

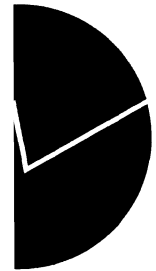
Statistics Norway
Research Department

Robin Choudhury

Documents

The Century Model

Technical Documentation of Computer
Programs and Procedures



Robin Choudhury

The Century Model

Technical Documentation of Computer Programs and Procedures

Abstract:

This document gives a detailed description of the computer programs and procedures developed to operate the Century Model, a long term macroeconomic model for Saudi Arabia, see Documents 96/7. The model has been developed by the Research Department of Statistics Norway for the Ministry of Planning in Saudi Arabia, as part of a United Nations Department for Development Support and Management Service (UN/DDSMS) project. The assignment has been an integral part of the UN/DDSMS Project SAU/94/001/A/01/01 Support for Economic Planning, Modelling and Management Information Systems Development under the contract TCD CON 4/95 with Statistics Norway. The motivation for the Century Model is to study the long run sustainability of the Saudi Arabian economy, which in stock terms, and due to the strong dependence upon exhaustible petroleum resources, can be thought of as managing national wealth. For the important issue of the overall sustainability of the Saudi economy, a longer time horizon is needed than the five-year development plans. Portable TROLL, a PC-based modelling and simulation system developed by Intex Solutions, Inc., has been used for the development and operating of the model. This document describes the organization of directories and the file structure of the computer programs used to operate the model, including a complete presentation of the programs controlling input and output. A description of the routines to generate the data base as well as a presentation of the model itself in full detail is also included. Finally, the document presents the programs and procedures made to organize simulations.

Keywords: Saudi Arabia, Sustainability.

JEL classification: 053, 011, Q32.

Address: Robin Choudhury, Statistics Norway, Research Department,
P.O.Box 8131 Dep., N-0033 Oslo, Norway. E-mail: rch@ssb.no.

Contents

1. Introduction	3
2. Directories and file structure	3
2.1. Directories.....	4
2.2. Files developed for the Century Model.....	5
2.2.1. Commands.....	5
2.2.2. Data.....	6
2.2.3. Programs.....	6
2.2.4. Models.....	8
2.2.5. Simres.....	8
3. Organization of input and output	9
3.1. Introduction.....	9
3.2. Access and search.....	9
3.2.1. Access.....	9
3.2.2. Search.....	10
4. Historical data	11
4.1. Introduction.....	11
4.2. Cmdata.inp.....	11
4.3. Transdat.inp.....	13
5. The model	15
5.1. Introduction.....	15
5.2. Cm.inp.....	15
6. Simulating the model	17
6.1. Introduction.....	17
6.2. Extrapolating the variables.....	17
6.2.1. Forecast.src.....	17
6.2.2. Gorecast.src.....	19
6.2.3. Extrap.src.....	20
6.2.4. Foreecon.src.....	21
6.2.5. Alt1ref1.src.....	22
6.2.6. Deviat.src.....	22
6.2.7. Changexa.src.....	23
6.2.8. Changexp.src.....	24
6.3. Simulating the variables.....	25
6.3.1. Refsim.src.....	25
6.3.2. Altsim.src.....	26
6.3.3. Sim.src.....	27
6.4. Printing results to screen.....	29
6.4.1. Prtref.src.....	29
6.5. Miscellaneous programs.....	30
6.5.1. Egeval.src.....	30
6.5.2. Db.src.....	31
References	32
Annex A. List of variables	33
Annex B. Analysing the model	35
Issued in the series Documents	37

1. Introduction¹

This document describes the model system and presents the computer programs made to operate the Century Model developed by the Research Department of Statistics Norway, for the Ministry of Planning, Saudi Arabia, as part of a United Nations project. Its purpose is to serve as a detailed documentation of the system specific computer programs, and thereby as a reference work for those operating the model.

The Saudi Arabian economy has strong links with the ecological environment, foremost by the economic dependence upon large, but yet depletable resources of petroleum. But there are also other links. Water is a particularly important resource and the current supply is to a great extent drawing down depletable resources. In this context, with the strong dependency upon the petroleum resources, the national income as measured in the national account, is not really income but liquidation of national wealth. The prime purpose of the Century Model, to study the long run sustainability of the Saudi economy, can in stock terms, be formulated as management of national wealth. This suggests the model to be considered as an intertemporal accounting framework.

The model itself, as well as the procedures used to handle the different tasks related to it, is programmed in the Portable TROLL modelling and programming language, and is run on a personal computer under a Windows environment. The TROLL system is particularly useful when it comes to simulation of simultaneous equations, but also provides useful routines for updating data bases and models and for estimation tasks. This document will focus in detail on the programs made specially for the Century Model. Among them are the routines for controlling input and output as well as programs used when forecasting and simulating.

The document is organized as follows: Chapter 2 gives a description of the directories and the file structure used in operating the model. Chapter 3 will present the files that organize the input to TROLL from external files (time series, models, programs, etc.), and the files that direct output to prespecified files or data bases (extrapolation and simulation results, etc.). In chapter 4 we describe how to generate the data to be used in the model into the TROLL data base. This chapter will also present, in full details, the program files made for this purpose. Chapter 5 presents the equations in the model. In chapter 6 programs used for extrapolation and simulation are presented.

In general we have tried to give ample comments inside the source files for the different programs. Because of this we have chosen to present almost all these source files, exactly as they have been written for TROLL, in this document. A /* (slash followed by a multiplication operator) at the beginning of a line indicates that the line contains a comment, and will not be processed by TROLL. Note that several of the chapters headings are, in fact, file names.

2. Directories and file structure

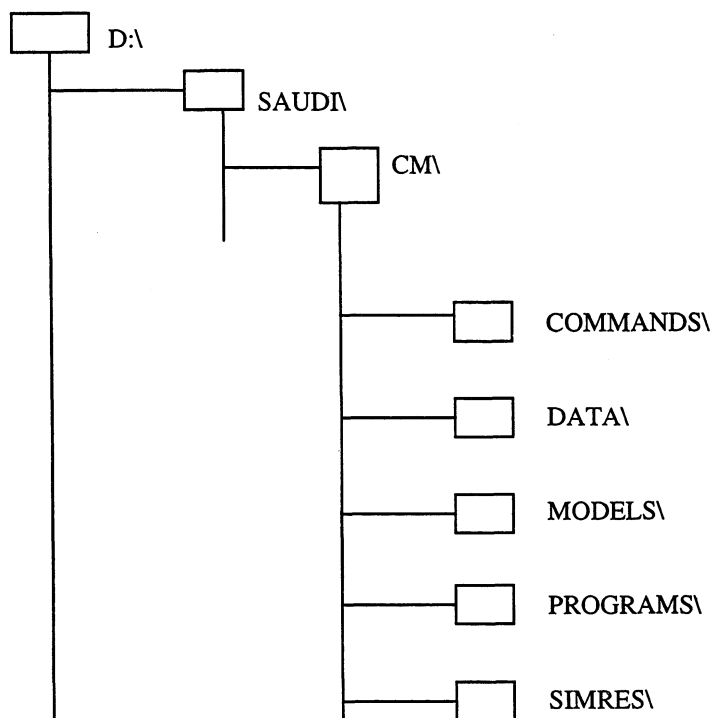
This chapter gives an overview of the files and directories created to operate the Century Model in a proper way. TROLL is very flexible when it comes to decide where to get input and where to direct output, and the organization of the system around the model, data bases, programs etc., is to a large extent limited by the model user, and not by technical restrictions. First we describes the directories and their content, before the different files attached to the model are explained.

¹ We owe special thanks to Einar Bowitz, Statistics Norway, whose assistance in the development of the model system, has been crucial. Many program modules made for the Century Model are modified versions of programs originally developed by him, Nils Ø. Mæhle and Rune Johansen, also Statistics Norway, see Bowitz et al. (1995).

2.1. Directories

If nothing else has been specified, TROLL will read input to the model from the TROLL directory, and save output, for instance simulation results, to the same location. This directory already contains originally designed programs and macros. If we, in addition to them, shall store time series, models, simulation results, etc. at the same directory, it will be almost impossible to keep it organized. Due to this we have chosen to keep the model system separated from the TROLL directory.

Figure 1. Directory structure for the Century Model



The organization of directories is shown in Figure 1. As can be seen, the directories are separated from the TROLL directory (not shown here). The D:\TROLL directory contains the original TROLL software (commands, macros and program files) which we don't want to mix with model specific programs. This requires some kind of link between the directories made for the Century Model and the TROLL system files. This link is established by running the access and search files. All the programs developed by us, except the access and search files, are located in the sub-directories on D:\SAUDI\CM. The access and search files are located at the D:\TROLL directory².

Figure 1 shows the different directories created to organize the different dataobjects in connection with the model system. SAUDI is a main directory where the Century Model, together with the other model systems developed (not shown here), are located. The CM directory itself consist of five sub-directories. The first one, COMMANDS, consists of the text files used to generate the historical time series and the model. These text files contain commands which, when executed by TROLL, establishes the data base and the model in TROLL format, located at the DATA and MODELS directories respectively. The historical time series are saved in a formdata file while the model is saved as an ordinary disk file. It should be noticed that, in contrast to for instance the Implementation Model, there is no sub-directory for estimated coefficients. This is because the coefficients has been

² This name of the root directory (D) is, of course, special for our machines, but it should be noticed that some of the programs are dependent of this name. This means that if these programs are transferred to a machine with another name of the root directory, the programs must be changed.

calibrated and is, technically speaking, defined as exogenous variables and stored together with the historical time series. The reason is to have the opportunity to extrapolate them with different values in different periods. The PROGRAMS directory contains the model specific programs and macros made to operate the Century Model. Note that this directory contains both the compiled and uncompiled versions of the programs and macros. The SIMRES directory, is meant to be used for simulation output. The files on this directory will be of formdata type.

The extensions following the file names indicates what kind of «datatype» the data object is, and plays an important role in controlling input and output. For instance, we have two data objects which are classified as datatype «data» in two different sub-directories. These are the formdata files for the historical time series and for the simulation (and extrapolation) output. This means that we, in some way, have to distinguish between the output from the two tasks (we don't want to mix them together in the same file). We have solved this by using two different search files. One search file is used when updating or reinstalling the historical time series, to ensure that they are stored at the DATA directory, another search file is used when simulating the model to ensure that the simulation results are saved at the SIMRES directory.

In Table 2.1 we show the different sub-directories and what kind of filetypes they contain. As mentioned above there are two directories that consist of formdata files with datatype «data». The sub-directory DATA will consist of one formdata file containing historical time series. The SIMRES sub-directory can consist of many different formdata files for different simulations. For a brief explanation of the different filetypes see Table 3.1 (a more complete description is given in the TROLL Reference Manual, part 3.1.1).

Table 2.1. The directories

<i>Sub-directory</i>	<i>Type (extension)</i>	<i>Description</i>
COMMANDS	disk (INP)	input files.
DATA	formdata (DAT)	historical time series.
MODELS	disk (MOD)	models.
PROGRAMS	disk (SRC and PRG)	source files and programs.
SIMRES	formdata (DAT)	simulated and extrapolated time series.

2.2. Files developed for the Century Model

This chapter presents the files in the different directories and gives a short explanation of their purpose. For more detailed information, specially of the programs, see inside the files and the uncompiled versions of the programs (in general we have tried to give ample information inside the source files. The following five parts, 2.2.1-2.2.5, refers to the sub-directories briefly described in Table 2.1.

2.2.1. Commands³

This directory contains input files. An input file is a «collection» of TROLL commands passed over to, and executed by TROLL, from a file rather than from the keyboard. Generating TROLL commands from a file has the advantage that the commands can be given many times by calling the file, instead of retyping the commands themselves each time we want them executed. Another advantage is that they can easily be edited using a text editor.

³ We have chosen the name «COMMANDS» on this directory because it contains input files (*.INP), which are files that give commands to TROLL through a text (ASCII) file. All commands in these files are possible to execute directly (online in interactive use of TROLL) but this is not recommendable, first of all because it's easier to undertake corrections and updating inside the text file rather than from the console inside TROLL.

Table 2.2.

<i>Files</i>	<i>Description</i>
CM.INP	generates the model.
CMDATA.INP	contains most of the historical time series data.
TRANSDAT.INP	used for calculating historical time series. It defines new variables using time series already defined by CMDATA.INP. In addition it calculates the residuals and calibrates the coefficients to make sure that they are recalculated each time the data base is installed.

2.2.2. Data

The historical time series are stored inside a formdata file. The file is generated from an input file, and it is, through the search and access rules, given the name TIMES.DAT. By running the two input files to generate this formdata file, the old version of the file is given a new name on the same directory. This is done because the old file must either be renamed or deleted before a new file with the same name can be installed.

Table 2.3.

<i>Files</i>	<i>Description</i>
TIMES.DAT	formdata file containing the historical time series.

2.2.3. Programs⁴

This directory contains programs developed by the staff at Statistics Norway to handle the model system efficiently. The programs are written as source files (SRC-files) and then compiled in TROLL and given the extension that shows that this is an executable program file (PRG-file).

It should be noticed that the generality of the programs varies a lot. This means that, for instance, while some programs may be used for all models, others may be specific to one model, or even one simulation. The main programs for organizing the simulations, REFSIM and ALTSIM, are examples of quite general programs, and may be used over and over again, for different simulations and for different models. The input to this programs, for instance EXTRAP and ALT1REF1, are simulation specific. EXTRAP is the name of a file with one specific set of assumptions about the growth rates for the exogenous variables, while ALT1REF1 is a name of a file where we give specific changes in one or more variables relative to the reference path. When another policy analysis is to be undertaken (other growth rates and/or shifting other variables), new names should be given to the files.

Table 2.4.

<i>Files</i>	<i>Description</i>
FOREECON.SRC	for extrapolation of time series, allows the user to set different levels of the variables in different sub-periods (this program is useful for extrapolation of the econometric residuals).
FORECAST.SRC	for extrapolation of time series, allows the user to set different growth rates in different sub-periods.
GORECAST.SRC	for extrapolation of time series, allows the user to set different growth rates in different sub-periods. This is a modified version of FORECAST. See the source file for more details (chapter 6.2.2).
EXTRAP.SRC	this program contains a list of all the exogenous variables in the model. It's purpose is to extrapolate these variables so that the endogenous variables can be simulated beyond the last year of historical data. Each exogenous variable can have different growth rates in different sub-periods. This file is called upon from

⁴ Programs are written as text (ASCII) files, then «compiled» into TROLL format. This means that all programs have two formats, a source file (*.SRC) and a program file (*.PRG). We only show the source files here.

	<p>the program REFSIM, where it is compiled, executed, and then used for input in a simulation. After simulation the formdata file containing the simulation output is stored with the name EXTRAP.DAT. Since this is done automatically, the different versions of EXTRAP should have different names (changes in the file, with respect to extrapolation period, growth rates, etc. should be saved with another name). In this context, the file EXTRAP.SRC is just <u>an example</u> of the program. This example will contain one special way of extrapolating the variables, while another way of extrapolating the variables should be saved with another file name.</p>
REFSIM.SRC	<p>program that organizes the reference path simulation. The program asks for the file where the exogenous variables are prepared for extrapolation, for instance EXTRAP.SRC, then it compiles it and simulates the model using these exogenous variables as input. The reference path is stored with the same name as the file that extrapolates the variables, but with extension DAT.</p>
ALTSIM.SRC	<p>a general program that organizes simulation scenarios, or scenarios where one or more exogenous variables are changed relative to the reference path. The exogenous variables are changed in a source file of the type ALT1REF1.SRC, which is called from this program.</p>
ALT1REF1.SRC	<p>file for making changes in one, or some, of the exogenous variables relative to the reference path, to make alternative scenarios. It passes the variables that are going to be changed to the program DEVIAT.SRC. When different scenarios are of interest, they should be stored with different names, so that this file is just <u>an example</u> of the program. This program is called upon and compiled from ALTSIM. The alternative simulation are then performed and stored in the same formdata file as the reference simulation. The formdata base are stored with the name of this file but with extension DAT e.g. ALT1REF1.DAT.</p> <p>Inside this formdata file the variables from the two different simulations are stored with prefix depending on the names from the simulations. As an example: Assume that we already have performed a reference simulation, saved in the formdata file BASIC.DAT. We want to compare the simulation results with a simulation where we have changed some of the variables, e.g. in the file ALT1REF1.SRC. By using the program ALTSIM, both simulations are stored in a new file called ALT1REF1.DAT. Inside this file the values of the variables from the reference simulation (originally stored in BASIC.DAT) has got the name BASIC_varname, and the values from the alternative simulation will have the name ALT1REF1_varname, where varname is the variable name.</p>
DEVIAT.SRC	<p>the program makes per cent changes in exogenous variables over user specified sub-periods. It receives variables from programs of the type ALT1REF1.SRC. The program can also be called separately.</p>
CHANGEXA.SRC	<p>program that makes absolute changes in exogenous variables over user specified sub-periods. The changes are stored in the dosave datatable and will be deleted when leaving TROLL or when typing DELSAVE (varname). Multiple pairs of sub-periods and changes are possible.</p>
CHANGEXP.SRC	<p>program that makes per cent changes in exogenous variables over user specified sub-periods. The changes are stored in the dosave datatable and will be deleted when leaving TROLL or when typing DELSAVE (varname). Multiple pairs of sub-periods and changes are possible.</p>
DEVIA2.SRC	<p>the program makes absolute changes in exogenous variables over user specified sub-periods. It receives variables from programs of the type ALT1REF1.SRC. The program can also be called separately.</p>
PRTREF.SRC	<p>the program prints output from a model simulation. Assumes that the simulation</p>

EQVAL.SRC	output has been stored in accordance to the REFSIM program. evaluates an individual equation from a model. The program evaluates the left hand side, the right hand side or the residual.
SIM	simulation program that stores the extrapolation and simulation results in a user specified DAT-file.
DB	program to open an «unaccessed» and «unsearched» data base. Prompt the model user for file name and nickname.

2.2.4. Models

The model file (Table 2.5) is generated from the input file CM.INP. This file contains commands to the model editor, the equations in the model and a symbol table. If the purpose is to change the model and then reinstall it, the procedure is to first delete the old version of the model CM.MOD, then perform the changes in the input file for the model CM.INP, before installing it by typing INPUT CM.

The model file is not, in contrast to data files, readable in a text editor. It can, however, be read inside TROLL, in the MODEDIT mode (see the TROLL Reference Manual, parts 4.11-4.12).

Table 2.5.

<i>Files</i>	<i>Description</i>
CM.MOD	the model in TROLL code, generated from the input file CM.INP.

2.2.5. Simres

This directory is aimed at storing the simulation results. Through the access and search rules, we have ensured that one formdata file is always created if the directory is empty. This file, SIMOUT.DAT, will be the target for output when using the FILESIM command when the model is simulated «manually» (Table 2.6). The variables will be saved inside this file with a user supplied prefix. If the file already exist, new variables will add to it.

However, when utilizing the simulation program REFSIM, the output will be saved in a file whose name is the same as the file used to extrapolate the exogenous variables. For example, we have a file called TESTFILE.SRC, consisting of all the exogenous variables to be extrapolated. We use the program REFSIM and, when prompted for «name of the file with the exogenous input to the reference scenario» we type the name TESTFILE. The resulting simulated reference scenario will automatically be saved as TESTFILE.DAT.

A new formdata file will add to this directory if a new reference scenario is simulated. This directory thus, in contrast to the directory for the historical time series, may consist of many formdata files.

Table 2.6.

<i>Files</i>	<i>Description</i>
SIMOUT.DAT	here simulation output are stored when the model are simulated «manually». This means that we call and execute the program that extrapolates the exogenous variables, and then simulates interactively (from the keyboard). Note that an empty version of this file will be created also when the simulation programs are used.

3. Organization of input and output

3.1. Introduction

The file system commands control access to information stored outside TROLL. The information is classified by the following filetypes:

Table 3.1.

<i>Filetype</i>	<i>Description (extension)</i>
DATA	data objects, for instance time series and simulation results (DAT).
INPUT	text files containing TROLL commands (INP).
MODEL	model created by the model editor.
SOURCE	source code for TROLL macros and functions written in text format (SRC).
PROGRAM	compiled code of the source file (PRG).

3.2. Access and search

TROLL can store and retrieve files through the host operating system on the computer. To organize where to get, and where to store the different data objects, some kind of link between the TROLL software and the model system must be established. This is the purpose of the access and search files. The access commands opens data bases and disk directories, while the search commands specifies where TROLL should look for files, and if there is readable or writeable search to them. We have chosen to use only one access file that opens all the relevant data bases, then we have made two different search files, which usage depends on where to store the datatype «data» (DAT-files). As mentioned earlier, both the historical and the simulated time series are defined as datatype «data» by TROLL, and given extension DAT. To be organized we want to store this two different «objects» in different directories. We use the search file that gives writeable search to the directory where we want to store our output. For instance, when we want to install or update the historical time series we must have writeable search to the directory (and formdata file) for the historical time series. We then use the search file SRCHTICM, which has writeable search to this location.

Note that, since the access and search files instructs where to look for other files, no commands are given to tell where to look for these files themselves. That is why they are stored on the TROLL directory, which has default access and search. This chapter will show the files in their full details. We have tried to give comment inside the files to provide ample information to the model user⁵.

3.2.1. Access

The following file opens all directories and data bases (formdata files) needed to use the Century Model.

3.2.1.1. Accesscm.inp

option screen off;

/* Input file ACCESSCM.INP.

/* Input file for the Century Model. Opens databases and disk directories and adds them
/* to the list of accessed data bases.

/* To ensure that the correct access rules are given the next command deletes all (eventually)
/* earlier accessed data bases.

⁵ Comments inside a TROLL program must be indicated by a /* (slash followed by a multiplication operator) at the beginning of the line.

```
delaccess all;
```

```
/* Open the data base for the original timeseries data.  
access cmtim type formdata id d:\saudi\cm\data\times.dat mode w;
```

```
/* Open the directory where the models are stored.  
access cmmod type disk id d:\saudi\cm\models mode w;
```

```
/* Open the directory where the input files are stored.  
access cmcom type disk id d:\saudi\cm\commands mode w;
```

```
/* Open the data base for the simulation output.  
access cmsim type formdata id d:\saudi\cm\simres\simout.dat mode w;
```

```
/* Open the directory where the programs, both compiled and uncompiled, are stored.  
access cmprog type disk id d:\saudi\cm\programs mode w;
```

3.2.2. Search

In this part we presents the two search files. They are distinguished only in where to write dataobjects of type «data» (DAT-files). The first one has writeable search to the historical time series data base, while the second writes to the data base for simulation output.

3.2.2.1. Srchticm.inp

```
option screen off;
```

```
/* Input file SRCHTICM.INP
```

```
/* Input file for the Century Model. Adds entries to the search list, which specifies where  
/* TROLL should look for files. This input file should be used when the time series database is to  
/* be updated or reinstalled. Note that there is given a readable search to the datatypes DATA for  
/* the simulation output (there is no R (read) after this searches because this  
/* is the default value).
```

```
/* To ensure that the correct search rules are given the next command deletes all (eventually)  
/* earlier searches.
```

```
delsearch all ;
```

```
/* Writeable search for the original time series data.  
search data cmtim w;
```

```
/* Readable search for the simulated values.  
search data cmsim ;
```

```
/* Writeable search for the models.  
search model cmmod w;
```

```
/* Writeable search for the input files (exclusive access and search files).  
search input cmcom;
```

```
/* Writeable search for the program files.  
search program cmprog w;
```

```
/* Writeable search for the source files.  
search source cmprog w;
```

3.2.2.2. Srchsicm.inp

option screen off;

/* Input file SRCHSICM.INP

/* Input file for the Century Model. Adds entries to the search list, which specifies where
/* TROLL should look for files. This input file should be used when the simulation output data base is to
/* be used. Note that there is given a readable search to the datatypes DATA for the historical data (there is
/* no R (read) after this searches because this is the default value).

/* To ensure that the correct search rules are given the next command deletes all (eventually)
/* earlier searches.

/* Readable search for the original time series data.
search data cmtim;

/* Writeable search for the simulated values.
search data cmsim w;

/* Writeable search for the models.
search model cmmod w;

/* Writeable search for the input files (exclusive access and search files).
search input cmcom;

/* Writeable search for the program files.
search program cmprog w;

/* Writeable search for the source files.
search source cmprog w;

4. Historical data

4.1. Introduction

The historical data for the variables in the Century Model is located at the \SAUDICM\DATA sub-directory. It consist of the base year observation including necessary lags. Most of the data are taken from the National Account, but also from other sources, such as SAMA, The Sixth Development Plan, OPECs Annual Statistical Bulletin, etc. The data are stored inside the formdata file TIMES.DAT. This file is generated from the two input files CMDATA and TRANSDAT. The first consisting of commands to install new time series into TIMES.DAT. When this variables is installed the remaining data is generated from the TRANSDAT input file. This file assigns values to new variables using variables already installed with CMDATA, for instance summation of government and private consumption into a new variable called total consumption. The advantage of doing it this way is that once the data in CMDATA is updated (by adding new observations), the variables in the TRANSDAT file is automatically updated.

4.2. Cmdata.inp

option screen off;

/* Input file CMDATA.INP.

/* This input file is used to generate the time series database.

/* Her we delete eventually existing accesses and searches, and replace them with the correct ones.

delaccess all;
delsearch all;

input accesscm;
input srchticm;

/* host "command.com /c if exist d:\saudi\lcm\data\times.dat del d:\saudi\lcm\data\times.dat";

/* host "command.com /c copy d:\saudi\lcm\data\times.dat d:\saudi\lcm\data\timesold.dat";

&DEDIT XX 1989A; DATA 7483; FILE;
&DEDIT CPN 1989A; DATA 8808; FILE;
&DEDIT XPS 1989A; DATA 8573; FILE;
&DEDIT M 1989A; DATA 135961; FILE;
&DEDIT EE 1989A; DATA 2071; FILE;
&DEDIT CG 1989A; DATA 114299; FILE;
&DEDIT JG 1989A; DATA 26285; FILE;
&DEDIT YP 1989A; DATA 152991; FILE;
&DEDIT CPR 1989A; DATA 145033 ; FILE;
&DEDIT JP 1989A; DATA 32590 ; FILE;
&DEDIT KP 1988A ; DATA 520730 529989; FILE;
&DEDIT E 1989A ; DATA 6049.5; FILE;
&DEDIT POPSA 1989A; DATA 12310; FILE;
&DEDIT POPNS 1989A; DATA 4638; FILE;
&DEDIT ES 1989A ; DATA 1981.5; FILE ;
&DEDIT EN 1989A ;DATA 4068; FILE;
&DEDIT PE 1989A; DATA 63.75; FILE;
&DEDIT PED 1989A; DATA 20; FILE;
&DEDIT LS 1989A; DATA 6424.2; FILE;
&DEDIT RW 1988A ; DATA 579836 565000; FILE;
&DEDIT AE 1989A; DATA 1609; FILE ;
&DEDIT D 1989A; DATA 714; FILE ;
&DEDIT R 1989A;DATA 0.05; FILE;
&DEDIT T 1988A; DATA 40381 6949; FILE;

&DEDIT ZCUM 1988A; DATA 732210 734481; FILE;
&DEDIT WNR 1989A; DATA 14836; FILE;
&DEDIT EDD 1989A; DATA 71.4; FILE;
&DEDIT EP 1989A; DATA 5300.4; FILE;
&DEDIT RE 1988A; DATA 292539 299080; FILE;
&DEDIT WDC 1989A; DATA 1240; FILE;
&DEDIT EG 1989A; DATA 749.1; FILE;
&DEDIT WD 1989A; DATA 714; FILE;
&DEDIT WDA 1989A; DATA 16400; FILE;
&DEDIT WDY 1989A; DATA 560; FILE;
&DEDIT WO 1989A; DATA 2650; FILE;
&DEDIT EDY 1989A ; DATA 657.81; FILE;
&DEDIT EDG 1989A ; DATA 0; FILE;
&DEDIT EDC 1989A ; DATA 59.82; FILE;
&DEDIT PK 1989A; DATA 0.094804 ; FILE;
&DEDIT GRTOT 1988A; DATA 87305.48 121427; FILE;
&DEDIT GETOT 1988A; DATA 140856.38 154859 ; FILE;
&DEDIT YG 1989A; DATA 57840; FILE;
&DEDIT DS 1989A; DATA 6756; FILE;
&DEDIT TAXE 1989A; DATA 0; FILE;
&DEDIT TAXK 1989A; DATA 0; FILE;
&DEDIT TAXL 1989A; DATA 0; FILE;
&DEDIT YID 1989A; DATA 6740.00 ; FILE;
quit;

4.3. Transdat.inp

option screen off;

/* Input file TRANSDAT.INP.

/* This file is used to generate new variables that is suitable for use in the model. The variables on the
/* right hand side is "assigned" to the variable on the left hand side. This implies that the variables on
/* the right hand side have to be either defined by the "dedit" command, in the input file CMDATA, or,
/* they have to be defined earlier in this file, TRANSDAT.INP, which means they have to be on the left

/* hand side in a previous line. Often the variable defined in this file is very simple definitions of variables,
/* like summation, linear combinations, products etc. The advantage of defining variables in this way,
/* instead of in CMDAT using the "dedit" command, is that once the variables in the CMDATA file is
/* updated, the left hand side variables in this file is automatically updated.

/* We can not use "Value added" in the private sector. Instead we have to use "Value added, included
/* the sum of deliveries from the sectors "Crude oil and natural gas" and "Petroleum refining" to the
/* private non-oil sector.

/* The sum of deliveries from the sectors "Crude oil and natural gas" and "Petroleum refining" to the
/* private sector is computed from the input-output table as:
DOFILE COPRYP = 21676.1;

/* Here we define the new variable, which consist of the sum of value added in the sector plus deliveries
/* from the sectors "Crude oil and natural gas" and "Petroleum refining" to the private non-oil sector.
DOFILE YPA = YP + COPRYP;

/* Here ALPHA is calibrated.
DOFILE ALPHA = PK*(1+TAXK)*KP/YPA;

/* Here ETA is calibrated.
DOFILE ETA = PED*(1+TAXE)*EDY/YPA;

/* We use the assumption of unity elasticity of scale to define BETA.
DOFILE BETA = 1-ALPHA-ETA;

/* Saudi participation ratio into the labour force.
DOFILE LAMBDA = ES/LS;

/* The ratio between population and labour supply.
DOFILE SIGMA = POPSA/LS;

/* The wage rate is calibrated from the equation stating the marginal condition for employment.
DOFILE W = BETA*YPA/((1 + TAXL)*EP);

/* Definition of the consumption of non-Saudis. POP is total population, while EN is non-Saudi
/* employment.
DOFILE CPRN = (POPNS/POP)*CPR;

/* On behalf of this results we can calculate the saving rate for the expatriates. It should be stressed that
/* this result is heavily dependent upon the calculation of deviding of the total residential consumption
/* into Saudi and non-Saudi.
DOFILE SEX = 1 - (CPRN/(W*EN));

/* Private consumption by resident, Saudis, is decided residually.
DOFILE CPRS = CPR - CPRN;

/* These scale factors is now technically defined as exogenous variables, implying that we are able to
/* give them different values in different future periods.
DOFILE AEC = EDC/(CPRS + CPRN);
DOFILE AEG = EDG/(CG+JG);
DOFILE AED = EDD/D;
DOFILE AWC = WDC/(CPRS + CPRN);

DOFILE FR = EE + RE - RE(-1);

/* Depreciation of the capital stock in the private sector.
DOFILE DELTA = (JP-KP+KP(-1))/KP(-1);

/* Link between government employment and government "value added".
DOFILE AYE = YG / EG;

/* Link between government "value added" and the sum of government consumption and government
/* investment.
DOFILE AYG = YG / (CG + JG);

/* Exports of "Other". Exports of goods and services, less "Exports of crude oil" and "Refined products",
/* which is equivalent to the sum of exports of "Petrochemicals", "Non-oil exports" and "Private consumption
/* by non-residents".
DOFILE XO = XPS + XX + CPN;

/* Residuals.

/* All variables starting with the letters RES are residuals. The following letters is the name of the
/* variable on the left hand side in the respective equation. They are defined by turning the equations, and
/* therefore defining the residuals as the left hand side minus the right hand side.

/* Energy extraction.
DOFILE RESEE = EE - (EDY + EDC + EDG + EDD + AE);

/* "Value added, including deliveries from the oil sectors to the private non-oil sector".
DOFILE RESYPA=
M + YPA + YG + YID - (CPRS + CPRN + CG + JG + JP + XO + DS);

/* Accumulated wealth.
DOFILE RESZCUM =
ZCUM - ZCUM(-1) - (PE * AE + XO - M - SEX * W * EN + R * ZCUM(-1));

/* Treasury.
DOFILE REST=
T - T(-1) - (PE * AE + TAXK * JP + TAXE * PED * (EE - AE) + TAXL * W * E - (CG + JG) - EDG * PED);

quit;

5. The model

5.1. Introduction

This chapter presents the input file for the model. A model is, in TROLL terms, a set of equations with an associated symbol table (p. 40 in Hollinger and Spivakovsky, 1993). A model can be made interactively in TROLL's model editor, but this is not recommendable. Instead the model should be written inside a text editor and then generated into TROLL code from this text file (input file). The advantage is that the model can be installed many times just by calling the file name, in addition, it is easier to edit the model in a text editor rather than in the TROLL model editor.

5.2. Cm.inp

option screen off;

/* Input file CM.

/* This input file is used to generate the Century Model into TROLL code.
usemod cm;
modedit;
addeq top,

/* Factor demand is determined according to the marginal conditions for the respective input factor.

$$PK * (1+TAXK) * KP = ALPHA * YPA ,$$

$$W * (1+TAXL) * EP = BETA * YPA ,$$

$$PED * (1+TAXE) * EDY = ETA * YPA ,$$

/* Labour demand.

$$ES + EN = E ,$$

$$E = EP + EG ,$$

$$ES = LAMBDA * LS ,$$

$$POPSA = SIGMA * LS ,$$

/* Energy and water demand.

$$RE(-1) - RE = EE - FR ,$$

$$EDC = AEC * (CPRS + CPRN) ,$$

$$EDG = AEG * (CG + JG) ,$$

$$EDD = AED * D ,$$

$$RW(-1) - RW = WNR ,$$

$$WDC = AWC * (CPRS + CPRN) ,$$

/* Capital accumulation.

$$PK = (R+DELTA) ,$$

$$JP = KP - KP(-1) + DELTA * KP(-1) ,$$

/* Market clearing.

$$M + YPA + YG + YID = CPRS + CPRN + CG + JG + JP + XO + DS + RESYPA ,$$

$$EE = EDY + EDC + EDG + EDD + AE + RESEE ,$$

$$WNR + D + WO = WDC + WDA + WDY ,$$

/* Government sector.

$$YG = AYE * EG ,$$

$$YG = AYG * (CG + JG) ,$$

/* Private final consumption, by non-Saudis.

$$CPRN = (1 - SEX) * W * EN ,$$

/* Balance of payment.

$$PE * AE + XO - M - SEX * W * EN + R * ZCUM(-1) = ZCUM - ZCUM(-1) + RESZCUM ,$$

/* Government budget.

$$T - T(-1) = R * T(-1) + PE * AE + TAXK * JP + TAXE * PED * (EE - AE) + TAXL * W * E - (CG + JG) - EDG * PED + REST ,$$

/* Definition.

$$T - T(-1) = GRTOT - GETOT ;$$

CHANGESYM ENDOGENOUS

KP YPA EP EDY EN ES E LS RE EDC EDG EDD RW WDC PK EE XO ZCUM T WNR GRTOT YG EG
CPRN;

filemod;

quit;

6. Simulating the model

6.1. Introduction

This chapter presents programs developed by Statistics Norway aimed at simplifying the extrapolation and simulation routines, in addition to organize the output from the simulations. The main idea is to write programs that prompt the model user for input, regarding for instance which model to use, which reference scenario and alternative scenario to use, start and end year of simulation, file name where to store output, etc.

6.2. Extrapolating the variables

Before endogenous variables can be simulated there must exist numerical values for the exogenous variables for the whole simulation period. When we perform a historical simulation the values of the exogenous variables are taken from the historical database, but when we want to simulate the endogenous variables ahead in time the exogenous variables must be given future values, or extrapolated. This chapter refers the programs used for extrapolating the exogenous variables. Because the program files (PRG-files) are in TROLL code, and not readable, we show the source files (SRC- files) which are compiled into executable TROLL programs. The source files contain information on how to use the programs.

6.2.1. *Forecast.src*

/* Source file FORECAST.SRC. (Programmed by Rune Johansen, Statistics Norway).

/* A general TROLL program for extrapolation of time series. A time serie can be extrapolated with
/* growth rates based on an existing observation. It is possible to specify many sub-periods with different
/* growth rates.

/* When the program is called it will prompt the user for the name of the variable to extrapolate, number
/* of periods to extrapolate from the last observation, start year of the first sub-period to extrapolate the
/* variable (or END to exit), per cent change in the variable from one period to the next in this sub-period,
/* start year of the next sub-period to extrapolate the variable (or END if finished), per cent change from one
/* year to the next in this sub-period, ... The program will continue to ask for "the start year of the next
/* sub-period" until the user type "END". This is because many pairs of sub-periods and growth rates are
/* possible.

/* Example: Assume we want to extrapolate the value for government consumption. The last observation is
/* in 1992 and we want to extrapolate 10 years. We want the variable to grow by 1.8 per cent, starting in
/* 1993 with 1996 the last year of this growth rate. Further, from 1998 to 2000, we want the variable to grow
/* by 2.1 per cent, and then, for the rest of the extrapolation period, stay constant (0 per cent growth).

/* There are two ways of providing this information to the FORCAST program: the program can be called
/* in the usual way from the TROLL command line, the user will then be prompted for the input needed to
/* perform the calculations. This method is all right for a few variables, but, as is often the case, when many
/* variables is to be extrapolated, it is better to provide the information from a file. To do this we need to know
/* the exact syntax of the program because we pass over all the information to the program without waiting for
/* the prompt.

/* For the above example it will look like this when the program is called from the TROLL
/* command line:

/* TROLL Command: &forecast

/* Give the name of the time serie to be extrapolated: cg

/* Give the number of periods to extrapolate from the last period: 10

/* Give the start year of the first sub-period to change the variable, or type 'END' to exit: 1993

/* Give the per cent change from one period to the next: 1.8

```

/* Give the first year of the next sub-period to change the variable, or type 'END' if finished: 1997
/* Give the per cent change from one period to the next: 0
/* Give the next forecasting year, or type 'END' if finished: 1998
/* Give the per cent change from one period to the next: 2.1
/* Give the first year of the next sub-period to change the variable, or type 'END' if finished: 2001
/* Give the per cent change from one period to the next: 0
/* Give the first year of the next sub-period to change the variable, or type 'END' if finished: end

/* When the information for all the variables is passed over from a file, it will look like this for one of them:

/* &forecast; >> cg 10 1993 1.8 1997 0 1998 2.1 2001 0 end

/* Notice that the answers to the prompt in the first alternative is exactly the same as on the line in the
/* second alternative.

addfun main;
procedure main()
begin;
    >>do;delcore all;
    get series"\nGive the name of the time serie to be extrapolated: " ;
    get number num"\nGive the number of periods to extrapolate from the last period: " ;

/* The variable 'num' decides how far beyond the end date of the original series we want to forecast. If
/* the user don't want to forecast beyond the end date of the original series, num may be set equal to zero.

/* 20 years forecasting period.
/* num = 20;

/* "temp" is a temporary time serie where subsequent extrapolated parts of the original series are stored. It is
/* first created as an extension of the original time serie decided by the variable "num".

    >>docore temp = reshape(seq(&(num)+1)*0,enddate(&(series)));
    >>docore temp = overlay(&(series), temp);
    get p2"\nGive the start year of the first sub-period , or type 'END' to exit: ";

/* The macro is running in a loop until the user type 'END'.

    while(p2 <>"END")
        begin;
            p1=convert(p2, "numeric");
            year = p1-1;
            get number fac"Give the per cent change from one periode to the next:";
            fac2=(1+fac/100);
            on warning nomsg;
            on error nomsg;

/* The extrapolation is done by the 'autocum' function.

            >>docore temp = autocum(temp,0,&(year)a,0,&(fac2));
            on error nomsg;
            get p2"Give the first year of the next sub-period, or type 'END' if finished: ";
        end;

/* The extrapolatioin is completed, and the temporary series is copied to a new series with the same name as
/* the original series, but belonging to a database specified by the search file.

    >>dofile &(series) = temp;
    >>delcore all;

```

```
/* print("Extrapolation is finished.");
```

```
end;
```

6.2.2. Gorecast.src

```
/* Source file GORECAST.SRC. (Programmed by Rune Johansen, Statistics Norway).
```

```
/* Troll program for extrapolation of time series. This program is more robust than the FORECAST program,  
/* but it runs more slowly. If the FORECAST program is used twice, the time series will be extrapolated  
/* twice. The GORECAST program will always start at the specified date (1990 in the previous program).
```

```
addfun main;
```

```
procedure main()
```

```
begin;
```

```
>>do;delcore all;
```

```
get series"\nGive the name of the timeseries to be forecasted: ";
```

```
get number aar2"\nGive the latest year for entering data: ";
```

```
/* The variable 'num' decides how far beyond the end date of the original series we want to forecast. If  
/* you don't want to forecast beyond the end date of the original series, you may set num equal to zero.
```

```
/* 20 years extrapolation period.
```

```
/* num = 20;
```

```
/* 'temp' is a temporary time serie where subsequent extrapolated parts of the original serie are stored.
```

```
/* First we create 'temp' as an extension of the original serie decided by the variable 'num'.
```

```
num=aar2-1990;
```

```
>>docore en = reshape((seq(&(num)+1)*0)+1,1989a);
```

```
>>docore &(series)=&(series)*en;
```

```
>>docore temp = reshape(seq(&(num)+1)*0,1990a);
```

```
>>docore temp = overlay(&(series), temp);
```

```
get p1"\nGive the first year of extrapolation, or type 'END' if finished: ";
```

```
/* The program is running in a loop until the parameter is 'END'.
```

```
while(p1 <> "END")
```

```
begin;
```

```
  p0=convert(p1, "numeric");
```

```
  year = p0-1;
```

```
  get number fac"\nGive the per cent growth rate: ";
```

```
>>do y=values(temp,&(year)A);
```

```
/* If the initial level of the series is < 0, then the factor is given the opposite sign. Otherwise, a negative  
/* annual change to a negative value, would give a positive change, due to the autocum function.
```

```
  x = getdata("Y",1);
```

```
  if (x < 0) then fac= -fac;
```

```
  fac2=(1+fac/100);
```

```
  on warning nomsg;
```

```
  on error nomsg;
```

```
>>docore temp = autocum(temp,0,&(year)a,0,&(fac2));
```

```
on error nomsg;
```

```

        get p1 "\nGive the next forecasting year, or type 'END' if finished: ";
    end;

/* The extrapolation is completed, and the temporary serie is copied to a new serie with the same name as
/* the original series, but belonging to a database specified by the search rules.

        >>dofile &(series) = temp;
        >>delcore all;

Print("");

/* print("Finished with forecasting.");

end;

```

6.2.3. *Extrap.src*

/* Source file EXTRAP.SRC.

/* This file is used to extrapolate the exogenous variables. It contains a list of all this variables,
/* including the residuals. This program make a call for the program 'FORECAST', and the information is
/* passed over to it. It is preferable to use the program "FORECAST" in this way, rather than calling it
/* directly from inside TROLL and give the information interactively. Calling the program from another
/* program, and supporting the information this way, means thay the program can be used over and over again.
/* This is advantageou when there exict many variables to be extrapolated.

/* How to use this program:

/* The program is called in the usual way:

/* &extrap

/* The program will give the message

/* Finished with extrapolating.

/* The information is, for instance in the first line: AE 50 1990 0 end, where AE is the variable
/* name, 50 means 50 year of extrapolation, 1990 means start year, 0 means percent growth and end means
/* that we are finished with entering. The "end" command is necessary because many pairs of "years"
/* and "per cent growth" is possible.

/* Note that this file is just an example of one way of extrapolating the exogenous variable. This means
/* that when we changes the growth rate of one or more of the variables (another policy) it should be
/* saved with another name.

```

addfun main;
procedure main()
begin;

```

/* Here the exogenous variables are "passes" over" to the program FORECAST.

```

&forecast; >> AE 50 1990 0 end
&forecast; >> AEC 50 1990 0 end
&forecast; >> AED 50 1990 0 end
&forecast; >> AEG 50 1990 0 end
&forecast; >> ALPHA 50 1990 0 end
&forecast; >> AWC 50 1990 0 end
&forecast; >> AYE 50 1990 0 end
&forecast; >> AYG 50 1990 0 end
&forecast; >> BETA 50 1990 0 end
&forecast; >> CG 50 1990 0 end

```

```

&forecast; >> CPRS 50 1990 3 2005 2 end
&forecast; >> D 50 1990 6.78 2000 4 end
&forecast; >> DELTA 50 1990 0 end
&forecast; >> DS 50 1990 0 end
&forecast; >> FR 50 1990 0 end
&forecast; >> ETA 50 1990 0 end
&forecast; >> GETOT 50 1990 0 end
&forecast; >> JG 50 1990 3 end
&forecast; >> JP 50 1990 3 end
&forecast; >> LAMBDA 50 1990 0 end
&forecast; >> M 50 1990 0 end
&forecast; >> PE 50 1990 0 2000 2 end
&forecast; >> PED 50 1990 0 2000 2 end
&forecast; >> POPSA 50 1990 3 2010 2 end
&forecast; >> R 50 1990 0 end
&forecast; >> RESEE 50 1990 0 end
&forecast; >> REST 50 1990 0 end
&forecast; >> RESYPA 50 1990 0 end
&forecast; >> RESZCUM 50 1990 0 end
&forecast; >> SIGMA 50 1990 0 end
&forecast; >> TAXE 50 1990 0 end
&forecast; >> TAXK 50 1990 0 end
&forecast; >> TAXL 50 1990 0 end
&forecast; >> W 50 1990 2 end
&forecast; >> WDA 50 1990 -3.66 2000 0 end
&forecast; >> WDY 50 1990 0 end
&forecast; >> WO 50 1990 2.77 2000 0.5 end
&forecast; >> YID 50 1990 0 end
&forecast; >> SEX 50 1990 0 end

```

```

print("");
print("Finished with extrapolating.");
print("");

```

```
end;
```

6.2.4. *Foreecon.src*

/* Source file FOREECON.SRC. (Programmed by Rune Johansen, Statistics Norway).

/* Program that is used to set levels of variables ahead in time, instead of growth rates. When the
 /* extrapolation is completed, the temporary serie is copied to a new serie with the same name as the
 /* original serie, but stored in the database specified by the search rules.

```

addfun main;
procedure main()
begin;

```

```
get series"\nGive the name of the time serie to be extrapolated: ";
```

/* 'temp' is a temporary time serie where subsequent extrapolated parts of the original serie is stored.

```
>>docore temp = &(series);
```

```
get year"\nGive the start year of the first extrapolation sub-period, or type 'END' to exit: ";
```

```
while (year <> "END")
begin;
```

```
get number val"\nGive the value (level) to extrapolate the variable with: ";
```

```

    get number num"\nGive the number of extrapolation years: ";
    >>docore temp2 = reshape(&(val)+seq(&(num))*0,&(year)a);
    >>docore temp3 = overlay(temp2, temp);
    >>docore temp = temp3;
    get year"\nGive the start year of the next extrapolation period, or type 'END' if finished.";
end;

>>dofile &(series) = temp;

```

end;

6.2.5. *Alt1refl.src*

/* Source file ALT1REF1.SRC. (Programmed by Rune Johansen, Statistic Norway).

/* This program is used for changing one or more of the exogenous variables from the reference path, to make an alternative scenario. It passes information about the exogenous variables to the program DEVIAT, where they are given shift in user specified sub-periods.

/* Example: Assume we have already extrapolated the exogenous variables, and made a reference simulation with these as input for the periode 1990 to 2010. In this extrapolation, and thus as input to the reference simulation, the government consumption was extrapolated with a 0 per cent growth rate. We are interested in studying the effects of government consumption growing 2.3 percent in the period 1998 to 2003. This will be our alternative scenario. The necessary information passed over to the program DEVIAT with the following syntax: CG 1998 2003 2.3 end

/* The 'END' statment is needed because it indicates that we are finished entering pairs of sub-periods and growth rates relative to the reference scenario.

/* The variable CG will be extrapolated in the same way as it was as input to the reference simulation, apart from the period 1998 - 2003, where it will stay 2.3 per cent above the extrapolated variable used in the reference simulation. The shift will be from 1997 to 1998, so that 1998 is the first year CG is 2.3 per cent higher. Thereafter, until 2003, CG will grow by 0 per cent, or, it will stay constant 2.3 per cent above than in the reference case (paralell). In 2004 CG will "be back" to the reference value.

/* Notice that the name of this file (source file) is arbitrary, a different scenario should be given a different name.

```

addfun main;
procedure main()
begin;

    &deviat; >> cg 1998 2003 2.3 end

    print("");
    print("Finished changing exogenous variables.");
    print("");

```

end;

6.2.6. *Deviat.src*

/* Source file DEVIAT.SRC. (Programmed by Rune Johansen, Statistic Norway).

/* This program makes changes in variables over user specified sub-periods. The main purpose of the program is to change (extrapolated values of) exogenous variables to be used in simulations.

/* Example: Assume there exist a time serie CG which is extrapolated from 1990 to 2010 with 1.2 per cent growth. This could be input to a reference simulation. We want to study the effects of a 2 per cent growth, rather than 1.2, in the sub-period 1998 - 2003.

```

/* In TROLL this example would look like this:

/* TROLL Command: &deviat
/* Give the name of the time serie to change: cg
/* Give the first year of the sub-period to change the variable, or type 'END' to exit: 1995
/* Give the last year of this sub-period: 2000
/* Give the per cent change: 10
/* Give the start year of the next sub-period, or type 'END' if finished: end

/* This example will increase CG from 1994 to 1995 by 10 per cent. CG will then stay at this level
/* until 2000, and, in year 2001, it will "be back" to the previous value.

/* It should be noticed that multiple pairs of sub-periods and per cent growth are possible.

addfun main;
procedure main()
begin;

    >>do;delcore all;
    get series"\nGive the name of the time serie to change: ";

/* Make a variablename 'temp', where subsequent parts parts of the original series are temporarily stored.

>>docore temp = &(series);

get p1"\nGive the first year of the sub-period to change the variable, or type 'END' to exit: ";

/* The program is running in a loop until the user type 'END'.

    while(P1<> "END")
        begin;
            year1 = p1;
            get year2"\nGive the last year of this sub-period: ";
            get number percent"\nGive the per cent change: ";
            on warning nomsg;
            on error nomsg;
            >>docore temp2 =
            >>((100+&(percent))/100)*subrange(temp,&(year1)a,&(year2)a);
            >>docore temp3 = subrange(temp,startdate(&(series)),&(year1)a);
            >>docore temp4 = subrange(temp,&(year2)a,enddate(&(series)));
            >>docore temp = overlay(temp2, temp3, temp4);
            on error nomsg;
            get p1"\n Give the start year of the next sub-period, or type 'END' if finished: ";
        end;

/* The new time series is stored under its original name, but at the database specified by the search file.

    >>dofile &(series) = temp;
    >>delcore all;

/* print("Finished with changing exogenous variable");

end;

```

6.2.7. *Changexa.src*

/* Source file CHANGEEXA.SRC. (Programmed by Rune Johansen and Einar Bowitz, Statistics Norway).

/* This program makes absolute changes in variables over user specified sub-periods. The user gives


```

/* variable name, start year of first sub-period, end year of first sub-period, and absolute change in values for
/* that period, start year of next sub-period, end year of next sub-period, absolute change in this sub-period, ...
/* The user type 'END' if he don't want to specify another sub-period (start year and end year). So multiple
/* sub-periods and changes in levels are allowed.

```

```

/* The change in the variables is only kept in the troll memory and will be deleted when leaving troll or
/* by typing delsave all.

```

```

addfun main;
procedure main()
begin;
    >>do; delcore all;
    get series "\nGive the name of the time serie to change: ";

```

```

/* 'temp' is a temporary timeseries where subsequent parts of the original series are stored.

```

```

    >>docore temp = &(series);
    get p1 "\nGive the start year of the first sub-period to change the variable, or type 'END' to exit: ";

```

```

/* The macro is running in a loop until the user types 'END'.

```

```

while(p1 <> "END")
    begin;
        year1 = p1;
        get year2 "\nGive the last year of this sub-period: ";
        get number absol "\nGive the absolute change: ";
        on warning nomsg;
        on error nomsg;
        >>docore temp2 =
        >> &(absol)+subrange(temp,&(year1)a,&(year2)a);
        >>docore temp3 = subrange(temp,startdate(&(series)),&(year1)a);
        >>docore temp4 = subrange(temp,&(year2)a,enddate(&(series)));
        >>docore temp = overlay(temp2, temp3, temp4);
        on error nomsg;
        get p1 "\nGive the start year of the next sub-period to change the variable, or type 'END' if finished:";

        end;

        >>do &(series) = temp;
        >>delcore all;

```

```

/* print("Finished changing the variable.");

```

```

end;

```

6.2.8. *Changexp.src*

```

/* Source file CHANGEXP.SRC.

```

```

/* Program that makes changes in variables over user specified sub-periods. The user gives variable
/* name, start year of sub-period, end year of sub-period, per cent change in values for that period, and, if
/* finished the user type 'END'. The 'END' is necessary because many pairs of sub-periods and per cent
/* growth is allowed.

```

```

/* The changes is only kept in the TROLL memory and will be deleted when leaving TROLL or by typing
/* delsave all.

```

```

addfun main;
procedure main()

```

```

begin;

    >>do;delcore all;
    get series"\nGive the name of the time serie to change: ";

/* 'temp' is a temporary timeseries where subsequent parts parts of the original series are stored.

    >>docore temp = &(series);

    get p1"\nGive the start year of the first sub-period to change the variable, or type 'END' to exit: ";

/* The macro is running in a loop until the user type 'END'.

    while(p1 <> "END")
    begin;
        year1 = p1;
        get year2"\nGive the last year of this sub-period: ";
        get number percent"\nGive the per cent change: ";
        on warning nomsg;
        on error nomsg;
        >>docore temp2 =
        >>((100+&(percent))/100)*subrange(temp,&(year1)a,&(year2)a);
        >>docore temp3 = subrange(temp,startdate(&(series)),&(year1)a);
        >>docore temp4 = subrange(temp,&(year2)a,enddate(&(series)));
        >>docore temp = overlay(temp2, temp3, temp4);
        on error nomsg;
    get p1"\nGive the start year of the next sub-period to change the variable, or type 'END' if finished: ";

    end;

    >>do &(series) = temp;

    >>delcore all;

/* print("Finished with changing the variable.");

end;

```

6.3. Simulating the variables

The previous section presented the program files aimed at extrapolate, and make changes in, exogenous variables. When the extrapolation is completed, this variables will be used as input to a simulation model. This section presents the main programs developed for organizing the simulations.

6.3.1. Refsim.src

/* Source file REFSIM.SRC.

/* This is the main program to organize the reference simulations. The program requires that the program used
/* to extrapolate the exogenous variables have been prepared in advance, and that it is ready to be called from
/* this program.

/* The program ask for which model to use, the name of the file (program) containing the exogenous variables
/* to extrapolate, and the start and end year of the simulation period.

/* This program compile and execute the program used to extrapolate the exogenous variables. Then these
/* variables are used as input to the model, which is used to simulate the endogenous variables for the specified
/* period.

```

/* The program automatically gives search to the directory for the simulation output. The results will be saved
/* in a formdata file at this directory. The name of this file will be the same as of the file used to extrapolate the
/* exogenous variables, but it will have extension DAT.

```

```

addfun main;
procedure main()
begin;

```

```

/* First we delete all searches and accesses eventually given, before we give the correct ones.

```

```

    >>delaccess all;
    >>delsearch all;
    >>input accesscm;
    >>input srchsicm

```

```

/* We ask for the names to be used.

```

```

    get modname"Give the name of the model: ";
    get refname"Give the name of the file containing the exogenous variables to be extrapolated: ";
    get stperiod"Give the start year of the simulation period (four digits only ): ";
    get endperiod"Give the end year of the simulation period (four digits only ): ";

```

```

/* We open the database for the simulation output, and create a new formdata file with the same name
/* as the file that generates the reference path. The internal name of this database is REF.

```

```

    >>access ref type formdata id d:\saudi\cm\simres\&(refname).dat mode create;

```

```

/* Here we give writeable search to this formdata file, so that the simulation result will be store here.

```

```

    >>Search data ref w;

```

```

/* Here we give search to the historical database, to be used when extrapolating the exogenous variables.

```

```

    >>search first data cmtim;

```

```

    >>compile &(refname);
    &(refname);
    >>search first data ref w;
    >>usemod &(modname);
    >>simulate;
    >>simstart &(stperiod)a;
    >>dotil &(endperiod)a;
    >>filesim &(refname);
    >>quit;

```

```

print("");
    print("Finished with simulation.");
    print("The reference simulation is saved in D:\SAUDI\CM\SIMRES\\",refname, ".DAT.\nwith the
prefix ",refname);
print("");

```

```

end;

```

6.3.2. *Altsim.src*

```

/* Source file ALTSIM.SRC.

```

```

/* Main program that organizes the simulations scenarios. Scenarios is defined as deviations from a reference
/* path due to changes in one or more of the exogenous variables.

```

```

/* This program requires an existing reference path (made for instance by the program REFSIM), and that a file
/* (program) which will be used for making change in one or more of the exogenous variables, has been

```

```

/* prepared.

/* The program ask the user for model name, the name of the reference path, the name of the alternative
/* scenario, and the start and end year of the simulation period.

/* When the necessary information has been provided the reference simulation is copied to a new name, then
/* the changes in the exogenous variables are undertaken and the endogenous variables are simulated.

addfun main;
procedure main()
begin;

    >>delsearch all;
    >>delaccess all;
    >>input accesscm;
    >>input srchticm;

    get modname"\nGive the name of the model: ";
    get refname"\nGive the name of the reference path: ";
    get sn"\nGive the name of the alternative scenario: ";
    get stperiod"\nGive the start year of the simulation period (four digits only): ";
    get endperiod"Give the end year of the simulation period (four digits only): ";

>>host "command.com /c copy d:\saudi\cm\simres\&(refname).dat d:\saudi\cm\simres\&(sn).dat";

/* Simulation input and output are stored here.
    >>access senn type formdata id d:\saudi\cm\simres\&(sn).dat mode w;

/* The reference simulation is stored here (the reference path).
    >> access ref type formdata id d:\saudi\cm\simres\&(refname).dat mode r;

    >>Search data senn w;
    >>compile &(sn);

/* Call macro that contains exogenous variables that is changed.
    &(sn);

    >> usemod &(modname);
    >> simulate;
    >> simstart &(stperiod)a; dotil &(endperiod)a;
    >> filesim &(sn);

    >>quit;

    print("");
    print("End of simulation.");
    print("The simulation results are stored in D:\SAUDI\CM\SIMRES\", sn, ".DAT." );
    print("The variables in the reference simulation has the prefix ", refname, " while the variables in the
alternative simulation has the prefix ", sn, ".");

    print("");

end;

```

6.3.3. *Sim.src*

/* Source file SIM.SRC. (Programmed by Robin Choudhury, Statistics Norway).

/* The purpose of this program is to prompt the user for the information needed to perform both

```
/* a reference and an alternative simulation. This information is: a file name to save the result in,  
/* the name of the file (program) containing the exogenous variables to extrapolate, the model name, the  
/* start and end year of the simulation period, the name of the file (program) containing the variable(s) to  
/* shift.
```

```
/* What distinguishes this program from the other simulation programs, is that it will save the results in a  
/* user specified file (formdata), and that it performs both a reference and an alternative simulation.
```

```
addfun main;  
procedure main()  
begin;  
  
    get file"\nGive a file name to save simulations results in (max. 8 characters, no extension): ";  
  
    >>access rrr type formdata id d:\saudi\cm\simres\&(file).dat mode w;  
    >>search data rrr w;  
  
    get refname"\nGive the name of the file containing the exogenous variables to extrapolate: ";  
  
    >>option screen off;  
    >>compile &(refname);  
    &(refname);  
    >>option screen on;  
  
    get modname"\nGive the name of the model: ";  
  
    print("");  
  
    >>usemod &(modname);  
    >>simulate;  
  
    print("");  
  
    get first"\nGive the first year of the simulation period: ";  
    get last"\nGive the last year of the simulation period: ";  
    get alname"\nGive the name of the file containing the exogenous variable(s) to shift: ";  
  
    print("");  
  
    >>option screen off;  
  
    >>simulate;  
    >>simstart &(first)a;  
    >>dotil &(last)a;  
    >>filesim &(refname);  
  
    >>compile &(alname);  
    &(alname);  
    >>simulate;  
    >>simstart &(first)a;  
    >>dotil &(last)a;  
    >>filesim &(alname);  
  
    >>option screen on;  
  
    print("");  
  
    print("End of simulations.");
```

```

print("The simulations results are saved as D:\SAUDI\CM\SIMRES\", file, ".DAT.");
print("The variables simulated using the file: ", refname, ".PRG has got prefix ", refname, ".");
print("The variables simulated using the file: ", altname, ".PRG has got prefix ", altname, ".");

print("");
>>quit;

end;

```

6.4. Printing results to screen

TROLL has built in programs for presenting the different dataobjects to screen. However, there have been made some programs that more easily recognize what variables we want to print and where to locate them.

6.4.1. Prtref.src

/* Source file prtref.src.

```

/* This program prints output from a simulation. The only information needed is the name of the
/* simulation. The program prompt the user for the name of the variable, and when program
/* REFSIM has been used the simulation has the same first name as the file extrapolating the
/* exogenous variables. The simulated variables has a prefix with the same name, but the user shall
/* not give this prefix, only "original" variable name.

```

```

/* The program print out the value and the per cent change in the variable from last periode.

```

```

addfun main ;
procedure main()
begin;
    on warning nomsg;
    get rs "Give the name of reference simulation: " ;

    >> access rssea TYPE formdata id d:\saudi\cm\simres\&(rs).dat MODE r;
    >> search FIRST data rssea ;

    start1:
        get va "Give the name of the variable (without prefix), or type 'END' if finished: ";
        if (va <> "END") then
            begin;
                >> do prt.(truncate(&(rs)_&(va),3));
                >> do print("Per cent change from last periode is: ");
                >> do prt.(truncate(100*(&(rs)_&(va)/&(rs)_&(va)-1),3));
            end;
        else
            goto start2;
    goto start1;
    start2:

    >> delsearch rssea ;
    >> delaccess rssea ;
    >> quit;

end;

```

6.5. Miscellaneous programs

6.5.1. *Eqeval.src*

/* Source file EQEVAL.SRC. (Programmed by Robin Choudhury, Statistics Norway).

/* The eqeval program evaluates the equations in a model by calling the built in eqeval TROLL function.
/* This is very useful to see that no errors prevent the left hand sides to be equal to the right hand sides. The
/* program prompt the user for: the model name, the number of the first equation, the number of the last
/* equation, and one of three possible choices: evaluation of the right hand side, the left hand side or the
/* residuals.

```
addfun main;
procedure main()
begin;

    on warning nomsg;
    get modname"\nGive the name of the model: ";
    get number eqfrom"\nGive the number of the first equation: ";
    get number eqto"\nGive the number of the last equation: ";

    start1:

get code"\nType '0' if you want to evaluate the right hand side, '1' for the left hand side or '2' for residuals: ";

        if (code <> "0" and code <> "1" and code <> "2") then
            begin;
            print(" ");

            print("ERROR. Not an option.");

            /* Retype if the choice is not 0, 1 or 2.
            goto start1;
        end;

        /* This is to make two extra line feed.
        print("");
        print("");

        >>usemod &(modname);

        if (code == "0") then
            print("Evaluation of the right hand side: ");

        else if (code == "1") then
            print("Evaluation of the left hand side: ");

        else if (code == "2") then
            print("Evaluation of the residuals: ");

        for(i= eqfrom; i<= eqto; i=i+1)
            begin;
            >> do prt.( eqeval(&(i), &(code) ));

        end;

print("\nFinished with eqeval.");
```

end;

6.5.2. Db.src

/ Source file DB.SRC. (Programmed by Robin Choudhury and Knut A. Magnussen, Statistics Norway).*

/ Program to open an unaccessed database.*

/ When a simulation is performed and the results are saved in the database, we are given temporarily
/* access and search to this. But as soon as we stop TROLL or type delaccess all anddelsearch all we loose
/* this access and search. This program can be used to avoid typing the long access and search commands.*

addfun main;

procedure main()

begin;

 get database"Give the name of the database to open (no extension): ";

 get nickname"Give a nickname for the database to open: ";

 >>access &(nickname) type formdata id d:\saudi\cm\simres\&(database).dat mode w;

 >>search data &(nickname) w;

end;

References

Coopers & Lybrand (1981): A Social Accounting Matrix for the Kingdom of Saudi Arabia 1396/97 (1976), Ministry of Planning.

Bjerkholt, O. (1993): Review of the Macroeconomic Modelling Needs of the Ministry of Planning of the Kingdom of Saudi Arabia, Ministry of Planning/UNDDSMS, August 1993.

Saudi Arabian Monetary Agency (1993): annual report 1411/1412 (1991).

Hollinger, P. and L. Spivakovsky (1993): Portable TROLL 0.86. Reference Manual, Intex Solutions, Inc.

Hollinger, P., L. Spivakovsky and P. Barrow (1994): Portable TROLL Programming Language for release 0.93, Intex Solutions, Inc.

Bowitz, E., N. Ø. Mæhle, V. S. Sasmitawidjaja and S. B. Widoyono (1995): MEMLI-An Environmental Model for Indonesia. Technical Documentation of Data Programs and Procedures, Documents 95/7, Statistics Norway.

Ministry of Planning, Kingdom of Saudi Arabia (1996): Sixth Development Plan, 1415-1420 A.H./1995-2000 A.D.

Bjerkholt, O., K.A. Brekke and R. Choudhury (1996): The Century Model - on the long term sustainability of the Saudi Arabian economy, Documents 96/7, Statistics Norway.

Annex A. List of variables

Exogenous

AE	Energy exports
AEC	Energy demand in private sector relative to private consumption by residents
AED	Energy demand for desalination relative to the level of desalination
AEG	Energy demand in government sector relative to government consumption
ALPHA	Elasticity of production with respect to capital in the private sector
AWC	Water demand in the private sector relative to private consumption
AYE	Value added in the government sector relative to employment in the government sector
AYG	Value added in the government sector relative to the sum of government consumption and gross fixed capital formation
BETA	Elasticity of production with respect to employment in the private sector
CG	Government consumption
CPRS	Private consumption by residents, Saudis
D	Desalination
DELTA	Rate of depreciation
DS	Increase in stocks
ETA	Elasticity of production with respect to use of energy in the private sector
GETOT	Total government expenditure
JG	Gross fixed capital formation in the government sector
JP	Gross fixed capital formation in the private sector
LAMBDA	Actual employment relative to labour supply
PE	Price on energy, Saudi Riyals
PED	Price on energy, domestic use
POPSA	Saudi population
R	Interest rate
RESRE	Residual for energy reserves
REST	Residual for treasury
RESYPA	Residual for «value added, including deliveries from the oil activities to the private non-oil sector»
RESZCUM	Residual for the accumulated wealth
SIGMA	Population relative to labour supply
SEX	Saving rate, expatriates
TAXE	Tax on energy
TAXK	Tax on capital
TAXL	Tax on labour
W	Wage rate
WDA	Water demand, agriculture
WDY	Water demand, private sector
WO	Water supply, other sources
XO	Exports of other goods and services. Total export of goods and services less export of crude oil and refined products.
YID	Import duties

Endogenous

CPRN	Private consumption by residents, non-Saudis
E	Total employment
EDC	Energy demand, household
EDD	Energy demand from desalination
EDG	Energy demand from government sector
EDY	Energy demand from private sector
EE	Energy extraction
EG	Employment in government sector
EN	Employment by non-Saudis
EP	Employment in private sector
ES	Employment by Saudis
FR	Future recoveries of oil and gas

GRTOT	Total government revenues
KP	Capital stock in the private sector
LS	Labour supply
M	Imports
PK	User price on capital
RE	Energy reserves
RW	Water reserves, non renewable
T	Treasury
WDC	Water demand by households
WNR	Water resources, non renewable (ground water)
YG	Value added in government sector
YP	Value added in private sector

Annex B. Analysing the model

The following table shows which variable appearing in which equation for the Century Model (CM).

<i>Block</i>	<i>Size Eqn Var</i>
CPRN	9 13 16 21
E	4 5 23
EDC	9 17
EDD	11 17
EDG	10 17 23
EDY	3 17
EE	8 17 23
EG	5 19
EN	4 21 22
EP	2 5
ES	4 6
GRTOT	24
KP	1 15
LS	6 7
M	16 22
PK	1 14
RE	8
RW	12
T	23 24
WDC	13 18
WNR	12 18
YG	16 19 20
YPA	1 2 3 16
ZCUM	22

In the next table we present the normalization and the block ordering of the Century Model. The normalization pairs endogenous variables to the equation used to solve for that particular variable, while the block ordering «breaks down» the model to the smallest possible number of equation that need to be solved simultaneous. If the largest block is one (consist of one equation), as is the case in the Century Model, the model is completely recursive (it is solved equation by equation). If the smallest block is greater than one, the equations in that block needs to be solved simultaneously.

<i>Block</i>	<i>Size Eqn Var</i>
1	1 14 PK
2	1 15 KP
3	1 1 YPA
4	1 2 EP
5	1 3 EDY
6	1 7 LS
7	1 6 ES
8	1 20 YG
9	1 19 EG
10	1 5 E
11	1 4 EN
12	1 21 CPRN
13	1 9 EDC
14	1 10 EDG

<i>Block</i>	<i>Size</i>	<i>Eqn</i>	<i>Var</i>
15	1	11	EDD
16	1	17	EE
17	1	8	RE
18	1	13	WDC
19	1	18	WNR
20	1	12	RW
21	1	16	M
22	1	22	ZCUM
23	1	23	T
24	1	24	GRTOT

Issued in the series Documents

- 94/1 H. Vennemo (1994): Welfare and the Environment. Implications of a Recent Tax Reform in Norway
- 94/2 K.H. Alfsen (1994): Natural Resource Accounting and Analysis in Norway
- 94/3 O. Bjerkholt (1994): Ragnar Frisch 1895-1995
- 95/1 A.R. Swensen (1995): Simple Examples on Smoothing Macroeconomic Time Series
- 95/2 E.Gjelsvik, T. Johnsen, H.T. Mysen and A. Valdimarsson (1995): Energy Demand in Iceland
- 95/3 C. Zhao, O. Bjerkholt, T. Halvorsen and Y. Zhu (1995): The Flow of Funds Accounts in China
- 95/4 Nordic Indicator Group (1995): Nordic Environmental Indicators. Draft document. English version with main points from comments received
- 95/5 H.A. Gravningsmyhr (1995): Analysing Effects of Removing Survivors' Pensions, Using the Microsimulation Model LOTTE
- 95/6 P. Boug (1995): User's Guide. The SEEM-model Version 2.0
- 95/7 E. Bowitz, N.Ø. Mæhle, V.S. Sasmitawidjaja and S.B. Widoyono (1995): MEMLI – An Environmental model for Indonesia. Technical Documentation of data programs and procedures
- 95/8 K. H. Alfsen, T. Bye, S. Glomsrød and H. Wiig (1995): Integrated Assessment of Soil Degradation and Economic Growth in Ghana
- 95/9 O. Bjerkholt (1995): Ragnar Frisch and the Foundation of the Econometric Society and *Econometrica*
- 95/10 P.J. Bjerve (1995): The Influence of Ragnar Frisch on Macroeconomic Planning and Policy in Norway
- 96/1 D. Kolsrud (1996): Documentation of Computer Programs that Extend the SEEM Model and Provide a Link to the RAINS Model
- 96/2 E. Bowitz, V.S. Sasmitawidjaja and G. Sugiarto (1996): The Indonesian Economy and Emission of CO₂: An Analysis Based on the Environmental-Macroeconomic-Model MEMLI, 1990-2020
- 96/3 H.V. Sæbø (1996): Use of Geographical Information Systems (GIS) in Statistics Norway
- 96/4 E. Berg (1996): Some Results from the Literature on the Impact of Carbon Taxes on the Petroleum Wealth
- 96/5 E.J. Fløttum (1996): Norwegian National Accounts – Documentation of the Compilation and Methods Applied. I General Description. II GDP From the Output Approach
- 96/6 E.J. Fløttum (1996): Norwegian National Accounts – Documentation of the Compilation and Methods Applied. III GDP by Expenditure Approach. IV GDP by Income Approach
- 96/7 O. Bjerkholt, K.A. Brekke and R. Choudhury (1996): The Century Model - on the Long Term Sustainability of the Saudi Arabian Economy
- 96/8 R. Choudhury (1996): The Century Model. Technical Documentation of Computer Programs and Procedures
- 96/9 R. Choudhury and K.A. Magnussen (1996): The Implementation Model. Technical Documentation of Computer Programs and Procedures
- 96/10 R. Choudhury (1996): The Selection Model. Technical Documentation of Computer Programs and Procedures
- 96/11 R. Choudhury (1996): The OM95 – An Oil Model for the Kingdom of Saudi Arabia. Technical Documentation of Computer Programs and Procedures

Statistics Norway
Research Department
P.O.B. 8131 Dep.
N-0033 Oslo

Tel.: + 47-22 86 45 00
Fax: + 47-22 11 12 38

ISSN 0805-9411



Statistics Norway