NEREUS

Núcleo de Economia Regional e Urbana da Universidade de São Paulo

The University of São Paulo Regional and Urban Economics Lab

OCP POLICY CENTER

THINK • STIMULATE • BRIDGE

Lecture 3: Introduction to GEMPACK

"Multi-regional Economic Modeling: Applications for Morocco"

October 10-13, 2017 – OCP Policy Center, Rabat

Eduardo A. Haddad and Vinicius Vale

History

The **Impact Project** started in 1975 as part of the Industries Assistance Commission (now Productivity Commission)

The aim of the Impact Project was to produce general tools of use to all economists. These include the ORANI model and GEMPACK software

GEMPACK development began in 1984.

The main developers of GEMPACK are Jill Harrison, Mark Horridge and Ken Pearson

Outline

- ✓ Introduction
- How to carry out a simulation?
- How to implement the SJ model in GEMPACK?

GEMPACK software

General purpose package for GE models, **not model specific**

Allows modelers to **concentrate on the economics** of their models instead of computing problems

Aims to **make modelers more productive**

Document models for others

Overview of GEMPACK Software

GEMPACK is a suite of general-purpose economic modeling software especially suitable for general and partial equilibrium models.

It can handle a wide range of economic behavior and also contains powerful capabilities for solving intertemporal models

GEMPACK provides software for calculating accurate solutions of an economic model, starting from an algebraic representation of the equations of the model.

Overview of GEMPACK Software

GEMPACK provides:

a) a simple language in which to describe and document the equations of your economic model;

b) a program which converts the equations of your model to a form ready for running simulations with the model;

c) options for varying the choice of exogenous and endogenous variables and the variables shocked;

d) utility programs to assist in managing the database on which the model is based.

Overview of GEMPACK Software

The (main) GEMPACK programs:

- ***WinGEM*** – windows interface to GEMPACK
- ***ViewHAR*** – for looking in the data in a Header Array File
- ***ViewSOL*** – for looking at Solutions files
- ***RunGEM*** – for automating simulations with models
- ***TABmate*** – text editor for developing TABLO Input files

Different versions

Source-Code version

- Fortran compiler
- Model size limited only by memory

Executable-Image version

- Can build and modify models
- Medium size models
- No Fortran compiler

Demonstration version

- Can build and modify small models
- Essentially free

Outline

Introduction

✓ How to carry out a simulation?

How to implement the SJ model in GEMPACK?

Overview of GEMPACK Software

In this part we will explain some of the terms used in GEMPACK;

Implementation

Simulation

Levels and percentage-change variables

Implementation

Write down equations in algebraic form

Collect data for initial solution

Construct TABLO Input file

Implementation

A model is implemented in GEMPACK when:

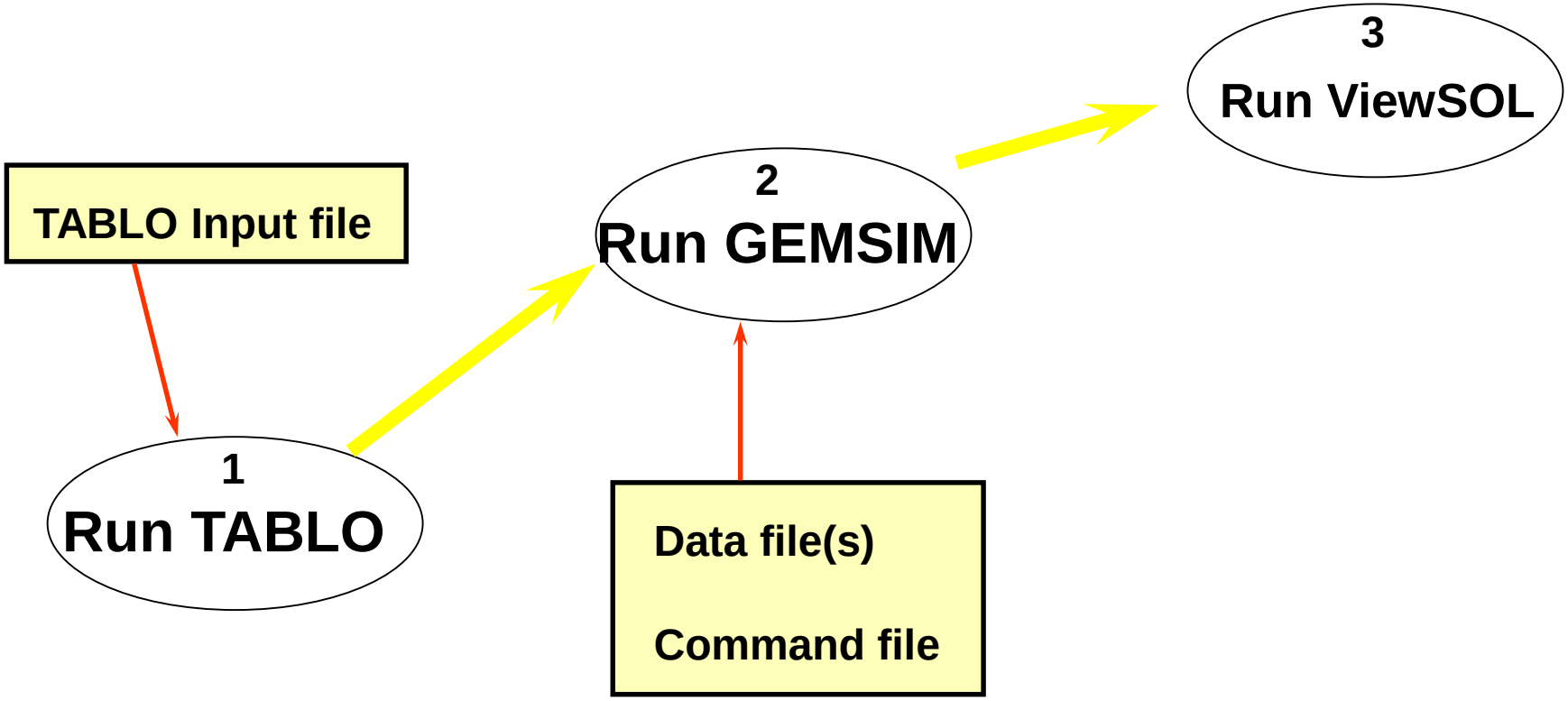
a) the equations describing its economic behavior are written down in an algebraic form following a syntax. (TABLO Input File)

b) data describing one solution of the model are assembled to be used as a starting point for simulations (ViewHAR)

Two GEMPACK Programs

Program	*Purpose*
TABLO	**Converts text file *TABLO Input file* into computer files called *GEMSIM Auxiliary files***
	e.g. converts SJ.TAB into SJ.GSS and SJ.GST
GEMSIM	**Does the arithmetic, carries out simulations**
	e.g. reads SJ data, calculates COEFFICIENTS etc.
	e.g. calculates results of simulation

Three GEMPACK Programs



Simulation

- Many simulations are the answer to "WHAT IF" question such as:

  "If the government were to increase tariffs by 10 percent, how much different would the economy be in 5 years time from what it would otherwise have been?"

- Comparative-statics

Simulation

- From the original solution supplied as the starting point, a simulation calculates a ***new solution*** to the equations of the model.
- Within GEMPACK, the results of a simulation are **usually reported as percentage changes from the original solution**.
- Solving models within GEMPACK is always done in the context of a simulation.

Simulation information

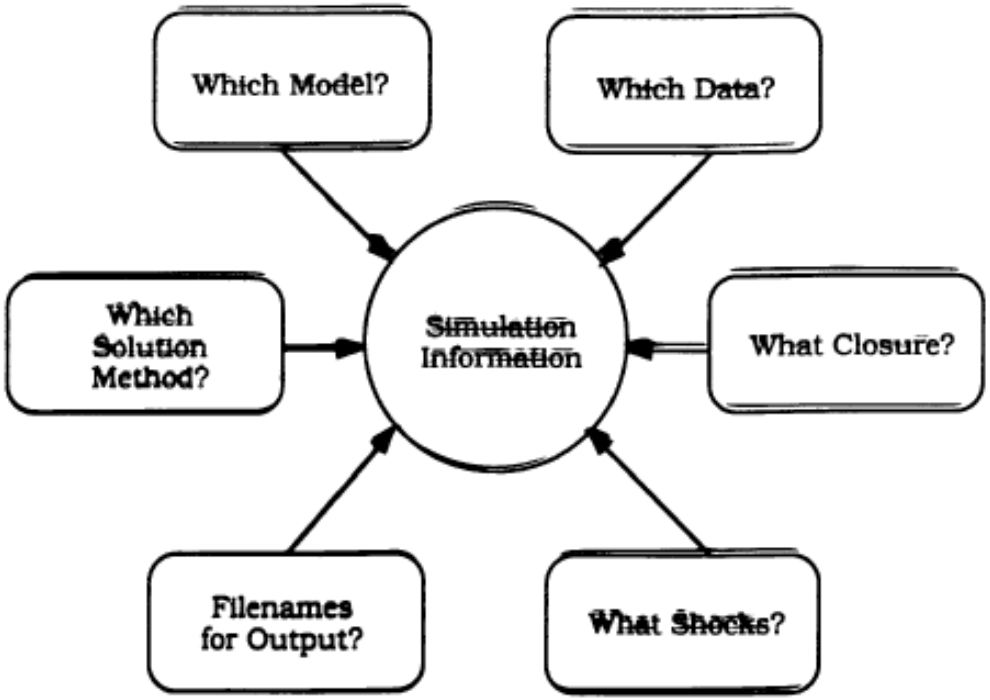
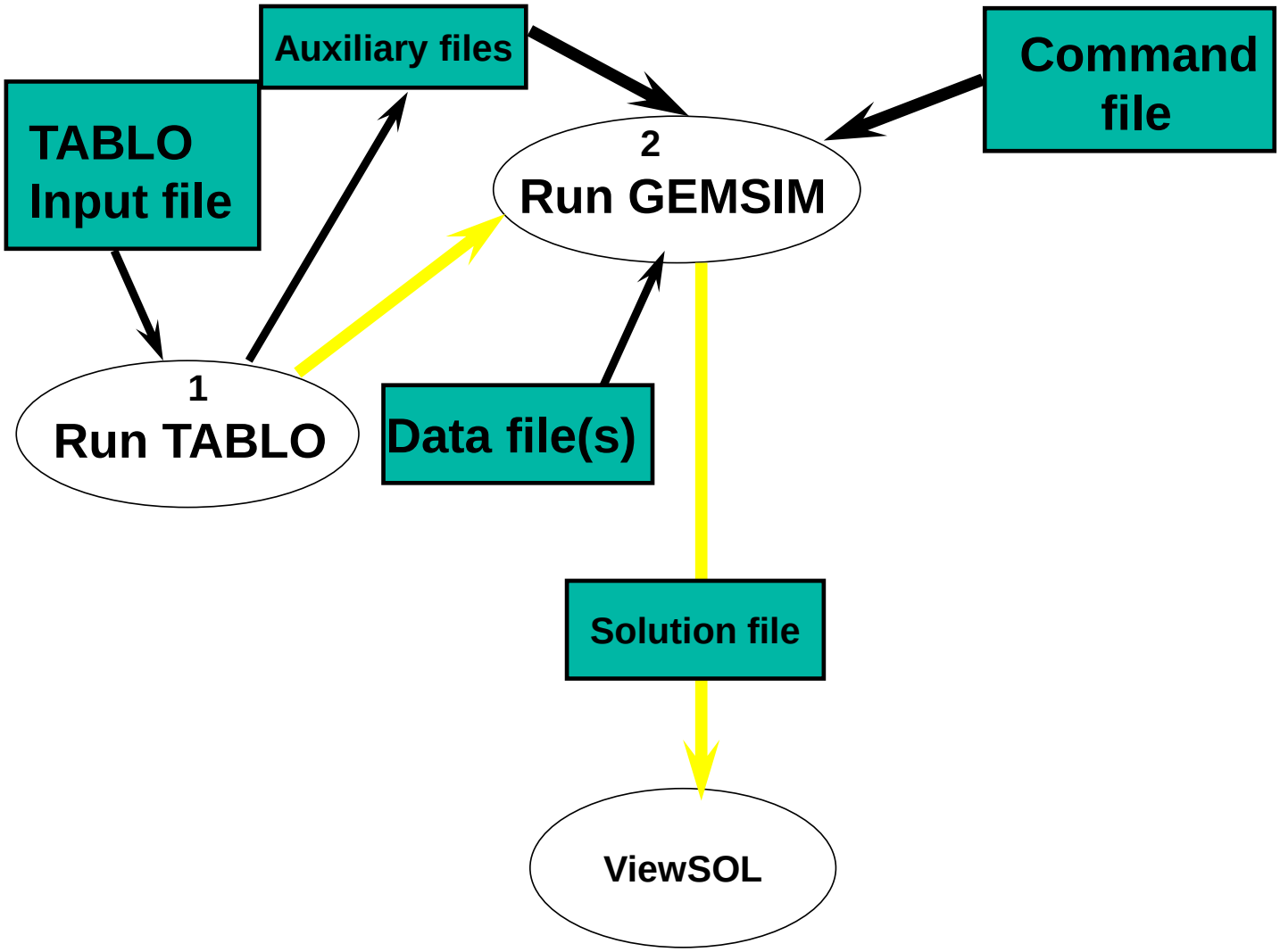
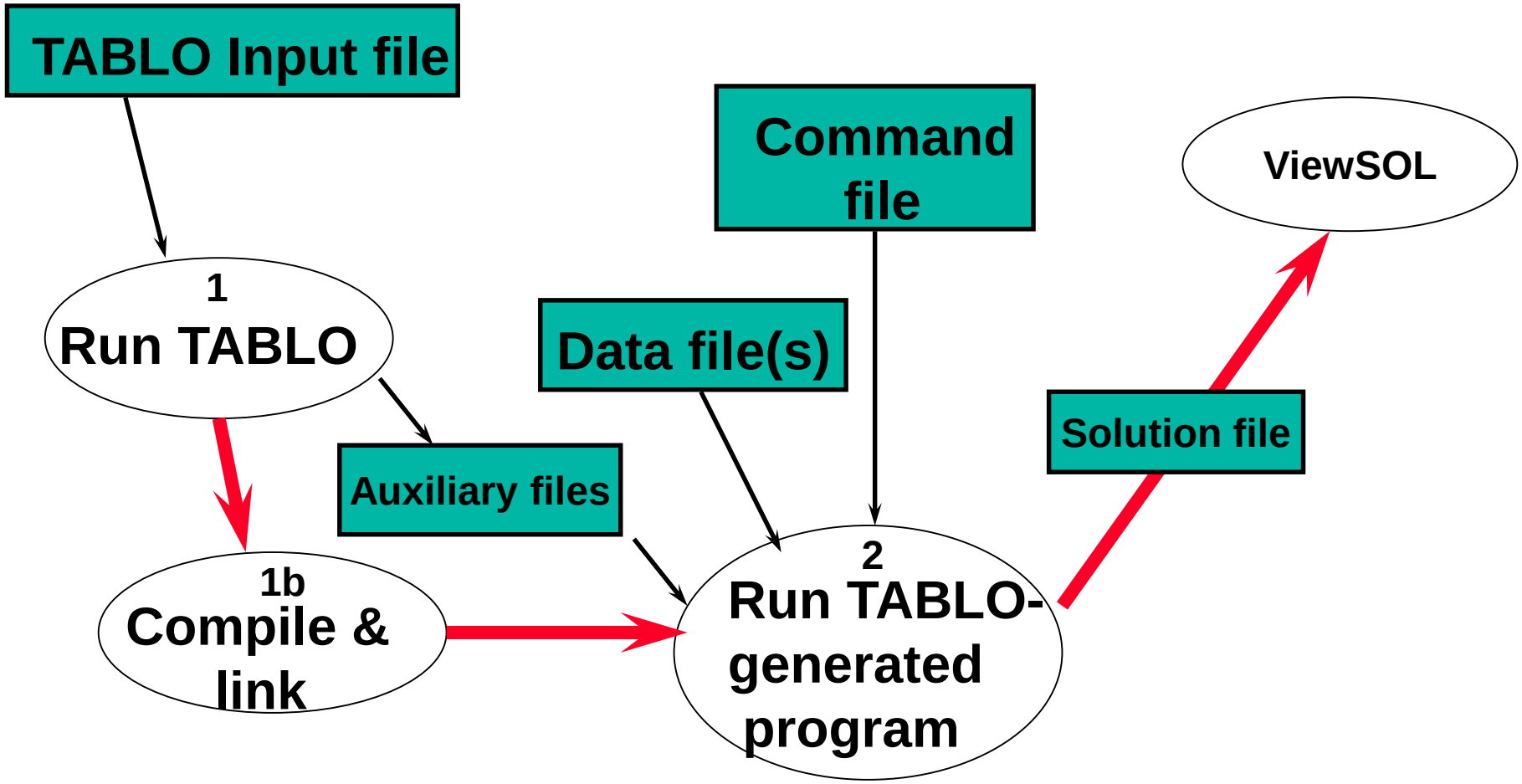


Figure 2.2.3: The Information Required to Specify a Simulation

The GEMSIM method



TABLO-generated program



The process

- There is a specification of the values of certain variables (“the exogenous ones” and the software calculates the values of the remaining variables (“the endogenous ones”).

- The new values of the exogenous variables are usually given by specifying the percentage changes (increases or decreases) from their values in the original solution given as part of the implementation.

Levels and percentage-change variables

- When the model is implemented, the equations may be linearized (that is, differentiated).
- The variables in these linearized equations are usually interpreted as percentage changes in the original variables.
- The original variables (prices, quantities, etc.) are referred as the **levels variables.**

Levels and percentage-change variables

- The (usually nonlinear) equations relating these levels variables are called **the levels equations.**

- **Levels equations:**

$$D = PQ$$

- The equation relates the dollar value, D, of a commodity to its price P (*$ per ton*) and its quantity Q (*tons*).

Levels and percentage-change variables

- **Linearized version:**

$$p\_D = p\_P + p\_Q$$

- The percentage change p_D in the dollar value is equal to the sum of the percentage changes p_P, in the price, and p_Q, in the quantity.

Data

- The data for a model often consists of input-output data (giving dollar values) and parameters (including elasticities).

- The data are usually sufficient to read off an initial solution to the levels equations (usually all basic prices are taken as 1 in the initial solution).

Outline

Introduction

How to carry out a simulation?

✓ How to implement the SJ model in GEMPACK?

Structure of the model

- Stylized Johansen described in Chapter 3 of Dixon et al (1992) and explained by Prof. Eduardo Haddad (Lecture 2).
- Single country
- Two sectors “s1” and “s2” producing a single commodity (1, 2)
- One household sector (0)
- Two primary factors (3, 4)

Levels and linearized equations of the Stylized Johansen Model

Levels Form*	Linearized Form	
consumer demands $X_{i0} = \alpha_{i0} Y / P_i$	$x_{i0} = y - p_i$	$i = 1,2$
intermediate demands $X_{ij} = \alpha_{ij} X_j \prod_{t=1}^{4} P_t^{\alpha_{tj}} \left[\prod_{t=1}^{4} (\alpha_{tj})^{-\alpha_{tj}} \right] / [A_j P_i]$	$x_{ij} = x_j - (p_i - \sum_{t=1}^{4} \alpha_{tj} p_t)$	$i=1, ..., 4$ $j=1,2$
price formation $P_j = \left[\prod_{t=1}^{4} (\alpha_{tj})^{-\alpha_{tj}} \right] \prod_{t=1}^{4} P_t^{\alpha_{tj}} / A_j$	$p_j = \sum_{t=1}^{4} \alpha_{tj} p_t$	$j = 1,2$
commodity market clearing $\sum_{j=0}^{2} X_{ij} = X_i$	$x_i = \sum_{j=0}^{2} [\frac{X_{ij}}{X_i}] x_{ij}$	$i = 1,2$
aggregate primary factor usage $\sum_{j=1}^{2} X_{ij} = X_i$	$x_i = \sum_{j=1}^{2} [\frac{X_{ij}}{X_i}] x_{ij}$	$i = 3,4$
Numeraire $P_1 = 1$	$p_1 = 0$	
Intermediate demands – dollar values $D_{ij} = P_i X_{ij}$	$d_{ij} = p_i + x_{ij}$	$i=1, ..., 4$ $j=1,2$
Consumer demands – dollar values $D_{i0} = P_i X_{i0}$	$d_{i0} = p_i + x_{i0}$	$i = 1,2$

Behavioral equations

Most are accounting identities
One is the numéraire equation

Levels variables for the Stylized Johansen

GEMPACK variable	Meaning	DPPW Notation
Y	Value of household income	Y
PC(i)	Price of commodity i	P_i (i=1,2)
PF(f)	Price of factor f	P_f (f=3,4)
XCOM(i)	Supply of commodity i	X_i (i=1,2)
XFAC(f)	Supply of factor f	X_f (f=3,4)
XH(i)	Household use of commodity i	X_{i0} (i=1,2)
XC(i,j)	Intermediate input of commodity i to industry j	X_{ij} (i,j=1,2)
XF(f,j)	Input of factor f to industry j	X_{fj} (f=3,4;j=1,2)
DVCOMIN(i,j)	Dollar values for intermediate inputs	(i,j=1,2)
DVFACIN(f,j)	Dollar values for factor use by industry	(f=3,4;j=1,2)
DVHOUS(i)	Dollar values for household consumption	(i=1,2)

Variables

- Most of variables have one or more arguments (indicating sectors and/or factors).
- They are **vector variables.**
- Variables which have no arguments (“Y” is the only one here) are referred to as **scalar or macro variables.**

Examples

- **PC(i)** is regarded as a vector variable with 2 components, one for each sector, namely PC("s1") and PC("s2").

- **XF(f,j)** is regarded as a vector variable with the following 4 components:

  - component 1 XF ("labor", "s1"): input of labor (factor 1) to sector 1.
  - component 2 XF ("capital", "s1"): input of capital (factor 2) to sector 1.
  - component 3 XF ("labor", "s2"): input of labor (factor 1) to sector 2
  - component 4 XF ("capital", "s2"): input of capital (factor 2) to sector 2.

Level *versus* percentage change

- Corresponding to each of these variables, there is an associated percentage change variable.
- TABLO adds the prefix "p_" to the name of the levels variable to indicate a percentage change.
- For example, p_XF is the percentage change in the levels variable XF.

Parameters for Stylized Johansen

Parameters		DPPW Notation
ALPHACOM(i,j)	Commodity exponents in production function for sector j (E3.1.4)	$ALPHA_{ij}$ (i,j=1,2)
ALPHAFAC(i,j)	Factor exponents in production function for sector j (E3.1.4)	$ALPHA_{fj}$ (f=3,4; j=1,2)

Stylized Johansen equations in GEMPACK notation

```
(E1) p_XH(i) = p_Y - p_PC(i)                                  i in SECT
                                                  [This is equation (E3.2.1) in DPPW]

(E2) p_XC(i,j) = p_XCOM(j) - [p_PC(i) - p_PC(j)]              i,j in SECT
                 [This is obtained from equation (E3.2.2) for i=1,2 in DPPW. The
                  term p_PC(j) is included because of equation (E3.2.3) in DPPW.]

(E3) p_XF(f,j) = p_XCOM(j) - [p_PF(f) - p_PC(j)]              f in FAC,j in SECT
                 [This is obtained from equation (E3.2.2) for i=3,4 in DPPW. The
                  term p_PC(j) is included because of equation (E3.2.3) in DPPW.]

(E4) p_PC(j) = SUM(i,SECT, ALPHACOM(i,j)*p_PC(i)) +
               SUM(f,FAC,  ALPHAFAC(f,j)*p_PF(f))              j in SECT
                                                  [This is equation (E3.2.3) in DPPW]

(E5) XCOM(i) = XH(i) + SUM(j,SECT, XC(i,j))                   i in SECT
                                                  [This is equation (E3.1.6) in DPPW]

(E6) XFAC(f) = SUM(j,SECT, XF(f,j))                           f in FAC
                                                  [This is equation (E3.1.7) in DPPW]

(E7) PC("s1") = 1
                                                  [This is equation (E3.1.23) in DPPW]

(E8) XC(i,j) = DVCOMIN(i,j) / PC(i)                          i,j in SECT
                                                  [This equation is not numbered in DPPW]

(E9) XH(i)   = DVHOUS(i) / PC(i)                             i in SECT
                                                  [This equation is not numbered in DPPW]

(E10) XF(f,j) = DVFACIN(f,j) / PF(f)                         f in FAC, j in SECT
                                                  [This equation is not numbered in DPPW]
```

SECT (1,2)

FAC (3,4)

Data requirements for the SJ model

		Industry		Households	Total Sales
		1	2		
Commodity Sectors	1	4.0	2.0	2.0	8.0
Commodity	2	2.0	6.0	4.0	12.0
Labor Factors	3	1.0	3.0		4.0
Capital	4	1.0	1.0		2.0
Total Production		8.0	12.0	6.0	

TABmate

TABmate is a text editor

TABmate colors items in the TAB file according to their function:

black for words that are part of the TABLO language;
green for variables and other items that are specific to this model;
blue for comments which GEMPACK ignores;
italics for strings that GEMPACK uses as labels or descriptions.

You might also see line numbers in the left hand margin.

SJ.tab – Example 1 (E9)

```
(E1) p_XH(i) = p_Y - p_PC(i)                          i in SECT
                                                  [This is equation (E3.2.1) in DPPW]

(E2) p_XC(i,j) = p_XCOM(j) - [p_PC(i) - p_PC(j)]    i,j in SECT
                          [This is obtained from equation (E3.2.2) for i=1,2 in DPPW. The
                           term p_PC(j) is included because of equation (E3.2.3) in DPPW.]

(E3) p_XF(f,j) = p_XCOM(j) - [p_PF(f) - p_PC(j)]    f in FAC,j in SECT
                          [This is obtained from equation (E3.2.2) for i=3,4 in DPPW. The
                            term p_PC(j) is included because of equation (E3.2.3) in DPPW.]

(E4) p_PC(j) = SUM(i,SECT, ALPHACOM(i,j)*p_PC(i)) +
               SUM(f,FAC,  ALPHAFAC(f,j)*p_PF(f))          j in SECT
                                                  [This is equation (E3.2.3) in DPPW]

(E5) XCOM(i) = XH(i) + SUM(j,SECT, XC(i,j))          i in SECT
                                                  [This is equation (E3.1.6) in DPPW]

(E6) XFAC(f) = SUM(j,SECT, XF(f,j))                  f in FAC
                                                  [This is equation (E3.1.7) in DPPW]

(E7) PC("s1") = 1
                                                  [This is equation (E3.1.23) in DPPW]

(E8) XC(i,j) = DVCOMIN(i,j) / PC(i)                  i,j in SECT
                                              [This equation is not numbered in DPPW]

(E9) XH(i)   = DVHOUS(i) / PC(i)                     i in SECT
                                              [This equation is not numbered in DPPW]

(E10) XF(f,j) = DVFACIN(f,j) / PF(f)                 f in FAC, j in SECT
                                              [This equation is not numbered in DPPW]
```

SJ.tab – Example 1 (E9)

```
EQUATION House
# Household demand for commodity i #
(all,i,SECT) XH(i) = DVHOUS(i) / PC(i) ;
```

SJ.tab – Example 1 (E9)

```
Variable (GE 0) (all,i,SECT)  XH(i)   # Household
demand for commodity i #

Variable (GE 0) (all,i,SECT)  DVHOUS(i)
# Dollar value of household use of commodity i # ;

Variable (GE 0) (all,i,SECT)  PC(i)
# Price of commodity i #
```

SJ.tab – Example 1 (E9)

```
VARIABLE (DEFAULT = LEVELS) ;
EQUATION (DEFAULT = LEVELS) ;
COEFFICIENT (DEFAULT = PARAMETER) ;
FORMULA (DEFAULT = INITIAL) ;

SET SECT # Sectors # (s1-s2) ;

READ DVHOUS  from FILE iodata HEADER "HCON" ;
FORMULA (all,i,SECT) PC(i) = 1.0 ;
FORMULA House # Household demand for commodity i #
(all,i,SECT) XH(i) = DVHOUS(i) / PC(i) ;
```

SJ.tab – Example 1 (E9)

```
FORMULA & EQUATION House
# Household demand for commodity i #
(all,i,SECT) XH(i) = DVHOUS(i) / PC(i) ;

FILE iodata # input-output data for the model # ;
```

SJ.tab – Example 2 (E4)

```
(E1) p_XH(i) = p_Y - p_PC(i)                          i in SECT
                                                  [This is equation (E3.2.1) in DPPW]

(E2) p_XC(i,j) = p_XCOM(j) - [p_PC(i) - p_PC(j)]    i,j in SECT
                          [This is obtained from equation (E3.2.2) for i=1,2 in DPPW. The
                           term p_PC(j) is included because of equation (E3.2.3) in DPPW.]

(E3) p_XF(f,j) = p_XCOM(j) - [p_PF(f) - p_PC(j)]    f in FAC,j in SECT
                          [This is obtained from equation (E3.2.2) for i=3,4 in DPPW. The
                            term p_PC(j) is included because of equation (E3.2.3) in DPPW.]

(E4) p_PC(j) = SUM(i,SECT, ALPHACOM(i,j)*p_PC(i)) +
               SUM(f,FAC,  ALPHAFAC(f,j)*p_PF(f))          j in SECT
                                                  [This is equation (E3.2.3) in DPPW]

(E5) XCOM(i) = XH(i) + SUM(j,SECT, XC(i,j))          i in SECT
                                                  [This is equation (E3.1.6) in DPPW]

(E6) XFAC(f) = SUM(j,SECT, XF(f,j))                  f in FAC
                                                  [This is equation (E3.1.7) in DPPW]

(E7) PC("s1") = 1
                                                  [This is equation (E3.1.23) in DPPW]

(E8) XC(i,j) = DVCOMIN(i,j) / PC(i)                  i,j in SECT
                                                  [This equation is not numbered in DPPW]

(E9) XH(i)   = DVHOUS(i) / PC(i)                     i in SECT
                                                  [This equation is not numbered in DPPW]

(E10) XF(f,j) = DVFACIN(f,j) / PF(f)                 f in FAC, j in SECT
                                                  [This equation is not numbered in DPPW]
```

SJ.tab – Example 2 (E4)

```
EQUATION(LINEAR) Price_formation
# Unit cost index for industry j #
(all,j,SECT) p_PC(j) =
SUM(i,SECT,ALPHACOM(i,j)*p_PC(i)) +
SUM(f,FAC,ALPHAFAC(f,j)*p_PF(f));
```

SJ.tab – Example 2 (E4)

```
Variable (GE 0) (all,i,SECT)  PC(i)
# Price of commodity i #;

Variable (GE 0) (all,f,FAC)   PF(f)
# Price of factor f #;

COEFFICIENT (all,i,SECT)(all,j,SECT)  ALPHACOM(i,j)
 # Share of intermediate use of commodity i in costs of
industry j # ;

COEFFICIENT (all,f,FAC)(all,j,SECT)   ALPHAFAC(f,j)
 # Share of factor input f in costs of industry j # ;
```

SJ.tab – Example 2 (E4)

```
FORMULA (all,i,SECT)(all,j,SECT) ALPHACOM(i,j) =
DVCOMIN(i,j) / [SUM(ii,SECT,DVCOMIN(ii,j)) + SUM
(ff,FAC,DVFACIN(ff,j))] ;

FORMULA (all,f,FAC)(all,j,SECT) ALPHAFAC(f,j) =
DVFACIN(f,j) / [SUM(ii,SECT,DVCOMIN(ii,j)) + SUM
(ff,FAC,DVFACIN(ff,j))] ;
```

Golden rule: each entity (VARIABLE, COEFFICIENT, etc.) must be declared on the TABLO Input file before it is used in EQUATIONs and FORMULAs!

Main statements in TABLO

EQUATION

FORMULA

READ

VARIABLE

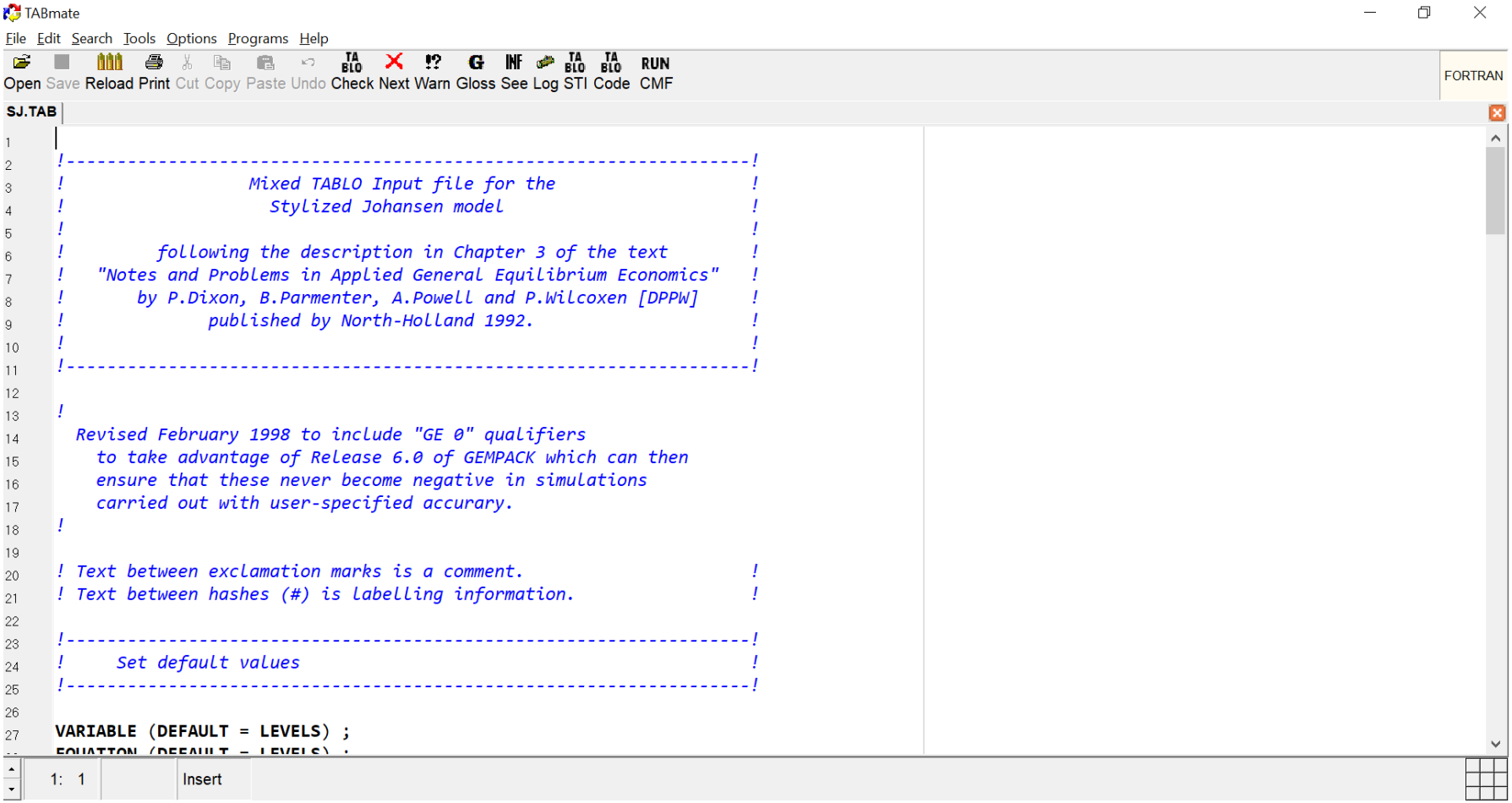
COEFFICIENT

SET

FILE

Chapters 3 and 4 of GEMPACK Document GPD-2!

SJ.tab

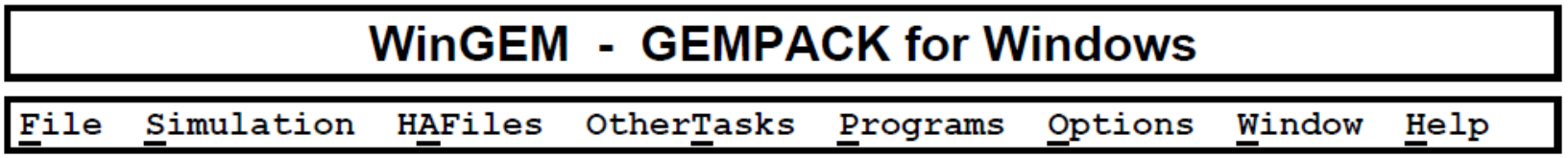

TABmate
File Edit Search Tools Options Programs Help
Open Save Reload Print Cut Copy Paste Undo Check Next Warn Gloss See Log STI Code CMF
FORTRAN
SJ.TAB
!----------!
! Mixed TABLO Input file for the !
! Stylized Johansen model !
! !
! following the description in Chapter 3 of the text !
! "Notes and Problems in Applied General Equilibrium Economics" !
! by P.Dixon, B.Parmenter, A.Powell and P.Wilcoxen [DPPW] !
! published by North-Holland 1992. !
! !
!----------!
!
Revised February 1998 to include "GE 0" qualifiers
to take advantage of Release 6.0 of GEMPACK which can then
ensure that these never become negative in simulations
carried out with user-specified accurary.
!
! Text between exclamation marks is a comment. !
! Text between hashes (#) is labelling information. !
!----------!
! Set default values !
!----------!
VARIABLE (DEFAULT = LEVELS) ;
1: 1
Insert

Examples

1. *Starting WinGEM*
2. *Preparing a directory for Model SJ*
3. *Setting the working directory*
4. *Editing text files in WinGEM*
5. *Looking at the data directly (VIEWHAR)*
6. *Simulations with the SJ model*

1. Starting WinGEM

- In Windows, double click on the WinGEM icon to start GEMPACK.
- This should give the main WinGEM menu



2. Preparing a directory for Model SJ

- To keep all example files for the SJ model together in one area, we should create a separate directory \SJ for these files and how to copy the relevant files into this directory.

3. Setting the working directory

Choose a working directory (for SJ model the working directory needs to be the directory \SJ you have just created)

To set this, first click on **File** in the main WinGEM menu

In the drop-down menu, click on the menu item

Change both default directories…

So the sequence of clicks (first File then Change both default directories) is

File|Change both default directories…

5. Looking at the data directly (VIEWHAR)

The input-output data used in SJ model are contained in the data file **SJ.DAT**

This is a special GEMPACK binary file **– called Header Array file**

Thus to look at SJ.DAT you have to use a special program to read Header Array files, called **ViewHAR**

HA Files*| *View VIEWHAR

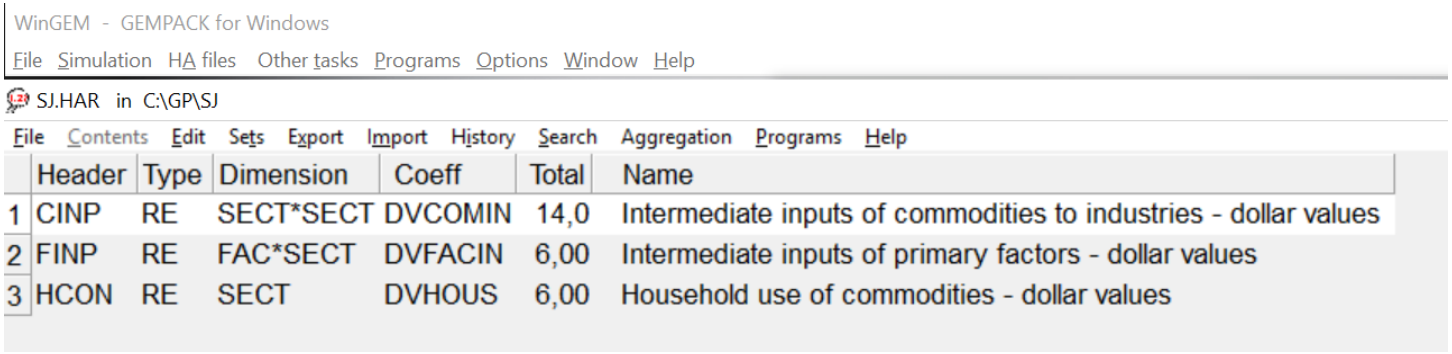
The viewHAR window will appear

Click on **File|Open…** and selected the file SJ.DAT

This will open the file SJ.DAT and show the contents on the screen

5. Looking at the data directly (VIEWHAR)

Each of the rows corresponds to a different array of data on the file. Look at the column under the heading Name to see what data are in these arrays.



	Header	Type	Dimension	Coeff	Total	Name
1	CINP	RE	SECT*SECT	DVCOMIN	14,0	Intermediate inputs of commodities to industries - dollar values
2	FINP	RE	FAC*SECT	DVFACIN	6,00	Intermediate inputs of primary factors - dollar values
3	HCON	RE	SECT	DVHOUS	6,00	Household use of commodities - dollar values

5. Looking at the data directly (VIEWHAR)

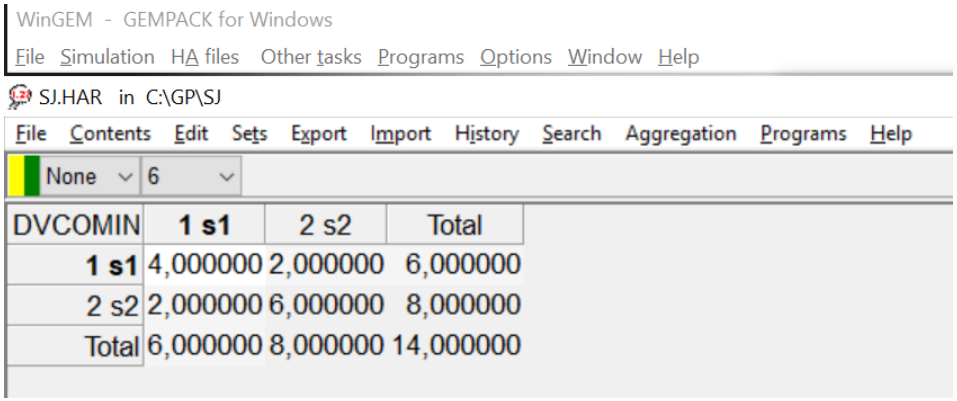
The first array is the "Intermediate inputs of commodities to industries – dollar values"

The Header CINP is just a label for this array (*headers can have up to 4 characters*).

The array is of Type RE. The R means this is an array of real numbers. The E means that this array has set and element labeling.

Double click on CINP to see the numbers in this array.

5. Looking at the data directly (VIEWHAR)



DVCOMIN	1 s1	2 s2	Total
1 s1	4,000000	2,000000	6,000000
2 s2	2,000000	6,000000	8,000000
Total	6,000000	8,000000	14,000000

Compare this numbers with the Input-Output data for SJ model.

To return to Contents Screen, click on **Contents** in the ViewHAR menu.

6. Simulations with the SJ model

- We have to choose a closure:

  - Supply of the two factors, labor and capital, are the exogenous variables

- Thus we will specify the percentage changes in the variable XFAC, namely p_XFAC, and solve the model to find the percentage changes in all the other variables.

6. Simulations with the SJ model

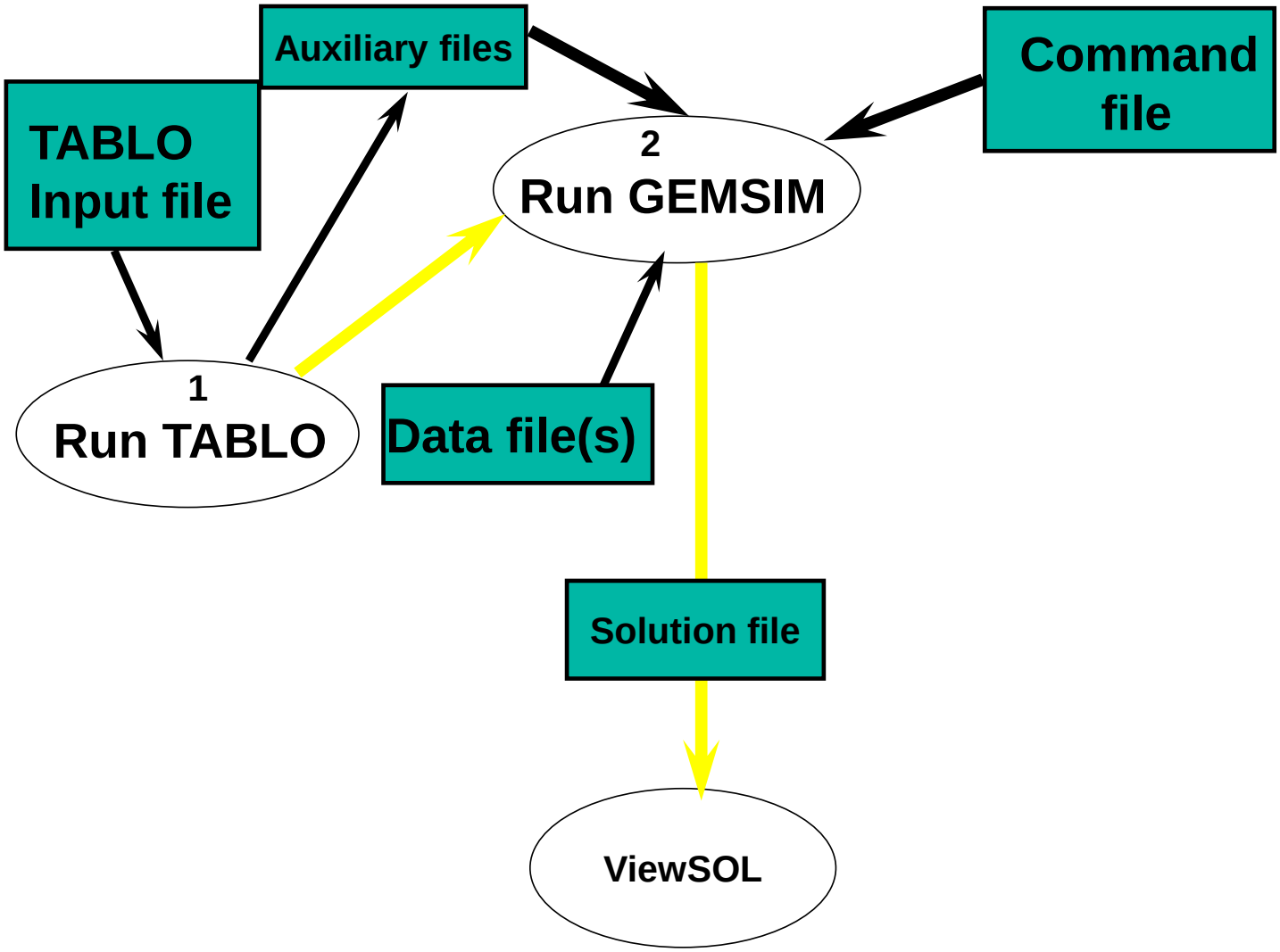
- For this simulation, we increase the supply of labor by 10 per cent and hold the supply of capital fixed

- The starting point for any simulations with Stylized Johansen model are:

  - the TABLO Input file (called SJ. TAB) and

  - the data file (called SJ.DAT)

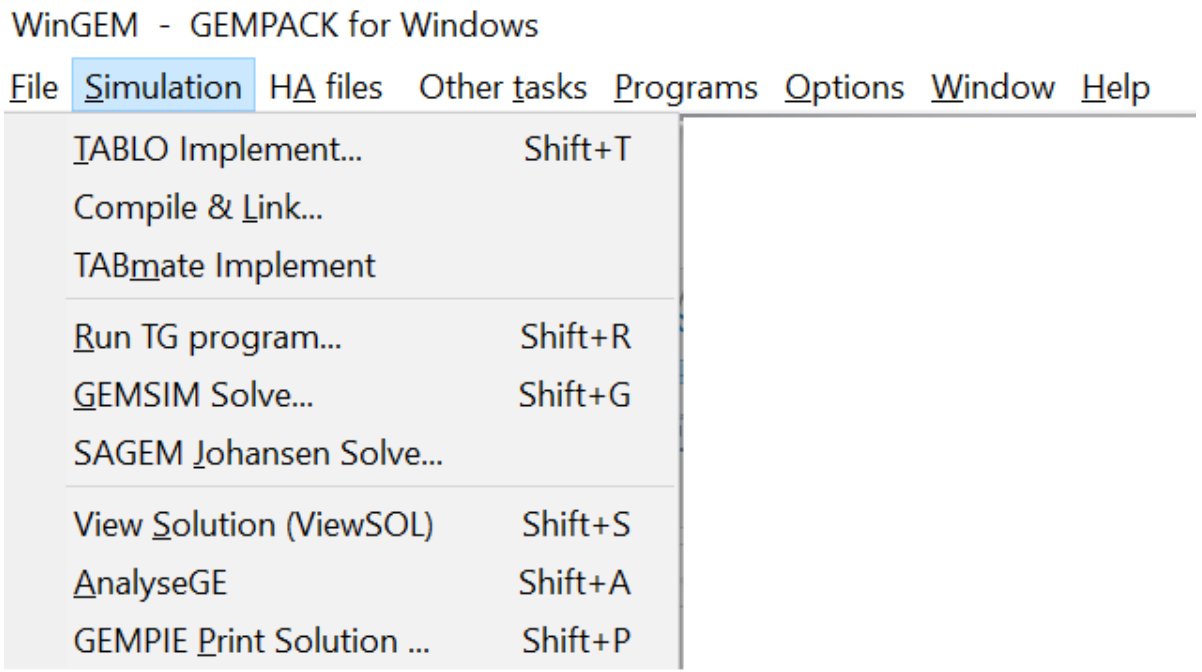
The GEMSIM method



6. Simulation using GEMSIM

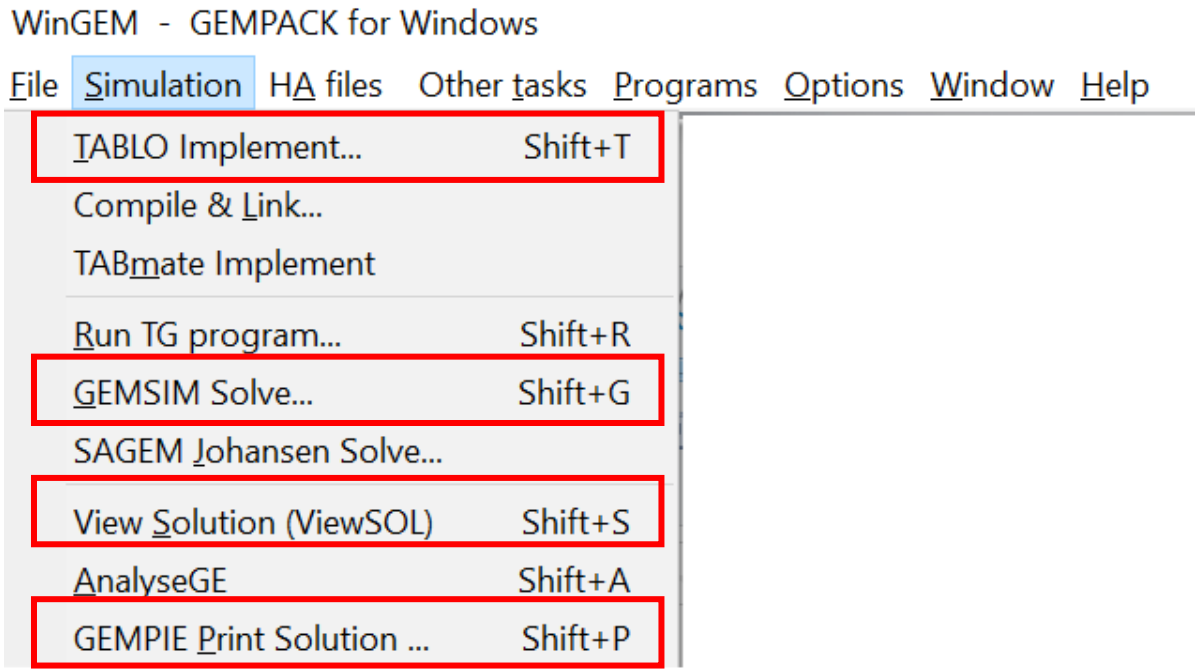
From the WinGEM menu at the top of the screen choose **Simulation**.

In the drop-down menu the choices are:



6. Simulation using GEMSIM

The items from this menu you will be using in this simulation are:



6. Simulation using GEMSIM

There are three steps involved in carrying out a simulation using GEMPACK:

STEP 1 – Implement the model (TABLO)

STEP 2 – Solve the equations of the model (GEMSIM)

STEP 3 – View the results (GEMPIE and VIEWSOL)

WinGEM will guide you through these steps and indicate what to do next.

6. Simulation using GEMSIM

Step 1:

Simulation / TABLO Implement…

TABLO Options… (PGS)

Go to GEMSIM

Step 2:

File *SJLB.CMF*

6. Simulation using GEMSIM

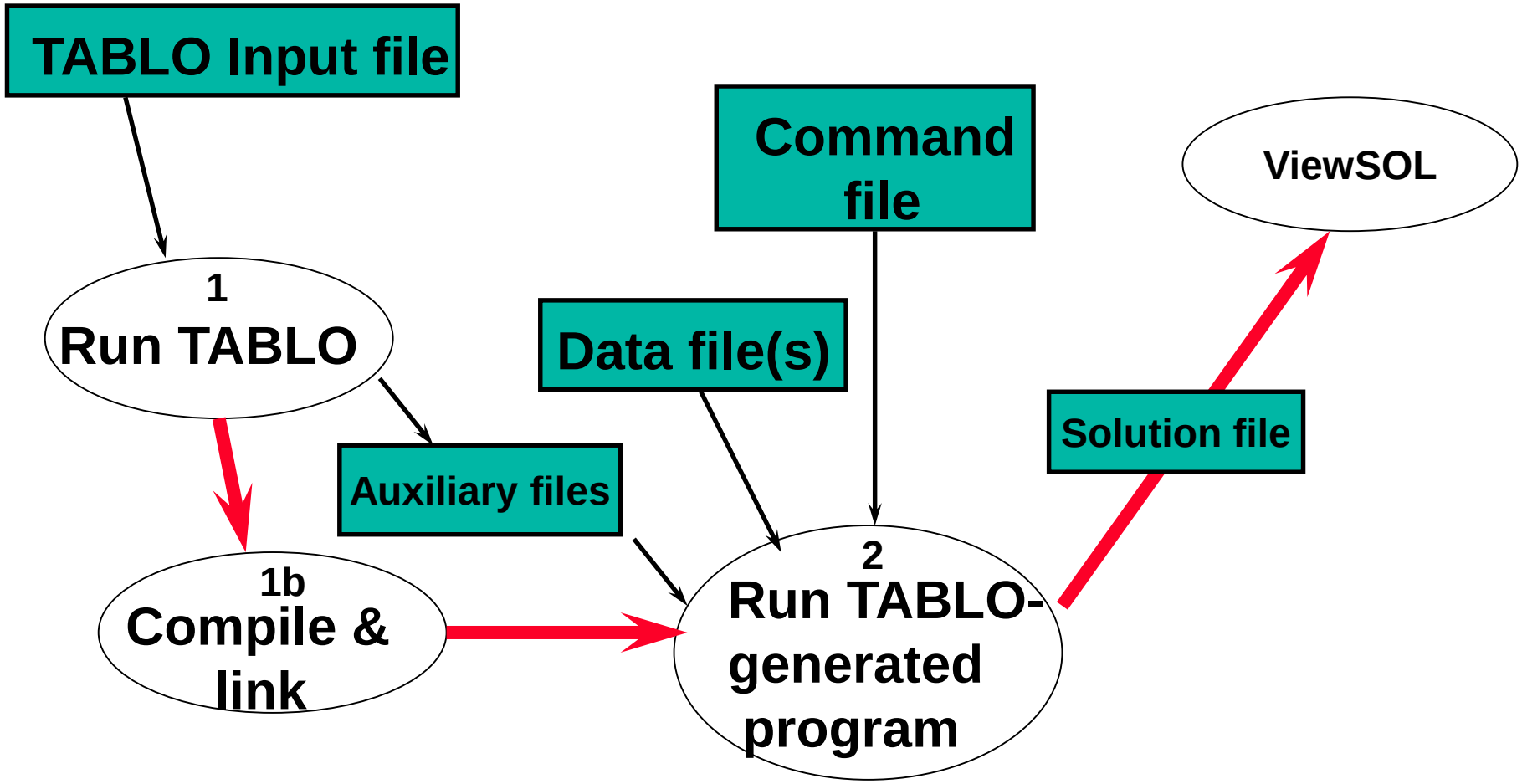
Step 3:

GEMPIE *versus* VIEWSOL

The updated data – another result of the simulation

File *SJLB.UPD*

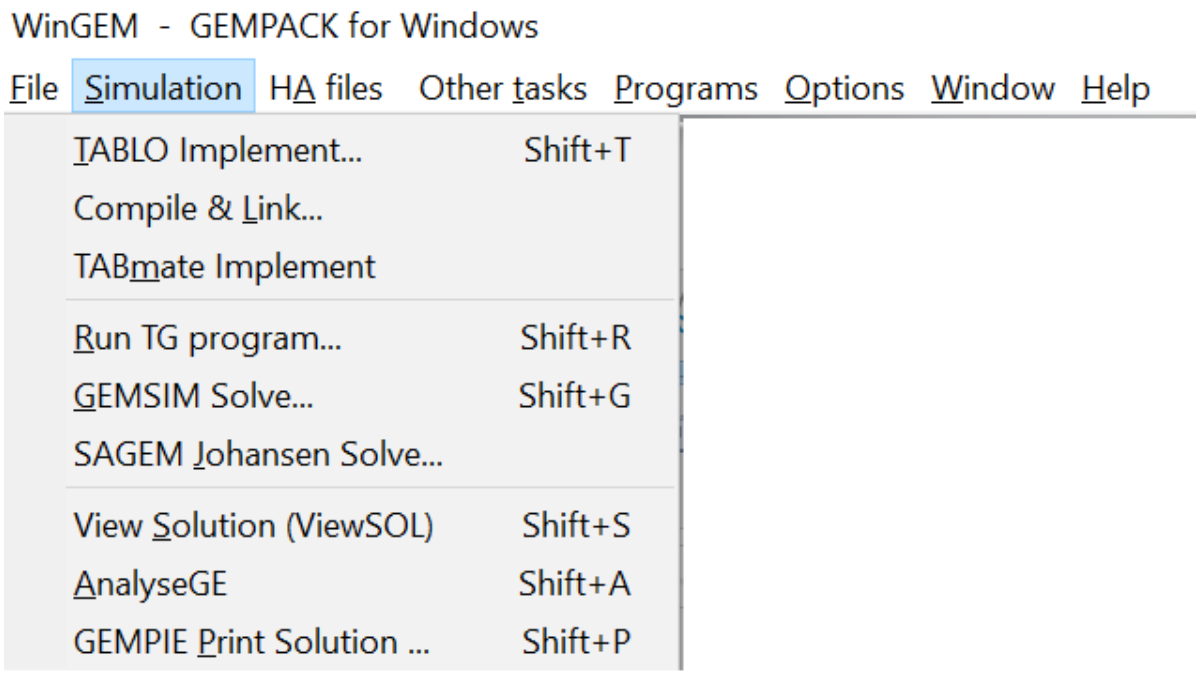
TABLO-generated program



6. TABLO-generated program

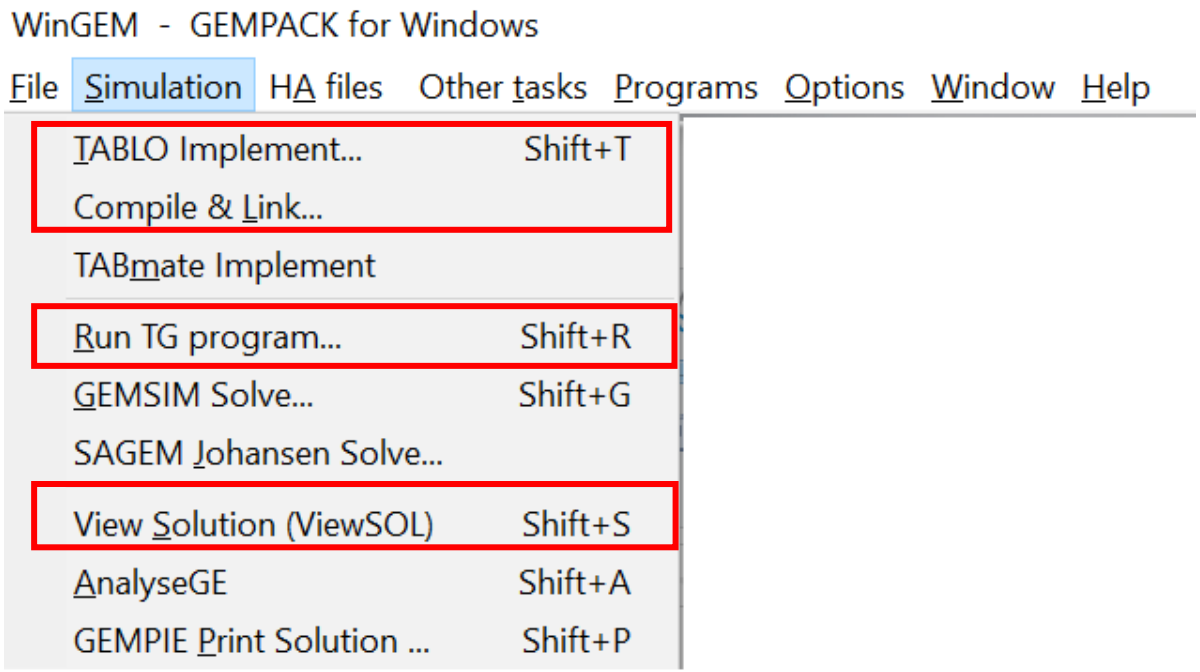
From the WinGEM menu at the top of the screen choose **Simulation**.

In the drop-down menu the choices are:



6. TABLO-generated program

The items from this menu you will be using in this simulation are:



6. TABLO-generated program

In the TABLO-generated program method, the GEMPACK program TABLO is used to convert the algebraic equations of the economic model into a Fortran program specific to your model.

This Fortran program (which is referred to as the TABLO-generated program or TG Program in the menu) is compiled and linked to a library of GEMPACK subroutines.

The executable image of the TABLO-generated program produced by the compiler is used to run simulations on the model.

6. TABLO-generated program

There are three steps involved in carrying out a simulation using GEMPACK:

STEP 1 – Implement the model

STEP 2 – Solve the equations of the model

STEP 3 – View the results

WinGEM and **RunGEM** will guide you through these steps and indicate what to do next.

6. TABLO-generated program

STEP 1 – Implement the model SJ using TABLO

Step 1 (a) – Run TABLO to create the TABLO-generated program

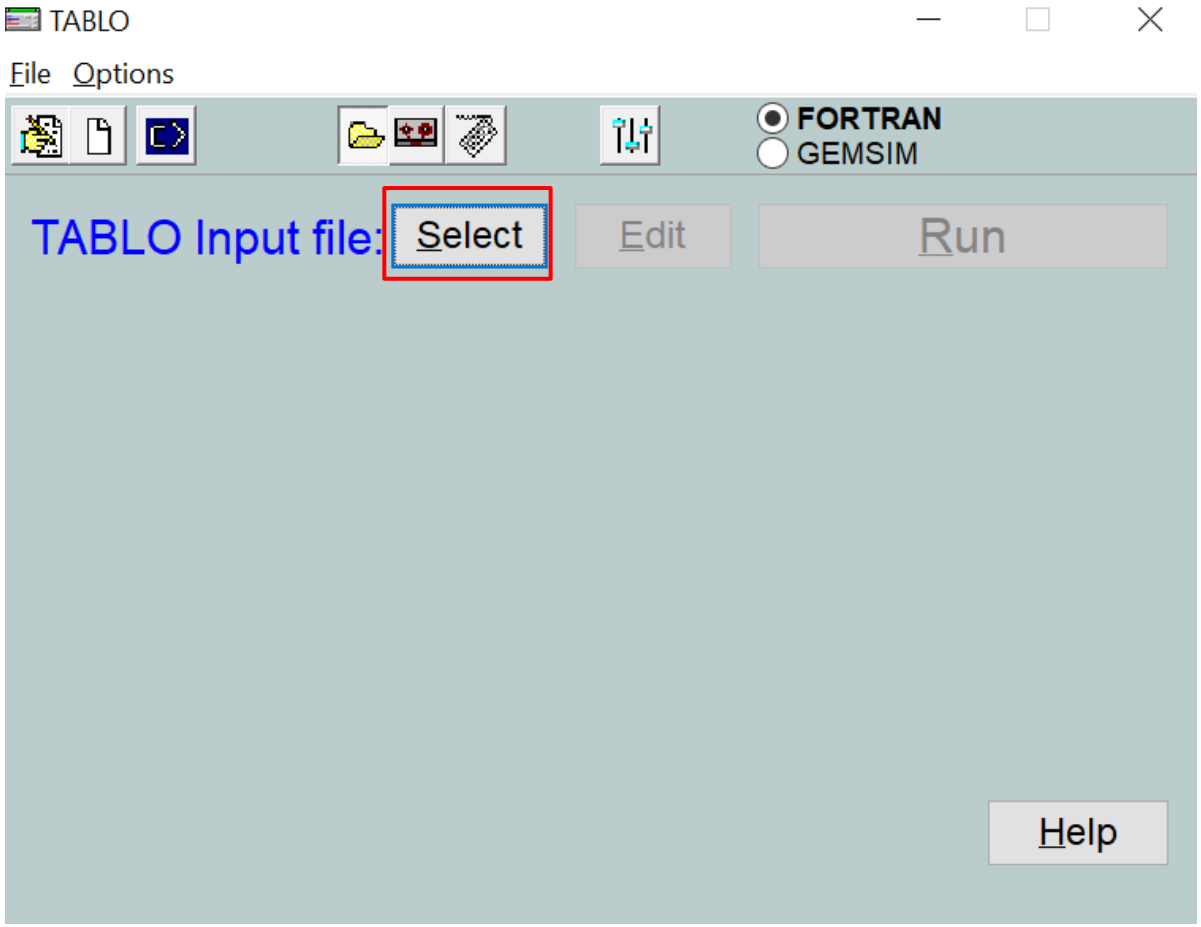
The TABLO Input file is called ***SJ.TAB***. It contains the theory of the SJ model.

Choose:

Simulation|TABLO Implement....

6. TABLO-generated program

A window for TABLO will appear:



6. TABLO-generated program

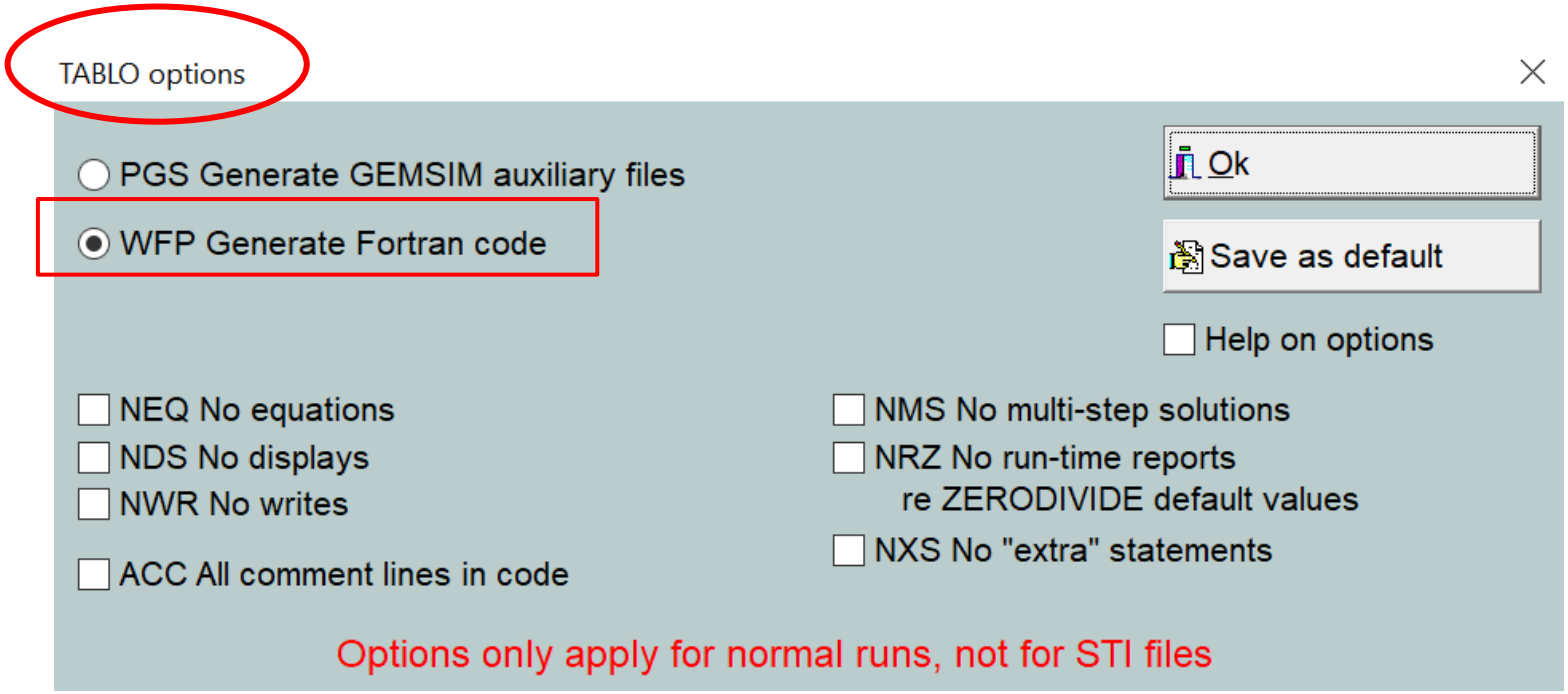
Click on the ***Select*** button to select the name of the TABLO Input file SJ.TAB. This is all TABLO need to implement a model.

In the menu for the TABLO window, select ***Options*** menu item. Then in this menu choose

TABLO Options.....

A new TABLO Options window will appear…

6. TABLO-generated program



6. TABLO-generated program

We will choose the option ***WFP*** because we want you to create the TABLO-generated program.

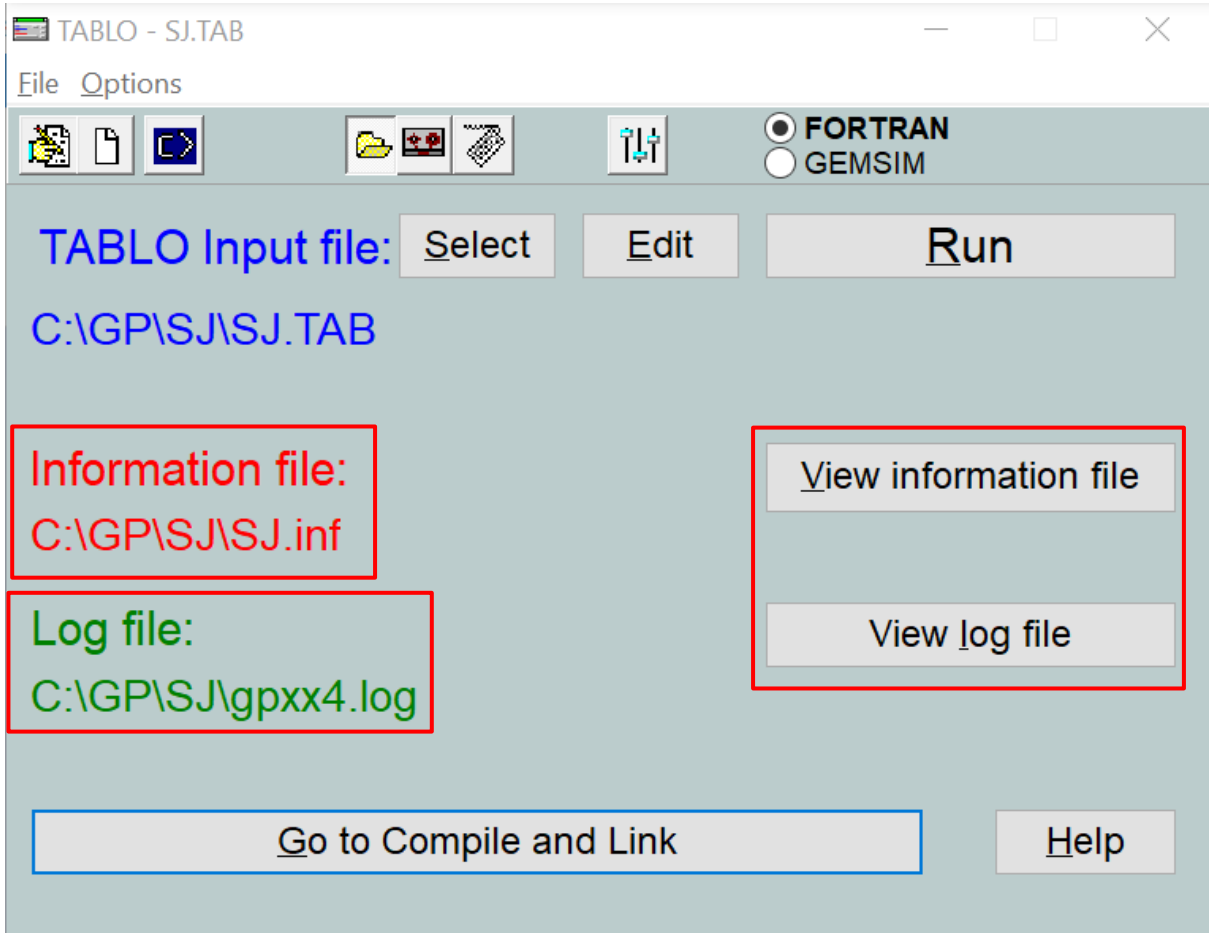
Then click on ***Ok*** button to return to the TABLO window.

Click on ***Run*** button

The program runs TABLO in a DOS box and when complete, returns to the TABLO window with the names of files it has created: the information file SJ.INF and Log file.

To look at files click on ***View*** buttons beside them.

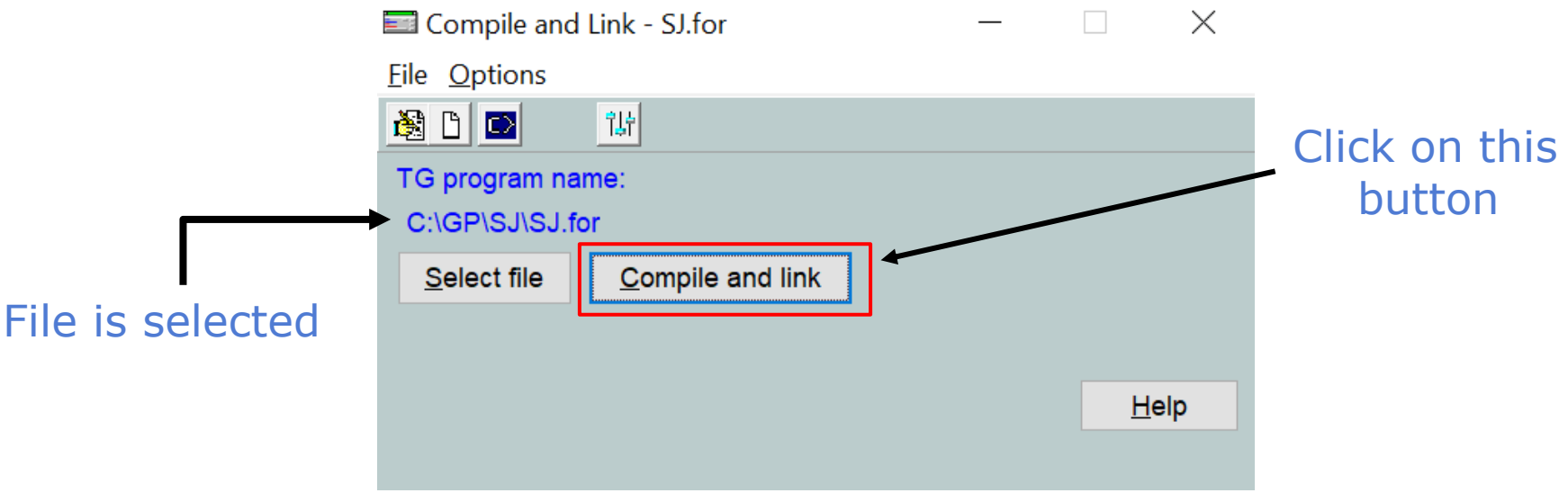
6. TABLO-generated program



6. TABLO-generated program

Step 1 (b) Compile and Link the TABLO-generated Program

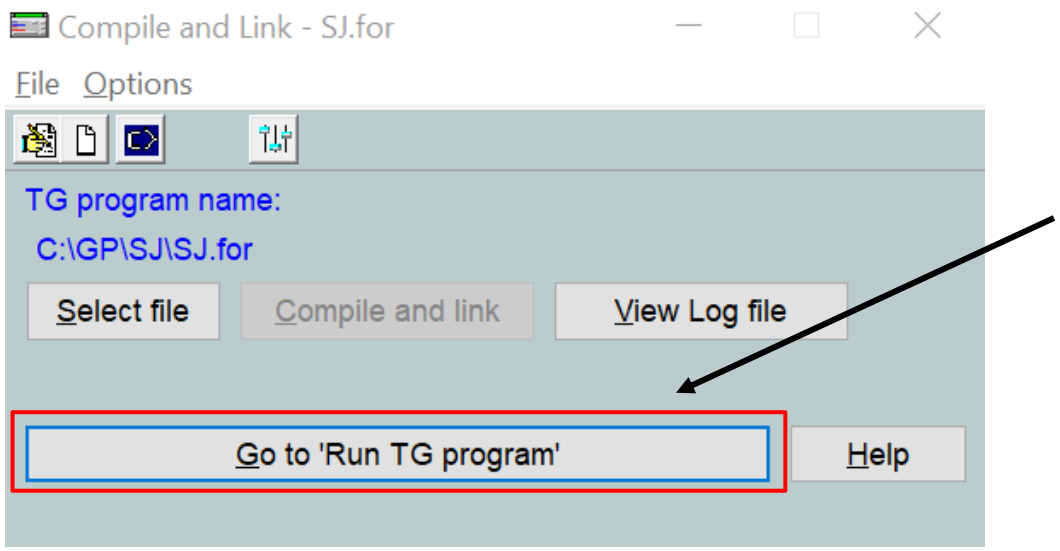
Click on the ***Go to compile and Link*** button at the button of the TABLO window to run the Fortran compiler.



6. TABLO-generated program

The compiler converts the Fortran file SJ.FOR into the executable image SJ.EXE.

When finished click on the button Go to ***'Run TG program' or go to RunGEM module***



Click on this button or go to RunGEM module

6. TABLO-generated program

STEP 2 – Solve the equations of the model

Starting RunGEM

Double click on RunGEM icon to start RunGEM module. This is the first part of RunGEM

6. TABLO-generated program

To change from Title to Model/Data for exemple is only necessary to click on the names.

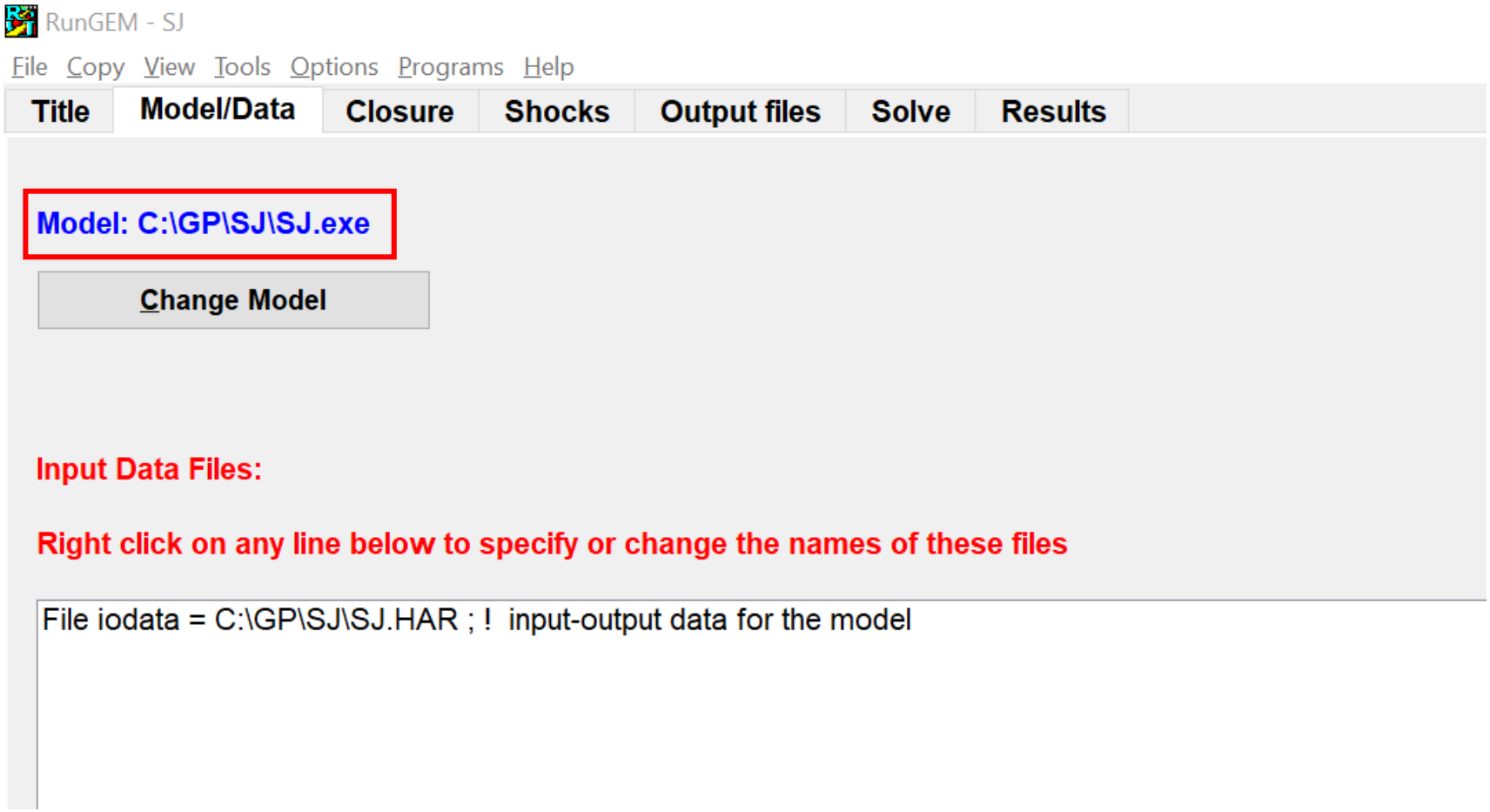
Select the Model:

Go to ***Model/Data*** page by clicking on its tab.

Click on button ***Change Model*** do select your model.

Select the file SJ.EXE.

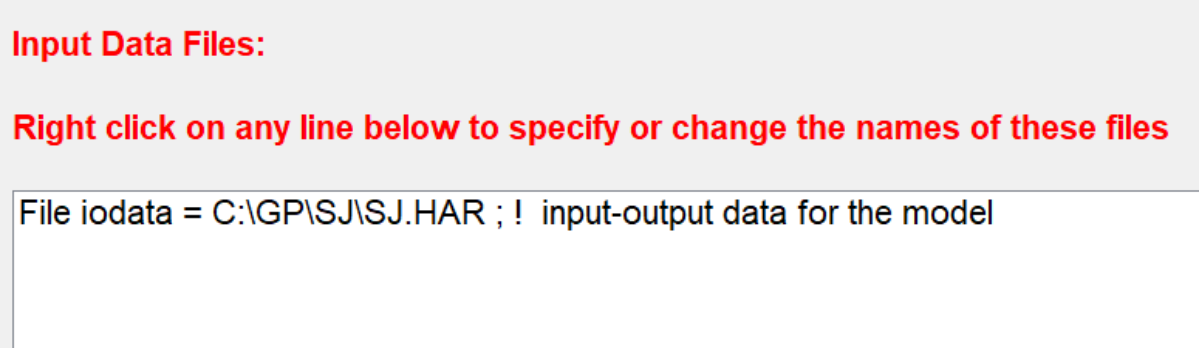
6. TABLO-generated program



6. TABLO-generated program

Select the Data File:

In the white box headed ***Input Data File*** a single line of text will appear.

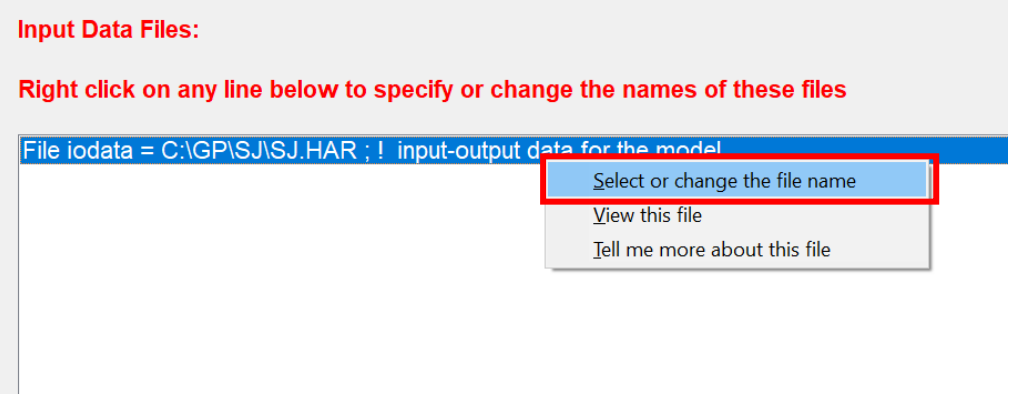


6. TABLO-generated program

Select the Data File:

Select the line (in blue) by right clicking on it

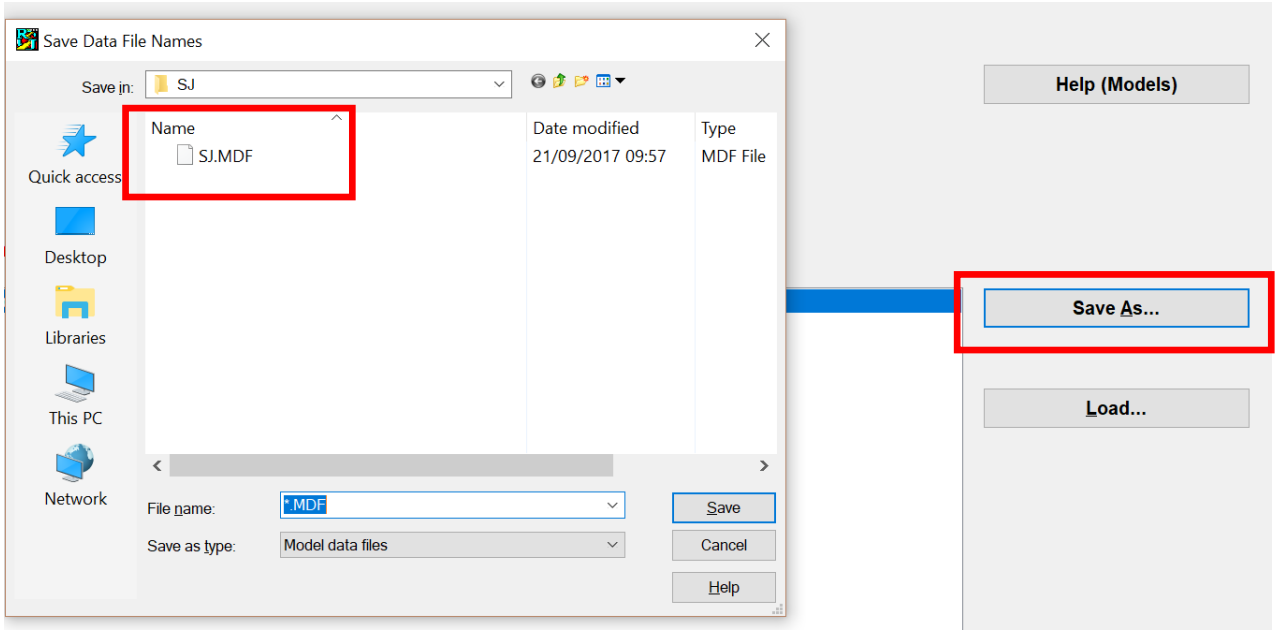
Select on the menu the option ***Select or change the file name***



6. TABLO-generated program

Select the Data File:

Click on the ***save as*** button



6. TABLO-generated program

Load Closure:

Select the ***Closure*** link

Select the link ***Load Closure*** (as you have a closured saved in the SJ directory the GEMPACK will automatic open it).

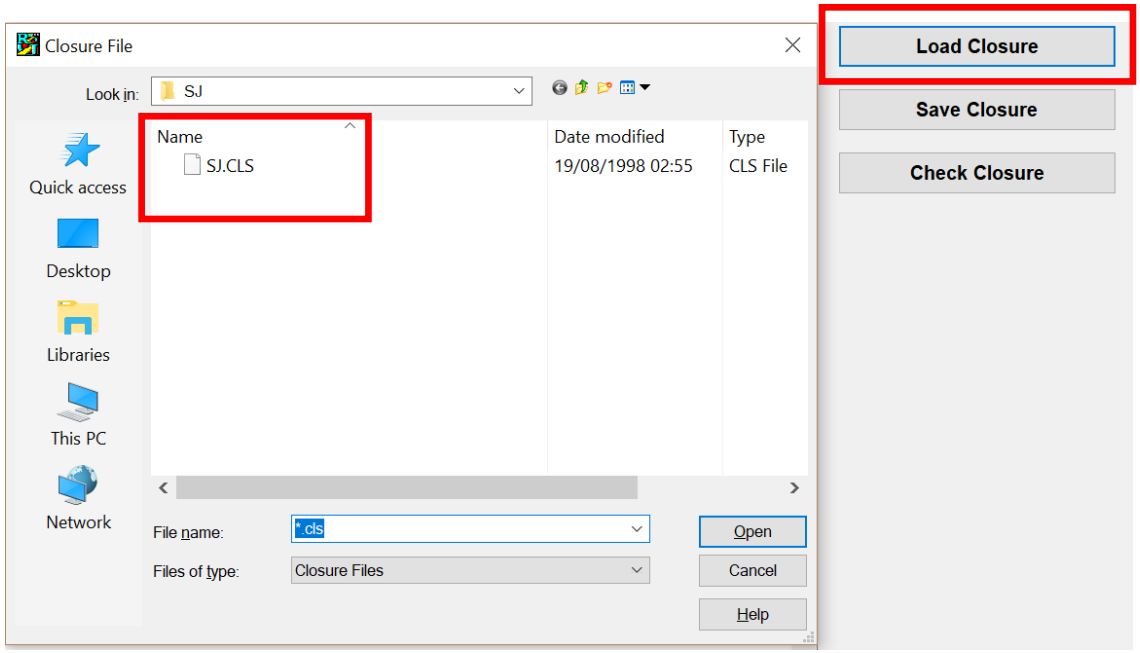
If you do not have the file .CLS you can type the closure and save it.

exogenous p_FAC;

rest endogenous;

6. TABLO-generated program

Load Closure:



6. TABLO-generated program

Load Closure:

You also have the opportunity to check the closure.

Thus click on the button ***Check Closure*** and RunGEM will check if this is a valid closure for the SJ model

Select the Shocks:

Click on ***Shocks*** tab, and in the box labeled ***Variable to shock***, click on the small arrow on the right hand side to get a drop-down list of exogenous variables.

6. TABLO-generated program

Select the Shocks:

For the closure chosen in the SJ model we have only one exogenous variable (p_XFAC).

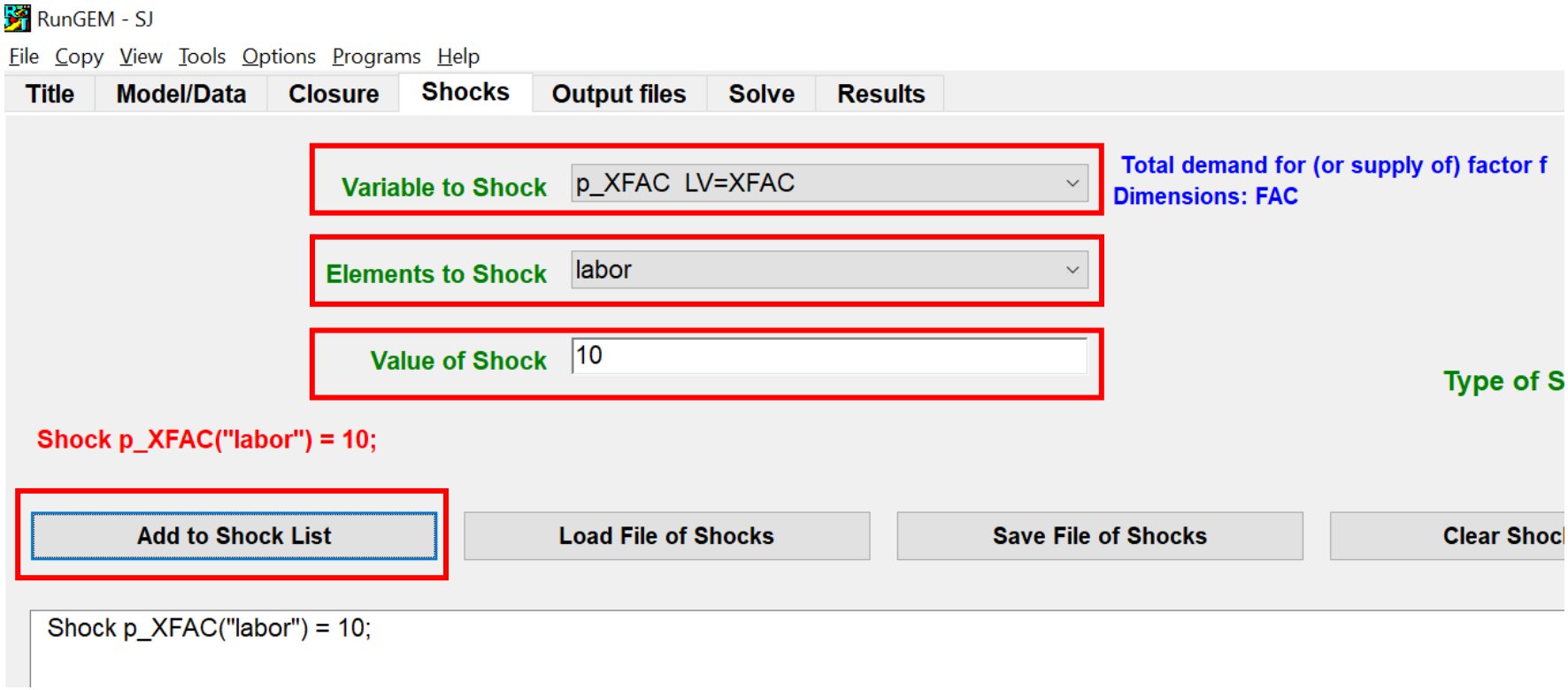
Click on this line to select p_XFAC and in the ***Elements to Shock*** box click on the arrow and select "labor".

Thus in this simulations you are shocking just one component of p_XFAC, the labor supply.

Type in the next box the value of shock – 10.

Click on the button ***Add to shock list***

6. TABLO-generated program



6. TABLO-generated program

Output Files:

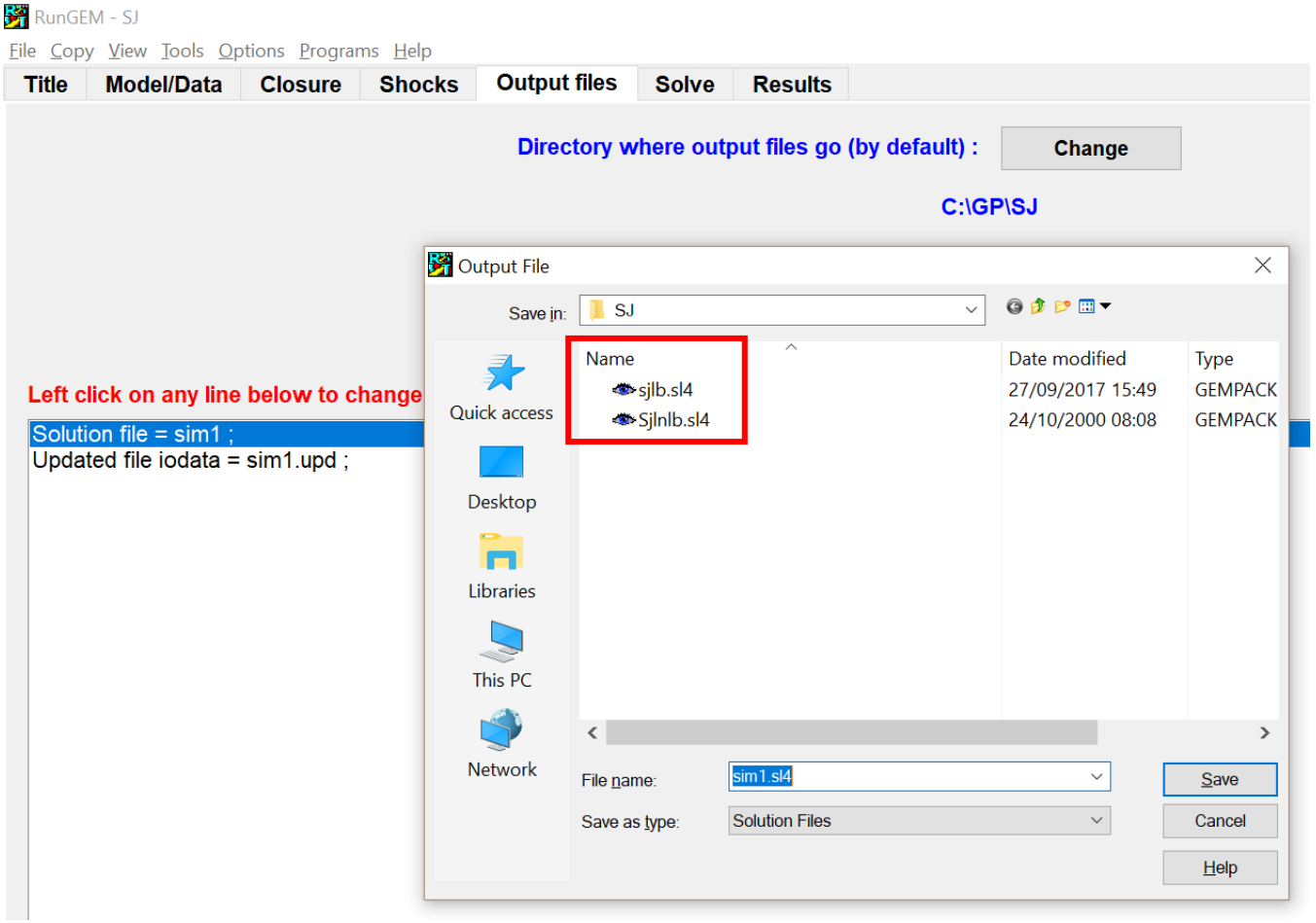
Click on ***Output files*** tab

To change the names of the output files, click (left click on this time) on the first line in the lower box:

Solution file = sim1.

Change the name of the Solution file to SJLB.SL4

6. TABLO-generated program



6. TABLO-generated program

Carry out a simulation:

Select the next page ***Solve*** and type in some verbal description to say

SJ. Standard closure – 10 percent increase in the labor supply

You need to select the solution method and steps.

Click on the ***Change*** button to the right of the text "Solution method".

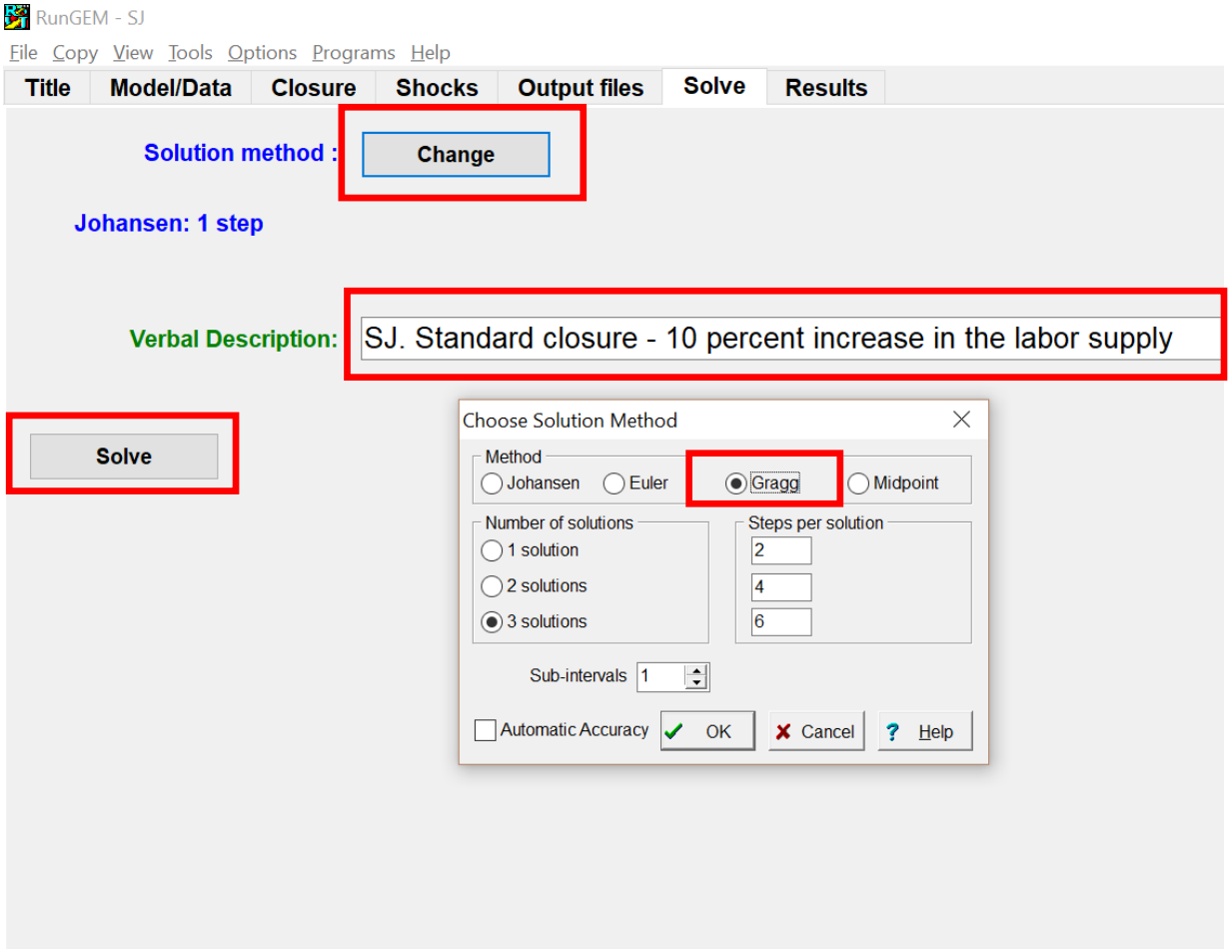
6. TABLO-generated program

Carry out a simulation:

Select “Gragg’s method with 2,4,6 steps calculations [One subinterval, not automatic accuracy]

Click on the ***Solve*** button and the RunGEM will calculate a solution.

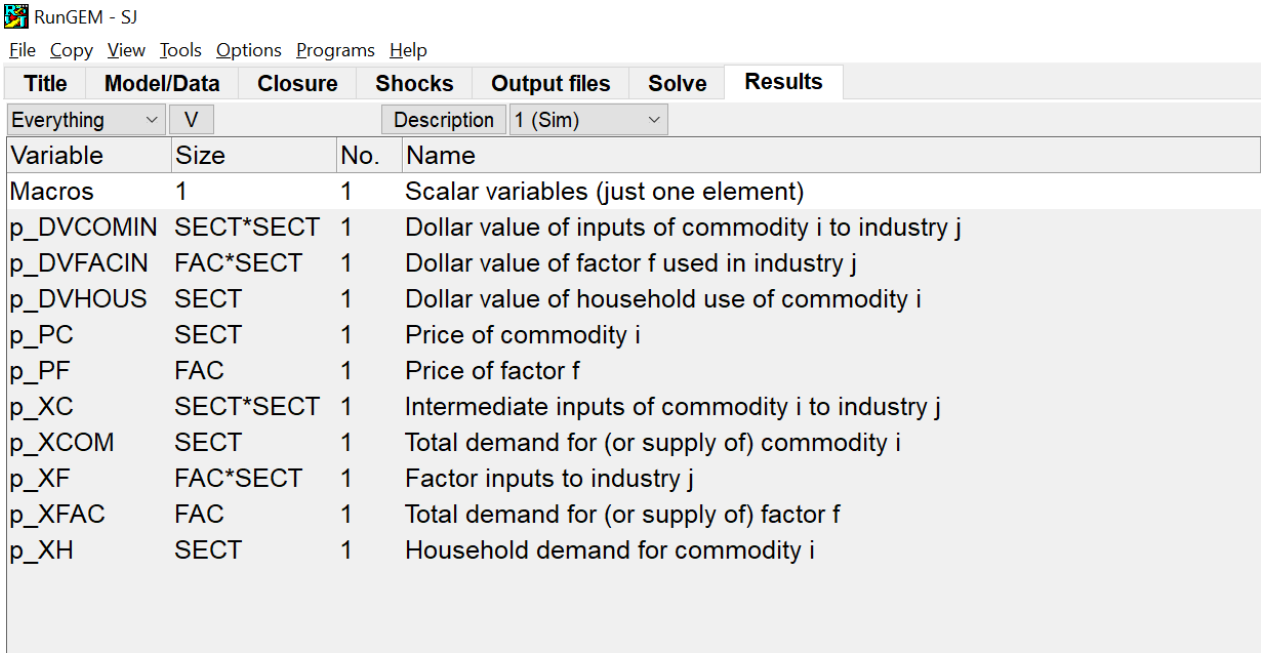
6. TABLO-generated program



6. TABLO-generated program

Look at the results:

Click on ***Results*** tab



Variable	Size	No.	Name
Macros	1	1	Scalar variables (just one element)
p_DVCOMIN	SECT*SECT	1	Dollar value of inputs of commodity i to industry j
p_DVFACIN	FAC*SECT	1	Dollar value of factor f used in industry j
p_DVHOUS	SECT	1	Dollar value of household use of commodity i
p_PC	SECT	1	Price of commodity i
p_PF	FAC	1	Price of factor f
p_XC	SECT*SECT	1	Intermediate inputs of commodity i to industry j
p_XCOM	SECT	1	Total demand for (or supply of) commodity i
p_XF	FAC*SECT	1	Factor inputs to industry j
p_XFAC	FAC	1	Total demand for (or supply of) factor f
p_XH	SECT	1	Household demand for commodity i

Examples (GPD-8)

2.1.12. Different closures and/or shocks

2.1.13. Correcting errors in TABLO Input files

2.1.15. Creating the base data header array file

2.1.17. Condensing the model

Bonus. Sensitivity analysis to alternative functional forms to intermediate demands equations (Leontief versus Cobb-Douglas versus CES – sigma = 2)