Working Paper

THE HUMUS (HUNGARIAN MULTISECTORAL) MODEL FAMILY: A USER'S GUIDE TO THE COMPUTER PROGRAMS

Ernö Zalai

November 1984
WP-84-99

International Institute for Applied Systems Analysis
A-2361 Laxenburg, Austria

NOT FOR QUOTATION
WITHOUT PERMISSION
OF THE AUTHOR

THE HUMUS (HUNGARIAN MULTISECTORAL) MODEL FAMILY: A USER'S GUIDE TO THE COMPUTER PROGRAMS

Ernö Zalai

November 1984
WP-84-99

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
2361 Laxenburg, Austria

PREFACE

Many of today's most significant socioeconomic problems, such as slower economic growth, the decline of some established industries, and shifts in patterns of foreign trade, are international or transnational in nature. But these problems manifest themselves in a variety of ways; both the intensities and the perceptions of the problems differ from one country to another, so that intercountry comparative analyses of recent historical developments are necessary. Through these analyses we attempt to identify the underlying processes of economic structural change and formulate useful hypotheses concerning future developments. The understanding of these processes and future prospects provides the focus for IIASA's project on Comparative Analysis of Economic Structure and Growth.

Our research concentrates primarily on the empirical analysis of interregional and intertemporal economic structural change, on the sources of and constraints on economic growth, on problems of adaptation to sudden changes, and especially on problems arising from changing patterns of international trade, resource availability, and technology. The project relies on IIASA's accumulated expertise in related fields and, in particular, on the data bases and systems of models that have been developed in the recent past.

In this paper, Ernö Zalai presents a user's guide and documentation of the "HUMUS" family of models. These computable general equilibrium models for the Hungarian economy were developed through collaborative efforts at IIASA and the Computer Center of the Hungarian Planning Commission. The paper describes the structure and mode of operation of the models, as well as providing a complete mathematical statement of the possible model variants. Also included is a sample of the communication option possible in the interactive model versions.

Anatoli Smyshlyaev
Project Leader
Comparative Analysis of
Economic Structure and
Growth

THE HUMUS (HUNGARIAN MULTISECTORAL) MODEL FAMILY: A USER'S GUIDE TO THE COMPUTER PROGRAMS

Ernö Zalai

INTRODUCTION

This paper documents and provides a user's guide to a suite of computer programs designed to solve a certain family of computable general equilibrium models. The models were developed at IIASA in close cooperation with colleagues at the Computer Center of the Hungarian Planning Commission. At various stages of model development Gy. Boda, I. Csekö, F-né Hennel, L. Lászlo, R. Nishimiya, S. Poviliaitis, F. Sivák, A.Tihanyi, and L. Zeöld assisted me in collecting data, preparing a solution algorithm, and writing the computer programs for the numerical models. Their valuable assistance is gratefully acknowledged.

A complete mathematical statement of the possible model variants is presented in Appendices 1 and 2. Various other IIASA publications describe the theoretical and conceptual considerations underlying the different models and subsections thereof (see, for example, Zalai 1980, 1982, 1983). Here we will only briefly outline the common general structure of the models and indicate the nature of possible variations.

The physical (real) part of the model is based on an input–output framework. The production, trade, and intermediate and final uses of various sectoral commodities are the major real endogenous variables of the model. On the production side, nonsubstitutable imports and two other primary factors (labor and capital) are considered, as well as intermediate inputs. A set of accounting identities (balance requirements) constrains the allocation of these resources.

Unlike traditional input–output models, however, the present model does not treat the various input coefficients or the sectoral composition of the different parts of final demand as fixed parameters. The coefficients can vary according to some predefined rules, mainly derived from substitutability assumptions. Thus, for example, labor and capital, or domestic products and imports from different regions with the same sectoral classification, can be treated as imperfect substitutes, as well as the commodities consumed.

The various commodity balances and constraints describing substitution possibilities and special restrictions (for example, the investment/consumption ratio) constitute what might be called the *primal* part of the model, to borrow a term from the mathematical programming literature. In fact, one version of the model is precisely a nonlinear multisectoral model of optimal resource allocation. In this variant, different types of objective function can be optimized subject to the aforementioned constraints. This model is represented by and solved on the basis of a manipulated version of the so-called Kuhn—Tucker conditions associated with the given constrained optimum problem. This formulation introduces several accounting (shadow) prices and various constraints that they have to satisfy (the *dual* part of the model). Structurally similar conditions can also be derived on the basis of competitive general equilibrium theory. This latter route is typically followed in formulating models of applied (or computable) general equilibrium. Either way we get a nonlinear equation system defined in terms of both the real and the price (cost) variables.

Thus, our model is essentially a nonlinear equation system, whose basic structure and specific variants are described in detail in Appendix 2. To handle these variations flexibly, we have made use of ***specification control variables***. This is a slightly inaccurate name for these variables, because sometimes they only control the way in which some input parameters (and not the model equations) are specified. A list of these control variables and their function can be found in Appendix 1. By choosing different values for the control variables the user can define the actual variables and equations of the model and the mathematical form of some relations, as well as determining some of the parameters.

1. GENERAL OUTLINE OF THE SOLUTION ALGORITHM

The model to be solved has a flexible structure, because, as indicated earlier, choosing values of specific *control variables* defines one or other variant of the model. In some instances only the mathematical form of a given (production or utility) function will be different; in other cases, however, the number of equations and variables may also be different. The special solution algorithm developed (more details of which can be found in Sivák *et al.* 1984) is flexible enough to handle the different model variants.

Two basically different cases can be distinguished, depending on whether the trade flows in the second ("rouble") trade area are fixed or not. If they are not fixed, then the variables can be grouped into three major subsets as follows:*

1. VMAIN = $\{wg, r, vr, vd\}$
2. VBLOCK = $\{pmbr, pmbd, pmb, mbrs, w, q, s, lk, kk,$
   $pmr, pmd, mrr, mdr, sh, sd, sr, phm,$
   $p, pinv\}$
3. VLINBLK = $\{zr, zd, z, pzd, txzd, ztar, pchm, pcmb, pc,$
   $vni, pvn, et, ev, ccv, chmv, cmbv, chm, cmb,$
   $ch, vninv, ninv, wf, ah, bh, fdh, x,$
   $ginv, l, k, ghus, md, mr, mb, mbr, mbd\}$

To these variables we can assign specific equations, also grouped in the same way:

*The variable names and equation numbers correspond to those in Appendices 1 and 2.

1. EMAIN = {6,7,8,9}
2. EBLOCK = {42,43,44,29,36,37,38,34,35,
   39,40,24,25,26,27,28*a*,41,45,48}
3. ELINBLK = {20,21,22,50,23*a*,23*b*,46*a*,46*b*,47,10,
   49,11,12,13,14*a*,14*b*,15*a*,15*b*,16*a*,17*a*,
   17*b*,19,32*a*,33*a*,2*c*,26,
   18,6,7,1,3,4,5,30,31}

This decomposition of variables and equations results in a useful structure that the solution algorithm exploits. It can be seen that if the variables in VMAIN are known, the variables in VBLOCK can be determined from equation group EBLOCK alone. And when the variables in both VMAIN and VBLOCK are known, the solution of the equations belonging to ELINBLK yields the corresponding values of the variables in VLINBLK. Moreover, BLOCK can be further decomposed into subgroups. The first (EBLOCK 1) consists of equations linear in the variables assigned to them. The remaining set is a nonlinear equation system, but even there a simple iteration method (based on the contractive mapping properties of the equation system) can be applied to reach the solution. ELINBLK is an equation system linear in the variables belonging to VLINBLK. Note also that the equations in EMAIN directly depend only on variables in VBLOCK and VLINBLK.

From what we have said above, the idea of the solution algorithm seems fairly simple and straightforward. On the one hand we decompose the solution in such a way that the resulting partial problems are either linear or nonlinear, but easy to solve (the groups BLOCK, LINBLK). On the other hand we reduce the solution of the whole equation system to four equations in four variables *wg*,*r*,*vr*, and *vd*. We start our solution algorithm by assigning initial values to these four variables. Then we solve the rest of the equations (EBLOCK, ELINBLK) and obtain a solution for the rest of the variables as functions of the first four variables. If the resulting variables do not fulfill equations VMAIN we try to find new values for those in VMAIN.

This latter task is performed by using a simple Newton iteration procedure, where the Jacobian is numerically estimated. To do this, in each main iteration we have to solve the rest of the model (BLOCK and LINBLK) five times (for different values of the four variables in VMAIN). Figure 1 illustrates the algorithm outlined above.

If the trade flows in the second ("rouble") trade area are fixed, the algorithm outlined above has to be modified. This is because the decomposition method described does not possess all of the useful properties discussed above, due to a different model specification. The crucial change is that two sets of variables (*sr*,*mbrs*) now depend directly on variables in VLINBLK, rather than on those in VBLOCK, while at the same time some variables in VBLOCK will depend on them. Thus the previous recursive structure no longer holds. We can, however, overcome this problem in the following simple way. We assign some initial values to these variables and recalculate them in the course of each main iteration (*sru*,*mbrsu*). We continue the iteration until these values converge to a sufficient degree of accuracy. This check on convergence and the updating of *sr* and *mbrs* is also executed in MAIN, which results in cri-

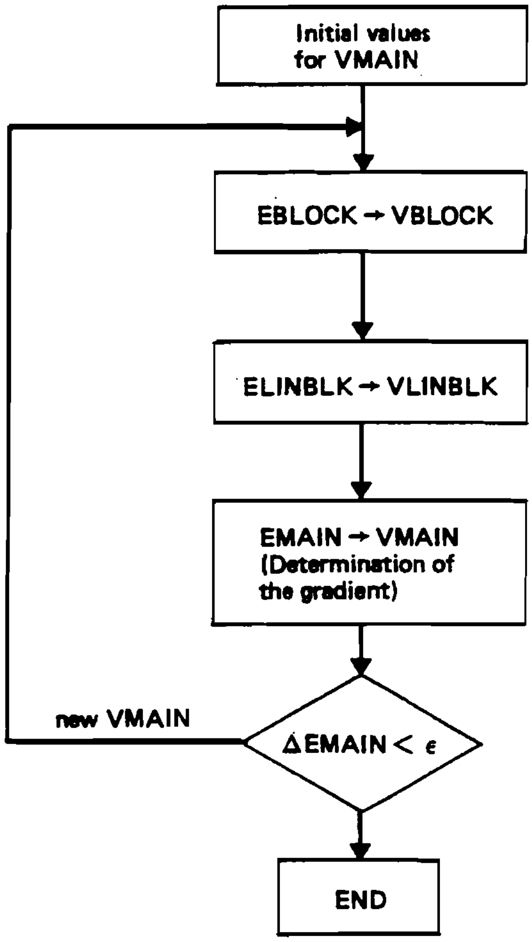

Figure 1. The flow chart of the solution algorithm.

teria additional to those discussed earlier for stopping the main iteration.

This concludes the overall outline of the solution algorithm and the corresponding part of the computer algorithm (MAIN, BLOCK, LINBLK). In the next section we turn to a somewhat more detailed description of the subroutines involved.

2. STRUCTURE OF THE COMPUTER PROGRAM

The computer program consists of a main program (MAIN) and several subroutines: READ, RF, KOMBI, CALC, BLOCK, LINBLK, INV, PRINT, and PRSTAT. One input channel and four output channels are used to handle information. The structure of the program can be best seen from Figure 2 (some of the less important links are not shown).

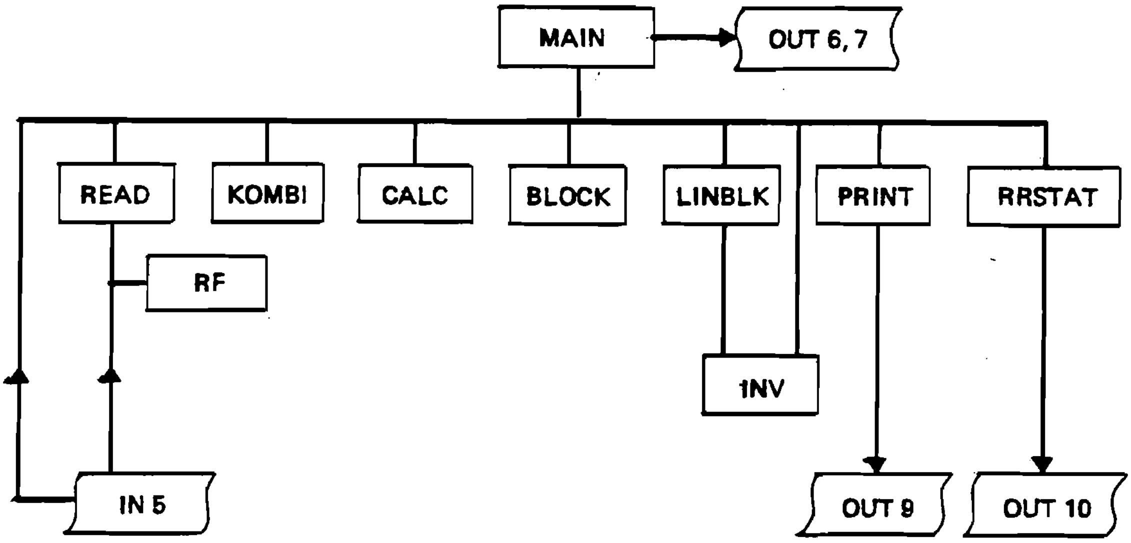

Figure 2. The structure of the computer program.

The Program MAIN

MAIN fulfills several tasks, one of which is the usual function of such programs, namely to organize the subroutines. Apart from that, MAIN first reads an item of information *(kom)* from the input (IN) file, which tells the program whether one or several runs are to be executed at the same time. If *kom* is greater than 1, then the program organizes a loop to repeat the calculations *kom* times. Differences between various runs have to be specified in the subroutine KOMBI in advance, since the input file cannot be modified during the running of the program. This option is useful if one has a clear idea in advance about the types of analysis or alternative assumptions to be explored. The subroutine PRSTAT summarizes and compares the various runs according to a number of main variables at the end. The input file is reread in each run, which is why we have to *rewind* it at the beginning of each run.

Next, the subroutines READ and CALC are called and executed. After this MAIN organizes the more important steps in estimating the Jacobian matrix needed for the main Newton iteration, as described in the previous section. For this the model is solved four or five times (depending on the exact specification), using various values of the VMAIN variables. (In some specifications one or other of the trade balance restrictions is lifted, and the related exchange rate variables are exogenously fixed.) The actual solution of the model is obtained in subroutines BLOCK and LINBLK. The variable *iter* records the number of main iterations needed to reach the "sufficiently accurate" solution (set by the parameter *eps)*. A maximum number of iterations can be set by the parameter *itera*, or a minimum by *mitera*. MAIN records some overall information on the iterative procedure, including the differences in the EMAIN equations and the gradient and new values of the variables in VMAIN. The other parameters that regulate the Newton iteration are del_i (size of the differences) and *trukk* (step size). The values of the major variables of the model obtained in various iterations can be stored in output file OUT6 after BLOCK and LINBLK have been executed. After each run, the subroutine PRINT is executed (preparing and printing output files) and at the end, when all runs are complete, PRSTAT is called to print the summary results.

The Subroutines READ and RF

The subroutines READ and RF have only one function: to read in various parameters and primary inputs for the model. These include, first of all, the number of sectors n. Since the program is flexible in this respect, special care must be taken that the number of sectors and the whole data file structure are accurate. Different input files (IN) must be prepared for different types of sectoral breakdown. The loops in reading the data depend on this variable number. The sectoral input data must be grouped so that five numbers are in each line (except possibly in the last line).

The Subroutine KOMBI

The subroutine KOMBI was designed to facilitate combined runs of the model, in which one can test a combination of alternative assumptions and their effects on the results. Thus, for example, the same problem can be analyzed using different model specifications, parameter sizes, etc. KOMBI is called when the initial data are read and it is possible to modify them according to the wishes of the user in the various runs. One typical way to do this is to make the value of some specification control variables or model parameters (e.g. elasticities) depend on the value of the "run" number, i.e. the loop variable *kom* (e.g. if *kom* = 2, *kdex*=2). The results of the various runs are summarized at the end in OUT10, an output file printed by PRSTAT.

The Subroutine CALC

The subroutine CALC is used for various purposes: to set the initial values of some variables (e.g. those required for the iteration), to calculate from the initial data various model parameters (e.g. single point estimates of production or utility function parameters, if needed), and to transform initial data if the model specification is different from that for which the input file was designed (e.g. treating noncompetitive imports as competitive or combining the two trade areas into one, etc.).

The Subroutine BLOCK

As described in Section 1, the values of the variables in VBLOCK are calculated in the subroutine BLOCK by solving the equations of EBLOCK for some fixed values of *wg*, *r*, *vr*, and *vd*. The first subgroup of variables in VBLOCK1 can be calculated directly and easily (LOOP 10). These variables and the corresponding equations (with the equation number given in parentheses after the corresponding variable) are as follows:

BLOCK 1: *pmbr* (42), *pmbd* (43), *pmb* (44), *mbrs* (29a)*, *w* (36)

*Only if *krtr*=1, i.e. "rouble" trade flows are also variable.

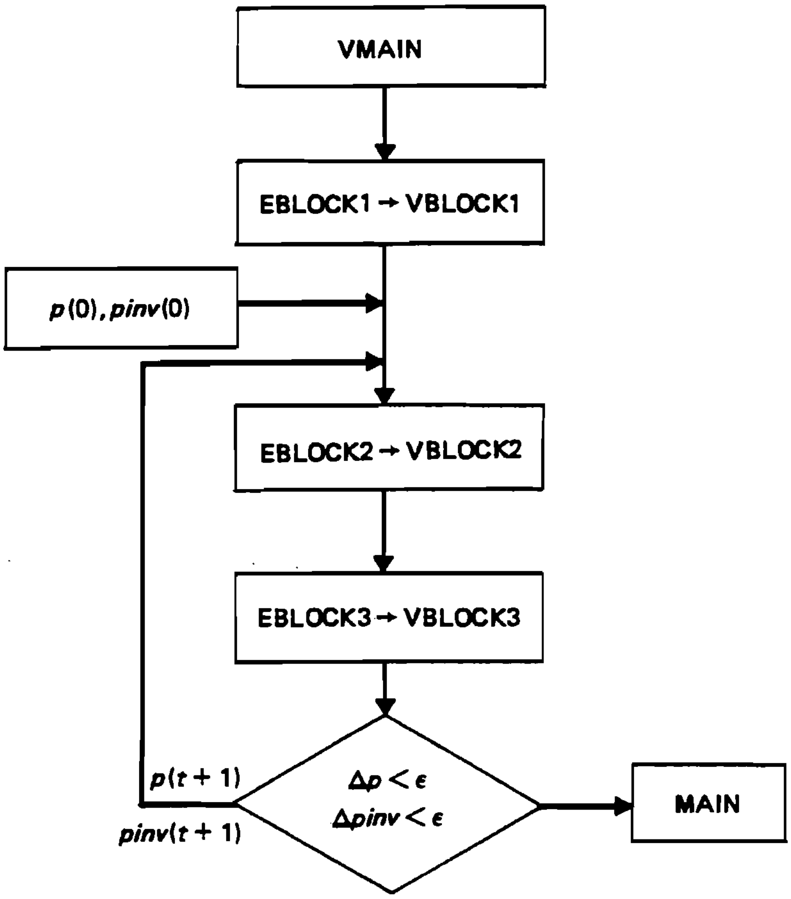

Figure 3. Flow chart of the subroutine BLOCK

The second subgroup comprises the nonlinear simultaneous equation system. If, however, we wish to solve for p and $pinv$ through simple iteration (based on the contractive property of the mappings involved), the solution is once again quite simple. First we assign some initial values to these variables (either in CALC or by using their values from previous iterations), then we recursively determine the values of the other variables and also the new values of p and $pinv$ themselves, and we continue until sufficient accuracy is attained. Thus, given some initial values $p(0)$ and $pinv(0)$, we proceed as follows:

BLOCK 2: *q* (37), *s* (38), *lk* (34), *kk* (35),
pmr (39), *pmd* (40), *mrr* (24), *mds* (25),
sh (26), *sd* (27), *sr* (28a)*, *phm* (41)

BLOCK 3: $p(t+1)$ (45), $pinv(t+1)$ (48)

and return to either BLOCK 2 or MAIN.

The flow chart of BLOCK is presented in Figure 3.

The Subroutine LINBLK

The subroutine LINBLK starts with previously calculated values for the variables in VMAIN and VBLOCK, and solves the equations of ELINBLK for variables in VLINBLK. It can be further decomposed into the following subgroups:

LINBLK 1 (Export and related variables)
zr (20), *zd* (21), *z* (22), *pzd* (50), *txzd* (23a), *ztar* (23b)

LINBLK 2 (Consumption and related variables)
pchm (46a), *pcmb* (46b), *pc* (47), *vni* (10),
pvn (49), *et* (12), *ev* (11), *ccv* (13),
chmv (14a), *cmbv* (14b), *chm* (15a), *cmb* (15b),
ch (16a), *vninv* (17a), *ninv* (17b), *wf* (19)

LINBLK 3 (Final demand and production)
ah (32a), *bh* (33a), *fdh* (2c), *x* (2b)

LINBLK 4 (Rest of VLINBLK)

ginv (18), *l* (6), *k* (7),

hus (1), *md* (3), *mr* (4),

sr (28b),* *mb* (5), *mbr* (30),

mbd (31), *mbrs* (29b)*

When the values of all these variables have been calculated, the program follows in MAIN. Since LINBLK contains only relatively straightforward calculations, there is no need to provide any chart of the subroutine.

The Subroutine PRINT

When the model is solved, the subroutine PRINT is called for two major purposes. The first of these is to prepare and print in the output file OUT9 the values of various parameters (for later use) and the values of the variables arranged into a series of tables. These tables are as follows:

1. Export
2. Prices related to export
3. Dollar import rates and rouble import rates
4. Competitive imports
5. Noncompetitive imports
6. Total consumption
7. Variable consumption
8. Primary factor allocation and production
9. Factor coefficients and costs
10. Prices before rescaling (*w*,*q*,*s*,*prof*)
11. Prices before rescaling (*p*,*pmr*,*pmd*,*phm*)
12. Prices before rescaling (*pinv*,*pc*,*vr*,*vd*,*w*,*r*,1+*w*, *etc.*)
13. Shadow (equilibrium) prices (*w*,*q*,*s*,*prof*)
14. Shadow (equilibrium) prices (*p*,*pmr*,*pmd*,*pa*)
15. Shadow (equilibrium) prices (*pinv*,*pc*,*vr*,*vd*,*w*,*r*, *etc.*)
16. Main indicators

*Only if *brtr* > 1, i.e. "rouble" trade flows are fixed.

The main indicators listed in table 16 include the following:

Variable consumption
Net investment
Rouble trade balance
Rouble terms of trade
Dollar trade balance
Dollar terms of trade
Export surplus/disposable GDP
Return rate of wages
Return rate on capital
Dollar exchange rate
Rouble exchange rate
Implicit welfare function
Final consumption
Gross accumulation
GDP
Net national production
Gross national production
Total export
Total import
Total trade/GDP ratio
Export surplus/import ratio
Total competitive import
Total noncompetitive import
Total rouble import
Total dollar import
Total rouble export
Total dollar export

The second task of PRINT is to calculate the statistics that will be printed, after all the required runs are complete, by the subroutine PRSTAT. Only one task relates directly to the solution of the model, and this is left for the subroutine PRINT; this is the chosen scaling of prices (see equation 51). This requires that the base and current value of total fixed consumption be the same. If they are not, then all variables in which the rest of the equations are homogeneous of degree 0 have to be rescaled accordingly. The scaling parameter is $gpl = cft/eft$.

The Subroutine STPRINT

Unlike PRINT, which produces tables containing the absolute values of the variables (except for a few indices), the subroutine PRSTAT mostly prints tables that show percentage changes with respect to the base solution. The other difference is that PRINT prints tables for each run separately in the file OUT9, whereas PRSTAT does it in a combined way in OUT10. The tables produced are as follows:

1. Dollar export
2. Rouble export
3. Dollar competitive import
4. Rouble competitive import
5. Dollar noncompetitive import
6. Rouble noncompetitive import
7. Variable consumption
8. Production
9. Labor employed
10. Capital used
11. Capital/labor ratio
12. Producers' price indexes
13. Domestic producer price/dollar exchange rate
14. Dollar export price indexes
15. Main indicators

for each of which Base, Run1, Run2,..., etc., values are reported.

3. AN INTERACTIVE VERSION OF THE PROGRAM

The program described above assumes that the user can edit files and run programs. In order to make the program more readily accessible for those who do not wish to do the programming themselves, an interactive version has been developed. The structure of this latter program is different only in so far as it contains a few additional subroutines to display information (data or tables) and to execute changes designed by the user.

These new subroutines fall into two categories. CHINP, MODATV, MODATS, and MODIS are called after the input file has been read. If the reader wishes to see and/or to change some (indicated by the program) input data (control specification variables or parameters), this can be done interactively. With the help of these subroutines the user may also display on the screen the meaning (content) of variables and parameters as well as the model equations in which they appear. The DISPLAY subroutine makes it possible to display on the screen after each run any of the tables that are prepared and printed by PRINT and PRSTAT. Note also that in this version PRINT also prepares the information for DISPLAY. Appendix 3 contains an example of the communication facility built into the interactive version.

One final word of warning for those who might wish to transfer the program to another computer. Unlike the first version, which was written in almost standard FORTRAN,* the interactive version contains many machine-specific commands. Special care and adaptation is therefore needed to implement the program on other computers.

*The program was developed on a VAX 11/780 computer using the UNIX operating system.

REFERENCES

Sivák, J., A. Tihanyi, and E. Zalai. (1984) *On the Solution of a Computable General Equilibrium Model.* CP-84-23. Laxenburg, Austria: International Institute for Applied Systems Analysis.

Zalai, E. (1980). *A Nonlinear Multisectoral Model for Hungary: General Equilibrium Versus Optimal Planning Approach.* WP-80-148. Laxenburg, Austria: International Institute for Applied Systems Analysis. Revised version in A.C. Kelley, W.C. Sanderson, and J.G. Williamson (Eds.). *Modeling Growing Economies in Equilibrium and Disequilibrium.* Durham, North Carolina: Duke University Press.

Zalai, E. (1982) *Foreign Trade in Macroeconomic Models: Equilibrium, Optimum, and Tariffs.* WP-82-132. Laxenburg, Austria: International Institute for Applied Systems Analysis.

Zalai, E. (1983) *Economic Reform, Allocative Efficiency, and Terms of Trade.* WP-83-112 Laxenburg, Austria: International Institute for Applied Systems Analysis.

APPENDIX 1: LIST OF VARIABLES AND PARAMETERS

Variables

x_j	home output of sectoral commodity j[*]
fdh_j	final demand for home output of sectoral commodity j
mr_j	competitive rouble import of sectoral commodity j
mrr_j	ratio of competitive rouble import of sectoral commodity j to home supply thereof
md_j	competitive dollar import of sectoral commodity j
mdr_j	ratio of competitive dollar import of sectoral commodity j to home supply thereof
m_j	total competitive import of sectoral commodity j
mbr_j	noncompetitive rouble import of sectoral commodity j
mbd_j	noncompetitive dollar import of sectoral commodity j
mb_j	total noncompetitive import of sectoral commodity j
$mbrs_j$	rouble share of noncompetitive imports at sectoral level of gross investment
$ginv$	level of gross investment
$ninv$	level of net investment
$vninv$	level of variable net investment

[*] Sector indices run from 1 to n. The value of a variable with index $(n+1)$ or $(n1)$ denotes the sum of the corresponding sectoral variables, e.g. $x(n+1) = \sum_{j=1}^{n} x(j)$.

ct	total consumption (at base prices)
chm_j	consumption of sectoral commodity j (home-competitive import composite)
$chmv_j$	variable part of the consumption of commodity j (home-competitive import composite)
ch_j	consumption of sectoral commodity j (home supply)
cr_j	consumption of sectoral commodity j (rouble competitive import)[*]
cd_j	consumption of sectoral commodity j (dollar competitive import)[*]
cmb_j	consumption of sectoral commodity j (noncompetitive import)
$cmbv_j$	variable part of the consumption of sectoral commodity j (noncompetitive import)
ccv	overall (implicit utility) level of variable consumption
hus_j	home use of sectoral (home-competitive import) composite commodity j
sh_j	home supply share of home use of sectoral commodity j (home-competitive import composite)
sr_j	rouble import share of home use of sectoral commodity j (home-competitive import composite)

[*] Appears only in the model description, but not in the program.

sd_j	dollar import share of home use of sectoral commodity j (home-competitive import composite)
$ah_{i,j}$	input–output coefficients (from home supply)
$ad_{i,j}$	input–output coefficients (from dollar-competitive imports)[*]
$ar_{i,j}$	input–output coefficients (from rouble-competitive imports)[*]
bh_i	investment input coefficients (from home supply)
bd_i	investment input coefficients (from dollar-competitive imports)[*]
br_i	investment input coefficients (from rouble-competitive imports)[*]
z_j	total export of sectoral commodity j
zr_j	rouble export of sectoral commodity j
zd_j	dollar export of sectoral commodity j
k_j	capital used in sector j
kk_j	capital coefficient in sector j
l_j	labor used in sector j
lk_j	labor coefficient in sector j
r	net rate of return on capital
q_j	cost of capital in sector j

[*] Appears only in the description of the model.

rs	shadow price of capital ($r.pinv$)
wg	net rate of return on labor
ws	shadow price of labor
w_j	cost of labor in sector j
s_j	joint cost of labor and capital in sector j
p_j	producer's price (index) of sector j
pmr_j	user's price (index) of sectoral commodity j (rouble-competitive import)
pmd_j	user's price (index) of sectoral commodity j (dollar-competitive import)
phm_j	average user's price (index) of sectoral commodity j (home-competitive import composite)
$pmbr_j$	user's price (index) of sectoral commodity j (rouble-noncompetitive import)
$pmbd_j$	user's price (index) of sectoral commodity j (dollar-noncompetitive import)
pmb_j	average user's price (index) of sectoral commodity j (noncompetitive import)
$pchm_j$	price of sectoral commodity j in consumption (home-competitive import composite)
$pcmb_j$	price of sectoral commodity j in consumption (noncompetitive import)

pc	price (index) of variable consumption
$pinv$	average price (index) of investment (capital goods)
pzd_j	dollar export price (index) of sectoral commodity j
$ztar_j$	endogenous export tariff on sectoral commodity j
$povd_j$	producer's price/dollar exchange rate in sector j
vr	index of rouble exchange rate
vd	index of dollar exchange rate
vni	value of variable national income
pvn	price index of variable (real) national income
et	total consumption expenditure
ev	expenditure left for variable consumption
wf	level of the implicit welfare (objective) function
sru_j	updated value of sr_j
$mbrsu_j$	updated value of $mbrs_j$
ind_k	value of main indicator k (used in program output)

Parameters

$ahm_{i\,j}$	input–output coefficients (home-competitive import composite)
$amb_{i,j}$	input–output coefficients (noncompetitive import)
bhm_i	investment input coefficients (home-competitive import composite)

bbm_i investment input coefficients (noncompetitive import)

$rinv$ net investment ratio (to disposable national income)

$rinco$ net investment/consumption ratio

$chmf_j$ fixed part of the consumption of commodity j (home-competitive import composite)

$cmbf_j$ fixed part of the consumption of commodity j (noncompetitive import)

$ckhm_j$ coefficients related to the variable consumption of commodity j (home-competitive import composite)

$ckmb_j$ coefficients related to the variable consumption of commodity j (noncompetitive import)

$elco$ elasticity of substitution in the (variable) consumption index function

$elci$ elasticity of substitution between variable consumption and investment (0, if $kinv$ = 1)

wd_j wage level in sector j

amr_j amortization rate in sector j

$repl_j$ capital replacement rate in sector j

$profr_j$ profit rate in sector j

$profc_j$ profit coefficient in sector j (with respect to gross revenue)

$knet_j$ rate of net to gross capital in sector j

kt total available capital stock

lt total available labor force

$kszi_j$ labor (share) coefficient in the jth sectoral production capacity function

$zeta_j$ scaling coefficient in the jth sectoral production capacity function

$sigma_j$ elasticity of substitution in the jth sectoral production capacity function

$beta_j$ CES elasticity parameter $(1/(1+beta) = sigma)$ in the jth sectoral production capacity function

$mrr0_j$ scaling parameter in the function of the competitive rouble import ratio, sectoral commodity j

$mrel_j$ elasticity parameter in the function of the competitive rouble import ratio, sectoral commodity j

$mdr0_j$ scaling parameter in the function of the competitive dollar import ratio, sectoral commodity j

$mdel_j$ elasticity parameter in the function of the competitive dollar import ratio, sectoral commodity j

eta_j elasticity parameter $(1/(1+eta) = mdel)$ in the home-dollar competitive import CES substitution function, sectoral commodity j

hh_j weight parameter of the home supply in the import substitution function, sectoral commodity j

hm_j weight parameter of the import in the import substitution function, sectoral commodity j

$mbrs0_j$ scaling parameter in the function of the noncompetitive rouble import share, sectoral commodity j

$mbrel_j$ elasticity parameter in the function of the noncompetitive rouble import share, sectoral commodity j

$zr0_j$ scaling parameter in the rouble export function, sectoral commodity j

$zrel_j$ elasticity parameter in the rouble export function, sectoral commodity j

$zd0_j$ scaling parameter in the dollar export function, sectoral commodity j

$zdel_j$ elasticity parameter in the dollar export function, sectoral commodity j

$zdeld_j$ demand elasticity of the dollar export, sectoral commodity j

$zdels_j$ supply elasticity of the dollar export, sectoral commodity j

$zdef_j$ relative cost of dollar earning in sector j

$zref_j$ relative cost of rouble earning in sector j

btd dollar balance-of-trade target

btr rouble balance-of-trade target

$pwzd_j$ dollar world market export price of sectoral commodity j

$pwzr_j$ rouble world market export price of sectoral commodity j

$pwmd_j$ dollar world market import price of competitive imports, sectoral commodity j

$pwmr_j$ rouble world market import price of competitive imports, sectoral commodity j

$pwmbd_j$ dollar world market import price of noncompetitive imports, sectoral commodity j

$pwmbr_j$ rouble world market import price of noncompetitive imports, sectoral commodity j

txh_j rate of change in user tax-subsidy factor on the home produced sectoral commodity j

$txmr_j$ rate of change in user tax-subsidy factor on the rouble competitive import of sectoral commodity j

$txmd_j$ rate of change in user tax-subsidy factor on the dollar competitive import of sectoral commodity j

$txmbr_j$ rate of change in user tax-subsidy factor on the rouble noncompetitive import of sectoral commodity j

$txmbd_j$ rate of change in user tax-subsidy factor on the dollar noncompetitive import of sectoral commodity j

$txzr_j$ rate of change in the tax-subsidy factor on the rouble export of sectoral commodity j

$txzd_j$ rate of change in the tax-subsidy factor on the dollar export of sectoral commodity j (in some versions this is itself a variable) *bvar* base value of variable *var* (for example bx_j, etc.)

stvar base related index of variable *var*

Specification Control Variables

kcof measure of variable consumption ('welfare') level (1–3)

1 Cobb–Douglas function (LES)

2 Kantorovitch–Leontief function (fixed structure)

3 CES function

kcoc coefficients of the excess consumption function (1–2)

1 exogenously given

2 same as that of the fixed part of consumption

col fixed part of consumption uniformly adjusted by the factor *col*[*]

kpft type of production function (1–2)

1 Cobb–Douglas

2 CES

kpfc coefficients of production function (1–2)

1 exogenously given

2 calculated from base data

[*] This does not appear in the equations that follow, because the program modifies the original data (read from the input file). Thus, the fixed consumptions in the model description should be understood as those in the input file multiplied by *col*.

kobj gains (losses) expressed in terms of (1–2)

1 variable national income (***kinv*** = 1 or 3) or consumption (***kinv*** = 2)

2 dollar trade balance

3 rouble trade balance

kinv determination of net investment (1–2)

1 fixed proportion of consumption

2 fixed amount

3 variable proportion of consumption (substitutes)

kpri price formation rule (1–6)

1 'nonprofit'

2 exogenous profit factor (multiplier)

3 exogenous profit ratio (additive)

4 calculated profit factor (multiplier)

5 calculated profit ratio (additive)

6 base profit factor (multiplier)

ktra number of trade areas (1–2)

1 only one trade area (rest of the world)

2 two trade areas with separate accounts

krtr treatment of rouble trade (1–4)

1 variable

2 fixed, rouble exchange rate constant

3 fixed, rouble exchange rate changes with the dollar rate

4 fixed, rouble import prices change as average prices

kdex dollar export (price) treatment (1–4)

1 demand (price varies with volume)

2 supply (constant price)

3 supply–demand equilibrium (price varies)

4 optimum (tariff) solution

kdim dollar import treatment (1–3)

1 perfect substitute – imperfect adjustment

2 imperfect substitute – perfect adjustment

3 as 2, plus nonlinear programming features

kcim noncompetitive import treatment (1–2)

1 taken into account

2 neglected (added to competitive)

APPENDIX 2: MODEL EQUATIONS

Balances

(1) Home use of home-competitive import composite goods

$$hus_i = \sum_{j=1}^{n} ahm_{i,j}x_j + bhm_i ginv + chm_i$$

(2) Domestically produced goods(only b) and c) used in the program)

a) $x_i = \sum_{j=1}^{n} ah_{i,j}x_j + bh_i ginv + ch_i + z_i = hus_i sh_i + z_i$

Equations 2a, 7, and 18 yield:

b) $x_i = \sum_{j=1}^{n} (ah_{i,j} + bh_i + bh_i repl_j kk_j)x_j + fdh_i$

c) $fdh_i = bh_i ninv + ch_i + z_i$

(3) Dollar competitive import goods (first eqn. used)

$$md_i = hus_i sd_i = \sum_{j=1}^{n} ad_{i,j}x_j + bd_i ginv + cd_i$$

(4) Rouble competitive import goods (first eqn. used)

$$mr_i = hus_i sr_i = \sum_{j=1}^{n} ar_{i,j}x_j + br_i ginv + cr_i$$

(5) Noncompetitive import goods

$$mb_i = \sum_{j=1}^{n} amb_{i,j}x_j + bmb_i ginv + cmb_i$$

(6) Labor

$$lt = \sum_{j=1}^{n} l_j , \qquad \text{where} \quad l_j = lk_j x_j$$

(7) Capital

$$kt = \sum_{j=1}^{n} k_j , \qquad \text{where} \quad k_j = kk_j x_j$$

(8) Dollar trade balance

$$btd = \sum_{i=1}^{n} (pzd_i zd_i / zdef_i - pwmd_i md_i - pwmbd_i mbd_i)$$

(9) Rouble trade balance

$$btr = \sum_{i=1}^{n} (pwzr_i zr_i / zref_i - pwmr_i mr_i - mbr_i pwmbr_i)$$

Consumption, Investment, and Related Variables

(10) Value of variable national income

$vni = bvni\, pvn$ if $kobj \neq 1$ and $kinv \neq 2$

$vni = pc\, ccv + pinv\, vninv$ if $kobj = 1$ or $kinv = 2$

(11) Total consumption expenditure

$et = bet\, gpl$ if $kobj = 1$

$et = ev + \sum_{i=1}^{n} pchm_i chmf_i + pcmb_i cmbf_i$ if $kobj \neq 1$

(12) Variable consumption expenditure

$ev = et - \sum_{j=1}^{n} (pchm_i chmf_i + pcmb_i cmbf_i)$ if $kobj = 1$

$ev = pc\, ccv$ if $kobj \neq 1$

(13) Global level of variable consumption

$ccv = ev / pc$ if $kobj = 1$

$ccv = bccv$ if $kobj \neq 1$ and $kinv = 2$

$ccv = (1 - rinv) pc^{-elci} pvn^{elci-1} vni$ otherwise

(14) Sectoral level of variable consumption

a) $chmv_i = ckhm_i pchm_i^{-elco} pc^{elco-1} ev$

b) $cmbv_i = ckmb_i pcmb_i^{-elco} pc^{elco-1} ev$

(15) Sectoral level of total consumption

a) $chm_i = chmf_i + chmv_i$

b) $cmb_i = cmbf_i + cmbv_i$

(16) Components of the consumption of composite goods (only a) appears in the program)

a) $ch_i = sh_i chm_i$

b) $cd_i = sd_i chm_i$

c) $cr_i = sr_i chm_i$

(17) Variable and global levels of net investment

a) variable

$vninv = rinco\, (pc / pinv)^{elci} ccv$ if $kinv = 1$ or 3

$vninv = 0$ if $kinv = 2$

b) global

$ninv = rinco\, cft + vninv$ if $kinv = 1$ or 3

$ninv =$ constant if $kinv = 2$

where $rinco = rinv / (1 - rinv)$

(18) Level of gross investment

$$ginv = \sum_{j=1}^{n} repl_j k_j + ninv$$

(19) Implicit welfare level

$wf = ccv$ if $kinv = 2$

$wf = vni/pvn$ if $kinv \neq 2$

Export and Related Variables

(20) Rouble export

$zr_i = zr0_i(p_i/txzr_i pwzr_i vr)^{zrel_i}$,

where $zrel_i = 0$ if $krtr \geq 2$

(21) Dollar export

$zd_i = zd0_i(p_i/txzd_i vd\, pwzd_i)^{zdel_i}$,

where

$zdel_i = zdels_i$ if $kdex = 2$

$zdel_i = zdeld_i$ if $kdex = 1$ or 4

$zdel_i = zdeld_i zdels_i/(zdeld_i + zdels_i)$ if $kdex = 3$

(22) Total sectoral export

$z_i = zr_i + zd_i$

(23) Optimal tariffs (only if kdex=4)

a) $txzd_i = (1+zdeld_i)/zdeld_i$

b) $ztar_i = txzd_i(zd0_i/zd_i)^{1/zdels_i}$

Import Ratios, Shares, and Volumes

(24) Ratio of competitive rouble import to home supply

$mrr_i = mrr0_i(p_i/pmr_i)^{mrel_i}$ if $krtr = 1$

$mrr_i = sr_i/sh_i$ if $krtr \neq 1$

(25) Ratio of competitive dollar import to home supply

$mdr_i = mdr0_i(p_i/pmd_i)^{mdel_i}$

(26) Home production share in total variable home use

$sh_i = 1/(1+mdr_i+mrr_i)$ if $kdim = 1$

$sh_i = (1-sr_i)(hm_i mdr_i^{-eta_i} + hh_i)^{1/eta_i}$ if $kdim = 2$

(27) Competitive dollar import share in total home use

$sd_i = sh_i mdr_i$

(28) Competitive rouble import share in total home use

a) $sr_i = sh_i mrr_i$ if $krtr = 1$

b) $sr_i = (mr_i - crf_i) / hus_i$ if $krtr \neq 1$

(29) Rouble share in total noncompetitive imports

a) $mbrs_i = mbrs0_i (pmbd_i / pmbr_i)^{mbrel_i}$ if $krtr = 1$

b) $mbrs_i = \min(mbr_i / mb_i\, ; 1) = mbrsu_i$ if $krtr \neq 1$

(30) Noncompetitive rouble import

$mbr_i = mbrs_i mb_i$ if $krtr = 1$

$mbr_i = constant$ if $krtr \neq 1$

(31) Noncompetitive dollar import

$mbd_i = (1 - mbrs_i) mb_i$ if $krtr = 1$

$mbd_i = \max(0; mb_i - mbr_i)$ if $krtr \neq 1$

Variable Input Coefficients

(32) Input–output coefficients for competitive goods (only a) used in the program)

a) $ah_{ij} = ahm_{ij} sh_i$

b) $ad_{ij} = ahm_{ij} sd_i$

c) $ar_{ij} = ahm_{ij} sr_i$

(33) Investment input coefficients for competitive goods (only a) used in the program)

a) $bh_i = bhm_i sh_i$

b) $bd_i = bhm_i sd_i$

c) $br_i = bhm_i sr_i$

(34) Labor input coefficients

$lk_i = (s_i kszi_i / blk_i w_i)^{sigma_i} blk_i$

(35) Capital input coefficients

$kk_i = (s_i (1 - kszi_i) / bkk_i q_i)^{sigma_i} bkk_i$

Costs and Prices

(36) Sectoral wage rate

$$w_i = (1+wg)wd_i$$

(37) Sectoral user's cost of capital

$$q_i = (amr_i + knet_i r)pinv$$

(38) Sectoral joint cost of labor and capital

$$s_i = (w_i blk_i / kszi_i)^{kszi_i}(q_i bkk_i/(1-kszi_i))^{1-kszi_i} \qquad \text{if } kpft = 1 \text{ (C-D)}$$

$$s_i = \left\{ kszi_i^{sigma_i}(blk_i w_i)^{beta_i sigma_i} + (1-kszi_i)^{sigma_i}(bkk_i q_i)^{beta_i sigma_i} \right\}^{1/(beta_i sigma_i)} \qquad \text{if } kpft = 2 \text{ (CES)}$$

(39) Price index of competitive rouble import

$$pmr_i = txmr_i pwmr_i vr \qquad \text{if } krtr \neq 4$$

$$pmr_i = phm_i \qquad \text{if } krtr = 4$$

($vr = 1$ if $krtr = 2$, $vr = vd$ if $krtr = 3$)

(40) Price index of competitive dollar import

$$pmd_i = txmd_i pwmd_i vd$$

(41) Average price index of sectoral composite goods

$$phm_i = sh_i p_i + sd_i pmd_i + sr_i pmr_i \qquad \text{if } krtr \neq 4$$

$$phm_i = (sh_i p_i + sd_i pmd_i)/(1-sr_i) \qquad \text{if } krtr = 4$$

(42) Price index of noncompetitive rouble import

$$pmbr_i = txmbr_i pwmbr_i vr \qquad \text{if } krtr \neq 4$$

$$pmbr_i = pmbd_i \qquad \text{if } krtr = 4$$

($vr = 1$ if $krtr = 2$, $vr = vd$ if $krtr = 3$)

(43) Price index of noncompetitive dollar import

$$pmbd_i = txmbd_i pwmbd_i vd$$

(44) Average price index of noncompetitive import

$$pmb_i = mbrs_i pmbr_i + (1-mbrs_i)pmbd_i$$

(45) Producers' price index of home products

$$p_i = \left\{ \sum_{j=1}^{n} (phm_j ahm_{ji} + pmb_j amb_{ji}) + s_i \right\} (1+profr_i)txh_i \qquad \text{if } kpri = 1,2,4 \text{ or } 6$$

(where $profr_i = 0$ if $kpri = 1$)

$$p_i = \left\{ \sum_{j=1}^{n} (phm_j ahm_{ji} + pmb_j amb_{ji}) + s_i + profc_i \right\} txh_i \qquad \text{if } kpri = 3 \text{ or } 5$$

(46) Price index of consumption

a) $pchm_i = txchm_i phm_i$

b) $pcmb_i = txcmb_i pmb_i$

(47) Global price index of variable consumption

$$pc = \prod_{i=1}^{n} pchm_i^{ckhm_i} pcmb_i^{ckmb_i} \qquad \text{if } kcof = 1$$

$$pc = \left\{ \sum_{i=1}^{n} (ckhm_i pchm_i^{1-elco} + ckmb_i pcmb_i^{1-elco}) \right\}^{1/(1-elco)} \qquad \text{if } kcof \neq 1$$

(48) Average price index of investments

$$pinv = \sum_{j=1}^{n} (phm_j bhm_j + pmb_j bmb_j)$$

(49) Price index of variable national income

$$pvn = pc \qquad \text{if } kinv = 2$$

$$pvn = pc^{1-rinv} pinv^{rinv} \qquad \text{if } kinv \neq 2 \text{ and } elci = 1$$

$$pvn = \left\{ (1 - rinv) pc^{1-elci} + rinv\, pinv^{1-elci} \right\}^{1/(1-elci)} \qquad \text{otherwise}$$

(50) Price index of dollar exports

$$pzd_i = (zd_i / zd0_i)^{1/zdeld_i} pwzd_i \qquad \text{if } kdex \neq 2$$

$$pzd_i = pwzd_i \qquad \text{if } kdex = 2$$

(51) Price normalization rule

$$cft = \sum_{i=1}^{n} (pchm_i chmf_i + pcmb_i cmbf_i)$$

(Done at the end only.)

APPENDIX 3: EXAMPLE OF THE COMMUNICATION FACILITY IN THE INTERACTIVE VERSION

```
Reading the inputs

Do you want to see or change input? y/n : (y)

  Specification control variables

 1.kcof     2.kcoc     3.col     4.kpft    5.kpfc
 6.kobj     7.kinv     8.kpri    9.krtr    10.ktra
11.kdim     12.kcim    13.kdex

  Parameters that can be changed

 1.coinv   2.chmf    3.cbmf   4.ckhm    5.ckbm
 6.elco    7.profr   8.profc  9.kt      10.lt
11.sigma   12.mrel   13.mdel  14.mbrel  15.zrel
16.zdeld   17.zdels  18.btd   19.btr    20.pwzd
21.pwzr    22.pwmd   23.pwmr  24.pwmbd  25.pwmbr
26.txh     27.txmr   28.txmd  29.txmbr  30.txmbd
31.txzr    32.txzd

 1: Help! Tell what is what. (You may ask
   for any model variable or parameter .)
 2: See or change  some control variable.
 3: See or change some parameter.
99: Continue.
Enter selection : (1)

(Re)enter name of variable or parameter : (eg. kpri)
```

```
   kpri    price formation rule (1-6)
        1  'nonprofit'
        2  exogenous profit factor (multiplier)
        3  exogenous profit ratio  (additive)
        4  calculated profit factor (multiplier)
        5  calculated profit ratio  (additive)
        6  base profit factor (multiplier)

Appears in following equations:
   p(i) = (sum(phm(j)*ahm(j,i)+pmb(j)*abm(j,i))+s(i))
        *(1.+profr(i))*(1.+txh(i))
                      if(kpri.eq.1.or.2.or.4.or.6)
   p(i) = (sum(phm(j)*ahm(j,i)+pmb(j)*abm(j,i))+s(i)
        +profc(i))*(1.+txh(i))
                      if(kpri.eq.3.or.5)
   profr(i) = 0.          if(kpri.eq.1)

Any other one you want to see? y/n : (n)
```

```
 Specification control variables

 1.kcof    2.kcoc    3.col    4.kpft    5.kpfc
 6.kobj    7.kinv    8.kpri    9.krtr    10.ktra
 11.kdim    12.kcim    13.kdex

 Parameters that can be changed

 1.coinv   2.chmf    3.cbmf   4.ckhm    5.ckbm
 6.elco    7.profr   8.profc   9.kt     10.lt
 11.sigma   12.mrel    13.mdel   14.mbrel  15.zrel
 16.zdeld   17.zdels   18.btd    19.btr    20.pwzd
 21.pwzr    22.pwmd    23.pwmr   24.pwmbd 25.pwmbr
 26.txh     27.txmr    28.txmd   29.txmbr 30.txmbd
 31.txzr    32.txzd

  1: Help! Tell what is what. (You may ask
     for any model variable or parameter .)
  2: See or change  some control variable.
  3: See or change some parameter.
 99: Continue.
 Enter selection : (2)


 Specification control variables

 1.kcof    2.kcoc    3.col    4.kpft    5.kpfc
 6.kobj    7.kinv    8.kpri    9.krtr    10.ktra
 11.kdim    12.kcim    13.kdex

 Current values

 kcof   kcoc   kinv   kpri   kpft   kpfc
   3      2      1      1      2      2
```

```
krtr   kobj   ktra   kcim   kdim   kdex   col
 -1      1      1      2      2      2   1.00



Enter the number of control variable to be changed or 99 to exit: (eg. 1)


   kcof    measure of excess consumption ('welfare') level (1-3)
      1  Cobb-Douglas function (LES)
      2  Kantorovits-Leontief function (fixed structure)
      3  CES function

Current value of kcof         3

Change it or not? y/n : (n)
```

```
 Specification control variables

1.kcof    2.kcoc    3.col    4.kpft    5.kpfc
6.kobj    7.kinv    8.kpri   9.krtr   10.ktra
11.kdim     12.kcim    13.kdex

 Current values

kcof  kcoc  kinv  kpri  kpft  kpfc
  3     2     1     1     2     2


krtr  kobj  ktra  kcim  kdim  kdex  col
  1     1     1     2     2     2  1.00



Enter the number of control variable to be changed or 99 to exit: (99)



 Specification control variables

1.kcof    2.kcoc    3.col    4.kpft    5.kpfc
6.kobj    7.kinv    8.kpri   9.krtr   10.ktra
11.kdim     12.kcim    13.kdex

 Parameters that can be changed

1.coinv   2.chmf    3.cbmf   4.ckhm    5.ckbm
6.elco    7.profr   8.profc  9.kt      10.lt
11.sigma   12.mrel   13.mdel   14.mbrel  15.zrel
16.zdeld  17.zdels  18.btd    19.btr    20.pwzd
21.pwzr   22.pwmd   23.pwmr   24.pwmbd  25.pwmbr
```

```
26.txh    27.txmr    28.txmd   29.txmbr  30.txmbd
31.txzr   32.txzd

 1: Help! Tell what is what. (You may ask
    for any model variable or parameter .)
 2: See or change  some control variable.
 3: See or change some parameter.
99: Continue.
Enter selection : (3)
```

```
 Parameters that can be changed

 1.coinv   2.chmf    3.cbmf    4.ckhm    5.ckbm
 6.elco    7.profr   8.profc   9.kt      10.lt
11.sigma   12.mrel   13.mdel   14.mbrel  15.zrel
16.zdeld   17.zdels  18.btd    19.btr    20.pwzd
21.pwzr    22.pwmd   23.pwmr   24.pwmbd  25.pwmbr
26.txh     27.txmr   28.txmd   29.txmbr  30.txmbd
31.txzr    32.txzd

Enter the number of parameter to be changed or 99 to exit: (eg. 16)

   zdeld(j) demand elasticity of the dollar export, sectoral commodity j

 Appears in following equations:
        zdel(i) = zdeld(i)                    if(kdex.eq.1.or.4)
        zdel(i) = zdeld(i)*zdels(i00)/(zdeld(i)+zdels(i))
        txzd(i) = (1.+zdeld(i))/zdeld(i)              if(kdex.eq.4)
        pzd(i) = (zd(i)/zd0(i))**(1/zdeld(i))*pwzd(i)  if(kdex.ne.2)

 Current values :

-4.50000 -6.00000 -5.50000 -5.00000 -4.50000
-6.00000 -4.50000 -4.50000 -3.50000 -30.0000

You need the list of sectors? y/n : (y)

 1.Mining and electricity        2.Metallurgy and machinery
 3.Chemicals                 4.Food processing
 5.Light and other ind.          6.Construction, -materials
 7.Agriculture and forestry      8.Productive infrastructure
 9.Trade                   10.Nonproductive infrastruct.

Change any of these values? y/n : (y)
```

```
Multiply uniformly in each sector, y/n: (n)
(Re)enter sector number and new value : (etc ...)

Enter selection : (99)

Continue with calc
Do you want to see or change input? y/n : (n)
Change the number of next run if you wish (1-9) or write 99 to continue: (99)

Main iteration, No. 1  (etc ...)
```

```
Done run no. (1)

Do you want to see
 1: Detailed results for individual runs
 2: Summary results
99: END
Enter selection : (1)

   0: Display TITLES
1-18: Display the TABLE of selected number
  99: RETURN to the upper level of selection menu
Enter selection : (0)

 1: Export
 2: Prices related to export
 3: Dollar import rates and Rouble import rates
 4: Competitive imports
 5: Noncompetitive imports
 6: Total consumption
 7: Variable consumption
 8: Primary factor allocation and production
 9: Factor coefficient and costs
10: Prices before rescaling (W,Q,S,Prof)
11: Prices before rescaling (P,Pmr,Pmd,Pa)
12: Prices before rescaling (P(n+1),Vr,Vd,W,R,1+W,etc)
13: Shadow (calculative) prices (W,Q,S,Prof)
14: Shadow (calculative) prices (P,Pmr,Pmd,Pa)
15: Shadow (calculative) prices (P(n+1),Vr,Vd,W,R,etc)
16: Main indicators
```

```
   0: Display TITLES
 1-18: Display the TABLE of selected number
  99: RETURN to the upper level of selection menu
Enter selection : (99)

  1: Detailed results for each run
  2: Summary results
 99: END
Enter selection : (2)

   0: Display TITLES
 1-16: Display the TABLE of selected number
  99: RETURN to the upper level of selection menu
Enter selection : (0)

  1: Dollar export
  2: Rouble export
  3: Dollar competitive import
  4: Rouble competitive import
  5: Dollar nocompetitive import
  6: Rouble nocompetitive import
  7: Variable consumption
  8: Production
  9: Labor employed
 10: Capital used
 11: Capital/labor ratio
 12: Producers price indices
 13: Domestic prod.price/dollar exchange rate
 14: Dollar export price indices
 15: Main indicators (at base prices)


   0: Display TITLES
1-16: Display the TABLE of selected number
```

```
 99: RETURN to the upper level of selection menu
Enter selection : (99)

  1: Detailed results for each run
  2: Summary results
 99: END
Enter selection : (99)

Do you want to have another run ? y/n (n)
```