International Institute for
Applied Systems Analysis
www.iiasa.ac.at

PC-POSTAN Version 3.0. Postoptimal Analysis Package. User's Manual

Rys, T.

IIASA Working Paper

December 1988

WORKING PAPER

PC-POSTAN VERSION 3.0
POSTOPTIMAL ANALYSIS PACKAGE
USER'S MANUAL

T. Rys

IIASA

International Institute
for Applied Systems Analysis

PC-POSTAN VERSION 3.0
POSTOPTIMAL ANALYSIS PACKAGE
USER'S MANUAL

T. Rys

December 1988
WP-88-119

Foreword

This paper is one of the series of 11 Working Papers presenting the software for interactive decision support and software tools for developing decision support systems. These products constitute the outcome of the contracted study agreement between the System and Decision Sciences Program at IIASA and several Polish scientific institutions. The theoretical part of these results is presented in the IIASA Working Paper WP-88-071 entitled *Theory, Software and Testing Examples in Decision Support Systems* which contains the theoretical and methodological bacgrounds of the software systems developed within the project.

This paper presents the PC–POSTAN package. This package constitutes a tool for postoptimal analysis for linear and linear-fractional programming problems. Contrary to the POSTAN 3 package it has been designed to be used on the IBM-PC computers running under the MS–DOS operating system. This is an interactive package which retains all options of the POSTAN 3 and adds some new features. PC–POSTAN consists of a number of FORTRAN routines which are incorporated into MINOS, the well known linear and nonlinear programming code developed at the Stanford University. The postoptimal analysis is performed after MINOS has found an optimal solution and is initiated by extending the original MINOS specification file. The main function of PC–POSTAN is ranging with respect to parameters specified by the user and computing the sensitivity coefficients. The user can control the program interactively, using hierarchical menu system. It is possible to perform series of experiments, selective inspection of results using the window interface, and modification of model data. For details regarding theoretical backgrounds of this package, the user is advised to consult the POSTAN 3 manual.

Alexander B. Kurzhanski
Chairman
System and Decision Sciences Program

PC–POSTAN Ver. 3.0.
Postoptimal Analysis Package.
User's Manual

Tomasz Ryś

Joint Systems Research Department,
Academy of Mining and Metallurgy, Cracow,
Industrial Chemistry Research Institute, Warsaw.

Contents

1 Introduction

PC–POSTAN is a PC–software implementation of the postoptimal analysis package named POSTAN–3 elaborated by G. Dobrowolski, K. Hajduk, A. Korytowski and T. Ryś (G. Dobrowolski at. al. 1983, 1984, 1985, 1988). It can be used on IBM-PC type computers under MS–DOS operating system. The PC–POSTAN is an interactive package which not only retains all facilities of the batch version (POSTAN–3) but has also some new additional options.

User controls PC–POSTAN interactively using hierarchical menu system. The program implementation enables to perform series of optimization experiments, selective review of results and modification of a model data. The output from PC–POSTAN is organized through the window system.

In the below documentation the theoretical part has been purposefully omitted since the mathematical algorithms remained unchanged with respect to those used in POSTAN–3.

PC–POSTAN as POSTAN–3 did is to be considered an extension of MINOS LP package (B. A. Mutagh and M. A. Saunders, 1977). It means that in the calculations, low level procedures from the MINOS library are utilized. For that reason description of certain parameters and details is omitted and appropriate references to MINOS documentation are given.

This documentation provides user's manual where detailed information on how to use PC–POSTAN is given. Hardware technical requirements, file formats and installation manual are presented in the appendixes.

2 How to start ?

The package is initialized using command **pcpost** entered on the MS–DOS operating system prompt level. In response two windows are displayed. The higher, horizontal one contains options of the main package menu, the lower, bigger one contains PC–POSTAN introductory screen. In the course of the interaction with the package in the higher level window menu options (which are in a sense higher hierarchically) or messages describing state of the package are displayed. In the case when with respect to the menu in question exists a sub–menu directly corresponding to it, the sub–menu is displayed in the new vertical window.

Option *Exit* on each menu passes control to the higher level menu. Only in the case of the main menu it terminates the work of the PC–POSTAN.

Option *Step_Back* passes control to the preceding menu, while option *Log* leads to display of so called *log file* (see section 4.1).

Particular menu option is picked up using arrow keys. Selected (active) option is highlighted and the selection should be confirmed by pressing RETURN key. At the beginning first option is always active. The arrow keys: right or down are used to move forward while left or up are used to move backward trough the menu options. First and last option can be also selected by using keys HOME and END respectively.

There is a possibility of direct option selection without necessity of confirmation (by RETURN key). It is done by pressing key with the first letter of the option name (regardless of the character case). Options are reviewed sequentially from the first to the last position and consequently if more then one of the options name starts from the same letter the first one is chosen. The use of the wrong key is signaled acoustically and the key is ignored.

The main PC–POSTAN menu consist of the following options:

- Help

1

- Parameters

- Load

- Exit

3 Help

Selection of this option leads to display a window containing the information explaining the use of a menu. Return to the main menu is triggered by pressing any key.

4 Parameters

This option enables user to set, modify or review the basic PC–POSTAN parameters. If selected, the sub–menu with following options is displayed:

- Files

- Mps input data

- Frequencies

- Lp parameters

- Miscellaneous

- Exit

The above options reflect a break down of the parameters into specific categories. Selection of any option displays a list of parameters. Values of those parameters depend on the actual package status: these are either default values assigned by the package or actual values set by the user. The parameter which can be changed is to be pointed by a cursor position. Change of the position is attained by using arrow keys (up or down). In order to change value of a parameter key INSERT should be pressed (on the screen comes out: **Enter New Value**) and then new value can be entered followed by the ENTER key. In the case of a formal error (e.g. wrong number format) clause **Error in field** is displayed and user is asked to enter new value once more. There is a possibility to enter empty value (e.g. discarding file name or a option) and in this case default value will be assumed. Return to the menu is obtained by pressing ESCAPE key.

In the description that follows parameters with the † sign have meaning exactly as they have in the MINOS package (B. A. Mutagh and M. A. Saunders, 1977).

4.1 Files

Using this category user may declare all input and output files. The file names must obey rules implied by MS–Dos.

Improper entering of the file name (e.g. name of non–existing file) causes particular consequences in the process of problem loading (see option *Load* further on in the documentation). The names of the following files can be declared:

- mps file (input file) †. Default name mps.

- old basis file (map) †

- solution file †

- new basis file (map) †

- insert file †

- backup basis file †

- punch file †

- load file †

- dump file †

- **data** ranging file. This file contains data for directional type of postoptimal analysis. The file format is described in the appendix A.

- **dictionary** file. File which enables for generating reports using real–life description of the variables. The file format and report programming rules are described in the appendix B (see also sec. 6).

- **report** file. The file name on which generated reports containing results of a particular optimization runs will be (on request) stored (see sec. 6.).

- **modification** file. File containing directives modifying loaded problem. Because PC– POSTAN enables for modification to be entered interactively, therefore the *modification file* should be used either when list of modification is very long or it can be generated by another program. This option was left also to sustain compatibility with POSTAN–3. The file format is described in the appendix C.

- **postan** log file. Default name **postan.log**. This file contains all information about optimization runs, possible errors that took place. modification performed and if user wishes so results of optimization.

4.2 Mps input data

This category of data is used to declare some parameters of LP–model stored in MPS file format. There are:

- **objective** name. MPS code of the criterion row.

- **denominator** name. MPS code of the denominator row in the case of fractional–type objective function. *Objective name* comes to be a numerator when this row is declared.

- **rhs** vector name †

- **bounds** vector name †

- **ranges** vector name †

- **row** limit †

- **list** limit †

- lower bound limit †

- column limit †

- error message limit †

- upper bound limit †

- elements limit †

- aij tolerance †

4.3 Frequencies

In this category of data user declares following frequencies of particular action of PC–POSTAN (MINOS):

- log iteration †

- check row error †

- cycle limit †

- save new basis map †

- factorize (invert) †

- cycle tolerance †

- cycle print †

4.4 Lp parameters

This category of data enables user to declare parameters directly related to optimization algorithm, that is:

- optimization direction. Admissible values are min (for minimizing, default) or max (for maximizing).

- iteration limit †

- feasibility tolerance †

- partial price factor †

- crash option †

- dj tolerance †

- multiple price †

- weight on objective †

- pivot tolerance †

4.5 Miscellaneous

For this category it is possible to declare following parameters:

- lu row tolerance †

- print level †

- imbed †. Admissible values: **yes** or **no**.

- lu col tolerance †

- debug level †

- print spikes †. Admissible values: **yes** or **no**.

- lu mod tolerance †

- unbounded objective value †

- unbounded step size †

- solution (log file). Defines a way of results reporting to the *log file*. Admissible values: **yes** †, **no** † or **default** †.

5 Load

This option should be utilized after making sure, that all parameters were adequately selected and set for the problem user wishes to solve.

Activation of this option loads the problem into memory. In the process of loading following activities are performed:

- declared files are opened. If any of the files which should exist had not been found a window is displayed which is to be used for entering proper file name. Entering empty name eliminates declaration of the file. This operation cannot be performed with respect to the MPS file, which must be properly declared.

- MPS file reading and analyzing. In case of errors in the MPS file user gets informed via error window. Warnings are reported only on the *log file*.

Proper loading is terminated by display in the main window of an information about problem size. The control is passed to the next, horizontal menu which contains following elements:

- Modify_Problem

- Run_Problem

- Log

- Step_Back

6 Modify_Problem

The option enables for change (modification) of the loaded model data. This operation can be performed interactively or via directives contained in the external file. The latter possibility can be utilized only when *modification file* was declared. In the case of selection of this option another menu is displayed:

- Interactive

- From_File

- Log

- Step_Back

Selection of the option *From_File* leads to reading in a deck containing modifications from the previously declared *modification file* (see appendix C for file format information). When some errors are detected in the process of performing modifications, relevant information is displayed in the error window and additional remarks are also stored in the *log file* which in turn can be reviewed by using option *Log*.

In case of successful completion of modifications control is switched automatically to the previous menu.

Modification *From_File* can be performed until end of file is reached. The use of this option each time implies sequential reading of one modification deck.

When *modification file* was not declared or if option *Interactive* was selected control is switched to the menu as follows:

- Obj

- Den

- Rows

- Columns

- Rhs

- Ranges

- Bounds

- View

- Exit

Items of this menu refer to sections in terms of MPS file, which are to be modified. It must be underlined however that execution of modifications does not change the original MPS file contents.

Selection of any of the options leads to preparation of modification commands. This commands are in the format similar to the MPS file records for its corresponding sections. Format (record fields) appropriate for the command is displayed on the screen before reading in the command. Entering empty first field in a given command ends modifications of the corresponding section. Formal errors in a command (e.g. wrong type of a row, wrong number format) are signaled in the same line as command in question and user is forced to enter a

6

proper value. Other types of errors (e.g. nonexisting name) are shown in the line bellow command line.

Each command is recorded also in the *log file*. In some cases previous values of elements (to be modified) are shown also on the screen. The command field is filled in only after pressing the ENTER key and when the field is formally correct, then package waits for the completion of the next field. When the last field is complete the next command is to be read in.

In the bellow description by the term *name* the MPS code is understood.

6.1 Obj

This option enables for change of the objective row and/or optimization direction. Its use is equivalent to changes that are performed through the option *Parameters* from the main menu. However in the case of *Obj* it is not necessary to reload a problem, therefore changes are executed much faster.
Fields of the command:

- *name* – name of the objective row

- *dir* – optimization direction. Admissible values **min** (for minimizing) or **max** (for maximizing)

The user is asked to accept or reject the realized modification after entering this command. Pressing the ESCAPE key means acceptance, while pressing any other key means rejection.

Type of the declared row is changed for N (free) and its right hand side (if exists) is canceled.

When there are no errors package displays previous criterion and control goes back to the menu.

6.2 Den

Through this option user can swiftly execute changes related to type of optimization and objective denominator.
Fields of the command are as follows:

- *name* – name of the row which is to be or actually is a denominator

- *on/off* – *switching on* or *switching off* denominator. Using "on" causes:

  - change of the optimization type for the *fractional* one
  - the objective row (*Obj*) becomes a numerator and the declared row becomes denominator of the criterion function. Type of row is set for N and its right hand side if exists is canceled.

  Using "**off**" causes:

  - change of the optimization type for the linear one
  - type of declared row is set as type free.

Similar to the previous option, after entering the command user is asked to accept it and previous denominator is displayed.

7

6.3 Rows

This option enables for change of the rows type. Right hand sides remain unchanged.
Fields of the command:

- *tp* – new type of a row. Admissible values: **l**, **g**, **e**, **n** as in MPS file

- *r–name* – name of a row.

After entering one command user may continue with the next or may terminate by entering empty first field.

6.4 Columns

This option enables for change of value of any LP–matrix element.
Fields of the command:

- *c–name* – name of a column

- *r–name* – name of a row

- *value* – new value of an element identified by the couple (c–name, r–name).

Modified may be only existing element. In the case of no error the previous value of the element is displayed on the screen.

6.5 Rhs

This option enables to modify RHSs declared for the rows in the MPS file. First field of the command is displayed automatically and it contains actual *rhs vector name* (see sec. 4.2).
Fields of the command:

- *r–name* – name of a row

- *value* – new RHS value

In the case of no error previous value of RHS is displayed.

6.6 Ranges

This option enables to introduce the left hand side constrains for the rows declared in MPS file. The above action is performed for each row of L, G, E type regardless if the constraint was introduced or not in the basic problem. First field of the command is displayed automatically and contains *ranges vector name* (see sec. 4.2).
Fields of the command:

- *r–name* – name of a row

- *value* – new value

6.7 Bounds

This option provides possibility to set or change constraints on the structural variables of the model.
Fields of the command:

- *tp* – type of the constraint. Admissible values: `up`, `lo`, `fx`, `mi`, `pl` as in MPS file.

- *c-name* – name of a variable

- *value* – value of the constraint. Filling this field is obligatory. In the case of `mi` and `pl` type it can be any value.

In the case of no errors previous value of constraints is displayed.

6.8 View

This option enables for examining of the MPS file contents.

7 Run_Problem

Selection of this option leads to solve the actually loaded problem. In the case of no errors resulting value of the objective function, number of performed iterations and cycle number are displayed. Then control is switched to the next menu which contains following options:

- Results

- Report

- Postan

- Rerun

- Log

- Step_Back

The option *Rerun* passes control to the menu presented in section 5 and option *Step_Back* to the main PC–POSTAN menu.

In the case of errors a message is displayed in the error window; in some instance an internal error number of MINOS is given (M. A. Saunders, 1977). Error information is also stored in the *log file*.

8 Results

This option enables for analysis of the optimization results of the problem currently solved. When picked up, option leads to its own menu:

- All

- Selected

- View_Mps

- Step_Back

8.1 All

This is used in order to review results for all model variables. The review is organized identically to reporting mode of MINOS (MPS style). In the PC–POSTAN presentation values of the variables and rows numbers are omitted. User may terminate review or may go to the next section by using ESCAPE key.

8.2 Selected

This option enables for selective review of the results. Its activation leads to display of a window which is used for entering MPS code of a row or a column. Entering of an empty code turns control back to the menu. This option presents results with greater accuracy then in the case of the previous option. Headings of particular values are the same as for MINOS.

8.3 View_Mps

This option is for MPS file reviewing.

9 Report

The Report option makes it possible to present the LP solution according to user's demands. It can be used only in the case when *dictionary file* is declared by *Parameters/Files* option (see sec. 4.1). It makes the printout of MINOS (MPS style) easier and more legible for the user. The option is especially convenient when package cooperates with the MPS file generator and a data base that usually contains description of LP model elements which may be used for reporting purposes.
The following functions are performed by this option:

- it gives the solution of LP problem described in natural language

- it searches for and outputs specified elements of the solution and formats them

- it formats printed numbers by re–scaling and rounding

- it formats the whole printout (display)

- it makes simple computations on the elements of solution possible and presents their results.

In the discussed version the functions mentioned above concern the primary solution data. This option brings in the next menu with the following items:

- Make_Report

- Edit_Dictionary

- View

- Step_Back

9.1 Make_Report

This option formats a report with results from the current optimization run. First of all the *dictionary file* is being analyzed. If some errors are detected package displays adequate message in the error window and also records it (in more detailed form) to the *log file*. Note that all records from *dictionary file* which are found as improper are ignored. Number of detected errors is limited by the *error message limit* option (see sec. 4.2).

After *dictionary file* analysis, formated report is displayed on the screen. It is assumed that records between two header lines (if any) of the *dictionary file* (see appendix B) compose a *report section* and therefore display comes in sequence *section after section*. This review can be terminated by pressing the ESCAPE key or by reaching end of the report.

If the *report file* is declared thorough *Parameters/Files* option (sec. 4.1) package also puts prepared report on this file.

9.2 Edit_Dictionary

When *dictionary file* is very long the user can be confused by time consuming analysis of the report. He can also find that some report records contain information not useful. In such a case the *Edit_Dictionary* option should be used. It enables user to temporary eliminate records from the *dictionary file* (change them into comment lines).

After choosing this option, content of the *dictionary file* is displayed. The active (not comment lines) records are highlighted. The user can toggle status of a record from active to comment line by pressing ENTER key. The record in question is pointed out by the cursor position which can be changed with the help of arrow keys (down or up). User can also use PAGEDOWN key for fast moving thorough dictionary contents. Header lines (see appendix B) are always active and they status can not be changed. All of changes are temporary and they do not any affect *dictionary file*. It should be also mentioned that it is possible to change status of comment line originally placed in the file.

The process of changes is terminated by pressing the ESCAPE key, after that package prepares scratch report file.

9.3 View

This option enables user to examine *report file* contents. It can be useful for comparison of results from different optimization runs.

10 Postan

Postoptimal analysis of currently obtained optimization results can be performed by this option. It can be activated only in case of an **optimal solution**. When the option is selected control is switched to the next menu with items as follows:

- On_Screen

- To_File

- log

- Step_Back

The *To_File* option produces results in the form exactly the same as composed by POSTAN–3 package and places them to the *log file*. The *On_Screen* option displays results on the computer screen but due to the limitations of a screen size some information is missing as compared to *To_File* option.

Both of above options pass control to the following menu:

- Directional Cost Ranging

- Directional Bound Ranging

- Directional Rhs Ranging

- Cost Ranging

- Bound Ranging

- Rhs Ranging

- Matrix Element Ranging

- Matrix Row Ranging

- Matrix Column Ranging

- Exit

The above items represent type of the postoptimal analysis which can be performed for all model variables. After selection of an option appropriate results are written to the file or displayed on the screen. The user can terminate reviewing the screen by pressing the ESCAPE key. In some cases this key switches display to the next section of the results and then package asks for termination or continuation of the review.

For the *directional* and *matrix* types of analysis user must prepare adequate data on the *data ranging file*. If this file is not declared the window with corresponding error message is displayed and control is passed to the menu.

In the *fractional* optimization only *Cost Ranging* analysis can be performed. All others are not yet supported by the PC–POSTAN package.

In the bellow sections the output from the postoptimal analysis for all cases will be described. Labels marked by the **typewriter font** are related to the *On_Screen* option, while labels written in upper case letters are adequate for *To_file* option. Labels marked with † are related to both screen and file options. In such case word *print* means *display* on screen or *print* to a file accordingly.

10.1 Directional Cost Ranging

Directional Cost Ranging analysis performs directional ranging on the costs. For a given increment of the cost vector determines the largest value such that for every cost the optimal solution is the same. The boundary cost components and some information on the change of state of variables at the boundary is also given.

Data: see appendix A.

The output is entitled DIRECTIONAL COST RANGING †. It takes one of two forms, depending on the value of range increment. If it is $< 10^{15}$, we have the finite range case, while if it is $> 10^{15}$ we have the infinite range case. Let us consider the finite range first.

12

In this case the sub-heading FINITE RANGE† is printed below the main title, with the corresponding value of range increment in brackets. Next, the following information is given for each structural variable:

NUMBER†	Number of structural variable
COLUMN†	Name of structural variable
DIRECTION†	Given increment component
OBJ GRADIENT†	Cost component
BOUNDARY VALUE†	Boundary value of cost component
M+J†	$m + 1 + i$ (as in Minos)

At the boundary either the optimal solution vanishes or one of the nonbasic variables changes its state. The name and original state of this variable are given in the last row of the output in the form: AT BOUNDARY VARIABLE *name* CEASES TO BE AT *bound* OR OPTIMAL (**opt**) SOLUTION VANISHES†. The letters LL are substituted for *bound* if the variable is no longer at its lower bound, while UL appears if the variable is no longer at its upper bound.

In the infinite range case the message INFINITE RANGE (TMAX.GE.1.E15)† is displayed. Beneath this the same information is given for each structural variable as in the finite range case, except for the BOUNDARY VALUE, which is no longer relevant.

10.2 Directional Bound Ranging

Directional Bound Ranging analysis performs directional ranging on the bounds. For a given vector of increments of the vector of bounds, this analysis determines two real numbers:

- related to the unchangeable the optimal solution

- related to the unchangeable the optimal basis.

The bound increments which correspond to fixed variables are automatically set to zero regardless of the values given in the data.
Data: see appendix A.
The output is entitled DIRECTIONAL BOUND RANGING (-- rows or -- columns). The first type information is given under the heading A. NO CHANGE IN OPTIMAL SOLUTION†. If calculated value is $< 10^{15}$ the message FINITE RANGE† is displayed, with the corresponding value in brackets. In the oposite case INFINITE RANGE (TMAXA .GE. 1.E15)† is printed. Similar information is given under heading B. NO CHANGE IN THE OPTIMAL BASIS†. The rest of the output is divided into two sections: SECTION 1 - ROWS and SECTION 2 - COLUMNS.

SECTION 1 - ROWS contains the following information for each slack variable (or for each row), except for the slack variable which corresponds to the objective row:

NUMBER	Number of slack variable
ROW†	Name of row
LL DIRECTION (ll direc)	Component of the lower bound increment vector
LL BOUNDARY A (ll bound a)	Boundary value of the lower bound; this is printed only if calculated value is $< 10^{15}$
LL BOUNDARY B (ll bound b)	Boundary value of the lower bound; this is printed only if calculated value is $< 10^{15}$

13

UL DIRECTION (ul direc)	Component of the upper bound increment vector
UL BOUNDARY A (ul bound a)	Boundary value of the upper bound; this is printed only if calculated value is $< 10^{15}$
UL BOUNDARY B (ul bound b)	Boundary value of the upper bound; this is printed only if calculated value is $< 10^{15}$
I	Row number

SECTION 2 – COLUMNS contains information analogous to that described above for each structural variable, with the following exceptions:

NUMBER	Number of structural variable
COLUMN†	Name of structural variable
M+J	$m + 1 + i$ (as in MINOS)

The last two rows of the output contain information on the change of state of variables at the boundaries. If calculated value is $< 10^{15}$, the message AT THE BOUNDARY A THE VARIABLE name HITS bound† is displayed. The letters LL are substituted for bound if the variable hits its lower bound and letters UL if it hits the upper bound. In the opposite case the message AT THE BOUNDARY B THE BASIC VARIABLE name BECOMES NONBASIC AT bound OR OPTIMAL SOLUTION VANISHES (or opt sol vanish) is displayed in the next row. Once again, LL is used to denote the lower bound and UL the upper bound.

10.3 Directional Rhs Ranging

Directional Rhs Ranging analysis performs directional ranging on the right-hand sides. For a given vector of increments of the rhs vector Directional Rhs Ranging analysis determines the largest real value such that for every rhs the optimal basis is the same
Data: see appendix A.
The output is entitled DIRECTIONAL RHS RANGING†. It takes one of two forms, depending on the value of calculated number. If it is $< 10^{15}$, one has the finite range case, while if it is $\geq 10^{15}$ it is the infinite range case. Let us consider the finite range first.

In this case the sub-heading FINITE RANGE† is printed below the main title, with the corresponding value in brackets. Next, the following information is given for each row (or each slack variable), except for the objective row:

NUMBER†	Number of slack variable
ROW†	Name of row
DIRECTION†	Component of the increment vector
RHS†	Right-hand side component
BOUNDARY VALUE†	Boundary value of the rhs component
I†	Row number

At the boundary either the optimal solution vanishes or one of the basic variables changes its state. The name and type of change are given in the last row of the output in the form: AT THE BOUNDARY VARIABLE name PASSES FROM THE BASIS TO bound OR OPTIMAL SOLUTION VANISHES†. The letters LL are substituted for bound if the variable reaches its lower bound and UL if it reaches its upper bound.

14

In the case of the infinite range the message INFINITE RANGE (TMAX.GE.1.E15)† is displayed. Below the same information is given for each non-objective row as in the case of finite range, except for the BOUNDARY VALUE, which is no longer relevant.

10.4 Cost Ranging

Cost Ranging analysis performs ordinary ranging on the costs. For each cost component the procedure determines the largest range in which cost coefficient may vary without affecting the optimal solution. Cost Ranging analysis also gives some information on the change of state of variables at the boundaries.

This analysis does not require any input data and can be used for linear and fractional optimization.

The output is entitled COST RANGING (Analysis). The following information is then given for each cost component:

NUMBER	Number of structural variable
COLUMN†	Name of structural variable
OBJ GRADIENT	Cost component
LOWER LIMIT†	Lower boundary of the range in which the cost component may vary without affecting the optimal solution
UPPER LIMIT†	Upper boundary of this range
CHANGE AT LOWER LIMIT (OR OPTSOL VANISHES)†	Name of the nonbasic variable which changes its state at the lower boundary; this is printed only if the lower boundary is finite. (Beware: Cost Ranging analysis does not know if there is an optimal solution beyond the boundary so that the name of a nonbasic variable may be printed even if the optimal solution vanishes)
CHANGE AT UPPER LIMIT (OR OPTSOL VANISHES)†	Name of the nonbasic variable which changes its state at the upper boundary (other explanations as above)
M+J	NUMBER $+m+1$ (as in MINOS)

10.5 Bound Ranging

Bound Ranging analysis performs ordinary ranging on the bounds. For each lower bound and for each upper bound the analysis determines two ranges: range A, which is the maximum range in which the bound may vary without affecting the optimal solution, and range B, which is the maximum range in which the bound may vary without affecting the optimal basis. While these ranges are being determined for particular bound, all other bounds remain fixed at their original values. Bound Ranging analysis also gives some information on the change of state of variables at the boundaries. This analysis is not performed for fixed variables. This analysis does not require any input data.

The output is entitled BOUND RANGING (-- No Solution Change or -- No Basis Change). It is divided into two parts, A and B, which will now be discussed separately.

Part A

Part A is entitled A. NO SOLUTION CHANGE and is divided into two subsections, SECTION 1 – ROWS† and SECTION 2 – COLUMNS†, which correspond to the sections of the

same name in the final output of MINOS .

SECTION 1 – ROWS contains the following information for each slack variable (or for each row constraint), except for the slack variable which corresponds to the objective row. In the first two columns there is:

NUMBER   Number of slack variable
ROW†   Name of row

If upper bound is equal to lower bound for the slack variable under consideration, the remaining columns contain only the message FIXED VARIABLE†.

     In the case when the slack variable is nonbasic at its lower bound the message VARIABLE AT LOWER BOUND† appears in the next two columns, which otherwise contain:

LL FOR L BOUND†   Lower boundary of range A
UL FOR L BOUND†   Upper boundary of range A

The next two columns give similar information about the upper bound. In other words, if the slack variable is nonbasic at its upper bound, the message VARIABLE AT UPPER BOUND† is printed; if it is not so the columns contain:

LL FOR U BOUND†   Lower boundary of range A
UL FOR U BOUND†   Upper boundary of range A
The last column contains:
I   Row number

SECTION 2 – COLUMNS contains information analogous to that described above for each structural variable. All the information delivered may be interpreted in the same way as in SECTION 1 – ROWS, with the following exceptions:

NUMBER   Number of structural variable
COLUMN†   Name of structural variable
M+J $m + 1 + i$ (as in MINOS)

Part B

Part B is entitled B. NO BASIS CHANGE†. It is also divided into two subsections, SECTION 1 – ROWS and SECTION 2 – COLUMNS.

SECTION 1 – ROWS contains the following information for each slack variable , except for the slack variable which corresponds to the objective row. The first two columns contain:

NUMBER   Number of slack variable
ROW†   Name of row

If upper bound is equal to lower bound for the slack variable under consideration, the remaining columns contain only the message FIXED VARIABLE†.

     In the case when the slack variable is nonbasic at its lower bound the message VARIABLE AT LOWER BOUND† appears in the next two columns, which otherwise contain:

LL FOR L BOUND (ll fl bnd)   Lower boundary of range B
UL FOR L BOUND (ul fl bnd)   Upper boundary of range B

The next two columns give similar information about the upper bound. In other words, if the slack variable is nonbasic at its upper bound, the message VARIABLE AT UPPER BOUND† is printed; if it is not the case the columns contain:

LL FOR U BOUND (ll fu bnd)   Lower boundary of range B
UL FOR U BOUND (ul fu bnd)   Upper boundary of range B

The columns which follow all appear under the heading CHANGES AT BDRIES (OR OPT SOL VANISHES)†. These are used only for nonbasic slack variables, remaining blank for basic variables.

If the slack variable is at its lower bound then the columns contain the names of the basic variables which change their state at the boundaries of range B for this lower bound, given that the solution does not vanish. The message LL indicates that the variable has reached its lower bound, while UL shows that the upper bound has been reached. The first column, headed LOWER†, gives the name of the variable which changes its state at the lower boundary of range B; the second column, headed UPPER†, gives the name of the variable which changes its state at the upper boundary. The name of the variable may also appear under the heading UPPER†. It means that the set of feasible solutions is then empty beyond the boundary.

If the slack variable is at its upper bound, these columns contain the names of the basic variables which change their states at the boundaries of range B, given that the solution does not vanish. The messages LL and UL have the same meaning as above. The first column, headed LOWER†, gives the name of the variable which changes its state at the lower boundary of the range B; the second column, headed UPPER†, gives the name of the variable which changes its state at its the upper boundary. If the name of slack variable appears under the heading LOWER†, then the set of feasible solutions is empty beyond this boundary.

The last column contains:

I   Row number

SECTION 2 – COLUMNS contains information analogous to that described above for each structural variable. All the information delivered may be interpreted in the same way as in SECTION 1 – ROWS, with the following exceptions:

NUMBER   Number of structural variable
COLUMN†   Name of structural variable
M+J $m + 1 + i$ (as in MINOS)

Beware: in most cases Bound Ranging analysis does not know if there is an optimal solution beyond the boundaries of range B and so a variable name may be printed under CHANGES AT BDRIES (OR OPT SOL VANISHES)† even if the optimal solution vanishes. The question whether the optimal solution exists has a negative answer (in the negative) only if the name of the nonbasic variable appears in the appropriate column of the output.

10.6 Rhs Ranging

Rhs Ranging analysis performs ordinary ranging on the right-hand sides (rhs). For each component of the vector of right-hand sides (except for the objective row), this procedure determines the maximum range in which rhs may vary without affecting the optimal basis. Rhs Ranging analysis gives some information on the change of state of variables at the boundaries.

This analysis does not require any input data.

The output is entitled RHS RANGING (Analysis). The following information is then given for each rhs component:

NUMBER	$n + i + 1$ (as in MINOS)
ROW†	Name of row
RHS†	Right-hand-side component
LOWER LIMIT†	Lower boundary of the range in which the rhs component may vary without affecting the optimal basis
UPPER LIMIT†	Upper boundary of this range
CHANGE (chng) AT LOWER LIMIT (ll) (OR OPT SOL VANISHES)†	Name of the basic variable which becomes nonbasic at the lower boundary. LL is printed if this variable reaches its lower bound and UL if it reaches its upper bound; the name is printed only if the boundary is finite. (Beware: Rhs Ranging analysis does not know if there is an optimal solution beyond the boundary and so a variable name may be printed even if the optimal solution vanishes)
CHANGE (chng) AT UPPER LIMIT (ul) (OR OPT SOL VANISHES)†	Name of the basic variable which becomes nonbasic at the upper boundary (other explanations as above)
M+J	Number of row

10.7 Matrix Element Ranging

Matrix Element Ranging analysis performs ordinary ranging on the elements of the constraint matrix. For selected elements the analysis determines the largest open range in which element may vary without affecting the optimal basis or the state of nonbasic variables. The list of the selected elements is given in the data. While the range for particular element is being determined, all other elements, are fixed at their original values. Matrix Element Ranging analysis also gives the sensitivity of the optimal cost with respect to the elements. In addition, some information on the change of state of variables at the boundaries is also given.

Data: see appendix A.

The output is entitled MATRIX ELEMENT RANGING (Analysis). The following information is then printed for each of the selected elements:

NUMBER	Number of slack variable
ROW†	Name of row
NUMBER	Number of column
COLUMN†	Name of column

18

ELEM VALUE†	Value of the selected element of the constraint matrix
LOWER LIMIT†	Lower boundary of the range in which the element may vary without affecting the optimal basis or the state of nonbasic variables
UPPER LIMIT†	Upper boundary of this range
SENSITIVITY†	Sensitivity of optimal cost with respect to the selected element
CHANGE AT LL	Name of the variable which changes its state at the lower boundary. No name is printed if the boundary is infinite or the optimal basis becomes singular at the boundary. TO LL is printed if this variable becomes nonbasic at its lower bound, OFF LL is printed if it is nonbasic and leaves its lower bound. Similarly, TO UL and OFF UL are printed for the upper bound. (Beware: Matrix Element Ranging analysis does not know if there is an optimal solution beyond the boundary and so a variable name may be printed even if the optimal solution vanishes)
CHANGE AT UL	Name of the variable which changes its state at the upper boundary. Other explanations as above.

If the row number is the number of objective row, the following message is displayed:
XXX THE REQUIRED NUMBER OF ROW IS THE NUMBER OF OBJECTIVE ROW†
and no analysis is performed for this particular element.

10.8   Matrix Row Ranging

Matrix Row Ranging analysis performs directional ranging on the rows of the constraint matrix.
Data: see appendix A.
The output is entitled MATRIX ROW RANGING (Analysis). The following information is given in the first two rows of the output:

NUMBER†	Number of slack variable
ROW†	Name of row
TMIN†	Lower boundary of the range
TMAX†	Upper boundary of the range
SENSITIVITY†	Sensitivity of optimal cost

19

CHANGE AT LL†	Name of the variable which changes its state at the lower boundary. No name is printed if the boundary is infinite or the optimal basis becomes singular at the boundary. TO LL is printed if this variable becomes nonbasic at its lower bound, OFF LL is printed if it is nonbasic and leaves its lower bound. Similarly, TO UL and OFF UL are printed for the upper bound. (Beware: Matrix Row Ranging analysis does not know if there is an optimal solution beyond the boundary and so a variable name may be printed even if the optimal solution vanishes)
CHANGE AT UL†	Name of variable which changes its state at the upper boundary. Other explanations as above
I†	Number of row

Next, the following information is given for each structural variable:

NUMBER	Number of column
COLUMN†	Name of column
DIRECTION†	Increment component
ELEM VALUE†	Component of the constraint vector
LOWER LIMIT†	Lower boundary value of the component of the constraint vector
UPPER LIMIT†	Upper boundary value
M+J	$m + 1 + k$ (as in MINOS)

10.9 Matrix Column Ranging

Matrix Column Ranging analysis performs directional ranging on the structural columns of the constraint matrix.
Data: see appendix A.
The output is entitled MATRIX COLUMN RANGING (Analysis). The following information is given in the first two rows of the output:

NUMBER†	Number of column
COLUMN†	Name of column
TMIN†	Lower boundary of the range
TMAX†	Upper boundary of the range
SENSITIVITY†	Sensitivity of optimal cost

20

CHANGE AT LL†	Name of the variable which changes its state at the lower boundary. No name is printed if the boundary is infinite or the optimal basis becomes singular at the boundary. TO LL is printed if this variable becomes nonbasic at its lower bound, OFF LL is printed if it is nonbasic and leaves its lower bound. Similarly, TO UL and OFF UL are printed for the upper bound. (Beware: Matrix Column Ranging analysis does not know if there is an optimal solution beyond the boundary and so a variable name may be printed even if the optimal solution vanishes)
CHANGE AT UL†	Name of the variable which changes its state at the upper boundary. Other explanations as above
M+J†	$m + 1 + i$ (as in MINOS)

Next, the following information is given for each row (except for the objective row):

NUMBER	Number of slack variable
ROW†	Name of row
DIRECTION†	Increment
ELEM VALUE†	Component of the constraint column
LOWER LIMIT†	Lower boundary value of the component of the constraint column
UPPER LIMIT†	Upper boundary value
I	Number of row

11 References

Dobrowolski G., A. Korytowski, T. Ryś (1983). POSTAN: A Postoptimal Analysis Package for Linear Programming. *Computing at IIASA*, Vol 3, pp. 14-17.

Dobrowolski G., K. Hajduk, A. Korytowski, T. Ryś (1984). Postan - A Package for Postoptimal Analysis. (An Extension of Minos). IIASA Colaborative Paper CP–84–32.

Dobrowolski G., K. Hajduk, A. Korytowski, T. Ryś (1985). POSTAN 2 - An Extended Postoptimal Analysis Package for MINOS. Theory, Software and Test Examples for Decision Support Systems, A. Lewandowski and A. Wierzbicki eds., pp. 142-176.

Dobrowolski G., K. Hajduk, A. Korytowski, T. Ryś (1988). POSTAN3 – Extended Postoptimal Analysis Package for MINOS. Theory, Software and Test Examples for Decision Support System. A. P. Wierzbicki and A. Lewandowski eds., IIASA Working Paper WP–88–071.

Saunders M. A. (1977). Minos System Manual. System Optimization Laboratory, Department of Operation Research, Stanford University, Stanford USA.

Murtagh B. M., M. A. Saunders (1977). Minos. A Large–Scale Nonlinear Programming System (For Problems With Linear Constraints). User's Guide. System Optimization Laboratory, Department of Operation Research, Stanford University, Stanford USA.

A Data ranging file format

The data required to perform the postoptimal analysis are to be prepared in an MPS-like format and placed in the file specified by the *Parameters/Files* option *data ranging file*. The data sets for different directional ranging procedures may be given in any order. The beginning of the data set for each procedure is identified by the line NAME and its end by the line ENDATA. The line 'SET' may occur immediately after the line NAME in each data set; this line defines the default values of all the variables which are not explicitly defined. Every data set is identified by the name given in the line NAME.

The records in the *data ranging file* should have the following (basic) form, which is analogous to MPS format:

Columns: 1–4,   5–12,   15–22,   25–36,   40–47,   50–61

Fields: f1,   f2,   f3,   f4,   f5,   f6

Below we give a detailed description of the data set for each directional and matrix ranging procedures.

Directional Bound Ranging

	f1	f2	f3	f4	f5	f6
1.	NAME		DBOU			
2.	'SET'		Comments	Value		
3.		LOWER	Row/col. name	Value	Row/col. name	Value
4.		UPPER	Row/col. name	Value	Row/col. name	Value
5.	ENDATA					

Remarks:

- If field f2 in a given record is empty, this means that it is the same as in the last record. Field f2 must not be empty in the first data record.

- The records with identifiers UPPER and LOWER may appear in an arbitrary order.

- LOWER is used for increments of the lower bounds and UPPER for increments of the upper bounds.

Directional Cost Ranging

	f1	f2	f3	f4	f5	f6
1.	NAME		DCOS			
2.	'SET'		Comments	Value		
3.			Col. name	Value	Col. name	Value
4.	ENDATA					

22

Directional RHS Ranging

	f1	f2	f3	f4	f5	f6
1.	NAME		DRHS			
2.	'SET'		Comments	Value		
3.			Row name	Value	Row name	Value
4.	ENDATA					

Matrix Element Ranging

	f1	f2	f3	f4	f5	f6
1.	NAME		MELE			
2.		Col. name	Row name		Row name	
3.	ENDATA					

Remarks:

- Field f2 of record 2 may be empty. This means that it is the same as in the previous record. It must not be empty in the first data record.

- The maximum number of elements which can be specified is the integer part of $N/2$ where N is the number of all variables, $N = columns + rows + 2$.

Matrix Row Ranging

	f1	f2	f3	f4	f5	f6
1.	NAME		MROW			
2.	'SET'		Comments	Value		
3.		Row name	Col. name	Value	Col. name	Value
4.	ENDATA					

23

Matrix Column Ranging

	f1	f2	f3	f4	f5	f6
1.	NAME		MCOL			
2.	'SET'	Comments		Value		
3.		Col. name	Row name	Value	Row name	Value
4.	ENDATA					

Remarks:

- In both data sets, field f2 of record 3 may be empty; this means that it is the same as in the previous record. Field f2 must not be empty in the first data record.

- Only one row and/or column may be specified.

General Rules

The following general rules apply to all data sets:

- One of the pairs of fields (f3,f4) and (f5,f6) may be empty.

- If 'SET' appears, it must follow immediately after NAME. If 'SET' does not occur, the default for all variables whose values are not specified is zero. This has the same effect as

```
                        'SET'        0.
```

- Comments may be entered in arbitrary positions in the data set. They are identified by an asterisk * in the first column.

- The values should be written as real numbers in a format accepted by FORTRAN

- *data ranging file* is read once to find the necessary data set (one cycle is performed). This file is not rewound if it is correctly constructed and the data sets occur in the following order: DCOS, DBOU, DRHS, MELE, MROW, MCOL.

A.1 An example of the data ranging file

```
name          dcos
'set'                         1.
*
*    Empty set of data for directional cost ranging.
*
endata
name          dbou
'set'                         1.
    lower     x4              0.
    lower     r1              0.
    lower     r2              0.
              r3              0.
    lower     r4              0.
```

```
    lower    de              0.
    upper    x4              0.    r1              0.
             r2              0.
             de              0.    r4              0.
endata
name     drhs
'set'                    1.
         r4              0.
endata
name     mele
    x3   r1                   r2
         r3                   r4
         de
endata
name     mrow
'set'                    1.
    r1   x3              0.
endata
name     mcol
'set'                    1.
    x3   r1              0.
endata
```

	Field	Pos.	Format
1	Indicator	1-8	right aligned integer number or string of characters
2	Description	9-48	string of characters
3	Dimension	49-56	string of characters
4	Scale factor	57-63	real number
5	Format	64-71	string of characters
6	Omit indicator	72	character from among: space,1,2
7	Row code	73-80	string of characters

Table 1: Dictionary File Record Format

B Dictionary file format

Both, printout/display description data and complementary data are to be placed in *dictionary file* arranged as specified below. The file is a sequential set of 80 character records.

A *report file* or display is created sequentially - one record of *dictionary file* causes in principle only one consecutive line to be printed/display.

A record of *dictionary file* contains 7 following fields:

Field 1

Indicator field content has three main functions:

a) when it is a MPS code (a string of characters, excluding case **c**) or a number greater than 0 it indicates the solution data that is to be printed out (displayed). A positive number is interpreted as a consecutive number of row and column of LP problem, that is established by the package in order to appear in MPS file and to be printed in NUMBER column of the standard printout of the solution. Note that changes in MPS file such as adding, removing or exchanging any rows or columns cause changes of the above numeration.

b) when it is a negative number or equals to 0 it controls the printout/display by calling one of implemented module functions

c) when it is * (asterisk) followed by at least three spaces it is omitted during reading (comment).

Field 2

The content of description field destined for the printout/display being:

- in case **a** - a description of data item,

- in case **b** - a heading.

Field 3

Dimension field content is a text describing dimension of output data. If the data it to to be non-dimensional the field should contain left aligned "n.d." string.

Field	Default	Description
3	n.d.	output value is non-dimensional
4	1.	scale of output value remains unchanged
5	(F16.0)	adequately rounded integer parts of number will be put out
6	2	value equal to 0 after rounding will be neglected
7		row indicated is presently an objective function

Table 2: Default values for record fields in the dictionary file.

Field 4

The field contains a factor by which data are to be multiplied before being located in the printout/display.

Field 5

The field contains a format description for REAL numbers according to FORTRAN standard. The format refers to one value and a field 16 character long. Output data will be rounded according to the given format.

Field 6

If the field contains "1" a row will be printed out (displayed) even if the rounded data value is equal to 0. If it contains "2" such a row is omitted in the printout/display.

Field 7

The field contains MPS code of the row being a reference for certain module functions what will be discussed further on.

Module actions in case of omitting any of the fields (the field is filled by space characters) are as follows:

Field 1.

Action as for an indicator equal to 0.

Field 2.

In appropriate place 40 space characters will be printed (displayed).

Field 3-6.

Default contents of field fixed earlier will be accepted and an appropriate action will be performed. The default contents are fixed at the beginning of the run; they may be also changed by filling appropriate fields in *dictionary file* records with an indicator less or equal to 0.
Default contents of fields are presented in the table 2.

27

Therefore when a printout/display row is being completed (case **a**) and any of the fields is omitted, the contents fixed by the last control record will be accepted (case **b**) or if this field was omitted in all control records recognized so far - the fixed primary value will be accepted.

The module ends its work after having met the end of file marker of *dictionary file*.

B.1 Implemented Report functions

In its current version the module performs the following functions denoted below by their coding indicator (field 1, case b):

0 printout of field 2 from a new page preceded and followed by 3 blank lines

-1 like in 0 function but without shift to new line

-11 printout of primary solution with single shift

-12 printout of primary solution with percentage, that is, the ratio of variable and its upper constraint, with a single shift

-13 like in "11" function with MPS codes printout for the variables if they exist

-14 like in "12" function with MPS codes printout for the variables if they exist

-15 printout of primary solution for structural variables with single shift and with the following information:

  − product of variable and coefficient of the objective function with scale and rounding according to contents of fields 4 and 5.

  − quotient of coefficient by a variable in the objective function and coefficient in the row indicated by field 7.

On the screen above *header lines* are highlighted. If for arithmetical reasons it is impossible to obtain any of printout values (division by 0) or searched coefficient appears to be equal to 0 - the module changes the printout format (omits appropriate fields) automatically.

Other functions apart from the described above may be implemented only after making necessary changes and extensions.

B.2 Example of a dictionary file

```
      -1SAMPLE RESULTS
      -1Values of Rows
ob        Numerator Value (ob)               UnitOfob
de        Denominator Value (de)             UnitOfde
r1        First Row (r1)                     UnitOfr1
r2        Second Row (r2)                    n.d.
r3        Third Row (r3)                     UnitOfr3
r4        Fourth Row (r4)                    UnitOfr4
      -1Values of Columns                                    (f16.2)
x3        Third Column (x3)                  UnitOfx3
x2        Second Column (x2)                 UnitOfx2
x1        First Column (x1)                  UnitOfx1
```

28

B.3 Example of a report

```
**************************************************************************
 problem    test        objective   ob    (max)   value    2.9009999970E+0
**************************************************************************

          SAMPLE RESULTS

          Values of Rows

Numerator Value (ob)                            29. UnitOfob
Denominator Value (de)                          10. UnitOfde
Second Row (r2)                                 -7.
Third Row (r3)                                   4. UnitOfr3

          Values of Columns

Second Column (x2)                            -1.40 UnitOfx2
First Column (x1)                              1.41 UnitOfx1

**************************************************************************
```

C   Modification File Format

This description will be regarded further on as development of MINOS documentation. The reader is assumed to be acquainted also with the MPSX standard. All the references to those positions are omitted in the text.

Modification file can be interpreted as a series of "exchange" type alterations concerning the MPS file (modification decks). This file is a character set built of records (lines) containing at most 80 characters. It may be divided into a few sections similar to MPSX standard. The record structure almost always accords with that standard. The extensions will be discussed below.

NAME section

Must occur as the first declaration in *modification file*. The package checks consistency of the given name with the name of the basic problem (MPS file).

ROWS section

The changes of rows types are described here. Values of right hand sides of modified rows remain unchanged.

COLUMNS section

The section introduces the changes of values of matrix elements existing in the basic problem. "Existing" means here not rejected by the package when loading MPS file.

RHS section

The section introduces changes of values of right-hand sides existing in the basic problem. "Existing" means here non-zero i.e. declared explicitly in MPS file.

RANGES section

The section introduces the opposite side constraints for the rows declared in MPS file. The above action is performed for each row of L,G,E type regardless if the constraint was introduced or not in the basic problem.

BOUNDS section

The section introduces the constraints for the structural variables of the model.

OBJECTIVE section

This is an extension to MPSX standard. It consists of one record having the following structure :

- positions 1–9 – "OBJECTIVE",

- positions 11–18 – the name of row in the basic problem that is to become the row of an objective function,

- positions 21–32 – "1" in the case of minimizing, "-1" in the case of maximizing.

30

The row declared in this section automatically becomes of the N type (its right-hand-side is deleted) in the case it was declared a non-zero one before.

DENOMINAT section

This is an extension to MPSX standard. It consists of one record of the following structure:

- positions 1–9 – "DENOMINAT",

- positions 11–18 – name of the row in the basic problem that is to become the denominator of the objective function.

The row declared in this section becomes automatically of N type and its right-hand-side is deleted if it was declared as a non-zero one before. This declaration of the section causes an automatic switch of the package into the mode of solving the linear-fractional programming problems. The numerator of the objective function is the row defined either by *Parameters* option or as a result of using OBJECTIVE section.

NODENOMIN section

This is an extension of MPSX standard. It consists of one record of the following structure:

- positions 1–9 – "NODENOMIN"

Declaration of this section causes switching the package into the mode of solving linear programming problem with an objective function declared before as a numerator. The row of denominator remains of the N type.

ENDATA section

The appearance of this section finishes the modification of the problem. The *simplex* algorithm is re-started automatically. By placing this section just after the NAME section that opens the sequence, the computation of the non-modified basic problem (MPS file) is performed.

C.1 General remarks on construction of modification file

1. A series of problems being solved is constructed following the rule "modification to modification".

2. A modification for a consecutive problem in the series is declared using any sequence of sections excluding NAME and ENDATA ended by ENDATA section.

3. The modifications are recorded successively in the package arrays according to their order of appearance in *modification file*. Their number is practically unlimited for one problem.

4. For RHS, RANGES and BOUNDS sections the package checks the names appearing in them accepting only those which accord with declarations in *Parameters/Lp parameters* main menu option.

31

5. The package memorizes the type of the last section opened in the series (from among ROWS, COLUMNS, RHS, RANGES, BOUNDS). Therefore if the series of problems consists in repeated modification of the same coefficient it is not necessary to repeat the respective section marker. If the first modification of the series is not preceded by any of the markers mentioned above it is assumed that the RHS section has been opened.

6. Only 4 first characters of a marker are analyzed by the module.

7. Restrictions to the modifications introduced by COLUMNS and RHS sections are imposed by the structure of MINOS package arrays. If necessary, this inconvenience may be evaded by including in the basic problem (MPS file) coefficients or right-hand-side values e.g. equal to 1, which will be deleted during the first modification and in consecutive problems fixed on desired values,

8. *modification file* may contain several series of modifications for different basic problems starting with the NAME section with an appropriate name.

C.2 Emergency situations in module processing

To the *log file*, under the heading:

MODIFICATION file FOR CYCLE No n

appropriate sections of *modification file* are reprinted by the package as well as information about errors in case they occurred ("n" is the consecutive number of the problem in the series).

Due to the rule of series construction: "modification to modification" most of errors are of fatal type and cause the current as well as following problems will be aborted.
Following errors may be detected:

- in *modification file* no NAME section was found with the name corresponding to the name of the base problem,

- no optimal solution exists for the preceding problem,

- formal incorrectness occurs in *modification file*,

- modifications concerning the admissible area (RHS, RANGES and BOUNDS sections) reduce it to an empty set,

- any of the names concerning the basic problem were not identified.

The module terminates its work correctly after having met the next NAME section or end of file marker just after the ENDATA section on the *modification file*.

C.3 An example of the modification file

```
name          test
columns
*
*   LP-Matrix element modification
*
    x1        ob                .79
*
```

```
endata
*
*   The same once more
*
    x1          ob              1.
    x1          de            5.07
*
endata
```

D Technical requirements

PC–POSTAN can be run on each IBM–XT/AT computer or compatible. The IBM–AT is recommended because of CPU speed.
Required computer configuration:

- 640 kb of RAM memory. PC–POSTAN , itself takes approximately 560 kb of RAM memory for loading therefore use of resident programs is strongly restricted. It should be also mentioned that same resident programs can be in conflict with PC–POSTAN (e.g. *dosedit* program).

- arithmetic co–processor

- hard disk. Usually the program requires three times more of free disk space then the space occupied by data files: *mps file* and *dictionary file*.

- color or monochrome monitor. PC–POSTAN can support colors therefore color monitor is recommended.

- MS–Dos operating system ver. 3.0 or later

- high density floppy drive $5\frac{1}{4}''$ for program installation.

E Program installation

PC–POSTAN is distributed on high density, $5\frac{1}{4}{}''$ floppy disks organized as follows:

- *root* directory containing executable version of the program (file **pcpost.exe**) and **read.me** text file (you are reading it now). Executable version can be easy installed by the COPY command, e.g.:

  <div align="center">

  `copy a:\pcpost.exe c:\bin`

  </div>

- *src* directory containing:

  - file **makefile** which allows to easy compile and install program by the *make* command. Contents of this file should be examined before installation because of proper set of disk drivers and path names.
  - files ***.f, *.cpp, *.h, *.hdr** which are source of the program
  - file **linkfile.msc** containing directives for linking program

- *test* directory containing example of all data files

- *lib* directory containing MINOS LP software library (**lpminos.lib**)

The following software is required for proper program installation:

- UNIX version of *make* program
- UNIX version of *cpp* program
- LAHEY Fortran 77 compiler ver. 2.21 or later
- SPINDRIFT [1] library ver. 1.64
- LPMINOS library of linear MINOS procedures
- Microsoft *link* program ver. 3.65 or later

PC–POSTAN can be installed from sources by the following command:

<div align="center">

make install

</div>

---

[1] see The SINDRIFT Library (1986), A Fortran 77 Subroutine Library for use with the F77L compiler. Reference Manual. Spindrift Laboratories, USA.