Research Report
No. 39 1991

MOSES ON PC

Manual, initialization, and calibration

by
Erol Taymaz

THE INDUSTRIAL INSTITUTE FOR
ECONOMIC AND SOCIAL RESEARCH

The Industrial Institute for Economic and Social Research

is an independent non-profit research institution, founded in 1939 by the Swedish Employers' Confederation and the Federation of Swedish Industries.

Objectives

To carry out research into economic and social conditions of importance for industrial development in Sweden.

Activities

The greater part of the Institute's work is devoted to long-term problems, especially to long-term changes in the structure of the Swedish economy particularly within manufacturing industry.

Address

Industriens Utredningsinstitut
Box 5501, S-114 85 Stockholm, Sweden
Tel. 08-783 80 00

The Industrial Institute for Economic
and Social Research

MOSES on PC

- manual, initialization, and calibration

by
Erol Taymaz

Distribution: Almqvist & Wiksell International, Stockholm

FOREWORD

The Swedish micro-to-macro simulation model, MOSES, has been developed over more than a decade. It has now been transferred from a mainframe computer into a personal computer.

This manual of Erol Taymaz documents the PC version of the MOSES model. It consists of three sections: *manual*, *initialization*, and *calibration*. Several improvements of the model specification have been entered. For instance the credit market module has been significantly revised. The calibration program is entirely new. It builds on a novel application of an old idea and has finally made possible systematic "estimation" of parameters in the complex model structure. Appendices contain the APL code written for the PC version.

The model and a synthetic database can now be made available for researchers, firms, and teachers.

Stockholm in May 1991
Gunnar Eliasson

CONTENTS

INTRODUCTION

MOSES^{PC} is the PC version of MOSES (Model Of the Swedish Economic System) which is a micro-to-macro, firm-based, econometric model of the Swedish economy. It has been transferred from the DEC-10 mainframe version of MOSES 7.3 in November 1989. The whole model is written in the programming language APL. The model with all current datasets requires less than 3 MByte memory. The current version of the model is implemented by using a Dyalog-APL interpreter with SCO Xenix Operating System V in a Toshiba T5200/100 portable PC (80386-20 microprocessor). A one-year simulation takes about 1.5-2.0 minutes depending on the experiment.

This manual describes how to *run, initialize, and calibrate* the model.(*) It is intended for users who have little or no familiarity with the model. Anyone following the instructions in Section 1.2 can start up and run the model without knowing what it is all about. At this level, it is also possible to make various experiments by changing the model parameters. However, to be able to carry out experiments based on changes in the model (e.g., changes in the behavioral equations, etc.), one needs a rather deep understanding of the model and the APL code which requires a considerable effort on the part of the user.

The model consists of two parts, the simulation model itself and the initialization procedure. There are micro and macro databases for two years, i.e., starting points for the simulation, namely, 1976 and 1982. The initialization procedures take as input micro and macro databases and converts them into a form that fits the simulation model. Thus, there are two basic initial datasets with various versions for 1976 and 1982. The simulation commences the first quarter of 1977 or 1983 depending on the choice of the initial year. Since the 1976 and 1982 datasets contain

*) Complete model code written in APL is published in *MOSES Code* (Albrecht *et al.* 1989: 247-354). Those functions written for the PC version are printed in Appendices.

confidential firm data collected in the Planning Surveys of IUI and the Federation of Swedish Industries, a synthetic dataset for 1990 was prepared for outside use (see Section 3.3).

This manual does not describe the initialization procedure in detail. For complementary detail and explanation see *MOSES Handbook* (Bergholm, 1989) and *MOSES Code* (Albrecht *et al.* 1989). Note that the first part of *MOSES Handbook* on "How to Run the MOSES Model" is superseded by this manual for the PC version of the model. Each section of this manual is a self-contained unit. The user does not have to read previous sections in order to understand the material in any particular section.

This manual uses the following notational conventions.

- Commands that you enter are printed in **boldface** type.
- Variables that you define are printed in *italics*.
- Options are shown within [square brackets]. You may not enter those options.
- Keys to be pressed are printed in SMALL CAPS. For example, the Return key is represented by RETURN.
- Key combinations are printed in **boldface** and are hyphenated. When you see a key combination, such as **Ctrl-d**, you are supposed to hold down the first key (CONTROL) and press the second key (d).
- System prompts are printed in `courier` characters.

SECTION 1

MOSES^{PC} MANUAL

1.1 FILE SYSTEM

The MOSES^{PC} model consists of the following files (or, workspaces, as named in the APL programming language).

moses : The initial interactive part of the model.

MOSES.PC : The main part of the model that contains the model functions.

MOSES.CALIB : File that contains functions used for calibration and policy experiments.

MOSES.FUNEXP : File prepared for experiment-specific functions.

MOSES.GRAPH : File for graphics functions (charts, bar charts, etc.).

MOSES.HELP : File that contains various useful functions to analyze the model, etc.

MOSES.INIT : The main initialization file.

DATA : A data file that contains various real macroeconomic data. Also used by the calibration program.

FUNCTI : File that contains some functions used by the MOSES.INIT workspace.

ISTART : File that contains initialization modification functions. Each function in this file is named ISTART*xx* where *xx* is the modification number.

MSTART : File for model modification functions. Each function in this file is named MSTART*xx* where *xx* is the modification (experiment) number.

MAC*xx* : Macro-dataset for year *xx*. It is used by the MOSES.INIT workspace. There are currently two macro-datasets: one for 1976, the other for 1982.

MIC*xx* : Micro-dataset for year *xx*. It is used by the MOSES.INIT workspace. There are currently two micro-datasets: one for 1976, the other for 1982.

MICRO.DBASE : A data file that contains the results of all planning surveys of establishments (firms or divisions) used in the model. This file is to be updated for every year.

PRT : The APL workspace that contains printing functions. (It is supplied by the Dyalog-APL interpreter.)

VLISTS : A file that contains variables names to be saved at the end of the initialization procedure. It is used by the MOSES.INIT workspace.

R1990.10 : The "synthetic" dataset for 1990. This dataset is created by simulating the model version 2.0 for 8 years by using the dataset R1982.91, and the modification function MSTART900.

R1990.10.SERIE : A file that contains various simulated time-series macro-data for 1983-1989. These data were obtained by simulation to generate the R1990.10 dataset.

R1982.*xx* : The 1982 dataset. *xx* denotes the dataset version number. There are currently six versions for this dataset, namely, 89, 91, 92, 93, 94, and 98 where the second one is the default version.

R1976.5 : The 1976 dataset. There is only one version for this set.

M*xx*R*yyyy.yy.zz* : Default name of output files. *xx*, *yyyy.yy* and *zz* denote the modification version, the initial dataset version, and the time period of the experiment. For example, the results of a 10 year experiment simulated by using MSTART19 (model) modification function and R1982.89 dataset will be M19.R1982.89.10. Note that, by appropriate changes, this file can be used as an initial dataset for further experiments.

R1976.5, R1982*xx*, MIC76, MIC82 and MICRO.DBASE workspaces contain confidential firm data. Because of the disclosure rules, the access to those workspaces is restricted to authorized users.

1.2 A QUICK START

The MOSES^{PC} model can be used in two ways. First, the model can be installed in your computer. In this case, you need a PC with 80386 microprocessor, SCO Xenix Operating System V, Dyalog-APL interpreter, and the model itself. (The model can be used with other APL interpreters and operating systems, but it may take time to transfer the model workspaces from this system to another.) The complete model comes with the moses, MOSES.PC, MOSES.CALIB, MOSES.FUNEXP, MOSES.GRAPH, MOSES.HELP, MOSES.INIT, DATA, FUNCTI, ISTART, MSTART, PRT, R1990.10, R1990.10.SERIE, and VLISTS workspaces.(*) Second, the PC in IUI can be accessed by, for example, using modems by another computer which is operating with the Xenix operating system.

Now, we assume that you use the Toshiba PC at the IUI, and you have a valid user account. When the PC is turned on, the following message is shown after the memory test.

```
XENIX System V
Boot
:
```

Press RETURN to continue (here, if you type **DOS** RETURN, you will log in into DOS operating system). After some messages about the PC's configuration, you are asked to type **Ctrl-D** to proceed with normal startup.(**) Type **Ctrl-D**, and enter time, your user name, and the password when you are asked to do so. Finally, a welcome message is shown and you will be in your home directory. Files in this directory can

*) For the installation of MOSES^{PC} see Appendix D.

**) Instead, you may get the following message if the system is not shutdown properly by the "shutdown" command before you turn it on.

```
The system was not shut down properly, and the root file
system should be cleaned.
Proceed with cleaning (y/n)?
```

Type y RETURN and wait a few seconds for cleaning the root file system.

be listed by the "list" command by typing

# l RETURN

where # is the Xenix prompt ($ or % may also appear. Note that the upper- and lowercase letters are considered to be different in the Xenix operating system.)

Type

# pwd RETURN

to see name of the current directory. If you are not in the /moses directory, type

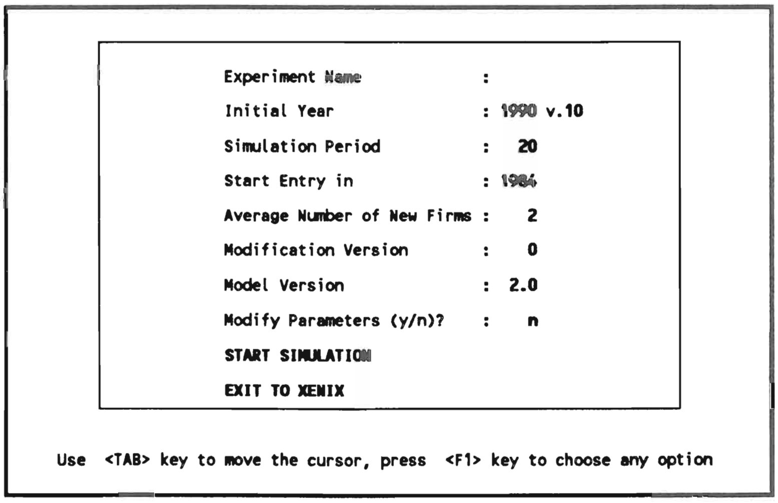
# cd /moses RETURN

to change it.

When you are in the /moses directory, type the following to run the model.

# apl moses RETURN

The MOSES^{PC} logo is shown 4 seconds and then the following menu appears on the screen:



```
Experiment Name                :
Initial Year                   : 1990 v.10
Simulation Period              :  20
Start Entry in                 : 1984
Average Number of New Firms :     2
Modification Version           :   0
Model Version                  :  2.0
Modify Parameters (y/n)?       :   n
START SIMULATION
EXIT TO XENIX

Use  <TAB> key to move the cursor, press  <F1> key to choose any option
```

The cursor can be moved by pressing TAB (downward) or Shift-TAB (upward) keys. After changing default values on the menu as desired, the simulation is started by pressing F1 key when the cursor (the highlighted rectangular) is on "START SIMULATION".

"Experiment Name" defines the workspace name. If you do not enter any name here, the default name as defined in Section 1.1 is used.

"Initial Year" denotes the starting point of the simulation experiment, i.e., the initial dataset year. The simulation commences the first quarter after the "initial year". Since there are only three datasets, this variable should be either 1976, 1982 or 1990. Only the 1990 dataset which is semi-synthetic (see Section 3.4) is available for external use.

The next variable on this line shows the version of data year. For the time being, there is only one version of the 1976 dataset (version 5), six versions of the 1982 dataset (versions 89, 91, 92, 93 ,94, and 98), and one version of the 1990 dataset (version 10).

"Simulation Period" is the number of years that the simulation will proceed. Note that although the MOSES^{PC} model is actually a quarterly model, the simulation period should be specified in years.

"Average Number of New Firms" specifies the entry feature of MOSES (the $\underline{\text{A}}$MAXENT variable in the START∆ENT2 function). It is equal to the maximum number of new firms in each industry in each year when the average industry profitability is equal to unity. The number of new firms is a probabilistic linear function of average industry profitability.[*] If you do not want to use the firm entry option, enter zero for this variable.

"Modification Version" determines which MSTART function will be used to modify the model. In the MOSES model, each experiment is carried out by making changes in the original model (changes in the behavioral

[*] $N_{i,t}$ = ran($\underline{\text{A}}$MAXENT*AVEPROF$_{i,t-1}$) where N is the number of new firms, and AVEPROF is equal to average industry profitability, and subscript i and t denote industry and time (year), respectively. ran(x) draws randomly an integer number from {0, 1, ..., int($\underline{\text{A}}$MAXENT*AVEPROF$_i$)}.

equations, variables, etc.) by using one of the modification functions in the MSTART workspace. This allows you to keep the model in its original form. The MSTARTxx functions where all changes connected with the specific experiment are defined modify the original model at the beginning of the simulation. If you want to make experiments with the original model, enter zero for this variable.

"Model Version" specifies the model version to be used for the experiment. As of May 1991, there are three versions of the MOSES^{PC} model: 1.0, 1.1, and (the default) 2.0. (For details, see Section 1.8.)

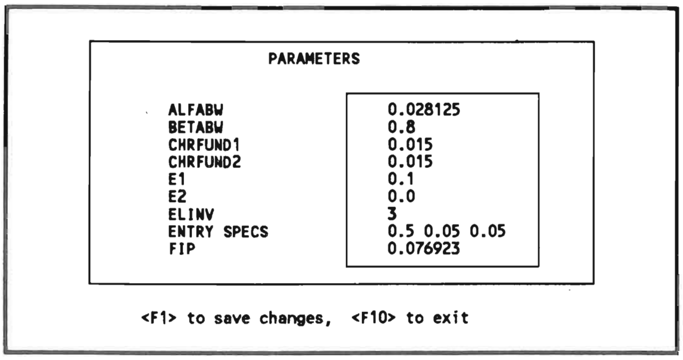
"Modify Parameters (y/n)?" question allows the user to change model parameters just before the experiment. Note that the parameters can also be changed by the initialization procedure and by the modification function (see Sections 1.4 and 2.2). The values entered by using this option has the highest priority (i.e., it supersedes the values specified in the modification function), and it may be convenient to use this option when various experiments are done only for different parameter values.

Pressing F1 key when the cursor is on the "Exit to Xenix" allows the user to exit form the model. To continue to use the model, the user should start by typing "apl moses" at the Xenix prompt.

When the user's answer is "y" for "Modify Parameters (y/n)?", a modification menu appears after some messages about loading the model, database, etc. This menu looks like as follows.

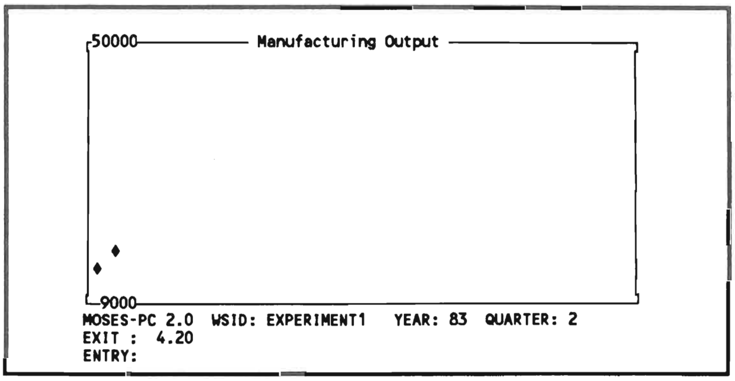
The cursor can be moved by pressing "cursor" keys. Note that all parameter names and values are not shown in the window. Therefore, the user can scroll the data using CURSOR RIGHT and CURSOR DOWN keys. The data in both fields (name and variable fields) scroll accordingly. After having changed the parameter values, press F1 key to resume simulation with modified parameter values. F10 key keeps the parameter values unchanged and start the simulation.

In a normal simulation, some messages about date, the model version, etc., are shown on the screen right before the beginning of the simulation. Then, the simulation is started.



```
                    PARAMETERS

   ALFABW                  0.028125
   BETABW                  0.8
   CHRFUND1                0.015
   CHRFUND2                0.015
   E1                      0.1
   E2                      0.0
   ELINV                   3
   ENTRY SPECS             0.5 0.05 0.05
   FIP                     0.076923

     <F1> to save changes,  <F10> to exit
```

During simulation, the chart below is shown on the screen. On this chart, the numbers at the top-left and bottom-left corners denote the maximum and minimum values of the output, i.e., of the chart. The first line under the chart shows the model version, the active workspace name, and the period of simulation, respectively. The last two lines show the firm codes that exit from or enter to the manufacturing industries. The first number of a firm code denotes the sector where 1 is the raw materials sector, 2 intermediate goods, 3 investment goods and consumer durables, and 4 the consumption goods sector.



```
MOSES-PC 2.0  WSID: EXPERIMENT1   YEAR: 83  QUARTER: 2
EXIT :  4.20
ENTRY:
```

When the simulation is completed, a message is printed on the screen. The results of the experiment can be printed on the screen and be saved in a text file that can be accessed by Xenix and DOS operating systems, and any other DOS-based program (WordPerfect, Lotus-123, etc.)[*].

SPRINT, PPRINT, and FPRINT functions send the output tables to the screen, printer, and a file, respectively. (These functions and other APL commands can be used at the end of the simulation when the system is in the APL's immediate execution mode. In this mode, the cursor stays at the eighth column from the left.) For example, to show all standard tables on the screen whose names are stored in the ALLREPORTS variable, type

SPRINT ALLREPORTS RETURN

When the PPRINT function is used, you are asked to type the printer's name. If the printer connected to the computer is an HP LaserJet printer, type **HPLJ**.

To print or to create a file of all tables, type, respectively,

PPRINT ALLREPORTS RETURN

FPRINT ALLREPORTS RETURN

If you want to print only one table at a time, type

PPRINT *'tablename'* RETURN

Note that the table name should be within single quotation marks. The names of standard output tables are stored in the variable ALLREPORTS. To see those names, type

ALLREPORTS RETURN

The graphics functions can be accessed by copying the MOSES.GRAPH workspace into the active workspace. To do this, first, type

*) To copy a text (ASCII) file from a XENIX directory to a DOS directory on the hard disk, use the following XENIX command.

# doscp */XENIXpathname/file.name* **/dev/hd0d:***/DOSpathname/filename* RETURN

Note that **/dev/hd0d:** denotes the (first) hard disk. Notice also that normal slash ("/") is used to specify DOS path names instead of back-slash ("\") as normally used by DOS.

)COPY MOSES.GRAPH RETURN

After copying the MOSES.GRAPH workspace, the PLOT, CHART, BARCHART, and PIECHART functions can be used. For example, to use the PIECHART function, simply type

PIECHART RETURN

When the graphics functions are invoked, a full-screen input menu which is self-explanatory appears. On-line help is also available for these functions.

The whole output workspace can be saved by the APL command

)SAVE RETURN

Before saving the active workspace, its name can be changed as follows.

[]WSID ← *workspacename* RETURN

where [] is the APL character entered by Shift-L when the keyboard is in the APL mode. (The Dyalog APL character set contains both the APL and ASCII characters, and uses the keys Ctrl-N and Ctrl-O to switch between these sets, respectively.)

This workspace can also be used, after appropriate changes, as an initial dataset for further experiment.

Finally, typing

)OFF RETURN

allows the user to exit to the operating system.

1.3 BASIC APL FUNCTIONS

A basic knowledge of the APL language is essential for designing and running simulation experiments on the MOSES model. Hence, some APL functions/commands are summarized in this section.

The place in the computer where the work is done is called *workspace* in the APL language. An APL workspace consists mainly of functions and variables. The user enters functions and variables in the workspace and can save them for future usage. The files mentioned in Section 1.1 are those APL workspaces that form the MOSES^{PC} model.

APL system commands provide services or information associated with the workspace and the external environment. All system commands begin with the symbol ")", known as a right parenthesis. (Note that when the keyboard is in APL mode, ")" is entered by typing "Shift-".) System commands may be entered from the immediate execution mode. In this mode, the blinking cursor stays at the eighth column from the left on the screen. In other words, the user can enter those commands only when the simulation is completed or has stopped for any reason.

Some useful system commands are as follows.

)FNS displays the names of globally defined functions in the active (current) workspace.

)VARS displays the names of globally defined variables in the active workspace.

)SAVE saves the active workspace (in a file).

)COPY *workspacename [names]* brings all or selected global objects from the *workspace* stored previously. If the list of *[names]* is excluded, all defined objects (functions and variables) are copied. Existing global objects in the active workspace with the same name as a copied object are replaced.

)OFF terminates the APL section, returning to the Xenix shell command level with the standard prompt (normally #, $ or %).

)SI displays the contents of the state indicator in the active workspace. The state indicator identifies those operations which are suspended or pendent for each suspension. The list consists of a line for each suspended or pendent operation, beginning with the most recently suspended function. This command is very useful to locate errors when the simulation is unexpectedly halted.

)RESET cancels all suspensions recorded in the state indicator.

[]WSID is a system variable that contains the identification name of the active workspace. If a new name is assigned, that name becomes the identification name of the active workspace, provided that it is named according to Xenix (UNIX) filename conventions. For example, typing

[]WSID RETURN

will return the active workspace name. If the user types

[]WSID ← 'EXPERIMENT_1' RETURN

a new workspace name, EXPERIMENT_1, will be assigned.

The environmental parameters of the APL are defined in the script /usr/bin/apl. Currently, the maximum workspace size is restricted to 6 MByte. An HP plotter (PLOTTER) and an HP laser printer (PRINTER) connected to the first parallel port are defined in this script file. If you wish to change any parameter, you must edit the script /usr/bin/apl. Note that this requires super-user privilege. (You may override the system-wide parameters by defining your own environmental parameter before invoking

APL. See Dyalog APL User Guide for details).

1.4 PREPARING AN EXPERIMENT

In the MOSES^{PC} model, each experiment is carried out by means of modifications in the original model (changes in the behavioral equations, variables, etc.) using the functions in the MSTART workspace only. This allows the user to keep the model in its original form. Therefore, the user should know about the model, the APL interpreter, and APL's full-screen editor, "VIA".

A modification function, MSTART*xx*, can have six types of modifications.

1) It may contain a description of the experiment which is printed at the top of all output tables. It can be defined by adding the following line into the MSTART*xx* function.

DSCR ← DSCR,' *description'*

DSCR is the MOSES variable that contains the description information.

2) It may contain commands for data storage. The model keeps track of time-series data for major industry- and economy-wide variables in various tables (see the following section for those tables). If the user would like to add more tables to this standard output, the following functions are available.

scale **Y▴R▴FIRM** *firmcode*
scale **Y▴R▴FIRM▴Q** *firmcode*
scale **Y▴R▴FIRM▴F** *firmcode*
scale **Y▴R▴FIRM▴FINANCE** *firmcode*
NEW▴FIRM
NEW▴FIRM▴Q
NEW▴FIRM▴F
scale **NEW▴FIRM▴FINANCE**
INITIAL▴NEW▴FIRMS
variable **SAVED▴IN▴Y** *tablename*
variable **SAVED▴IN▴Q** *tablename*
Y▴R▴MARKET▴F
Y▴R▴MARKET▴Q
Y▴R▴INDUSTRY▴F
Y▴R▴INDUSTRY▴Q

The first four functions are used to keep track of various firm data. *scale* is the scaling factor for some variables. (100 is used by the new firm functions.) *firmcode* is a number that denotes the specific firm under investigation. The first number of the firm code denotes the sector where 1 is the raw materials sector, 2 intermediate goods, 3 investment goods and consumer durables, and 4 the consumption goods sector, and the second number denotes the position of the firm in its sector. For example, 3.4 is used for the fourth firm in the investment goods sector.

The NEW▴FIRM functions are identical to the previous functions, except that these functions save time-series data of *all* new firms. Therefore, there is no need to specify a firm code. Note that in long-run simulation experiments, the memory requirements of those tables created by the NEW▴FIRM functions can increase considerably.

The INITIAL▴NEW▴FIRMS function keeps track of data for the characteristics of new firms when they enter the industry.

The SAVED▴IN▴Y and SAVED▴IN▴Q functions are used to store any annual and quarterly variable, respectively, in a table whose name is defined by *tablename* (for more information about these functions and output tables created by them, see the following section).

The Y▴R▴MARKET▴F, Y▴R▴MARKET▴Q, Y▴R▴INDUSTRY▴F, and Y▴R▴INDUSTRY▴Q functions are similar to the Y▴R▴MARKET, and Y▴R▴INDUSTRY functions but they contain different market and industry data.

3) The modification functions may contain commands to modify model variables. For example, the following line changes the KSI variable from its default value, .15, to .25.

KSI ← .25

4) The modification function may contain commands that change the MOSES functions. For this purpose, the following functions can be used.

'functionname' **MODADD** *'linebeginningwnewline'*
'functionname' **MODADDLAST** *'newline'*

'functionname' **MODSUBST** *'oldlinebeginning⍵newline'*
'functionname' **MODDEL** *'oldlinebeginning'*
'functionname' **MODADDLINE** *'newline' linenumber*

In those functions, the user should use single quotation marks (') as shown above. The omega symbol, ⍵, is the APL character entered by **Shift-W.** *functionname* denotes the function that is to be modified.

The MODADD function adds a *newline* into the *functionname* after the line whose initial characters correspond to the *linebeginning*. In other words, **only** the beginning characters of the old line stands before the omega symbol, and the whole new line after this symbol. (The length of the *linebeginning* should be chosen such that there is no more than one line with the same *linebeginning*.)

The MODADDLAST function adds the *newline* into the *functionname* as the last line.

The MODSUBST function substitutes the *newline* for the line whose initial characters correspond to the *linebeginning* in the *functionname*.

The MODDEL function deletes the line whose initial characters correspond to the *oldlinebeginning* in the *functionname*.

It might be ambiguous what line you are referring to when too few letters are specified in the *linebeginning* or *oldlinebeginning*. Therefore, when these modification functions (MODADD, MODSUBST, and MODDEL) are used, the user should verify that the changes are made properly.

Finally, the MODADDLINE function adds the *newline* into the *functionname* as the line defined by its *linenumber*.

5) During the simulation of the model, a chart showing a variable defined by the user is shown on the screen. It is also possible to print some quarterly data defined by the TRACE1 (or TRACE2) function instead of this chart.

The function that prepares the chart is as follows.

'varname' 'title' [*'background'*] PREP∆INT∆GRAPH *min max period*

In this function, *varname* is the name of the variable to be plotted, *title* is the title of the chart (it should be less than 80 characters), *background* is the name of a 22x80 text matrix that is to be shown as the background of the chart, *min* and *max* are the minimum and maximum values of the variable (i.e., the scale of the Y-axis), and *period* is the period of simulation in years. *background* can be the chart prepared during a reference simulation, and it can be used to show the progress of the current simulation compared to the reference case. You can also use the actual data for *background*. To do this, prepare a 22x80 text matrix that contains the chart of the actual data as specified by the *min* and *max* values, by using the PLOT function (see Appendix C).

This function is called by the RUNEXP function. If you want to use the TRACE1 (or TRACE2) function instead of this default option, the line in the RUNEXP function that calls this function should be deleted by using the MODDEL function. Note that TRACE1 and TRACE2 functions can only be used with versions 1.0 and 1.1, respectively.

6) The STOP▲HERE function can be called in any place to stop the simulation temporarily. This function is used as follows.

STOP▲HERE *'message'*

where *message* is the message to be shown on the screen when the simulation is stopped. This function can be used for, for example, manual data entry for "interactive" simulation games, etc.

An example can be used to clarify the above-mentioned types of modification.

```
     ▼  MSTART99
[1]  DSCR ← DSCR,' A MODIFICATION EXAMPLE'
[2]  KSI ← .25
[3]  1 Y▲R▲FIRM 3.16
[4]  'HOUSEHOLD▲UPDATE' MODADD 'QC[NDUR]wMTEC←10+MTEC'
[5]  'YEARLY▲EXP' MODSUBST 'EXPDW←wEXPDW←(.8xEXPXDW)+
     (.2xEXPIDW)'
[6]  'RUNEXP'    MODSUBS    '"Q▲Q"w"RU"    "Rate    of
     Unemployment" PREP▲INT▲GRAPH 0 100 20'
[7]  'YEARLY▲TARG' MODADDLAST 'STOP▲HERE "MODIFY
     YEARLY TARGETS" '
     ▼
```

In this example, the first line adds "A MODIFICATION EXAMPLE" into the description variable, DSCR. The second line changes the KSI parameter from its default value to .25. The third line executes the Y∆R∆FIRM function that stores the time-series data for the firm 3.16. The fourth line means that a new line, MTEC ← 10 + MTEC, will be added into the HOUSEHOLD∆UPDATE function after the line beginning with QC[NDUR]. The fifth line means that a new line, EXPDW ← (.8xEXPXDW) + (.2xEXPIDW) will be substituted for the line beginning with EXPDW← in the YEARLY∆EXP function. In other words, this line changes the EXPDW equation in the function. The sixth line changes the on-line plot function so that the rate of unemployment variable will be plotted for 20 years. The last line stops the simulation after the yearly profit targets are set in the YEARLY∆TARG function. The user can "manually" modify profit targets before allowing the simulation to continue. For example, the fifth firms' profit target can be modified as follows.

```
MHIST[5]←.80 RETURN
→[]LC RETURN
```

(The second line resumes the simulation.)

These changes in the model take place when the function UPDATEMOSES is called. That is, the MSTARTxx function is called on a line in the UPDATEMOSES function. If you wish to check that the changes in the model have been performed correctly, press once

Ctrl-BACKSPACE

after the messages about the modification of the model. List the functions you are interested in by typing

∇*functionname*∇ RETURN

where ∇ is the APL symbol entered by Shift-G. (You can also use the editor VIA, and make changes). Type

```
→[]LC RETURN
```

to continue the simulation of the model, where → and [] are APL symbols. []LC is a system constant that is equal to the number of the line where the simulation is stopped.

All modification functions should be in the MSTART workspace. Therefore, load MSTART workspace into the active workspace when a new modification function is to be written. The full-screen function editor "VIA" can be used for writing and editing (for more information about this editor, see Dyalog-APL User Guide).

1.5 OUTPUT REPORTING

Standard output tables that contain various annual time-series data are prepared by the "Y▲R▲*tablename*" functions. In these functions the table header is setup and the table name for storage of data is defined, usually as "YEARLY▲*tablename*". Finally a function by the name "Y▲*tablename*" is called for defining and storing the data.

The variables going into a table are defined and entered into the table, "YEARLY▲*tablename*" by the function "Y▲*tablename*" each year. The variable "YEARLY▲*tablename*" is in matrix form, the first row containing the variable names stored in the table.

Currently, the following tables are standard output of the model.

Table name: YEARLY▲INDUSTRY▲TOTAL
Aggregate statistics for manufacturing industry

Colon	Variable	Description
no	name	
1	QTOP	Potential output level (MS)(*)
2	TEC	Average potential labor productivity (MS)
3	L	Employment level (MS)
4	PROD	Actual labor productivity
5	DQ	Annual change in production (MS)
6	A21	A slack variable (MS)
7	A22	A slack variable (MS)
8	SUM	A21 + A22
9	A23	A slack variable (MS)
10	M*S	Gross income (total sale * profit margin)
11	STO	Output stock level relative to the desired level (MS)
12	DS	Annual change in sales (MS)
13	DP	Annual change in prices (MS)
14	DW	Annual change in wages (MS)
15	M	Profit margin (MS)
16	INV	Investment level
17	LTOT	Total labor force
18	RU	Rate of unemployment (MS)

Table name: YEARLY▲GNPFIX▲PROD
GNP accounts, supply side, 1982 prices

Colon	Variable	Description
no	name	
1	TOT	Total GNP
2	RAW	Raw materials
3	IMED	Intermediate goods
4	INV/DUR	Investment and durable goods
5	NDUR	Non-durable (consumer) goods
6	A/F/F	Agriculture/forestry/fishery
7	ORE	Mining
8	BLD	Building and construction
9	EL	Electricity, gas, water
10	SERVICE	Services
11	WSG	Government (wage payments) (MS)

*) MS: MOSES variable, i.e., the same variable name is used in the model. For a detailed description of these variables, see *MOSES Code* (Albrecht *et al*, 1989: 196-220).

Table name: YEARLY▲GNPFIX▲USE
GNP accounts, demand side, 1982 prices

Colon no	Variable name	Description
1	TOT	Total GNP
2	GTOT	Government total (WSG+PURCHG)
3	WSG	Government wage payments (MS)
4	PURCHG	Government consumption (MS)
5	HH	Household consumption
6	INV-TOT	Total gross investment
7	INV-MKT	Investment in manufacturing
8	INV-BLD	Investment in building and construction
9	INV-GOV	Public investment
10	CHSTO	Change in stocks (MS)
11	EXPORT	Exports (MS)
12	IMPORT	Imports (MS)

Table name: YEARLY▲GNPCUR▲USE
GNP accounts, demand side, current prices

[Same as YEARLY▲GNPFIX▲USE]

Table name: YEARLY▲GNPCUR▲PROD
GNP accounts, supply side, current prices

[Same as YEARLY▲GNPFIX▲PROD]

Table name: YEARLY▴FINANCE
Financial aggregates for manufacturing industry

Colon no	Variable name	Description
1	M*S	Gross income (total sale * profit margin)
2	INTPAY	Interest payments (MS)
3	DPER	Depreciation
4	TAXES	Taxes
5	DIV	Dividends
6	SUBS	Subsidies (MS)
7	CHBW	Change in borrowing (value) (MS)
8	INV	Gross investment
9	CHK2	Change in current assets (value) (MS)
10	K1	Value of physical assets (MS)
11	K2	Value of current assets (MS)
12	K3-IN	Value of input inventories (K3IMED)
13	K3-OUT	Value of output inventories (K3FINISH)
14	BW	Level of borrowing (debt) (MS)
15	NW	Net worth (MS)
16	TOT	Total value

Table name: YEARLY▴GOVERNMENT
Variables on public finance

Colon no	Variable name	Description
1	LG	Government employment (MS)
2	WG	Wage level (MS)
3	DWG	Annual change in government wages (MS)
4	WSG	Total wage bill (MS)
5	PURCHG	Government consumption (MS)
6	TRANS	Transfer payments (MS)
7	SUBS	Subsidies (MS)
8	SPG	Total spending
9	INVG	Gross government investment (MS)
10	ITAX	Income taxes (MS)
11	WTAX	Wage tax (MS)
12	VATAX	Value added tax (MS)
13	CTAX	Corporate tax (MS)
14	INCOME	Total income
15	INTPAY	Interest payments (INTPAYG)
16	SURPLUS	Surplus (or deficit) (SURPLUSG)

Table name: YEARLY▲BANK▲TRANSACTIONS
Variables on the bank transactions

Colon	Variable	Description
no	name	
1	INTF	Interest payments by firms
2	INTK2	Interest payments for current assets (paid to firms)
3	INTH	Interest payments to/from households
4	INTG	Interest payments to/from government
5	INTGFOR	Government interest payments to foreigners
6	CHBW	Change in borrowing (value)
7	CHK2	Change in firms' current assets (value)
8	SAVH	Households' savings
9	CHDEPG	Change in government's position (MS)
10	CHDEPGF	Change in government's position (foreign accounts)
11	EXPORT	Exports (MS)
12	FASSPAY	Payments of foreign trade credits (MS)
13	IMPORT	Imports (MS)
14	FDPAY	Payments of foreign trade debts (MS)
15	CHNBW	Change in net borrowing (value)

Table name: YEARLY▲BANK▲POSITION
Variables on the bank position

Colon	Variable	Description
no	name	
1	BW	Total borrowing (from the bank)
2	K2	Firms' current assets held in the bank (MS)
3	HH	Households' deposits
4	G	Government deposits
5	LIQB	Bank's holding of "liquidity" (MS)
6	LIQBFOR	Bank's holding of foreign "liquidity" (MS)
7	FASS	Foreign trade credit (MS)
8	FD	Foreign trade debts (MS)
9	FNASS	FASS-FD
10	NETFOR	Net foreign position
11	NW	Bank's net worth (NWB)
12	GFOR	Government's foreign assets (POSGFOR)

Table name: YEARLY▲COUNTRY▲TOTAL
Some national statistics

Colon no	Variable name	Description
1	GNPFIX	GNP (1982 prices)
2	GNPCUR	GNP (current prices)
3	MONEY	Money stock (MS)
4	VEL	Velocity of money
5	RI	Rate of interest (MS)
6	PRINT	New issue of money (MPRINT)
7	CHLIQB	Change in liquid balances (value)
8	CHINV	Change in investment (value)
9	CHDIV	Change in dividends (value)
10	CHKIN	Change in inventories (value)
11	TOT	Total of 6-10.

Table name: YEARLY▲PRICES
Price statistics

Colon no	Variable name	Description
1	QPDOM 1	Level of domestic prices of raw materials (MS)
2	QPDOM 2	Same, intermediate goods (MS)
3	QPDOM 3	Same, investment and durable goods (MS)
4	QPDOM 4	Same, non-durable goods (MS)
5	QPFOR 1	Level of foreign prices of raw materials (MS)
6	QPFOR 2	Same, intermediate goods (MS)
7	QPFOR 3	Same, investment and durable goods (MS)
8	QPFOR 4	Same, non-durable goods (MS)
9	W 1	Wages in raw materials sector
10	W 2	Same, intermediate goods sector
11	W 3	Same, investment and durable goods sector
12	W 4	Same, non-durable goods sector
13	M 1	Profit margins in raw materials sector (MS)
14	M 2	Same, intermediate goods sector (MS)
15	M 3	Same, investment and durable goods sector (MS)
16	M 4	Same, non-durable goods sector (MS)
17	RR 1	Rate of returns in raw materials sector (MS)
18	M 2	Same, intermediate goods sector (MS)
19	M 3	Same, investment and durable goods sector (MS)
20	M 4	Same, non-durable goods sector (MS)

Table name: YEARLY▲FOREIGN▲TRADE
Foreign trade statistics

Colon	Variable	Description
no	name	
1	X 1	Export share in raw materials (MS)
2	X 2	Same, intermediate goods (MS)
3	X 3	Same, investment and durable goods (MS)
4	X 4	Same, non-durable goods (MS)
6	X 5	Average export share
7	IMP 1	Export share in raw materials (MS)
8	IMP 2	Same, intermediate goods (MS)
9	IMP 3	Same, investment and durable goods (MS)
10	IMP 4	Same, non-durable goods (MS)
11	IMP 5	Average import share

Table name: YEARLY▲HOUSEHOLDS
Household statistics

Colon	Variable	Description
no	name	
1	DDI	Annual change in disposable income (MS)
2	SP 1	Share of raw materials consumption in disposable income
3	SP 2	Share of intermediate goods consumption in disposable income
4	SP 3	Share of durable goods consumption in disposable income
5	SP 4	Share of non-durable goods consumption in disposable income
6	SP 5	Share of A/F/F products consumption in disposable income
7	SP 6	Share of mining consumption in disposable income
8	SP 7	Share of oil consumption in disposable income
9	SP 8	Share of building consumption in disposable income
10	SP 9	Share of electricity/gas/water consumption in disposable income
11	SP 10	Share of services consumption in disposable income
12	SAVH	Share of savings in disposable income
13	DCPI	Annual change in consumer price index
14	PURCH	Total purchasing (value)
15	SAVH	Total savings (value)

Table name: YEARLY▲MARKET1
Production statistics of raw materials sector

Colon no	Variable name	Description
1-16	same as in YEARLY▲INDUSTRY▲TOTAL	
17	QPDOM	Domestic price level (MS)
18	QPFOR	Foreign price level (MS)

Table name: YEARLY▲MARKET2
Production statistics of intermediate goods sector

[Same as YEARLY▲MARKET1]

Table name: YEARLY▲MARKET3
Production statistics of investment and durable goods sector

[Same as YEARLY▲MARKET1]

Table name: YEARLY▲MARKET4
Production statistics of nun-durable goods sector

[Same as YEARLY▲MARKET1]

Table name: YEARLY▴FINANCE1
Financial statistics of raw materials sector

[Same as YEARLY▴FINANCE table]

Table name: YEARLY▴FINANCE2
Financial statistics of intermediate goods sector

[Same as YEARLY▴FINANCE table]

Table name: YEARLY▴FINANCE3
Financial statistics of investment and durable goods sector

[Same as YEARLY▴FINANCE table]

Table name: YEARLY▴FINANCE4
Financial statistics of non-durable goods sector

[Same as YEARLY▴FINANCE table]

In Version 2.0, following tables are also prepared.

Table name: YEARLY▲RATES
Exchange rate and interest rate data

Colon	Variable	Description
no	name	
1	XRATE	Exchange rate (MS)
2	RI LEND	Interest rate (credits) (MS)
3	RI DEP	Interest rate (deposits)
4	RIDIFF	Interest rate differential (RI DEP/RI LEND)
5	RFUND1	A coefficient used in the CREDIT▲MARKET function (MS)
6	RFUND2	A coefficient used in the CREDIT▲MARKET function (MS)

Table name: YEARLY▲EXPECTATIONS
Data on manufacturing firms' expectations

Colon	Variable	Description
no	name	
1	EXPDS	Expected annual change in sales (MS)
2	DS	Actual change in sales (MS)
3	HIGH S	Percentage of firms that overestimated sales growth
4	LOW S	Percentage of firms that underestimated sales growth
5	EXPDP	Expected annual change in prices (MS)
6	DP	Actual change in prices (MS)
7	HIGH P	Percentage of firms that overestimated price increase
8	LOW P	Percentage of firms that underestimated price increase
9	EXPDW	Expected annual change in wages (MS)
10	DW	Actual change in wages (MS)
11	HIGH W	Percentage of firms that overestimated wage increase
12	LOW W	Percentage of firms that underestimated wage increase

Moreover, there are two tables called ED▴NULLIFIED and ED▴PARAMETERS. The form of these two tables are different from those of others. They contain data on the nullified firms, and the simulation parameters, respectively.

The user could add more tables to this standard (default) output by calling any one of the following functions by the MSTART*xx* function.

Y▴R▴MARKET▴F
Y▴R▴MARKET▴Q
Y▴R▴INDUSTRY▴F
Y▴R▴INDUSTRY▴Q
scale **Y▴R▴FIRM** *firmcode*
scale **Y▴R▴FIRM▴Q** *firmcode*
scale **Y▴R▴FIRM▴F** *firmcode*
scale **Y▴R▴FIRM▴FINANCE** *firmcode*
NEW▴FIRM
NEW▴FIRM▴Q
NEW▴FIRM▴F
NEW▴FIRM▴FINANCE
INITIAL▴NEW▴FIRMS
variable **SAVED▴IN▴Y** *tablename*
variable **SAVED▴IN▴Q** *tablename*

The Y▴R▴MARKET▴F and Y▴R▴MARKET▴Q functions create tables for each sector that contain the following variables.

Table name	**Variables in the table**
YEARLY▴MARKET▴F	DIV (dividends)
	INV (investment level)
	NW (net worth)
	DNW (annual change in net worth)
	DBW (annual change in borrowing)
	RR (rate of return)
	K1 (value of physical assets)
	VA (value added)
	RWCIF (profitability)
YEARLY▴MARKET▴Q	L (employment level)
	DS (annual change in sales)
	DQ (annual change in production)
	DW (annual change in wages)
	M (profit margin)
	XS (export share)
	MKTSHIND (market share index)

When one of the firm level data storage functions (Y▲R▲FIRM, Y▲R▲FIRM▲F, Y▲R▲FIRM▲Q, and Y▲R▲FIRM▲FINANCE) are used, a new table for each firm is created. The name of the table is formed as follows.

YEARLY▲*functionname*▲*c1*X̲*c2*

where *functionname* is the last part of the function that create the table (e.g., if the table is created by the Y▲R▲FIRM function, "FIRM" is the *functionname* and *c1* and *c2* are the integer and decimal parts of the firm code. For example, the function

100 Y▲R▲FIRM▲F 3.16

will create a table for the firm 3.16 with the following name:

YEARLY▲FIRM▲F▲3X̲16

New firm functions (NEW▲FIRM, NEW▲FIRM▲F, NEW▲FIRM▲Q, and NEW▲FIRM▲FINANCE) activate the corresponding firm level data storage functions for *all* new firms. If these functions are used for long run simulations with a large number of new firms (entries), memory requirements may grow substantially. Therefore, it is recommended not to use these "NEW▲FIRM" functions unless it is absolutely necessary. The variables stored in these tables at the firm-level are the same as the corresponding industry tables shown before.

The name of the table created by the INITIAL▲NEW▲FIRMS is FIRMCHARC, and it contains the data about the initial characteristics of new firms.

Names of all these tables will be added to the variable, ALLREPORTS. Names of those tables created by the SAVED▲IN▲Y and SAVED▲IN▲Q functions are *not* added to the ALLREPORTS variable.

The output tables are in the form of a (T+1)xV matrix where T is the duration of simulation in years (i.e., the number of annual data for each variable), and V the number of variables in the table. The first row contains variable names as shown above. Thus, for example, the first ten years of the QTOP series for the second sector can be shown by typing

YEARLY▲MARKET2[1+*ι*20;1] RETURN

where $\hat{\imath}$ is the APL indexing character entered by Shift-I.

Tables created by the SAVED∆IN∆Y and SAVED∆IN∆Q functions are in the form of a (T+1) element vector. The first element of the vector contains the variable name. Note that when a cross-sectional variable is saved, the elements of this vector are also in the vector form.

For example, assume that the following line is added into the MSTART*xx* function to store time-series data of all firms' potential output, QTOP.

QTOP SAVED∆IN∆Y 'QTOP∆SERIES'

The QTOP∆SERIES table will be a T+1 element vector. Each element of this table, Q_t, $1<t$, will be an N_{ti} element vector where t and i denote time and the number of firms at time t, respectively.

Three functions are available for printing the data tables: SPRINT, PPRINT, and FPRINT. SPRINT shows tables on the screen, PPRINT sends them to a printer, and FPRINT creates a text file that contains those tables.

To show any table(s) on the screen, type

SPRINT *tablename* RETURN

Note that *tablename* should be in the form of a text vector (for a single table) or a text matrix (for many tables) whose rows contain table names. Therefore, when a single table is to be shown on the screen, the name should be in single quotation marks. For example, to show ED∆PARAMETERS on the screen, type

SPRINT 'ED∆PARAMETERS' RETURN

To show all tables whose names are in the ALLREPORTS variable, type

SPRINT ALLREPORTS RETURN

To show the first and sixth tables of the ALLREPORTS variable (i.e., YEARLY∆INDUSTRY∆TOTAL and YEARLY∆FINANCE tables), type

SPRINT ALLREPORTS[1 6;] RETURN

The PPRINT and FPRINT functions are used similarly. When these two functions are used, the user is asked to supply the print width in number of columns, a printer name (for the PPRINT function), and a file name (for the FPRINT function). (The maximum print width for the HP LaserJet printer is 91.)

Printer names currently supported are as follows.

Name	**Printer**
STD	ASCII or standard APL printer
EPSON	Epson FX printers
HPLJ	HP LaserJet printer
PROPRINTER	IBM Proprinter
TOSHIBA	Toshiba 351P with APL font cartridge

File names should be in accordance with the Xenix file name conventions. Recall that small and capital letters are considered different in the Xenix operating system. Therefore, if you want to use small letters, type **Ctrl-O** to switch the keyboard to ASCII mode. Type **Ctrl-N** to go back to the APL mode.

For those tables that are not in the form of standard tables and all other types of variables, the function FILE▲OF can be used to create a copy of the variable in a text file. This function can be used as follows.

FILE▲OF *variablename* RETURN

Similarly, any variable can be manually printed by using those functions in the PRT workspace. For this purpose, copy the PRT workspace into the active workspace, turn the printer on, print the variable, and turn the printer off as follows.

)COPY PRT RETURN

)PRTON▲HPLJ RETURN [for other types of printers, use the appropriate function]

)PRT *variablename* RETURN

)PRTOFF RETURN

The file created by the FPRINT and FILE▲OF functions is a text

(ASCII) file, and can be accessed by the Xenix text editor, "vi". This file can also be copied into any DOS-based media (hard diskette or floppy diskette) by the Xenix's "doscp" command, and can be accessed by any popular DOS-based program such as WordPerfect, T^3, Lotus, etc.

The MOSES.GRAPH workspace contains various graphics functions. These graphics functions can be accessed by copying the MOSES.GRAPH workspace into the active workspace. To do this, first, type

)COPY MOSES.GRAPH RETURN

After copying the MOSES.GRAPH workspace, the PLOT, CHART, BARCHART, and PIECHART functions can be used. For example, to use the PIECHART function, type

PIECHART RETURN

When the graphics functions are invoked, a full-screen input menu which is self-explanatory appears. On-line help is also available for these functions. After filling out the required sections of this full-screen input menu, type **F10** or RETURN key to show the chart. To go back to the menu, type RETURN or any one of the function keys.

In the menu of those charts, you can enter any valid APL expression for the X and Y values to be plotted. (Remember to press Ctrl-N if you want to input APL symbols.) For example, to draw a chart of total GNP in current prices, enter the following expression in any one of the data lines:

YEARLY∆GNPCUR∆PROD[1+⍳T;1]

where T is the time period. The number after the semi-colon (;) denotes the position of the variable in the table. In this case, total GNP is the first variable in the YEARLY∆GNPCUR∆PROD table.

The CHART, PIECHART, and BARCHART functions can also be used to plot or to print the graphics of variables. Use the **F5** key after the expressions are filled out on the menu. You will be asked to select between a plotter and a printer. Press **P** for plotter, and **R** for printer. (The default plotter and printer are HP plotter and HP laserjet printer, respectively, using the first parallel port. Those options can be changed by editing the script /usr/bin/apl. See Section 4 for details.) If you use a plotter, you may be asked to change the paper.

The PLOT function does not use graphics facilities. It is based on Dyalog-APL's full-screen applications of the []SM and []SR functions. The plot that is shown on the screen can also be saved as a 25x80 text matrix.

Another graphics function, SHOWFUN, shows an animated time series chart of the production functions of industries or firms. The output tables should be available for this function. This function is used as follows.

SHOWFUN *'firmcode'* RETURN

For the manufacturing industry, and the raw materials, intermediate goods, capital goods, and consumer goods sectors, enter 0, 1, 2, 3, and 4, respectively, for the *firmcode*. (Those graphics functions are explained in detail in Appendix C.)

The CHART, PIECHART, and BARCHART functions use the SCO CGI Graphics Run-Time System, and are written by modifying some graphics functions supplied in the CGIDEMO workspace of the Dyalog-APL interpreter.

The SCO Graphics^{TM} 1.0 Run-Time System (by The Santa Cruz Operation, Inc.) has some known bugs and omissions. A list of those problems are stated in the variable CGIBUGS which is contained in the CGIDEMO workspace. To read this variable, type

)COPY CGIDEMO RETURN

CGIBUGS RETURN

There is also another list in the "Release and Installation Notes" of the SCO Graphics^{TM} package. The most important problem of the Graphics Run-Time System is the fact that the computer may be locked during graphics applications without any apparent reason. Although the graphics functions are modified so that almost all sources of problems are checked by the function before using the CGI auxiliary processor, this problem may still persist to some extent. Therefore the user is strongly recommended to save the active workspace *before* using the graphics functions.

If the computer is locked, turn it off, wait a few seconds, and turn it on. Since the system is not properly shut down in this case, the root file system *should be cleaned* before login into your account.

1.6 PREPARING A NEW DATASET AND ADDING FIRMS INTO EXISTING DATASETS

1.6.1 Preparing a new dataset

A new dataset can be generated by three methods.

First, a "synthetic" dataset can be produced by a simulation experiment. For this purpose, simulate the model by using any one of the modification functions, and use the SAVE∆OUTPUT function to create the new dataset. The SAVE∆OUTPUT function saves the active workspace and deletes all variables and functions which are not required in a MOSES dataset. You can use this dataset for further experiments.

Second, you can add new variables or firms into one of the datasets. If a new variable is added to the dataset, the name of the variable should be added to VARIABLE∆NAMES or PARAMETER∆NAMES variables in the MOSES∆PC workspace. These variables contain the names of variables in the dataset used in some functions. If the added variable is a firm-level variable, an appropriate line should be added into NULLIFY, FIRMENTRY2, FIRMENTRY∆PC and MERGE∆WITH functions in the MOSES.PC workspace.

Third, a completely new dataset can be generated by using new micro and macro data. See Section 3 for this procedure.

1.6.2 Entering new (real) firms into existing datasets

Real firms can also be entered into the model in any year by using functions saved in the MOSES.FUNEXP workspace. Although macro-consistency is affected in this case, if the size of new firms is small relative to the economy, this may not be a serious problem. To enter new real firms, a variable called REAL∆FIRM∆DATA should be prepared. This variable can easily be prepared by using the REAL∆DATA∆ENTRY function. To do so, first load the MOSES.FUNEXP workspace by typing

```
) LOAD  MOSES.FUNEXP  RETURN
```

when you are in APL.(*)

Then type

REAL▲DATA▲ENTRY

This function will ask you to enter following firm data. (Enter the values in millions of SEK, and ratios as percentages. For example enter 10 for 10%.)

```
YEAR
SECTOR NUMBER
DESIRED K2/SALES RATIO
VALUE OF A21
VALUE OF A22
MARKET VALUE OF PHYSICAL ASSETS
BOOK VALUE OF PHYSICAL ASSETS
VALUE OF CURRENT ASSETS
TOTAL DEBT
TOTAL INVESTMENT IN THIS YEAR
TOTAL INVESTMENT IN LAST YEAR
DOMESTIC SALES IN THIS YEAR
DOMESTIC SALES IN LAST YEAR
FOREIGN SALES IN THIS YEAR
FOREIGN SALES IN LAST YEAR
INCREASE IN PRODUCTION VOLUME THIS YEAR (PERCENT)
TOTAL WAGE BILL THIS YEAR
TOTAL WAGE BILL LAST YEAR
NUMBER OF EMPLOYEES IN THIS YEAR
NUMBER OF EMPLOYEES IN LAST YEAR
VALUE OF TOTAL INPUTS (RAW MATERIALS, SERVICES,
     ETC.) THIS YEAR
VALUE OF TOTAL INPUTS (RAW MATERIALS, SERVICES,
     ETC.) PAST YEAR
EXPECTED VALUE OF SALES IN NEXT YEAR
EXPECTED INCREASE IN PRODUCTION VOLUME IN NEXT
     YEAR (PERCENT)
EXPECTED INCREASE IN OUTPUT PRICES IN NEXT YEAR
     (PERCENT)
CURRENT INPUT INVENTORIES / SALES RATIO
DESIRED INPUT INVENTORIES / SALES RATIO
CURRENT OUTPUT INVENTORIES / SALES RATIO
DESIRED OUTPUT INVENTORIES / SALES RATIO
```

The REAL▲DATA▲ENTRY function creates the REAL▲FIRM▲DATA variable if it does not exist. Otherwise, it adds new firm data into that variable. The REAL▲FIRM▲DATA variable is a q*f matrix where q is the

*) If you are in XENIX, change your directory to /moses and invoke the APL interpreter by the following commands.

```
# cd /moses RETURN
# apl MOSES.FUNEXP RETURN
```

number of data-elements (i.e., number of questions), and f is the number of firms. In other words, each column of this variable contains data of a specific firm.

To enter any firm whose data are saved in the REAL▲FIRM▲DATA variable, copy the REAL▲FIRM▲DATA variable to the dataset workspace to be changed, and type

REAL▲FIRM▲ENTRY *columnnumbers*

where *columnnumbers* refer to the REAL▲FIRM▲DATA variable. For example, if you type

REAL▲FIRM▲ENTRY 1 2 5

the first, second, and fifth firms of the REAL▲FIRM▲DATA variable will be added to the current dataset. The REAL▲FIRM▲ENTRY function, after preparing the dataset, prints the firm codes of these firms and asks if you want to compare these firms with their sector averages. If you enter these firms for the first time, compare them with sector averages to check for data consistency and typing errors.

1.6.3 Forming new firms by mergers

New firms can be formed by merging two existing firms during a simulation experiment. For this purpose, use the MERGE▲WITH function as follows.

firmcode1 MERGE▲WITH *firmcode2 conv eff period*

where *firmcode1* and *firmcode2* are firm codes of the firms to be merged, *conv* is the converge factor of the new firm's TEC variable to the TEC variable of the firm that has higher technological level (if *cons* is equal to 1, linear convergence, 2 quadratic converge, etc.), *eff* is the "efficiency" of converge in the TEC variable, and *period* is the time period of convergence in quarters. Unit variables are added, and fractional variables are weighted by appropriate weights to determine the new firm's variables. The TEC variable of the new firm initially is equal to the weighted average of both firms and gradually increases up to the level *eff**max(TEC_1, TEC_2) within *period* quarters by the convergence factor, *conv*.

This function should be used in the YEAR function *before* the line, YEARLY▴INIT.

1.7 ON-LINE REPORTING

During a simulation experiment, quarterly manufacturing output is displayed on the screen. This default option is allowed by the PREP▲INIT▲GRAPH function used in the RUNEXP function. Any other variable can be displayed during the simulation by modifying the command line for this function (see Section 1.4 for editing the model functions.)

There are two alternatives to this default on-line reporting. In the first case, some basic quarterly data for each manufacturing sector (production plans, actual production, exports, etc.) can be printed on the screen. When the version 1.0 is used, this option is turned on by adding a line TRACE1 to the modification function, MSTART*xx*, to execute the TRACE1 function, and to turn the graphics functions off (see Section 1.4). Use TRACE2 function to get the same on-line reporting for version 1.1. (For version 2.0, you should modify TRACE2 function appropriately.)

In the second case, an animated presentation of a firm's search for quarterly output and employment targets is shown for each quarter. You can also modify the firm's output and employment targets graphically.

This option is allowed by using TARG▲SEARCH▲FUNCTION instead of using TARG▲SEARCH (versions 1.0 and 1.1) or TARG▲SEARCH▲PC (version 2.0) functions in the PRODPLAN function. The ▲F variable contains the firm code whose target search is to be displayed. Define the ▲F variable in your modification function as follows.

▲F ← *'firmcode'*

If the ▲F variable is not defined, you will be asked to enter the firm code at the beginning of the simulation. (When you use this option, delete the line in RUNEXP function that invokes the PREP▲INIT▲GRAPH function.)

If this option is used, a chart will be displayed on the screen. The firm code, the time period for planning, the current profit target, the planned levels of output and labor, and the production function of the firm are shown on this chart. There will be explanations of the search procedure in the top-left corner of the chart. Press RETURN key or any one of the function keys to proceed. When the search procedure is complete, a pointing hand will be shown under the planned level of labor. If you do not want to change the planned levels, press the RETURN key to proceed. If you want to change them, use the cursor keys to move the pointing hand. (Press the F10 key to increase the cursor steps, and F9 key to decrease.) The level of output and labor at the tip of the pointing finger will be simultaneously shown on the screen. Press F1 key to change the planned level of output and labor, and press the RETURN key not to change them.

1.8 MODEL VERSIONS

This section explains the MOSES^{PC} versions and compares simulation results of each version. If you are not interested in technical differences between model versions, you may want to read only the last section, "Comparing model versions".

As of May 1991, there are three versions of the MOSES^{PC} model: 1.0, 1.1, and 2.0.

1.8.1 Versions 1.0 and 1.1

The PC Version 1.0 is identical to the model version which was transferred from the DEC-10 mainframe computer to a PC. (The mainframe version was then called 7.2). This version includes modifications made by PERMANENT∆CHANGES and MOSES∆VARIANTS functions (MC, 285, 286[*]). Although the PRICE∆CHANGES in the MOSES∆VARIANTS function seems to be turned on in the Moses Code (MC, 286), it was off in the program.

This version has been used for most of the experiments on the mainframe computers (the PRIME computer at the Stockholm School of Economics, DEC-10 in QZ Center, Stockholm, and DEC-20 in Bergen, Norway). Since the APL interpreters installed in these computers use different initial seed numbers for the pseudo-random number generators, and, presumably, different formulas, the results of the PC computer may differ somewhat from those of the mainframe computers. However, the PRIME computer and the PC have identical pseudo-random number generators, and their results are exactly the same.

The PC Version 1.1 is similar to the Version 1.0 but it also includes the changes made by the PRICE∆CHANGES function in the

[*] References are either to "The MOSES Technical Specification Code" or "The MOSES APL Program Code" in Albrecht *et al.* (*MOSES Code*, Stockholm: IUI 1989), pp.147-220 and 245-354, respectively.

MOSES∆VARIANTS function. Note that since PERMANENT∆CHANGES and MOSES∆VARIANTS functions are explained in detail in the IUI Research Report No. 36, *MOSES Code*, 1989, the modifications of these functions were incorporated into the PC model to reduce simulation time. Thus, the program does not execute these functions at the beginning of each simulation.[*]

1.8.2 Version 2.0

Using the model, a number of "bugs" have been found in this specification. Version 2.0 rectifies those problems. The most important changes in version 2.0 involve the price mechanism, credit market, labor market, and entry/exit functions.

The modifications in the Version 2.0 are as follows.

1.8.2.1 Changes in price specifications

Price adjustment specifications are considerably changed in Version 2.0. During the autumn of 1986 the domestic price mechanism in MOSES was reprogrammed to appropriately account for the price and size of imports (MC, 228). However, some of the specifications related to prices were not modified in these changes. Before explaining new changes, basic price variables need to be explained.

QPFOR is foreign prices (before trade margins). QPDOM is the consumer price of goods supplied by domestic producers. It includes the value added tax (VAT). PT is the average price of domestic goods. In other words,

$$\text{PT} = (\text{QPDOM}*(1\text{-IMP})) + ((\text{QPFOR}*\text{IMP})/(1\text{-TXVA2})) ,$$

where IMP is the import share, and TXVA2 is the VAT including trade margins. (The VAT for producers, TXVA1, is set to zero.)

A firms's average price is the weighted average of its domestic and foreign sales.

[*] There is also a number of minor differences between the PC and mainframe versions, especially in the transcription functions. For these differences, see Appendix E.

QP = (QPDOM[MARKET]*(1-TXVA2)* (1-X))+(X*QPFOR[MARKET])

Now, the changes can be summarized as follows.

1) As specified in Line 238 of the VERSION20 function[*], the new FINALQPQSQM▴PC function uses average domestic prices, PT, for input goods in the calculation of net sales and value added (see Lines 8 and 9 of the FINALQPQSQM▴PC function). The previous function, FINALQPQSQM (MC, 330), uses domestic prices of domestic producers, QPDOM.

2) The new QUARTERLY▴EXP▴PC function uses average domestic prices in the calculation of expected input prices (see Lines 14-16 of the QUARTERLY▴EXP▴PC function). The previous function, QUARTERLY▴EXP (MC, 312), uses QPDOM.

3) The new DOMESTIC▴MARKET▴PC function uses a number of new ...PC functions instead of ...NEWP functions.

4) The MARKET▴ENTRANCE▴PC function limits the difference between preliminary prices, QPRELPDOM, and last quarter's prices.

5) The MARKET▴CONFRONT▴PC function changes government spending specification. In the previous specification of the MARKET▴CONFRONT▴NEWP function (MC, 324), government spending is kept equal to government wage payments. In this case, the share of government expenditures in current GNP declines sharply. In the new specification, government spending *plus* government wage payments increase at the quarterly rate of DPURCHG, corrected by the ratio (tax income - deficit) / tax income.

6) The ADJUST▴PRICES▴PC function makes adjustments to domestic prices as a function of excess demand. The previous ADJUST▴PRICES▴NEWP function (MC, 326) makes adjustments to domestic prices a fixed amount if there is an excess demand (or supply).

7) The COMPUTE▴IMPORTS▴PC function calculates the share of imports, IMP. The previous COMPUTE▴IMPORTS▴NEWP function does not

[*] For the code, see Appendix F. Unless otherwise specified, all line numbers refer to the function VERSION20.

change the IMP variable, and it remains as a "desired" share of imports. This causes problems in the calculation of domestic prices since the IMP variable is used as the weight for foreign prices.

8) The DOMESTIC∆RESULT∆PC function changes the QDPDOM equation. In the DOMESTIC∆RESULT∆NEWP function, this equation computes erroneous domestic prices. (In that formulation, the average prices could be higher or lower than its components.)

9) Line 243 of the VERSION20 function defines the AGGPROF variable which is used in the determination of interest rate.

10) Line 244 values input goods inventories at the average domestic prices. The previous function uses domestic producers' domestic prices (MC, 331).

11) Line 245 changes QVATAX∆IMP definition so that customs duties ("the value added tax" for foreign goods) are included in government income.

12) Line 246 values the input goods purchases of the external sectors at average domestic prices. The previous function uses only domestic producers' domestic prices (MC, 328).

13) Line 247 uses average domestic prices in the calculation of the rate of change in capital goods prices. The previous function uses domestic producers' domestic prices for this purpose (MC, 331).

1.8.2.2 Credit market changes

There are some significant modifications made in Version 2.0 regarding the determination of interest rates. Lines 110-150 of the VERSION20 function modify the code and set new parameters used in the new specifications. In the previous versions (MC, 188), the change in the interest rate was determined as QCHRI = LAMDA1 * (QDEMFUND-QSUPFUND)/QSUPFUND where LAMDA1 is a constant. This specification is problematic on two counts. First, the change in interest rate depends only on the level of current excess demand. Second, the interest rate is the only variable that is used to adjust demand and supply of funds. The major changes in interest rate specification in the CREDIT▴MARKET▴PC function are as follows.

1) The "desired" change in the interest rate is determined by the excess demand for funds, and the level of net debt to banks. This new specification takes into consideration the fact that an increase in the rate of interest will also increase the supply of funds. Then the actual change in the interest rate is found by a smoothing function that puts a limit on the quarterly change in the rate of interest.

2) The difference between lending and borrowing rates (IDIFF) is endogenous in the new specification. It increases when there is excess demand for funds.

3) The coefficients that determine the credit supply as a the proportion of the bank's liquid assets (RFUND1 and RFUND2) are also endogenous in the new specification. If, for example, there is excess supply of funds in the last three quarters *and* the average rate of return in the industry is somewhat higher than the current rate of interest, then the bank tightens the credit supply by increasing RFUND1 and/or RFUND2.

Government spending and monetary policy are found to be critical for the interest rate determination. Therefore, a number of changes have also been made in government accounting by using a new function, GOVERNMENT▴ACCOUNTING▴PC. In the previous specification, GOVERNMENT▴ACCOUNTING, (MC, 332) the change in the government's position in the bank is defined as income + (new) money supply - spending. New money supply then is equal to the money stock (total deposits in the bank including government's deposits) times the quarterly change in current GNP. In other words, it is intended to make new money supply proportional to current GNP. However, the money stock also grows endogenously by households' and firms' deposits and the newly issued money makes the money stock grow exponentially.

In the new government accounting, the rules that govern money supply are rather complicated. First, the "estimated" money supply is found as reserve money times quarterly growth in current GNP. Then, if government income is higher than spending, i.e., if there is a surplus, the money supply is reduced by the amount of surplus, because the government surplus has the same effect as money supply. The "estimated" change in the government position in the bank is found by using this level of money supply. There are limits to the share of government's borrowing from the bank, and the share of government in total deposits. Within these limits, the final level of change in the government's position in the bank is determined. The actual money supply is then determined by the difference between the change in the government's position in the bank and its surplus. This specification, hence, takes into consideration funds available in the bank. If the government has a high share in total debts, it tends to print money. Otherwise, it tends to use bank resources instead of printing more money.

1.8.2.3 Labor market changes

The following changes have been made in the labor market functions. Line 72 of the VERSION20 function changes the required change in the labor force specification in the LABOR∆SEARCH∆INPUT function (MC, 321). In the previous specification, a firm that experienced a negative change in its profit margin determines its labor force requirement according to the optimum use of its labor force. This, however, implies an inconsistency between the firms' production planning and its labor search procedures. This special treatment of firms with declining profit margins has been eliminated in the new specification.

Line 76 changes the firms' probability to be attacked in the labor market. In the previous specification (MC, 321), the probability to be attacked depended only on the relative size of each firm. In the new specification, it also depends on firms' relative wage rate. Those firms that pay lower wages are more likely to be attacked in the labor market.

Lines 167-174 put a limit on maximum changes in the wage rate. It is determined by the expected net prices and profit margin targets.

20. Lines 210-216 allow workers to quit their jobs depending on their current wage rate, and the unemployment rate (see the QUITS function).

1.8.2.4 Changes in entry/exit functions

Lines 84-91 of the VERSION20 function add new specifications for nullified firms. In the previous versions, nullified firms simply "evaporate" after leaving their labor unemployed. In this version, the capital equipment, and input and output inventories of nullified firms are sold in the markets, debts to the bank are paid, and remaining income, if positive, is paid to the households.

Lines 92-108 add some specifications for new firms. In the previous versions, new firms come from nowhere. Nobody pays for their initial capital equipment and inventories. In the new version, they raise their capital from households. They buy new capital equipment and input goods inventories in the first quarter, and start production in the second quarter. At the first quarter, they do not carry any debt from banks.

In accordance with this new specification, the borrowing specification in the INVFIN function has also been modified. In the previous case (MC, 182, 331), a firm with zero borrowing never wanted to borrow! Moreover, the money allocated for new investments was a residual fund from the total quarterly cash flow. In the new specification, firms determine their cash requirements (for new investments, current assets, etc.), and their desired borrowing is the difference between net cash flows and requirements.

1.8.2.5 Other changes

1) Line 9 specifies the number of real firms in the model by the variable N∆REAL∆FIRMS. This variable is used in the AVG∆TOP function.

2) A new table (YEARLY∆EXPECTATIONS) that stores information about firms' expectations is created in Line 13 of the VERSION20 function (see function code below). There are 12 columns in the

YEARLY▴EXPECTATIONS variable. The first column of this variable contains the (weighted) average of the expected growth in sales, the second column the (weighted) average of the actual growth in sales, the third column the percentage of firms that overestimated sales growth, and the fourth column the percentage of firms that underestimated sales growth. Column 5-8 and 9-12 similarly contains information on expected changes in prices and wages, respectively.

3) Lines 19-24 prepare new exogenous foreign prices for the 1982 datasets. Lines 28-41 change some parameters that seem to produce better long-term results. These parameters and foreign prices were used in MSTART functions in most of the previous experiments.

4) Line 48 modifies the SOLVE function (MC, 316) so that the planned level of employment will be equal to, or larger than 0.1. This was used in most of the previous experiments in MSTART functions. This ensures a positive planned level of employment. (Even though we have had no problem with negative employment levels for any firm in a large number of 20 years simulations we have chosen to introduce this modification.)

5) Lines 52-56 set the MTEC values for each sector to 10% higher than the maximum TEC level in that sector for the 1976 and 1982 datasets. These changes ensure that the best available technology is better than all currently employed technologies, as implied by the definitions of these variables. However, in the initialization process of both the 1976 and the 1982 datasets, the MTEC variable was calculated as the weighted average of the TEC values (see the SECONDARY DATA function, MC, 275). Note that the technological level (labor productivity) of new investments is equal to the MTEC value (MC, 153). Thus, when the MTEC values are equal to the industry average, the technological level of better firms declines when they invest in versions 1.0 and 1.1.

6) Line 61 changes the specification of the BAD variable. In the previous specification (MC, 288), a firm is nullified when it has negative

net worth in any six quarters during a simulation experiment. The new specification nullifies those firms that have negative net worth in any six quarters in a row.

7) Lines 65-68 introduce a new target search function (TARG▴SEARCH▴PC) instead of the TARG▴SEARCH function, and set new parameters used by this function. There are three major differences between these functions. First, the new function turns on the SOLVE▴MONEY function that ascertains that no step in SEARCH leads to less expected profits in money terms than in the position before (MC, 159). Second, the old function nullifies those firms that cannot satisfy profit margin targets (MC, 315, line 67). However, the new function enables those firms to revise their profit margins. If their expected profit margins are higher than the revised margin which is equal to (LOW▴TARGET * targeted profit margin), then they are not nullified. Third, those firms that are on the production frontier, and that are capable of producing more at higher monetary profits, reduce their profit margin targets.

There is also another function, TARG▴SEARCH▴SHOW, which is similar to the TARG▴SEARCH▴PC function. The only difference between these functions is that the TARG▴SEARCH▴SHOW function shows on the screen the planning procedure of a firm with explanatory notes. In other words, it shows graphically the planning process of a firm as specified by the model in TARG▴SEARCH▴PC function. The user is recommended to use this option to understand the model's planning procedures. You can also change manually the employment and output plans of that firm. (For details, see Section 1.7.)

8) Line 76 ensures that money spent for new investments is non-negative. (Probably as a result of a typing error, a firm had a negative value in the 1982 dataset.)

9) Line 82 changes the optimum sale units in a quarter (QOPTSU) in the PLANQREVISE function. The previous specification (MC, 166) has two problems: First, it uses the TMIMSTO parameter for the adjustment of

finished goods inventories instead of the correct one, the TMSTO variable. (TMIMSTO is the adjustment parameter for input goods inventories. See MC, 165.) Second, the QOPTSU may become higher than available output goods, QQ+STO. The new specification simply divides available output goods into two groups, optimum sales (QOPTSU), and output stocks depending on the level of expected sales, QEXPSU, and optimum stock level adjusted by the TMSTO parameter.

10) Lines 152-165 make the *exchange rate endogenous*. It is determined by the net export ratio. The value of foreign assets and liabilities are modified in accordance with exchange rate changes. Exchange rate changes also affect foreign prices via the parameter XRPEFF, which should be less than one.

11) Lines 177-186 add additional specifications for sales, wage rate, and price expectations. Sales expectations are adjusted according to new investment, and exchange rate changes. Wage rate expectations are adjusted according to new labor requirements and the level of unemployment. The expected change in any variable is constrained to be higher than -50% and lower than 100%.

12) Lines 188-192 change profit margin targeting. The LOW▴TARGET parameter is introduced and set to 0 (see parameter definitions below). The YEARLY▴TARG and TARGM functions are changed so that profit margin targets are to be non-negative.

13) Lines 194-208 change the production frontier specifications. In the previous specification (MC 153, 313), any new investment has three components: one goes to increase productive capacity (QCHQTOP1), one increases the "slack" (QCHQTOP2), and the last part is "lost". The lost part of the investment is determined by the LOSS and RESMAX parameters, and the initial level of slack, RES. Therefore, for example, an average firm with initial slack of 10%, wastes 5% of its new investment capacity. 81% of the new investment goes to productive capacity, and 14% to the slack variable (under current values of LOSS and RESMAX parameters). Thus, with the

new investment, the level of slack increases for all firms, and investment efficiency declines. The new specification eliminates the lost component. Moreover, the INVEFF variable is redefined in physical terms, and is allowed to increase quarterly by the factor QDINVEFF.

14) Lines 217-229 introduce a new function, OBSOLETE K. This new function makes part of capital equipment obsolete. Firms may eliminate the portion of their capital equipment that is not likely to be used in the future. However, the obsolescence of capital equipment does not affect the slope of the production function (see the OBSOLETE K function).

15) Line 233 changes the household spending function so that households' spending is affected by the real rate of interest. (The effect, however, is rather small. The ALFA3 parameter determines its impact.)

16) Line 259 changes the BETA2 parameter to be compatible with the I-O table.

17) Lines 262-266 introduce a limit to the quarterly real growth rate.

18) Lines 269-274 redefine the growth of public employment in relative terms and set a new variable for the exogenous growth of public employment.

19) Lines 276-285 define the investment composition parameters (OMEGAs) in real terms.

1.8.3 New parameters and variables introduced in Version 2.0

CHRFUND1 Change in the RFUND1 variable. Used in the CREDIT MARKET PC function.

CHRFUND2 Change in the RFUND1 variable. Used in the CREDIT MARKET PC function.

DPURCHG Quarterly increase in government expenditures (spending plus wage payments). Used in the MARKET CONFRONT PC function.

ENTRY SPECS Specifications for new firms. This is a three-element vector. The first element defines average size of the new firms, and the second and third elements define the relative efficiency of new firms in TEC and RES variables. (For example, if the second element of the ENTRY▵SPECS parameter is .05, then new firms' TEC value will be 5% higher than the average.) Used in the ENTER NEWFIRMS function.

GBRWRAT Maximum share of government in total debt. Used in the GOVERNMENT ACCOUNTING PC function.

GDEPRAT Maximum share of government in total deposits. Used in the GOVERNMENT ACCOUNTING PC function.

INTDIFF The interest rate differential between lending and borrowing. Used in the CREDIT MARKET PC function.

IOTALOW Reduction in the initial wage offer. Used in the LABOR SEARCH INPUT function.

LOW TARGET The amount of decline in profit margin target if no satisfactory plan is found. Used in the TARG SEARCH PC and TARG SEARCH SHOW functions.

LOWER MHIST The amount of decline in the firm's profit margin history. Used in the TARG SEARCH PC and TARG SEARCH SHOW functions.

MAXCHIDIFF Maximum change in the INTDIFF variable. Used in the CREDIT MARKET PC function.

MAXEXPDP Maximum annual change in expected prices. Used in the MDIFY▵EXP function.

MAXEXPDS Maximum annual change in expected sales. Used in the MDIFY▵EXP function.

MAXEXPDW Maximum annual change in expected wages. Used in the MDIFY▵EXP function.

MAXIDIFF Maximum level of the INTDIFF variable. Used in the CREDIT MARKET PC function.

MAXQCXRATE Maximum quarterly change in the exchange rate. Used in the BANK UPDATE function.

MAXQDSUFOR Maximum quarterly increase in volume of exports. Used in the EXPORT▲MARKETS function.

MAXRFUND1 Maximum level of change the RFUND1 variable. Used in the CREDIT MARKET PC function.

MAXRFUND2 Maximum level of change the RFUND2 variable. Used in the CREDIT MARKET PC function.

MAXWCOEFF Maximum wage change coefficient. Used in the TARG SEARCH PC and TARG SEARCH SHOW functions.

MINEXPDP Minimum annual change in expected prices. Used in the MDIFY▲EXP function.

MINEXPDS Minimum annual change in expected sales. Used in the MDIFY▲EXP function.

MINEXPDW Minimum annual change in expected wages. Used in the MDIFY▲EXP function.

MINIDIFF Minimum level of the INTDIFF variable. Used in the CREDIT MARKET PC function.

MINRFUND1 Minimum level of change in the RFUND1 variable. Used in the CREDIT MARKET PC function.

MINRFUND2 Minimum level of change in the RFUND2 variable. Used in the CREDIT MARKET PC function.

N▲REAL▲FIRMS Number of real firms. Used in the AVG▲TOP function.

NEWFUND The share of household savings that can be used to finance the formation of new firms. Used in the START ENTRY PC function.

NORMRU Reference rate of unemployment. Used in the QUIT function.

OBSRATE The coefficient for obsolescence. Used in the OBSOLETE K function.

PARAMETER∆NAMES A vector of parameter names.

QDINVEFF A four-element vector. Each element defines the quarterly change in investment efficiency in each sector. Used in the PRODFRONT function.

QINVBLDREAL Quarterly volume of investment in building and construction.

QINVGREAL Quarterly volume of public investment.

QINVINREAL Quarterly volume of investment in non-manufacturing sectors.

QSUFOROLD Exports volume of each sector in the last quarter. Used in the EXPORTS∆MARKET function.

QUITCOEFF Quit coefficient. Used in the QUIT function.

RWEXP Weighing coefficient for internal and external wage expectations. Used in the YEARLY∆EXP function.

VARIABLE∆NAMES A vector of model variables' names.

XRATECOEFF Exchange rate coefficient. Used in the BANK UPDATE function.

XRPEFF Exchange rate price effect coefficient. Used in the EXPORT MARKETS function.

1.8.4 Comparing model versions

The results of all versions are shown in Figures 1.1-1.7. Version 2.1 refers to version 2.0 with calibrated parameters. The same exogenous foreign prices were used in all experiments. The MSTART19 function was used for versions 1.0 and 1.1, and the MSTART900 function (that contains calibrated parameters) for version 2.1.

Figure 1.1 shows the level of GDP. GDP did not grow at the first ten years of the simulation in version 1.1, although it started to accelerate in the early 1990s. Other versions exhibit similar growth characteristics until the mid 1990s. Then the model economy stopped growing in version 1.0. The second version exhibited relatively smooth growth rates (around

3.7% per year).

Manufacturing labor productivity is shown in Figure 1.2. Versions 2 exhibited considerably higher labor productivity, especially in the second half of the simulation. This is partially a result of the specification of the MTEC variable. As shown in Figure 1.3, the labor productivity factor (the TEC variable) was much higher in versions 2 than versions 1. Versions 1.0 and 1.1 have a very small increase in the TEC variable due to the erroneous initial level of the MTEC variable. Recall that in versions 2, *all* firms increase their labor productivity factor by investment since the best practice technology embodied in investment has a higher technology factor. In versions 1, however, the labor productivity factor of the best practice technology (the MTEC variable in the model) is equal to sector averages for the initial year.

Figure 1.4 shows the "unemployment rates." All versions had a 9-10% "unemployment rate" at the end of simulations. Although these figures seem to be on the high side for Sweden, note that the "unemployment rate" in the model is calculated on the basis of hours worked. This means that decreases in working hours and increases in part-time employment are also reflected as an increase in the "unemployment rate."

The government share in current GNP is shown in Figure 1.5. It increases in version 2.1 in the 1980s and remains almost constant aterwards. Version 2.0 has a fluctuating government share around 34%. Versions 1.0 and 1.1 have almost symmetric changes. In version 1.1 it increases until 1992, then declines. The pattern is the opposite in version 1.0 where results are quite contrary to our expectations (a 5% points decline in five years in Sweden!). Note that the rate of increase in government spending is determined by the coefficient DPURCHG in versions 2.0 and 2.1. Therefore, the share of government in GNP can actually be controlled in those versions.

Figure 1.6 shows the changes in the velocity of money (current GDP

/ deposits in the bank). In versions 2, it fluctuates around 1.15 while the amplitude of fluctuations is higher for version 2.0. In versions 1, it declines almost continuously from 1.35 to .95. This difference between versions 1 and 2 is due to changes in the GOVERNMENT▲-ACCOUNTING function (see Section 1.8.2.2).

The rate of interest is shown in Figure 1.7. It was exogenously determined in versions 1.0 and 1.1. (If endogenously determined in this version, it would rapidly increase or decrease until it reaches the limits, to stay there.) Although the rate of interest increases in version 2.1 in the 1990s, it declines to 9-10% in the 2000s (not shown in this figure).

Figure 1.1 GDP level (1982 prices)

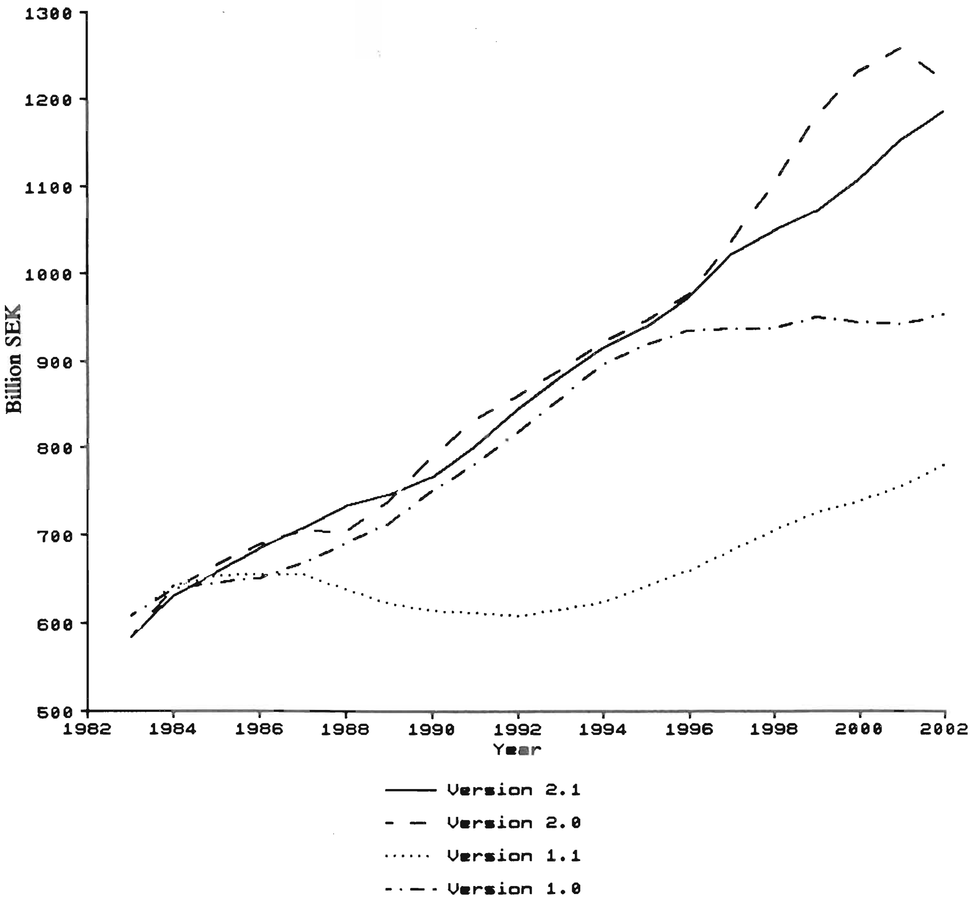


Figure 1.2 Labor productivity in manufacturing

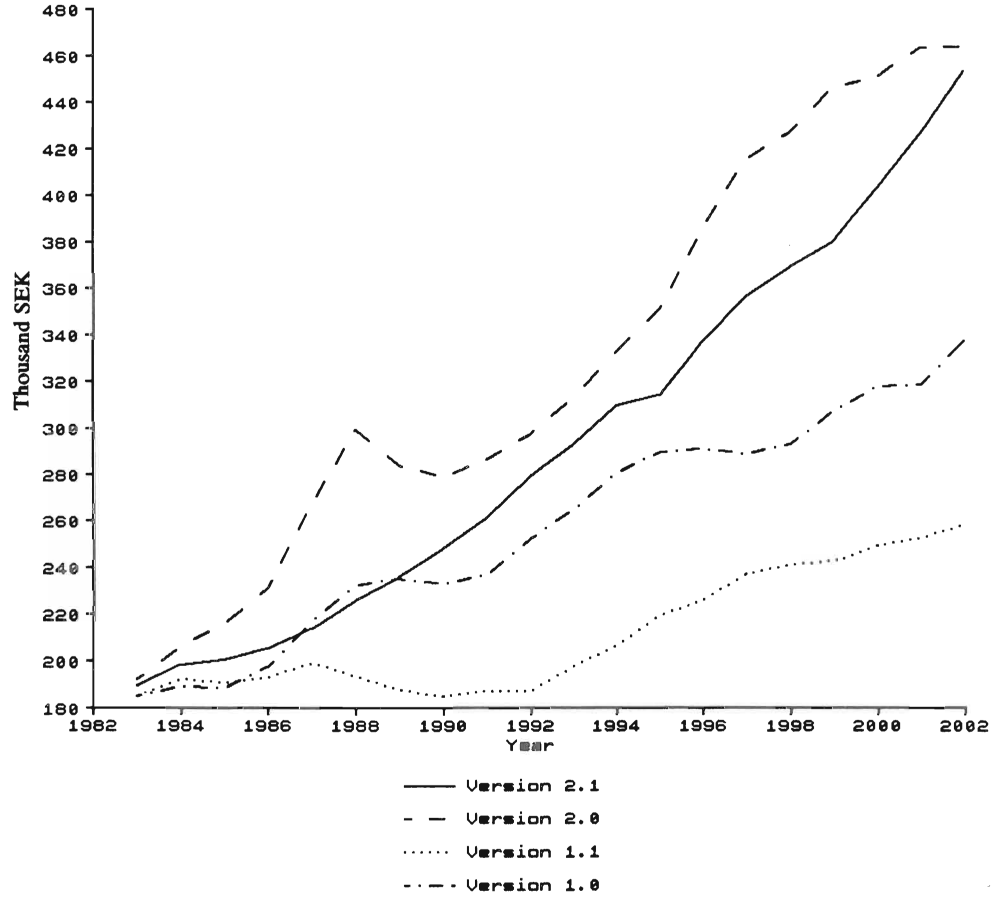


Figure 1.3 Average labor productivity factor (The average of the TEC variable)

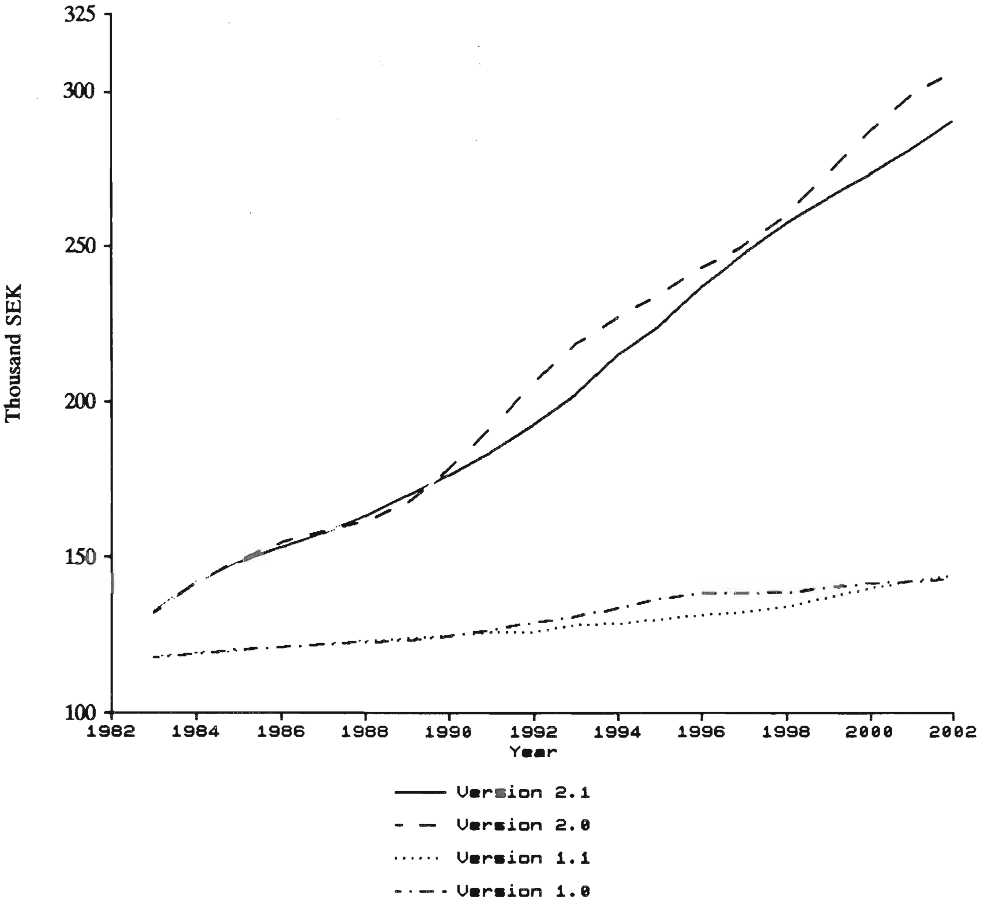


Figure 1.4 Unemployment rate

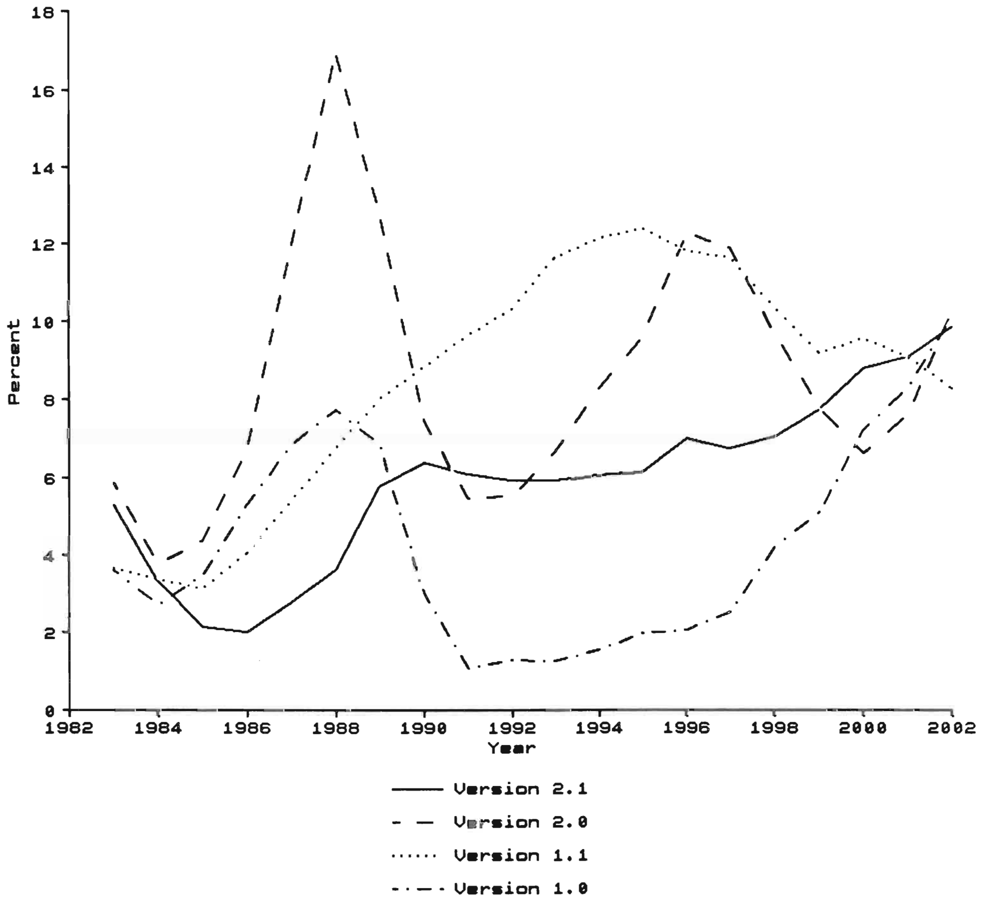


Figure 1.5 Government's share in GDP
(Based on current prices)

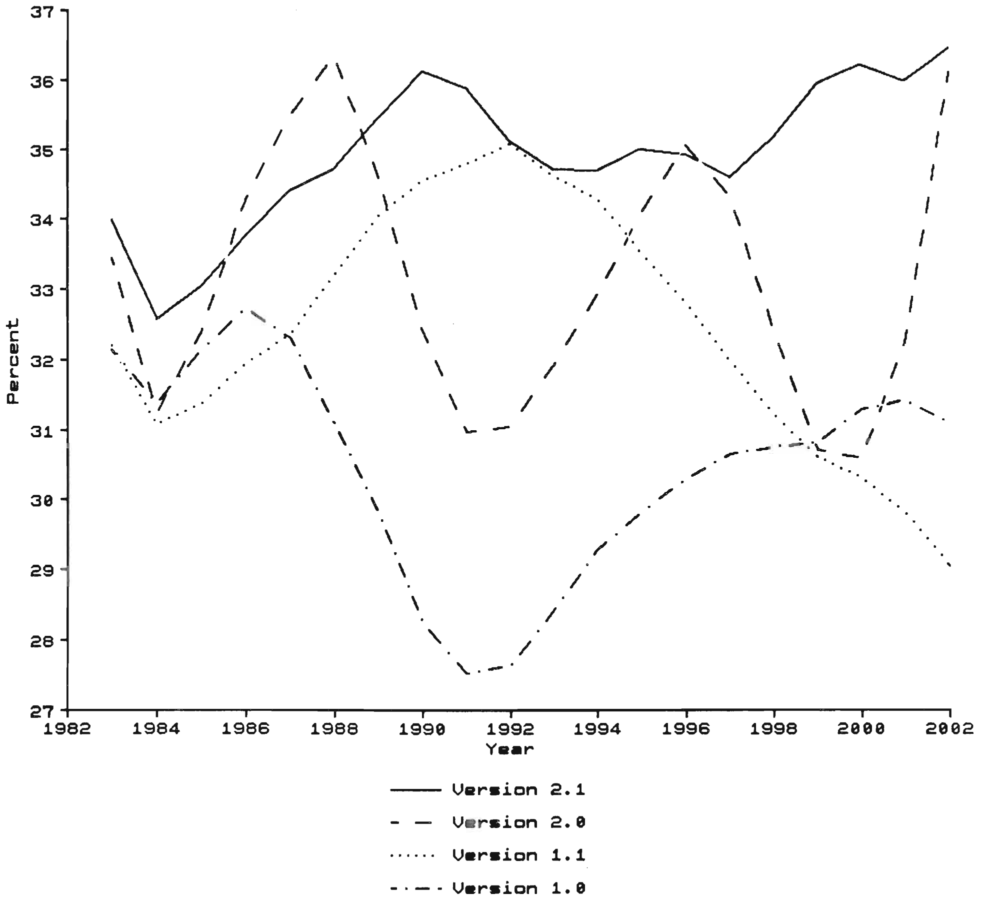


Figure 1.6 Velocity of money

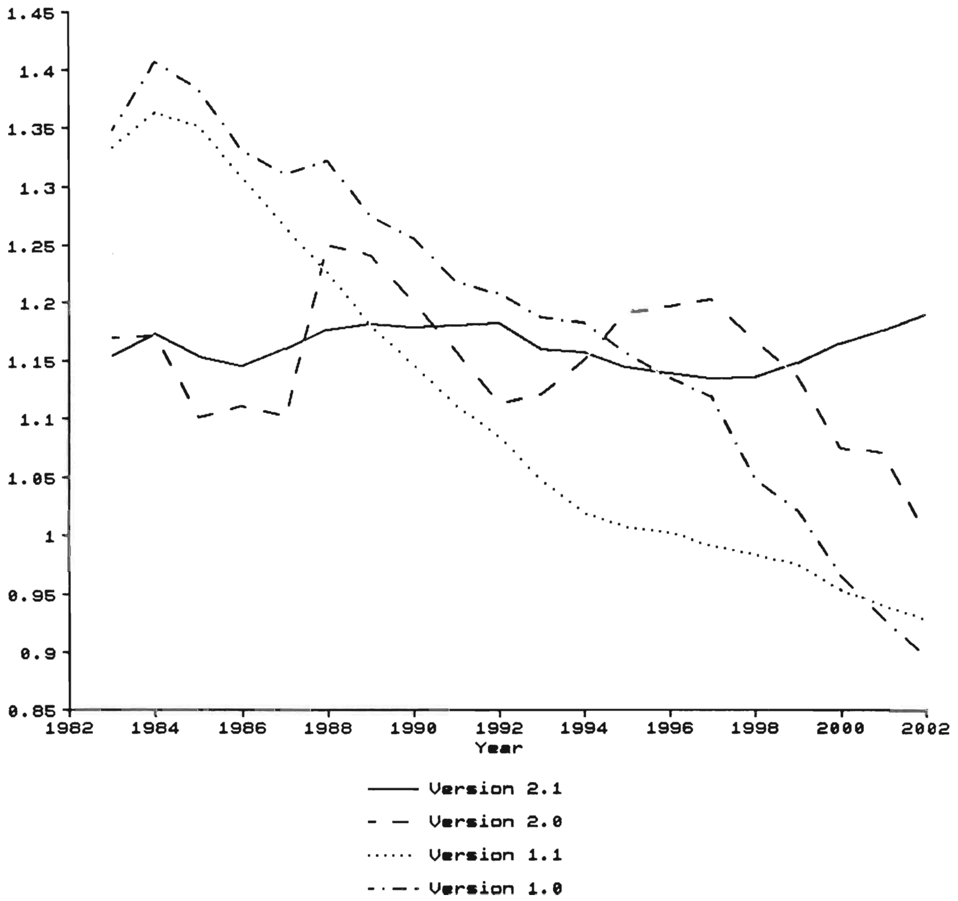
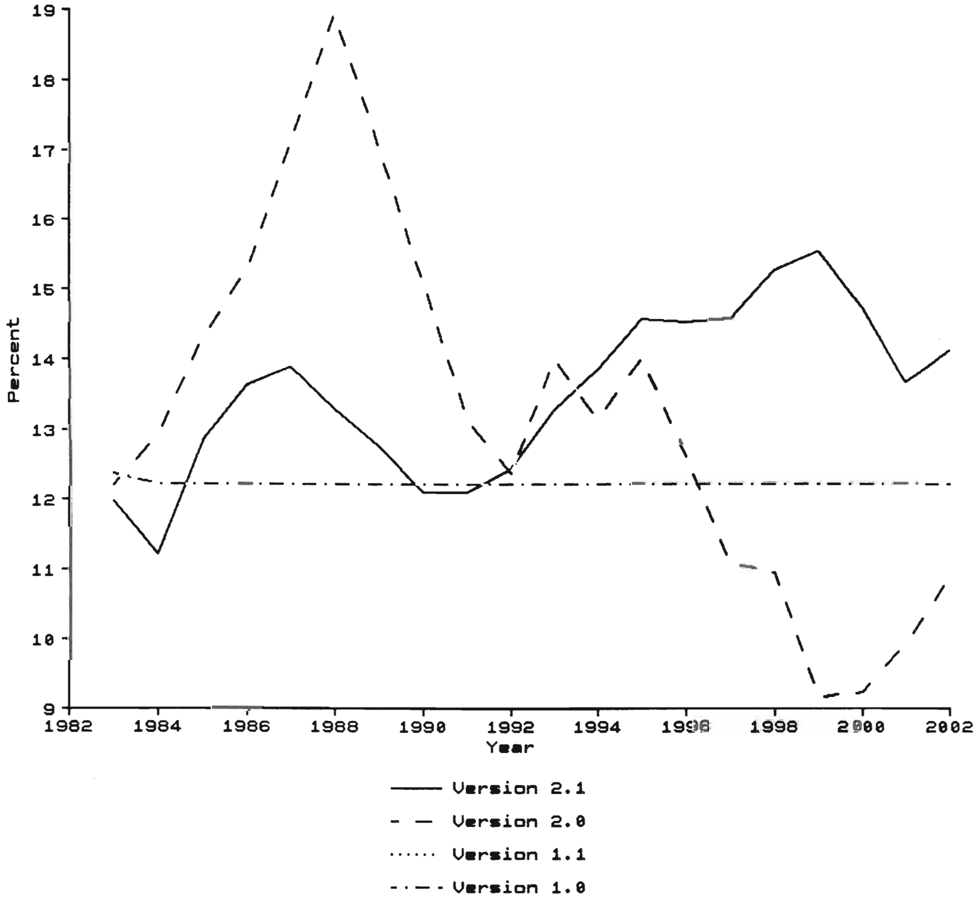


Figure 1.7 Rates of interest



SECTION 2

INITIALIZATION

2.1 INITIALIZATION PROCEDURE

The micro-to-macro simulation model MOSES simulates the economy quarter by quarter from a given starting year. Before one can start a simulation of the model economy, one has to initialize a large number of variables.

"Initialization" means that three kinds of variables are prepared for the model.

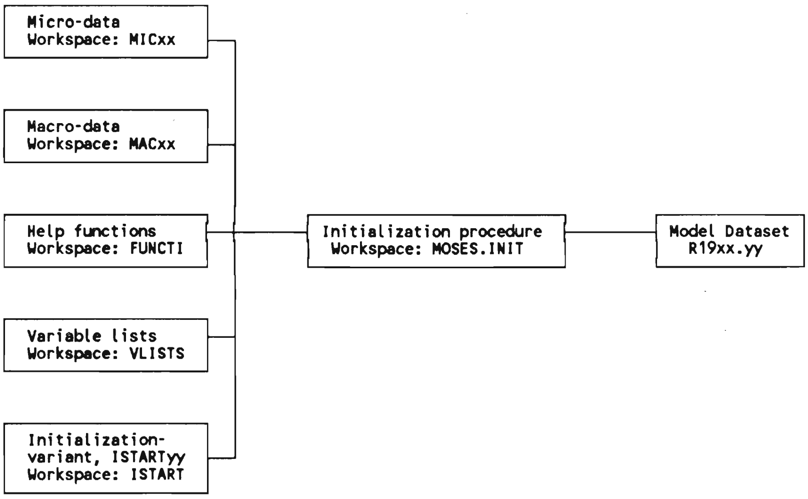
1) Micro and macro variables needed to start up the model (initial endogenous variables).

2) Exogenous variables for the simulation period.

3) Certain constants, some of which are parameters affecting the behavior of firms, households and market mechanisms.

The initialization program, MOSES.INIT, converts raw micro and macro data to variables that are used in the model. A schematic overview of the initialization procedure is shown in Figure 2.1.

Figure 2.1 The initialization procedure



The raw macro and micro data are saved in the MAC*xx* and MIC*xx* workspaces, respectively, where *xx* is the base year. There are currently two sets of data: one for 1976 (MAC76 and MIC76) and the other for 1982 (MAC82 and MIC82).The initialization program, MOSES.INIT, retrieves these workspaces, and converts them into a form that can be used by the MOSES model by using the help functions in the FUNCTI workspace and initialization modification function ISTART*yy* in the ISTART workspace. The OUTPUT∆OPERATIONS function of the MOSES.INIT workspace deletes all functions and variables that are not specified in any one of the variable name lists of the VLISTS workspace. The name of the final model dataset will be R19*xx*.*yy*, where *xx* is the dataset year, and *yy* is the initialization modification number.

If one wants to generate a dataset for other years or other countries, the MAC*xx* and MIC*xx* raw datasets should be prepared. These workspaces are explained in detail in Part 2 of *MOSES Handbook*(*), (Bergholm 1989: 45-213), and Chapter II of *MOSES Code* (Albrecht *et al.* 1989: 67-132). The user is referred to these publications. *MOSES Handbook* includes a complete initialization code and the list of macro and micro variables that are used in the MAC*xx* and MIC*xx* workspaces. Note, however, that there are some minor changes made for the PC version of the model as explained in Appendix E.

*) The first part of the handbook, "How to Run the MOSES Model", is replaced by this manual for the PC version of the model.

2.2 USING THE MOSES.INIT WORKSPACE

To use the initialization program, load the MOSES.INIT workspace when you are in the Xenix operating system by typing

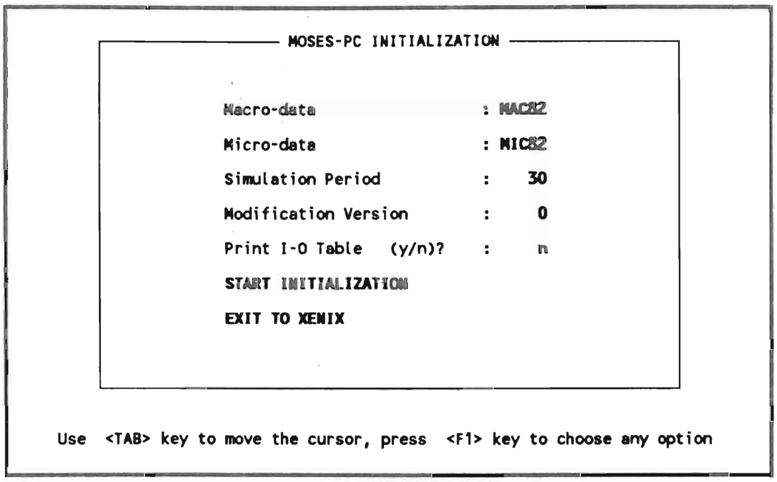
# apl MOSES.INIT RETURN

or, if you are in APL already, by typing

) LOAD MOSES.INIT RETURN

where ")" is the right parenthesis entered by typing **Shift-"** when the keyboard is in the APL mode.

Then, the following menu will appear on the screen.



```
MOSES-PC INITIALIZATION

Macro-data                   : MAC82
Micro-data                   : MIC82
Simulation Period            :    30
Modification Version         :     0
Print I-O Table   (y/n)?     :     n
START INITIALIZATION
EXIT TO XENIX

Use <TAB> key to move the cursor, press <F1> key to choose any option
```

The cursor can be moved by pressing TAB (downward) or Shift-TAB (upward) keys. After changing default values on the menu as desired, the simulation is started by pressing F1 key when the cursor (the highlighted rectangular) is on "START INITIALIZATION".

"Macro-data" and "Micro-data" define the macro and micro-data workspaces. If you do not enter any name here, the default workspaces, MAC82 and MIC82 will be used. (The micro-datasets, MIC76 and MIC82, contain confidential firm data. Therefore, access to these workspaces is restricted to authorized users, and these workspaces are not supplied with the MOSES model.)

"Simulation Period" is the number of years that the simulation will proceed. Note that although the MOSES^{PC} model is actually a quarterly model, the simulation period should be specified in years. The initialization program prepares exogenous variables for the number of years as specified by this option.

"Modification Version" determines which ISTART function to be used to modify the initialization code. In the MOSES.INIT program, each initialization is carried out by making changes in the original code by using one of the modification functions in the ISTART workspace. This allows you to keep the program in its original form. The ISTART*xx* functions contain modifications for specific initialization variants. ISTART*xx* functions are executed at the beginning of an initialization. If you want the initialization code without any modifications, enter zero for this variable.

"Print I-O Table (y/n)?" question allows the user to print out the I-O table of the final dataset, on the screen during the initialization procedure.

Pressing F1 key when the cursor is on the "Exit to Xenix" allows the user to exit from the model. To continue to use the model, the user should start by typing "apl MOSES.INIT" at the Xenix prompt.

The initialization takes at most a couple of minutes depending on the number of firms. During the initialization process, some variables are printed out on the screen. If you enter "y" for "Print I-O Table (y/n)?", then the I-O table of the dataset prepared by the initialization program is also printed out on the screen. Note, however, that when the I-O table is shown on the screen, the execution of the program is temporarily suspended. You have to press RETURN key to proceed.

At the end of the initialization program, the following message is shown.

```
******************************************
INITIALIZATION COMPLETED
WORKSPACE NAME IS R19xx.yy
PLEASE DO NOT FORGET TO SAVE THE WORKSPACE
******************************************
```

As stated in this message, the workspace is *not* automatically saved at the end of initialization process, and you have to save if for further use by typing

) **SAVE** RETURN

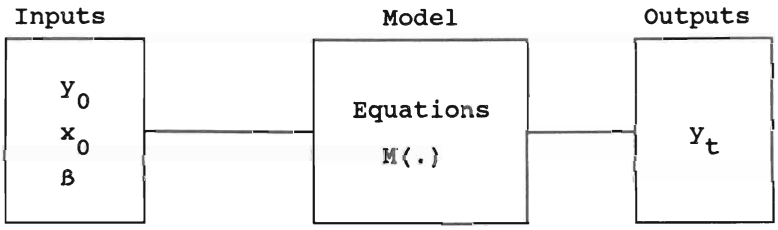
SECTION 3

CALIBRATION

3.1 ANALYZING MODEL'S STRUCTURE Calibration, sensitivity, linearity

The MOSES model consists of a large number of non-linear equations that, given a set of parameters, convert a set of exogenous variables and the initial values of endogenous variables into a future series of endogenous variables (see Figure 3.1).

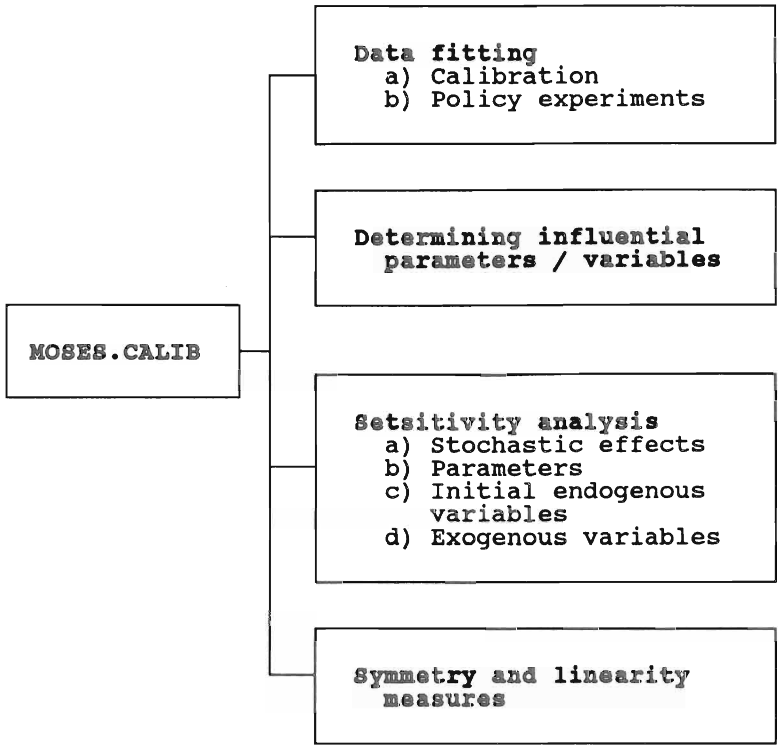
Figure 3.1 The simulation model



Notation: y_0: endogenous variables vector at the base year
x_0: exogenous variables vector
ß: parameter vector
y_t: endogenous variables at time t
M(.): a set of model equations

The model, i.e., the set of equations M(.), is so complex that even those who have the complete model code cannot foresee the impact of changes in initial conditions, {y_0, x_0, ß}, on output, {y_t}, without simulating the model. The complexity of the model may also make understanding of the dynamics of the model difficult, as well as its representation of economic relations.

Figure 3.2 MOSES.CALIB tools



The MOSES.CALIB workspace contains various functions for the analysis of the model's structure. Figure 3.2 summarizes the tools that are available in the MOSES.CALIB workspace.

The major use of the MOSES.CALIB workspace is for *data fitting*, i.e., to determine the values of a set of parameters(*) that generates the best fit for a series of endogenous variables. In the calibration case, the problem is to determine the values of parameters that generate the best fit for a set of observed endogenous variables for a certain time period. In

*) There is computationally no difference between the parameters of the model and the exogenous variables. Therefore, hereafter, we use the term "parameter" in a broad sense including all variables whose values are constant during a simulation experiment.

the policy experiment case, the problem is to determine the values of some exogenous (policy) variables (for example, the quarterly increase in government spending), that come as close to a certain target variable (for example, zero unemployment rate) as you specify, provided it is feasible. Note that there is no technical difference between the calibration procedure and making policy experiments. The calibration procedure is explained in detail in the following sections.

Although there are a large number of parameters in the model, some of them exert little influence on the model's endogenous variables that are considered as "important". (Only manufacturing sectors are explicitly modeled on the basis of micro data in MOSES. Therefore, for example, the simulated growth rates of manufacturing output may be regarded more important than those of other sectors.) It is important to determine "key" parameters that have significant effects on certain endogenous variables. The "Global Random Search" option of the MOSES.CALIB program can be used for this purpose as explained in Sections 3.2 and 3.4.

Sensitivity analysis is used to examine the effects of minor changes in initial conditions and stochastic factors on simulation results. It is normally preferred to have a "robust" model which is not very sensitive to minor changes in initial conditions. Otherwise, it may be difficult to evaluate simulation results because initial variables inevitably have a noise component.(*) The model's sensitivity to small variations in parameters values and stochastic factors are analyzed in Section 3.4.

Finally, the MOSES.CALIB workspace allows the user to apply Zellner and Peck's (1973) relative symmetry and linearity measures (see

*) As mentioned by Kuh, Neese and Hollinger (1985: 18, fn 2), "[i]f a particular model has low parameter sensitivity for endogenous variables of interest, then the Lucas critique (1976) -that parameters change in response to stochastic behavior of variables of concern to individual agents- will have less potential practical importance. Conversely, if significant policy parameters induce large responses, the potential importance of Lucas's observations will be all the greater. Even more to the point, parameters thought to be endogenous should be treated that way from the outset."

also Kuh, Neese and Hollinger, 1985: 110-113). Zellner-Peck measures are used

> to determine the extent to which induced changes in the model's endogenous variables are symmetric and/or linear. Symmetry is of interest for its own sake while finding of linearity of near-linearity may be useful in efforts to simplify the model's structure. Also, since the effects of both relatively small and large changes are reported, we gain information on both the local and global properties of the model. (Zellner and Peck, 1973: 152, quoted by Kuh, Neese, and Hollinger, 1985: 110)

The symmetry and linearity measures are summarized in Section 3.3 and results of an experiment are presented in Section 3.6.

3.2 CALIBRATION PROCEDURE

Thanks to improvements in information technology, large scale micro simulation models have been increasingly used in the last decades for economic analysis and policy simulations. A major problem of such complex models is the fact that the model specification includes a large number of unknown parameters. Parameter values have to be guesstimated since it is usually not possible to get their econometric estimates due to lack of necessary data. Thus, the "calibration" of large-scale micro simulation models is a problem that needs further research. This section presents a method used for the "calibration" of the MOSES model.(*)

Calibration is used to estimate the parameter values in a predetermined set that "minimizes" the difference between simulation results and some choosen control variables, observed for the calibration period. This "minimization" problem can be summarized as follows.

The model can be thought of as a system of equations containing an r-element parameter vector, P, such that a vector of initial and exogenous variables, Z, produces a v-element vector of endogenous control variables Y^s (**).

$$Y^s = M(P;Z)$$

Our purpose is to estimate the parameter vector, P, that makes the simulated control variables as close to their real values, Y^r, as possible. If the standard weighted sum-of-squared-errors criterion is used, then the objective function is

*) Eliasson and Olavi (1978) formulated a calibration procedure for the model in the early stages of the development of the MOSES model. However, it could not be implemented in practice because computers available in late 1970s were not fast enough to perform the calibration work at a reasonable cost. See also Klevmarken, 1978.

**) Note that M(.) is a stochastic function of P due to some stochastic variables in the model. Hence, the Y^s is distributed by mean Y^{smean} and standard deviation σ_y. If a short time period is used for the calibration procedure, σ_y can be assumed to be zero since the stochastic variances in the model have significant effects only in the long run.

$$\min_{P \epsilon S} d(P) = \Sigma_{t \epsilon T} (Y_t^s - Y_t^r)' W (Y_t^s - Y_t^r) \, ,$$

where S is the predetermined parameter space, T the time period for calibration, and W a diagonal weighing matrix. It is assumed that the parameter set S contains the global minimum in its interior.

If the model could mimic exactly the real world, and the actual values of the exogenous variables could be same as those used in the model, then the minimum distance value, d^{min} would be equal to zero. Since this is not the case, $d^{min} > 0$.

Recall that there are many non-linear and non-smooth functions in the model. M(.) and, consequently, d(P) are very complex systems of non-linear functions. It is very difficult, if at all possible, to find the minimum distance value by conventional optimization methods (Fletcher, 1987). The d(P) function may have many local minima and it is typically a function with a large number of parameters (in our case, more that 60 parameters). Thus the minimum of the d(P) function can only be searched by random search methods since, as Brooks (1958) stated, the number of experiments in these methods do not depend on the number of parameters and they are not usually restricted by the nature of the distance surface.(*)

Finding the "minimum" of the distance function is only one aspect of the calibration process. The model needs to be robust in the neighborhood of the selected parameter vector P^c, i.e., small variations in parameter values should not have significant effects on system's propeties. In other words,

$$| \; d(P) - d(P+\delta) \; | \leq \epsilon \; ,$$

where δ is an r-element vector of variations, and ϵ a small number.

Finally, the model's long-run properties generated by the parameter

*) These methods were first proposed by Anderson (1953). For some recent studies, see Solis and Wets (1981), Boender *et al.* (1982), and references therein.

vector P^c should be "reasonable". Hence we seek to establish convergence to a small region surrounding (in some sense) the candidates for the global minimum. The optimality region then is defined by

$$S_o = \{ P \mid P \in S, d(P) < d^{min} + \epsilon \} \quad , \text{where } \epsilon > 0.$$

In other words, 1) the parameter vector P^c does not need to be global minimum of the distance function but $d(P^c)$ should be a local minimum "close" enough to d^{min}; 2) the model should be robust at the neighborhood of $d(P^c)$; and 3) the long-run properties of P^c should be satisfactory. To determine the parameter vector P^c a simple two-stage calibration procedure can be designed as follows.

Stage 1. Global search

At this stage n parameter vectors are randomly drawn from the parameter space, S, and the distance values for each parameter vector are found by simulation. The number of experiments is determined by two criteria.

1) $n \geq \log(1\text{-}s)/\log(1\text{-}b)$

where b is a predetermined proportion of the parameter space that contains parameter combinations having lower distance values (S_b), and s is the probability of at least one experiment choosen from the parameter sub-space S_b. For example, if we want to get at least one parameter vector drawn from the lowest 5% of the parameter space with 99% probability, then

$n \geq \log(1\text{-}.99)/\log(1\text{-}.05) \approx 90.$

2) The cumulative distance distribution (CDD) of the sequence of experiments with randomly selected parameter vectors will converge to the population distribution as $n \rightarrow \infty$. Thus n can be choosen so that additional experiments do not change the CDD to a significant extent.

By global search with n random parameter vectors, we can get information about 1) the global minimum, 2) the shape of the distance surface, and 3) influential parameters.

There is a number of techniques, generaly based on de Haan's (1981) analysis of order statistics, for determining the confidence interval for the global minimum. (For an application of this method, see Veall, 1990.) Suppose

$d^{min} = \min \{ d(P) \mid P \in S \}$

exists. Using a random sample of $P^1, P^2, \dots P^n$ from a uniform distribution over S, a confidence interval at the p significance level for d^{min} (under mild conditions on d(.)[(*)]) can be constructed as follows.

$\{ d^m, Y^1 \}$,

where $d^m=[Y^1-(Y^2-Y^1)/p^{-1/\alpha}-1]$, $Y^1, Y^2, \dots, Y^n$ is the order statistic from $d(P^1), d(P^2), \dots d(P^n)$, that is, $Y^1 \leq Y^2 \leq \dots \leq Y^n$, $\alpha=\log(k)/[\log(Y^k-Y^3)/(Y^3-Y^2)]$, and k is any sequence of integer numbers depending on n such that $k(n) \to \infty$ and $k(n)/n \to 0$ as $n \to \infty$. Note that Y^1 tends to d^{min} almost surely as $n \to \infty$, and d^m is a monotone decreasing function of p such that $d^m \to -\infty$ as $p \to 1$.

This confidence interval can be used for testing the hypothesis that any given minimum is global. Since our purpose is to find a parameter vector in the optimality region only, the confidence interval can be used to assess the "closeness" of the distance value found at Stage 2 to the global minimum.[(**)]

The CDD obtained in this stage gives an idea about the shape of the distance surface. A relatively smooth curve at low values may show that the model is not sensitive to small variations in parameter values (see Figures 3.3.)

Influential parameters can also be determined at this stage. Recall that parameter values are chosen randomly so that they are independently

[*)]For the conditions on d(.), see de Haan (1981) and Boenden *et al.* (1982).

[**)] There are some techniques to estimate the value of global minimum. See, for example, Smith (1987).

distributed across experiments. Ignoring joint effects, we can estimate the distance values as an approximate function of each parameter separately. For example, regression analysis may be employed by using linear, quadratic and cubic functions of parameters as explanatory variables. The F-statistic of the regression may show the significance of that parameter.

Stage 2. Local search

Local search is based on a method of random "hill-climbing" from an initial point. Direction of the next move is determined randomly. If the distance value decreases at the randomly selected direction, parameter values are changed accordingly. Otherwise, a new random direction is selected. This process is repeated until no further improvement in the distance value is obtained in a predetermined number of trials.

The algorithm we use is rather simple and can be summarized as follows.

Step 0.

Pick $P^0 \epsilon S$, set $k=nf=ns=0$, fix μ_{min}, μ_{-1}, mf, ms, sr, si, con

k : iteration number
P^k : parameter vector for k^th iteration
S : parameter space
nf : number of failures
ns : number of successes
μ_k : "diameter" of random search space (step size). P^k is the center point.
μ_{min} : minimum diameter
mf : number of failures before step size reduction
ms : number of successes before step size increase
sr : step size reduction factor
si : step size increase factor
con : convergence criterion

Step 1.

$$\text{Set } \mu_k = \left[\begin{array}{l} \mu_{k-1}{*}\mathrm{sr} \text{ if } \mathrm{nf} \geq \mathrm{mf} \\ \mu_{k-1}{*}\mathrm{si} \text{ if } \mathrm{ns} \geq \mathrm{ms} \\ \mu_k \end{array}\right.$$

Stop if $\mu_k \leq \mu_{min}$. Set $P^c = P^k$.
Otherwise set $P^{k+1} = \mathrm{ran}(P^k + \mu_k)$ where ran(.) is a random vector on the "circle" defined by its center P^k and diameter μ_k.

Step 2.

$$\text{Set } P^{k+1} = \left[\begin{array}{l} P^{k+1} \text{ if } d(P^{k+1}) < d(P^k)/1+\mathrm{con}\ , \ \text{ set } \mathrm{ns}=\mathrm{ns}+1, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad k=k+1,\ \mathrm{nf}=0 \\ P^k \ \text{ otherwise , set } \mathrm{ns}=0,\ \mathrm{nf}=\mathrm{nf}+1,\ k=k+1 \end{array}\right.$$

Return to Step 1.

There are two alternatives for initial parameter vector, P^0: 1) the parameter vector found at Stage 1 that has the lowest distance value, 2) the parameter vectors that is currently being used in the model. We prefer to use the second alternative because *some* of the parameters currently used may incorporate our *a priori* information about their values. Moreover, these parameters are known to have "reasonable" long-run properties and the model is not sensitive to small variations in these values.

If the random local search ends up with a high distance value compared to the "estimated" global minimum, repeat the search process either with a new initial parameter vector or use the same vector with longer step sizes.

Finally the long-run properties of the selected vector, P^c, should be tested. If it generates nonstable, chaotic results after the calibration period, we need to make another search. The sensitivity of the model is easy to check since at least nf number of experiments are done already in the neighborhood of P^c during the local search.

We used only two parameters (the MAXDP and BETABW parameters) to visualize the search process and the shape of the distance space. By performing a complete grid search (total 122 experiments for 11 values of each parameter) the distance surface[*] was found as in Figure 3.12. As can be seen in this figure, even with two parameters there may be many local mimima. (The global minimum is marked by a small square on the figure.) However the shape is rather smooth and mostly influenced by changes in the MAXDP parameter. The BETABW parameter seem to have less influence as also indicated by regression analysis in Section 3.4 (see Table 3.5).

The local search process which was started at the middle of the parameter space converged successfully to a local minimum which is very close to the global optimum. Note however that this may not be so in other cases since the starting point was in a relatively smooth valley in this experiment.

*) It is practically impossible to find global optimimum by grid search when the number of parameters is high. For example, if 11 values are tried for of each parameters, then n^{11} experiments is needed for complete search. If it takes 10 minutes for each experiment as in the case of calibration procedure used in Section 3.4, calibration of 10 parameters takes almost 2 million years!

3.3 USING THE MOSES.CALIB WORKSPACE

The calibration functions are stored in the APL workspace MOSES.CALIB (for the code, see Appendix H). Before running the calibration procedure, you should create two matrix variables in the workspace DATA. The first matrix variable (the control variables matrix) contains data about the control variables, and it should be a (t+2)*n matrix where t is the number of years, and n the number of variables. The first row of the control variable matrix contains the names of control variables, and the second row contains the variable weights that are to be used in calculating the distance between the simulation results and the control variables. Note that the variable names should be the same as those used in the model. Other rows contain the observed values of control variables for each year. For example, the following matrix contains data about the rate of interest (RI), the rate of unemployment (RU), and annual industrial output growth rates (DPROD) for 1983-1986. The weights of the RI and RU variables are higher than that of the DPROD variable. Note that the sum of weights does not need to be equal to one in this matrix. Weights are recalculated by the program.

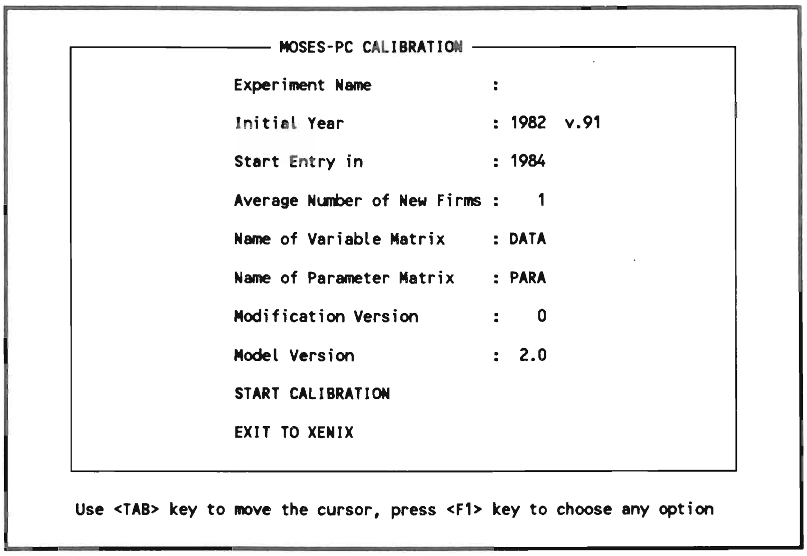
Table 3.1 Control variables matrix

RI	RU	DPROD	← *Control variable name*
2	2	1	← *Variable weight*
.1085	.035	.04	← *Variable values*
.1177	.031	.08	←
.1385	.028	.02	←
.1015	.027	.00	←

The second matrix variable (the parameter matrix) contains some information about parameters to be calibrated (or policy variables to be found). It is a 6*m matrix where m denotes the number of parameters. Parameter names should be in the first row, the maximum and minimum permissible values for the parameters in the second and third rows, respectively, the starting parameter values in the fourth row, the step size in the fifth row, and a value for the initial search direction in the sixth row. If you want to start the iterative, stepwise search upward (downward), enter a positive (negative) value for this row.

Table 3.2 Parameter matrix

ALFABW	BETABW	CHRFUND1	← *Parameter name*
.056	1.35	.03	← *Maximum parameter value*
.0	.25	.0	← *Minimum parameter value*
.025	.855	.005	← *Starting parameter value*
.003	.055	.002	← *Step-size*
1	1	1	← *Initial search direction for iterative search (positive to increase, and negative to decrease the parameter values in the first iteration)*



```
                    MOSES-PC CALIBRATION
      Experiment Name              :
      Initial Year                 : 1982  v.91
      Start Entry in               : 1984
      Average Number of New Firms  :   1
      Name of Variable Matrix      : DATA
      Name of Parameter Matrix     : PARA
      Modification Version         :   0
      Model Version                :  2.0
      START CALIBRATION
      EXIT TO XENIX

Use <TAB> key to move the cursor, press <F1> key to choose any option
```

After preparing these two matrix variables in the DATA workspace,(*) load the MOSES.CALIB workspace by typing

) **LOAD MOSES.CALIB** RETURN

When the MOSES.CALIB is loaded, the MOSES-PC CALIBRATION menu will appear.

"Experiment Name" defines the workspace name. If you do not enter any name here, the default name, DELETE.ME, is used.

"Initial Year" means the starting point of the simulation experiment, i.e., the initial dataset year. The simulation commences the first quarter after the "initial year". (As of May 1, 1991, there are only two datasets. In this case this variable should be either 1976 or 1982.)

The next variable on this line shows the version of data year. For the time being, there is only one version of the 1976 dataset (version 5), and six versions of the 1982 dataset (versions 89, 91, 92, 93 ,94, and 98).

"Start Entry in" enables the endogenous firm entry after the first quarter of the specified time. It should be at least one year after the initial simulation period. (In other words, if you use a 1982 dataset, start entry can start in 1984.)

"Average Number of New Firms" specifies the entry feature of MOSES (the A̲MAXENT variable in the START∆ENT2 function). It is equal to the maximum possible number of new firms in each industry in each year when the average industry profitability is equal to unity. The number of new firms is a probabilistic linear function of average industry profitability. If you do not want to use the firm entry option, enter zero for this variable.

"Variable Matrix" and "Parameter Matrix" are as explained above. The default variable matrix, DATA, contains observed data for five variables for 1983-1988. (The control variables in this matrix are the

*) Control variables and parameter matrixes have a generic form that is used for all options of the MOSES.CALIB program. However these matrices may contain redundant information for a specific option. For example, data in the sixth row of the parameter matrix that specify initial search directions are used only by the ITERATIVE SEARCH process.

interest rate, the annual growth rates of manufacturing employment, output and prices, and annual growth rates of GDP.) The default parameter matrix, PARA, is the variable used in Section 3.4. Both matrices are saved in the DATA workspace, and can be modified for a specific calibration exercise. (For these matrices, see Section 3.4.)

"Modification Number" determines which MSTART function will be used to modify the model. In the MOSES model, each experiment is carried out by making changes in the original model (changes in the behavioral equations, variables, etc.) by using one of the modification functions in the MSTART workspace. This allows you to keep the model in its original form. The MSTART*xx* functions in which all changes connected with the specific experiment are defined modifies the original model at the beginning of the simulation. If you want to make experiments with the original model, enter zero for this variable.

There are three versions of the MOSES model available: 1.0, 1.1, and (the default) 2.0. (For details, see Section 1.8.)

Pressing F1 key when the cursor is on the "Exit to Xenix" allows the user to exit from the model.

When the experiment is started by pressing the F1 key when the cursor is on the "START CALIBRATION" option, the program asks some interactive questions about the search method. The first question is

```
CALIBRATION PROCEDURE
(Global random search / Local random search /
Iterative search / Seed number / Zellner's test)
```

Enter the first letter of the search method. (The default option is the global random search.) To choose the default option, press the RETURN key.

In the GLOBAL RANDOM SEARCH, the parameter values are chosen randomly by using a pseudo-random number generator within the range defined by the second and third rows of the parameter matrix (Table 3.2).

The step-size values defined in the fifth row of the parameter matrix is used in choosing parameter values, and the same parameter vector is not used twice.[*] Therefore, if you want to obtain completely random parameter values, enter very small step sizes. If you want to perform a complete grid search for a few number of parameters, enter large step sizes.

If you choose global random search, the minimization criterion is asked:

```
MINIMIZE Squared / Absolute / Maximum Distance :
```

Enter the first letter of the distance criterion. (The default option is the squared distance.) The distance criteria are calculated as follows:

Squared Distance = $\Sigma_T \, \Sigma_V \, w_v{*}(a_{vt} - s_{vt})^2$,
Absolute Distance = $\Sigma_T \, \Sigma_V \, w_v{*}\,|a_{vt} - s_{vt}|$,
Maximum Distance = $\max\,(w_v{*}\,|a_{vt} - s_{vt}|;\, v \in V,\, t \in T)$,

where a_{vt} is the observed value of the vth control variable at time t, s_{vt} the simulated value of the same variable, w_v the weight of the vth variable, V the set of control variables, and T the time period.

Then, the program asks if you want to standardize the variables:

```
STANDARDIZE VARIABLES (No/Yes) :
```

"No" is the default option. If you enter "Yes", actual and simulated variables are standardized by the means and standard deviations of real variables before calculating the distance value. In other words, actual and

*) Parameter values for the ith experiment are selected as follows.
$P^i = P^{min} + n{*}P^{step}$, where n is a vector of integers randomly drawn form a uniform distribution over $\{0, \mathrm{int}((P^{max} - P^{min}) \,/\, P^{step})\}$. P^{max}, P^{min} and P^{step} correpond to the second, third and fourth rows of the parameter matrix, respectively.

simulated variables are transformed in the following way.

$a^{*}_{vt} = (a_{vt} - a^{m}_{vt}) / a^{s}_{vt}$, and

$s^{*}_{vt} = (s_{vt} - a^{m}_{vt}) / a^{s}_{vt}$

where a^{m}_{vt} and a^{s}_{vt} are the mean value and the standard deviation of the control variable a_v. Standardization is advised when the control variables have different units of measure.

Then, the program asks three more questions.

```
MAXIMUM DEVIATION IN A YEAR (1) :
MAXIMUM ITERATION NUMBER (25)   :
CHANGE THESE VALUES (No/Yes)    :
```

The model may generate erratic results for some parameter sets. In such a case, it is necessary to stop the simulation because, otherwise, the program may be halted. If the distance value exceeds the value of the MAXIMUM DEVIATION *IN A YEAR*, the program stops execution of the simulation for that parameter set. The maximum iteration number defines the number of parameter sets to be simulated. If your entries are correct, then answer **No** to the last question to proceed. If you want to change the values entered, answer **Yes**. In this case, the program will start with the question about the "calibration procedure."

The results of global random search are stored in the RESULTS and BESTTAB matrix variables. RESULTS is a (n+1)*(p+v+2) matrix where n is the number of experiments, p the number of parameters, and v the number of control variables. The first row contains variable names. The first column shows experiment number. The next p number of columns store parameter values of that experiment. Then the weighed average of the distance value, and distance values for each control variables are stored. BESTTAB is a t*v matrix where t is the period of simulation in years. It stores the values of control variables for the experiment that have minimum (average) distance value.

If you choose LOCAL RANDOM SEARCH, the program executes a local random search as summarized in the previous section. You will be asked to answer the following questions.

```
MINIMIZE Squared / Absolute /
           Maximum distance       :
STANDARDIZE VARIABLES (No/Yes)    :
MAXIMUM DEVIATION IN A YEAR (1)   :
MAX NUMBER OF STEP REDUCTIONS (1) :
# FAILURES BEFORE ▲ STEP (20)     :
# SUCCESSES BEFOR ▲ STEP (20)     :
STEP SIZE REDUCTION FACTOR (.5)   :
STEP SIZE INCREASE FACTRO (1)     :
CONVERGENCE VALUE (0.01)          :
CHANGE THESE VALUES (No/Yes)      :
```

These variables are explained in the previous section. As in the case of global random search, results are saved in RESULTS and BESTTAB variables.

In the ITERATIVE SEARCH, the program starts with the initial parameter values as defined in the fourth row of the parameter matrix. Then the program changes the value of the first parameter by a step length which is defined in the fifth row until the "distance" increases. When the objective function cannot be decreased by changing the first parameter in either directions, or the maximum and minimum values are reached, the second parameter is changed. This stepwise procedure goes on until all parameters have been tried. You can try another iterative search by using those parameter values found in the first iteration.

The minimum number of iterations in this method is equal to 1+2*number of parameters. A one year simulation takes about 90 seconds. In other words, an iteration in the calibration of the model for 1983-1988 (6 years) takes about 10 minutes. Therefore, this method is not recommended when there are many parameters to be calibrated.

When you choose the iterative search process, the following questions will be asked.

```
MINIMIZE Squared / Absolute /
         Maximum distance         :
STANDARDIZE VARIABLES (No/Yes)    :
CONVERGENCE VALUE (0.01)          :
CHANGE THESE VALUES (No/Yes)      :
```

The results of the iterative search are saved in the RESULTS variable.

If you choose the option SEED NUMBERS, the following questions will be asked.

```
MINIMIZE Squared / Absolute /
           Maximum distance       :
STANDARDIZE VARIABLES (No/Yes)    :
MAXIMUM DEVIATION IN A YEAR (1)   :
NUMBER OF EXPERIMENTS (25)        :
CHANGE THESE VALUES (No/Yes)      :
```

Answer these questions as explained before. In this case, n experiments will be performed with different initial seed numbers to examine the effects of stochastic effects on simulation results. Note that, in this case, the program does *not* use the parameter matrix you defined on the MOSES-PC CALIBRATION menu because parameter values are the same in all experiments. Experiment results are saved in the RESULTS variable.

If the option ZELLNER'S TEST is chosen, the following questions will be asked.

```
NAME OF THE CONTROL VARIABLE                  :
MAXIMUM CHANGE IN THE CONTROL VARIABLE (25%) :
NAMES OF THE TEST VARIABLES                   :
```

```
SIMULATION PERIOD                          :
```

In this option, neither control variables matrix nor parameter matrix (Tables 3.1 and 3.2) are used. (However, their names should be entered in the MOSES-PC CALIBRATION menu.) This option calculates Zellner-Peck symmetry and linearity statistics that measure the extent of departures from symmetry and linearity of *test variables* as a result of changes in the value of the *control variable*. You should enter only one variable (or parameter) name for the control variable since the program runs simulations for one variable at a time. However, you may enter as many test variables as you like since an increase in the number of test variables does not increase the number of experiments.

For the calculation of symmetry and linearity measures, the base run is obtained by running the model for the initial value of the control variable. Then four experiments are made by changing the control variable by the amount specified in the MAXIMUM CHANGE IN THE CONTROL VARIABLE.(*) The symmetry and linearity statistics are calculated as follows (see Kuh, Neese and Hollinger, 1985: 110-113).

A deviation from the base run for a test variable y in time t is defined by

$\delta_t^{\blacktriangle} = y_t^{\blacktriangle} - y_t^{b}$

where y_t^b is the value of y (the test variable) at time t for the base run and $y_t^{\blacktriangle}$ is the value of y where the control variable was changed by $\blacktriangle$ units.

Deviation from symmetry is defined as

$\tau_t = |\delta_t^{\blacktriangle} + \delta_t^{-\blacktriangle}|$

If y were perfectly symmetric with respect to changes in the control variable, τ_t would be equal to zero.

The symmetry measure is the sum of deviations scaled by absolute

) Denote the maximum change in the control variable by Γ^{max}. Then experiments will be made by increasing the control variable Γ^{max}% and 0.1Γ^{max}%, and decreasing it with the same amount.

deviations as follows.

$SYM = \Sigma\tau \; / \; \Sigma(|\delta_t^{\blacktriangle}| + |\delta_t^{-\blacktriangle}|)/2$

The symmetry measure, SYM, is constrained to be in the region [0, 2]. A value of SYM near zero indicates high symmetry in response to the test variable to changes in the control variable, and a larger value indicates less symmetry.

The linearity measure is defined similarly as the sum of deviation from linearity scaled by absolute deviations. Deviation from linearity is defined as follows.

$\Theta_t = \delta_t^{10\blacktriangle} - 10 \; \delta_t^{\blacktriangle}$

Then the measure of linearity is

$LIN = \Sigma\Theta \; / \; \Sigma(|\delta_t^{10\blacktriangle}| + |10 \; \delta_t^{\blacktriangle}|)/2$

Similarly, the linearity measure, LIN, is constrained in the region [0, 2]. If the response of the test variable to changes in the control variable were perfectly linear, LIN = 0.

The results of Zellner-Peck tests are saved in a variable called RESULTS. This is a vector variable that contains the symmetry and linearity measures for each test variables. (For the form of this variable, see Section 3.6.)

In a usual experiment, some messages about date, the model version, etc., are shown on the screen right before the beginning of the simulation. Then, the simulation is started. Depending on the method, some information about current iteration are shown on the screen.

The MOSES.CALIB workspace stores all initial variables in an external file called DELETE.CAL. If the procedure ends normally, this file is deleted from the hard disk. Since a calibration experiment can take a long time, you may want to stop the experiment to start it later. In this case, you have to untie the external file first. To do so, type

[]FUNTIE 1 RETURN

where [] is the APL character entered by **Shift-L** when the keyboard is in the APL mode. (If the keyboard is in the ASCII mode, type

Ctrl-N to switch back to the APL mode.) Then save the workspace and exit from the APL by typing

```
)SAVE  RETURN
)OFF  RETURN
```

When you want to continue the experiment, start the APL interpreter and load the experiment workspace. Then, tie the external file and continue the experiment by typing

```
'DELETE.CAL'  []FTIE  1  RETURN
→ []LC  RETURN
```

where → is the APL character entered by **Shift-]**.

When the simulation is not finished properly, the external file DELETE.CAL may not be deleted from the hard disk. In that case, a file tie error occurs in the SAVE∆DATA function. If you get the file tie error message, delete the external file DELETE.CAL and resume the simulation by typing

```
)DROP  DELETE.POL  RETURN
→ []LC  RETURN
```

3.4 PREPARATION OF THE SYNTHETIC DATABASE, R1990.10

The synthetic database R1990.10 is prepared for external use of the model, since other datasets (R1976.5 and R1982.*xx*) contain confidential firm data. This dataset is prepared by running the model version 2.0 for eight years. R1982.91 is the initial dataset. Before the simulation experiment, the parameters were calibrated to get a good fit for certain exogenous variables for 1982-1988.(*) (The calibrated parameters are saved in MSTART900 function.) This section explains the calibration process of the synthetic database R1990.10.(**)

Before beginning the description of the calibration of the MOSES model, let us specify some notations. P^{max}, P^{min} and P^0 are the vectors of maximum, minimum, and initial parameter values, respectively. [P^{max}, P^{min}] defines the parameter space. μ_{-1} is the initial step size vector. There were 65 parameters to be calibrated. Rates of interest (RRI), annual growth rates of manufacturing employment, output, and prices (MEMP, MOUT, and MPRICE) and annual growth rate of GDP (GGDP) were used as control variables.(***) The DATA matrix is shown in Table 3.3.

To determine the cumulative distance distribution (CDD), 100 experiments were run by choosing parameters randomly within the range [P^{max}, P^{min}]. See Table 3.4 for the (inverted) PARA matrix. The CDDs for

*) The complete process described here took about 20 hours of CPU time. Since the model is installed in a PC computer and the calibration process needs almost no user interaction, it can be done in a day with almost no additional cost.

**) For the calibration process, the MSTART999 function was used to define control variables.

***) These variables were selected for calibration because of two reasons. First, in MOSES, only manufacturing sectors are modeled explicitly on the basis of micro data. Thus the model should have better results for manufacturing sectors. Second, the calibrated parameter set was prepared for a study by Carlsson and Taymaz (1991) on the role of technological progress and economic competence of manufacturing firms in economic growth.

the first 50 and 100 experiments are shown in Figure 3.3. There is no significant change in the frequency distributions between 50 and 100 experiments. The maximum and minimum distance values obtained in 100 experiments are .0259 and .0026, respectively. The CDD figure reveals that the landscape defined by the distance values has a relatively flat surface. The distance value was within the range of .0026-.0100 in almost 70% of the experiments.

Table 3.3 The DATA matrix

RRI	MEMP	MOUT	MPRICE	GGDP
1	1	1	1	1
0.109	-0.033	0.052	0.113	0.024
0.118	0.007	0.049	0.078	0.034
0.139	0.001	0.028	0.056	0.023
0.102	-0.002	0	0.061	0.023
0.092	-0.001	0.027	0.037	0.029
0.101	-0.011	0.027	0.061	0.023

Table 3.4 The PARA matrix

Parameter name	Max value	Min value	Initial value	Step size	Direction
ALFABW	0.05625	0	0.0298	0.005625	1
BETABW	1.35	0.25	0.885	0.11	1
CHRFUND1	0.03	0	0.0183	0.003	1
CHRFUND2	0.03	0	0.0163	0.003	1
E1	0.2	0	0.1	0.02	1
FIP	0.1538462	0	0.0871	0.01538462	1
FIS	0.1538462	0	0.0697	0.01538462	1
FIW	0.1538462	0	0.0807	0.01538462	1
GAMMA	0.6	0	0.3106	0.06	1
GBRWRAT	1	0.25	0.68	0.075	1
GDEPRAT	0.4	0	0.154	0.04	1
INTDIFF	1.5	1	1.19	0.05	1
IOTA	1	0.8	0.9	0.02	1
IOTALOW	0.3	0	0.14	0.03	1
KSI	0.3	0	0.15	0.03	1
LAMDA1	1	0	0.588	0.1	1
LAMDA2	1	0	0.766	0.1	1
LOWER MHIST	1	0.6	0.77	0.04	1
MAXCHIDIFF	0.01	0	0.005	0.001	1
MAXDP	0.02	0	0.01	0.002	1
MAXIDIFF	1	0.98	0.99	0.002	1
MAXQCHRI	0.01	0	0.005	0.001	1
MAXQCXRATE	0.02	0	0.01	0.002	1
MAXRIDIFF	0.1	0	0.046	0.01	1
MAXWCOEFF	0.45	0.05	0.204	0.04	1
MINIDIFF	0.99	0.61	0.8	0.038	1
MINRFUND1	0.17	0.13	0.15	0.004	1
MINRFUND2	0.17	0.13	0.15	0.004	1
NEWFUND	0.75	0.25	0.495	0.05	1
NORMRU	0.03	0.01	0.019	0.002	1
OBSRATE	4	2	3.075	0.2	1
QUITCOEFF	1.5	0.5	1.04	0.1	1
R	0.75	0.25	0.5	0.05	1
RFUND1	0.2	0.17	0.16	0.003	1
RFUND2	0.2	0.17	0.18	0.003	1
RHO	0.00914	0.005	0.00724	0.000414	1
RHOBOOK	0.06112	0.01	0.03	0.005112	1
RLU	0.8	0.4	0.607	0.04	1
RTRANS	0.7	0.3	0.5	0.04	1
SKREPA	140	10	85	13	1
SMP	0.75	0.25	0.51	0.05	1
SMS	0.75	0.25	0.54	0.05	1
SMT	0.75	0.25	0.47	0.05	1
SMW	0.75	0.25	0.43	0.05	1
THETA	0.019	0.001	0.0109	0.0018	1
TMFASS	0.4	0.1	0.241	0.03	1
TMFD	0.28334	0.05	0.1644	0.023334	1
UTREF	0.99	0.71	0.84	0.028	1
XRATECOEFF	0.2	0	0.11	0.02	1
XRPEFF	0.2	0	0.104	0.02	1
TMSTO	1.75	0.25	0.95	0.15	1
TMIMSTO	1.75	0.25	0.93	0.15	1
MAXQDSUFOR	0.1	0.014	0.015	0.01	1
BETA1[1]	1.1	0.9	1.01	0.02	1
BETA1[2]	1.1	0.9	1.02	0.02	1
BETA1[3]	0.8	0.6	0.69	0.02	1
BETA1[4]	0.85	0.65	0.78	0.02	1
BETA1[5]	1	0.8	0.88	0.02	1
BETA1[6]	1.1	0.9	1.03	0.02	1
BETA1[7]	1.1	0.9	0.98	0.02	1
BETA1[8]	1	0.8	0.9	0.02	1
BETA1[9]	1.1	0.9	1.01	0.02	1
BETA1[10]	0.85	0.65	0.75	0.02	1
BETA1[11]	1.1	0.9	1	0.02	1
BETA1[12]	0.6	0.4	0.49	0.02	1

Table 3.5 Correlations between parameters and control variables (R^2)

Parameter		Control Variables				
Name	AVERAGE	RRI	MEMP	MOUT	MPRICE	GGDP
ALFABW	29.45	2.15	17.22	50.25	8.03	46.87
BETABW	1.78	3.29	0.16	5.64	2.40	3.10
CHRFUND1	0.30	3.42	0.63	3.97	1.07	3.57
CHRFUND2	5.96	3.54	6.09	8.01	4.48	5.98
E1	4.91	5.03	6.73	2.22	4.40	1.10
FIP	4.76	1.35	6.37	3.64	6.96	3.13
FIS	0.08	4.39	0.98	0.90	0.68	0.74
FIW	5.39	1.04	7.13	4.54	4.73	1.78
GAMMA	1.41	1.17	3.67	1.24	0.32	2.96
GBRWRAT	0.46	14.72	1.71	1.91	3.23	1.77
GDEPRAT	1.24	1.53	1.18	2.98	0.93	4.26
INTDIFF	7.74	1.36	3.84	9.94	6.15	6.65
IOTA	0.90	2.69	0.61	1.35	1.60	1.35
IOTALOW	0.55	2.39	0.21	2.03	2.64	3.08
KSI	1.23	7.09	0.48	1.57	0.43	3.39
LAMDA1	5.01	2.01	0.33	6.33	1.43	7.80
LAMDA2	8.33	13.49	8.92	2.08	3.49	3.57
LOWER MHIST	1.30	3.11	0.41	1.72	0.09	1.91
MAXCHIDIFF	16.58	2.65	29.49	2.63	28.38	2.04
MAXDP	16.36	4.94	54.15	3.80	55.09	8.07
MAXIDIFF	...	...	...	...	...	...
MAXQCHRI	14.04	21.75	7.80	6.32	6.07	2.44
MAXQCXRATE	6.36	1.36	2.85	4.49	5.59	3.52
MAXRIDIFF	5.59	4.07	3.51	3.73	1.12	1.89
MAXWCOEFF	5.27	0.74	9.86	0.26	7.61	0.78
MINIDIFF	10.80	9.05	3.11	6.82	2.66	4.52
MINRFUND1	7.49	3.76	4.78	4.96	7.16	6.71
MINRFUND2	2.72	0.76	0.85	2.52	3.47	3.94
NEWFUND	1.13	2.10	0.31	1.23	1.02	1.02
NORMRU	1.74	1.30	1.91	1.95	1.94	2.91
OBSRATE	1.25	2.23	2.02	3.45	1.04	4.68
QUITCOEFF	6.03	4.98	3.36	2.94	3.51	1.19
R	1.14	3.34	1.90	2.52	2.41	1.06
RFUND1	...	...	...	...	...	...
RFUND2	...	...	...	...	...	...
RHO	1.24	5.23	1.29	2.14	1.96	3.78
RHOBOOK	5.27	2.12	2.59	8.82	5.32	3.57
RLU	5.70	0.73	1.35	4.88	4.39	2.80
RTRANS	2.46	2.94	2.92	1.36	5.37	10.76
SKREPA	5.06	3.15	1.83	1.72	0.71	2.15
SMP	1.16	1.51	5.96	0.71	7.44	1.29
SMS	2.86	2.34	1.86	2.45	0.22	2.86
SMT	5.66	1.83	5.56	3.78	1.36	4.70
SMW	0.53	3.68	3.37	4.24	4.05	5.02
THETA	3.72	6.21	0.98	3.02	0.80	0.46
TMFASS	1.33	1.74	1.17	2.81	2.03	1.96
TMFD	2.31	7.17	4.44	3.29	1.70	0.66
UTREF	9.16	5.31	3.05	6.99	5.46	7.21
XRATECOEFF	5.88	2.54	7.55	5.11	2.67	4.98
XRPEFF	5.77	1.76	4.14	3.31	3.52	2.64
TMSTO	3.49	6.23	8.04	0.59	4.28	0.77
TMIMSTO	5.69	2.73	2.37	7.81	0.93	5.21
MAXQDSUFOR	2.20	1.00	3.74	6.25	1.28	6.93
BETA1[1]	2.77	4.71	5.35	1.90	1.89	0.72
BETA1[2]	2.72	2.64	1.13	0.80	1.58	4.26
BETA1[3]	0.20	5.22	0.57	0.57	0.71	2.32
BETA1[4]	2.10	2.85	4.19	3.82	3.94	3.64
BETA1[5]	1.72	0.50	1.85	3.56	1.97	4.31
BETA1[6]	1.44	2.48	2.83	1.55	1.95	1.15
BETA1[7]	5.31	2.25	3.36	6.67	3.77	6.79
BETA1[8]	2.73	2.89	2.35	0.89	2.12	0.79
BETA1[9]	8.31	1.56	4.94	6.07	1.81	5.19
BETA1[10]	1.98	0.57	0.78	1.03	1.84	3.88
BETA1[11]	3.22	0.84	2.37	1.96	6.51	1.11
BETA1[12]	3.07	3.49	2.39	2.38	0.39	5.81

To establish the "confidence interval" for the global minimum, we used de Haan's equation with $k = \text{int}(n^{1/2}) = 100^{1/2} = 10$. The value of α is equal to 19.4, and the confidence interval at the 5% significance level is {.00209-0.0026}.

At this stage, a number of regression analyses were performed to approximate the relations between distance and parameter values. Note that parameters were chosen randomly so that they are independently distributed across experiments. Thus, we can regress the distance values as a function of each parameter separately to get unbiased estimates of the effects of parameters. First, second and third order functions, i.e., linear, quadratic, and cubic functions of parameters were used in the regression analysis. Table 3.5 shows the coefficients of determination, R^2. (An R^2 less than 7.5% means that that equation is not significant at the 5% level.)

The ALFABW parameter (which is used in the investment function) has the highest coefficient of determination. It explains almost 30% of the variation in the distance values as also shown in the following estimated regression equation.

$$\begin{array}{lllll} d_i = & .0061 + & .303\ \text{ALFABW}_i - & 15.008\ \text{ALFABW}_i^2 + & 244.5\ \text{ALFABW}_i^3 + \epsilon_i \\ & (3.99) & (1.17) & (-1.37) & (1.85) \end{array}$$

$$R^2 = 29.5, \text{ Number of observations} = 100$$

It is interesting to observe the impact of parameters on various endogenous variables. For example, the MAXDP parameter that is used in the price adjustments functions has a very strong impact on the growth rate of manufacturing employment and prices. (It explaines more than 50% of the variation in these variables.) But the changes in the MAXDP parameter have less impact on the growth rate of manufacturing output and rate of interest.

Having estimated the minimum possible distance value, a local random search was performed as shown in Figure 3.4. The solid line on this figure connects the minimum distance values found during the search process.

The distance value for P^0 was found .00562. Note that this distance value is very close to our confidence interval. It is within the best 15% of the parameter space as shown in Figure 3.3. The first random search around the P^0 vector had a distance value of .00547. Since it is less than $d(P^0)/(1+con)$ where the convergence criterion, con, is 0%, P^1 was set to that parameter vector.

The minimum distance value by the local random search, the P^c vector, was found to be .00203 at the 73th experiment. 60 more experiments were made around the P^c vector by decreasing the step size twice.(*) However none of these searches produced lower distance values. Incidentally, the minimum distance value found by the local random search is lower than the confidence interval, i.e., we cannot reject, at the 5% significance level, the hypothesis that P^c is the global minimum. This is rather an unexpected result, because the local random search does not guarantee to find the global minimum or any point that has a distance value lower than the confidence interval.

Sensitivity of results to small variations in the parameter values is shown in Figure 3.4 The last 60 experiments that have parameter values in the neighborhood of the P^c vector have close distance values. Results appear not to be very sensitive to minor changes in parameter values. The effects of stochastic factors (the use of random number generator in labor market and firm entry functions) were tested by making 15 simulations with different initial seed numbers. Results of these experiments show that stochastic factors have a negligible effect on those control variables used in the calibration process.(**) The long-run properties of this parameter set was checked by running the model for 20 years. Since the long-run

*) Figure 3.5 depicts how the local search proceeded for the first parameter, ALFABW. The value of ALFABW gradually declined during the first 25-30 experiments and then fluctuated around the "best" value that was obtained at the 73th experiment. Step size reductions are clearly shown on this figures.

**) All but one of experiments have a distnace value less than .0002.

results also seem to be reasonable, we use the P^c vector as the calibration vector. (These parameter values are saved in the MSTART900 function.)

Actual and simulated values of control variables used for the calibration process are shown in Figures 3.6-3.10. Simulated results are fairly close to their actual values. The highest divergence is obtained for the growth rate of GDP although the difference between simulated and actual values has been decreasing by time. High growth rates in GDP in initial years may be partially caused by the consistency problems of the macro data. However it seems that the model adjusts itself in a short time and the simulated growth rates converge to actual values.

Simulated annual growth rates of manufacturing employment and output are slightly higher that actual values except the last two years of calibration. Thus, the simulated labor productivity index which is of particular interest to Carlsson and Taymaz's (1991) study has a very good fit as shown in Figure 3.11.

3.5 A POLICY EXPERIMENT

This section explains how to make policy experiments by using the MOSES.CALIB workspace. As mentioned before, there is computationally no difference between exogenous variables and parameters in the model. By defining exogenous variables as parameters, policy experiment can be carried out.

In this example, we determine the rate of increase in government spending such that unemployment and price increases are minimized and that the rate of interest rate is forced to be close to 10% for five years. Thus, the data variable is as follows:

Table 3.6 The data variable for policy experiment

RI	RU	DPDOM
2	2	1
.1	0	0
.1	0	0
.1	0	0
.1	0	0
.1	0	0

Note that the weights for the unemployment and interest rates are twice that of price increases.

The rate of increase in government spending is initially 2% per quarter in all years, and is constrained to be 1-3% per quarter. The stepsize is chosen as .2%. Thus, the "parameter" variable for this experiment is as follows. (DPG*x* variable denotes the quarterly rate of increase in government spending in year *x*.)

Table 3.7 The parameter variable for policy experiment

DPG1	DPG2	DPG3	DPG4	DPG5
.03	.03	.03	.03	.03
.01	.01	.01	.01	.01
.02	.02	.02	.02	.02
.002	.002	.002	.002	.002
1	1	1	1	1

In this experiment, we used a iterative search and absolute distance criterion. The MSTART997 function was used to create the DPDOMU variable in simulations. Results (the RESULTS variable) are shown in Table 3.8.

In the first experiment, all "parameters" were set to their starting values. The distance value of this experiment was .200. Then, the first "parameter" was increased to .022. The distance value increased in this case. Thus, this "parameter" was decreased to .018, i.e., the direction of search was changed. The distance value was .189. Since the distance value was decreased, the program continued to decrease the first "parameter" to .016. In this case, however, the distance increase tod .194. The program then set the value of the first "parameter" back to .018 and increased the value of the second "parameter" to .022. The process continued in this manner. The lowest distance value was obtained in Experiment 16.

Table 3.8 Results of policy experiment

EXP NO	DPG1	DPG2	DPG3	DPG4	DPG5	DISTANCE
1	0.02	0.02	0.02	0.02	0.02	0.2002543957
2	0.022	0.02	0.02	0.02	0.02	0.2022882013
3	0.018	0.02	0.02	0.02	0.02	0.1888736674
4	0.016	0.02	0.02	0.02	0.02	0.1935819636
5	0.018	0.022	0.02	0.02	0.02	0.1922877454
6	0.018	0.018	0.02	0.02	0.02	0.1877501188
7	0.018	0.016	0.02	0.02	0.02	0.1904035338
8	0.018	0.018	0.022	0.02	0.02	0.1868195915
9	0.018	0.018	0.024	0.02	0.02	0.1888771772
10	0.018	0.018	0.022	0.022	0.02	0.1903165364
11	0.018	0.018	0.022	0.018	0.02	0.189356288
12	0.018	0.018	0.022	0.02	0.022	0.1872731091
13	0.018	0.018	0.022	0.02	0.018	0.1865172446
14	0.018	0.018	0.022	0.02	0.016	0.185931347
15	0.018	0.018	0.022	0.02	0.014	0.1856100519
16	0.018	0.018	0.022	0.02	0.012	0.1853771228
17	0.018	0.018	0.022	0.02	0.01	0.1854018131

3.6 AN EXAMPLE OF SYMMETRY AND LINEARITY MEASURES

In this section, we made an experiment with the RTRANS parameter to examine its relationships with government surplus (SURPLUSG), manufacturing output (+/S), and wealth of households (WH).[*] The RTRANS parameter was used to determine the level of transfer payments to households. The maximum change in the RTRANS parameter was set 10%.

To calculate symmetry and linearity measures, the program made five runs. The first run is the base run in which the current value of the RTRANS parameter was used. In the second and third runs, the value of RTRANS was decreased 10% and 1%, respectively. In the fourth and fifth runs, the value of RTRANS was increased 1% and 10%, respectively. The results of this experiment was saved in the RESULTS variable which is reproduced in Table 3.9.

As shown in Table 3.9, symmetry measures were calculated (by the program) for ±10% and ±1% changes in the RTRANS paramater, and linearity measures for -1/-10%, +1/+10% changes. Experiment results show that all test variables under consideration have very high symmetry and linearity measures, i.e., changes in the value of the RTRANS parameter does not have symmetric effects on these variables, and they are highly nonlinear. In other words, these results show that, at least for these variables and the RTRANS parameter, the model is highly nonlinear and it is very difficult to simplify it by linear approximations.

[*] In other words, the control variable of this experiment is RTRANS, and test variables are SURPLUSG, (+/S), and WH. Model version 2.0 was simulated 6 years with R1982.91 database and MSTART900 modification function (calibrated parameter values). S is a vector of firms' sales. In APL language +/S means the sum of S vector, i.e., total manufacturing sales.

Table 3.9 Results of symmetry and linearity tests (The RESULTS variable)

```
CONTROL VARIABLE :  RTRANS
VARIABLE TESTED  :  SURPLUSG
Symmetry statistics
Year   10.0%     1.0%
      -10.0%    -1.0%
   1  1.807    0.095
   2  1.86     0.638
   3  1.911    1.274
   4  1.933    1.434
   5  1.951    1.332
   6  1.963    1.355
Linearity statistics
Year  -10.0%    10.0%
       -1.0%     1.0%
   1  1.069    1.473
   2  0.763    1.5
   3  0.548    1.634
   4  0.595    1.654
   5  0.815    1.75
   6  0.99     1.73

VARIABLE TESTED :  (+/S)
Symmetry statistics
Year   10.0%     1.0%
      -10.0%    -1.0%
   1   2         2
   2   1.996     1.99
   3   1.983     1.994
   4   1.981     1.996
   5   1.946     1.997
   6   1.934     1.997
Linearity statistics
Year  -10.0%    10.0%
       -1.0%     1.0%
   1   1.281     1.591
   2   1.622     1.575
   3   1.763     1.632
   4   1.829     1.692
   5   1.748     1.761
   6   1.68      1.788
```

Table 3.9 Continued

```
VARIABLE TESTED :  WH
Symmetry statistics
Year   10.0%    1.0%
      -10.0%   -1.0%
   1  2        2
   2  2        2
   3  2        2
   4  2        2
   5  1.961    1.994
   6  1.952    1.995
Linearity statistics
Year  -10.0%   10.0%
       -1.0%    1.0%
   1  1.007    1.712
   2  1.075    1.625
   3  1.045    1.602
   4  1.052    1.701
   5  1.094    1.714
   6  1.074    1.783
```

Figure 3.3 Cumulative distance distributions

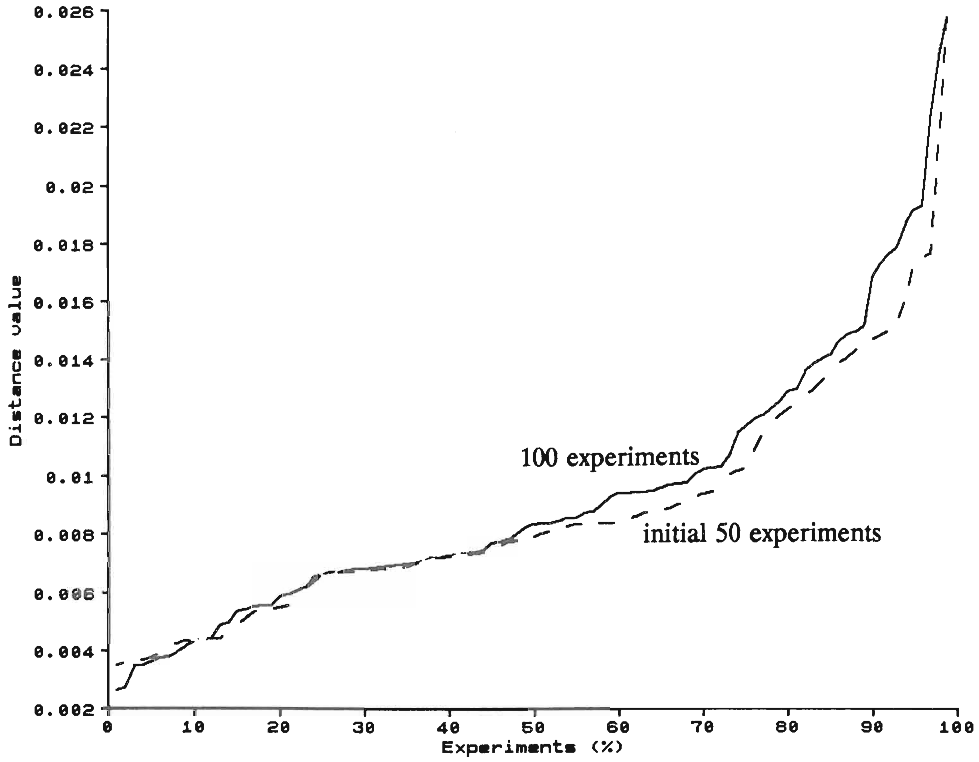


Figure 3.4 Local random search (Distance values)

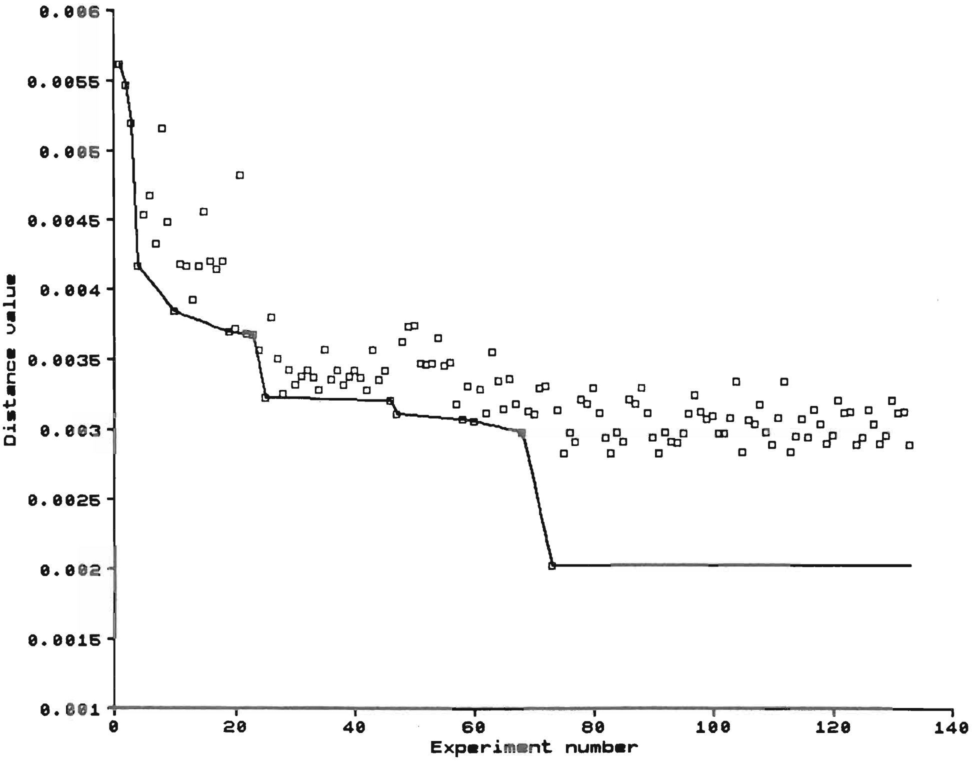
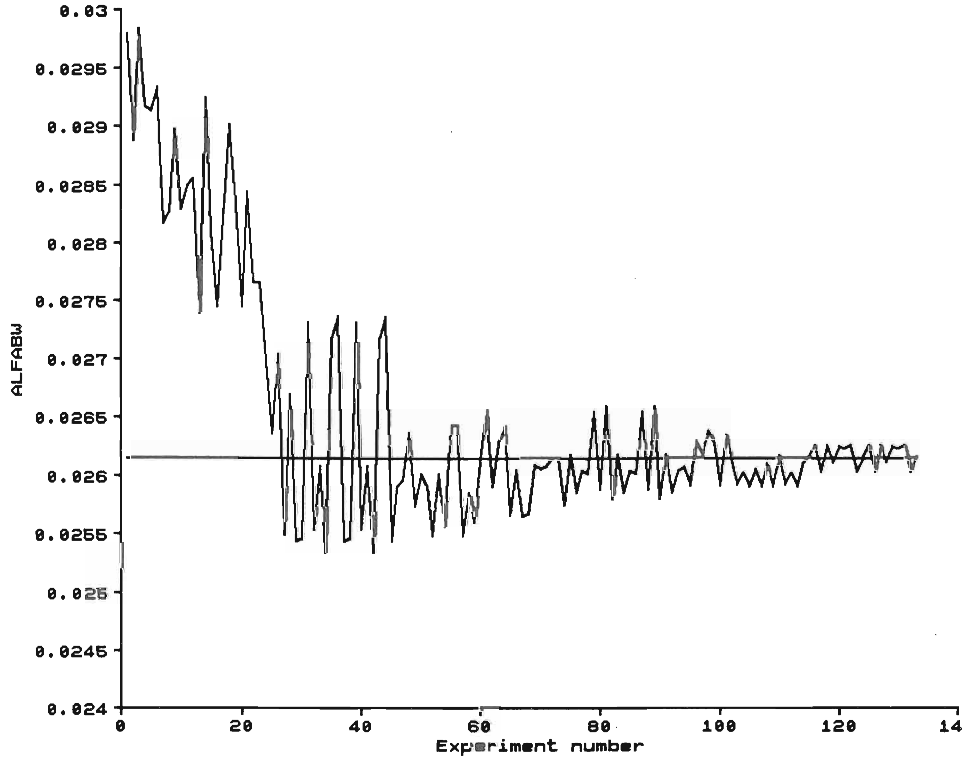


Figure 3.5 Local random search
(Trials of ALFABW parameter)



Note: Horizontal line shows the calibrated value of ALFABW parameter.

Figure 3.6 Interest rate

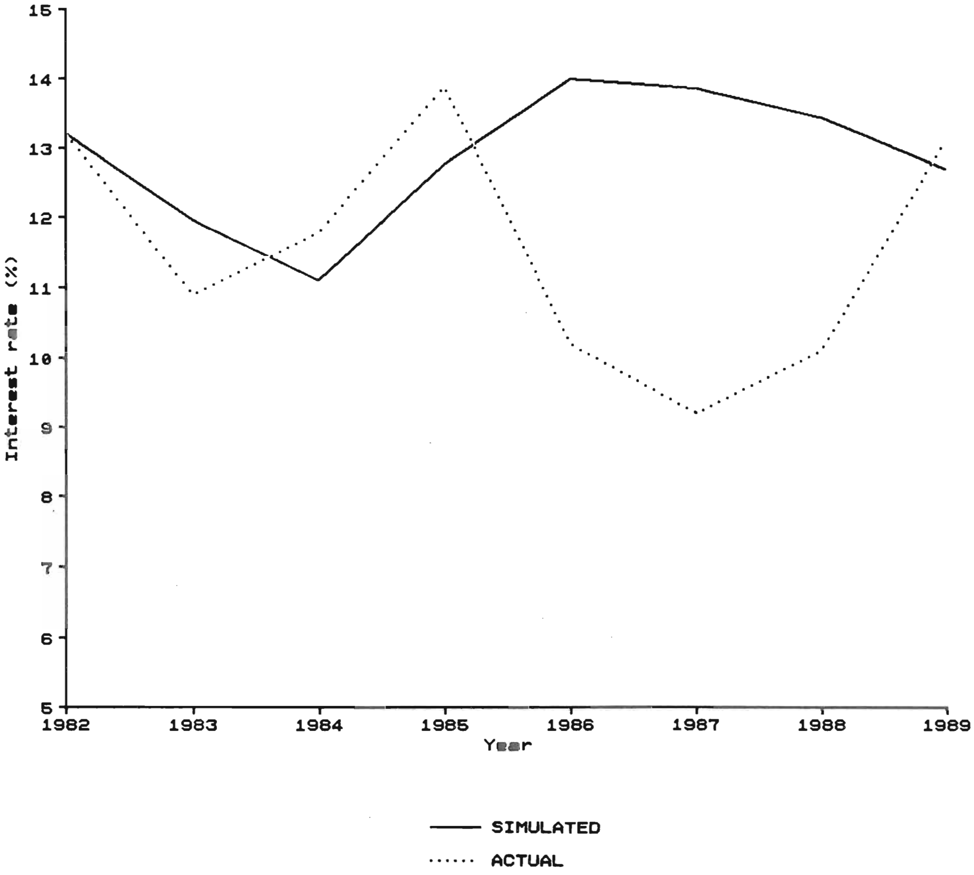


Figure 3.7 Annual growth rate of manufacturing employment

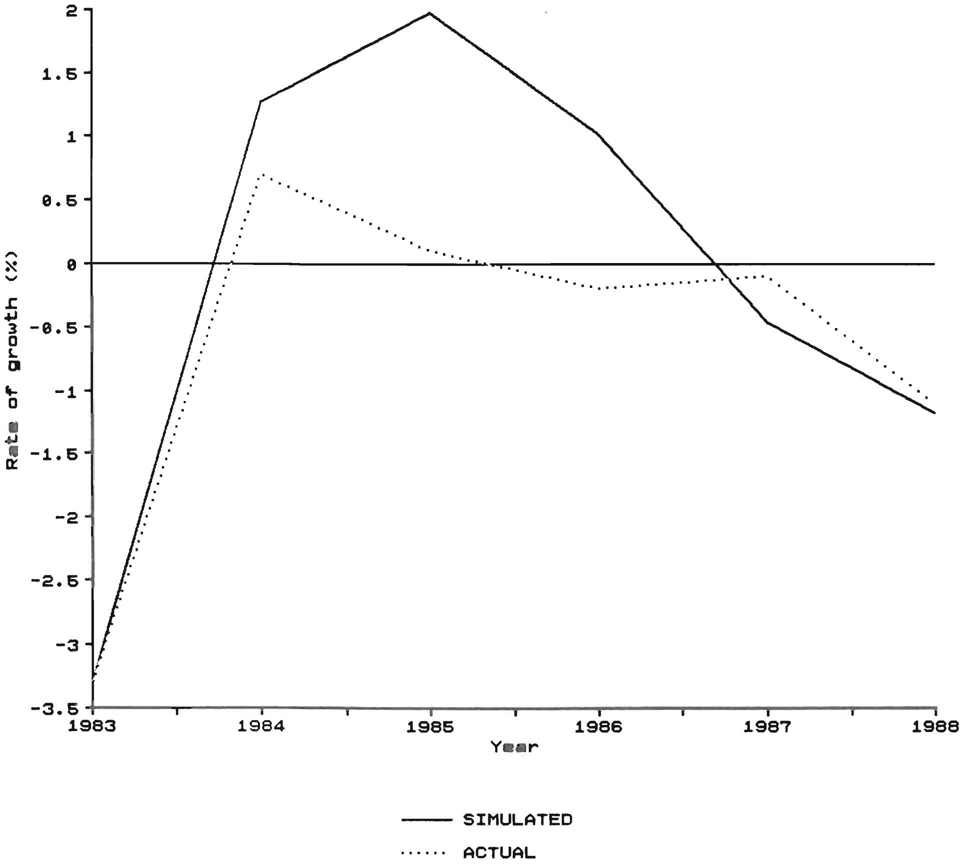


Figure 3.8 Annual growth rate of manufacturing output

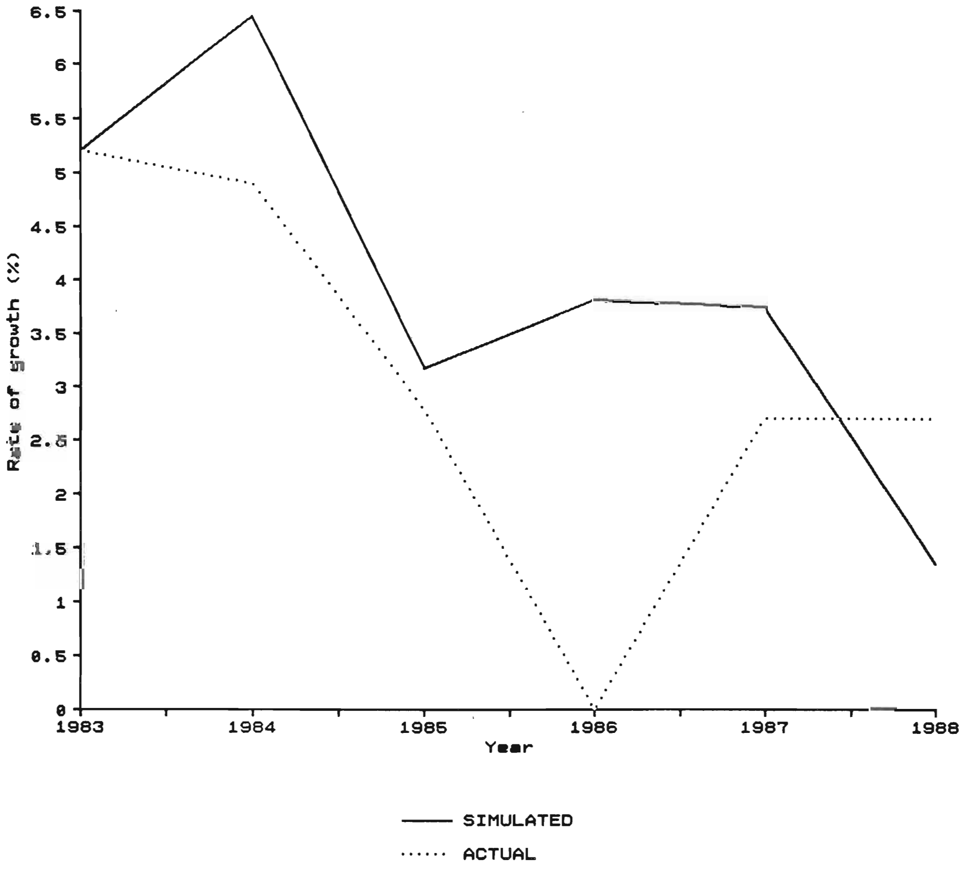


Figure 3.9 Annual growth rate of manufacturing prices

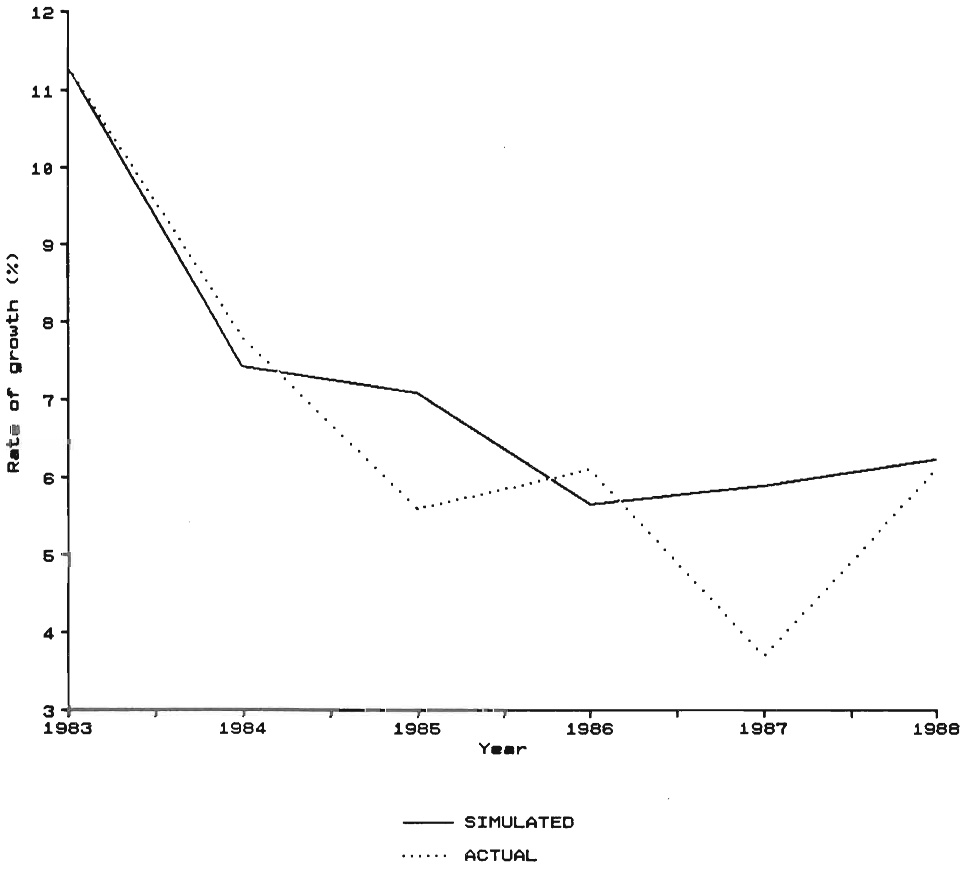


Figure 3.10 Annual growth rate of GDP

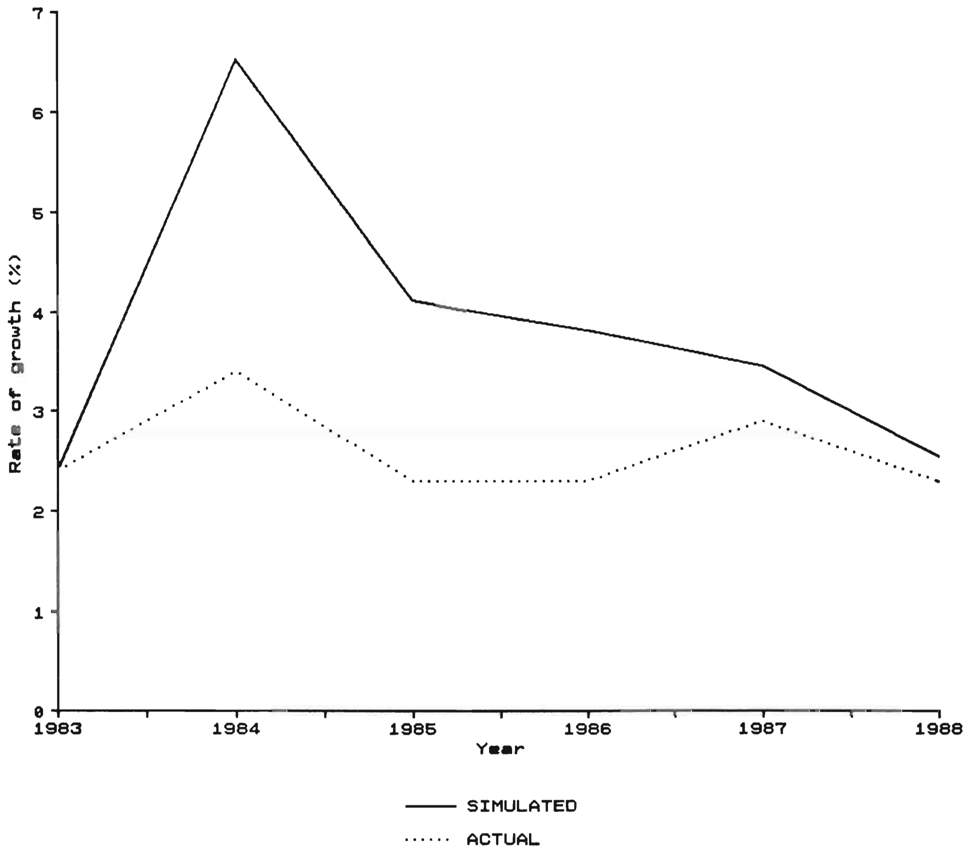


Figure 3.11 Indices of labor productivity in manufacturing

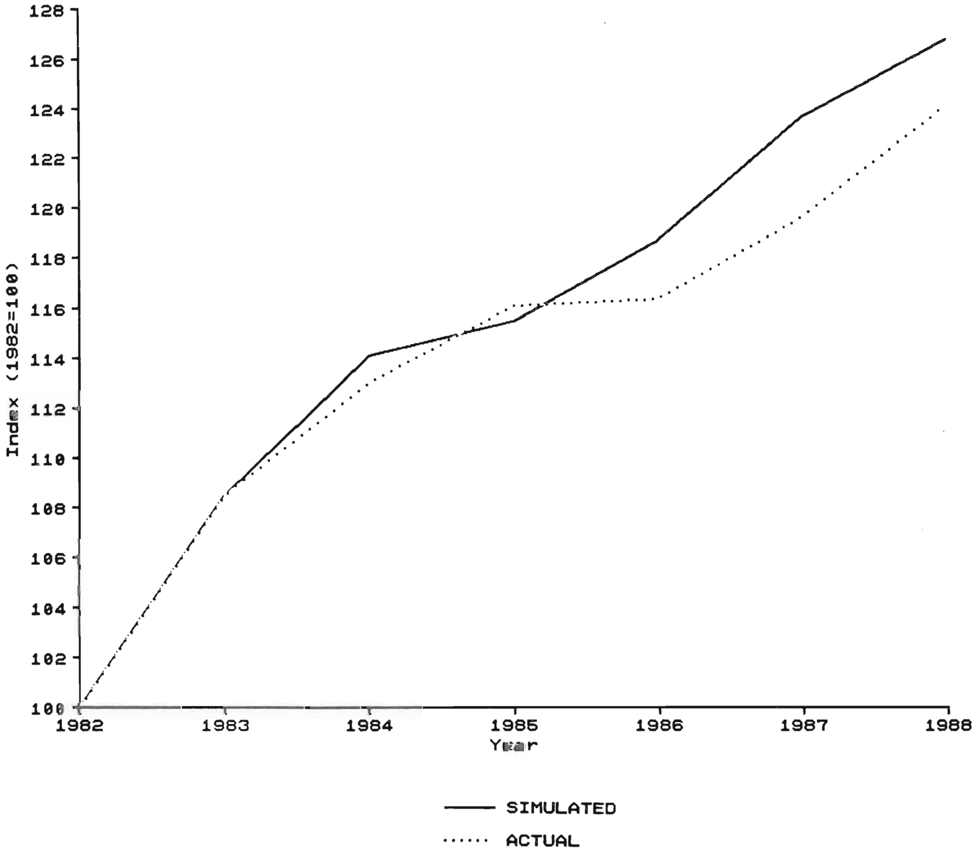
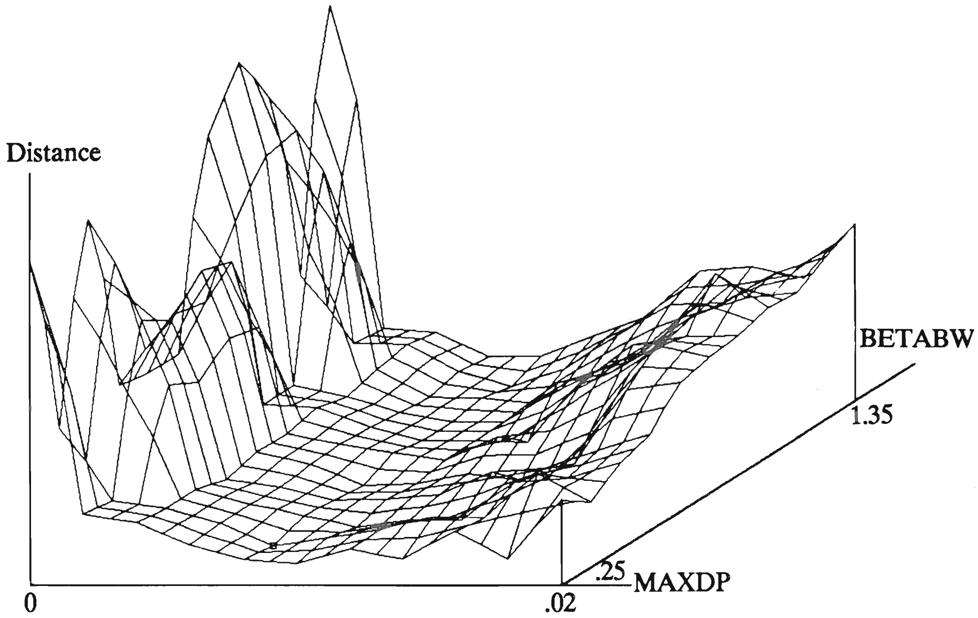


Figure 3.12 Distance values as a function of MAXDP and BETABW parameters



REFERENCES

There are many papers on the MOSES model, and the studies based on the simulations of this model. The following list contains only those publications that are essential to understanding the structure of the model. Albrecht *et al.* (1989) contains the complete list of the model's code. The modifications to the code after the transfer of the model from the mainframe computer to a personal computer are explained in the Appendix F.

Anderson, R.L. (1953), "Recent Advances in Finding Best Operating Conditions", *Journal of the American Statistical Association* (48): 789-798.

Bergholm, F. (1989), *MOSES Handbook*. Stockholm: IUI.

Boender, C.G.E., Rinnooy, A.H.G., Timmer, G.T. and Stougie, L. (1982), "A Stochastic Method for Global Optimization", *Mathematical Programming* (22): 125-140.

Brooks, S.H. (1958), "A Discussion of Random Methods for Seeking Maxima", *Operations Research* (6): 244-251.

Carlsson, B. and Taymaz, E. (1991), "The Role of Technological Progress and Economic Competence in Economic Growth." Paper presented at the conference on *Capital, its Value, its Rate of Return, and its Productivity*, March 5-6, 1991, Stockholm.

Dyadic Systems Limited (1986), *Dyalog APL User Guide*. Alton, Hampshire: DSL.

Eliasson, G. (1977), "Competition and Market Processes in a Simulation Model of the Swedish Economy", *American Economic Review* (67): 277-281.

Eliasson, G. ed. (1978), *A Micro-to-Macro Model of the Swedish Economy: Papers on the Swedish Model from the Symposium on Micro Simulation Methods in Stockholm Sep.19-22, 1977*. Stockholm: IUI.

Eliasson, G. (1985), *The Firm and Financial Markets in the Swedish Micro-to-Macro Model*. Stockholm: IUI.

Eliasson, G. and Olavi, G. (1978), "Stepwise Parameter Estimation of a Micro Simulation Model", in Eliasson (1978): 95-103.

Fletcher, R. (1987), *Practical Methods of Optimization*, New York: John Wiley and Sons.

de Haan, L. (1981), "Estimation of the Minimum of a Function Using Order Statistics", *Journal of the American Statistical Association* (76): 467-469 (Corrigendum, 78, 1008).

Klevmarken, A. (1978), "On Estimation and Other Problems of Statistical Inference in the Micro Simulation Approach", in Eliasson (1978): 87-93.

Kuh, E., Neese, J.W. and Hollinger, P. (1985), *Structural Sensitivity in Econometric Models*. New York: John Wiley & Sons.

Lucas, R.E. (1976), "Econometric Policy Evaluation: A Critique", in K.Bruner and A.H.Meltzer (eds.), *The Phillips Curve and Labor Markets* (Vol 1 of a Supplementary Series to the *Journal of Monetary Economics*): 19-46.

Petkovski, D.B. (1987), "Time-Domain Robustness Criteria for Large-Scale Economic Systems", *Journal of Economic Dynamics and Control* (11): 249-254.

Smith, R.L. (1987), "Estimating Tails of Probability Distributions", *Annals of Statistics* (15): 1174-1207.

Solis, F.J. and Wets, R.J-B. (1981), "Minimization by Random Search Techniques", *Mathematics of Operations Research* (6): 19-30.

Veall, M.R. (1990), "Testing for a Global Maximum in an Econometric Context", *Econometrics* (58): 1459-1465.

Zellner, A. and Peck, S.C. (1973), "Simulation and Experiments with a Quarterly Macroeconomic Model of the U.S. Economy", in A.A.Powell and R.A.Williams (eds.), *Econometric Studies of Macro and Monetary Relations* (Amsterdam: North Holland): 149-168.

APPENDICES

A. IN CASE OF TROUBLE

An experiment can be halted during simulation because of the following reasons.

1) The STOP▴HERE function can be called by the user at any time to stop simulation, for example, for manual data entry.

This function can also be used in a number of functions to remind the user about the possible errors made in editing model functions and the modification function. For example, when two firms to be merged are not in the same industry, this function stops the execution of the model, and sends the following message:

```
**********************************
FIRMS IN THE MERGE▴WITH FUNCTION
ARE NOT IN THE SAME INDUSTRY
FUNCTION(S): MERGE▴WITH, ...
TYPE  <→[]LC>  TO CONTINUE
**********************************
```

At this point, you can either call the MERGE▴WITH function properly and continue to simulation by typing

→[]LC RETURN

or continue the simulation without merging two firms.

2) The ENS function is used in a number of functions to ensure i) the "logical" consistency of a model variable, and ii) the proper execution of commands.

The APL system command, the system indicator, can be used to locate the source of the problem. This command identifies those operations which are suspended or pendent for each suspension. Type

)SI RETURN

to display the list that consists of a line for each suspended or pendent operation beginning with the most recently suspended function.

In the first case, the variable does not have the "logical", expected value. For example, in the FIRMENTRY2 function, the ENS function is used to ensure that total labor demand of new firms does not exceed the level of unemployment since it is implicitly assumed that new firms hire their labor from the ranks of unemployed. When their demand becomes larger than the unemployment level, i.e., when the level of unemployment becomes negative after the new entries, the ENS function stops simulation and gives an error message. Since the actual level of firms' employment will be determined in the labor market, this "temporary" negative unemployment may not be considered a serious problem, and the simulation can be continued without any changes by typing

```
→[]LC RETURN
```

In the second case, the experiment is halted because of improper or impossible execution of a command. Therefore, the cause of error should be corrected before the experiment is resumed.

3) The most serious reason for the suspension of the experiment is due to errors or misspecifications in the model and the dataset. If the experiment is interrupted for some reasons, an APL event message will be shown on the screen. You can use the system indicator to locate the functions that are suspended of pendent (for the explanation of event messages, see Dyalog APL "User Guide", Section 11).

A possible source of an error is the so-called "domain error" that occurs, for example, when a number is divided by zero.

The quarterly profit margin in the FINALQPQSQM function is calculated as a fraction of quarterly net sales. Thus, if the quarterly net sales of a firm for any reason is equal to zero a domain error occurs. In this case, either the system variable that determines the division method or the definition of the variable can be changed.[*]

Although no new type of error that halts the simulation has been seen in a large number of experiments, there may still be some unknown

[*] "The value of the system variable []DIV determines how division by zero is to be treated. If []DIV=0, division by 0 produces a domain error except that the special case of 0/0 returns 1. If []DIV=1, division by 0 returns 0.

[]DIV may be assigned the value of 0 or 1. The value in a clear workspace is 0."

misspecifications in the model that reveal themselves in a specific experiment design. Such unknown misspecifications may take a long time to identify and correct since the model should be understood thoroughly for corrections.

B. FUNCTIONS IN THE MOSES.HELP WORKSPACE

The MOSES.HELP function contains a number of functions that can be used to analyze the model. These functions and their use can be summarized as follows.

B ← BACK▲TREE A

A is a function name, B is a vector of names that (eventually) calls the function A.

B ← CLEAR▲BLANK A

A is a vector of names, B is the same vector of names where all blanks in names are removed.

A CORR B

A and B are variable vectors. CORR returns the correlation coefficient and the t-value.

B ← F▲ID A

A is a number less than the number of firms in the current dataset. B is the firm code of the A^{th} firm.

B ← F▲PL A

A is a firm code. B is that firm's position in micro (firm) variable in the model. (For example, if A denotes the firm 3.12 -12^{th} in the third sector-, and B is found to be 25 by the F▲PL function, then Firm 3.12's employment level is saved in the 25^{th} element of the labor vector, L.) Thus, A = F▲ID F▲PL A.

[F] FNGREP A

A is a string, F is a matrix of function names (each row of F is a function name). This function returns the names of all those functions in the F matrix that contain the string A. If F is not specified, all functions in the active workspace are searched.

B ← FOR▴TREE A

B is a function name, A is a vector of function names that are eventually called by A. (Those functions whose names are within single quotation marks cannot be found by this function.)

B ← IN▴FUNCTIONS A

A is a function name, B is a vector of function names that are directly called by A.

B ← IN▴VARIABLES A

A is a function name, B is a vector of variable names used in A.

KEEP▴NAMES A

A is a vector of names. This function deletes all variables and functions in the current workspace other than those in A.

LIST

This function lists all the variables and their ranks in the active workspace.

B ← [F] NAME▴USED A

A is a vector of strings (names), F is a matrix of function names, and B is a vector of function names in F that contain any one of the strings in A. If F is not specified, all functions in the active workspace are searched.

[F] PRINT▴ALL▴F A

A is a number that specifies print width (in number of columns), F is a matrix of function names (each row of F is a function name). This function prints the functions whose names are found in F. If F is not specified, all functions in the current workspace are printed.

'REG' REG 'X_1 X_2 ... X_n'

This function is used for regression analysis. It calculates R^2, the estimated coefficients, and t-values. Note that variable names together should be within single quotation marks.

C ← A SALTER B

This function prepares variables for a salter curve. A is the variable on the vertical axis, B is the variable whose cumulative value will be shown on the horizontal axis. C is a three-element vector whose first element contains the ordered pairs of the variable A, the second element the corresponding cumulative pairs of the variable B, and the third element the vector of 0's whose rank is equal to the rank of the first (and second) element. Use C[1] and C[2] to show the salter curve, and C[3] to connect two or more salter curves to be shown on the same chart.

SET▲MONITOR

This function sets monitor for all functions. It is useful to determine various attributes of functions used in an experiment. To use the SET▲MONITOR function, copy it to the MOSES.PC workspace and call in the MSTART*xx* function. At the end of a simulation, type

[]MONITOR '*functionname*' RETURN

This will return a four-element vector where the first element denotes the line number, the second element the number of times the line was executed, the third element the CPU time in milliseconds, and the fourth element the elapsed time in milliseconds.

scale Y▲RFIRM *firmcode*

scale and *firmcode* are defined same as in the case of the Y▲R▲FIRM function (see Section 1.5 on output reporting). This function creates a table called YEARLY▲RFIRM▲*firmcode*. The form of this table is exactly same as those created by the Y▲R▲FIRM function, but it contains real data for that firm, if data are available in the Planning Surveys. If the *firmcode* belongs to a synthetic firm, an error message is shown.

C. FUNCTIONS IN THE MOSES.GRAPH WORKSPACE

Six functions are available in the MOSES.GRAPH workspace: PLOT, CHART, BARCHART, PIECHART, SHOWFUN, and SHOWREAL . There are a number of functions in this workspace that are used by those main functions. Therefore, it is better to use this workspace after saving the original output workspace in order not to increase the size of the workspace.

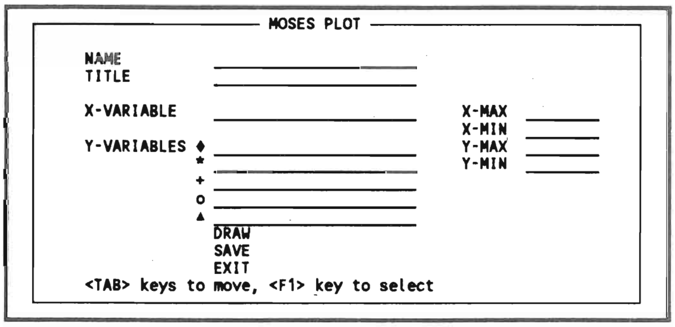
These graphics functions can be accessed by copying the MOSES.GRAPH workspace into the active workspace. To do this, type

) COPY MOSES.GRAPH RETURN

After copying the MOSES.GRAPH workspace, the graphics functions can be used. For example, to use the PIECHART, simply type

PIECHART RETURN

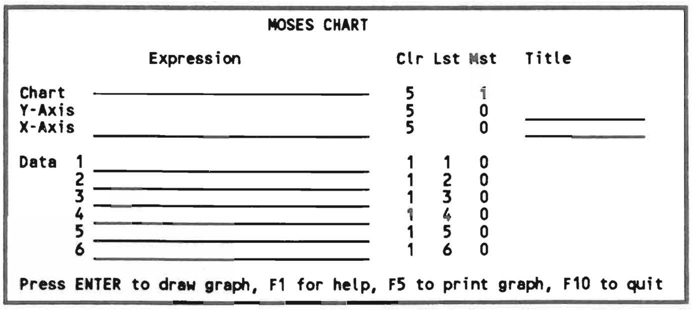
The PLOT function does not use the SCO Graphics^{TM} run-time system. It is based on Dyalog-APL's full-screen applications of the []SM and []SR functions, and creates a simple plot of up to six variables. The plot that is shown on the screen can also be saved as a 25x80 text matrix.



<TAB> keys to move, <F1> key to select

When the PLOT function is invoked, the following menu appears. This function creates a plot of up to five variables. Enter the name of the variables against the Y-variables. At least one of the Y-variables should be entered in the first line. All other entries in the menu are optional. To show the plot on the screen, move the cursor on "DRAW" and press the **F1** key. "SAVE" creates a text matrix of the plot in a variable defined in the "NAME" entry.

The CHART function creates a chart of up to six variables. (The number of variables that can be plotted can be increased by modifying the CHART function.) When the CHART function is invoked, the following menu appears on the screen.



```
                          MOSES CHART

              Expression                 Clr Lst Mst   Title

Chart    ______________________________   5       1
Y-Axis                                    5       0   _________
X-Axis   ______________________________   5       0   _________

Data  1  ______________________________   1   1   0
      2  ______________________________   1   2   0
      3  ______________________________   1   3   0
      4  ______________________________   1   4   0
      5  ______________________________   1   5   0
      6  ______________________________   1   6   0

Press ENTER to draw graph, F1 for help, F5 to print graph, F10 to quit
```

You can enter any valid APL expression against the X-axis and data expressions. If you do not enter any expression against the X-axis, a line chart will be shown. In other words, the X variable will be equal to the index of the rank of the Y-variables.

Enter the chart and axes titles under the titles section. The column "Clr" defines the color of the chart (borders, chart title, etc.), axes (axes lines, labels, etc.), and data (charts). "Lst" defines line types (1 for continues line, 2 for broken line, 3 for dotted line, etc.) "Mst" defines the chart style (0 for no labels, 1 for line labels), grids on the horizontal (Y-axis) and vertical (X-axis) axes, and marker styles for the data. If the chart style is set to 1, the first five characters of the data expressions are used in labels.

To get on line help, press the **F1** key. If you want to show the chart on the screen, press the RETURN key. To go back to the menu, press the

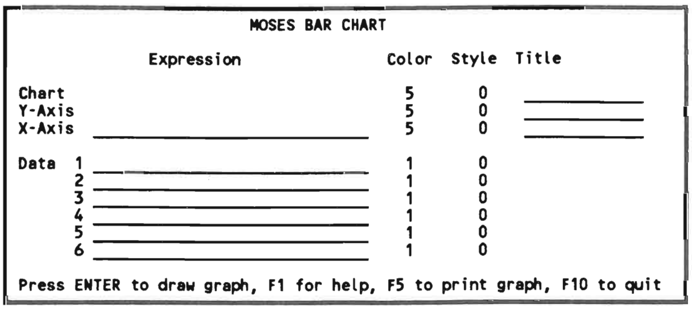
RETURN key or any one of the function keys when the chart is shown on the screen. To go back to the APL interpreter, i.e., to exit from the CHART function, press the **F10** key. To print the chart, press **F5**. When you press the **F5** key, you will be asked if you want to use a plotter or a printer. Type

P RETURN , for the plotter, and
R RETURN , for the printer.

The default options for the plotter and printer are the HP plotter and the HP laserjet printer using the first parallel port. If you use a different configuration, change the APL environmental parameters in the script /usr/bin/apl.

If you use a plotter, you will be asked to change the paper and press the RETURN key. Printing of the chart may take up to a few minutes depending on the complexity of the chart. You cannot use any other function during printing. Note also that the computer may be locked out without any apparent reason because of the bugs in the SCO Graphics package although several checks for possible sources of errors are included into these functions. We have had no problems so far using the HP printer. Therefore, be sure to save the output file before using any one of the graphics functions.

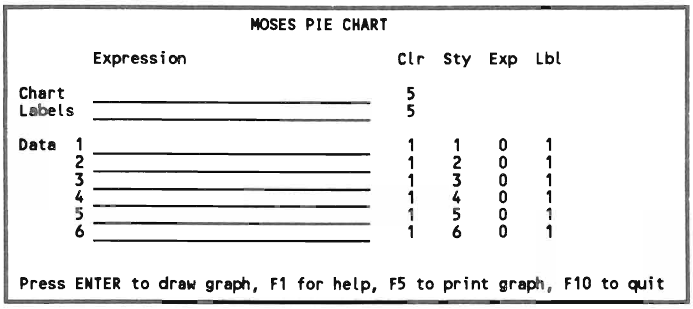
The following menu appears on the screen when the BARCHART function is invoked.



```
                           MOSES BAR CHART

                   Expression              Color  Style  Title

Chart                                        5      0     __________
Y-Axis                                       5      0     __________
X-Axis   _____________________________       5      0     __________

Data  1  _____________________________       1      0
      2  _____________________________       1      0
      3  _____________________________       1      0
      4  _____________________________       1      0
      5  _____________________________       1      0
      6  _____________________________       1      0

Press ENTER to draw graph, F1 for help, F5 to print graph, F10 to quit
```

This function is used exactly as the CHART function. The only difference is that there are no styles for markers which are not used in bar charts, and the chart style now defines a regular bar chart (enter 0), or a stacked bar chart (enter 1).

The PIECHART function prepares a pie chart. This function can simultaneously show at most three pie charts. When you type **PIECHART**, the following menu is displayed.



```
                          MOSES PIE CHART

        Expression                         Clr  Sty  Exp  Lbl

Chart   ______________________________      5
Labels  ______________________________      5

Data  1 ______________________________      1    1    0    1
      2 ______________________________      1    2    0    1
      3 ______________________________      1    3    0    1
      4 ______________________________      1    4    0    1
      5 ______________________________      1    5    0    1
      6 ______________________________      1    6    0    1

Press ENTER to draw graph, F1 for help, F5 to print graph, F10 to quit
```

Enter the title of the chart under the "Expression" column. The name of the variable that contains the pie labels should be entered against the "Labels" line. The "Sty" column defines the interior style of each slice (enter a number between 0 and 9). The "Exp" column defines if the slice is exploded (0 for not, 1 for exploded). The "Lbl" column defines if the percentage shares are to be shown on the slices (0 for not, 1 for otherwise).

Since this function can show at most three pie charts, if the variable names of expressions entered against the data lines contain more than three variables, only the first three of them are used. The labels of slices to be displayed under the bottom of the pie chart are taken from the first five characters of the data expressions (or variable names).

The SHOWFUN graphics function shows an animated time series chart of the production functions of industries or firms. The output tables should be available for this function, i.e., the YEARLY∆INDUSTRY∆TOTAL table for the manufacturing industry, YEARLY∆MARKET*x* table for each sector, and YEARLY∆FIRM∆*firmcode* tables for firms. This function is used as follows.

SHOWFUN '*firmcode*' RETURN

For the manufacturing industry, and the raw materials, intermediate goods, capital goods, and consumer goods sectors, enter 0, 1, 2, 3, and 4, respectively, for the *firmcode*.

This function displays the production function for each year starting from the initial year of the simulation experiment, and shows the actual position of the firm/industry on the production function. Press the RETURN key or any one of the function keys to move from one year to the next year. For the final year, a message will be shown on the screen. The following figure shows the production function of the manufacturing industry for various years.

The SHOWREAL graphics function creates the same type of graphics as the SHOWFUN function for real firms, if their data are available in the MICRO.DBASE workspace.

D. INSTALLING MOSES ON PC

The current PC version of the model is implemented by using a Dyalog-APL interpreter with SCO Xenix Operating System V on a 386 personal computer. (Although the model can be used with a 286 PC, a 386 PC with a math-coprocessor is strongly recommended.) Xenix operating system and APL should be ready before the installation of MOSES. If you plan to use the MOSES.GRAPH functions, SCO Graphics™ should also be installed. Note that Xenix operating systems can be used together with DOS system on the same computer and it is possible to transfer files from Xenix to DOS or vice versa.

After installing Xenix and APL[*], create a directory for the MOSES model and change the current directory as follows.

```
# mkdir /moses RETURN
# cd /moses RETURN
```

Recall that small and capital letters are considered different in Xenix. Then insert the MOSES diskette into *first* floppy drive[**] and type

*) For the installation of Xenix and APL, see their manuals. These programs can be obtained from the following companies:

Xenix operating system and SCO Graphics package:
The Santa Cruz Operation
400 Encinal Street, P.O.Box 1900
Santa Cruz, CA 95061
USA
Tel: 1 - 800 626 8649
The Santa Cruz Operation
P.O.Box 4YN, 18 Noel Street
London W1A 4YN
United Kingdom

Dylaog APL:
Dyadic Systems Limited
Park House, The High Street
Alton, Hampshire
United Kingdom
Tel: 44 - 420 87024

**) If 3.5 inch diskette drive is your second drive (drive B in DOS) then use the following command.

```
# tar xvf /dev/fd1135ds18 RETURN
# tar xvf /dev/fd0135ds18 RETURN
```

to copy all files in the floppy diskette to /moses directory. If there are more than one diskette, repeat the last step for all diskettes. To start the MOSES program, type

```
# apl moses RETURN
```

Information about the APL environment are stored in a text file, /usr/bin/apl. Add the following line into /usr/bin/apl file[*] to use APL keyboard that allows to use, for example, underlined characters by "Alt" key.

```
mapkey /usr/dyalog/apl.iso_2.3
```

The maximum workspace size is also defined in this file. The default size is 1024 KBytes. If you want to use larger workspaces, change the line for maximum workspace size (MAXWS).

There are a number of functions in the model that requires an HP laser printer for graphics. To use these functions properly, add following two lines into /usr/bin/apl file.

```
:{RINTER=laserjet}          ;export RINTER
:{laserjet=/dev/lp0i}       ;export laserjet
```

These lines specifies an HP laser printer connected to the first parallel port. If you want to use another printer for graphics, or another port, use the appropriate definitions. (For details, see SCO Graphics™ Installation Notes.)

The SCO Graphics™ program is designed for computers using EGA (Enhanced Graphics Adaptor) graphics card. If your computer has a VGA (Video Graphics Array) card, you have to use VGA drive for graphics. To do so, enter the following two commands (after installing SCO Graphics™ program) when you are in Xenix.

If you do not use regular MOSES program which is available in 3.5 inch, 1.44 KByte diskettes, use appropriate **tar** command for that diskette. (For details of using the **tar** command, see Xenix User's Guide.)

[*] Use Xenix's full-screen editor, vi, to modify text files. For editing commands, see Xenix Users's Guide.

```
# rm /dev/ega RETURN
# ln /dev/vga /dev/ega RETURN
```

E. DIFFERENCES BETWEEN THE PC AND MAINFRAME VERSIONS

There are a number of differences between those workspaces that were in DEC-10 and those on PC. Some of those differences are related with the APL program and machine characteristics. They can be summarized as follows.

1) The monadic ϵ function in DEC-10 APL means "execute" whereas it means "type" in Dyalog-APL. Therefore all monadic ϵ functions should be changed to Dyalog-APL's execute function. There is a function called FNGREP that finds a given string in all functions in the current workspace.

2) The MAXCORE command in DEC-10 APL is used to set the amount of allocated memory for the APL program. Since it is no longer necessary and does not have any meaning in Dyalog-APL, this command in all functions should be deleted.

3) In DEC-10 APL, some negative variables have the suffix "-". They are converted to "¯" sign during the transfer process by the function VCR.

4) Some commands (namely, WSID and ERASE commands) should be changed to their corresponding syntax for Dyalog-APL.

5) During the transfer process, some blank spaces between single quotation marks (') have been deleted. This may create a problem in some cases (especially in the transcription functions). Those function that were affected by this problem were modified manually and the model runs without any problem so far.

6) The []PW command is used in a number of functions to set up the print width to 120 columns. Since this is not convenient for the PC, this command has been deleted form the PREPARE∆RUN function.

After the transfer of APL workspaces, some functions have been changed. These changes can be summarized as follows.

1) A new workspace called "moses" have been created. This workspace contains functions and variables to start a simulation. It copies all necessary functions into the current workspace. The UPDATEMOSES function in the MOSES workspace is also modified to work with this new function. The MOSES workspace was renamed MOSES.PC.

2) The results of a simulation were being saved in text matrices in the mainframe model. But this method of keeping track of data is not convenient to use when output variables are to be processed further (graphics, etc.). Therefore all transcription functions (except Y▲R▲DISTRIBUTION and quarterly data functions) in the MOSES.PC workspace have been modified so that the results of a simulation are stored in arrays in numerical form. This form of data allows us to use the results as a variable. Accordingly, PRINT▲OUT and PRINT functions have also been changed. New functions, SPRINT, PPRINT, and FPRINT were written to send output tables to the screen, printer, and a file, respectively. Moreover, the calculation of the QTOP variable in the Y▲MARKET function has been changed to obtain comparable data with respect to those collected in the Y▲INDUSTRY▲TOTAL and Y▲FIRM functions.

3) Functions related to firm entry were considerably modified. A few lines that are used in the modification functions of the entry experiments (MSTART3*x*) were added into the function START▲ENT▲MOD since they are required in any simulation with the entry option. START▲ENT2 function was also modified to allow probabilistic firm entry. START▲ENT1 function was modified in accordance with these changes. In the FIRMENTRY2 function, the comment mark before the definition of FIRMCHARC variable that collects the data on the characteristics of new firms was removed. It has been modified to have an array structure (instead of a text matrix form, as explained before). Consequently, there is no difference between the RUNEXP and RUNENTRY functions. Thus, hereafter, all experiments (with or without the entry option) can be started with the RUNEXP function. (Additional changes have been made in version

2.0. For these changes, see the VERSION20 function in Appendix F.)

4) Some lines that were apparently experiment-specific (e.g., collection of data on some firms, etc.) were deleted in the function RUNEXP.

5) The ENS function was modified so that it stops the execution of the program when an "ERROR" message is printed. The execution of the program is proceeded by the APL command →[]LC. (In the previous form of the function, it only prints "ERROR" message on the screen.)

6) Since the "updates" are well documented, the modifications made by these update functions have been saved in the MOSES.PC workspace, and the UPDATEMOSES function is modified accordingly. Only those changes made by the PRICE∆CHANGES function have not been saved.

7) The INIT file was renamed MOSES.INIT. The START function in this file was modified to use a menu. Those functions that copy macro and micro datasets were modified accordingly.

8) The macro and micro datasets were renamed as MAC*xx* and MIC*xx*, respectively, where *xx* is the dataset year.

9) ESTABLISHMENTS functions in the INIT workspace were divided into two functions because the PC version of APL cannot handle very long functions.

10) Data-specific variable names in the INIT workspace were changed. (IO76, IOCOEFF76, SALES76 (SALES82), HH76, and SCB82∆QINV were renamed as IO∆∆, IOCOEFF∆∆, SALES∆∆, HH∆∆, and SCB∆∆∆QINV, respectively.)

11) The OUTPUT∆OPERATIONS function in the INIT workspace was simplified considerably.

12) The random link variable, []RL, in the INIT workspace was changed to the default values of Dyalog-APL, because the mainframe and PC versions of APL may use different formulae for random-number generation.

13) A new variable, N∆REAL∆FIRMS, is introduced in ESTABLISHMENTS functions in the INIT workspace. This variable specifies

the number of real firms in the dataset, and is used by the AVG∆TOP function of the model. Accordingly, its name was added to VARIABLEGRUPP3 variable in the VLISTS workspace.

14) The RANDOMIZE function in the INIT workspace was modified so that it now allows to define of more than 100 synthetic firms. (The previous specification does not allow more than 100 synthetic firms.)

F. MOSES VERSION 2.0 FUNCTIONS

VERSION20
MSTART900
ADJUSTPRICES PC
COMPUTE IMPORTS PC
DOMESTIC MARKET PC
DOMESTIC RESULT PC
ENTER NEWFIRMS
FINALQPQSQM PC
FIRMENTRY PC
GOVERNMENT ACCOUNTING PC
MARKET CONFRONT PC
MARKET ENTRANCE PC
MDIFY EXP
NATIONAL ACCOUNTING PC
OBSOLETE K
QUARTERLY EXP PC
QUITS
START ENTRY PC
TARG SEARCH PC
TARG SEARCH SHOW

```
********************************************************************************

VERSION20

[0]    VERSION20;PARAM
[1]   ⍝
[2]    →DOIF∧/THISΔYEAR≠76 82 ◊ VERSION2A ◊ →0
[3]    VERSION←'MOSES-PC 2.0'
[4]   ⍝*************************************
[5]   ⍝********  MOSES VERSION 2.0  ********
[6]   ⍝*************************************
[7]   ⍝
[8]   ⍝ DEFINE THE NUMBER OF REAL FIRMS
[9]    →DOIF 2≠⎕NC'NΔREALΔFIRMS' ◊ NΔREALΔFIRMS←(THISΔYEAR=76 82)/97 154
[10]   ⍝
[11]   ⍝ NEW TABLE FOR EXPECTATIONS
[12]   ⍝
[13]    YΔRΔEXPECTATIONS
[14]   ⍝
[15]   ⍝ NEW FOREIGN PRICES !
[16]   ⍝
[17]    →(THISΔYEAR≠82)/JMPR
[18]    →(2≠⎕NC'FIRMΔEXOΔQDPFOR')/JMPP
[19]    FIRMΔEXOΔQDPFOR[;12+⍳28]←FIRMΔEXOΔQDPFOR[;12+⍳28]+0.01
[20]    EXOΔQDPFOR←FIRMΔEXOΔQDPFOR
[21]    ⎕EX'FIRMΔEXOΔQDPFOR'
[22]    ⎕EX'EXOΔQDSUFOR'
[23]   ⍝
[24]    QPFOR←(1÷(1-(0.16×1÷4)))×(103.975 104.275 104.9875 104.3875)
[25]   ⍝
[26]   JMPP:
[27]   ⍝
```

```
[28]   ⍝ ********* PARAMETER CHANGES *********
[29]    TID←0
[30]    KSI←0.15
[31]    NITER←3
[32]    GAMMA←0.3
[33]    IOTA←0.5
[34]    SKREPA←75
[35]    RESDOWN←0.2
[36]    EXO∆TXC←EXO∆TXC÷2
[37]    RTD←1
[38]    TMIMP←7 7 7 7
[39]    TMX←7 7 7 7
[40]    ALFA3←ALFA4←0
[41]    SAVRAT←0.005÷4
[42]   JMPR:
[43]   ⍝
[44]   ⍝
[45]   ⍝
[46]   ⍝ MIN LABOUR IS .1
[47]   ⍝
[48]    'SOLVE'MODSUBST'QPLANL[I]⍵QPLANL[I]←0.1⌈(Y×QTOP[I]÷TEC[I])'
[49]   ⍝
[50]   ⍝ SET MTEC VALUES
[51]   ⍝
[52]    →((THIS∆YEAR≠76)∧THIS∆YEAR≠82)/JMPZ
[53]    MTEC[1]←1.1×⌈/(MARKET=1)/TEC
[54]    MTEC[2]←1.1×⌈/(MARKET=2)/TEC
[55]    MTEC[3]←1.1×⌈/(MARKET=3)/TEC
[56]    MTEC[4]←1.1×⌈/(MARKET=4)/TEC
[57]   JMPZ:
[58]   ⍝
[59]   ⍝ CHANGE BAD SPEC
[60]   ⍝
```

```
[61]     'INVFIN∆ADJUSTMENTS'MODSUBST'BAD←BAD+⍵BAD←0⌈BAD+(NW<0)-NW≥0'
[62]    ⍝
[63]    ⍝ USE NEW TARGET SEARCH FUNCTION
[64]    ⍝
[65]     'PRODPLAN'MODSUBST'TARG∆SEARCH⍵TARG∆SEARCH∆PC'
[66]     LOWER∆MHIST←0.8
[67]     LOW∆TARGET←0.5
[68]     PARAM←'LOWER∆MHIST' 'LOW∆TARGET'
[69]    ⍝
[70]    ⍝ CHANGE CHL SPEC
[71]    ⍝
[72]     'LABOUR∆SEARCH∆INPUT'MODSUBST'CHL←⍵CHL←QPLANL-L'
[73]    ⍝
[74]    ⍝ CHANGE THE PROBABILITY TO BE TARGETTED IN HIT FUNCTION
[75]    ⍝
[76]     'CONFRONT'MODSUBST'SKR←⍵SKR←(1÷W÷⌈/W) SKREPA×(⍴W)÷+/W÷⌈/W'
[77]    ⍝
[78]     DELAY∆INV←|DELAY∆INV
[79]    ⍝
[80]    ⍝ CHANGE QOPTSU SPEC
[81]    ⍝
[82]     'PLANQREVISE'MODSUBST'QOPTSU←⍵QOPTSU←0⌈QEXPSU×(QQ+STO)÷QEXPSU+STO+
(OPTSTO-STO)÷4×TMSTO'
[83]    ⍝
[84]    ⍝ ADD CHANGES FOR NULLIFIED FIRMS
[85]    ⍝
[86]     'Q∆EXO'MODADDLAST'NULL∆CURRENT←0 ⋄ NULL∆GOODS←4⍴0'
[87]     'NULLIFY'MODADDLINE'NULL∆CURRENT+←+/OUT/K2-BW' 1
[88]     'NULLIFY'MODADDLINE'NULL∆GOODS+←4↑(+⌿OUT⌿IMSTO)+OMEGA×(+/OUT/K1)÷+
/OMEGA×PT[MKT IN]' 2
[89]     'NULLIFY'MODADDLINE'NULL∆GOODS[OUT/MARKET]+←OUT/STO' 3
[90]     'HOUSEHOLD∆UPDATE'MODADD'WH←⍵∆NULLS←(0⌈NULL∆CURRENT++/NULL∆GOODS×Q
PDOM[MKT]×1-TXVA2)÷NH ⋄ WH+←∆NULLS ⋄ QSAVH+←∆NULLS×NH ⋄ ⎕EX ''∆NULLS'''
```

```
[91]    ⍝
[92]    ⍝ ADD CHANGES FOR NEW FIRMS
[93]    ⍝
[94]     'START∆ENTRY'MODADDLINE'START∆ENTRY∆PC MM∆ENTRY ⋄ →0' 1
[95]     'COMPUTE∆BUYING'MODADDLINE'QBUY[MKT IN;NEW∆FIRMS[1;]]+←⌽↑NEW∆FIRMS
[2;]×NEW∆FIRMS[4;]' 3
[96]     'COMPUTE∆BUYING'MODADD'QINVTOT←⍵QINVTOT+←+⌿↑NEW∆FIRMS[2;]×NEW∆FIRM
S[3;]'
[97]     'HOUSEHOLD∆UPDATE'MODADD'WH←⍵→DOIF NRS=1 ⋄ WH-←(+/×⌿NEW∆FIRMS[2 5;
])÷NH ⋄ QSAVH-←+/×⌿NEW∆FIRMS[2 5;]'
[98]     'QUARTER'MODADD'Q∆EXO⍵→DOIF NRS=2 ⋄ ENTER∆NEWFIRMS '
[99]     'INVFIN'MODSUBST'QDESCHBW←(0⌈1⍵QDESINVP←(0⌈1-ELINV×UTREF-QQ÷QFR1 L
)×K1×ALFABW+BETABW×QDPK+(QRR-RIF)÷4'
[100]     'INVFIN'MODADD'QDESINVP←(0⌈1⍵QDESCHBW←QDESINVP+QDESCHK2-QCASH'
[101]     'INVFIN'MODADD'QDESCHBW←(BW⍵QDESCHBW←QDESCHBW-(0>BW+QDESCHBW)×BW+
QDESCHBW'
[102]     'INVFIN'MODSUBST'QDESCHBW[THO]⍵QDESCHBW[THO]←REDCHBW×(0⌈NW)[THO←(
QDESCHBW>(REDCHBW×0⌈NW))/⍳⍴L]'
[103]     'INVFIN∆ADJUSTMENTS'MODSUBST'QINVLAG←⍵QINVLAG←0⌈(QCHBW+QCASH)×1⌊Q
DESINVP÷QDESINVP+QDESCHK2'
[104]     NEW∆FIRMS←5 1⍴1 0 (⊂10⍴0) (⊂10⍴0) 0
[105]     NEWFUND←0.5
[106]     ENTRY∆SPECS←0.5 0.05 0.05
[107]     RTD←1.25
[108]     PARAM ←'NEWFUND' 'ENTRY∆SPECS'
[109]    ⍝
[110]    ⍝ ENDOG INTEREST RATE
[111]    ⍝
[112]     RI∆IS∆EXOGENOUS←0
[113]    ⍝
[114]    ⍝ CHANGE GOVERNMENT ACCOUNTING AND CREDIT MARKET FUNCTIONS
[115]    ⍝
[116]     'MONETARY∆SECTOR'MODSUBST'CREDIT⍵CREDIT∆MARKET∆PC'
```

```
[117]     'QUARTER'MODSUBST'GOVERNMENT⍵GOVERNMENT∆ACCOUNTING∆PC'
[118]     'RIDEPG'MODADDLAST'R←IDIFF×RI'
[119]     'RIH'MODADDLAST'R←IDIFF×RI'
[120]     'RIK2'MODADDLAST'R←IDIFF×RI'
[121]     'INVFIN'MODSUBST'QDIV←⍵QDIV←QTAX×((1-TXC)÷TXC)⌊0⌈RTD×(QRR+4×QDPK)
÷RIF'
[122]     'INVFIN'MODADDLAST'AGGPROF←10⌊.1⌈INTDIFF×RIF÷(K1+K2+K3) AVG2 QRR+
4×QDPK'
[123]    ⍝
[124]     QPURCHG←10⍴0.1×6.83E10
[125]     QSURPLUSG←¯0.3×QTTAX
[126]     QCHPOSG←0
[127]     BETABW←0.8
[128]     ALFABW×←1.5
[129]     RTD←1.5
[130]     EXO∆QDINVBLD←120⍴0.0075
[131]     EXO∆QDINVIN←120⍴0.0089
[132]     QGNPCUR←10⍴0.1×1.45E11
[133]     INTDIFF←1.25
[134]     KAPPA1←0
[135]     GDEPRAT←0.2
[136]     GBRWRAT←0.75
[137]     MAXQCHRI←0.005
[138]     IDIFF←0.92
[139]     MAXIDIFF←0.99
[140]     MINIDIFF←0.8
[141]     MAXCHIDIFF←0.005
[142]     MINRFUND1←0.15
[143]     MAXRFUND1←0.5
[144]     MINRFUND2←0.15
[145]     MAXRFUND2←0.5
[146]     CHRFUND1←CHRFUND2←0.015
[147]     RFUND1←0.18 ◊ RFUND2←0.18
```

```
[148]      LIMITED∆BY∆C1←LIMITED∆BY∆C2←0

[149]      RESERVEMONEY←(47÷385)×MONEY

[150]      PARAM ←'INTDIFF' 'GDEPRAT' 'GBRWRAT' 'MAXIDIFF' 'MINIDIFF' 'MAXCH
IDIFF' 'MINRFUND1' 'MAXRFUND1' 'MINRFUND2' 'MAXRFUND2' 'CHRFUND1' 'CHRFUND2
'

[151]     ⍝

[152]     ⍝ ENDOG EXCHANGE RATE

[153]     ⍝

[154]      'EXPORT∆MARKETS'MODSUBST'QPFOR←⍵QPFOR←(1+QCXRATE×XRPEFF)×QPFOR×1+
QDPFOR'

[155]      'BANK∆TRANSACTIONS'MODSUBST'QCHLIQB←⍵QCHLIQB←QINTF+QSAVH+QCHPOSG+
QIMPORT-QINTK2+QINTH+QINTG+QEXPORT'

[156]     ⍝'BANK∆TRANSACTIONS'MODADDLAST'XRATE←XRATE+QCXRATE←(-XRATE)+(XRATE
COEFF×XRATE+(-MAXQCXRATE)⌈MAXQCXRATE⌊(QIMPORT-QEXPORT)÷QIMPORT+QEXPORT)+XRA
TE×1-XRATECOEFF '

[157]      'BANK∆UPDATE'MODADDLAST'XRATE←XRATE×1+QCXRATE←(-MAXQCXRATE)⌈MAXQC
XRATE⌊XRATECOEFF×(QIMPORT-QEXPORT)÷QIMPORT+QEXPORT'

[158]      'BANK∆UPDATE'MODADDLAST'FASS×←1+QCXRATE ⋄ FD×←1+QCXRATE ⋄ LIQBFOR
×←1+QCXRATE'

[159]     ⍝

[160]      XRATE←1 ⋄ QCXRATE←0

[161]     ⍝ DXRATE←0

[162]      XRPEFF←0.1

[163]      XRATECOEFF←0.1 ⋄ MAXQCXRATE←0.01

[164]      Y∆R∆RATES

[165]      PARAM ←'XRPEFF' 'XRATECOEFF' 'MAXQCXRATE'

[166]     ⍝

[167]     ⍝ PUT A LIMIT FOR MAX WAGE CHANGE

[168]     ⍝

[169]      'TARG∆SEARCH∆PC'MODADDLAST'QMAXW←(QEXPW×1-MAXWCOEFF)+MAXWCOEFF×QE
XPW+0⌈4×((QQ÷L)⌈(QPLANQ÷QPLANL))×(QPLANL-L)×QEXPPNET÷QPLANL'

[170]      'TARG∆SEARCH∆SHOW'MODADDLAST'QMAXW←(QEXPW×1-MAXWCOEFF)+MAXWCOEFF×
QEXPW+0⌈4×((QQ÷L)⌈(QPLANQ÷QPLANL))×(QPLANL-L)×QEXPPNET÷QPLANL'
```

```
[171]     'CONFRONT'MODSUBST'→ATK WWω→ATK WW[I]←WW[I]+(QMAXW[I]-WW[I])⌊KSI×
(WW[II]×1+GAMMA)-WW[I]'
[172]     'LABOUR∆SEARCH∆INPUT'MODSUBST'WW←ωWW←QW+(IOTA-IOTALOW×CHL>0)×QEXP
W-QW'
[173]     MAXWCOEFF←0.25
[174]     IOTA←0.9 ◊ IOTALOW←0.15
[175]     PARAM ←'IOTALOW' 'MAXWCOEFF' 'RWEXP'
[176]    ⍝
[177]    ⍝ MODIFY EXPECTATIONS
[178]    ⍝
[179]     'YEARLY∆EXP'MODSUBST'EXPDW←ωEXPDW←(EXPIDW×1-RWEXP)+RWEXP×S AVG2 D
P'
[180]     'YEARLY∆EXP'MODADDLAST'MDIFY∆EXP'
[181]     MAXEXPDP←MAXEXPDS←MAXEXPDW←1
[182]     MINEXPDP←MINEXPDS←MINEXPDW←0.5
[183]     PARAM ←'MAXEXPDP' 'MAXEXPDS' 'MAXEXPDW' 'MINEXPDP' 'MINEXPDS' 'MI
NEXPDW'
[184]     RWEXP←0.25
[185]     EXPXDS←0.09
[186]     EXPXDP←0.04
[187]    ⍝
[188]    ⍝ CHANGE PROFIT MARGIN TARGETING AND INITIAL PRICES
[189]    ⍝
[190]     LOW∆TARGET←0
[191]     'YEARLY∆TARG'MODADD'MHIST←ωMHIST←MHIST⌈0'
[192]     'TARGM'MODADDLAST'Z←0⌈Z'
[193]    ⍝
[194]    ⍝ CHANGE PRODFRONT SPECS
[195]    ⍝
[196]     'INVFIN'MODSUBST'INVEFF←ωINVEFF←INVEFF'
[197]     'PRODFRONT'MODADD'MTEC←ωINVEFF×←1+QDINVEFF[MARKET]'
[198]     'PRODFRONT'MODSUBST'QCHQTOP1←ω⍝ THIS LINE IS REPLACED.'
[199]     'PRODFRONT'MODSUBST'QCHQTOP2←ω⍝ THIS LINE IS REPLACED.'
```

```
[200]      'PRODFRONT'MODSUBST'QCHQTOP←ωQCHQTOP←QINV×INVEFF÷OMEGA+.×PT[MKT I
N]×(1-TXVA2)÷1-TXVA1 ◊ QCHQTOP2←(.5×RES+RESMAX)×QCHQTOP'
[201]      'PRODFRONT'MODSUBST'RES←ωRES←(QCHQTOP2+RES×QTOP)÷QTOP+QCHQTOP'
[202]     ⍝
[203]      PT←(QPDOM[MKT]×1-IMP[MKT])+IMP[MKT]×QPFOR÷1-TXVA2
[204]      PT ←QPDOM[IN] 1
[205]      INVEFF←QTOP×(OMEGA+.×PT[MKT IN]×(1-TXVA2)÷1-TXVA1)÷K1
[206]      QDINVEFF←4⍴0.0005
[207]      RESMAX←0.11
[208]      PARAM ←⊂'QDINVEFF'
[209]     ⍝
[210]     ⍝ ADD QUITS
[211]     ⍝
[212]      'LUUPDATE'MODADDLINE'QUITS' 1
[213]      'LABOUR∆UPDATE'MODSUBST'RU←ωQCHRU+←(LU÷LU+LG+SUM2 L)-RU ◊ RU←LU÷L
U+LG+SUM2 L'
[214]      QUITCOEFF←1
[215]      NORMRU←0.02
[216]     ⍝
[217]     ⍝ ADD OBSOLENCE
[218]     ⍝
[219]      'TARG∆SEARCH∆PC'MODADDLAST'OBSOLETE∆K'
[220]      'TARG∆SEARCH∆SHOW'MODADDLAST'OBSOLETE∆K'
[221]      QTOPMAX←QTOP
[222]      'NULLIFY'MODADDLAST'SHRINK ''QTOPMAX'' '
[223]      'TARG∆SEARCH∆PC'MODADDLAST'QPLANQ←QPLANQ⌊QTOPMAX ◊ QPLANL←QPLANL⌊
RFQ1 QTOPMAX'
[224]      'TARG∆SEARCH∆SHOW'MODADDLAST'QPLANQ←QPLANQ⌊QTOPMAX ◊ QPLANL←QPLAN
L⌊RFQ1 QTOPMAX'
[225]      'PLANQREVISE'MODSUBST'QPLANQ←QPLANQ⌊ωQPLANQ←QTOPMAX⌊QPLANQ⌊QFR1 L
'
[226]      'ENTER∆NEWFIRMS'MODADD'FIRMENTRYωQTOPMAX←QTOPMAX (⍴RELSIZE)⍴(¯1↑Q
TOP)⌊QTOP AVG∆TOP QTOPMAX'
```

```
[227]     'PRODFRONT'MODADDLAST'QTOPMAX←QCHQTOP+QTOPMAX×1-RHO'
[228]     OBSRATE←3
[229]     PARAM ←'QUITCOEFF' 'NORMRU' 'OBSRATE'
[230]    ⍝
[231]    ⍝ INTEREST SENSITIVE HOUSEHOLD SPENDING
[232]    ⍝
[233]     'COMPUTE∆SPENDING'MODSUBST'SWAP←⍵SWAP←.05⌊¯.05⌈ALFA3×(RI÷4)-QDCPI
'
[234]     ALFA3←0.5
[235]    ⍝
[236]    ⍝ CHANGE PRICE ADJUSTMENTS SPECS
[237]    ⍝
[238]     'QUARTERLY∆RESULT'MODSUBST'FINAL⍵FINALQPQSQM∆PC'
[239]     'QUARTER'MODSUBST'QUARTERLY∆EXP⍵QUARTERLY∆EXP∆PC'
[240]     'QUARTER'MODSUBST'DOMESTIC∆M⍵DOMESTIC∆MARKET∆PC'
[241]     'QUARTER'MODSUBST'NATIONAL∆A⍵NATIONAL∆ACCOUNTING∆PC'
[242]    ⍝ 'INVFIN'MODSUBST'AGGPROF←⍵AGGPROF←10⌊.1⌈INTDIFF×RIF÷(K1+K2+K3) A
VG2 QRR+QDPK'
[243]     'INVFIN'MODSUBST'AGGPROF←⍵AGGPROF←10⌊.1⌈INTDIFF×RIF÷(K1+K2+K3) AV
G2 QRR'
[244]     'K3∆IMED'MODSUBST'R←I⍵R←IMSTO+.×PT[MKT IN]×1-TXVA2'
[245]     'INDIRECT∆TAXES'MODSUBST'QVATAX∆IMP⍵QVATAX∆IMP←0'
[246]     'EXTERNAL∆SECTORS'MODSUBST'QIMPURCH∆IN←⍵QIMPURCH∆IN←(PT[MKT IN]×1
-TXVA2)+.×QBUY[MKT IN;IN]'
[247]     'INVFIN'MODSUBST'QDPK←⍵QDPK←(OMEGA×PT[MKT IN]÷+/OMEGA×PT[MKT IN])
+.×((QDPDOM×1-IMP)+IMP×(QDPFOR+QCXRATE×XRPEFF) QDPDOM[IN])-QCHTXVA2-QCHTXVA
1'
[248]     OLDIMP←IMP[MKT]
[249]     DPURCHG←0.02
[250]     QPURCHG←10⍴1700000000
[251]     MAXDP←0.01
[252]     MARKET∆ITER←5
[253]    ⍝
```

```
[254]    ⍝
[255]    ⍝ OTHER CHANGES
[256]    ⍝
[257]     QTDIV←|QTDIV
[258]    ⍝
[259]     BETA2[MKT IN]←(+/BETA2[MKT IN])×((⍴MKT IN)⍴CVA)÷+/CVA[MKT IN;1]
[260]    ⍝
[261]    ⍝
[262]     'EXPORT∆MARKETS'MODADD'QSUFOR←⍵QSUFOR×←(((QSUFOROLD×1+MAXQDSUFOR)
⌊SUM1 QSUFOR)÷SUM1 QSUFOR)[MARKET] ⋄ QSUFOROLD←SUM1 QSUFOR'
[263]     QSUFOROLD←SUM1 X×QS÷QP
[264]     MAXQDSUFOR←0.025
[265]     PARAM ←⊂'MAXQDSUFOR'
[266]     VARIABLE∆NAMES ←⊂'MAXQDSUFOR'
[267]    ⍝
[268]    ⍝
[269]    ⍝ Growth rate of public employment
[270]     ∆←1+0.01×1.3 1.3 0.9 0.9 0.1 0.9 0.6
[271]     ∆←(∆*0.25)-1 ⋄ ∆∆←(+/∆)÷⍴∆
[272]     ∆←(30⍴4)/30⍴∆ (0.5×∆∆+¯1↑∆) 30⍴∆∆
[273]     EXO∆REALCHLG←∆
[274]     'Q∆EXO'MODSUBST'QREALCHLG←⍵QREALCHLG←LG×EXO∆REALCHLG[1]'
[275]    ⍝
[276]    ⍝ OMEGA CHANGES
[277]    ⍝
[278]     QINVINREAL←OMEGAIN×QINVIN×P∆REF÷PT[MKT IN]×(1-TXVA2)÷1-TXVA1
[279]     QINVGREAL←OMEGA×QINVG×P∆REF÷PT[MKT IN]×(1-TXVA2)÷1-TXVA1
[280]     QINVBLDREAL←OMEGABLD×QINVBLD×P∆REF÷PT[MKT IN]×(1-TXVA2)÷1-TXVA1
[281]     'Q∆EXO'MODSUBST'QINVIN⍵QINVINREAL×←1+EXO∆QDINVIN[1]'
[282]     'Q∆EXO'MODSUBST'QINVG⍵QINVGREAL×←1+EXO∆QDINVG[1]'
[283]     'Q∆EXO'MODSUBST'QINVBLD⍵QINVBLDREAL×←1+EXO∆QDINVBLD[1]'
[284]     'COMPUTE∆BUYING'MODSUBST'QINVTOT←⍵QINVTOT←((QINVGREAL +QINVINREAL
+ QINVBLDREAL)×PT[MKT IN]×(1-TXVA2)÷P∆REF)+((OMEGA×PT[MKT IN])÷+/OMEGA×PT[M
KT IN])×SUM2 QINVLAG'

[285]     'COMPUTE∆BUYING'MODADD'QINVTOT←⍵QINVG QINVBLD QINVIN←+/¨QINVGREAL
 QINVBLDREAL QINVINREAL×⊂PT[MKT IN]×(1-TXVA2)÷(1-TXVA1)×P∆REF'

[286]     ⍝

[287]     ⍝

[288]     ⍝ ADD NEW PARAMETERS

[289]     ⍝

[290]     CON:

[291]      PARAM[3;1]←⊂(⊃PARAM[3;1])ABOVE↑PARAM

[292]      PARAMETER∆NAMES←↓⊃PARAM[3;1]

[293]      VARIABLE∆NAMES ←'QINVINREAL' 'QINVBLDREAL' 'QINVGREAL' 'QSUFOROLD
' 'OLDIMP' 'PARAMETER∆NAMES'

[294]     ⍝


*******************************************************************************

MSTART900

[0]      MSTART900

[1]     ⍝

[2]     ⍝ Calibrated parameters.

[3]     ⍝ Model vesrion 2.0 is used with R1982.91 and MSTART999

[4]     ⍝ for calibration. Control variables and parameter space are

[5]     ⍝ defined by DATL and PARL variables  respectively  in DATA

[6]     ⍝ workspace. AMAXENT variable should be equal to 2.

[7]     ⍝ Prepared by E.Taymaz  April 1991.

[8]     ⍝

[9]      MARKET∆ITER←3

[10]      NITER←4

[11]      ALFABW←0.0261490475

[12]      BETABW←0.9029394488

[13]      CHRFUND1←0.0193471812

[14]      CHRFUND2←0.0141927157

[15]      E1←0.1089252605

[16]      FIP←0.0842019195
```

```
[17]     FIS←0.072185089
[18]     FIW←0.0879668459
[19]     GAMMA←0.3006141805
[20]     GBRWRAT←0.7255351373
[21]     GDEPRAT←0.184362314
[22]     INTDIFF←1.213976264
[23]     IOTA←0.8917630875
[24]     IOTALOW←0.1399296093
[25]     KSI←0.1425520581
[26]     LAMDA1←0.5646107603
[27]     LAMDA2←0.7566414578
[28]     LOWERΔMHIST←0.760064294
[29]     MAXCHIDIFF←0.0053066724
[30]     MAXDP←0.0076168657
[31]     MAXIDIFF←0.9885746857
[32]     MAXQCHRI←0.0044792517
[33]     MAXQCXRATE←0.0104058657
[34]     MAXRIDIFF←0.0491137389
[35]     MAXWCOEFF←0.215619196
[36]     MINIDIFF←0.8115695092
[37]     MINRFUND1←0.1506980568
[38]     MINRFUND2←0.1504663515
[39]     NEWFUND←0.5174949139
[40]     NORMRU←0.0182401716
[41]     OBSRATE←3.070525545
[42]     QUITCOEFF←1.036312969
[43]     R←0.5171516728
[44]     RFUND1←0.1506980568
[45]     RFUND2←0.1823497932
[46]     RHO←0.0073760815
[47]     RHOBOOK←0.0308450053
[48]     RLU←0.576037509
[49]     RTRANS←0.5201723303
```

```
[50]      SKREPA←79.16787031
[51]      SMP←0.5104608729
[52]      SMS←0.510506079
[53]      SMT←0.4708944958
[54]      SMW←0.4594676663
[55]      THETA←0.0112773529
[56]      TMFASS←0.2410732285
[57]      TMFD←0.1810916951
[58]      UTREF←0.8379947406
[59]      XRATECOEFF←0.1132212279
[60]      XRPEFF←0.1131868332
[61]      TMSTO←0.9131268107
[62]      TMIMSTO←0.9195760704
[63]      MAXQDSUFOR←0.0144811424
[64]      B11←1.00702343
[65]      B12←1.018319013
[66]      B13←0.6882757419
[67]      B14←0.7995462478
[68]      B15←0.8799771684
[69]      B16←1.012275678
[70]      B17←0.9839670222
[71]      B18←0.9063180729
[72]      B19←1.011074046
[73]      B110←0.7413158738
[74]      B111←0.996439764
[75]      B112←0.4714071077
[76]      BETA1←B11 B12 B13 B14 B15 B16 B17 B18 B19 B110 B111 B112
```

```
*******************************************************************************

ADJUSTΔPRICESΔPC

[0]       ADJUSTΔPRICESΔPC;Δ;MAXDP2
[1]       MAXDP2←MAXDP×1-IMP[MKT]
```

```
[2]      Δ←((QTBUY[MKT]×1-IMP[MKT])÷NULLΔGOODS+SUM1 QOPTSUDOM)-1

[3]      PT[MKT]×←1+(MAXDP2×Δ)÷MAXDP2+|Δ


********************************************************************************

COMPUTEΔIMPORTSΔPC

[0]      COMPUTEΔIMPORTSΔPC;IMPZ;QMAXTSUDOM

[1]      QTBUYFOR←QTBUY×IMP IMPZ←0

[2]      QTBUYDOM←QTBUY-QTBUYFOR

[3]      QTBUYDOM[MKT]←QTBUYDOM[MKT]⌊QMAXTSUDOM←NULLΔGOODS+SUM1 0⌈QQ+(STO-MI

NSTO)-QSUFOR

[4]      QTBUYFOR←⍉(2 ρNDURΔDUR)ρQTBUYFOR QTBUY-QTBUYDOM+QTBUYFOR

[5]      ALWAYS'QTBUYFOR≥0 '

[6]      ALWAYS' 0=QTBUYFOR[Z;]'

[7]     ⍝ QIMPORT←(QPFOR[MKT] (1-TXVA2)×(QPDOM[IN]×1+QDPIN))+.×+/QTBUYFOR[MK

T IN;]

[8]      QIMPORT←(QPFOR[MKT] (1-TXVA2)×(1+QCXRATE×XRPEFF)×(QPDOM[IN]×1+QDPIN

))+.×+/QTBUYFOR[MKT IN;]

[9]      IMP[MKT]←(+/QTBUYFOR[MKT;])÷QTBUY[MKT]


********************************************************************************

DOMESTICΔMARKETΔPC

[0]      DOMESTICΔMARKETΔPC;QOPTSUDOM;QPRELPDOM;INMONEYHH;QSPSAVREQ

[1]      MARKETΔENTRANCEΔPC

[2]      HOUSEHOLDΔINIT

[3]     ⍝

[4]      MARKETΔCONFRONTΔPC

[5]     ⍝

[6]      COMPUTEΔIMPORTSΔPC

[7]      NOTIFY'QTBUY[MKT IN]=0 '

[8]      DOMESTICΔRESULTΔPC

[9]      EXTERNALΔSECTORS

[10]      HOUSEHOLDΔUPDATE
```

```
[11]    INDIRECTΔTAXES
[12]    ⍝
[13]    ALWAYS'INMONEYHH=QWTAX+QITAX+QSAVH++/QSP[NDURΔDUR;]+.×NH'
[14]    ALWAYS'((+/QPURCHG+QSP[MKT IN;]+.×NH)+(QINVG+QINVIN+QINVBLD+SUM2 Q
INVLAG)+(+/QIMPURCHΔIN)+((SUM2 QIMQ)+.×QPDOM×1-TXVA2))=(+/QVATAX[MKT IN])+(
+/QSDOMΔIN)+QIMPORT+QTSUDOM+.×(QPDOM×1-TXVA2)[MKT]'

*****************************************************************************
DOMESTICΔRESULTΔPC

[0]     DOMESTICΔRESULTΔPC
[1]    ⍝ QDPDOM[MKT]←((PT[MKT]-IMP[MKT]×QPFOR[MKT]÷1-TXVA2)÷((1-IMP[MKT])×Q
PDOM[MKT]))-1
[2]     QDPDOM[MKT]←((PT[MKT]-OLDIMP×QPFOR[MKT]÷1-TXVA2)÷((1-OLDIMP)×QPDOM[
MKT]))-1
[3]     QDPDOM[IN]←(PT[IN]÷QPDOM[IN])-1
[4]     QPDOM←(1+QDPDOM)×QPDOM
[5]     PT[MKT]←(QPDOM[MKT]×1-IMP[MKT])+(QPFOR[MKT]×IMP[MKT])÷1-TXVA2
[6]    ⍝ QPZ←PT[Z]×1-TXVAZ
[7]     QTSUDOM←0⌈QTBUYDOM[MKT]-NULLΔGOODS
[8]    ⍝ QSZ←QTBUYDOM[Z]×QPZ
[9]     ALWAYS'QSZ=(1-TXVAZ)×NH+.×QSP[Z;]'

*****************************************************************************
ENTERΔNEWFIRMS

[0]     ENTERΔNEWFIRMS;MM;NL;NF;N;KKI;ΔTEC;ΔRES;RELSIZE
[1]     N←¯1↑⍴NEWΔFIRMS
[2]     →(N=0)/0
[3]     →(0=NF←+/NEWΔFIRMS[2;])/0
[4]     KKI←0
[5]    BEG:KKI←KKI+1
[6]     MM←NEWΔFIRMS[1;KKI]
[7]     RELSIZE←NEWΔFIRMS[2;KKI]⍴ENTRYΔSPECS[1]
```

```
[8]      ∆TEC←NEW∆FIRMS[2;KKI]⍴ENTRY∆SPECS[2]

[9]      ∆RES←NEW∆FIRMS[2;KKI]⍴ENTRY∆SPECS[3]

[10]      FIRMENTRY∆PC NEW∆FIRMS[1 2;KKI]

[11]      →(KKI<N)/BEG

[12]      NEW∆FIRMS←5 1⍴1 0 (⊂10⍴0) (⊂10⍴0) 0

[13]      NL←L[(⍴L)-NF-⍳NF]

[14]      →DOIF LU≥+/NL ◊ LU-←+/NL ◊ →0

[15]      NL←NL×LU÷+/NL

[16]      LU-←+/NL

[17]      L[(⍴L)-NF-⍳NF]←NL
```

****

FINALQPQSQM∆PC

```
[0]      FINALQPQSQM∆PC

[1]     ⍝

[2]      QSU←QSUFOR+QSUDOM

[3]      QDS←(QSFOR+QSDOM-QS)÷QS

[4]      QS←QSFOR+QSDOM

[5]      ALWAYS'QS>0 '

[6]      QDP←((QS÷QSU)-QP)÷QP

[7]      QP←QS÷QSU

[8]      QVA←QVA×1+QDVA←¯1+(QQ×QP-SHARE×((PT[⍳10]×1-TXVA2)+.×IO)[MARKET])÷QV

A

[9]      QM←1-(L×QW)÷4×QSNET←QS-QIMQ+.×PT[⍳10]×1-TXVA2
```

****

FIRMENTRY∆PC

```
[0]      FIRMENTRY∆PC MMM;K2;BW;I;MM;NUM;SCALE;A22;A21;S;DP;DVA;L;M;P;Q;QP;Q

Q;QTOP;QVA;RES;SHARE;VA;W;K1;TEC;DS;DQ;INVEFF

[1]      MM←MMM[1]

[2]      NUM←MMM[2]

[3]      RW←RW (⍴RELSIZE)⍴S AVG∆TOP RW
```

```
[4]    A21←(ρRELSIZE)ρ0
[5]    A22←(ρRELSIZE)ρ(VA AVG∆TOP A22)
[6]    K1←K1 K1←(ρRELSIZE)ρ+/(⊃NEW∆FIRMS[3;KKI])×PT[MKT IN]×1-TXVA2
[7]    K1BOOK←K1BOOK K1
[8]    K2←NEW∆FIRMS[5;KKI]-K1++/(⊃NEW∆FIRMS[4;KKI])×PT[MKT IN]×1-TXVA2
[9]    BW←0⌈-K2
[10]    K2←K2 0⌈K2
[11]    BW←BW BW
[12]    QINV←QINV RELSIZE×0
[13]    QINVLAG←QINVLAG RELSIZE×0
[14]    DELAY∆INV←DELAY∆INV [1](NUM 3)ρ0
[15]    X←X (ρRELSIZE)ρS AVG∆TOP X
[16]    P←P P←(ρRELSIZE)ρS AVG∆TOP P
[17]    QP←QP QP←(ρRELSIZE)ρQS AVG∆TOP QP
[18]    DP←DP DP←(ρRELSIZE)ρS AVG∆TOP DP
[19]    W←W W←(ρRELSIZE)ρL AVG∆TOP W
[20]    DW←DW (ρRELSIZE)ρVA AVG∆TOP DW
[21]    QDW←QDW (ρRELSIZE)ρ(L×QW)AVG∆TOP QDW
[22]    QW←QW (ρRELSIZE)ρL AVG∆TOP QW
[23]    ⍎(0≠⎕NC'CHM')/'CHM←CHM (ρRELSIZE)ρS AVG∆TOP CHM'
[24]   ⍝
[25]    DVA←DVA DVA←(ρRELSIZE)ρVA AVG∆TOP DVA
[26]    VA←VA VA←RELSIZE×(S AVG∆TOP VA)
[27]    QVA←QVA QVA←RELSIZE×(S AVG∆TOP QVA)
[28]    SHARE←SHARE SHARE←(ρRELSIZE)ρS AVG∆TOP SHARE
[29]    Q←Q Q←VA÷P-SHARE×((PT[MKT IN]×1-TXVA2)+.×IO)[MM]
[30]    QQ←QQ QQ←QVA÷QP-SHARE×((PT[MKT IN]×1-TXVA2)+.×IO)[MM]
[31]    DQ←DQ DQ←DVA-DP
[32]   ⍝
[33]    S←Q×P
[34]    DS←DS DS←(Q×DP+DQ×P)÷S
[35]    QS←QS QQ×QP
[36]    AMAN←((ρAMAN)+(ρRELSIZE) 0)↑AMAN
```

```
[37]   ⍝
[38]    EXPDP←EXPDP (ρRELSIZE)ρS AVGΔTOP EXPDP
[39]    EXPDS←EXPDS (ρRELSIZE)ρS AVGΔTOP EXPDS
[40]    EXPDW←EXPDW (ρRELSIZE)ρS AVGΔTOP EXPDW
[41]    HISTDP←HISTDP (ρRELSIZE)ρS AVGΔTOP HISTDP
[42]    HISTDPDEV←HISTDPDEV (ρRELSIZE)ρS AVGΔTOP HISTDPDEV
[43]    HISTDPDEV2←HISTDPDEV2 (ρRELSIZE)ρS AVGΔTOP HISTDPDEV2
[44]    HISTDS←HISTDS (ρRELSIZE)ρS AVGΔTOP HISTDS
[45]    HISTDSDEV←HISTDSDEV (ρRELSIZE)ρS AVGΔTOP HISTDSDEV
[46]    HISTDSDEV2←HISTDSDEV2 (ρRELSIZE)ρS AVGΔTOP HISTDSDEV2
[47]    HISTDW←HISTDW (ρRELSIZE)ρS AVGΔTOP HISTDW
[48]    HISTDWDEV←HISTDWDEV (ρRELSIZE)ρS AVGΔTOP HISTDWDEV
[49]    HISTDWDEV2←HISTDWDEV2 (ρRELSIZE)ρS AVGΔTOP HISTDWDEV2
[50]   ⍝
[51]    CUMINV←CUMINV (ρRELSIZE)ρ0
[52]    CUML←CUML (ρRELSIZE)ρ0
[53]    LLASTYR←LLASTYR (ρRELSIZE)ρ1
[54]    DL←DL (ρRELSIZE)ρ0
[55]    DNW←DNW (ρRELSIZE)ρ0
[56]    NWLASTYR←NWLASTYR (ρRELSIZE)ρ1
[57]    CUMM←CUMM (ρRELSIZE)ρ0
[58]    CUMQ←CUMQ (ρRELSIZE)ρ0
[59]    CUMS←CUMS (ρRELSIZE)ρ0
[60]    CUMSNET←CUMSNET (ρRELSIZE)ρ0
[61]    CUMSU←CUMSU (ρRELSIZE)ρ0
[62]    CUMVA←CUMVA (ρRELSIZE)ρ0
[63]    CUMWS←CUMWS (ρRELSIZE)ρ0
[64]    CUMINTPAYF←CUMINTPAYF (ρRELSIZE)ρ0
[65]    CUMDEPR←CUMDEPR (ρRELSIZE)ρ0
[66]    CUMTAXF←CUMTAXF (ρRELSIZE)ρ0
[67]    CUMDIV←CUMDIV (ρRELSIZE)ρ0
[68]    CUMSUBSF←CUMSUBSF (ρRELSIZE)ρ0
[69]    CUMCHBWF←CUMCHBWF (ρRELSIZE)ρ0
```

```
[70]      CUMCHK2F←CUMCHK2F (ρRELSIZE)ρ0

[71]      SNET←SNET (ρRELSIZE)ρ0

[72]      REALLY∆BAD←REALLY∆BAD (ρRELSIZE)ρ0

[73]      QDP←QDP (ρRELSIZE)ρ0

[74]      QDQ←QDQ (ρRELSIZE)ρ0

[75]      QDS←QDS (ρRELSIZE)ρ0

[76]      QDVA←QDVA (ρRELSIZE)ρ0

[77]      QM←QM (ρRELSIZE)ρ0

[78]      QOPTSU←QOPTSU (ρRELSIZE)ρ0

[79]      QSDOM←QSDOM (ρRELSIZE)ρ0

[80]      QSFOR←QSFOR (ρRELSIZE)ρ0

[81]      QSNET←QSNET (ρRELSIZE)ρ0

[82]      QSU←QSU (ρRELSIZE)ρ0

[83]      QSUDOM←QSUDOM (ρRELSIZE)ρ0

[84]      QSUFOR←QSUFOR (ρRELSIZE)ρ0

[85]      STO←STO (ρRELSIZE)ρ0

[86]      IMSTO←IMSTO [1](NUM 10)ρ⊃NEW∆FIRMS[4;KKI]

[87]      QIMQ←QIMQ [1](NUM 10)ρ0

[88]     ⍝

[89]      RES←RES RES←(ρRELSIZE)ρ(1-∆RES)×(S AVG∆TOP RES)

[90]      QTOP←QTOP QTOP←QQ÷1-A21+A22+RES

[91]      TEC←TEC MTEC[MM]⌊TEC←(ρRELSIZE)ρ(1+∆TEC)×(TEC AVG∆TOP TEC)

[92]      L←L L←1⌈((QTOP÷TEC)×⍟(1-RES)÷A22)

[93]      INVEFF←INVEFF INVEFF←QTOP×QP÷K1

[94]      M←M M←0.1⌈(1-W×L÷VA)

[95]      MHIST←MHIST M×0.5

[96]     ⍝

[97]      RSUBS∆CASH←RSUBS∆CASH 0×RELSIZE

[98]      RSUBS∆EXTRA←RSUBS∆EXTRA 0×RELSIZE

[99]      BIG←BIG (ρRELSIZE)ρS AVG∆TOP BIG

[100]      SMALL←SMALL (ρRELSIZE)ρS AVG∆TOP SMALL

[101]      IMBIG←IMBIG (ρRELSIZE)ρS AVG∆TOP IMBIG

[102]      IMSMALL←IMSMALL (ρRELSIZE)ρS AVG∆TOP IMSMALL
```

```
[103]     BAD←BAD (⍴RELSIZE)⍴0

[104]     RWC←RWC RWC←(⍴RELSIZE)⍴S AVG∆TOP RWC

[105]     ENTRY∆EPSILON←ENTRY∆EPSILON RWC-(⍴RWC)⍴RIF

[106]     S←S S

[107]    ⍝

[108]     MARKET←MARKET (⍴RELSIZE)⍴MM

[109]     ORIGMARKET←ORIGMARKET (⍴RELSIZE)⍴MM

[110]     LEFT←LEFT RELSIZE=RELSIZE

[111]     →(2≠⎕NC'FIRMCHARC')/OUT

[112]     FIRMCHARC←FIRMCHARC ABOVE THIS∆YEAR MM NUM (12 NUM⍴((100×M) L (0.
0001×VA QQ) QP DS (0.0001×K1 QTOP) TEC (100×INVEFF) (100×RES) (100×A22)))[;
1]

[113]    OUT:


********************************************************************************

GOVERNMENT∆ACCOUNTING∆PC

[0]     GOVERNMENT∆ACCOUNTING∆PC;MAXCHPOSG

[1]    ⍝ QWTAX QITAX QVATAX QCTAX ARE ALREADY AVAILABLE FROM THROUGHOUT THE
 QUARTER

[2]     QINTG←(DEPG×RIDEPG÷4)-(BWG×RIBWG÷4)

[3]     QINTGFOR←(DEPGFOR×RIDEPGFOR÷4)-(BWGFOR×RIBWGFOR÷4)

[4]     QWSG←LG×QWG÷4

[5]     QSUBS←QSUBSFOR+QSUBSDOM+QSUBS∆CASH

[6]     QSPG←QWSG+(+/QPURCHG)+QTRANS+QSUBS

[7]     QSURPLUSG←(QTTAX←QWTAX+QITAX+(+/QVATAX)+QCTAX)+QINTG+QINTGFOR-QSPG+
QINVG

[8]     →DOIF 2=⎕NC'QDGNPCUR' ◊ QMPRINT←0⌈(-QSURPLUSG×QSURPLUSG>0)+RESERVEM
ONEY×QDGNPCUR

[9]     →DOIF 2≠⎕NC'QDGNPCUR' ◊ QMPRINT←0⌈(-QSURPLUSG×QSURPLUSG>0)+RESERVEM
ONEY×S AVG2 DS

[10]     POSGFOR←POSGFOR+QCHPOSGFOR←0

[11]     QCHPOSG←QSURPLUSG+QMPRINT-QCHPOSGFOR

[12]     →(QCHPOSG≤0)/JMP
```

```
 MAXCHPOSG←1⌈(GDEPRAT×(SUM2 K2)+FD+0⌈WH+.×NH)-POSG
 →DEV
JMP:
 MAXCHPOSG←¯1×1⌈POSG+GBRWRAT×(SUM2 BW)+FASS+LIQBFOR+LIQB+BWG-0⌈WH+.
×NH
DEV:
 QCHPOSG←(MAXCHPOSG×QCHPOSG)÷MAXCHPOSG+QCHPOSG
 POSG←POSG+QCHPOSG←QSURPLUSG-QCHPOSGFOR-QMPRINT←(¯0.01×RESERVEMONEY
)⌈QCHPOSG+QCHPOSGFOR-QSURPLUSG
 RESERVEMONEY+←QMPRINT
⍝
 CUMWTAX←CUMWTAX+QWTAX
 CUMITAX←CUMITAX+QITAX
 CUMVATAX←CUMVATAX++/QVATAX
 CUMCTAX←CUMCTAX+QCTAX
 CUMWSG←CUMWSG+QWSG
 CUMLG←(LG+CUMLG×NRS-1)÷NRS
 CUMPURCHG←CUMPURCHG+QPURCHG
 CUMTRANS←CUMTRANS+QTRANS
 CUMSUBS←CUMSUBS+QSUBS
 CUMMPRINT←CUMMPRINT+QMPRINT
 CUMINTG←CUMINTG+QINTGFOR+QINTG
 CUMINVG←CUMINVG+QINVG
 →(NRS<4)/OUT
 WTAX←CUMWTAX
 ITAX←CUMITAX
 VATAX←CUMVATAX
 CTAX←CUMCTAX
 MPRINT←CUMMPRINT
 INTG←CUMINTG
 INVG←CUMINVG
 WG←WG×1+DWG←¯1+CUMWSG÷CUMLG×WG
 WSG←WSG×1+DWSG←¯1+CUMWSG÷WSG
```

```
[44]      PURCHG←CUMPURCHG

[45]      TRANS←CUMTRANS

[46]      SUBS←CUMSUBS

[47]      SPG←WSG+(+/PURCHG)+TRANS+SUBS

[48]      SURPLUSG←WTAX+ITAX+VATAX+CTAX+INTG-SPG+INVG

[49]     OUT:
```

****

MARKET∆CONFRONT∆PC

```
[0]      MARKET∆CONFRONT∆PC;J

[1]     ⍝

[2]      IMP[MKT]←OLDIMP

[3]      IMP[MKT]←1⌊0⌈IMP[MKT]-(IMP[MKT]+(1-2×IMP[MKT])×(QPDOM×1-TXVA2)[MKT]
>QPFOR)×(÷4×TMIMP)×(QPFOR-(QPDOM×1-TXVA2)[MKT])÷(QPFOR⌊(QPDOM×1-TXVA2)[MKT]
)

[4]      OLDIMP←IMP[MKT]

[5]      PT←(QPRELPDOM×1-IMP[MKT])+IMP[MKT]×QPFOR[MKT]÷1-TXVA2

[6]     ⍝ PT←PT (QPDOM[IN]×1+QDPIN) 1

[7]      PT←PT ((1+QCXRATE×XRPEFF)×QPDOM[IN]×1+QDPIN) 1

[8]     ⍝ QPURCHG←0⌈(((1+DPURCHG)×(LG×QWG÷4)++/QPURCHG)-LG×QWG÷4)×GKOFF×PT[⍳
10]÷+/GKOFF×PT[⍳10]

[9]      QPURCHG←0⌈(((1+0⌈DPURCHG×(QTTAX+QSURPLUSG)÷QTTAX)×(LG×QWG÷4)++/QPUR
CHG)-LG×QWG÷4)×GKOFF×PT[⍳10]÷+/GKOFF×PT[⍳10]

[10]      J←1

[11]     L:

[12]      COMPUTE∆SPENDING

[13]      COMPUTE∆BUYING

[14]      →(MARKET∆ITER<J←J+1)/0

[15]      ADJUST∆PRICES∆PC

[16]      →L
```

****

```
MARKET∆ENTRANCE∆PC

[0]     MARKET∆ENTRANCE∆PC;∆∆;MAXDP2
[1]     MAXDP2←(1-IMP[MKT])×MAXDP×MARKET∆ITER
[2]     QOPTSUDOM←(1-X)×QOPTSU
[3]     ∆∆←¯1+QOPTSUDOM AVG1 QEXPP÷QP
[4]     ∆∆←∆∆×MAXDP2÷MAXDP2+|∆∆
[5]     QPRELPDOM←(QPDOM×1+QCHTXVA2)[MKT]×1+∆∆

********************************************************************************

MDIFY∆EXP

[0]     MDIFY∆EXP;W̲;∆
[1]     EXPDS×←1+X×4×QCXRATE×XRPEFF
[2]    ⍝
[3]     W̲←(L AVG1 W)[MARKET]
[4]     ∆←((((UTREF×QFR1 L)-QQ)÷UTREF×QFR1 L)<(EXPDS-EXPDP)÷1+EXPDP)×(1÷W)×
((+/L÷W)÷(+/L÷W)+SKREPA×LU÷+/W÷⍴W)×0⌈(W̲×(1+GAMMA))-W
[5]    ⍝
[6]     EXPDW←(2×EXPDP)⌊∆+EXPDW×1-AMAN[;1]÷L
[7]     EXPDP←(MAXEXPDP-(MAXEXPDP+MINEXPDP)×EXPDP<0)×EXPDP÷(MAXEXPDP-(MAXEX
PDP+MINEXPDP)×EXPDP<0)+EXPDP
[8]     EXPDW←(MAXEXPDW-(MAXEXPDW+MINEXPDW)×EXPDW<0)×EXPDW÷(MAXEXPDW-(MAXEX
PDW+MINEXPDW)×EXPDW<0)+EXPDW
[9]     EXPDS←(MAXEXPDS-(MAXEXPDS+MINEXPDS)×EXPDS<0)×EXPDS÷(MAXEXPDS-(MAXEX
PDS+MINEXPDS)×EXPDS<0)+EXPDS

********************************************************************************

NATIONAL∆ACCOUNTING∆PC

[0]     NATIONAL∆ACCOUNTING∆PC;QCHTSTOCURF;QCHTSTOCURM;OLD
[1]     TSTOCURF←TSTOCURF+QCHTSTOCURF←(SUM1 K3∆FINISH)-TSTOCURF
[2]     TSTOCURM←TSTOCURM+QCHTSTOCURM←(PT[MKT]×(1-TXVA2)×SUM1 STO)-TSTOCURM
[3]     ⍎(2=⎕NC'QGNPCUR')/'OLD←Q̲G̲N̲P̲CUR'
[4]     QGNPCUR←((SUM1 QSNET)+QCHTSTOCURF) QVA∆IN ((+/QVATAX)+(+/QCHTSTOCUR
M)-(+/QCHTSTOCURF)+QVATAX∆IMP) (-QSUBS-QSUBS∆CASH) QWSG
[5]     QGNPCUR←QGNPCUR (+/QPURCHG) (+/QSP[MKT IN;]+.×NH) ((SUM2 QINVLAG)-Q
TCHINV) QINVIN QINVBLD QINVG (+/QCHTSTOCURM)
[6]     QGNPCUR←QGNPCUR QEXPORT (-QIMPORT+QVATAX∆IMP)
[7]     ALWAYS'(+/(3+ρMKT IN)↑QGNPCUR)=(+/(2+ρMKT IN)↓QGNPCUR)'
[8]     ⍎(2=⎕NC'OLD')/'QDGNPCUR←¯1+QGNPCUR÷OLD'
[9]     CUMGNPCUR←CUMGNPCUR+QGNPCUR
[10]    ⍝
[11]     QGNPFIX←(P∆REF×((SUM1 QQ)-QWASTE) QQIN)-(P∆REF+.×QBUY[MKT IN;MKT I
N])
[12]     QGNPFIX←QGNPFIX LG×WG∆REF÷4
[13]     QGNPFIX←QGNPFIX ((P∆REF÷1-TXVA2)+.×QPURCHG÷PT[MKT IN]) ((P∆REF÷1-T
XVA2)+.×(QSP[MKT IN;]+.×NH)÷PT[MKT IN])
[14]     QGNPFIX←QGNPFIX P∆REF+.×OMEGA×((SUM2 QINVLAG)-QTCHINV)÷PT[MKT IN]×
(1-TXVA2)÷(1-TXVA1)
[15]     QGNPFIX←QGNPFIX P∆REF+.×OMEGAIN×QINVIN÷PT[MKT IN]×(1-TXVA2)÷(1-TXV
A1)
[16]     QGNPFIX←QGNPFIX P∆REF+.×OMEGABLD×QINVBLD÷PT[MKT IN]×(1-TXVA2)÷(1-T
XVA1)
[17]     QGNPFIX←QGNPFIX P∆REF+.×OMEGAG×QINVG÷PT[MKT IN]×(1-TXVA2)÷(1-TXVA1
)
[18]    ⍝ QGNPFIX←QGNPFIX P∆REF+.×OMEGA×((SUM2 QINVLAG)-QTCHINV)÷+/OMEGA×PT
[MKT IN]×1-TXVA2
[19]    ⍝ QGNPFIX←QGNPFIX +/QINVINREAL
[20]    ⍝ QGNPFIX←QGNPFIX +/QINVBLDREAL
[21]    ⍝ QGNPFIX←QGNPFIX +/QINVGREAL
[22]    ⍝ QGNPFIX←QGNPFIX (P∆REF[MKT]+.×QCHTSTO) (P∆REF∆FOR+.×(SUM1 QSUFOR)
 (XIN×QQIN)) (-P∆REF∆FOR+.×+/QTBUYFOR[MKT IN;])
[23]     QGNPFIX←QGNPFIX (P∆REF[MKT]+.×QCHTSTO) (P∆REF+.×(SUM1 QSUFOR) (XIN
×QQIN)) (-P∆REF+.×+/QTBUYFOR[MKT IN;])
[24]     ALWAYS'(+/(1+ρMKT IN)↑QGNPFIX)=(+/(ρMKT IN)↓QGNPFIX)'
[25]     CUMGNPFIX←CUMGNPFIX+QGNPFIX
[26]    ⍝
```

```
[27]     CUMEXPORT←CUMEXPORT+QEXPORT

[28]     CUMIMPORT←CUMIMPORT+QIMPORT

[29]    ⍝

[30]    ⍝FOR THE LAST QUARTER

[31]    ⍝

[32]     →(NRS<4)/OUT

[33]     GNPFIX←CUMGNPFIX

[34]     GNPCUR←CUMGNPCUR

[35]     EXPORT←CUMEXPORT

[36]     IMPORT←CUMIMPORT

[37]    OUT:
```

****

OBSOLETEΔK

```
[0]     OBSOLETEΔK;B;Y;J;D;Δ

[1]     J←⍳⍴Y←÷B←QEXPW÷4×WTIX×(1-RES)×TEC×QEXPPNET

[2]     ALWAYS'B>0 '

[3]    NR:→NR+0=⍴J←(D>0.001×Y[J]←Y[J]-D←((B[J]×Y[J])+(*-Y[J])-1)÷B[J]-*-Y[J
])/J

[4]    ⍝ Δ←(1+OBSRATE)×QFR1 Y×QTOP÷TEC

[5]     Δ←QFR1 OBSRATE×Y×QTOP÷TEC

[6]     Δ←(QTOPMAX⌊Δ)÷QTOPMAX

[7]     QTOPMAX←QTOPMAX×Δ

[8]     K1×←Δ
```

****

QUARTERLYΔEXPΔPC

```
[0]     QUARTERLYΔEXPΔPC;QEXPDW;QCHTXVA2IN;ΔΔ

[1]    ⍝

[2]     QEXPDP←EXPDP÷4

[3]     QEXPDW←EXPDW÷4

[4]     QEXPDS←EXPDS÷4
```

```
[5]      →(NRS=1)/L
[6]      QEXPDP←QEXPDP+FIP×QDP-QEXPDP
[7]      QEXPDW←QEXPDW+FIW×QDW-QEXPDW
[8]      QEXPDS←QEXPDS+FIS×QDS-QEXPDS
[9]     L:
[10]      QEXPP←QP×1+QEXPDP
[11]      QEXPW←QW×1+QEXPDW
[12]      QEXPS←QS×1+QEXPDS
[13]     ⍝
[14]      ΔΔ←(IMP[MKT]×QPFOR[MKT])+(1-IMP[MKT])×QPDOM[MKT]×1-TXVA2
[15]      ΔΔ←ΔΔ QPDOM[IN]×1-TXVA2
[16]      QEXPPIM←ΔΔ×1+(QQ AVG1 QEXPDP) QDPIN-QCHTXVA2IN←(ρMKT)↓(ρMKT IN)ρQC
HTXVA2

********************************************************************************

QUITS

[0]      QUITS;Δ
[1]      Δ←0.01×QUITCOEFF×0⌈¯1+(+/L×W)÷(+/L)×W×1+GAMMA×RU÷NORMRU
[2]      LU←LU++/Δ×L++/AMAN
[3]      L×←1-Δ
[4]      AMAN←AMAN×⌽(3 ρΔ)ρ1-Δ
[5]      RU←RU+QCHRU←(LU÷LU+LG+SUM2 L)-RU

********************************************************************************

STARTΔENTRYΔPC

[0]      STARTΔENTRYΔPC MMΔENTRY;ENTRYΔMKTNR;RELSIZE;I̲;MM;NUM;LIST;NEWΔCAP;K̲
1;K̲2;E̲XPDP;M̲STO;S̲TO;DIZI
[1]      NEWΔCAP←LIST←0ρ0
[2]      NEWΔFIRMS←5 4ρ1 2 3 4 16ρ0
[3]      RELSIZE←ENTRYΔSPECS[1]
[4]      I̲←1
[5]     ST:→END×ι(ρMMΔENTRY)<I̲←I̲+1
```

```
[6]      ENTRY∆MKTNR←MM←1ρMM∆ENTRY[I]
[7]      →(4>+/MARKET=MM)/ST
[8]      NUM←START∆ENT2 ENTRY∆EPSILON
[9]      →(NUM=0)/ST
[10]      LIST ←NUMρMM
[11]      EXPDP←1+(S AVG∆TOP EXPDP)÷4
[12]      K1←RELSIZE×EXPDP×K1 AVG∆TOP K1
[13]      K2←RELSIZE×EXPDP×K2 AVG∆TOP K2
[14]      MSTO←RELSIZE×I0+.×(+/MINIMSTO)AVG∆TOP+/MINIMSTO
[15]      STO←EXPDP×MSTO+.×10 1ρPT[MKT IN]×1-TXVA2
[16]      NEW∆CAP ←NUMρK1+K2+STO
[17]      NEW∆FIRMS[3 4 5;I-1]←(⊂OMEGA×K1÷+/OMEGA×PT[MKT IN]×(1-TXVA2)×EXPDP
) (⊂MSTO) K1+K2+STO
[18]      →ST
[19]     END:
[20]      →((NEWFUND×WH+.×NH)≥+/NEW∆CAP)/JMP
[21]      →(0=ρLIST)/JMP
[22]      DIZI←(ρLIST)?ρLIST
[23]      LIST←(1≠DIZI)/LIST
[24]      NEW∆CAP←(1≠DIZI)/NEW∆CAP
[25]      →END
[26]     JMP:
[27]      NEW∆FIRMS[2;LIST]+←LIST∊MM∆ENTRY
[28]      NEW∆FIRMS←(0≠NEW∆FIRMS[2;])/NEW∆FIRMS
[29]      ⍎(0=¯1↑ρNEW∆FIRMS)/'NEW∆FIRMS←5 1ρ1 0 (⊂10ρ0) (⊂10ρ0) 0'

********************************************************************************

TARG∆SEARCH∆PC

[0]      TARG∆SEARCH∆PC;KEEPS;NEXT;I;OK;II;Q2;Q3;Q7;LAYOFF;QEXPPNET;WHERE;L5
[1]     ⍝
[2]     ⍝ SIMULTANEOUS SEARCH FOR ALL FIRMS:
[3]     ⍝   NEXT - INDICATES NEXT STEP TO BE TRIED FOR EACH FIRM
```

```
[4]     ⍝   I    - INDICES OF THE FIRMS TO BE HANDLED IN A STEP
[5]     ⍝   OK   - TELLS WHICH FIRMS IN 'I' THAT ARE SUCCESSFUL IN A STEP
[6]     ⍝   II   - SHORT FOR 'OK/I'
[7]     ⍝
[8]      QEXPPNET←QEXPP-SHARE×(QEXPPIM+.×IO)[MARKET]
[9]      QPLANL←(⍴L)⍴¯1
[10]      ALWAYS'QEXPPNET>0 '
[11]      →DOIF(∨/QEXPPNET≤0) ◊ NULLIFY∆LINE←'****** NULLIFIED (NEG PR) (YEA
R=' ⍕THIS∆YEAR ◊ NULLIFY(QEXPPNET≤0) ◊ QEXPPNET←(QEXPPNET>0)/QEXPPNET
[12]      Q3←QFR1 L
[13]      Q7←Q3⌊QPLANQ
[14]      Q2←Q3⌊QEXPSU+MAXSTO-STO
[15]
[16]     ⍝
[17]      NEXT←(1 5 6)[1+(QPLANQ≥QTOP×WTIX×1-RES)+(QPLANQ>Q3)]
[18]     ⍝
[19]      WHERE←(⍴L)⍴9
[20]      II←(OK←QPLANQ[I]SAT L[I])/I←(NEXT=1)/⍳⍴QPLANQ
[21]      WHERE[II]←1
[22]      QPLANL[II]←L[II]
[23]      NEXT[I]←(2 10)[1+OK]
[24]     ⍝
[25]      II←(OK←Q2[I]SAT L[I])/I←(NEXT=2)/⍳⍴QPLANQ
[26]      QPLANQ[II]←(QPLANL[II]←L[II])×QEXPW[II]÷4×(1-QTARGM[II])×QEXPPNET[
II]
[27]      WHERE[II]←2
[28]      NEXT[I]←(3 4 10 10)[1+(Q2[I]=Q3[I])+2×OK]
[29]     ⍝
[30]      II←(OK←Q2[I]SAT I RFQ2 Q2)/I←(NEXT=3)/⍳⍴QPLANQ
[31]      WHERE[II]←3
[32]      QPLANL[II]←(1-QTARGM[II])×(QPLANQ[II]←Q2[II])×QEXPPNET[II]÷QEXPW[I
I]÷4
[33]      NEXT[I]←(4 10)[1+OK]
```

```
[34]    ⍝
[35]     II←(OK←QPLANQ[I]SAT I RFQ2 QPLANQ)/I←(NEXT=4)/⍳⍴QPLANQ
[36]     SOLVE II
[37]     (I RFQ2 QPLANQ)SOLVE∆MONEY(I RFQ2 Q2)
[38]     QPLANQ[II]←II QFR2 QPLANL
[39]     WHERE[II]←4
[40]     NEXT[I]←(7 10)[1+OK]
[41]    ⍝
[42]     II←(OK←QPLANQ[I]SAT L5[I]←I RFQ2 QPLANQ)/I←(NEXT=5)/⍳⍴L5←(⍴L)⍴LU+S
UM2 L
[43]     WHERE[II]←5
[44]     QPLANL[II]←II RFQ2 QPLANQ
[45]     NEXT[I]←(6 10)[1+OK]
[46]    ⍝
[47]     II←(OK←Q3[I]SAT L[I])/I←(NEXT=6)/⍳⍴QPLANQ
[48]     SOLVE II
[49]     L[I]SOLVE∆MONEY L5[I]
[50]     QPLANQ[II]←II QFR2 QPLANL
[51]     WHERE[II]←6
[52]     NEXT[I]←(7 10)[1+OK]
[53]    ⍝
[54]     II←(OK←Q7[I]SAT I RFQ2 Q7×(1-RES)÷(1-RESDOWN×RES))/I←(NEXT=7)/⍳⍴QP
LANQ
[55]     WHERE[II]←7
[56]     QPLANL[II]←(1-QTARGM[II])×(QPLANQ[II]←Q7[II])×QEXPPNET[II]÷QEXPW[I
I]÷4
[57]     RES[II]←1-QPLANQ[II]×(1-RES[II])÷II QFR2 QPLANL
[58]     RES[(~OK)/I]←RESDOWN×RES[(~OK)/I]
[59]     ALWAYS'(RES≥0)∧(RES≤RESMAX)'
[60]     NEXT[I]←(8 10)[1+OK]
[61]    ⍝
[62]     II←(OK←(I×0)SAT(I×0))/I←(NEXT=8)/⍳⍴QPLANQ
[63]     SOLVE II
```

```
[64]     0 SOLVE∆MONEY(I RFQ2 Q7)

[65]     QPLANQ[II]←II QFR2 QPLANL

[66]     WHERE[II]←8

[67]     NEXT[I]←(9 10)[1+OK]

[68]    ⍝

[69]     II←(OK←SAT∆LOW I)/I←(NEXT=9)/⍳⍴QPLANQ

[70]     QPLANQ[II]←QQ[II]

[71]     QPLANL[II]←II RFQ2 QPLANQ

[72]     WHERE[II]←9

[73]     NEXT[I]←(13 10)[1+OK]

[74]     KEEPS←(QPLANL<(QTOP÷TEC)×⍟QEXPPNET×TEC×WTIX×(1-RES)÷QEXPW÷4)∧WHERE
∊4 6 8

[75]     →DOIF 0<+/KEEPS ◊ (KEEPS/MHIST)×←LOWER∆MHIST

[76]    ⍝

[77]     →DOIF(0<⍴I←(NEXT=13)/⍳⍴QPLANQ) ◊ NULLIFY∆TARG(⍳⍴QPLANQ)∊I

[78]    ⍝ WARNING: 'NEXT' 'Q2' 'Q3' 'Q7' ARE NOT SHRINKED.

[79]    ⍝

[80]     AMAN←(0 1↓AMAN) L

[81]     LAYOFF←0⌈L-QPLANL

[82]     AMAN←0⌈AMAN-0⌈(+\AMAN)MINUS7 LAYOFF

[83]    ⍝

[84]     ALWAYS'QPLANQ≥0 '

[85]     ALWAYS' 0<SUM1(⍴MARKET)⍴1 '

[86]     ALWAYS'QPLANL≥0 '

[87]     I←⍳⍴QPLANQ

[88]     ALWAYS'QPLANQ SAT QPLANL'
```

```
*********************************************************************************

TARG∆SEARCH∆SHOW

[0]      TARG∆SEARCH∆SHOW;KEEPS;NEXT;I;OK;II;Q2;Q3;Q7;LAYOFF;QEXPPNET;WHERE;
L5

[1]     ⍝

[2]     ⍝ SIMULTANEOUS SEARCH FOR ALL FIRMS:
```

```
[3]    ⍝  NEXT - INDICATES NEXT STEP TO BE TRIED FOR EACH FIRM
[4]    ⍝  I    - INDICES OF THE FIRMS TO BE HANDLED IN A STEP
[5]    ⍝  OK   - TELLS WHICH FIRMS IN 'I' THAT ARE SUCCESSFUL IN A STEP
[6]    ⍝  II   - SHORT FOR 'OK/I'
[7]    ⍝ THIS FUNCTION ALSO SHOWS THE TARGET SEARCH OF THE FIRM 'ΔF'.
[8]    ⍝
[9]     →DOIF 0=⎕NC'ΔF' ◊ ⎕ARBOUT 7 ◊ 3 1 1ρ(40ρ'*')('   ENTER THE FIRM COD
E   ') ◊ ΔF←⍞
[10]    QEXPPNET←QEXPP-SHARE×(QEXPPIM+.×IO)[MARKET]
[11]    QPLANL←(ρL)ρ¯1
[12]    ALWAYS'QEXPPNET>0 '
[13]    →DOIF(∨/QEXPPNET≤0) ◊ NULLIFYΔLINE←'****** NULLIFIED (NEG PR) (YEA
R=' ⍕THISΔYEAR ◊ NULLIFY(QEXPPNET≤0) ◊ QEXPPNET←(QEXPPNET>0)/QEXPPNET
[14]    Q3←QFR1 L
[15]    Q7←Q3⌊QPLANQ
[16]    Q2←Q3⌊QEXPSU+MAXSTO-STO
[17]
[18]   ⍝
[19]    NEXT←(1 5 6)[1+(QPLANQ≥QTOP×WTIX×1-RES)+(QPLANQ>Q3)]
[20]   ⍝
[21]    WHERE←(ρL)ρ9
[22]    PPVAL←0 4ρ0
[23]    PPΔDATA 1
[24]    II←(OK←QPLANQ[I]SAT L[I])/I←(NEXT=1)/⍳ρQPLANQ
[25]    WHERE[II]←1
[26]    QPLANL[II]←L[II]
[27]    NEXT[I]←(2 10)[1+OK]
[28]    PPΔDATA 1
[29]   ⍝
[30]    II←(OK←Q2[I]SAT L[I])/I←(NEXT=2)/⍳ρQPLANQ
[31]    QPLANQ[II]←(QPLANL[II]←L[II])×QEXPW[II]÷4×(1-QTARGM[II])×QEXPPNET[
II]
[32]    WHERE[II]←2
```

```
[33]     NEXT[I]←(3 4 10 10)[1+(Q2[I]=Q3[I])+2×OK]

[34]     PP∆DATA 2

[35]    ⍝

[36]     II←(OK←Q2[I]SAT I RFQ2 Q2)/I←(NEXT=3)/ιρQPLANQ

[37]     WHERE[II]←3

[38]     QPLANL[II]←(1-QTARGM[II])×(QPLANQ[II]←Q2[II])×QEXPPNET[II]÷QEXPW[I

I]÷4

[39]     NEXT[I]←(4 10)[1+OK]

[40]     PP∆DATA 3

[41]    ⍝

[42]     II←(OK←QPLANQ[I]SAT I RFQ2 QPLANQ)/I←(NEXT=4)/ιρQPLANQ

[43]     SOLVE II

[44]     (I RFQ2 QPLANQ)SOLVE∆MONEY(I RFQ2 Q2)

[45]     QPLANQ[II]←II QFR2 QPLANL

[46]     WHERE[II]←4

[47]     NEXT[I]←(7 10 14)[1+OK]

[48]     PP∆DATA 4

[49]    ⍝

[50]     II←(OK←QPLANQ[I]SAT L5[I]←I RFQ2 QPLANQ)/I←(NEXT=5)/ιρL5←(ρL)ρLU+S

UM2 L

[51]     WHERE[II]←5

[52]     QPLANL[II]←II RFQ2 QPLANQ

[53]     NEXT[I]←(6 10)[1+OK]

[54]     PP∆DATA 5

[55]    ⍝

[56]     II←(OK←Q3[I]SAT L[I])/I←(NEXT=6)/ιρQPLANQ

[57]     SOLVE II

[58]     L[I]SOLVE∆MONEY L5[I]

[59]     QPLANQ[II]←II QFR2 QPLANL

[60]     WHERE[II]←6

[61]     NEXT[I]←(7 10 14)[1+OK]

[62]     PP∆DATA 6

[63]    ⍝
```

```
[64]     II←(OK←Q7[I]SAT I RFQ2 Q7×(1-RES)÷(1-RESDOWN×RES))/I←(NEXT=7)/⍳⍴QP
LANQ
[65]     WHERE[II]←7
[66]     QPLANL[II]←(1-QTARGM[II])×(QPLANQ[II]←Q7[II])×QEXPPNET[II]÷QEXPW[I
I]÷4
[67]     RES[II]←1-QPLANQ[II]×(1-RES[II])÷II QFR2 QPLANL
[68]     RES[(~OK)/I]←RESDOWN×RES[(~OK)/I]
[69]     ALWAYS'(RES≥0)∧(RES≤RESMAX)'
[70]     NEXT[I]←(8 10)[1+OK]
[71]     PP∆DATA 7
[72]    ⍝
[73]     II←(OK←(I×0)SAT(I×0))/I←(NEXT=8)/⍳⍴QPLANQ
[74]     SOLVE II
[75]     0 SOLVE∆MONEY(I RFQ2 Q7)
[76]     QPLANQ[II]←II QFR2 QPLANL
[77]     WHERE[II]←8
[78]     NEXT[I]←(9 10 14)[1+OK]
[79]     PP∆DATA 8
[80]    ⍝
[81]     II←(OK←SAT∆LOW I)/I←(NEXT=9)/⍳⍴QPLANQ
[82]     QPLANQ[II]←QQ[II]
[83]     QPLANL[II]←II RFQ2 QPLANQ
[84]     WHERE[II]←9
[85]     NEXT[I]←(13 10)[1+OK]
[86]     PP∆DATA 9
[87]     PP∆DATA 10
[88]     SHOW∆TARGET∆SEARCH
[89]     KEEPS←(QPLANL<(QTOP÷TEC)×⍟QEXPPNET×TEC×WTIX×(1-RES)÷QEXPW÷4)∧WHERE
∊4 6 8
[90]     →DOIF 0<+/KEEPS ⋄ (KEEPS/MHIST)×←LOWER∆MHIST
[91]     →DOIF(0<⍴I←(NEXT=13)/⍳⍴QPLANQ) ⋄ NULLIFY∆TARG(⍳⍴QPLANQ)∊I
[92]    ⍝ WARNING: 'NEXT' 'Q2' 'Q3' 'Q7' ARE NOT SHRINKED.
[93]    ⍝
```

```
[94]      AMAN←(O 1↓AMAN) L

[95]      LAYOFF←O⌈L-QPLANL

[96]      AMAN←O⌈AMAN-O⌈(+\AMAN)MINUS7 LAYOFF

[97]     ⍝

[98]      ALWAYS'QPLANQ≥O '

[99]      ALWAYS' O<SUM1(⍴MARKET)⍴1 '

[100]      ALWAYS'QPLANL≥O '

[101]      I←⍳⍴QPLANQ

[102]      ALWAYS'QPLANQ SAT QPLANL'
```

****

G. MERGE WITH FUNCTION

```
****************************************************************************

MERGEΔWITH

[0]     ΔA MERGEΔWITH ΔB;ΔEFF;ΔN;ΔG;ΔK;ΔΔM;ΔF;ΔCON;ΔT;P1;P2;⎕TRAP
[1]    ⍝
[2]    ⍝  This function merges two firms.
[3]    ⍝  Written by E.Taymaz  Jan.1989.
[4]    ⍝
[5]     ⎕TRAP←6 'E' '→⎕LC+1'
[6]    ⍝
[7]    ⍝ ΔF IS THE NEW FIRM. ΔCON IS THE TECHNOLOGY CONVERGENCE FACTOR.
[8]    ⍝ ΔEFF IS THE MAXIMUM CONVERGE TO THE INNOVATIVE FIRM.
[9]    ⍝ ΔT IS THE TIME PERIOD FOR CONVERGENCE (IN QUARTERS).
[10]    ⍝
[11]     ΔF ΔCON ΔEFF ΔT←ΔB
[12]     ⍎((⌊ΔF)≠(⌊ΔA))/'MESS←''FIRMS IN THE MERGEΔWITH FUNCTION ARE NOT IN
 THE SAME INDUSTRY'' ⋄ →ERR'
[13]    ⍝
[14]    ⍝ P1 AND P2 ARE FIRMS' PLACE IN VARIABLES.
[15]    ⍝
[16]     P2←(+\ORIGMARKET=⌊ΔA)⍳100×(ΔA-⌊ΔA)
[17]     P1←(+\ORIGMARKET=⌊ΔF)⍳100×(ΔF-⌊ΔF)
[18]     →(P1=P2)/0
[19]     →((LEFT[P1]=0)∨LEFT[P2]=0)/0
[20]     P1←(+/LEFT[⍳P1])
[21]     P2←(+/LEFT[⍳P2])
[22]    ⍝
[23]    ⍝
[24]    ⍝
[25]     AMAN[P2;]+←AMAN[P1;]
[26]     BAD[P2]←⌊/BAD[P1] BAD[P2]
```

```
[27]      BIG←K1 WEIGHT BIG

[28]      DNW←NW WEIGHT DNW

[29]      ΔΔM←((M[P2]×QSNET[P2])+(M[P1]×QSNET[P1]))÷(QSNET[P1]+QSNET[P2])

[30]      CUMM[P2]←((CUMSNET[P1]+CUMSNET[P2]-CUMWS[P1]+CUMWS[P2])÷(CUMSNET[P
1]+CUMSNET[P2]))

[31]      CHM[P2]←CUMM[P2]-ΔΔM

[32]      CUMINV[P1]+←CUMINV[P1]

[33]      CUMSNET[P2]+←CUMSNET[P1]

[34]      QIMQ[P2;]+←QIMQ[P1;]

[35]      ΔK←(ρK1)ρ0

[36]      ΔK[P2]←K1[P2]+K2[P2]+K3[P2]

[37]      ΔK[P1]←K1[P1]+K2[P1]+K3[P1]

[38]      DELAYΔINV[P2;]+←DELAYΔINV[P1;]

[39]      DVA←VA WEIGHT DVA

[40]      DS←S WEIGHT DS

[41]      DW←VA WEIGHT DW

[42]      DP←S WEIGHT DP

[43]      DQ←DVA-DP

[44]      CUMQ[P2]+←CUMQ[P1]

[45]      CUMS[P2]+←CUMS[P1]

[46]      CUML[P2]+←CUML[P1]

[47]      CUMWS[P2]+←CUMWS[P1]

[48]      CUMSU[P2]+←CUMSU[P1]

[49]      CUMVA[P2]+←CUMVA[P1]

[50]      P[P2]←CUMS[P2]÷CUMSU[P2]

[51]      W[P2]←CUMWS[P2]÷CUML[P2]

[52]      EXPDP←S WEIGHT EXPDP

[53]      EXPDS←S WEIGHT EXPDS

[54]      EXPDW←S WEIGHT EXPDW

[55]      HISTDP←S WEIGHT HISTDP

[56]      HISTDPDEV←S WEIGHT HISTDPDEV

[57]      HISTDPDEV2←S WEIGHT HISTDPDEV2

[58]      HISTDW←S WEIGHT HISTDW
```

```
[59]     HISTDWDEV←S WEIGHT HISTDWDEV
[60]     HISTDWDEV2←S WEIGHT HISTDWDEV2
[61]     HISTDS←S WEIGHT HISTDS
[62]     HISTDSDEV←S WEIGHT HISTDSDEV
[63]     HISTDSDEV2←S WEIGHT HISTDSDEV2
[64]     IMBIG←S WEIGHT IMBIG
[65]     IMSMALL←S WEIGHT IMSMALL
[66]     INVEFF←QTOP WEIGHT INVEFF
[67]     QDVA←QVA WEIGHT QDVA
[68]     QDW←(L×QW)WEIGHT QDW
[69]     QEXPP←QS WEIGHT QEXPP
[70]     QEXPS←QS WEIGHT QEXPS
[71]     QEXPSU←QSU WEIGHT QEXPSU
[72]     QEXPW←QW WEIGHT QEXPW
[73]     QDP←QS WEIGHT QDP
[74]     QDQ←QQ WEIGHT QDQ
[75]     QDS←QS WEIGHT QDS
[76]     QP←QS WEIGHT QP
[77]     QS[P2]+←QS[P1]
[78]     QQ[P2]+←QQ[P1]
[79]     K1BOOK[P2]+←K1BOOK[P1]
[80]     QW←L WEIGHT QW
[81]     DL←L WEIGHT DL
[82]     L[P2]+←L[P1]
[83]     M[P2]←ΔΔM
[84]     QDSUFOR←QSUFOR WEIGHT QDSUFOR
[85]     QSUFOR[P2]+←QSUFOR[P1]
[86]     QIMQ[P2;]+←QIMS[P1;]
[87]     QINV[P2]+←QINV[P1]
[88]     QINVLAG[P2]+←QINVLAG[P1]
[89]     QOPTSU[P2]+←QOPTSU[P1]
[90]     QPLANL[P2]+←QPLANL[P1]
[91]     QPLANQ[P2]+←QPLANQ[P1]
```

```
[92]      QSDOM[P2]+←QSDOM[P1]
[93]      QSFOR[P2]+←QSFOR[P1]
[94]      QSNET[P2]+←QSNET[P1]
[95]      QSU[P2]+←QSU[P1]
[96]      QSUDOM[P2]+←QSUDOM[P1]
[97]      QTARGM←K1 WEIGHT QTARGM
[98]      QVA[P2]+←QVA[P1]
[99]      REALLYΔBAD[P2]←⌊/REALLYΔBAD[P2] REALLYΔBAD[P1]
[100]      RES←QTOP WEIGHT RES
[101]      SHARE←S WEIGHT SHARE
[102]      SMALL←S WEIGHT SMALL
[103]      SNET[P2]+←SNET[P1]
[104]      VA[P2]+←VA[P1]
[105]      X←S WEIGHT X
[106]      RW←S WEIGHT RW
[107]      RSUBSΔCASH←S WEIGHT RSUBSΔCASH
[108]      RSUBSΔEXTRA←S WEIGHT RSUBSΔEXTRA
[109]      CUMINTPAYF[P2]+←CUMINTPAYF[P1]
[110]      CUMDEPR[P2]+←CUMDEPR[P1]
[111]      CUMTAXF[P2]+←CUMTAXF[P1]
[112]      CUMDIV[P2]+←CUMDIV[P1]
[113]      CUMSUBSF[P2]+←CUMSUBSF[P1]
[114]      CUMCHBWF[P2]+←CUMCHBWF[P1]
[115]      CUMCHK2F[P2]+←CUMCHK2F[P1]
[116]      Q[P2]+←Q[P1]
[117]      S[P2]+←S[P1]
[118]      ΔEFF←ΔEFF×⌈/TEC[P2] TEC[P1]
[119]      TEC←QTOP WEIGHT TEC
[120]      →(ΔEFF≤TEC[P2])/JMP
[121]      ΔG←TEC[P2]+(ΔEFF-TEC[P2])×((⍳ΔT)÷ΔT)*1÷ΔCON
[122]      ΔG←TEC[P2] ΔG ◊ ΔG←ΔG[1+⍳ΔT]÷ΔG[⍳ΔT]
[123]      ΔN←'UΔTΔ' (⍕⌊ΔF) '⍙' ⍕100×(ΔF-⌊ΔF)
[124]      ⍎ΔN '←ΔG'
```

```
[125]     'PRODFRONT'MODADD'TEC←(QωUPDATE∆TECH∆OF ' ⍕∆F
[126]    JMP:
[127]     QTOP[P2]+←QTOP[P1]
[128]     QTOPMAX[P2]+←QTOPMAX[P1]
[129]     MHIST←K1 WEIGHT MHIST
[130]     K1[P2]+←K1[P1]
[131]     K2[P2]+←K2[P1]
[132]     IMSTO[P2;]+←IMSTO[P1;]
[133]     STO[P2]+←STO[P1]
[134]     BW[P2]+←BW[P1]
[135]     QM[P2]←1-(L[P2]×QW[P2])÷4×QSNET[P2]
[136]     OUT←(ρMARKET)ρO
[137]     OUT[P1]←1
[138]     NULLIFY∆LINE←'****** MERGED WITH ' (⍕∆A) '   (YEAR=' (⍕THIS∆YEAR)
 ' QUARTER=1'
[139]     NULLIFY OUT
[140]     →O
[141]    ERR:
[142]     STOP∆HERE MESS
```

****

H. MOSES CALIB FUNCTIONS

ADDRESULT
ADDRESULTB
CHECK MM
CHECK PARS
CLEAR BLANK
COMPUTE L
COMPUTE S
DISTANCE
DOIF
EVA
GET METHOD
KEEP NAMES
MAKEVECTOR
MPAR
PREPARE RESULTS
READ DATA
SAVE DATA
START I
START L
START R
START S
START Z
VIA
<u>INTERACT</u>

```
*****************************************************************************

ADDRESULT

[0]     ADDRESULT RRR

[1]     RESULTS←RESULTS [1](1 2+NPARS+NVARS)ρRRR (⍎⍕ΔPARS[1;]) DISTANCE

*****************************************************************************

ADDRESULTB

[0]     ADDRESULTB R

[1]     RESULTS←RESULTS [1](1 2+NPARS+NVARS)ρR (⍎⍕ΔPARS[1;]) (⊂'EXCEED MAX
DEV') (NVARS)ρ⊂'...'

*****************************************************************************

CHECKΔMM

[0]     R←CHECKΔMM;ΔV

[1]     ΔV←⍎⍕ΔPARS[1;ΔP]

[2]     →DOIF(ΔPARS[2;ΔP]≥ΔV)∧ΔPARS[3;ΔP]≤ΔV ◊ R←1 ◊ →0

[3]     R←0

[4]     ΔPARS[6;ΔP]×←¯1

[5]     MPAR ΔP

*****************************************************************************

CHECKΔPARS

[0]     R←CHECKΔPARS

[1]     →DOIF 6≠1↑ρΔPARS ◊ →ERR

[2]     →DOIF 0=∧/(ΔPARS[2;]≥ΔPARS[4;]) ΔPARS[3;]≤ΔPARS[4;] ◊ →ERR

[3]     R←0 ◊ →0

[4]    ERR:' ' ◊ '*****  ERROR IN PARAMETER MATRIX  *****'

[5]     R←1
```

****

CLEAR∆BLANK

```
[0]      ∆∆∆NEW←CLEAR∆BLANK ∆∆∆OLD;∆I;∆RANK

[1]     ⍝

[2]     ⍝  This function deletes all blanks from a vector of names.

[3]     ⍝  Written by E.Taymaz  Dec.1989

[4]     ⍝

[5]      ∆RANK←ρ∆∆∆OLD

[6]      ∆∆∆NEW←∆RANKρ''

[7]      ∆I←0

[8]     BEG:∆I←∆I+1

[9]      →(∆I>∆RANK)/END

[10]      ∆∆∆NEW[∆I]←⊂((~((⊃∆∆∆OLD[∆I])=' '))/(⊃∆∆∆OLD[∆I]))

[11]      →BEG

[12]     END:
```

****

COMPUTE∆L

```
[0]      Z←COMPUTE∆L;AA;BB;NT;II;DEV;ADEV;RDEV

[1]      NT←¯1+ρTESTVALUES

[2]      II←0

[3]      AA←(NYEAR NT)ρ0

[4]     BAS:II←II+1

[5]      AA[;II]←(⊃RESULTSALL[II])[;I]

[6]      →(II<NT)/BAS

[7]      AA←AA[⍳J;] ◊ BB←BASECASE[⍳J;I]

[8]      BB←⍉(NT J)ρBB

[9]      DEV←BB-AA

[10]      ADEV←(+/|DEV[;1]-10×DEV[;2]) +/|DEV[;4]-10×DEV[;3]

[11]      RDEV←(+/0.5×(|DEV[;1])+10×|DEV[;2]) +/0.5×(|DEV[;4])+10×|DEV[;3]

[12]      Z←0.001×⌈¯0.5+1000×ADEV÷RDEV
```

```
**************************************************************************

COMPUTE∆S

[0]     Z←COMPUTE∆S;AA;BB;NT;II;DEV;ADEV;RDEV

[1]     NT←¯1+⍴TESTVALUES

[2]     II←0

[3]     AA←(NYEAR NT)⍴0

[4]    BAS:II←II+1

[5]     AA[;II]←(⊃RESULTSALL[II])[;I]

[6]     →(II<NT)/BAS

[7]     AA←AA[⍳J;] ◊ BB←BASECASE[⍳J;I]

[8]     BB←⍉(NT J)⍴BB

[9]     DEV←BB-AA

[10]     ADEV←(+/|+/DEV[;1 4]) +/|+/DEV[;2 3]

[11]     RDEV←(+/0.5×+/|DEV[;1 4]) +/0.5×+/|DEV[;2 3]

[12]     Z←0.001×⌈¯0.5+1000×ADEV÷RDEV


**************************************************************************

DISTANCE

[0]     R←DISTANCE;OLD;NEW;ST;MD;⎕DIV;WGS

[1]     ⎕DIV←1

[2]     WGS←∆VARS[2;]÷+/∆VARS[2;]

[3]     OLD←OLDTAB ◊ NEW←NEWTAB

[4]     →(SVAR='N')/JMP

[5]     →(1=NYEAR)/JMP

[6]     MD←(1÷NYEAR)×(NYEAR NVARS)⍴+⌿OLDTAB

[7]     ST←(NYEAR NVARS)⍴(+⌿(OLDTAB-MD)*2)*0.5

[8]     OLD←(OLDTAB-MD)÷ST

[9]     NEW←(NEWTAB-MD)÷ST

[10]    JMP:

[11]     →DOIF SMIN='A' ◊ R←⍋⍒+/WGS×+⌿|OLD-NEW ◊ R←R ⍋⍒+⌿|OLD-NEW ◊ →0

[12]     →DOIF SMIN='M' ◊ R←⍋⍒⌈/WGS×⌈⌿|OLD-NEW ◊ R←R ⍋⍒⌈⌿OLD-NEW ◊ →0

[13]     R←⍋⍒+/WGS×+⌿(OLD-NEW)*2 ◊ R←R ⍋⍒+⌿(OLD-NEW)*2
```

```
******************************************************************************

DOIF

[0]      R←DOIF TEST
[1]     ⍝Return next line number if TEST false
[2]      R←(~TEST)/1+1↓2↑⎕LC


******************************************************************************

EVA

[0]      RR←EVA
[1]      ADDRESULT ΔE
[2]      RR←1
[3]      →(ΔE≠1)/J1
[4]      OLDDIST←DISTANCE[1]
[5]      MPAR ΔP
[6]      →SON
[7]     J1:
[8]      →(0>DISTANCE[1]-OLDDIST)/J2
[9]      ΔPARS[6;ΔP]×←¯1 ◊ MPAR ΔP
[10]      →DOIF 2≠ΔIT ◊ →JJ2
[11]      MPAR ΔP
[12]      →DOIF CHECKΔMM ◊ →SON
[13]     JJ2:
[14]      ΔP+←1 ◊ RR←0
[15]      →DOIF ΔP≤NPARS ◊ RR←1 ◊ ΔIT←1 ◊ MPAR ΔP
[16]      →SON
[17]     J2:
[18]      →DOIF CONVERGENCE≥|DISTANCE[1]-OLDDIST ◊ ΔP+←1 ◊ ΔIT←0 ◊ →JJ3
[19]      →DOIF CONVERGENCE<|DISTANCE[1]-OLDDIST ◊ MPAR ΔP ◊ →DOIF~CHECKΔMM
◊ ΔP+←1 ◊ ΔIT←0
[20]     JJ3:
[21]      OLDDIST←DISTANCE[1]
```

```
[22]     →DOIF ∆P>NPARS ◊ RR←0 ◊ →SON
[23]    SON:
[24]

**********************************************************************

GET∆METHOD

[0]     GET∆METHOD;CVAR;TVAR;NTVAR;NYEAR;MINSTEPF;NFAILF;NSUCF;STEPRF;STEPI
F;SDEV;SMET;SMIN;SVAR;CONVERGENCE;MITER;∆
[1]    ⍝ →CHECK∆PARS/0
[2]    TEK:3 1ρ' ' ◊ 2 70ρ'**' ◊ 2 1ρ' '
[3]     ⎕ARBOUT 7
[4]     '      PROCEDURE '
[5]     '      (Global random search / Local random search / Iterative searc
h '
[6]     ⍞←'      Seed number / Zellner''s test)     : ' ◊ SMET←⍞ ◊ ' '
[7]     SMET←(SMET 'R')[41] ◊ ⍎(0=+/SMET='RLSIZ')/'SMET←''R'''
[8]     →(SMET≠'Z')/JMPZ
[9]     ⍞←'      NAME OF THE CONTROL VARIABLE       : ' ◊ CVAR←⍞ ◊ ' '
[10]     CVAR←40↓CVAR
[11]     ⍞←'      NAMES OF THE TEST VARIABLES        : ' ◊ TVAR←⍞ ◊ ' '
[12]     TVAR←MAKEVECTOR 40↓TVAR ◊ NTVAR←ρTVAR
[13]     ⍞←'      MAX CONTROL VARIABLE CHANGE (10%) : ' ◊ STEPRF←⍞ ◊ ' '
[14]     →DOIF 41≥ρSTEPRF ◊ STEPRF← 10
[15]     →DOIF 41<ρSTEPRF ◊ STEPRF←⍎41↓STEPRF
[16]     ⍞←'      SIMULATION PERIOD                  : ' ◊ NYEAR←⍞ ◊ ' '
[17]     NYEAR←⍎40↓NYEAR
[18]     →JMP
[19]     JMPZ:
[20]     '      MINIMIZE Squared / Absolute / '
[21]     ⍞←'                       Maximum distance   : ' ◊ SMIN←⍞ ◊ ' '
[22]     SMIN←(SMIN 'S')[41] ◊ ⍎(0=+/SMIN='SAM')/'SMIN←''S'''
[23]     ⍞←'      STANDARDIZE VARIABLES (No/Yes)    : ' ◊ SVAR←⍞ ◊ ' '
```

```
[24]    SVAR←(SVAR 'N')[41] ◊ ⍎(O=+/SVAR='YN')/'SVAR←''N'''

[25]    →(SMET='LI')/JMPL JMPI

[26]    ⍞←'      MAXIMUM DEVIATION IN A YEAR (1)   : ' ◊ SDEV←⍞ ◊ ' '

[27]    →DOIF 40≥ρSDEV ◊ SDEV← 1

[28]    →DOIF 40<ρSDEV ◊ SDEV←⍎40↓SDEV

[29]    ⍞←'      NUMBER OF EXPERIMENTS (25)        : ' ◊ MITER←⍞ ◊ ' '

[30]    →DOIF 40≥ρMITER ◊ MITER← 25

[31]    →DOIF 40<ρMITER ◊ MITER←⍎40↓MITER

[32]    →JMP

[33]   JMPL:

[34]    ⍞←'      MAXIMUM DEVIATION IN A YEAR (1)   : ' ◊ SDEV←⍞ ◊ ' '

[35]    →DOIF 40≥ρSDEV ◊ SDEV← 1

[36]    →DOIF 40<ρSDEV ◊ SDEV←⍎40↓SDEV

[37]   S3:

[38]    ⍞←'      MAX NUMBER OF STEP REDUCTIONS (1): ' ◊ MINSTEPF←⍞ ◊ ' '

[39]    →DOIF 40≥ρMINSTEPF ◊ MINSTEPF← 1

[40]    →DOIF 40<ρMINSTEPF ◊ MINSTEPF←⍎40↓MINSTEPF

[41]    →DOIF MINSTEPF<1 ◊ ⎕ARBOUT 7 ◊ '*** MINIMUM STEP SIZE FACTOR SHOUL

D BE HIGHER THAN ONE! ***' ◊ →S3

[42]    ⍞←'      # FAILURES BEFORE Δ STEP (20)     : ' ◊ NFAILF←⍞ ◊ ' '

[43]    →DOIF 40≥ρNFAILF ◊ NFAILF← 20

[44]    →DOIF 40<ρNFAILF ◊ NFAILF←⍎40↓NFAILF

[45]    ⍞←'      # SUCCESSES BEFORE Δ STEP (20)    : ' ◊ NSUCF←⍞ ◊ ' '

[46]    →DOIF 40≥ρNSUCF ◊ NSUCF← 20

[47]    →DOIF 40<ρNSUCF ◊ NSUCF←⍎40↓NSUCF

[48]   S1:

[49]    ⍞←'      STEP SIZE REDUCTION FACTOR (.5)   : ' ◊ STEPRF←⍞ ◊ ' '

[50]    →DOIF 40≥ρSTEPRF ◊ STEPRF← 0.5

[51]    →DOIF 40<ρSTEPRF ◊ STEPRF←⍎40↓STEPRF

[52]    →DOIF STEPRF>1 ◊ ⎕ARBOUT 7 ◊ '*** STEP SIZE REDUCTION FACTOR SHOUL

D BE LESS THAN ONE! ***' ◊ →S1

[53]   S2:

[54]    ⍞←'      STEP SIZE INCREASE FACTOR (1)     : ' ◊ STEPIF←⍞ ◊ ' '
```

```
[55]     →DOIF 40≥⍴STEPIF ◊ STEPIF← 1
[56]     →DOIF 40<⍴STEPIF ◊ STEPIF←⍎40↓STEPIF
[57]     →DOIF STEPIF<1 ◊ ⎕ARBOUT 7 ◊ '*** STEP SIZE INCREASE FACTOR SHOULD
 BE HIGHER THAN ONE! ***' ◊ →S2
[58]    JMPI:
[59]     ⍞←'     CONVERGENCE VALUE (0.01)          : ' ◊ CONVERGENCE←⍞ ◊ ' '
[60]     →DOIF 40≥⍴CONVERGENCE ◊ CONVERGENCE← 0.01
[61]     →DOIF 40<⍴CONVERGENCE ◊ CONVERGENCE←0⌈⍎40↓CONVERGENCE
[62]    JMP:
[63]     ⍞←'     CHANGE THESE VALUES (No/Yes)     : ' ◊ ∆←⍞ ◊ ' '
[64]     →('Y'=(∆ 'N')[41])/TEK
[65]     2 1⍴' ' ◊ 2 70⍴'**' ◊ 2 1⍴' '
[66]     ⍎⊃(SMET='RLSIZ')/'START∆R' 'START∆L' 'START∆S' 'START∆I' 'START∆Z'
```

```
********************************************************************************
```

KEEP∆NAMES

```
[0]     KEEP∆NAMES ∆NAMES;∆EXNAMES
[1]    ⍝
[2]    ⍝  This function deletes all functions and variables in the current
[3]    ⍝  workspace other than those given in ∆NAMES as a vector of names.
[4]    ⍝  Written by E.Taymaz  Dec.1989
[5]    ⍝
[6]     ∆EXNAMES←CLEAR∆BLANK↓⎕NL 2 3
[7]     ∆NAMES←CLEAR∆BLANK ∆NAMES
[8]     ∆EXNAMES←↑(~(∆EXNAMES∊∆NAMES))/∆EXNAMES
[9]     ⎕EX ∆EXNAMES
```

```
********************************************************************************
```

MAKEVECTOR

```
[0]     DUMMY←MAKEVECTOR NAMES;POS
[1]     DUMMY←0⍴''
[2]    L:→(0=⍴NAMES)/0
```

```
[3]      POS←( NAMES)⍳' '

[4]      →(1=POS)/LL

[5]      DUMMY ←⊂(POS-1)↑NAMES

[6]     LL:NAMES←POS↓NAMES

[7]      →L


********************************************************************************

MPAR


[0]      MPAR R

[1]      ⍎(⍕ΔPARS[1;R]) '+←' ⍕ΔPARS[5;R]××ΔPARS[6;R]

[2]      ΔPARS[4;R]+←ΔPARS[5;R]××ΔPARS[6;R]


********************************************************************************

PREPAREΔRESULTS


[0]      PREPAREΔRESULTS;I;RES

[1]      RESULTS←0⍴''

[2]      RESULTS←⊂' CONTROL VARIABLE : ' ⍕CVAR

[3]      I←0

[4]     BEG:I←I+1

[5]      RESULTS ←⊂' '

[6]      RESULTS ←⊂' VARIABLE TESTED : ' ⍕TVAR[I]

[7]      RESULTS ←⊂' '

[8]      RESULTS ←⊂' Symmetry statistics'

[9]      RESULTS ←⊂' '

[10]      RES←((2+NYEAR) 1+NTVAR)⍴''

[11]      RES[1;]←(⊂'Year') (⊂(5 1⍕100×TESTVALUES[5]) '%') ⊂(5 1⍕100×TESTVAL
UES[4]) '%'

[12]      RES[2;]←(⊂' ') (⊂(5 1⍕100×TESTVALUES[2]) '%') ⊂(5 1⍕100×TESTVALUES
[3]) '%'

[13]      J←0

[14]     BEGS:J←1+J

[15]      RES[2+J;]←J COMPUTEΔS
```

```
[16]      →(J<NYEAR)/BEGS
[17]      RESULTS ←⊂RES
[18]      RESULTS ←⊂' '
[19]      RESULTS ←⊂' Linearity statistics'
[20]      RESULTS ←⊂' '
[21]      RES←((2+NYEAR) 1+NTVAR)⍴''
[22]      RES[1;]←(⊂'Year') (⊂(5 1⍕100×TESTVALUES[2]) '%') ⊂(5 1⍕100×TESTVAL
UES[5]) '%'
[23]      RES[2;]←(⊂' ') (⊂(5 1⍕100×TESTVALUES[3]) '%') ⊂(5 1⍕100×TESTVALUES
[4]) '%'
[24]      J←0
[25]     BEGL:J←1+J
[26]      RES[2+J;]←J COMPUTEΔL
[27]      →(J<NYEAR)/BEGL
[28]      RESULTS ←⊂RES
[29]      →(I<NTVAR)/BEG
[30]      RESULTS←((⍴RESULTS) 1)⍴RESULTS
```

```
***************************************************************************
READΔDATA

[0]      READΔDATA R;RANK;ΔII;ΔΔΔ
[1]      RANK←⍴R
[2]      ΔII←1
[3]     START:
[4]      ΔΔΔ←⎕FREAD 1 ΔII
[5]      ⍎(⍕R[ΔII]) '←ΔΔΔ'
[6]      →(RANK≥ΔII←ΔII+1)/START

***************************************************************************
SAVEΔDATA

[0]      SAVEΔDATA R;RANK;ΔII
[1]      →DOIF 1=1↑⎕FNUMS ◊ 'DELETE.CAL'⎕FERASE 1
```

```
[2]      'DELETE.CAL'⎕FCREATE 1

[3]      RANK←ρR

[4]      ∆II←1

[5]     START:

[6]      (⍎⊃R[∆II])⎕FAPPEND 1

[7]      →(RANK≥∆II←∆II+1)/START


*****************************************************************************

START∆I


[0]      START∆I;∆PARS;∆VARS;∆KEEP;OLDVAR;NPARS;NYEAR;NVARS;∆I;∆E;∆P;∆IT;NPA
RS;NYEAR;NVARS;OLDTAB;NEWTAB;KEEPVAR

[1]      KEEPVAR←CLEAR∆BLANK↓⎕NL 2

[2]      SAVE∆DATA KEEPVAR

[3]      ∆VARS←⍎⊃MENU[10;1] ◊ ∆PARS←⍎⊃MENU[16;1]

[4]      NPARS←¯1↑ρ∆PARS

[5]      NYEAR←¯2+1↑ρ∆VARS

[6]      NVARS←¯1↑ρ∆VARS

[7]      OLDTAB←2 0↓∆VARS

[8]      ∆P←1

[9]      ∆IT←0

[10]      ∆E←0

[11]     ⍝

[12]      RESULTS←(1 (2+NPARS+NVARS))ρ(⊂'EXP NO') ∆PARS[1;] (⊂'DISTANCE')

[13]      START∆ENT1

[14]      ∆KEEP←(↓⎕NL 2) (↓⎕NL 3) (⊂'OLDDIST') ⊂'∆KEEP'

[15]     STR:

[16]      ∆E+←1 ◊ ∆IT+←1

[17]      →DOIF 1≠∆E ◊ READ∆DATA KEEPVAR ◊ KEEP∆NAMES ∆KEEP

[18]      NEWTAB←0×OLDTAB

[19]      ⍎(⍕∆PARS[1;]) '←∆PARS[4;]'

[20]      ⎕RL←16807

[21]      (⍕⎕TS[4]) ':' (¯2↑'0' ⍕⎕TS[5]) '     Experiment : ' (⍕∆E) '   Para
meter : ' ⍕∆PARS[1;∆P]
```

```
[22]     ∆I←0
[23]    START:
[24]     →(NYEAR<∆I←∆I+1)/END
[25]     THIS∆YEAR+←1
[26]     YEAR
[27]     LAST∆YEAR+←1
[28]     NEWTAB[∆I;]←⍎⍕∆VARS[1;]
[29]     →START
[30]    END:
[31]     'Iteration: ' (⍕∆IT) '    Distance: ' ⍕DISTANCE[1]
[32]     →EVA/STR
[33]    ENSON:
[34]     'DELETE.CAL'⎕FERASE 1


******************************************************************************

START∆L

[0]     START∆L;NSTEPR;NSUC;NFAIL;NSTEPR;MIND;∆∆∆;∆VARS;∆PARS;∆KEEP;NNUM;∆P
ARS;OLDVAR;NPARS;NYEAR;NVARS;∆I;∆IT;NPARS;NYEAR;NVARS;OLDTAB;NEWTAB;KEEPVAR
[1]     KEEPVAR←CLEAR∆BLANK↓⎕NL 2
[2]     SAVE∆DATA KEEPVAR
[3]     ∆VARS←⍎⊃MENU[10;1] ⋄ ∆PARS←⍎⊃MENU[16;1]
[4]     ⍎(⍕∆PARS[1;]) '←∆PARS[4;]'
[5]     NPARS←¯1↑⍴∆PARS
[6]     NYEAR←¯2+1↑⍴∆VARS
[7]     NVARS←¯1↑⍴∆VARS
[8]     OLDTAB←2 0↓∆VARS ⋄ BESTTAB←OLDTAB
[9]     MIND←1E100
[10]     ∆IT←0
[11]     NSUC←NFAIL←NSTEPR←0
[12]    ⍝
[13]     RESULTS←(1 (2+NPARS+NVARS))⍴(⊂'EXP NO') ∆PARS[1;] (⊂'DISTANCE') ∆V
ARS[1;]
```

```
  ΔPARS←(1 NPARS)ρ0
  STARTΔENT1
  ΔKEEP←(↓⎕NL 2) (↓⎕NL 3) (⊂'OLDDIST') ⊂'ΔKEEP'
  ΔΔΔ←NPARSρ0
 STR:
  →(ΔIT=0)/ILKI
  →DOIF NFAIL≥NFAILF ◊ ΔΔΔ←ΔPARS[5;] ◊ ((ΔΔΔ≠1)/ΔΔΔ)←(ΔΔΔ≠1)/STEPRF×
ΔΔΔ ◊ ΔPARS[5;]←ΔΔΔ ◊ NFAIL←0 ◊ NSTEPR+←1
  →DOIF NSUC≥NSUCF ◊ ΔΔΔ←ΔPARS[5;] ◊ ((ΔΔΔ≠1)/ΔΔΔ)←(ΔΔΔ≠1)/STEPIF×ΔΔ
Δ ◊ ΔPARS[5;]←ΔΔΔ ◊ NSUC←0 ◊ NSTEPR-←1
  →(NSTEPR>MINSTEPF)/ENDD
  READΔDATA KEEPVAR ◊ KEEPΔNAMES ΔKEEP
  ΔΔΔ←1○(0.001×¯1001+?NPARSρ2001)×○1
  ΔΔΔ÷←(+/ΔΔΔ*2)*0.5 ◊ ((ΔPARS[5;]=1)/ΔΔΔ)←(ΔPARS[5;]=1)/¯2+?NPARSρ3
  ΔΔΔ←ΔΔΔ×ΔPARS[5;]
  ⍎(⍕ΔPARS[1;]) '←ΔPARS[2;]⌊ΔPARS[3;]⌈ΔPARS[4;]+ΔΔΔ'
 ILKI:
  ΔPARS←ΔPARS [1](1 NPARS)ρΔPARS[4;]
  ΔIT+←1
  NEWTAB←0×OLDTAB
  ⎕RL←16807
  ΔI←0
  '         EXPERIMENT  ' (⍕ΔIT)
 START:
  →(NYEAR<ΔI←ΔI+1)/END
  THISΔYEAR+←1
  (⍕⎕TS[4]) ':' (¯2↑'0' ⍕⎕TS[5]) '    Year         : ' ⍕THISΔYEAR
  YEAR
  LASTΔYEAR+←1
  NEWTAB[ΔI;]←⍎⍕ΔVARS[1;]
  →DOIF SDEV≤⊃⌈/|NEWTAB[ΔI;]-OLDTAB[ΔI;] ◊ ADDRESULTB ΔIT ◊ →STR
  →START
 END:
```

```
[45]     →DOIF ΔIT=1 ◊ MIND←DISTANCE[1] ◊ BESTTAB←NEWTAB ◊ →CNT
[46]     →DOIF DISTANCE[1]>MIND÷1+CONVERGENCE ◊ NSUC←0 ◊ NFAIL+←1 ◊ →CNT
[47]     MIND←DISTANCE[1] ◊ BESTTAB←NEWTAB
[48]     NSUC+←1 ◊ NFAIL←0
[49]     ΔPARS[4;]←ΔPARS[2;]⌊ΔPARS[3;]⌈ΔPARS[4;]+ΔΔΔ
[50]    CNT:
[51]     (⍕⎕TS[4]) ':' (¯2↑'0' ⍕⎕TS[5]) '    Distance    : ' ⍕DISTANCE[1]
[52]     '          Min distance: ' ⍕MIND
[53]     ADDRESULT ΔIT
[54]     →STR
[55]    ENDD:
[56]     2 1⍴' ' ◊ '***** CALIBRATION IS COMPLETED. *****' ◊ 2 1⍴' '
[57]     'DELETE.CAL'⎕FERASE 1
```

****

STARTΔR

```
[0]     STARTΔR;MIND;ΔVARS;ΔPARS;ΔKEEP;NNUM;ΔPARS;OLDVAR;NPARS;NYEAR;NVARS;
ΔI;ΔIT;NPARS;NYEAR;NVARS;OLDTAB;NEWTAB;KEEPVAR
[1]     KEEPVAR←CLEARΔBLANK↓⎕NL 2
[2]     SAVEΔDATA KEEPVAR
[3]     ΔVARS←⍎⊃MENU[10;1] ◊ ΔPARS←⍎⊃MENU[16;1]
[4]     ⍎(⍕ΔPARS[1;]) '←ΔPARS[4;]'
[5]     NNUM←1+⌊(ΔPARS[2;]-ΔPARS[3;])÷ΔPARS[5;]
[6]     NPARS←¯1↑⍴ΔPARS
[7]     NYEAR←¯2+1↑⍴ΔVARS
[8]     NVARS←¯1↑⍴ΔVARS
[9]     OLDTAB←2 0↓ΔVARS
[10]     BESTTAB←OLDTAB
[11]     MIND←1E100
[12]     ΔIT←0
[13]    ⍝
[14]     RESULTS←(1 (2+NPARS+NVARS))⍴(⊂'EXP NO') ΔPARS[1;] (⊂'DISTANCE') ΔV
ARS[1;]
```

```
[15]     ∆PARS←(1 NPARS)⍴0
[16]     START∆ENT1
[17]     ∆KEEP←(↓⎕NL 2) (↓⎕NL 3) (⊂'OLDDIST') ⊂'∆KEEP'
[18]    STR:
[19]     →DOIF ∆IT≠0 ◊ READ∆DATA KEEPVAR ◊ KEEP∆NAMES ∆KEEP
[20]    RTPR:∆PARS[4;]←∆PARS[3;]+∆PARS[5;]×¯1+?NNUM
[21]     →(⍎⍕1=∨/∧/∆PARS=(⍴∆PARS)⍴∆PARS[4;])/RTPR
[22]     ⍎(⍕∆PARS[1;]) '←∆PARS[4;]'
[23]     ∆PARS←∆PARS [1](1 NPARS)⍴∆PARS[4;]
[24]     ∆IT+←1
[25]     NEWTAB←0×OLDTAB
[26]     ⎕RL←16807
[27]     ∆I←0
[28]     '          EXPERIMENT  ' (⍕∆IT)
[29]    START:
[30]     →(NYEAR<∆I←∆I+1)/END
[31]     THIS∆YEAR+←1
[32]     (⍕⎕TS[4]) ':' (¯2↑'0' ⍕⎕TS[5]) '    Year        : ' ⍕THIS∆YEAR
[33]     YEAR
[34]     LAST∆YEAR+←1
[35]     NEWTAB[∆I;]←⍎⍕∆VARS[1;]
[36]     →DOIF SDEV≤⊃⌈/|NEWTAB[∆I;]-OLDTAB[∆I;] ◊ ADDRESULTB ∆IT ◊ →STR
[37]     →START
[38]    END:
[39]     →DOIF DISTANCE[1]<MIND ◊ MIND←DISTANCE[1] ◊ BESTTAB←NEWTAB
[40]     (⍕⎕TS[4]) ':' (¯2↑'0' ⍕⎕TS[5]) '     Distance : ' ⍕DISTANCE[1]
[41]     '      Min distance  : ' ⍕MIND
[42]     ADDRESULT ∆IT
[43]     →(∆IT<MITER)/STR
[44]     'DELETE.CAL'⎕FERASE 1
```

```
******************************************************************************

STARTΔS

[0]      STARTΔS;ΔVARS;ΔPARS;ΔKEEP;OLDVAR;NPARS;NYEAR;NVARS;ΔI;ΔIT;NPARS;NYE

AR;NVARS;OLDTAB;NEWTAB;KEEPVAR

[1]      KEEPVAR←CLEARΔBLANK↓⎕NL 2

[2]      SAVEΔDATA KEEPVAR

[3]      ΔVARS←⍎⊃MENU[10;1] ◊ ΔPARS←⍎⊃MENU[16;1]

[4]      NPARS←¯1↑⍴ΔPARS

[5]      NYEAR←¯2+1↑⍴ΔVARS

[6]      NVARS←¯1↑⍴ΔVARS

[7]      OLDTAB←2 0↓ΔVARS

[8]      ΔIT←0

[9]     ⍝

[10]      RESULTS←(1 (2+NPARS+NVARS))⍴(⊂'EXP NO') ΔPARS[1;] (⊂'DISTANCE') ΔV

ARS[1;]

[11]      STARTΔENT1

[12]      ΔKEEP←(↓⎕NL 2) (↓⎕NL 3) (⊂'OLDDIST') ⊂'ΔKEEP'

[13]     STR:

[14]      →DOIF ΔIT≠0 ◊ READΔDATA KEEPVAR ◊ KEEPΔNAMES ΔKEEP

[15]      ⍎(⍕ΔPARS[1;]) '←ΔPARS[4;]'

[16]      ΔIT+←1

[17]      NEWTAB←0×OLDTAB

[18]      ΔI←0

[19]      '          EXPERIMENT  ' (⍕ΔIT)

[20]     START:

[21]      →(NYEAR<ΔI←ΔI+1)/END

[22]      THISΔYEAR+←1

[23]      (⍕⎕TS[4]) ':' (¯2↑'0' ⍕⎕TS[5]) '    Year       : ' ⍕THISΔYEAR

[24]      YEAR

[25]      LASTΔYEAR+←1

[26]      NEWTAB[ΔI;]←⍎⍕ΔVARS[1;]

[27]      →DOIF SDEV≤⊃⌈/|NEWTAB[ΔI;]-OLDTAB[ΔI;] ◊ ADDRESULTB ΔIT ◊ →STR

[28]      →START

[29]     END:
```

```
[30]     (⍕⎕TS[4]) ':' (¯2↑'0' ⍕⎕TS[5]) '     Distance : ' ⍕DISTANCE[1]
[31]     ADDRESULT ΔIT
[32]     →(ΔIT<MITER)/STR
[33]     'DELETE.CAL'⎕FERASE 1

***************************************************************************
STARTΔZ

[0]     STARTΔZ;RESULTSALL;BASECASE;TESTVALUES;ΔKEEP;OLDVAR;ΔI;ΔIT;NPARS;KE
EPVAR
[1]     KEEPVAR←CLEARΔBLANK↓⎕NL 2
[2]     SAVEΔDATA KEEPVAR
[3]     TESTVALUES←0 ¯0.01 ¯0.001 0.001 0.01×STEPRF
[4]     ΔIT←0
[5]    ⍝
[6]     BASECASE←RESULTS←(NYEAR NTVAR)⍴0
[7]     RESULTSALL←0⍴0
[8]     STARTΔENT1
[9]     ΔKEEP←(↓⎕NL 2) (↓⎕NL 3) (⊂'OLDDIST') ⊂'ΔKEEP'
[10]    STR:
[11]     →DOIF ΔIT≠0 ◊ READΔDATA KEEPVAR ◊ KEEPΔNAMES ΔKEEP
[12]     ΔIT+←1
[13]     ⍎CVAR '×←1+' ⍕TESTVALUES[ΔIT]
[14]     ⎕RL←16807
[15]     ΔI←0
[16]     '         EXPERIMENT  ' (⍕ΔIT)
[17]    START:
[18]     →(NYEAR<ΔI←ΔI+1)/END
[19]     THISΔYEAR+←1
[20]     (⍕⎕TS[4]) ':' (¯2↑'0' ⍕⎕TS[5]) '    Year      : ' ⍕THISΔYEAR
[21]     YEAR
[22]     LASTΔYEAR+←1
[23]     →DOIF ΔIT=1 ◊ BASECASE[ΔI;]←⍎⍕TVAR
[24]     →DOIF ΔIT>1 ◊ RESULTS[ΔI;]←⍎⍕TVAR
```

```
  →START
 END:
  RESULTSALL ←⊂RESULTS
  →(ΔIT<ρTESTVALUES)/STR
  PREPAREΔRESULTS
  'DELETE.CAL'⎕FERASE 1
```

****

VIA

```
 VIA ΔΔN;ΔΔR;ΔΔI;⎕IO;⎕TRAP;ΔΔD
⍝Invoke via external function editor on named fn or fns
 ⎕IO←1 ◊ ⎕TRAP←( 239)'C' '→ΔΔL4'
 ΔΔD← '∇'
 →(1≠≡ΔΔN← ΔΔN)ρΔΔL1
 ΔΔN← ⊂ΔΔN
ΔΔL1:ΔΔR←⎕NR¨ΔΔN
ΔΔL2:ΔΔR←1↓⊃ /(⊂⊂ΔΔD) ¨ΔΔR
 →(0≠⎕NC'via')ρΔΔL3
 'via'⎕SH'via'
 ΔΔL3:ΔΔR←via ΔΔR
  ΔΔR←¯1⌽ΔΔR (ΔΔD≠1ρ(ρΔΔR)⊃ΔΔR)/⊂ΔΔD
  ΔΔR←1↓¨(( ↑1ρ¨ΔΔR)=ΔΔD)⊂ΔΔR
  ΔΔI←ΔΔI/ιρΔΔI←' '≠ ↑1ρ¨0ρ¨⎕FX¨ΔΔR
  →(0=ρΔΔI)ρ0
  'Unable to fix function' ((1<ρΔΔI)/'s') ' ' (⍕ΔΔI) ' Type Q to qui
t'
  →('Q'=1↑⍞)ρ0
  ΔΔR←ΔΔR[ΔΔI]
  →ΔΔL2
 ΔΔL4:'Quit'
```

****

INTERACT

```
 INTERACT;RESULTS;EXITKEYS;INITWORKSPACE
⍝
⍝ This function prepares the calibration or policy simulations.
⍝ Written by E.Taymaz  Dec. 1990
⍝
 EXITKEYS←'F1' 'F2' 'F9'
 ⎕SM←MENU
DEV:RESULTS←EXITKEYS 19 ⎕SR 19 9 17 11 12 10 16 13 21 7 8
 →(RESULTS[1]=7)/STR
 →(RESULTS[1]=8)/XNX
  →(RESULTS[4]≡⊂'F9')/0
  →DEV
 STR:
  2 1 1ρ' '
  'MODEL VERSION    : ' ⍕⎕SM[21;1]
  'INITIAL YEAR     : ' ⍕⎕SM[9;1]
  'ENTRY STARTS IN  : ' ⍕⎕SM[11;1]
  'AVERAGE ENTRIES  : ' ⍕⎕SM[12;1]
  'MSTART NUMBER    : ' ⍕⎕SM[13;1]
  INITWORKSPACE←'R' (⍕⎕SM[9;1]) '.' ⍕⎕SM[17;1]
  MENU←⎕SM
 ⍝
  3 1 1ρ' '(⊂'*****   LOADING MOSES   *****')
  ⎕CY'MOSES.PC'
  3 1 1ρ' '(⊂'*****   LOADING DATABASE   *****')
  ⎕CY INITWORKSPACE
  3 1 1ρ' '(⊂'*****   UPDATING THE MODEL   *****')
  ⍎'VERSION' ⍕10×⎕SM[21;1]
  3 1 1ρ' '(⊂'*****   MODIFYING THE MODEL   *****')
  VARIANTS←''
  ⍎(0<MENU[12;1])/'VARIANTS←VARIANTS '' + Firm entry: '' ⍕MENU[12;1]
'' firm(s)'''
```

```
[31]     UPDATEMOSES MENU[13;1]
[32]     AMAXENT←MENU[12;1]
[33]    ⍝
[34]    ⍝
[35]     3 1 1⍴' '(⊂'*****   LOADING VARIABLE MATRIX   *****')
[36]     ((' '≠⊃MENU[10;1])/⊃MENU[10;1])⎕CY'DATA'
[37]     3 1 1⍴' '(⊂'*****   LOADING PARAMETER MATRIX   *****')
[38]     ((' '≠⊃MENU[16;1])/⊃MENU[16;1])⎕CY'DATA'
[39]     ⎕WSID←(~(⊃MENU[19;1])=' ')/⊃MENU[19;1]
[40]     →DOIF 1=∧/' '=⎕WSID ◊ ⎕WSID←'DELETE.ME'
[41]     GET∆METHOD
[42]     ⎕ARBOUT 7
[43]     ' ' ◊ 2 70⍴'*' ◊ ' ' ◊ ⎕ARBOUT 7
[44]     →0
[45]    XNX:
[46]     ⎕SM←0 13⍴0
[47]     25⍴' '
[48]     ⎕OFF
```

****

I. MOSES GRAPH FUNCTIONS

APL
AXES
BARCHART
CGINIT
CGISTART
CHART
DOIF
EPI
FRAME
GCD
GHELP
GRAPH
GRAPHS
GXMENU
HIST
HYPO
LEGEND
LNORM
MAXSTRLEN
MESSAGE
NICE
PAT
PAUSE
PDEFINE
PFUN2
PIE
PIECHART
PLOT
PLOTINIT
SALTERC
SETSCALE
SHOWFUN
SHOWREAL
STRLEN
TITLE
VIA
WAIT
WINC
WINPP

```
***************************************************************************
APL

[0]     APL
[1]    ⍝ Reloads APL character font into EGA / VGA
[2]    ⍝ This only needs to be done when running under xenix 2.2
[3]    ⍝ ⎕SH gives DOMAIN ERROR when the exit code of the expression to be
[4]    ⍝ executed is non-zero. Both uname and aplfont2 return non-zero code
s
[5]    ⍝ but the 'exit 0' avoids this problem
[6]    ⍝
[7]     →0ρ⍨'2.2'≢3↑⊃⎕SH'uname -r; exit 0'
[8]     ⎕SH'$DYALOG/fonts/aplfont2; exit 0'

***************************************************************************
AXES

[0]     AXES XLABPOS;SINK
[1]    ⍝ Sub function of HIST and GRAPH
[2]    ⍝ Draws axes  ticks  labels  titles  grid lines
[3]    ⍝ All dimensions are in physical coordinates
[4]    ⍝ Uses following semi-globals ...
[5]    ⍝  W       (window - x1 y1 x2 y2)
[6]    ⍝  TL      (tick length)
[7]    ⍝  GRID    (grid lines)
[8]    ⍝  ∆CX     (character width)        ∆CY    (character height)
[9]    ⍝  XLAB    (X-Axis labels)          YLAB   (y-Axis labels)
[10]    ⍝  XTITLE (X-Axis title)          YTITLE (Y-Axis title)
[11]    ⍝  XTICS  (No. ticks on X-Axis)   YTICS  (No. ticks on Y-Axis)
[12]    ⍝  XCOL   (X-Axis colour)         YCOL   (Y-Axis colour)
[13]    ⍝
[14]    ⍝ XLABPOS (position of X-axis labels ) ...
```

```
[15]    ⍝         0 = centred on tick-marks (GRAPH)
[16]    ⍝         1 = centred between tick-marks (HIST)
[17]    ⍝
[18]    ⍝ Defines following semi-globals ...
[19]    ⍝  XO     (X position of origin)   YO     (Minimum Y-value)
[20]    ⍝  X1     (Maximum X-value)        Y1     (Maximum Y-value)
[21]    ⍝  XGAP   (space between X-ticks)  YGAP   (space between Y-ticks)
[22]    ⍝ YORIGIN (Y position of origin)
[23]    ⍝
[24]    ⍝
[25]    ⍝                                 |
[26]    ⍝                                 |
[27]    ⍝                              ---|
[28]    ⍝ <ΔCY><ΔCX><ΔCX×¯1↑ρYLAB><TL><TL>|
[29]    ⍝                                 |
[30]    ⍝           +------------+        | (XO YORIGIN)
[31]    ⍝ +---+     |   YLAB     |     ---|--------------------------------
--
[32]    ⍝ | Y |     +------------+        |
[33]    ⍝ | T |                           |
[34]    ⍝ | I |     +------------+        |
[35]    ⍝ | T |     |   YLAB     |     ---|
[36]    ⍝ | L |     +------------+        |
[37]    ⍝ | E |                           |
[38]    ⍝ +---+     +------------+        | (XO YO)
[39]    ⍝           |   YLAB     |    ^---|--------------------------------
---
[40]    ⍝           +------------+    TL             |          |          |
[41]    ⍝                             v              |          |          |
[42]    ⍝                             ^               +------+   +------+
[43]    ⍝                             ΔCY             | XLAB |   | XLAB |
[44]    ⍝                             v               +------+   +------+
[45]    ⍝                             ^
```

```
[46]    ⍝                             ∆CY×0.15
[47]    ⍝                             v
[48]    ⍝                             ^            +----------------------
-+
[49]    ⍝                             ∆CY          |         XTITLE
 |
[50]    ⍝                             v            +----------------------
-+
[51]    ⍝
[52]     X0←⊃W[1]+∆CY+∆CX+(TL×2)+MAXSTRLEN YLAB                ⍝ X-origin
[53]    ⍝ (Note that YTITLE takes up an x-distance of ∆CY because of rotati
on)
[54]     YORIGIN←Y0←W[2]+(2.15×∆CY)+TL                             ⍝ Min Y
[55]     X1←W[3] ◊ Y1←W[4]-∆CY
[56]     →DOIF~XLABPOS ◊ X1←X1-0.5×MAXSTRLEN XLAB
[57]     XGAP←(X1-X0)÷XTICS-1
[58]     YGAP←(Y1-Y0)÷YTICS-1
[59]     →DOIF(0 0≠GRID[;2]) ◊ (⊂GRID[;2])VPLINE(⊂X0 X1) [1.5]¨Y0+YGAP×¯1+⍳
YTICS
[60]     X←X0+XGAP× ⍉(3 XTICS)⍴¯1+⍳XTICS
[61]     Y←Y0-(3×XTICS)⍴TL×0 1 0
[62]     1 XCOL VPLINE X [1.5]Y
[63]    ⍝ Calculate position of X-axis labels
[64]     X←X0+XGAP×0 ⍳XTICS-1
[65]     Y←Y0-∆CY+TL×~XLABPOS
[66]     →DOIF XLABPOS ◊ X←(0.5×XGAP)+¯1↓X
[67]     →(YMIN≥0)/L1
[68]     YORIGIN←Y0+(Y1-Y0)×(-YMIN)÷YMAX-YMIN     ⍝ Draw X-axis if
[69]     VPLINE(X0 X1)YORIGIN                     ⍝ YORIGIN≠Y0
[70]    L1:
[71]    ⍝ (Note that X-labels are centred between tick marks)
[72]     FONT XCOL 0 1 0 VGTEXT X Y XLAB
[73]     X←X0+(X1-X0)÷2 ◊ Y←W[2]
```

```
[74]      VGTEXT X Y XTITLE

[75]      →DOIF(0 0≠GRID[;1]) ◊ (⊂GRID[;1])VPLINE(X0+XGAP×¯1+⍳XTICS) [1.5]¨⊂
Y0 Y1

[76]      X←2↑X0-(3×YTICS)⍴TL×0 0 1

[77]      Y←Y0+0 0 ¯1↓ YGAP×⌽(3 YTICS-1)⍴⍳YTICS-1

[78]      1 YCOL VPLINE X [1.5]Y

[79]      X←X0-(2×TL) ◊ Y←Y0+YGAP×¯1+⍳YTICS

[80]      FONT YCOL 0 2 1 VGTEXT X Y YLAB

[81]      X←W[1]+∆CY ◊ Y←0.5×Y0+Y1

[82]      FONT YCOL 900 1 0 VGTEXT X Y YTITLE

[83]      SINK←VSTALN 0 0             ⍝ Reset alignment

[84]      VSTROT 0                    ⍝ Reset text direction to l-right
```

```
***************************************************************************

BARCHART

[0]      BARCHART;PLOT;BCOL;BOXED;TEXT;XLAB;DEVICE;DEVINFO;DINT;FORMS;CCOL;C
TITLE;YCOL;YTITLE;XEXP;XCOL;XTITLE;YEXP;DCOL;DSTYLE;HGR;VGR;STACKED;GRID;XV
AL;YVAL;WIN;X;KEY;MASK;WIN;FONT;SIZE;MSG

[1]      PLOT←'R'

[2]      CCOL←YCOL←XCOL←5 ◊ CTITLE←XTITLE←YTITLE←XEXP←YEXP←' '

[3]      HGR←VGR←STACKED←0 ◊ DCOL←6⍴1 ◊ DSTYLE←0 ⍳5

[4]      PSTART

[5]      PENTER 1

[6]      X←PKEY 0 1 5 10

[7]      MSG←'Press ' (⊃X) ' to draw barchart  ' (2⊃X) ' for help  '

[8]      MSG ←(3⊃X) ' to print barchart ' (4⊃X) ' to quit.'

[9]      PDEFINE ∆PLOTSCREENB

[10]      STACKED HGR VGR CTITLE XTITLE YTITLE XEXP YEXP CCOL XCOL YCOL DCOL
 DSTYLE PPUT'STACKED' 'HGR' 'VGR' 'CTITLE' 'XTITLE' 'YTITLE' 'XEXP' 'YEXP'
'CCOL' 'XCOL' 'YCOL' 'DCOL' 'DSTYLE'

[11]     Reset:MSG PPUT'MSG'

[12]     Again:→(0 1 5 10=KEY←PREAD)/Doit Help Doit Exit ◊ PPUT 2 ◊ →Again
```

```
[13]    Help:GHELP 2 ◊ →Again
[14]    Doit:
[15]     →DOIF 5=KEY ◊ 'PLOTTER OR PRINTER (P/R)?' ◊ PLOT←⍞
[16]     YEXP←PGET'YEXP'
[17]     →Again×ι~∨/MASK←(∨/YEXP≠' ')∧~∨/YEXP∊'←→'
[18]     CTITLE XTITLE YTITLE XEXP←1 PGET'CTITLE' 'XTITLE' 'YTITLE' 'XEXP'
[19]     STACKED HGR VGR CCOL XCOL YCOL DCOL DSTYLE←PGETN'STACKED' 'HGR' 'V
GR' 'CCOL' 'XCOL' 'YCOL' 'DCOL' 'DSTYLE'
[20]     YEXP DCOL DSTYLE/⍨←⊂MASK
[21]     STACKED HGR VGR CCOL XCOL YCOLρ⍨←⊂⍬⍝  Ensure these are scalars
[22]     HGR←2⊤HGR ◊ VGR←2⊤VGR ◊ STACKED←2⊤STACKED
[23]     →(∧/DSTYLE∊0 ι9)/Ok1
[24]     MESSAGE'Valid bar styles are in the range 0-9'
[25]     →Again
[26]    Ok1:⎕EX¨'X' 'XVAL'
[27]     →L1×ι∧/' '=XEXP
[28]     X←XVAL←⍎XEXP
[29]    L1:
[30]     YVAL←⍉↑⍎¨↓YEXP
[31]     YVAL←(2↑(ρYVAL) 1)ρYVAL
[32]     ⍎(2≠⎕NC'XVAL')/'XVAL←(1↑ρYVAL)ρ'' '''
[33]     →(⊃∧/(1↑ρYVAL)=ρ¨⍎¨↓YEXP)/L2
[34]     MESSAGE'Variables must be in same size'
[35]     →Again
[36]    L2:
[37]     XLAB←⍕¨XVAL
[38]     →((ρXLAB)=1↑ρYVAL)/L2
[39]     MESSAGE'Number of labels should be equal to the number of clusters
'
[40]     →Again
[41]     CTITLE XTITLE YTITLE←dbr° ¨CTITLE XTITLE YTITLE
[42]    L2:
[43]     TEXT←↓YEXP ◊ TEXT←5↑¨TEXT
```

```
[44]     GRID←2 2⍴VGR HGR(VGR×XCOL)(HGR×YCOL)
[45]     WIN←WINC KEY PLOT'C' ◊ DINT←(¯1↑⍴YVAL)⍴2
[46]     FORMS←0 1 2 4 5 20 22 11 13 14
[47]     DSTYLE←FORMS[1+DSTYLE]
[48]     BCOL←0
[49]     FONT SIZE BOXED←1 349 0
[50]     CGISTART
[51]     CGINIT KEY PLOT
[52]     →DOIF 0=DEVINFO[46] ◊ VSBCOL BCOL
[53]     →DOIF 0=KEY ◊ VCLRWK
[54]     →DOIF 0=DEVINFO[46] ◊ FONT SIZE BOXED←1 586 0 ◊ →LL1
[55]     FONT SIZE BOXED←1 349 0
[56]    LL1:
[57]     HIST WIN STACKED BOXED YVAL DINT DCOL DSTYLE GRID XLAB XCOL YCOL X
TITLE YTITLE FONT SIZE
[58]     WIN←WINC KEY PLOT'L'
[59]     →DOIF 0=DEVINFO[46] ◊ FONT SIZE BOXED←1 586 0 ◊ →LL2
[60]     FONT SIZE BOXED←1 349 0
[61]    LL2:
[62]     LEGEND WIN FONT SIZE DINT DCOL DSTYLE TEXT XCOL BOXED BCOL
[63]     WIN←WINC KEY PLOT'T'
[64]     →DOIF 0=DEVINFO[46] ◊ FONT SIZE BOXED←2 911 1 ◊ →LL3
[65]     FONT SIZE BOXED←2 750 1
[66]    LL3:
[67]     TITLE WIN CTITLE CCOL FONT SIZE BOXED
[68]     FRAME 5 KEY PLOT
[69]     →DOIF 0=KEY ◊ PAUSE
[70]     RESET ◊ PREDRAW
[71]     →Reset
[72]    Exit:PRESET
```

****

```
CGINIT
[0]     CGINIT PRS;K;P
[1]     K P←PRS
[2]     →DOIF K=0 ◊ DEVICE DEVINFO←VOPNWK'DISPLAY' ◊ →0
[3]     →DOIF'P'=1↑P ◊ DEVICE DEVINFO←VOPNWK'PLOTTER' ◊ →0
[4]     DEVICE DEVINFO←VOPNWK'RINTER'

***************************************************************************
CGISTART

[0]     CGISTART
[1]     →(3=⎕NC'VARC')/0
[2]     'cgi'⎕SH'cgi'

***************************************************************************
CHART

[0]     CHART;MCOL;PLOT;CTYPE;BCOL;BOXED;TEXT;DEVICE;DEVINFO;GRID;HGR;VGR;C
TITLE;XTITLE;YTITLE;XEXP;YEXP;CCOL;XCOL;YCOL;DCOL;DSTYLE;XVAL;YVAL;WIN;X;KE
Y;MASK;WIN;PLYM;FONT;SIZE;MSG
[1]     PLOT←'R'
[2]     CTITLE←XTITLE←YTITLE←' ' ◊ CCOL←XCOL←YCOL←5 ◊ DCOL←6⍴1 ◊ DSTYLE←⍳6
[3]     XEXP←YEXP←' ' ◊ HGR←VGR←CTYPE←0 ◊ PLYM←6⍴0
[4]     PSTART
[5]     PENTER 1
[6]     X←PKEY 0 1 5 10
[7]     MSG←'Press ' (⊃X) ' to draw graph  ' (2⊃X) ' for help  '
[8]     MSG ←(3⊃X) ' to print graph  ' (4⊃X) ' to quit.'
[9]     FONT SIZE←1 586
[10]     PDEFINE ∆PLOTSCREEN
[11]     PLYM CTYPE HGR VGR CTITLE XTITLE YTITLE XEXP YEXP CCOL XCOL YCOL D
COL DSTYLE PPUT'PLYM' 'CTYPE' 'HGR' 'VGR' 'CTITLE' 'XTITLE' 'YTITLE' 'XEXP'
 'YEXP' 'CCOL' 'XCOL' 'YCOL' 'DCOL' 'DSTYLE'
[12]    Reset:MSG PPUT'MSG'
[13]    Again:→(0 1 5 10=KEY←PREAD)/Doit Help Doit Exit ◊ PPUT 2 ◊ →Again
```

```
[14]    Help:GHELP 1 ◊ →Again
[15]    Doit:
[16]     →DOIF 5=KEY ◊ 'PLOTTER OR PRINTER (P/R)?' ◊ PLOT←⍞
[17]     YEXP←PGET'YEXP'
[18]     →Again×⍳~∨/MASK←(∨/YEXP≠' ')∧~∨/YEXP∊'←→'
[19]     CTITLE XTITLE YTITLE XEXP←1 PGET'CTITLE' 'XTITLE' 'YTITLE' 'XEXP'
[20]     PLYM CTYPE HGR VGR CCOL XCOL YCOL DCOL DSTYLE←PGETN'PLYM' 'CTYPE'
'HGR' 'VGR' 'CCOL' 'XCOL' 'YCOL' 'DCOL' 'DSTYLE'
[21]     PLYM YEXP DCOL DSTYLE⌿¨←⊂MASK
[22]     CTYPE HGR VGR CCOL XCOL YCOLρ¨←⊂⍬⍝  Ensure these are scalars
[23]     HGR←2⊤HGR ◊ VGR←2⊤VGR
[24]     GRID←2 2ρVGR HGR(VGR×XCOL)(HGR×YCOL)
[25]     →(∧/DSTYLE∊0 ⍳7)/Ok1
[26]     MESSAGE'Valid line styles are in the range 1-7'
[27]     →Again
[28]    Ok1:→(∧/PLYM∊0 ⍳7)/OK2
[29]     MESSAGE'Valid marker styles are in the range 1-7'
[30]     →Again
[31]    OK2:
[32]     MCOL←DCOL ◊ ((PLYM=0)/MCOL)←0 ◊ ((DSTYLE=0)/DCOL)←0
[33]     ⎕EX¨'X' 'XVAL'
[34]     →L1×⍳∧/' '=XEXP
[35]     X←XVAL←⍎XEXP
[36]    L1:YVAL←⍉↑⍎¨↓YEXP
[37]     YVAL←(2↑(ρYVAL) 1)ρYVAL
[38]     ⍎(2≠⎕NC'XVAL')/'XVAL←⍳1↑ρYVAL'
[39]     →(⊃∧/(ρXVAL)=ρ¨⍎¨↓YEXP)/L2
[40]     MESSAGE'Variables must be in same size'
[41]     →Again
[42]    L2:
[43]     →(0=∧/(↓YVAL)[1]≡¨↓YVAL)/L8
[44]     MESSAGE'Variable(s) have same values' ◊ →Again
[45]    L8:
```

```
[46]     →DOIF 1≠≡XVAL ◊ MESSAGE'Dimension of X variable is not equal to on
e' ◊ →Again
[47]     →DOIF 1≠≡YVAL ◊ MESSAGE'Dimension of Y variables are not equal to
one' ◊ →Again
[48]     CTITLE XTITLE YTITLE←dbr° ¨CTITLE XTITLE YTITLE
[49]     TEXT←↓YEXP ◊ TEXT←5↑¨TEXT
[50]     WIN←WINC KEY PLOT'C'
[51]     CGISTART
[52]     CGINIT KEY PLOT
[53]     BCOL←0 ◊ FONT SIZE BOXED←1 586 0
[54]     →DOIF 0=DEVINFO[46] ◊ FONT SIZE BOXED←1 586 0 ◊ →LL2
[55]     FONT SIZE BOXED←1 349 0
[56]    LL2:
[57]     GRAPH MCOL WIN CCOL XVAL YVAL DSTYLE DCOL PLYM GRID XCOL YCOL XTIT
LE YTITLE FONT SIZE
[58]     WIN←WINC KEY PLOT'L'
[59]     →DOIF 0≠CTYPE ◊ LEGEND_L MCOL WIN FONT SIZE DSTYLE DCOL PLYM TEXT
XCOL BOXED BCOL
[60]     WIN←WINC KEY PLOT'T'
[61]     →DOIF 0=DEVINFO[46] ◊ FONT SIZE BOXED←2 911 1 ◊ →LL9
[62]     FONT SIZE BOXED←2 698 1
[63]    LL9:
[64]     TITLE WIN CTITLE CCOL FONT SIZE BOXED
[65]     →DOIF 0=KEY ◊ PAUSE
[66]     RESET ◊ PREDRAW
[67]     →Reset
[68]    Exit:PRESET
```

****

DOIF

```
[0]     R←DOIF TEST
[1]    ⍝Return next line number if TEST false
```

```
[2]     R←(~TEST)/1↓1↓2↑⎕LC

****************************************************************************

EPI

[0]      RΔFIXED EPI RΔROLL;THETA;G;X;Y
[1]     ⍝ Draws an Epicycloid which is a curve described by a point
[2]     ⍝ on the circumference of a circle as the circle rolls
[3]     ⍝ without slipping  on the outside of a fixed circle.
[4]     ⍝ RΔFIXED is the radius of the fixed circle
[5]     ⍝ RΔROLL  is the radius of the rolling circle
[6]     ⍝
[7]      THETA←2×○40÷⍨⍳40×RΔROLL÷RΔFIXED GCD RΔROLL
[8]      G←(RΔFIXED+RΔROLL)÷RΔROLL
[9]      X←((RΔFIXED+RΔROLL)×2○THETA)-RΔROLL×2○G×THETA
[10]      Y←((RΔFIXED+RΔROLL)×1○THETA)-RΔROLL×1○G×THETA
[11]      X Y×←(⌊/DEVINFO[52 53])÷500                        ⍝ Scale to N
DC
[12]      X Y+←0.5×DEVINFO[52 53]                            ⍝ Centralise
[13]      VPLINE X [1.5]Y

****************************************************************************

FRAME

[0]      FRAME PARS;W;PAR;K;P
[1]     ⍝ Draw frame around window
[2]      PAR K P←PARS
[3]      W←0 0 ¯1+DEVINFO[52 53]
[4]      →(0=K)/JM
[5]      →DOIF'R'=1↑P ◊ W[1]←0.1×W[3] ◊ W[2]←0.2×W[4]
[6]      W[3 4]←0.95×W[3 4]
[7]     JM:(¯2↑ PAR)VPLINE(W[1 1 3 3 1])(W[2 4 4 2 2])

****************************************************************************
```

```
GCD

[0]     R←A GCD R
[1]     →0≠⊃A R←(A|R)A

****************************************************************************

GHELP

[0]     GHELP ΔX;X;Y
[1]     X←⊃PDEFN 1 6ρ1 1 24 80 0 0
[2]     HELPMESSAGE[ΔX]PPUT 1
[3]     Y←PREAD
[4]     PDEFN-X

****************************************************************************

GRAPH

[0]     GRAPH PARS;YVALL;MCOL;W;XMIN;XMAX;XINC;XTICS;XLAB;YMIN;YMAX;YINC;YT
ICS;YLAB;X;Y;X0;Y0;XGAP;YGAP;X1;Y1;I;XSCALE;YSCALE;ΔCX;ΔCY;WIN;BOXED;XVAL;Y
VAL;DCOL;DSTYLE;XCOL;YCOL;XTITLE;YTITLE;FONT;TL;PLYM;GRID;XINC;YINC;SIZE;GS
QDSP
[1]     MCOL WIN BOXED XVAL YVAL DSTYLE DCOL PLYM GRID XCOL YCOL XTITLE YTI
TLE FONT SIZE←PARS
[2]     →DOIF~36⊃DEVINFO ◊ DCOL XCOL YCOL BOXED≠←0 ⍝ If mono  set cols to b
lack
[3]     VSTFNT FONT ◊ ΔCX ΔCY←⌈/2 2ρVSTHGT SIZE
[4]     W←(0.01× WIN)× 2 2ρ¯1+DEVINFO[52 53]
[5]     →DOIF 0≠BOXED ◊ 1 BOXED VPLINE RECTANGLE 2 2ρW
[6]     W+←1 1 ¯1 ¯1×4ρ0.5×ΔCX ΔCY
[7]     TL←⌈ΔCX÷2 ⍝ Tick length
[8]     XINC←(-/W[3 1])÷10×ΔCX
[9]     XMIN XMAX XINC XTICS XLAB←XINC SETSCALE XVAL
[10]     YINC←(-/W[4 2])÷5×ΔCY
[11]     YMIN YMAX YINC YTICS YLAB←YINC SETSCALE YVAL
[12]     AXES 0
```

```
[13]      XSCALE←(X1-X0)÷XMAX-XMIN

[14]      YSCALE←(Y1-Y0)÷YMAX-YMIN

[15]      X←X0+XSCALE×XVAL-XMIN

[16]      YVAL←Y0+((ρYVAL)ρYSCALE)×YVAL-(ρYVAL)ρYMIN

[17]      →(∧/0=DSTYLE)/JMP

[18]      DCOL←(DSTYLE≠0)/DCOL

[19]      YVALL←(DSTYLE≠0)/YVAL

[20]      DSTYLE←(DSTYLE≠0)/DSTYLE

[21]      (DSTYLE ¨DCOL)VPLINE Y←(⊂⊂X) ∘⊂¨↓[1]YVALL

[22]     JMP:→(∧/0=PLYM)/EXIT

[23]      MCOL←(PLYM≠0)/MCOL

[24]      YVAL←(PLYM≠0)/YVAL

[25]      PLYM←(PLYM≠0)/PLYM

[26]     ⍝ XVAL←(~∨/YVAL=Y0)⌿XVAL

[27]     ⍝ YVAL←(~∨/YVAL=Y0)⌿YVAL

[28]      (PLYM ¨MCOL)VPMARK Y←(⊂⊂X) ∘⊂¨↓[1]YVAL

[29]     EXIT:


***************************************************************************

GRAPHS

[0]      GRAPHS PARS;ΔNWEY;ΔNWEX;ΔSYNY;ΔSYNX;YVALL;MCOL;W;XMIN;XMAX;XINC;XTI
CS;XLAB;YMIN;YMAX;YINC;YTICS;YLAB;X;Y;X0;Y0;XGAP;YGAP;X1;Y1;I;XSCALE;YSCALE
;ΔCX;ΔCY;WIN;BOXED;XVAL;YVAL;DCOL;DSTYLE;XCOL;YCOL;XTITLE;YTITLE;FONT;TL;PL
YM;GRID;XINC;YINC;SIZE;GSQDSP

[1]      MCOL WIN BOXED XVAL YVAL DSTYLE DCOL PLYM GRID XCOL YCOL XTITLE YTI
TLE FONT SIZE←PARS

[2]      →DOIF~36⊃DEVINFQ ◊ DCOL XCOL YCOL BOXED≠←0 ⍝ If mono  set cols to b
lack

[3]      VSTFNT FONT ◊ ΔCX ΔCY←⌈⌿2 2ρVSTHGT SIZE

[4]      W←(0.01× WIN)× 2 2ρ¯1+DEVINFQ[52 53]

[5]      →DOIF 0≠BOXED ◊ 1 BOXED VPLINE RECTANGLE 2 2ρW

[6]      W+←1 1 ¯1 ¯1×4ρ0.5×ΔCX ΔCY
```

```
[7]     TL←⌈∆CX÷2 ⍝ Tick length
[8]     XINC←(-/W[3 1])÷10×∆CX
[9]     XMIN XMAX XINC XTICS XLAB←XINC SETSCALE XVAL
[10]     YINC←(-/W[4 2])÷5×∆CY
[11]     YMIN YMAX YINC YTICS YLAB←YINC SETSCALE YVAL
[12]     AXES 0
[13]     XSCALE←(X1-XO)÷XMAX-XMIN
[14]     YSCALE←(Y1-YO)÷YMAX-YMIN
[15]     X←XO+XSCALE×XVAL-XMIN
[16]     YVAL←YO+((⍴YVAL)⍴YSCALE)×YVAL-(⍴YVAL)⍴YMIN
[17]     →DOIF 2=⎕NC'∆SYN' ◊ ∆SYNY←(¯1↑YVAL) YO+((⍴∆SYN)⍴YSCALE)×∆SYN-(⍴∆SY
N)⍴YMIN ◊ ∆SYNX←(1↑X) X ◊ 1 1 1 VFAREA ∆SYNX ∆SYNY
[18]     →DOIF 2=⎕NC'∆NWE' ◊ ∆NWEY←(¯1↑YVAL) YO+((⍴∆NWE)⍴YSCALE)×∆NWE-(⍴∆NW
E)⍴YMIN ◊ ∆NWEX←(1↑X) X ◊ 2 1 5 VFAREA ∆NWEX ∆NWEY
[19]     →(∧/0=DSTYLE)/JMP
[20]     DCOL←(DSTYLE≠0)/DCOL
[21]     YVALL←(DSTYLE≠0)/YVAL
[22]     DSTYLE←(DSTYLE≠0)/DSTYLE
[23]     (DSTYLE ¨DCOL)VPLINE Y←(⊂⊂X) ∘⊂¨↓[1]YVALL
[24]    JMP:→(∧/0=PLYM)/EXIT
[25]     MCOL←(PLYM≠0)/MCOL
[26]     YVAL←(PLYM≠0)/YVAL
[27]     PLYM←(PLYM≠0)/PLYM
[28]    ⍝ XVAL←(~∨/YVAL=YO)⌿XVAL
[29]    ⍝ YVAL←(~∨/YVAL=YO)⌿YVAL
[30]     (PLYM ¨MCOL)VPMARK Y←(⊂⊂X) ∘⊂¨↓[1]YVAL
[31]    EXIT:
```

****

GXMENU

```
[0]     CHOICE←GXMENU PARS;LOC;TEXT;FONT;TSIZE;TCOL;LCOL;∆CX;∆CY;X;Y;RECT;C
URSOR;POS;L;TX;XO;YO;CT;TY;EFLAG;NCELLS;BOX
[1]    ⍝
```

```
[2]     ⍝ Displays a pop-up menu  returns user's selection
[3]     ⍝ Argument is a nested vector as follows :
[4]     ⍝  1. LOC    : (x y) coordinates for bottom-left corner of menu box
[5]     ⍝  2. TEXT   : Vector of text vectors (the choices)
[6]     ⍝  3. FONT   : Text font
[7]     ⍝  4. TSIZE  : Text font size
[8]     ⍝  5. BCOL   : Background colour for menu boxes
[9]     ⍝  6. TCOL   : Text colour
[10]     ⍝  7. LCOL   : Line colour for borders around menu boxes
[11]     ⍝  8. EFLAG  : Flag to indicate if/if not menu is to be erased on e
xit
[12]     ⍝  9. CHOICE : Initial (default) choice
[13]     ⍝
[14]     LOC TEXT FONT TSIZE BCOL TCOL LCOL EFLAG CHOICE←PARS
[15]     VGTATT FONT TCOL 0
[16]     ΔCX ΔCY←⌈⌿2 2ρVSTHGT TSIZE
[17]     NCELLS←⌽ρTEXT←(2↑(ρTEXT) 1)ρTEXT ⍝ Default to vertical menu
[18]     L←ΔCX×1+⊃⌈/ρ¨TEXT←⌽ TEXT
[19]     RECT←(⌊LOC) [0.5]⌈LOC+NCELLS×L ΔCY×2
[20]     X←LOC[1]+L×0 ⍳⊃NCELLS
[21]     Y←LOC[2]+2×ΔCY×0 ⍳2⊃NCELLS
[22]     TX←LOC[1]+0.5×L-ΔCX×⊃ /ρ¨TEXT
[23]     TX+←(×/NCELLS)ρ⌽L×0 ¯1↓⍳⊃NCELLS
[24]     TY←(⊃NCELLS)/¯1↓Y+0.5×ΔCY
[25]     TY+←EXT[4]
[26]     VSDBIT BACK                  ⍝ Use same bitmap  but above visible
 part
[27]     VCPBIT SCREEN RECT(EXT[2 4]+RECT[1;])
[28]     VSDBIT FORE                  ⍝ Use same bitmap  but above visible
 part
[29]     1 BCOL VFAREA RECTANGLE RECT+2 2ρEXT[2 4]
[30]     VSBCOL BCOL
[31]     VGTEXT TX TY TEXT
```

```
[32]     VSBCOL 0

[33]     VGLATT 1 1 ⍝ LCOL

[34]     VPLINE((⊂⊂X) ¨(Y+EXT[4])) (⊂⊂(EXT[4]+Y[1(ρY)])) ⍨¨X

[35]     VSDBIT SCREEN

[36]     VCPBIT FORE(RECT+2 2ρEXT[2 4])(RECT[1;]) ⍝ Copy menu to screen

[37]     POS←(CHOICE⊃⌽TX) (CHOICE⊃⌽TY-EXT[4]) ⍝ Position cursor in chosen b

ox

[38]    LOOP:POS←⊃VRQLOC DEVICE POS 0 0

[39]     →(∨/(POS≤RECT[1;])∨POS≥RECT[2;])/LOOP

[40]     CHOICE←(XO←''ρ¯1↑(POS[1]≥X)/⍳ρX)+(⊃NCELLS)×YO←¯1+''ρ¯1↑(POS[2]≤⌽Y)

/⍳ρY

[41]     1 TCOL VBAR BOX←X[XO+0 1] [1.5]Y[(ρY)-YO+1 0]

[42]     0 LCOL VBAR BOX

[43]     VSBCOL TCOL

[44]     CT←1+(ρTEXT)-CHOICE

[45]     (⊃VQTATT)BCOL VGTEXT TX[CT] (TY-EXT[4])[CT] TEXT[CT]

[46]     VSBCOL 0

[47]     →EFLAG/L2

[48]     CHOICE←CHOICE(BACK RECT) ◊ →Exit

[49]    L2:VCPBIT BACK(RECT+2 2ρEXT[2 4])(RECT[1;]) ⍝ Erase menu if needed

[50]    Exit:


*************************************************************************

HIST

[0]     HIST PARS;W;XMIN;XMAX;XINC;XTICS;XLAB;YMIN;YMAX;YINC;YTICS;YLAB;X;Y

;XO;YO;X1;Y1;YGAP;I;XSCALE;YSCALE;ΔCX;ΔCY;XGAP;N;BAR;WIN;XLAB;XTITLE;YTITLE

;DATA;XCOL;YCOL;BOXED;FONT;TL;STACKED;BBAR;XY;DSTYLE;GRID;SIZE;YORIGIN;DCOL

;DINT

[1]   ⍝

[2]     WIN STACKED BOXED DATA DINT DCOL DSTYLE GRID XLAB XCOL YCOL XTITLE

YTITLE FONT SIZE←PARS

[3]     →DOIF 1=DEVINFO[46] ◊ DSTYLE←1+6|DSTYLE ⍝ plotters only have 6 patt
```

```
erns

[4]     →DOIF~36⊃DEVINFO ◊ DCOL XCOL YCOL BOXED≠←0 ⍝ If mono  set cols to b

lack

[5]     VSTFNT FONT ◊ ∆CX ∆CY←⌈/2 2⍴VSTHGT SIZE

[6]     W←(0.01× WIN)× 2 2⍴¯1+DEVINFO[52 53]

[7]     →DOIF 0≠BOXED ◊ 1 BOXED VPLINE RECTANGLE 2 2⍴W

[8]     W+←1 1 ¯1 ¯1×4⍴0.5×∆CX ∆CY

[9]     TL←⌈∆CX÷2 ⍝ Tick length

[10]     N←1↑⍴DATA

[11]     →DOIF''≡XLAB ◊ XLAB←⍕¨⍳N

[12]     XMIN XMAX XINC XTICS←0 N 1 1+N

[13]     YMIN YMAX YINC YTICS YLAB←((-/W[4 2])÷5×∆CY)SETSCALE DATA+\IF STAC

KED

[14]     AXES 1

[15]     YSCALE←(Y1-Y0)÷YMAX-YMIN

[16]     DATA←((⍴DATA)⍴YSCALE)×DATA-(⍴DATA)⍴YMIN

[17]     Y←YORIGIN ¨(2⍴¨DATA+Y0) ¨⊂2⍴YORIGIN

[18]     →DOIF STACKED ◊ Y←Y++\0 0 ¯1↓DATA

[19]     BBAR←BAR←XGAP÷2

[20]     →DOIF~STACKED ◊ BAR←BAR÷¯1↑⍴DATA

[21]     X←X0+(0.5×XGAP-BBAR)+XGAP×¯1+⍳N

[22]     →DOIF~STACKED ◊ X←X∘.+BAR×0 ⍳¯1+¯1↑⍴DATA

[23]     X←↓X∘.+0 0 1 1 0×BAR

[24]     →DOIF STACKED ◊ X←(¯1↑⍴Y)/((⍴X) 1)⍴X

[25]     XY←⊃ /↓[1]X [1.5]¨Y

[26]     (N/↓⌽↑DINT DCOL DSTYLE)VFAREA XY

[27]     (N/0 ¨DCOL)VFAREA XY    ⍝ Draw outlines

***************************************************************************

HYPO

[0]     R∆FIXED HYPO R∆ROLL;THETA;H;X;Y

[1]     ⍝ Draws a Hypocycloid which is a curve described by a point

[2]     ⍝ on the circumference of a circle as the circle rolls
```

```
[3]     ⍝ without slipping  on the inside of a fixed circle.

[4]     ⍝ RΔFIXED is the radius of the fixed circle

[5]     ⍝ RΔROLL  is the radius of the rolling circle

[6]     ⍝

[7]      THETA←2×○40÷⍨⍳40×RΔROLL÷RΔFIXED GCD RΔROLL

[8]      H←(RΔFIXED-RΔROLL)÷RΔROLL

[9]      X←((RΔFIXED-RΔROLL)×2○THETA)+RΔROLL×2○H×THETA

[10]      Y←((RΔFIXED-RΔROLL)×1○THETA)-RΔROLL×1○H×THETA

[11]      X Y×←(⌊/DEVINFO[52 53])÷500                      ⍝ Scale to N

DC

[12]      X Y+←0.5×DEVINFO[52 53]                         ⍝ Centralise

[13]      VPLINE X [1.5]Y


****************************************************************************

LEGEND

[0]      LEGEND PARS;N;L;H;COLS;ROWS;SIZE;START;X;DCOL;DINT;DSTYLE;BOX;TPOS;

LT;ΔCX;ΔCY;FONT;BOXED;RECT;LBOX;TCOL;SIZE;XO;YO;XTEXT;YTEXT;XBOX;YBOX

[1]     ⍝ Displays a legend into given rectangle RECT

[2]      RECT FONT SIZE DINT DCOL DSTYLE TEXT TCOL BOXED BCOL←PARS

[3]      →DOIF 1=DEVINFO[46] ⋄ DSTYLE←1+6|DSTYLE ⍝ plotters only have 6 patt

erns

[4]      →DOIF~36⊃DEVINFO ⋄ TCOL BOXED≠←0 ⍝ If mono  set cols to black

[5]      RECT⍴⍨←2 2

[6]      RECT×←0.01×2 2⍴¯1+DEVINFO[52 53]

[7]      VSTFNT FONT ⋄ ΔCX ΔCY←⌈⌿2 2⍴VSTHGT SIZE

[8]      LBOX←4⌊LT←⍴⊃TEXT ⍝ LBOX = length of legend box  LT = length of text

[9]      L←(ΔCX×3+LBOX)+MAXSTRLEN TEXT⍝ Length of each legend

[10]      N←⍴TEXT ⍝ Number of legends

[11]      H←2.5×ΔCY⍝Height of each legend

[12]      ROWS←N⌊⌊((-⍨/RECT[;2])-ΔCY)÷H

[13]      'Legend too high'⎕SIGNAL(ROWS≤0)/201

[14]      COLS←⌊(N÷ROWS)⌊((-⍨/RECT[;1])-2×ΔCX)÷L
```

```
[15]     'Legend too wide'⎕SIGNAL(COLS≤0)/202
[16]     COLS←⌈N÷ROWS ◊ ROWS←⌈N÷COLS
[17]     SIZE←((2×ΔCX) ΔCY)+(COLS ROWS)×L H
[18]     'Legend too large'⎕SIGNAL(∨/SIZE>-⍨⌿RECT)/203
[19]     START←RECT[1;]+0.5×(-⍨⌿RECT)-SIZE
[20]     XO←START[1]+ΔCX+L×¯1+⍳COLS
[21]     XO←(ROWS×⍴XO)⍴XO
[22]     YO←COLS/START[2]+(ΔCY×0.5)+H×⌽¯1+⍳ROWS
[23]     XBOX←↓(XO+ΔCX)∘.+0 0 1 1 0×LBOX×ΔCX
[24]     YBOX←↓(YO+ΔCY×0.5)∘.+0 1 1 0 0×1.5×ΔCY
[25]     BOX←N⍴XBOX [1.5]¨YBOX
[26]     XTEXT←N⍴XO+ΔCX×2+LBOX
[27]     YTEXT←N⍴YO+ΔCY×0.75
[28]     RECT←START [0.5]START+SIZE
[29]     →DOIF 0=DEVINFO[46] ◊ 1 BCOL VBAR RECT          ⍝   Blank out backg
round
[30]     (↓⍉↑DINT DCOL DSTYLE)VFAREA BOX
[31]     (0 ¨DCOL)VFAREA BOX
[32]     →DOIF 0=DEVINFO[46] ◊ VBMODE 1
[33]     FONT TCOL VGTEXT XTEXT YTEXT TEXT
[34]     →DOIF 0=DEVINFO[46] ◊ VBMODE 0
[35]     →(0=BOXED)/Exit
[36]      1 BOXED VPLINE RECTANGLE RECT
[37]    Exit:

****************************************************************************

LEGEND_L

[0]     LEGEND_L PARS;MCOL;N;L;H;COLS;ROWS;SIZE;START;X;DCOL;DSTYLE;LINE;TP
OS;LT;ΔCX;ΔCY;FONT;BOXED;RECT;LBOX;TCOL;PLYM;SIZE;XO;YO;XTEXT;YTEXT;XLINE;Y
LINE;BCOL
[1]    ⍝ Displays a line legend into given rectangle RECT
[2]     MCOL RECT FONT SIZE DSTYLE DCOL PLYM TEXT TCOL BOXED BCOL←PARS
[3]     →DOIF~36⊃DEVINFO ◊ TCOL BOXED≠←0 ⍝ If mono  set cols to black
```

```
[4]     TEXT←↓↑TEXT

[5]     RECTρ⍨←2 2

[6]     RECT×←0.01×2 2ρ¯1+DEVINFO[52 53]

[7]     VSTFNT FONT ◊ ∆CX ∆CY←⌈⌿2 2ρVSTHGT SIZE

[8]     LBOX←4⌊LT←ρ⊃TEXT ⍝ LBOX = length of legend box  LT = length of text

[9]     L←(∆CX×3+LBOX)+MAXSTRLEN TEXT⍝ Length of each legend

[10]     N←ρTEXT ⍝ Number of legends

[11]     H←2×∆CY⍝Height of each legend

[12]     ROWS←N⌊⌊((-⍨/RECT[;2])-∆CY)÷H

[13]     'Legend too large'⎕SIGNAL(ROWS≤0)/201

[14]     COLS←⌊(N÷ROWS)⌊((-⍨/RECT[;1])-2×∆CX)÷L

[15]     'Legend too large'⎕SIGNAL(COLS≤0)/201

[16]     COLS←⌈N÷ROWS ◊ ROWS←⌈N÷COLS

[17]     SIZE←((2×∆CX) ∆CY)+(COLS ROWS)×L H

[18]     'Legend too large'⎕SIGNAL(∨/SIZE>-⍨⌿RECT)/201

[19]     START←RECT[1;]+0.5×(-⍨⌿RECT)-SIZE

[20]     XO←START[1]+∆CX+L×¯1+⍳COLS

[21]     XO←(ROWS×ρXO)ρXO

[22]     YO←COLS/START[2]+(∆CY×0.5)+H×⌽¯1+⍳ROWS

[23]     XLINE←Nρ↓XO∘.+∆CX×+\1 LBOX

[24]     YLINE←NρYO+∆CY

[25]     XTEXT←NρXO+∆CX×2+LBOX

[26]     YTEXT←NρYO+∆CY×0.75

[27]     RECT←START [0.5]START+SIZE

[28]     →DOIF 0=DEVINFO[46] ◊ 1 BCOL VBAR RECT          ⍝   Blank out backg

round

[29]     (DSTYLE ¨DCOL)VPLINE XLINE [1.5]¨YLINE

[30]     →DOIF 0<ρ PLYM ◊ (PLYM ¨MCOL)VPMARK 1 2∘ρ¨(NρXO+∆CX×1+LBOX×0.5) ¨Y

LINE

[31]     →DOIF 0=DEVINFO[46] ◊ VBMODE 1

[32]     FONT TCOL VGTEXT XTEXT YTEXT TEXT

[33]     →DOIF 0=DEVINFO[46] ◊ VBMODE 0

[34]     →(0=BOXED)/Exit
```

```
[35]     1 BOXED VPLINE RECTANGLE RECT
[36]    Exit:

******************************************************************************

LNORM

[0]     RECT←LNORM RECT;MIN;MAX;XS;YS;DIFFX;DIFFY
[1]    ⍝ Adjust Legend Rectangle to be integer
[2]    ⍝ and have first row minimum  second row maximum.
[3]    ⍝ Scale the rectangle dimensions to be multiples of pixel dimensions
[4]     RECT←2 2⍴RECT
[5]     RECT←(⌊RECT[1;]) [0.5]⌈RECT[2;]
[6]     MIN←⌊⌿RECT
[7]     MAX←⌈⌿RECT
[8]     XS YS←DEVINFO[52 53]÷DEVINFO[1 2]
[9]     MAX←⌊XS YS×⌈MAX÷XS YS
[10]     MIN←⌊XS YS×⌈MIN÷XS YS
[11]     RECT←MIN [0.5]MAX

******************************************************************************

MAXSTRLEN

[0]     R←MAXSTRLEN LABELS
[1]    ⍝  Calculates maximum length of LABELS when written in current font
[2]     →DOIF 0=DEVINFO[46] ◊ →VQTDES[4]/Prop   ⍝ proportionally spaced fon
t ?
[3]     R←(⌈/VQTATT[7 9])×⌈/⊃ /⍴¨LABELS ◊ →Exit
[4]    Prop:R←⌈/STRLEN¨LABELS
[5]    Exit:

******************************************************************************

MESSAGE

[0]     MESSAGE MSG
```

```
[1]     MSG PPUT'MSG'

[2]     PPUT¨(0=⍴MSG)↓2 1  ⍝ Ring bell (unless message empty) and flush


****************************************************************************

NICE


[0]     V←NICE X;A;M1;NAL;I

[1]     I←((A÷10*NAL←(¯1×A<1)+(×M1)×⌊|M1←10⍟A←(X[2]-X[1])÷X[3])<1.414214 3.
162278 7.071068)⍳1

[2]     V[1]←(×V[1])×⌊|V[1]←X[1]÷V[3]←(1 2 5 10)[I]×10*NAL 0⍴V←3⍴0

[3]     V[1]←V[1]-0>X[1]÷V[3]

[4]     V[1]←V[3]×V[1]+(V[1]+1-X[1]÷V[3])<0.00002

[5]     V[2]←(×V[2])×⌊1+V[2]←X[2]÷V[3]

[6]     V[2]←V[2]-¯1>X[2]÷V[3]

[7]     V[2]←V[3]×V[2]-(|(X[2]÷V[3])+1-V[2])<0.00002


****************************************************************************

PAT


[0]     FIXED PAT ROLLING

[1]     VSLCOL⊃COLS

[2]     FIXED HYPO ROLLING

[3]     VSLCOL 2⊃COLS

[4]     COLS←2⌽COLS

[5]     FIXED EPI ROLLING


****************************************************************************

PAUSE


[0]     PAUSE;X

[1]     X←VRQSTR 2


****************************************************************************
```

```
 PDEFINE
[0]     {R}←PDEFINE P
[1]    ⍝Define nested panel P - P[1] 6col mx  P[2] standard txt  P[3] label
s
[2]     PDEFN 1⊃P
[3]     (2⊃P)PPUT⍳1↑⊃⍴¨P
[4]     R←PDEFN 3⊃P

******************************************************************************
PFUN2

[0]     PARS←PFUN2;RE;QTOP;TEC;RES;LA1;LL1;Q1;Q2;Q3
[1]     RE←REF[I-1]
[2]     QTOP←1000000×TAB[I;1]÷RE ◊ TEC←TAB[I;2] ◊ RES←TAB[I;9]
[3]     LL1←0 S×⍳⌊LLMAX÷S
[4]     LA1← S×⌊TAB[I;3]÷S
[5]     Q1←WTIX×QTOP×1-*-LL1×TEC÷QTOP
[6]     Q2←Q1×(1-RES÷100)
[7]     Q3← QTOP×(1-0.01×RES+TAB[I;8])
[8]     LL1 LA1←LCON×LL1 LA1
[9]     Q1 Q2 Q3←QCON×Q1 Q2 Q3
[10]     PARS←Q1 Q2 Q3 LL1 LA1

******************************************************************************
PIE

[0]     R←PIE X;WIN;DATA;COLOUR;PAT;OFFSET;LABELS;BOXED;ΔCX;ΔCY;W;RADIUS;XO
;YO;GDT_THETA;THETA;S;C;X1;Y1;TX;TY;FONT;STYLE;SIZE
[1]     WIN DATA STYLE COLOUR PAT OFFSET LABELS FONT SIZE BOXED←X
[2]     →DOIF 1=DEVINFO[46] ◊ PAT←1+6|PAT ⍝ plotters only have 6 patterns
[3]     →DOIF~36⊃DEVINFO ◊ COLOUR BOXED≠←0 ⍝ If mono  set cols to black
[4]     VSTFNT FONT ◊ ΔCX ΔCY←⌈/2 2⍴VSTHGT SIZE
[5]     W←(0.01× WIN)× 2 2⍴¯1+DEVINFO[52 53]
[6]     →DOIF 0≠BOXED ◊ 1 BOXED VPLINE RECTANGLE 2 2⍴W
[7]     W+←1 1 ¯1 ¯1×4⍴0.5×ΔCX ΔCY
```

```
[8]      RADIUS←0.5×⌊/W[3 4]-W[1 2]

[9]      X0 Y0←0.5×W[1 4]+W[3 2]

[10]      GDT_THETA←⌊+\0 3600×0.01×DATA  ⍝ Angles in GDT coordinates (1/10th
 deg)

[11]      THETA←+\0 0.01×DATA×○2        ⍝ Angles in radians

[12]      S←1○(¯1↓THETA)+0.5×(1↓THETA)-(¯1↓THETA)

[13]      C←2○(¯1↓THETA)+0.5×(1↓THETA)-(¯1↓THETA)

[14]      X1←X0+OFFSET×RADIUS×0.05×C

[15]      Y1←Y0+OFFSET×RADIUS×0.05×S

[16]      TX←LABELS/X1+(0.5×RADIUS×C)-3×ΔCX×C<0

[17]      TY←LABELS/Y1+(0.5×RADIUS×S)-1×ΔCY×S<0

[18]      DATA←LABELS/↓'X1 I2 <% >'⎕FMT DATA

[19]      STYLE COLOUR PAT VPIESL X1 Y1 RADIUS(¯1↓GDT_THETA)

[20]      0 COLOUR PAT VPIESL X1 Y1 RADIUS(¯1↓GDT_THETA)

[21]      COLOUR←LABELS/COLOUR

[22]      FONT COLOUR VGTEXT TX TY DATA

************************************************************************

PIECHART

[0]      PIECHART;PLOT;BOXED;ΔRANK;BCOL;DINT;OFFSETS;WINL;WINP;DEVICE;DEVINF
Q;ΔINC;ΔI;PDATA;XLABB;XLAB;FORMS;TEXT;CCOL;CTITLE;XEXP;XCOL;YEXP;DCOL;DSTYL
E;LABELS;XVAL;YVAL;X;KEY;MASK;FONT;SIZE;MSG

[1]      PLOT←'R'

[2]      CCOL←XCOL←5 ◊ OFFSETS←6⍴0 ◊ DCOL←LABELS←6⍴1 ◊ DSTYLE←0 ⍳5

[3]      CTITLE←YEXP←XEXP←' '

[4]      PSTART

[5]      PENTER 1

[6]      X←PKEY 0 1 5 10

[7]      MSG←'Press ' (⊃X) ' to draw piechart  ' (2⊃X) ' for help  '

[8]      MSG ←(3⊃X) ' to print piechart  ' (4⊃X) ' to quit.'

[9]      PDEFINE ΔPLOTSCREENP

[10]      OFFSETS LABELS CTITLE XEXP YEXP CCOL XCOL DCOL DSTYLE PPUT'OFFSETS
' 'LABELS' 'CTITLE' 'XEXP' 'YEXP' 'CCOL' 'XCOL' 'DCOL' 'DSTYLE'
```

```
[11]    Reset:MSG PPUT'MSG'
[12]    Again:→(0 1 5 10=KEY←PREAD)/Doit Help Doit Exit ◊ PPUT 2 ◊ →Again
[13]    Help:GHELP 3 ◊ →Again
[14]    Doit:
[15]     →DOIF 5=KEY ◊ 'PLOTTER OF PRINTER (P/R)?' ◊ PLOR←⍞
[16]     YEXP←PGET'YEXP'
[17]     →Again×⍳~∨/MASK←(∨/YEXP≠' ')∧~∨/YEXP∊'↔'
[18]     CTITLE XEXP←1 PGET'CTITLE' 'XEXP'
[19]     OFFSETS LABELS CCOL XCOL DCOL DSTYLE←PGETN'OFFSETS' 'LABELS' 'CCOL
' 'XCOL' 'DCOL' 'DSTYLE'
[20]     OFFSETS LABELS YEXP DCOL DSTYLE/⍨←⊂MASK
[21]     CCOL XCOLρ⍨←⊂⍬⍝  Ensure these are scalars
[22]     OFFSETS←2⊤OFFSETS ◊ LABELS←2⊤LABELS
[23]     →(∧/DSTYLE∊0 ⍳9)/Ok1
[24]     MESSAGE'Valid bar styles are in the range 0-9'
[25]     →Again
[26]    Ok1:⎕EX¨'X' 'XVAL'
[27]     →L1×⍳∧/' '=XEXP
[28]     X←XVAL←⍎XEXP
[29]    L1:
[30]     YVAL←⍉↑⍎¨↓YEXP
[31]     YVAL←(¯2↑1 ρYVAL)ρYVAL
[32]     ⍎(2≠⎕NC'XVAL')/'XVAL←(1↑ρYVAL)ρ'' '''
[33]     →((ρXVAL)=1↑ρYVAL)/L2
[34]     MESSAGE'Number of labels should be equal to the number of pie char
ts'
[35]     →Again
[36]    L2:
[37]     →(2≤¯1↑ρYVAL)/LL2
[38]     MESSAGE'There must be at least two variables' ◊ →Again
[39]    LL2:
[40]    ⍝ CTITLE←dbr∘ ¨CTITLE
[41]     TEXT←↓YEXP ◊ TEXT←5↑¨TEXT
```

```
[42]    WINP←WINPP KEY PLOT'C' ◊ DINT←(¯1↑⍴YVAL)⍴2
[43]    WINL←WINPP KEY PLOT'B'
[44]    XLAB←⍕¨XVAL
[45]    FORMS←0 2 5 10 20 22 18 24 11 29
[46]    DSTYLE←FORMS[1+DSTYLE]
[47]    BCOL←0
[48]    ΔRANK←1↑⍴YVAL ◊ ⍎(ΔRANK≥3)/'ΔRANK←3'
[49]    ΔINC←(WINP[3]-WINP[1])÷ΔRANK
[50]    WINP[3]←WINL[3]←ΔINC+WINP[1]
[51]    ΔI←0
[52]    CGISTART
[53]    CGINIT KEY PLOT
[54]    →DOIF 0=DEVINFO[46] ◊ VSBCOL BCOL
[55]    →DOIF 0=KEY ◊ VCLRWK
[56]   pies:ΔI←ΔI+1
[57]    →DOIF 0=DEVINFO[46] ◊ FONT SIZE BOXED←4 586 0 ◊ →LL1
[58]    FONT SIZE BOXED←4 349 0
[59]   LL1:
[60]    PDATA←100×YVAL[ΔI;]÷+/YVAL[ΔI;]
[61]    PIE WINP PDATA DINT DCOL DSTYLE OFFSETS LABELS FONT SIZE BOXED
[62]    FONT SIZE BOXED←2 500 0
[63]    XLABB←⊃XLAB[ΔI]
[64]    TITLE WINL XLABB XCOL FONT SIZE BOXED
[65]    WINL[1]←WINP[1]←WINP[3] ◊ WINP[3]+←ΔINC ◊ WINL[3]+←ΔINC
[66]    →(ΔI<ΔRANK)/pies
[67]    WINL←WINPP KEY PLOT'L'
[68]    →DOIF 0=DEVINFO[46] ◊ FONT SIZE BOXED←1 586 0 ◊ →LL4
[69]    FONT SIZE BOXED←1 349 0
[70]   LL4:
[71]    LEGEND WINL FONT SIZE DINT DCOL DSTYLE TEXT XCOL BOXED BCOL
[72]    WINL←WINPP KEY PLOT'T'
[73]    →DOIF 0=DEVINFO[46] ◊ FONT SIZE BOXED←2 911 1 ◊ →LL3
[74]    FONT SIZE BOXED←2 750 1
```

```
[75]    LL3:
[76]     TITLE WINL CTITLE CCOL FONT SIZE BOXED
[77]     FRAME 5 KEY PLOT
[78]     →DOIF 0=KEY ⋄ PAUSE
[79]     RESET ⋄ PREDRAW
[80]     →Reset
[81]    Exit:PRESET
```

****

PLOT

```
[0]     PLOT;∆NAM;∆MENU;EXITKEYS;RESULTS
[1]    ⍝
[2]    ⍝  This function prepares a menu for drawing a graph of one (optiona
1)
[3]    ⍝  X-variable and up to 5 Y-variables.
[4]    ⍝  Written by E.Taymaz  Jan.1990.
[5]    ⍝
[6]     ⎕SM←MENUG
[7]    DE:EXITKEYS←'F1' 'F9'
[8]     RESULTS←EXITKEYS 3 ⎕SR 2+⍳15
[9]     →(RESULTS[4]≡⊂'F9')/0
[10]     →(RESULTS[1]=15 16 17)/DRW SAV EXT
[11]     →DE
[12]    DRW:
[13]     →(~((∧/' '=⊃⎕SM[5;1])∨(2=⎕NC⊃⎕SM[5;1])))/DD
[14]     →(~((∧/' '=⊃⎕SM[8;1])∨(2=⎕NC⊃⎕SM[8;1])))/DD
[15]     →(~((∧/' '=⊃⎕SM[11;1])∨(2=⎕NC⊃⎕SM[11;1])))/DD
[16]     →(~((∧/' '=⊃⎕SM[12;1])∨(2=⎕NC⊃⎕SM[12;1])))/DD
[17]     →(~((∧/' '=⊃⎕SM[13;1])∨(2=⎕NC⊃⎕SM[13;1])))/DD
[18]     →(~((∧/' '=⊃⎕SM[14;1])∨(2=⎕NC⊃⎕SM[14;1])))/DD
[19]     →DEVV
[20]    DD:
```

```
[21]     ⎕SM[18;1]←⊂5 36ρ'                                     VARIABLE NAM
E IS NOT VALID.          PLEASE RE-ENTER THE VARIABLE NAMES. PRESS ANY KEY T
O CONTINUE.             '
[22]     →ERR
[23]    DEVV:→(~(∧/' '=⊃⎕SM[8;1]))/DEV
[24]     ⎕SM[18;1]←⊂4 36ρ(36ρ' ') ' PLEASE ENTER THE FIRST Y-VARIABLE.   PR
ESS ANY KEY TO CONTINUE.' 36ρ' '
[25]     →ERR
[26]    DEV:Δ̲M̲E̲N̲U̲←⎕SM
[27]     ΔNAM←PLOTΔDRAW
[28]     ⎕SM←Δ̲M̲E̲N̲U̲
[29]     →DE
[30]    SAV:
[31]     ⍎(2≠⎕NC'ΔNAM')/'⎕SM[18;1]←⊂4 30 ρ(30ρ'' '') '' GRAPH SHOULD BE DRA
WN FIRST.   PRESS ANY KEY TO CONTINUE.'' 30ρ'' '' ◊ →ERR'
[32]     ⍎(∧/' '=⊃⎕SM[3;1])/'⎕SM[18;1]←⊂4 30 ρ(30ρ'' '') '' PLEASE ENTER A
GRAPH NAME.      PRESS ANY KEY TO CONTINUE.'' 30ρ'' '' ◊→ERR'
[33]     ⍎(⊃⎕SM[3;1]) '←ΔNAM'
[34]     →DE
[35]    ERR:
[36]     EXITKEYS←''
[37]     ⎕ARBOUT 7
[38]     RESULTS←EXITKEYS 18 ⎕SR 18
[39]     ⎕SM[18;]←M̲E̲N̲U̲G̲[18;]
[40]     →DE
[41]    EXT:⎕SM←0 13ρ0
[42]     →0
```

****

PLOTΔDRAW

```
[0]     ΔGR←PLOTΔDRAW;ΔEX;ΔR;ΔI;ΔMS;ΔX;ΔY1;ΔY2;ΔY3;ΔY4;ΔY5;ΔXX;ΔXM;ΔYX;ΔYM
[1]    ⍝
[2]    ⍝  This function is used in the 'PLOT' function.
```

```
[3]    ⍝  Written by E.Taymaz  Jan.1990.
[4]    ⍝
[5]     ΔY1←⍎⊃ΔM̲E̲N̲U̲[8;1]
[6]     ΔY5←ΔY4←ΔY3←ΔY2←ΔY1
[7]     ΔX←⍳ρΔY1
[8]     ⍎(~(∧/' '=⊃ΔM̲E̲N̲U̲[5;1]))/'ΔX←⍎⊃ΔM̲E̲N̲U̲[5;1]'
[9]     ⍎(~(∧/' '=⊃ΔM̲E̲N̲U̲[11;1]))/'ΔY2←⍎⊃ΔM̲E̲N̲U̲[11;1]'
[10]    ⍎(~(∧/' '=⊃ΔM̲E̲N̲U̲[12;1]))/'ΔY3←⍎⊃ΔM̲E̲N̲U̲[12;1]'
[11]    ⍎(~(∧/' '=⊃ΔM̲E̲N̲U̲[13;1]))/'ΔY4←⍎⊃ΔM̲E̲N̲U̲[13;1]'
[12]    ⍎(~(∧/' '=⊃ΔM̲E̲N̲U̲[14;1]))/'ΔY5←⍎⊃ΔM̲E̲N̲U̲[14;1]'
[13]    ΔXX←⌈/ΔX ⋄ ΔXM←⌊/ΔX ⋄ ΔYX←⌈/ΔY1 ΔY2 ΔY3 ΔY4 ΔY5 ⋄ ΔYM←⌊/ΔY1 ΔY2 ΔY
3 ΔY4 ΔY5
[14]    ⍎((⎕VFI⊃ΔM̲E̲N̲U̲[6;1])[1]≡⊂ 1)/'ΔXX←⊃(⎕VFI ⊃ΔM̲E̲N̲U̲[6;1])[2]'
[15]    ⍎((⎕VFI⊃ΔM̲E̲N̲U̲[7;1])[1]≡⊂ 1)/'ΔXM←⊃(⎕VFI ⊃ΔM̲E̲N̲U̲[7;1])[2]'
[16]    ⍎((⎕VFI⊃ΔM̲E̲N̲U̲[9;1])[1]≡⊂ 1)/'ΔYX←⊃(⎕VFI ⊃ΔM̲E̲N̲U̲[9;1])[2]'
[17]    ⍎((⎕VFI⊃ΔM̲E̲N̲U̲[10;1])[1]≡⊂ 1)/'ΔYM←⊃(⎕VFI ⊃ΔM̲E̲N̲U̲[10;1])[2]'
[18]
[19]    ΔR←⌊/(ρΔX) (ρΔY1) (ρΔY2) (ρΔY3) (ρΔY4) (ρΔY5)
[20]    ΔX←ΔRρΔX ⋄ ΔY1←ΔRρΔY1 ⋄ ΔY2←ΔRρΔY2
[21]    ΔY3←ΔRρΔY3 ⋄ ΔY4←ΔRρΔY4 ⋄ ΔY5←ΔRρΔY5
[22]
[23]    ((ΔX>ΔXX)/ΔX)←ΔXX
[24]    ((ΔX<ΔXM)/ΔX)←ΔXM
[25]    ((ΔY1>ΔYX)/ΔY1)←ΔYX
[26]    ((ΔY2>ΔYX)/ΔY2)←ΔYX
[27]    ((ΔY3>ΔYX)/ΔY3)←ΔYX
[28]    ((ΔY4>ΔYX)/ΔY4)←ΔYX
[29]    ((ΔY5>ΔYX)/ΔY5)←ΔYX
[30]    ((ΔY1<ΔYM)/ΔY1)←ΔYM
[31]    ((ΔY2<ΔYM)/ΔY2)←ΔYM
[32]    ((ΔY3<ΔYM)/ΔY3)←ΔYM
[33]    ((ΔY4<ΔYM)/ΔY4)←ΔYM
[34]    ((ΔY5<ΔYM)/ΔY5)←ΔYM
```

```
[35]
[36]      ΔGR←23 80ρ' ' ◊ ΔGR[2 23;]←' ' ◊ ΔGR[;1 80]←' '
[37]      ΔGR[2;1]←' ' ◊ ΔGR[2;80]←' ' ◊ ΔGR[23;1]←' ' ◊ ΔGR[23;80]←' '
[38]      ΔGR[1;]←80ρ(28ρ' ') ⊃ΔMENU[4;1]
[39]      ΔMS←'X-MAX: ' ⍕ΔXX '  X-MIN: ' ⍕ΔXM '  Y-MAX: ' ⍕ΔYX '  Y-MIN: ' ⍕
ΔYM
[40]      ΔX←1+⌊0.5+79×(ΔX-ΔXM)÷ΔXX-ΔXM
[41]      ΔY1←22-⌊0.5+20×(ΔY1-ΔYM)÷ΔYX-ΔYM
[42]      ΔY2←22-⌊0.5+20×(ΔY2-ΔYM)÷ΔYX-ΔYM
[43]      ΔY3←22-⌊0.5+20×(ΔY3-ΔYM)÷ΔYX-ΔYM
[44]      ΔY4←22-⌊0.5+20×(ΔY4-ΔYM)÷ΔYX-ΔYM
[45]      ΔY5←22-⌊0.5+20×(ΔY5-ΔYM)÷ΔYX-ΔYM
[46]
[47]
[48]      ΔEX←''
[49]      ΔMENU[6 7 9 10;1]←⊂12ρ' '
[50]      ΔI←0
[51]     BEG:ΔI←ΔI+1
[52]      ΔGR[ΔY5[ΔI] ΔY4[ΔI] ΔY3[ΔI] ΔY2[ΔI] ΔY1[ΔI];ΔX[ΔI]]←'Δ' 'o' '+' '*
' '◊'
[53]      →(ΔI<ΔR)/BEG
[54]      ⎕SM←2 13ρ(ΔGR)1 1 0 0 0 0 1 ¯1 1 1 0 0(ΔMS)25 5 0 0 0 0 0 ¯1 1 1 0
 0
[55]      ΔEX←ΔEX 2 ⎕SR 2
[56]      ΔGR←⎕FMT ΔGR [1]1 80ρΔMS 80ρ' '

************************************************************************
PLOTINIT

[0]      PLOTINIT
[1]      DEVICE DEVINFO←VOPNWK'PLOTTER'

************************************************************************
```

```
PRESET

[0]      PRESET
[1]     ⍝Release all panels
[2]      →(0≠1↑PDEFN-1↑PDEFN 0)⍴⎕LC

******************************************************************************

PSTART

[0]      PSTART
[1]     ⍝Start up prefect external function library if not there yet
[2]      →(0≠⎕NC'PDEFN')⍴SDBR
[3]      'prefect'⎕SH'prefect'
[4]     ⍝Start up the dbr function from xutils  too
[5]     SDBR:→(0≠⎕NC'dbr')⍴0
[6]      'xutils'⎕SH'xutils' 'dbr'

******************************************************************************

RECTANGLE

[0]      R←RECTANGLE X
[1]      R←X[1 1 2 2 1;]
[2]      R[2;2]←X[2;2]
[3]      R[4;2]←X[1;2]

******************************************************************************

RESET

[0]      RESET
[1]     ⍝ Closes CGI workstation and reloads APL font
[2]      VCLSWK
[3]      APL

******************************************************************************

SALTERC
```

```
[0]     SALTERC;∆RR;∆SYN;∆NWE;MCOL;PLOT;CTYPE;BCOL;BOXED;TEXT;DEVICE;DEVINF
O;GRID;HGR;VGR;CTITLE;XTITLE;YTITLE;XEXP;YEXP;CCOL;XCOL;YCOL;DCOL;DSTYLE;XV
AL;YVAL;WIN;X;KEY;MASK;WIN;PLYM;FONT;SIZE;MSG
[1]     PLOT←'R'
[2]     CTITLE←XTITLE←YTITLE←' ' ◊ CCOL←XCOL←YCOL←5 ◊ DCOL←6⍴1 ◊ DSTYLE←⍳6
[3]     XEXP←YEXP←' ' ◊ HGR←VGR←CTYPE←0 ◊ PLYM←6⍴0
[4]     PSTART
[5]     PENTER 1
[6]     X←PKEY 0 1 5 10
[7]     MSG←'Press ' (⊃X) ' to draw graph  ' (2⊃X) ' for help  '
[8]     MSG ←(3⊃X) ' to print graph  ' (4⊃X) ' to quit.'
[9]     FONT SIZE←1 586
[10]     PDEFINE ∆PLOTSCREEN
[11]     PLYM CTYPE HGR VGR CTITLE XTITLE YTITLE XEXP YEXP CCOL XCOL YCOL D
COL DSTYLE PPUT'PLYM' 'CTYPE' 'HGR' 'VGR' 'CTITLE' 'XTITLE' 'YTITLE' 'XEXP'
 'YEXP' 'CCOL' 'XCOL' 'YCOL' 'DCOL' 'DSTYLE'
[12]    Reset:MSG PPUT'MSG'
[13]    Again:→(0 1 5 10=KEY←PREAD)/Doit Help Doit Exit ◊ PPUT 2 ◊ →Again
[14]    Help:GHELP 1 ◊ →Again
[15]    Doit:
[16]     →DOIF 5=KEY ◊ 'PLOTTER OR PRINTER (P/R)?' ◊ PLOT←⍞
[17]     YEXP←PGET'YEXP'
[18]     →Again×⍳~∨/MASK←(∨/YEXP≠' ')∧~∨/YEXP∊'←→'
[19]     CTITLE XTITLE YTITLE XEXP←1 PGET'CTITLE' 'XTITLE' 'YTITLE' 'XEXP'
[20]     PLYM CTYPE HGR VGR CCOL XCOL YCOL DCOL DSTYLE←PGETN'PLYM' 'CTYPE'
'HGR' 'VGR' 'CCOL' 'XCOL' 'YCOL' 'DCOL' 'DSTYLE'
[21]     PLYM YEXP DCOL DSTYLE/¨←⊂MASK
[22]     CTYPE HGR VGR CCOL XCOL YCOL⍴¨←⊂⍬⍝  Ensure these are scalars
[23]     HGR←2⊤HGR ◊ VGR←2⊤VGR
[24]     GRID←2 2⍴VGR HGR(VGR×XCOL)(HGR×YCOL)
[25]     →(∧/DSTYLE∊0 ⍳7)/Ok1
[26]     MESSAGE'Valid line styles are in the range 1-7'
[27]     →Again
```

```
[28]    Ok1:→(∧/PLYM∈0 ⍳7)/OK2
[29]     MESSAGE'Valid marker styles are in the range 1-7'
[30]     →Again
[31]    OK2:
[32]     MCOL←DCOL ◊ ((PLYM=0)/MCOL)←0 ◊ ((DSTYLE=0)/DCOL)←0
[33]     ⎕EX¨'X' 'XVAL'
[34]     →L1×⍳∧/' '=XEXP
[35]     X←XVAL←⍎XEXP
[36]    L1:YVAL←⍉↑⍎¨↓YEXP
[37]     YVAL←(2↑(⍴YVAL) 1)⍴YVAL
[38]     ⍎(2≠⎕NC'XVAL')/'XVAL←⍳1↑⍴YVAL'
[39]     →(⊃∧/(⍴XVAL)=⍴¨⍎¨↓YEXP)/L2
[40]     MESSAGE'Variables must be in same size'
[41]     →Again
[42]    L2:
[43]     →(0=∧/(↓YVAL)[1]≡¨↓YVAL)/L8
[44]     MESSAGE'Variable(s) have same values' ◊ →Again
[45]    L8:
[46]     YVAL←YVAL[;1]
[47]     →DOIF 2=⎕NC'SHOW1' ◊ ∆SYN←SHOW1/YVAL
[48]     →DOIF 2=⎕NC'SHOW2' ◊ ∆NWE←SHOW2/YVAL
[49]     XVAL←XVAL[⍒YVAL] ◊ YVAL←YVAL[⍒YVAL]
[50]     XVAL←100×(+\XVAL)÷+/XVAL
[51]     ∆RR←(⍴YVAL)⍴2 ◊ YVAL←∆RR\YVAL ◊ XVAL←∆RR\XVAL
[52]     YVAL←0 YVAL 0 0 ◊ XVAL←0 0 XVAL 0
[53]     →DOIF 2=⎕NC'SHOW1' ◊ ∆RR←YVAL∈∆SYN ◊ ∆SYN←YVAL×0 ◊ (∆RR/∆SYN)←∆RR/
YVAL
[54]     →DOIF 2=⎕NC'SHOW2' ◊ ∆RR←YVAL∈∆NWE ◊ ∆NWE←YVAL×0 ◊ (∆RR/∆NWE)←∆RR/
YVAL
[55]     CTITLE XTITLE YTITLE←dbr° ¨CTITLE XTITLE YTITLE
[56]     TEXT←↓YEXP ◊ TEXT←5↑¨TEXT
[57]     WIN←WINC KEY PLOT'C'
[58]     CGISTART
```

```
[59]     CGINIT KEY PLOT
[60]     BCOL←0 ◊ FONT SIZE BOXED←1 586 0
[61]     →DOIF 0=DEVINFO[46] ◊ FONT SIZE BOXED←1 586 0 ◊ →LL2
[62]     FONT SIZE BOXED←1 349 0
[63]    LL2:
[64]     GRAPHS MCOL WIN CCOL XVAL YVAL DSTYLE DCOL PLYM GRID XCOL YCOL XTI
TLE YTITLE FONT SIZE
[65]     WIN←WINC KEY PLOT'L'
[66]     →DOIF 0≠CTYPE ◊ LEGEND_L MCOL WIN FONT SIZE DSTYLE DCOL PLYM TEXT
XCOL BOXED BCOL
[67]     WIN←WINC KEY PLOT'T'
[68]     →DOIF 0=DEVINFO[46] ◊ FONT SIZE BOXED←2 911 1 ◊ →LL9
[69]     FONT SIZE BOXED←2 698 1
[70]    LL9:
[71]     TITLE WIN CTITLE CCOL FONT SIZE BOXED
[72]     →DOIF 0=KEY ◊ PAUSE
[73]     RESET ◊ PREDRAW
[74]     →Reset
[75]    Exit:PRESET
```

****

SETSCALE

```
[0]     R←INCREMENT SETSCALE VALUES;LABEL;MIN;MAX
[1]    ⍝ Calculates MIN  MAX  INCREMENT  and LABEL to annotate an axis
[2]    ⍝ Suggested INCREMENT and data VALUES are supplied
[3]     MIN MAX INCREMENT←NICE(⌊/VALUES) (⌈/VALUES← VALUES) INCREMENT
[4]     LABEL←⍕¨MIN+0 INCREMENT×⍳(MAX-MIN)÷INCREMENT
[5]     R←MIN MAX INCREMENT(1+⌊(MAX-MIN)÷INCREMENT)LABEL
```

****

SHOWFUN

```
[0]     SHOWFUN ∆F;I;TIT;⎕TRAP;REF;∆M;∆T;S;TAB;QT1;QT2;QP2;QP2;QA1;QA2;LL1;
```

```
LL2;LA1;LA2;LCO;QCON;LMAX;QMAX;RANK;W
[1]     ⍝
[2]     ⍝ This function prepares an animated chart of the production functio
n of
[3]     ⍝ a firm whose data table is created before.
[4]     ⍝ Written by E.Taymaz  March 1990.
[5]     ⍝
[6]      REF←IMPLPΔREF
[7]      →(0 1 2 3 4∊⍎ΔF)/IND SEC SEC SEC SEC
[8]      TIT←'Production Function of Firm ' ΔF
[9]      ΔM←⌊⍎ΔF ◊ ((ΔF='.')/ΔF)←'X'
[10]     TAB←'YEARLYΔFIRMΔ' ΔF
[11]     TAB←⍎TAB
[12]     REF←(¯1+1↑⍴TAB)⍴REF[ΔM]
[13]     →JMP
[14]    IND:TAB←YEARLYΔINDUSTRYΔTOTAL
[15]     ΔM←1
[16]     TIT←'Production Function of the Manufacturing Industry'
[17]     ΔT←1+⍳(¯1+1↑⍴TAB)
[18]     REF←((YEARLYΔMARKET1[ΔT;1]×REF[1])+(YEARLYΔMARKET2[ΔT;1]×REF[2])+(
YEARLYΔMARKET3[ΔT;1]×REF[3])+(YEARLYΔMARKET4[ΔT;1]×REF[4]))÷TAB[ΔT;1]
[19]     →JMP
[20]    SEC:TAB←'YEARLYΔMARKET' ΔF ◊ ΔM←⍎ΔF
[21]     TIT←(1 2 3 4=ΔM)/'Raw Materials' 'Intermediate Goods' 'Investment
Goods' 'Consumer Goods'
[22]     TIT←(⍕TIT) ' Production Function'
[23]     TAB←⍎TAB ◊ REF←(¯1+1↑⍴TAB)⍴REF[ΔM]
[24]    JMP:
[25]     CGISTART
[26]     DEVICE DEVINFO←VOPNWK'DISPLAY'
[27]     ⎕TRAP←0 'E' 'RESET'
[28]     TII←TIT
[29]     TITLE(0 90 100 100)(TIT)1 2 750 1
```

```
[30]    TITLE(0 80 10 87)('Output')1 2 500 0
[31]    TITLE(90 0 100 7)('Labour')1 2 500 0
[32]    W←0 0 ¯1+DEVINFO[52 53]
[33]    VPLINE(W[1 1 3 3 1])(W[2 4 4 2 2])
[34]    RANK←¯1+1↑ρTAB
[35]    LMAX←⌈1.25×⌈/TAB[1+⍳RANK;3]
[36]    S←⌊0.5+LMAX÷100
[37]    QMAX←TAB[1+⍳RANK;1]÷REF
[38]    QMAX←⌈1.1×1000000×⌈/QMAX
[39]    LCON←W[3]÷LMAX
[40]    QCON←W[4]÷QMAX
[41]    I←1
[42]    →DOIF RANK≤0 ◊ RESET ◊ →0
[43]    I←I+1
[44]    TITLE(80 90 100 100)(⍕1981+I)1 2 500 0
[45]    QT1 QP1 QA1 LL1 LA1←PFUN2
[46]    (3 1)VPLINE LL1 QT1
[47]    (1 1)VPLINE LL1 QP1
[48]    (6 1)VPMARK LA1 QA1
[49]    TITLE(2 75 10 80)('Output: ')1 2 500 0
[50]    TITLE(2 70 10 75)('Labour: ')1 2 500 0
[51]    TITLE(10 75 20 80)(⍕⌈TAB[I;1]×0.001×TAB[I;8]+TAB[I;9])1 2 500 0
[52]    TITLE(10 70 20 75)(⍕⌈TAB[I;3])1 2 500 0
[53]    PAUSE
[54]    →DOIF I>RANK ◊ RESET ◊ →0
[55]   BAS:I←I+1
[56]    TITLE(80 90 100 100)(⍕1981+I)1 2 500 0
[57]    TITLE(2 75 10 80)('Output: ')1 2 500 0
[58]    TITLE(2 70 10 75)('Labour: ')1 2 500 0
[59]    TITLE(10 75 20 80)(' ' (⍕⌈TAB[I;1]×0.001×TAB[I;8]+TAB[I;9]) ' ')1
2 500 0
[60]    TITLE(10 70 20 75)(' ' (⍕⌈TAB[I;3]) ' ')1 2 500 0
[61]    QT2 QP2 QA2 LL2 LA2←PFUN2
```

```
[62]     (3 2)VPLINE LL1 QT1

[63]     (1 2)VPLINE LL1 QP1

[64]     (6 2)VPMARK LA1 QA1

[65]     (3 1)VPLINE LL2 QT2

[66]     (1 1)VPLINE LL2 QP2

[67]     (6 1)VPMARK LA2 QA2

[68]     PAUSE

[69]     →DOIF I>RANK ◊ TITLE(25 75 75 85)('Final Year')1 2 500 1 ◊ PAUSE ◊
 RESET ◊ →0

[70]     (3 0)VPLINE LL1 QT1

[71]     (1 0)VPLINE LL1 QP1

[72]    ⍝ (6 0)VPMARK LA1 QA1

[73]     QT1←QT2 ◊ QP1←QP2 ◊ QA1←QA2 ◊ LL1←LL2 ◊ LA1←LA2

[74]     →BAS
```

```
***************************************************************************

SHOWREAL

[0]     SHOWREAL ∆F;I;TIT;⎕TRAP;REF;∆M;∆T;S;TAB;QT1;QT2;QP2;QP2;QA1;QA2;LL1
;LL2;LA1;LA2;LCO;QCON;LMAX;QMAX;RANK;W

[1]    ⍝

[2]    ⍝ This function prepares an animated chart of the production functio
n of

[3]    ⍝ a real firm whose data table is created before.

[4]    ⍝ Written by E.Taymaz  April 1990.

[5]    ⍝

[6]     REF←IMPLP⍙REF

[7]     TIT←'Production Function of Firm ' ∆F

[8]     ∆M←⌊⍎∆F ◊ ((∆F='.')/∆F)←'X'

[9]     TAB←'YEARLY⍙RFIRM⍙' ∆F

[10]     TAB←⍎TAB

[11]     REF←(¯1+1↑⍴TAB)⍴REF[∆M]

[12]     CGISTART
```

```
[13]    DEVICE DEVINFO←VOPNWK'DISPLAY'
[14]    ⎕TRAP←0 'E' 'RESET'
[15]    TIT←TIT
[16]    TITLE(0 90 100 100)(TIT)1 2 750 1
[17]    TITLE(0 80 10 87)('Output')1 2 500 0
[18]    TITLE(90 0 100 7)('Labour')1 2 500 0
[19]    W←0 0 ¯1+DEVINFO[52 53]
[20]    VPLINE(W[1 1 3 3 1])(W[2 4 4 2 2])
[21]    RANK←¯1+1↑⍴TAB
[22]    LMAX←⌈1.25×⌈/TAB[1+⍳RANK;3]
[23]    S←⌊0.5+LMAX÷100
[24]    QMAX←TAB[1+⍳RANK;1]÷REF
[25]    QMAX←⌈1.1×1000000×⌈/QMAX
[26]    LCON←W[3]÷LMAX
[27]    QCON←W[4]÷QMAX
[28]    I←1
[29]    →DOIF RANK≤0 ⋄ RESET ⋄ →0
[30]    I←I+1
[31]    TITLE(80 90 100 100)(⍕TAB[I;17])1 2 500 0
[32]    QT1 QP1 QA1 LL1 LA1←PFUN2
[33]    (3 1)VPLINE LL1 QT1
[34]    (1 1)VPLINE LL1 QP1
[35]    (6 1)VPMARK LA1 QA1
[36]    TITLE(2 75 10 80)('Output: ')1 2 500 0
[37]    TITLE(2 70 10 75)('Labour: ')1 2 500 0
[38]    TITLE(10 75 20 80)(⍕⌈TAB[I;1]×0.001×TAB[I;8]+TAB[I;9])1 2 500 0
[39]    TITLE(10 70 20 75)(⍕⌈TAB[I;3])1 2 500 0
[40]    PAUSE
[41]    →DOIF I>RANK ⋄ RESET ⋄ →0
[42]   BAS:I←I+1
[43]    TITLE(80 90 100 100)(⍕TAB[I;17])1 2 500 0
[44]    TITLE(2 75 10 80)('Output: ')1 2 500 0
[45]    TITLE(2 70 10 75)('Labour: ')1 2 500 0
```

```
[46]     TITLE(10 75 20 80)(' ' (⍕⌈TAB[I;1]×0.001×TAB[I;8]+TAB[I;9]) ' ')1
2 500 0
[47]     TITLE(10 70 20 75)(' ' (⍕⌈TAB[I;3]) ' ')1 2 500 0
[48]     QT2 QP2 QA2 LL2 LA2←PFUN2
[49]     (3 2)VPLINE LL1 QT1
[50]     (1 2)VPLINE LL1 QP1
[51]     (6 2)VPMARK LA1 QA1
[52]     (3 1)VPLINE LL2 QT2
[53]     (1 1)VPLINE LL2 QP2
[54]     (6 1)VPMARK LA2 QA2
[55]     PAUSE
[56]     →DOIF I>RANK ◊ TITLE(25 75 75 85)('Final Year')1 2 500 1 ◊ PAUSE ◊
 RESET ◊ →0
[57]     (3 0)VPLINE LL1 QT1
[58]     (1 0)VPLINE LL1 QP1
[59]    ⍝ (6 0)VPMARK LA1 QA1
[60]     QT1←QT2 ◊ QP1←QP2 ◊ QA1←QA2 ◊ LL1←LL2 ◊ LA1←LA2
[61]     →BAS

*****************************************************************************
STRLEN

[0]     R←STRLEN LABEL
[1]    ⍝ Calculates length of string using current
[2]    ⍝ (proportionally spaced) font
[3]     R←2⊃VQTEXT 10000 10000 LABEL
[4]     R←R[3;1]-R[1;1]

*****************************************************************************
TITLE

[0]     TITLE Z;TEXT;COLOUR;FONT;X;Y;W;BOXED;SIZE;SINK
[1]     W TEXT COLOUR FONT SIZE BOXED←Z
[2]     →DOIF~36⊃DEVINFO ◊ COLOUR BOXED≠←0 ⍝ If mono  set cols to black
```

```
[3]     Wρ⍨←2 2
[4]     W×←0.01×2 2ρ¯1+DEVINFO[52 53]
[5]     X Y←W[1;]+0.5×-⌿W[2 1;]
[6]    ⍝ 1 0 VBAR 2 2ρW                  ⍝ Ensure blank background
[7]     VSTFNT FONT ◊ SINK←VSTHGT SIZE
[8]     FONT COLOUR 0 1 1 VGTEXT X Y TEXT
[9]     →(0=BOXED)/Exit
[10]     1 BOXED VPLINE RECTANGLE 2 2ρW
[11]    Exit:SINK←VSTALN 0 0
```

```
****************************************************************************
VIA

[0]     VIA ∆∆N;∆∆R;∆∆I;⎕IO;⎕TRAP;∆∆D
[1]    ⍝Invoke via external function editor on named fn or fns
[2]     ⎕IO←1 ◊ ⎕TRAP←( 239)'C' '→∆∆L4'
[3]     ∆∆D← '∇'
[4]     →(1≠≡∆∆N← ∆∆N)ρ∆∆L1
[5]     ∆∆N← ⊂∆∆N
[6]    ∆∆L1:∆∆R←⎕NR¨∆∆N
[7]    ∆∆L2:∆∆R←1↓⊃ /(⊂⊂∆∆D) ¨∆∆R
[8]     →(0≠⎕NC'via')ρ∆∆L3
[9]     'via'⎕SH'via'
[10]    ∆∆L3:∆∆R←via ∆∆R
[11]     ∆∆R←¯1⌽∆∆R (∆∆D≠1ρ(ρ∆∆R)⊃∆∆R)/⊂∆∆D
[12]     ∆∆R←1↓¨(( ↑1ρ¨∆∆R)=∆∆D)⊂∆∆R
[13]     ∆∆I←∆∆I/⍳ρ∆∆I←' '≠ ↑1ρ¨0ρ¨⎕FX¨∆∆R
[14]     →(0=ρ∆∆I)ρ0
[15]     'Unable to fix function' ((1<ρ∆∆I)/'s') ' ' (⍕∆∆I) ' Type Q to qui
t'
[16]     →('Q'=1↑⍞)ρ0
[17]     ∆∆R←∆∆R[∆∆I]
[18]     →∆∆L2
[19]    ∆∆L4:'Quit'
```

```
*****************************************************************************

WAIT

[0]     {ERR}WAIT MSG;X;ΔCX;ΔCY;L;N;M;RECT;SINK;SAVE;SCREEN;SIZE
[1]    ⍝ Displays a message and optional error
[2]    ⍝ Waits for user to press a key
[3]     VSTFNT 1 ◊ ΔCX ΔCY←⌈⌿2 2ρVSTHGT 400
[4]     →(2=⎕NC'ERR')/L1
[5]     ERR←''
[6]    L1:L←1+(ρMSG)⌈ρERR
[7]     N←1000+ΔCX×L                 ⍝ Calculate rectangle which will
[8]     M←1000+ΔCY×2 4[1+0<ρERR]             ⍝ be overwritten ...
[9]     RECT←1000 1000 N M
[10]     SCREEN SIZE←VQDBIT
[11]     SAVE←VCCBIT SIZE                     ⍝ Create off-screen bitmap
[12]     VSDBIT SAVE                          ⍝ select it
[13]     VCPBIT SCREEN RECT(RECT[1 2])        ⍝ and copy to it
[14]     VSDBIT SCREEN
[15]     1 0 VBAR RECT ◊ 0 5 VBAR RECT
[16]     1(1 3)VGTEXT(1000+ΔCX÷2)(1000+ΔCY×2 0.5)(ERR MSG)
[17]     SINK←VRQSTR 1
[18]     VCPBIT SAVE RECT(RECT[1 2])        ⍝ Restore display


*****************************************************************************

WINC

[0]     W←WINC PRS;KEY;PR;TY
[1]     KEY PR TY←PRS
[2]     →(KEY≠0)/JMP
[3]     →DOIF TY='C' ◊ W←0 10 100 90 ◊ →0
[4]     →DOIF TY='L' ◊ W←0 0 100 10 ◊ →0
[5]     →DOIF TY='T' ◊ W←0 90 100 100 ◊ →0
[6]    JMP:→('R'=1↑PR)/JRR
```

```
[7]      →DOIF TY='C' ◊ W←0 10 95 85 ◊ →0
[8]      →DOIF TY='L' ◊ W←0 0 95 10 ◊ →0
[9]      →DOIF TY='T' ◊ W←0 85 95 95 ◊ →0
[10]     JRR:
[11]      →DOIF TY='C' ◊ W←10 30 95 90 ◊ →0
[12]      →DOIF TY='L' ◊ W←10 20 95 30 ◊ →0
[13]      →DOIF TY='T' ◊ W←10 90 95 95 ◊ →0
```

****

WINPP

```
[0]      W←WINPP PRS;KEY;PR;TY
[1]      KEY PR TY←PRS
[2]      →(KEY≠0)/JMP
[3]      →DOIF TY='C' ◊ W←0 10 100 80 ◊ →0
[4]      →DOIF TY='L' ◊ W←0 0 100 10 ◊ →0
[5]      →DOIF TY='T' ◊ W←0 90 100 100 ◊ →0
[6]      →DOIF TY='B' ◊ W←0 75 100 90 ◊ →0
[7]     JMP:→('R'=1↑PR)/JRR
[8]      →DOIF TY='C' ◊ W←0 10 95 75 ◊ →0
[9]      →DOIF TY='L' ◊ W←0 0 95 10 ◊ →0
[10]      →DOIF TY='T' ◊ W←0 85 95 95 ◊ →0
[11]      →DOIF TY='B' ◊ W←0 70 95 85 ◊ →0
[12]     JRR:
[13]      →DOIF TY='C' ◊ W←10 30 95 75 ◊ →0
[14]      →DOIF TY='L' ◊ W←10 20 95 30 ◊ →0
[15]      →DOIF TY='T' ◊ W←10 85 95 95 ◊ →0
[16]      →DOIF TY='B' ◊ W←10 70 95 85 ◊ →0
```

****

J. MOSES HELP FUNCTIONS

BACK TREE
CLEAR BLANK
CORR
DOIF
F ID
F PL
FIND INT
FNGREP
FOR TREE
IN FUNCTIONS
IN VARIABLES
KEEP NAMES
LIST
MORE
NAME USED
PFUN
PRINT ALL F
PRT
PRTOFF
PRTON
PRTON 5202
PRTON EPSON
PRTON HPLJ
PRTON PROPRINTER
PRTON STD
PRTON TOSHIBA
REG
SALTER
SET MONITOR
VIA XUSTART
Y RFIRM

```
***************************************************************************

BACKΔTREE

[0]     ΔTREE←BACKΔTREE ΔFNAME
[1]    ⍝
[2]    ⍝  This function finds out all functions that call ΔFNAME.
[3]    ⍝   Written by E.Taymaz  Dec.1989
[4]    ⍝
[5]    ⍝
[6]     ΔTREE←NAMEΔUSED ΔFNAME
[7]    BAS:ΔFNAME←NAMEΔUSED ΔFNAME
[8]     ΔFNAME←(~(ΔFNAME∊ΔTREE))/ΔFNAME
[9]     →(0=⍴ΔFNAME)/END
[10]     →BAS
[11]    END:

***************************************************************************

CLEARΔBLANK

[0]     ΔΔΔNEW←CLEARΔBLANK ΔΔΔOLD;ΔI;ΔRANK
[1]    ⍝
[2]    ⍝  This function deletes all blanks from a vector of names.
[3]    ⍝  Written by E.Taymaz  Dec.1989
[4]    ⍝
[5]     ΔRANK←⍴ΔΔΔOLD
[6]     ΔΔΔNEW←ΔRANK⍴''
[7]     ΔI←0
[8]    BEG:ΔI←ΔI+1
[9]     →(ΔI>ΔRANK)/END
[10]     ΔΔΔNEW[ΔI]←⊂((~((⊃ΔΔΔOLD[ΔI])=' '))/(⊃ΔΔΔOLD[ΔI]))
[11]     →BEG
[12]    END:
```

```
****************************************************************************

CORR

[0]     C←X CORR Y

[1]     X←X-(+/X)÷ρX

[2]     Y←Y-(+/Y)÷ρY

[3]     C←2 2ρ'Corr coef:' ' ' ' 't-value  :' ' ' '

[4]     C[1;2]←(+/X×Y)÷((+/X×X)×(+/Y×Y))*0.5

[5]     C[2;2]←((¯2+ρX)×(C[1;2]*2)÷1-C[1;2]*2)*0.5


****************************************************************************

DOIF

[0]     R←DOIF TEST

[1]    ⍝Return next line number if TEST false

[2]     R←(~TEST)/1+1↓2↑⎕LC


****************************************************************************

F∆ID

[0]     ∆ID←F∆ID ∆N;IN∆LEFT;∆M;∆F

[1]     IN∆LEFT←(+\LEFT=1)⍳∆N

[2]     ∆M←ORIGMARKET[IN∆LEFT]

[3]     ∆F←(+\ORIGMARKET=∆M)[IN∆LEFT]

[4]     ∆ID←(⍕∆M) '.' ¯2↑'0' ⍕∆F


****************************************************************************

F∆PL

[0]     ∆PL←F∆PL ∆N;∆M;∆I;IN∆LEFT

[1]     ∆M←⌊∆N

[2]     ∆I←100×(∆N-⌊∆N)

[3]     IN∆LEFT←(+\ORIGMARKET=∆M)⍳∆I

[4]     ⍎(LEFT[IN∆LEFT]=0)/'⎕ARBOUT 7 ◊''***** THIS FIRM HAS BEEN NULLIFIED
```

```
 *****'' ◊ →0'

[5]      ∆PL←+/LEFT[⍳IN∆LEFT]


****************************************************************************

FIND∆INT

[0]      FIND∆INT;AA;I

[1]      AA←⎕NL 3

[2]      I←0

[3]     BEG:I←I+1

[4]      →(I>1↑⍴AA)/END

[5]      BB←(~(AA[I;]=' '))/AA[I;]

[6]      →((⊂BB)∊∆LIS)/BEG

[7]      AB←BACK∆TREE BB

[8]      ⍎((⊂'INTERACT')∊AB)/'∆LIS←∆LIS ⊂BB'

[9]      →BEG

[10]     END:


****************************************************************************

FNGREP

[0]      {F}FNGREP V;I;M;C;N;X;Z

[1]     ⍝ Report matches of regular expression V in fns F

[2]      XUSTART

[3]      →S↑⍨0≠⎕NC'F' ◊ F←⎕NL 3

[4]     S:N←1↑⍴F←(¯2↑1 1 ⍴F)⍴F

[5]      I←0

[6]     L:→0↑⍨N<I←I+1

[7]      Z←⍴C←⎕CR F[I;] ◊ →L↑⍨1=⍴Z ◊ →L↑⍨0=⍴M←∪1+(Z⊤ss( C)V)[1;]

[8]      X←(X≠' ')/X←F[I;] '['

[9]      (((⍴M) ⍴X)⍴X) (⍕((⍴M) 1)⍴M-1) ']' ' ' C[M;]

[10]      →L


****************************************************************************
```

```
FORΔTREE

[0]     ΔTREE←FORΔTREE ΔFNAME;ΔLIST;ΔΔI

[1]    ⍝

[2]    ⍝  This function finds out all functions called by ΔFNAME.

[3]    ⍝   Written by E.Taymaz  Dec.1989

[4]    ⍝

[5]    ⍝

[6]     ΔTREE←⊂ΔFNAME

[7]     ΔLIST←INΔFUNCTIONS ΔFNAME

[8]     ΔΔI←0

[9]    BEG:ΔΔI←ΔΔI+1

[10]     →(ΔΔI>⍴ΔLIST)/END

[11]     ⍎(~(ΔLIST[ΔΔI]∊ΔTREE))/'ΔTREE←ΔTREE ΔLIST[ΔΔI] ◊ ΔLIST←ΔLIST (INΔF
UNCTIONS ⊃ΔLIST[ΔΔI])'

[12]     →BEG

[13]    END:

***************************************************************************

INΔFUNCTIONS

[0]     ΔΔΔFUNS←INΔFUNCTIONS FNAME

[1]    ⍝

[2]    ⍝  This function finds out the names of functions used in a function
.

[3]    ⍝  Written by E.Taymaz  Dec.1989

[4]    ⍝

[5]     ΔΔΔFUNS←(↑(⎕NC¨↓(⎕REFS FNAME))=3)[;1]/[1](↓⎕REFS FNAME)

[6]     ΔΔΔFUNS←CLEARΔBLANK ΔΔΔFUNS

***************************************************************************

INΔVARIABLES

[0]     ΔΔΔVARS←INΔVARIABLES VNAME
```

```
[1]     ⍝
[2]     ⍝  This function finds out the names of variables used
[3]     ⍝  in a function.
[4]     ⍝  Written by E.Taymaz  Dec. 1989
[5]     ⍝
[6]      ΔΔΔVARS←(↑(⎕NC¨↓(⎕REFS VNAME))=2)[;1]/[1](↓⎕REFS VNAME)
[7]      ΔΔΔVARS←CLEARΔBLANK ΔΔΔVARS
```

```
****************************************************************************
```

KEEPΔNAMES

```
[0]      KEEPΔNAMES ΔNAMES;ΔEXNAMES
[1]     ⍝
[2]     ⍝  This function deletes all functions and variables in the current
[3]     ⍝  workspace other than those given in ΔNAMES as a vector of names.
[4]     ⍝  Written by E.Taymaz  Dec.1989
[5]     ⍝
[6]      ΔEXNAMES←CLEARΔBLANK↓⎕NL 2 3
[7]      ΔNAMES←CLEARΔBLANK ΔNAMES
[8]      ΔEXNAMES←↑(~(ΔEXNAMES∊ΔNAMES))/ΔEXNAMES
[9]      ⎕EX ΔEXNAMES
```

```
****************************************************************************
```

LIST

```
[0]      LIST N;I;R
[1]      I←0
[2]      R←3+¯1↑⍴N
[3]      'VARIABLE NAME' ((R-13)⍴' ') 'RANK'
[4]     BEG:I←I+1
[5]      →DOIF 2≠⎕NC N[I;] ◊ (⍕N[I;]) '   IS NOT DEFINED.'
[6]      →DOIF 2=⎕NC N[I;] ◊ (⍕N[I;]) ((R-⍴⍕N[I;])⍴' ') ': ' ⍕¯2↑'S ' ⍴⍎N[I;
]
[7]      →DOIF 0=22⊤I ◊ 'PRESS ENTER KEY TO CONTINUE' ◊ ⍞
```

```
[8]      →(I<1↑ρN)/BEG
```

****

MORE

```
[0]      {Z}←{PL}MORE V;R;C                  ⍝ Paginate output of matrix.

[1]      ⍎(0=⎕NC'PL')/'PL←23'                ⍝ Default lines-per-page.

[2]      V←⎕FMT Z←V                          ⍝ Convert to char matrix  assign

rslt.

[3]      C←⊃⌽ρV←⎕FMT↑((⊃⌽ρV)ρ⎕PW↑1)⊂V         ⍝ And chop up into ⎕PW width chun

ks.

[4]     LP:→0ρ⍨0=R←⊃ρV                       ⍝ While rows remain to do ...

[5]      PL⌊←R                               ⍝ This page is up to PL rows.

[6]      ⎕←PL C↑V                            ⍝ Display them.

[7]      R←1 ⎕ARBIN'More' 4ρ⊃⎕TC             ⍝ Prompt and read user response

[8]      ⎕ARBOUT'     ' 4ρ⊃⎕TC               ⍝ Erase the prompt.

[9]      V↓⍨←PL 0                            ⍝ Adjust the data.

[10]      →(0ρ⍨R∊81 113) LP                   ⍝ End loop.
```

****

NAME∆USED

```
[0]      ∆LIST←{F}NAME∆USED ∆V;∆I;V;I;M;C;N;X;Z

[1]     ⍝

[2]     ⍝  This function finds out those functions that call the string "∆V"

[3]     ⍝  in a set of functions  "F".(If "F" is not defined  all functions

[4]     ⍝  in the current workspace are searched. "∆V" can be a name vector.

)

[5]     ⍝  Written by E.Taymaz  Jan.1990

[6]     ⍝

[7]      ∆LIST←''

[8]      XUSTART

[9]      →S↑⍨0≠⎕NC'F' ⋄ F←⎕NL 3

[10]     S:∆I←0
```

```
[11]      ⍎(1=(≡ΔV))/'ΔV←1ρ⊂ΔV'

[12]     SIR:ΔI←ΔI+1

[13]      →(ΔI>ρΔV)/END

[14]      V←⊃ΔV[ΔI]

[15]      N←1↑ρF←(¯2↑1 1 ρF)ρF

[16]      I←0

[17]     L:→SIR↑⍨N<I←I+1

[18]      Z←ρC←⎕CR F[I;] ◊ →L↑⍨1=ρZ ◊ →L↑⍨0=ρM←∪1+(Z⊤ss( C)V)[1;]

[19]      ⍎((~(⊂F[I;])∊ΔLIST))/'ΔLIST←ΔLIST ⊂F[I;]'

[20]      →L

[21]     END:ΔLIST←CLEAR⍙BLANK ΔLIST
```

```
*****************************************************************************

PFUN

[0]      PARS←ΔPRS PFUN ΔF;R̲E̲F̲;ΔM;ΔT;ΔS;S̲;TAB;Q̲T̲O̲P̲;T̲E̲C̲;R̲E̲S̲;L̲;L̲L̲;Q̲1;Q̲2;Q̲3

[1]      ΔT ΔS←ΔPRS

[2]      R̲E̲F̲←IMPLP⍙R̲E̲F̲ ◊ ΔT←ΔT+1

[3]      →(0 1 2 3 4∊⍎ΔF)/IND SEC SEC SEC SEC

[4]      ΔM←⌊⍎ΔF ◊ ((ΔF='.')/ΔF)←'X̲'

[5]      TAB←'YEARLY⍙FIRM⍙' ΔF ◊ S̲←25 ◊ →JMP

[6]     IND:TAB←'YEARLYΔINDUSTRYΔTOTAL' ◊ S̲←1000 ◊ ΔM←1

[7]      R̲E̲F̲[1]←((YEARLY⍙MARKET1[ΔT;1]×R̲E̲F̲[1])+(YEARLY⍙MARKET2[ΔT;1]×R̲E̲F̲[2])
+(YEARLY⍙MARKET3[ΔT;1]×R̲E̲F̲[3])+(YEARLY⍙MARKET4[ΔT;1]×R̲E̲F̲[4]))÷(⍎TAB)[ΔT;1]
◊ →JMP

[8]     SEC:TAB←'YEARLY⍙MARKET' ΔF ◊ S̲←1000 ◊ ΔM←⍎ΔF

[9]     JMP:

[10]      TAB←⍎TAB

[11]      →DOIF ΔT>1↑ρTAB ◊ STOP⍙HERE'ERROR IN TIME PERIOD' ◊ →0

[12]      Q̲T̲O̲P̲←1000000×TAB[ΔT;1]÷ΔS×R̲E̲F̲[ΔM] ◊ T̲E̲C̲←TAB[ΔT;2] ◊ R̲E̲S̲←TAB[ΔT;9]

[13]      L̲L̲←S̲×0 ⍳1000

[14]      L̲←S̲×⌊TAB[ΔT;3]÷S̲

[15]      →DOIF L̲>¯1↑L̲L̲ ◊ L̲L̲←S̲×0 ⍳⌈1.3×L̲÷S̲

[16]      L̲
```

```
[17]     Q1←WTIX×QTOP×1-*-LL×TEC÷QTOP

[18]     Q2←Q1×(1-RES÷100)

[19]     Q3←(ρLL)ρ0

[20]     Q3[LLιL]←QTOP×(1-(RES+TAB[ΔT;8])÷100)

[21]     PARS←0.0001×Q1 Q2 Q3 LL
```

****

PRINTΔALLΔF

```
[0]      {ΔLIST}PRINTΔALLΔF ΔPP;⎕PW;ΔI;ΔJ;AA;ΔF

[1]     ⍝

[2]     ⍝ This function prints all functions whose names are defined in ΔLIS

T.

[3]     ⍝ If ΔLIST is not defined  all functions in the active workspace wil

l

[4]     ⍝ be printed. Written by E.Taymaz  Feb.1990.

[5]     ⍝

[6]     ⍝

[7]      ⎕PW←ΔPP

[8]      →S↑⍨0≠⎕NC'ΔLIST' ◊ ΔLIST←⎕NL 3

[9]     S:ΔLIST←(¯2↑1 1 ρΔLIST)ρΔLIST

[10]      ⎕CY'PRT'

[11]      ΔI←0

[12]      PRTONΔHPLJ

[13]      arbout 12

[14]      PRT' ' ◊ PRT 88ρ'*' ◊ PRT' ' ◊ PRT' '

[15]      PRT'WORKSPACE  ' ⎕WSID

[16]      PRT' ' ◊ PRT' ' ◊ PRT 88ρ'*' ◊ PRT' ' ◊ PRT' '

[17]     BEG:ΔI←ΔI+1

[18]      ΔJ←0

[19]      ΔF←⎕CR ΔLIST[ΔI;]

[20]      →(0=1↑ρΔF)/BEG

[21]      ΔLIST[ΔI;] ' is being printed.'
```

```
[22]      PRT 88ρ'*'
[23]      PRT' '
[24]      PRT ⎕FMT ∆LIST[∆I;]
[25]      PRT' '
[26]      PRT' '
[27]    BE:∆J←∆J+1
[28]      AA←∆F[∆J;]
[29]      AA←⊖((~∧\' '=⊖AA)/⊖AA)
[30]      PRT'[' (⍕∆J-1) ']     ' AA
[31]      →(∆J<1↑ρ∆F)/BE
[32]      PRT' '
[33]      PRT' '
[34]      →(∆I<1↑ρ∆LIST)/BEG
[35]      PRT 88ρ'*'
[36]      PRTOFF
```

****

PRT

```
[0]     PRT DATA
[1]    ⍝ Format DATA and pass to prt
[2]     prt ⎕FMT DATA
```

****

PRTOFF

```
[0]     PRTOFF
[1]    ⍝ Terminate printer output
[2]     ⎕EX¨'prt' 'arbout'
```

****

PRTON

```
[0]     PRTON ARGS;CMD;TRANSLATION;FONT;DESTINATION
[1]    ⍝ General setup function  called by specific printer setup functions
```

```
[2]      →(0≠⎕NC'prt')/0                         ⍝ Exit if already starte
d
[3]      TRANSLATION FONT DESTINATION←ARGS       ⍝ Split ARGS
[4]      CMD←''                                  ⍝ Construct shell comman
d
[5]      →(0=⍴ TRANSLATION)/L1                   ⍝ Translation required?
[6]      CMD←'APLT1=' TRANSLATION '; export APLT1;' ⍝ Select translation tab
le
[7]     L1:CMD ←' trap "" 1 2 3; { '             ⍝ Start command list
[8]      →(0=⍴ FONT)/L2                          ⍝ Font required?
[9]      CMD ←' cat $DYALOG/fonts/' FONT ';'     ⍝ Select font file
[10]     L2:CMD ←' exec $DYALOG/xflib/prt;'       ⍝ Invoke prt AP
[11]      CMD ←' } '                              ⍝ End command list
[12]      CMD ←' ¦ '                              ⍝ Redirect output ...
[13]      CMD ←'exec lpr '                        ⍝ ... to the print spoo
ler
[14]      →(0=⍴ DESTINATION)/L3                   ⍝ Destination printer?
[15]      CMD ←'-d' DESTINATION                   ⍝ Select printer
[16]     L3:'/bin/sh'⎕SH'sh' '-c'CMD             ⍝ Run command via the sh
ell
```

```
*****************************************************************************
PRTON∆5202

[0]      PRTON∆5202
[1]     ⍝ Setup function for IBM Quietwriter III with APL font
[2]      TRANSLATION←'ibm5202'
[3]      FONT←''
[4]      DESTINATION←''
[5]      PRTON TRANSLATION FONT DESTINATION

*****************************************************************************
PRTON∆EPSON
```

```
[0]      PRTON∆EPSON;TRANSLATION;FONT;DESTINATION
[1]     ⍝ Setup function for Epson FX printers
[2]      TRANSLATION←'epsonFX'
[3]      FONT←'epsonFX'
[4]      DESTINATION←''
[5]      PRTON TRANSLATION FONT DESTINATION
```

****

PRTON∆HPLJ

```
[0]      PRTON∆HPLJ;TRANSLATION;FONT;DESTINATION
[1]     ⍝ Setup function for Laserjet plus
[2]      TRANSLATION←'HPLJplus'
[3]      FONT←'HPLJplus'
[4]      DESTINATION←''
[5]      PRTON TRANSLATION FONT DESTINATION
```

****

PRTON∆PROPRINTER

```
[0]      PRTON∆PROPRINTER;TRANSLATION;FONT;DESTINATION
[1]     ⍝ Setup function for IBM Proprinter
[2]      TRANSLATION←'proprinter'
[3]      FONT←'proprinter'
[4]      DESTINATION←''
[5]      PRTON TRANSLATION FONT DESTINATION
```

****

PRTON∆STD

```
[0]      PRTON∆STD;TRANSLATION;FONT;DESTINATION
[1]     ⍝ Setup function for standard printer
[2]      TRANSLATION←'ascii_apl'
[3]      FONT←''
```

```
[4]     DESTINATION←''

[5]     PRTON TRANSLATION FONT DESTINATION
```

****

PRTONΔTOSHIBA

```
[0]     PRTONΔTOSHIBA;TRANSLATION;FONT;DESTINATION

[1]    ⍝ Setup function for Toshiba 351P with APL font cartridge

[2]     TRANSLATION←'toshiba351P'

[3]     FONT←''

[4]     DESTINATION←''

[5]     PRTON TRANSLATION FONT DESTINATION
```

****

REG

```
[0]     Y REG X;ΔDV;ΔEV;ΔC;ΔB;ΔS;ΔT

[1]     ΔDV←Y

[2]     →DOIF 1=≡X ◊ X←⊂X

[3]     ΔEV←(⊂'Constant') X

[4]     Y←⍎Y ◊ X←⍎¨X

[5]     X←⍉↑X

[6]     X←(2↑(⍴X) 1)⍴X

[7]     X←((1↑⍴X 1)⍴1) [2]X

[8]     ΔC←⌹((⍉X)+.×X)

[9]     ΔB←ΔC+.×(⍉X)+.×Y

[10]     ΔS←(⍉Y-X+.×ΔB)+.×(Y-X+.×ΔB)÷((⍴Y)-¯1↑⍴X)

[11]     ΔT←ΔB÷(ΔS×ΔC[2\¨⍳¯1↑⍴X])*0.5

[12]     40⍴'*'

[13]     ' Regression results'

[14]     ' Dependent variable : ' ⍕ΔDV

[15]     ' R-square           : ' 6↑⍕1-((⍉Y-X+.×ΔB)+.×(Y-X+.×ΔB))÷((⍉Y)+.×Y
)-(⍴Y)×(+/Y÷⍴Y)*2

[16]     ''
```

```
[17]     ⍉(3 1+ρΔEV)ρ(⊂'Variable') ΔEV (⊂'Coefficient') ΔB (⊂'t-statistic')
 ΔT
[18]     ''
[19]     40ρ'*'
```

****

SALTER

```
[0]      ΔC←ΔΔA SALTER ΔΔB
[1]     ⍝
[2]     ⍝ This function prepares variables to draw salter curves.
[3]     ⍝ ΔΔA and ΔΔB will be on the Y
[4]     ⍝  and X-axes  respectively.
[5]     ⍝ Written by E.Taymaz  March 1990
[6]     ⍝
[7]      ⍎((ρΔΔA)≢(ρΔΔB))/'''RANKS OF VARIABLES SHOULD BE EQUAL'' ◊ ⎕ARBOUT
7 ◊ →0 '
[8]      ΔC←3ρ''
[9]      ΔΔB←ΔΔB[⍒ΔΔA] ◊ ΔΔA←ΔΔA[⍒ΔΔA]
[10]      ΔΔB←+\ΔΔB ◊ ΔΔB←100×ΔΔB÷⌈/ΔΔB
[11]      ΔR←(ρΔΔA)ρ2 ◊ ΔΔA←ΔR\ΔΔA ◊ ΔΔB←ΔR\ΔΔB
[12]      ΔΔA←ΔΔA 0 ◊ ΔΔB←0 ΔΔB
[13]      ΔC←(ΔΔA)(ΔΔB)((ρΔΔA)ρ0)
```

****

SETΔMONITOR

```
[0]      SETΔMONITOR;ΔI;AA
[1]      AA←⎕NL 3
[2]      ΔI←0
[3]     BEG:ΔI←ΔI+1
[4]      →(ΔI>1↑ρAA)/END
[5]      0 ⎕MONITOR AA[ΔI;]
[6]      →BEG
```

```
[7]     END:
```

****

VIA

```
[0]      VIA ΔΔN;ΔΔR;ΔΔI;⎕IO;⎕TRAP;ΔΔD
[1]     ⍝Invoke via external function editor on named fn or fns
[2]      ⎕IO←1 ◊ ⎕TRAP←( 239)'C' '→ΔΔL4'
[3]      ΔΔD← '∇'
[4]      →(1≢≡ΔΔN← ΔΔN)ρΔΔL1
[5]      ΔΔN← ⊂ΔΔN
[6]     ΔΔL1:ΔΔR←⎕NR¨ΔΔN
[7]     ΔΔL2:ΔΔR←1↓⊃ /(⊂⊂ΔΔD) ¨ΔΔR
[8]      →(0≠⎕NC'via')ρΔΔL3
[9]      'via'⎕SH'via'
[10]     ΔΔL3:ΔΔR←via ΔΔR
[11]      ΔΔR←¯1⌽ΔΔR (ΔΔD≠1ρ(ρΔΔR)⊃ΔΔR)/⊂ΔΔD
[12]      ΔΔR←1↓¨(( ↑1ρ¨ΔΔR)=ΔΔD)⊂ΔΔR
[13]      ΔΔI←ΔΔI/⍳ρΔΔI←' '≠ ↑1ρ¨0ρ¨⎕FX¨ΔΔR
[14]      →(0=ρΔΔI)ρ0
[15]      'Unable to fix function' ((1<ρΔΔI)/'s') ' ' (⍕ΔΔI) ' Type Q to qui
t'
[16]      →('Q'=1↑⍞)ρ0
[17]      ΔΔR←ΔΔR[ΔΔI]
[18]      →ΔΔL2
[19]     ΔΔL4:'Quit'
```

****

XUSTART

```
[0]      XUSTART
[1]     ⍝Start up the "xutils" auxilary processor if not already going
[2]      →(0≠⎕NC'ss')ρ0
[3]      'xutils'⎕SH'xutils'
```

```
*****************************************************************************

YΔRFIRM

[0]     SCALE YΔRFIRM ΔF;MISS;ΔP;I̲;M;ΔT;RANK;L̲;TIME;A̲21;A̲22;A̲23;S̲U̲M̲;S̲T̲O̲;D̲S̲;

D̲P̲;D̲W̲;I̲N̲V̲;D̲Q̲;P̲;S̲;Q̲S̲;Q̲P̲;ΔK3F;Q̲Q̲;Q̲;V̲A̲;Q̲T̲O̲P̲;T̲E̲C̲;P̲R̲O̲D̲;M̲;M̲S̲

[1]     ⍝

[2]     ⍝ This function prepares a data table for a real firm. This table is

 comparable to those created by the YΔRΔFIRM function during the simulation

.

[3]     ⍝ Written by E.Taymaz  April 1990.

[4]     ⍝

[5]      M←⌊⍎ΔF

[6]      ΔP←(+\MARKET=M)⍳⍎2↓ΔF

[7]      →DOIF ΔP≥155 ◊ ⎕ARBOUT 7 ◊ ΔF ' IS NOT A REAL FIRM' ◊ →0

[8]      →DOIF 0=⎕NC'LISTC' ◊ ⎕CY'MICRO.DBASE'

[9]      →DOIF 0=LISTC[ΔP] ◊ ⎕ARBOUT 7 ◊ 'NO DATA ARE AVAILABLE FOR THE FIRM

 ' ΔF ◊ →0

[10]      I̲←⍉(LISTC[ΔP]=TABLE[;1])⌿TABLE

[11]      ((ΔF='.')/ΔF)←'⍙'

[12]      ΔT←1 17⍴'QTOP' 'TEC' 'L' 'PROD' 'DQ' 'A21' 'A22' 'SUM' 'A23' 'M×S'

 'STO' 'DS' 'DP' 'DW' 'M' 'INV' 'TIME'

[13]      MISS←∨⌿I̲[5 8 9 10 11 12 13 20 26 29 30 31 32 44 45 46 47;]∊¯99999

[14]      →(0=+/MISS)/JMP

[15]      ⎕ARBOUT 7 ◊ 'MISSING VALUES FOR THE FOLLOWING YEAR(S): ' ⍕MISS/I̲[2

;⍳¯1↑⍴I̲]

[16]      M1←MISS⍳0

[17]      →DOIF M1>⍴MISS ◊ ⎕ARBOUT 7 ◊ 'MISSING VALUES FOR ALL YEARS' ◊ →0

[18]      MISS[⍳M1]←0

[19]      M2←MISS⍳1

[20]      I̲←I̲[;¯1+M1+⍳(M2-M1)]

[21]     JMP:

[22]      RANK←1+⍳(¯1+¯1↑⍴I̲)

[23]      L̲←I̲[5;RANK]
```

```
[24]       TIME←I[2;RANK]

[25]       A21←0⌈50⌊I[32;RANK]-I[29;RANK]

[26]       A22←2.5⌈50⌊I[31;RANK]-I[32;RANK]

[27]       SUM←A21+A22

[28]       A23←(ρRANK)ρ10

[29]       STO←100×I[46;RANK]÷I[47;RANK]

[30]       DS←100×(I[7;RANK]÷I[7;RANK-1])-1

[31]       DP←DS-I[29;RANK]

[32]       DW←100×(I[5;RANK-1]×I[13;RANK]÷I[13;RANK-1]×I[5;RANK])-1

[33]       INV←I[26;RANK]

[34]       DQ←DS-DP

[35]       P←100-+/(0≥TIME-1982)/DP

[36]       P←P++\DP

[37]       S←I[7;RANK]

[38]      ⍝

[39]      ⍝ QS←0.25×(0.625×S)+0.375×DS

[40]       QS←S÷4

[41]      ⍝ QP←(0.625×P)+0.375×DP

[42]       QP←P+DP÷4

[43]       ΔK3F←S×0.01×I[46;RANK]-I[47;RANK]

[44]       QQ←(QS+ΔK3F÷4)÷QP

[45]       Q←(S+ΔK3F)÷P

[46]       VA←S+ΔK3F+((+/I[10 11 12;RANK])×0.01×I[44;RANK]-I[45;RANK])-+/I[10
 11 12;RANK]

[47]       QTOP←IMPLPΔREF[M]×(QQ×100+A21+A22)÷100-A23

[48]       TEC←¯1×(⍟A22÷100+A21+A22)×1000000×QTOP÷L×IMPLPΔREF[M]

[49]       PROD←1000000×Q×IMPLPΔREF[M]÷L

[50]       M←100×1-I[13;RANK]÷VA

[51]       MS←M×S

[52]       ΔT←ΔT [1]⍉(17 ρRANK)ρ(SCALE×QTOP) TEC L PROD DQ A21 A22 SUM A23 (S
CALE×MS) STO DS DP DW M (SCALE×INV) TIME

[53]       ⍎'YEARLYΔRFIRMΔ' ΔF '←ΔT'

**********************************************************************
```