Research Report
No. 35 1989

MOSES HANDBOOK

by
Fredrik Bergholm

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.

Board

Curt Nicolin, chairman
Gösta Bystedt
Anders Carlberg
Torbjörn Ek
Per-Olof Eriksson
Lennart Johansson
Olof Ljunggren
Lars Nabseth
Sven H Salén
Hans Stahle
Peter Wallenberg
Sven Wallgren
Karl Erik Önnesjö
Gunnar Eliasson, director

Address

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

THE INDUSTRIAL INSTITUTE FOR ECONOMIC
AND SOCIAL RESEARCH

Fredrik Bergholm

MOSES HANDBOOK

Distributor: Almqvist & Wiksell International, Stockholm, Sweden

(c) The Industrial Institute for Economic and Social Research,
Stockholm, Sweden

FOREWORD

The Swedish Micro-to-Macro Model has been developed over more than a decade. The actual operation of the model requires a not insignificant input of "on site learning". The size of the model system will make this unavoidable also in the future. To make the model more accessible to outside researchers, however, we are currently preparing a series of technical publications.

The MOSES Handbook, written by Fredrik Bergholm, documents in detail the complex initialization of the micro-macro model economy. It is a manual that will allow anyone reasonably experienced with computer work to run the model. The MOSES Code, to be published shortly, documents the code and the program. It will allow the model user to identify exactly how economic processes are represented in the model.

Stockholm in January 1989

Gunnar Eliasson

CONTENTS

Part 1

HOW TO RUN THE MOSES MODEL

CONTENTS

Preface

MOSES is short for "Model for Simulating the Economy of Sweden". Different versions of the model have been used within the institute for many years. A number of simulation experiments have been performed.* The whole model is written in the programming language APL. The size of the model can be measured by the required memory space when loading the APL-functions, and is currently approximately 550 kbytes. The current version of the model is installed on a DEC computer in Bergen, Norway. This has come about through cooperation with IØI (Bergen), the industrial institute for economic research in our neighbouring country. Previously the model has been installed in IBM computers in Stockholm and at other places. The model is currently being transformed onto PRIME APL to be run on the PRIME computer at the Stockholm School of Economics.**

Technical changes in the program code are being made in connection with this, e.g. speeding up the simulation and reducing required memory space during simulation.

* An earlier version is described in full detail in Eliasson (1978, abbreviated in (1980)). A new, updated presentation of the full model plus a complete bibliography is presented in Eliasson (1985).

** A shorter version of the MOSES model (based on synthetic micro data) is also written in FORTRAN and installed on a Norsk Data computer in the Central Institute of Mathematical Economics (CEMI), USSR Academy of Sciences, Moscow.

For some time there has been a demand for a full documentation of the current version of the model. "The MOSES Manual" fulfills one part of this request. Anyone interested in a large-scale simulation model of this kind needs to get acquainted with the techniques involved in starting up (initializing) and running the model. Experiments have shown that the initialization procedure, which constructs an initial state of the model economy, is crucial for the behaviour of the model. Part 2 of the MOSES Manual has been devoted to describing this initialization process.***

This part of the MOSES Manual describes how to actually run the model. Previously, only a few persons knew how to handle the machinery. The intention behind Part 1 is that anyone - following the instructions - should be able to start up and run the model.

Part 1 of the manual is mainly a cookbook. The user can run the model without knowing what it is all about. To be able to carry out meaningful analytical experiments, however, one needs a rather deep understanding of the model and the initialization itself, which requires a considerable and time-consuming effort on the part of the user.

*** A more detailed description of the micro (firm) database can be found in Albrecht-Lindberg (1982).

Section 1 Introduction

The model consists of two parts, the simulation model itself and the initialization procedure, which takes as input micro and macro databases and converts them into a form that fits the simulation model. There are micro and macro databases for two years, i.e. starting points for the simulation, namely 1976 or 1982. So, a simulation of the Swedish economy with the "micro-to-macro model" MOSES, is preceded by an initialization of a vast number of variables, and the simulation proper commences the first quarter 1983 or 1977 depending on the choice of starting point. "Initialization" means (mainly) that three kinds of variables are given values.

(1) Variables for 1982[1] needed to start up the model.

(2) Variables needed to determine the future of certain variables which get their values irrespective of what happens during the simulation.

(3) Certain constants, some of which are parameters affecting for example the behaviour of firms.

The required memory space for a simulation with the model version installed in Bergen is approximately 1,000 - 1,400 kbytes.

Let us see how the initialization and simulation can be performed on the DEC20-computer in Norway.

----

[1] or 1976, if we choose to use this year as the first year of the simulation. This is the earliest year for which a complete micro-to-macro database exists.

We assume that you are sitting in front of a terminal equipped with a keyboard with APL-symbols. To be able to run the model one needs some elementary knowledge of APL. On the following pages we will therefore try to give a mini-lesson in APL. For more detailed information, see the APL reference manual.

1.1 The Concept **Workspace** and System Commands

An APL **workspace** consists of functions and variables. The user enters functions and variables in the workspace and can save them for future usage.

A system command in the APL language starts with a right-hand bracket. One needs system commands to handle APL workspaces. The most frequently used system commands are:

```
) LOAD workspace name
) CLEAR
) SAVE
```

The) **LOAD-command** loads a previously saved workspace into the computer memory. Thus all functions and variables are instantaneously transferred to computer memory. In APL there is in general no program reading variables from a database into computer memory. Instead one makes a) **LOAD**-command.

Example:

```
) LOAD PROV
```

The workspace PROV is loaded into the computer memory. If you wish to look at a variable in the workspace, just write the name of the variable. If you want to add a variable just write for example:

a←10.

You have then added a new variable **a** which has the value 10.

If you want to save the extended workspace with new or changed variables (for example) simply write:

) SAVE

If you just want to erase computer memory and start all over again write:

) CLEAR

The **) CLEAR**-command erases the workspace which is in the memory at that particular moment. The workspace in the library is of course unchanged. Note that the **) SAVE**-command destroys the old version of the workspace in the library.

Example:

We want to look at the workspace containing the MOSES model. This workspace lies in the library and is called MOSES.

Perform:

(1) **) LOAD** MOSES
(2) we look at some variables or functions
(3) **) CLEAR**

A function is a program in APL. How to write and read functions is not described here; see instead the APL reference manual. To run an old experiment you don't have to be able to write functions, but if you wish to make new experiments this is necessary!

The most common system commands can be found in the appendix to the next section. A full description can be found in an APL-manual.

Information about the "login procedure", e.g. how to get in touch with the computer for instance in Bergen, Norway, or at the Stockholm School of Economics, can be received from authorized persons at IUI.

Section 2 How to Run the Initialization and How to Start the Model

2.1 Overview

We assume that you have logged in and that you have access to:

1) the APL-language
2) workspaces containing the micro-to-macro model, which are:

MOSES (= the model itself)
,INIT (= the initialization procedure)
,MACR82 (= macro database, for 1982)
      or MACRO (= macro database, for 1976)
,SI82 (= micro database, for 1982)
      or SI76 (= micro database, for 1976)
,VLISTS (= variable names, initialized variables)
,FUNCTI, ISTART,
MSTART (= miscellaneous workspaces in connection with initialization etc).

One can check that all workspaces needed are in the (IUI) library by using the system command:

```
) LIB
```

Then all saved workspaces are listed.

It might also be a good idea to perform the **) MAXCORE**-command in the beginning. This command is described in an appendix to this section, and extends maximum available space in the computer memory. It is not possible to use more than 352 p (pages).

To start the model enter:

```
) LOAD INIT            (step 1)
  START NR1            (step 2)
) LOAD MOSES           (step 3)
  UPDATEMOSES NR2      (step 4)
  RUNEXP N             (step 5)
) SAVE                 (step 6)
```

Step 1+2 = initialization of the model, which means that start-up variables for 1982 (1976) are given values etc.

Step 3+4 = update the MOSES model code to its latest version

Step 5 = run the model

Step 6 = save the result

NR1, NR2 and N are chosen numbers.
N = number of years to simulate (run the model)
NR1 = number of **initialization version**
NR2 = number of **experiment version**

For a definition of "**initialization version**" and "**experiment version**", see the following pages.

This 6-step procedure to run the model will be explained in more detail below.

In what follows, we will write system commands and function calls (for example the function **START**) in boldface letters, whereas parameters to the functions (for example NR1, NR2, N above) or workspace names (for example INIT) will be written in ordinary letters. Note that there must be at least one blank between a function call and a parameter!

2.1.1 "Initialization Version"

Definition:

Instead of changing the original[1] initialization version, one puts all changes in a separate function. A new initialization version is made by using the original initialization program (workspace INIT) and by making a function **ISTARTXX** where all changes are defined. XX is the number of the initialization version. **ISTARTXX** should be put into workspace ISTART before anything else is done.

Example:

Make a function **ISTART4** where all changes are defined. Put it into workspace ISTART by using the system commands **LOAD** and **SAVE**. Then perform:

```
) LOAD INIT            (step 1)
  START 4              (step 2)
```

Thus, initialization version 4 has been run, and the result is saved (automatically) in workspace R4.

How to make new initialization versions is described in Section 3. If you just wish to repeat an old experiment just check that the proper **ISTARTXX**-function is stored in the ISTART-workspace.

The reason for this somewhat cumbersome way of labelling the experiments is that it is extremely difficult to keep track of changes in a program (in this case the initialization program **START**), if one were allowed to make small modifications all the time. If one works in the fashion outlined above,

[1] which dates back to January 1982.

the original initialization code is unchanged, and all modifications thereafter are defined by small modules called ISTARTXX. To be more specific, the original initialization program is unchanged before a new initialization. During the initialization, the initialization program is updated[1] with the changes defined in an ISTARTXX-module. For more details see Section 3.

[1] A program can update another program during execution in the APL-language. This is a somewhat unusual feature for a programming language.

2.1.2 "Experiment Version"

Definition:

Instead of changing the original[1] model one puts changes of the model, for experiment purposes, in a separate function. A new experiment version is made by using the original model program (workspace MOSES) and by making a function **MSTARTXX**, where all changes connected with the specific experiment are defined. **XX** is the number of the experiment version. Put **MSTARTXX** into workspace MSTART before anything else is done.

Example: (We extend the example given above on the previous page)

Make a function **MSTART8** where experiment-specific changes are defined. Put it into workspace MSTART.

Then perform:

```
) LOAD INIT              (step 1)
  START  4               (step 2)
) LOAD MOSES             (step 3)
  UPDATEMOSES  8         (step 4)
```

How to make new experiment versions is described in Section 3. If you just wish to repeat an old experiment, simply check that the proper **MSTARTXX** function is stored in the MSTART workspace.

Note that each experiment is uniquely determined by the lines in an **MSTART** function, provided that the same input data are used. Note also that the above procedure means that one can use a certain set of indata and make a large number of different experi-

----

[1] which dates back to the version as it stood in 1978.

ments with these. Step 2 yields output from initialization (= input to the model). By varying the parameter to **UPDATEMOSES** (in this example 8) one can achieve different experiments with the same indata to the model.

One point ought to be clarified in this context. The word "experiment" in our concept "experiment version" means that we are experimenting with the model itself, for example introducing a certain micro or macro behaviour. One can also make experiments by varying input to the model by using different "initialization versions". Some parameters produced by the initialization procedure affect the micro behaviour of firms. So it is equally natural to keep the **MSTARTXX** function constant and vary the **ISTARTXX** functions. The difference between an **ISTARTXX** function and an **MSTARTXX** function is that the former can't change lines in the model.

2.2 Comments on the 6-step Procedure to Start and Run the Model

Step 1 The initialization program is loaded into computer memory

Step 2 "**START** XX" starts the initialization. **ISTARTXX** should be stored beforehand in workspace ISTART. The result from the initialization is stored in workspace RXX. XX = number of initialization version. Step 2 takes about 5 minutes. The monitor prompter resumes the original position when this step is ready.

Step 3 The model itself is loaded.

Step 4 "**UPDATEMOSES** XX" updates the model from the 1976 version to the 1982 version and makes experiment-specific changes in the model code. **MSTARTXX** should have been stored beforehand in workspace MSTART. **XX** = number of experiment version. The monitor prompter resumes the original position when this step is ready.

Step 5 The model is run for N years. This takes about 10 minutes per simulated year. The monitor prompter resumes the original position when the run is over.

Step 6 The simulation result is saved.

During step 2 the program asks whether one wants to see the input-output matrix or not (described in Part 2 of the handbook). Answer, yes or no.

During step 4 a question comes up, where the user should give the name of the workspace from which the result of the initialization should be fetched. The name is R1 if you used initialization version 1, R2 if you used initialization version 2, etc.

IMPORTANT NOTE: If you already, (for example, some days before) have made the initialization (steps 1+2) you can start with step 3 at once. This might be a convenient way to work.

The simulation result is not stored automatically. The user has to make the) **SAVE** command himself (step 6). The name of the simulation result is SXXVYY, where XX = number of experiment version, and YY = number of initialization version. S stands for simulation result and V for version. We conclude by giving two examples of a MOSES run.

Example 1:

```
) LOAD INIT
START 1
) LOAD MOSES
UPDATEMOSES  7
RUNEXP  10
) SAVE
```

Initialization version 1 and experiment version 7 are made. The model is run for 10 years. The result from the initialization is stored in workspace R1. The result from the model simulation is stored in workspace S7V1.

Example 2:

```
) LOAD MOSES
UPDATEMOSES  7
RUNEXP  10
) SAVE
```

The initialization is already made. The user is asked for the name of the workspace where the result from the initialization is stored (for example R1 if initialization version 1 is used). The result from the simulation is stored in a workspace called S7V1.

IMPORTANT:
If one wants to look at the contents of a workspace, one makes the) **LOAD** command. There is one exception, though.

To be able to look at the result of the initialization, RXX, after having performed steps 1 and 2 in the 6-step procedure, one has to use the) **COPY** command.

Thus:) **COPY** R1 must be done to look at workspace R1, instead of) **LOAD** R1. (If you try to do the **LOAD** command instead, a lot of nonsense will be printed out, for technical reasons.)

A complete list of variables coming out from initialization can be found in Part 2 of the MOSES Handbook.

2.3 Batch-Job

So far we have assumed that the user of the model makes the commands (the 6-step procedure to start and run the model) actually sitting in front of the computer terminal, doing so-called time-sharing. This is by no means necessary.

The same commands can be written in a so-called "command file" beforehand. To perform a run on the computer in this way is called a batch-job. It is very convenient because the program can be run during low-cost time (for example) during the night, without anybody at the computer terminal. The terminal need not even be switched on.

On the DEC20 computer (in Bergen) a batch-job is done in this way:

A "command file" is made (using the editor on the DEC20). In the command file prov.CTL we have:

```
monitor prompter APLSF
* TTY
* ) MAXCORE  352
* ) LOAD  INIT
* START  4
* yes
* ) LOAD  MOSES
* UPDATEMOSES  8
* R4
* RUNEXP  10
* ) SAVE
* ) MONITOR
```

The file could have any name, but the extension must always be CTL. In this example we called the file "prov.CTL". A DEC20 monitor command (the command **APLSF**)[1] should start with a monitor prompter (a helix-shaped symbol) in a command file. An APL system command should start with an asterisk. More information about batch-jobs can be found in "DEC-SYSTEM Batch Reference Manual".

The batch-job is ordered by using the DEC20 command:

```
submit prov.CTL
```

One could add extra instructions to the operator, by writing remarks after "slashes", according to syntax (see the Batch Ref. Manual).

We found the following extra instructions useful in practical work:

```
submit prov.CTL/after:24:00:00/TIME:03:00:00/
restart:YES/PAGE:300
```

The result from a batch-run is stored on a file called "prov.log". This file contains the output which would have been typed on the screen if the job had been done as time-sharing.

A disadvantage with batch-jobs is that if anything (no matter how trifling) goes wrong during the model run, nothing can be done about it. The program is simply interrupted. The batch-job has to be connected and repeated.

[1] There are different versions of APL, one of which is APLSF, working on the DEC20 system.

APPENDIX

"System commands"

) **LOAD** workspace name
Loads the workspace into computer memory.

) **CLEAR**
Clears the computer memory.

) **MAXCORE** 352
If one has too little space in the computer memory, one has to ask for more space. The command above gives the user maximum available space. One can choose any number between 6 and 352 (pages). 1 page ≈ 1/2 Kwords.

) **WSID**
This is a question. The computer answers by telling the user the name of the workspace which is in the computer memory right now.

) **WS** name
Changes the name of workspace to the name written after WS.

) **SAVE**
Saves the workspace under current workspace name.

) **FNS**
Lists all functions in the current workspace.

) **VARS**
Lists all variables in the current workspace.

) **COPY** workspace name
A copy of the workspace is added to the workspace one is working with.

The system commands might differ somewhat on different computers. The commands mentioned above are used on DEC20 in Bergen, Norway.

Section 3 How to Make New Initialization Versions and New Experiment Versions (simulation variants)

3.1 New Initialization Versions

Do:

) LOAD ISTART (step 1)

This loads the workspace ISTART, where all previous initialization versions are stored (**ISTART1, ISTART2 ...**).

Make a function

ISTARTXX (step 2)

XX = number of initialization version.

How to make a function is described in the APL reference manual.

Save the extended workspace ISTART.

) SAVE (step 3)

ISTART1 and **ISTART2** on the next page show the pattern to be followed when making an **ISTARTXX** function.

Comment lines in the APL language start with a very particular symbol, the so-called cap-null symbol. It appears frequently in the initialization code (see for example Section 5) and looks like an A which is smaller and more smooth than an ordinary A. For typographical reasons we write this symbol as a boldface **A** in the examples below.

Example 1:

```
    ∇   ISTART1
[1]   A TEST1
[2]   A FREDRIK B
[3]     SYNTHΔFIRMS← 8 16 18 8
[4]   A
    ∇
```

Example 2:

```
    ∇   ISTART2
[1]   A TEST
[2]   A JANUARI 1982
[3]   A FREDRIK B
[4]   A
[5]     SYNTHΔFIRMS← 8 16 18 8
[6]     'MARKETSΔDATA' MODSUBST 'GAMMA←ωGAMMA←0.5'
[7]     'MARKETSΔDATA' MODDEL 'KSI←'
[8]     'MARKETSΔDATA' MODADD 'NITER←ωKSI←0.3'
    ∇
```

Note: If you by mistake put in an extra blank somewhere between the '-signs, you could fail to make the proper changes.

GAMMA, KSI and NITER are three parameters in the MOSES model. **MARKETSΔDATA** is the name of a subfunction in the initialization procedure (a subfunction to the function **START**). To be able to make changes in the initialization one has to get well acquainted with the initialization program, which is described in Part 2. Many parameters which guide firm behaviour are given their values in the subfunction **MARKETSΔDATA** which is easy to read. One can without difficulty change such parameters according to the pattern above. The examples above are explained in detail on the following pages.

ISTART1:

No changes in the initialization program are made. The line "SYNTH∆FIRMS← 8 16 18 8" is compulsory in any **ISTARTXX** function. This line means that there will be 8 synthetic[1] firms in sector 1, 16 synthetic firms in sector 2, 18 in sector 3 and 8 in sector 4. Thus, this line tells how many synthetic firms will be created during the initialization procedure.

ISTART2:

The APL functions **MODDEL, MODADD** and **MODSUBST** are used to change lines in the initialization program. The changes will take place when the **ISTARTXX** function is called, and this happens in the very beginning of the initialization.

Thus: Line 6, example 2, means that we will change a line in the subfunction **MARKETS∆DATA** (see Appendix C, Part 2). The text before **MODSUBSTS** (all textstrings should stand between '-symbols) tells the name of the function where the changes are to be made. The text after **MODSUBST** tells what line is to be changed and defines the new line. The beginning of the old line stands before the "omega symbol" (ω) and the new line after this symbol. **MODSUBST** deletes the old line beginning with "GAMMA" and inserts the new line: GAMMA←0.5. **MODADD** works like **MODSUBST**, with one exception; **MODADD** does not delete the old line. The new line is put immediately after the old line. Line 7 means that we will delete any line in the subfunction **MARKETS∆DATA** beginning with KSI←.

----

[1] artificial firms, which define the difference between macro data (national accounting) and micro data (real firms). See Albrecht-Lindberg (1982).

3.1.1 Summing Up

Syntax:

'function name' **MODSUBST** 'old line ω new line'

'function name' **MODADD** 'old line ω new line'

'function name' **MODDEL** 'line'

Don't take too many letters! This may cause overflow during execution. Error: Workspace full. Don't take too few letters! Ambiguous commands might be misunderstood and applied to the wrong lines. Any line (in the function at which you are looking) with the same string of letters (in the **beginning** or in the **middle** of the line) might be affected by the **MODSUBST**, **MODDEL** or **MODADD** command.[1]

3.2 New Experiment Versions

Do:

) **LOAD** MSTART (step 1)

This loads the workspace MSTART, where all previous experiment versions are stored (**MSTART1**, **MSTART2**, ...). Make a function

MSTARTXX (step 2)

) **SAVE** (step 3)

On the next page you can see an example of an **MSTART** function. **MODSUBST**, **MODDEL** and **MODADD** are used in the same way as was done above, in connection with new initialization versions.

----

[1] Note: It is not allowed to change the function where you are at the moment. Thus **MODSUBST**, **MODDEL**, **MODADD** cannot make changes in the function **START**.

An example of a MSTART function:

```
     ∇   MSTART1
[13]    A
[14]    A EXPERIMENT
[15]    A **********************
[16]      'INVFIN' MODSUBST 'QDIV←ωQDIV←0.6·QTAX'
[17]    A
     ∇
```

Note: **A** is a symbol starting comment lines, for typographical reasons written as a boldface A.

In a subfunction, called **INVFIN**, in the MOSES workspace one line is altered.

The new line

```
    QDIV←0.6·QTAX
```

means that each firm will, in this particular experiment, pay dividends to the household sector amounting to 60 percent of the corporate tax.

The changes in the MOSES program take place when the function **UPDATEMOSES** is called. An **MSTART** function is, namely, called on a line in the function **UPDATEMOSES**.

If you wish to check that the changes in the MOSES program have been performed correctly, list the functions you are interested in (in the example above the function **INVFIN**) after the call of the function **UPDATEMOSES** (step 4 in the 6-step procedure presented in Section 2). How to read (list) functions is described in the APL reference manual.

Section 4 How to Add New Model Variables

Say that you have a new variable I which you wish to give as input to the model. The variable I is to be given a value in the initialization procedure.

It is <u>not enough</u> just to add this variable to the initialization program in an **ISTARTXX** function, for example with a "MODADD line".

You must also add the name of the variable to a variable list in workspace VLISTS before making the initialization. All output variables from initialization should be listed in this workspace. If this is not done, <u>the variable I will be deleted during the initialization procedure</u>, since it is not mentioned among the output variables in a variable list in workspace VLISTS. Thus, this system forces the user to mention all new output variables from initialization.

There are 6 variable lists in workspace VLISTS, and the user can extend 5 of them, namely:

Variabelgrupp 1	(= exogenous variables)
Variabelgrupp 2	(= endogenous variables)
Variabelgrupp 3	(= constants)
Variabelgrupp 4	(= indices, parameters of a technical nature)
Variabelgrupp 5	(= miscellaneous)

To add the variable I to a variable list in workspace VLISTS do:

```
) LOAD VLISTS                              (step 1)
Variabelgrupp 1←variabelgrupp 1, ' I'      (step 2)
) SAVE                                     (step 3)
```

Note: We assumed above that I was an exogenous variable, and that was why variabelgrupp 1 was used. The division of variables into 6 groups is purely for book-keeping purposes. If you put a new variable in the wrong group, nothing will happen, but the book-keeping will be messed up.

Note: The text string between the '-signs in step (2) must begin with a blank!

Section 5 Description of the Function UPDATEMOSES

UPDATEMOSES is the function doing changes in the model (workspace MOSES) itself, and is step 4 in the 6-step procedure presented in Section 2.

The function is documented below, with the APL code itself. It should be noted that this documents all permanent changes in the model program since 1978.[1]

In **UPDATEMOSES** four functions are called:

1) **PREPARE∆RUN** Fixes headlines etc. for printing out simulation results)

2) **PERMANENT∆CHANGES**[2] (Permanent changes in the model program since 1978)

3) **MOSES∆VARIANTS**[2] (Larger permanent changes in the model program since 1978)

4) **MSTARTXX** (The experiment version function)

Note: The MSTART function should never be used for permanent changes of the model.

The model program is updated in this fashion, so as to make it possible to repeat old experiments from 1978 and onwards. Another reason for this updating procedure is to ensure that the changes are clearly defined, and can be checked by anyone wishing to convince himself that the changes have been properly made.

----

[1] For a complete understanding of these changes one needs, however, the model program itself, the so-called MOSES code, which will be presented in another part of the MOSES Handbook.

[2] The changes in **PERMANENT∆CHANGES** are perhaps more permanent than **MOSES∆VARIANTS**, which is why they are separate.

Description of UPDATEMOSES

```
    ∇  UPDATE∆MOSES NUM
[1]   ⍝THIS FUNCTION DOES:
[2]   ⍝(1): PREPARES HEADLINES ETC. FOR PRINT-OUT FROM MOSES-RUN
[3]   ⍝(2): MAKES CHANGES IN THE MOSES PROGRAM FROM 1978
[4]   ⍝
[5]    PREPARE∆RUN
[6]    PERMANENT∆CHANGES
[7]    MOSES∆VARIANTS
[8]   ⍝EXPERIMENT-MODULES IN WORKSPACE MSTART.
[9]   ⍝THEY ARE CALLED MSTARTXX WHERE XX IS THE NUMBER IN
[10]  ⍝THE CALL 'UPDATEMOSES XX '   .
[11]  ⍝
[12]   'GIVE THE NAME OF THE WORKSPACE WITH START-'
[13]   'VALUES FROM INITIALIZATION (FOR EXAMPLE R1 ETC.) :'
[14]   INITWORKSPACE←⎕
[15]   ⍎')COPY ',INITWORKSPACE
[16]  ⍝THIS LINE FETCHES INDATA FROM INITIALIZATION...
[17]   PREPARE2
[18]   ⍎')COPY MSTART  MSTART',⍕NUM
[19]   ⍎'MSTART',⍕NUM
[20]  ⍝LINE ABOVE MEANS THAT MSTARTXX IS EXECUTED.
[21]  ⍝
[22]  ⍝ IF YOU WANT MARKET TIME-SERIES RESULTS TO BE PRINTED OUT
[23]  ⍝ DURING SIMULATION REMOVE COMMENT ON NEXT LINE...
[24]  ⍝ TRACE1
[25]  ⍝
[26]  ⍝TRACE1 PRINTS OUT TIME-SERIES RESULTS...
[27]  ⍝
    ∇
```

```
     ∇  PERMANENT∆CHANGES
[1]     ⍝ 'PERMANENT' CHANGES IN MOSES:
[2]     ⍝
[3]      'MARKET∆CONFRONT' MODADD 'PTωQPURCHG←QPURCHG×PT[⍳10]×(1÷(QWG÷WG∆R
        EF))÷100'
[4]     ⍝CORRECTION OF DEFLATOR FOR QOVERNMENT'S PURCHASES
[5]     ⍝
[6]      'INVFIN' MODSUBST 'K1BOOK←K1BOOKωK1BOOK←K1BOOK-QDEPRBOOK←0⌈QREVLR
        HOBOOK×K1BOOK←QINV+K1BOOK'
[7]     ⍝ BOOK-VALUE SHOULD NOT BE UPDATED WITH INFLATION
[8]     ⍝
[9]     ⍝ CHANGES TO VECTOR-FORMAT ON MAX/MIN/OPT-STO MADE PERM.
[10]    ⍝ CORRESPONDING FOR MAX/MIN/OPT-IMSTO IN FN.'IND∆PURCH∆SHARES'
[11]    ⍝
[12]    ⍝
[13]     'PLANQREVISE' MODSUBST 'QIMQ←0⌈ω⍝ QIMQ-LINE MOVED ABOVE IMSTO'
[14]     'PLANQREVISE' MODADD 'QQ←QQωQIMQ←0⌈((⍉IO)[MARKET;]MULT7 QPLANQSAV
        E)+(OPTIMSTO-IMSTO)÷4×TMIMSTO'
[15]    ⍝
[16]     EXO∆QDWG←''
[17]    ⍝THE Z-SECTOR IS OBSOLETE AND ISN'T ANY LONGER USED.
[18]    ⍝THE FOLLOWING LINES DELETE OR CHANGE LINES SO THAT
[19]    ⍝THIS SECTOR IS IGNORED.
[20]     'LABOUR∆MARKET' MODDEL 'ZLABOUR'
[21]     'MARKET∆CONFRONT' MODSUBST 'PT←QωPT←QPRELPDOM,(QPDOM[IN]×1+QDPIN)
        ,1'
[22]     'HOUSEHOLD∆INIT' MODSUBST 'INMONEYωINMONEYHH←QTRANS+QINPAY+QTDIV+
        (SUM2 L×QW÷4)+(LG×QWG÷4)+(QINTH←NH+.×RIH×WH÷4)'
[23]     'HOUSEHOLD∆INIT' MODSUBST 'QTWS←(LGωQTWS←(LG×QWG÷4),SUM2 L×QW÷4'
[24]     'HOUSEHOLD∆INIT' MODSUBST 'QTI←QTDIωQTI←QTDIV+QTRANS+((+/QTWS)-QW
        TAX)+QINTH'
[25]     'LUUPDATÉ' MODSUBST 'LF←ωLF←LU+LG+SUM2 L'
[26]     'Q∆EXO' MODDEL 'TXVAZ←'
[27]     'LABOUR∆UPDATE' MODSUBST 'RU←RU+QωRU←RU+QCHRU←(LU÷LU+LG+SUM2 L)-R
        U'
[28]     'DOMESTIC∆RESULT' MODDEL 'QPZ←'
[29]     'FINALQPQSQM' MODDEL 'QMZ←'
[30]     'FINALQPQSQM' MODDEL 'QCHKZ←'
```

```
[31]   'FINALQPQSQM' MODDEL 'QDIVZ←'
[32]   'INVFIN' MODSUBST 'QCTAX←(SU⍵QCTAX←(SUM2 QTAX)'
[33]   'Y∆COUNTRY∆TOTAL' MODDEL 'CT∆CHKZ←'
[34]   'DOMESTIC∆RESULT' MODDEL 'QSZ←'
[35]   'INDIRECT∆TAXES' MODSUBST 'QVATAX←⍵QVATAX←(TXVA2×QPURCHG+QSP[MKT,
      IN;]+.×NH),0'
[36]   'Y∆INDUSTRY∆TOTAL' MODSUBST 'LTOT←⍵LTOT←LG+SUM2 L'
[37]  ⍝
[38]   VARIANTS←VARIANTS,' PERMANENT CHANGES 1980-81 '
    ∇
    ∇  MOSES∆VARIANTS
[1]     NEG∆IMSTO
[2]     IND∆PURCH∆SHARES
[3]     POSITIVE∆NET∆WORTH
    ∇


     ∇  PREPARE∆RUN
 [1]     LOEPNR←NUM
 [2]     DATUM
 [3]     DSCR←'**** EXPERIMENT ',⍕NUM
 [4]     ⎕PW←120
 [5]     ⍎')MAXCORE 352'
 [6]    ⍝MAXIMUM CORE IN COMPUTER-MEMORY...
     ∇

     ∇  PREPARE2
 [1]    ⍝ THIS LINE GIVES THE WORKSPACE A NEW NAME,
 [2]    ⍝ FOR EXAMPLE  S3V7
 [3]    ⍝
 [4]     ⍎')WSID S',(⍕NUM),'V',1↓INITWORKSPACE
 [5]     DSCR←DSCR,' ,ISTART= ',1↓INITWORKSPACE
 [6]     DSCR←DSCR,'  *****'
     ∇


    ∇  DATUM;TS
[1]     TS←⎕TS
[2]     TIMESTAMP←(TWO TS[1]),'-',(TWO TS[2]),'-',(TWO TS[3]),' ',(TWO TS
      [4]),':',(TWO TS[5])
    ∇


    ∇  RESULT←TWO NUMBER
[1]    ⍝
[2]    ⍝ TO REPRESENT ANY NUMBER WITH TWO INTEGER PLACES.
[3]    ⍝ FRACTIONAL PART IS ROUNDED, BIGGER THAN 99 GETS TRUNCATED,
[4]    ⍝ AND SMALLER THAN 10 GETS LEADING ZEROES.
[5]    ⍝
[6]     ALWAYS '(NUMBER≥0)∧(0=⍴⍴NUMBER)'
[7]     RESULT←¯2↑'00',⍕⌊0.5+NUMBER
    ∇
    ∇  NEGΔIMSTO
[1]    ⍝ ALLOWS INPUT-GOODS INVENTORIES TO HAVE LEVELS BELOW ZERO
[2]     ENS 1=1↑⍴'PLANQREVISE' MODFNP 'SHORTAGE←0⌈⍵SHORTAGE←0×'
[3]     VARIANTS←VARIANTS,' NEG-IMSTO'
    ∇
```

```
    ∇  POSITIVEΔNETΔWORTH
[1]    ⍝ FROM 81-02-02 A COMBINATION WITH 'NULLIFYΔNEGΔNW'
[2]    ⍝
[3]    ⍝ TO MAKE SURE BORROWING DO NOT EXCEED ASSETS,
[4]    ⍝ AND TO , THOUGH IN QUITE A CRUDE WAY , ADJUST
[5]    ⍝ NEW-BORROWING TO THE DEBT/EQUITY-RATIO
[6]    ⍝ STEP 1: QΔBW ≤ REDCHBW(=.15)×BW
[7]    ⍝      2: QΔBW REDUCED IF 0.1<(NW/A)≤0.3
[8]    ⍝      3: QΔBW =0       IF 0.1≥(NW/A)
[9]    ⍝      4: FIRMS ARE NULLIFIED THE 6'TH QUARTER WITH   NW<0
[10]   ⍝
[11]    'INVFIN' MODADD 'QDESCHBW←(0⍵QDESCHBW[THO]←REDCHBW×BW[THO←(QDESCH
       BW>REDCHBW×BW)/⍳⍴BW]'
[12]    'INVFIN' MODADD 'QDESCHK2←(RW⍵QDESCHBW←(BWΔCHECK ((BW+QDESCHBW)÷(
       K1+(K2+QDESCHK2)+K3)))×QDESCHBW'
[13]   ⍝
[14]    'INVFINΔADJUSTMENTS' MODADD '⍝ NW IS⍵BAD←BAD+(NW<0)'
[15]    'INVFINΔADJUSTMENTS' MODADD 'BAD⍵REALLYΔBAD←BAD=6'
[16]    'INVFINΔADJUSTMENTS' MODADD 'REALLY⍵⍳(0<(+/REALLYΔBAD))/'' NULLIF
       YΔNW REALLYΔBAD '''
[17]   ⍝
[18]    'NULLIFY' MODADD 'SHRINK ''AMAN⍵SHRINK ''BAD'''
[19]    'NULLIFY' MODADD 'SHRINK ''QW''⍵SHRINK ''REALLYΔBAD'''
[20]   ⍝
[21]    VARIANTS←VARIANTS,'  POS-NET-WORTH-ELSE-NULLIFY '
[22]   ⍝
    ∇
```

```
     ∇  IND∆PURCH∆SHARES
[1]    ⍝ PURCHASING-SHARE INDIVIDUALIZED IN THE FOLLOWING WAY :
[2]    ⍝ I/O-MATRIX ENDOGENOUS IN VOLUME TERMS
[3]    ⍝ PURCHASING SHARE: (SUM I=1...10 IO[X I])÷(SUM I=1...13 IO[X I])
[4]    ⍝ PURCH.-SHARES ARE INDIVIDUAL FOR MKT X=1...4
[5]    ⍝ THE RELATION (IO[X I])÷(IO[X J]); I,J∊[1,10],I≠J; IS
[6]    ⍝ FIXED THOUGH........
[7]    ⍝ NOTE: IF FN. *ADDFIRM* SHOULD BE USED, CHANGE LINES
[8]    ⍝ 'Q←' AND 'QQ←'
[9]    ⍝
[10]   ⍝ FROM FRED OCT-80
[11]   ⍝
[12]    'TARG∆SEARCH' MODSUBST 'QEXPPNET←⍵QEXPPNET←QEXPP-SHARE×(QEXPPIM+.
       ×IO)[MARKET]'
[13]    'MAXIMSTO' MODSUBST 'R←(⍉IO⍵R←((⍉IO)[MARKET;] MULT7 SHARE)MULT7 R
       EFQIMSTO × IMBIG'
[14]    'MINIMSTO' MODSUBST 'R←(⍉IO)⍵R←((⍉IO)[MARKET;] MULT7 SHARE)MULT7
       REFQIMSTO × IMSMALL'
[15]    'OPTIMSTO' MODSUBST 'R←(⍉IO)⍵R←((⍉IO)[MARKET;] MULT7 SHARE)MULT7
       REFQIMSTO × IMSMALL+IMBETA×IMBIG-IMSMALL'
[16]    'PLANQREVISE' MODSUBST 'MAT←(⍉IO)⍵MAT←(⍉IO)[MARKET;] MULT7 SHARE'
[17]    'PLANQREVISE' MODSUBST 'IMSTO←⍵IMSTO←MAXIMSTO⌊IMSTO+QIMQ-MAT MULT
       7 QQ'
[18]    'PLANQREVISE' MODSUBST 'QIMQ←0⌈((⍉IO⍵QIMQ←0⌈(MAT MULT7 ⍉PLANQSAVE
       )+(OPTIMSTO-IMSTO)÷4×TMIMSTO'
[19]    'FINALQPQSQM' MODSUBST 'QVA←⍵QVA←QVA×1+QDVA←¯1+(QQ×QP-SHARE×((QPD
       OM×1-TXVA2)+.×IO)[MARKET])÷QVA'
[20]   ⍝ WE ALSO SHRINK VAR. *SHARE* IN NULLIFY
[21]   ⍝
[22]   VARIANTS←VARIANTS,'  INDIVIDUAL-PURCHASING-SHARES '
     ∇
```

Section 6 In Case of Trouble ...

Here are some tricky situations in connection with the running of the model:

1) Error during simulation. Not the model itself.

Use the system command

```
    ) SI.
```

You can then see in what function the simulation was interrupted.

If you are interrupted in a "print-out" function (not the model itself) you can usually continue the simulation by entering:[1]

```
    →[]LC+1
```

This means that you skip the line where you were interrupted. This can damage the printing out of results in some tables, but not the simulation itself.

2) Error during simulation. The model itself.

If you are interrupted in a function belonging to the model itself, this is usually due to one firm (out of 147) behaving in a perverse way (getting production volume less or equal to zero or something like that).

You must then either change some lines in the model or some lines in the initialization. Due to the character of the problem it may take anything from a couple of hours to weeks to correct this error,

----

[1] In front of LC should be an APL symbol which is an empty square. For typographical reasons this is written as [].

since one has to get well acquainted with the model code itself to understand why things have gone wrong. (A quick solution may be to nullify (= delete) this firm in the beginning of the simulation, but this is not the best solution.)

3) Error during initialization

Usually due to some technicalities when using the functions MODADD, MODSUBST, etc. See Section 3.

4) Editing an APL function

While making changes in an APL function one might sometimes wind up in a situation where it is impossible to get a new line number, when pressing the RETURN key. This comes about if one is writing an expression with ' in, and for some reason has omitted an aphostrophe. One must write both a left-hand apostrophe and a right-hand apostrophe to get a new line number.

Section 7 Standard Print-Out of Result

The result is, as mentioned previously, stored in a workspace called SXXVYY, where XX is the number of the experiment version, and YY is the number of the initialization version.

A function **PRINT** should be used to print out the result.

The result is a number of tables with different names.

To print out a table enter:

PRINT 'name of table'.

The names of the tables, available after a run, are shown if one performs the function call:

ALLREPORTS

Example:

We wish to print out the table called YEARLY∆INDUSTRY∆TOTAL.

Perform:

PRINT 'YEARLY∆INDUSTRY∆TOTAL'

You then get yearly performance of some main economic indicators.

This result is printed out on the screen and on a printer connected in series with the terminal, after each simulated year. Deleted firms (bankruptcy) are also printed out during simulation.

LITERATURE to Part 1

"The APL Reference Manual": For example: APLSF programmer's reference manual, DEC system 10, DEC-10-LPLSA-A-D, 1976.

Batch Reference Manual, DEC system 20, DEC-20-OBRMA-A-D, 1978.

Albrecht, J. - Lindberg,T., The Micro Initialization of MOSES, IUI Working Paper No. 72, Stockholm 1982.

Eliasson, G., A Micro-to-Macro Model of the Swedish Economy, IUI Conference Reports 1978:1, Stockholm 1978.

Eliasson, G., Experiments with Fiscal Policy Parameters on a Micro-to-Macro Model of the Swedish Economy, Reprint from Robert H. Haveman and Kevin Hollenbeck, eds., Microeconomic Simulation Models for Public Policy Analysis, Vol. 2, pp. 49-95, 1980.

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

Part 2

THE INITIALIZATION PROCESS

CONTENTS

* The Micro Data Base and the Micro initialization are also described in much detail in Albrecht-Lindberg (1982).

Preface

This paper describes the initialization process.[1] One can divide the **initialization process** into three stages: **data base work**, the **initialization procedure** and **consistency controls.**

A tremendous amount of information is needed to start up the model. The main reason for this is the fact that the model simulates the behaviour of the economy mainly by summing up the dynamic performance of individual firms (micro simulation). Each firm is described by about 100 variables. About 250[2] firms participate (in the present version of the model). There are also a large number of variables needed to describe the "macro sectors" in the model, e.g. the household sector, the Government sector, non-industrial sectors etc. The collection of **micro data** and **macro data** needed for the initialization has been a drawn out research project at IUI, where a number of people have been involved.[3] The **data base work** is documented by the Sections 3, 5, Appendix B and the Supplement.

Micro and macro data have to be transformed in several ways before they finally can be fed into the model. The **initialization procedure** is the name of this transformation process. In this procedure we also include the task of giving values to **parameters** affecting the behaviour of households and firms. The initialization procedure is documented by Sections 1, 2, 4 plus Appendices A, C and D. This has also been a lengthy research project at IUI with several people involved.[3] The Micro Data Base and the Micro Initialization are also described in much detail in Albrecht-Lindberg (1982). This is an <u>integrated</u> description of the way firms are modelled and how initial data are processed. The main emphasis in this handbook. Section 4, is to provide a brief guide for someone going through the computer code (Appendix C). Section 4 is also a necessary background in order to understand Sections 5 and 6, where consistency problems (mismatch between micro and macro data) are discussed.

Finally, one has to check that all variables in the model (for the initial year) are consistent and that micro in all senses add up to the macro totals. This **consistency checking** has been done by the author of this paper, and is described in the Sections 3 (subsection 3.2) and 6.

Part 1 of this handbook describes how to run the model, in a technical sense. To be able to make experiments with the model one has to know the initialization procedure in some detail. Thus the user of the model must be well acquainted with both Part 1 and Part 2 of the handbook. Section 7 is a bridge between Part 1 and Part 2 where some examples of simulation techniques are presented.

In conclusion, it should be pointed out that this paper has a twofold purpose. It is a documentation of the initialization process and an aid for the future users of the model.

Section 1 Introduction

The "micro-to-macro model" MOSES simulates the economy quarter by quarter from a given starting year. Before one can start a simulation of the Swedish economy with the model one has to initialize a vast number of variables. The starting year is either 1982 or 1976.[4] The reason for this is that we only have complete micro and macro data bases for these years.

"Initialization" means, mainly, that three kinds of variables are given values.

(1) Variables for 1982 (1976) needed to start up the model.
(2) Variables needed to determine the future of certain variables which get their values irrespective of what happens during the simulation, i.e. exogenous variables.
(3) Certain constants, some of which are parameters affecting the behaviour of firms, households and market mechanisms.

In what follows, the first kind of variables will be referred to as "**start-up variables**", the second will be referred to as "**exogenous time-series**" and the third will simply be called "**constants**". A constant which affects the behaviour of firms or households is called a **parameter**.

An example of a "start-up variable" is RU, the rate of unemployment 1982 (1976). An example of an "exogenous time-series" is the growth of the government employment in the model. There is a variable[5] called EXOΔREALCHLG, which is a vector (over time) containing the number of people to be added to the government sector each quarter. An example of a "constant" is SMT which is a factor determining to what extent profit targets are updated with recent development during the simulation. More precisely:[6]

Targ(n+1) gets the value Targ(n) • SMT + M(n) • (1-SMT)

where

Targ(n+1) = profit target, quarter n+1
Targ(n) = profit target, quarter n
M(n) = actual profit, quarter n

The three mentioned kinds of variables can be **micro variables** or **macro variables**.

A "**micro variable**" is a variable which is connected with firms. Such variables are often vectors. A micro variable can be some characteristic of the firms (for example the value added share), a behaviour parameter (for example SMT above) or a variable which the firm can influence (for example L below).

Example:
L is the labour force (number of people) in each firm.
L(n) is the labour force in firm n.
n = 1,2,3,...,147 for the present.

The length of the vector is equal to the number of firms participating in the simulation. A micro variable can also be a constant, equal for all firms (a scalar). The constant SMT, mentioned above, determines the way profit target changes in each firm between any two quarters, and is an example of such a micro variable.

Typical macro variables are (for example) the rate of unemployment, the growth of the government-sector and tax-rates.

Certain macro variables apply to macro-entities but are used as micro variables as well. Such variables obviously lie somewhere between the two categories micro and macro.

An example of this is the variable IO, the input-output matrix.

IO(i,j) tells how much of production in sector j comes from input from sector i, and is a number between 0 and 1.

During the initialization IO gets the true values from real data for the economy for 1982 (1976). Throughout the whole simulation these shares are used (cf. Section 3) to determine each firm's demand from other sectors.

An alphabetical list of all variables (about 200) coming out from the initialization can be found in Appendix A. An alphabetical list of all variables (about 400) in the model itself can be found in Eliasson-Heiman-Olavi (1978).

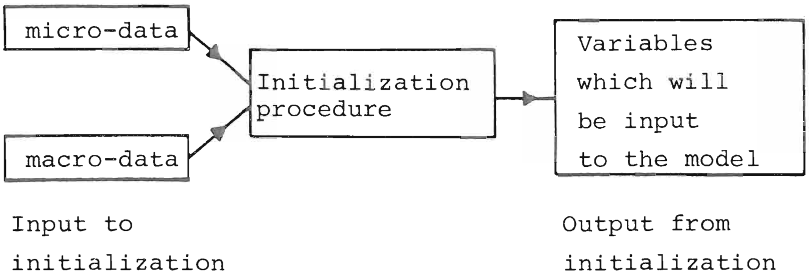
The main features of the initialization procedure are described in the next section. In Section 3 the input-output system is described, since it is an essential part of the initialization procedure and the data base work. The input-output system is described in rather much detail. The techniques involved are of general interest for builders of large scale simulation models of this kind. Section 4 presents the initialization procedure in more detail. Sections 5 and 6 are devoted to a documentation of the data bases and the consistency control system. Section 7 is of interest for users of the model.

Section 2 The Initialization, Main Features

The "initialization procedure" is a matter of converting raw-data (micro and macro data) to the variables, mentioned in the previous section, needed for the model-simulation.

Schematically:

Figure 1


micro-data
macro-data
Initialization procedure
Variables which will be input to the model
Input to initialization
Output from initialization

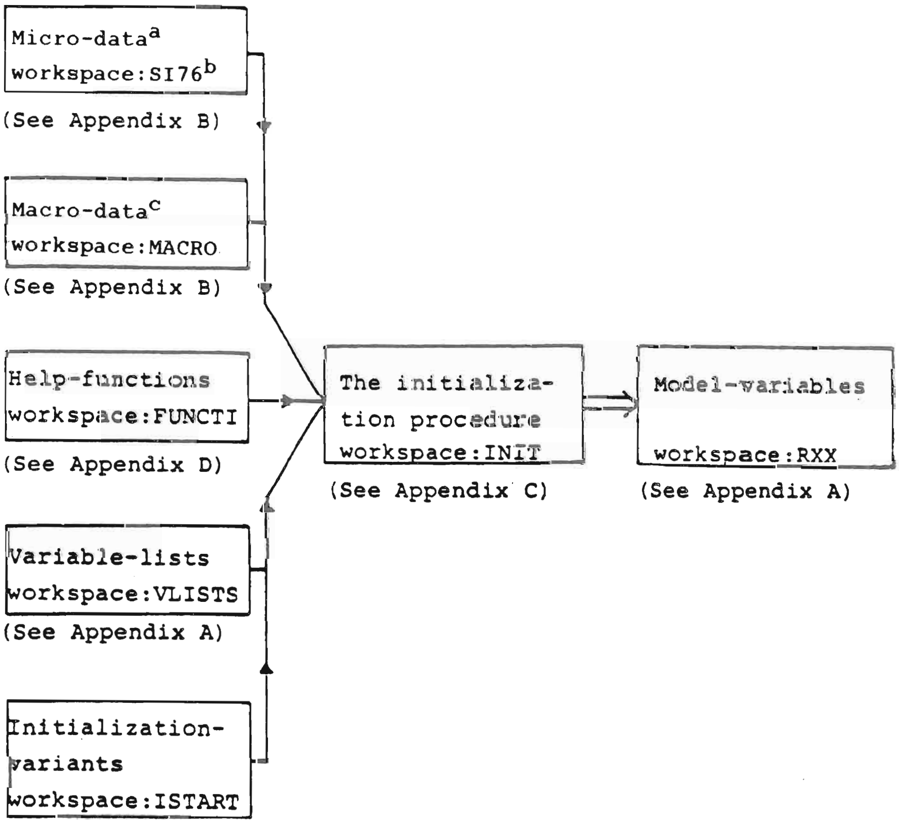
We will refer to "variables which will be input to the model" as "**model variables**", in what follows.

Micro data, which is data for real firms in the model, are stored in one work-space and macro data in another. Micro variables and macro variables are formed from micro data and macro data.

A more detailed schematic overview of the initialization procedure is shown in Figures 2 and 3. The whole initialization program was rewritten between 1980 and 1981 by the author of this paper. The logical structure of the initialization and the consistency checking were done during this period. An important addition was a part of the program called **OUTPUTΔOPERATIONS**, where the **model vari-**

ables were sifted from other data. Previously everything - raw data, intermediate variables and model variables - came out together after initialization. This meant that output from the initialization was "hidden" among a lot of redundant data. The new initialization procedure has notably facilitated practical work with the model.

Figure 2 The initialization procedure, if the starting year is 1976, input and output



[a] for 1982 use SI82 instead.

[b] SI stands for the "Federation of Swedish Industries", which collected the micro data, through the so-called Planning Survey. Reference persons: Ola Virin, Kerstin Wallmark

[c] for 1982 use MACR82 instead.

The contents of each workspace in Figure 2 will be described below. The initialization procedure is written in the programming language APL. In APL both variables and functions are stored in so-called workspaces which can be immediately transferred to computer-memory, by aid of certain system commands (cf. Part 1 of this handbook). **We write APL-functions in boldface letters** in what follows, but not workspace names.

The program for the initialization procedure lies in a workspace called INIT. This program consists of a main-function **START** and a number of part, so-called sub-functions. Figure 3 shows the structure of the initialization program, in workspace INIT.

Figure 3 The initialization program

Main program	Sub-functions, level 1[a]	Sub-functions, level 2[a]
START	**ISTARTXX**[b]	**TAXΔPARAMETERS**
	SIΔINIT	**PUBLICΔSECTOR**
		MONETARY
		MARKETS
		HOUSEHOLDS
		ESTABLISHMENTS
		DISPOSEΔVARΔINPUT
		MARKETSΔDATA
		SECONDARYΔDATA
		PUBLICΔDATA
		MONETARYΔDATA
		HOUSEHOLDSΔDATA
		OUTPUTΔOPERATIONS

[a] The greek letter "delta", Δ, is used in function-names in the APL-code instead of blanks, if the function-name consists of several words. Thus **PUBLICΔSECTOR** should be read "public sector" etc.

[b] XX in **ISTARTXX** stands for a number indicating different initialization variants.

The initialization is, as seen from Figure 3, divided into parts (sub-functions, level 2) according to the type of the variable.

Variables connected with taxes are set in **TAXΔPARAMETERS**. Variables connected with the government sector are set in **PUBLICΔSECTOR**. Variables connected with individual firms (micro variables) are given values in the function **ESTABLISHMENTS**, etc.

SIΔINIT (sub-function, level 1) calls all the sub-functions at level 2, and does some administration.[7]

Let us now turn to Figure 2 again.

Macro data are fetched from workspace MACRO in the beginning of **SIΔINIT** and **micro data** are fetched from workspace SI76 in the beginning of the sub-function **ESTABLISHMENTS**.

Help-functions for different applications are fetched from workspace FUNCTI.

To know the names of the model-variables **variable lists** are fetched from workspace VLISTS. In the sub-function **OUTPUTΔOPERATIONS** the variables mentioned in these variable lists are saved in a workspace RXX and the rest are **deleted**. XX stands for a number given by the user, which refers to the number of the initialization variant. The user gives this number when starting the initialization, by the call[8] **START** XX.

If one wishes to make an **initialization variant**, one makes a function **ISTARTXX** and stores this function in the workspace ISTART. The main-function **START** calls **ISTARTXX** before calling **SIAINIT**. (See sub-functions, level 1 in Figure 3.) How to make **ISTARTXX**-functions and initialization variants is described in Part 1 of this handbook.

2.1 Summary

What the user should particularly bear in mind is this: **New macro data** should be added to workspace MACRO. **New micro data** should be added to workspace SI76. Micro data are mainly used in the part of the initialization program called **ESTABLISHMENTS**.

To make **initialization variants**, use workspace ISTART and check the instructions in Part 1 of the handbook. As soon as **new model variables** are used, add the names of these in the variable lists in workspace VLISTS according to the instructions in Part 1. (If you forget this, the new variables will be **deleted**!)

The result from the initialization (= the model variables) winds up in a workspace RXX, where XX is the number used in the call "**START** XX", which starts the initialization.

A more detailed description of the initialization program will be presented in Section 4.

Section 3 The Input-Output System*

It is worthwhile knowing more about the input-output system in the initialization and in the model for three reasons:

a) Among macro data (input to initialization, workspace MACRO) there is an input-output matrix for the Swedish economy for 1976, called IO76. This matrix is used to give many of the model variables (output from initialization) their values. We describe this in Section 3.1.

b) To check up the consistency of the whole initialization the input-output system is used. We describe this in Section 3.2.

c) To be able to understand how the input-output system is used in the model, one has to know more about the model-variable IO, which is a matrix of input-output coefficients constructed from IO76. We describe this in Section 3.3.

The input-output system can be described as a matrix with 14 rows and 21 columns. This matrix, IO76, stored in workspace MACRO, has the structure shown in Table 1. The economy is divided into 10 sectors of production (=the first 10 rows and columns) and a number of final demand categories (columns 11, 12...). The first 4 sectors are inhabited by individual firms after the initialization.

Let us first turn our attention to the first 10 columns. This part of the matrix shows the product-flows between the 10 sectors and the value added in each sector. For example:

* This discussion is based on data for 1976.

Table 1 Input-output matrix (IO76) for the Swedish economy 1976

(Unit: Million of SEK in 1975 year's prices)
Explanations for column- and row-numbers, see next page

PRODUCTION MATRIX
Row 1,2,.....14 Column 1,2.....10

	1	2	3	4	5	6	7	8	9	10
1	5272	2890	5869	1321	245	94	0	4192	942	1943
2	2029	5195	4805	4465	908	117	0	3498	170	2035
3	954	2354	12296	915	503	213	0	6294	171	3079
4	803	2428	2041	14872	2078	87	0	2648	102	6484
5	2400	1964	341	10768	383	1	0	1689	0	312
6	2951	210	79	63	26	140	0	418	5	0
7	4136	600	151	261	130	213	0	1009	488	842
8	1235	1198	2838	961	1383	162	0	10928	708	9874
9	904	941	475	485	238	171	0	1118	328	1010
10	3293	3338	5919	4402	1792	640	0	9143	426	25656
11	63	71	142	-2377	163	10	0	350	5	2261
12	8736	14351	27422	19551	11452	2529	0	50892	6395	64383
13	154	-119	178	51	41	35	0	238	0	1
14	32933	35423	62558	55738	19341	4413	0	92417	9738	117881

FINAL DEMAND MATRIX
Row 1,2.....14 Column 11,12.....21

	11	12	13	14	15	16	17	18	19	20	21
1	758	5399	0	0	0	380	2754	12137	-11478	214	32933
2	1953	9075	558	0	869	2170	1135	14735	-12965	-5329	35423
3	3522	14903	3110	0	4836	10231	1687	29947	-24563	-7896	62558
4	5102	55944	112	0	175	132	752	7450	-15980	-29493	55738
5	243	6807	0	0	128	408	-95	1351	-3597	-3763	19341
6	81	24	0	0	0	0	67	1134	-3015	2230	4413
7	374	2346	0	0	0	0	188	1778	-6491	-6025	0
8	2929	26970	17893	12436	4682	765	1067	7062	-4453	-6221	92417
9	973	3580	0	0	0	0	-76	319	-306	-421	9738
10	8849	30617	379	0	591	0	-316	10370	-16362	29496	117881
11	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0
14	24785	155664	22052	12436	11281	14085	7163	86284	-99209	-27209	430440

Source: Louise Ahlström, SAF. See also the Supplement.

Table 1 (cont)

Row 1:	Raw material sector
Row 2:	Intermediate goods
Row 3:	Investment goods and consumer durable goods
Row 4:	Consumption goods (excl. consumer durable goods)
Row 5:	Agriculture, forestry, fishing
Row 6:	Mining and quarrying
Row 7:	Oil
Row 8:	Construction
Row 9:	Electricity
Row 10:	Other services
Row 11:	Commodity based indirect taxes (Not value added tax (moms))
Row 12:	Value added in producer's prices
Row 13:	Corrections
Row 14	Vertical sum = production (producer's prices)
Column1,2 through 10	Corresponding rows
Column 11:	Government's consumption
Column 12:	Household's consumption
Column 13:	Government's investments
Column 14:	Investments, buildings, residential housing
Column 15:	Investments in sector 5.....10
Column 16:	Other investments (= Investments made by firms)
Column 17:	Change in stock (inventories)
Column 18:	Exports
Column 19:	Imports
Column 20:	Moms etc. (Indirect taxes) are deducted
Column 21:	Horizontal sum = production (producer's prices)

Note : Wages in the Government Sector are not included in the Final Demand Matrix, so GNP cannot be calculated from Table 1 alone.

Column 1: IO76(m,1), m=1,2,...,10, shows how much sector 1 is buying from the other 10 sectors.

NOTE: Imports are included.

Rows 11 and 13 consist of rather small values and are described further in the Supplement. Row 12 is the value added in each production sector.

Value added (row 12) + Inputs (rows 1,2...10) equals total production in each sector (row 14).

Columns 11, 12 and onwards show the final demand-side in the economy.

NOTE: Imports are included.

The division into demand categories follows the usual pattern from national accounting where gross national product is described as C+I+G+X-M+ΔL. (C=household's consumption, G=government spending, I=investments, X=exports, M=imports and ΔL=change in stock.) Investments have been divided somewhat more, though.

The vertical sum of production (row 14, column 1,2...10) shall by definition be the same as the horizontal sum (rows 1,2...10; column 21). Column 20 has to be present to make this work. Columns 11,12... are defined in final prices including indirect taxes, value added taxes (VAT), whereas production in columns 1,2...10 are defined without these taxes. Thus, these taxes (VAT)* are subtracted in column 20 to make "vertical sum of production" match "horizontal sum of production". Some

* In Swedish called MOMS.

other adjustments of a technical nature are also made in column 20. For a more detailed description, see the Supplement to this handbook.

3.1 How the Input-Output Matrix IO76 is Used in the Initialization

IO76 is a matrix with flows in SEK (Swedish crowns) These flows are, in general, not used directly to give values to model variables during the initialization procedure. In 95 % of all cases shares, fractions, based on IO76, are used for this purpose. These shares are called IOCOEFF76 and are defined as:

$$\frac{IO76(i,j)}{IO76(14,j)} = IOCOEFF76(i,j)$$

$i=1,2,3,\ldots,13 \quad j=1,2,\ldots,19.$

The coefficient matrix IOCOEFF76 can be found in Appendix B.

The following **model-variables**, shown in Figure 4, (cf. Appendix A) get their values from the "input-output coefficient matrix" IOCOEFF76.

The first six model variables, in Figure 4, are constants throughout the simulation. The input-output matrix IO will be described more thoroughly below, in Section 3.3. QINVG, QINVBLD and QINVIN are start-up variables for the corresponding[9] exogenous time series, concerning non-industrial investments.

Export and import shares XIN and IMP are estimated from IO76.

Figure 4 Model variables created from IO76

Model variable	Coming from:	
OMEGA	Column 16	IOCOEFF76
OMEGAIN	Column 15	IOCOEFF76
OMEGABLD	Column 14	IOCOEFF76
OMEGAG	Column 13	IOCOEFF76
GKOFF	Column 11	IOCOEFF76
HH76 (household coefficients)	Column 12	IOCOEFF76
IO (input-output matrix)	Columns 1,2,...,10, rows 1,2,...,10	IOCOEFF76
IO2 (submatrix of IO)		IOCOEFF76
IO3 (submatrix of IO)		IOCOEFF76
QINVG		IO76(14,13)
QINVBLD		IO76(14,14)
QINVIN		IO76(14,15)
IMP (import shares)	Estimated from	IO76
XIN (export shares)	Estimated from	IO76

XIN is the export share of production in non-industrial sectors 5,6,7,...10, and is estimated as: IO76(5,18)/IO76(14,5) etc. This is export shares for sectors which are modelled as macro units. Export shares, called X, for individual firms in the model come from micro data.

IMP is the import share of Swedish demand and is estimated as:

$$\frac{IO76(i,19)}{IO76(i,21)-IO76(i,20)-IO76(i,19)-IO76(i,18)}$$

where i=1,2,...,10, and column 19 consists of negative numbers (cf. Table 1).

The same import shares apply to both consumers and firms. We lack information about individual firms' import shares. Hence import shares IMP refer to markets, in contrast to export shares X which refer to individual firms. Thus the macro shares are used for the individual firms in the import block of the model. In Appendix A one can see that IMP is classified as a micro variable for sectors 1, 2, 3, 4 and as a macro variable for the remaining sectors.

3.2 Consistency checking

For the purpose of checking the consistency of the initialization one would expect that IO76 should be used. This is, however, only the case to some extent. In principle IO76 can not be used since it is expressed in 1975 year's prices instead of 1976 year's prices. All model-variables coming out from the initialization should be in current prices, i.e. 1976 year's prices. This makes a direct comparison between IO76 and the input-output matrix coming out from the initialization a bit difficult.

Even if one managed to express IO76 in 1976 year's prices it would, all the same, be almost practically impossible to check the consistency of the initialization just by direct comparison with IO76. The explanation is as follows:

After the initialization the four industrial sectors (columns 1,2,3,4 in IO76) are inhabited with firms. S and Q are important firm-variables. S is individual firm sales and Q is individual firm production.

To determine the sum of S in each of the four sectors one must use SCB's national accounting statistics. Q is by definition equal to S minus changes in finished goods inventories. This also determines the sum of Q in each of the four sectors (approximately) and these figures of the production (in sectors 1,2,3,4) may differ substantially from figures from IO76 (row 14 IO76, reestimated in 1976 year's prices) due to errors[10] of different kinds.

The consistency of the initialization is instead tested as follows:

a) Form a matrix $IO76_{II}$ from the initialization by using the sum of micro-variables (for example Q above) when this is possible, and fill in with values from IO76 when this is not possible.

b) $IO76_{II}$ is considered consistent if (1) the values in $IO76_{II}$ don't differ "unreasonably much" from IO76 and (2) horizontal sum of production ≈ vertical sum of production in $IO76_{II}$.

For more details about the consistency check, see Section 6.

3.3 How the Model-Variable IO is Used

We now give a short description of how the model-variable IO, which is a 10x10 sub-matrix of IOCOEFF76 (the input-output coefficients), is used in the model.

The variable IO, with some exceptions[11], is **not** used for the purpose of determining macro variables during the simulation.

IO(i,j) tells how much of production in sector j comes from input from sector i, and is a number between 0 and 1, and i=1,2,...,10. Thus IO(1,j), IO(2,j), IO(3,j),...,IO(10,j) are the input-**shares** for each product (input from sectors 1,2,...,10) in sector j. The firms belong to sector 1, 2, 3 or 4.

The main use of the input-output matrix during the simulation is to determine **each firm's** demand for goods from other sectors. Thus, a firm in sector j producing q SEK (Swedish crowns) a certain quarter demands IO(1,j) • q • c SEK production from sector 1 and IO(2,j) • q • c SEK production from sector 2, etc.

$$c = \frac{\text{the individual firm's input-share}}{\text{average input-share in the sector}}$$

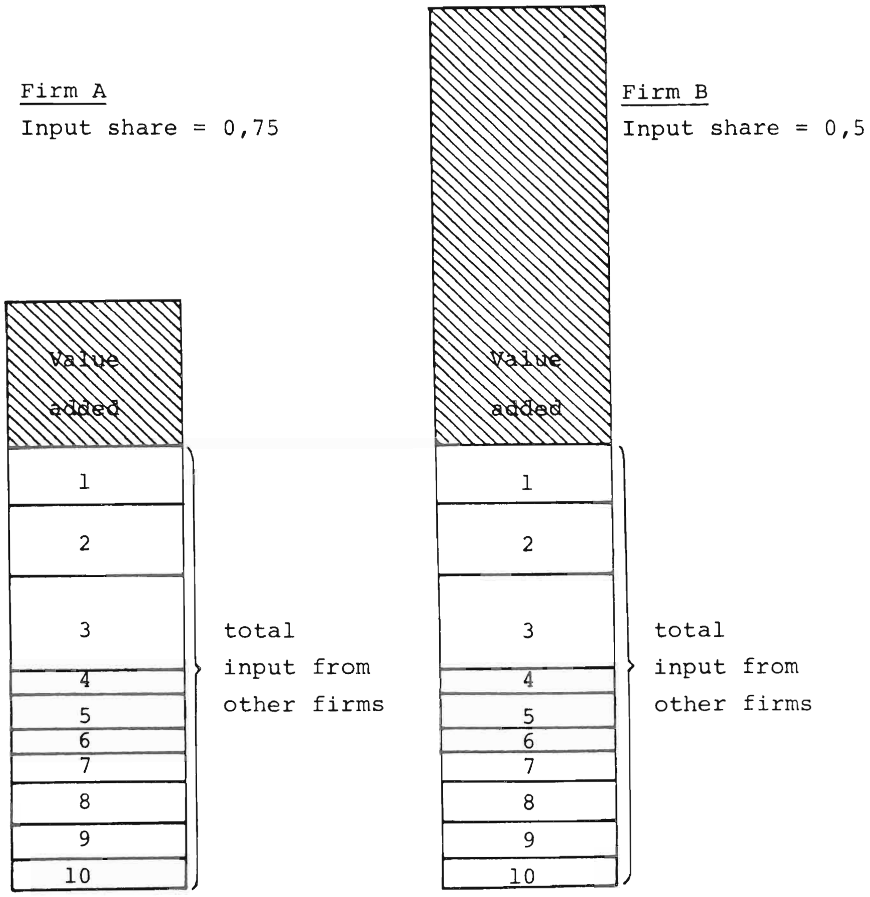
$$\text{The average input-share} = \sum_{i=1}^{10} IO(i,j)$$

The fractions c are only estimated for 1976 and are called SHARE in the initialization procedure.

Note that this specification means that the **macro** input-output coefficients are **variable** over time during a simulation. Since firms have **individual** input-shares (see c above) and firms grow at different rates, the macro input-output coefficients **vary endogenously** although the coefficients IO are **constant** over time and **exogenous**.

This can be clarified by Figure 5.

Figure 5 The production in individual firms



1 = input from sector 1
2 = input from sector 2
3 = input from sector 3
4 = input from sector 4
5 = input from sector 5
6 = input from sector 6
7 = input from sector 7
8 = input from sector 8
9 = input from sector 9
10 = input from sector 10

Production = Total input+value added

Firms A and B have different **individual input shares**. The individual input share is equal to 0.75 for firm A, whereas it is 0.5 for firm B. Information about such shares come from micro data. Thus the value added share is 0.25 and 0.5 respectively. How the inputs are divided onto the 10 sectors is determined from the input output matrix (the variable IO). These proportions are the same for all firms, which means that the quotient **(input from sector j)/(total input)** is the same, but not the quotient **(input from sector j)/(production)**.

The **macro input shares**[12] will in general change if the relative size of the firms changes from one year to another during the simulation.

Let us give a concrete example of this. From a simulation performed in 1983[13] the following macro input shares were obtained:

Table 2 Simulation results

Year:	Macro input share (=total input/total production) Sector 1	Sector 2	Sector 3	Sector 4
1976 (real values)	0.73	0.60	0.56	0.69
1977 (simulated values)	0.72	0.62	0.55	0.69
1978 (simulated values)	0.73	0.61	0.52	0.67
1979 (simulated values)	0.73	0.56	0.49	0.66
1980 (simulated values)	0.70	0.52	0.48	0.65

This illustrates some kind of structural change in the four industrial sectors in the simulated economy. In principle one could describe this structural change by investigating the chains of causes at the macro - and the micro - level during the simulation.

Section 4 The Initialization, Overview

This section will give a more detailed description of the initialization program. Those who wish a complete description of the program may turn to the programming code itself in Appendix C and use this section as a guide. The techniques involved are of general interest for someone wishing to construct a micro-to-macro simulation model.

As was shown in Section 2, Figure 3, the initialization essentially consists of 13 parts (subfunctions, level 2 in Figure 3). 11 of these parts deal with giving values to model-variables, namely: **TAX∆PARAMETERS**, **PUBLIC∆SECTOR**, **MONETARY**, **MARKETS**, **HOUSEHOLDS**, **ESTABLISHMENTS**, **MARKETS∆DATA**, **SECONDARY∆DATA**, **HOUSEHOLDS∆DATA**, **MONETARY∆DATA** and **PUBLIC∆DATA**.

In the programming code, Appendix C, comment lines start with the symbol **A***. Such comment lines are inserted to make the program easier to understand. In the beginning of each sub-function there is a comment line beginning with "output from initialization". Thereafter follows a list of the names of those model-variables which have been given values in that particular subfunction. This is an important guide to the reader of the program, because he then knows what is to be considered as output from the sub-function. Other variables in the sub-function are either local variables (help-variables used to form the model-variables) or indata from the macro-data workspace or the micro-data workspace. The macro-data workspace and

* This symbol looks like an A, but is smoother than an ordinary A. For typographical reasons we write this as a boldface A, in this text.

the (non-confidential part of the) micro-data workspace are listed in Appendix B.

An alphabetical list of all model-variables can be found in Appendix A. Now we will comment on each of the 11 parts of the initialization program.

o **TAXΔPARAMETERS**

"Start-up tax variables" (=tax last quarter 1976) are transferred directly from workspace MACRO. These variables are TXVA1,TXVA2.

The rest of the tax-variables in this part of the program are "exogenous time-series" which are formed by extending time-series for the period 1976 and onwards from workspace MACRO.

o **PUBLICΔSECTOR**

Some model-variables (OMEGAG, QINVG, GKOFF), mentioned in Section 3, get their values from the input-output system.

The number of people to be added to the government-sector (EXOΔREALCHLG) each quarter during the simulation is an exogenous time-series, and is determined as follows:

a) Quarterly labour force in the government sector is estimated from time-series data (1976-), TIMΔOFF, from workspace MACRO. For the present this determines EXOΔREALCHLG for the first 4 quarters of the model simulation.

b) A trend change of the government sector growth, estimated from historical time-series during the

1970s (the trend change LGTRENDCH comes from workspace MACRO), is used for the remaining quarters in EXO∆REALCHLG vector.

Wages in the government sector, the model variables QWG and WG, are determined from wage data in workspace MACRO for 1976-77 (LON∆OFF).

○ **MARKETS**

Most of the model variables mentioned in Section 3.1 get their values in this part of the initialization program. It is variables connected with the input-output system, for example input-output coefficients of various kinds (="constants"), investments in different sectors (="start-up variables"), import shares of Swedish demand (=IMP ="start up variable") and export shares of production in certain sectors (=XIN="constant").

Model variables starting with "EXO" are "exogenous time-series".

The important model variable EXO∆QDPFOR (=changes in foreign price index) is set using historical price-behaviour (extrapolation). EXO∆QDPFOR is a matrix with the format "4 x number of quarters in the simulation" because it yields foreign price changes in each of the 4 industrial sectors.

All price-indices are equal to **100** for the base year (1976).

○ **HOUSEHOLDS** and **MONETARY**

Household coefficients HH76, i.e. how the consumers distribute their purchases on products from the 10 sectors in 1976, are set. These coefficients are used later on in the initialization procedure (cf. **HOUSEHOLDSΔDATA** below). Some exogenous time-series in connection with the rate of interest (EXOΔRI and others) in the bank-system are set.

o **ESTABLISHMENTS**

This is the first time micro variables are given values. Real firms are given their values, and the residuals on each variable are split up on synthetic firms. By residuals we mean deviations from the national accounting level, 1976. For the present, 1983, we have 97 real firms and 50 synthetic firms. The synthetic firms have been created to be able to model the **whole** industrial sector by a **micro** simulation process in MOSES.

Only the 4 industrial sectors consist of micro-units, i.e. firms in the model. Micro-data are fetched from workspace SI76 (the first line in this sub-function).

Firm-data from this workspace are mainly stored in two variables: X and FΔDATA. X is a matrix where the first index is the firm index and the second is the number of the variable. For example: X(17;7) is export sales (question number 7 in the questionnaire) for the 17th firm.

The function **ESTABLISHMENTS** is rather complicated and only the main features will be described here.

Let us look at the variable sales, to get a picture of how the initialization of this variable is performed. A similar pattern can be found for many other micro variables.

(STEP 1) Real∆sales (=help variable) is a vector with R components, where R=number of real firms. Real∆sales(i) gets the value:

$$[\underset{\substack{\text{export-}\\ \text{sales}}}{\underline{X}(i,7)} + \underset{\substack{\text{domestic}\\ \text{sales}}}{\underline{X}(i,12)}] \cdot 10^{6}.$$

i = firm-index = 1,2,3...R

The rest of the sales value in each of the 4 industrial sectors is split up on the <u>synthetic firms</u>.

(STEP 2) Res∆sales (=help variable) is a vector of length 4 and is the rest of the sales value in the 4 sectors. RES∆sales(j) gets the value:

$$\text{SALES76}(j) - \sum_{i=1\ \text{and } i \text{ belongs to } j}^{R} (\text{Real}\Delta\text{sales}(i))$$

j=1,2,3,4=sector-index

SALES76(j) is sales for 1976 in each of the 4 sectors, fetched from SCB national accounting statistics. "i belongs to j" means summation of those real firms (i=1,2,...,R) which belong to sector j.

(STEP 3) Synth∆sales(i) gets the value Scale(i) • res∆sales(M(i)).

M(i) is the sector to which firm 'i' belongs. i=1,2,3,...,Q Q=number of synthetic firms, M(i)=1 or 2 or 3 or 4, R=number of real firms.

$$\sum_{i=1}^{Q} \text{Scale}(i) = 1$$

and i belongs to j

Scale is a vector with sizes (fractions), within a sector.

(STEP 4) The model-variable S (= individual firm sales) gets the values: S(i) = RealΔsales(i) for i=1,2,...,R and S(i) = SynthΔsales(i) for i=R+1,R+2,...,R+Q.

Thus:

STEP 1: RealΔsales(=sales for real firms) is set.

STEP 2: ResΔsales(=residuals between macro and sum of real firms) are set.

STEP 3: ResΔsales is distributed onto synthetic firms. SynthΔsales(=sales for synthetic firms) is set.

STEP 4: S(=sales=model variable) is the combination of RealΔsales and SynthΔsales.

This 4-step procedure is repeated for many other micro variables. Thus, W(wage-level in firms), L(labour-force in firms), X(export shares in firms) etc are set in much the same fashion.

In connection with "synthetic firm initialization" there are two other important technical points. Namely:

o (a) As soon as ratios appear, an inbuilt check insures that the synthetic firms get the same mean and dispersion (standard deviation) as the real firms.

o (b) Certain variables ought to co-vary with other variables in the synthetic firms, and this is also taken into account.

Example:
L(i)=labor in each firm R=number of real firms
S(i)=sales in each firm Q=number of synthetic firms
i=1,2,...,R+Q.
ratio=L(i)/S(i)

This ratio is randomized for the synthetic firms in such a fashion that the mean and dispersion for the synthetic firms (i=R+1,R+2,...,R+Q) are the same, as that of the real firms. (Actually, it is a bit more complicated than this, since each sector (1,2,3,4) is treated independently.)

The export share for each firm (an important model-variable), X, is set in a similar manner.

Jim Albrecht, Columbia University, has made these randomization procedures in **ESTABLISHMENTS**.[14]

Production for each firm Q(i) is estimated as

$$Q(i) = (S(i)+\Delta K3\Delta FINISH(i))/100$$

where $\Delta K3\Delta FINISH$ is the change in the finished goods stock (a help variable) and 100=price index (the index equals 100 by definition 1976). Thus production in both synthetic and real firms is set indirectly, that is, by aid of sales figures and changes in finished goods stocks.

Each firm in the model has an individual input share (input/production), which is estimated from micro-data. Thus the model variable Share(i) is created:

$$\frac{\text{the individual firm's input share}}{\text{average input share in the sector}}\ .$$

Section 3.3 describes how this share is used, during the simulation.

- **MARKETSΔDATA**

Most of the constants in the model (mainly **parameters**) are set in this part of the initialization program. Constants connected with the banks system and the household sector ar not set here, though. These constants (parameters) are instead created in **HOUSEHOLDSΔDATA** and **MONETARYΔDATA**. Inventory-constants for firms (maximum- and minimum-inventory levels) are set in **ESTABLISHMENTS**.

- **MONETARYΔDATA**

Constants connected with the bank system are set. If RIΔISΔEXOGENOUS=1[15] then the bank system is partly set out of function, since the rate of interest in the economy is set exogenously in this case. In that case most of the other constants in this part become redundant. This is the case for the present (1983), since the bank system is not quite ready yet. Even when this module is ready it is of interest to be able to, for analytical simplicity, perform simulations with an exogenous rate of interest.

- **SECONDARYΔDATA** and **PUBLICΔDATA**

Certain labour market variables are given values, for example LU, the number of unemployed during the last quarter 1976. MTEC, a constant describing "the production function" for firms in each market, is set.

○ **HOUSEHOLDSΔDATA**

Constants connected with the household part of the model are set here. For example; the coefficient-vectors BETA1, BETA2 (cf. Appendix A) are given values. BETA1 tells how much consumers tend to stick to historical consumption levels during the simulation and BETA2 are marginal propensities to consume when disposable income varies. Consumption levels last quarter 1976 are set.

QC(j) = HH76(j) • QDI
j=1,2,...,10=sector index.

QC=consumption, QDI=disposable income, HH76=input-output shares (see Section 3).

QDI is estimated in a certain function which takes into account the whole tax system, wage system etc. This is done in the function **QDIΔINIT**.

Section 5 The Data Base

The macro data for 1976 for the initialization come from workspace MACRO and the micro data from workspace SI76, see Appendix B. If we instead use the 1982 data bases we take macro data from MACR82, documented in Appendix B2, and micro data from SI82.

Below, there is a brief documentation of the variables appearing in these two workspaces.

(MACR82 and SI82 contain the same variables as MACRO and SI, i.e. we have a standardized format.)

5.1 Workspace MACRO

In general, most of the variables refer to 1976 or 1976 and a couple of years ahead (to form exogenous time series). The only exceptions are:

	Period
TLΔEXP (export price indices)	1970-80
IMPLΔPRIS, IMPLΔPRISΔIN (domestic price-indices)	1974-77
HISTΔTXVA2("moms")	1974-77

TLΔEXP is a long time series which is used to extrapolate a` future time series starting 1977, i.e. the variable EXOΔQDPFOR mentioned previously. IMPLΔPRIS etc are a bit longer to be able to transform data on a quarterly basis for 1976, 1977. The values for 1974 are redundant, though.

Sources:
Reference person for all variables except SALES76, TLΔEXP, LON and TIM: Louise Ahlström (previously IUI).

The national accounting statistics from SCB has been used. Reference persons for SALES76, TLΔEXP. LON, TIM: Thomas Lindberg, Fredrik Bergholm, IUI.

5.1.1 The Problem of Distributing Macro Data

There is a general problem of a practical nature in connection with the three variables LON(=total wage sum in sector 1,2,3,4), TIM(=total number of working hours in sector 1,2,3,4) and SALES76(=total sales in sector 1,2,3,4). LON, TIM and SALES76 are used for micro initialization, as was mentioned in Section 4. They are the macro totals for model variables like labour L and sales S.

The problem is that from SCB-figures we have
a) total wage sum in the industry
b) total number of working hours in the industry
c) total sales in the industry.

When a), b) and c) are distributed across the 4 sectors (1-4 in the input-output system) we get the variables LON, TIM and SALES76. There is a so-called "weighting matrix" which has been constructed to do this job. However, the result seems to be a bit unsatisfactory. In the consistency check (Section 6) we find residuals indicating that sector 1 is <u>too small</u> and sector 3 and (or) 4 are <u>too large</u>. A consequence of this is that synthetic firms in sector 1 get input shares FΔINKOP=the quotient input/production, see Appendix A) <u>larger than 1</u>. The behaviour of these companies disturb the simulation during the first three to four years in quite a conspicuous manner.

Apparently this problem is a crucial one to be overcome in order that a proper initialization may be obtained. In 1983 some measures were taken to improve matters. Of course there can be many reasons for the inconsistencies.

However, the distribution process clearly yields different results compared with the figures in the input-output system IO76 in Section 3. This can be seen as follows:

Total sales in the 4 (industrial) sectors 1976 is 207 150 million Swedish crowns. SALES76 is a vector with four components where this amount has been distributed onto the 4 sectors by aid of the weighting matrix mentioned above. The following result is then obtained:

SALES76 = (0.14, 0.18, 0.34, 0.34) • 207 150

On the other hand, if one distributes total sales according to the proportions for gross production (assuming that sales ≈ production and thus neglecting changes in finished goods inventories) in the input-output system IO76 (cf. Table 1, row 14, columns 1 through 4) the following result is obtained:

NYSALES76 = (0.18, 0.19, 0.33, 0.30) • 207 150

In 1983 we started using NYSALES76 instead of SALES76 in the initialization procedure (initialization variant ISTART10). This reduced the inconsistencies in the initialization (cf. Section 6).

Future work in connection with the variables LON, TIM and SALES76 should be directed towards obtaining more precise distribution procedures, which at the same time are reasonably consistent with the input-output system.

5.1.2 Changes in the Input-Output Matrix

If the input-output matrix IO76 is changed (corrected) the function **COEFFΔIO** has to be executed to get new input-output coefficients IOCOEFF76.

5.2 Workspace SI76

A good description of this workspace can be found in Albrecht-Lindberg (1982). Sources: Reference persons: Thomas Lindberg, IUI, Jim Albrecht, Columbia University, New York. The Planning Survey ("Planenkäten"), collected by the Federation of Swedish Industries, has been used (Ola Virin, Kerstin Wallmark).

Section 6 The Consistency Control System

Many micro and macro variables are set during the initialization procedure.

One important question is: Are the variables consistent on the macro level?

To check this, one has to sum the micro variables up to country total or sector total (4 industrial sectors) and check whether macro variables obtained in this way "fit the 1976 input-output system". This has already, briefly, been discussed in Section 3.2. The "input-output consistency check" of the initialization is done as follows:

a) Form a matrix $IO76_{II}$, having the same form as the input-output matrix IO76 (see Section 3), from the initialization by using the sum of micro variables when this is possible, and fill in with macro values from IO76 when it is not possible. We will call the input-output matrix $IO76_{II}$ the "<u>control matrix</u>" in this section.

b) $IO76_{II}$ is considered to be consistent if (1) the values in $IO76_{II}$ do not differ "unreasonably much"[16] from IO76 and (2) horizontal sum of production ≈ vertical sum of production in IO76.

A print-out of the control matrix $IO76_{II}$ is done during the initialization in the sub function **IOAMATRIX**, see Appendix C. On the following pages an example from 1982 of such a print-out is presented. It is from the present initialization version (that is, the one which can be found in Appendices C and D).

By definition, the horizontal sum (col 1,2 through 20 in row 1,2,...10) should be equal to the vertical sum in col 1,2...10. The residual between the horizontal and the vertical sum is printed out under the headline "residual". The first number is the residual in sector 1, the second in sector 2, etc. The residuals in this case indicate that there is too little production in sector 1 and too much in sectors 3 and 4. (sector 1: -1820, sector 3: 2573, sector 4: 7611).

This problem has already been discussed in Section 5 and is probably due to a bad distribution of production and sales between the 4 industrial sectors. The values in the "control-matrix" do not, in general, seem to differ unreasonably much from those of IO76. But the values in column 1 (sector 1) are apparently too small and the values in column 17 (yearly inventory changes) seem to be somewhat large in comparison with IO76.

The negative residual values in rows 5,6,...,10, are due to the fact that the values in columns 5,6,...,10 are expressed in 1975 year's prices. This error need not affect the simulation much, though, since production in sectors 5,6,...,10 in the simulation is determined by inverting the input-output coefficient matrix IO.

Some other consistency controls are effected in the subfunction **CONTROLS**, see Appendix C. For example:

a) wages (average wage times number of employees) in firm i + profits[17] in firm i = value added in firm i. (i=firm index).

b) the input share (compare the variables FΔINKOP and BRINKOP in Appendix A) in sectors 1,2,3,4 obtained by summing the micro-units (Σpurchases/ Σproduction) should be equal to the input share from the input-output matrix IO76.

The printout of the consistency control matrix IO76_{II} can be made (option) during any initialization, and the user can thus check whether the residuals can be considered to be small enough for performing the simulation experiment afterwards.

Table 3 The control matrix

```
INPUT-OUTPUT MATRIX FROM INITIALIZATION:
                                                                             unit=10^6 SEK
         1          2          3          4          5          6          7          8          9         10

      5195       3291       6881       1713        245         94          0       4192        942       1943
      1999       5916       5633       5790        908        117          0       3498        170       2035
       940       2681      14416       1187        503        213          0       6294        171       3079
       791       2765       2393      19287       2078         87          0       2648        102       6484
      2365       2237        400      13965        383          1          0       1689          0        312
      2908        239         93         82         26        140          0        418          5          0
      4076        683        177        338        130        213          0       1009        488        842
      1217       1364       3327       1246       1383        162          0      10928        708       9874
       891       1072        557        629        238        171          0       1118        328       1010
      3245       3802       6939       5709       1792        640          0       9143        426      25656
        62         81        166      ¯3083        163         10          0        350          5       2261
      8608      16344      32150      25355      11452       2529          0      50892       6395      64383
       152       ¯136        209         66         41         35          0        238          0          1
     32449      40339      73341      72285      19341       4413          0      92417       9738     117881

        11         12         13         14         15         16         17         18         19         20         21

       758       5666          0          0          0        464       4938      10737     ¯13004        214      32449
      1953       9524        576          0        894       2652       2035      15759     ¯14535      ¯5329      40339
      3522      15640       3209          0       4979      12500       3025      33620     ¯27314      ¯7896      73341
      5102      58711        116          0        180        161       1348       9340     ¯17427     ¯29493      72285
       243       7144          0          0        132        498       ¯170       1351      ¯4143      ¯3763      19341
        81         25          0          0          0          0        120       1134      ¯3069       2230       4413
       374       2462          0          0          0          0        337       1778      ¯6728      ¯6025          0
      2929      28304      18464      12630       4820        935       1913       7062      ¯4647      ¯6221      92417
       973       3757          0          0          0          0       ¯136        319       ¯320       ¯421       9738
      8849      32131        391          0        609          0       ¯567      10370     ¯17123      29496     117881
         0          0          0          0          0          0          0          0          0          0          0
         0          0          0          0          0          0          0          0          0          0          0
         0          0          0          0          0          0          0          0          0          0          0
     24784     163363      22755      12630      11614      17210      12843      91470    ¯108309     ¯27209     462204
```

Table 3 (cont)

Rows and columns in the control matrix:

Row 1: Raw material sector
Row 2: Intermediate goods
Row 3: Investment goods and consumer durable goods
Row 4: Consumption goods
Row 5: Agriculture, forestry, fishing
Row 6: Mining and quarrying
Row 7: Oil
Row 8: Construction
Row 9: Electricity
Row 10: Other services
Row 11: Commodity-based indirect taxes
Row 12: Value added in producer's prices
Row 13: Correction
Row 14: Sum = production
Column 1, 2 through 10: Corresponding rows
Column 11: Government's consumption
Column 12: Household's consumption
Column 13: Government's investments
Column 14: Investments, buildings
Column 15: Investments in sector 6..10
Column 16: Other investments
Column 17: Change in stock
Column 18: Exports
Column 19: Imports
Column 20: Moms etc.
Column 21: Horizontal sum = production

Residuals R(i): (million Swedish crowns)

Definition: $$R(i) = A(i,21) - \sum_{j=1}^{20} A(i,j)$$

where A is the control matrix in Table 3.

R(1) =	-1 820	R(6) =	-18
R(2) =	742	R(7) =	-154
R(3) =	2 573	R(8) =	-3 981
R(4) =	7 611	R(9) =	-447
R(5) =	-3 302	R(10) =	-3 627

Section 7 On Simulation Techniques*

This section is a bridge between Part 1 and Part 2 of the MOSES Handbook. We give some examples of how this manual could be used in connection with simulation experiments. This section presupposes knowledge of the three first chapters in Part 1.

Let us assume, for example, that we wish to extend the micro data base with more real firms and that we want to experiment with varying the foreign export price index and the growth of the government sector. How do we go about accomplishing this?

To begin with, the micro data base must be updated. This means that the 5 variables in the workspace SI76, see Appendix B, must be updated. This can be done according to instructions in Albrecht, Lindberg (1982). There is one problem, though; we **cannot repeat old experiments** if we simply update SI76 without taking extra measures. Therefore we must look at the function **ESTABLISHMENTS** where micro data are processed (cf. Section 4). We read the beginning of the function in Appendix C.

There is a line in the beginning containing a test for whether a firm belongs to the list of firms chosen to be included in the experiment. This is line [31] in Appendix C, function **ESTABLISHMENTS**. Apparently this means that if we extend all other micro data base variables (i.e. X̲, F∆DATA, FIRMID, R∆MARKET), new firms won't enter the simulation unless LIST is updated as well. If we update LIST **during the initialization procedure** new firms enter the simulation as an **initialization variant**, which in turn means that old experi-

* This discussion is based on the case when we use 1976 as starting year.

ments can be repeated. Therefore we use the **ISTARTXX**-function (cf. Section 2) to extend LIST. The techniques involved can be found in part 1. The new line needed to update LIST will be (for example)

LIST ← LIST, 4.95 4.96.

The numbers 4.95 and 4.96 are code-numbers for new firms. We call the initialization version 11, i.e. we use **ISTART11**. **ISTART11** is shown in Figure 6.

Let us now change the export price index. From Section 4 we know that it is an exogenous time series called EXOΔQDPFOR. We find EXOΔQDPFOR in Appendix C in the subfunction **MARKETS** on line [56]. In **ISTART11** we should swap that line for a new one. The matrix EXOΔQDPFOR is (as we see in Appendix A) the **change** in the export price index each quarter during the simulation, for each of the 4 industrial sectors. If we, for example, wish to make an experiment with a 2 percent change every quarter for all firms, each component of EXOΔQDPFOR should be given the value 0.02. The size of EXOΔQDPFOR is not quite obvious. How many quarters ought one to use in the matrix? The maximum number of years to simulate[18] in the standard initialization is 30 years. Therefore it might be appropriate to use 120 quarters. The export price index must have a start value too. Close inspection of the subfunction **MARKETS** reveals that the model variables QPFOR and QDPFOR should be given new values too. If we don't care much about the first simulated quarter one could, however, skip this and let QPFOR and QDPFOR keep their values from the standard initialization version.

Let us finally change the government sector employment growth, which also is an exogenous time series (cf. Appendix A), called EXOΔREALCHLG. From Section 4 we know that one line in the subfunction **PUBLICΔSECTOR** should be changed. We do this in **ISTART11**. If we let each component of the vector EXOΔREALCHLG take the value 2,500, this means that 2,500 people will be taken from "the pool of unemployment" each quarter during the simulation (unless the pool is empty). The government sector has priority, i.e. firms take people from the "pool of unemployment" after the government sector has satisfied its demand for people.[19]

ISTART11 is shown on the next page, together with another example, **ISTART12**.

Let us go on by describing another experiment, corresponding to **ISTART12**.

In this experiment we wish to change the behaviour of firms in connection with profit targets. We also want to make changes in the production function of individual firms. From Section 4 we know that most **parameters** (definition in Section 1) can be given values in the function **MARKETSΔDATA**. After having checked the parameters in this function with the description in Appendix A we find that SMT is a parameter affecting the profit target behaviour. According to Appendix A, SMT is not a vector,[20] so we can not change the behaviour of an individual firm, only all the firms at the same time. SMT could be given any value between 0 and 1. The construction of SMT is described in Section 1, but in general one has to consult the MOSES code, i.e. the simulation program itself, to check the construction of the parameter. The MOSES code is not included in this paper, but is available at IUI. It will be included in another part of the documentation. We set SMT equal to 1, which means that we don't update profit targets over time. Similarly, we find a parameter INVEFF affecting the production function. The parameter INVEFF yields ΔQTOP/INV where QTOP is maximum production capacity and INV is investments in machinery and buildings. Apparently INVEFF describes the marginal efficiency of new equipment, i.e. how much the production frontier is pushed upwards due to investments. Since INVEFF is a vector (length = number of firms) we could change this parameter for individual firms. SMT and INVEFF are changed in ISTART12, in Figure 6.

Figure 6 Initialization variants and experiment variants, examples

```
      ∇ISTART11[⎕]∇
    ∇ ISTART11
[1]   SYNTH∆FIRMS← 8 16 18 8
[2]   'ESTABLISHMENTS' MODADD   ')CO⍵LIST←LIST,  4.95 4.96   '
[3]  ⍝THIS MEANS THAT THE LINE
[4]  ⍝   LIST←LIST, 4.95 4.96
[5]  ⍝ IS INSERTED AS A NEW LINE AFTER THE LINE
[6]  ⍝   ∊')COPY SI76..  '
[7]   'MARKETS' MODSUBST 'EXO∆QDPFOR←⍵EXO∆QDPFOR←(4 120)⍴0.02'
[8]   'MARKETS' MODSUBST 'QPFOR← 1⍵QPFOR←4⍴100+(3÷8)×2×4'
[9]   'MARKETS' MODSUBST 'QDPFOR←(T⍵QDPFOR←4⍴0.02'
[10]  'PUBLIC∆SECTOR' MODDEL 'EXO∆REALCHLG←'
[11]  'PUBLIC∆SECTOR' MODADD 'LG←QLG[4]⍵EXO∆REALCHLG←120⍴2500'
    ∇
```

```
      ∇ISTART12[⎕]∇
    ∇ ISTART12
[1]   SYNTH∆FIRMS← 8 16 18 8
[2]   'MARKETS∆DATA' MODSUBST 'SMT←⍵SMT←1'
[3]   'MARKETS∆DATA' MODADD 'GAMMA←⍵INVEFF←147⍴0.5'
    ∇
```

```
      ∇MSTART13[⎕]∇
    ∇ MSTART13
[1]  ⍝EXAMPLE
[2]   'NULLIFY' MODADDLAST ' SHRINK ''QF'' '
[3]  ⍝ MODADDLAST MEANS THAT THE LINE IS ADDED AS THE LAST LINE
    ∇
```

Note: These functions are examples which no longer are stored in ISTART- and MSTART-workspaces.

7.1 New variables **(IMPORTANT)**

If new variables are added to the model two extra measures have to be taken. Firstly, add the name of the variable to a variable list in workspace VLISTS (cf. Part 1). All model variables coming out from initialization should be registered[21] there. Secondly, add a line in the MOSES-model in a subfunction called **NULLIFY**. Say that the new variable is called QF. Then the line:
SHRINK 'QF'
should be added in the function **NULLIFY**.

The reason for this procedure is that some firms go bankrupt during the simulation and then all micro variables which are **vectors** become shrunk (one firm is deleted from the vector). Micro variables which are vectors[22] must be part of this "shrinking system" and that's why the line above must be added. One should extend **NULLIFY by using** the function **MODADD** in a **MSTARTXX**-function (cf. Part 1). This is done in **MSTART13** in Figure 6.

7.2 Experiment variants, exogenous exports

One common experiment variant is to make some endogenous variable in the simulation exogenous instead. One can, for example, make exports exogenous. This was done in connection with experiments concerning multiplier effects on the Swedish economy described in Bergholm (1984).

The necessary changes can be made in an **MSTART**-function. In Figure 7 below some[23] lines from experiment version 10, **MSTART10**, are shown (cf. Part 1 of the handbook):

Figure 7

```
    ∇  MSTART10
[10]   EXO∆EXPORT←X×QS÷QP
[12]   TID←1
[14]   RATE←(ρQ)ρ1.05*(1÷4)
[18]   'EXPORT∆MARKETS' MODADD 'QSUFOR←ωQSUFOR←EXO∆EXPORT×RATE*TID'
[19]   'EXPORT∆MARKETS' MODADDLAST 'X←QSUFOR÷QOPTSU'
[20]   'EXPORT∆MARKETS' MODADDLAST 'TID←TID+1'
[21]   'NULLIFY' MODADDLAST ' SHRINK ''RATE'' '
[22]   'NULLIFY' MODADDLAST ' SHRINK ''EXO∆EXPORT'' '
    ∇
```

Line [18] in **MSTART10** makes exports QSUFOR exogenous.

$$QSUFOR = EXO\Delta EXPORT \cdot (RATE)^{TID},$$

where EXO∆EXPORT is the export volume for TID = 0, TID is the time variable (measured in simulated quarters) and RATE is the growth rate. The function **MODADDLAST** adds the line at the end of the function specified as left hand argument. Note the necessary additions to the subfunction NULLIFY (cf. Section 7.1). To be able to construct and fully understand MSTART-functions like MSTART10 one has to consult the MOSES code, cf. Eliasson-Heiman-Olavi (1978).

7.3 Change of Starting Year of Simulation

To be able to start the simulation in some other year than 1976 or 1982 requires the creation of a new data base for that year.

This manual provides the user with essential information for that task. The whole macro data base is described in Appendix B. This should be updated to the year in question. Price indices come (in general) from SCB data and taxes, sales etc. from national accounting statistics. The cumbersome task is, above all, the input-output system. To update this to, for example, 1980 would probably mean months of work. There is, however, a **short cut** method of updating the input-output matrix (coefficients).

One could simply run the model for five years and let the simulated coefficients be an approximation of the real coefficients. Consistency problems will probably appear, though (cf. Sections 5 and 6).

Micro data can rather easily be fetched from the Planning Survey, since it is collected yearly since 1975 with small changes in the format of the questionnaire.

The initialization procedure (Appendix C) applies, for the time being, only to the starting year 1976. However, one need not change it much to be able to use it in connection with another starting year. Some changes are necessary: New workspaces for macro and micro data should be input to the initialization program. One should not change the names of the variables (for example IO76 etc.)

although that would be natural, or, alternatively, rename them (for example IO80 ← IO76) in the beginning of the initialization.

7.4 Simulation extension, the ENTRY block

In the MOSES code* there is a function making the entry of firms possible. It is called **ADDFIRM** and is documented in Appendix D.

The idea behind this facility was to remedy the asymmetry connected with the exit-mechanism in the standard version of the model. Firms (cf. Section 7.1) go bankrupt (exit) during the simulation but no inflow of new firms takes place. This ENTRY module is rather primitive and an improved module is documented in Hanson (1989). **ADDFIRM** should be used in MSTART-functions.

* Not documented in this handbook. See Eliasson-Heiman-Olavi (1978).

APPENDIX A VARIABLES COMING OUT FROM INITIALIZATION, AN ALPHABETICAL LIST

The concepts "start-up variable", "exogenous time series", "constants", "parameters" and "micro variables" from Section 1 are used to describe types of variable.

This variable list is of utmost importance in connection with simulation experiments. To be able to set parameter values, change exogenous time series or start-up variables, this list must be consulted.

This list also specifies the result of the initialization. Appendix A is a guide which considerably facilitates work with the model and enables one to check this result. Previously a lot of "time consuming detective work" was needed for almost any little change in the initialization procedure or the simulation. Knowledge about the meaning of the model variables below was, in the lack of written documentation, based on experience and scattered notes.

Appendix A is also needed if one wants to extend the model. In such a situation one must do considerable checking up on the input to the model, i.e. the variables below.

Note that Appendix A yields a specification of input needed to start the model any year, not necessarily the present starting year 1976. Thus, this appendix is a piece of information needed when constructing another starting year for the simulation.

All the variables below (with exception of FΔINKOP and BRINKOP), are inputs to the model. Start-up variables usually refer to the last quarter 1976 since the model is run by quarters. Some variables also refer to the whole year 1976, though. In the "code-column" we write vectors and matrices with indexes; we write v(i) instead of v, if v is a vector. In the "type-column" we tell the range of the index i.

MODEL VARIABLES
- An Alphabetical List

Code	Type	Used in (purpose)
ALFABW	Constant, micro variable, parameter	INVFIN to determine firms' desired change in borrowing. INVFIN = investment part of model
ALFA3 ALFA4	Constants, parameters	The household sector part of the model
AMAN(i,j)	Start-up variable i=1,2,3... number of firms j=1,2,3 micro variable	For each firm, a three component vector accomodating the two-quarter lag of layoffs
AMANΔyear	Technical variable, needed for simulation	
BAD(i)	Start-up variable, micro variable i=1,2...number of firms	Investment financing part of the model. Counts number of quarters a firm has negative net worth. If BAD > 6 then it is nullified in the model
BETA	Constant, micro variable parameter	Constant used to compute optimum finished-goods inventory level

Code	Type	Used in (purpose)
BETA1(j)	Constant. Vector of length 12 j=1,2,...12	COMPUTE EXPENDITURES to adjust household expenditures in different categories to the income constraint
BETA2(j)	Constant, j=1,2,...12	COMPUTE EXPENDITURES to adjust household expenditures in different categories to the income constraint SUM(BETA2) = 1
BETA3(j)	Constant, j=1,2,...12	COMPUTE EXPENDITURES to adjust household expenditures in different categories to the income constraint. All BETA3(j) = 0 for the present
BETABW	Constant, micro variable, parameter	INVFIN to determine firms' desired change in borrowing. INVFIN = investment financing part of the model
BIG(i)	Constant, micro variable i=1,2,3... number of firms	Maximum inventory level (fraction of sales). Finished goods
BRINKOP(j)	Information variable j=1,2,3,4	Average input share in each industrial sector. For the definition of input share, see Section 3.3 in this handbook. BRINKOP=input/production
BW(i)	Start-up variable, micro variable i=1,2,..., number of firms	A firm's total borrowing Last quarter 1976
CHM(i)	Start-up variable, micro variable i=1,2,3..., number of firms	Yearly change in M (profit margin). Change 1975-76
CVA(j)	Start-up variable j=1,2,...11	CVA = QC but in fixed prices

Code	Type	Used in (purpose)
DELAY△INV(i,j)	Start-up variable micro variable i=1,2,..., number of firms j=1,2,3	Investments between plan and fulfilment. Three stages

DP(i)
DW(i)
DS(i)
DQ(i)
	Start-up variables, micro variables i=1,2,..., number of firms	Yearly change (a fraction) of P, W, S and Q respectively

DVA(i)	Start-up variable, micro variable i=1,2,..., number of firms	Change in VA (a fraction)
DUR	Index	DUR = 3
E1 E2	Constant, micro variable, parameter	Used in YEARLY EXPECTATIONS in the model. E2=0 at present (Jan. 1982)
ELINV	Constant, micro variable, parameter	An elasticity, reducing firms' desired new borrowing (and hence in vestments) whenever capacity utilization is low. Used in INVFIN.
ENTRY	Constant, parameter	A parameter regulating the inflow of new persons to the labour market (quarterly fraction of the total labour force)
EPS	Constant, micro variable, parameter	EPS = 0 and thus redundant at present (Jan. 1982)
EXO△QCHTXVA1(j) EXO△QCHTXVA2(j)	Exogenous time-series j=1,2,...NQR, NQR = number of quarters in the simulation	TAXVA2 = value added tax rate = "MOMS"EXO△QCHTXVA2 is change in the "MOMS-rate". TXVA1 refers to investment goods. EXO△QCHTXVA1 is the change in that tax rate

Code	Type	Used in (purpose)
EXO△QDINVBLD(j)	Exogenous time-series j=1,2..NQR NQR = number of quarters in the simulation	Quarterly change (a fraction) of QINVBLD = investments in residential construction
EXO△QDINVG(j)	Exogenous time-series j=1,2,..NQR NQR = number of quarters in the simulation	Quarterly change (a fraction) of QINVG = investments in the government sector
EXO△DINVIN(j)	Exogenous time-series j=1,2,..NQR NQR = number of quarters in the simulation	Quarterly change (a fraction) of QINVIN = investments in sectors 5,6,...10.
EXO△QDPFOR(i,j)	Exogenous time-series (a matrix) i=1,2,3,4 j=1,2,3..NQR. NQR = number of quarters in the simulation. micro variable	The change (a fraction) in foreign price index, for each of the 4 industrial sectors

EXO△QDPIN(i,j)	Exogenous time-series
	i=5,6,7,8,9,10
	j=1,2,...NQR
	NQR = number of quarters in the simulation
		Quarterly change in domestic price index in sectors 5,6,7,8,9,10

EXO△REALCHLG(j)	Exogenous time-series j=1,2,..NQR NQR = number of quarters in the simulation	A variable (vector) telling the number of people to be added to the government sector each quarter (Government demand)

EXO△RI(j)	Exogenous time-series
	NQR = number of
	j=1,2...NQR
	quarters in the simulation
		The rate of interest

Code	Type	Used in (purpose)
EXO△RIBWFOR(j)	Exogenous time-series j=1,2...NQR NQR = number of quarters in the simulation	The foreign lending rate of interest
EXO△RIDEPFOR(j)	Exogenous time-series j=1,2...NQR NQR = number of quarters in the simulation	The foreign deposit rate of interest

EXO△RSUBS(i,j)	Exogenous time-series, micro variable
	i=1,2,3,4
	j=1,2,...NQR
	NQR = number of quarters in the simulation. i = sector index
		Subventions to the individual firm, expressed as a fraction of sales. Equal for all firms in a sector

EXO△TXC(j)	Exogenous time-series j=1,2...NYR NYR=number of years in the simulation	Corporate tax-rate (Tax on firms)
EXO△TXI1(j)	Exogenous time-series j=1,2,..NYR NYR=number of years in the simulation	Income-tax rate (for households)
EXO△TXI2(j)	Exogenous time-series	Some kind of income tax rate used in another version of the MOSES-model than the present (Jan. 1982). Can't be omitted for technical reasons but redundant
EXO△TXW(j)	Exogenous time-series j=1,2...NYR NYR=number of years in the simulation	Payroll-tax rate for the non-government sectors

Code	Type	Used in (purpose)
EXO△TXWG(j)	Exogenous time-series j=1,2...NYR NYR=number of years in the simulation	Payroll-tax rate for the government sector

EXPDW(i)
EXPDS(i)
EXPDP(i)
	Start-up variables, micro variables i=1,2... number of firms etc.	Expected change (a fraction) in P, W and S

EXPXDP	Constants, micro variables	Expected rate of price-change
EXPXDW	parameters	Expected rate of wage-change
EXPXDS		Expected rate of sales-change These are the constant components of expectations, entered exogenously
FASS	Constant	Bank-parameter
FD	Constant	Bank-parameter
FIP FIW FIS	Constants, micro variables parameters	Used in "Quarterly-Expectations" in the model
First△sim△year	Technical variable, needed for simulation	
Funds△are△ enough	Constant	Bankparameter
F△INKOP(i)	Information variable i=1,2,... number of firms	F△INKOP is not used in the model. Each firms's input share (fraction of production) of input goods, 1976. See Section 3.3, Part 2
GAMMA	Constant, micro variable, parameter	A constant telling how big a wage increase is needed, for making a person leave his job for another job. GAMMA = 0.1 at present

Code	Type	Used in (purpose)
GKOFF(j)	Constant j=1,2,...10	Government purchasing (less investments) in each sector, as a fraction of Government wage sum. GKOFF is a vector

HISTDP(i)
HISTDW(i)
HISTDS(i)
	Constants, micro variables i=1,2,... number of firms	For each firm a time-smoothed average of its experienced (historical) price changes (HISTDP), wage changes (HISTDW) and sales changes (HISTDS)

HISTDPDEV(i)
HISTDSDEV(i)
HISTDPDEV(i)
	Start-up variables, micro variables i=1,2,... number of firms	For each firm a time-smoothed average of the difference between actual and expected increase in price level, wage level and sales

HISTDPDEV2(i)
HISTDSDEV2(i)
HISTDWDEV2(i)
	Constants, micro variables i=1,2,... number of firms	Redundant at present because E2 = 0

IMBETA	Constant, micro variable, parameter	Constant used to compute optimum input-goods inventory level = 0.5
IMBIG(i)	Constant, micro variable i=1,2,... number of firms	Maximum inventory level (fraction of sales). Input goods
IMP(i)	i=1,2,3,4 Start-up variable micro variable	Import share in sectors 1,2,3,4 (the industrial sectors). Start-up value
IMP(j)	j=5,6,...10 Constant, macro variable	Import share in external sectors 5,...10. Constant. NOTE: IMP is a start-up variable and a constant at the same time!
IMPLPΔREF	Technical variable needed for simulation	

Code	Type	Used in (purpose)
IMSMALL(i)	Constant, micro variable i=1,2,... number of firms	Minimum inventory level (fraction of sales). Input goods

IMSTO(i,j)	Start-up variable (matrix), micro variable
	i=1,2,3.. number of firms
	j=1,2,3,...10
		Inventory level of input goods for each type of product (10 sectors). Fixed (1976 year's) prices

IN	Vector-index	Index for external sectors IN = 5,6,7,8,9,10
INVEFF(i)	Start-up variable micro variable i=1,2,3... number of firms	The quotient $\frac{\text{change in QTOP}}{\text{investment}}$ QTOP = potential maximum production level. Production function parameter

IO(i,j)	Constant, micro variable
	i=1...10
	j=1...10
		Input-output coefficients, 10x10 matrix. Tells the share of production in sector j coming from sector i
		$\sum_{i=1}^{10} IO(i,j)$ + value added share = 1

IO2 (i,j)	Constant,
	i=1,2,3,4
	j=5,6,...10
		Input-output coefficient Submatrix of IO(i,j)

IO3 (i,j)	Constant,
	i=5,6..10
	j=5,6,...10
		Input-output coefficients. Submatrix of IO(i,j), which is inverted during simulation

IOTA	Constant, micro variable, parameter	A constant used by firms to form their initial wage offer in LABOUR SEARCH. IOTA=0.5 at present
K1(i)	Micro variable start-up variable i=1,2... number of firms	For each firm, the replacement value of its production equipment

Code	Type	Used in (purpose)
K2(i)	Micro variable start-up variable i=1,2,... number of firms	For each firm, its current assets last quarter 1976
Kappal Kappa2	Constants	Bankparameters
K1BOOK(i)	Start-up variable micro variable	For each firm, the book value (1976) (for taxation purposes) of its production equipment
KSI	Constant, micro variable, parameter	A constant, used in LABOUR SEARCH which tells by how much a firm raises its own wage level after it has performed an unsuccessful attack KSI = 0.25 at present
L(i)	Start-up variable i=1,2... number of firms micro variable	Number of people in each firm. Last quarter 1976
Lamdal Lamda2	Constants	Bank-parameters
Last∆TXI2∆year	Technical variable needed for simulation	
Last∆year	Technical variable needed for simulation	Last∆year = 1976
LEFT(i)	Logical vector (start-up variable) i=1,2... number of firms	Logical vector indicating whether a firm is out of business or not. During simulation LEFT(i) takes the value zero if firm i is nullified (deleted)
LG	Start-up variable	Number of people employed in the government sector last quarter 1976

Code	Type	Used in (purpose)
LIQB	Start-up variable	The bank's holdings of "liquidity" of an unspecified nature. Updated in BANK UPDATE
LIQBFOR	Start-up variable	The bank's current holdings of foreign "liquidity" of an unspecified nature. Updated in BANK TRANSACTIONS
LOSS	Constant, micro variable, parameter	Used in connection with production function
LU	Start-up variable	Number of people unemployed last quarter 1976
M(i)	Start-up variable micro variable i=1,2... number of firms	Profit margin (profit/value added) for each firm the whole 1976
MARKET(i)	Start-up variable micro variable i=1,2,3... number of firms	MARKET(i)=1 or 2 or 3 or 4. This variable tells to which sector a certain firm belongs
MARKETΔITER	Parameter	Telling the number of iterations in the product market process in the model
MAXDP	Constant, micro variable, parameter	ADJUST-PRICES in the model
Maxqchri	Constant	Bank-parameter. Maximum change in rate of interest
Maxri	Constant	Bank-parameter
Maxridiff	Constant	Bank-parameter
MB	Constant	Bank-parameter
Minri	Constant	Bank-parameter

Code	Type	Used in (purpose)
MHIST(i)	Start-up variable micro variable i=1,2,3... number of firms	For each firm, an average of past profit margins (a fraction)
MKT	"Vector index" MKT=1,2,3,4	Index for industrial sectors=1,2,3,4. In the APL-language "vector indices" are allowed.
MTEC(j)	Start-up variable micro variable j=1,2,3,4	On each market, sector 1,2,3,4, a technology factor of modern equipment (potentially produced units per person and quarter). Last quarter 1976. Production function parameter
NDUR	"Vector-index" =1,2,4,5...11	
NDURΔDUR	"Vector-index" =1,2,3,4...11	
NITER	Parameter	Telling the number of labour-market iterations in the labour market process in the model
NH	Constant	The number of households in the model
NMARKETS	Index	The number of industrial sectors in the model=4
NWB	Start-up variable	The net value of the bank. Residual between assets and liabilities
OMEGA(j)	Constant, micro-variable j=1,2,...10	A distribution vector indicating how firms' outlays for investments are allocated on purchases from different model sectors. Assumed to be equal for all firms

Code	Type	Used in (purpose)
OMEGABLD(j)	Constant j=1,2,...10	A distribution vector indicating how investments in residential construction result in purchases from different model sectors
OMEGAG(j)	Constant j=1,2...10	A distribution vector indicating how government investments result in purchasing from different model sectors
OMEGAIN(j)	Constant j=1,2,...10	A distribution vector indicating how investments from external sectors (5,6,7...10) (less residential construction) result in purchases from different model sectors
ORIGMARKET(i)	Vector i=1,2... number of firms	Copy of the vector MARKET. Needed because MARKET will be changed during simulation
P(i)	Start up variable i=1,2,3... micro variable	Yearly price index 1976 =100 for all i (IMPORTANT)
POSG	Start up variable	The government's net position in the bank
POSGFOR	Start up variable	The government's net foreign deposit/borrowing position 1976
PΔREF(j)	Constant j=1,2...10	Reference-price level. QPDOM+"value added tax" (=MOMS) value
Q(i)	Start up variable micro variable i=1,2,3... number of firms	Yearly production in each firm 1976, in fixed (1976) prices

Code	Type	Used in (purpose)
QC(j)	Start up variable j=1,2...11	Each household's consumption of products from the 10 sectors. QC•(number of households) yields aggregate consumption. The 11th component is redundant. Last quarter 1976. Current prices
QCHRI	Start up variable	Change in RI (rate of interest)
QCPI	Start up variable	Quarterly consumer price index. Last quarter 1976
QDCPI	Start up variable	Quarterly change (a fraction) of quarterly price index QCPI. Last quarter 1976
QDI	Start up variable	Disposable income per household. QDI•(number of households)=aggregate disposable income. Last quarter 1976
QDMTEC(j)	Constant micro variable j=1,2,3,4 parameter	On each market, the rate of technology upgrade for production equipment (a fraction on quarterly basis). Entered exogenously
QDPDOM(i)	Start up variable micro variable i=1,2...10	Change in QPDOM. A fraction. Last quarter 1976. 10 sectors
QDWIND	Start-up variable	Average wage increase in the industry (sector 1+2+3+4) during one quarter (a fraction)
QIMQ(i,j)	Start-up variable micro variable i=1,2,3..., number of firms j=1,2...10	Each firm's quarterly purchases of each kind of product (10 sectors). Fixed (1976) prices. Last quarter 1976

Code	Type	Used in (purpose)
QINPAY	Start-up variable	Households' aggregate wage and capital income from the external sectors (sectors 5,6...10) during one quarter. Computed in EXTERNAL SECTORS. Last quarter 1976
QINV(i)	Start-up variable micro variable i=1,2,3... number of firms	Each firm's investments during a quarter. Will enter the bookkeeping next quarter (last quarter 1976). NOTE: QINV is in current prices
QINVBLD	Start-up variable	Investments in the construction sector last quarter 1976
QINVG	Start-up variable	Government investments last quarter 1976
QINVLAG(i)	Start-up variable micro variable i=1,2,3... number of firms	Each firm's investment plans during a quarter. (There is a couple of quarter's delay between plan and fulfilment of investment.) Last quarter 1976
QINVIN	Start-up variable	Investments in sectors 5,6...10. Last quarter 1976
QP(i)	Start-up variable i=1,2... number of firms micro variable	Quarterly price-index for each firm. Last quarter 1976
QPDOM(j)	Start-up variable (micro-variable to some extent) j=1,2,3,4...11	Domestic quarterly price index in the four industrial sectors last quarter 1976. Each firm has the same domestic price in a sector
QPFOR(j)	Start-up variable j=1,2,...4 micro variable	The foreign price index last quarter 1976. 4 sectors. Each firm has the same foreign price in a sector

Code	Type	Used in (purpose)
QPH(j)	Start-up variable j=1,2...11	Domestic prices for households for 10 sectors. The 11:th component is redundant, but must be there for technical reasons
QQ(i)	Start-up variable micro variable	Same as Q, but applies to quarter instead of year. Last quarter 1976
QS(i)	Start-up variable micro variable	Same as S, but quarterly variable. Last quarter 1976
QSAVHREQ	Start-up variable	One quarter's reduction in aggregate household borrowing
QTOP(i)	Start-up variable micro variable i=1,2,3... number of firms	Potential maximum production in each firm's production function Last quarter 1976
QTDIV	Start-up variable	One quarter's aggregate payments of dividends from firms to households Last quarter 1976
QTTAX	Start-up variable	Total tax receipts by the government during one quarter. Updated in GOVERNMENT ACCOUNTING. Last quarter 1976
QVA(i)	Start-up variable micro variable	Same as VA, but last quarter 1976 instead of the whole year
QW(i)	Start-up variable micro variable	Same as W, but refers to quarter instead. (Wage is expressed as the yearly wage-sum though)
QWG(i)	Start-up variable micro variable i=1,2... number of firms	Same as WG, but refers to last quarter 1976. (Still expressed as early wage-level)

Code	Type	Used in (purpose)
R	Constant, micro variable parameter	Used in YEARLY-EXPECTATIONS in the model
REDCHBW	Constant, micro variable parameter	Maximum allowed change in borrowing (fraction of borrowing)
RES(i)	Start-up variable micro variable i=1,2... number of firms	Parameter connected with the production function
RESDOWN	Constant, micro variable parameter	Used in connection with production function
RESMAX	Constant micro variable, parameter	A constant telling maximum slack any firm can possibly have RESMAX = 0.2 (Jan. 1982)
RET	Constant, parameter	Retirement rate on the labour market (a fraction on quarterly basis)
RHO	Constant micro variable parameter	Physical depreciation rate of production equipment (a fraction on quarterly basis)
RHOBOOK	Constant micro-variable, parameter	Maximum allowed depreciation rate of production equipment, for taxation purposes. A fraction quarterly basis
RHODUR	Constant, parameter	Depreciation rate of consumer durable goods (a fraction on quarterly basis)
Rfund1 Rfund2	Constant	Bank parameters
RI	Start-up variable	Rate of interest (a fraction). Last quarter 1976

Code	Type	Used in (purpose)
R1△1S△EXOGENOUS	Logical variable	R1△1S△EXOGENOUS = 1 means that EXO△RI will be used, i.e. rate of interest will be exogenous
RLU	Constant, parameter	Fraction used in HOUSEHOLD INIT to compute unemployment compensation in proportion to average wage level in the industry. RLU=0.6 (Dec.1982)
RSUBS△CASH(i)	Constant micro-variable i=1,2,3... number of firms parameter	Government subventions to individual firms. Temporary subvention. The amount is expressed as a fraction of sales
RSUBS△EXTRA(i)	Constant micro variable i=1,2... number of firms parameter	Government subventions to individual firms expressed as a fraction of sales in the firm. Nontemporary subvention
RTD	Constant micro variable parameter	Ratio between firms' dividend payments and corporate taxes
RTRANS	Constant, parameter	Ratio between total transfer payments to households (less unemployment compensation) and total taxes. Used in HOUSEHOLD INIT; assumed constant
RU	Start-up variable	Rate of unemployment (fraction of total labour-force) last quarter 1976
RW(i)	Constant, micro variable i=1,2... number of firms parameter	A constant giving firms' desired amount of work ing capital (K2) as a fraction of current yearly sales
S(i)	Start-up variable micro variable i=1,2,3... number of firms	Yearly sales in each firm (current prices) 1976

Code	Type	Used in (purpose)
SAV	Index. SAV=12	
SHARE(i)	Constant, micro variable i=1,2... number of firms	SHARE(i) = $\frac{\text{individual firm's input share}}{\text{average inputshare in sector}}$ See Section 3.3
SKREPA	Constant, parameter	A constant factor by which the probability for the pool of unemployed to be selected at a labour market attack is upgraded, as compared with the probability for any firm to be selectd. Used in CONFRONT
SMALL(i)	Constant, micro variable i=1,2,3... number of firms	Minimum inventory level (fraction of sales) Finished goods
SMOOTH(j)	Constant j=1,2...12	Used in the household part of the model
SMP	Constant, micro variable, parameter	This variable is used by firms to (each year) time-smooth their price-experiences. Equal for all firms
SMS	Constant micro-variable parameter	This variable is used by firms to (each year) time-smooth their sales experiences
SMT	Constant micro variable value jan-82: 0.5 parameter	This variable controls how quickly the profit-target is changed between two quarters. Equal for all firms
SMW	Constant, micro variable, parameter	Used by firms to (each year) time-smooth their wage experiences
STO(i)	Start-up variable micro variable i=1,2,3... number of firms	Inventory level of finished goods. Fixed (1976 year's) prices. Last quarter 1976

Code	Type	Used in (purpose)
STODUR	Start-up variable	Each household's stock of durable goods, current prices, last quarter1976
TEC(i)	Start-up variable i=1,2... number of firms	Parameter connected with the production function of the individual firm
THETA	Constant micro-variable parameter	Parameter used in the labour market-process in the model
ThisΔyear	Technical variable needed for simulation	= 1976
TMFASS	Constant	Bank-parameter
TMFD	Constant	Bank-parameter
TMIMP(j)	Constant j=1,2,3,4 micro variable parameter	Time constant for Swedish consumers to adjust import share (of demand) in each of the 4 industrial sectors
TMIMSTO	Constant micro variable parameter	Constant used for inventories. See the PLANQREVISE-part of the model. Has to do with adjustment-speed to optimum inventory level
TMINV(j)	Constant micro variable j=1,2,3,4 parameter	Average delay time to install investments in new production equipment Used in INVFIN; assumed to be equal for all firms in a sector. Sectors 1,2,3,4
THMSTO	Constant micro variable parameter	Constant used for inventories. See the PLANQREVISE-part of the model. Has to do with adjustment-speed to achieve optimum inventory level
TMX(j)	Constant micro-variable j=1,2,3,4 parameter	Time constant for firms when they adjust export share. Common to all firms in a sector

Code	Type	Used in (purpose)
TSTOCURF(j)	Start-up variable j=1,2,3,4	For each industrial sector (1..4) the aggregate finished goods inventories at current factor prices
TSTOCURM(j)	Start-up variable =1,2,3,4	For each industrial sector, the aggregate finished goods inventories at current market prices
TXI3	Technical variable needed for simulation	
TXVA1	Start-up variable	Value added tax, last quarter 1976. Compare with EXOΔQCHTXVA1
TXVA2	Start-up variable	Value added tax rate = "Moms". Last quarter 1976
VA(i)	Start-up variable micro-variable i=1,2,3... number of firms	Valued added for each firm 1976. Current prices in the model
UTREF	Constant micro-variable	A "reference" level of capacity utilization. Used in INVFIN when firms form their desired new borrowing and correct it for their current degree of utilization. Assumed equal for all firms
W(i)	Start-up variable micro-variable i=1,2,3... number of firms	Wage-sum per employee (expressed as wage sum per year) the whole 1976
WG	Start-up variable	Wage level in government sector 1976. Expressed as: yearly wages/number of people

Code	Type	Used in (purpose)
WG∆REF	Copy of WG for technical reasons	
WH	Start-up variable	Each household's wealth last quarter 1976 (current value of its bank deposits)
WHRA	Start-up variable	Each household's so called wealth ratio (quotient between bank deposits and quarterly disposable income)
WSG	Start-up variable	Total government wage sum last quarter 1976. Expressed as yearly wage sum
WTIX	Constant WTIX=1	Probably redundant, at present
X(i)	Start-up variable micro-variable i=1,2,3,4,... number of firms	Export share (exports/ production) for each firm in the 4 industrial sectors. Last quarter 1976
XIN(j)	Constant j=5,6,...10	Export share (exports/ production) in external sectors (5,6...10)
Z	Index Z=11	

The names of all the model-variables are stored in a workspace VLISTS.

The contents of this workspace are listed below. The names are stored in the text-variables: VARIABELGRUPP1,...VARIABELGRUPP5, GRUPP1.

Two functions **COPYSAVE** and **KILL** are also stored in this workspace (documented in Appendix C).

APPENDIX A: WORKSPACE VLISTS

```
      DOKUMENTATION
DOKUMENTATION
⍝ DOCUMENTATION:
⍝
⍝ COMPLETE LISTS OF OUTPUT-VARIABLES FROM INITIALIZATION
⍝
⍝ VARIABELLISTA1= EXOGENOUS VARIABLES
⍝ VARIABELLISTA2= ENDOGENOUS VARIABLES
⍝ VARIABELLISTA3= CONSTANTS
⍝ VARIABLELISTA4,VARIABELLISTA5= OTHER VARIABLES (TECHNICAL)
⍝ GRUPP1 = VARIABLES WHICH ARE TAKEN DIRECTLY FROM INPUT-
⍝ WORKSPACE MACRO.
⍝
⍝
⍝ IF NEW VARIABLES ARE ADDED TO THE INITIALIZATION,THE
⍝ VARIABLELISTS ABOVE HAVE TO BE UPDATED WITH THE NEW
⍝ VARIABLES,OTHERWISE THE VARIABLES WILL BE DELETED
⍝ IN THE FUNCTION OUTPUTΔOPERATIONS.
⍝
⍝ FREDRIK BERGHOLM ,DEC 1981
⍝============================================
```

APPENDIX A: WORKSPACE VLISTS

```
      VARIABELGRUPP1
EXOΔRSUBS QINVG EXOΔREALCHLG EXOΔQDINVG GKOFF OMEGAG XIN IMP IO IO2 IO3
       OMEGA OMEGABLD QINVBLD QINVIN EXOΔQDINVIN EXOΔQDINVBLD QPFOR EXOΔ
      QDPFOR EXOΔQDPIN   SHARE QDMTEC EXPXDP EXPXDW EXPXDS RET ENTRY EXO
      ΔQCHTXVA1 EXOΔQCHTXVA2 MTEC WSG RSUBSΔEXTRA RSUBSΔCASH NH  OMEGAIN
       EXOΔTXC EXOΔTXI1 EXOΔTXW EXOΔTXWG EXOΔRI EXOΔRIBWFOR EXOΔRIDEPFOR
       RET ENTRY EXOΔQDINVBLD
```

```
      VARIABELGRUPP2
  LG QWG WG LU IMP QPDOM X HISTDP HISTDW HISTDS HISTDPDEV2 HISTDWDEV2 HI
      STDSDEV2 MHIST QIMQ L EXPDP EXPDW  EXPDS DP DW DS DQ QP QW QS QQ Q
      VA Q P S W VA M DVA AMAN IMSTO STO QTOP TEC RES K1 K2    BW INVEFF
       QINV QINVLAG DELAYΔINV QTDIV K1BOOK QDWIND              TSTOCURF
      TSTOCURM QPH WH WHRA QC CVA QDCPI STODUR QSAVHREQ QCPI  K1BOOK QDP
      DOM HISTDPDEV HISTDWDEV HISTDSDEV  CHM QDI
```

APPENDIX A: WORKSPACE VLISTS

```
      VARIABELGRUPP3
BETA TMSTO IMBIG IMSMALL TMIMSTO IMBETA RHO RHOBOOK RESMAX LOSS RESDOWN
      WTIX RW ALFABW BETABW ELINV RTD TMINV EPS TMX TMIMP RLU MAXDP UTRE
      F_R E1 E2 SMP SMW SMS FIP FIW FIS GAMMA THETA KSI SKREPA IOTA SMAL
      L BIG RTRANS POSGFOR TMFASS TMFD FD FASS  KAPPA1 KAPPA2 RFUND1 RFU
      ND2 LAMDA1 LAMDA2 MAXQCHRI MB MAXRIDIFF MINRI MAXRI FUNDSΔAREΔENOU
      GH RHODUR ALFA3 ALFA4 BETA1 BETA2 BETA3 SMOOTH  SMT BAD REDCHBW
```

```
      VARIABELGRUPP4
RIΔISΔEXOGENOUS MARKET MKT IN NDURΔDUR DUR NITER MARKETΔITER SAV Z NDUR
       LEFT            FΔINKOP BRINKOP
```

```
      VARIABELGRUPP5
WGΔREF PΔREF ORIGMARKET
```

```
      GRUPP1
TXVA1 TXVA2 RI NWB LIQB  POSG LIQBFOR RU QCHRI  QTTAX QINPAY LASTΔYEAR T
      HISΔYEAR FIRSTΔSIMΔYEAR AMANΔYEAR LASTΔTXI2ΔYEAR NMARKETS EXOΔTXI2
        IMPLPΔREF  TxI3
```

APPENDIX B1 MACRO- AND MICRO-DATA DOCUMENTATION DEC. 1983 WORKSPACE MACRO AND SI76

All variables (dec 83) in workspace MACRO are listed in this appendix. This is a complete documentation of the macro data base. The micro data base is also complete, although firm variables are not printed since they are confidential. This appendix is needed as a pattern if one wants to initialize the micro-to-macro model for another starting year.

There are also 3 functions in workspace MACRO. They are used to form certain variables during the initialization procedure (**AGGRITAX** and **TLΔEXPΔPRISΔ76**) or before the initialization (**COEFFΔIO**).

The variables are:

AMANΔYEAR
BLDΔRATE1
BLDΔRATE2
EXOΔQTXVA1
EXOΔQTXVA2
EXOΔRI
EXOΔRIBWFOR
EXOΔRIDEPFOR
EXOΔTXC
EXOΔTXI1
EXOΔTXI2
EXOΔTXW
EXOΔTXWG
FIRSTΔSIMΔYEAR
GΔRATE1
GΔRATE2
HISTΔTXVA2
HOURSΔPERΔYEAR
HUSHALLSDEP
IMPLΔPRIS
IMPLΔPRISΔIN
IMPLPΔREF
INΔRATE1
INΔRATE2
INITΔGROWTH
IO76
IOCOEFF76
LASTΔTXI2ΔYEAR
LASTΔYEAR
LGTRENDCH
LIQB
LIQBFOR
LON
LONΔOFF
MACROLIST
NMARKETS
NWB
POSQ
QCHRI
QINPAY
QTTAX
RI
RU
SALES76
THISΔYEAR
TIM
TIMΔOFF
TLΔEXP
TRENDM
TXC
TXI1
TXVA1
TXVA2
TXVAZ
TXW
RSUBS

APPENDIX B1 WORKSPACE MACRO

```
         AMANΔYEAR                   Probably redundant (jan 1982) ,but needed for technical reasons.
75
         BLDΔRATE1
1.04215                              Growth-rate of investments in residential housing,1976.
         BLDΔRATE2
1.00275                              Long term growth rate,investments in residential housing.(yearly change)
         EXOΔQTXVA1
0 0 0 0 0                            Value added tax on investments goods.Quarterly series starting with
         EXOΔQTXVA2                  first quarter 1977.
0.15 0.15 0.15 0.171 0.171Value added tax       .Quarterly series starting with first quarter 1977.
         EXOΔRI                     Rate of interest,quarterly series starting with first quarter 1977.
0.0981 0.0986 0.0979 0.098 0.0987 0.1011 0.0998 0.0999 0.0983 0.0954 0.0947 0.0992 0.1069
         EXOΔRIBWFOR                See appendix A.Quaterly series starting with first quarter 1977.
0.0614 0.0714 0.0746 0.0768 0.0847 0.0931 0.1191 0.1081 0.1093 0.1243 0.1508 0.1737
         EXOΔRIDEPFOR               See appendix A.Quarterly  series starting with first quarter 1977.
0.0519 0.0569 0.0677 0.0709 0.0731 0.081 0.0894 0.1154 0.1044 0.1056 0.1206 0.1471 0.17
         EXOΔTXC
0.564 0.58 0.575 0.575      Tax-rate,firms.Yearly series starting with 1977.
         EXOΔTXI1
0.392 0.395                 Income-tax rate ,households.Yearly series starting with 1977.
         EXOΔTXI2           Probably redundant,but needed for technical reasons.
0.00055124 0.0005466
         EXOΔTXW
0.280 0.289 0.280 0.294     Wage-tax rate.Yearly series starting with 1977.
         EXOΔTXWG
0.307 0.309 0.309 0.312     Wage-tax rate,goverment sector.Yearly series starting with 1977.
```

APPENDIX B1   WORKSPACE MACRO

```
      FIRSTΔSIMΔYEAR                                    First year of simulation.77 stands for 1977.
77
      GΔRATE1                                           Growth-rate of investments in the Goverment-sector,1976.
1.08732
      GΔRATE2                                           Long term growth-rate,investments in the government-sector,
1.03269                                                 yearly change.
      HISTΔTXVA2
  0.15            0.12            0.12                   0.15          VALUE-ADDED TAX ,"moms".
  0.15            0.15            0.15                   0.15
  0.15            0.15            0.15                   0.15          Rows:  Years ,starting with 1974.
  0.15            0.15            0.171                  0.171         Columns: Quarters .
      HOURSΔPERΔYEAR
1600                                                    Average number of working hours per year,1976.Roughly.
      HUSHALSDEP
1.133900000E11                                          Household´s bank deposits 1976.
      IMPLΔPRIS
   88.27192527    94.18785677   100                      107.3170732   YEARLY PRICE-INDEX SERIES,domestic prices.
   84.99043977    96.36711281   100                      103.5372849
   80.23072889    89.77451494   100                      106.6072365   Rows: Sector 1,2,3,4   (Industrial sectors)
   82.23609535    89.78433598   100                      111.8047673   Columns: Years;1974,1975,1976,1977


                                                                      YEARLY PRICE-INDEX SERIES,domestic prices.
      IMPLΔPRISΔIN
   74.98647333    90.96869026   100                      111.1604083  Rows: Sector 5,6,7,8,9,10
   74.10440123    96.87819857   100                       94.06345957
   83.47457627    85.2672751    100                      108.8657106   Columns: Years;1974,1975,1976,1977
   75.1002004     87.6252505    100                      105.511022
   83.97033657    89.50370793   100                      111.8653736
   81.640625      89.67633929   100                      111.1049107
```

APPENDIX B1 WORKSPACE MACRO

```
      IMPL∆PAREF
27.18 40.38 44.34 30.91 60.26 58.35 100 55.71 65.7 56.54
```

Value added shares,from input-output matrix 1976.

Value added share=1-inputshare.10 sectors.(percentage-figure$). This variable is used in some print-out functions in the MOSES-workspace.

```
      IN∆RATE1
1.08065
```

Growth rate of investments in non-industrial sectors (sector 5,6..10),1976.

```
      IN∆RATE2
1.02519
```

Long term growth rate of investments in non-industrial sectors,(yearly change).

```
      INIT∆GROWTH
0.064 0.056 0.06 0.023
```

Growth rate,labour productivity in the 4 industrial sectors (sector 1,2,3,4).Used in function secondary∆data in the initialization procedure.

APPENDIX B1 WORKSPACE MACRO

INPUT-OUTPUT matrix ,1976,in kr,expressed in 1975 year´s prices.
14 rows and 21 columns.

Documentation ,see section 3.

The first 10 columns.

```
       I076
      5272      2890      5869      1321       245        94         0      4192       942      1943
      2029      5195      4805      4465       908       117         0      3498       170      2035
       954      2354     12296       915       503       213         0      6294       171      3079
       803      2428      2041     14872      2078        87         0      2648       102      6484
      2400      1964       341     10768       383         1         0      1689         0       312
      2951       210        79        63        26       140         0       418         5         0
      4136       600       151       261       130       213         0      1009       488       842
      1235      1198      2838       961      1383       162         0     10928       708      9874
       904       941       475       485       238       171         0      1118       328      1010
      3293      3338      5919      4402      1792       640         0      9143       426     25656
        63        71       142     ¯2377       163        10         0       350         5      2261
      8736     14351     27422     19551     11452      2529         0     50892      6395     64383
       154      ¯119       178        51        41        35         0       238         0         1
     32933     35423     62558     55738     19341      4413         0     92417      9738    117881
```

The 11 remaining columns.Final Demand side of the matrix.

```
         I076
       758      5399         0         0         0       380      2754     12137    ¯11478       214     32933
      1953      9075       558         0       869      2170      1135     14735    ¯12965     ¯5329     35423
      3522     14903      3110         0      4836     10231      1687     29947    ¯24563     ¯7896     62558
      5102     55944       112         0       175       132       752      7450    ¯15980    ¯29493     55738
       243      6807         0         0       128       408       ¯95      1351     ¯3597     ¯3763     19341
        81        24         0         0         0         0        67      1134     ¯3015      2230      4413
       374      2346         0         0         0         0       188      1778     ¯6491     ¯6025         0
      2929     26970     17893     12436      4682       765      1067      7062     ¯4453     ¯6221     92417
       973      3580         0         0         0         0       ¯76       319      ¯306      ¯421      9738
      8849     30617       379         0       591         0      ¯316     10370    ¯16362     29496    117881
         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0
     24785    155664     22052     12436     11281     14085      7163     86284    ¯99209    ¯27209    430440
```

APPENDIX B1 WORKSPACE MACRO

INPUT-OUTPUT coefficients estimated from IO76.Vertical sum=1.13 rows,19 columns.

See function COEFF∆IO on p.12   in this appendix.See also section 3.

```
      IOCOEFF76                 The first 10 columns
0.16        0.08        0.09        0.02        0.01        0.02        0.00        0.05        0.10        0.02
0.06        0.15        0.08        0.08        0.05        0.03        0.00        0.04        0.02        0.02
0.03        0.07        0.20        0.02        0.03        0.05        0.00        0.07        0.02        0.03
0.02        0.07        0.03        0.27        0.11        0.02        0.00        0.03        0.01        0.06
0.07        0.06        0.01        0.19        0.02        0.00        0.00        0.02        0.00        0.00
0.09        0.01        0.00        0.00        0.00        0.03        0.00        0.00        0.00        0.00
0.13        0.02        0.00        0.00        0.01        0.05        0.00        0.01        0.05        0.01
0.04        0.03        0.05        0.02        0.07        0.04        0.00        0.12        0.07        0.08
0.03        0.03        0.01        0.01        0.01        0.04        0.00        0.01        0.03        0.01
0.10        0.09        0.09        0.08        0.09        0.15        0.00        0.10        0.04        0.22
0.00        0.00        0.00       ¯0.04        0.01        0.00        0.00        0.00        0.00        0.02
0.27        0.41        0.44        0.35        0.59        0.57        0.00        0.55        0.66        0.55
0.00        0.00        0.00        0.00        0.00        0.01        0.00        0.00        0.00        0.00
```

APPENDIX B1 WORKSPACE MACRO

```
          INPUT-OUTPUT coefficients,continued.
        IOCOEFF76                                      Column 11,12..19 .Final Demand coefficients.
0.03       0.03       0.00       0.00       0.00       0.03       0.38       0.14       0.12
0.08       0.06       0.03       0.00       0.08       0.15       0.16       0.17       0.13
0.14       0.10       0.14       0.00       0.43       0.73       0.24       0.35       0.25
0.21       0.36       0.01       0.00       0.02       0.01       0.10       0.09       0.16
0.01       0.04       0.00       0.00       0.01       0.03      ¯0.01       0.02       0.04
0.00       0.00       0.00       0.00       0.00       0.00       0.01       0.01       0.03
0.02       0.02       0.00       0.00       0.00       0.00       0.03       0.02       0.07
0.12       0.17       0.81       1.00       0.42       0.05       0.15       0.08       0.04
0.04       0.02       0.00       0.00       0.00       0.00      ¯0.01       0.00       0.00
0.36       0.20       0.02       0.00       0.05       0.00      ¯0.04       0.12       0.16
0.00       0.00       0.00       0.00       0.00       0.00       0.00       0.00       0.00
0.00       0.00       0.00       0.00       0.00       0.00       0.00       0.00       0.00
0.00       0.00       0.00       0.00       0.00       0.00       0.00       0.00       0.00


      LASTΔTXI2ΔYEAR              Probably redundant at present (jan 1982).
78


      LASTΔYEAR                   "Year-counter" in the model.Start-value=76 (stands for 1976).
76
      LGTRENDCH                   Trend growth in the goverment sector.Number of people added (net)
7875                              each quarter.
      LIQB
4.630900000E10                    See appendix A. 1976.
      LIQBFOR                     See appendix A. 1976.
10924000000
      LON                                                         Total wage-sum in the 4 industrial sectors,
8376281000 1.065502900E10  2.400718000E10  1.450339000E10        1,2,3,4  .      1976.
```

APPENDIX B1 WORKSPACE MACRO

```
      LONΔOFF                                  Total wage-sum in the goverment-sector.1976 and 1977.
5.807200000E10  6.994700000E10

      NMARKETS                                 Number of industrial sectors in the model.(=4)
4
      NWB                                      See appendix A.1976.
7.779457670E10
      POSG
¯7.396300000E10                                See appendix A.1976.
      QCHRI
0.0002                                         Change in rate of interest,last quarter 1976.
      QINPAY
3.240000000E10                                 See appendix A.Last quarter 1976.
      QTTAX
3.780000000E10                                 Total tax receipts by the government,last quarter 1976.
      RI
0.0979                                         Rate of interest,last quarter 1976.
      RU
0.016                                          Rate of unemployment,1976.
      SALES76 *)
2.913290600E10  3.788546400E10  7.025235800E10  6.988083000E10   Total sales in the 4 industrial sectors,
      THISΔYEAR                                                  (sector 1,2,3,4) in producer"s prices,1976.
76                         "Year counter"in the model.
      TIM                                                        Total number of working-hours during a
204338800 264942430 606865110 398119570                          a year in the 4 industrial sectors,1976.
      TIMΔOFF
1465950000 1498760000
                              Total number of working-hours
                              during a year,in the government-
                              sector.1976 and 1977 .
```

*) Since 1983 the following variable is,usually,used instead of SALES76:

```
      NYSALES76
3.660000000E10  3.930000000E10  6.950000000E10  6.180000000E10
```

APPENDIX B1 WORKSPACE MACRO

Export price index,the four industrial sectors.(sector 1,2,3,4)
Price-series, 38 quarters.   1971:1 ..1980:2

These series are used to form future price-series.See function TL∆E(P∆PRIS∆76
on p.12 in this appendix.

```
   TL∆EXP
101.4       101         98.6        96.5        95.2        95.2        96.6
      99.7       108.3       120.5       132.3       143.9       168.7       177      SECTOR 1
     183.1       190.4       184.4       177.1       172.6       173.9       177.3
     182.9       185.1       181.7       183         182.6       186.7       180.4
     175.7       178.1       185.7       193         203.9       210.2       223.3
     233.5       252         255.6
 99.4       100         100.2       101.8       103.5       105.7       105.3
     105.4       109.2       113.4       118.2       123.2       149.4       163.5
     180.5       188         194.4       190.6       187.2       184.9       182.4
     183.9       184.8       182.4       184.2       187.5       195.9       199      SECTOR 2
     203         201.1       204.6       207         216.8       228.8       235.8
     244.5       258.6       266.5
 99.9        99.6       100         101.5       103.2       104.8       108.1
     108.5       110.3       111.8       115.9       120.9       127.9       131.2
     137         141         144.9       150         154.3       157         165.2    SECTOR 3
     168.3       173.2       174.2       179.9       183         188.9       190.5
     200.3       202.8       207.4       209.9       213.2       220.1       224.3
     225.7       235.1       242.7
 99         100.5       100.4       101.5       104.2       103.2       106.3
     106.5       112.3       115.8       118.7       122.9       133.2       139.5
     142.7       145.5       148.2       150.3       152.6       155.2       162      SECTOR 4
     164.8       167.8       170.5       178.2       183.5       185.7       191.7
     200.8       203.3       204.1       208.3       219.8       221.9       226.8
     230.2       240.6       239.4
```

APPENDIX B1 WORKSPACE MACRO

```
     TRENDM              Trend change (quarterly change) in domestic price index
 0.0133237               for sector 5,6..10 .
 0.00840916
 0.01729822
 0.0124923
 0.00989622
 0.0141188

     TXC
0.561                      Corporate tax-rate.   1976.
     TXI1                  Income tax-rate (households).1976.
0.354
     TXVA1                 Value added tax,investment goods.Last quarter 1976.
0
     TXVA2                 Value added tax-rate,"moms".  Last quarter 1976.
0.15
     TXW                   Wage-tax rate . 1976.
0.267
     TXWG                  Wage-tax (government-sector) rate.1976.
0.277
     RSUBS
     -----                 Subventions to the 4 industrial sectors,(sector 1,2,3,4).
 0
 0                         1976-."Food subventions " to sector 4.
 0
 0.035                     Subventions are expressed as fractions of sales in each sector.
```

APPENDIX B1 WORKSPACE MACRO

```
     ∇AGGRITAX[⎕]∇
   ∇  R←AGGRITAX Y
[1]   R←TXI1×Y
   ∇
```

This function estimates income-tax ,in kr.

Usage: See function QDI_INIT,subfunction in appendix C.

```
     ∇COEFF∆IO[⎕]∇
   ∇  COEFF∆IO;S;SUMMA;SUMMAMAT
[1]   S←(13 19)↑IO76
[2]   SUMMA←+/[1]S
[3]   SUMMAMAT←(13 19)⍴SUMMA
[4]   IOCOEFF76←S÷SUMMAMAT
[5]  ⍝ 0÷0 GER 1,MASTE KORRIGERAS
[6]   IOCOEFF76[;7]←0
   ∇
```

This function estimates input-output coefficients from the input-output matrix IO76.

```
     ∇TL∆EXP∆PRIS∆76[⎕]∇
   ∇  R←TL∆EXP∆PRIS∆76 N;AR;CYCL;DU;DUM;DUMMY;FUT
[1]  ⍝ EXPORT-PRICE CHANGES WITH NEW DATA, COVERING PERIOD
[2]  ⍝ 1971:1 THROUGH 1980:2
[3]  ⍝ OUTPUT IS QUARTERLY CHANGES FROM 1Q-76 UP TO END OF
[4]  ⍝ SIMULATION = ARG. N. DUR AND NDUR ON THE AVER-
[5]  ⍝ AGE TREND 1971-76. RAW AND IMED WITH A CYCLE FROM
[6]  ⍝ 1980:3 AS THE ONE FROM 1975:1
[7]   AR←(1↓⍴TL∆EXP)
[8]   FUT←(1+4×N)-(¯1+AR-22)
[9]   CYCL←(¯16+1↓⍴TL∆EXP)
[10]  DUMMY←(¯1+(TL∆EXP[;1+⍳AR-1]÷TL∆EXP[;⍳AR-1]))
[11]  DUM← 0 22 ↓DUMMY
[12]  DU←(4,FUT)⍴((1,FUT)⍴(DUMMY[1;15+⍳CYCL])),[1]((1,FUT)⍴(DUMMY[2;15+⍳CYCL])),[1](⍉(FUT,2)⍴(+/DUMMY[3 4 ;])
     ÷(AR-1))
[13]  DU[;⍳FUT]←DU[;⍳FUT]×2÷3
[14] ⍝TEMPORARY CHANGE 4/12 1980,TO LOWER FOREIGN INFLATION RATE
[15]  R←DUM,DU
   ∇
```

This function estimates future export price-changes.

APPENDIX B1 WORKSPACE SI76 - MICRO DATA

A vector telling what firm-group a certain firm belongs to.

```
      FIRMID
16 16 18 0 25 25 14 27 24 16 8 0 0 18 32 21 0 18 32 21 0 19 19 1 8 8 0 7 19 0 16 0 22 7 14 3 9 9 9 9 22 16 18 25 28
   28 17 0 0 27 32 0 30 0 18 6 24 1 2 2 2 15 5 5 0 3 23 23 28 11 11 26 7 11 31 3 23 23 23 28 28 7 20 20 20 23 12 3
 23 1 1 11 30 31 29 0 4 11 0 23 28 30 30 26 26 26 0 16 15 13 13 13 13 13 33 34 35 36 37 38 39 40
```

Firm-code.

```
      LIST
1,01 1.02 1.03 1.07 1.08 1.09 1.12 1.13 1.17 1.18 1.26 1.29 1.41 1.44 2.01 2.02 2.03 2.06 2.07 2.12 2.13 2.19 2.21
 2.26 2.27 2.28 2.3 2.31 2.32 2.33 2.35 2.4 2.42 2.44 2.46 2.47 2.51 2.61 2.72 3.01 3.05 3.06 3.07 3.08 3.09 3.1
 3.12 3.13 3.16 3.18 3.19 3.2 3.22 3.23 3.25 3.29 3.32 3.34 3.36 3.37 3.38 3.39 3.4 3.41 3.43 3.44 3.47 3.48 3.54
 3.55 3.56 3.57 3.58 3.61 3.68 4.06 4.22 4.3 4.32 4.33 4.38 4.39 4.44 5.01 5.03 5.09 5.11 5.14 5.18 5.19 5.24 5.25
 1.91 1.92 1.93 3.91 4.91 4.92 4.93 4.94
```

A vector telling what sector (1,2,3 or 4) a certain firm belongs to.

```
      RΔMARKET
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3
  3 33 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 3 3 3 3 4 4 4 2 2 2 2 3 2 2 2 2 2 1
  1 13 4 4 4 4
```

```
      ρX
122 50
      ρFΔDATA
40 26
```

All firm-data lie in an enormous matrix with 122 rows and 50 columns.

All firm-group data lie in an enormous matrix with 40 rows and 26 columns.

X and FΔDATA are not listed in this documentation,because the figures are given by the firms provided that the figures aren´t published.

APPENDIX B2   MACRO DATA 1982 (WORKSPACE MACR82)

```
*****    AMANΔYEAR                 *****
75
------------------------------------------------------------
*****    BLDΔRATE1                 *****
1.05
------------------------------------------------------------
*****    BLDΔRATE2                 *****
1.12
------------------------------------------------------------
*****    EXOΔQTXVA1                *****
0 0
------------------------------------------------------------
*****    EXOΔQTXVA2                *****
0.11 0.11 0.11 0.11
------------------------------------------------------------
*****    EXOΔRI                    *****
0.1326 0.1284 0.1287 0.1237 0.1198 0.1223
------------------------------------------------------------
*****    EXOΔRIBWFOR               *****
0.0958 0.0971 0.1035 0.101 0.105 0.1198
------------------------------------------------------------
*****    EXOΔRIDEPFOR              *****
0.0921 0.0934 0.0998 0.0973 0.1013 0.1161
------------------------------------------------------------
*****    EXOΔTXC                   *****
0.583 0.527 0.527
------------------------------------------------------------
*****    EXOΔTXI1                  *****
0.28
------------------------------------------------------------
*****    EXOΔTXI2                  *****
0.00055124 0.0005466
------------------------------------------------------------
*****    EXOΔTXW                   *****
0.275
------------------------------------------------------------
*****    EXOΔTXWG                  *****
0.288
------------------------------------------------------------
```

APPENDIX B2 continued

```
-----------------------------------------------------------------
*****    FIRSTΔSIMΔYEAR          *****
83
-----------------------------------------------------------------
*****    GΔRATE1                 *****
1.06
-----------------------------------------------------------------
*****    GΔRATE2                 *****
1.03269
-----------------------------------------------------------------
*****    HISTΔT×VA2              *****
0.11 0.11 0.11 0.11
0.12 0.12 0.12 0.12
0.11 0.11 0.11 0.11
0.11 0.11 0.11 0.11
-----------------------------------------------------------------
*****    HOURSΔPERΔYEAR          *****
1600
-----------------------------------------------------------------
*****    HUSHALLSDEP             *****
2.236E11
-----------------------------------------------------------------
*****    IMPLPΔREF               *****
15.7019888 35.3175231 39.4864013 32.38852283 57.58724748 55.10135583 0
    57.7265516 62.77758359 51.78873938
-----------------------------------------------------------------
*****    IMPLΔPRIS               *****
 88.5  90.3 100   111.4
 83.2  90.2 100   110.6
 87.1  92.2 100   110.5
 78.6  88.1 100   110.5
-----------------------------------------------------------------
*****    IMPLΔPRISΔIN            *****
 83.7  90.4 100   110.4
 83.7  90.4 100   110.4
 83.7  90.4 100   110.4
 81.1  91.4 100   109
 78.6  88.1 100   110.5
 78.6  88.1 100   110.5
-----------------------------------------------------------------
*****    INITΔGROWTH             *****
0.064 0.056 0.06 0.023
-----------------------------------------------------------------
*****    INΔRATE1                *****
1.18
-----------------------------------------------------------------
*****    INΔRATE2                *****
1.02519
-----------------------------------------------------------------
```

APPENDIX B2 continued

```
 IO76          INPUT-OUTPUT matrix, 1982, 14 rows and 11 columns, below: the first 10 columns
11036    7975   12347    2637     553     116       0    8227    2909    5804
 5839   16460   13209   11089    2764     261       0   11600    1271   13399
 1763    6031   24928    1301     857     224       0   11136     420    7236
 1364    6102    3607   26732    3888      75       0    4041     171   15960
 4966    5292     698   19734     748       1       0    3155       0     824
 5481       0       0       0       0     109       0       0       0       0
 8597    1670     381     540     275     243       0    1997    1494    2428
 1305    1360    3331       0    1953      85       0   16617    1567   20444
 2023    2765    1286    1158     557     205       0    2562    1063    3447
 7265    9672   13108    8863    3778     758       0   18508    1497   71042
    0       0       0       0       0       0       0       0       0       0
 9246   31097   46845   33819   20429    2549       0  104724   17060  147745
    0    ¯373   ¯1103   ¯1456    ¯327       0       0   ¯1153    ¯276   ¯3046
58884   88049  118635  104416   35474    4626       0  181413   27175  285284
```

The remaining 11 columns. Final demand side of the matrix

```
 IO76
 1710   10900       0       0       0     281   ¯1290   20367  ¯25254     566   58884
 4405   18000       0       0    2050    1610   ¯1500   43914  ¯42223  ¯14098   88049
 5700   30800   12673       0   11408   13373   ¯2841   64969  ¯50458  ¯20883  118635
11510  115000       0       0     412     100    ¯202   18523  ¯24872  ¯77994  104416
  546   13000       0       0     300     300     427    1967   ¯6530   ¯9952   35474
  182       0       0       0       0       0       0    2828   ¯2596   ¯1378    4626
  845   14500       0       0       0       0     ¯90    3703  ¯20652  ¯15931       0
 6615   67000   28742   27028   11043     570     ¯54   12881   ¯2167  ¯16907  181413
 2198   11800       0       0       0       0    ¯490     340    ¯624   ¯1117   27175
19970   52000       0       0    1394       0    ¯950   33128  ¯32754   78006  285284
    0       0       0       0       0       0       0       0       0       0       0
    0       0       0       0       0       0       0       0       0       0       0
    0       0       0       0       0       0       0       0       0    7733       0
53681  333000   41415   27028   26611   16234   ¯6990  202620 ¯208130  ¯71955  903955
```

NOTE: The name of the matrix is misleading. A more logical name would have been IO82.
We have, however,refrained from changing names of variables.

APPENDIX B2 continued

INPUT-OUTPUT coefficients for 1982 estimated from IO76 on the previous page.

The matrix contains 13 rows and 19 columns. The vertical sum is equal to 1.0

```
          IOCOEFF76
0.19      0.09      0.10      0.03      0.02      0.03      0.00      0.05      0.11      0.02
0.10      0.19      0.11      0.11      0.08      0.06      0.00      0.06      0.05      0.05
0.03      0.07      0.21      0.01      0.02      0.05      0.00      0.06      0.02      0.03
0.02      0.07      0.03      0.26      0.11      0.02      0.00      0.02      0.01      0.06
0.08      0.06      0.01      0.19      0.02      0.00      0.00      0.02      0.00      0.00
0.09      0.00      0.00      0.00      0.00      0.02      0.00      0.00      0.00      0.00
0.15      0.02      0.00      0.01      0.01      0.05      0.00      0.01      0.05      0.01
0.02      0.02      0.03      0.00      0.06      0.02      0.00      0.09      0.06      0.07
0.03      0.03      0.01      0.01      0.02      0.04      0.00      0.01      0.04      0.01
0.12      0.11      0.11      0.08      0.11      0.16      0.00      0.10      0.06      0.25
0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
0.16      0.35      0.39      0.32      0.58      0.55      0.00      0.58      0.63      0.52
0.00      0.00     ¯0.01     ¯0.01     ¯0.01      0.00      0.00     ¯0.01     ¯0.01     ¯0.01
```

APPENDIX B2 continued

INPUT-OUTPUT coefficients for 1982, continued. Below:column 11-19.

```
IOCOEFF76
0.03   0.03   0.00   0.00   0.00   0.02   0.18   0.10   0.12
0.08   0.05   0.00   0.00   0.08   0.10   0.21   0.22   0.20
0.11   0.09   0.31   0.00   0.43   0.82   0.41   0.32   0.24
0.21   0.35   0.00   0.00   0.02   0.01   0.03   0.09   0.12
0.01   0.04   0.00   0.00   0.01   0.02  ¯0.06   0.01   0.03
0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.01   0.01
0.02   0.04   0.00   0.00   0.00   0.00   0.01   0.02   0.10
0.12   0.20   0.69   1.00   0.42   0.04   0.01   0.06   0.01
0.04   0.04   0.00   0.00   0.00   0.00   0.07   0.00   0.00
0.37   0.16   0.00   0.00   0.05   0.00   0.14   0.16   0.16
0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
```

source: Fredrik Bergholm

```
-----------------------------------------------------------------
*****     LASTΔTXI2ΔYEAR           *****
78
-----------------------------------------------------------------
*****     LASTΔYEAR                *****
82
-----------------------------------------------------------------
*****     LGTRENDCH                *****
7875
-----------------------------------------------------------------
*****     LIBB                     *****
1.02018E11
-----------------------------------------------------------------
*****     LIBBFOR                  *****
2.5115E10
-----------------------------------------------------------------
*****     LON                      *****
7344000000 2.2779E10 4.072E10 2.4456E10
-----------------------------------------------------------------
*****     LONΔOFF                  *****
1.29076E11 1.39981E11
```

APPENDIX B2 continued

```
----------------------------------------------------------------
*****    NEWEXOΔQTXVA2          *****
0.19 0.19 0.19 0.19
----------------------------------------------------------------
*****    NEWHISTΔTXVA2          *****
0.171  0.171  0.171  0.19
0.19   0.19   0.19   0.1835
0.177  0.177  0.177  0.177
0.19   0.19   0.19   0.19
----------------------------------------------------------------
*****    NEWPOSG                *****
¯1E11
----------------------------------------------------------------
*****    NEWTXVA2               *****
0.19
----------------------------------------------------------------
*****    NMARKETS               *****
4
----------------------------------------------------------------
*****    NWB                    *****
2.2E11
----------------------------------------------------------------
*****    NYSALES76              *****
5.8562E10 8.7381E10 1.17627E11 1.04227E11
----------------------------------------------------------------
 ----------------------------------------------------------------
 *****    POSG                   *****
 ¯3.77089E11
 ----------------------------------------------------------------
 *****    QCHRI                  *****
 ¯0.0003
 ----------------------------------------------------------------
 *****    QINPAY                 *****
 5.019225E10
 ----------------------------------------------------------------
 *****    QPFOR                  *****
 104.050405 105.5776892 103.755102 102.9388403
 ----------------------------------------------------------------
 *****    QTTAX                  *****
 7.198675E10
 ----------------------------------------------------------------
 *****    RI                     *****
 0.1329
 ----------------------------------------------------------------
 *****    RU                     *****
 0.031
 ----------------------------------------------------------------
 *****    RSUBS                  *****
 0
 0
 0
 0.035
 ----------------------------------------------------------------
 *****    SALES82                *****
 5.8562E10 8.7381E10 1.17627E11 1.04227E11
 ----------------------------------------------------------------
 *****    THISΔYEAR              *****
 82
 ----------------------------------------------------------------
 *****    TIM                    *****
 100130000 316880000 526070000 364690000
 ----------------------------------------------------------------
 *****    TIMΔOFF                *****
 1698580000 1730070000
 ----------------------------------------------------------------
```

NOTE: A number of new variables,compared to Workspace MACRO are introduced.

APPENDIX B2 continued

```
      Export prices,cf. Workspace MACRO. The change of these  prices are, now,
*****    TL∆DEXP                  *****  also a variable.
¯0.0524861879 ¯0.0364431487  0.0015128593  0.0196374622  0.04
      0.0213675214 ¯0.0083682008  0.0014064698 ¯0.0014044944
      0.0070323488 ¯0.0125698324 ¯0.0325318246  0.0014619883
      0.0291970803  0.0439716312  0.0543478261  0.0476804124
      0.0479704797  0.0492957747  0.0872483222  0.0360082305
      0.005958292  ¯0.0039486673 ¯0.0039643211  0.007960199
      0.0078973347  0.0391772772  0.0527803959  0.0188003581
     ¯0.0237258348  0.0405040504  0.0311418685  0.0343959732
      0.0373073804  0.0132916341  0.0462962963  0.0678466077
      0.0165745856 ¯0.0095108696
¯0.0093457944 ¯0.0215633423 ¯0.0123966942 ¯0.0264993027  0.0085959885
      0.0042613636  0.0014144272 ¯0.0042372881  0.0184397163
      0.0292479109  0.0365358593  0.0195822454 ¯0.0038412292
      0.0077120823  0.0178571429  0.0388471178  0.0458383595
      0.0253748558  0.03712036    0.0455531453  0.0394190871
      0.0119760479  0.0059171598  0.0460784314  0.0215557638
      0.0385321101  0.0600706714  0.03          0.0113268608
      0.004         0.0557768924  0.039245283   0.0196078431
      0.0142450143  0.0126404494  0.0194174757  0.0231292517
      0.0239361702  0.0123376623
 0.0246710526  0.0321027287  0.0171073095  0.0489296636  0.0247813411
      0.0199146515  0.0153417015  0.0288461539  0.0240320427
      0.0273794003  0.0139593909  0.0450563204  0.0179640719
      0.0164705882  0.0173611111  0.0182025028  0.0290502793
      0.0152008686  0.0085561497  0.0339342524  0.0164102564
      0.0211907165  0.0088932806  0.027424094   0.0314585319
      0.0175600739  0.0272479564  0.0477453581  0.0210970464
      0.0123966942  0.0375510204  0.064516129   0.0206947524
      0.0108616944  0.0107449857  0.019135365   0.0111265647
      0.0068775791  0.0054644809
 0.0101010101  0.0216666667  0.0114192496  0.0483870968  0.0215384615
      0.0256024096  0.0073421439  0.0422740525  0.0265734266
      0.0136239782  0.0309139785  0.0521512386  0.0173482032
      0.0024360536  0.0194410693  0.0452920143  0.0250855188
      0.0222469411 ¯0.0043525571  0.0655737705  0.0092307692
      0.0264227642  0.0198019802  0.0504854369  0.0194085028
      0.0108794198  0.0511210762  0.0341296928  0.0222772277
      0.01614205    0.0293884035  0.0493827161  0.0176470588
      0.0216763006  0.013437058   0.0272156315  0.0360054348
      0.0131147541  0.0077669903
------------------------------------------------------------

*****    TL∆EXP                   *****
 72.4  68.6  66.1  66.2  67.5  70.2  71.7  71.1  71.2  71.1  71.6
     70.7  68.4  68.5  70.5  73.6  77.6  81.3  85.2  89.4  97.2 100.7
    101.3 100.9 100.5 101.3 102.1 106.1 111.7 113.8 111.1 115.6 119.2
    123.3 127.9 129.6 135.6 144.8 147.2 145.8
 74.9  74.2  72.6  71.7  69.8  70.4  70.7  70.8  70.5  71.8  73.9
     76.6  78.1  77.8  78.4  79.8  82.9  86.7  88.9  92.2  96.4 100.2
    101.4 102   106.7 109   113.2 120   123.6 125   125.5 132.5 137.7
    140.4 142.4 144.2 147   150.4 154   155.9
 60.8  62.3  64.3  65.4  68.6  70.3  71.7  72.8  74.9  76.7  78.8
     79.9  83.5  85    86.4  87.9  89.5  92.1  93.5  94.3  97.5  99.1
    101.2 102.1 104.9 108.2 110.1 113.1 118.5 121   122.5 127.1 135.3
    138.1 139.6 141.1 143.8 145.4 146.4 147..5  97.5  98.4
    101   103   108.2 110.3 111.5 117.2 121.2 123.9 125.9 129.6 136
    138.4 141.4 143.3 147.2 152.5 154.5 155.7
------------------------------------------------------------
```

APPENDIX B2 continued

```
*****    TRENDM                      *****
0.0133237
0.00840916
0.01729822
0.0124923
0.00989622
0.0141188
--------------------------------------------------------------
*****    TXC                         *****
0.579
--------------------------------------------------------------
*****    TXI1                        *****
0.276
----- --------------------------------------------------------
*****    TXVA1                       *****
0
--------------------------------------------------------------
*****    TXVA2                       *****
0.11
--------------------------------------------------------------
*****    TXVAZ                       *****
0.15
--------------------------------------------------------------
*****    TXW                         *****
0.33
s-------------------------------------------------------------
*****    TXWG                        *****
0.361
--------------------------------------------------------------
```

APPENDIX B2 continued

For some reason, these exogenous price series are alsc contained in MACR82.

```
   EXOΔGDPFOR
 0.0344   0.0373   0.0133   0.0463   0.0678   0.0166  ¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095
¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095
¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095
¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095  ¯0.0095
 0.0196   0.0142   0.0126   0.0194   0.0231   0.0239   0.0123   0.0123   0.0123   0.0123
 0.0123   0.0123   0.0123   0.0123   0.0123   0.0123   0.0123   0.0123   0.0123   0.0123
 0.0123   0.0123   0.0123   0.0123   0.0123   0.0123   0.0123   0.0123   0.0123   0.0123
 0.0123   0.0123   0.0123   0.0123   0.0123   0.0123   0.0123   0.0123   0.0123   0.0123
 0.0207   0.0109   0.0107   0.0191   0.0111   0.0069   0.0055   0.0055   0.0055   0.0055
 0.0055   0.0055   0.0055   0.0055   0.0055   0.0055   0.0055   0.0055   0.0055   0.0055
 0.0055   0.0055   0.0055   0.0055   0.0055   0.0055   0.0055   0.0055   0.0055   0.0055
 0.0055   0.0055   0.0055   0.0055   0.0055   0.0055   0.0055   0.0055   0.0055   0.0055
 0.0176   0.0217   0.0134   0.0272   0.0360   0.0131   0.0078   0.0078   0.0078   0.0078
 0.0078   0.0078   0.0078   0.0078   0.0078   0.0078   0.0078   0.0078   0.0078   0.0078
 0.0078   0.0078   0.0078   0.0078   0.0078   0.0078   0.0078   0.0078   0.0078   0.0078
 0.0078   0.0078   0.0078   0.0078   0.0078   0.0078   0.0078   0.0078   0.0078   0.0078
```

```
FIRMΔEXOΔGDPFOR
 0.0311   0.0344   0.0373   0.0133  ¯0.0113  ¯0.0113  ¯0.0113  ¯0.0113  ¯0.0015  ¯0.0015
¯0.0015  ¯0.0015   0.0063   0.0063   0.0063   0.0063   0.0119   0.0119   0.0119   0.0119
 0.0119   0.0119   0.0119   0.0119   0.0119   0.0119   0.0119   0.0119   0.0119   0.0119
 0.0119   0.0119   0.0119   0.0119   0.0119   0.0119   0.0119   0.0119   0.0119   0.0119
 0.0393   0.0196   0.0142   0.0126  ¯0.0122  ¯0.0122  ¯0.0122  ¯0.0122   0.0136   0.0136
 0.0136   0.0136   0.0093   0.0093   0.0093   0.0093   0.0111   0.0111   0.0111   0.0111
 0.0111   0.0111   0.0111   0.0111   0.0111   0.0111   0.0111   0.0111   0.0111   0.0111
 0.0111   0.0111   0.0111   0.0111   0.0111   0.0111   0.0111   0.0111   0.0111   0.0111
 0.0645   0.0207   0.0109   0.0107   0.0115   0.0115   0.0115   0.0115   0.0097   0.0097
 0.0097   0.0097  ¯0.0314  ¯0.0314  ¯0.0314  ¯0.0314   0.0093   0.0093   0.0093   0.0093
 0.0093   0.0093   0.0093   0.0093   0.0093   0.0093   0.0093   0.0093   0.0093   0.0093
 0.0093   0.0093   0.0093   0.0093   0.0093   0.0093   0.0093   0.0093   0.0093   0.0093
 0.0494   0.0176   0.0217   0.0134   0.0284   0.0284   0.0284   0.0284   0.0102   0.0102
 0.0102   0.0102   0.0089   0.0089   0.0089   0.0089   0.0098   0.0098   0.0098   0.0098
 0.0098   0.0098   0.0098   0.0098   0.0098   0.0098   0.0098   0.0098   0.0098   0.0098
 0.0098   0.0098   0.0098   0.0098   0.0098   0.0098   0.0098   0.0098   0.0098   0.0098
```

APPENDIX C THE INITIALIZATION CODE, MAIN CODE

The functions listed below are the functions stored (jan 82) in workspace INIT. They are described in Sections 2 and 4 in Part 2.

The functions have line-numbers leftmost. A function stands between the symbols ∇ (upside-down delta).

Local variables in each function can be found on line zero after the semicolon (;). After the function-name a parameter to the function may appear. For example: **START** N. N is a parameter (an integer) to the function **START**.

APPENDIX C FUNCTION START

```
    ∇  START N
[1]     ⍎')MAXCORE 160 '
[2]    ⍝ NEEDED SPACE IN COMPUTER...
[3]     WORKSPACENAME←'R',⍕N
[4]    ⍝THE RESULT FROM THE INITIALIZATION WILL BE STORED IN A WORKSPACE
[5]    ⍝CALLED RXX,WHERE XX IS THE NUMBER N GIVEN IN THE CALL START N
[6]      ' RESULT FROM INITIALIZATION IS STORED IN WORKSPACE ',⍕WORKSPACENAME
[7]    ⍝
[8]    ⍝WORKSPACENAME IS USED IN FUNCTION OUTPUT∆OPERATIONS...
[9]    ⍝
[10]    NYR←30
[11]   ⍝NUMBER OF YEARS TO INITIALIZE VARIABLES.
[12]   ⍝CAN BE CHANGED IN FUNCTION ISTARTXX.
[13]   ⍝
[14]    ⍎')COPY FUNCTI MODADD MODDEL MODSUBST SCANMAT PACK ENS EQUALS ABOVE'
[15]    NAME←'ISTART',⍕N
[16]    ⍎')COPY ISTART'
[17]   ⍝START-FUNCTIONS SHOULD LIE IN WORKSPACE ISTART
[18]    ⍎NAME
[19]   ⍝THE LINE ABOVE MEANS THAT THE FUNCTION ISTARTXX WILL BE EXECUTED.
[20]   ⍝XX IS THE NUMBER OF THE INITIALIZATION.(XX=N)
[21]   ⍝ISTARTXX IS SPECIFIC FOR A CERTAIN EXPERIMENT.
[22]   ⍝IN ISTARTXX ONE CAN CHANGE LINES BELOW WITH 3 SPECIAL
[23]   ⍝FUNCTIONS MODADD,MODSUBST,MODDEL.
[24]   ⍝THUS ISTARTXX CAN CHANGE THE PROGRAM BELOW DURING EXECUTION.
[25]   ⍝
[26]   ⍝
[27]    SI∆INIT NYR
[28]    'INITIALIZATION COMPLETED'
[29]    ⍎')CLEAR'
[30]    ⍎')WS CLEAR'
    ∇
```

APPENDIX C   FUNCTION SIΔINIT

```
    ∇  SIΔINIT NYR;DUMMY
     ⍝ DUMMY←⍎')COPY SI76 FΔDATA X FIRMID'
     ⍝LINE ABOVE EXECUTED IN FUNCTION ESTABLISHMENTS
      DUMMY←⍎')COPY MACRO'
      DUMMY←⍎')COPY FUNCTIONS'
     ⍝
     ⍝FIRMDATA FROM WORKSPACE SI76
     ⍝MACRODATA FROM WORKSPACE MACRO
     ⍝HELPFUNCTIONS FROM WORKSPACE FUNCTIONS
     ⍝
     ⍝
      TESTUTSKRIFT←0
     ⍝NYR=NUMBER OF YEARS TO RUN THE SIMULATION.
     ⍝
     ⍝
      NQR←4×NYR
     ⍝NQR=NUMBER OF QUARTERS
      NMARKETS←4
     ⍝
     ⍝
      TAXΔPARAMETERS
      PUBLICΔSECTOR
      MONETARY
      MARKETS
      HOUSEHOLDS
      ESTABLISHMENTS
     ⍝FUNCTION DISPOSEΔVARΔINPUT DELETES VARIABLES FROM WORKSPACE MACRO...
     ⍝XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
     ⍝  SECOND PART OF INITIALIZATION
     ⍝XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
     ⍝THE FOLLOWING VARIABLES ARE NEEDED IN THE SECOND PART
     ⍝OF THE INITIALIZATION.COPIES ARE TAKEN BECAUSE IT SEEMS LOGICAL
     ⍝TO FORBID READING FROM INPUTFILES IN SECOND PART OF
     ⍝INITIALIZATION...
      GROWTH←INITΔGROWTH
      TXVA2COPY←TXVA2
      RUΔCOPY←RU
      TXWCOPY←TXW
```

APPENDIX C FUNCTION STARTIT (cont.)

```
[38]    TXWGCOPY←TXWG
[39]    QINPAYCOPY←QINPAY
[40]    RIΔCOPY←RI
[41]    TXI1COPY←TXI1
[42]   ⍝FROM NOW ON NO MORE READING FROM INPUT-WORKSPACES
[43]   ⍝(MACRO AND SI76).THERE WILL BE,ONLY,FURTHER WORK WITH
[44]   ⍝VARIABLES AND PARAMETER-SETTING.
[45]    DISPOSEΔVARΔINPUT
[46]    MARKETSΔDATA
[47]    SECONDARYΔDATA
[48]    PUBLICΔDATA
[49]    MONETARYΔDATA
[50]    HOUSEHOLDSΔDATA
[51]   ⍝
[52]   ⍝
[53]    OUTPUTΔOPERATIONS
[54]   ⍝THIS FUNCTION HANDLES OUTPUT.(UNNECESSARY VARIABLES ARE DELETED).
[55]    'TESTUTSKRIFT2'
      ∇
```

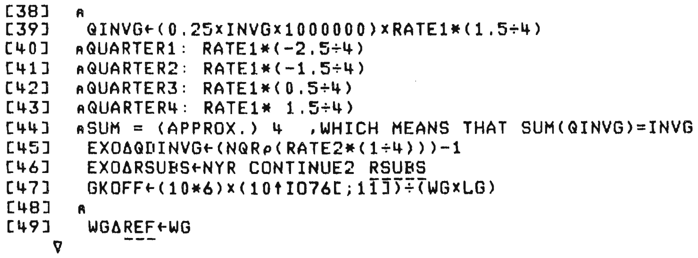
APPENDIX C FUNCTION TAXΔPARAMETERS

```
    ∇  TAXΔPARAMETERS
[1]    ⍝VARIABLES IN WORKSPACE MACRO WHICH IS FINAL OUTPUT FROM INITIALIZATION:
[2]    ⍝ TXVA1,TXVA2
[3]    ⍝
[4]    ⍝ OTHER VARIABLES IN TAXΔPARAMETERS WHICH WILL BE FINAL
[5]    ⍝ OUTPUT FROM INITIALIZATION:
[6]    ⍝ ALL EXO-VARIABLES TO THE LEFT OF '←' BELOW AND TXI3
[7]    ⍝
[8]     EXOΔQCHTXVA1←NQR↑DIFF EXOΔQTXVA1
[9]     EXOΔQCHTXVA2←NQR↑DIFF EXOΔQTXVA2
[10]    EXOΔTXC←NYR CONTINUE1 EXOΔTXC
[11]    EXOΔTXI1←NYR CONTINUE1 EXOΔTXI1
[12]    EXOΔTXW←NYR CONTINUE1 EXOΔTXW
[13]    EXOΔTXWG←NYR CONTINUE1 EXOΔTXWG
[14]    TXI3←1.6
    ∇
```

APPENDIX C FUNCTION PUBLIC∆SECTOR

```
    ∇  PUBLIC∆SECTOR;ALG;QLG;WAGES;RATE1;RATE2;QCHLG
[1]    ⍝
[2]    ⍝  VARIABLES IN PUBLIC∆SECTOR WHICH WILL BECOME
[3]    ⍝ FINAL OUTPUT FROM ININTIALIZATION:
[4]    ⍝ OMEGAG,QINVG,EXO∆QDINVG,EXO∆RSUBS,QWG,WG,LG,WG∆REF
[5]    ⍝GKOFF,EXO∆REALCHLG
[6]    ⍝
[7]    ⍝
[8]     OMEGAG←10↑IOCOEFF76[;13]
[9]     INVG←IO76[14;13]
[10]    RATE1←G∆RATE1
[11]    RATE2←G∆RATE2
[12]   ⍝ RATE1=YEARLY PERCENTAGE CHANGE IN INVG,RATE2=TREND CHANGE
[13]    ALG←TIM∆OFF÷HOURS∆PER∆YEAR
[14]   ⍝
[15]    WAGES←2⍴0
[16]    WAGES[1]←LON∆OFF[1]÷ALG[1]
[17]    WAGES[2]←LON∆OFF[2]÷ALG[2]
[18]   ⍝
[19]    QLG←(4×(⍴ALG))⍴0
[20]    QLG←MAKEQUARTERS ALG
[21]   ⍝RESULT FROM MAKEQUARTERS:QLG=
[22]   ⍝AVERAGE LABOUR FORCE IN EACH QUARTER.QLG(1)=
[23]   ⍝QUARTER 1 BASE YEAR AND SO ON...
[24]    QCHLG←DIFF QLG
[25]    LG←QLG[4]
[26]    EXO∆REALCHLG←NQR CONTINUE1(3↓QCHLG),LGTRENDCH
[27]    EXO∆REALCHLG←EXO∆REALCHLG×0.4
[28]   ⍝ATTEMPT TO MODIFY GOVERNMENT DEMAND FOR LABOUR DUE TO
[29]   ⍝FICTIOUS LABOUR-FORCE IN THE MODEL...
[30]   ⍝(GOVERNMENT LABOUR+INDUSTRY LABOUR)÷(TOTAL LABOUR FORCE)=1.7÷4.1 MILLION PEOPLE
[31]   ⍝THAT IS: FICTIOUS LABOURFORCE=1.7 MILL. PEOPLE IS
[32]   ⍝APPROXIMATELY 0.4×TOTAL LABOUR FORCE.
[33]   ⍝THAT'S WHY DEMAND IS MULTIPLIED WITH 0.4...
[34]   ⍝FREDRIK B
[35]   ⍝
[36]    QWG←WAGES[1]+0.375×(WAGES[2]-WAGES[1])
[37]    WG←WAGES[1]
```

APPENDIX C FUNCTION PUBLICAΔSECTOR (cont.)



```
[38]  ⍝
[39]   QINVG←(0.25×INVG×1000000)×RATE1*(1.5÷4)
[40]  ⍝QUARTER1: RATE1*(-2.5÷4)
[41]  ⍝QUARTER2: RATE1*(-1.5÷4)
[42]  ⍝QUARTER3: RATE1*(0.5÷4)
[43]  ⍝QUARTER4: RATE1* 1.5÷4)
[44]  ⍝SUM = (APPROX.) 4  ,WHICH MEANS THAT SUM(QINVG)=INVG
[45]   EXOΔQDINVG←(NQRρ(RATE2*(1÷4)))-1
[46]   EXOΔRSUBS←NYR CONTINUE2 RSUBS
[47]   GKOFF←(10*6)×(10↑IO76[;1])÷(WG×LG)
[48]  ⍝
[49]   WGΔREF←WG
    ∇
```

APPENDIX C   FUNCTION MARKETS

```
    ∇  MARKETS;PDOM;M∆PRICE
[1]    ⍝FINAL OUTPUT FROM THIS FUNCTION:
[2]    ⍝XIN,IO,IO2,IO3,OMEGA,OMEGABLD,OMEGAIN,IMP,
[3]    ⍝QINVBLD,QINVIN,EXO∆QDINVIN,EXO∆QDINVBLD,
[4]    ⍝QPDOM,QDPDOM,EXO∆QDPIN,P∆REF,QPFOR,EXO∆QDPFOR
[5]    ⍝
[6]    ⍝OUTPUT TO FUNCTION HOUSEHOLDS∆DATA:
[7]    ⍝QDPIN,QDPFOR
[8]    ⍝
[9]    ⍝
[10]   ⍝
[11]   ⍝
[12]   ⍝
[13]    IMP←10⍴0
[14]    XIN←6⍴0
[15]    XIN[3]←0
[16]    XIN[1,2,4,5,6]←IO76[5,6,8,9,10;18]÷IO76[14;5,6,8,9,10]
[17]   ⍝XIN=EXPORT SHARES IN SECTORS OUTSIDE OUR 4 MARKETS
[18]    SWEDISH∆DEMAND←IO76[⍳10;21]-(IO76[⍳10;20]+IO76[⍳10;19]+IO76[⍳10;18])
[19]   ⍝SWEDISH∆DEMAND←PRODUCTION(INCL. IMPORTS)-(DIFF+IMPORTS+EXPORTS).
[20]   ⍝NOTE THAT IMPORTS IS STORED WITH NEGATIVE SIGN IN IO76...
[21]   ⍝
[22]    IMP←(|IO76[⍳10;19])÷SWEDISH∆DEMAND
[23]   ⍝IMP= IMPORT-SHARE OF_SWEDISH_CONSUMER'S DEMAND ...
[24]   ⍝ IMP=IMPORTS VECTOR FOR MARKETS 1,2..10
[25]   ⍝
[26]   ⍝
[27]   ⍝
[28]    IO←IOCOEFF76[⍳10;⍳10]
[29]    IO2←IOCOEFF76[⍳4;4+⍳6]
[30]    IO3←IOCOEFF76[4+⍳6;4+⍳6]
[31]    OMEGA←10↑IOCOEFF76[;16]
[32]    OMEGABLD←10↑IOCOEFF76[;14]
[33]    OMEGAIN←10↑IOCOEFF76[;15]
[34]   ⍝
[35]   ⍝
[36]   ⍝
[37]   ⍝
```

APPENDIX C   FUNCTION MARKETS (cont.)

```
[38]    INVBLD←IO76[14;14]
[39]    INVIN←IO76[14;15]
[40]    QINVBLD←(0.25×INVBLD×1000000)×BLD∆RATE1*(1.5÷4)
[41]    QINVIN←(0.25×INVIN×1000000)×IN∆RATE1*(1.5÷4)
[42]    EXO∆QDINVIN←¯1+(NQR⍴(IN∆RATE2*(1÷4)))
[43]    EXO∆QDINVBLD←¯1+(NQR⍴(BLD∆RATE2*(1÷4)))
[44]   ⍝
[45]   ⍝
[46]   ⍝
[47]   ⍝ HIST∆TXVA2[YEARS;QUARTERS] YEAR=1,2,3,4  YEAR 1=1974
[48]   ⍝ P[MARKETS;YEARS]YEAR=1,2,3,4
[49]    P←IMPL∆PRIS,[1]IMPL∆PRIS∆IN
[50]    PDOM←P DIV8 1-0.25×+/HIST∆TXVA2[⍳4;]
[51]    ENS P[;3]=100
[52]   ⍝QPFOR ESTIMATED FROM VARIABLE EXPORT∆PRIS IN
[53]   ⍝OLD INITIALIZATION (BEFORE JULY 1980)...
[54]    QPFOR← 101.4 100.8 102.1 101
[55]    QDPFOR←(TL∆EXP∆PRIS∆76 NYR)[;1]
[56]    EXO∆QDPFOR← 0 1 ↓TL∆EXP∆PRIS∆76 NYR
[57]   ⍝
[58]   ⍝THOMAS LINDBERG  HAS MADE THE FUNCTION TL∆EXP∆PRIS∆76
[59]   ⍝WHICH YIELDS QUARTERLY EXPORTPRICE-CHANGES...
[60]   ⍝
[61]   ⍝
[62]    QPDOM←PDOM[;3,4]+.× 0.625 0.375
[63]    QDPDOM←¯1+(PDOM[;4]÷PDOM[;3])*(1÷4)
[64]    QDPIN←¯1+((IMPL∆PRIS∆IN[;4]÷IMPL∆PRIS∆IN[;3])*(1÷4))+(HIST∆TXVA2[3;4]-HIST∆TXVA2[3;3])
[65]    M∆PRICE←(6,4×(⍴IMPL∆PRIS∆IN)[2])⍴0
[66]    J←1
[67]   ST:→(J=7)/SL
[68]    M∆PRICE[J;]←MAKEQUARTERS IMPL∆PRIS∆IN[J;]
[69]    J←J+1
[70]    →ST
[71]   SL:
[72]    M∆PRICE←(0,11)↓M∆PRICE
[73]    EXO∆QDPIN←NQR CONTINUE2((RELDIFF M∆PRICE),TRENDM)
[74]    P∆REF←PDOM[;3]
     ∇
```

APPENDIX C SUBFUNCTION MONETARY AND HOUSEHOLDS

```
      ∇MONETARY[⎕]∇
    ∇  MONETARY
[1]    ⍝ VARIABLES FROM WORKSPACE MACRO   WHICH WILL REMAIN
[2]    ⍝ UNCHANGED  AND WHICH WILL BECOME FINAL OUTPUT FROM
[3]    ⍝INITIALIZATION: RI,LIQB,POSG,LIQBFOR
[4]    ⍝ OTHER VARIABLES WHICH WILL
[5]    ⍝ BECOME FINAL OUTPUT FROM INITIALIZATION: ALL EXO-VARIABLES HERE
[6]    ⍝
[7]     EXO∆RI←NQR CONTINUE1 EXO∆RI
[8]     EXO∆RIBWFOR←NQR CONTINUE1 EXO∆RIBWFOR
[9]     EXO∆RIDEPFOR←NQR CONTINUE1 EXO∆RIDEPFOR
    ∇
```

```
    ∇  HOUSEHOLDS
[1]    ⍝OUTPUT FROM INITIALIZATION: SEE HOUSEHOLDS∆DATA INSTEAD
[2]    ⍝WHSUM AND HH76 WILL  BE USED IN HOUSEHOLDS∆DATA IN
[3]    ⍝THE SECOND PART OF INITIALIZATION...
[4]     HH76←IOCOEFF76[⍳10;12]
[5]     WHSUM←HUSHALLSDEP
    ∇
```

APPENDIX C FUNCTION ESTABLISHMENTS

```
    ∇  ESTABLISHMENTS;R;F;ALPHA;SCALE;RATIO;RATIO1;RATIO2;HELP;FLAG;DUMMY
        ∊')COPY SI76 X FΔDATA FIRMID   LIST RΔMARKET'
       ⍝FIRM-VARIABLES FROM WORKSPACE SI76.
       ⍝
       ⍝
       ⍝INPUT FROM FUNCTION MARKETS:IO (INPUT-OUTPUT-MATRIX)
       ⍝INPUT FROM ISTARTXX-FUNCTION: SYNTHΔFIRMS
       ⍝
       ⍝
       ⍝OUTPUT FROM THIS FUNCTION:
       ⍝MARKET,P,QP,DP,W,QW,DW,S,QS,DS,Q,QQ,DQ,
       ⍝L,EXPDP,EXPDS,EXPDW,HISTDP,HISTDS,HISTDW,
       ⍝HISTDPDEV2,HISTDWDEV2,HISTDSDEV2,MHIST,CHM
       ⍝VA,QIMQ,QVA,DVA,M,AMAN,STO,IMSTO,
       ⍝QTOP,TEC,QINV,QINVLAG,DELAYΔINV,K1,K1BOOK,K2,BW,
       ⍝QTDIV,RSUBSΔCASH,RSUBSΔEXTRA,RES,INVEFF,RESMAX,BETA,
       ⍝IMBETA,TMINV,BIG,SMALL,IMBIG,IMSMALL,FΔINKOP,BRINKOP,
       ⍝SHARE,X,ORIGMARKET,LEFT
       ⍝
       ⍝
       ⍝
       ⍝INFORMATION ABOUT INDATA:
       ⍝X IS FIRM-DATA.
       ⍝FΔDATA IS INDATA ABOUT FIRM-GROUPS.
       ⍝X IS A MATRIX WITH FIRST COMPONENT= FIRM
       ⍝AND SECOND COMPONENT= VARIABLE (SALES,LABOUR,ETC..).
       ⍝X CONSISTS MAINLY OF DATA FOR THE YEAR 1976.
       ⍝
       ⍝
       ⍝ REDUCTION ON LIST
       ⍝FIRMS WITH INCONSISTENT VARIABLES ARE OMITTED
       L0:F←FIRMID[(X[;1]∊LIST)/⍳ρX[;1]]
        NAMNΔMARKET←RΔMARKET[(X[;1]∊LIST)/⍳ρX[;1]]
        ALPHA←(+/X[(X[;1]∊LIST)/⍳ρX[;1]; 7 12])÷FΔDATA[F;15]
       ⍝ CHECK ON ALPHA
        →(0=ρFLAG←(1<ALPHA+.×F∘.=⍳⌈/F)/⍳⌈/F)/L2
        HELP←⍳0
       ⍝ OLD: L1:HELP←HELP,F⍳1↑FLAG
```

```
[38]  L1:HELP←HELP,ALPHA⍳⌊/ALPHA[((1↑FLAG)=F)/⍳⍴F]
[39]   →(0<⍴FLAG←1↓FLAG)/L1
[40]   'DROPPING ',(5 2 ⍕LIST[HELP]),' FROM LIST.'
[41]   LIST←(~(⍳⍴LIST)∊HELP)/LIST
[42]   →L0
[43]  L2:X←X[(X[;1]∊LIST)/⍳⍴X[;1];]
[44]  ⍝
[45]  ⍝
[46]  ⍝
[47]  ⍝ R=NUMBER OF REAL FIRMS.
[48]  ⍝MARKET=VECTOR WITH MARKET NUMBERS FOR EACH FIRM,
[49]  ⍝FOR EXAMPLE:  1 1 1 2 1 3 1 4 1 4 ...ETC.
[50]  ⍝S∆MARKET=VECTOR WITH MARKET-NUMBERS FOR SYNTHETIC FIRMS.
[51]  ⍝
[52]   S∆MARKET←SYNTH∆FIRMS DUP⍳4
[53]   MARKET←NAMN∆MARKET,S∆MARKET
[54]   R←1↑⍴X
[55]  ⍝
[56]   'SIZE-UTSKRIFT 2'
[57]   ⍎')SIZE'
[58]  ⍝
[59]  ⍝
[60]  ⍝
[61]  ⍝ SETTING SCALE FOR SYNTHETIC FIRMS:
[62]   SCALE←⍳0
[63]   SCALE←SCALE,SYNTH∆FIRMS[1]SCALE 0.02
[64]   SCALE←SCALE,SYNTH∆FIRMS[2]SCALE 0.001
[65]   SCALE←SCALE,SYNTH∆FIRMS[3]SCALE 0.02
[66]   SCALE←SCALE,SYNTH∆FIRMS[4]SCALE 0.0001
[67]   ENS 1=SYNTH∆SUM1 SCALE
[68]  ⍝
[69]   ⎕RL←123476
[70]  ⍝⎕RL YIELDS START-VALUE FOR PSEUDO-RANDOM-NUMBERS:
[71]  ⍝THIS MEANS THAT THE SAME 'RANDOM-NUMBERS' WILL BE
[72]  ⍝GENERATED IN DIFFERENT EXECUTIONS ,AS LONG AS ONE
[73]  ⍝DOESN'T CHANGE ⎕RL.
[74]  ⍝RANDOMNUMBERS OCCUR IN THE FUNCTIONS 'USING' AND 'RANDOMIZE'.
[75]  ⍝
[76]  ⍝
[77]  ⍝
```

```
[78]  ⍝
[79]  ⍝
[80]  ⍝SALES:
[81]  ⍝SUM1,REAL∆SUM1,SYNTH∆SUM1 ETC. SUM FIRMVARIABLES TO
[82]  ⍝MARKET-VARIABLES,A FIRM-VECTOR IS SUMMED UP TO A
[83]  ⍝MARKET-VECTOR OF LENGTH 4.
[84]   REAL∆SALES←(+/X̲[; 7 12]×1000000)
[85]   RES∆SALES←SALES̄76-REAL∆SUM1(REAL∆SALES)
[86]   SYNTH∆SALES←SCALE×RES∆SALES[S̲∆MARKET]
[87]   S←REAL∆SALES,SYNTH∆SALES
[88]  ⍝
[89]  ⍝
[90]  ⍝
[91]  ⍝LABOUR:
[92]   REAL∆LABOUR←X̲[;3]
[93]   RES∆LABOUR←(T̄IM÷HOURS∆PER∆YEAR)-REAL∆SUM1(REAL∆LABOUR)
[94]   SYNTH∆LABOUR←R↓S×RATIO←(REAL∆LABOUR÷REAL∆SALES)USING S
[95]  ⍝
[96]  ⍝FUNCTION 'USING' HAS THE FORM 'A USING B'
[97]  ⍝FUNCTION 'USING' DOES:
[98]  ⍝(1) EXTENDS VARIABLE A WITH RANDOMIZED VALUES FOR
[99]  ⍝    SYNTHETIC FIRMS.
[100] ⍝ (2)THE RANDOMIZED VALUES OF A COVARIES WITH B.
[101] ⍝    THE VARIABLES A AND B ARE FIRM-VECTORS...
[102] ⍝
[103]  SYNTH∆LABOUR←SYNTH∆LABOUR×(RES∆LABOUR÷(SYNTH∆SUM1 SYNTH∆LABOUR))[S̲∆MARKET]
[104]  L←REAL∆LABOUR,SYNTH∆LABOUR
[105] ⍝
[106] ⍝
[107] ⍝
[108] ⍝EXPORT FRACTIONS   (EXPORTS÷SALES)   :
[109] ⍝XM=  EXPORT-SHARE (MARKET-AVERAGE). FROM
[110] ⍝IO-MATRIX. XM IS A VECTOR OF LENGTH=4 .
[111] ⍝SALES IS APPROXIMATED WITH PRODUCTION.
[112]  XM←4⍴0
[113]  XM←IO76[⍳4;18]÷IO76[14;⍳4]
[114] ⍝XM←EXPORTS (MARKETS 1,2,3,4) ÷ PRODUCTION (MARKETS 1,2,3,4)
[115]  REAL∆RATIO←(X̲[;7]÷(+/X̲[; 7 12]))
[116]  SYNTH∆RATIO←R̄EAL∆RATIŌ RANDOMIZE S
[117]  RES∆EXPORT←(XM×(SUM1 S))-REAL∆SUM1(REAL∆RATIO×REAL∆SALES)
```

```
[118]  SYNTH∆RATIO←SYNTH∆RATIO×(RES∆EXPORT÷(SYNTH∆SUM1(SYNTH∆RATIO×SYNTH∆SALES)))[S̲∆MARKET]
[119]  X←REAL∆RATIO,SYNTH∆RATIO
[120]  'TEST PA EXPORTANDEL:X>0.95
[121]  (X<0)∨(X>0.95)
[122]  X←0⌈0.95⌊X
[123] ⍝
[124] ⍝
[125] ⍝
[126] ⍝
[127] ⍝PRICES
[128]  P←(⍴MARKET)⍴100
[129] ⍝
[130] ⍝
[131] ⍝INVENTORIES
[132] ⍝RATIO=ACTUAL STOCK-RATIO=STOCK÷SALES
[133]  RATIO←(X̲[;48]÷100)USING S
[134]  STO←(S÷P̄)×RATIO
[135] ⍝RATIO1=NORMAL  LEVEL OF STOCK-RATIO
[136]  RATIO1←(X̲[;50]÷100)USING RATIO⌈0.01
[137] ⍝ NOTE WE¯ARE SETTING BIG, SMALL, ETC FOR EACH FIRM
[138]  BIG←RATIO⌈(1+∆←0.5)×RATIO1
[139]  SMALL←RATIO⌊(1-∆)×RATIO1
[140]  BIG[HELP/⍳⍴BIG]←(HELP←(RATIO<(1-∆)×RATIO1))/(2×RATIO1)-RATIO
[141]  BIG←0⌈0.5⌊BIG
[142]  SMALL[HELP/⍳⍴BIG]←(HELP←(RATIO>(1+∆)×RATIO1))/(2×RATIO1)-RATIO
[143]  SMALL←0⌈SMALL
[144]  ∆K3∆FINISH←S×RATIO-RATIO1
[145] ⍝ THAT WAS PRODUCT INVENTORIES..NEXT IS INPUT GOODS INVENTORIES.
[146] ⍝
[147] ⍝INPUTRATIO=(PURCHASES OF RAW MATERIALS)÷SALES
[148]  INPUTRATIO←(X̲[;17]÷+/X̲[; 7 12])USING S
[149] ⍝
[150]  RATIO1←(X̲[;44]÷100)USING INPUTRATIO
[151] ⍝ RATIO1=ĀCTUAL STOCK-RATIO.
[152]  RATIO2←(X̲[;46]÷100)USING RATIO1⌈0.01
[153] ⍝RATIO2= N̄ORMAL STOCK LEVEL.
[154]  K3∆IMED←S×INPUTRATIO×RATIO1
[155]  ĪMBIG←RATIO1⌈(1+∆)×RATIO2
[156]  IMSMALL←RATIO1⌊(1-∆)×RATIO2
[157]  IMBIG[HELP/⍳⍴IMBIG]←(HELP←(RATIO1<(1-∆)×RATIO2))/(2×RATIO2)-RATIO1
```

APPENDIX C   FUNCTION ESTABLISHMENTS (cont.)

```
[158]  IMBIG←0⌈0.5⌊IMBIG
[159]  IMSMALL[HELP/⍳ρIMBIG]←(HELP←(RATIO1>(1+Δ)×RATIO2))/(2×RATIO2)-RATIO1
[160]  IMSMALL←0⌈IMSMALL
[161]  BETA←IMBETA←0.5
[162]  ΔK3ΔIMED←S×INPUTRATIO×RATIO1-RATIO2
[163] ⍝IMSTO  IS A FIRM×PRODUCT-MATRIX (=FIRM×10-MATRIX)
[164] ⍝MULT7 MULTIPLIES A MATRIX WITH A COLUMN-VECTOR.
[165] ⍝
[166] ⍝M  MULT7  V .M=MATRIX M(I,J)  V=VECTOR V(I)
[167] ⍝RESULT: A  MATRIX WITH ELEMENTS  M(I,J)×V(I)
[168] ⍝
[169] ⍝ NEXT: SPREAD K3ΔIMED ACROSS SECTORS USING IO-MATRIX
[170]  IMSTO←((((⍉IO)DIV7+/⍉IO)[MARKET;])MULT7 K̲3ΔIMED)÷100
[171] ⍝ NOTE: WE HAVE DIVIDED BY 100 ASSUMING BĀSE YEAR=START YEAR.
[172] ⍝IMSTO SHOULD BE IN FIXED PRICES,THUS DIVISION BY 100
[173] ⍝,WHICH IS THE PRICEINDEX FOR 1976
[174] ⍝ THE IDEA BEHIND THAT COMPUTATION WAS AS FOLLOWS:
[175] ⍝   (⍉IO)[1;] LOOKS LIKE A[1,1],.....,A[1,10], WHERE
[176] ⍝   A[1,J]=FRACTION OF GROSS PRODUCTION IN SECTOR 1 ACCTD FOR BY
[177] ⍝     INPUTS FROM SECTOR J.
[178] ⍝   THEN A[1,J]÷SUM ON J OF A[1,J]  =  FRACTION OF INPUT GOODS
[179] ⍝     COMING FROM SECTOR J
[180] ⍝
[181] ⍝
[182] ⍝
[183] ⍝
[184] ⍝ COMPUTATION OF INPUT GOODS PURCHASES
[185]  REALΔINP←X̲[;17]×1000000
[186]  QCURR←S+ΔK̄3ΔFINISH
[187] ⍝
[188] ⍝QCURR=PRODUCTION IN CURRENT PRICES:SALES+CH. IN STOCK
[189] ⍝HELP (BELOW) IS TOTAL INPUT CONSUMPTION BY THE
[190] ⍝SYNTHETIC FIRM UNITS PER SECTOR (1,2,3,4).
[191] ⍝
[192]  HELP←(+/(⍉IO)[⍳4;]MULT7 SUM1 QCURR)-(REALΔSUM1(REALΔINP-R↑ΔK3ΔIMED))
[193]  HELP←HELP+SYNTHΔSUM1(R↓ΔK3ΔIMED)
[194] ⍝ HELP=TOTAL INPUT GOODS PURCHASES BY THE SYNTHETIC UNITS (ρHELP=4)
[195] ⍝     IN EACH SECTOR
[196] ⍝ INP=INPUT GOOD PURCHASES FOR EACH PRODUCTION UNIT, SUMMED OVER SECTORS
[197] ⍝ ρINP = ρMARKETS
```

APPENDIX C   FUNCTION ESTABLISHMENTS (cont.)

```
[198]  INP←REAL∆INP,(R↓S×INPUTRATIO)×(HELP÷(SYNTH∆SUM1 R↓S×INPUTRATIO))[S̲∆MARKET]
[199] ⍝
[200] ⍝ QIMQ=INP SPREAD ACROSS THE 10 SECTORS. JUST LIKE IMSTO ABOVE.
[201]  QIMQ←((((⍉IO)DIV7+/⍉IO)[MARKET;])MULT7 INP)÷100
[202]  QIMQ←QIMQ÷4
[203] ⍝ SAME COMMENT AS APPLIES TO THE DEFLATION OF IMSTO
[204] ⍝ VALUE ADDED
[205]  VA←QCURR+∆K3∆IMED-INP
[206]  DISPOSE1∆FIRMS
[207] ⍝
[208] ⍝CONSUMPTION=INP-∆K3∆IMED=PURCHASES-CHANGE IN STOCK
[209] ⍝ VALUE ADDED=PRODUCTION-CONSUMPTION
[210] ⍝
[211]  RES∆FORVF←SYNTH∆SUM1(R↓VA)
[212]  FORVF←SUM1(VA)
[213]  REAL∆FORVF←R↑VA
[214]  SYNTH∆FORVF←R↓VA
[215] ⍝FORVF,REAL∆FORVF ETC. ARE USED  IN FUNCTION CONTROLS BELOW...
[216] ⍝
[217] ⍝
[218] ⍝
[219] ⍝
[220] ⍝
[221] ⍝ WAGES
[222]  REAL∆KR∆LON←X̲[;5]×1000000
[223]  REAL∆W←REAL∆KR∆LON÷(R↑L)·
[224]  SYNTH∆W←R↓S×(RATIO←(REAL∆KR∆LON÷REAL∆SALES)USING L)÷L
[225]  RES∆KR∆LON←LON-REAL∆SUM1(REAL∆W×(R↑L))
[226]  SYNTH∆W←SYNTH∆W×(RES∆KR∆LON÷(SYNTH∆SUM1(R↓L)×SYNTH∆W))[S̲∆MARKET]
[227]  W←REAL∆W,SYNTH∆W
[228]  SYNTH∆KR∆LON←SYNTH∆W×(R↓L)
[229]  DW←(¯1+(×/X̲[; 2 5])÷×/X̲[; 3 4])USING W
[230]  QDW←DW÷4
[231]  QW←((⍉((2,(ρW))ρ(W,W+DW)))+.×(0.625,0.375))
[232]  DVA←DS←(¯1+(+/X̲[; 7 12])÷+/X̲[; 6 11])USING DW
[233]  QS←((⍉((2,(ρS))ρ(S,S+DS)))+.×(0.625,0.375))÷4
[234]  QVA←VA×(1+DVA÷4)÷4
[235] ⍝
[236] ⍝
[237] ⍝
```

APPENDIX C FUNCTION ESTABLISHMENTS (cont.)

```
[238] ⍝
[239] ⍝ MARGINS
[240]  M←1-W×L÷VA
[241]  M75←1-(X̲[;4]÷+/X̲[; 6 11])×R↑S÷VA
[242] ⍝ M75=PRŌFIT MARḠIN 1975.
[243]  HELP←(R↑M)-M75
[244]  MHIST←0.5×(2×M)-CHM←HELP USING DS
[245] ⍝VARIABLES FOR FUNCTION CONTROL BELOW
[246] ⍝
[247]  OVERSKOTT←SUM1(M×VA)
[248]  SYNTH∆OVERSKOTT←R↓(M×VA)
[249]  REAL∆OVERSKOTT←R↑(M×VA)
[250]  DP←((R↑DS)-X̲[;26]÷100)USING DS
[251]  QP←((⍉((2,(ρ̄P))ρ(P,P+DP)))+.×(0.625,0.375))
[252] ⍝ QUANTITIES
[253]  Q←(S+∆K3∆FINISH)÷P
[254]  QQ←(QS+∆K3∆FINISH÷4)÷QP
[255]  DQ←DS-DP
[256] ⍝ SOME VARIABLES ADDED 27 OCT 1980...
[257]  F∆INKOP←(INP-∆K3∆IMED)÷(100×Q)
[258] ⍝PURCHASING-SHARE PER FIRM =F∆INKOP
[259]  BRINKOP←4↑(+/[1]IO)
[260] ⍝PURCHASING SHARE PER MARKET =BRINKOP
[261]  SHARE←F∆INKOP÷BRINKOP[MARKET]
[262] ⍝SHARE IS USED IN THE MODEL IN THIS WAY:
[263] ⍝SHARE×(MARKET AVERAGE INPUT SHARE)=
[264] ⍝THE INDIVIDUAL INPUT SHARE FOR EACH FIRM.
[265] ⍝MARKET AVERAGE INPUT SHARE=BRINKOP[1]..BRINKOP[4]
[266] ⍝
[267] ⍝
[268] ⍝
[269] ⍝
[270] ⍝
[271] ⍝ A21 AND A22
[272]  A22←(-/X̲[; 30 32]÷100)USING A21←(-/X̳[; 32 26]÷100)USING M
[273]  A21←0⌈0̄.5⌊A21
[274]  A22←0.025⌈0.5⌊A22
[275] ⍝ MUST ENSURE A22>0 SO TEC CAN BE COMPUTED..
[276] ⍝ AMAN--BASED ON APPROXIMATION GIVEN IN INDUSTRIKONJUNKTUREN PAPER
[277]  AMAN←⍉(3,ρL)ρ(L×A21÷1+A21)÷3
```

APPENDIX C FUNCTION ESTABLISHMENTS (cont.)

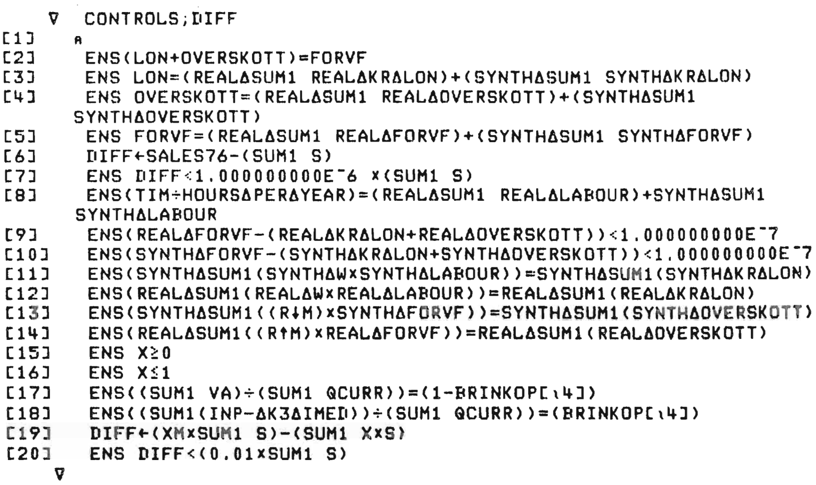
```
[278] ⍝ EXPECTATIONS...NOTE THAT EXPDW SHOULD BE FIXED
[279]  HISTDS←EXPDS←(¯1+(+/X[; 8 13])÷+/X[; 7 12])USING DS
[280]  HISTDSDEV2←(HISTDSDEV̄←-0.02 BETWE̅EN(⍴HISTDS)⍴0.02)*2
[281]  HISTDP←EXPDP←((R↑EXPDS)-X[;28]÷100)USING EXPDS
[282]  HISTDPDEV2←(HISTDPDEV←-0.̄02 BETWEEN(⍴HISTDP)⍴0.02)*2
[283]  HISTDW←EXPDW←EXPDS-EXPDP
[284]  HISTDWDEV2←(HISTDWDEV←-0.02 BETWEEN(⍴HISTDW)⍴0.02)*2
[285] ⍝ PRODUCTION FUNCTION PARAMETERS.
[286]  QTOP←(QQ×1+A21+A22)÷1-RES←(⍴QQ)⍴0.5×RESMAX←0.2
[287]  TEC←¯1×(⍟A22÷1+A21+A22)×QTOP÷L
[288]  ENS(QQ-QFR1 L)<0.5
[289] ⍝ FINANCIAL VARIABLES
[290]  K1BOOK←S×((÷/FΔDATA[F; 5 15])USING S)
[291]  K1←S×((÷/FΔDATA[F; 26 15])USING K1BOOK)
[292]  K2←K1BOOK×(((+/FΔDATA[F; 1 2 4 6])÷FΔDATA[F;5])USING K1)
[293]  A←K1+K2+K1BOOK×((÷/FΔDATA[F; 3 5])USING S)
[294]  BW←K1BOOK×(((+/FΔDATA[F; 8 9 10])÷FΔDATA[F;5])USING K1)
[295]  BAD←(⍴BW)⍴0
[296]  QTDIV←SUM2 ¯0.25×K1BOOK×((÷/FΔDATA[F; 20 5])USING M)
[297]  INVEFF←QTOP×QP÷K1
[298]  QINV←S×(((+/X̲[; 21 24])÷+/X̲[; 7 12])USING S)÷4
[299]  QINVLAG←QINV×̄1+(VA AVG1 DP¯DDIV 4)[DUR←3]
[300]  TMINV← 2 1 1 0.5
[301]  DELAYΔINV←⍉(3,⍴QINV)⍴QINV MULT1(4×TMINV)÷3
[302]  RSUBSΔCASH←RSUBSΔEXTRA←L×0
[303] ⍝
[304] ⍝
[305]  CONTROLS
[306] ⍝
[307] ⍝
[308] ⍝ CONSISTENCY-CONTROLS ARE MADE IN FUNCTION CONTROLS
[309] ⍝
[310]  IOΔMATRIX
[311] ⍝IO-MATRIX IN FLOWS  IS WRITTEN OUT
[312] ⍝
[313]  DISPOSE2ΔFIRMS
[314] ⍝THIS FUNCTION DELETES VARIABLES OF NO FURTHER USE
[315] ⍝
[316] ⍝ SOME VARIABLES NEEDED FOR NULLIFY AND SHRINK
[317]  LEFT←MARKET=ORIGMARKET←MARKET
[318]  'SIZEUTSKRIFT 3'
[319]  ⍎')SIZE'
[320] ⍝
    ∇
```

Note: Line 290,..302 FINANCIAL variables.
Function CONTROLS is listed on the following page.
Function IO-MATRIX is listed on the following pages.

APPENDIX C SUBFUNCTION CONTROLS

(subfunction to ESTABLISHMENTS)

Consistency Control



```
    ∇  CONTROLS;DIFF
[1]     ⍝
[2]     ENS(LON+OVERSKOTT)=FORVF
[3]     ENS LON=(REALΔSUM1 REALΔKRΔLON)+(SYNTHΔSUM1 SYNTHΔKRΔLON)
[4]     ENS OVERSKOTT=(REALΔSUM1 REALΔOVERSKOTT)+(SYNTHΔSUM1
       SYNTHΔOVERSKOTT)
[5]     ENS FORVF=(REALΔSUM1 REALΔFORVF)+(SYNTHΔSUM1 SYNTHΔFORVF)
[6]     DIFF←SALES76-(SUM1 S)
[7]     ENS DIFF<1.000000000E¯6 ×(SUM1 S)
[8]     ENS(TIM÷HOURSΔPERΔYEAR)=(REALΔSUM1 REALΔLABOUR)+SYNTHΔSUM1
       SYNTHΔLABOUR
[9]     ENS(REALΔFORVF-(REALΔKRΔLON+REALΔOVERSKOTT))<1.000000000E¯7
[10]    ENS(SYNTHΔFORVF-(SYNTHΔKRΔLON+SYNTHΔOVERSKOTT))<1.000000000E¯7
[11]    ENS(SYNTHΔSUM1(SYNTHΔW×SYNTHΔLABOUR))=SYNTHΔSUM1(SYNTHΔKRΔLON)
[12]    ENS(REALΔSUM1(REALΔW×REALΔLABOUR))=REALΔSUM1(REALΔKRΔLON)
[13]    ENS(SYNTHΔSUM1((R↓M)×SYNTHΔFORVF))=SYNTHΔSUM1(SYNTHΔOVERSKOTT)
[14]    ENS(REALΔSUM1((R↑M)×REALΔFORVF))=REALΔSUM1(REALΔOVERSKOTT)
[15]    ENS X≥0
[16]    ENS X≤1
[17]    ENS((SUM1 VA)÷(SUM1 QCURR))=(1-BRINKOP[⍳4])
[18]    ENS((SUM1(INP-ΔK3ΔIMED))÷(SUM1 QCURR))=(BRINKOP[⍳4])
[19]    DIFF←(XM×SUM1 S)-(SUM1 X×S)
[20]    ENS DIFF<(0.01×SUM1 S)
    ∇
```

<u>Note</u>: The subfunction ENS is documented in Appendix D.

APPENDIX C SUBFUNCTION IOΔMATRIX

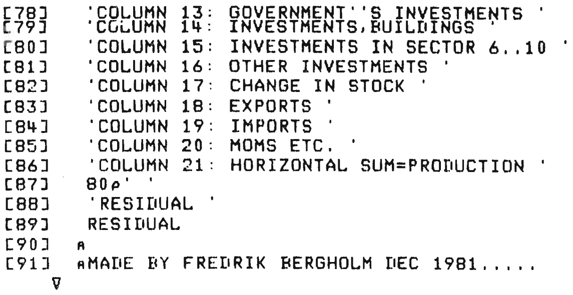
(subfunction to ESTABLISHMENTS)

(Consistency Control is performed)

```
    ∇ IO∆MATRIX;MA;PROD;CHAR;RESIDUAL;SWEDISH∆DEMAND
[1]    ⍝THIS FUNCTION DOES:
[2]    ⍝(1)    AN INPUT-OUTPUT MATRIX FOR THE SWEDISH
[3]    ⍝       ECONOMY IN FLOWS IS PRINTED OUT.
[4]    ⍝       THE INITIALIZED VARIABLES ARE USED.
[5]    ⍝(2)    VERTICAL SUM SHOULD BY DEFINITION BE
[6]    ⍝       EQUAL TO HORIZONTAL SUM.THE UNEXPLAINED
[7]    ⍝       RESIDUAL IS PRINTED OUT.
[8]    ⍝
[9]    ⍝
[10]   ⍝
[11]    'DO YOU WANT THE INPUT-OUTPUT-MATRIX PRINTED OUT?'
[12]    'YES OR NO :    '
[13]    CHAR←⎕
[14]    →(∧/(CHAR[1 2]='NO'))/0
[15]   ⍝
[16]    MA←IO76
[17]    PROD←SUM1(Q×100)
[18]    MA[;⍳4]←(IOCOEFF76[;⍳4],[1]1)MULT8(PROD÷10*6)
[19]   ⍝ THE FIRST 4 COLUMNS IN MA ARE REPLACED WITH FLOWS
[20]   ⍝ COMING FROM INITIALIZATION.
[21]   ⍝ COLUMN 5..10 UNCHANGED.
[22]    MA[⍳13;11]←(GKOFF×WG×LG÷10*6),(0,0,0)
[23]    MA[14;11]←+/[1]MA[⍳13;11]
[24]    MA[⍳13;12]←(HH76×4×QDI∆INIT2÷10*6),(0,0,0)
[25]   ⍝ QDI∆INIT2 YIELDS THE HOUSEHOLD'S DISPOSABLE INCOME
[26]    MA[14;12]←+/[1]MA[⍳13;12]
[27]    MA[;13]←(OMEGAG×QINVG×4÷10*6),(0,0,0,4×QINVG÷10*6)
[28]    MA[;14]←(OMEGABLD×QINVBLD×4÷10*6),(0,0,0,QINVBLD×4÷10*6)
[29]    MA[;15]←(OMEGAIN×QINVIN×4÷10*6),(0,0,0,4×QINVIN÷10*6)
[30]    MA[;16]←(OMEGA×(+/QINV)×4÷10*6),(0,0,0,4×(+/QINV)÷10*6)
[31]   ⍝
[32]    MA[⍳13;17]←(+/(∆K3∆IMED+∆K3∆FINISH)÷10*6)×IOCOEFF76[⍳13;17]
[33]    MA[14;17]←+/[1]MA[⍳13;17]
[34]   ⍝
[35]    MA[1 2 3 4 ;18]←(SUM1(X×S))÷10*6
[36]    MA[14;18]←+/[1]MA[⍳13;18]
[37]   ⍝
```

APPENDIX C SUBFUNCTION IOΔMATRIX (cont.)

```
[38]   SWEDISHΔDEMAND←+/MA[⍳10;⍳17]
[39]   MA[⍳13;19]←(IMP×SWEDISHΔDEMAND),(0,0,0)
[40]   MA[14;19]←+/[1]MA[⍳13;19]
[41]   MA[;19]←(¯1)×MA[;19]
[42]  ⍝
[43]   MA[⍳13;21]←MA[14;⍳10],(0,0,0)
[44]   MA[14;21]←+/[1]MA[⍳13;21]
[45]  ⍝
[46]   RESIDUAL←MA[;21]-((+/MA[⍳10;⍳20]),(0,0,0),(+/MA[14;⍳20]))
[47]  ⍝
[48]  ⍝*********O*********************
[49]  ⍝
[50]   ⎕PW←110
[51]  ⍝PAGE WIDTH
[52]   'INPUT-OUTPUT MATRIX FROM INITIALIZATION: '
[53]   80⍴' '
[54]   '        1         2          3         4          5          6         7        8         9          10'
[55]   80⍴' '
[56]   (8,0)⍕MA[;⍳10]
[57]   80⍴' '
[58]   '        11        12         13        14         15         16        17        18        19         20         21'
[59]   80⍴' '
[60]   (8,0)⍕MA[;10+⍳11]
[61]   'ROW 1: RAW MATERIAL SECTOR '
[62]   'ROW 2: INTERMEDIATE GOODS  '
[63]   'ROW 3: INVESTMENT GOODS AND CONSUMER DURABLE GOODS'
[64]   'ROW 4: CONSUMPTION GOODS '
[65]   'ROW 5: AGRICULTURE,FORESTRY,FISHING '
[66]   'ROW 6: MINING AND QUARRYING '
[67]   'ROW 7:   OIL  '
[68]   'ROW 8 : CONSTRUCTION  '
[69]   'ROW 9 : ELECTRICITY '
[70]   'ROW 10: OTHER SERVICIES '
[71]   'ROW 11: COMMODITY BASED INDIRECT TAXES '
[72]   'ROW 12: VALUE ADDED  IN PRODUCER''S PRICES '
[73]   'ROW 13: CORRECTIONS'
[74]   'ROW 14: SUM =PRODUCTION '
[75]   'COLUMN 1,2 THROUGH 10 = CORRESPONDING ROWS '
[76]   'COLUMN 11: GOVERNMENT''S CONSUMPTION '
[77]   'COLUMN 12: HOUSEHOLDS''S CONSUMPTION '
```



```
[78]   'COLUMN 13: GOVERNMENT''S INVESTMENTS '
[79]   'COLUMN 14: INVESTMENTS,BUILDINGS '
[80]   'COLUMN 15: INVESTMENTS IN SECTOR 6..10 '
[81]   'COLUMN 16: OTHER INVESTMENTS '
[82]   'COLUMN 17: CHANGE IN STOCK '
[83]   'COLUMN 18: EXPORTS '
[84]   'COLUMN 19: IMPORTS '
[85]   'COLUMN 20: MOMS ETC. '
[86]   'COLUMN 21: HORIZONTAL SUM=PRODUCTION '
[87]   80ρ' '
[88]   'RESIDUAL '
[89]   RESIDUAL
[90]  ⍝
[91]  ⍝MADE BY FREDRIK BERGHOLM DEC 1981.....
     ∇
```

APPENDIX C SUBFUNCTION MARKETS-DATA

```
    ∇  MARKETSΔDATA;TMEXP;TMTARG
[1]    ⍝
         output from initialization:All variables below except TMEXP,TMTARG,NPER
[2]     NPER←4
[3]     DUR←3
[4]     MKT←⍳4
[5]     IN←4+⍳6
[6]    ⍝

[7]     RET←¯1+1.035*(1÷4)
[8]     ENTRY←RET+0.0068÷NPER
[9]     EXPXDP←0.03
[10]    EXPXDW←0.07
[11]    EXPXDS←0.07
[12]    R←0.5
[13]    E1←0.1
[14]    E2←0
[15]    SMP←SMW←SMS←1-2÷1+TMEXP←3
[16]    FIP←FIW←FIS←(1-R)×2÷1+NPER×TMEXP
[17]    SMT←1-2÷1+TMTARG←3
[18]   ⍝

[19]    GAMMA←0.1
[20]    THETA←0.01
[21]    KSI←0.25
[22]    SKREPA←50
[23]    IOTA←0.5
[24]    NITER←9
[25]   ⍝

[26]    TMSTO←1
[27]   ⍝

[28]    TMIMSTO←1
[29]   ⍝

[30]    RHO←¯1+(1+1÷35)*(1÷4)
[31]    RHOBOOK←¯1+(1.15)*(1÷4)
[32]    QDMTEC←¯1+(1.056 1.03 1.026 1.004)*(1÷4)
[33]   ⍝ RESMAX←0.2 IS SET IN ESTABLISHMENTS...
[34]    LOSS←0.1
[35]    RESDOWN←0.9
[36]    WTIX←1
[37]   ⍝
```

```
[38]   RW←K2÷S
[39]   ALFABW←0.075÷NPER
[40]   BETABW←1
[41]   UTREF←0.85
[42]   ELINV←3
[43]   RTD←1
[44]  ⍝TMINV IS SET IN ESTABLISHMENTS
[45]  ⍝

[46]   EPS←0
[47]   TMX← 3 3 3 3
[48]   TMIMP← 3 3 3 3
[49]  ⍝

[50]   RLU←0.6
[51]   MAXDP←0.06
     ∇
```

APPENDIX C SUBFUNCTIONS PUBLIC-DATA AND SECONDARY-DATA

```
    ∇  PUBLIC∆DATA
[1]   ⍝ VARIABLES WHICH WILL BE OUTPUT FROM INITIALIZATION: WSG,RTRANS,T
      STOCURF,TSTOCURM
[2]    WSG←WG×LG
[3]    RTRANS←0.5
[4]   ⍝TSTOCURF IS A MARKET-VECTOR (4 MARKETS).FUNCTION SUM1 TRANSFORMS
      FIRMS-DATA TO MARKET-DATA...
[5]    TSTOCURF←SUM1(STO×QP)
[6]    TSTOCURM←QPDOM[⍳4]×(SUM1 STO)
    ∇
```

```
    ∇  SECONDARY∆DATA;MTEC∆PER∆FIRM
[1]   ⍝VARIABLES WHICH WILL BE OUTPUT FROM INITIALIZATION:
[2]   ⍝MTEC,LU,QDWIND
[3]   ⍝RU∆COPY IS A COPY OF RU WHICH COMES FROM INPUTFILE.
[4]   ⍝L,QW,QDW,QDMTEC,TEC COMES FROM ESTABLISHMENTS
[5]   ⍝GROWTH COMES FROM INPUTFILE (INIT∆GROWTH=GROWTH)
[6]   ⍝LG COMES FROM FUNCTION PUBLIC∆SECTOR
[7]    LU←(LG+SUM2(L))×RU∆COPY÷(1-RU∆COPY)
[8]   ⍝LG+SUM2 L=WORKING LABOUR FORCE=TOTAL LABOUR FORCE-UNEMPLOYED
[9]   ⍝UNEMPLOYED=R×'WORKING LABOUR FORCE'
[10]  ⍝WHERE R SHOULD BE UNEMPLOYED÷WORKING LABOUR FORCE
[11]  ⍝SINCE RU IS DEFINED AS UNEMPLOYED÷TOTAL LABOUR FORCE R=   RU÷(1-R
      U)...
[12]   QDWIND←¯1+(L AVG2 QW×(1+QDW))÷(L AVG2 QW)
[13]   MTEC∆PER∆FIRM←TEC DIV1(1-QDMTEC÷((RHO+GROWTH)*(1÷4)))
[14]   MTEC←L AVG1 MTEC∆PER∆FIRM
[15]  ⍝AVG1 YIELDS MARKET-AVERAGES FROM FIRMS-DATA (MTEC∆PER∆FIRM) WEIGH
      TED BY LABOUR-SHARES (L÷SUM L)
[16]   ENS 0<MTEC
    ∇
```

APPENDIX C SUBFUNCTION MONETARY-DATA

```
    ∇  MONETARYΔDATA
[1]   ⍝ALL VARIABLES BELOW WILL BE OUTPUT FROM INITIALIZATION
[2]    POSGFOR←0
[3]    TMFASS←3÷12
[4]    TMFD←2÷12
[5]    FD←FASS←(SUM2 X×S)×TMFASS
[6]    KAPPA1←0.02
[7]    KAPPA2←0.3
[8]    RFUND1←0.5
[9]    RFUND2←0.25
[10]   LAMDA1←0.6
[11]   LAMDA2←0.8
[12]   MAXQCHRI←0.01
[13]   MB←0.015
[14]   MAXRIDIFF←0.05
[15]   RIΔISΔEXOGENOUS←1
[16]   MINRI←MB
[17]   MAXRI←0.25
[18]   FUNDSΔAREΔENOUGH←0
[19]   REDCHBW←0.15
    ∇
```

```
     ∇  HOUSEHOLDSΔDATA;PRICECHANGES;DUR
[1]    ⍝INPUT TO THIS FUNCTION:
[2]    ⍝GKOFF,LG,WG,L,QW,QTDIV,LU,QDWIND FROM FUNCTION PUBLICΔSECTOR,ESTA
       BLISHMENT,SECONDARYΔDATA
[3]    ⍝QPDOM,QDPFOR,QDPIN FROM FUNCTION ESTABLISHMENTS
[4]    ⍝RTRANS,RLU,RHO  FROM FUNCTION MARKETSΔDATA
[5]    ⍝TXI1,TXW,TXWG,QINPAY,RI (INDIRECTLY) FROM WORKSPACE MACRO
[6]    ⍝HH76,WHSUM  FROM HOUSEHOLDS...
[7]    ⍝xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
[8]    ⍝OUTPUT FROM THIS FUNCTION,WHICH WILL BE FINAL OUTPUT FROM INITIAL
       IZATION:
[9]    ⍝Z,SAV,NDUR,NDURΔDUR,NH,WH,WHRA,QPH,QC,CVA,QDCPI,QCPI,QDI
[10]   ⍝QSAVHREQ,RHODUR,STODUR,ALFA AND BETA-COEFFICIENTS,SMOOTH  ,MARKET
       ΔITER...
[11]   ⍝
[12]    DUR←3
[13]    NDURΔDUR←⍳11
[14]    Z←11
[15]    SAV←12
[16]    NDUR←(DUR≠⍳11)/⍳11
[17]   ⍝NDUR,Z,SAV ARE INDEX-VARIABLES...
[18]    NH←LG+(SUM2 L)+LU
[19]    WH←WHSUM÷NH
[20]    QDIΔINIT
[21]   ⍝FUNCTION QDIΔINIT IS CALLED TO GIVE A VALUE TO QDI,AND THIS IS TH
       E ONLY PURPOSE OF THIS FUNCTION.QDI=DISPOSABLE INCOME
[22]    WHRA←WH÷QDI
[23]    QPH←QPDOM,0
[24]   ⍝QPH USED TO BE A VECTOR OF LENGTH 11.QPH(11) WAS THE PRICE IN THE
        SERVICE SECTOR.THERE IS NO LONGER AN ELEVENTH SERVICE- SECTOR,SO
       QPH=QPDOM.FOR TECHNICAL REASONS WE SEE TO THAT QPH
[25]   ⍝HAS THE LENGTH 11 DESPITE THIS,FOR THE TIME BEING,WHERE WE WILL H
       AVE A REDUNDANT 0 AT THE END...
[26]    QC←(HH76×QDI),0
[27]    QC←(1,⍴QC)⍴QC
[28]    QC←⍉QC
[29]   ⍝QC AND CVA MUST BE COLUMN-VECTORS FOR TECHNICAL REASONS...
[30]   ⍝SEE MOSES-FUNCTION CPI1...
[31]    CVA←QC DIV7 QPH
[32]    QCPI←CPI1(QPH)
[33]    PRICECHANGES←QDPFOR,QDPIN,0
[34]    QDCPI←(PRICECHANGES+.×,QC)÷(+/,QC)
```

Note: QDI-INIT is a subfunction listed later on.

APPENDIX C SUBFUNCTION HOUSEHOLDS-DATA (cont.)

```
[36]  ⍝
[37]   RHODUR←RHO
[38]   STODUR←QPH[DUR]×CVA[DUR;1]÷RHODUR
[39]  ⍝
[40]   ALFA3←0.3
[41]   ALFA4←0.5
[42]   BETA1← 1 1 0.7 0.75 0.9 1 1 0.9 1 0.75 1 0.5
[43]   BETA2← 0 0.02 0.1 0.22 0.01 0 0 0.08 0 0.36 0 0.21
[44]   BETA3←0×BETA2
[45]   SMOOTH←(11⍴0.9),1
[46]  ⍝
[47]   MARKET∆ITER←3
[48]  ⍝MARKET∆ITER TELLS HOW MANY ITERATIONS WILL BE DONE IN THE  MARKET
       PROCESS DURING SIMULATION...
[49]   NH←1⍴NH
     ∇
```

APPENDIX C **SUBFUNCTION DISPOSE1-FIRMS**

(deletes a number of variables)

This function is called in subfunction ESTABLISHMENTS.

```
     ∇DISPOSE1∆FIRMS[⎕]∇
   ∇  DISPOSE1∆FIRMS
[1]     →(TESTUTSKRIFT=0)/START
[2]     'REAL∆RATIO'
[3]     REAL∆RATIO
[4]     'SYNTH∆RATIO'
[5]     SYNTH∆RATIO
[6]     'INPUTRATIO'
[7]     INPUTRATIO
[8]     'REAL∆SALES'
[9]     REAL∆SALES
[10]    'SYNTH∆SALES'
[11]    SYNTH∆SALES
[12]    'SLUT PA TESTUTSKRIFT I DISPOSE1∆FIRMS
[13]   START:
[14]   ⍝
[15]    KILL 'SCALE MAKEQUARTERS'
[16]    KILL 'RAMARKET FIRMID RES∆LABOUR SYNTH∆SALES RES∆SALES RATIO1 RAT
       IO2 INPUTRATIO'
[17]    KILL 'REAL∆RATIO SYNTH∆RATIO RES∆EXPORT REAL∆INP LIST K3∆IMED '
[18]   ⍝THIS FUNCTION DELETES VARIABLES AND FUNCTIONS OF NO FURTHER USE.
   ∇
```

APPENDIX C SUBFUNCTION DISPOSE2-FIRMS

(deletes a number of variables)
This function is called in subfunction ESTABLISHMENTS.

```
      ∇DISPOSE2ΔFIRMS[⎕]∇
    ∇  DISPOSE2ΔFIRMS
[1]     →(TESTUTSKRIFT=0)/START
[2]     'S̲ΔMARKET'
[3]     S̲ΔMARKET
[4]     'A21'
[5]     A21
[6]     'A22'
[7]     A22
[8]     'INP'
[9]     INP
[10]    'QCURR'
[11]    QCURR
[12]    'M75'
[13]    M75
[14]    'ΔK3ΔIMED'
[15]    ΔK3ΔIMED
[16]    'ΔK3ΔFINISH'
[17]    ΔK3ΔFINISH
[18]    'REALΔFORVF'
[19]    REALΔFORVF
[20]    'SYNTHΔFORVF'
[21]    SYNTHΔFORVF
[22]    'FORVF'
[23]    FORVF
[24]    'REALΔLABOUR'
[25]    REALΔLABOUR
[26]    'SYNTHΔLABOUR'
[27]    SYNTHΔLABOUR
[28]    'REALΔW'
[29]    REALΔW
[30]    'SYNTHΔW'
[31]    SYNTHΔW
[32]    'REALΔOVERSKOTT'
[33]    REALΔOVERSKOTT
[34]    'SYNTHΔOVERSKOTT'
[35]    SYNTHΔOVERSKOTT
[36]    'OVERSKOTT'
[37]    OVERSKOTT
[38]    'REALΔKRΔLON'
[39]    REALΔKRΔLON
[40]    'SYNTHΔKRΔLON'
[41]    SYNTHΔKRΔLON
[42]    'LON'
[43]    LON
```

```
[45]  START:
[46]   KILL 'X̲ FΔDATA S̲ΔMARKET NAMNΔMARKET A21 A22 INP QCURR M75'
[47]   KILL 'Δ̄K3ΔIMED Δ̄K3ΔFINISH REALΔSALES REALΔFORVF SYNTHΔFORVF FORVF
       REALΔLABOUR SYNTHΔLABOUR '
[48]   KILL 'REALΔW SYNTHΔW REALΔOVERSKOTT SYNTHΔOVERSKOTT OVERSKOTT'
[49]   KILL 'REALΔKRΔLON SYNTHΔKRΔLON LON SCALE HELP'
[50]   KILL 'IOΔMATRIX CONTROLS REALΔSUM1 SYNTHΔSUM1 DISPOSE1ΔFIRMS RAND
      OMIZE USING QFR1 HISTORY BETWEEN'
[51]  ⍝
[52]  ⍝THIS FUNCTION DELETES FUNCTIONS AND VARIABLES OF NO FURTHER USE..
      '
    ∇
```

```
      ∇KILL[⎕]∇
    ∇  KILL NAMES;POS;DUMMY
[1]   L̲:→(0=⍴NAMES)/0
[2]    POS←NAMES⍳' '
[3]    DUMMY←⎕EX(POS-1)↑NAMES
[4]    NAMES←POS↓NAMES
[5]    →L̲
    ∇
```

This function is stored in workspace VLISTS.

APPENDIX C SUBFUNCTION DISPOSE-VAR-INPUT

```
      ∇DISPOSEΔVARΔINPUT[⎕]∇
    ∇  DISPOSEΔVARΔINPUT;COPΔRI;COPΔTXW;COPΔTXWG;COPΔRIDEPFOR;
      COPΔRIBWFOR;COPΔTXC;COPΔTXI1
[1]    ⍝THIS FUNCTION GETS RID OF INPUTVARIABLES FROM
[2]    ⍝FIRST PART OF INITIALIZATION
[3]    ⍝
[4]     COPΔRIDEPFOR←EXOΔRIDEPFOR
[5]     COPΔRIBWFOR←EXOΔRIBWFOR
[6]     COPΔRI←EXOΔRI
[7]     COPΔTXW←EXOΔTXW
[8]     COPΔTXWG←EXOΔTXWG
[9]     COPΔTXC←EXOΔTXC
[10]    COPΔTXI1←EXOΔTXI1
[11]   ⍝
[12]   ⍝MACROLIST CONTAINS VARIABLENAMES FOR INPUT-VARIABLES
[13]    KILL MACROLIST
[14]    EXOΔRIDEPFOR←COPΔRIDEPFOR
[15]    EXOΔRIBWFOR←COPΔRIBWFOR
[16]    EXOΔRI←COPΔRI
[17]    EXOΔTXW←COPΔTXW
[18]    EXOΔTXWG←COPΔTXWG
[19]    EXOΔTXC←COPΔTXC
[20]    EXOΔTXI1←COPΔTXI1
[21]   ⍝VARIABLES FROM WORKSPACE MACRO HAVE SOMETIMES THE SAME
[22]   ⍝ NAME AS AN OUTPUT-VARIABLE,SUCH VARIABLES MUST NOT
[23]   ⍝BE DELETED BY THE CALL ''KILL MACROLIST''
[24]   ⍝
    ∇
```

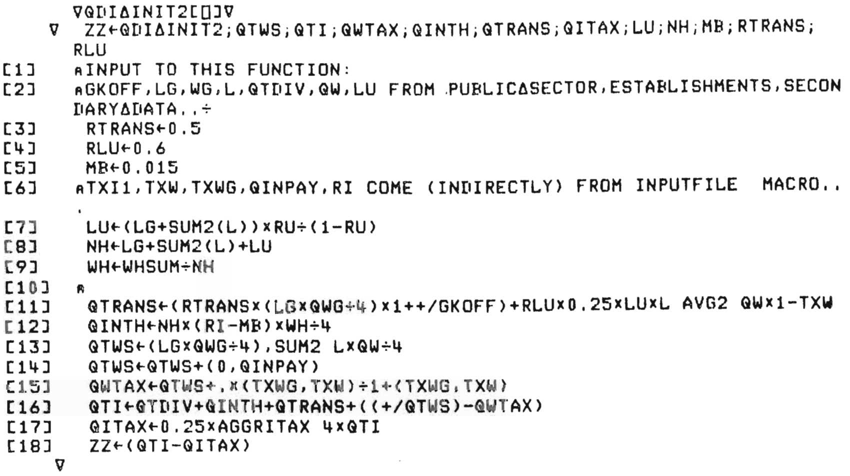
APPENDIX C **SUBFUNCTION QDI-INIT**

This function is called in subfunction HOUSEHOLDS DATA

```
      ∇QDIΔINIT[⎕]∇
    ∇  QDIΔINIT;QTWS;QTI;QWTAX;QINTH;QTRANS;QITAX;TXI1
[1]    ⍝INPUT TO THIS FUNCTION:
[2]    ⍝GKOFF,LG,WG,L,QTDIV,QW,LU FROM PUBLICΔSECTOR,ESTABLISHMENTS,SECON
       DARYΔDATA,,÷
[3]    ⍝RTRANS,RLU  FROM MARKETSΔDATA
[4]    ⍝TXI1,TXW,TXWG,QINPAY,RI COME (INDIRECTLY) FROM INPUTFILE  MACRO,,
       ,
[5]    ⍝LOCAL COPIES OF TXW,TXWG,,,ARE USED...
[6]    ⍝NH,WH  FROM HOUSEHOLDSΔDATA
[7]    ⍝
[8]     QTRANS←(RTRANS×(LG×QWG÷4)×1++/GKOFF)+RLU×0.25×LU×L AVG2 QW×1-
       TXWCOPY
[9]     QINTH←NH×(RIΔCOPY-MB)×WH÷4
[10]    QTWS←(LG×QWG÷4),SUM2 L×QW÷4
[11]    QTWS←QTWS+(0,QINPAYCOPY)
[12]    QWTAX←QTWS+.×(TXWGCOPY,TXWCOPY)÷1+(TXWGCOPY,TXWCOPY)
[13]    QTI←QTDIV+QINTH+QTRANS+((+/QTWS)-QWTAX)
[14]    TXI1←TXI1COPY
[15]    QITAX←0.25×AGGRITAX 4×QTI
[16]    QDI←(QTI-QITAX)÷NH
    ∇
```

APPENDIX C SUBFUNCTION QDI-INIT2

This function is called in subfunction IO-MATRIX.



```
      ∇QDIΔINIT2[⎕]∇
    ∇  ZZ←QDIΔINIT2;QTWS;QTI;QWTAX;QINTH;QTRANS;QITAX;LU;NH;MB;RTRANS;
      RLU
[1]   ⍝INPUT TO THIS FUNCTION:
[2]   ⍝GKOFF,LG,WG,L,QTDIV,QW,LU FROM PUBLICΔSECTOR,ESTABLISHMENTS,SECON
      DARYΔDATA..÷
[3]    RTRANS←0.5
[4]    RLU←0.6
[5]    MB←0.015
[6]   ⍝TXI1,TXW,TXWG,QINPAY,RI COME (INDIRECTLY) FROM INPUTFILE  MACRO..
      .
[7]    LU←(LG+SUM2(L))×RU÷(1-RU)
[8]    NH←LG+SUM2(L)+LU
[9]    WH←WHSUM÷NH
[10]  ⍝
[11]   QTRANS←(RTRANS×(LG×QWG÷4)×1++/GKOFF)+RLU×0.25×LU×L AVG2 QW×1-TXW
[12]   QINTH←NH×(RI-MB)×WH÷4
[13]   QTWS←(LG×QWG÷4),SUM2 L×QW÷4
[14]   QTWS←QTWS+(0,QINPAY)
[15]   QWTAX←QTWS+.×(TXWG,TXW)÷1+(TXWG,TXW)
[16]   QTI←QTDIV+QINTH+QTRANS+((+/QTWS)-QWTAX)
[17]   QITAX←0.25×AGGRITAX 4×QTI
[18]   ZZ←(QTI-QITAX)
     ∇
```

APPENDIX C SUBFUNCTION OUTPUT-OPERATIONS

```
      ∇OUTPUTΔOPERATIONS[⎕]∇
    ∇  OUTPUTΔOPERATIONS;LIST;TOTLIST
[1]    ⍝OUTPUT FROM INITIALIZATION IS BEING GROUPED:
[2]    ⍝VARIABELGRUPP1,VARIABELGRUPP2,..COME FROM WORKSPACE VLISTS,
[3]    ⍝AND ARE TEXT-VECTORS .THIS WORKSPACE ALSO CONTAINS SOME
[4]    ⍝ EXTRA VARIABLES AND FUNCTIONS...
[5]     ∊')WSID TEMPORARY'
[6]     ∊')SAVE'
[7]     LIST←⎕NL 2,3
[8]     LIST←,LIST
[9]     ∊')COPY VLISTS'
[10]    MN←WORKSPACENAME
[11]    KILL LIST
[12]    ⎕RL←123467
[13]    ⍳'∊'')COPY MACRO  ',GRUPP1,''''
[14]    TOTLIST←VARIABELGRUPP1,' ',VARIABELGRUPP2,' ',VARIABELGRUPP3
[15]    TOTLIST←TOTLIST,' ',VARIABELGRUPP4,' ',VARIABELGRUPP5
[16]    ∊')ERASE VARIABELGRUPP1 VARIABELGRUPP2 VARIABELGRUPP3'
[17]    ∊')ERASE VARIABELGRUPP4 VARIABELGRUPP5 GRUPP1 LIST'
[18]    ∊')ERASE DOKUMENTATION  '
[19]   ⍝
[20]   ⍝
[21]    MN COPYSAVE TOTLIST
[22]   ⍝OUTPUT FROM INITIALIZATION,AND NOTHING ELSE,IS SAVED
[23]   ⍝IN WORKSPACE(WHOSE NAME IS STORED IN WORKSPACENAME).
[24]   ⍝
[25]   ⍝
[26]    ∊')DROP TEMPORARY'
    ∇
```

```
    ∇  Y COPYSAVE X
[1]    ⍝THIS FUNCTION TAKES VARIABLES FROM WORKSPACE TEMPORARY
[2]    ⍝,TAKING ONLY THOSE SPECIFIED IN LIST X.AND SAVES THEM IN A WORKSP
       ACE  WITH NAME Y...
[3]     ⍳'∊'')COPY TEMPORARY ',X,''''
[4]     ⍳'∊'')WSID ',Y,''''
[5]     ∊')SAVE'
    ∇
```

This function is stored in workspace VLISTS.

APPENDIX D THE INITIALIZATION CODE, HELP-FUNCTIONS

The help-functions, in general, perform operations which occur many times during the initialization or which are so technical that they preferably should not be part of the main initialization code.

The help-functions are, in alphabetical order:

ABOVE, AVG1, AVG2, BETWEEN, CONTINUE1, CONTINUE2, CPI1, DDIV, DEV, DIFF, DIV1, DIV7, DIV8, DUP, ENS, EQUALS, HISTORY, MAKEQUARTERS, MODADD, MODDEL, MODSUBST, MULT1, MULT7, MULT8, PACK, QFR1, RANDOMIZE, REALΔSUM1, RELDIFF, SCANMAT, SUM1, SUM2, SYNTHΔSUM1, USING, SCALE

They are stored in workspace FUNCTI.

A short description of what some of the help-functions do:

AVG1:
Has 2 parameters W(=vector) and D(=vector).

Result: $\sum_{\mathbf{i\ in\ 1}} \left(\frac{W(i)\cdot D(i)}{\Sigma W(i)}\right), \sum_{\mathbf{i\ in\ 2}} \left(\frac{W(i)\cdot D(i)}{\Sigma W(i)}\right), \sum_{\mathbf{i\ in\ 3}} \left(\frac{W(i)\cdot D(i)}{\Sigma W(i)}\right), \sum_{\mathbf{i\ in\ 4}} \left(\frac{W(i)D(i)}{\Sigma W(i)}\right)$

i in 1 means: Summation over index **i** (usually number of firms) but only taking those **i** which belongs to sector 1, etc...

Thus we get a weighted average in each industrial sector (1,2,3,4) of a micro-variable. The result is a vector of length equal to 4.

SUM1:

Has 1 parameter V(=vector). (V=micro-variable).

Result: A vector of length=4 with the sum of V in one and each of the four industrial sectors (compare with **AVG1** above).

MODADD, MODDEL, MODSUBST:

These functions can change lines in another function, i.e. the programming code itself.* They are described in Part 1, Section 2.

MULT7:

Example:

$$\text{M } \mathbf{MULT7} \text{ V} = \begin{bmatrix} v_1 m_{11} & v_1 m_{12} \\ v_2 m_{21} & v_2 m_{22} \end{bmatrix}$$

$$\text{where M} = \begin{bmatrix} m_{11} m_{12} \\ m_{21} m_{22} \end{bmatrix} \text{ and V} = (v_1, v_2)$$

MULT7 is an operator which performs a kind of multiplication between a matrix and a vector.

* The possibility of changing lines in one program by aid of another program is a particular feature of the APL-language.

APPENDIX D

```
      ∇ABOVE[⎕]∇
    ∇  M←M1 ABOVE M2
[1]    ⍝ TO FORM A MATRIX WITH M1 ABOVE M2, PADDING WITH BLANKS OR ZEROES
        IF NEEDED.
[2]    ⍝ EACH OF M1 AND M2 IS MATRIX, VECTOR, OR SCALAR.

[3]     M←(((1↑⍴M1),1↓(⍴M1)⌈⍴M2)↑M1),[1]((1↑⍴M2),1↓(⍴M2)⌈⍴M1←(¯2↑ 1 1 ,⍴
       M1)⍴M1)↑M2←(¯2↑ 1 1 ,⍴M2)⍴M2
    ∇
```

```
      ∇AVG1[⎕]∇
    ∇  A←W AVG1 D
[1]    ⍝
[2]    ⍝ TO GET MARKET AVERAGES FROM FIRM DATA:
[3]    ⍝ 'D' IS THE FIRM (VECTOR) DATA TO BE AVERAGED.
[4]    ⍝ 'W' IS A WEIGHTING VECTOR.
[5]    ⍝ GLOBAL VECTOR 'MARKET' TELLS MARKET NUMBER OF EACH FIRM.
[6]    ⍝ GLOBAL 'NMARKETS' TELLS NUMBER OF MARKETS.
[7]    ⍝ 'A' IS THE (VECTOR) AVERAGE.
[8]    ⍝
[9]     A←((W×D)+.×MARKET∘.=⍳NMARKETS)÷(W+.×MARKET∘.=⍳NMARKETS)
    ∇
```

APPENDIX D

```
      ∇AVG2[⎕]∇
    ∇  A←W AVG2 D
[1]    ⍝
[2]    ⍝ TO GET A COUNTRY AVERAGE FROM FIRM DATA:
[3]    ⍝ 'D' IS THE FIRM (VECTOR) DATA TO BE AVERAGED.
[4]    ⍝ 'W' IS A WEIGHTING VECTOR.
[5]    ⍝ 'A' IS THE (SCALAR) AVERAGE.
[6]    ⍝
[7]     A←(+/W×D)÷(+/W)
    ∇
```

```
      ∇BETWEEN[⎕]∇
    ∇  R←A BETWEEN B
[1]     R←A+(B-A)×0.01×¯1+?101×B=B
    ∇
```

```
      ∇CONTINUE1[⎕]∇
    ∇  R←N CONTINUE1 V
[1]     R←N↑V,N⍴¯1↑V
    ∇
```

APPENDIX D

```
      ∇CONTINUE2[⎕]∇
    ∇  R←N CONTINUE2 M
[1]    R←((1↑⍴M),N)↑M,⍉(N,1↑⍴M)⍴M[;(⍴M)[2]]
    ∇
```

```
      ∇CPI1[⎕]∇
    ∇  Z←CPI1 PRICES
[1]   ⍝A÷B WHERE A=QC1×NH+QC2×NH... AND
[2]   ⍝B= QC1×NH÷P1 + QC2×NH÷P2 +...
[3]   ⍝
[4]    Z←(+/QC+.×NH)÷((QC+.×NH)+.÷PRICES)
    ∇
```

APPENDIX D

```
      ∇DDIV[⎕]∇
    ∇  Z←A DDIV B
[1]    ⍝
[2]    ⍝ TO 'DIVIDE' A TREND PERCENTAGE.
[3]    ⍝ 'Z' IS COMPUTED AS THE SOLUTION TO: (1+A)=(1+Z)*B
[4]    ⍝
[5]     Z←¯1+*(⍟1+A)÷B
    ∇
```

```
      ∇DEV[⎕]∇
    ∇  A←DEV X
[1]     A←X-+/X÷⍴X
    ∇
```

```
      ∇DIFF[⎕]∇
    ∇  R←DIFF F
[1]     R←((((¯1+⍴⍴F)⍴0),1)↓F)-((((¯1+⍴⍴F)⍴0),¯1)↓F)
    ∇
```

APPENDIX D

```
      ∇DIV1[⎕]∇
    ∇  Z←F DIV1 M
[1]    ⍝
[2]    ⍝ TO DIVIDE FIRMS' DATA WITH A MARKET VECTOR:
[3]    ⍝ 'F' IS THE FIRMS' DATA VECTOR.
[4]    ⍝ 'M' IS THE MARKET VECTOR.
[5]    ⍝ GLOBAL VECTOR 'MARKET' CONTAINS MARKET NUMBER OF EACH FIRM.
[6]    ⍝ 'Z' IS THE RESULTING (FIRM VECTOR) DATA.
[7]    ⍝
[8]     Z←F÷M[MARKET]
    ∇
```

```
      ∇DIV7[⎕]∇
    ∇  Z←M DIV7 V
[1]     ENS(⍴V)=(⍴M)[1]
[2]    ⍝
[3]    ⍝ TO DIVIDE A MATRIX WITH A VECTOR:
[4]    ⍝ EACH ELEMENT 'M[I;J]' IS DIVIDED BY 'V[I]'.
[5]    ⍝ THUS, 'M' MUST HAVE AS MANY ROWS AS 'V' HAS ELEMENTS.
[6]    ⍝
[7]     Z←M÷⍉(⌽⍴M)⍴V
    ∇
```

APPENDIX D

```
      ∇DIV8[⎕]∇
    ∇  Z←M DIV8 V
[1]     ENS(⍴V)=(⍴M)[2]
[2]    ⍝ TO DIVIDE A MATRIX WITH A VECTOR:
[3]    ⍝ EACH ELEMENT M[I;J] IS DIVIDED BY V[J].
[4]    ⍝ THUS, M MUST HAVE AS MANY COLUMNS AS V HAS ELEMENTS.
[5]     Z←M÷(⍴M)⍴V
    ∇
```

```
      ∇DUP[⎕]∇
    ∇  Z←NUM DUP EL
[1]    ⍝ Z←(NUM[1]⍴EL[1]),(NUM[2]⍴EL[2]), ... ,(NUM[N]⍴EL[N])
[2]     ENS(1≥⍴⍴NUM),(1≥⍴⍴EL)
[3]     ENS(1≤⍴,NUM),(2≤⍴,EL)
[4]     ENS(1=⍴,NUM)∨((⍴,NUM)=(⍴,EL))
[5]     NUM←(⍴EL)⍴NUM
[6]     Z←EL[(0≠Z)/Z←,⍉(((⌈/NUM),⍴NUM)⍴⍳⍴NUM)×(⍳⌈/NUM)∘.≤NUM]
    ∇
```

APPENDIX D

```
       ∇ENS[⎕]∇
     ∇  ENS STRING
[1]      →(∧/STRING=1)/0
[2]      'ERROR DETECTED BY FUNCTION ENS'
[3]      1÷0
[4]     ⍝LINE ABOVE STOPS EXECUTION
     ∇
```

```
       ∇EQUALS[⎕]∇
     ∇  Z←A EQUALS B
[1]      →((⍴⍴A)≠⍴⍴B)/Z←0
[2]      →((,⍴A)∨.≠,⍴B)/0
[3]      Z←(,A)∧.=,B
     ∇
```

```
       ∇HISTORY[⎕]∇
     ∇  R←SM HISTORY DATA;W
[1]      R←DATA+.×W÷+/W←⌽×\(¯1↑⍴DATA)⍴SM
     ∇
```

APPENDIX D

```
       ∇MAKEQUARTERS[⎕]∇
     ∇  W←MAKEQUARTERS V;FUNKA;FUNKB;DELTA;DIFF;F0;F1;F2;NIVA0;NIVA1;R;I;
       J;K;M;N;LEVEL;EXPR1;EXPR2;FUNKX;FIKTIV1;FIKTIV2
[1]    ⍝THIS FUNCTION DISTRIBUTES VARIABLES ON QUARTERS.FLOW-VARIABLES MU
       ST BE DIVIDED BY 4 AFTERWARDS...
[2]    ⍝V=INPUT=YEARLY FIGURES   W=RESULT=QUARTERLY FIGURES
[3]     W←(4×(⍴V))⍴0
[4]     FUNKB←'DELTA×X*((DELTA-N)÷N)'
[5]     FUNKA←'(((3×DELTA)-(6×N))×X*2)+((6×N)-(2×DELTA))×X'
[6]    ⍝
[7]     FIKTIV1←V[1]-(V[2]-V[1])
[8]     FIKTIV2←V[⍴V]+(V[⍴V]-V[¯1+⍴V])
[9]     V←FIKTIV1,V,FIKTIV2
[10]    M←(⍴V)-1
[11]    R←4⍴0
[12]   ⍝
[13]    I←1
[14]   START:→(I=M)/SLUT
[15]    F0←V[I]
[16]    F1←V[I+1]
[17]    F2←V[I+2]
[18]    K←4×(I-1)
[19]    NIVA0←F0+(F1-F0)÷2
[20]    NIVA1←F1+(F2-F1)÷2
[21]    DELTA←NIVA1-NIVA0
[22]    N←(F1-F0)÷2
[23]   ⍝
[24]    FUNKX←FUNKB
[25]    ⍎((×(F2-F1))≠×(F1-F0))/'FUNKX←FUNKA'
[26]   ⍝
[27]    J←1
[28]   S:→(J=5)/L
[29]    X←(J-1)÷4
[30]    LEVEL←⍎'F0+N+',FUNKX
[31]    EXPR1←⍎FUNKX
[32]    X←J÷4
[33]    EXPR2←⍎FUNKX
[34]    R[J]←LEVEL+(EXPR2-EXPR1)÷2
[35]    J←J+1
[36]    →S
[37]   L:
[38]    DIFF←F1-(+/R)÷4
[39]    →(TESTUTSKRIFT=0)/L3
[40]    'TESTUTSKRIFT'
```

APPENDIX D

```
[41]    R[1],R[2],R[3],R[4]
[42]    ⎕ ←'DIFF'
[43]    DIFF
[44]   L3:
[45]    W[K+⍳4]←R+DIFF
[46]    I←I+1
[47]    →START
[48]   SLUT:
[49]    →(TESTUTSKRIFT=0)/EXIT
[50]    ⎕ ←'RESULTAT'
[51]    I←0
[52]   S2:→(I=(M-1))/L2
[53]    ⎕ ←V[I+1]
[54]    ⎕ ←W[(⍳4)+I×4]
[55]    I←I+1
[56]    →S2
[57]   L2:'OK'
[58]   EXIT:
     ∇
```

APPENDIX D

```
      ∇MODADD[⎕]∇
    ∇  NAME MODADD OLDNEW;BREAK;CR;ROWS
[1]     ENS 'MOD'∨.≠3↑NAME←,NAME
[2]     ENS 3=⎕NC NAME
[3]     ENS(BREAK>1),1=⍴BREAK←('ω'=OLDNEW)/⍳⍴OLDNEW
[4]     ENS 1=⍴ROWS←(CR←⎕CR NAME)SCANMAT(BREAK-1)↑OLDNEW
[5]     ENS ⎕EX NAME
[6]     ENS(PACK NAME)EQUALS ⎕FX CR[⍳ROWS;]ABOVE(BREAK↓OLDNEW)ABOVE(ROWS,
      0)↓CR
    ∇
```

```
      ∇MODDEL[⎕]∇
    ∇  N←NAME MODDEL STRING;CR;ROWS
[1]     ENS~'MOD'∧.=3↑NAME←,NAME
[2]     ENS 3=⎕NC NAME
[3]     N←''⍴⍴ROWS←(CR←⎕CR NAME)SCANMAT STRING
[4]     ENS~1∊ROWS
[5]     ENS ⎕EX NAME
[6]     ENS NAME EQUALS ⎕FX(∧⌿ROWS∘.≠⍳1↑⍴CR)/[1]CR
    ∇
```

APPENDIX D

```
      ∇MODSUBST[⎕]∇
    ∇  NAME MODSUBST OLDNEW;BREAK;CR;ROWS
[1]    ENS 'MOD'∨.≠3↑NAME←,NAME
[2]    ENS 3=⎕NC NAME
[3]    ENS(BREAK>1),1=⍴BREAK←('ω'=OLDNEW)/⍳⍴OLDNEW
[4]    ENS 1=⍴ROWS←(CR←⎕CR NAME)SCANMAT(BREAK-1)↑OLDNEW
[5]    ENS ⎕EX NAME
[6]    ENS(PACK NAME)EQUALS ⎕FX CR[⍳ROWS-1;]ABOVE(BREAK↓OLDNEW)ABOVE(
      ROWS,0)↓CR
    ∇
```

```
      ∇SCANMAT[⎕]∇
    ∇  R←M SCANMAT S
[1]     R←(∨/∧⌿(⌽(⍴,S)≤⍳¯1↑⍴M)/(⍉((1↑⍴M),⍴,S)⍴¯1+⍳⍴,S)⌽(,S)∘.=M)/⍳1↑⍴M
    ∇
```

```
      ∇PACK[⎕]∇
    ∇  Z←PACK S
[1]     Z←1↓(Z∨1⌽Z←0,' '≠S)/' ',S
    ∇
```

APPENDIX D

```
      ∇MULT1[⎕]∇
    ∇  Z←F MULT1 M
[1]    ⍝
[2]    ⍝  TO MULTIPLY FIRMS' DATA WITH A MARKET VECTOR:
[3]    ⍝ 'F' IS THE FIRMS' DATA VECTOR.
[4]    ⍝ 'M' IS THE MARKET VECTOR.
[5]    ⍝ GLOBAL VECTOR 'MARKET' CONTAINS MARKET NUMBER OF EACH FIRM.
[6]    ⍝ 'Z' IS THE RESULTING (FIRM VECTOR) DATA.
[7]    ⍝
[8]     Z←F×M[MARKET]
    ∇
```

```
      ∇MULT7[⎕]∇
    ∇  Z←M MULT7 V
[1]     ENS((⍴V)=(⍴M)[1]),(2=⍴⍴M),(1=⍴⍴V)
[2]    ⍝
[3]    ⍝ TO MULTIPLY A MATRIX WITH A VECTOR:
[4]    ⍝ EACH ELEMENT 'M[I;J]' IS MULTIPLIED WITH 'V[I]'.
[5]    ⍝ THUS, 'M' MUST HAVE AS MANY ROWS AS 'V' HAS ELEMENTS.
[6]    ⍝
[7]     Z←M×⍉(⌽⍴M)⍴V
    ∇
```

```
      ∇MULT8[⎕]∇
    ∇  Z←M MULT8 V
[1]     ENS((⍴V)=(⍴M)[2]),(2=⍴⍴M),(1=⍴⍴V)
[2]    ⍝ TO MULTIPLY A MATRIX WITH A VECTOR:
[3]    ⍝ EACH ELEMENT 'M[I;J]' IS MULTIPLIED WITH 'V[J]'.
[4]    ⍝ THUS, 'M' MUST HAVE AS MANY COLUMNS AS 'V' HAS ELEMENTS.
[5]    ⍝
[6]     Z←M×(⍴M)⍴V
    ∇
```

APPENDIX D

```
      ∇QFR1[⎕]∇
    ∇  Q←QFRI L
[1]    Q←(1-RES)×QTOP×1-*-L×TEC÷QTOP
    ∇
```

```
      ∇RANDOMIZE[⎕]∇
    ∇  C←A RANDOMIZE B;D;E;AID
[1]    C←((REALΔSUM1 A)÷+/NAMNΔMARKET∘.=⍳4)[S̲ΔMARKET]
[2]   ⍝ EACH ELEMENT OF C EQUALS CORRESPONDING REAL MARKET  AVERAGE
[3]    →((0=B)∧1=⍴B)/END
[4]   ⍝ IF B=0, SKIP CORRELATION ASPECT
[5]    D←(⍴NAMNΔMARKET)↑B
[6]    E←(⍴D)↓B
[7]   ⍝ HELP VBLES: D=REAL PART OF B, E=SYNTHETIC PART OF B
[8]    AID←E-((E+.×S̲ΔMARKET∘.=⍳4)÷+/S̲ΔMARKET∘.=⍳4)[S̲ΔMARKET]
[9]   ⍝ AID=DEVIATION OF ELEMENTS OF E FROM THEIR MKT AVERAGES
[10]   C←C+AID×((+/(DEV D)×DEV A)÷+/(DEV E)*2)×(⍴E)÷⍴D
[11]  ⍝ THAT USED THE APPROXIMATION COV(C,E)=COV(A,D)
[12]  END:AID←A-((A+.×NAMNΔMARKET∘.=⍳4)÷+/NAMNΔMARKET∘.=⍳4)[NAMNΔMARKET]
[13]  ⍝ AID=DEVIATION OF ELEMENTS OF A FROM THEIR MKT AVERAGES
[14]   C←C+((¯50+(⍴C)?100)÷50)×(((REALΔSUM1 AID*2)÷+/NAMNΔMARKET∘.=⍳4)*
      0.5)[S̲ΔMARKET]
[15]  ⍝ C[I,J]=C[I]×(1+EPS[I,J])×SD(A[I])
[16]  ⍝ WHERE: C[I]=C FOR MARKET I AS COMPUTED ABOVE
[17]  ⍝        EPS[I,J] IS UNIFORM OVER [¯0.5, 0.5]
[18]  ⍝        SD(:)=STANDARD DEVIATION OF A ON THE ITH MARKET
    ∇
```

APPENDIX D

```
      ∇REALΔSUM1[⎕]∇
    ∇  A←REALΔSUM1 V
[1]    ⍝
[2]    ⍝ TO SUM FROM FIRMS TO MARKETS:
[3]    ⍝ 'V' IS THE FIRM DATA TO BE AGGREGATED, IF IT HAS MORE THAN
[4]    ⍝ ONE AXIS, FIRST DIMENSION MUST INDICATE FIRM NUMBER.
[5]    ⍝ GLOBAL VECTOR 'NAMNΔMARKET' TELLS MARKET NUMBER OF EACH FIRM.
[6]    ⍝ GLOBAL 'NMARKETS' TELLS NUMBER OF MARKETS.
[7]    ⍝ 'A' IS THE AGGREGATE.
[8]    ⍝
[9]     A←((⍳NMARKETS)∘.=NAMNΔMARKET)+.×V
    ∇
```

```
      ∇RELDIFF[⎕]∇
    ∇  R←RELDIFF F
[1]     R←(DIFF F)÷(((¯1+⍴⍴F)⍴0),¯1)↓F
    ∇
```

```
      ∇SUM1[⎕]∇
    ∇  A←SUM1 V
[1]    ⍝
[2]    ⍝ TO SUM FROM FIRMS TO MARKETS:
[3]    ⍝ 'V' IS THE FIRM DATA TO BE AGGREGATED, IF IT HAS MORE THAN
[4]    ⍝ ONE AXIS, FIRST DIMENSION MUST INDICATE FIRM NUMBER.
[5]    ⍝ GLOBAL VECTOR 'MARKET' TELLS MARKET NUMBER OF EACH FIRM.
[6]    ⍝ GLOBAL 'NMARKETS' TELLS NUMBER OF MARKETS.
[7]    ⍝ 'A' IS THE AGGREGATE.
[8]    ⍝
[9]     A←((⍳NMARKETS)∘.=MARKET)+.×V
    ∇
```

APPENDIX D

```
      ∇SUM2[⎕]∇
    ∇  A←SUM2 V
[1]    ⍝
[2]    ⍝ TO SUM FROM FIRMS TO A COUNTRY TOTAL:
[3]    ⍝ 'V' IS THE FIRM DATA TO BE AGGREGATED. IF IT HAS MORE THAN
[4]    ⍝ ONE AXIS, FIRST DIMENSION MUST INDICATE FIRM NUMBER.
[5]    ⍝ 'A' IS THE AGGREGATE.
[6]    ⍝
[7]     A←+⌿V
    ∇
```

```
      ∇SYNTHΔSUM1[⎕]∇
    ∇  R←SYNTHΔSUM1 V
[1]     R←((⍳NMARKETS)∘.=SΔMARKET)+.×V
    ∇
```

```
      ∇USING[⎕]∇
    ∇  OUT←REAL USING V
[1]     OUT←REAL,(REAL RANDOMIZE V)
    ∇
```

```
      ∇SCALE[⎕]∇
    ∇  S←N SCALE PAR
[1]     ENS(0<PAR),(1≤⍴PAR),(PAR≤S←1,¯1↓PAR)
[2]    ⍝ TO GET N SCALED NUMBERS IN DESCENDING ORDER.

[3]    ⍝ (¯1↓PAR) ARE SIZES OF NUMBERS 2,3,... RELATIVE TO FIRST NUMBER.

[4]    ⍝ AFTER THAT, MORE NUMBERS ARE GENERATED IN A LOGARITHMICALLY DECL
       INING FASHION DOWN TO (¯1↑PAR).
[5]    ⍝ NUMBERS ARE NORMALIZED TO HAVE SUM=1.

[6]     →(N=⍴S←(N⌊⍴S)↑S)/L
[7]     S←S,⌽(¯1↑PAR)×((÷/¯2↑1,PAR)*÷N-⍴S)*¯1+⍳N-⍴S
[8]    L:S←S÷+/S
```

APPENDIX D ENTRY VARIANT, ADDFIRM

```
        ∇ADD∆FIRM

        ∇ MMM ADDFIRM PARMS;MM;NEWSYMBOL;∆;RELSIZE;A2̲2̲P̲;⍙P̲;D̲P̲;W̲;M̲;D̲V̲A̲;⍙V̲
        A̲;V̲A̲;⍙̲;⍙̲⍙̲;L̲;R̲E̲S̲;⍙̲T̲O̲P̲
[1]     ⍝ TO INSERT NEW FIRM(S) INTO ONE MARKET; TO BE USED AT A YEAR
        LIMIT ONLY.
[2]     ENS(0 1∨.=ρρMMM),(1 2∨.=ρ,MMM),(1=ρρPARMS),(2≤ρPARMS)
[3]     NEWSYMBOL←(MMM←2ρMMM)[2]
[4]     MM←MMM[1]
[5]     ∆←PARMS[1]
[6]     RELSIZE←1↓PARMS
[7]     ⍝ MM IS MARKET NUMBER
[8]     ⍝ NEWSYMBOL GIVES NUMERICAL CODE FOR PLOTTING
[9]     ⍝ ∆ IS PROFIT-MARGIN ADVANTAGE COMPARED TO THE AVERAGE FIRM
[10]    ⍝ RELSIZE IS SIZE OF NEW FIRM(S) AS A FRACTION OF CURRENT MARK
        ET AGGREGATE
[11]    ENS 0=⎕NC 'NRS'
[12]    ⍝ THAT WAS TO ENSURE A YEAR LIMIT
[13]    ⍝
[14]    RW←RW,(ρRELSIZE)ρS AVG5 RW
[15]    ⍝
[16]    A21←VA AVG5 A̲21
[17]    A22←VA AVG5 A̲22
[18]    ⍝
[19]    INVEFF←INVEFF,(ρRELSIZE)ρK1 AVG5 INVEFF
[20]    K1←K1,RELSIZE×SUM5 K1
[21]    K1BOOK←K1BOOK,RELSIZE×SUM5 K1BOOK
[22]    K2←K2,RELSIZE×SUM5 K2
[23]    BW←BW,RELSIZE×SUM5 BW
[24]    ⍙INV←⍙INV,RELSIZE×SUM5 ⍙INV
[25]    ⍙INVLAG←⍙INVLAG,RELSIZE×SUM5 ⍙INVLAG
[26]    DELAY∆INV←DELAY∆INV,[1]RELSIZE∘.×SUM5 DELAY∆INV
[27]    ⍝
[28]    X←X,(ρRELSIZE)ρS AVG5 X
[29]    ⍝
[30]    P←P,P̲←(ρRELSIZE)ρS AVG5 P
[31]    ⍙P←⍙P,⍙P̲←(ρRELSIZE)ρRS AVG5 ⍙P
[32]    DP←DP,D̲P̲←(ρRELSIZE)ρS AVG5 DP
[33]    ⍝
[34]    W←W,W̲←(ρRELSIZE)ρL AVG5 W
[35]    DW←DW,(ρRELSIZE)ρVA AVG5 DW
[36]    ⍙DW←⍙DW,(ρRELSIZE)ρ(L×⍙W)AVG5 ⍙DW
[37]    ⍙W←⍙W,(ρRELSIZE)ρL AVG5 ⍙W
```

APPENDIX D ENTRY VARIANT, ADDFIRM (cont.)

```
[38]    ⍝
[39]    ⍎(0≠⎕NC 'CHM')/'CHM←CHM,(ρRELSIZE)ρS AVG5 CHM'
[40]    M←M,M̲←(ρRELSIZE)ρΔ+S AVG5 M
[41]    ⍝
[42]    DVA←DVA,D̲V̲A̲←(ρRELSIZE)ρVA AVG5 DVA
[43]    VA←VA,V̲A̲←RELSIZE×SUM5 VA
[44]    QVA←QVA,Q̲V̲A̲←RELSIZE×SUM5 QVA
[45]    ⍝
[46]    Q←Q,Q̲←V̲A̲÷P̲−((QPDOM×1−T×VA2)+.×IO)[MM]
[47]    QQ←QQ,Q̲Q̲←Q̲V̲A̲÷P̲−((QPDOM×1−T×VA2)+.×IO)[MM]
[48]    DQ←DQ,D̲V̲A̲−D̲P̲
[49]    ⍝
[50]    DS←DS,(ρRELSIZE)ρS AVG5 DS
[51]    S←S,Q̲×P̲
[52]    QS←QS,Q̲Q̲×Q̲P̲
[53]    ⍝
[54]    L←L,L̲←V̲A̲×(1−M̲)÷W̲
[55]    LU←LU−+/,L̲
[56]    ENS LU≥0
[57]    AMAN←((ρAMAN)+(ρRELSIZE),0)↑AMAN
[58]    ⍝
[59]    EXPDP←EXPDP,(ρRELSIZE)ρS AVG5 EXPDP
[60]    EXPDS←EXPDS,(ρRELSIZE)ρS AVG5 EXPDS
[61]    EXPDW←EXPDW,(ρRELSIZE)ρVA AVG5 EXPDW
[62]    HISTDP←HISTDP,(ρRELSIZE)ρS AVG5 HISTDP
[63]    HISTDPDEV←HISTDPDEV,(ρRELSIZE)ρS AVG5 HISTDPDEV
[64]    HISTDPDEV2←HISTDPDEV2,(ρRELSIZE)ρS AVG5 HISTDPDEV2
[65]    HISTDS←HISTDS,(ρRELSIZE)ρS AVG5 HISTDS
[66]    HISTDSDEV←HISTDSDEV,(ρRELSIZE)ρS AVG5 HISTDSDEV
[67]    HISTDSDEV2←HISTDSDEV2,(ρRELSIZE)ρS AVG5 HISTDSDEV2
[68]    HISTDW←HISTDW,(ρRELSIZE)ρVA AVG5 HISTDW
[69]    HISTDWDEV←HISTDWDEV,(ρRELSIZE)ρVA AVG5 HISTDWDEV
[70]    HISTDWDEV2←HISTDWDEV2,(ρRELSIZE)ρVA AVG5 HISTDWDEV2
[71]    MHIST←MHIST,(ρRELSIZE)ρΔ+S AVG5 MHIST
[72]    ⍝
[73]    STO←STO,RELSIZE×SUM5 STO
[74]    IMSTO←IMSTO,[1]RELSIZE∘.×SUM5 IMSTO
[75]    QIMQ←QIMQ,[1]RELSIZE∘.×SUM5 QIMQ
[76]    ⍝
[77]    RES←RES,R̲E̲S̲←(ρRELSIZE)ρVA AVG5 RES
[78]    QTOP←QTOP,Q̲T̲O̲P̲←Q̲Q̲÷1−A21+A22+R̲E̲S̲
[79]    TEC←TEC,(Q̲T̲O̲P̲÷L̲)×⍟(1−R̲E̲S̲)÷A22
[80]    ⍝
[81]    MARKET←MARKET,(ρRELSIZE)ρMM
[82]    ORIGMARKET←ORIGMARKET,(ρRELSIZE)ρMM
[83]    SYMBOL←SYMBOL,(ρRELSIZE)ρNEWSYMBOL
[84]    LEFT←LEFT,RELSIZE=RELSIZE
[85]    ⍝
[86]    RSUBSΔCASH←RSUBSΔCASH,0×RELSIZE
[87]    RSUBSΔEXTRA←RSUBSΔEXTRA,0×RELSIZE
        ∇

[88]
```

NOTES to Part 2

[1] A more detailed description of the micro (firm) database can be found in Albrecht-Lindberg (1982). In this working paper the micro initialization is also described.

[2] When starting the simulation 1976 only about 150 firms participate, whereas 250 firms participate when starting 1982.

[3] See Albrecht-Lindberg (1982), Ahlström (1978), Bergholm (1982), and Eliasson-Heiman-Olavi (1978).

[4] Comments about choosing another starting year are made in Section 7.

[5] For example:
EXO∆REALCHLG(1) = 3000 means that 3,000 persons will be added to the sector the first quarter 1977.

EXO∆REALCHLG(5) = 2500 means that 2,500 persons will be added to the fifth simulated quarter 1977.

EXO∆REALCHLG etc. is a vector with a number of components = the number of quarters to be simulated ...

[6] For the present SMT = 1/2 so Targ(n+1) is an ordinary average of Targ(n) and M(n). Targ is specified for the individual firm.

[7] The only thing the function **SI∆INIT** does is to fetch data from workspace MACRO and FUNCTI and to call the sub-functions on level 2 i Figure 3.

[8] Formally XX is a parameter to the main function **START.**

[9] The corresponding exogenous time-series are EXO∆QINVG, EXO∆QINVBLD etc.

[10] a) Statistical errors in SCB statistics.
b) IUI computation errors when distributing total industry sales on the 4 sectors in the model.

[11] In the sectors 5,6,...,10, where there are no firms in the model, IO is used to determine these sectors' demand for products. This is done in a conventional input-output fashion by inverting a sub-matrix of IO.

[12] Both the share **(total input)/(total production)** and the share **(input from sector j)/(total production)** will change at the macro level.

[13] Initialization version 19 and experiment version 11 were used. The experiment is labeled S11V19 (cf. Part 1 of this handbook).

[14] The sub-functions **RANDOMIZE** and **USING**. August 1980. See Appendix D. See also "The micro initialization of MOSES" by Albrecht-Lindberg (1982).

[15] The variable RI∆IS∆EXOGEOUS is a logical variable being zero or one.

[16] Remember that IO76 is in 1975 year's prices (see Section 3) whereas $IO76_{II}$ should be in 1976 year's prices.

[17] Profits = gross operating surplus.

[18] To simulate more than 30 years, one has to set the variable NYR equal to that number in a ISTARTXX function. For example NYR ← 50.

[19] This is behaviour of the government sector in the labour market in the present version of the model. One could think of other possibilities.

[20] If a parameter is a vector of length = the number of firms, one can change the behaviour of individual firms, otherwise it's much more tricky, i.e. one must make changes in the model to be able to do that.

[21] However, if one is sure that the new variable does not affect the initialization procedure and the consistency in any way, one can introduce the new variable directly in a MSTART function.

[22] i.e. vectors of length = number of firms in the simulation.

[23] MSTART10 is a function stored (since 1983) in the MSTART workspace.

LITERATURE to Part 2

"The APL Reference Manual": For example: APLSF programmer's reference manual, DEC system 10, DEC-10-LPLSA-A-D, 1976.

Albrecht-Lindberg, The Micro Initialization of MOSES, IUI, 1982, Working Paper No. 72.

Ahlström, L., 1978, "The Market Oriented Inter-Industry Stock and Flow Data Aggregation Scheme Used in the Swedish Model" in Eliasson (1978), op.cit.

Bergholm, F., 1984, "De utlandsetablerade företagens multiplikatoreffekter på svensk industri - några simuleringar" i De utlandsetablerade företagen och den svenska ekonomin, Eliasson (1984), IUI, Stockholm.

Eliasson, G, 1978, A Micro-to-Macro Model of the Swedish Economy, IUI Conference Reports 1978:1, Stockholm.

Eliasson, G, 1980, Experiments with Fiscal Policy Parameters on a Micro-to-Macro Model of the Swedish Economy, Reprint from Robert H. Haveman and Kevin Hollenbeck, eds., Microeconomic Simulation Models for Public Policy Analysis, vol. 2, pp.49-95.

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

Eliasson, G. with the assistance of Heiman, M., Olavi, G., 1976, A Micro Macro Interactive Simulation Model of the Swedish Economy, Preliminary Documentation, Economic Research Reports B15, Federation of Swedish Industries, Stockholm.

Eliasson, G., Heiman, M., Olavi, G., 1978, "Technical Specifications for the Swedish Micro Based Macro Model" in Eliasson (1978), op.cit.

Hansen, K., (1989) Chapter VI "Firm Entry in MOSES" in Albrecht, J.W. et al., MOSES Code, IUI (forthcoming).

SUPPLEMENT A MICRO-TO-MACRO DATA BASE

by Louise Ahlström

Even though the presence of advanced high-speed computers make the development of complex micro simulation models possible, the obstacle to rapid progress is not modeling per se or computor capacity but rather the need to obtain a consistent micro-macro data base and to incorporate the micro data into a macro model format.

Data base work, if not organized efficiently, easily overwhelms the project group. Certain approximations are always needed and according to our experience problems often arise when micro data sets are forced into consistency with official and often internally inconsistent macro data. It is necessary to make a series of decisions as to how to bypass these problems. The process of making these decisions is a painful one, especially since one does not know which problem will come next. Frequently, after having successfully dealt with one problem, it is necessary to give up the "solution", since the solution of the problem that one stumbles on thereafter is not in line with the solution chosen for the first. The construction of a data base can thus be described as a tedious process of two steps forward and one step back - sometimes one step forward and two steps back. By necessity there will be many versions of the model and the corresponding data bases before the model project is terminated.

The structure we have finally chosen for the construction of the data bases must be seen as one way among a theoretically vast number of ways to deal with an operation analysis problem. We dare not claim that we have managed to find the best one. Our endevours ought to be judged only in the light of the objectives for our particular project. We will now point to some aspects of a general character.

The objectives for the construction of the Swedish micro-to-macro model were

1. to formulate a micro explanation for inflation and

2. to study the relationships between inflation, profits, investment and growth.

The chosen problems relate to typical dynamic processes and place heavy emphasis on the market process and its importance for price and income determination and growth at the macro level. For this reason an aggregation scheme that centers on markets and the use of industrial products rather than on the ordinary classification according to the production technique and raw material base is necessary. The aggregation scheme includes four industrial production sectors:

Raw Material Processing Industries (RAW)
Intermediate Goods Industries (IMED)
Investment and Consumer Durable Goods Industries (DUR)
Non Durable Consumption Goods Industries (NDUR)

Each industrial sector holds a large number of individual firms which constitute the micro feature in MOSES. The market processes in the model operate both between and within the four sectors. The aggregation scheme has been designed on the same format as that of the Annual Planning Survey of the Federation of Swedish Industries. This means that the capacity utilization data of this survey can be directly incorporated into the micro data base. Regarding the macro data base, we have had to develop a market-oriented classification scheme of our own in order to adapt the national accounts macro statistics to our micro-based sector classification. Lack of some firm data necessitates the use of industrial macro data as substitutes. The input-output matrix is one example where such simplifications have been necessary.[1]

In the early stages of constructing the model it was built around a 1968 base year macro data base. It was our ambition to be able to start the model in any year from 1950 and onwards. Thus a great deal of effort was put into collecting time series for macro variables. Due to lack of relevant data it was difficult to obtain time series that were consistent over time as well as with each other. The calibration of the model was done by starting simulations in 1968 and running them for a 10-year period. We compared the behavior of key macro variables in the model with reality, adjusted the parameters according to the results and started the process over again.

[1] For a description of how macro data are combined with real firm data see Eliasson, G., A Micro Simulation Model of a National Economy, Chapter 3 on estimation methods, in A Micro-to-Macro Model of the Swedish Economy, IUI Conference Reports 1978:1. Also see Eliasson, G., The Firm and Financial Markets in the Swedish Micro-to-Macro Model - Theory, Model and Verification, IUI Stockholm 1985, Chapter VIII.

We have had to put in substantial effort to overcome inconsistencies in the data base that have crept in not only because of our new aggregation type but also because of inconsistencies between the various parts of the national accounts statistics themselves. We have found by experience that a consistent data base for the first period of a simulation is imperative for a proper tracking by the model of historic macro test data. During the next stage of model life we wanted to update the data base in order to be able to start simulations in 1976 - we therefore decided to concentrate our efforts on obtaining a good base year. For this reason it was necessary to create an accounting system as a framework for the construction of the macro data base.

As the core around which we chose to build the accounting system (see Tables 1:1 through 1:3) we used the input-output system. The input-output coefficient matrix for 1976 used in the model is calculated from an input-output matrix expressed in producer's prices. Since the final demand, on the other hand, is expressed in purchaser's prices we have had to adjust the accounting system for the difference in price levels (DIF, Column 20). On the macro level the difference (DIF) is equal to the total of the commodity-based indirect taxes on final demand. The production value in market prices is identical to the production value in purchaser's prices on the macro level, since the transport and trade margins are included in the production value of the trade sector. For the individual sectors this is not true, which in turn depends on the existence of margins as well as on the mix-up of industrial sectors in the basic statistical material. The residual (RES, row 13) arises from differences in the gross production data if calculated from the demand side or from the production side. Since there is reason to believe that the data on the demand side have greater reliability, we have chosen to treat the residual as a row vector.

The input-output system has been created through the aggregation of input-output matrices expressed in producer's prices, provided by the central Bureau of Statistics. Since data in producer's prices were only available for 1975, we had to project the 1976 I/O matrix by assuming the same change between 1976 and 1975 in the coefficients expressed in producer's prices as in those in purchaser's prices. The input-output matrices shown in Section 2 in Part 2 of this handbook were constructed in this way. The aggregation scheme consisting of a weighting matrix based on value added is documented in Table 2.

Technically speaking the input-output coefficients are kept constant over time in the model. The model is not solved by inverting the input-output matrix in the traditional way. For the four industrial production sectors, the production volume is determined in the business system block, while the corresponding input-output coefficients determine the amount of inputs needed to make this level of production possible. At both ends of these sectors (that is, at both ends of each individual firm) there are buffer stocks to even out production flows. For the remaining six "external sectors" on the other hand, the input-output matrix is operating as in a conventional macro input-output model complemented with a Keynesian demand system. Since individual firms within and between markets meet with success and failure differently, they grow at very different rates. Consequently, the macro input-output coefficients in the model vary endogenously over time.

Table 1:1

The Accounting System in MOSES

Gross production in producers' prices	Value added	Commodity based indirect taxes, net	Raw Material Processing Industries	RAW	INPUTS into the production system
			Intermediate Goods Industries	IMED	
			Investment and Consumer Durable Goods Industries	DUR	
			Non Durable Consumption Goods Industries	NDUR	
			Agriculture, Forestry and Fishing	A/F/F	
			Mining and Quarrying	ORE	
			Petroleum Products Imports	OIL	
			Construction	CONSTR	
			Electricity	EL	
			Other Services	SERVICE	
			Government consumption	GOVT	FINAL DEMAND
			Private consumption	CONS	
			Investments	INV	
			Change in stocks	ΔSTO	
			Exports	EXP	
			Imports and duties	IMP	

Table 1:2

The Accounting System in MOSES

Vertically:	Row	
Inputs into sectors 1-10	1-10	
Total inputs		INPUTS
Commodity based indirect taxes, net	11	TAX
Value added in producers' prices	12	VA
Residual	13	RES
Gross production in producers' prices	14	TOTAL

Horizontally:	Column	
Input deliveries into sectors 1-10	1-10	
Total input deliveries		INPUTS
Government consumption	11	GOVT
Private consumption	12	CONS
Investments	13-16	INV
Change in stocks	17	ΔSTO
Exports	18	EXP
Total use		USE
Imports and duties	19	IMP
Gross production	(USE-IMP)	TOT
Difference	20	DIF
Gross production in producers' prices	21 (TOT-DIF)	TOTAL

Sectors 1-10: *		
Agriculture, Forestry and Fishing	1	A/F/F
Mining and Quarrying	2	ORE
Petroleum Products Imports	3	OIL
Raw Material Processing Industries	4	RAW
Intermediate Goods Industries	5	IMED
Investment and Consumer Durable Goods Industries	6	DUR
Construction	7	CONSTR
Non Durable Consumption Goods Industries	8	NDUR
Electricity	9	EL
Other Services	10	SERVICE

* The column and row numbers for sectors 1-10 have been altered in the present (March 1983) data base.

Table 1:3

The Accounting System in MOSES - A Schematic Description

FINAL DEMAND IN PURCHASER'S PRICES

GOVT + CONS + INV + ΔSTO + EXP - IMP

INPUTS	TAX	VA			DIF	MARG
in producers prices, incl. imports and duties	commodity based indirect taxes, net	wages	operating surplus incl. depreciation	non commodity taxes, net	commodity based indirect taxes on final demand, net	transport and trade margins

GROSS PRODUCTION IN PRODUCER'S PRICES

INPUTS in purchaser's prices   VALUE ADDED in producer's prices

GROSS PRODUCTION IN PURCHASER'S PRICES

Comment: On the macro level the production value in market prices is identical to the production value in purchaser's prices, since transport and trade margins are included in the production value of the trade sector (part of SERVICE).

Table 2 **The MOSES Aggregation Scheme According to the Standard for Swedish Classification of Economic Activities (SNI)** with Comparisons to the Industry Group in the Input-Output Statistics (I/O), the National Accounts Statistics (SNR) and the Classification Used in the IUI Long Term Survey (LB)

				1	2	3	4	5	6	7	8	9	10	
I/O	Sector	SNR	LB	A/F/F	ORE	OIL	RAW	IMED	DUR	CONSTR	NDUR	EL	SERVICE	SNI
1	Agriculture	1100	1	11										11
2	Forestry	1200	1	12										12
3	Fishing	1300	1	13										13
4	Mining and quarrying	2000	3		20 excl. (220)	(220)[a]								20
5	Sheltered food manufacturing	3111	4								3111/2 3116-8			3111/2 3116-8
6	Import - competing food manufacturing	3112	5								3113-5 3119-22			3113-5 3119-22
7	Beverage and tobacco manufacturing	3120	6								313/4			313/4
8	Textile and leather industries	3200	7				-	321·0.25	321·0.25	-	321·0.5 322-324			32
9	Manufacture of wood and wood products	3410	8				33111	3312/9	3320·0.4	33112/9	3320·0.6			33

Table 2 (cont)

				1	2	3	4	5	6	7	8	9	10	
I/O	Sector	SNR	LB	A/F/F	ORE	OIL	RAW	IMED	DUR	CONSTR	NDUR	EL	SERVICE	SNI
10a	Manufacture of pulp	3420 part	8				34111	34112	-	34113	-			34111-3
10b	Manufacture of paper products	3420 part	8				-	3412	-	-	3419			3412/9
11	Printing and publishing	3430	9				-	-	-	-	3420			342
12	Manufacture of chemicals and chemical products	3520 part	11				-	351 3521/9	-	-	3522/3			351/2
13	Petroleum refineries, manufacture of products of petroleum and coal	3530	12			(353/4)[a]	353/4 excl. (353/4)	-	-	-	-			353/4
14	Manufacture of rubber products	3510	10				-	355·0.8	-	-	355·0.2			355
15	Manufacture of plastic-products	3520 part	11				-	35601	-	-	35609			356
16	Manufacture of non-metallic mineral products	3600	13				-	36202	-	36201/9 369	361 36203			36

Table 2 (cont)

				1	2	3	4	5	6	7	8	9	10	
I/O	Sector	SNR	LB	A/F/F	ORE	OIL	RAW	IMED	DUR	CONSTR	NDUR	EL	SERVICE	SNI
17	Iron-, steel- and ferro-alloys industries	3700 part	14				37101/2	37103	-	-	-			371
18	Non-ferrous metal industries	3700 part	14				37201-3	37204	-	-	-			371
19	Manufacture of fabricated metal products, machinery and equipment	3810	15				-	3811 38199 38191/2	3812,382 3842-9 385	3813 38193/4	38195			381/2, 385 3842-9
20	Manufacture of electrical machinery, apparatus appliances and supplies	3830	15				-	3839	3831-3	-	-			383
								3521/9	-	-	3522/3	351/2		
21	Shipbuilding and repairing	3843	16						3841					3841
22	Manufacturing industries not elsewhere classified	3900	17								39			39
23	Repair of household applicances and motor vehicles etc.	3600 3842	13				36202		36201/9-		361			36
		9511/3	15										951	951

Table 2 (cont)

				1	2	3	4	5	6	7	8	9	10	
I/O	Sector	SNR	LB	A/F/F	ORE	OIL	RAW	IMED	DUR	CONSTR	NDUR	EL	SERVICE	SNI
24	Electricity, gas and water	4000	18									40		40
25	Construction	5000	19							50				50
26	Wholesale and retail trade	6100	20										61/2	61/2
27	Restaurants and hotels	6300	23										63	63
28	Transport and storage	7100	21										71	71
29	Communication	7200	21										72	72
30	Financial institutions and insurance	8100	23										81/2	81/2
31	Letting of dwellings and use of owner-occupied dwellings	8300	22							83101/3				83101/3
32	Letting of other premises	8400	23							83102				83102
33	Business services	8500	23										832/3	832/3
34	Private services not elsewhere classified	9600 excl. 9511/3	23										92-4 952-9	92-4 952-9

[a] The SNI code within parentheses refers to imports.

LITERATURE to the Supplement

Ahlström, L., 1978, The Market Oriented Inter-Industry Stock and Flow Aggregation Scheme Used in the Swedish Model in A Micro-to-Macro Model of the Swedish Economy. IUI Conference Reports 1978:1, Stockholm.

Ahlström, L. 1980, Input-output systemet och dess utveckling in Kalkyler för 80-talet. Specialstudier för IUIs långtidsbedömning 1979. Del 2.

Dorfman, R., Samuelson, P. and Solow, R., 1958, Linear Programming and Economic Analysis, New York.

Höglind, B. and Werin, L., 1964, The Production System of the Swedish Economy, An Input-Output Study, IUI, Stockholm.

Statistical Report N 1972:44, Input-Output Tables for Sweden 1968. National Central Bureau of Statistics.

Statistical Report N 1980:3. Input-Output Tables for Sweden 1975. National Central Bureau of Statistics.