Interactive management for time series

Calzolari, Giorgio

Centro Scientifico IBM, Pisa, Italy

1974

Interactive Management for Time Series

Giorgio Calzolari, Pisa

1. INTRODUCTION

A very important element in order to study the structure of a national economy is the organization of a data file consisting of the observations, over time, on the variables which characterize the economy.

The main purpose of the package presented in this paper is to offer researcher a supply of information, which can be increased revised and updated during the research.

While the application of some form of computer processing can be great help whenever data must be manipulated in a repetitive way, an interactive approach, by allowing a direct dialogue between the user and the computer and, making it possible for a large number of users to have simultaneous access to the same information, facilitates this iterative procedure very much.

Data are handled globally by a package, called IMTS (Interactive Management for Time Series), running under the operating system CP-67/CMS. By means of this package, it is possible to use data both from the common central file and from a private single user file; these data are accessed using only the abbreviated name of the series to be used. The user has therefore to know only the abbreviated names of the series, whose complete list may be obtained by issuing a simple command at the terminal.

The research capabilities supplied by the IMTS package are analysis, transformation and regression of stored data. As regards analysis and transformation of stored data, algebraic, not algebraic, logical operators can be applied as described in sections 4.1 to 4.6.

The estimation of the parameters of a model can be performed by means of three different methods: Ordinary Least Squares, Two Stage Least Squares and Limited Information Single Equation. They are described in section 4.7.

These features can be used independently or in any combination by means of the IMTS language, which provides a single command for each application: it is therefore possible to write regression equations as well as algebraic expressions defining transformations of the economic variables.

For the standard commands no programming skill is required; the researcher can devote virtually all his attention to problem analysis.

Moreover, any user with some experience of Fortran and/or Assembler languages, can create special functions (additional estimation methods, particular cross-sections studies, etc) by means of subroutines which, after having been compiled, can be used immediately in the IMTS environment, in conjunction with any other standard or user's operator.

The research capabilities of IMTS are supported by many other , such as the on-line formal errors recovery procedure (see section 4.9), the Macro facility (see section 3), the on-line plotter (see section 4.7), the linkage with the simulation package (DMS command, see section 4.7 and ref.), etc.

The IMTS system was originally designed for use in econometrics, but it is expected to be applicable to many other fields such as management science or social sciences.

2. CONNECTIONS BETWEEN THE PACKAGE AND THE OPERATING SYSTEM

As mentioned in the introduction, the IMTS package operates under the control of the operating system CP-67/CMS. This is general purpose time-sharing system for the IBM System/360: the CP (Control Program) creates the time-sharing part of the system to allow many users to access the computer simultaneously. The CMS (Cambridge Monitor System) provides the conversational part of the system to allow a user to monitor his work from a remote terminal.

CMS interprets a simple command language typed at the user's remote terminal. One of its purposes is to provide the user with various file-handling facilities, referring to data stored on disk, cards, or magnetic tape.

The description of the technical features of this operating system is beyond the scope of this paper; what we want to emphasize here is that, taking into account the interactive features of CP-67/CMS and suitably operating on the main storage of the computer, the package and the operating system are linked very closely together to perform any operation in a very effective way.

The IMTS environment must be entered by issuing a command in the CMS environment; in the turn, any CMS command may be executed while operating in the IMTS environment. In particular, all the commands dealing with "File Creation, Maintenance and Manipulation" of CMS, and with utility programs of IMTS may be issued without leaving the IMTS environment.

3. FILE MANAGEMENT

The input/output routines of the package provide for the maintenance of four classes of file.

The first is made up of one single file: the file of permanent time series available to all users.

The second is also made up of only one file, containing the abbreviated names of all the permanent time series. It is the directory.

The third class of files is made up of all the files containing the private time series of individual users, or the time series obtained as intermediate

results of one procedure, for use in subsequent ones.

The fourth class includes only one file in which the commands to be executed are stored as and when they arise. It is used both for correcting formal errors in commands, and for carrying out burdensome sequences of commands, which may be stored once and for all and then recalled as and when needed (macro facilities).

Each series, in the central or in a private file, holds the following :

- the complete name of the time series;
- the abbreviated name of the series;
- the data source;
- the unit of measurement;
- a code to indicate whether the data are annual, semi annual, quarterly or monthly;
- the first year;
- the last year;
- the first quarter (or half-year or month);
- the last quarter (or half-year or month);
- the number of data in the series;
- the numerical data.

The data must be complete from beginning to end (missing data are not allowed).

4. ACCESS TO THE FILE AND DATA PROCESSING

Issuing of commands and printing of results are carried out by the user through the terminal. The user must issue at the terminal the command to be executed, eventually followed by the names of the time series (from the central file or private series: in the event of the same names, the private series is the first to be searched for), numerical constants, operator codes and function names. Generally speaking a command must be contained in one line (nevertheless, the continuation on one or more subsequent lines is allowed); the program decodes and executes each command-line as soon as it is written.

After executing a command, the program prints the results on the terminal, but does not store these results. Should the user wish to retain these results (and this is only possible when the results constitute a time series or when a single numerical result is involved, thus excluding plotter output, for example), it is necessary to write the key word FILE at the end of the data string, following it up immediately - without any intermediate space - with the name of the file to be created: for example FILEGNP, the results of the command go into the FILE GNP P1, (CMS file identifier) and may be used in all subsequent operations merely by referring to the GNP time series.

The following operations may be carried out on time series and on any constant :

(i) Operations and functions executed by routines held in the storage

.AND., .OR., .LT., .LE., .EQ., .NE., .GE., .GT.

that are logical dyadic operators.

.NOT.

that is a logical monadic operator.

+	addition
-	subtraction
*	multiplication
/	division
**	raise to power

that are algebraic dyadic operators.

LOG	natural logarithm
LOG10	decimal logarithm
EXP	exponential (base e)
SIN	sine (Radians)
COS	cosine (Radians)
ATAN	arc tangent (Radians)
SINH	hyperbolic sine
COSH	hyperbolic cosine
ABS	absolute value
MAX	maximum value
MIN	minimum value
MEAN	mean value
VAR	variance
INT	integer part
RV	rate of variation
DIF	first difference
LAG(nn)	data lagging
ONE(nn/mm)	selection of one single item of data from a series
SEL(nn/mm,ll/kk)	selection of part of a series
CPR(nn)	compression of a series

that are monadic operators.

IF(THEN,
ELSE)

that are for discontinuities and decision rules.

(ii) Operations and functions carried out by special routines.

PLOT	in line plotter
OLS, OLSR	ordinary least squares estimate
TSLS	two-stage least squares estimate
LISE	limited information single equation estimate

DMS connection to DMS/2

user defined functions

4.1 Operator priority

Highest priority is given to the following functions: LOG, LOG10, EXP, SIN, COS, SINH, COSH, ATAN, ABS, MAX, MIN, MEAN, VAR, INT, RV, ONE, SEL, CPR, IF, OLSR. In the event of functions being chained, priority goes from right to left (the internal function being executed first).

These are followed, in order of precedence, by: raising to power, multiplication and division, in order from left to right, then addition and subtraction, in order from left to right. Then comparison operations in order from left to right. Then the logical operator .NOT. At the end, the logical operators .AND. and .OR. in order from left to right. In order to vary priority, parenthesis may be used.

4.2 Arithmetic dyadic operators

+, -, *, /, **

The operator must be written between the two operands to which it refers. No special separator signs are necessary.

The execution of these operations is subjected to the following rules. Between two constants, the result is a new constant. Between a constant and a series, the result is a new series with the same characteristics (starting date and finishing date, division into periods) as the original series. The operation between two series is allowed only if both series are divided into the same periods (for example, both with monthly data or both with annual data), in which case the result is a series limited to the period common to both the original series (for example, the sum of two series of annual data - one from 1953 to 1972 and the other from 1951 to 1970 - is a series of annual data from 1953 to 1970).

4.3 Logical dyadic operators

.AND., .OR., .LT., .LE., .EQ., .NE., .GE., .GT.

As for the arithmetic dyadic operators, the result may be a constant or a series, whose numerical values may be 0. (FALSE) or 1. (TRUE).

For the .AND. and .OR. operators, the operation can have any value, but they are tested only to control if the value is 0. (FALSE) or $\neq$ 0. (TRUE).

4.4 Monadic operators

(i) EXP, LOG, LOG10, SIN, COS, ATAN, SINH, COSH, ABS, INT, .NOT.

The operator symbol must precede the variable (or constant) to which it is to be applied. It is possible to chain several operators, without using separator signs between them.

When applied to a constant, these functions transform it into a new constant. Applied to a series, they transform it into a new series with the same characteristics as the original one (starting date, finishing date, division into periods).

(ii) MAX, MIN, MEAN, VAR

They return only one numerical value, which has all the characteristics of a costant (there are no dates, etc.). However, if they are related to the operator CPR (compression of a series in conformity with another operator), they return a series (see CPR).

(iii) ONE(nn/mm)

Extracts from the series the numerical value corresponding to the year and the month (or quarter or half-year) indicated. For example, ONE(67/11)'SERIES1' returns the value for the month of November and for the year 1967 (SERIES1 must have monthly data); ONE(67)'SERIES1' returns the first (or only) item of data for the year 1967. The numerical value returned has all the characteristics of a constant.

(iv) LAG(nn)

This operator shifts the data of a time series (in one direction only) for as many positions as are indicated by nn (from 0 to 99). For example, LAG(4) 'SERIES1', where SERIES1 (annual data) starts from 1961 and ends in 1969, returns a series starting from 1965 and ending in 1969, with the original value for '61 assigned to '65 etc.

(v) SEL(nn/mm, ll/kk)

Selects from the series the values included between the year and the month (or quarter) indicated first, and that indicated second. For example, SEL(55/10, 66/2)'SERIES1' returns a series with monthly data (like SERIES1) included between October 1955 and February 1966.

(vi) RV

Calculates the variation rates of a series, taking into account the difference between one data and the preceeding, whatever the periodicity. The results are in percentage form.

4.5 Composite operators

CPR(nn)

The function CPR(nn) (compression of a series according to another operator) must be used together with one of the operators MAX, MIN, MEAN, VAR, +, *, : it must never be used alone. This function gives a series with characteristics as illustrated in the following examples: MAX CPR(1) 'SERIES1' (where SERIES1 has monthly data) gives a series of annual values (1 per year) equal to the maximum value among the monthly values for

each year; MEAN CPR(4)'SERIES1' gives a series of quarterly values (4 per year) equal to the mean from among the values for each quarter in the series; 0.+CPR(2)'SERIES1' gives a series of half-yearly values (2 per year) equal to the sum of all the values for each half year in the series indicated. In these cases, the operators MAX, MIN, MEAN, VAR give a series instead of a constant.

4.6 The IF operator

Discontinuities and decision rules can be analyzed by means of the IF operator. It is used together with the pseudo-operators THEN and ELSE as illustrated in the following examples :

(i) IF ('SERIES1'.LT.'SERIES2')THEN'SERIES3'ELSE'SERIES4'

First of all the program performs the operation with highest priority (.LT. because it is enclosed in parenthesis); the result is a series, limited to the period common to SERIES1 and SERIES 2 and with the same division into periods, with numerical values 0. and 1.
This resulting series should have the same division into periods as SERIES3 and SERIES4 in order to perform the operation for the period common to three series, selecting for every year the value from SERIES3 if the corresponding value of the logical calculated series is nonzero (TRUE) or from SERIES4 if it is 0. (FALSE). These values build the resulting series.

(ii) IF('SERIES1'.LT.1000.)THEN'SERIES2'ELSE IF(178.LT.'SERIES3') THEN'SERIES4'ELSE 250.

In this example, comparisons are made between series and constants (every value of the series with the constant) and the resulting values are selected from series or from the constant. Each ELSE refers to the last unclosed THAN; each THEN must be closed by an ELSE.

4.7 Special functions

This group includes the on-line plotter routines, estimate routines (OLS, OLSR, TSLS, LISE), the routine for connection to DMS/2, and user-defined functions.

(i) OLS (ordinary least squares)

Multiple linear regression by the ordinary least squares method is executed by means of the SSP (Scientific Subroutine Package) library routines (see ref.). On the basis of the ordinary least squares method, is possible with these routines to calculate the standard deviations and means of both dependent and independent variables, simple and multiple correlation coefficients, and also to calculate regression coefficients. Variance analysis is also executed and, on request, the residue table (differences between the observed and interpolated values is printed.

On user's request, it is possible to specify the output unit (6=teletype terminal, 8= high-speed printer) when printed output of the results is required.
Example: OLS('SERIES1', RV('SERIES2'+'SERIES3'), 'SERIES4'); SERIES1 is the dependent variable (written first in the list of variables separated by commas), while the other two, that is the series resulting from the RV('SERIES2'+'SERIES3') operation and 'SERIES4' are the independent variables.

(ii) OLSR

It performs the same operations as OLS, but instead of printing a table of statistics, it gives, as a result, the series of the residuals of the linear regression. It can be so used as each other function. For example: 'SERIES1'-OLSR('SERIES1', 'SERIES2'); the result is the series of the interpolated values of SERIES1.
PLOT('SERIES2', OLSR('SERIES1', 'SERIES2'))(see. PLOT) plots at the terminal the residuals against the explaining variable SERIES2.
Among the various single-equation methods for estimating systems of simultaneous equations, TSLS and LISE are considered hereunder.

(iii) TSLS (two-stage least squares)

The TSLS instruction ('SERIES1', 'SERIES2', ...) sorts all the endogenous and predetermined variables (lagged endogenous and exogenous) involved in the system to be estimated: all the endogenous variables are defined first, then the predetermined variables. After preliminary control of the total number of variables sorted, by means of specification of the number of endogenous and predetermined variables, the program executes the OLS estimate of the 'reduced form' (first stage). Then (the second stage), for each structural equation (at most as many as the endogenous variables), the user must insert the specifications (number of variables and sorting index) needed for sorting the subset of variables involved in the equation. For each equation, the estimate of the coefficients and the estimate of the standard error referred to each coefficient are supplied.

(iv) LISE (Limited information single equation)

The operational characteristics of the LISE command are the same specified for TSLS: however, even if the user is not aware of this, it should be noted that the 'reduced form' estimate applies only to the 'standard error' estimate, while for estimating the coefficients of structural equations, in the iterative process which leads to the maximization of the variance ratio, the OLS estimate of the same equation is taken as the point of departure.

(v) PLOT

The PLOT command permits the printing on the terminal of a graph

in points for certain (dependent) variables versus another (independent) variable. The scale is automatically adjusted on the basis of the minimum and maximum values of the variables.
Example: PLOT('SERIES1','SERIES2',RV'SERIES3'*MAX'SERIES3'/100.);'SERIES1' is the independent variable (abscissa), while 'SERIES2' and the series resulting from the RV'SERIES3'*MAX'SERIES3'/100 operation are the dependent variables (ordinates).
In conversational mode the user is asked to choose between two different types of on-line plotter that is the normal plotter, with numerical values and scales which takes considerable time to print, or the high-speed, qualitative plotter, without numerical values and scales which expecially in the case of a single dependent variable, permits a considerable saving in time.

(vi) DMS

The DMS command ('SERIES1','SERIES2'....) allows the selection of a certain number of time series, writing their names and numerical values on the FILE DMS P1, according to the format required by the DMS/2 input; the data are, however, arranged variable by variable, while DMS/2 requires them arranged year by year. It is therefore necessary to use the CMS SORT command which permits re-arrangement of the file in date order. In this case, series of different lenghts are not cut as this operation is handled by DMS/2.

4.8 User defined functions

The user himself can create special functions of interest for him, according to the general philosophy of the IMTS. He must write a program, whose FILENAME must be the name he wants to give the function, as a subroutine with a standard list of parameters. All the necessary operative information can be easily obtained by entering a standard subroutine, which is available for any user. After a correct compilation, this function is immediately available with no other formalities. User defined functions are treated as the other functions for the priority.

4.9 Formal error recovery procedure

During the operations for the coding and decoding of commands some types of formal errors are likely to occur in the program as, for example, number of open parenthesis different from number of closed parenthesis quotes opened but not closed to contain names of series, names of series exceeding 8 characters, and so on.

After reporting the type of error and the name of the routine containing the error (indication normally not essential for the user, but very useful for the system engineer in setting up new functions), the program calls the EDIT program of the CMS and applies it to the file in which each command is stored before decoding.

Using the usual EDIT requests the user corrects the given command, cancels from the file any other records not required, gives the FILE command, which retains the file content and returns control to the main program, then, if required, gives the GO command to repeat operation of the corrected command. If this still contains formal errors, the procedure starts again.

Other types of errors are also contemplated as, for example, time series which are not present either in the central file or in the user's private file operations which do not yield any result (for example, arithmetic operations on two incompatible series, one with monthly data and one with annual data, or on two compatible series without a common period) divisions by zero, logarithms of negative numbers, and so on. In the case of all these errors, which are not in the formal error category, the correction procedure is not called automatically. Should he think it advisable, however, the user may edit the command file and then proceed as for formal errors.

5. UTILITY PROGRAMS

The category of utility programs includes all routines in the operating system accessible to the user (EDIT, COMBINE, LISTF, SORT, APL etc.) and certain special programs which are external to the main program but are accessible from within it (CREARCH, NEW). The CMS routines are accessed from within the program by means of the usual CMS command. As the special programs are held on the same disk as the central file in module format, the are called by typing only their name.

(i) CREARCH

This program was used at the outset to create the central file, and it is used whenever new time series are to be added to this file. It may be used only by the responsible for the file maintenance, and the file, according to the program philosophy, may be read contemporaneously by more than one virtual machine, but may be modified by only one.

(ii) NEW

This is a program which permits the creation of a file containing a new time series in user's private file, without any problem. It reads the data as and when they are written and then stores them in the required file, in correct format. Once it has been created and completed, the new series may be added by the central file becoming available for all the users.

REFERENCES

I. B. M. (CP-67/CMS) Version 3 CMS Program Logic Manual - GY20-0591-0

I. B. M. (CP-67/CMS) Version 3 User's Guide - GH20-0859-0

Corsi P., A. Stajano, An interactive programming system to solve econometric non linear model - CSP010/513-3505

I. B. M. System/360 Scientific Subroutine Package Version III - Programmer's Manual - H20-0203

Kmenta J., Elements of Econometrics - The Macmillan Company, New York

EXAMPLE (see Kmenta op. cit., p. 564)
(with formal error)

```
--->
ols('qt','pt','ft','time'
(D2)ERROR : PARENTHESIS OR THEN-ELSE DO NOT BALANCE
(AMDM1) ERROR RECOVERY PROCEDURE
USE EDIT COMMANDS.
DEFAULT TABS SET.
EDIT:
p
OLS('QT','PT','FT','TIME'
c./' /')/
n
EOF
file
--->
go
OLS('QT','PT','FT','TIME')
THERE ARE 20 DATA FOR EACH SERIES, FROM 1922 TO 1941:1 EACH YEAR
1 SIMBOLIC UNIT (6/8)
2-9 NAME
10 RESIDUALS (0/1)
-.........-
6supply  0
MULTIPLE REGRESSION.....SUPPLY
VARIABLE    CORRELATION  REGRESS.   STD. ERROR      COMPUTED
  NO.         X VS Y     COEFF.     OF REG. COEF.   T VALUE
    2         0.09810    0.16037      0.09488       1.69014
    3         0.68057    0.24813      0.04619       5.37226
    4         0.15004    0.24830      0.09752       2.54622
DEPENDENT
    1
INTERCEPT                  58.27541
MULTIPLE CORRELATION       0.80920
R**2                       0.65481
STD. ERROR OF ESTIMATE     2.40509
STD. DETERMINANT           0.88357
SOURCE       D.F.   SUM OF SQUARES   MEAN SQUARES   F VALUE
REGRESSION    3        175.5632         58.5211      10.11697
RESIDUAL     16         92.5512          5.7844
TOTAL        19        268.1144
              DURBIN WATSON STATISTIC = 2.1097
```