

NBSIR 75-666

Post Optimality and Parametric Analysis with the National Bureau of Standards' Linear Programming Subroutine RVSMPX

Tyrone B. Ayers

Applied Mathematics Division

February 1975

Final

Technical Report to
Computer Services Division
Center for Computer Sciences and Technology

207
NBSIR 75-666

**POST OPTIMALITY AND PARAMETRIC
ANALYSIS WITH THE NATIONAL BUREAU OF
STANDARDS' LINEAR PROGRAMMING SUBROUTINE
RVSMPX**

Tyrone B. Ayers

Applied Mathematics Division

February 1975

Final

Technical Report to
Computer Services Division
Center for Computer Sciences and Technology



U.S. DEPARTMENT OF COMMERCE, Rogers C.B. Morton, *Secretary*
James A. Baker, III, *Under Secretary*
Dr. Betsy Ancker-Johnson, *Assistant Secretary for Science and Technology*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Acting Director*

CONTENTS

1. Introduction.....	1
2. Sensitivity Analysis on Cost and RHS Vectors (RANGES).....	2
3. Parametric Programming on Cost Vector (PAROBJ) or on RHS Vector (PARRHS).....	3
4. Deck Set-up and Parameter Settings.....	4
5. Output.....	8
5.1 Output from RANGES.....	8
5.2 Output from PAROBJ and PARRHS.....	8
6. References.....	18
Appendix A: Listing of RANGES.....	19
Appendix B: Listing of PAROBJ.....	25
Appendix C: Listing of PARRHS.....	33
Appendix D: Timing Considerations.....	39

LIST OF FIGURES

Figure 1: PAROBJ Output - Regular Format.....	10
Figure 2: PARRHS Output - Regular Format.....	12
Figure 3: PAROBJ Output - Compressed Format.....	14
Figure 4: PARRHS Output - Compressed Format.....	16

ABSTRACT

This report is a sequel to NBS Report 10695 (February 1972), The National Bureau of Standards' Linear and Quadratic Programming Subroutines, which documented one phase of an effort to provide users of the facility operated by the National Bureau of Standards' Computer Services Division, with reliable clearly-described solution algorithms for selected frequently-arising classes of special mathematical problems. The present report presents subroutines which perform post-optimality analysis and parametric programming studies on linear programming problems solved by the National Bureau of Standards' RVSMPX subroutine. (The present versions of these codes use internal storage only.)

Keywords: Algorithms; post-optimality analysis; parametric programming; linear programming.

ACKNOWLEDGEMENT

The author would like to acknowledge the aid provided by W.G. Hall, R.H.F. Jackson, and P.B. Saunders of the Applied Mathematics Division in the formulation and testing of the subroutines discussed in this report.

1. INTRODUCTION

This report documents another phase of a continuing effort to provide users of the computer facility operated by the National Bureau of Standards' Computer Services Division with reliable, clearly-described solution algorithms for selected, frequently-arising classes of mathematical problems. This phase of the effort has increased NBS's capabilities in the area of linear programming by producing three subroutines: **RANGES**, a subroutine which provides post-optimality ranges for the objective function **coefficients** (costs) and for the constants which constitute the right-hand-sides (RHS) of the linear program's constraints; **PAROBJ**, a subroutine which performs a parametric study of the effects of changes in the costs; and **PARRHS**, a subroutine which performs a **similar parametric study for changes in the RHS**.

The subroutines were written in FORTRAN V, solve the problems "in core," and were designed to be used with RVSM PX, a subroutine for performing the revised simplex method of linear programming. RVSM PX is described in the previous report [2], and familiarity with that document is assumed here.

Sections 2 and 3 of this report discuss, respectively, the methods used in **RANGES**, and those employed for **PAROBJ**, and **PARRHS**. Section 4 presents the deck set-up, and parameter settings required to use the routines, and Section 5 provides sample outputs with explanations. The appendices contain program listings, and also the results of sample problem runs made for purposes of code verification and timing.

2. SENSITIVITY ANALYSIS ON COST AND RHS VECTORS (RANGES)

The input to RANGES, consisting of a linear programming problem together with its solution, is the output from RVSMXP (see [2]). The output from RANGES is a set of upper and lower bounds. An upper-lower bound pair appears for each objective function coefficient and also for each constraint right hand side, and represents the maximum range over which that quantity can vary (with all other problem data fixed) without forcing a change from the optimal basis for the original LP problem.* That is to say, the set of variables in the optimal basis will remain the same so long as no cost or RHS is changed beyond its bounds. It should be noted that in the case of a change in an RHS, although the basis remains the same, the values of the variables in the basis would change. Correspondingly, in the case of a change in a cost, the values of the variables in the basis as well as the membership of the basis will remain the same, while the value of the objective function changes.

Note, that due to round-off error the computed bound values may occasionally vary from the actual values by as much as 3 times the epsilon (EPS) computed in RVSMXP. However, the computed values, when substituted for their corresponding cost or RHS, do indeed yield the same optimal basis.

A number of computer runs were made to check the "correctness" of the code. The times for these runs (which, by the way, employed the very same problems that were used to verify RVSMXP in [2]) are reported in Appendix D. A potentially useful statistic from these runs is that the time consumed by RANGES was on the average roughly 20% of that required by RVSMXP to solve the original problem.

* This technique is described in detail in [1] and [3].

3. PARAMETRIC PROGRAMMING ON COST VECTOR (PAROBJ) OR RHS VECTOR (PARRHS)

Consider the linear programming problem given in matrix notation as

$$\max cx, \text{ subject to } Ax:b \text{ and } x \geq 0,$$

where ":" represents $\leq$, $\geq$, or $=$.^{*} If B is the basis matrix, and c_B and x_B are the corresponding subvectors of c and x respectively, then the solution to this problem is given by $x_B = B^{-1}b$, with optimal objective value of $c_B x_B$.

Parametric programming on the cost vector is concerned with replacing c in the above problem by $c^* = c + \epsilon f$, where f is an arbitrary vector supplied by the user and ϵ is a nonnegative scalar parameter. PAROBJ solves this problem by first solving the original problem using RVSMPX, then computing the largest value of ϵ for which the current optimal basis remains optimal. Call this value $\epsilon_{\max}$. A new basic solution is then calculated which is optimal in some new interval of ϵ values beginning with $\epsilon_{\max}$. The upper bound of that new interval is the new value of $\epsilon_{\max}$.

This procedure continues until either a.) the problem becomes unbounded (linear programming theory assures that it will remain so for all larger values of ϵ), or b.) the last of the critical values of ϵ is reached (the same solution remains optimal for all larger values of ϵ), or c.) a user-specified limit on ϵ is exceeded.

If the solution to the original problem is unbounded, it is still possible that for some positive value of ϵ , the solution to the modified problem "becomes" bounded. If such is the case for the original linear programming problem, then PAROBJ will start the parametric study with the smallest positive value of ϵ for which a bounded, feasible solution is obtained.

As with RANGES, a number of sample problems were run on PAROBJ for the purpose of code verification and timing. The results of these runs are reported in appendix D.

* * *

PARRHS is a subroutine written to perform parametric programming on the RHS vector. This is the situation in which vector b of the original problem stated in the beginning of this section is replaced by $b^* = b + \epsilon r$, where r is an arbitrary vector supplied by the user, and ϵ is as before. All the comments stated above for PAROBJ with regard to boundedness, apply for PARRHS except that the concern here is with feasibility.

^{*}By way of clarification, RVSMPX accepts the A matrix in non-canonical form, so that the constraints need not be all equalities or inequalities.

4. DECK SET-UP AND PARAMETER SETTINGS

The previous sections of this report discussed the subroutines in general terms, stressing the functions they perform. This section will provide the reader with the information needed actually to use the subroutines. The deck set-ups will be discussed first, followed by the form of the call to the subroutines, followed further by an explanation of each of the parameters employed.

RANGES is used in what can be termed a "post-RVSMPX" mode. In the user's main program, after the call to RVSMPX in the form described in [2], the user simply inserts a FORTRAN CALL statement of the following form:

```
CALL RANGES (A,MA,B,MB,MT,NT,L,X,RHSUP,RHSLOW,COEUP,COELOW,NSET),
```

and the range information is printed out.

The observant reader will note that many of the parameters in the above CALL statement appear identical to those in the CALL statement for RVSMPX. In most cases they are exactly the same. There are, however, some variations, which are listed below and should be NOTED by the user.

Both PAROBJ and PARRHS operate in a "super-RVSMPX" mode. The user-supplied main program referred to earlier and discussed in [2] should pass control to one of these routines, which then takes charge of the parametric study and uses RVSMPX as a subroutine. This, of course, means that any parameters required by RVSMPX which would normally be passed directly to it, must now be passed through the parametric programming routines. The FORTRAN CALL statements, as they should appear in the main program, are:

```
CALL PAROBJ (A,MA,B,MB,MT,NT,L,X,TOLP,INV,F,BOUND,NSET),
```

```
CALL PARRHS (A,MA,B,MB,MT,NT,L,X,TOLP,INV,BOUND,R,NSET).
```

Again it should be noted that the similarity between these CALL statements and the one for RVSMPX is not accidental. A full understanding of the parameters, especially as they relate to the various subroutines, is absolutely imperative. Listed below are the parameters that appear in RANGES, PAROBJ, and PARRHS. After each variable name, there appears an (R), an (O), or an (H) if the variable is a parameter of either RANGES, PAROBJ, or PARRHS, respectively. If the variable pertains to all three subroutines without variation, no letters will appear. But if variations are present, an explanation will be given. The m and n used with reference to dimensions have the same meaning as in the RVSMPX documentation: m is the number of constraints, and n is the number of real (original) variables.

* * *

A (R) This is the augmented constraint matrix. It should be dimensioned at least (m+1) by (n+1) and is unchanged by the subroutine.

(O) Because of the nature of PAROBJ, A must be dimensioned at least $(m+2)$ by $(n+1)$ to provide storage for the original coefficients of the objective function. A is unchanged by PAROBJ.

(H) Because of the methods used in storing the original coefficients of the objective function and in dealing with an originally infeasible problem, A must be dimensioned at least $(m+2)$ by $(n+2)$. A is unaltered by PARRHS.

MA This variable equals the value of the first dimension of A. It is unchanged upon exiting from the subroutines.

B (R) This matrix provides storage for the inverse of the basis matrix and should be dimensioned $(m+2)$ by $(m+2)$. B is unchanged by the subroutine.

(O,H) Since the subroutines perform parametric studies which cover several bases, B will contain the inverse of the last optimal basis matrix upon exiting from the subroutines. B should be dimensioned at least $(m+2)$ by $(m+2)$.

MB This is the value of the first dimension of B. It is not altered by the subroutines.

MT (R) This is the number of rows of information in the A matrix, i.e., $(m+1)$. MT is not changed by RANGES.

(O,H) Recall that the A matrix for PAROBJ and PARRHS has $(m+2)$ rows. Hence MT must equal $m+2$. Upon exiting from the subroutine, MT is unaltered.

NT (R,O) This variable equals the number of columns of information in the A matrix, i.e., $(n+1)$. It is unchanged by the subroutines.

(H) Recall that the A matrix for PARRHS has $(n+2)$ columns. Therefore, NT must equal $(n+2)$. Upon exiting from the subroutine, NT is not changed.

L (R) This is a multipurpose vector providing work space needed by RVSMFX. L is not changed in any way by RANGES.

(O,H) In order to maintain a certain consistency in output formulation, the user-supplied values of L only function for the initial reporting of the problem, after which certain elements of L are automatically set by the subroutines. Upon exiting, $L(5)$, $L(8)$, $L(13)$, and $L(14)$ will equal 1. $L(4)$ and $L(12)$ will equal 0. The value of all other $L(I)$, $I \leq 14$, will be the same as they are upon returning from RVSMFX.

X (R) Just as in RVSMFX, X provides storage for the solution of the problem. X is not changed by RANGES.

(O,H) These subroutines perform a parametric study over a series of optimal bases. Therefore, the solution in X upon exiting from the subroutines will not be the original solution, but rather the solution corresponding to the final optimal basis of the parametric study.

TOLP (O,H) This is the variable used by RVSMFX to compute the value of EPS and CEPS. Its value on input should be exactly the same as it is for RVSMFX. TOLP is not changed by the subroutine.

INV (O,H) This is the switch allowing the user to supply RVSM PX with an initial feasible solution. As in RVSM PX, INV may still be used to indicate a user-supplied initial basis. The value of INV is altered in the subroutines.

RHSUP (R) This vector provides storage for the upper bounds on the ranges of the right-hand sides of the given problem being solved. Since each right-hand side will have an upper bound, RHSUP should be dimensioned at least m in the main program, where m is the number of rows in the unaugmented A matrix.

RHSLOW (R) This vector provides storage for the lower bounds on the ranges of the right-hand sides. Like RHSUP, RHSLOW should be dimensioned at least m in the main program.

COEUP (R) This vector provides storage for the upper bounds on the ranges of the coefficients of the objective function of the given problem. COEUP should be dimensioned at least n in the main program, where n is the number of columns in the unaugmented A matrix.

COELOW (R) This vector provides storage for the lower bounds on the ranges of the coefficients. Like COEUP, COELOW should be dimensioned at least n.

R (H) This is the vector supplied by the user to indicate the rates of change of each of the right-hand sides with respect to the scalar ϵ . Therefore, R must be dimensioned at least m. R should be set up so that the ith element of R is the rate of change of the ith right-hand side. For example, if the original right-hand side vector is $b = (1, 2, 3.7, 5)^T$ and if R is $(0, 2, -3, .6)^T$, then the new right hand side vector b^* varies as follows:

$$\begin{aligned} b_1^* &= b_1 + 0 \epsilon = 1 \\ b_2^* &= b_2 + 2 \epsilon = 2 + 2 \epsilon \\ b_3^* &= b_3 - 3 \epsilon = 3.7 - 3 \epsilon \\ b_4^* &= b_4 + 0.6 \epsilon = 5 + 0.6 \epsilon \end{aligned}$$

where $\epsilon \geq 0$. The subroutine does not change R.

F (O) This is the vector supplied by the user to indicate the rates of change of each of the objective function coefficients with respect to the scalar ϵ . Therefore, F must be dimensioned by at least n. F should be set up so that the jth element of F is the rate of change of the jth cost coefficient. For example, if the original cost vector is $c = (2, 3, 1.6, -4)$ and if F is $(2, 0, -.5, 5)$, then the new cost vector c^* varies as follows:

$$c_1^* = c_1 + 2 \epsilon = 2 + 2 \epsilon$$

$$c_2^* = c_2 + 0 \epsilon = 3$$

$$c_3^* = c_3 - 0.5 \epsilon = 1.6 - 0.5 \epsilon$$

$$c_4^* = c_4 + 5 \epsilon = -4 + 5 \epsilon$$

where $\epsilon \geq 0$. F is not altered by the subroutine.

BOUND (O,H) This is the user-specified limit on the non-negative scalar parameter ϵ . The scalar ϵ is incremented up to the value of BOUND. Once the value of ϵ exceeds BOUND, the subroutines will terminate by reporting that the limit has been exceeded and that the current basis remains optimal at that limit. However, the subroutines may terminate before ϵ reaches BOUND if:

- 1) the largest critical value of ϵ is less than BOUND,
- 2) the problem becomes infeasible (PARRHS),
- 3) the problem becomes unbounded (PAROBJ).

The value of BOUND is unchanged by the subroutines.

NSET (R) This parameter is a user-specified switch. If the user sets NSET = 0, ranges are calculated for both right-hand sides and objective function coefficients. If NSET = 1, ranges are determined only for the right-hand sides. If NSET = 2, ranges are obtained only for the objective function coefficients.

(O,H) For these subroutines, this user-specified switch determines the output format. If NSET = 0, a detailed format is used. If NSET = 1, a compressed tabular summary format is used.

5. OUTPUT

This section of the report will discuss the output from the three subroutines, and illustrate them through actual output samples.

5.1 THE OUTPUT FROM RANGES

The printing of the output is controlled by RANGES. Information on the original right-hand side vector is printed first, followed by information on the objective function coefficient vector. The user may obtain information on both the right-hand side and the objective function coefficient vectors or either of them separately, depending on his specified value for the parameter NSET (see section 4). The output is printed in an E14.8 format as follows:

-14845019+02	LE	RHS	1	LE	POS.INFINITY
.54586695+01	LE	RHS	2	LE	POS.INFINITY
.30275573+01	LE	RHS	3	LE	.15395222+02
.15161557+02	LE	RHS	4	LE	POS.INFINITY
NEG.INFINITY	LE	COEFFICIENT	1	LE	-.14055693+00
-.20215248+01	LE	COEFFICIENT	2	LE	-.64413020+00
NEG.INFINITY	LE	COEFFICIENT	3	LE	-.54610899+01
NEG.INFINITY	LE	COEFFICIENT	4	LE	-.40288613+01
-.12985771-01	LE	COEFFICIENT	5	LE	.57053266+00

Line 1, for example, reads $-14.845019 \leq \text{RHS } 1 \leq \text{POSITIVE INFINITY}$. This means that the reported final basis will remain optimal as long as RHS 1 (the first element of the right-hand side vector of the problem being solved) is greater than -14.85019 . Similarly, bounds are reported for each remaining element of the right-hand side vector, i.e., RHS 2, RHS 3, and RHS 4.

Line 5 reads

$\text{NEGATIVE INFINITY} \leq \text{COEFFICIENT } 1 \leq -.14055693$.

This means that coefficient 1, the first element of the cost vector of the problem being solved, has no lower bound and has an upper bound of $-.14055693$. Following line 5 are the bounds on the remaining elements of the cost vector.

5.2 THE OUTPUT FROM PAROBJ AND PARRHS

All output is completely controlled by the subroutines, and the output formats for the two routines are nearly identical. The first item to be printed is a statement that the first critical value of Θ is zero. With this value of Θ , the solution to the parametric programming problem is the same as the solution to the simple linear programming problem, and that

solution is printed next. A formula for the calculation of the objective function value for all values of θ from zero through the next critical value is then printed.

A cycle of output then begins. The "current" critical value of θ is printed, along with the basis change required when passing through that value for θ . The "current" basis is then given, followed by the objective function calculation formula. This output grouping is repeated until one of the termination conditions in section 3 is met. This output format is illustrated in Figures 1 and 2.

A more compressed tabular format is also available. The value of NSET (see Section 4) controls the output format selection. The compressed format is illustrated in Figures 3 and 4.

CRITICAL VALUE (01 OF THETA = .00000000

EPSILON = .364000-04 CAPITAL EPSILON = .364000-03 0 NON-ZERO ENTRIES ARE EFFECTIVELY EQUAL TO ZERO.

END OF PHASE 1. OBJECTIVE FUNCTION = .74505806-08 THERE WERE 2 ITERATIONS.
MINIMUM PIVOT WAS 1.0000 AT ITERATION 1. REAL OBJECTIVE FUNCTION = -1.2676858

END OF PHASE 2. OBJECTIVE FUNCTION = -.66359763 THERE WERE 1 ITERATIONS.
MINIMUM PIVOT WAS 1.0000 AT ITERATION 1.

BASIC VARIABLES

X1 1) = .2211992

S1 1) = 6.538539 S1 2) = 10.48937 S1 3) = 12.37799 S1 5) = .1006814

OBJECTIVE FUNCTION VALUE BETWEEN CRITICAL VALUES 1 01 AND (1) OBTAINED FROM .44239841+00 THETA -.66359762+00

CRITICAL VALUE (11 OF THETA = .62500000+00
AT THAT VALUE, X1 41 ENTERS THE BASIS AND S1 11 LEAVES.

Figure 1: PAROBJ Output
Regular Format

BASIC VARIABLES

X(1)= .5184055 X(4)= 2.080444
S(2)= 12.56981 S(3)= 9.405929 S(5)= 2.478332

OBJECTIVE FUNCTION VALUE = -.18709863+00

OBJECTIVE FUNCTION VALUE BETWEEN CRITICAL VALUES (1) AND (2) OBTAINED FROM .51976991+01* THETA --.36356604+01

CRITICAL VALUE (2) OF THETA = .86764706+00
AT THAT VALUE, X(5) ENTERS THE BASIS AND X(1) LEAVES.

BASIC VARIABLES

X(4)= 5.579681 X(5)= 1.425615
S(2)= 27.47397 S(3)= .3338330 S(5)= 9.736009

OBJECTIVE FUNCTION VALUE = .87410781+00

OBJECTIVE FUNCTION VALUE BETWEEN CRITICAL VALUES (2) AND (3) OBTAINED FROM .14010593+02* THETA --.11282142+02

CRITICAL VALUE (3) OF THETA = .12500000+01
AT THAT VALUE, S(4) ENTERS THE BASIS AND S(3) LEAVES.

BASIC VARIABLES

X(4)= 5.746598 X(5)= 1.592532
S(2)= 28.30855 S(4)= .6676660 S(5)= 10.40367

OBJECTIVE FUNCTION VALUE = .42310992+01

OBJECTIVE FUNCTION VALUE BETWEEN CRITICAL VALUES (3) AND (4) OBTAINED FROM .14678259+02* THETA --.12116724+02

CRITICAL VALUE (4) OF THETA = .16111111+01
PROBLEM GOES UNBOUNDED BEYOND THIS CRITICAL VALUE.

Figure 1: PAROBJ Output
Regular Format (continued)

```

CRITICAL VALUE ( U) OF THETA = .00000000
EPSILON = .364000-04 CAPITAL EPSILON = .364000-03 U NON-ZERO ENTRIES ARE EFFECTIVELY EQUAL TO ZERO.

.....
END OF PHASE 1. OBJECTIVE FUNCTION = .74505806+08 THERE WERE 2 ITERATIONS.
MINIMUM PIVOT WAS 1.0000 AT ITERATION 1. REAL OBJECTIVE FUNCTION = -1.2676858
.....

.....
END OF PHASE 2. OBJECTIVE FUNCTION = -.66359763 THERE WERE 1 ITERATIONS.
MINIMUM PIVOT WAS 1.0000 AT ITERATION 1.
.....

BASIC VARIABLES
X( 1) = .2211992
S( 1) = 6.538539 S( 2) = 10.48937 S( 3) = 12.63779 S( 5) = .1006814

OBJECTIVE FUNCTION VALUE BETWEEN CRITICAL VALUES ( U) AND ( 1) OF THETA OBTAINED FROM -.85714285+00+THETA =.66359763+00
. . . . .

CRITICAL VALUE ( 1) OF THETA = .58730794+01
AT THAT VALUE, X( 5) ENTERS THE BASIS AND S( 5) LEAVES.

BASIC VARIABLES
X( 1) = .2379794 X( 5) = .0000000
S( 1) = 6.521758 S( 2) = 10.43064 S( 3) = 12.68004

```

Figure 2: PARRHS Output
Regular Format

OBJECTIVE FUNCTION VALUE = -.71393830+00
 OBJECTIVE FUNCTION VALUE BETWEEN CRITICAL VALUES (1) AND (2) OF THETA OBTAINED FROM -.22500000+01*THETA -0.58179401+00

.....

CRITICAL VALUE (2) OF THETA = .10106485+01
 AT THAT VALUE, X(4) ENTERS THE BASIS AND X(1) LEAVES.

BASIC VARIABLES

X(4)=	.0000000	X(5)=	-.7139383
S(1)=	11.75731	S(2)=	15.19023
		S(3)=	15.53579

OBJECTIVE FUNCTION VALUE = -.78557532+01
 OBJECTIVE FUNCTION VALUE BETWEEN CRITICAL VALUES (2) AND (3) OF THETA OBTAINED FROM -.25000000+01*THETA -0.32913167+00

.....

THE CRITICAL VALUE OF THETA IS INFINITE.

.....

Figure 2: PARRHS Output
 Regular Format (continued)

THE R VECTOR IS AS FOLLOWS

R(1) = 2.00
R(2) = 1.00
R(3) = 4.00
R(4) = 2.00
R(5) = 2.00

BOUND= 6.00

CRITICAL VALUE (0) OF THETA = .00000000

EPSILON = .364000-04 CAPITAL EPSILON = .364000-03 0 NON-ZERO ENTRIES ARE EFFECTIVELY EQUAL TO ZERO.

*
* END OF PHASE 1. OBJECTIVE FUNCTION = .74505806+08 THERE WERE 2 ITERATIONS.
* MINIMUM PIVOT WAS 1.0000 AT ITERATION 1. REAL OBJECTIVE FUNCTION = -1.2676858
*

*
* END OF PHASE 2. OBJECTIVE FUNCTION = -.66359763 THERE WERE 1 ITERATIONS.
* MINIMUM PIVOT WAS 1.0000 AT ITERATION 1.
*

BASIC VARIABLES

X(1)= .221199
S(1)= 6.538539 S(2)= 10.48936A S(3)= 12.37792 S(5)= .100681

OBJECTIVE FUNCTION VALUOF BETWEEN CRITICAL VALUES (0) AND (1) OF THETA OBTAINED FROM -.85714285+00*THETA -.66359763+00

Figure 3: PAROBJ Output
Compressed Format

```

* * * * *
CRIT VAL  CRIT VAL  BASIS INFORMATION  OBJ FUNC VAL  ORJ FUNC VAL BETWEEN THIS
A.O.                                     LEAVES      AND NEXT CRIT VAL
* * * * *
1      .6250      X( 4)  S( 1)  -.3871      5.1977* THETA  -3.6357
* * * * *
2      .8676      X( 5)  X( 1)  .8741      14.0106* THETA  -11.2821
* * * * *
3      1.2500      S( 4)  S( 3)  6.2311      14.6783* THETA  -12.1167
* * * * *
4      1.6111      NO FURTHER INFORMATION-PROBLEM GOES UNBOUNDED BEYOND THIS CRITICAL VALUE
* * * * *

```

Figure 3: PAROBJ Output
Compressed Format (continued)

THE F VECTOR IS AS FOLLOWS

F(1) = 2.00
F(2) = 1.00
F(3) = 4.00
F(4) = 2.00
F(5) = 2.00
BOUND= 6.00

CRITICAL VALUE (0) OF THETA = .00000000

EPSILON = .364000-04 CAPITAL EPSILON = .364000-03 0 NON-ZERO ENTRIES ARE EFFECTIVELY EQUAL TO ZERO.

*
* END OF PHASE 1. OBJECTIVE FUNCTION = .74505806-08 THERE WERE 2 ITERATIONS.
* MINIMUM PIVOT WAS 1.0000 AT ITERATION 1. REAL OBJECTIVE FUNCTION = -1.2676858
*

*
* END OF PHASE 2. OBJECTIVE FUNCTION = -.66359763 THERE WERE 1 ITERATIONS.
* MINIMUM PIVOT WAS 1.0000 AT ITERATION 1.
*

BASIC VARIABLES

X(1)= .221199

S(1)= 6.534539 S(2)= 10.48936A S(3)= 12.377992 S(5)= .100681

OBJECTIVE FUNCTION VALUE BETWEEN CRITICAL VALUES (0) AND (1) OBTAINED FROM .44239841+00* THETA -.66359762+00

Figure 4: PARRHS Output
Compressed Format

CRIT VAL NO.	CRIT VAL	BASIS INFORMATION ENTERS	LEAVES	OBJ FUNC VAL	OBJ FUNC VAL BETWEEN THIS AND NEXT CRIT VAL
1	.0587	X(5)	S(5)	-.7139	-2.2500* THETA -.5818
2	1.0106	X(4)	X(1)	-2.8558	-2.5000* THETA -.3291

CRITICAL VALUE (3) OF THETA IS INFINITE.

* * * * *

Figure 4: PARRHS Output
Compressed Format (continued)

6. REFERENCES

1. Hadley, G., Linear Programming, Addison-Wesley, Reading, Mass., 1962.
2. Hall, W.G., Jackson, R.H.F., Saunders, P.B., The National Bureau of Standards' Linear and Quadratic Programming Subroutines, NBS Report No. 10695, February 1972.
3. Simonnard, M., Linear Programming, Prentice Hall, Englewood Cliffs, N.J., 1966.

APPENDIX A: Listing of RANGES

WFO9,15 RANGES
FOR 01DA-07/21/73-21:06:00 (,0)

SUBROUTINE RANGES ENTRY POINT 001274

STORAGE USED: CODE(1) 001344; DATA(0) 000234; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

0003 WFOUS
0004 NIOZS
0005 NEXPI5
0006 NERR35

STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000331	IL	0001	000453	106L	0000	000033	11F	0001	000120	12L	0001	000473	120L
0000	000342	13F	0001	000534	130L	0000	000053	14F	0001	000166	144G	0000	000064	15F
0001	000555	150L	0001	000174	151G	0001	000312	16L	0001	000227	167G	0001	000322	17L
0001	000605	170L	0001	000351	18L	0001	000654	180L	0001	000665	190L	0001	000214	2L
0001	000714	200L	0001	000712	201L	0001	000726	220L	0001	000771	230L	0001	000416	234G
0001	000724	237G	0001	000449	250G	0001	001012	250L	0001	000470	261G	0001	000510	267G
0001	001042	270L	0001	001122	280L	0001	001133	290L	0001	001161	300L	0001	001157	301L
0001	000550	305G	0001	001216	310L	0000	000100	311F	0001	000577	315G	0001	001226	320L
0000	000112	321F	0001	001243	322L	0001	001235	330L	0000	000124	331F	0001	000723	352G
0001	000750	360G	0001	001035	375G	0001	001034	405G	0000	000135	500F	0001	000247	7L
0001	000037	8L	0001	000061	9L	0000	000016	1	0000	000145	1NJP5	0000	000011	J
0000	000023	JTEL	0000	000022	J1	0000	000024	J2	0000	000025	J3	0000	000030	J4
0000	000010	K	0000	000020	KL	0000	000014	KU	0000	000006	L1	0000	000005	L10FF
0000	000007	L2	0000	000013	L20FF	0000	000000	M	0000	000003	MP1	0000	000004	MP2
0000	000021	M2	0000	000031	N	0000	000032	N0WN	0000	000012	NPLPG	0000	000002	NPI
0000	000031	NUPP	0000	000017	RATIO	0000	000015	RATMIN	0000	000026	XX	0000	000027	YY

COL	1*	SUBROUTINE RANGES(A,MA,R,MB,MT,MT,L,X,RHSUP,RHSLOW,COEUP,COEL0W,NS	RG1000
00101	2*	1ET)	RG1005
00103	3*	DIMENSION A(MA,1),R(MB,1),L(1),X(1),RHSUP(1),RHSLOW(1)	RG1010
00104	4*	DIMENSION COEUP(1),COEL0W(1)	RG1020
00105	5*	IF(L(3).EQ.0.OR.L(3).EQ.1) GO TO 8	RG1030
00107	6*	WRITE(4,15)	RG1040
00111	7*	GO TO 22	RG1050
00112	8*	R=MNT-1	RG1060
00113	9*	M=MNT-1	RG1070
00114	10*	NPI=M+1	RG1080
00115	11*	MP1=M+1	RG1090
00116	12*	MP2=M+2	RG1100
00117	13*	L10FF=14	RG1110
00120	14*	L1=0	RG1120
00121	15*	L2=0	RG1130
00122	16*	K=NPI	RG1140
00123	17*	9 J=L(L10FF+K)	RG1150
00124	18*	IF(J.EQ.3) GO TO 12	RG1160

```

00126 19* IF(J.EQ.1) LI=LI+1
00130 20* IF(J.EQ.2) L2=L2+1
00132 21* IF(L1*L2.EQ.M) GO TO 12
00134 22* K=K+1
00135 23* GO TO 5
00136 24* 12 NPLPG=M+LI+L2
00137 25* L2OFF=110FF*M+N+L2
00140 26* NPLPG=N+LI+L2
00141 27* IF(NSFT.EQ.2) GO TO 18
00141 28* C .....
00141 29* C ..BEGIN RHS UPPER BOUND COMPUTATION PROCEEDURE..
00141 30* C .....
00143 31* DO 1 J=1,M
00146 32* KU=0
00147 33* RATMIN=1E35
00147 34* C ..CHECK FOR NEGATIVE VALUES IN INVERSE OF BASIS MATRIX..
00150 35* DO 2 I=1,M
00153 36* IF (R(I,J).GE.0.0) GO TO 2
00155 37* KU=1
00155 38* C ..RATIO OF OPTIMAL BASIC SOLUTION TO INVERSE OF BASIS MATRIX..
00156 39* RATIO=-R(I,MP2)/R(I,J)
00156 40* C ..FIND MINIMUM RATIO..
00157 41* IF(RATIO.LT.RATMIN) RATMIN=RATIO
00161 42* 2 CONTINUE
00161 43* C ..COMPUTE UPPER BOUND ON RIGHT HAND SIDE..
00163 44* RHSUP(I)=RATMIN+A(J,MP1)
00163 45* C .....
00163 46* C ..BEGIN RHS LOWER BOUND COMPUTATION PROCEEDURE..
00163 47* C .....
00164 48* 4 KL=0
00165 49* 3 RATMIN=1E35
00165 50* C ..CHECK FOR POSITIVE VALUES IN INVERSE OF BASIS MATRIX..
00166 51* DO 7 I=1,M
00171 52* IF (R(I,J).LE.0.0) GO TO 7
00173 53* KL=1
00173 54* C ..RATIO OF OPTIMAL BASIC SOLUTION TO INVERSE OF BASIS MATRIX..
00173 55* RATIO=-R(I,MP2)/R(I,J)
00174 56* C ..FIND MINIMUM RATIO..
00175 57* IF(RATIO.LT.RATMIN) RATMIN=RATIO
00177 58* 7 CONTINUE
00177 59* C ..COMPUTE LOWER BOUND ON RIGHT HAND SIDE..
00201 60* RHSLOW(J)=A(J,MP1)-RATMIN
00201 61* C .....
00201 62* C ..BEGIN PRINTING OF RHS ROUNDS..
00201 63* C .....
00202 64* 10 IF(KL.EQ.0.AND.KU.GT.0) GO TO 16
00204 65* IF(KL.GT.0.AND.KU.EQ.0) GO TO 17
00206 66* WRITE(4,11)PHSLOW(J),J,RHSUP(J)
00213 67* GO TO 1
00214 68* 16 WRITE(4,13)J,PHSUP(J)
00220 69* GO TO 1
00221 70* 17 WRITE(4,14)PHSLOW(J),J
00225 71* 1 CONTINUE
00227 72* WRITE(4,500)
00231 73* IF(NSFT.EQ.1) GO TO 322
00231 74* C .....
00231 75* C ..BEGIN OBJECTIVE FUNCTION UPPER BOUND COMPUTATION PROCEEDURE..
00231 76* C .....

```

```

RG1170
RG1180
RG1190
RG1200
RG1210
RG1220
RG1230
RG1240
RG1245
RG1250
RG1260
RG1270
RG1280
RG1290
RG1300
RG1310
RG1320
RG1330
RG1340
RG1350
RG1360
RG1370
RG1380
RG1390
RG1400
RG1410
RG1420
RG1430
RG1440
RG1450
RG1460
RG1470
RG1480
RG1490
RG1500
RG1510
RG1520
RG1530
RG1540
RG1550
RG1560
RG1570
RG1580
RG1590
RG1600
RG1610
RG1620
RG1630
RG1640
RG1650
RG1660
RG1670
RG1680
RG1690
RG1695
RG1700
RG1710
RG1720

```

```

00233 77.  IR DO 330 J=1,N
00236 78.  DO 100 I=1,M
00241 79.  100 B(MP2,I)=0.0
00243 80.  M2=0
00243 81.  C .....
00243 82.  C **STOPE APPROPRIATE ROW OF A INVERSE IN ROW M+2 OF B**
00243 83.  C .....
00244 84.  IF(X(J),LT=0.0) GO TO 120
00246 85.  M2=1
00247 86.  DO 105 J=1,M
00252 87.  K=L(L2OFF+J)
00253 88.  JTEL=J
00254 89.  IF(K,FO,J) GO TO 106
00256 90.  105 CONTINUE
00260 91.  106 DO 110 J2=1,M
00263 92.  110 B(MP2,I2)=R(JTEL,J2)
00263 93.  C **COMPUTE INDICATORS AND DENOMINATORS**
00265 94.  120 RATMIN=1F35
00266 95.  DO 190 J2=1,NPLG
00271 96.  IF(X(J2),GE=0.0) GO TO 190
00273 97.  J3=L(L1OFF+J2)
00274 98.  IF(J3,GT=0) GO TO 170
00274 99.  C **REAL VARIABLES**
00276 100.  XX=0.0
00277 101.  IF(J2,FO,J) XX=-1.0
00301 102.  IF(M2,FO,1) GO TO 130
00303 103.  GO TO 150
00304 104.  130 DO 140 I=1,M
00307 105.  140 XY=XX*B(MP2,I)*A(I,J2)
00311 106.  150 IF(XY,GE=0.0) GO TO 190
00313 107.  :YY=ATN(XI,J2)
00314 108.  DO 160 I=1,M
00317 109.  160 YY=YY+(MPI,I)*A(I,J2)
00321 110.  GO TO 180
00322 111.  170 J4=J2-N
00322 112.  C **SLACK AND SURPLUS VARIABLES**
00323 113.  XX=0.0
00324 114.  IF(M2,FO,1) XX=(-1)*((J3+1)*B(MP2,J4)
00326 115.  IF(XY,GE=0.0)) GO TO 190
00330 116.  YY=(-1)*((J3+1)*R(MPI,J4)
00330 117.  C **COMPUTE RATIO OF INDICATORS TO DENOMINATORS**
00331 118.  180 RATIO=YY/XX
00331 119.  C **FIND MINIMUM RATIO**
00332 120.  IF(RATIO,GE,RATMIN) GO TO 190
00334 121.  RATMIN=RATIO
00335 122.  190 CONTINUE
00337 123.  NUPP=C
00340 124.  IF(RATMIN,GT,1E34) NUPP=1
00342 125.  IF(NUPP,FO,1) GO TO 201
00342 126.  C **COMPUTE UPPER ROUND ON COEFFICIENT**
00344 127.  COEUP(I)=RATMIN+ATN(MPI,J)
00345 128.  GO TO 200
00346 129.  201 COEUP(I)=RATMIN
00346 130.  C .....
00346 131.  C **BEGIN OBJECTIVE FUNCTION LOWER BOUND COMPUTATION PROCEDURE
00346 132.  C .....
00347 133.  200 IF(M2,FO=0) GO TO 220
00347 134.  C **CHANGE SIGN OF ROW M+2 OF R**

```

RG1730
RG1740
RG1750
RG1760
RG1770
RG1780
RG1790
RG1800
RG1810
RG1820
RG1830
RG1840
RG1850
RG1860
RG1870
RG1880
RG1890
RG1900
RG1910
RG1920
RG1930
RG1940
RG1950
RG1960
RG1970
RG1980
RG1990
RG2000
RG2010
RG2020
RG2030
RG2040
RG2050
RG2060
RG2070
RG2080
RG2090
RG2100
RG2110
RG2120
RG2130
RG2140
RG2150
RG2160
RG2170
RG2180
RG2190
RG2200
RG2210
RG2220
RG2230
RG2240
RG2250
RG2260
RG2270
RG2280
RG2290
RG2300

```

00351 DO 210 JI=1,M
00354 210 B(MP2,JI)=-P(MP2,JI)
00354 C **COMPUTE INDICATORS AND DENOMINATORS**
00356 220 RATMIN=1E35
00357 DO 290 J2=1,NPLPG
00362 IF(X(J2)*GE.O.O) GO TO 290
00364 IF(J3*.T.O) GO TO 270
00364 C **REAL VARIABLES**
00366 XX=O.O
00367 IF(J2.EQ.O) XX=1.O
00371 IF(M2.EQ.O) GO TO 230
00373 GO TO 250
00374 230 DO 240 I=1,M
00377 240 XY=XX+P(MP2,I)*A(I,J2)
00400 250 IF(XX.EQ.O.O) GO TO 290
00403 YY=A(MP1,J2)
00404 DO 240 I=1,M
00407 260 YY=YY+P(MP1,I)*A(I,J2)
00411 GO TO 290
00412 270 J4=J2-M
00412 C **SLACK AND SURPLUS VARIABLES**
00413 XX=O.O
00414 IF(M2.EQ.O) XX=(-1)*((J3+1)*B(MP2,J4)
00416 JF(XX.EQ.O.O) GO TO 290
00420 YY=(-1)*((J3+1)*B(MP1,J4)
00420 C **COMPUTE RATIO OF INDICATORS TO DENOMINATORS**
00421 280 RATIO=YY/XX
00421 C **FIND MINIMUM RATIO**
00422 IF(RATIO*GE.RATMIN) GO TO 290
00424 RATMIN=RATIO
00425 290 CONTINUE
00427 NDWN=O
00430 JF(RATMIN*GT.E34) NDWN=1
00432 IF(NDWN.EQ.O) GO TO 301
00432 C **COMPUTE LOWER BOUND ON COEFFICIENT**
00434 COELQ(J)=RATMIN+A(MP1,J)
00435 GO TO 300
00436 301 COELQ(J)=RATMIN
00436 C .....
00436 C **BEGIN PRINTING OF OBJECTIVE FUNCTION BOUNDS**
00436 C .....
00437 300 IF(NDWN.EQ.O.AND.NDWN.EQ.O) GO TO 310
00441 IF(NDWN.EQ.O.AND.NDWN.EQ.O) GO TO 320
00443 WRITE(4,311) COELQ(J),J,COFUP(J)
00445 GO TO 300
00445 310 WRITE(4,311) J,COEUP(J)
00445 GO TO 300
00456 320 WRITE(4,321) COELQ(J),J
00457 330 CONTINUE
00457 11 FORMAT(4,A,2X,'LE RHS',13,' LE',E14.8)
00457 13 FORMAT(' NFG.INFINITY LE RHS',13,' LE',E14.8)
00457 14 FORMAT('E14.8,2X,'LE RHS',13,' LE POS.INFINITY')
00457 15 FORMAT(' UN RANGING DONE DUE TO EITHER INFEASIBILITY OR UNBOUNDEDNESS..')
00470 311 FORMAT(' NFG.INFINITY LE COEFFICIENT',13,' LE',E14.8)
00471 321 FORMAT('E14.8,2X,'LE COEFFICIENT',13,' LE POS.INFINITY')
00472 331 FORMAT('E14.8,2X,'LE COEFFICIENT',13,' LE',E14.8)
00473 500 FORMAT('')

```

RG2310

RG2320

RG2330

RG2340

RG2350

RG2360

RG2370

RG2380

RG2390

RG2400

RG2410

RG2420

RG2430

RG2440

RG2450

RG2460

RG2470

RG2480

RG2490

RG2500

RG2510

RG2520

RG2530

RG2540

RG2550

RG2560

RG2570

RG2580

RG2590

RG2600

RG2610

RG2620

RG2630

RG2640

RG2650

RG2660

RG2670

RG2680

RG2690

RG2700

RG2710

RG2720

RG2730

RG2740

RG2750

RG2760

RG2770

RG2780

RG2790

RG2800

RG2810

RG2820

RG2830

RG2840

RG2850

RG2860

RG2870

RG2880

00474
00475

193*
194*

322 RETURN
END

RG2890
RG2900

END OF COMPILATION: NO DIAGNOSTICS.

8 FIN

APPENDIX B: Listing of PAROBJ

@FOR,IS PAROBJ
FOR 010A-09/20/73-11:43:01 (,0)

<SUBROUTINE PAROBJ ENTRY POINT 002743

STORAGE USED: CODE(1) 003156; DATA(0) 000533; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

0003 RVSMXP
0004 NWDUS
0005 NIO2\$
0006 NIO1\$
0007 NEXP1\$
0010 NERR3\$

<TORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0000	000126	10F	0001	000415	100L	0001	001736	1001L	0001	002021	1002L	0001	002073	1003L
0000	000277	1004F	0000	000320	1005F	0000	000337	1006F	0000	000354	1007F	0000	000362	1000F
0000	000366	1010F	0001	002411	1021G	0001	002323	1099L	0001	000157	11L	0001	002335	1100L
0001	002617	1101G	0001	002346	1101L	0000	000371	1102F	0000	000406	1103F	0000	000413	1104F
0000	000420	1105F	0001	002660	1123G	0001	002707	1130G	0001	000532	130L	0001	000130	143G
0001	000131	146G	0001	000203	160G	0001	000217	167G	0000	000127	201F	0001	000335	226G
0001	000670	240L	0001	000411	254G	0001	000753	260L	0001	000470	261G	0001	000476	267G
0001	000764	270L	0001	000525	275G	0001	001016	300L	0001	000551	307G	0001	000602	320G
0001	000623	327G	0001	001134	330L	0001	000826	333G	0001	001202	340L	0001	000657	345G
0001	001221	350L	0001	000733	361G	0000	000051	4F	0001	001250	400L	0000	000143	401F
0000	000157	402F	0001	001011	407G	0001	001054	416G	0001	001105	427G	0001	001126	437G
0000	000062	5F	0001	001340	500L	0001	001333	520G	0001	001361	526G	0001	001457	543G
0001	001463	547G	0001	001541	557G	0001	001573	566G	0000	000063	6F	0001	001422	600L
0001	001703	617G	0001	001477	620L	0000	000064	7F	0001	001545	700L	0001	002114	706G
0001	001614	710L	0001	001755	720L	0001	001762	730L	0001	001772	740L	0001	002004	750L
0000	000204	751F	0001	002130	760L	0001	002277	762G	0001	000351	8L	0001	002304	800L
0001	002425	810L	0000	000101	9F	0001	002624	900L	0000	000220	901F	0000	000230	902F
0000	000241	903F	0000	000254	904F	0000	000266	905F	0000	000270	906F	0001	002632	910L
0001	002640	920L	0001	002646	930L	0001	002664	999L	0000	R	000016	C91T	0000	R
0000	R	000037	ENUM	0000	R	000021	EPS	0000	I	000024	I	0000	I	000030
0000	I	000025	J	0000	I	000033	J1	0000	I	000035	J2	0000	I	000017
0000	I	000020	KT1	0000	I	000032	K1	0000	I	000042	K3	0000	I	000047
0000	I	000050	LET2	0000	I	000006	L1	0000	I	000014	L10FF	0000	I	000015
0000	I	000010	L3	0000	I	000001	M	0000	I	000002	MP1	0000	I	000031
0000	I	000000	N	0000	I	000011	NPL	0000	I	000013	NPLPG	0000	I	000004
0000	I	000041	NTER	0000	R	000026	R	0000	R	000036	RATIO	0000	R	000027
0000	R	000043	SMALL	0000	R	000005	STAR	0000	R	000045	TEST	0000	R	000023
											XX			YY

00101 1* SUBROUTINE PAROBJ(A,MA,R,MB,MT,NT,L,X,TOLP,INV,F,BOUND,NSET) 001000
00103 2* DIMENSION A(MA,1),B(MB,1),L(1),X(1),F(1) 001010
00104 3* N=NT-1 001020
00105 4* M=MT-2 001030
00106 5* MP1=M+1 001040

```

00107 6*
00110 7*
00111 8*
00112 9*
00113 10*
00114 11*
00115 12*
00116 13*
00117 14*
00120 15*
00121 16*
00122 17*
00123 18*
00124 19*
00125 20*
00126 21*
00127 22*
00130 23*
00131 24*
00132 25*
00133 26*
00134 27*
00136 28*
00137 29*
00141 30*
00142 31*
00145 32*
00150 33*
00151 34*
00153 35*
00156 36*
00156 37*
00157 38*
00162 39*
00164 40*
00166 41*
00171 42*
00176 43*
00201 44*
00203 45*
00207 46*
00207 47*
00211 48*
00212 49*
00214 50*
00217 51*
00221 52*
00221 53*
00223 54*
00224 55*
00225 56*
00230 57*
00232 58*
00233 59*
00234 60*
00236 61*
00237 62*
00241 63*

MP2=MP1+1
NP1=NT
MT=MP1
STAR='*'
L1=L(1)
L2=L(2)
L3=L(3)
L(5)=1
L(8)=1
L(12)=1
L(13)=1
L(14)=1
NPL=N+L1
NPLP1=N+L1+1
NPLPG=N+L1+L2
L1OFF=14
L2OFF=L1OFF+M+N+L2
CRIT=0.0
KT=1
KT1=0
EPS=TOLP
IF(EPS.GT.0.0) GO TO 11
XX=1.0E-5
IF(EPS.LT.0.0) XX=-EPS
YY=0.0
DO 1 I=1,M
DO 1 J=1,M
EPS=A(I,J)
IF(EPS.LT.0.0) EPS=-EPS
1 YY=YY+EPS
: EPSE=XX*YY/(M*N)
C **STORE ORIGINAL CJ'S**
11 DO 3 J=1,N
3 A(MP2,J)=A(MP1,J)
WRITE(6,1007)
DO 1008 I=1,N
1008 WRITE(6,1009) I,F(I)
WRITE(6,1010) BOUND
WRITE(6,6)
WRITE(6,4) KT1,CRIT
WRITE(6,5)
C **SOLVE ORIGINAL PROBLEM**
CALL RVSMXP(A,MA,B,MB,MT,NT,L,X,EPS,INV,0)
WRITE(6,6)
IF(L(3).EQ.1) WRITE(6,7)
IF(L(3).EQ.2) GO TO 910
IF(L(3).EQ.4) GO TO 920
C **COMPUTE C + THETA*F FOR OBJECTIVE FUNCTION VALUE**
R=0.0
S=0.0
DO 8 I=1,N
IF (X(I).LE.0.0) GO TO 8
R=R+A(MP2,I)*X(I)
S=S+F(I)*X(I)
8 CONTINUE
ISG=I+1
IF(R.LT.0.0) ISG=I
WRITE(6,9) KT1,KT,S,ISG,R

```

```

OR1050
OR1060
OR1070
OR1071
OR1080
OR1090
OR1100
OR1110
OR1120
OR1130
OR1140
OR1150
OR1160
OR1170
OR1180
OR1190
OR1200
OR1210
OR1211
OR1212
OR1220
OR1230
OR1240
OR1250
OR1260
OR1270
OR1280
OR1290
OR1300
OR1310
OR1311
OR1320
OR1330
OR1340
OR1341
OR1342
OR1343
OR1344
OR1345
OR1350
OR1360
OR1370
OR1380
OR1390
OR1400
OR1410
OR1420
OR1430
OR1440
OR1450
OR1460
OR1470
OR1480
OR1490
OR1500
OR1510
OR1520
OR1530

```

```

00250      64*      WRITE(6,10)
00250      65*      C **COMPUTE FR*B IN ORDER OF BASIC VARIABLES**
00252      66*      WRITE(6,905) (STAR,I=1,40)
00260      67*      100 DO 110 I=1,M
00263      68*      110 R(MP2,I)=0.0
00265      69*      M2=0
00266      70*      DO 130 J=1,M
00271      71*      K1=L(L2OFF+J)
00272      72*      IF(K1.GT.N) GO TO 130
00274      73*      DO 120 J1=1,M
00277      74*      120 R(MP2,J1)=B(MP2,J1)*(F(K1)*R(J,J1))
00301      75*      130 CONTINUE
00301      76*      C *****
00301      77*      C **COMPUTE INDICATORS, DENOMINATORS, AND MINIMUM RATIO**
00301      78*      C ** FOR UNROUNDED ORIGINAL PROBLEM **
00301      79*      C *****
00301      80*      IF(L(3).NE.3) GO TO 300
00305      81*      RATMIN=IE35
00306      82*      DO 270 J=1,NPLPG
00311      83*      IF(X(J).GE.0.0) GO TO 270
00313      84*      J1=L(L1OFF+J)
00314      85*      IF(J1.GT.0) GO TO 240
00314      86*      C **REAL VARIABLES**
00316      87*      YY=A(MP1,J)
00317      88*      DO 200 I=1,M
00322      89*      200 YY=YY+B(MP1,I)*A(I,J)
00324      90*      IF(YY.LE.EPS) GO TO 270
00326      91*      DO 220 I=1,M
00331      92*      XX=0.0
00332      93*      DO 210 J1=1,M
00335      94*      210 XX=XX+R(I,J1)*A(J1,J)
00337      95*      IF(XX.GT.0.0) GO TO 270
00341      96*      220 CONTINUE
00343      97*      XX=-F(J)
00344      98*      DO 230 I=1,M
00347      99*      230 XX=XX+R(MP2,I)*A(I,J)
00351      100*      IF(XX.LE.0.0) GO TO 270
00353      101*      GO TO 260
00354      102*      C **SLACK AND SURPLUS VARIABLES**
00354      103*      240 J2=J-N
00355      104*      YY=(-1)*(J1+1)*B(MP1,J2)
00356      105*      IF(YY.LE.EPS) GO TO 270
00360      106*      DO 250 I=1,M
00363      107*      XX=(-1)*(J1+1)*R(I,J2)
00364      108*      IF(XX.GT.0.0) GO TO 270
00366      109*      250 CONTINUE
00370      110*      XX=(-1)*(J1+1)*R(MP2,J2)
00371      111*      IF(XX.LE.0.0) GO TO 270
00371      112*      C **FIND MINIMUM RATIO**
00373      113*      260 YY=YY/XX
00374      114*      IF(YY.GF.RATMIN) GO TO 270
00376      115*      RATMIN=YY
00377      116*      270 CONTINUE
00401      117*      IF(RATMIN.LE.IE34) GO TO 400
00403      118*      WRITE(6,201)
00405      119*      WRITE(6,905) (STAR,I=1,40)
00415      120*      GO TO 999
00413      121*      C *****
00413      121*
081540
081550
081551
081560
081570
081580
081590
081600
081601
081610
081620
081630
081640
081650
081660
081670
081680
081690
081700
081710
081720
081730
081740
081750
081760
081770
081780
081790
081800
081810
081820
081830
081840
081850
081860
081870
081880
081890
081900
081910
081920
081930
081940
081950
081960
081970
081980
081990
082000
082010
082020
082030
082040
082050
082060
082061
082070
082080

```

```

00413 122* C **COMPUTE INDICATORS, DENOMINATORS, AND MINIMUM RATIO**
00413 123* C ** FOR ROUNDED ORIGINAL PROBLEM **
00413 124* C *****
00415 125* 300 RATMIN=1E35
00420 126* DO 350 J=1,NPLPG
00422 127* IF(X(J).GE.0.0) GO TO 350
00423 128* J1=L(L1OFF+J)
00423 129* IF(J1.GT.0) GO TO 330
00423 130* C **REAL VARIABLES**
00425 131* XX=-F(J)
00426 132* DO 310 I=1,M
00426 133* 310 XX=XX+B(MP2,I)*A(I,J)
00426 134* IF(XX.GE.0.0) GO TO 350
00426 135* YY=A(MP1,J)
00426 136* DO 320 I=1,M
00426 137* 320 YY=YY+R(MP1,I)*A(I,J)
00426 138* GO TO 340
00426 139* C **SLACK AND SURPLUS VARIABLES**
00426 140* 330 J2=J-N
00426 141* XX=(-1)**(J1+1)*B(MP2,J2)
00426 142* IF(XX.GE.0.0) GO TO 350
00426 143* YY=(-1)**(J1+1)*B(MP1,J2)
00426 144* C **FIND MINIMUM RATIO**
00426 145* 340 RATIO=YY/XX
00426 146* IF(RATIO.GE.RATMIN) GO TO 350
00426 147* RATMIN=RATIO
00426 148* ENUM=YY
00426 149* DENOM=XX
00426 150* NTER=J
00426 151* 350 CONTINUE
00426 152* IF(RATMIN.GT.1E34) GO TO 900
00426 153* IF(RATMIN.GT.EPS) GO TO 400
00426 154* M2=1
00426 155* RATMIN=-(1.2*EPS)-ENUM/DENOM
00426 156* C **ACCUMULATE CRITICAL VALUE**
00426 157* 400 CRIT=CRIT+RATMIN
00426 158* IF(L(3).EQ.3) WRITE(6,401) CRIT
00426 159* IF(CRIT.LE.ROUND) GO TO 500
00426 160* WRITE(6,402) ROUND
00426 161* ISG=+1
00426 162* IF(R.LT.0.0) ISG=-1
00426 163* KT1=KT
00426 164* KT=KT+1
00426 165* WRITE(6,9) KT1,KT,S,ISG,R
00426 166* WRITE(6,905) (STAR,I=1,40)
00426 167* GO TO 999
00426 168* C *****
00426 169* C **UPDATE OBJECTIVE FUNCTION WITH NEW THETA AND F VECTOR**
00426 170* C *****
00426 171* 500 DO 510 J=1,N
00426 172* 510 CONTINUE
00426 173* A(MP1,J)=A(MP1,J)+(RATMIN*F(J))
00426 174* IF(L(3).EQ.3.0R.M2.EQ.1) GO TO 800
00426 175* C *****
00426 176* C **GET APPROPRIATE Y(J) AND STORE IN COLUMN MP1 OF R**
00426 177* C *****
00426 178* K3=L(L1OFF+NTER)
00426 179* IF(K3.EQ.0) GO TO 600
00436 00413 122*
00436 00413 123*
00436 00413 124*
00436 00415 125*
00436 00420 126*
00436 00422 127*
00436 00423 128*
00436 00423 129*
00436 00423 130*
00436 00425 131*
00436 00426 132*
00436 00426 133*
00436 00426 134*
00436 00426 135*
00436 00426 136*
00436 00426 137*
00436 00426 138*
00436 00426 139*
00436 00426 140*
00436 00426 141*
00436 00426 142*
00436 00426 143*
00436 00426 144*
00436 00426 145*
00436 00426 146*
00436 00426 147*
00436 00426 148*
00436 00426 149*
00436 00426 150*
00436 00426 151*
00436 00426 152*
00436 00426 153*
00436 00426 154*
00436 00426 155*
00436 00426 156*
00436 00426 157*
00436 00426 158*
00436 00426 159*
00436 00426 160*
00436 00426 161*
00436 00426 162*
00436 00426 163*
00436 00426 164*
00436 00426 165*
00436 00426 166*
00436 00426 167*
00436 00426 168*
00436 00426 169*
00436 00426 170*
00436 00426 171*
00436 00426 172*
00436 00426 173*
00436 00426 174*
00436 00426 175*
00436 00426 176*
00436 00426 177*
00436 00426 178*
00436 00426 179*

```

```

00540 180*
00542 181*
00545 182*
00546 183*
00551 184*
00554 185*
00555 186*
00556 187*
00561 188*
00561 189*
00561 190*
00561 191*
00563 192*
00564 193*
00565 194*
00570 195*
00572 196*
00573 197*
00574 198*
00576 199*
00577 200*
00600 201*
00602 202*
00603 203*
00604 204*
00607 205*
00614 206*
00623 207*
00625 208*
00627 209*
00633 210*
00633 211*
00635 212*
00636 213*
00637 214*
00641 215*
00642 216*
00643 217*
00644 218*
00645 219*
00647 220*
00650 221*
00651 222*
00652 223*
00653 224*
00655 225*
00657 226*
00665 227*
00666 228*
00667 229*
00670 230*
00671 231*
00672 232*
00673 233*
00674 234*
00676 235*
00700 236*
00703 237*

IF(K3.EQ.1.OR.K3.EQ.2) GO TO 620
600 DO 610 I=1,M
    R(I,MP1)=0.0
    DO 610 J=1,M
610 R(I,MP1)=B(I,MP1)+B(I,J1)*A(J1,NTER)
    GO TO 700
620 J2=NTER-N
    DO 630 I=1,M
630 R(I,MP1)=(-1)**(K3+1)*B(I,J2)
C *****
C **FIND THE LEAVING AND ENTERING VARIABLES**
C *****
700 SMALL=1E35
    J5=0
    DO 710 J=1,M
    IF(R(J,MP1).LF.0.0) GO TO 710
    J5=1
    TEST=B(J,MP2)/B(J,MP1)
    IF(1TEST.GT.SMALL) GO TO 710
    SMALL=TEST
    LEAVE=L(L2OFF+J)
710 CONTINUE
    KT1=KT
    KT=KT+1
    IF(NSET.EQ.1.AND.J5.EQ.0) WRITE(6,5)
    IF(NSET.EQ.1.AND.J5.EQ.0) WRITE(6,1105)KT1,CRIT
    IF(NSET.EQ.1.AND.J5.EQ.0) WRITE(6,905) (STAR,I=1,40)
    IF(NSET.EQ.1.AND.J5.EQ.0) GO TO 999
    IF(NSET.EQ.1) GO TO 1001
    WRITE(6,4) KT1,CRIT
    IF(J5.EQ.0) GO TO 930
C **SET X VECTOR FOR RVSM PX WITH ADVANCED START**
1001 X(LEAVE)=-1.0
    X(NTER)=1.0
    IF(NTER.GT.N) GO TO 720
    LET1=X(
    GO TO 730
720 LET1=X(
    NTER=NTER-N
730 IF(LEAVE.GT.N) GO TO 740
    LET2=X(
    GO TO 750
740 LET2=X(
    LEAVE=LEAVE-N
    IF(NSET.EQ.1) L(13)=0
750 IF(NSET.EQ.1) GO TO 1002
    WRITE(6,751) LET1,NTER,LET2,LEAVE
1002 L(1)=L1
    L(2)=L2
    L(3)=0
    L(4)=0
    L(12)=0
    INV=1
    CALL RVSM PX(A,MA,B,MB,MT,NT,L,X,EPS,INV,0)
    WRITE(6,6)
    IF(NSET.EQ.1) GO TO 1003
    WRITE(6,906) X(M+N+L2+1)
1003 R=0.0
OR2640
OR2650
OR2660
OR2670
OR2680
OR2690
OR2700
OR2710
OR2720
OR2730
OR2740
OR2750
OR2760
OR2770
OR2780
OR2790
OR2800
OR2810
OR2820
OR2830
OR2840
OR2850
OR2860
OR2861
OR2862
OR2863
OR2864
OR2865
OR2866
OR2867
OR2870
OR2880
OR2890
OR2900
OR2910
OR2920
OR2930
OR2940
OR2950
OR2960
OR2970
OR2980
OR2990
OR3000
OR3002
OR3004
OR3010
OR3020
OR3030
OR3040
OR3050
OR3051
OR3060
OR3070
OR3080
OR3082
OR3084
OR3090

```

```

00704 238*
00705 239*
00710 240*
00712 241*
00713 242*
00714 243*
00716 244*
00717 245*
00721 246*
00724 247*
00727 248*
00732 249*
00732 250*
00747 251*
00751 252*
00760 253*
00766 254*
00767 255*
00770 256*
00771 257*
00773 258*
00777 259*
01000 260*
01001 261*
01003 262*
01004 263*
01005 264*
01007 265*
01010 266*
01011 267*
01012 268*
01013 269*
01014 270*
01016 271*
01017 272*
01020 273*
01023 274*
01025 275*
01026 276*
01027 277*
01031 278*
01032 279*
01034 280*
01037 281*
01042 282*
01045 283*
01056 284*
01061 285*
01064 286*
01066 287*
01075 288*
01077 289*
01105 290*
01106 291*
01110 292*
01111 293*
01113 294*
01114 295*

S=0.0
DO 760 I=1,N
  IF(X(I).LE.0.0) GO TO 760
  R=R+A(MP2,I)*X(I)
  S=S+F(I)*X(I)
760 CONTINUE
  ISG=I+1
  IF(R.LT.0.0) ISG=1
  IF(NSET.EQ.1.AND.KT1.EQ.1) WRITE(6,1004)
  IF(NSET.EQ.1.AND.KT1.EQ.1) WRITE(6,1005)
  IF(NSET.EQ.1.AND.KT1.EQ.1) WRITE(6,5)
  IF(NSET.EQ.1) WRITE(6,1102) KT1,CRIT,X(M+N+L2+1),S,ISG,R
  11) S,ISG,R
  IF(NSET.EQ.1) GO TO 100
  WRITE(6,9) KT1,KT,S,ISG,R
  WRITE(6,905) (STAR,I=1,40)
  GO TO 100
800 KT1=KT
  KT=KT+1
  IF(NSET.EQ.1) GO TO 1099
  WRITE(6,4) KT1,CRIT
1099 L(1)=L1
  L(2)=L2
  IF(NSET.EQ.1) GO TO 1100
  L(3)=L3
  INV=1
1100 IF(NSET.NE.1) GO TO 1101
  L(3)=0
  L(4)=0
  L(12)=0
  L(13)=0
1101 CALL RVSMXP(A,MA,B,MB,MT,NT,L,X,EPS,INV,0)
  WRITE(6,5)
  R=0.0
  S=0.0
DO 810 I=1,N
  IF(X(I).LE.0.0) GO TO 810
  R=R+A(MP2,I)*X(I)
  S=S+F(I)*X(I)
810 CONTINUE
  ISG=I+1
  IF(R.LT.0.0) ISG=1
  IF(NSET.EQ.1.AND.KT1.EQ.1) WRITE(6,1004)
  IF(NSET.EQ.1.AND.KT1.EQ.1) WRITE(6,1005)
  IF(NSET.EQ.1.AND.KT1.EQ.1) WRITE(6,5)
  IF(NSET.EQ.1) WRITE(6,1102) KT1,CRIT,X(M+N+L2+1),S,ISG,R
  IF(NSET.EQ.1) WRITE(6,1103)
  IF(NSET.EQ.1) WRITE(6,1104)
  IF(NSET.EQ.1) GO TO 100
  WRITE(6,9) KT1,KT,S,ISG,R
  WRITE(6,10)
  WRITE(6,905) (STAR,I=1,40)
  GO TO 100
900 WRITE(6,901)
  GO TO 999
910 WRITE(6,902)
  GO TO 999
920 WRITE(6,903)

```

```

083100
083110
083120
083130
083140
083150
083160
083170
083172
083174
083176
083177
083178
083179
083180
083190
083200
083210
083211
083212
083213
083220
083230
083231
083240
083242
083244
083246
083248
083250
083252
083260
083270
083280
083290
083300
083310
083320
083330
083340
083350
083360
083361
083362
083363
083364
083365
083366
083367
083370
083380
083381
083390
083400
083410
083420
083430
083440

```

```

01116 296* GO TO 999
01117 297* WRITE(6,904)
01121 298* WRITE(6,905) (STAR,I=1,40)
01127 299* DO 1000 J=1,N
01142 300* A(MP1,J)=A(MP2,J)
01133 301* 1000 CONTINUE
01145 302* 4 FORMAT(1H0,'CRITICAL VALUE (' ,I3,' ) OF THETA = ' ,E14.8)
01146 303* 5 FORMAT(//)
01137 304* 6 FORMAT(//)
01140 305* 7 FORMAT(' NUMERICAL DIFFICULTIES IN RVSM PX. QUESTION RESULTS OF PA
01140 306* IRAMETRIC STUDY')
01141 307* 9 FORMAT(' OBJECTIVE FUNCTION VALUE BETWEEN CRITICAL VALUES (' ,I3,' )
01141 308* 1 AND (' ,I3,' ) OBTAINED FROM ' ,E15.8,' * THETA 'A1,E15.8)
01142 309* 10 FORMAT(//)
01143 310* 201 FORMAT(' PROBLEM REMAINS UNBOUNDED THROUGHOUT RANGE OF CRITICAL VA
01143 311* LUES')
01143 312* 401 FORMAT(' WHEN THETA = ' ,F15.10,' YOUR PROBLEM BECOMES BOUNDED AND PR
01144 313* OCEEDS')
01145 314* 402 FORMAT(' CRITICAL VALUE OF THETA BYPASSES YOUR LIMITATION (BOUND) ,
01145 315* 1 THEREFORE THIS BASIS REMAINS OPTIMAL AT YOUR LIMIT OF ' ,F10.4)
01146 316* 751 FORMAT(' AT THAT VALUE ' ,A3,I4,' ) ENTERS THE BASIS AND ' ,A3,I4,' ) LEA
01146 317* VES')
01147 318* 901 FORMAT(' THE CRITICAL VALUE OF THETA IS INFINITE.')
01150 319* 902 FORMAT(' PROBLEM IS INFEASIBLE. NO STUDY PERFORMED')
01151 320* 903 FORMAT(' SYSTEM ERROR ENCOUNTERED BY RVSM PX. NO STUDY PERFORMED')
01152 321* 904 FORMAT(' PROBLEM GOES UNBOUNDED BEYOND THIS CRITICAL VALUE.')
01153 322* 905 FORMAT(/40A3)
01154 323* 906 FORMAT(' OBJECTIVE FUNCTION VALUE = ' ,E14.8)
01155 324* 1004 FORMAT(' CRIT VAL CRIT VAL BASIS INFORMATION ORJ FUNC
01156 325* IVAL OBJ FUNC VAL BETWEEN THIS')
01156 326* 1005 FORMAT(' NO. ENTERS LEAVES ' ,I5X,' )
01157 327* 1 AND NEXT CRIT VAL')
01160 328* 1006 FORMAT(16,F15.4,5X,A3,I4,' ) ,3X,A3,I4,' ) ,F14.4,6X,F10.4,' * THETA
01160 329* ' ,A1,F10.4)
01160 330* 1007 FORMAT(' THE F VECTOR IS AS FOLLOWS')
01161 331* 1009 FORMAT(' F(' ,I4,' ) = ' ,F6.2)
01162 332* 1010 FORMAT(' BOUND= ' ,F6.2)
01163 333* 1102 FORMAT(16,F15.4,5X,' NO REPORT - THETA ' ,F14.4,6X,F10.4,' * THETA ' ,
01163 334* 1A1,F10.4)
01164 335* 1103 FORMAT(26X,' LESS THAN EPS OR ')
01164 336* 1104 FORMAT(26X,' ORIG PROB UNBOUNDED')
01165 337* 1105 FORMAT(16,F15.4,5X,' NO FURTHER INFORMATION-PROBLEM GOES UNBOUNDED
01166 338* 1 BEYOND THIS CRITICAL VALUE')
01167 339* MT=MT+1
01170 340* RETURN
01171 341* END

```

END OF COMPILATION: NO DIAGNOSTICS.

APPENDIX C: Listing of PARRHS

FOR,IS PARPHS

FOR C10A-C9/21/73-19:31:09 (,0)

SUBROUTINE PARPHS ENTRY POINT 002027

STORAGE USED: CODE(11 002233); DATA(0) 000447; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

0003 RVSNPX
0004 WNDUR
0005 NI02\$
0006 NI01\$
0007 WFXPI\$
0010 NFRR3\$

STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

BLOCK	TYPE	RELATIVE LOCATION	NAME
0000	000060	10F	
0001	0010A3	1006L	
0001	000141	133G	
0000	000103	16F	
0001	000626	200L	
0000	000164	231F	
0001	001130	300L	
0001	000615	320G	
0001	001057	414G	
0001	001273	501G	
0001	001370	550L	
0001	001745	657G	
0001	001512	910L	
0000	000267	951F	
0000	000021	I	
0000	000044	J2	
0000	000037	LEAVE	
0000	000011	L2	
0000	000002	HP2	
0000	000014	NPLP1	
0000	R 000040	RATMAX	
0000	000060	10F	
0001	001206	1007L	
0000	000121	14F	
0000	000121	17F	
0001	000323	206G	
0001	000441	254G	
0000	000212	301F	
0001	000664	330G	
0001	001124	423G	
0001	001314	511G	
0001	001403	560L	
0001	001772	664G	
0001	001522	920L	
0001	001751	990L	
0000	000403	INJP\$	
0000	000027	K	
0000	000032	LET1	
0000	000020	L2OFF	
0000	000001	N	
0000	000004	NP1	
0000	R 000025	RATMIN	
0000	000031	15G	
0000	000035	KLEAVE	
0000	000034	LET2	
0000	000012	L3	
0000	000016	NMG1	
0000	000005	NP2	
0000	R 000006	STAR	
0000	000304	1003F	
0000	001556	1008L	
0001	000155	140G	
0000	000132	18F	
0001	000345	217G	
0001	000524	266G	
0001	001676	302L	
0001	000670	334G	
0001	001166	453G	
0001	001427	540G	
0001	000236	561F	
0001	000446	8L	
0001	001527	930L	
0000	R 000026	CBY	
0000	000031	15G	
0000	000023	KT	
0000	000010	L1	
0000	000000	M	
0000	000013	NPL	
0000	000033	NTER	
0000	R 000042	XX	
0000	000310	1004F	
0000	000330	1009F	
0001	001716	15L	
0001	000144	19F	
0001	000354	224G	
0000	000547	277G	
0001	000557	304G	
0001	000707	345G	
0000	001243	470G	
0001	001322	540L	
0001	001434	570L	
0001	000454	9L	
0000	000253	931F	
0000	R 000030	COMPUT	
0000	000022	J	
0000	000024	KT1	
0000	000017	L1OFF	
0000	000003	MPI	
0000	000015	NPLPG	
0000	R 000036	RATIO	
0000	R 000043	YY	
0000	000313	1005F	
0000	000351	1010F	
0001	000174	151G	
0001	000622	199L	
0001	000726	230L	
0000	000045	3F	
0001	001173	310L	
0000	000056	4F	
0000	000057	5F	
0001	001454	546G	
0001	001711	642G	
0001	001505	900L	
0000	000251	950F	
0000	R 000007	CRIT	
0000	000041	J1	
0000	000024	KT1	
0000	000017	L1OFF	
0000	000003	MPI	
0000	000015	NPLPG	
0000	R 000036	RATIO	
0000	R 000043	YY	

CO101	1*	SURROUTINE PARPHS(A,MA,R,MB,MT,NT,L,X,TOLP,INV,BOUND,R,NSET)	RH1000
CO103	2* <td>01MENSTON A(MA,I),R(MB,I),L(II),X(II),R(II)</td> <td>RH1010</td>	01MENSTON A(MA,I),R(MB,I),L(II),X(II),R(II)	RH1010
CO104	3* <td>M=MT-2</td> <td>RH1020</td>	M=MT-2	RH1020
CO105	4* <td>N=NT-2</td> <td>RH1030</td>	N=NT-2	RH1030
CO106	5* <td>MP2=M+2</td> <td>RH1040</td>	MP2=M+2	RH1040
CO107	6* <td>MPI=M+1</td> <td>RH1050</td>	MPI=M+1	RH1050
CO110	7* <td>NP1=N+1</td> <td>RH1060</td>	NP1=N+1	RH1060
CO111	8* <td>NP2=N+2</td> <td>RH1070</td>	NP2=N+2	RH1070
CO112	9* <td>STAR=*</td> <td>RH1071</td>	STAR=*	RH1071
CO113	10* <td>CRIT=0.0</td> <td>RH1080</td>	CRIT=0.0	RH1080
CO114	11* <td>L1=L(11)</td> <td>RH1090</td>	L1=L(11)	RH1090

```

00115      L2=L(2)
00116      L3=L(3)
00117      L(5)=1
00120      L(8)=1
00121      L(12)=1
00122      L(13)=1
00123      L(14)=1
00124      NPL=N+1
00125      NPLP1=NPL+1
00126      NPLPG=NPL+L2
00127      NMGI=N+M+L2+1
00130      L1OFF=14
00131      L2OFF=110FF+M+N+L2
00132      DO 1 I=1,M
00135      1 A(1,NP2)=A(1,NP1)
00137      DO 2 J=1,NP1
00142      2 A(MP2,J)=A(MP1,J)
00144      KTI=1
00145      KTI=0
00146      WRITE(A,1001)
00150      DO 100 I=1,M
00153      1002 WRITE(A,1003) I,R(1)
00160      WRITE(A,5)
00162      WRITE(A,1004) ROUNO
00165      WRITE(A,5)
00167      WRITE(A,3) KTI,CRIT
00173      WRITE(A,4)
00173      C .....
00173      C **SOLVE ORIGINAL PROBLFM**
00173      C .....
00175      HT=NT-1
00176      NT=NT-1
00177      CALL RVSMXP(A,MA,B,MB,MT,NT,L,X,TOLP,INV,0)
00203      WRITE(A,5)
00202      IF(L(3),LT,2) GO TO 200
00204      WRITE(A,950) (STAR,I=1,40)
00212      IF(L(3),EQ,3) GO TO 15
00212      C **INFEASIBILITY METHOD FOR INFEASIBLE ORIGINAL PROBLEM**
00214      NT=NT+1
00215      A(MP1,NP1)=-1.0
00216      DO 6 J=1,N
00221      6 A(MP1,J)=0.0
00223      DO 7 I=1,M
00226      7 A(I,NP1)=-R(1)
00227      CONTINUE
00231      L(1)=L1
00231      L(2)=L2
00233      L(3)=L3
00234      INV=0
00235      CALL RVSMXP(A,MA,B,MB,MT,NT,L,X,TOLP,INV,0)
00236      WRITE(A,14)
00240      NT=NT-1
00241      IF(L(3),EQ,2) GO TO 9
00243      RATMIN=X(MP1)
00244      CRIT=RATMIN
00245      IF(CRIT,LE,POUND) GO TO 8
00247      WRITE(A,10) CRIT
00252      WRITE(A,950) (STAR,I=1,40)

```

RH1100
 RH1110
 RH1120
 RH1130
 RH1140
 RH1150
 RH1160
 RH1170
 RH1180
 RH1190
 RH1200
 RH1210
 RH1220
 RH1230
 RH1240
 RH1250
 RH1260
 RH1270
 RH1271
 RH1272
 RH1273
 RH1274
 RH1275
 RH1276
 RH1277
 RH1278
 RH1280
 RH1290
 RH1300
 RH1310
 RH1320
 RH1330
 RH1340
 RH1350
 RH1360
 RH1361
 RH1370
 RH1380
 RH1390
 RH1400
 RH1410
 RH1420
 RH1430
 RH1440
 RH1450
 RH1460
 RH1470
 RH1480
 RH1490
 RH1500
 RH1510
 RH1520
 RH1530
 RH1540
 RH1550
 RH1560
 RH1570
 RH1571

```

00260      GO TO 9
00261      A WRITE(4,19) CRIT
00262      9 A(MP1,MPI)=A(MP2,MPI)
00263      DO 11 I=1,M
00264      11 A(MPI,I)=A(MP2,I)
00265      IF(L(3).EQ.2) GO TO 15
00270      IF(CRIT.GT.ROUND) GO TO 999
00271      DO 12 I=1,M
00272      12 A(I,MPI)=A(I,MP2)
00273      DO 13 I=1,M
00274      13 CONTINUE
00275      A(I,MPI)=A(I,MPI)+RATHIN*R(1)
00276      L(1)=L1
00277      L(2)=L2
00278      L(3)=L3
00279      INV=0
00280      C **SOLVE PROBLEM WITH THETA FROM INFEASIBILITY METHOD**
00281      CALL RVSMPL(A,M,R,M,B,NT,NT,L,X,TOLP,INV,0)
00282      WRITE(4,950) (STAR,I=1,40)
00283      C *****
00284      C **COMPUTE Y AND STORE IN COLUMN MPI OF B**
00285      C *****
00286      GO TO 200
00287      199 K1=KT
00288      KT=KT+1
00289      DO 200 J=1,M
00290      200 B(I,MPI)=0.0
00291      DO 210 J=1,M
00292      210 B(I,MPI)=R(1,MPI)+(B(I,J)*R(J))
00293      220 CONTINUE
00294      CRY=0.0
00295      DO 230 J=1,M
00296      230 K=IL2OFF+J
00297      IF(K.GT.N) GO TO 230
00298      CRY=CRY+(A(MP2,K)*R(J,MPI))
00299      230 CONTINUE
00300      COMPUT=X(NMGI)-(CRY*CRIT)
00301      ISG=1.0
00302      IF(COMPUT.LT.0.0) ISG=-1.0
00303      IF(INSET.EQ.1.AND.KT1.GT.0) WRITE(6,1005)KT1,CRIT,LETI,INTER,LET2,KL
00304      LEAVE,X(NMGI),CRY,ISG,COMPUT
00305      IF(INSET.EQ.1.AND.KT1.GT.0) WRITE(6,4)
00306      IF(INSET.EQ.1.AND.KT1.GT.0) GO TO 1006
00307      WRITE(4,231) KT1,KT,CRY,ISG,COMPUT
00308      WRITE(4,950) (STAR,I=1,40)
00309      C **FIND MINIMUM RATIO AND LEAVING VARIABLE**
00310      1006 K=0
00311      RATHIN=1E35
00312      DO 300 I=1,M
00313      300 IF(B(I,MPI).GE.0.0) GO TO 300
00314      RATIO=-R(1,MPI)/R(1,MPI)
00315      IF(RATIO.GT.RATHIN) GO TO 300
00316      RATHIN=RATIO
00317      K=I
00318      LEAVE=1
00319      300 CONTINUE
00320      KLEAVE=L(IL2OFF+LEAVE)
00321
00322
00323
00324
00325
00326
00327
00328
00329
00330
00331
00332
00333
00334
00335
00336
00337
00338
00339
00340
00341
00342
00343
00344
00345
00346
00347
00348
00349
00350
00351
00352
00353
00354
00355
00356
00357
00358
00359
00360
00361
00362
00363
00364
00365
00366
00367
00368
00369
00370
00371
00372
00373
00374
00375
00376
00377
00378
00379
00380
00381
00382
00383
00384
00385
00386
00387
00388
00389
00390
00391
00392
00393
00394
00395
00396
00397
00398
00399
00400
00401
00402
00403
00404
00405
00406
00407
00408
00409
00410
00411
00412
00413
00414
00415
00416
00417
00418
00419
00420
00421
00422
00423
00424
00425
00426
00427
00428
00429
00430
00431
00432
00433
00434
00435
00436
00437
00438
00439
00440
00441
00442
00443
00444
00445
00446
00447
00448
00449
00450
00451
00452
00453
00454
00455
00456
00457
00458
00459
00460
00461
00462
00463
00464
00465
00466
00467
00468
00469
00470
00471
00472
00473
00474
00475
00476
00477
00478
00479
00480
00481
00482
00483
00484
00485
00486
00487
00488
00489
00490
00491
00492
00493
00494
00495
00496
00497
00498
00499
00500
00501
00502
00503
00504
00505
00506
00507
00508
00509
00510
00511
00512
00513
00514
00515
00516
00517
00518
00519
00520
00521
00522
00523
00524
00525
00526
00527
00528
00529
00530
00531
00532
00533
00534
00535
00536
00537
00538
00539
00540
00541
00542
00543
00544
00545
00546
00547
00548
00549
00550
00551
00552
00553
00554
00555
00556
00557
00558
00559
00560
00561
00562
00563
00564
00565
00566
00567
00568
00569
00570
00571
00572
00573
00574
00575
00576
00577
00578
00579
00580
00581
00582
00583
00584
00585
00586
00587
00588
00589
00590
00591
00592
00593
00594
00595
00596
00597
00598
00599
00600
00601
00602
00603
00604
00605
00606
00607
00608
00609
00610
00611
00612
00613
00614
00615
00616
00617
00618
00619
00620
00621
00622
00623
00624
00625
00626
00627
00628
00629
00630
00631
00632
00633
00634
00635
00636
00637
00638
00639
00640
00641
00642
00643
00644
00645
00646
00647
00648
00649
00650
00651
00652
00653
00654
00655
00656
00657
00658
00659
00660
00661
00662
00663
00664
00665
00666
00667
00668
00669
00670
00671
00672
00673
00674
00675
00676
00677
00678
00679
00680
00681
00682
00683
00684
00685
00686
00687
00688
00689
00690
00691
00692
00693
00694
00695
00696
00697
00698
00699
00700
00701
00702
00703
00704
00705
00706
00707
00708
00709
00710
00711
00712
00713
00714
00715
00716
00717
00718
00719
00720
00721
00722
00723
00724
00725
00726
00727
00728
00729
00730
00731
00732
00733
00734
00735
00736
00737
00738
00739
00740
00741
00742
00743
00744
00745
00746
00747
00748
00749
00750
00751
00752
00753
00754
00755
00756
00757
00758
00759
00760
00761
00762
00763
00764
00765
00766
00767
00768
00769
00770
00771
00772
00773
00774
00775
00776
00777
00778
00779
00780
00781
00782
00783
00784
00785
00786
00787
00788
00789
00790
00791
00792
00793
00794
00795
00796
00797
00798
00799
00800
00801
00802
00803
00804
00805
00806
00807
00808
00809
00810
00811
00812
00813
00814
00815
00816
00817
00818
00819
00820
00821
00822
00823
00824
00825
00826
00827
00828
00829
00830
00831
00832
00833
00834
00835
00836
00837
00838
00839
00840
00841
00842
00843
00844
00845
00846
00847
00848
00849
00850
00851
00852
00853
00854
00855
00856
00857
00858
00859
00860
00861
00862
00863
00864
00865
00866
00867
00868
00869
00870
00871
00872
00873
00874
00875
00876
00877
00878
00879
00880
00881
00882
00883
00884
00885
00886
00887
00888
00889
00890
00891
00892
00893
00894
00895
00896
00897
00898
00899
00900
00901
00902
00903
00904
00905
00906
00907
00908
00909
00910
00911
00912
00913
00914
00915
00916
00917
00918
00919
00920
00921
00922
00923
00924
00925
00926
00927
00928
00929
00930
00931
00932
00933
00934
00935
00936
00937
00938
00939
00940
00941
00942
00943
00944
00945
00946
00947
00948
00949
00950
00951
00952
00953
00954
00955
00956
00957
00958
00959
00960
00961
00962
00963
00964
00965
00966
00967
00968
00969
00970
00971
00972
00973
00974
00975
00976
00977
00978
00979
00980
00981
00982
00983
00984
00985
00986
00987
00988
00989
00990
00991
00992
00993
00994
00995
00996
00997
00998
00999
01000
01001
01002
01003
01004
01005
01006
01007
01008
01009
01010
01011
01012
01013
01014
01015
01016
01017
01018
01019
01020
01021
01022
01023
01024
01025
01026
01027
01028
01029
01030
01031
01032
01033
01034
01035
01036
01037
01038
01039
01040
01041
01042
01043
01044
01045
01046
01047
01048
01049
01050
01051
01052
01053
01054
01055
01056
01057
01058
01059
01060
01061
01062
01063
01064
01065
01066
01067
01068
01069
01070
01071
01072
01073
01074
01075
01076
01077
01078
01079
01080
01081
01082
01083
01084
01085
01086
01087
01088
01089
01090
01091
01092
01093
01094
01095
01096
01097
01098
01099
01100
01101
01102
01103
01104
01105
01106
01107
01108
01109
01110
01111
01112
01113
01114
01115
01116
01117
01118
01119
01120
01121
01122
01123
01124
01125
01126
01127

```

```

00440 128* CRIT=CRIT+RATMIN RH2090
00441 129* IF(K.E.0) GO TO 302 RH2100
00443 130* IF(CRIT.LE.POUND) GO TO 310 RH2110
00445 131* WRITE(4,301) KT,ROUND RH2120
00451 132* WRITE(4,950) (STAR,I=1,40) RH2121
00457 133* GO TO 99 RH2130
00460 134* 310 IFINSET.EQ.11 GO TO 1007 RH2135
00462 135* WRITE(4,31) KT,CRIT RH2140
00462 136* C ..... RH2150
00462 137* C ..COMPUTE INDICATORS AND DENOMINATORS FOR ENTERING VARIABLES.. RH2160
00462 138* C ..... RH2170
00466 139* 1007 RATMAX=-1E35 RH2180
00467 140* DO 540 J=1,MPLPG RH2190
00472 141* IFIX(J).GE.O.0) GO TO 560 RH2200
00474 142* JL=L(L)OFF+J1 RH2210
00475 143* IF(J1.GT.O) GO TO 540 RH2220
00475 144* C ..REAL VARIABLES.. RH2230
00477 145* XX=O.C RH2240
00500 146* DO 510 I=1,M RH2250
00503 147* 510 XX=XX+O(LEAVE,I)*A(I,J) RH2260
00505 148* IF(XX.E.O.O) GO TO 560 RH2270
00507 149* YY=A(M,I,J) RH2280
00510 150* DO 520 I=1,M RH2290
00513 151* 520 YY=YY+(MPL,I)*A(I,J) RH2300
00515 152* GO TO 550 RH2310
00515 153* C ..SLACK AND SUPPLUS VARIABLES.. RH2320
00516 154* 540 J2=J-N RH2330
00517 155* XX=(-1)*((J)+1)*O(LEAVE,J2) RH2340
00520 156* IF(XX.E.O.O) GO TO 560 RH2350
00522 157* YY=(-1)*((J)+1)*O(MPL,J2) RH2360
00523 158* 550 RATIO=-YY/XX RH2370
00524 159* IF(RATIO.LE.RATMAX) GO TO 560 RH2380
00526 160* RATMAX=RATIO RH2390
00527 161* NTER=J RH2400
00530 162* 560 CONTINUE RH2410
00532 163* IF(RATMAX.GT.-1E35) GO TO 570 RH2420
00534 164* WRITE(4,561) RH2430
00536 165* WRITE(4,950) (STAR,I=1,40) RH2431
00544 166* GO TO 99 RH2440
00544 167* C ..... RH2450
00544 168* C ..CHANGE RIGHT HAND SIDES.. RH2460
00544 169* C ..... RH2470
00545 170* 570 DO 580 I=1,M RH2480
00550 171* A(I,MPL)=A(I,MPL)+(RATMIN*(I)) RH2490
00551 172* 580 CONTINUE RH2500
00551 173* C ..SET X VECTOR FOR RVSMPLX WITH ADVANCED START.. RH2510
00553 174* X(INTER)=1.0 RH2520
00554 175* X(KLEAVE)=-1.0 RH2530
00555 176* L(1)=L1 RH2540
00556 177* L(2)=L2 RH2550
00557 178* L(3)=0 RH2560
00560 179* L(4)=0 RH2570
00561 180* L(12)=N RH2580
00562 181* INV=1 RH2590
00563 182* IF(INTER.GT.N1 GO TO 900 RH2600
00565 183* LETI=X(I) RH2610
00566 184* GO TO 910 RH2620
00567 185* 900 LETI=S(I)

```

```

00570 186*
00571 187*
00572 188*
00573 189*
00574 190*
00575 191*
00576 192*
00577 193*
00601 194*
00603 195*
00611 196*
00613 197*
00614 198*
00616 199*
00621 200*
00624 201*
00627 202*
00631 203*
00634 204*
00635 205*
00640 206*
00646 207*
00647 208*
00652 209*
00655 210*
00663 211*
00666 212*
00670 213*
00671 214*
00672 215*
00673 216*
00674 217*
00675 218*
00676 219*
00677 220*
00678 221*
00679 222*
00700 223*
00701 224*
00702 225*
00703 226*
00704 227*
00705 228*
00706 229*
00707 230*
00708 231*
00709 232*
00710 233*
00711 234*
00712 235*
00713 236*
00714 237*
00715 238*
00716 239*
00717 240*
00718 241*
00719 242*
00720 243*

      NTER=NTER+1
      910 IF (KLEAVE.GT.N) GO TO 920
      LET2=X(1)
      GO TO 930
      920 LET2=X(5)
      KLEAVE=KLEAVE+N
      930 IF (NSET.EQ.1) L(13)=0
      IF (NSET.EQ.1) GO TO 1008
      WRITE(4,931) LET1,NTER,LET2,KLEAVE
      WRITE(4,5)
      WRITE(4,5)
      1008 CALL P5MPX(A,MA,B,MR,MT,NT,L,X,TOLP,INV,0)
      WRITE(4,5)
      IF (NSET.EQ.1.AND.KT1.EQ.0) WRITE(6,1009)
      IF (NSET.EQ.1.AND.KT1.EQ.0) WRITE(6,1010)
      IF (NSET.EQ.1.AND.KT1.EQ.0) WRITE(6,4)
      IF (NSET.EQ.1) GO TO 199
      WRITE(4,951) X(NMGI)
      GO TO 199
      302 WRITE(4,18) KT
      WRITE(4,950) (STAR,I=1,40)
      GO TO 999
      15 IF (L(13).EQ.2) WRITE(6,16)
      IF (L(13).EQ.3) WRITE(6,17)
      WRITE(4,950) (STAR,I=1,40)
      999 DO JCC I=1,M
      1000 A(1,NPI)=A(1,MP2)
      3 FORMAT(1H9,'CRITICAL VALUE ('',13,'') OF THETA = ',E14.8)
      4 FORMAT(1H9)
      5 FORMAT(1H9)
      10 FORMAT(1H9) WHEN THETA CRITICAL =',F15.5, 'YOUR PROBLEM BECOMES FEAS
      11 RLE RUT THIS BYPASSES YOUR LIMIT.'')
      14 FORMAT(1H9)
      16 FORMAT(1H9) PROBLEM REMAINS INFEASIBLE THROUGHOUT RANGE OF THETA. NO
      17 STUDY PERFORMED.'')
      17 FORMAT(1H9) PROBLEM IS UNROUTED. NO STUDY PERFORMED.'')
      18 FORMAT(1H9) CRITICAL VALUE ('',13,'') OF THETA IS INFINITE.'')
      19 FORMAT(1H9) WHEN THETA CRITICAL =',F15.5, 'YOUR PROBLEM BECOMES FEAS
      21 RLE AND THE STUDY CONTINUES.'')
      231 FORMAT(1H9) OBJECTIVE FUNCTION VALUE BETWEEN CRITICAL VALUES ('',13,'')
      ) AND ('',13,'') OF THETA OBTAINED FROM',E15.8,'THETA ',A1,E15.8)
      301 FORMAT(1H9) CRITICAL VALUE ('',13,'') BYPASSES YOUR LIMITATION, THE
      56) PROBLEM REMAINS OPTIMAL AT YOUR LIMIT OF',F10.5)
      950 FORMAT(1H9) PROBLEM BECOMES INFEASIBLE BEYOND THIS CRITICAL VALUE.'')
      931 FORMAT(1H9) AT THAT VALUE,'A3,14,'') ENTERS THE BASIS AND',A3,14,'') LEA
      VES.'')
      951 FORMAT(1H9) OBJECTIVE FUNCTION VALUE =',E14.8)
      1001 FORMAT(1H9) THE 9 VECTOR IS AS FOLLOWS.'')
      1003 FORMAT(1H9) R(14,'') =',F6.2)
      1004 FORMAT(1H9) R(14,'') =',F6.2)
      1005 FORMAT(1H9),F15.4,5X,A3,14,''),3X,A3,14,''),F14.4,6X,F10.4,'') THETA
      1',A1,F10.4)
      1009 FORMAT(1H9) CRIT VAL CRIT VAL BASIS INFORMATION OBJ FUNC
      1VAL OBJ FUNC VAL BETWEEN THIS')
      1010 FORMAT(1H9) NO. ENTERS LEAVES',15X,'
      1 AND NEXT CRIT VAL')
      MT=MT+1
      NT=NT+1

```

RH2630
 RH2640
 RH2650
 RH2660
 RH2670
 RH2680
 RH2682
 RH2684
 RH2690
 RH2691
 RH2700
 RH2710
 RH2712
 RH2714
 RH2716
 RH2718
 RH2720
 RH2721
 RH2730
 RH2731
 RH2732
 RH2740
 RH2750
 RH2751
 RH2760
 RH2770
 RH2780
 RH2790
 RH2800
 RH2810
 RH2820
 RH2830
 RH2840
 RH2850
 RH2860
 RH2870
 RH2880
 RH2890
 RH2900
 RH2910
 RH2920
 RH2930
 RH2940
 RH2941
 RH2950
 RH2960
 RH2961
 RH2962
 RH2964
 RH2966
 RH2968
 RH2970
 RH2972
 RH2974
 RH2976
 RH2978
 RH2980
 RH2982

APPENDIX D: Timing Considerations

APPENDIX D

TIMING CONSIDERATIONS

RANGES SAMPLES: Time Per Right-Hand Side
and Objective Function Coefficient

PROBLEM SIZE		ELAPSED TIME RANGES (SEC.)	TIME PER R.H.S.	TIME PER COEFFICIENT
m	n			
10	20	.1388	.0031	.0054
15	20	.1674	.0031	.0060
20	20	.2216	.0035	.0075
20	30	.2750	.0032	.0071
20	80	.9258	.0033	.0108
50	150	4.0446	.0039	.0257

APPENDIX D (continued)

PAROBJ SAMPLES: Time Per Critical Value

PROBLEM	SIZE	ELAPSED TIME PAROBJ (SEC.)	NO. OF COMPUTED CRITICAL VALUES	TIME PER CRITICAL VALUE
m	n			
10	20	.7608	3	.2536
15	20	.0554	1	.0554
20	20	14.1665	5	2.8333
20	30	12.5100	9	1.3900
20	80	9.4508	8	1.1814
50	150	190.2708	23	8.2726

APPENDIX D (continued)

PARRHS SAMPLES: Time Per Critical Value

PROBLEM SIZE		ELAPSED TIME PARRHS (SEC.)	NO. OF COMPUTED CRITICAL VALUES	TIME PER CRITICAL VALUE
m	n			
10	20	.7930	2	.3965
15	20	1.8704	3	.6235
20	20	4.3180	10	.3925
20	30	1.9914	1	1.9914
20	80	4.7322	1	4.7322
50	150	73.2580	2	36.6290

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET		1. PUBLICATION OR REPORT NO. NBSIR 75-666	2. Gov't Accession No.	3. Recipient's Accession No.
4. TITLE AND SUBTITLE "Post Optimality and Parametric Analysis with the National Bureau of Standards' Linear Programming Subroutine"			5. Publication Date February 1975	
			6. Performing Organization Code	
7. AUTHOR(S) Tyrone B. Ayers			8. Performing Organ. Report No.	
9. PERFORMING ORGANIZATION NAME AND ADDRESS NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234			10. Project/Task/Work Unit No.	
			11. Contract/Grant No.	
12. Sponsoring Organization Name and Complete Address (Street, City, State, ZIP) Same as No. 9			13. Type of Report & Period Covered Final	
			14. Sponsoring Agency Code	
15. SUPPLEMENTARY NOTES				
16. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.) This report is a sequel to NBS Report 10695 (February 1972), "The National Bureau of Standards' Linear and Quadratic Programming Subroutines," which documented one phase of an effort to provide users, of the facility operated by the National Bureau of Standards' Computer Services Division, with reliable, clearly-described solution algorithms for selected frequently-arising classes of special mathematical problems. The present report presents subroutines which perform post-optimality analysis and parametric programming studies on linear programming problems solved by the National Bureau of Standards' RVSMPPX subroutine. (The present versions of these codes use internal storage only.)				
17. KEY WORDS (six to twelve entries; alphabetical order, capitalize only the first letter of the first key word unless a proper name; separated by semicolons) Algorithms; linear programming; parametric programming; post-optimality analysis.				
18. AVAILABILITY ☒ Unlimited ☐ For Official Distribution. Do Not Release to NTIS ☐ Order From Sup. of Doc., U.S. Government Printing Office Washington, D.C. 20402, SD Cat. No. C13 ☒ Order From National Technical Information Service (NTIS) Springfield, Virginia 22151		19. SECURITY CLASS (THIS REPORT) UNCLASSIFIED 20. SECURITY CLASS (THIS PAGE) UNCLASSIFIED		21. NO. OF PAGES 47 22. Price \$ 3.75



