1969q2 0.4083654 0.0925
1969q3 0.4206297 0.0960667
1969q4 0.4136589 0.0897
1970q1 0.4171974 0.0910667
1970q2 0.4292845 0.0804667
1970q3 0.4460275 0.0751
1970q4 0.4606331 0.072
1971q1 0.3978946 0.0751
1971q2 0.4389922 0.0652
1971q3 0.4402502 0.0580333
1971q4 0.4482596 0.0479
1972q1 0.418719  0.0496
1972q2 0.4048007 0.0568667
1972q3 0.3948362 0.0775333
1972q4 0.4388275 0.0808667
1973q1 0.5047194 0.1008667
1973q2 0.4497358 0.0892
1973q3 0.5118994 0.1273
1973q4 0.5517743 0.1475
```

1974q1	0.5914837	0.1556333
1974q2	0.6421138	0.1331667
1974q3	0.6729907	0.1277333
1974q4	0.649741	0.1260333
1975q1	0.6796549	0.1137667
1975q2	0.6843259	0.0977
1975q3	0.6849031	0.1060333
1975q4	0.7137509	0.1147
1976q1	0.7134444	0.0930333
1976q2	0.7313867	0.1093667
1976q3	0.7527727	0.1172
1976q4	0.8034339	0.1513
1977q1	0.7934583	0.1141667
1977q2	0.7898481	0.079
1977q3	0.7422794	0.0674
1977q4	0.7187624	0.0614667
1978q1	0.6988144	0.0659667
1978q2	0.7048116	0.0909667
1978q3	0.696776	0.0967
1978q4	0.6946978	0.1182667
1979q1	0.6852785	0.1292
1979q2	0.7555839	0.1269
1979q3	0.7629781	0.142
1979q4	0.801616	0.1624333
1980q1	0.8380404	0.1821333
1980q2	0.8323959	0.1725333
1980q3	0.8596979	0.1588
1980q4	0.8461346	0.1536
1981q1	0.8291904	0.1310667
1981q2	0.8149488	0.1247333
1981q3	0.8422555	0.1482333
1981q4	0.8534618	0.1572333
1982q1	0.8503275	0.1404333
1982q2	0.8609113	0.1339
1982q3	0.8403321	0.1117
1982q4	0.8200183	0.1004667
1983q1	0.8196285	0.1127333
1983q2	0.8045014	0.1002333
1983q3	0.8223725	0.0983667
1983q4	0.8110266	0.0929
1984q1	0.7757497	0.0921
1984q2	0.7591695	0.0937667
1984q3	0.749437	0.1104771
1984q4	0.7648844	0.0960813
1985q1	0.7415238	0.1142263
1985q2	0.7184946	0.084368
1985q3	0.6721155	0.0540316
1985q4	0.6455615	0.0421961
1986q1	0.6032669	0.0361592
1986q2	0.5597042	0.026584
1986q3	0.520312	0.0304446
1986q4	0.517358	0.0388338
1987q1	0.4653731	0.0291873
1987q2	0.4381253	0.0277219
1987q3	0.4394221	0.0368852
1987q4	0.418032	0.0265008

```
1988q1 0.3813962 0.0298168
1988q2 0.3671117 0.0291997
1988q3 0.3813599 0.0493664
1988q4 0.3873666 0.0507167
1989q1 0.3660495 0.0463667
1989q2 0.3465607 0.0526
1989q3 Nan       0.0574667
1989q4 Nan       0.0632
```

```
                    *
                *     *
```

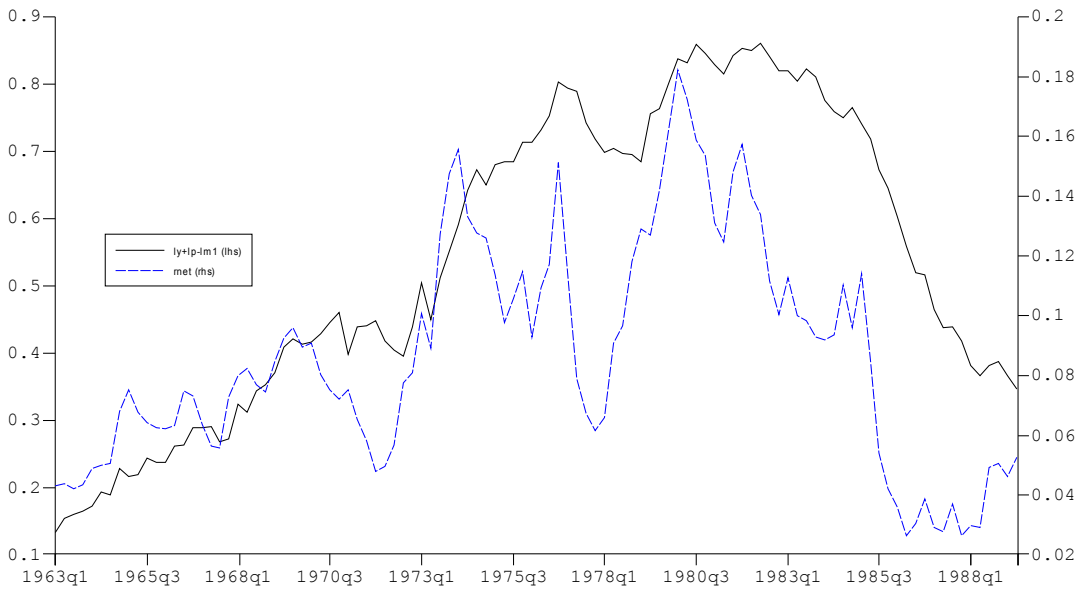
// Print log of m1 velocity and the interest rate. See chapter 3.

```
-->lagm1=lagts(exp(lm1))
```

// creates the lag of M1 (see chapter 3 for all manipulations on time series).

```
-->pltseries('ly+lp-lm1','rnet','yaxis=[1; 2]')
```

// Plots log of m1 velocity and the interest rate on different scales (see chapter 21).



Grocer 1.2

```
-->kpss('lm1',1)

KPSS stationarity test for variable: lm1
with a time term
and estimation period: 1964q3-1989q2
is equal to:  0.1850943
this values should be compared to the following critical values:
1% level 5% level 10% level
 .216     .146     .119

conclusion: the null hypothesis of stationarity is accepted at a 1% level, but
rejected at a 5% level

                         *
                      *     *
```

// Computes Kwiatkowski, Phillips, Schmidt, Shin stationarity test (see chapter 8, a number of other unit root tests exist also).

```
-->[r1]=automatic('delts(lm1-lp)','lagts(lm1-lp-ly)',...
-->'delts(lp)','rnet',...
-->'delts(lagts(lm1-lp))','delts(lagts(2,lm1-lp))','delts(lagts(3,lm1-
lp))','delts(lagts(4,lm1-lp))',...
-->'delts(ly)','delts(lagts(1,ly))','delts(lagts(2,ly))','delts(lagts(3,ly))',
'delts(lagts(4,ly))',...
-->'delts(delts(lp))','delts(delts(lagts(1,lp)))','delts(delts(lagts(2,lp)))',
'delts(delts(lagts(3,lp)))','delts(delts(lagts(4,lp)))',...
-->'delts(rnet)','delts(lagts(1,rnet))','delts(lagts(2,rnet))','delts(lagts(3,
rnet))','delts(lagts(4,rnet))',...
-->'const','prt=final,test_final');
 ____________________________________________________
 ____________________________________________________
| results of the automatic regression package      |
 ____________________________________________________

final model

ending reason: only one stage 1 model not rejected against their union


ols estimation results for dependent variable: delts(lm1-lp)
estimation period: 1964q3-1989q2
number of observations: 100
number of variables: 8
R² = 0.7834693      adjusted R² =0.7669942
Overall F test: F(7,92) = 47.554581       p-value = 0
standard error of the regression: 0.0127155
sum of squared residuals: 0.0148750
DW(0) =2.023762
Belsley, Kuh, Welsch Condition index: 12

variable                  coeff      t-statistic p value
lagts(lm1-lp-ly)          -0.1153950 -10.355618  0
delts(lp)                 -0.8001655 -6.3224265  9.122D-09
rnet                      -0.8088969 -9.9557857  4.441D-16
```

```
delts(lagts(lm1-lp))      -0.2661552 -3.3172529  0.0013029
delts(lagts(3,lm1-lp))    -0.1658617 -2.5066203  0.0139447
delts(delts(lagts(2,lp))) 0.3051644  1.9005401   0.0604933
delts(rnet)               0.3633122  2.7771577   0.0066452
const                     0.0316593  6.9538925   5.051D-10


                        *
                    *     *


tests results:
**************
test                   test value p-value
Chow pred. fail. (50%) 0.4506833  0.9963452
Chow pred. fail. (90%) 0.5741256  0.8305430
Doornik & Hansen       6.8029378  0.0333243
AR(1-4)                0.3262070  0.8596993
hetero x_squared       1.6339596  0.1189528


                        *
                    *     *



variable                  reliability

lagts(lm1-lp-ly)          1
delts(lp)                 1
rnet                      1
delts(lagts(lm1-lp))      0.7
delts(lagts(3,lm1-lp))    0.7
delts(delts(lagts(2,lp))) 0
delts(rnet)               0.7
const                     1

                        *
                    *     *
```

// from a large set of potentially relevant variables, extract the "best model", that is the one that has
// only significant variables, that passes 5 default specification tests and that has the better fit among
// the ones that verify the two previous conditions. See chapter 14.

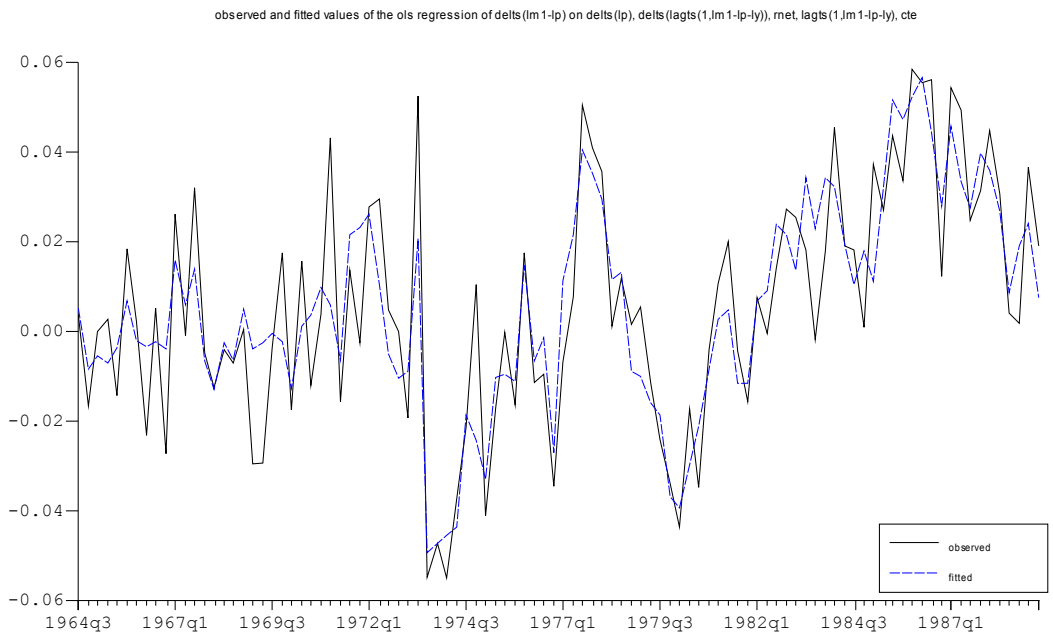
```
-->bounds('1964q3','1989q2');            rols=ols('delts(lm1-lp)','delts(lp)',...
'delts(lagts(1,lm1-lp-ly))','rnet','lagts(1,lm1-lp-ly)','cte');

-->rols=ols('delts(lm1-lp)','delts(lp)','delts(lagts(1,lm1-lp-
ly))','rnet','lagts(1,lm1-lp-ly)','cte')

ols estimation results for dependent variable: delts(lm1-lp)
estimation period: 1964q3-1989q2
number of observations: 100
number of variables: 5
R² = 0.7616191      adjusted R² =0.7515820
```

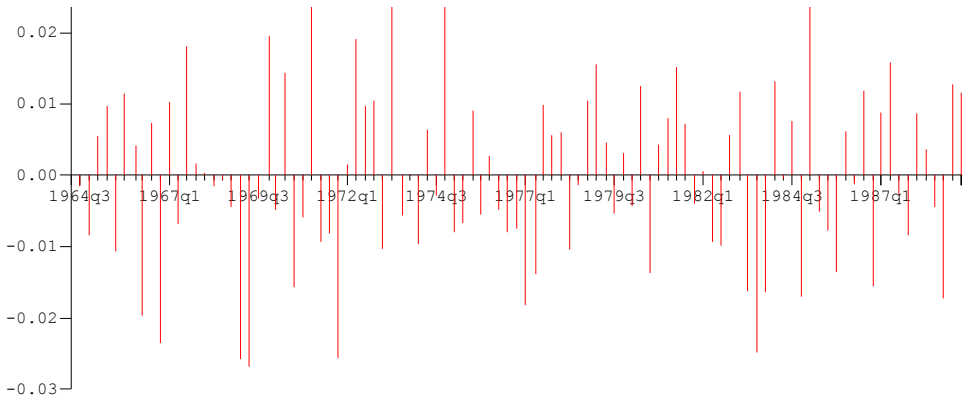
```
Overall F test: F(4,95) = 75.880479       p-value = 0
standard error of the regression: 0.0131293
sum of squared residuals: 0.0163760
DW(0) =2.1774387
Belsley, Kuh, Welsch Condition index: 9

variable                  coeff      t-statistic p value
delts(lp)                 -0.6870384 -5.4783521  0.0000004
delts(lagts(1,lm1-lp-ly)) -0.1746079 -3.0101518  0.0033442
rnet                      -0.6296267 -10.46407   0
lagts(1,lm1-lp-ly)        -0.0928557 -10.873421  0
cte                       0.0234367  5.8185613   7.986D-08
```

*

* *

// estimates equation # (6) in D.F.Hendry et N.R.Eriesson (1991) on period 1994q3-1989q2





// Results are saved in tlist rols and can be used later: reprinted or used in other grocer
// functions such as the testing functions. 2 graphs are also automatically produced, one in graphic
// window n°1 with observed and fitted values and another one, in graphic window n°2, with the
// residuals. See chapter 6.

```
-->hetero_sq(rols);
F(6,88)=1.7883885
(p -value                  = 0.1104757)
```

// heteroskedasticity test from the regression of the square of ols residuals on the exogenous
// variables and their square. This is one of the numerous tests presented in chapter 7.

```
-->rjoh=johansen(1,2,'lm1','ly','lp','rnet')

Johansen estimation results for variables:
lm1, ly, lp, rnet
time order: 1
# of lags: 2

NULL:    Trace Statistic Crit 90% Crit 95%  Crit 99%
r <= 0   56.151045       51.6492  55.2459   62.5202
r <= 1   21.664995       32.0645  35.0116   41.0815
r <= 2   9.1360287       16.1619  18.3985   23.1485
r <= 3   0.2887943       2.7055   3.8415    6.6349

conclusions from the trace statistics:
at a 10% level, there are 1 cointegration relation(s)
at a 5% level, there are 1 cointegration relation(s)
at a 1% level, there are 0 cointegration relation(s)

NULL:    Max Eigenvalues Statistic  Crit 90%  Crit 95%  Crit 99%
l <= 0   34.486051                  28.2398   30.8151   36.193
l <= 1   12.528966                  21.8731   24.2522   29.2631
l <= 2   8.8472344                  15.0006   17.1481   21.7465
l <= 3   0.2887943                  2.7055    3.8415    6.6349

conclusions from the maximal eigenvalues statistics:
at a 10% level, there are 1 cointegration relation(s)
at a 5% level, there are 1 cointegration relation(s)
at a 1% level, there are 0 cointegration relation(s)

                         *
                      *     *
```

// Perform Johansen cointegration test on the system (lm1,ly,lp,rnet) and save the results in tlist rjoh.
// See chapter 8.

```
-->recm_he=ecm(2,'endo=lm1;ly;lp;rnet','jres=rjoh','nbr=1');

*** cointegrating vectors from johansen estimation ***

variable vector # 1
lm1      -3.7891204
```

```
ly       54.594419
lp       4.4111627
rnet     27.829891


                         *
                      *     *


ecm estimation results for variables
del(lm1), del(ly), del(lp) and del(rnet)

AIC criterion: -35.064497
BIC criterion: -32.159349
Hannan-Quinn criterion: -34.640258

estimation results for dependent variable del(lm1)
estimation period: 1964q3-1989q2
number of observations: 99
number of variables: 10
R² = 0.4503667      adjusted R² =0.3947858
Overall F test: F(9,89) = 8.1029042       p-value = 1.098D-08
standard error of the regression: 0.0162651
sum of squared residuals: 0.0002378
DW(0) =1.9852811
Belsley, Kuh, Welsch Condition index: 155

variable                coeff      t de student p value
del(lm1)(-1)            -0.0833737 -0.8103282   0.4177516
del(ly)(-1)             0.2059791  1.6244647    0.1042767
del(lp)(-1)             0.0389578  0.1753093    0.8608366
del(rnet)(-1)           -0.5232754 -3.7519321   0.0001755
del(lm1)(-2)            0.2579669  2.6180296    0.0088439
del(ly)(-2)             0.1375873  1.0527438    0.2924585
del(lp)(-2)             0.1040476  0.4637328    0.6428392
del(rnet)(-2)           -0.3231208 -2.3884206   0.0169210
lag 1 of coint. vec. #1 0.0007140  3.3979415    0.0006789
cte                     -0.4019059 -3.3349953   0.0008530

                         *
                      *     *

estimation results for dependent variable del(ly)
estimation period: 1964q3-1989q2
number of observations: 99
number of variables: 10
R² = 0.2219690      adjusted R² =0.1432917
Overall F test: F(9,89) = 2.8212587       p-value = 0.0057847
standard error of the regression: 0.0131471
sum of squared residuals: 0.0001554
DW(0) =2.0709672
Belsley, Kuh, Welsch Condition index: 155

variable                coeff      t de student p value
del(lm1)(-1)            0.1552515  1.8667838    0.0619318
```

```
del(ly)(-1)              -0.2319213 -2.2628413   0.0236455
del(lp)(-1)              -0.2286450 -1.2729123   0.2030492
del(rnet)(-1)            0.0179749  0.1594478    0.8733161
del(lm1)(-2)             0.2284578  2.8684215    0.0041253
del(ly)(-2)              -0.0536103 -0.5074797   0.6118183
del(lp)(-2)              -0.0870498 -0.4799874   0.6312364
del(rnet)(-2)            0.1925850  1.7611443    0.0782140
lag 1 of coint. vec. #1 -0.0003170 -1.8661079   0.0620263
cte                      0.1914192  1.965091     0.0494037

                          *
                       *     *

estimation results for dependent variable del(lp)
estimation period: 1964q3-1989q2
number of observations: 99
number of variables: 10
R² = 0.7098118      adjusted R² =0.6804669
Overall F test: F(9,89) = 24.188608       p-value = 0
standard error of the regression: 0.0080471
sum of squared residuals: 0.0000582
DW(0) =2.0626045
Belsley, Kuh, Welsch Condition index: 155

variable                 coeff      t de student p value
del(lm1)(-1)             0.0404454  0.7945449    0.4268783
del(ly)(-1)              0.0305806  0.4874732    0.625923
del(lp)(-1)              0.6313144  5.7421347    9.349D-09
del(rnet)(-1)            0.0608335  0.8816262    0.3779790
del(lm1)(-2)             0.0475717  0.9758346    0.3291465
del(ly)(-2)              -0.0636447 -0.9842920   0.3249719
del(lp)(-2)              0.2112037  1.9026316    0.0570886
del(rnet)(-2)            0.1291106  1.9289686    0.0537348
lag 1 of coint. vec. #1 -0.0000556 -0.5350546   0.5926121
cte                      0.0338979  0.5685401    0.5696683

                          *
                       *     *

estimation results for dependent variable del(rnet)
estimation period: 1964q3-1989q2
number of observations: 99
number of variables: 10
R² = 0.1297543      adjusted R² =0.0417519
Overall F test: F(9,89) = 1.4744404       p-value = 0.1697386
standard error of the regression: 0.0137479
sum of squared residuals: 0.0001699
DW(0) =1.9877036
Belsley, Kuh, Welsch Condition index: 155

variable                 coeff      t de student p value
del(lm1)(-1)             0.1792167  2.0607754    0.0393245
del(ly)(-1)              0.1412890  1.3183051    0.1874015
del(lp)(-1)              0.0826369  0.4399514    0.6599723
del(rnet)(-1)            0.1740334  1.4763107    0.1398605
del(lm1)(-2)             0.0847650  1.0177656    0.3087894
```

```
del(ly)(-2)              -0.0155285 -0.1405702   0.8882095
del(lp)(-2)              -0.1167511 -0.6156262   0.5381412
del(rnet)(-2)            0.1217966  1.0651274    0.2868183
lag 1 of coint. vec. #1 -0.0003590 -2.0211445   0.0432648
cte                      0.2033469  1.9963137    0.0458998

                          *
                       *     *
```

// estimate a vecm model for the system (lm1,ly,lp,rnet). Number of lags has been fixed to 2 ;
// the estimation uses the johansen tlist result estimated previously and imposes to 1 the
// number of cointegration relations to take into account. See chapter 9.

```
-->varf(recm_he,'1992q4');

forecasting results for the VAR over the period 1989q3-1992q4
*************************************************************

obs\variable lm1       ly        lp        rnet
1989q3       11.680641 11.836153 0.1782128 0.0506976
1989q4       11.720257 11.844305 0.1979197 0.0486624
1990q1       11.764288 11.850625 0.2170368 0.0455549
1990q2       11.808136 11.857857 0.2364220 0.0423646
1990q3       11.853573 11.865666 0.2560431 0.0395380
1990q4       11.899136 11.873328 0.2758818 0.0369955
1991q1       11.945102 11.881297 0.2959967 0.0345704
1991q2       11.991191 11.889179 0.3164012 0.0322251
1991q3       12.037558 11.897021 0.3370531 0.0298381
1991q4       12.084148 11.904793 0.3579362 0.0273968
1992q1       12.131042 11.912513 0.3790164 0.0248878
1992q2       12.178222 11.920184 0.4002695 0.0223208
1992q3       12.225707 11.92783  0.4216745 0.0197039
1992q4       12.273482 11.935453 0.4432152 0.0170464

                          *
                       *     *
```

// use the estimated model to make a forecast until 1992q4 (see chapter 9).

6. Future developments

There are very numerous extensions that can be programmed in GROCER. Here are the main extensions we plan to make in the near future:

- (at a one year horizon): structural VARs and decomposition of the variance; ordered logit estimates; Generalized Method of Moments (GMM); static and dynamic factor analysis;
- (at a two years horizon): trend-cycle unobserved component models; E-GARCH, GARCH-M and other garch methods that are currently lacking in GROCER;
- (at a three years horizon): a syntactic parser, in order to improve nls speed on ts and to provide user-friendly dynamic simulation; more unit roots and cointegration tests.

If you have specific needs, send us an e-mail and we will do our best to satisfy them. Try to be the most precise in the description of your needs (a reference to the literature would notably help).

7. Bugs and contributions

If you discover a bug in GROCER, do not hesitate to inform us at the following e-mail address: grocer.toolbox(at)free.fr[6]. In that case, we will try to correct it in the shortest time. Do not forget to indicate the Scilab and GROCER versions[7]. Much better, you can correct it: in that case, e-mail us also the nature of the bug and the correction you propose. In the text of the concerned function, indicate the lines of programs that you have corrected along with your name as follows:

```
// correction provided by M. X
....
corrected text
...
// end of correction
```

You can also correct the Copyright to add your own name.

If you develop your own programs and you think they can interest other users, do not hesitate to send them to me, so that wecan incorporate them in GROCER, with due acknowledgement to its author! You can also send your program directly to the Scilab group as a Scilab contribution. In any case, if your program uses GROCER functions or is complementary to GROCER, then please check that they conform to GROCER principles and that they do not give rise to side-effects with Scilab or GROCER own programs.

8. Citing GROCER.

You can use GROCER as you want, provided that you respect the constraints described in the licenses that apply to Scilab and GROCER. We would however greatly appreciate that you make a proper citation of GROCER in any publication built on work made with GROCER. In that case, you should make the following citation:

Éric Dubois and Emmanuel Michaux (2006): "GROCER 1.2: an econometric toolbox for Scilab", available at *http://dubois.ensae.net/grocer.html*.

[6] To limit spam, the required symbol @ is replaced by (at): do not forget to do the reverse substitution.
[7] Each time we release a new version (because of the addition of a new program or a of the correction of a bug), we upgrade the number of the version.