A list of Working Papers on the last pages

No. 261, 1990

THE MOSES PC MANUAL

by

Erol Taymaz

This is a preliminary paper.

June, 1990

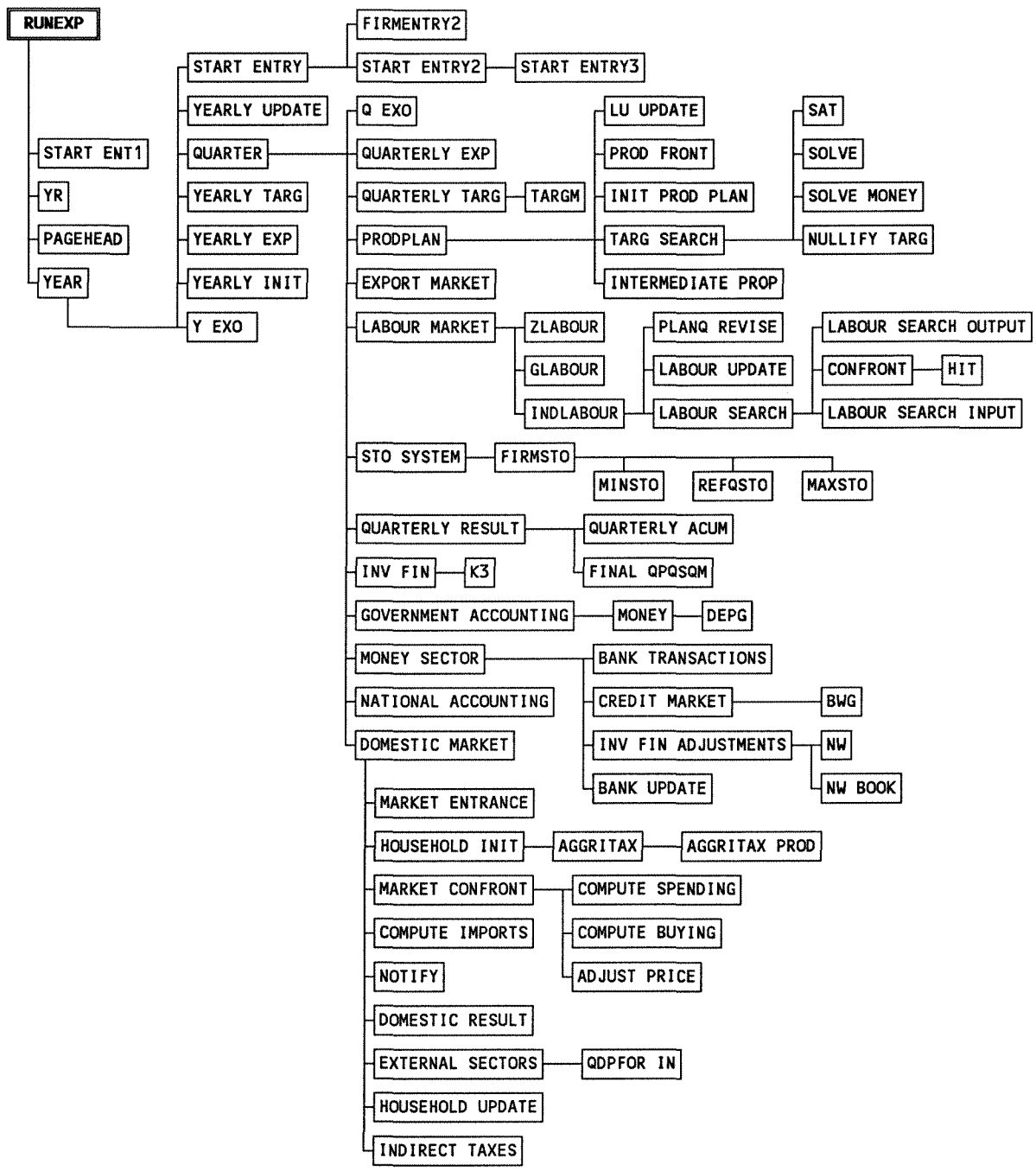
THE MOSES^{PC} MANUAL

Erol Taymaz
IUI
Box 5501
11485 Stockholm

March 26, 1990

CONTENTS

MOSES FUNCTIONS



1. Introduction

MOSES^{PC} is the PC version of MOSES (**M**odel **O**f the **S**wedish **E**conomic **S**ystem) 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 (including the APL interpreter) seems to require less than 2 MByte RAM and 5 MByte hard-disk 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* the model. It is intended for users who have little or no familiarity with the model. Anyone following the instructions in Section 3 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.

This manual does not describe the initialization procedure which is explained in detail in MOSES HANDBOOK (Bergholm, 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 chapter 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 chapter.

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.

The manual is organized as follows. The file system of the model (the names and functions of model files) are summarized in Section 2. Section 3 is a quick start that explain how to run the model. Basic APL commands that are useful in using the MOSES^{PC} model are shown in Section 4. The modification method to prepare simulation experiments are explained in Section 5. Section 6 explains the functions that are used in displaying the model results (printing output tables, graphics, etc.). Section 7 summarizes the procedure to create a new database for the model. The last section presents an overview of the model variants and experiment functions.

2. 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.HELP	: Various useful functions to analyze the model, etc.
MOSES.GRAPH	: Various functions for graphics (charts, bar charts, etc.).
R1990.91	: The "synthetic" dataset for 1990. This dataset is created by simulating the model for 8 years by using the dataset R1982.91, and the modification function MSTART91.
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 first one is the default version.
R1976.5	: The 1976 dataset. There is only one version for this set.
MSTART	: The file for modification functions. Each

function is named MSTART*xx* where *xx* is the modification (experiment) number.

M*xx*R*yyyy.yy.zz* : An output file that contains the results of an experiment. *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 years 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.

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

DATA : A data file that contains various real macroeconomic data.

DATASET.SURVEY : A data file that contains the results of all planning surveys for 1977-1989. This file is to be updated for every year.

The R1976.5, R1982.*xx*, and DATASET.SURVEY workspaces contain confidential firm data. Because of the disclosure rules, the access to those

workspaces is restricted to authorized users.

3. 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.HELP, MOSES.GRAPH, R1990.91, MSTART, and PRT workspaces. Second, the PC in IUI can be accessed by, for example, using modems by another computer which is operating with Xenix operating system.

Now, we assume that you directly use the Toshiba PC at the IUI, and 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 login to 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 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.)

To run the model, type

$ apl moses RETURN

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

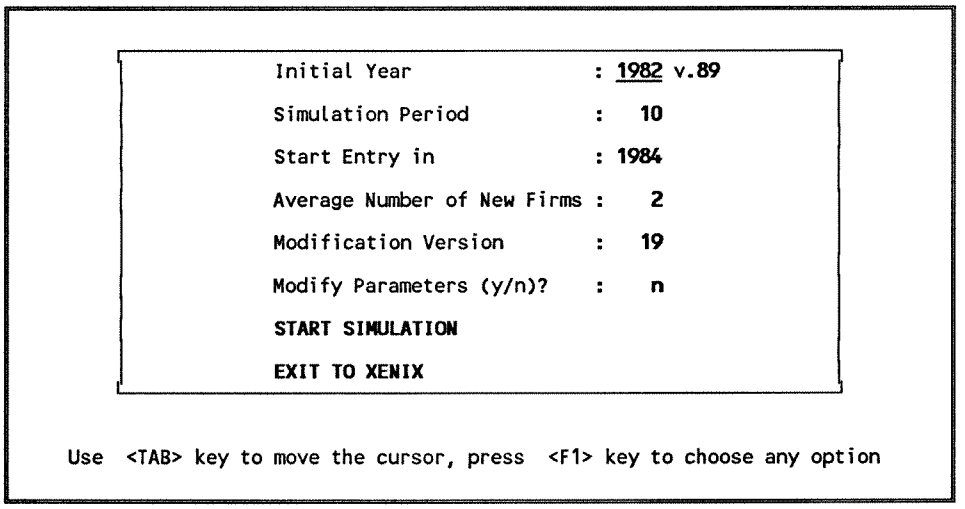
The cursor can be moved by pressing TAB (downward) or Shift-TAB (upward) keys. After changing default values in the menu as desired, the simulation is started by pressing F1 key when the cursor (the highlighted rectangular) is on "START SIMULATION".

---

*) 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.



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

"Simulation Period" is the number of years 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 AMAXENT 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 the average profitability. If you want not to use the firm entry option, enter zero for this variable.

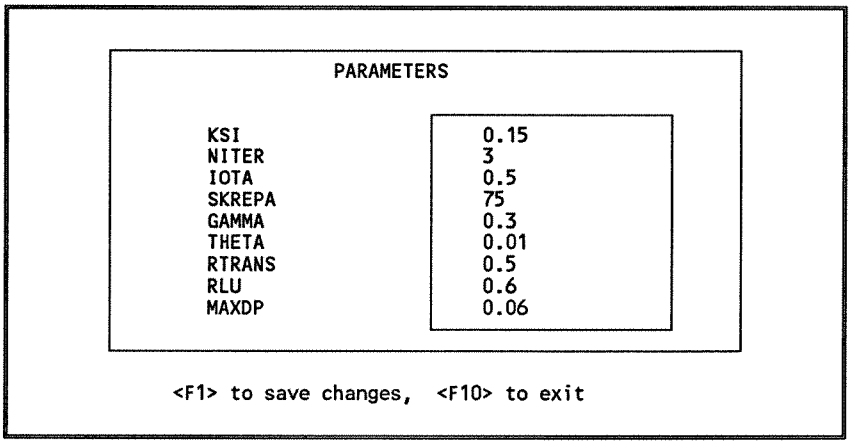
"Modification Number" determines which MSTART function will be used to modify the model. In the MOSES^{PC} model, each experiment is carried out by making changes in the original model (changes in the behavioral equations, variables, etc.) by using one of the modification functions in the MSTART workspace. This allows to keep model in its original form. The MSTART*xx* functions where 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.

"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. The values entered by using this option has the highest priority (i.e., it supercedes 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" let 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.



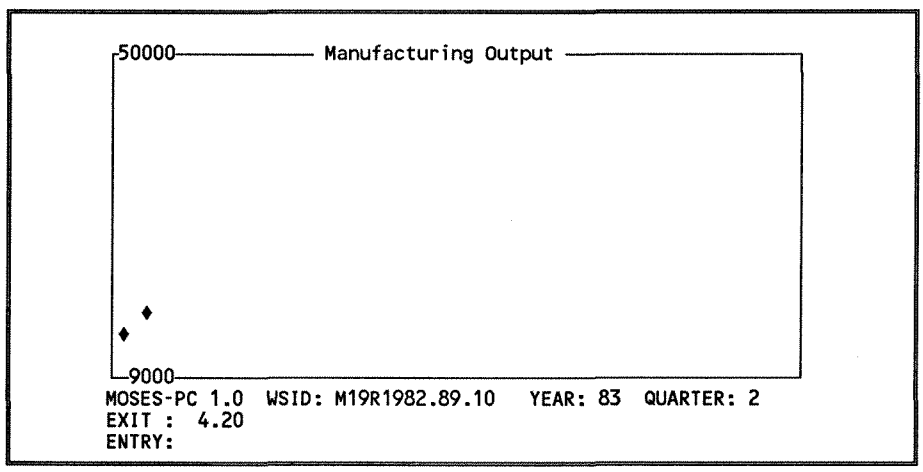
The cursor can be moved by pressing "cursor" keys. Note that all parameter names and values are not shown in this menu. Therefore, the user can scroll the data using CURSOR RIGHT and CURSOR DOWN. The data in both fields (name and variable fields) scrolls accordingly. After changing the parameter values, pressing F1 key changes those values in the model and starts the execution of the experiment. F10 key keeps the parameter values unchanged and start the simulation.

In a usual 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.

There are two alternative types of on-line output during the simulation. In the first case which is the default option, quarterly industrial output is

plotted on a graph during simulation. This option is allowed by the "PREP▲INT▲GRAPH" function used in the "RUNEXP" function. The user can plot any other variable by modifying the command line for this function (see Section 5 for editing the modification functions). In this case, a chart is shown as follows.

In 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 material sector, 12 the intermediate goods, 3 the investment goods and consumer durables, and 4 the consumption goods sector.



In the second case, basic quarterly data for each industry (production plans, actual production, exports, etc.) are printed on the screen. This option is turned on by adding a line "TRACE1" to the modification function, MSTARTxx, to execute the TRACE1 function, and to turn the graphics functions off (see Section 5). When the PRICE▲CHANGES function is called in the MSTARTxx function (i.e., the ADJUST▲PRICES, COMPUTE▲-IMPORTS, DOMESTIC▲RESULTS, and MARKET▲CONFRONT functions are replaced by their modified forms), "TRACE2" should be used instead of the TRACE1 function.(*)

When the simulation is completed, a message is printed on the screen. The results of the experiment can be printed on the screen, and saved in a text file that can be accessed by Xenix and DOS operating systems, and any other DOS-based program (T^3, 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 eight 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

---

*) An interactive graphics that shows the search procedure of a firm for appropriate quarterly production and employment levels can be shown during the simulation. The planned levels of output and employment of the firm can also be changed manually. For details, see Section 8.

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 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

)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

let the user to exit to the operating system.

4. Basic APL functions

A basic knowledge of the APL language is essential for using the MOSES^{PC} 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 2 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. System commands may be entered from immediate execution mode. In this mode, the blinking cursor stays at the eight column from the left on the screen. In other words, the user can enter those commands only when the simulation is completed or stopped by any reason.

Some useful system commands are as follows.

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

)VARS displays the names of global 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 (usually $ 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 the errors when the simulation is unexpectedly halted.

)RESET cancels all suspensions recorded in the state indicator.

[]WSIS 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 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 3 Mb, and an HP plotter (PLOTTER) and an HP laser printer (RINTER) using a 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 for Dyalog APL Üser Guide for details).

5. 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.) only by using the functions in the MSTART workspace. 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 material sector, 2 the intermediate goods, 3 the 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 exactly same as the previous functions. The only difference is 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 to 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. Finally, the modification function may contain commands that change the MOSES functions. For this purpose, the following functions can be used.

'functionname' **MODADD** *'linebeginning⍵newline'*
'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, and *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 should be 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 two few letters are specified in the *linebegining* 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 MODLNP function adds the *newline* into the *functionname* as the line defined by its *linenumber.*

5. During the simulation of the model, a chart that plots a variable defined by the user is shown. It is also possible to print some quarterly data defined by the TRACE1 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 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 with respect to the reference case.

This function is called by the RUNEXP function. If you want to use TRACE1 (or TRACE2) function instead of this default option of showing a chart during the simulation, the line in the RUNEXP function that calls this function should be deleted by using the MODDEL function.

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 causes to store 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 this 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 will cause to stopping simulation after the yearly profit targets are set in the YEARLY▲TARG function. The user can modify and targets "manually" before continuing to the simulation. 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 takes place when the function UPDATEMOSES is called. That is, the MSTART*xx* 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's "User Guide").

6. 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 the matrix form whose first row contains the variable names stored in the table.

Currently, following tables are standard output of the model.

Table Name	**Variables in the table**
YEARLY▲INDUSTRY▲TOTAL	QTOP, TEC, L, PROD, DQ, A21, A22, SUM, A23, M*S, STO, DS, DP, DW, M, INV, LTOT, RU
YEARLY▲GNPFIX▲PROD	TOT, RAW, IMED, INV/DUR, NDUR, A/F/F, ORE, BLD, EL, SERVICE, WSG
YEARLY▲GNPFIX▲USE	TOT, GTOT, WSG, PURCHG, HH, INV-TOT, INV-MKT, INV-BLD, INV-G, CHSTO, EXPORT, IMPORT
YEARLY▲GNPCUR▲PROD	TOT, RAW, IMED, INV/DUR, NDUR, A/F/F, ORE, BLD, EL, SERVICE, WSG
YEARLY▲GNPCUR▲USE	TOT, GTOT, WSG, PURCHG, HH, INV-TOT, INV-MKT, INV-BLD, INV-G, CHSTO, EXPORT, IMPORT
YEARLY▲FINANCE	M*S, INTPAY, DPER, TAXES, DIV, SUBS, CHBW, INV, CHK2, K1, K2, K3-IN, K3-OUT, BW, NW, TOT
YEARLY▲GOVERNMENT	LG, WG, DWG, WSG, PURCHG, TRANS, SUBS, SPG, INVG, ITAX, WTAX, VATAX, CTAX, INCOME, INTPAY, SURPLUS

YEARLY▲BANK▲TRANSACTIONS	INTF, INTK2, INTH, INTG,INTGFOR, CHBW, CHK2, SAVH, CHDEPG, CHDEPGF, EXPORT, FASSPAY, IMPORT, FDPAY, CHNBW
YEARLY▲BANK▲POSITION	BW, K2, HH, G, LIQB, LIQBFOR, FASS, FD, FNASS, NETFOR, NW, GFOR
YEARLY▲COUNTRY▲TOTAL	GNPFIX, GNPCUR, MONEY, VEL, RI, PRINT, CHLIQB, CHINV, CHDIV, CHKIN, REST
YEARLY▲PRICES	QPDOM 1, QPDOM 2, QPDOM 3, QPDOM 4, QPFOR 1, QPFOR 2, QPFOR 3, QPFOR 4, W 1, W 2, W 3, W 4, M 1, M 2, M 3, M 4, RR 1, RR 2, RR 3, RR 4
YEARLY▲FOREIGN▲TRADE	X 1, X 2, X 3, X 4, IMP1 1, IMP1 2, IMP1 3, IMP1 4, IMP2 1, IMP2 2, IMP2 3, IMP2 4
YEARLY▲HOUSEHOLDS	DDI, SP 1, SP 2, SP 3, SP 4, SP 5, SP 6, SP 7, SP 8, SP 9, SP 10, SAVH, DCPI, PURCH, SAVH
YEARLY▲MARKET1	QTOP, TEC, L, PROD, DQ, A21, A22, SUM, A23, M*S, STO, DS, DP, DW, M, INV, QPDOM, QPFOR
YEARLY▲MARKET2	QTOP, TEC, L, PROD, DQ, A21, A22, SUM, A23, M*S, STO, DS, DP, DW, M, INV, QPDOM, QPFOR
YEARLY▲MARKET3	QTOP, TEC, L, PROD, DQ, A21, A22, SUM, A23, M*S, STO, DS, DP, DW, M, INV, QPDOM, QPFOR
YEARLY▲MARKET4	QTOP, TEC, L, PROD, DQ, A21, A22, SUM, A23, M*S, STO, DS, DP, DW, M, INV, QPDOM, QPFOR
YEARLY▲FINANCE1	M*S, INTPAY, DPER, TAXES, DIV, SUBS, CHBW, INV, CHK2, K1, K2, K3-IN, K3-OUT, BW, NW, TOT
YEARLY▲FINANCE2	M*S, INTPAY, DPER, TAXES, DIV, SUBS, CHBW, INV, CHK2, K1, K2, K3-IN, K3-OUT, BW, NW, TOT
YEARLY▲FINANCE3	M*S, INTPAY, DPER, TAXES, DIV, SUBS, CHBW, INV, CHK2, K1, K2, K3-IN, K3-OUT, BW, NW, TOT
YEARLY▲FINANCE4	M*S, INTPAY, DPER, TAXES, DIV, SUBS, CHBW, INV, CHK2, K1, K2, K3-IN, K3-OUT, BW, NW, TOT
ED▲NULLIFIED	
ED▲PARAMETERS	

The form of the last two tables are different than those of others. These tables contain data on the nullified firms, and the simulation parameters, respectively.

The user could add more table to this standard (default) output by calling any one of the following functions by the MSTARTxx function.

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*
Y▲R▲MARKET▲F
Y▲R▲MARKET▲Q
Y▲R▲INDUSTRY▲F
Y▲R▲INDUSTRY▲Q

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*X*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▲3X16

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

The firm-level variables stored in these tables are as follows.

Table name	**Variables in the table**
YEARLY▲FIRM▲*c1X̲c2*	QTOP, TEC, L, PROD, DQ, A21, A22, SUM, A23, M*S, STO, DS, DP, DW, M, INV
YEARLY▲FIRM▲F▲*c1X̲c2*	DIV, INV, NW, DNW, DBW, RR, K1, VA
YEARLY▲FIRM▲Q▲*c1X̲c2*	L, DS, DQ, DP, DW, M, XS, MKTSH
YEARLY▲FIRM▲FINANCE▲*c1X̲c2*	M*S, INTPAY, DEPR, TAXES, DIV, SUBS, CHBW, INV, CHK2, K1, K2, K3-IN, K3-OUT, BW, NW, TOT
FIRMCHARC	YEAR, SCTR, NUM, M, L, VA, QQ, QP, QS, K1, QTOP, TEC, INVEFF, RES, A22

Optional industry- and sector (market) level data storage functions keep track of the following variables. (The Y▲R▲MARKET▲F and Y▲R▲MARKET▲Q functions create tables for each sector.)

Table name	**Variables in the table**
YEARLY▲MARKET▲F▲1	DIV, INV, NW, DNW, DBW, RR, K1, VA, RWCIF
YEARLY▲MARKET▲F▲2	DIV, INV, NW, DNW, DBW, RR, K1, VA,

	RWCIF
YEARLY▲MARKET▲F▲3	DIV, INV, NW, DNW, DBW, RR, K1, VA, RWCIF
YEARLY▲MARKET▲F▲4	DIV, INV, NW, DNW, DBW, RR, K1, VA, RWCIF
YEARLY▲MARKET▲Q▲1	L, DS, DQ, DP, DW, M, XS, MKTSHIND
YEARLY▲MARKET▲Q▲2	L, DS, DQ, DP, DW, M, XS, MKTSHIND
YEARLY▲MARKET▲Q▲3	L, DS, DQ, DP, DW, M, XS, MKTSHIND
YEARLY▲MARKET▲Q▲4	L, DS, DQ, DP, DW, M, XS, MKTSHIND
YEARLY▲INDUSTRY▲F	DIV, INV, NW, DNW, DBW, RR, K1, VA
YEARLY▲INDUSTRY▲Q	L, DS, DQ, DP, DW, M, XS, RU, Q

Names of all these table 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.

These 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+⍳20;1] RETURN

where *⍳* is the APL indexing character entered by Shift-I.

Those tables that are 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, let's assume that the following line is added into the

MSTARTxx 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.

There are three functions that are available to for printing those 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 highest possible 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 UNIX file name conventions. Recall that small and capital letters are considered different in the UNIX 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 function 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 programs such as 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, 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. After filling out the required sections of this full-screen input menu, type **F10** 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, the 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, respectivly, 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 save 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 availabale 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™ 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™ 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 it is strongly recommended that the user should 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 (see f.n.1, p.6).

7. 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 any one of the list of variables, VG1, VG2, VG3, VG4, or VG5.(*) These lists contains the names of variables in the dataset and used in various functions. If the added variable is a firm-level variable, an appropriate line should be added into NULLIFY, FIRMENTRY2, and MERGE▲WITH functions in the MOSES.PC workspace. (Data for new, real firms obtained from the Planning Surveys can also be added into the dataset. A new function will be written to perform this procedure.)

Third, a completely new dataset can be generated by using new micro and macro data. See Bergholm (1989) for this procedure.

---

*) The variables are divided into those five groups as follows: VG1: exogenous variables, VG2: endogenous variables, VG3: constants, VG4: indices, parameters of a technical nature, VG5: miscellaneous.

8. Model variants and experiment functions

The MOSES^{PC} Version 1.0 is "updated" by the modifications in the PERMANENT▲CHANGES (permanent changes in the mainframe version since 1986) and MOSES▲VARIANTS (larger permanent changes in the mainframe version since 1986) functions because of the fact that these modifications are permanent and well-documented in Albrecht *et al.* (1989). Therefore, the VARIANTS variable does not contain any information about the changes made by these functions.

During the development of the model, new functions are written to generate model variants that modifies the behavioral equations in the model. As of April 1, 1990, the following model variants are available.

PRICE▲CHANGES : This function was a part of the MOSES▲VARIANTS function, but not used in the PC Version 1.0. The MARKET▲CONFRONT▲NEWP, COMPUTE▲IMPORTS▲NEWP, and DOMESTIC▲RESULT▲NEWP functions are substituted for the MARKET▲-CONFRONT, COMPUTE▲IMPORTS, and DOMESTIC▲RESULT functions in the DOMESTIC▲MARKET function.

TARGET▲CHANGES : This function modifies firms' output expectations, uses a new TARG▲SEARCH function, and changes the money supply equation in the GOVERNMENT▲ACCOUNTING function. The new TARG▲SEARCH function which is created by the definition given by ▲TARG▲SEARCH variable nullifies firms that have negative expected net

price, and allows them to reduce their quarterly targets by an amount defined in the LOW▲TARGET variable when they cannot meet their profit targets. The default value of the LOW▲TARGET variable is equal to 1. If you want to use a different value, enter a value between 0-1 for this variable in in your modification function.

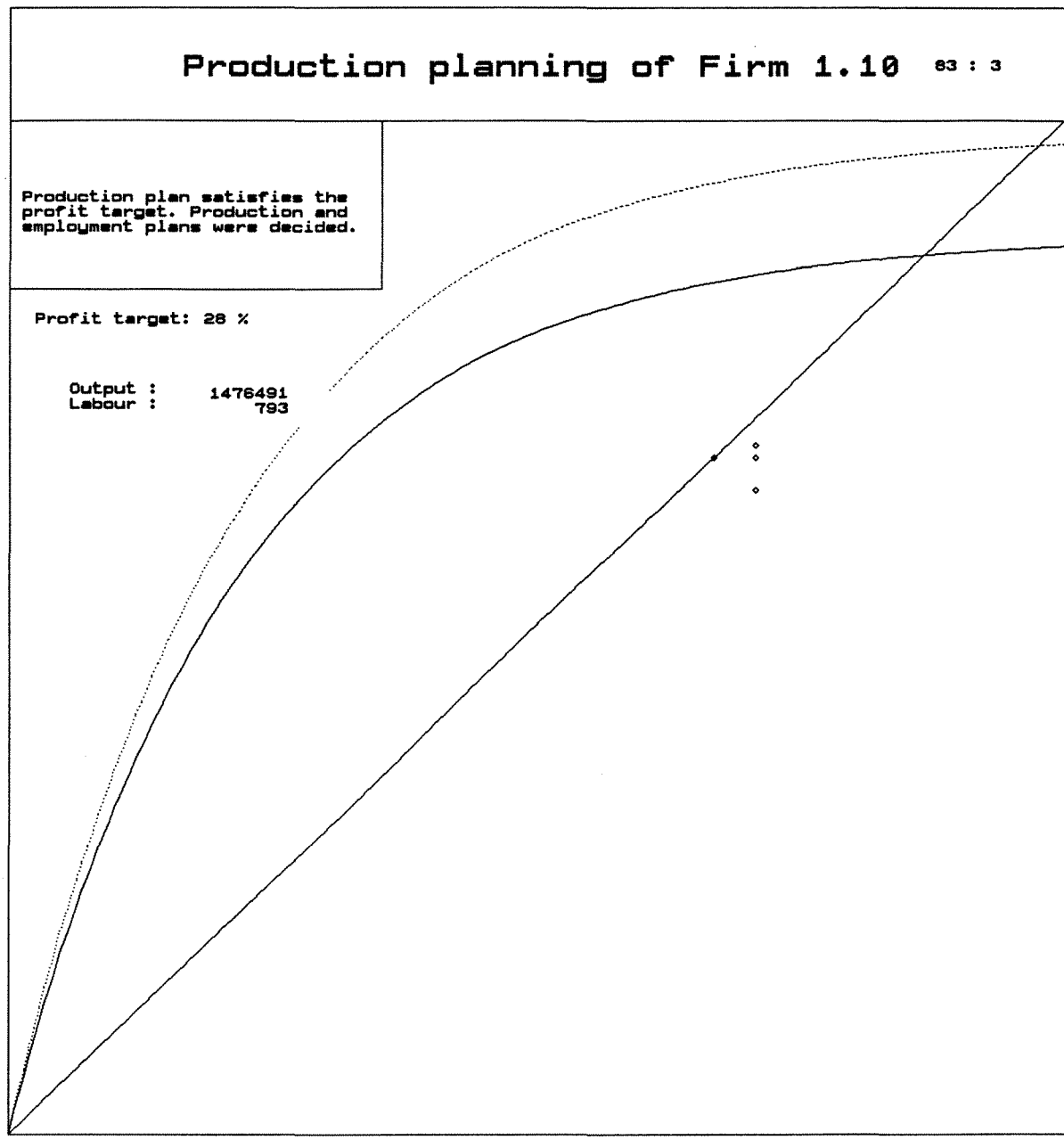
There is an alternative for the ▲TARG▲SEARCH variable. The ▲TARG▲SEARCH▲SHOW variable creates the same TARG▲SEARCH function as the ▲TARG▲SEARCH variable but it also demonstrates an animated presentation of a firm's search for quarterly output and employment targets, and allows the user to modify graphically the firm's output and employment target. 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. To use this target search function, enter the following line into the TARGET▲CHANGES function as the last command.

[]FX ▲TARG▲SEARCH▲SHOW

If this option is used, the following chart will be displayed on the screen when the TARG▲SEARCH function is invoked. The firm code, the time period for planning, the current profit target, the planned levels of


Production planning of Firm 1.10 83 : 3
Production plan satisfies the profit target. Production and employment plans were decided.
Profit target: 28 %
Output : 1476491
Labour : 793

output and labour, and the production function of the firm are shown on this chart. There will be explanations on 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 labour. If you want not 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 labour 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 labour, and press the RETURN key not to change them.

A number of new functions have been added into the model to be used in various experiment. As of April 1, 1990, these functions are as follows.

firmcode1 MERGE▲WITH *firmcode2 conv eff period* : *firmcode1* and *firmcode2* are firm codes of those firms to be merged, *conv* is the converge factor of new firm's TEC variable to the TEC variable of that 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 weighed by appropriate weights to determine the new firm's variables. The TEC variable of the new firm initially is equale to the weighed 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".

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 understand the structure of the model. Albrecht *et al.* (1989) contain the complete list of the model's APL 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 D.

Albrecht, James W., *et al.* (1989), *MOSES Code*. Stockholm: IUI.

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

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

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

APPENDICES

A. In case of trouble

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

1. The STOP▲HERE function is called: This function is used in a number of functions to remind the user about the possible errors made in editing model functions and 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 to simulation without merging two firms.

The STOP▲HERE function can also be called by the user in any place to stop simulation, for example, for manual entry of some data.

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 this 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 the total labour demand of new firms does not exceed the level of unemployment since it is implicitly assumed that new firms hire their labour from the ranks of unemployed. When their demand becomes larger that 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 labour 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 to resume the experiment.

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 because of these 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, 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 is equal to zero for any reason, 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 other type of error that halts the simulation has been seen in a large number of experiment until now, there may be some unknown misspecifications in the model that reveal themselves in a specific experiment design. In such a case, it may take a long time to remedy the errors since the model should be understood thoroughly for corrections.

---

*) "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."

B. MOSES.HELP functions

The MOSES.HELP function contains a number of useful 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.

[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 in the single quotation marks can not 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.

REG 'Y X_1 X_2 ... X_n' : This function is used for the regression analysis. It returns 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 horizontal axis, B is the variable whose cumulative value will be shown on the vertical 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) elements. 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 attributed of functions used in an experiment. To use the SET▲MONITOR function, copy it to the MOSES.PC workspace and

call in the MSTARTxx function. At the end of 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 miliseconds, and the fourth element the elapsed time in miliseconds.

scale Y▲RFIRM *firmcode* : *scale* and *firmcode* are defined same as in the case of the Y▲R▲FIRM function (see Section 6 on output reporting). This function creates a table called YEARLY▲RFIRM▲*frimcode*. 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. MOSES.GRAPH functions

Four functions are available in the MOSES.GRAPH workspace: PLOT, CHART, BARCHART, PIECHART, and SHOWFUN. There are a number of functions in this workspace that are used those five main functions. Therefore, it is better to use this workspace after saving the original output workspace 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

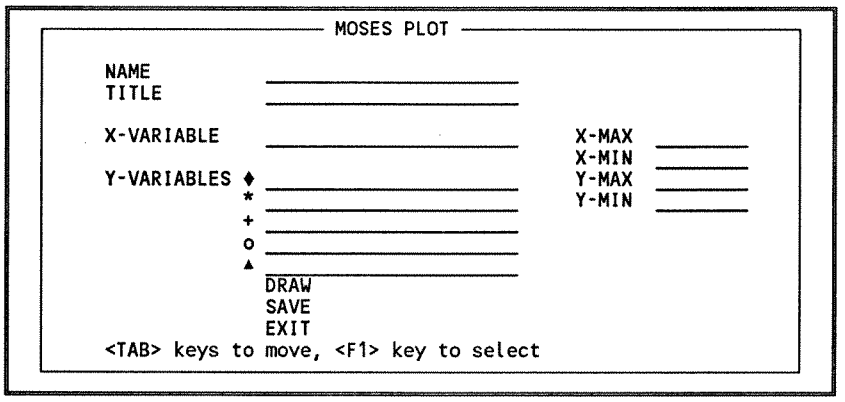
) COPY MOSES.GRAPH RETURN

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

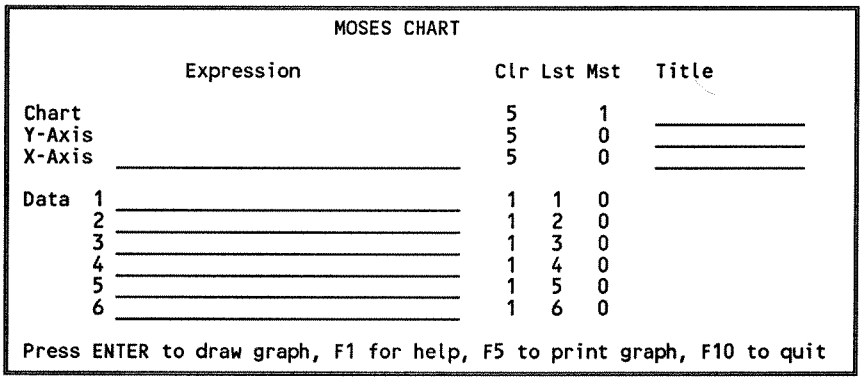
PIECHART RETURN

The PLOT function does not use the SCO Graphics™ 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.

When the PLOT function is invoked, the following menu appears. This function creates a plot of up to five variables. Enter the name of 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.



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 colour 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 HP plotter and 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 (see Section 4).

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

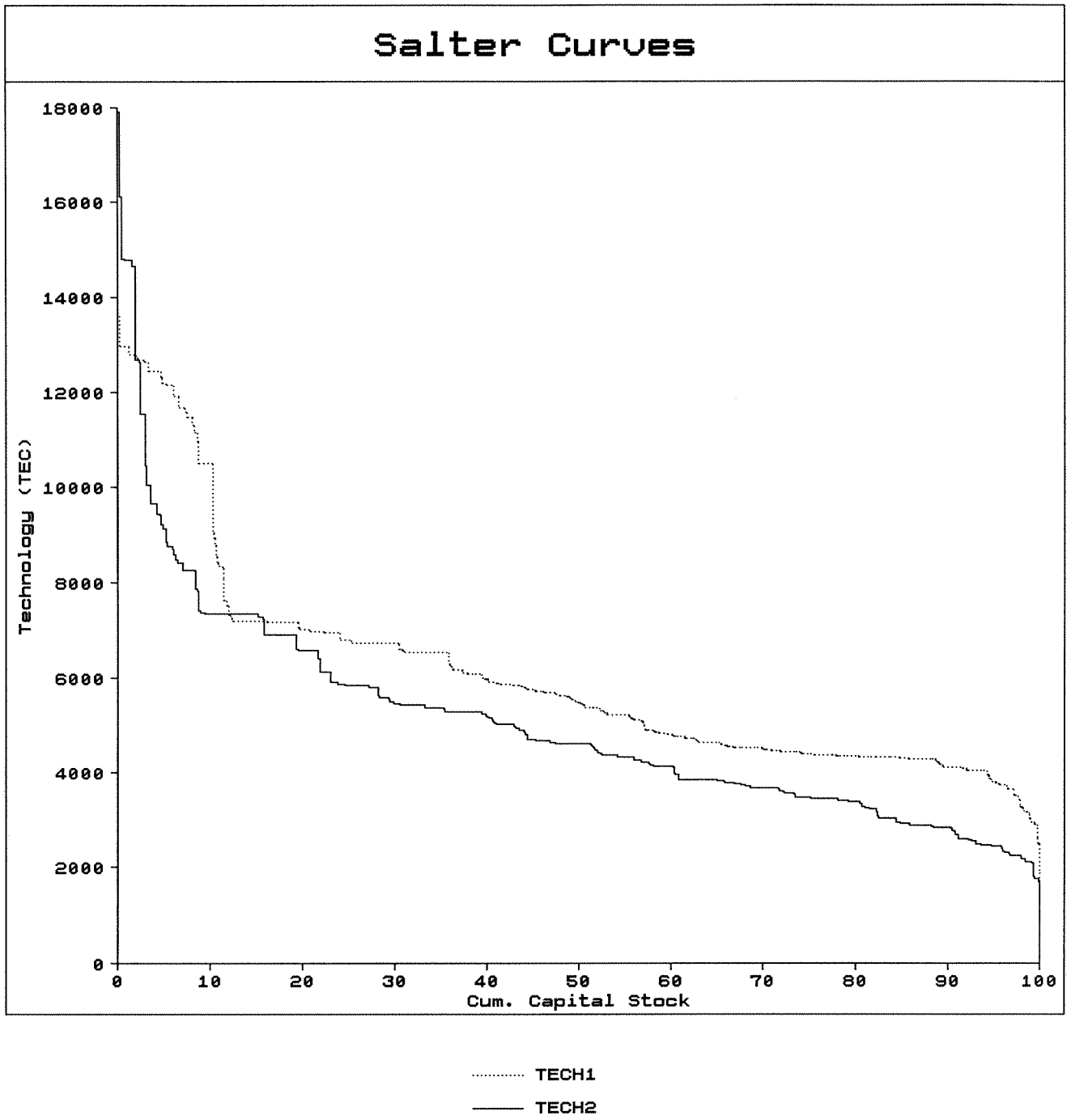
depending on the complexity of the chart. You cannot use any other function during printing. Note also that the computer may be locked out without any apparent reason because of the bugs in the SCO Graphics package although several checks for possible sources of errors are included into these functions and there has not been any problem so far using the HP printer. Therefore, be sure to save the output file before using any one of the graphics functions.

The following chart of the "Salter Curves" was printed by using an HP printer. Note that in this chart, the Mst option for the chart was set to 1 (labels at the bottom of the chart), and no grids were used.

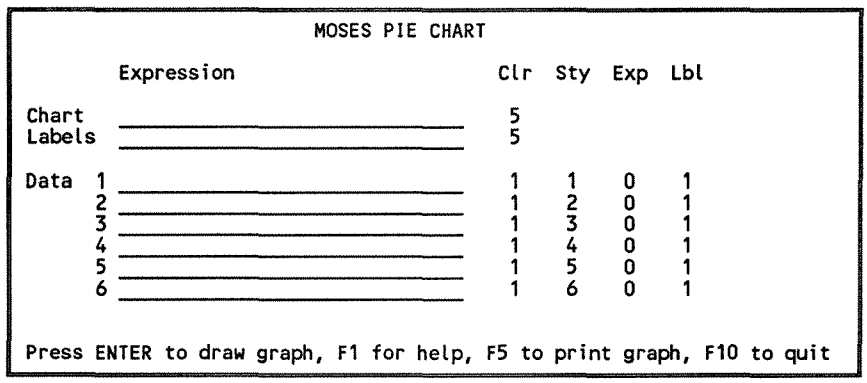
The following menu appears on the screen when the BARCHART function is invoked.

```
                        MOSES BAR CHART

                Expression                  Colour  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 following two bar charts were printed by this


Salter Curves
Technology (TEC)
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
0
10
20
30
40
50
60
70
80
90
100
Cum. Capital Stock
TECH1
TECH2

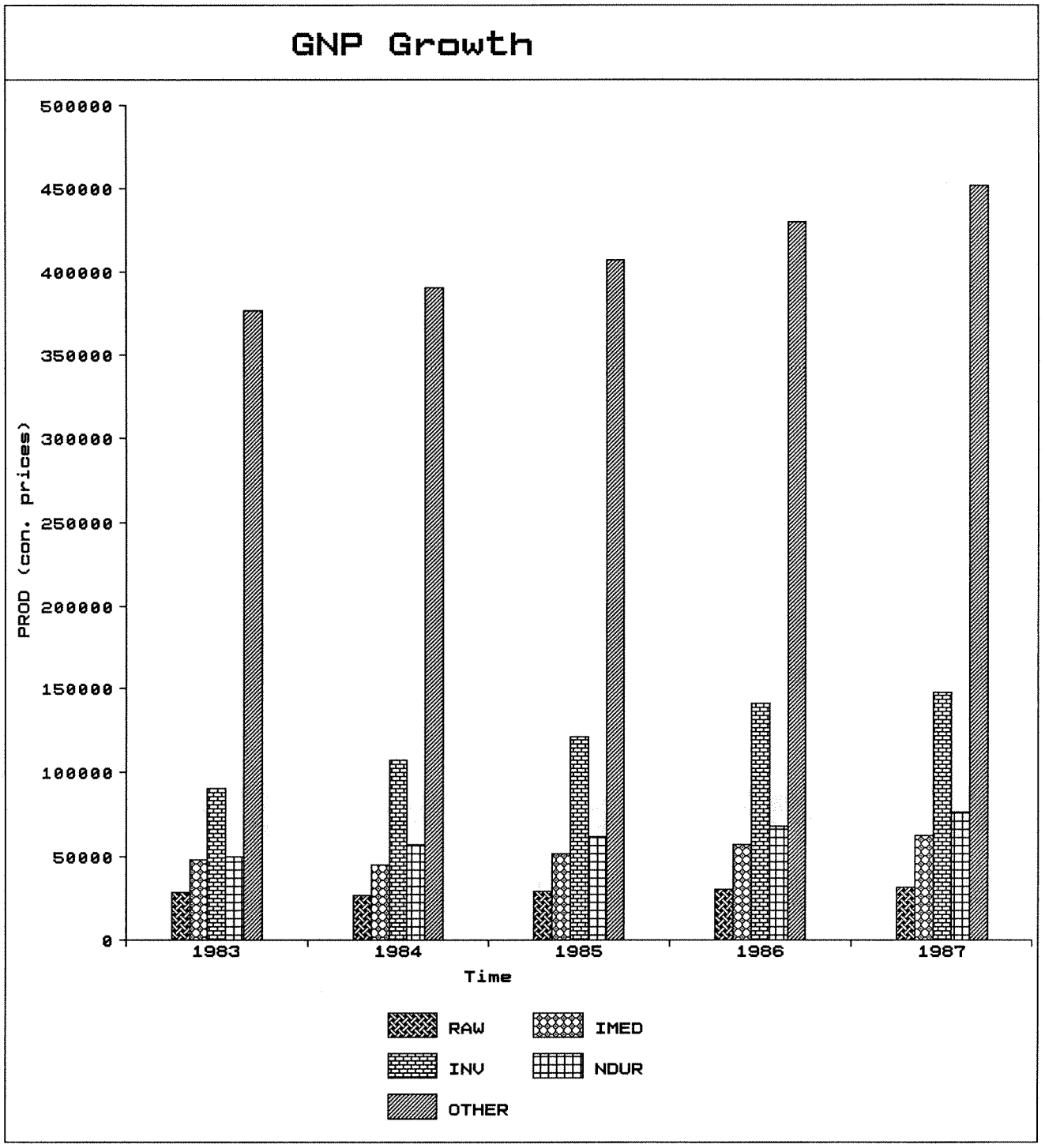
function for the same data. The first chart is a regular bar chart, and the second chart is a stacked bar chart. The style for the Y-axis in the second bar chart was set to 1 to draw the horizontal grids.

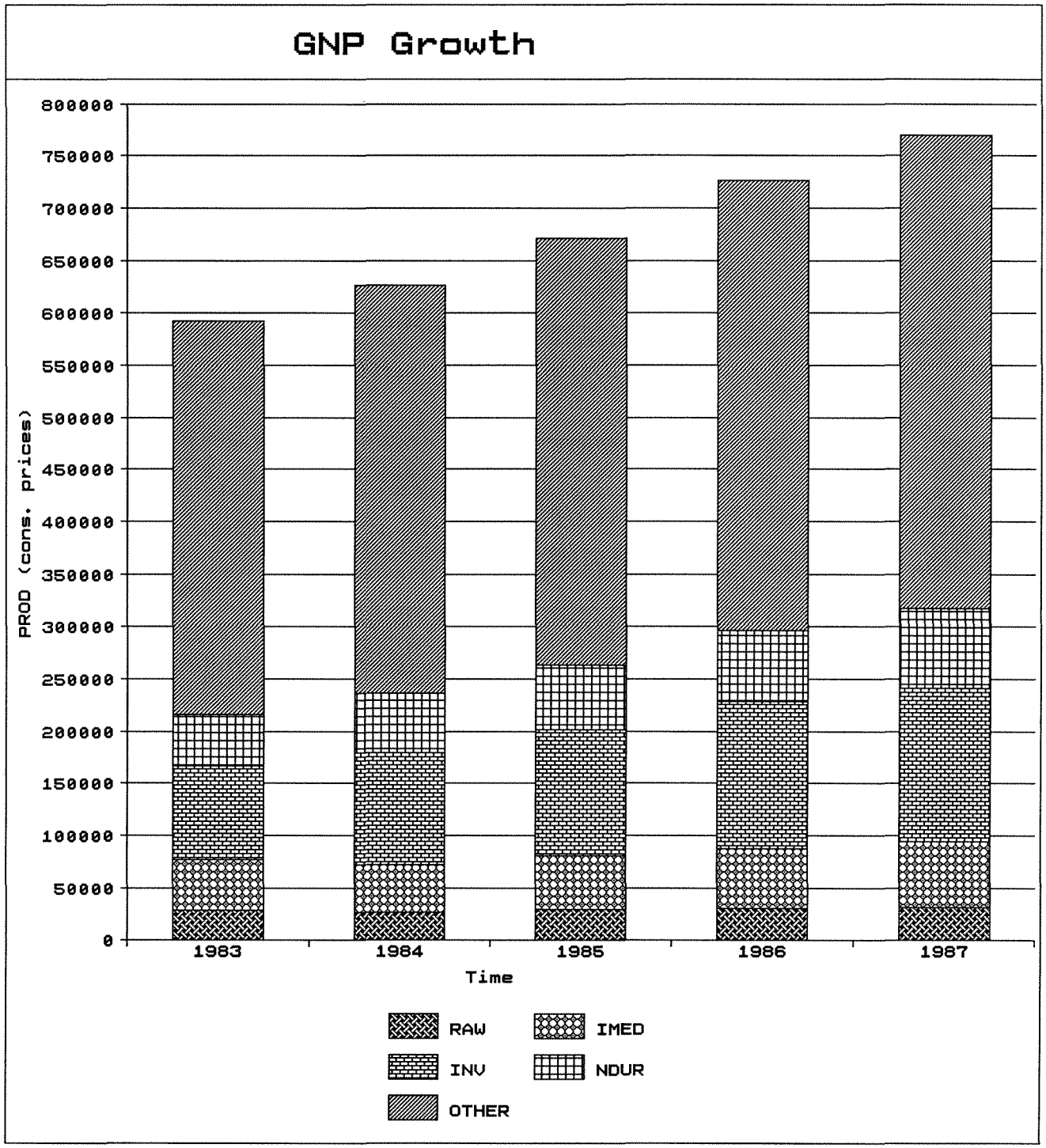


Enter the title of the chart under the "Expression" column. The name of the variable that contains the pie labels should be entered against the "Labels" line. The "Sty" column defines the interior style of each slice (enter a number between 0 and 9). The "Exp" column defines if the slice is exploded (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).

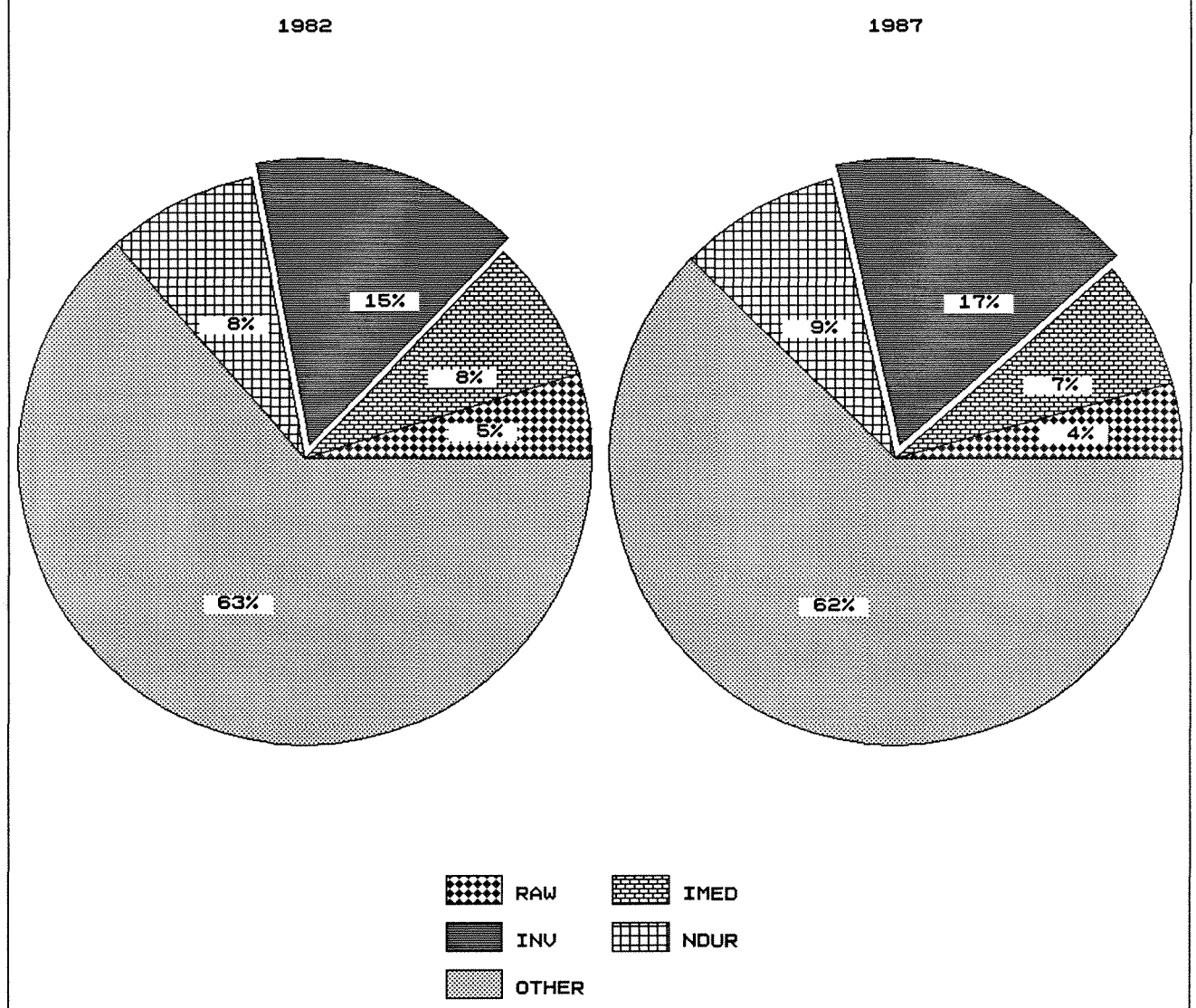
This function can simultaneously show at most three pie charts. Therefore, if the variable names of expressions entered against the data lines contain more than three variables, only the first three of them are used. The labels of slices to be displayed under the bottom of the pie chart are taken from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

The following figure was printed by this function. The "Lbl" option for


GNP Growth
500000
450000
400000
350000
300000
250000
200000
150000
100000
50000
0
PROD (con. prices)
1983
1984
1985
1986
1987
Time
RAW
IMED
INV
NDUR
OTHER


GNP Growth
800000
750000
700000
650000
600000
550000
500000
450000
400000
350000
300000
250000
200000
150000
100000
50000
0
PROD (cons. prices)
1983
1984
1985
1986
1987
Time
RAW
IMED
INV
NDUR
OTHER

Components of GNP



all variables, i.e., for all slices, were equal to 1. The "Exp" option for the third variable, INV, was set to 1 to display this slice as exploded.

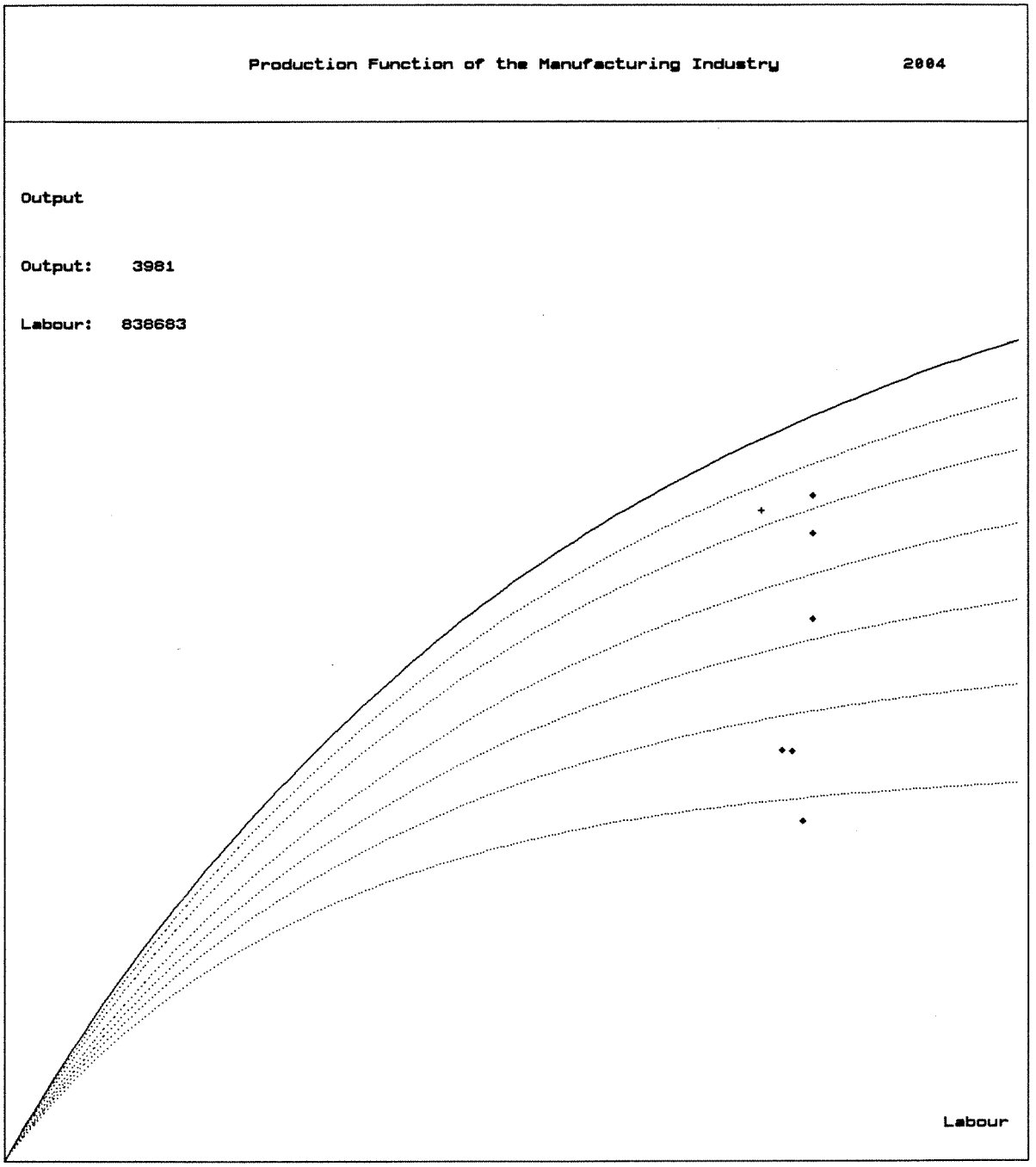
The SHOWFUN graphics function shows an animated time series chart of the production functions of industries or firms. The output tables should be availabale for this function, i.e., the YEARLY▲INDUSTRY▲TOTAL table for the manufacturing industry, YEARLY▲MARKET*x* table for each sector, and YEARLY▲FIRM▲*firmcode* 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 is available in the DATASET.SURVEY workspace.


Production Function of the Manufacturing Industry 2004
Output
Output: 3981
Labour: 838683
Labour

D. Differences between the PC and mainframe versions

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

a) The monadic Ä function in DEC-10 APL means "execute" whereas it means "type" in Dyalog-APL. Therefore all monadic Ä 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.

b) 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.

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

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

e) 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.

f) 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 PREPAREâRUN function.

After the transfer of APL workspaces, a number of changes to the model have been done. These changes can be summarized as follows.

a) 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 is also modified to work with this new function.

b) The results of simulation were being saved in text matrices in the mainframe model. But this method of keeping track of data is not convenient to use those variables for further processing (graphics, etc.). Therefore all transcription functions (except YâRâDISTRIBUTION and quarterly data functions) have been modified so that the results of simulation are stored in arrays in numerical form. This form of data allows us to use the results as a variable. Accordingly, PRINTâOUT 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.

c) Functions related with firm entry were considerably modified. A few lines that are used in the modification functions of the entry experiments (MSTART3x) were added into the function STARTâENTâMOD since they are required in any simulation with the entry option. STARTâENT2 function was also modified to allow probabilistic firm entry. STARTâENT1 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 and it is 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.

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

e) 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 proceed by the APL command →[]LC. (In the previous form of the function, it only prints "ERROR" message on the screen.)

f) Since the "updates" are well documented, the modifications made by these update functions have been done, and the UPDATEMOSES function is modified accordingly. Only those changes in the PRICEâCHANGES function have not been made.