NATIONAL BUREAU OF STANDARDS REPORT

10 695

THE NATIONAL BUREAU OF STANDARDS' LINEAR AND QUADRATIC PROGRAMMING SUBROUTINES

Technical Report

to

Computer Services Division

Center for Computer Sciences and Technology

U.S. DEPARTMENT OF COMMERCE

NATIONAL BUREAU OF STANDARDS

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards [1] was established by an act of Congress March 3, 1901. Today, in addition to serving as the Nation's central measurement laboratory, the Bureau is a principal focal point in the Federal Government for assuring maximum application of the physical and engineering sciences to the advancement of technology in industry and commerce. To this end the Bureau conducts research and provides central national services in four broad program areas. These are: (1) basic measurements and standards, (2) materials measurements and standards, (3) technological measurements and standards, and (4) transfer of technology.

The Bureau comprises the Institute for Basic Standards, the Institute for Materials Research, the Institute for Applied Technology, the Center for Radiation Research, the Center for Computer Sciences and Technology, and the Office for Information Programs.

THE INSTITUTE FOR BASIC STANDARDS provides the central basis within the United States of a complete and consistent system of physical measurement; coordinates that system with measurement systems of other nations; and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. The Institute consists of an Office of Measurement Services and the following technical divisions:

Applied Mathematics—Electricity—Metrology—Mechanics—Heat—Atomic and Molecular Physics—Radio Physics [2]—Radio Engineering [2]—Time and Frequency [2]—Astrophysics [2]—Cryogenics.[2]

THE INSTITUTE FOR MATERIALS RESEARCH conducts materials research leading to improved methods of measurement standards, and data on the properties of well-characterized materials needed by industry, commerce, educational institutions, and Government; develops, produces, and distributes standard reference materials; relates the physical and chemical properties of materials to their behavior and their interaction with their environments; and provides advisory and research services to other Government agencies. The Institute consists of an Office of Standard Reference Materials and the following divisions.

Analytical Chemistry—Polymers—Metallurgy—Inorganic Materials—Physical Chemistry.

THE INSTITUTE FOR APPLIED TECHNOLOGY provides technical services to promote the use of available technology and to facilitate technological innovation in industry and Government; cooperates with public and private organizations in the development of technological standards, and test methodologies; and provides advisory and research services for Federal, state, and local government agencies. The Institute consists of the following technical divisions and offices:

Engineering Standards—Weights and Measures — Invention and Innovation — Vehicle Systems Research—Product Evaluation—Building Research—Instrument Shops—Measurement Engineering—Electronic Technology—Technical Analysis

THE CENTER FOR RADIATION RESEARCH engages in research, measurement, and application of radiation to the solution of Bureau mission problems and of problems of other agencies and institutions. The Center consists of the following divisions:

Reactor Radiation—Linac Radiation—Nuclear Radiation—Applied Radiation.

THE CENTER FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides technical services designed to aid Government agencies in the selection, acquisition, and effective use of automatic data processing equipment; and serves as the principal focus for the development of Federal standards for automatic data processing equipment, techniques, and computer languages. The Center consists of the following offices and divisions:

Information Processing Standards—Computer Information — Computer Services — Systems Development—Information Processing Technology.

THE OFFICE FOR INFORMATION PROGRAMS promotes optimum dissemination and accessibility of scientific information generated within NBS and other agencies of the Federal government; promotes the development of the National Standard Reference Data System and a system of information analysis centers dealing with the broader aspects of the National Measurement System, and provides appropriate services to ensure that the NBS staff has optimum accessibility to the scientific information of the world. The Office consists of the following organizational units:

Office of Standard Reference Data—Clearinghouse for Federal Scientific and Technical Information [3]—Office of Technical Information and Publications—Library—Office of Public Information—Office of International Relations.

---

[1] Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D.C. 20234.
[2] Located at Boulder, Colorado 80302.
[3] Located at 5285 Port Royal Road, Springfield, Virginia 22151.

NATIONAL BUREAU OF STANDARDS REPORT

NBS PROJECT		NBS REPORT
2053588	February, 1972	10 695

THE NATIONAL BUREAU OF STANDARDS' LINEAR AND QUADRATIC PROGRAMMING SUBROUTINES

W. G. Hall
R. H. F. Jackson
P. B. Saunders
Applied Mathematics Division

U.S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS

ACKNOWLEDGEMENT

The authors would like to acknowledge the aid provided by Richard Ku of the Technical Analysis Division; not only for proofing the manuscript and providing numerous unusual test problems, but also for playing the role of "typical user" on whom we tried our ideas.

ABSTRACT

This report documents one phase of an effort to provide users, of the facility operated by the National Bureau of Standards' Computer Services Division, with reliable,well-tested,clearly-described solution algorithms for selected frequently-arising classes of special mathematical problems. The report presents algorithms for the simplex and revised simplex methods of linear programming, as well as their adaptations to quadratic programming. Set up as subroutines, the present versions of these codes use internal storage only, with resultant limitations on the size of the problems which can be treated.

Key Words: Algorithms, linear programming, nonlinear programming, quadratic programming.

CONTENTS

1. INTRODUCTION

The "package" documented in this report consists of two subroutines: SIMPLX, which is an implementation of the simplex method for linear programming, and RVSMPX, which is an implementation of the revised simplex method for linear programming. In addition, both subroutines can solve some types of quadratic programming problems.

The primary goal during the development of this package was to provide reliable user-oriented subroutines for solving linear programming problems. Other desirable program attributes, such as efficiency with respect to core usage and time, and the extension of problem size, were considered to be of less importance. An abundance of monitoring prints is available; each is optional and may be printed or suppressed independently of the other outputs. All output appears on unit 6; there are no other references to peripheral devices within the subroutines. Since each program is a subroutine, there is a return to the calling routine regardless of whether the exit is normal (a solution has been found) or abnormal (the problem is infeasible, unbounded, ill-stated, or numerically unstable). Hence the user must, in his calling routine, test a status parameter upon return from the subroutine to determine the cause of the return.

The calling statements for the subroutines are as similar as possible. In all of these, A is the augmented constraint matrix, X is the output vector, L contains the output switches, etc. In fact, in the interests of standardization, the two subroutines were designed

to be as similar as possible with respect to variable names, error-handling, and output messages, without sacrificing any speed or efficiency intrinsic to the respective algorithms.

It is due to this similarity between SIMPLX and RVSMPX, and to an attempt at avoiding duplication, that the documentation here presented is in a somewhat "factored" form: sections 4, 5, and 6 contain information that is common to both subroutines and would have been duplicated had it not been "factored out" into its own section.

It is well-known [7] that the revised simplex method is faster and more efficient computationally than the original simplex method. The main reason for the inclusion of the simplex method is that since its computations are actually performed in the memory space occupied by the A matrix (the A matrix is the simplex tableau) and there is no need to store separately the inverse of the basis matrix, SIMPLX can solve somewhat larger problems than RVSMPX. This should be advantageous to the user whose problem is slightly too large for RVSMPX but not large enough to warrant the use of peripheral storage.

In the interests of completeness we mention here the rule used by RVSMPX and SIMPLX to break ties for the variable to leave the basis. The variable that is chosen to leave the basis when there is a tie is the variable whose pivot element is largest in magnitude. Although this rule does not guarantee the prevention of cycling, practical experience thus far has shown it to be effective.

This report is intended as a user's manual for RVSMPX and SIMPLX. The reader should have some familiarity with both linear and computer programming. Although some parts of the text do require more than a basic familiarity, the user will normally have no need to understand these sections.

2. THE SIMPLEX SUBROUTINE (SIMPLX)

SIMPLX finds, using the simplex method for linear programming, the maximum value of a linear objective function subject to a set of linear constraints with non-negative variables and non-negative right-hand-sides. The problem is:

$$\text{maximize:} \quad \sum_{j=1}^{n} c_j x_j$$

$$\text{subject to:} \quad \sum_{j=1}^{n} a_{ij} x_i \leq b_i, \quad i = 1, 2, \ldots, \ell$$

$$\sum_{j=1}^{n} a_{ij} x_i \geq b_i, \quad i = \ell + 1, \ell + 2, \ldots, \ell + g$$

$$\sum_{j=1}^{n} a_{ij} x_i = b_i, \quad i = \ell + g + 1, \ell + g + 2, \ldots \ell + g + e$$

$$x_j \geq 0 \ , \quad j = 1, 2, \ldots, n$$

$$b_i \geq 0 \ , \quad i = 1, 2, \ldots, m$$

where n is the number of "real" (i.e.,original) variables; m, which is equal to $\ell + g + e$, is the total number of constraints; ℓ is the number of contraints with $\leq$ signs (LE constraints); g is the number of constraints with $\geq$ signs (GE constraints); e is the number of constraints with $=$ signs (EQ constraints). Note the requirement that all $b_i \geq 0$.

Within the SIMPLX subroutine, the constraint set is transformed into a system of equations with an initial basic feasible solution through the addition of slack variables for the LE constraints, surplus variables for the GE constraints, and artificial variables for the GE and EQ constraints. The subroutine then proceeds with the "Two-Phase Method" for solving the problem.

During Phase I, an artificial objective function is constructed in such a way that when simplex iterations have driven it to zero, the basis contains no non-zero artificial variables,and is a basic feasible solution to the original problem. However, if the maximum of the artificial objective function is not zero, there is no solution to the original problem, and that problem is declared infeasible. On the other hand, if at the end of Phase I one or more artificial variables remain in the solution set (basis) at a zero level, then the constraints associated with those artificial variables are redundant (i.e., they are not needed). These variables will remain in the basis at a zero level throughout Phase II, and will be members of the final solution. Note that this allows an arbitrary relationship between the number of constraints and the number of variables in the original problem. Phase I is not required if all the constraints in the original systems are LE constraints, since the slack variables alone provide an initial basic feasible solution in this case.

A few words need to be said about the redundancy indication mentioned above. At the "textbook" end of Phase I (all the indicators,the "$z_j - c_j$," non-positive), the NBS simplex and revised simplex subroutines go one step further. If at that point there are any artificial variables in the basis at an "effectively" zero level (see section 5.9), an attempt is made to remove them by replacing them with any non-basic, non-artificial variable whose indicator is zero. Any artificial variable removed in this way was in the basis as a result of degeneracy, and any artificial variable that remains in the basis at a zero level after this procedure, is present as a result of redundancy.

Phase II consists of simplex iterations whose task is to maximize the real (original) objective function. If no maximum exists, the problem is declared unbounded.

SIMPLX consists of approximately 400 FORTRAN V statements that compile into slightly less than 2,000 computer words. Data storage requires approximately mn+6m+5n words. With a small main program on the NBS UNIVAC 1108, under the EXEC II operating system, mn may be as large as 44,000.

The calling statement is as follows:

CALL SIMPLX (A,MA,MT,NT,L,X,TØLP,KQP).

For an explanation of the parameters in the calling statement and a discussion of their meaning on entering and exiting from the subroutine, see Section 5.

For the purpose of illustration, and to give the user an idea of what the output from a run will look like, the output from an example problem is included here. The problem is:

maximize: $x_1 + 1.5x_2 + 5\,x_3 + 2\,x_4$

subject to :
$$\begin{aligned} 3x_1 + 2x_2 + x_3 + 4x_4 &\leq 6 \\ 2x_1 + x_2 + 5x_3 + x_4 &\leq 4 \\ 2x_1 + 6x_2 - 4x_3 + 8x_4 &= 0 \\ x_1, \quad x_2, \quad x_3, \quad x_4, &\geq 0 \end{aligned}$$

The A matrix was set up as follows:

3	2	1	4	6
2	1	5	1	4
2	6	-4	8	0
1	1.5	5	2	0
0	0	0	0	0

The other input parameters were:

MA=100 L(1)=2 L(4) through L(14)=1

MT=5 L(2)=0 TOLP=0

NT=5 L(3)=0 KQP=0

The following page contains the output from this run.

```
EPSILON =    .325000-04  CAPITAL EPSILON =    .325000-03      0 NON-ZERO ENTRIES ARE EFFECTIVELY EQUAL TO ZERO.


PHASE 1 ITERATION   1.  PIVOT=   8.00000     OBJECTIVE FUNCTION=   .00000000      X(  4) ENTERED THE BASIS, A(  1) LEFT.

BASIC VARIABLES

   X(  4)=     .000000

   S(  1)=    6.000000    S(  2)=    4.000000



*****************************************************************************************************************
*                                                                                                               *
*             END OF PHASE 1.    OBJECTIVE FUNCTION =      .00000000       THERE WERE   1 ITERATIONS.           *
*             MINIMUM PIVOT WAS   8.0000    AT ITERATION   1.   REAL OBJECTIVE FUNCTION =   .00000000         *
*                                                                                                               *
*****************************************************************************************************************

BASIC VARIABLES

   X(  4)=     .000000

   S(  1)=    6.000000    S(  2)=    4.000000



PHASE 2 ITERATION   1.  PIVOT=   5.50000      OBJECTIVE FUNCTION=   4.3636363      X(  3) ENTERED THE BASIS, S(  2) LEFT.

BASIC VARIABLES

   X(  3)=     .727273    X(  4)=     .363636

   S(  1)=    3.818182



*****************************************************************************************************************
*                                                                                                               *
*             END OF PHASE 2.    OBJECTIVE FUNCTION =      4.3636363       THERE WERE   1 ITERATIONS.           *
*             MINIMUM PIVOT WAS   5.5000    AT ITERATION   1.                                                 *
*                                                                                                               *
*****************************************************************************************************************

BASIC VARIABLES

   X(  3)=     .727273    X(  4)=     .363636

   S(  1)=    3.818182
```

3. THE REVISED SIMPLEX SUBROUTINE (RVSMPX)

The formulation of the problem, restrictions pertaining thereto, and definitions of the variables are the same here as that in section 2 for SIMPLX. The method of solution is, of course, different. The method used is the revised simplex method as presented in chapter 3 of [3], with one important modification. The modification is the provision for re-inverting the basis matrix every [m/2]+5 iterations, in order to reduce round-off error. (The value [m/2]+5 is one that appears in other codes, see[1], and has been shown in practice to be a reasonably good choice. However, if the user feels another value is more appropriate, he may use his value by changing the value of INVC on card number 6 of the subroutine.) A discussion of the advantages of re-inverting the basis matrix appears in section 7-8 of [4]. This provision also allows for the "restart" or "advanced start" capability whereby the user may choose to supply the subroutine with an initial basic feasible solution, if known, in order to speed up the algorithm. For instructions on how to use this advanced start capability, see section 5.10.

RVSMPX consists of approximately 450 FORTRAN V statements that compile into slightly more than 2,000 computer words. Data storage is approximately $mn + m^2 + 2n + 3m$ computer words.

The calling statement is:

CALL RVSMPX (A,MA,B,MB,MT,NT,L,X,TØLP,INV,KQP).

For an explanation of the parameters in this calling statement and a description of their values on entering and exiting from the subroutines, see section 5.

The output messages that may appear are described in section 6.

4. SOLVING QUADRATIC PROGRAMS.

As was mentioned in the introduction, both RVSMPX and SIMPLX have the capability to solve quadratic programming problems. This is accomplished via the parameter KQP, discussed below in section 5.

A quadratic program is of the following form:

maximize: $\sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_i x_j$

subject to the same set of constraints that appears in section 2. There is, however, one important additional restriction--the objective function above must be concave. If it is not concave, although a feasible solution to the quadratic program may be found, there is no guarantee that this solution will be optimal.

The objective function above is concave if

$$\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_i x_j \leq 0,$$

for all x_i, where i=1,2,...,n. This is equivalent to saying that the matrix of the d_{ij} is negative semi-definite. The quadratic form above may be shown to be concave by showing that it can be written as the negative of a sum of linear forms,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_i x_j = -\sum_{k} [L_k(x)]^2 ;$$

in many applications (e.g., least squares estimation) this is known a priori. Equivalently, one could attempt to diagonalize the matrix of the d_{ij} and check the resulting diagonal elements (the eigenvalues of the d_{ij} matrix). If all are non-positive, the matrix is negative semi-definite. See chapter 8 of [2] for more on this.

The method of solution that is used was originally developed by Wolfe [9]. He noted that the Kuhn-Tucker conditions (see Appendix A and [6]) for the quadratic programming problem are "almost linear", and that, under the concavity assumption mentioned above, they are necessary and sufficient--which means that if one can find a set of x_j, $j=1,2...n$, that satisfy the Kuhn-Tucker conditions, then those x_j are the optimal x_j for the original problem. Thus, the emphasis was shifted to solving a set of "almost linear" equations. This was accomplished in [9] by utilizing Phase I of the simplex method for linear programming with one modification. That modification is that certain pairs of variables cannot simultaneously be in the basis; it is implemented by restricting the choice of the entering basic variable. This approach is used in both RVSMPX and SIMPLX.

All that is necessary to solve a quadratic program is to set KQP to n, where n is as defined in this section, and to set up the constraint matrix as follows:

A_1	0	0	0	0	0	
A_2	0	0	0	0	0	b
A_3	0	0	0	0	0	
-2D	A_1^T	$-A_2^T$	A_3^T	$-A_3^T$	$-I_n$	c
0	0	0	0	0	0	*
*	*	*	*	*	*	*

where the original constraint matrix has been partitioned into A_1, A_2 and A_3, consisting of the LE, GE, and EQ constraint coefficients respectively. A^T indicates the transpose of the matrix A, D is the matrix of the d_{ij}, and I_n is an nxn identity matrix. The solution to the quadratic programming problem will be found in X(1) through X(N) where N is the number of real variables in the original problem.

For a more rigorous development of the algorithm and an explanation of some of the information that might be gleaned from the solution, see appendix A.

5. THE PARAMETERS.

Listed below are the parameters that appear in either one or both of the subroutines. After the name of each variable appears an (R) if the variable is a parameter of RVSMPX; an (S) if the variable is a parameter of SIMPLX; and an (RS) if the variable is a parameter of both RVSMPX and SIMPLX. All variables conform to the FORTRAN implicit naming conventions regarding mode.

1. A (RS)

A is the constraint matrix augmented by a right-hand-side column, an objective function row, and in the case of SIMPLX, a row to be used (by the subroutine) for the coefficients of the artificial objective function. The first l rows of A contain the LE constraint coefficients, the next g rows contain the GE constraint coefficients, and the next e rows contain the EQ constraint coefficients. Row m+1 contains the coefficients of the real objective function; row m+2, whose values are not supplied by the user, contains the artificial objective function coefficients for SIMPLX, and column n+1 contains the non-negative right-hand-sides. Hence, A must be dimensioned at least (m+1)x(n+1) for RVSMPX and (m+2)x(n+1) for SIMPLX. Pictorially, where "*" indicates a value that is not supplied by the user, A is:

a_{11}	a_{12}	.	.	.	a_{1n}	b_1
—	—	.	.	.	—	—
$a_{\ell+1,1}$	$a_{\ell+1,2}$	.	.	.	$a_{\ell+1,n}$	$b_{\ell+1}$
—	—	.	.	.	—	—
$a_{\ell+g+1,1}$	$a_{\ell+g+1,2}$	.	.	.	$a_{\ell+g+1,n}$	$b_{\ell+g+1}$
—	—	.	.	.	—	—
a_{m1}	a_{m2}	.	.	.	a_{mn}	b_m
c_1	c_2	.	.	.	c_n	*
*	*	.	.	.	*	*

We note, again, that the last row is not needed for RVSMPX.

Since the A matrix is used by SIMPLX as explicit storage for the condensed simplex tableau (see pp. 119 of [8]), the original values in the A matrix will have been completely destroyed by the time SIMPLX returns to the main program. In most cases there is nothing in the A matrix of value to the user at this point. For RVSMPX, however, the A matrix is completely unchanged during the course of the algorithm.

2. MA (RS)

This variable contains the value of the first dimension of A, as A is dimensioned in the calling routine; e.g., if A is dimensioned as A(20,50), then MA should be set to 20 in the main program before the subroutine is called.

This parameter is unchanged by either of the subroutines.

3. B (R)

This parameter is a matrix whose dimensions must be at least (m+2)x(m+2). Upon exiting from RVSMPX, B will contain the inverse of the matrix of the current basic columns in ((B(I,J),J=1,M),I=1,M); the negatives of the current values of the dual variables in (B(M+1,J),J=1,M); the current values of the basic variables in (B(I,M+2),I=1,M), in an order specified by a portion of the L vector to be explained later; the negative of the value of the real objective function in B(M+1,M+2); and the negative of the value of the artificial objective function in B(M+2,M+2).

4. MB (R)

This contains the value of the first dimension of the B matrix as it is dimensioned in the calling routine. It too is unchanged by the subroutine.

5. MT (RS)

MT is the number of rows of information in the A matrix, which is m+2 for SIMPLX and m+1 for RVSMPX. Note that MT is less than or equal to MA. This variable is not changed by the subroutines.

6. NT (RS)

The value of this variable is the number of columns in A, i.e., n+1. It too is unchanged by SIMPLX or RVSMPX.

7. L (RS)

L is a multipurpose vector that provides storage space for a number of distinct working vectors constructed by the subroutines, in addition to providing indicators and constants for both entry and exit. It also contains output switches which are used to determine how

much (if any) output from the subroutines is given. For RVSMPX, L must be dimensioned at least 14+2m+n+g; for SIMPLX, 14+2(m+n)+g. The meaning of L(1) through L(14) is the same for both subroutines for both entry and exit; the differences in the use of the L vector by RVSMPLX and SIMPLX occur beyond L(14). Although the typical user need not understand what happens to this latter portion of the L vector, a description is included here for the user who may wish to modify the subroutines.

L(1): For entry, this must contain the number of LE constraints, or ℓ. The exit value of L(1) is the number of non-zero elements in the original A matrix that are between (-EPS) and EPS and are, therefore, considered by the subroutines to be equal to zero [see 9 below]. If the exit value of L(1) is positive, the user is providing the subroutine with information it does not use. If the exit value of L(1) is large, it is suggested that the user either force the subroutine to use the enumerated small matrix entries by making EPS smaller (see 9 below), or else not provide these numbers (set them to zero). If one of these actions is not taken, numerical problems might arise.

L(2): This should contain the number of GE constraints, or g, when entering the subroutine. The exit value of this variable is the total number of iterations the subroutine performed before termination. It includes the iterations for both phases.

L(3): When entering the subroutines, this is a default print switch. If L(3) is non-zero, the subroutine sets L(4), L(5), L(8), L(12), L(13), and L(14) to 1, and L(6), L(7), L(9), L(10), and L(11) to 0. If L(3) is zero, the user sets L(4) through L(14) to obtain the desired outputs. When leaving the subroutine, the value of L(3) is very important -- it indicates what caused the termination. A value of 0 indicates an

optimal basic feasible solution has been obtained; 1 indicates that an optimal basic feasible solution has been obtained but that numerical difficulties arose during the course of the algorithm, and that the user should question the results; 2 indicates that the problem is infeasible; 3 indicates the problem is unbounded; and 4 indicates a system error.

L(4): If this parameter is non-zero when the subroutines are entered, the values of EPS, CEPS, and L(1) are printed. The value of L(4) remains unchanged by the subroutine.

L(5): For entry, if L(5) is non-zero, a warning message will be printed if necessary. This variable, too, is untouched by the subroutine.

L(6): If L(6) is greater than zero when entering RVSMPX or SIMPLX, the iteration summary will be printed after every L(6) iterations in Phase I. Neither of the subroutines changes the value of L(6).

L(7): If L(7) is greater than zero when entering the subroutines, the basic variables and their values will be printed after every L(7) iterations in Phase I. Neither RVSMPX nor SIMPLX changes this variable.

L(8): If this switch is non-zero when the subroutines are entered, the final summary for Phase I will be printed. No change occurs in the value of this switch.

L(9): This switch, if non-zero when entry occurs, will cause the basic variables and their values to be printed at the end of Phase I. Again, no change occurs in the value of this variable.

L(10) through L(13): These variables perform the same task as L(6) through L(9), except that they refer to Phase II.

L(14): When entering the subroutine, if L(14) is non-zero, error messages will be printed if errors occur. No change in the value of L(14) occurs.

Values for the remaining portions of the L vector are not supplied by the user, and, as was mentioned earier, the exit values of these variables will probably not be of any importance to the casual user of these subroutines. Nevertheless, in the interests of completeness, their descriptions are provided here.

RVSMPX and SIMPLX use this storage space differently. The description of the manner in which RVSMPX uses this space is provided first. To facilitate the discussion, let

$$\begin{aligned} M &= m \\ N &= n \\ MG &= g \\ L1\emptyset FF &= 14 \\ L2\emptyset FF &= L1\emptyset FF + n + \ell + g + g + e \\ &= 14 + n + m + g \end{aligned}$$

Then, for J = 1 through N + M + MG, if L(L1ØFF + J) equals 0, X(J) is a real variable; if 1, X(J) is a slack variable; if 2, X(J) is a surplus variable; if 3, X(J) is an artifical variable; if 4, X(J) is an artificial variable that has been removed from the basis, and, therefore, from future consideration (see p. 119 of [4]); if 5, X(J) is an artifical variable that is in the basis at a zero level and cannot be removed. For J = 1 through M, if L(L2ØFF + J) = K, X(K) isthe J^{th} basic variable, and its value appears in B(J,M + 2).

To facilitate the discussion of the manner in which SIMPLX uses this space, let

L1ØFF = 14
L2ØFF = L1ØFF + n = 14 + n,
L3ØFF = L2ØFF + m = 14 + n + m
INØFF = L3ØFF + g = 14 + n+ m + g,
NØTØFF = INØFF + m = 14 + n + m + g + m
L10 = the number of columns in the A matrix after the artificial variables have been removed by the subroutine,
L20 = the number of rows in the A matrix after redundant constrain- have been implicitly removed by the subroutine.

Then, for J = 1 through L10, if L(L1ØFF + J) = K, the J^{th} column of the current A matrix is the K^{th} column of the original A matrix. Also, for J = 1 through L20, if L(L2ØFF + J) = K, the J^{th} row of the current A matrix is the K^{th} row of the original A matrix. Since the surplus and artificial variables for each GE constraint are never needed at the same time, L(L3ØFF + J) for J = 1 through MG is set up as follows: if L(L3ØFF + J) equals 1, the artificial variable for the J^{th} GE constraint is still in the basis; if L(L3ØFF + J) equals 0, the artificial variable for the J^{th} GE constraint has been removed from the basis, and the storage space previously occupied by it is occupied by the surplus variable for that constraint. For J = 1 through M, if L(INØFF + J) = K, the J^{th} basic variable is X(K). Finally, for J = 1 through N, if L(NØTØFF + J) = K, the J^{th} non-basic variable is X(K).

8. X (RS)

This vector is the solution vector. Its dimension should be at least n + ℓ + g + g + e + 1, or equivalently, n + m + g + 1. For the SIMPLX subroutines, the values in the X vector when entry occurs have no effect on the subroutine, since SIMPLX initializes X to zero. For RVSMPX, however, the initial values in the X vector are important when the advanced start option (see 10 below) is used, and unimportant otherwise.

When termination occurs, X contains the terminal values of the variables in the original problem, including any slack, surplus, or artificial variables that were added by the subroutine. The first n elements of X contain the values of x_1 through x_n. Next in order are the slack variables for the LE constraints, then the surplus variables for the GE constraints, the artificial variables for the GE constraints, the artificial variables for the EQ constraints, and finally, in X(N + ML + 2*MG + ME + 1), the value of the objective function when termination occurs.

The scheme by which values are stored in X is as follows: for each x_i that is not in the basis, X(I) = -EPS/100; hence, for each I for which X(I) is non-negative, x_i is in the basis at its given level. This scheme was employed so as to permit both easy identification of the basic variables and immediate use of the solution vector at the same time, with no undue round-off error.

X will contain these values according to the scheme noted above regardless of the reason for termination. Of course, if termination is with L(3) greater than 1, all the values in X will be the current ones and not necessarily optimal or even feasible.

9. TØLP (RS)

This is a parameter which is used in the construction of epsilon (EPS) and capital epsilon (CEPS), variables that are used as tolerance parameters. Within the subroutines, $|a|<$EPS is equivalent to a=0. The user has the following options regarding the value of EPS:

If TØLP is less than zero, EPS = $\dfrac{|\text{TØLP}|\sum_{i,j}|a_{i,j}|}{m * n}$

If TØLP equals 0, EPS = $\frac{10^{-5} \sum_{i,j} |a_{i,j}|}{m * n}$.

If TØLP is greater than 0, EPS = TØLP.

Tests have shown the default choice (TØLP=0) to be reasonably effective. The word "reasonably" is used here since there are cases when this choice is inappropriate. It is for this reason that the two other choices for TØLP and EPS have been provided. Furthermore, to aid the user in determining when an inappropriate value of EPS has been selected, warning messages have been provided (see section 6) that indicate that numerical problems have arisen. This may possibly be overcome by changing the value of EPS.

In any case, CEPS is set to 10*EPS and is used solely to determine when a number is "close to zero".

The value of TØLP is not changed within either of the subroutines.

10. INV (R)

This is a switch that allows the user to provide RVSMPX with an initial basic feasible solution. If INV is non-zero, an initial basic feasible solution will be expected. If INV equals zero, the subroutine will start from scratch with Phase I.

The initial basis is passed via the X vector as follows: for i=1 through $n + \ell + g + g + e$, if x_i is a member of the basis, then X(I) is greater than or equal to zero; if x_i is not a member of the basis, then X(I) is less than zero. The actual values are irrelevent so long as the signs are correct. RVSMPX will form a basis with these variables and proceed with Phase I or Phase II, depending on whether or not the basis contains any artificial variables.

When RVSMPX terminates, INV may have a value of 0,1, or 2. In any event, its value is irrelevant.

11. KQP (RS)

This is a switch that allows the user to solve quadratic programming problems. If KQP is zero, the subroutines will consider the problem being solved to be a linear programming problem. In order to solve a quadratic programming problem, KQP must be set to n, the number of real variables in the original quadratic programming problem. For a discussion of the quadratic programming algorithm that is used, the method of setting up the problem, and the interpretation of the results, see section 4.

The value of this switch remains unchanged throughout the subroutine.

6. OUTPUT

As indicated in section 5.7, the output messages are controlled (with one exception) by the values of L(3) through L(14), which are parameters in the subroutine call. This section dicusses the output messages that may result from the use of RVSMPX or SIMPLX. Since there are a few messages that are indigenous to one or the other of the two subroutines, the same notational scheme that was used in section 5 will be used here; i.e., if the message can result from RVSMPX, an (R) will appear after the name of the message; if the message can result from SIMPLX, an (S) will appear after the name of the message; and if the message can result from either one, an (RS) will appear.

1. The Negative Right Hand Side Message (RS).

```
* * AT LEAST ONE ELEMENT OF THE RIGHT HAND SIDE COLUMN IS LESS THAN ZERO. SUBROUTINE TERMINATES.
```

This message is exceptional in that it is not under control of any switch in the L vector. Immediately upon entry, the subroutines examine the problem data. If any entry in column NT of matrix A (i.e., the right hand sides) is negative, this message is printed, the value of L(3) becomes 2, and the subroutine returns to the calling routine. The problem should be reformulated so that all right hand sides are non-negative.

2. The Epsilon Print (RS).

```
EPSILON =    .964732-02  CAPITAL EPSILON =    .964732-01      0 NON-ZERO ENTRIES ARE EFFECTIVELY EQUAL TO ZERO.
```

This message is printed whenever L(4) is non-zero. The values of epsilon and capital epsilon are based on TØLP, a parameter in the calling sequence (see section 5.9). Major ways in which these values are used are discussed in 5, 9, and 11 below, and in the discussion of L(1) in section 5.7. The number of non-zero entries in the A matrix that are "effectively" equal to zero is stored in L(1). For a discussion of this value see section 5.7.

3. The "Infeasible Initial Basis" Message (R).

```
**********************************************************************************************************
*                                                                                                        *
*  ERROR - THE VARIABLES SPECIFIED AS COMPRISING AN INITIAL SOLUTION DO NOT FORM A BASIC FEASIBLE SOLUTION.*
*                                                                                                        *
*  THE ERROR WAS DETECTED AT ITERATION   0 OF PHASE 1. AT THAT TIME THE OBJECTIVE FUNCTION VALUE WAS -21.000000*
*  AND THE FOLLOWING VARIABLES WERE BASIC.                                                               *
*                                                                                                        *
**********************************************************************************************************

BASIC VARIABLES

   X(  6)=    2.000000     X(  8)=    1.000000     X( 10)=    3.000000     X( 12)=    2.000000     X( 14)=    6.000000

   S(  2)=    5.000000

   A(  5)=   11.000000     A(  6)=   10.000000
```

If the advanced start feature is used (INV non-zero), and the variables specified in the X vector as forming a basis do not form a feasible basis, this message is printed, (if L(14) is non-zero), L(3) is set to 2, and RVSMPX returns to the calling routine. The basic variables that are printed are the variables that had been put into the basis before the error was detected.

There are two ways in which this error may occur:

(i) The number of non-negative variables passed in the X vector is not equal to m. However, in some cases when the number is greater than m, no error will occur. That is, if the m variables that are the first to enter the basis do form a feasible basis, then they are used as the basis and no error is recorded.

(ii) The number of non-negative variables passed in the X vector is equal to m, but these variables do not form a feasible basis. In this case, at least one of the m basic variables printed will be negative.

(Note: if a case (i) error occurs, there may be negative basic variables. There is no cause for alarm; the basis forming procedure (actually a matrix inversion) allows basic variables to become negative during its course since, if the variables do form a feasible basis all the variables will be non-negative at the end of this procedure.)

4. The "Small Pivot" Message (RS):

```
* * WARNING * *  SMALL PIVOT ELEMENT AT ITERATION  10 OF PHASE 1. PIVOT =  .1234568-06
```

This message is printed if the chosen pivot element is between EPS and CEPS and if L(5) is non-zero. This is considered a warning message; computations continue, but the user is strongly advised to view the results critically.

5. The alternate Pivot Element Message (S):

```
* * WARNING * *  IN PHASE 1, THERE HAVE BEEN 10 ATTEMPTS TO FIND AN ALTERNATE PIVOT ELEMENT.
                 OF THESE, 6 FOUND NO ALTERNATE ELEMENT GREATER THAN CAPITAL EPSILON.
```

This warning message requires a rather detailed explanation. If the pivot element that is chosen using the standard simplex criterion is "close to zero" (between EPS and CEPS), SIMPLX goes into the alternate pivot selection routine. This routine scans the A matrix in

ascending order of the column number, checking the non-basic variables (including the one chosen by the standard simplex criterion, if necessary) for one which has a pivot element greater than CEPS. If one is found, it is used (without scanning the remaining columns) in lieu of the one found by the standard simplex criterion. If none greater than CEPS is found, the largest one is kept. (This may be the one originally found by standard simplex.) If that one is between EPS and CEPS, it is used, and if L(5) is non-zero, the "small pivot" message (number 4 above) is printed. If the largest is less than EPS, the subroutine returns to the column that contained the standard simplex pivot, and sets that pivot element to zero. That column is then scanned and quotients are again formed to get a new minimum quotient, thus determining a new pivot candidate. If this new pivot candidate is acceptable (greater than EPS), Simplex continues and performs the operation. If it is not acceptable, it too is set to zero and this latter process is repeated until either an acceptable pivot element is found or until each element in that column is non-positive and the problem is declared unbounded.

As for the warning message itself, it appears at the end of a phase and gives the number of times the subroutines left the standard simplex pivot and went to the alternate pivot selection routine, and the number of times it returned to the standard simplex pivot from the alternate routine without having found an acceptable pivot.

6. The Iteration Summary (RS):

E 1 ITERATION 16. PIVOT= 14.7480 OBJECTIVE FUNCTION= .27186017-04 S(4) ENTERED THE BASIS, A(16) LEFT.

This message is printed after every L(6)th iteration in Phase I if L(6) is positive, and after every L(10)th iteration in Phase II if L(10) is positive. It is useful in tracing the progress of the algorithm.

7. The Basis Print (RS).

```
BASIC VARIABLES
 X(  1)=     .824861   X(  2)=     .367047   X(  3)=     .002276   X(  4)=     .011032   X(  5)=     .696950
 X(  6)=     .731616   X(  7)=     .607440   X(  8)=     .684551   X(  9)=     .605376   X( 10)=     .129697

 S(  1)=    5.951955   S(  2)=     .234925   S(  3)=   13.179356   S(  4)=     .000000

 A(  5)=     .000000   A(  7)=     .000000   A(  9)=     .000000   A( 10)=     .000000   A( 15)=     .000000
 A( 16)=     .000000
```

This print occurs after every L(7)th iteration in Phase I if L(7) is positive, after every L(11)th iteration in Phase II if L(11) is positive, at the end of Phase I if L(9) is non-zero, and at the end of Phase II if L(13) is non-zero. It is also printed if the problem is determined to be infeasible or unbounded and L(14) is non-zero.

This print simply gives the indices and values of the basic variables. The letter X refers to real variables with indices between 1 and n. The letter S refers to slack or surplus variables with indices between 1 and ℓ + g. The letter A refers to artifical variables, with indices between 1 and g + e.

8. The End-of-Phase Summary (RS).

```
*****************************************************************************************************
*                                                                                                   *
*          END OF PHASE 1.    OBJECTIVE FUNCTION =      .27186017-04    THERE WERE  17 ITERATIONS.   *
*          MINIMUM PIVOT WAS   .20277    AT ITERATION  12.   REAL OBJECTIVE FUNCTION =  -24.319439  *
*                                                                                                   *
*****************************************************************************************************
```

This message is printed at the end of Phase I if L(8) is non-zero, and at the end of Phase II if L(12) is non-zero. The phrase "REAL OBJECTIVE FUNCTION =" is printed only at the end of Phase I, since at that point the objective function is the artificial objective function.

9. The Infeasible Message (RS).

```
*****************************************************************************
*                                                                           *
*  ERROR - THE PROBLEM IS INFEASIBLE. THE CONSTRAINTS ASSOCIATED WITH THE ARTIFICIAL VARIABLES BELOW ARE INCONSISTENT. *
*  IF NONE APPEAR, NUMERICAL DIFFICULTIES HAVE BEEN ENCOUNTERED. THE LARGEST ENTRY IN THE OBJECTIVE FUNCTION ROW *
*  IS   -5.3214                                                             *
*                                                                           *
*  THE ERROR WAS DETECTED AT ITERATION  10 OF PHASE 1. AT THAT TIME THE OBJECTIVE FUNCTION VALUE WAS  -21.000000 *
*  AND THE FOLLOWING VARIABLES WERE BASIC.                                  *
*                                                                           *
*****************************************************************************
```

This is printed when the problem is infeasible and L(14) is non-zero. A problem is said to be infeasible when it is impossible to satisfy one or more constraints. It is detected in the subroutine when all the indicator variables (the $z_j - c_j$) are less than EPS and either one or both of the following is true: The value of the artificial objective function is greater than (g + e)*CEPS, or one or more artificial variables which cannot be made non-basic have value greater than CEPS.

If the maximum value in the objective function row (the indicators) is "close to " EPS and/or the objective function is "close to" CEPS and/or the artificial variables in the basis are "close to" CEPS, then the infeasibility may result from numerical instability. In this case, it may be possible to "achieve" feasibility by changing the value of EPS (see section 5.9). It should be noted, however, that a different value of EPS could cause the algorithm to follow a different path toward the solution. Thus it is not obvious what the direction and magnitude of the change should be.

If this message is printed, it is always followed by a print of the

variables which are basic at the time the error was detected.

Whether or not this message is printed, L(3) is set to 2 and the subroutine returns to the calling routine.

10. The "Re-inversion Infeasibility" Message (R) .

```
*************************************************************************************************
*                                                                                               *
*  ERROR - THE PROBLEM IS INFEASIBLE.  INFEASIBILITY INDICATED DURING RE-INVERSION OF THE BASIS MATRIX.  *
*                                                                                               *
*  THE ERROR WAS DETECTED AT ITERATION  10 OF PHASE 1. AT THAT TIME THE OBJECTIVE FUNCTION VALUE WAS  -21.000000  *
*  AND THE FOLLOWING VARIABLES WERE BASIC.                                                      *
*                                                                                               *
*************************************************************************************************
```

This message is printed, when L(14) is non-zero, if one or more basic variables became negative as a result of re-inverting the basis matrix. This error, in every case, is a result of numerical difficulties and is an indication that the problem, as it is posed, is just not numerically tractable. To overcome this problem, on might try changing the value of EPS (see section 5.9) or even INVC (see section 3); however, it is suggested that the user modify his problem to make it more numerically stable. Techniques such as normalization, scaling values, removing redundancies, etc. can be used to accomplish this.

11. The "Unbounded" Message (RS).

```
*************************************************************************************************
*                                                                                               *
*  ERROR - THE PROBLEM IS UNBOUNDED. THE VARIABLE X( 13) CAN ASSUME AN ARBITRARILY LARGE VALUE, THEREBY YIELDING  *
*  AN ARBITRARILY LARGE VALUE OF THE OBJECTIVE FUNCTION.                                        *
*                                                                                               *
*  THE ERROR WAS DETECTED AT ITERATION   4 OF PHASE 2. AT THAT TIME THE OBJECTIVE FUNCTION VALUE WAS   43138.569  *
*  AND THE FOLLOWING VARIABLES WERE BASIC.                                                      *
*                                                                                               *
*************************************************************************************************
```

This error message is printed, when L(14) is non-zero, if the problem is found to be unbounded. A problem is unbounded when at least one variable, together with the objective function, can assume an arbitrarily large value without violating any of the constraints. Unboundedness is detected in the subroutines when all the entries in the column associated with the variable that is about to enter the basis are

less than EPS. It is this variable that is identified as "causing" the unboundedness.

In some problems (those that are numerically unstable), this situation might be alleviated by changing the value of EPS (see section 5.9). However, as was mentioned in 9 above, the direction and magnitude of the required change are not obvious.

If this message is printed, it is always followed by a print of the variables that were in the basis at the time the error was detected.

Whether or not this message is printed, L(3) is set to 3 and the subroutine returns to the calling routine.

12. The "Computational Inconsistency" Message (RS).

```
**********************************************************************************************************
*                                                                                                        *
*  WARNING - COMPUTATIONAL INCONSISTENCY INDICATED AT THE END OF PHASE 1. THE ALGORITHM WILL CONTINUE WITH PHASE 2,  *
*  BUT THE USER IS ADVISED TO CRITICIZE THE RESULTS.                                                     *
*                                                                                                        *
**********************************************************************************************************
```

If L(14) is non-zero, this message is printed at the end of Phase I if one or more artificial variables which cannot be removed from the basis is between EPS and CEPS, or if the artificial objective function is between EPS and CEPS. This situation is an indication of possible numerical difficulties. The user is advised at least to check the solution for feasibility.

7. ADDITIONAL WORK AND FUTURE PLANS

The subroutines described in this report are part of a longer-term effort, in which the goal of the Applied Mathematics Division's Operations Research Section is to provide reliable user-oriented solution algorithms for those mathematical optimization problems which are of particular importance for operations-research applications (they often arise in other contexts as well). In addition to SIMPLX and RVSMPX, we also have operational on the NBS computer:

(a) a 0-1 integer programming routine acquired from the RAND Corporation;

(b) a subroutine to solve transportation problems, obtained from the Carnegie-Mellon University;

(c) a prototype code implementing the Dantzig-Wolf decomposition principle for linear programming;

(d) a quadratic programming algorithm, limited to "separable" objective functions (no cross-products of variables), that solves larger problems than SIMPLX or RVSMPX,

(e) a univariate dynamic programming algorithm, developed at Johns Hopkins University and supported by the National Bureau of Standards under Contract CST-1279,

(f) a multivariate dynamic programming algorithm, obtained from Johns Hopkins University , and

(g) several algorithms for finding shortest paths between pairs of nodes in networks.

Except perhaps for (e) and (g), these items have not been tested or documented to the standards set for SIMPLX and RVSMPX. Future work, in addition to such "completion" efforts for selected items from the above list, may include:

(A) testing of the new UNIVAC mathematical programming package (Functional Mathematical Programming System),

(B) as appropriate, extension of preceding items (quite likely beginning with RVSMPX) to handle much larger problems through use of peripheral storage,

(C) provision of capability for parametric and sensitivity analyses,

(D) extension of (d), above, to more general separable nonlinear problems, and

(E) provision of a mixed-integer programming capability.

8. REFERENCES

1. Clasen, R. "Using Linear Programming as a Simplex Subroutine," Rand Report P -3267, Nov. 1965.

2. Finkbeiner, D.T., Introduction to Matrices and Linear Transformations, San Francisco, W.H. Freeman, 1960.

3. Garvin, W.W., Introduction to Linear Programming, New York, McGraw-Hill, 1960.

4. Hadley, G., Linear Programming, Reading, Mass., Addison-Wesley, 1962.

5. Hadley, G., Nonlinear and Dynamic Programming, Reading, Mass., Addison-Wesley, 1964.

6. Kuhn, H.W. and A.W. Tucker, "Nonlinear Programming", in Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, J. Neyman, ed., Berkeley, Cal., University of Cal. Press, 1951, pp. 481-492.

7. Wagner, H.M., "A Comparison of the Original and Revised Simplex Methods," Operations Research, 5(3), 1957.

8. Wagner, H.M., Principles of Operations Research, Englewood Cliffs, N.J., Prentice Hall, 1969.

9. Wolfe,P., "The Simplex Method for Quadratic Programming," Econometrica, 27, 1959, pp. 382-298.

APPENDIX A: THE "ALMOST LINEAR" KUHN-TUCKER CONDITIONS FOR QUADRATIC PROGRAMMING

One form (see section 6-3 of [4]) of the Kuhn-Tucker necessary conditions for the quadratic programming problem given in section 4 to have a solution at x_j (j = 1 through n) is:

1. $c_j + 2\sum_{k=1}^{n} d_{jk}x_k - \sum_{i=1}^{m} \lambda_i a_{ij} = 0$, if $x_j > 0$, for j = 1 through n

   $c_j + 2\sum_{k=1}^{n} d_{jk}x_k - \sum_{i=1}^{m} \lambda_i a_{ij} \leq 0$, if $x_j = 0$, for j = 1 through n

2. $\sum_{j=1}^{n} a_{ij}x_j = b_i$, if $\lambda_i > 0$, i = 1 through ℓ

   $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$, if $\lambda_i = 0$, i = 1 through ℓ

   $\sum_{j=1}^{n} a_{ij}x_j = b_i$, if $\lambda_i < 0$, i = $\ell + 1$ through $\ell + g$

   $\sum_{j=1}^{n} a_{ij}x_j \geq b_i$, if $\lambda_i = 0$, i = $\ell + 1$ through $\ell + g$

   $\sum_{j=1}^{n} a_{ij}x_j = b_i$, , i = $\ell + g + 1$ through m

3. $\lambda_i \geq 0$, i = 1 through ℓ

   $\lambda_i \leq 0$, i = $\ell + 1$ through $\ell + g$

   λ_i unrestricted, i = $\ell + g + 1$ through m

   $x_j \geq 0$, j = 1 through n

As was noted earlier, these conditions are also sufficient when the objective function is concave. Therefore, the problem reduces to one of finding x_j (j = 1 through n) that satisfy conditions 1-3 above for some set of λ_i (i = 1 through m).

These condtions may be written as:

1'. $c_j + 2\sum_{k=1}^{n} d_{ik} x_k - \sum_{i=1}^{m} \lambda_i a_{ij} + y_i = 0$, $j = 1$ through n

$x_j y_j = 0$, $j = 1$ through n

2'. $\sum_{j=1}^{n} a_{ij} x_j + s_i = b_i$, $i = 1$ through ℓ

$\sum_{j=1}^{n} a_{ij} x_j - s_i = b_i$, $i = \ell + 1$ through $\ell + g$

$\sum_{j=1}^{n} a_{ij} x_j = b_i$, $i = \ell + g + 1$ through m

$\lambda_i s_i = 0$, $i = 1$ through $\ell + g$

3'. $\lambda_i \geq 0$, $i = 1$ through ℓ

$\lambda_i \leq 0$, $i = \ell + 1$ through $\ell + g$

λ_i unrestricted, $i = \ell + g + 1$ through m

$x_j \geq 0$, $j = 1$ through n

$y_j \geq 0$, $j = 1$ through n

$s_i \geq 0$, $i = 1$ through $\ell + g$

With the exception of the requirements $x_j y_j = 0$ and $\lambda_i s_i = 0$, these conditions form a set of simultaneous linear equations. Futhermore, the variables either are, or can be made to be, non-negative. In order to make all the variables non-negative, substitute $\lambda_i = y_i$, $i = 1$ through ℓ; $\lambda_i = -u_i$, $i = \ell + 1$ through $\ell + g$; $\lambda_i = u_i - v_i$, $i = \ell + g + 1$ through m, into (1')-(3') to get:

1". $c_j + 2\sum_{k=1}^{n} d_{jk} x_k - \sum_{i=1}^{\ell} u_i a_{ij} + \sum_{i=\ell+1}^{\ell+g} u_i a_{ij}$

$$- \sum_{i=\ell+g+1}^{m} u_i a_{ij} + \sum_{i=\ell+g+1}^{m} v_i a_{ij} + y_j = 0, \quad j = 1 \text{ through } n$$

2". $\sum_{j=1}^{n} a_{ij} x_j + s_i = b_i, \quad i = 1 \text{ through } \ell$

$$\sum_{j=1}^{n} a_{ij} x_j - s_i = b_i, \quad i = \ell + 1 \text{ through } \ell + g$$

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = \ell + g + 1 \text{ through } m$$

3". $u_i \geq 0$, $i = 1$ through m $\qquad$ $v_i \geq 0$, $i = \ell + g + 1$ through m

$x_j \geq 0$, $y_j \geq 0$, $j = 1$ through n $\qquad$ $x_j y_j = 0$, $j = 1$ through n

$s_i \geq 0$, $i = 1$ through $\ell + g$ $\qquad$ $u_i s_i = 0$, $i = 1$ through $\ell + g$

These conditions are a set of simultaneous "almost linear" equations in non-negative variables. The solution to these equations can be found with Phase I of the simplex method with a modification to account for the nonlinear equations above. That modification is that x_j and y_j are not allowed to be simultaneously in the basis; nor are s_i and u_i.

This modification has been made to both SIMPLX and RVSMPX and is available with the use of the parameter KQP (see section 5.11). Note that in the discussion of this option in section 4, the s_i do not appear. This is so because RVSMPX and SIMPLX store slack and surplus variables implicitly. Therefore it is necessary only to put the original constraint coefficients in this part of the matrix.

APPENDIX B:

LISTING OF RVSMPX

UNIVAC 1108 FORTRAN V LEVEL 2206 0018 F5018P
THIS COMPILATION WAS DONE ON 17 AUG 71 AT 19:51:16

SUBROUTINE RVSMPX ENTRY POINT 004300

STORAGE USED (BLOCK, NAME, LENGTH)

0001	*CODE	004372
0000	*DATA	000770
0002	*BLANK	000000

EXTERNAL REFERENCES (BLOCK, NAME)

0003	NWSU$
0004	NIO2$
0005	NEXP1$
0006	NIO1$
0007	NERR2$
0010	NERR3$

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

0001	000245 10L	0001	002560 1002G	0001	002635 1023G	0001	002725 1050G	0001	000701 1050L		
0001	002744 1062G	0001	002766 1073G	0001	000070 110G	0001	000720 1100L	0001	003003 1103G		
0001	000721 1105L	0001	000746 1110L	0001	001021 1120L	0001	003061 1124G	0001	001110 1130L		
0001	001166 1140L	0001	001223 1142L	0001	003126 1143G	0001	001244 1143L	0001	001313 1150L		
0001	001336 1153L	0001	001351 1160L	0001	003235 1174G	0001	001366 1200L	0001	001416 1203L		
0001	001521 1205L	0001	001546 1215L	0001	003300 1217G	0001	001575 1220L	0001	001625 1225L		
0001	003330 1230G	0001	001646 1230L	0001	001766 1240L	0001	003344 1242G	0001	002000 1250L		
0001	002004 1260L	0001	002063 1270L	0001	002122 1280L	0001	002132 1300L	0001	002213 1302L		
0001	002311 1305L	0001	002323 1310L	0001	003531 1311G	0001	003573 1332G	0001	003634 1350G		
0001	003733 1374G	0001	002346 1400L	0001	004005 1410G	0001	004015 1416G	0001	002456 1420L		
0001	004115 1444G	0001	004132 1453G	0001	004146 1464G	0001	000215 150G	0001	004212 1500G		
0001	004220 1506G	0001	002513 1520L	0001	000216 153G	0001	002565 1530L	0001	000267 16L		
0001	002606 1600L	0001	002652 1601L	0001	002677 1602L	0001	002753 1605L	0001	002775 1612L		
0001	003031 1616L	0001	003051 1621L	0001	003073 1630L	0001	003076 1635L	0001	003200 1645L		
0001	000256 167G	0001	003212 1700L	0001	003304 1730L	0001	003312 1735L	0001	003425 1740L		
0001	003457 1750L	0001	003465 1760L	0001	003474 1770L	0001	003505 1780L	0001	003517 1790L		
0001	003553 1800L	0001	003604 1810L	0001	003627 1900L	0001	003666 1905L	0001	003673 1910L		
0001	003677 1920L	0001	003702 1930L	0001	003711 1940L	0001	003745 1950L	0001	003751 1955L		
0001	004063 1960L	0001	004065 1970L	0001	004156 1980L	0001	004240 1990L	0001	004246 1995L		
0001	004250 1997L	0001	000306 204G	0001	000307 207G	0001	000404 211L	0001	000430 221L		
0001	000357 230G	0001	000447 231L	0001	000374 240G	0001	000526 241L	0001	000411 250G		
0001	000506 25L	0001	000435 261G	0001	000473 267G	0001	000475 272G	0001	000605 29L		
0001	000637 30L	0001	000523 306G	0001	000547 314G	0001	000642 32L	0001	001013 375G		
0001	000114 4L	0001	001216 434G	0001	001331 464G	0001	001455 517G	0001	001617 546G		
0001	001712 567G	0001	002040 630G	0001	002056 642G	0001	002270 711G	0001	002377 733G		
0001	002406 741G	0001	002451 750G	0000	000105 9000F	0000	000130 9001F	0000	000144 9002F		
0000	000175 9003F	0000	000204 9005F	0000	000224 9006F	0000	000231 9007F	0000	000263 9008F		
0000	000313 9010F	0000	000343 9011F	0000	000347 9012F	0000	000353 9014F	0000	000440 9015F		
0000	000507 9016F	0000	000510 9017F	0000	000511 9018F	0000	000557 9019F	0000	000631 9020F		
0000	000675 913F	0000 R	000066 APS	0000 R	000033 ASTAR	0000 R	000042 CEPS	0000 R	000036 EPS		

```
0000 I 000022 I          0000 I 000000 IB         0000 I 000055 ICPSX      0000 I 000053 IMP        0000 I 000021 INVC
0000 I 000005 IP         0000 I 000047 IPHASE     0000 I 000070 IQ         0000 I 000071 IQ1        0000 I 000063 IR
0000 I 000035 ITC        0000 I 000051 ITER       0000 I 000034 ITP        0000 I 000067 IX         0000 I 000045 I1
0000 I 000065 I2         0000 I 000072 I3         0000 I 000073 I4         0000 I 000074 I5         0000 I 000041 J
0000 I 000056 J5         0000 I 000060 J1         0000 I 000061 J2         0000 I 000062 J3         0000 I 000057 J5
0000 I 000064 K          0000 I 000043 L1OFF      0000 I 000044 L2OFF      0000 I 000017 M          0000 I 000031 ME
0000 I 000026 MG         0000 I 000024 ML         0000 I 000050 MO3J       0000 I 000027 MPN        0000 I 000030 MP1
0000 I 000025 MSMX       0000 I 000020 M2         0000 I 000023 N          0000 I 000046 NA         0000 I 000032 NMAX
0000 R 000054 O2J        0000 R 000052 PIVM       0000 R 000012 PR         0000 R 000037 XX         0000 R 000040 Y
```

```
00101     1*          SUBROUTINE RVSMPX (A,MA,B,MB,MT,NT,L,X,TOLP,INV,KOP)                R10000
00103     2*          DIMENSION A(MA,1),B(MB,1),L(1),X(1),IB(5),IP(5),PR(5)               R10100
00104     3*          M=MT-1                                                              R10200
00105     4*          M2=M+2                                                              R10300
00106     5*          INVC=M/2+5                                                          R10400
00106     6*     C                                                                        R10500
00106     7*     C * * CHECK FOR NEGATIVE RIGHT HAND SIDES.                               R10600
00106     8*     C                                                                        R10700
00107     9*          DO 4 I=1,M                                                          R10800
00112    10*          IF (A(I,NT).GE.0) GO TO 4                                           R10900
00114    11*          WRITE(6,9020)                                                       R11000
00116    12*          L(3)=2                                                              R11100
00117    13*          RETURN                                                              R11200
00120    14*        4 CONTINUE                                                            R11300
00122    15*          N=NT-1                                                              R11400
00123    16*          ML=L(1)                                                             R11500
00124    17*          MSMX=14                                                             R11600
00125    18*          MG=L(2)                                                             R11700
00126    19*          MPN=M+N                                                             R11800
00127    20*          MP1=M+1                                                             R11900
00130    21*          ME=M-ML-MG                                                          R12000
00131    22*          NMAX=N+ML+ME+2*MG                                                   R12100
00132    23*          ASTAR='*'                                                           R12200
00133    24*          ITP=N-KOP                                                           R12300
00134    25*          ITC=KOP+ML+MG                                                       R12400
00135    26*          L(2)=0                                                              R12500
00136    27*          IF (INV.GT.0) INV=1                                                 R12600
00136    28*     C                                                                        R12700
00136    29*     C * * COMPUTE EPSILON AND CAPITAL EPSILON.                               R12800
00136    30*     C                                                                        R12900
00140    31*          EPS=TOLP                                                            R13000
00141    32*          IF (EPS.GT.0.0)  GO TO 10                                           R13100
00143    33*          XX=1.0E-5                                                           R13200
00144    34*          IF (EPS.LT.0.0)  XX=-EPS                                            R13300
00146    35*          Y=0.0                                                               R13400
00147    36*          DO 5  I=1,M                                                         R13500
00152    37*          DO 5  J=1,N                                                         R13600
00155    38*          EPS=A(I,J)                                                          R13700
00156    39*          IF (EPS.LT.0.0)  EPS=-EPS                                           R13800
00160    40*        5 Y=Y+EPS                                                             R13900
00163    41*          EPS=XX*Y/(M*N)                                                      R14000
00164    42*       10 IF (L(3).EQ.0) GO TO 16                                             R14100
00166    43*       14 DO 15  I=3,MSMX                                                     R14200
00171    44*       15 L(I)=0                                                              R14300
00173    45*          L(4)=1                                                              R14400
```

```
00174    46*          L(5)=1                                                    R14500
00175    47*          L(6)=1                                                    R14600
00176    48*          L(12)=1                                                   R14700
00177    49*          L(13)=1                                                   R14800
00200    50*          L(MSMX)=1                                                 R14900
00201    51*       10 CEPS=10.0*EPS                                             R15000
00202    52*          L(1)=0                                                    R15100
00203    53*          DO 17 I=1,M                                               R15200
00206    54*          DO 17 J=1,N                                               R15300
00211    55*          XX=ABS(A(I,J))                                            R15400
00212    56*       17 IF (XX.LT.EPS.AND.XX.GT.0.0) L(1)=L(1)+1                  R15500
00216    57*          IF (L(4).NE.0)  WRITE (6,9000) EPS,CEPS,L(1)              R15600
00216    58*    C                                                               R15700
00216    59*    C * * SET UP L1(.)-VARIABLE TYPE INDICATOR.                     R15800
00216    60*    C                                                               R15900
00224    61*          L1OFF=MSMX                                                R16000
00225    62*          L2OFF=L1OFF+NMAX                                          R16100
00226    63*          I=1                                                       R16200
00227    64*          DO 20  J=1,N                                              R16300
00232    65*          L(L1OFF+I)=0                                              R16400
00233    66*       20 I=I+1                                                     R16500
00235    67*          IF (ML.EQ.0)  GO TO 211                                   R16600
00237    68*          DO 21  J=1,ML                                             R16700
00242    69*          L(L1OFF+I)=1                                              R16800
00243    70*       21 I=I+1                                                     R16900
00245    71*      211 IF (MG.EQ.0)  GO TO 221                                   R17000
00247    72*          DO 22 J=1,MG                                              R17100
00252    73*          L(L1OFF+I)=2                                              R17200
00253    74*          L(L1OFF+I+MG)=3                                           R17300
00254    75*       22 I=I+1                                                     R17400
00256    76*      221 IF (ME.EQ.0)  GO TO 231                                   R17500
00260    77*          DO 23  J=1,ME                                             R17600
00263    78*          L(L1OFF+I+MG)=3                                           R17700
00264    79*       23 I=I+1                                                     R17800
00264    80*    C                                                               R17900
00264    81*    C * * INITIALIZE B(.,.), THE BASIS INVERSE, TO THE IDENTITY MATRIX.  R18000
00264    82*    C                                                               R18100
00266    83*      231 DO 26  I=1,M2                                             R18200
00271    84*          DO 25  J=1,M2                                             R18300
00274    85*       25 B(I,J)=0.0                                                R18400
00276    86*          IF (I.GT.M)  GO TO 26                                     R18500
00300    87*          B(I,I)=1.0                                                R18600
00301    88*       26 CONTINUE                                                  R18700
00303    89*          IF (INV.GT.0) GO TO 241                                   R18800
00303    90*    C                                                               R18900
00303    91*    C * * INITIALIZE THE X'S, B(M2,.)-THE MULTIPLIERS, B(.,M2)-THE VALUES  R19000
00303    92*    C * * OF THE BASIC VARIABLES, AND B(M2,M2)-THE ARTIFICIAL OBJ. VALUE.  R19100
00303    93*    C                                                               R19200
00305    94*          DO 28 I=1,NMAX                                            R19300
00310    95*       28 X(I)=-EPS/100.0                                           R19400
00312    96*      241 J=1                                                       R19500
00313    97*          DO 32 I=1,NMAX                                            R19600
00316    98*          I1=L(L1OFF+I)                                             R19700
00317    99*          IF (I1.EQ.0.OR.I1.EQ.2)  GO TO 32                         R19800
00321   100*          IF (INV.GT.0) GO TO 29                                    R19900
00323   101*          X(I)=A(J,NT)                                              R20000
00324   102*          L(L2OFF+J)=I                                              R20100
00325   103*       29 B(J,M2)=A(J,NT)                                           R20200
```

```
00326   104*          IF (I1.LT.3)  GO TO 30                                    R20300
00330   105*          B(M2,J)=1.0                                               R20400
00331   106*          B(M2,M2)=B(M2,M2)+B(J,M2)                                 R20500
00332   107*       30 J=J+1                                                     R20600
00333   108*       32 CONTINUE                                                  R20700
00335   109*          IF (INV.EQ.2) GO TO 1900                                  R20800
00335   110*    C                                                               R20900
00335   111*    C * * INITIALIZATIONS FOR PHASE I.                              R21000
00335   112*    C                                                               R21100
00337   113*          NA=0                                                      R21200
00340   114*          IPHASE=1                                                  R21300
00341   115*          MOBJ=M2                                                   R21400
00342   116*          ITER=0                                                    R21500
00343   117*          PIVM=1.0E28                                               R21600
00344   118*          IMP=0                                                     R21700
00345   119*          OBJ=-B(MOBJ,M2)                                           R21800
00346   120*          IF (INV.EQ.1) GO TO 1900                                  R21900
00346   121*    C                                                               R22000
00346   122*    C * * CHECK FOR IMMEDIATE BASIC FEASIBLE SOLUTION.              R22100
00345   123*    C                                                               R22200
00350   124*          IF (ML.LT.1) GO TO 1100                                   R22300
00350   125*    C                                                               R22400
00350   126*    C * * INITIALIZATIONS FOR PHASE II.                             R22500
00350   127*    C                                                               R22600
00352   128*     1050 MOBJ=MP1                                                  R22700
00353   129*          IPHASE=2                                                  R22800
00354   130*          ITER=0                                                    R22900
00355   131*          PIVM=1.0E28                                               R23000
00356   132*          IMP=0                                                     R23100
00357   133*          OBJ=-B(MOBJ,M2)                                           R23200
00357   134*    C                                                               R23300
00357   135*    C ********************************************                R23400
00357   136*    C *** STEPS 1 AND 2-COMPUTE THE C' TO GET S.***                 R23500
00357   137*    C ********************************************                R23600
00357   138*    C                                                               R23700
00360   139*     1100 ICPSX=0                                                   R23800
00361   140*     1105 NA=0                                                      R23900
00362   141*          IF (INV.GT.0) GO TO 1110                                  R24000
00364   142*          JS=0                                                      R24100
00365   143*          IF (L(2).EQ.0) GO TO 1110                                 R24200
00367   144*          IF (L(2)/INVC*INVC.EQ.L(2)) INV=2                         R24300
00371   145*          IF (INV.EQ.2) GO TO 231                                   R24400
00371   146*    C                                                               R24500
00371   147*    C * * CALCULATE THE INDICATORS-C'.                              R24600
00371   148*    C                                                               R24700
00373   149*     1110 B(MOBJ,MP1)=0.0                                           R24800
00374   150*          DO 1160 JS=1,NMAX                                         R24900
00377   151*          IF (INV.EQ.0) GO TO 1120                                  R25000
00401   152*          J=JS                                                      R25100
00402   153*          GO TO 1130                                                R25200
00403   154*     1120 J=JS                                                      R25300
00404   155*          IF (X(J).GE.0.0) GO TO 1160                               R25400
00406   156*          IF (KOP.EQ.0) GO TO 1130                                  R25500
00410   157*          IF (J.LE.ITC.AND.X(ITP+J).GE.0.0) GO TO 1160              R25600
00412   158*          IF (J.GT.ITP.AND.J.LE.ITP+ITC.AND.X(J-ITP).GE.0.0) GO TO 1160  R25700
00414   159*     1130 J1=L(LIOFF+J)                                             R25800
00415   160*          IF (J1.EQ.4)  GO TO 1160                                  R25900
00417   161*          IF (J1.EQ.0)  GO TO 1140                                  R26000
```

```
00421   162*          J2=J-N                                                  R26100
00422   163*          IF (J.GT.N+ML+MG) J2=J2-MG                              R26200
00422   164*    C                                                             R26300
00422   165*    C * * COMPUTE C' FOR SLACK, SURPLUS, OR ARTIFICIAL VARIABLES. R26400
00422   166*    C                                                             R26500
00424   167*          XX=(-1)**(J1+1)*B(MOBJ,J2)                              R26600
00425   168*          IF (J1.EQ.3) XX=XX-1.0                                  R26700
00427   169*          GO TO 1142                                              R26800
00427   170*    C                                                             R26900
00427   171*    C * * COMPUTE C' FOR REAL VARIABLES.                          R27000
00427   172*    C                                                             R27100
00430   173*     1140 XX=0.0                                                  R27200
00431   174*          IF (IPHASE.EQ.2) XX=A(MP1,J)                            R27300
00433   175*          DO 1141 I=1,M                                           R27400
00436   176*     1141 XX=XX+B(MOBJ,I)*A(I,J)                                  R27500
00436   177*    C                                                             R27600
00436   178*    C * * TRANSFER IF INVERTING OR PIVOTING OUT ARTIFICIALS.      R27700
00436   179*    C                                                             R27800
00440   180*     1142 IF (INV.GT.0.OR.ICPSX.EQ.1) GO TO 1143                  R27900
00440   181*    C                                                             R28000
00440   182*    C * * TEST FOR MINIMUM C' IF DOING A NORMAL PIVOT.            R28100
00440   183*    C                                                             R28200
00442   184*          IF (XX.LE.B(MOBJ,MP1)) GO TO 1160                       R28300
00444   185*     1143 B(MOBJ,MP1)=XX                                          R28400
00445   186*          JS=J                                                    R28500
00446   187*          JS=J2                                                   R28600
00447   188*          IF (INV.GT.0) GO TO 1200                                R28700
00451   189*          IF (ICPSX.EQ.0) GO TO 1160                              R28800
00451   190*    C                                                             R28900
00451   191*    C * * IF TRYING TO PIVOT OUT ARTIFICIALS, CHECK THIS COLUMN.  IF OK TO R29000
00451   192*    C * * ENTER, COMPUTE PIVOT ELEMENT.  IF PIVOT IS BIG ENOUGH, PIVOT.  R29100
00451   193*    C * * OTHERWISE, CONTINUE ON TO NEXT COLUMN.                  R29200
00451   194*    C                                                             R29300
00453   195*     1145 IF (XX.LT.-EPS) GO TO 1160                              R29400
00455   196*          JS=J                                                    R29500
00456   197*          IF (J1.EQ.0)  GO TO 1150                                R29600
00460   198*          B(IR,MP1)=(-1)**(J1+1)*B(IR,J2)                         R29700
00461   199*          GO TO 1153                                              R29800
00462   200*     1150 B(IR,MP1)=0.0                                           R29900
00463   201*          DO 1151 K=1,M                                           R30000
00466   202*     1151 B(IR,MP1)=B(IR,MP1)+B(IR,K)*A(K,JS)                     R30100
00470   203*     1153 IF (B(IR,MP1).GT.EPS) GO TO 1200                        R30200
00472   204*          IF (B(IR,MP1).GT.-EPS) GO TO 1160                       R30300
00474   205*          GO TO 1200                                              R30400
00475   206*     1160 CONTINUE                                                R30500
00477   207*          IF (ICPSX.EQ.1) GO TO 1635                              R30600
00501   208*          IF (B(MOBJ,MP1).LT.EPS) GO TO 1600                      R30700
00501   209*    C                                                             R30800
00501   210*    C ***************************************                     R30900
00501   211*    C *** STEP 3-CALCULATE A'(I,S). ***                           R31000
00501   212*    C ***************************************                     R31100
00501   213*    C                                                             R31200
00503   214*     1200 I1=L(L1OFF+JS)                                          R31300
00504   215*          IB(1)=' X(   '                                          R31400
00505   216*          IP(1)=JS                                                R31500
00506   217*          IF (I1.EQ.0) GO TO 1203                                 R31600
00510   218*          IB(1)=' S(   '                                          R31700
00511   219*          IP(1)=JS-N                                              R31800
```

```
00512   220*          IF (I1.LT.3) GO TO 1203                                  R31900
00514   221*          IJ(1)=' A(    '                                          R32000
00515   222*          IP(1)=IP(1)-ML-MG                                        R32100
00516   223*    1203  DO 1225 I=1,MP1                                          R32200
00516   224*    C                                                              R32300
00516   225*    C * * COMPUTE REAL OBJECTIVE COST IF NOT ALREADY COMPUTED.     R32400
00516   226*    C                                                              R32500
00521   227*          IF (I.LT.MP1) GO TO 1205                                 R32600
00523   228*          IF (IPHASE.EQ.2) GO TO 1225                              R32700
00525   229*          B(I,MP1)=0.0                                             R32800
00526   230*          IF (I1.EQ.0) B(I,MP1)=A(MP1,JS)                          R32900
00530   231*          IF (I1.EQ.3) B(I,MP1)=-1.0                               R33000
00532   232*          GO TO 1215                                               R33100
00532   233*    C                                                              R33200
00532   234*    C * * GENERATE THE REST OF THE COLUMN.                         R33300
00532   235*    C                                                              R33400
00533   236*    1205  B(I,MP1)=0.0                                             R33500
00534   237*    1207  I2=L(L2OFF+I)                                            R33600
00535   238*          IF (I2.EQ.-1) GO TO 1215                                 R33700
00537   239*          IF (L(L1OFF+I2).EQ.5) GO TO 1225                         R33800
00541   240*    1215  IF (I1.EQ.0)  GO TO 1220                                 R33900
00543   241*          B(I,MP1)=B(I,MP1)+(-1)**(I1+1)*B(I,J3)                   R34000
00544   242*          GO TO 1225                                               R34100
00545   243*    1220  DO 1223  K=1,M                                           R34200
00550   244*    1223  B(I,MP1)=B(I,MP1)+B(I,K)*A(K,JS)                         R34300
00552   245*    1225  CONTINUE                                                 R34400
00552   246*    C                                                              R34500
00552   247*    C * * IF PIVOTING OUT AN ARTIFICIAL, SET THE VALUE OF THIS ARTIFICIAL  R34600
00552   248*    C * * TO ZERO.  IF IT IS GT EPS SET L(3)=1 (COMPUTATIONAL INCONSISTENCY)  R34700
00552   249*    C                                                              R34800
00554   250*          IF (ICPSX.EQ.0) GO TO 1230                               R34900
00556   251*          IF (B(IR,M2).GT.EPS) L(3)=1                              R35000
00560   252*          B(IR,M2)=0.0                                             R35100
00561   253*          GO TO 1300                                               R35200
00561   254*    C                                                              R35300
00561   255*    C ***************************************                      R35400
00561   256*    C *** STEP 4-FORM QUOTIENTS TO GET R. ***                      R35500
00561   257*    C ***************************************                      R35600
00561   258*    C                                                              R35700
00562   259*    1230  IR=0                                                     R35800
00563   260*          APS=1.0E28                                               R35900
00564   261*          IF (INV.GT.0) APS=0.0                                    R36000
00566   262*          DO 1260 I=1,M                                            R36100
00571   263*          IF (INV.GT.0) GO TO 1240                                 R36200
00573   264*          IF (B(I,MP1).LE.0.0) GO TO 1260                          R36300
00575   265*          XX=B(I,M2)/B(I,MP1)                                      R36400
00576   266*          IF (XX.GT.APS)  GO TO 1260                               R36500
00600   267*          IF (XX.LT.APS.OR.IR.EQ.0) GO TO 1250                     R36600
00602   268*          IF (B(IR,MP1).GE.B(I,MP1)) GO TO 1260                    R36700
00604   269*          GO TO 1250                                               R36800
00604   270*    C                                                              R36900
00604   271*    C * * IF INVERTING, FIND THE LARGEST ABSOLUTE ELEMENT AMONG    R37000
00604   272*    C * * THOSE ROWS NOT ALREADY PIVOTED IN.                       R37100
00604   273*    C                                                              R37200
00605   274*    1240  IF (L(L2OFF+I).NE.-1) GO TO 1260                         R37300
00607   275*          XX=ABS(B(I,MP1))                                         R37400
00610   276*          IF (XX.LE.APS) GO TO 1260                                R37500
00612   277*    1250  IR=I                                                     R37600
```

```
00613   278*          APS=XX                                                  R37700
00614   279*    1260 CONTINUE                                                 R37800
00616   280*          IF (IR.GT.0) GO TO 1270                                 R37900
00616   281*    C                                                             R38000
00616   282*    C * * IF INVERTING, AND THERE IS NO PLACE TO PUT THIS         R38100
00616   283*    C * * VARIABLE, SKIP IT AND TRY AGAIN LATER.                  R38200
00616   284*    C                                                             R38300
00620   285*          IF (INV.GT.0) GO TO 1940                                R38400
00620   286*    C                                                             R38500
00620   287*    C * * THE PROBLEM IS UNBOUNDED.                               R38600
00620   288*    C                                                             R38700
00622   289*          L(3)=3                                                  R38800
00623   290*          IX=1                                                    R38900
00624   291*          IF (L(MSMX).EQ.0) RETURN                                R39000
00626   292*          WRITE(6,9015) (ASTAR,I=1,120),JS                        R39100
00635   293*          WRITE(6,9018) ITER,IPHASE,OBJ,(ASTAR,I=1,120)           R39200
00646   294*          GO TO 1735                                              R39300
00647   295*    1270 XX=ABS(B(IR,MP1))                                        R39400
00650   296*          IF (XX.GT.CEPS) GO TO 1300                              R39500
00650   297*    C                                                             R39600
00650   298*    C * * IF PIVOT IS TOO SMALL, SET TO ZERO AND TRY AGAIN.       R39700
00650   299*    C                                                             R39800
00652   300*          IF (XX.LT.EPS) GO TO 1280                               R39900
00652   301*    C                                                             R40000
00652   302*    C * * IF EPS.LT.PIVOT.LT.CEPS, PRINT WARNING AND CONTINUE.    R40100
00652   303*    C                                                             R40200
00654   304*          IF (L(5).NE.0) WRITE(6,9005) ITER,IPHASE,B(IR,MP1)      R40300
00662   305*          GO TO 1300                                              R40400
00663   306*    1280 B(IR,MP1)=0.0                                            R40500
00664   307*          GO TO 1230                                              R40600
00664   308*    C                                                             R40700
00664   309*    C ***************************************                     R40800
00664   310*    C *** STEP 5-CALCULATE NEW B(.,MT) AND X. ***                 R40900
00664   311*    C ***************************************                     R41000
00664   312*    C                                                             R41100
00665   313*    1300 IF (INV.GT.0) GO TO 1302                                 R41200
00667   314*          I1=L(L2OFF+IR)                                          R41300
00670   315*          I2=L(L1OFF+I1)                                          R41400
00671   316*          X(I1)=-EPS/100.0                                        R41500
00672   317*          IF (L(L1OFF+I1).EQ.3)  L(L1OFF+I1)=4                    R41600
00674   318*          IB(2)=' X(    '                                         R41700
00675   319*          IP(2)=I1                                                R41800
00675   320*          IF (I2.EQ.0) GO TO 1302                                 R41900
00700   321*          IB(2)=' S(    '                                         R42000
00701   322*          IP(2)=I1-N                                              R42100
00702   323*          IF (I2.LT.3) GO TO 1302                                 R42200
00704   324*          IB(2)=' A(    '                                         R42300
00705   325*          IP(2)=IP(2)-ML-MG                                       R42400
00705   326*    1302 L(L2OFF+IR)=JS                                           R42500
00707   327*          B(IR,M2)=B(IR,M2)/B(IR,MP1)                             R42600
00710   328*          DO 1310  I=1,M2                                         R42700
00713   329*          IF (I.EQ.IR)  GO TO 1305                                R42800
00715   330*          B(I,M2)=B(I,M2)-B(IR,M2)*B(I,MP1)                       R42900
00715   331*          IF (I.GT.M)  GO TO 1310                                 R43000
00720   332*    1305 IF (INV.GT.0) GO TO 1310                                 R43100
00722   333*          I1=L(L2OFF+I)                                           R43200
00723   334*          X(I1)=B(I,M2)                                           R43300
00724   335*    1310 CONTINUE                                                 R43400
```

```
00726   336*          IF (INV.EQ.2) GO TO 1400                                  R43500
00730   337*          X(NMAX+1)=-B(MP1,M2)                                      R43600
00731   338*          OBJ=-B(MOBJ,M2)                                           R43700
00731   339*    C                                                               R43800
00731   340*    C *****************************************************         R43900
00731   341*    C *** STEPS 6 AND 7-UPDATE THE INVERSE AND THE MULTIPLIERS. ***  R44000
00731   342*    C *****************************************************         R44100
00731   343*    C                                                               R44200
00732   344*     1400 DO 1402 K=1,M                                             R44300
00735   345*          B(IR,K)=B(IR,K)/B(IR,MP1)                                 R44400
00736   346*     1402 CONTINUE                                                  R44500
00740   347*          DO 1420 I=1,M2                                            R44600
00743   348*          IF (I.EQ.IR)  GO TO 1420                                  R44700
00745   349*          IF (I.EQ.M2.AND.IPHASE.EQ.2) GO TO 1420                   R44800
00747   350*          DO 1410 K=1,M                                             R44900
00752   351*          B(I,K)=B(I,K)-B(I,MP1)*B(IR,K)                            R45000
00753   352*     1410 CONTINUE                                                  R45100
00755   353*     1420 CONTINUE                                                  R45200
00757   354*          IF (INV.GT.0) GO TO 1930                                  R45300
00761   355*          ITER=ITER+1                                               R45400
00762   356*          L(2)=L(2)+1                                               R45500
00762   357*    C                                                               R45600
00762   358*    C * * STORE THE MINIMUM PIVOT.                                  R45700
00762   359*    C                                                               R45800
00763   360*          IF (B(IR,MP1).GT.PIVM) GO TO 1520                         R45900
00765   361*          PIVM=B(IR,MP1)                                            R46000
00765   362*          IMP=ITER                                                  R46100
00767   363*     1520 I1=5+(IPHASE-1)*4                                         R46200
00767   364*    C                                                               R46300
00767   365*    C * * CHECK TO PRINT ITERATION SUMMARY.                         R46400
00767   366*    C                                                               R46500
00770   367*          IF (L(I1+1).EQ.0) GO TO 1530                              R46600
00772   368*          IF (ITER/L(I1+1)*L(I1+1).NE.ITER) GO TO 1530              R46700
00774   369*          WRITE(6,9010) IPHASE,ITER,B(IR,MP1),OBJ,(IB(I),IP(I),I=1,2)  R46800
00774   370*    C                                                               R46900
00774   371*    C * * CHECK TO PRINT ITERATION BASIS.                           R47000
00774   372*    C                                                               R47100
01007   373*     1530 IF(L(I1+2).EQ.0) GO TO 1100                               R47200
01011   374*          IF (ITER/L(I1+2)*L(I1+2).NE.ITER) GO TO 1100              R47300
01013   375*          IX=2                                                      R47400
01014   376*          GO TO 1735                                                R47500
01014   377*    C                                                               R47600
01014   378*    C ********************                                          R47700
01014   379*    C *** END OF PHASE. ***                                         R47800
01014   380*    C ********************                                          R47900
01014   381*    C                                                               R48000
01015   382*     1600 IX=3                                                      R48100
01016   383*          IF ( IPHASE.EQ.2)  GO TO 1700                             R48200
01020   384*          IQ=0                                                      R48300
01020   385*    C                                                               R48400
01020   386*    C * * COMPUTE THE NUMBER OF ARTIFICIALS STILL IN THE BASIS.     R48500
01020   387*    C                                                               R48600
01021   388*          NA=0                                                      R48700
01022   389*          DO 1601 J=1,NMAX                                          R48800
01025   390*          IF (L(L1OFF+J).NE.3) GO TO 1601                           R48900
01027   391*          IF (X(J).GT.CEPS) GO TO 1602                              R49000
01031   392*          NA=NA+1                                                   R49100
01032   393*     1601 CONTINUE                                                  R49200
```

```
01034   394*          IF (B(MOBJ,M2).GT.CEPS)  GO TO 1602                              R49300
01034   395*    C                                                                      R49400
01034   396*    C * * IF THE ARTIFICIAL OBJECTIVE FUNCTION IS ACCEPTABLE BUT ARTIFICIALS R49500
01034   397*    C * * ARE STILL IN THE BASIS, GO TRY TO PIVOT THEM OUT.                R49600
01034   398*    C                                                                      R49700
01036   399*          IF (NA.GT.0)  GO TO 1612                                         R49800
01036   400*    C                                                                      R49900
01036   401*    C * * IF NO ARTIFICIALS ARE IN THE BASIS AND THE ARTIFICIAL OBJECTIVE  R50000
01036   402*    C * * FUNCTION IS NOT QUITE ZERO, PRINT WARNING OF ROUND-OFF ERROR.    R50100
01036   403*    C                                                                      R50200
01040   404*          IF (B(MOBJ,M2).GT.EPS) GO TO 1605                                R50300
01042   405*          GO TO 1700                                                       R50400
01042   406*    C                                                                      R50500
01042   407*    C * * THE PROBLEM IS INFEASIBLE.                                       R50600
01042   408*    C                                                                      R50700
01043   409*     1602 L(3)=2                                                           R50800
01044   410*          IF(L(MSMX).EQ.0) RETURN                                          R50900
01046   411*          WRITE(6,9014) (ASTAR,I=1,120),B(MOBJ,MP1)                        R51000
01055   412*          WRITE(6,9018) ITER,IPHASE,OBJ,(ASTAR,I=1,120)                    R51100
01056   413*          IX=1                                                             R51200
01057   414*          GO TO 1735                                                       R51300
01070   415*     1605 IF (L(MSMX).NE.0) WRITE(6,9019) (ASTAR,I=1,240)                  R51400
01077   416*          L(3)=1                                                           R51500
01100   417*          GO TO 1700                                                       R51600
01100   418*    C                                                                      R51700
01100   419*    C * * FIND THE ARTIFICIAL STILL IN THE BASIS.                          R51800
01100   420*    C                                                                      R51900
01101   421*     1612 IQ1=0                                                           R52000
01102   422*          DO 1615 J=1,NMAX                                                 R52100
01105   423*          J1=L(L1OFF+J)                                                    R52200
01106   424*          IF (J1.EQ.3)  IQ1=IQ1+1                                          R52300
01110   425*          IF (IQ1.GT.IQ) GO TO 1621                                        R52400
01112   426*     1615 CONTINUE                                                         R52500
01114   427*     1616 L(3)=4                                                           R52600
01115   428*          IF (L(MSMX).NE.0)  WRITE (6,913)                                 R52700
01120   429*      913 FORMAT ('0SYSTEM ERROR- COMPUTATIONAL IMPOSSIBILITY')            R52800
01121   430*          IX=1                                                             R52900
01122   431*          GO TO 1735                                                       R53000
01122   432*    C                                                                      R53100
01122   433*    C * * DETERMINE WHICH BASIC VARIABLE IT IS.                            R53200
01122   434*    C                                                                      R53300
01123   435*     1621 DO 1625  IR=1,M                                                 R53400
01125   436*          I=L(L2OFF+IR)                                                    R53500
01127   437*          IF (I.EQ.J)  GO TO 1630                                          R53600
01131   438*     1625 CONTINUE                                                         R53700
01133   439*          GO TO 1616                                                       R53800
01133   440*    C                                                                      R53900
01133   441*    C * * SET ARTIFICIALS IN AT ZERO SWITCH, AND GO TRY TO PIVOT IT OUT.   R54000
01133   442*    C                                                                      R54100
01134   443*     1630 ICPSX=1                                                         R54200
01135   444*          GO TO 1105                                                       R54300
01135   445*    C                                                                      R54400
01135   446*    C * * INCREMENT IQ, THE COUNTER FOR CURRENT NUMBER OF ARTIFICIALS THAT R54500
01135   447*    C * * CAN'T BE PIVOTED OUT.                                            R54600
01135   448*    C                                                                      R54700
01136   449*     1635 IQ=IQ+1                                                         R54800
01137   450*          IF (NA.GT.IQ) GO TO 1612                                         R54900
01137   451*    C                                                                      R55000
```

```
01137   452*   C * * COUNT AND CHECK THE ARTIFICIALS STILL IN THE BASIS.                    R55100
01137   453*   C                                                                           R55200
01141   454*         JS=0                                                                  R55300
01142   455*         DO 1645  I=1,M                                                        R55400
01145   456*         I1=L(L2OFF+I)                                                         R55500
01146   457*         IF (L(L1OFF+I1).NE.3) GO TO 1645                                      R55600
01150   458*         JS=JS+1                                                               R55700
01150   459*   C                                                                           R55800
01150   460*   C * * IF IN AND GT EPSILON, INFEASIBLE. OTHERWISE SET TO 0 AND CONTINUE.    R55900
01150   461*   C                                                                           R56000
01151   462*         IF (B(I,M2).GT.CEPS) GO TO 1602                                       R56100
01153   463*         IF (B(I,M2).GT.EPS) L(3)=1                                            R56200
01155   464*         X(I1)=0.0                                                             R56300
01156   465*         L(L1OFF+I1)=5                                                         R56400
01157   466*         B(I,M2)=0.0                                                           R56500
01160   467*    1640 CONTINUE                                                            R56600
01161   468*    1645 CONTINUE                                                            R56700
01163   469*         IF (JS.EQ.0)  GO TO 1616                                              R56800
01165   470*         IF (L(3).EQ.1) GO TO 1605                                             R56900
01165   471*   C                                                                           R57000
01165   472*   C **********************                                                  R57100
01165   473*   C *** SUMMARY PRINT.  ***                                                 R57200
01165   474*   C **********************                                                  R57300
01165   475*   C                                                                           R57400
01167   476*    1700 I1=5+(IPHASE-1)*4                                                   R57500
01170   477*         IF(L(I1+3).EQ.0) GO TO 1730                                           R57600
01172   478*         WRITE(6,9008) (ASTAR,I=1,120),IPHASE,OBJ,ITER                         R57700
01203   479*         IF(ITER.EQ.0) PIVM=0.0                                                R57800
01205   480*         WRITE(6,9001) PIVM,IMP                                                R57900
01211   481*    1710 IF(IPHASE.EQ.1) WRITE(6,9003) X(NMAX+1)                             R58000
01215   482*         WRITE(6,9006) (ASTAR,I=1,120)                                         R58100
01215   483*   C                                                                           R58200
01215   484*   C **********************                                                  R58300
01215   485*   C *** BASIS PRINT.  ***                                                   R58400
01215   486*   C **********************                                                  R58500
01215   487*   C                                                                           R58600
01223   488*    1730 IF (L(I1+4).EQ.0) GO TO  1810                                       R58700
01225   489*    1735 WRITE(6,9012)                                                       R58800
01227   490*         DO 1737 I2=1,5                                                        R58900
01232   491*         IB(I2)='      '                                                       R59000
01233   492*         PR(I2)=0.0                                                            R59100
01234   493*    1737 IP(I2)=0                                                            R59200
01235   494*         I3=0                                                                  R59300
01237   495*         I4=0                                                                  R59400
01240   496*         IQ1=0                                                                 R59500
01241   497*         DO 1800 I2=1,NMAX                                                     R59600
01244   498*         IF (X(I2).LT.0.0.AND.X(I2).GT.-EPS) GO TO 1800                        R59700
01246   499*         IF (I4.EQ.L(L1OFF+I2)) GO TO 1740                                     R59800
01250   500*         IF (I4.EQ.1.AND.L(L1OFF+I2).EQ.2) GO TO 1740                          R59900
01252   501*         IF (I4.EQ.3.AND.L(L1OFF+I2).EQ.5) GO TO 1740                          R60000
01254   502*         IF (L(L1OFF+I2).EQ.4) GO TO 1780                                      R60100
01256   503*         IQ1=1                                                                 R60200
01257   504*         GO TO 1790                                                            R60300
01260   505*    1740 I4=L(L1OFF+I2)                                                      R60400
01261   506*         IF (I3.EQ.5) GO TO 1790                                               R60500
01263   507*         I3=I3+1                                                               R60600
01264   508*         PR(I3)=X(I2)                                                          R60700
01265   509*         I5=I4+1                                                               R60800
```

```
01256   510*          GO TO (1750,1760,1760,1770,1780,1770),I5                  R60900
01267   511*    1758 IB(I3)=' X(    '                                           R61000
01270   512*          IP(I3)=I2                                                 R61100
01271   513*          GO TO 1800                                                R61200
01272   514*    1760 IB(I3)=' S(    '                                           R61300
01273   515*          IP(I3)=I2-N                                               R61400
01274   516*          GO TO 1800                                                R61500
01275   517*    1770 IB(I3)=' A(    '                                           R61600
01276   518*          IP(I3)=I2-N-ML-MG                                         R61700
01277   519*          GO TO 1800                                                R61800
01300   520*    1780 L(3)=4                                                     R61900
01301   521*          IX=1                                                      R62000
01302   522*          WRITE(6,913)                                              R62100
01304   523*          GO TO 1800                                                R62200
01305   524*    1790 IF (I3.EQ.0) GO TO 1740                                    R62300
01307   525*          WRITE(6,9011) (IB(J),IP(J),PR(J),J=1,I3)                  R62400
01317   526*          IF (IQ1.EQ.1) WRITE(6,9017)                               R62500
01322   527*          IQ1=0                                                     R62600
01323   528*          I3=0                                                      R62700
01324   529*          GO TO 1740                                                R62800
01325   530*    1800 CONTINUE                                                   R62900
01327   531*          IF (I3.GT.0) WRITE(6,9011) (IB(J),IP(J),PR(J),J=1,I3)     R63000
01340   532*    1810 IF (IX.EQ.1) RETURN                                        R63100
01342   533*          IF (IX.EQ.2)  GO TO 1100                                  R63200
01344   534*          IF (IPHASE.EQ.2)  RETURN                                  R63300
01346   535*          GO TO 1050                                                R63400
01346   536*    C                                                               R63500
01346   537*    C ********************************                            R63600
01346   538*    C *** TOTAL INVERSION ROUTINE.  ***                             R63700
01346   539*    C ********************************                            R63800
01346   540*    C                                                               R63900
01347   541*    1900 DO 1910 I=1,M                                              R64000
01352   542*          I1=L(L2OFF+I)                                             R64100
01353   543*          IF (L(L1OFF+I1).NE.5) GO TO 1905                          R64200
01355   544*          B(I,M2)=0.0                                               R64300
01356   545*          GO TO 1910                                                R64400
01357   546*    1905 L(L2OFF+I)=-1                                              R64500
01360   547*    1910 CONTINUE                                                   R64600
01362   548*    1920 I4=0                                                       R64700
01363   549*          JS=0                                                      R64800
01364   550*          GO TO 1940                                                R64900
01364   551*    C                                                               R65000
01364   552*    C * * I4=1 INDICATES THAT A PIVOT WAS PERFORMED.                R65100
01364   553*    C                                                               R65200
01365   554*    1930 I4=1                                                       R65300
01366   555*          X(JS)=-EPS/100.0                                          R65400
01367   556*    1940 I5=JS                                                      R65500
01370   557*          IF (I5.EQ.NMAX) GO TO 1955                                R65600
01370   558*    C                                                               R65700
01370   559*    C * * FOR EACH X(I) THAT IS NON-NEGATIVE AND NOT AN ARTIFICIAL IN AT  R65800
01370   560*    C * * A ZERO LEVEL, GO TRY TO PUT IT IN THE BASIS.  DON'T DO ANYTHING R65900
01370   561*    C * * FOR AN ARTIFICIAL IN AT A ZERO LEVEL.                     R66000
01370   562*    C                                                               R66100
01372   563*          I5=I5+1                                                   R66200
01373   564*          DO 1950 JS=I5,NMAX                                        R66300
01375   565*          IF (X(JS).LT.0.0) GO TO 1950                              R66400
01400   566*          IF (L(L1OFF+JS).EQ.5) GO TO 1930                          R66500
01402   567*          GO TO 1100                                                R66600
```

```
01403   566*    1950 CONTINUE                                                          R66700
01403   569*    C                                                                      R66800
01403   570*    C * * IF AT LEAST ONE PIVOT OPERATION WAS DONE, GO THROUGH X AGAIN.    R66900
01403   571*    C                                                                      R67000
01405   572*    1955 IF (I4.EQ.1) GO TO 1920                                           R67100
01407   573*         DO 1957 J=1,NMAX                                                  R67200
01412   574*    1957 X(J)=-EPS/100.0                                                   R67300
01412   575*    C                                                                      R67400
01412   576*    C * * SET X TO PROPER VALUES AND CHECK FOR FEASIBILITY.                R67500
01412   577*    C                                                                      R67600
01414   578*         I5=0                                                              R67700
01415   579*         DO 1970 I=1,M                                                     R67800
01420   580*         IF (B(I,M2).LT.-EPS) I5=1                                         R67900
01422   581*         IF (B(I,M2).LT.0.0.AND.B(I,M2).GE.-EPS) B(I,M2)=0.0               R68000
01424   582*         I1=L(L2OFF+I)                                                     R68100
01425   583*         IF (I1.LT.0) GO TO 1960                                           R68200
01427   584*         X(I1)=B(I,M2)                                                     R68300
01430   585*         GO TO 1970                                                        R68400
01431   586*    1960 I5=1                                                              R68500
01432   587*    1970 CONTINUE                                                          R68600
01434   588*         IF (I5.EQ.0) GO TO 1980                                           R68700
01434   589*    C                                                                      R68800
01434   590*    C * * INFEASIBLE AFTER INVERSION.                                      R68900
01434   591*    C                                                                      R69000
01435   592*         L(3)=2                                                            R69100
01437   593*         IF (L(MSMX).EQ.0) RETURN                                          R69200
01441   594*         IF (INV.EQ.1) WRITE(6,9007) (ASTAR,I=1,120)                       R69300
01450   595*         IF (INV.EQ.2) WRITE(6,9002) (ASTAR,I=1,120)                       R69400
01457   596*         WRITE(6,9018) ITER,IPHASE,OBJ,(ASTAR,I=1,120)                     R69500
01470   597*         IX=1                                                              R69600
01471   598*         GO TO 1735                                                        R69700
01471   599*    C                                                                      R69800
01471   600*    C * * IF INVERTING BECAUSE OF INITIAL BASIS, FIX UP L1(.).             R69900
01471   601*    C                                                                      R70000
01472   602*    1980 IF (INV.EQ.2) GO TO 1997                                          R70100
01474   603*         I5=M+ML+MG+1                                                      R70200
01475   604*         IF (I5.GT.NMAX) GO TO 1995                                        R70300
01477   605*         DO 1985 J=I5,NMAX                                                 R70400
01502   606*    1985 L(L1OFF+J)=4                                                      R70500
01504   607*         I5=0                                                              R70600
01505   608*         DO 1990 J=1,M                                                     R70700
01510   609*         J1=L(L2OFF+J)                                                     R70800
01511   610*         IF (L(L1OFF+J1).NE.4) GO TO 1990                                  R70900
01513   611*         L(L1OFF+J1)=3                                                     R71000
01514   612*         I5=1                                                              R71100
01515   613*    1990 CONTINUE                                                          R71200
01517   614*         IF (I5.EQ.1) GO TO 1997                                           R71300
01521   615*    1995 INV=0                                                             R71400
01522   616*         GO TO 1050                                                        R71500
01523   617*    1997 INV=0                                                             R71600
01524   618*         GO TO 1110                                                        R71700
01524   619*    C                                                                      R71800
01524   620*    C                                                                      R71900
01525   621*    9000 FORMAT(' EPSILON = 'G13.6,'  CAPITAL EPSILON = 'G13.6,'  'I5,' NON R72000
01525   622*        *-ZERO ENTRIES ARE EFFECTIVELY EQUAL TO ZERO.')                    R72100
01526   623*    9001 FORMAT(' *'14X'MINIMUM PIVOT WAS'G12.5,' AT ITERATION'I4,'.'57X'*' R72200
01526   624*        *)                                                                 R72300
01527   625*    9002 FORMAT(//////' '120A1/' *'118X'*'/' *  ERROR - THE PROBLEM IS INFEA R72400
```

```
01527   625*        *SIBLE.  INFEASIBILITY INDICATED DURING RE-INVERSION OF THE BASIS M  R72500
01527   627*        *ATRIX.'16X'*')                                                      R72600
01530   628*     9003 FORMAT(1H+65X'REAL OBJECTIVE FUNCTION ='G15.8)                     R72700
01531   629*     9005 FORMAT('0* * WARNING * *  SMALL PIVOT ELEMENT AT ITERATION'I4,' OF  R72800
01531   630*        * PHASE'I2,'. PIVOT ='G14.7)                                         R72900
01532   631*     9006 FORMAT(' *'118X'*'/' '120A1)                                       R73000
01533   632*     9007 FORMAT(/////' '120A1/' *'118X'*'/' *  ERROR - THE VARIABLES SPECIF  R73100
01533   633*        *IED AS COMPRISING AN INITIAL SOLUTION DO NOT FORM A BASIC FEASIBLE   R73200
01533   634*        * SOLUTION.'12X'*')                                                  R73300
01534   635*     9008 FORMAT(/////' '120A1/' *'118X'*'/' *'14X'END OF PHASE'I2,'.    OBJ  R73400
01534   636*        *ECTIVE FUNCTION ='G18.8,4X'THERE WERE'I4,' ITERATIONS.'17X'*')      R73500
01535   637*     9010 FORMAT (///'0PHASE'',I2,' ITERATION',I4,'.  PIVOT=',G13.6,'   OBJEC  R73600
01535   638*        *TIVE FUNCTION='G15.8,'  ',A3,I3,') ENTERED THE BASIS,',A3,I3,') LE  R73700
01535   639*        *FT.')                                                               R73800
01536   640*     9011 FORMAT(5(3XA3,I3,')='F12.6))                                       R73900
01537   641*     9012 FORMAT('0BASIC VARIABLES'/)                                        R74000
01540   642*     9014 FORMAT(/////' '120A1/' *'118X'*'/' *  ERROR - THE PROBLEM IS INFEA  R74100
01540   643*        *SIBLE. THE CONSTRAINTS ASSOCIATED WITH THE ARTIFICIAL VARIABLES BE   R74200
01540   644*        *LOW ARE INCONSISTENT. *'/' *  IF NONE APPEAR, NUMERICAL DIFFICULTI   R74300
01540   645*        *ES HAVE BEEN ENCOUNTERED. THE LARGEST ENTRY IN THE OBJECTIVE FUNCT   R74400
01540   646*        *ION ROW',7X,'*'/' *  IS ',G12.5,101X,'*')                           R74500
01541   647*     9015 FORMAT(/////' '120A1/' *'118X'*'/' *  ERROR - THE PROBLEM IS UNBOU  R74600
01541   648*        *NDED. THE VARIABLE X('I3,') CAN ASSUME AN ARBITRARILY LARGE VALUE,   R74700
01541   649*        * THEREBY YIELDING'7X'*'/' *  AN ARBITRARILY LARGE VALUE OF THE OBJ   R74800
01541   650*        *ECTIVE FUNCTION.'63X'*')                                            R74900
01542   651*     9016 FORMAT (1H1)                                                       R75000
01543   652*     9017 FORMAT (1H )                                                       R75100
01544   653*     9018 FORMAT(' *'118X'*'/' *  THE ERROR WAS DETECTED AT ITERATION'I4,' O  R75200
01544   654*        *F PHASE'I2,'. AT THAT TIME THE OBJECTIVE FUNCTION VALUE WAS'G15.8,   R75300
01544   655*        *4X'*'/' *  AND THE FOLLOWING VARIABLES WERE BASIC.'77X'*'/' *'118X   R75400
01544   656*        *'*'/' '120A1)                                                       R75500
01545   657*     9019 FORMAT(///' '120A1/' *'118X'*'/' *  WARNING - COMPUTATIONAL INCONS  R75600
01545   658*        *ISTENCY INDICATED AT THE END OF PHASE 1. THE ALGORITHM WILL CONTIN   R75700
01545   659*        *UE WITH PHASE 2,'4X'*'/' *  BUT THE USER IS ADVISED TO CRITICIZE T   R75800
01545   660*        *HE RESULTS.'67X'*'/' *'118X'*'/' '120A1)                            R75900
01546   661*     9020 FORMAT('0* *  AT LEAST ONE ELEMENT OF THE RIGHT HAND SIDE COLUMN I  R76000
01546   662*        *S LESS THAN ZERO. SUBROUTINE TERMINATES.')                          R76100
01547   663*         END                                                                 R76200

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =   1 SEC.
PHASE 2 TIME =   0 SEC.
PHASE 3 TIME =   3 SEC.
PHASE 4 TIME =   0 SEC.
PHASE 5 TIME =   2 SEC.
PHASE 6 TIME =   1 SEC.


TOTAL COMPILATION TIME =   7  SEC
```

APPENDIX C:

LISTING OF SIMPLX

```
$IT FOR SIMPLX,SIMPLX                                    19 AUG 71          08:46:50.064
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F50182
THIS COMPILATION WAS DONE ON 19 AUG 71 AT 08:46:50

   SUBROUTINE SIMPLX    ENTRY POINT 003640

   STORAGE USED (BLOCK, NAME, LENGTH)

      0001   *CODE   003704
      0000   *DATA   000763
      0002   *BLANK  000000


   EXTERNAL REFERENCES (BLOCK, NAME)

      0003   NWDJ$
      0004   NIO2$
      0005   NIO1$
      0006   NERR2$
      0007   NERR3$


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

 0001   000262 1UL      0001   002373 1023G    0001   002423 1031G    0001   002517 1051G    0001   000120 11L
 0001   002653 1113G    0001   000074 112G     0001   003004 1147G    0001   000324 12L      0001   003173 1226G
 0001   003201 1233G    0001   003262 1257G    0001   003276 1270G    0001   003343 1305G    0001   000134 132G
 0001   003400 1324G    0001   003413 1334G    0001   003462 1351G    0001   003515 1370G    0001   003530 1400G
 0001   003577 1415G    0001   000214 161G     0001   000217 164G     0001   001106 2000L    0001   001126 2001L
 0001   001140 2002L    0001   001141 2005L    0001   001177 2006L    0001   001155 2007L    0001   001220 2008L
 0001   001246 2009L    0001   001276 2010L    0001   001321 2012L    0001   001324 2015L    0001   000302 202G
 0001   001347 2025L    0001   000303 205G     0001   000405 232G     0001   000421 240G     0001   000446 254G
 0001   000473 263G     0001   000475 267G     0001   000532 302G     0001   000451 32L      0001   000634 325G
 0001   000677 340G     0001   000751 351G     0001   001025 365G     0001   000145 4L       0001   001053 402G
 0001   001071 414G     0001   001225 467G     0001   000237 5L       0001   000515 50L      0001   001364 5000L
 0001   001454 5005L    0001   001476 5010L    0001   001511 5020L    0001   001521 5025L    0001   001524 5030L
 0001   001527 5100L    0001   001572 5110L    0001   001272 513G     0001   001667 5150L    0001   001713 5155L
 0001   001722 5170L    0001   001724 5175L    0001   001725 5300L    0001   002023 5305L    0001   002045 5310L
 0001   002066 5315L    0001   002140 5320L    0001   002206 5330L    0001   002216 5335L    0001   002232 5344L
 0001   002300 5345L    0001   002322 5350L    0001   002326 5400L    0001   002354 5401L    0001   002453 5410L
 0001   002503 5415L    0001   002603 5425L    0001   002624 5428L    0001   002700 5440L    0001   002730 5445L
 0001   002752 5448L    0001   003007 5460L    0001   000573 55L      0001   003016 5500L    0001   003071 5530L
 0001   003073 5535L    0001   001415 554G     0001   003134 5540L    0001   003135 5550L    0001   003147 5600L
 0001   003233 5608L    0001   003241 5609L    0001   003245 5610L    0001   003313 5620L    0001   003352 5625L
 0001   003432 5630L    0001   003471 5635L    0001   003474 5637L    0001   003547 5640L    0001   003606 5645L
 0001   001501 574G     0001   000242 6L       0001   000643 60L      0001   001552 615G     0001   001634 634G
 0001   000737 66L      0001   001761 663G     0001   002050 705G     0001   002066 714G     0001   001000 75L
 0001   002255 753G     0001   002273 775G     0001   001030 85L      0001   001041 90L      0000   000125 9000F
 0000   000150 9001F    0000   000164 9003F    0000   000173 9005F    0000   000213 9006F    0000   000220 9007F
 0000   000230 9008F    0000   000310 9010F    0000   000340 9011F    0000   000344 9012F    0000   000350 9014F
 0000   000435 9015F    0000   000504 9016F    0000   000505 9017F    0000   000506 9018F    0000   000554 9019F
 0000   000626 9020F    0001   001075 91L      0000 R 000040 AA       0000 R 000021 ASTAR    0000 R 000037 CEPS
 0000 R 000033 EPS      0000 I 000026 I        0000 I 000071 IABC     0000 I 000072 IABC1    0000 I 000000 IB
 0000 I 000077 IFOUND   0000 I 000067 IC1      0000 I 000070 IC2      0000 I 000115 IENT     0000 I 000050 IG
 0000 I 000103 ILOOK    0000 I 000116 ILVE     0000 I 000074 IMP      0000 I 000121 IND      0000 I 000044 INOFF
```

```
0000 I 000005 IP          0000 I 000051 IPHASE      0000 I 000063 IPIV        0000 I 000054 IPRINT      0000 I 000114 IS
0000 I 000100 ISPIV       0000 I 000031 ITC         0000 I 000065 ITER        0000 I 000120 ITER1       0000 I 000030 ITP
0000 I 000020 IWARN       0000 I 000032 IX          0000 I 000104 IY          0000 I 000106 I7          0000 I 000055 I1
0000 I 000111 I1X         0000 I 000060 I2          0000 I 000057 I3          0000 I 000056 I4          0000 I 000036 J
0000 I 000112 JCPIV       0000 I 000064 JPIV        0000 I 000101 JSPIV       0000 I 000102 JTRUE       0000 I 000061 JY
0000 I 000002 J1          0000 I 000017 KFLAG       0000 I 000117 KH          0000 I 000041 L10FF       0000 I 000046 L10
0000 I 000042 L20FF       0000 I 000047 L20         0000 I 000043 L30FF       0000 I 000022 M           0000 I 000023 ME
0000 I 000025 MG          0000 I 000024 ML          0000 I 000053 MOBJ        0000 I 000052 MX          0000 I 000027 N
0000 I 000045 NOTOFF      0000 I 000122 NOX         0000 I 000123 N1          0000 I 000124 N2          0000 R 000066 OBJ
0000 R 000076 PIV         0000 R 000073 PIVM        0000 R 000113 PIVX        0000 R 000012 P3          0000 R 000107 Q
0000 R 000105 Q1          0000 R 000075 ROBJ        0000 R 000110 SPIV        0000 R 000034 XMAX        0000 R 000035 Y
```

```
00101     1*          SUBROUTINE SIMPLX (A,MA,MT,NT,L,X,TOLP,KOP)                    S10000
00103     2*          DIMENSION A(MA,1),L(1),X(1),IB(5),IP(5),PR(5)                  S10100
00104     3*          KFLAG = 0                                                      S10200
00105     4*          IWARN = 0                                                      S10300
00106     5*          ASTAR = '*'                                                    S10400
00107     6*          MENT=2                                                         S10500
00110     7*          ME = M-ML-MG                                                   S10600
00111     8*          DO 11 I=1,M                                                    S10700
00114     9*          IF (A(I,NT) .GE. 0.) GO TO 11                                  S10800
00115    10*          WRITE (6,9020)                                                 S10900
00120    11*          L(3) = 2                                                       S11000
00121    12*          RETURN                                                         S11100
00122    13*       11 CONTINUE                                                       S11200
00124    14*          ML=L(1)                                                        S11300
00125    15*          MG=L(2)                                                        S11400
00126    16*          L(2) = 0                                                       S11500
00127    17*          IF (L(3) .EQ. 0) GO TO 4                                       S11600
00131    18*          DO 1 I=3,13                                                    S11700
00134    19*        1 L(I)=0                                                         S11800
00135    20*          L(4)=1                                                         S11900
00137    21*          L(5)=1                                                         S12000
00140    22*          L(8) = 1                                                       S12100
00141    23*          L(12) = 1                                                      S12200
00142    24*          L(13)=1                                                        S12300
00143    25*          L(14)=1                                                        S12400
00144    26*        4 N=NT-1                                                         S12500
00145    27*          ITP=N-KOP                                                      S12600
00146    28*          ITC=KOP+ML+MG                                                  S12700
00147    29*          IX=1                                                           S12800
00147    30*    C     IF (L(14) .GT. 0) WRITE(6,9016)                                S12900
00150    31*          L(3)=0                                                         S13000
00151    32*          EPS=TOLP                                                       S13100
00152    33*          IF (EPS.GT.0.0)  GO TO 10                                      S13200
00154    34*          XMAX=1.0E-5                                                    S13300
00155    35*          IF (EPS.LT.0)  XMAX=-EPS                                       S13400
00157    36*          Y=0.0                                                          S13500
00160    37*          DO 6 I=1,M                                                     S13600
00163    38*          DO 6 J=1,N                                                     S13700
00166    39*          IF (A(I,J) .LT. 0.) GO TO 5                                    S13800
00170    40*          Y = Y + A(I,J)                                                 S13900
00171    41*          GO TO 6                                                        S14000
00172    42*        5 Y = Y-A(I,J)                                                   S14100
00173    43*        6 CONTINUE                                                       S14200
```

```
00176    44*          EPS=XMAX*Y/(M*N)                                          S14300
00177    45*       10 CEPS = 10.*EPS                                            S14400
00200    46*          L(1) = 0                                                  S14500
00201    47*          DO 12 I=1,M                                               S14600
00204    48*          DO 12 J=1,N                                               S14700
00207    49*          AA = ABS(A(I,J))                                          S14800
00210    50*          IF (AA .LE. 0 .OR. AA .GT. EPS) GO TO 12                  S14900
00212    51*          L(1) = L(1) + 1                                           S15000
00213    52*       12 CONTINUE                                                  S15100
00216    53*          IF (L(4) .NE. 0) WRITE (6,9000) EPS,CEPS,L(1)             S15200
00216    54*    C                                                               S15300
00216    55*    C * * CALCULATE L VECTOR OFFSETS AND INITIALIZE L VECTOR.       S15400
00216    56*    C                                                               S15500
00224    57*          L1OFF=14                                                  S15600
00225    58*          L2OFF=L1OFF+N                                             S15700
00226    59*          L3OFF=L2OFF+M                                             S15800
00227    60*          INOFF=L3OFF+MG                                            S15900
00230    61*          NOTOFF=INOFF+M                                            S16000
00231    62*          DO 20  J=1,N                                              S16100
00234    63*          L(L1OFF+J)=J                                              S16200
00235    64*       20 L(NOTOFF+J)=J                                             S16300
00237    65*          DO 25  I=1,M                                              S16400
00242    66*          L(L2OFF+I)=I                                              S16500
00243    67*       25 L(INOFF+I)=I+N                                            S16600
00245    68*          L1O=N                                                     S16700
00246    69*          L2O=M                                                     S16800
00247    70*          IF (ML.EQ.0)  GO TO 2000                                  S16900
00251    71*          IF (MG.EQ.0)  GO TO 32                                    S17000
00253    72*          DO 30 IG=1,MG                                             S17100
00256    73*       30 L(L3OFF+IG)=1                                             S17200
00260    74*       32 IPHASE=1                                                  S17300
00260    75*    C                                                               S17400
00260    76*    C * * CONSTRUCT ARTIFICIAL OBJECTIVE FUNCTION.                  S17500
00260    77*    C                                                               S17600
00261    78*          MX=ML+1                                                   S17700
00262    79*          DO 35  J=1,NT                                             S17800
00265    80*          A(MT,J)=0.0                                               S17900
00266    81*          DO 35  I=MX,M                                             S18000
00271    82*       35 A(MT,J)=A(MT,J)+A(I,J)                                    S18100
00274    83*          MOBJ=M+2                                                  S18200
00275    84*          IPRINT=6                                                  S18300
00276    85*          GO TO 2001                                                S18400
00275    86*    C                                                               S18500
00275    87*    C * * THE FOLLOWING SECTION IS ENTERED IN PHASE 1 WHEN ALL COEFFICIENTS  S18600
00275    88*    C * * IN THE OBJECTIVE FUNCTION ROW ARE NON-POSITIVE. FEASIBILITY AND    S18700
00275    89*    C * * POSSIBLE COMPUTATIONAL INCONSISTENCIES ARE INVESTIGATED.          S18800
00275    90*    C                                                               S18900
00277    91*       50 IF (A(MT,NT) .GT. CEPS) GO TO 90                          S19000
00301    92*          DO 55  I1=1,L2O                                           S19100
00304    93*          I4=I1                                                     S19200
00305    94*          I3=L(L2OFF+I1)                                            S19300
00306    95*          I=L(INOFF+I3)                                             S19400
00307    96*          IF (I.GT.N+ML+MG)  GO TO 60                               S19500
00311    97*          IF(I.LE.N+ML)  GO TO 55                                   S19600
00313    98*          I2=L(L3OFF+I-N-ML)                                        S19700
00314    99*          IF (I2.EQ.1)  GO TO 60                                    S19800
00315   100*       55 CONTINUE                                                  S19900
00320   101*          IF (A(MT,NT) .GT. EPS .OR. IWARN .EQ. 1) L(3) = 1         S20000
```

```
00322  102*          IF (L(3) .EQ. 1 .AND. L(14) .NE. 0) WRITE (6,9019) (ASTAR,      S20100
00322  103*         * I=1,240)                                                       S20200
00322  104*     C                                                                    S20300
00322  105*     C * * COMPUTATIONAL INCONSISTENCY IMPLIES THE ARTIFICIAL OBJECTIVE    S20400
00322  106*     C * * FUNCTION AND/OR AT LEAST ONE ARTIFICIAL VARIABLE HAVE ABSOLUTE  S20500
00322  107*     C * * VALUE BETWEEN EPS AND CEPS.                                    S20600
00322  108*     C                                                                    S20700
00331  109*          IA=4                                                            S20800
00332  110*          GO TO 2007                                                      S20900
00333  111*       60 IF (A(I3,NT) .GT. CEPS .OR. A(I3,NT) .LT. -CEPS) GO TO 90       S21000
00335  112*          IF (L10 .LE. 0) GO TO 66                                        S21100
00335  113*     C                                                                    S21200
00335  114*     C * * THERE IS AN ARTIFICIAL VARIABLE IN THE BASIS WITH              S21300
00335  115*     C * * A VALUE LESS THAN OR EQUAL TO CAPITAL EPSILON.                 S21400
00335  116*     C * * THERE WILL BE AN ATTEMPT TO FIND AN ACCEPTABLE                 S21500
00335  117*     C * * PIVOT ELEMENT.                                                 S21600
00335  118*     C                                                                    S21700
00337  119*       61 DO 65 J=1,L10                                                   S21800
00342  120*          JX=L(L10FF+J)                                                   S21900
00343  121*          IF (A(I3,JX).GT.EPS.OR.A(I3,JX).LT.-EPS)  GO TO 75              S22000
00345  122*       65 CONTINUE                                                        S22100
00345  123*     C                                                                    S22200
00345  124*     C * * IF THERE EXISTS NO ENTRY GREATER IN MAGNITUDE THAN EPSILON, ROW I3  S22300
00345  125*     C * * REPRESENTS A REDUNDANT CONSTRAINT.                             S22400
00345  126*     C                                                                    S22500
00347  127*       66 L20=L20-1                                                       S22600
00350  128*          DO 70 I1=I4,L20                                                 S22700
00353  129*          L(L20FF+I1)=L(L20FF+I1+1)                                       S22800
00354  130*       70 CONTINUE                                                        S22900
00356  131*          IF (A(I3,NT) .GT. EPS) IWARN = 1                                S23000
00360  132*          A(I3,NT)=0.0                                                    S23100
00361  133*          GO TO 50                                                        S23200
00362  134*       75 IF (A(I3,JX).GT.0.0)  GO TO 85                                  S23300
00364  135*          DO 80  J1=1,N                                                   S23400
00367  136*          A(I3,J1)=-A(I3,J1)                                              S23500
00370  137*       80 CONTINUE                                                        S23600
00372  138*       85 A(I3,NT) = 0.                                                   S23700
00372  139*     C                                                                    S23800
00372  140*     C * * THE ARTIFICIAL CAN AND WILL BE REMOVED FROM THE BASIS.         S23900
00372  141*     C                                                                    S24000
00373  142*          IPIV = I3                                                       S24100
00374  143*          JPIV=JX                                                         S24200
00375  144*          GO TO 5400                                                      S24300
00375  145*     C                                                                    S24400
00375  146*     C * * THE PROBLEM IS INFEASIBLE.                                     S24500
00375  147*     C                                                                    S24600
00376  148*       90 IF (L(14) .EQ. 0) GO TO 91                                      S24700
00400  149*          WRITE (6,9014) (ASTAR,I=1,120),XMAX                             S24800
00407  150*          WRITE (6,9018) ITER,IPHASE,A(MOBJ,NT),(ASTAR,I=1,120)           S24900
00420  151*       91 L(3) = 2                                                        S25000
00421  152*          IX=3                                                            S25100
00422  153*          IF (L(14) .NE. 0) KFLAG = 1                                     S25200
00424  154*          GO TO 5600                                                      S25300
00425  155*     2000 IPHASE=2                                                        S25400
00426  156*          IA=1                                                            S25500
00427  157*          OBJ=-A(M+1,NT)                                                  S25600
00430  158*          MOBJ=M+1                                                        S25700
00431  159*          IPRINT=10                                                       S25800
```

```
00432   160*          L(2) = L(2)+ 1                                                  S25900
00433   161*     2001 ITER=0                                                          S26000
00434   162*          IC1=1                                                           S26100
00435   163*          IC2=1                                                           S26200
00436   164*          IABC=0                                                          S26300
00437   165*          IABC1=0                                                         S26400
00440   166*          PIVM=1.0 E 8                                                    S26500
00441   167*          IMP=10000                                                       S26600
00442   168*     2002 GO TO 5000                                                      S26700
00443   169*     2005 IF (XMAX.GE.EPS)  GO TO 5100                                    S26800
00445   170*          IX=IPHASE+1                                                     S26900
00446   171*          IF (IPHASE .EQ. 1) GO TO 50                                     S27000
00450   172*     2007 IF (IABC.EQ.0.AND.IABC1.EQ.0)  GO TO 2006                       S27100
00452   173*          IF (L(5) .EQ. 0) GO TO 2005                                     S27200
00454   174*          WRITE (6,9007) IPHASE,IABC,IABC1                                S27300
00461   175*     2006 IF (L(IPRINT+2) .NE. 0) GO TO 2008                              S27400
00463   176*          IF (L(14) .EQ. 0 .OR. L(3) .LE. 1) GO TO 2010                   S27500
00465   177*     2008 WRITE (6,9008) (ASTAR,I=1,120),IPHASE,OBJ,ITER                  S27600
00475   178*          IF (ITER .EQ. 0) GO TO 2009                                     S27700
00500   179*          WRITE (6,9001) PIVM,IMP                                         S27800
00504   180*     2009 ROBJ = -A(I+1,NT)                                               S27900
00505   181*          IF (IPHASE .EQ. 1) WRITE (6,9003) ROBJ                          S28000
00511   182*          WRITE (6,9006) (ASTAR,I=1,120)                                  S28100
00517   183*     2010 IF (L(IPRINT+3) .NE. 0) KFLAG = 1                               S28200
00521   184*          IF(IPHASE.EQ.1.AND.KFLAG.EQ.0) GO TO 5550                       S28300
00523   185*          GO TO 5600                                                      S28400
00524   186*     2012 RETURN                                                          S28500
00525   187*     2015 IF (IPIV .EQ. 0) GO TO 5344                                     S28600
00527   188*          IF (PIV.GT.CEPS)  GO TO 5400                                    S28700
00531   189*          IF (IBOUND .EQ. 0) GO TO 5300                                   S28800
00533   190*          IF (PIV .GT. EPS) GO TO 5400                                    S28900
00535   191*          ISPIV = IPIV                                                    S29000
00536   192*          JSPIV = JPIV                                                    S29100
00537   193*          GO TO 5335                                                      S29200
00540   194*     2025 ITER=ITER+1                                                     S29300
00541   195*          L(2) = L(2)+ 1                                                  S29400
00542   196*          IC1=IC1+1                                                       S29500
00543   197*          IC2=IC2+1                                                       S29600
00544   198*          GO TO 2002                                                      S29700
00544   199*     C*************************************************************     S29800
00544   200*     C                                                              *     S29900
00544   201*     C     FIND THE MAXIMUM COEFFICIENT IN THE OBJECTIVE FUNCTION ROW.*     S30000
00544   202*     C                                                              *     S30100
00544   203*     C*************************************************************     S30200
00545   204*      5000 XMAX=0.0                                                       S30300
00546   205*          JPIV=0                                                          S30400
00547   206*          IF (L10.LT.1) GO TO 5030                                        S30500
00551   207*          XMAX=-1.0E35                                                    S30600
00552   208*          IBOUND=0                                                        S30700
00553   209*          DO 5025 J=1,L10                                                 S30800
00556   210*          JX=L(L1OFF+J)                                                   S30900
00557   211*          IF (A(MOBJ,JX).LE.XMAX) GO TO 5025                              S31000
00561   212*          IF (KOP.LE.0) GO TO 5020                                        S31100
00563   213*          JTRUE=L(NOTOFF+JX)                                              S31200
00564   214*          IF (JTRUE.GT.ITC) GO TO 5005                                    S31300
00566   215*          ILOOK=ITP+JTRUE                                                 S31400
00567   216*          GO TO 5010                                                      S31500
00570   217*     5005 IF (JTRUE.LE.ITP.OR.JTRUE.GT.ITP+ITC) GO TO 5020                S31600
```

```
00572   218*          ILOOK=JTRUE-ITP                                          S31700
00573   219*    5010  DO 5015 I=1,M                                           S31800
00575   220*          IF (L(INOFF+I).EQ.ILOOK) GO TO 5025                      S31900
00600   221*    5015  CONTINUE                                                 S32000
00602   222*    5020  JPIV=JX                                                  S32100
00603   223*          XMAX=A(MOBJ,JX)                                          S32200
00604   224*    5025  CONTINUE                                                 S32300
00605   225*    5030  JSPIV=JPIV                                               S32400
00607   226*          GO TO 2005                                               S32500
00607   227*   C*****************************************************************S32600
00607   228*   C                                                               *S32700
00607   229*   C     FIND THE MINIMUM QUOTIENT.                                *S32800
00607   230*   C                                                               *S32900
00607   231*   C*****************************************************************S33000
00610   232*    5100  IPIV=0                                                   S33100
00611   233*          PIV=0.0                                                  S33200
00612   234*          IF (L20.LT.1)  GO TO 5175                                S33300
00614   235*          DO 5105  I=1,L20                                         S33400
00617   236*          IY = L(L20FF+I)                                          S33500
00620   237*          IF (A(IY,JPIV) .GT. 0.0) GO TO 5110                      S33600
00622   238*    5105  CONTINUE                                                 S33700
00624   239*          GO TO 5175                                               S33800
00625   240*    5110  Q1 = A(IY,NT)/A(IY,JPIV)                                S33900
00626   241*          IPIV = IY                                                S34000
00627   242*          PIV=A(IPIV,JPIV)                                         S34100
00630   243*          IZ=I+1                                                   S34200
00631   244*          IF (IZ.GT.L20)  GO TO 5175                               S34300
00633   245*          DO 5170  I=IZ,L20                                        S34400
00636   246*          IY = L(L20FF+I)                                          S34500
00637   247*          IF (A(IY,JPIV) .LE. 0.0) GO TO 5170                      S34600
00641   248*          Q = A(IY,NT)/A(IY,JPIV)                                  S34700
00642   249*          IF (Q.GE.Q1)  GO TO 5150                                 S34800
00644   250*          Q1=Q                                                     S34900
00645   251*          GO TO 5155                                               S35000
00646   252*    5150  IF (Q.GT.Q1)  GO TO 5170                                 S35100
00650   253*          IF (A(IY,JPIV) .LE. A(IPIV,JPIV)) GO TO 5170             S35200
00652   254*    5155  IPIV = IY                                                S35300
00653   255*          PIV=A(IPIV,JPIV)                                         S35400
00654   256*    5170  CONTINUE                                                 S35500
00656   257*    5175  GO TO 2015                                              S35600
00656   258*   C*****************************************************************S35700
00656   259*   C                                                               *S35800
00656   260*   C     FIND AN ALTERNATE PIVOT ELEMENT GREATER THAN CEPS.        *S35900
00656   261*   C                                                               *S36000
00656   262*   C*****************************************************************S36100
00657   263*    5300  SPIV=PIV                                                 S36200
00660   264*          ISPIV=IPIV                                               S36300
00661   265*          JSPIV=JPIV                                               S36400
00662   266*          DO 5330 J=1,L10                                          S36500
00665   267*          J1=L(L10FF+J)                                            S36600
00666   268*          I1X=0                                                    S36700
00667   269*          Q1=1.0E20                                                S36800
00670   270*          IF (A(MOBJ,J1).LE.EPS)  GO TO 5330                       S36900
00672   271*          IF (KOP.LE.0) GO TO 5315                                 S37000
00674   272*          JTRUE=L(NOTOFF+J1)                                       S37100
00675   273*          IF (JTRUE.GT.ITC) GO TO 5305                             S37200
00677   274*          ILOOK=ITP+JTRUE                                          S37300
00700   275*          GO TO 5310                                               S37400
```

```
00701   276*     5305 IF (JTRUE.LE.ITP.OR.JTRUE.GT.ITP+ITC) GO TO 5315                  S37500
00703   277*          ILOOK=JTRUE-ITP                                                   S37600
00704   278*     5310 DO 5312 I=1,M                                                     S37700
00707   279*          IF (L(INOFF+I).EQ.ILOOK) GO TO 5330                               S37800
00711   280*     5312 CONTINUE                                                          S37900
00713   281*     5315 DO 5320 I=1,L20                                                   S38000
00715   282*          I1=L(L20FF+I)                                                     S38100
00717   283*          IF (A(I1,J1).LE.0.0)  GO TO 5320                                  S38200
00721   284*          Q=A(I1,NT)/A(I1,J1)                                               S38300
00722   285*          IF (Q.GT.Q1)  GO TO 5320                                          S38400
00724   286*          IF (Q.GE.Q1.AND.A(I1,J1).LE.PIV)  GO TO 5320                      S38500
00726   287*          Q1=Q                                                              S38600
00727   288*          I1X=I1                                                            S38700
00730   289*     5320 CONTINUE                                                          S38800
00732   290*          IF (I1X.EQ.0.OR.A(I1X,J1).LE.PIV)  GO TO 5330                     S38900
00734   291*          PIV=A(I1X,J1)                                                     S39000
00735   292*          IPIV=I1X                                                          S39100
00736   293*          JPIV=J1                                                           S39200
00737   294*          Y=PIV/EPS                                                         S39300
00740   295*          IF (PIV.GT.CEPS)  GO TO 5350                                      S39400
00742   296*     5330 CONTINUE                                                          S39500
00744   297*          IF (PIV.GT.EPS)  GO TO 5345                                       S39600
00746   298*     5335 A(ISPIV,JSPIV) = 0.                                               S39700
00747   299*          IBOUND = 1                                                        S39800
00750   300*          JPIV = JSPIV                                                      S39900
00751   301*          GO TO 5100                                                        S40000
00751   302*     C                                                                      S40100
00751   303*     C * * THE PROBLEM IS UNBOUNDED.                                        S40200
00751   304*     C                                                                      S40300
00752   305*      5344 L(3) = 3                                                         S40400
00753   306*          IX=3                                                              S40500
00754   307*          JPIV = JSPIV                                                      S40600
00755   308*          JCPIV = L(NOTOFF+JPIV)                                            S40700
00756   309*          IF (L(14) .EQ. 0) GO TO 5600                                      S40800
00760   310*          KFLAG = 1                                                         S40900
00761   311*          WRITE (6,9015) (ASTAR,I=1,120),JCPIV                              S41000
00770   312*          WRITE (6,9018) ITER,IPHASE,OBJ,(ASTAR,I=1,120)                    S41100
01001   313*          GO TO 5600                                                        S41200
01002   314*     5345 IF (PIV.GT.CEPS)  GO TO 5350                                      S41300
01004   315*          IF (L(5) .NE. 0) WRITE (6,9005) ITER,IPHASE,PIV                   S41400
01012   316*          IABC1=IABC1+1                                                     S41500
01013   317*     5350 IABC = IABC+1                                                     S41600
01014   318*          GO TO 5400                                                        S41700
01014   319*     C*************************************************************       S41800
01014   320*     C                                                            *       S41900
01014   321*     C     EXCHANGE A BASIC AND A NON-BASIC VARIABLE, I.E. PIVOT. *       S42000
01014   322*     C                                                            *       S42100
01014   323*     C*************************************************************       S42200
01015   324*      5400 PIVX=1.0/A(IPIV,JPIV)                                           S42300
01016   325*          IF (A(IPIV,JPIV).GE.PIVM) GO TO 5401                              S42400
01020   326*          PIVM=A(IPIV,JPIV)                                                 S42500
01021   327*          IMP=ITER+1                                                        S42600
01022   328*     5401 DO 5415 I=1,MOBJ                                                  S42700
01025   329*          IF (I.EQ.IPIV)  GO TO 5415                                        S42800
01027   330*          A(I,JPIV) = -PIVX*A(I,JPIV)                                       S42900
01030   331*          DO 5410 JX = 1,L10                                                S43000
01033   332*          J = L(L10FF+JX)                                                   S43100
01034   333*          IF (J.EQ.JPIV)  GO TO 5410                                        S43200
```

```
01036   334*          A(I,J) = A(I,J)+A(I,JPIV)*A(IPIV,J)                    S43300
01037   335*     5410 CONTINUE                                                S43400
01041   336*          A(I,NT) = A(I,NT) + A(I,JPIV)*A(IPIV,NT)                S43500
01042   337*          IF (A(I,NT).GE.0.0)  GO TO 5415                         S43600
01044   338*          IF (I.LE.N)  A(I,NT)=0.0                                S43700
01046   339*     5415 CONTINUE                                                S43800
01050   340*          DO 5420 JX = 1,L10                                      S43900
01053   341*          J = L(L10FF+JX)                                         S44000
01054   342*          IF (J.NE.JPIV)  A(IPIV,J)= PIVX*A(IPIV,J)               S44100
01056   343*     5420 CONTINUE                                                S44200
01060   344*          A(IPIV,NT) = PIVX*A(IPIV,NT)                            S44300
01061   345*          OJJ=-A(NOBJ,NT)                                         S44400
01062   346*          A(IPIV,JPIV)=PIVX                                       S44500
01063   347*          IS=L(INOFF+IPIV)                                        S44600
01064   348*          L(INOFF+IPIV)=L(NOTOFF+JPIV)                            S44700
01065   349*          L(NOTOFF+JPIV)=IS                                       S44800
01066   350*          IENT = L(INOFF+IPIV)                                    S44900
01067   351*          ILVE = IS                                               S45000
01070   352*          IB(1) = ' X(    '                                       S45100
01071   353*          IF (IENT .LE. N) GO TO 5425                             S45200
01073   354*          IB(1) = ' S(    '                                       S45300
01074   355*          IENT = IENT-N                                           S45400
01075   356*     5425 IF (IPHASE .EQ. 1) GO TO 5428                           S45500
01077   357*          IB(2) = ' X(    '                                       S45600
01100   358*          IF (ILVE .LE. N) GO TO 5460                             S45700
01102   359*          IB(2) = ' S(    '                                       S45800
01103   360*          ILVE = ILVE-N                                           S45900
01104   361*          GO TO 5460                                              S46000
01105   362*     5428 IF (IS .LE. N+ML+MG) GO TO 5440                         S46100
01105   363*     C                                                            S46200
01105   364*     C * * TRANSFER FOR NON-EQUALITY CONSTRAINT.                  S46300
01105   365*     C                                                            S46400
01107   366*          IB(2) = ' A(    '                                       S46500
01110   367*          ILVE = ILVE-N-ML                                        S46600
01111   368*          J=1                                                     S46700
01112   369*          DO 5430 I=1,L10                                         S46800
01115   370*          I1=L(L10FF+I)                                           S46900
01116   371*          IF (I1.EQ.JPIV)  J=J+1                                  S47000
01120   372*          L(L10FF+I)=L(L10FF+J)                                   S47100
01121   373*     5430 J=J+1                                                   S47200
01123   374*          L10=L10-1                                               S47300
01124   375*          GO TO 5460                                              S47400
01125   376*     5440 IB(2) = ' S(    '                                       S47500
01126   377*          IF (IS .LE. N) IB(2) = ' X(    '                        S47600
01130   378*          IF (IS .GT. N+ML) GO TO 5445                            S47700
01132   379*          IF (IS .GT. N) ILVE = ILVE-N                            S47800
01134   380*          GO TO 5460                                              S47900
01134   381*     C                                                            S48000
01134   382*     C * * TRANSFER FOR SLACK.                                    S48100
01134   383*     C                                                            S48200
01135   384*     5445 KH = L(NOTOFF+JPIV)-ML-N                                S48300
01136   385*          IF (L(L3OFF+KH) .NE. 0) GO TO 5448                      S48400
01140   386*          ILVE = ILVE-N                                           S48500
01141   387*          GO TO 5460                                              S48600
01142   388*     5448 IB(2) = ' A(    '                                       S48700
01143   389*          ILVE = ILVE-N-ML                                        S48800
01144   390*          L(L3OFF+KH)=0                                           S48900
01145   391*          A(MT,JPIV)=A(MT,JPIV)+1                                 S49000
```

```
                DO 5450  I=1,MT                                                 S49100
                A(I,JPIV)=-A(I,JPIV)                                            S49200
 5450 CONTINUE                                                                  S49300
 5460 IF (IX .NE. 2) IX = 1                                                     S49400
                GO TO 5500                                                      S49500
C***************************************************************************  S49600
C                                                                           *   S49700
C       PRINT ITERATION SUMMARIES.                                          *   S49800
C                                                                           *   S49900
C***************************************************************************  S50000
 5500 IF (L(IPRINT) + L(IPRINT+1) .EQ. 0) GO TO 5550                            S50100
                ITER1 = ITER+1                                                  S50200
                IND=0                                                           S50300
                IF (IC1 .LT. L(IPRINT) .OR. L(IPRINT) .EQ. 0) GO TO 5530        S50400
                IC1=0                                                           S50500
                WRITE (6,9010) IPHASE,ITER1,PIV,OBJ,IB(1),IENT,IB(2),ILVE       S50600
                GO TO 5535                                                      S50700
 5530 IND=1                                                                     S50800
 5535 IF (IC2 .LT. L(IPRINT+1) .OR. L(IPRINT+1) .EQ. 0) GO TO 5550              S50900
                KFLAG = 1                                                       S51000
                IC2=0                                                           S51100
                IF (IND.EQ.0)  GO TO 5540                                       S51200
 5538 WRITE (6,9010) IPHASE,ITER1,PIV,OBJ,IB(1),IENT,IB(2),ILVE                 S51300
 5540 GO TO 5600                                                                S51400
 5550 GO TO (2025,50,2012,2000),IX                                              S51500
C***************************************************************************  S51600
C                                                                           *   S51700
C       GET X AND PRINT BASIS IF KFLAG NE 0.                                *   S51800
C                                                                           *   S51900
C***************************************************************************  S52000
 5600 NOX = N+M+MG                                                              S52100
                DO 5605 I2=1,NOX                                                S52200
 5605 X(I2) = -.001*EPS                                                         S52300
                DO 5610 I2=1,M                                                  S52400
                IND = L(INDOFF+I2)                                              S52500
                IF (IND .LE. N+ML) GO TO 5609                                   S52600
                IF (IND .GT. N+ML+MG) GO TO 5608                                S52700
                KH = IND-N-ML                                                   S52800
                IF (L(L3OFF+KH) .EQ. 0) GO TO 5609                              S52900
 5608 X(IND+MG) = A(I2,NT)                                                      S53000
                GO TO 5610                                                      S53100
 5609 X(IND) = A(I2,NT)                                                         S53200
 5610 CONTINUE                                                                  S53300
                X(NOX+1) = OBJ                                                  S53400
                IF (KFLAG .EQ. 0) GO TO 5550                                    S53500
                KFLAG = 0                                                       S53600
                DO 5606 I=1,5                                                   S53700
                IB(I) = ' X(    '                                               S53800
 5606 CONTINUE                                                                  S53900
                WRITE (6,9012)                                                  S54000
                I3 = 0                                                          S54100
                DO 5625 I2=1,N                                                  S54200
                IF (X(I2) .LT. 0.) GO TO 5620                                   S54300
                I3 = I3+1                                                       S54400
                PR(I3) = X(I2)                                                  S54500
                IP(I3) = I2                                                     S54600
 5620 IF (I3 .LT. 5 .AND. I2 .LT. N) GO TO 5625                                 S54700
                IF (I3 .EQ. 0) GO TO 5625                                       S54800
```

```
01303   450*          WRITE (6,9011) (IB(J),IP(J),PR(J),J=1,I3)                    S54900
01313   451*          I3 = 0                                                       S55000
01314   452*     5625 CONTINUE                                                     S55100
01316   453*          N1 = N+1                                                     S55200
01317   454*          N2 = N+NLHG                                                  S55300
01320   455*          IF (N2 .LT. N1) GO TO 5637                                   S55400
01322   456*          I3 = 0                                                       S55500
01323   457*          DO 5629 I=1,5                                                S55600
01326   458*          IB(I) = ' S(   '                                             S55700
01327   459*     5629 CONTINUE                                                     S55800
01331   460*          WRITE (6,9017)                                               S55900
01333   461*          DO 5635 I2=N1,N2                                             S56000
01336   462*          IF (X(I2) .LT. 0.) GO TO 5630                                S56100
01340   463*          I3 = I3+1                                                    S56200
01341   464*          PR(I3) = X(I2)                                               S56300
01342   465*          IP(I3) = I2-N1+1                                             S56400
01343   466*     5630 IF (I3 .LT. 5 .AND. I2 .LT. N2) GO TO 5635                   S56500
01345   467*          IF (I3 .EQ. 0) GO TO 5635                                    S56600
01347   468*          WRITE (6,9011) (IB(J),IP(J),PR(J),J=1,I3)                    S56700
01357   469*          I3 = 0                                                       S56800
01360   470*     5635 CONTINUE                                                     S56900
01362   471*     5637 N1 = N2+1                                                    S57000
01363   472*          N2 = NOX                                                     S57100
01364   473*          IF (N2 .LT. N1) GO TO 5550                                   S57200
01366   474*          I3 = 0                                                       S57300
01367   475*          DO 5639 I=1,5                                                S57400
01372   476*          IB(I) = ' A(   '                                             S57500
01373   477*     5639 CONTINUE                                                     S57600
01375   478*          WRITE (6,9017)                                               S57700
01377   479*          DO 5645 I2=N1,N2                                             S57800
01402   480*          IF (X(I2) .LT. 0.) GO TO 5640                                S57900
01404   481*          I3 = I3+1                                                    S58000
01405   482*          PR(I3) = X(I2)                                               S58100
01406   483*          IP(I3) = I2-N1+1                                             S58200
01407   484*     5640 IF (I3 .LT. 5 .AND. I2 .LT. N2) GO TO 5645                   S58300
01411   485*          IF (I3 .EQ. 0) GO TO 5645                                    S58400
01413   486*          WRITE (6,9011) (IB(J),IP(J),PR(J),J=1,I3)                    S58500
01423   487*          I3 = 0                                                       S58600
01424   488*     5645 CONTINUE                                                     S58700
01425   489*          GO TO 5550                                                   S58800
01427   490*     9000 FORMAT(' EPSILON = 'G13.6,'  CAPITAL EPSILON = 'G13.6,'  'I5,' NON S58900
01427   491*         *-ZERO ENTRIES ARE EFFECTIVELY EQUAL TO ZERO.')               S59000
01430   492*     9001 FORMAT(' *'14X'MINIMUM PIVOT WAS'G12.5,' AT ITERATION'I4,'.'57X'*' S59100
01430   493*         *)                                                           S59200
01431   494*     9003 FORMAT(1H+65X'REAL OBJECTIVE FUNCTION ='G15.8)               S59300
01432   495*     9005 FORMAT('0* * WARNING * *  SMALL PIVOT ELEMENT AT ITERATION'I4,' OF S59400
01432   496*         * PHASE'I2,'. PIVOT ='G14.7)                                 S59500
01433   497*     9006 FORMAT(' *'118X'*'/' '120A1)                                S59600
01434   498*     9007 FORMAT(//'0* * WARNING * *  IN PHASE'I2,', THERE HAVE BEEN'I3,' AT S59700
01434   499*         *TEMPTS TO FIND AN ALTERNATE PIVOT ELEMENT.'//19X'OF THESE,'I3,' FO S59800
01434   500*         *UND NO ALTERNATE ELEMENT GREATER THAN CAPITAL EPSILON.')     S59900
01435   501*     9009 FORMAT(/////' '120A1/' *'118X'*'/' *'14X'END OF PHASE'I2,'.    OBJ S60000
01435   502*         *ECTIVE FUNCTION ='G18.8,4X'THERE WERE'I4,' ITERATIONS.'17X'*') S60100
01436   503*     9010 FORMAT (///'0PHASE'I2,' ITERATION',I4,'.  PIVOT='G13.6,'   OBJEC S60200
01436   504*         *TIVE FUNCTION='G15.8,'  ',A3,I3,') ENTERED THE BASIS,',A3,I3,') LE S60300
01436   505*         *FT.')                                                       S60400
01437   506*     9011 FORMAT(5(3XA3,I3,')='F12.6))                                S60500
01440   507*     9013 FORMAT('0BASIC VARIABLES'/)                                 S60600
```

```
01441   508*     9014 FORMAT(/////' '120A1/' *'118X'*'/' *  ERROR - THE PROBLEM IS INFEA  S60700
01441   509*         *SIBLE. THE CONSTRAINTS ASSOCIATED WITH THE ARTIFICIAL VARIABLES BE  S60800
01441   510*         *LOW ARE INCONSISTENT. *'/' *  IF NONE APPEAR, NUMERICAL DIFFICULTI  S60900
01441   511*         *ES HAVE BEEN ENCOUNTERED. THE LARGEST ENTRY IN THE OBJECTIVE FUNCT  S61000
01441   512*         *ION ROW',7X,'*'/' *  IS ',G12.5,101X,'*')                          S61100
01442   513*     9015 FORMAT(/////' '120A1/' *'118X'*'/' *  ERROR - THE PROBLEM IS UNBOU  S61200
01442   514*         *NDED. THE VARIABLE X('I3,') CAN ASSUME AN ARBITRARILY LARGE VALUE,  S61300
01442   515*         * THEREBY YIELDING'7X'*'/' *  AN ARBITRARILY LARGE VALUE OF THE OBJ  S61400
01442   516*         *ECTIVE FUNCTION.'63X'*')                                            S61500
01443   517*     9016 FORMAT (1H1)                                                        S61600
01444   518*     9017 FORMAT (1H )                                                        S61700
01445   519*     9018 FORMAT(' *'118X'*'/' *  THE ERROR WAS DETECTED AT ITERATION'I4,' O  S61800
01445   520*         *F PHASE'I2,'. AT THAT TIME THE OBJECTIVE FUNCTION VALUE WAS'G15.8,   S61900
01445   521*         *4X'*'/' *  AND THE FOLLOWING VARIABLES WERE BASIC.'77X'*'/' *'118X   S62000
01445   522*         *'*'/' '120A1)                                                       S62100
01446   523*     9019 FORMAT(///' '120A1/' *'118X'*'/' *  WARNING - COMPUTATIONAL INCONS  S62200
01446   524*         *ISTENCY INDICATED AT THE END OF PHASE 1. THE ALGORITHM WILL CONTIN   S62300
01446   525*         *UE WITH PHASE 2,'4X'*'/' *  BUT THE USER IS ADVISED TO CRITICIZE T  S62400
01446   526*         *HE RESULTS.'67X'*'/' *'118X'*'/' '120A1)                            S62500
01447   527*     9020 FORMAT('0* *  AT LEAST ONE ELEMENT OF THE RIGHT HAND SIDE COLUMN I  S62600
01447   528*         *S LESS THAN ZERO. SUBROUTINE TERMINATES.')                          S62700
01450   529*          END                                                                 S62800

      END OF UNIVAC 1108 FORTRAN V COMPILATION.      0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =    1 SEC.
PHASE 2 TIME =    0 SEC.
PHASE 3 TIME =    2 SEC.
PHASE 4 TIME =    0 SEC.
PHASE 5 TIME =    1 SEC.
PHASE 6 TIME =    2 SEC.

TOTAL COMPILATION TIME =    6  SEC
```

APPENDIX D: TIMING CONSIDERATIONS

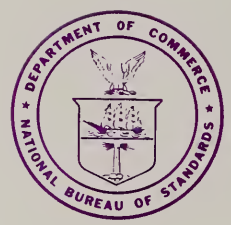
In order to provide the reader with some idea of the time involved in solving linear programming problems with RVSMPX and SIMPLX, we include the following table of problems that were run on the National Bureau of Standards' UNIVAC 1108. Each of the problems was randomly generated in such a way as to be bounded and feasible; and for each problem, all constraints were "greater than" constraints, so that a full Phase I was required.

Note that, as n increases from 20 to 120 while m is held fixed at 20, RVSMPX becomes faster than SIMPLX. The same is true for m held fixed at 50 and n increasing from 75 to 200. This illustrates the concept alluded to in section 1, and discussed at length in [7]. Basically, the concept is that for $n > 3m$, the revised simplex method unequivocally appears to be better computationally. Futhermore, in [7], Wagner argues that even for smaller n the revised simplex method is more desirable than the standard simplex method.

TABLE I: RESULTS OF SAMPLE RUNS

		RVSMPX				SIMPLX		
m	n	Elapsed Time	Number of Iterations	Time per Iteration		Elapsed Time	Number of Iterations	Time per Iterations
10	20	.3984	29	.0137	*	.3686	29	.0127
10	20	.3876	28	.0138		.3338	28	.0119
15	20	.5826	29	.0201	*	.4580	29	.0158
20	20	1.0838	44	.0246	*	.7898	44	.0180
20	30	1.9954	74	.0270		1.6608	74	.0224
20	60	2.5174	74	.0340	*	2.9472	74	.0398
20	80	4.3960	112	.0393	*	5.6848	112	.0508
20	120	3.3164	66	.0502	*	5.0066	66	.0759
50	75	43.5036	322	.1351		37.1222	329	.1128
50	150	58.7630	335	.1754	*	73.0950	339	.2156
50	200	87.8642	459	.1914	*	133.0888	474	.2808

Note: the times listed above are in seconds.


DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS