



Calhoun: The NPS Institutional Archive
DSpace Repository

Reports and Technical Reports

All Technical Reports Collection

1974-10

LPI - An interactive linear programming package

Shudde, Rex H.

Monterey, California. Naval Postgraduate School

<http://hdl.handle.net/10945/29376>

Downloaded from NPS Archive: Calhoun



Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>

NPS55Su74101

NAVAL POSTGRADUATE SCHOOL

Monterey, California



LPI--An Interactive Linear Programming Package

R. H. Shudde

October 1974

Approved for public release; distribution unlimited

FEDDOCS
D 208.14/2:
NPS-55SU74101

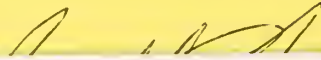
Naval Postgraduate School
Monterey, California

Rear Admiral Isham Linder
Superintendent

Jack R. Borsting
Provost

Reproduction of all or part of this report is authorized.

This report was prepared by:

A handwritten signature in dark ink, appearing to be "A. N. A.", written over a horizontal line.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS55Su741001	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) LPI--An Interactive Linear Programming Package		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) R. H. Shudde		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		12. REPORT DATE
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) <div style="display: flex; justify-content: space-between;"> <div> Linear Programming Separable Programming Wolfe's Algorithm </div> <div> Interactive Linear Programming Remote Terminal Linear Programming System System, Linear Programming </div> </div>		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) LPI is an interactive linear programming (L.P.) package designed primarily for instructional usage with the Cambridge Monitor System on the IBM/360 Computer. LPI removes the computational burden from the student without depriving him of the decision-making processes necessary for the successful solution of a L.P. LPI is self-instructing as to its own use; a minimum of CP/CMS commands are required to interface the student with LPI. LPI will allow primal simplex and/or dual simplex pivoting; sensitivity analysis of the "cost" coefficients and		

20. Continued

the "requirement" coefficients; the Separable Programming Algorithm; and the Wolfe Quadratic Programming Algorithm.

CONTENTS

I.	INTRODUCTION AND PURPOSE	3
II.	PROGRAM DESIGN CONSIDERATIONS	4
III.	MATHEMATICAL CONSIDERATIONS AND USAGE	5
	A. Standard Form	5
	B. Starting Basis and Cost Coefficient	5
	C. Pivoting	6
	D. Optimal Solutions	7
	E. Option Mode	7
	F. Sensitivity Analysis Mode	8
	G. The Restart File	8
	H. Error Handling	10
IV.	REFERENCES	
V.	APPENDICES	
	A. LPI User Instructions	13
	B. Sample Terminal Session	17
	C. LPI Listing	21
	DISTRIBUTION LIST	40

I. INTRODUCTION AND PURPOSE

LPI is an interactive linear programming program written in FORTRAN IV for use with the Cambridge Monitor System (CP/CMS) as implemented for the IBM/360 computer. LPI is intended to supplement courses in which the methodology of linear programming is taught.

The use of the simplex methods (primal and dual) to solve a linear program (LP) requires the student to be familiar with a battery of procedural rules for "determining feasibility", "maintaining feasibility", "improving the objective value", "determining unboundedness", "recognizing optimality", etc. If the student is performing hand computations and "something" goes wrong, he is usually at a loss to know whether he has violated one of the procedural rules or has performed a mathematical blunder. If the student is solving a LP using a standard computer program, the procedural decisions are made in the computer program and the student loses the insight gained by having to make his own decisions. Further, the use of a standard computer program often burdens the student with learning a rigid set of "input" requirements of the program which, if violated, lead to a plethora of error messages which are usually incomprehensible to all but the computer specialist.

LPI was written for the express purpose of providing the student with the best of the two worlds; namely it removes the burden of hand computations while it provides a computer interactive means of solving LP problems. LPI will not solve a LP, the student must do this for himself. He may use standard procedural rules to solve the LP; he may freely investigate the consequences of violating one or more of the standard rules; he may develop his own criteria for proceeding. Unfortunately there are a few steps which are necessary to link to the computer; these have been kept to the bare essentials. Once the LPI environment has been entered, the system is

self-explanatory¹ and practically user-proof.²

II. PROGRAM DESIGN CONSIDERATIONS

LPI was designed with the following objectives:

- A. Minimize the amount of computer instructions required to "get on" and use the system. Necessary and sufficient instructions for the user are contained in Appendix A.
- B. Minimize the chances of the user inadvertently "bombing out" of the system. Every user response is interrogated for validity. Non-valid responses are re-queried while distracting system generated error messages are suppressed.
- C. Allow the user every opportunity to make corrections. The input elements are usually via the keyboard. The user is given the opportunity to correct typing errors before preceding.
- D. Provide maximum recovery in the event of computer malfunction. To this end, a "restart file" is created to obviate retyping the input elements should the users virtual-computer be lost through a computer malfunction.
- E. Keep the options limited for the novice user while allowing extended options for the advanced user. An attempt is made to restrict unfamiliar queries from the novice. This is particularly true for the sensitivity analysis options. There are certain options (such as tableau print-suppression) available to the advanced user which are not self-documented by LPI queries and responses.

¹By "self-explanatory" it is meant that the LPI requirements are defined to the user by LPI itself. It does not mean that any of the simplex procedures are defined by LPI.

²There does not exist a completely user proof system. However, it is believed that the user of LPI cannot destroy the environment without a conscious and malicious effort.

F. Keep output formats clean when possible. When the entries in a tableau are of "reasonable" magnitude, fixed point formats are used. Should the fixed point boundaries be exceeded, LPI will automatically switch to a floating point format.

III. MATHEMATICAL CONSIDERATIONS AND USAGE

A. Standard Form

LPI may be used to solve any LP which is written in standard form:

$$\begin{array}{l} \min \\ \max \end{array} \sum_{j=1}^n c_j x_j = z$$

subject to

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, \dots, m$$

and

$$x_j \geq 0 \quad \text{for } j = 1, \dots, n.$$

The number of rows, m , is limited to 20 and the number of columns, n , is limited to 30.

All input information, the a_{ij} 's, the b_i 's, and the c_j 's is preserved in LPI for possible restarting or later modification. All of the pivoting and row transformations are performed in a separate matrix.

B. Starting Basis and Cost Coefficients

An initial tableau and starting basis are formed using artificial variables and the Charnes' M-method [1,2]. The cost coefficients of the artificial variables are set equal to M if z is to be minimized and are set equal to $-M$ if z is to be maximized, where $M > 0$.

For identification purposes, the artificial variables are subscripted 71 through 70 + m . The matrix of legitimate vectors is then searched for unit slack vectors (with zero cost coefficients) which can be substituted

for artificial vectors in the starting basis. The number of artificial vectors in the initial basis is thus minimized.

The indirect cost minus the direct cost, $z_j - c_j$, is then computed for each legitimate vector. The $z_j - c_j$ which is a linear function of M , is printed as two rows: the constant term and the coefficient of M . If there are no artificial vectors in the basis, then the printing of the coefficient of M is suppressed.

C. Pivoting

There are two pivoting responses which become apparent to the LPI user and two which are "hidden." LPI queries: DO YOU WANT YOUR PIVOT CHECKED FOR MIN-RATIO VIOLATION?

1. If the user responds 'no', then the user may pivot on any non-zero element in the tableau by specifying the column and row coordinates of the pivot element. In other words, the user may find a new basic solution without regard to maintaining primal or dual feasibility. If the user wishes to perform dual simplex pivoting, then the response must be 'no'.
2. If the user responds 'yes', then the user must still specify the column and row coordinates of the pivot element. If the primal simplex min-ratio criterion is violated by the user's choice of row, then pivoting does not take place. The user is informed that the min-ratio criterion has been violated and he is requered for a column and row number.
3. The two "hidden" responses are 'auto' and 'ntbl'. If the user responds 'auto' he is then queried for a pivot column. The row choice is automatically made using the primal min-ratio criterion and the entire new transformed tableau is displayed. The response

'ntbl' has the same affect as the 'auto' response except that most of the tableau printing is suppressed; the only elements that print are the basic x_j 's and the $(z_j - c_j)$'s. With this option, one may rapidly solve a LP or use the Wolfe algorithm, for example, with a minimum of printing. At a later point in LPI, these responses may be overridden if desired.

4. When the user no longer desires to pivot, he should respond by hitting the 'carriage return' key when queried for a new pivot column. This response will enter the option mode if the user has not previously entered the sensitivity analysis mode through the option mode. If the sensitivity analysis mode is in effect, then control returns to the next sensitivity analysis query.

D. Optimal Solutions

LPI will not inform the user when optimality has been reached. The user must recognize the occurrence of any of the possible stopping criteria: no feasible solution, unbounded solution, maximum or minimum solution, or optimal solution with alternate primal or dual optima.

E. Option Mode

To enter the option mode, the user must respond as directed in the last paragraph on pivoting. In the event that the user inadvertently entered the option mode, his first option is to return to the pivoting query. The next option is to either perform an option or not. If not, then a new problem may be started. If an option is desired, LPI will either list the available options or allow the knowledgeable user to supply one or more option keywords. The options allow for printing the dual variables, the basis inverse, removing the min-ratio violation check (including the 'auto' and 'ntbl' pivoting options), modify and rework the problem. The user may also enter the sensitivity analysis mode at this point.

F. Sensitivity Analysis Mode

The sensitivity analysis options are well documented by LPI.

Cost elements, c_j 's and "right-hand-side" elements, b_i 's, may be changed one at a time and in any order.

To replace c_j by $c_j + \Delta c_j$, the value of j and Δc_j must be specified. When Δc_j has been specified, the updated value of z and of all $z_j - c_j$ are printed. The user is then queried: PIVOT? If the user's response is 'yes', then he will be queried for a new pivot column and row, the users current pivot option will remain in effect. If the user's response is 'no', then he may change another cost or right-hand-side element.

To replace b_i by $b_i + \Delta b_i$, the value of i and Δb_i must be specified, the updated value of z and of all basic x_j 's are printed. Then the pivot options of 'auto', 'ntbl', or 'min-ratio' violation checking are replaced by 'no min-ratio' violation checking, so that dual simplex iterations may be performed. The user is then queried PIVOT? If the users response is 'yes' then he will be queried for a new pivot column and row number. If the user's response is 'no' then he may change another cost or right-hand-side element.

The user may leave the sensitivity analysis mode by responding 'quit' to the PIVOT? query or to the change in c_j or b_i query. Control is then passed back to the battery of option queries.

NOTE: Whenever a Δb_i or Δc_j is specified, the original b_i or c_j is changed accordingly. It is thus possible to generate an updated restart file by using the modify and rework option.

G. The Restart File

In the Naval Postgraduate School version of CP/CMS, the FORTRAN command READ(ℓ ,... is implemented as follows: If ℓ = 5, then the input

to CP/CMS is via the typewriter (remote terminal). If $\underline{\ell} = 1, 2, 3, 4$, or 7, then the CP/CMS input is assumed to be via "file ft0 $\underline{\ell}$ f001" and the file is further assumed to be in 80 column card images. LPI will generate 'file ft04f001' as a restart file when the user has completed typing in his input. Should the CP/CMS system malfunction, this input file will be available to re-establish the users input data. When queried '...TYPE, READ, OR RESTART', the user may type 'restart' to automatically implement the READ($\underline{\ell}$,... command with $\underline{\ell} = 4$).

The advanced LPI user may wish to prepare a data deck prior to his terminal session. The data deck may be read by LPI provided it is in the proper format and has the name "file ft0 $\underline{\ell}$ f001" where $\underline{\ell} = 1, 2, 3, 4$, or 7. If the data deck does not have the proper name, the user should consult the CP/CMS manual and learn how to use the CMS 'alter' command [3].

The format for an LPI data deck is given below in the proper input sequence.

1. First card:

m = no. of rows, n = no. of columns

(I2, 1X, I2)

2. Cost coefficients (one per card for as many cards as needed):

$j, c_j, (\text{name})_j$

(I2, 4X, F21.10, 5X, A4)

3. Delimiter card:

One blank card, or one card with 00 in columns 1 and 2 denotes the end of the cost coefficient input.

4. Matrix elements (one per card):

i, j, a_{ij}

(I2, 1X, I2, 1X, F21.10)

5. Delimiter card:

One blank card or one card with 00 in columns 1 and 2.

6. Right-hand-side elements:

i, b_i

(I2, 4X, F21.10)

7. Delimiter card optimal.

H. Error Handling

The most common error is that of the user inadvertently typing a non-numeric character in a data field which must consist of numeric characters only. This is the IHC215I-CONVERT error [4]. The FORTRAN error handling system will convert each erroneous character to a zero and generate an error message for each conversion. The diagnostic messages are suppressed in LPI by using the ERRSET subroutine [4] which is standard part of the FORTRAN error handling system. The user will not be aware of this conversion unless the numeric value generated is subsequently used as a row or column index and the index is out of the range allowed by LPI. In this case LPI will type a warning message to the user and requery. Generally this minimal error handling is sufficient. In all other cases, the user is given an opportunity to review all of the LP coefficients before proceeding. It is the users responsibility to detect such input errors.

The only other error message which is suppressed by means of ERRSET is the IHC218I - I/O ERROR. This error occurs if the user attempts to READ a non-existent data file or use the RESTART file when the restart file does not exist. LPI will detect such an error and a message will be typed which explains the probable sources of error to the user. An exit from LPI is then taken so that the user may take corrective action.

The "summary of errors" cannot be suppressed by any of the IBM standard routines. In order to suppress this final error message, a local subroutine KTT is used. A listing of this routine may be found in Appendix C. KTT should be implemented only by a resident CP/CMS systems programmer. KTT may be removed from LPI without affecting any of the LPI functions.

REFERENCES

1. Gass, S., "Linear Programming" 3rd ed. McGraw-Hill, 1969.
2. Hadley, C., "Linear Programming," Addison-Wesley, 1962.
3. Control Program-67/Cambridge Monitor System (CP/67/CMS). IBM Document GH20-0859.
4. IBM System/360 Operating System FORTRAN IV (G&H) Programmer's Guide Form C28-6817.

APPENDIX A

LPI USER INSTRUCTIONS

LOGIN, LINK, LOAD, EXECUTION, LOGOUT PROCEDURES:

In order to use LPI it is necessary to LOGIN to the Control Program (CP), LINK to the file on which LPI is stored, LOAD the Cambridge Monitor System (CMS), LOGIN the general user and LPI files, EXECUTE LPI and to LOGOUT of CMS and CP at the completion of EXECUTION. The commands below will enable you to do this.

(Those commands marked with an * are entered by the user.)

- * Turn the terminal on, depress the RETURN key, and wait patiently for the system to respond

cp-67 online xd.65 qsyosu

- * Depress the ATTN key once. The roll bar will advance one line and the keyboard will unlock. Then enter:

- * login xxxxgnn

nn is the terminal number. Terminal numbers are painted on the right side of the terminal stand.

xxxx is your user identification number.

EX: 0405g07 is terminal 07 and user 0405

The system will respond with:

ENTER PASSWORD:

Then you enter:

- * npg

The system will then give:

ENTER 4-DIGIT PROJECT NUMBER FOLLOWED BY 4-CHARACTER COST CENTER CODE:

Then you enter:

- * aaaabbbb

aaaa is the assigned project number.

bbbb is your section designator (or code, for faculty members)

The system will respond with the message of the day, such as:

HELLO...GEN USERS LOGIN WITH TERM NO + OWN ID...DUFFY

then:

READY AT 12.13.42 on 04/21/74

You are now logged in to CP. Next you must LINK to the LPI file. This is done by entering:

* link 3024p 191 193

The system will respond:

ENTER PASSWORD:

Then you enter:

* linprog

The system will respond:

SET TO READ ONLY

Now you must LOAD CMS by entering:

* ip1 cms

The system will respond:

CMS,.VERSION 01/21/74

Now you must LOGIN your general user file and the LPI file by entering:

* login 191

To which the system will respond:

R; T=0.01/0.10 12.15.52

Then enter:

* login 193 a,p

To which the system will respond:

** A (193) READ ONLY **

R; T=0.01/0.08 12.16.32

If you have gotten this far, take a deep breath and relax a minute before pushing on. You have succeeded in getting everything hooked up, now you are ready to execute the program.

Execute LPI by entering:

* begin

The system will respond with something like:

EXECUTION BEGINS...

GOOD AFTERNOON. YOU WISH TO SOLVE A LINEAR PROGRAM...

You are on your own now.

Eventually you will be asked:

DO YOU WANT TO SOLVE ANOTHER PROBLEM:

Note - This is not the precise message, but is a typical message.

If you respond 'yes' you will start over again, if you respond 'no' the system will respond:

IT HAS BEEN A PLEASURE SERVING YOU. HAVE A GOOD DAY.

R; T=0.01/0.07 13.15.16

Execution of the LPI Routine has ceased and now you can LOGOUT by entering:

* logout

The system will respond:

T=100.67/196.38 13.15.42

CP ENTERED, REQUEST, PLEASE.

CP

Then enter:

* log

The system will respond:

CONNECT=01.02.07 VIRTCPU=002.36.10 TOTCPU=003.51.34

LOGOUT AT 13.15.48 on 04/21/74

- * Turn off the terminal, roll up the paper to a perforation point and tear off your output.

NOTE:

A useful feature of CP/CMS to keep in mind is the procedure to correct typing errors. You can effectively back space by typing the character @ as many times as spaces you want to go back. This will delete what you have already typed in these spaces. DO NOT USE THE BACKSPACE KEY. Delete an entire line by typing the character ¢ and then depressing the RETURN key. You can then start the line over again. This feature will work while responding to the LPI queries.

If you are familiar with the more lengthy LPI statements and wish to suppress them, you may do so as follows:

1. Wait until typing of the message starts.
2. Press the 'ATTN' key once. They typing will stop with the characters 'CP'.
3. Press the 'ATTN' key once more. Wait for the click which indicates that the keyboard has unlocked.
4. Type the characters 'kt'. The CMS command KT is an abbreviation for 'kill typing'.
5. Depress the RETURN key.
6. Again wait for the keyboard to unlock. If you know what the suppressed query is, you may type in the appropriate response, otherwise depress the RETURN key once more to let LPI take over.

APPENDIX B

SAMPLE TERMINAL SESSION

begin

IF YOU ENCOUNTER ERRORS IN LINPROG, PLEASE REPORT THEM TO PROF. SHUDDE,
R-201B. BRING YOUR CONSOLE PRINT-OUT FOR DOCUMENTATION.
EXECUTION BEGINS...

GOOD AFTERNOON.

YOU WISH TO SOLVE A LINEAR PROGRAM, I PRESUME. VERY WELL, I WILL PROMPT
YOU FOR ENTRIES & TRY TO KEEP YOU OUT OF TROUBLE.
MAXIMUM SIZE IS CURRENTLY 20-BY-30, FOR $AX = B$ ONLY.
TYPE IN YOUR RESPONSES UNDER THE XXX'S. RIGHT ADJUST ALL INTEGER VALUES.
IMPLIED DECIMAL POINT IS SHOWN, BUT USE YOUR OWN TO MINIMIZE ERRORS.
YOU NEED TO ENTER ONLY THE NON-ZERO ELEMENTS, BUT YOU MUST ENTER YOUR OWN
SLACK VARIABLES, BUT NO ARTIFICIAL VARIABLES.
WHEN YOU ARE THROUGH WITH A SECTION, HIT THE "RETURN" KEY.
THE "MAX" OR "MIN" ENTRY AFFECTS ONLY THE SIGN OF THE ARTIFICIAL VARIABLES,
WHICH ARE SUBSCRIBED STARTING WITH 71.
WHEN YOU HAVE A BASIC SOLUTION, I WILL NO LONGER PRINT THE M-COEFFICIENTS
FOR THE ARTIFICIAL VARIABLES.
PIVOT ON ANY NON-ZERO ELEMENT. YOU MAY USE ANY PIVOTING CRITERIA THAT
YOU WANT.
IT'S UP TO YOU TO RECOGNIZE YOUR SOLUTION!
GOOD LUCK

DO YOU WISH TO TYPE IN YOUR TABLEAU, READ IN FROM YOUR OWN FILE, OR RESTART
FROM A SYSTEM WIPE-OUT?

THIS PROGRAM GENERATES 'FILE FT04F001' FOR RESTARTING.

RESPOND WITH -- TYPE, READ, OR RESTART.

type

INPUT THE SIZE OF THE A-MATRIX BELOW.

XX - NUMBER OF ROWS

02

XX - NUMBER OF COLUMNS.

05

NROW = 2 NCOL = 5

IF YOU WISH TO CORRECT EITHER OF THE ABOVE, RESPOND WITH "YES" NOW..

IF YOU WISH TO PROCEED, HIT THE "RETURN" KEY.

IF NEEDED, DO YOU WANT THE M-COEFFICIENTS SET UP FOR A MIN OR MAX PROBLEM?

min

OBJECTIVE FUNCTION (COL. NAME OPTIONAL)

```

C
0
L ELEMENT NAME
XX XXXXXXXX. XXXX
01 30¢
01 2. a-1
02 5. a-2
03 7. a-3
04 7.¢
04 0. s-1
05 0. s-2

```

A-MATRIX

```

R C
0 0
W L ELEMENT
XX XX XXXXXXXX.
01 01 3.
01 02 1.
01 03 2.

```

```

01 04 -1.
02 10 1.
YOUR INDEX IS OUT OF RANGE. TRY AGAIN.
02 01 1.
02 02 3.
02 3¢
02 03 4.
02 05 -1.

```

B-VECTOR

```

R
0
W ELEMENT
XX XXXXXXXX.
1 150.
YOUR INDEX IS OUT OF RANGE. TRY AGAIN.
01 150.
02 250.

```

	B	1	2	3	4	5
		A-1	A-2	A-3	S-1	S-2
OBJ FUNCT		2.0000	5.0000	7.0000	0.0	0.0
1	150.0000	3.0000	1.0000	2.0000	-1.0000	0.0
2	250.0000	1.0000	3.0000	4.0000	0.0	-1.0000

THIS IS YOUR INPUT. DO YOU HAVE CHANGES TO MAKE? YES, NO, OR RESTART.

no

RESTART FILE CREATED

DO YOU WANT YOUR PIVOT CHECKED FOR MIN-RATIO VIOLATION?

no

THEN - HERE WE GO. JUST BE CAREFUL OF YOUR PIVOTING CRITERIA.

TABLEAU NUMBER 0

	B	1	2	3	4	5
		A-1	A-2	A-3	S-1	S-2
1 X(71)	150.0000	3.0000	1.0000	2.0000	-1.0000	0.0
2 X(72)	250.0000	1.0000	3.0000	4.0000	0.0	-1.0000
OBJ FUNCT	0.0	-2.0000	-5.0000	-7.0000	0.0	0.0
M-COEFF.	400.0000,	4.0000	4.0000	6.0000	-1.0000	-1.0000

XX - PIVOT COLUMN?

03

XX - PIVOT ROW?

02

TABLEAU NUMBER 1

	B	1	2	3	4	5
		A-1	A-2	A-3	S-1	S-2
1 X(71)	25.0000	2.5000	-0.5000	0.0	-1.0000	0.5000
2 X(3)	62.5000	0.2500	0.7500	1.0000	0.0	-0.2500
OBJ FUNCT	437.5000	-0.2500	0.2500	0.0	0.0	-1.7500
M-COEFF.	25.0000	2.5000	-0.5000	0.0	-1.0000	0.5000

XX - PIVOT COLUMN?

01

XX - PIVOT ROW?

01

TABLEAU NUMBER 2

	B	1	2	3	4	5
		A-1	A-2	A-3	S-1	S-2
1 X(1)	10.0000	1.0000	-0.2000	0.0	-0.4000	0.2000
2 X(3)	60.0000	0.0	0.8000	1.0000	0.1000	-0.3000
OBJ FUNCT	440.0000	0.0	0.2000	0.0	-0.1000	-1.7000

XX - PIVOT COLUMN?

02

XX - PIVOT ROW?

02

TABLEAU NUMBER 3

	B	1	2	3	4	5
		A-1	A-2	A-3	S-1	S-2
1 X(1)	25.0000	1.0000	0.0	0.2500	-0.3750	0.1250
2 X(2)	75.0000	0.0	1.0000	1.2500	0.1250	-0.3750
OBJ FUNCT	425.0000	0.0	0.0	-0.2500	-0.1250	-1.6250

XX - PIVOT COLUMN?

YOU INDICATE THAT YOU'RE THROUGH PIVOTING. ARE YOU QUITE SURE?
yes

THERE ARE NUMEROUS OPTIONS THAT YOU MAY PERFORM AT THIS TIME. DO YOU WISH TO
USE ONE OR MORE OF THEM?
no

HOW ABOUT STARTING A NEW PROBLEM THEN?
no

ABOUT ALL THAT IS LEFT IS TO CALL IT A DAY. DO YOU WANT TO QUIT,? IF NO,
THEN WE WILL REVIEW THE OPTIONS.
yes

I GUESS I SHOULD HAVE ASKED YOU THAT IN THE FIRST PLACE. SORRY ABOUT THAT...
IT HAS BEEN A PLEASURE SERVING YOU.

...HAVE A GOOD DAY!...
R; T=1.09/3.22 13.34.47

LPI Listing

```

C
C INTERACTIVE LINEAR PROGRAMMING ROUTINE. SHLDD E APR '74
C
IMPLICIT REAL*8(A-H,O-Z)
INTEGER*4 YES//YES//,NO//NO//,MIN//MIN//,MAX//MAX//
1 REWORK//REWC//,MODIFY//MODI//,DUAL//DUAL//,INVERS//INVE//,
2 RATIO//RATI//,SENS//SENS//,DONE//DONE//,ALTC//ALTC//
3 TYPE//TYPE//,READ//READ//,REST//REST//,BLNK//BLNK//,LABL(51)
4 STOP//STOP//,QUIT//QUIT//,OBJ//OBJ//,RHS//RHS//,STO//STO//
5 QUI//QUI//,NOTBL//NOTBL//
COMMON /TABL$/A(22,51),T(22,51),IBV(20)
COMMON /STAT$/NROW,NCOL,NROW1,NROW2,NCOL1,NCOL2,NCOLM
C
CALL ERRSET(218,256,-1,1)
CALL ERRSET(215,256,-1,1)
INTPE = 4
C
WRITE(6,6000)
6000 FORMAT('IGOOD AFTERNOON..// YOU WISH TO SOLVE A LINEAR PROGRAM, I
1 PRESUME. VERY WELL, I WILL PROMPT YOU FOR ENTRIES & TRY.
2 .. TO KEEP YOU OUT OF TROUBLE..// MAXIMUM SIZE IS CURRENTLY.
3 .. 20-BY-30, FOR AX = B ONLY..//
4 .. TYPE IN YOUR RESPONSES UNDER THE XXX'S..//
5 .. RIGHT ADJUST ALL INTEGER VALUES..// IMPLIED DECIMAL POINT.
6 .. IS SHOWN, BUT USE YOUR OWN TC MINIMIZE ERRORS..//
7 .. YOU NEED TO ENTER ONLY THE NON-ZERO ELEMENTS, BUT YOU MUST.
8 .. ENTER YOUR OWN SLACK VARIABLES, BUT NC ARTIFICIAL VARIABLES..//
9 .. WHEN YOU.
10 .. ARE THROUGH WITH A SECTION, HIT THE "RETURN" KEY..// THE "MAX"
11 .. OR "MIN" ENTRY AFFECTS ONLY THE SIGN OF THE.
12 .. ARTIFICIAL VARIABLES, WHICH ARE SUBSCRIPTED STARTING WITH 71..//
13 .. WHEN YOU HAVE A BASIC SOLUTION, I WILL NC LONGER PRINT THE.
14 .. M-COEFFICIENTS FOR THE ARTIFICIAL VARIABLES..//
15 .. PIVOT ON ANY NON-ZERO ELEMENT. YOU MAY USE ANY.
16 .. PIVOTING CRITERIA THAT YOU WANT..//
17 .. IT'S UP TO YOU TO RECOGNIZE YOUR SOLUTION.//
18 .. GOOD LUCK .....)
C
INITIALIZE A-MATRIX
1010 REWIND 5
1020 DO 1030 J = 1,51
LABL(J) = BLNK
DO 1030 I = 1,22
A(I,J) = 0.D0
CONTINUE
1030 REWIND 5
1040 WRITE(6,6001)
6001 FORMAT('ODO YOU WISH TO TYPE IN YOUR TABLEAU, READ IN FROM ',
1 'YOUR OWN FILE, OR RESTART FROM A SYSTEM WIPE-OUT?')
2 'THIS PROGRAM GENERATES FILE FT04F001..// FOR RESTARTING..//
3 ' RESPOND WITH -- TYPE, READ, OR RESTART..')

```

```

1050 READ(5,5004,END=1040) NRES
      IF (NRES .EQ. TYPE) GC TO 1130
      IF (NRES .EQ. READ) GC TO 1060
      IF (NRES .EQ. REST) GC TO 1050
      IF (NRES .EQ. STOP) GC TO 2130
      WRITE(6,6020)
      GC TO 1040
      INFILE = 4
      GO TO 1070
1060 REWIND 5
6002 WRITE(6,6002)
      FORMAT(1, OFROM, WHAT FILE DO YOU WISH TO READ? YOUR CHOICES ARE - ,
1      1, 2, 3, 4 OR 7.)
5000 READ(5,5000,END=1040) INFILE
      FORMAT(11)
      IF (1 .LE. INFILE .AND. INFILE .LE. 4) GO TO 1070
      IF (INFILE .EQ. 7) GO TO 1070
      IF (INFILE .EQ. STOP) GO TO 2130
      WRITE(6,6003) INFILE
6003 FORMAT(1, SORRY, FILE '14,' IS NOT ALLOWED. TRY AGAIN..)
1070 GO TO 1060
      CONTINUE
      REWIND INFILE
      READ(INFILE,5003,END=1120) NROW,NCOL
      IF (NROW .LT. 1 .OR. NROW .GT. 20) GO TO 1120
      IF (NCOL .LT. 1 .OR. NCOL .GT. 30) GO TO 1120
      NCOL1 = NCOL + 1
      NROW1 = NROW + 1
1080 READ(INFILE,5001,END=1120) I,TEMP,NAME
5001 FORMAT(12,4X,F21.10,5X,A4)
      IF (1 .EQ. 0) GO TO 1090
      IF (1 .LT. 0 .OR. I .GT. NCOL) GC TO 1120
      A(NROW1,I) = TEMP
      LABL(I) = NAME
      GO TO 1080
1090 READ(INFILE,5002,END=1120) I,J,TEMP
5002 FORMAT(12,1X,I2,1X,F21.10)
      IF (1 .EQ. 0) GO TO 1100
      IF (1 .LT. 0 .OR. I .GT. NROW) GO TO 1120
      IF (J .LE. 0 .OR. J .GT. NCOL) GC TO 1120
      A(I,J) = TEMP
      GO TO 1090
1100 READ(INFILE,5001,END=1110) I,TEMP
      IF (1 .EQ. 0) GO TO 1110
      IF (1 .LT. 0 .OR. I .GT. NROW) GO TO 1120
      A(I,NCOL1) = TEMP
      GC TO 1100
1110 WRITE(6,6004)
6004 FORMAT(1, YOUR DATA HAS BEEN SUCCESSFULLY READ IN. YOU WILL ,
1      1, BE ASKED FOR NEW ELEMENTS SHORTLY.)/

```

LPI100510
 LPI100520
 LPI100530
 LPI100540
 LPI100550
 LPI100560
 LPI100570
 LPI100580
 LPI100590
 LPI100600
 LPI100610
 LPI100620
 LPI100630
 LPI100640
 LPI100650
 LPI100660
 LPI100670
 LPI100680
 LPI100690
 LPI100700
 LPI100710
 LPI100720
 LPI100730
 LPI100740
 LPI100750
 LPI100760
 LPI100770
 LPI100780
 LPI100790
 LPI100800
 LPI100810
 LPI100820
 LPI100830
 LPI100840
 LPI100850
 LPI100860
 LPI100870
 LPI100880
 LPI100890
 LPI100900
 LPI100910
 LPI100920
 LPI100930
 LPI100940
 LPI100950
 LPI100960
 LPI100970
 LPI100980
 LPI100990
 LPI101000

```

2  ! RESPOND BY HITTING THE "RETURN" KEY.))
GC TO 1170
C
1120 REWIND 5
6005 WRITE(6,6005)
      FORMAT(10Y)OUR DATA HAS NOT BEEN SUCCESSFULLY READ. THE MOST
1  ! , PROBABLE CAUSE IS THAT ONE OF YOUR
2  ! , ROW OR COLUMN NUMBERS IS OUT OF RANGE. I ASSUME YOU WILL
3  ! , WANT TO EDIT YOUR DATA BEFORE CONTINUING.
4  ! , ALSO CHECK TO SEE THAT YOUR LOGIN & LINK COMMANDS,
5  ! , ARE EXACTLY AS SPECIFIED.// JOB TERMINATED.))
      GO TO 2140
      INPUT MATRIX SIZE
C
1130 REWIND 5
6006 WRITE(6,6006)
1140 FORMAT(10I)NPUT THE SIZE OF THE A-MATRIX BELOW.))
      REWIND 5
6007 WRITE(6,6007)
      FORMAT(10X) - NUMBER OF ROWS))
5003 READ(5,5003,END=1140) NROW
      FORMAT(12,1X,12,1X,F7.0)
      IF(1.LE.NROW.AND.NROW.LE.20) GO TO 1150
6008 WRITE(6,6008)
      FORMAT(10I)YOUR RESPONSE IS OUT OF MY RANGE. TRY AGAIN.))
      GO TO 1010
1150 REWIND 5
6009 WRITE(6,6009)
      FORMAT(10X) - NUMBER OF COLUMNS.))
      READ(5,5003,END=1150) NCOL
      IF(1.LE.NCOL.AND.NCOL.LE.30) GO TO 1160
      WRITE(6,6008)
      GO TO 1150
1160 REWIND 5
6010 WRITE(6,6010) NROW,NCOL
      FORMAT(10I)NROW = ,12, NCOL = ,12/
1  ! , IF YOU WISH TO CORRECT EITHER OF THE ABOVE, RESPOND
2  ! , WITH "YES" NOW.// IF YOU WISH TO PROCEED, HIT THE
3  ! , "RETURN" KEY.))
      READ(5,5004,END=1170) NRES
5004 FORMAT(A4)
      IF (NRES.EQ. STOP) GO TO 2130
      GO TO 1140
1170 REWIND 5
C
      BOOKKEEPING
      IBL = 2 IF COLUMNS ARE SPECIFIED. IBL = 1 OTHERWISE
      IPRT = 1 IF NORMAL FIXED POINT OUTPUT FORMATS ONLY HAVE BEEN
      REQUIRED. IPRT = 2 OTHERWISE.
      IBL = 1
      IPRT = 1
C
C
C
C
C

```



```

NROW1 = NROW + 1
NROW2 = NROW + 2
NCOL1 = NCOL + 1
NCOL2 = NCOL + 2
NCOLM = NCOL1 + NROW
SETUP IDENTITY MATRIX TO STORE INVERSE & DUAL VARIABLES
DO 1180 I = 1, NROW
  J = NCOL1 + I
  1180 A(I,J) = 1.D0
C
C
C      MIN OR MAX??
      WRITE(6,6011)
6011 FORMAT('OIF NEEDED, DO YOU WANT THE M-COEFFICIENTS SET UP FOR A MI
1 N CR MAX PROBLEM?')
      READ(5,5004,END=1170) LPTPE
      IF(LPTPE.EQ. MIN.OR. LPTPE.EQ. MAX) GO TO 1190
      IF(LPTPE.EQ. STOP) GO TO 2130
      WRITE(6,6020)
      GO TO 1170
C
C      REWIND 5
      INPUT L.P. DATA
      WRITE(6,6012)
6012 FORMAT('OBJECTIVE FUNCTION (COL. NAME CPTCNAL)') C'/' C'/' C'/'
1200 CONTINUE
      READ(5,5005,END=1220) J,TEMP,NAME
      FORMAT(12,1X,F7.0,2X,A4)
      IF(J.LT. 1 .OR. J.GT. NCOL) GO TO 1210
      A(NROW1,J) = TEMP
      LABL(J) = NAME
      GO TO 1200
1210 WRITE(6,6013)
6013 FORMAT('YOUR INDEX IS OUT OF RANGE. TRY AGAIN.')
1220 REWIND 5
      WRITE(6,6014)
6014 FORMAT('OA-MATRIX') R C'/' O O'/' W L ELEMENT'/'
1230 CCNTINUE
      READ(5,5003,END=1250) I,J,TEMP
      IF(J.LT. 1 .OR. J.GT.NCOL) GO TO 1240
      IF(I.LT. 1 .OR. I.GT. NROW) GO TO 1240
      A(I,J) = TEMP
      GO TO 1230
1240 WRITE(6,6013)
      GO TO 1230
1250 REWIND 5
      WRITE(6,6015)

```

```

6015 FORMAT('OB-VECTOR',//, R//, O//, W ELEMENT',//
1, XX XXXXXX. '),
1260 CCNTINUE
READ(5,5005,END=1280) I,TEMP
IF(I.LT.1.OR.I.GT.NROW) GO TO 1270
A(I,NCCL1) = TEMP
GO TO 1260
1270 WRITE(6,6013)
GO TO 1260
C
C END DATA INPUT
1280 REWIND 5
C
C PRINT OUT ALL INPUT
CC 1290 J = 1,NCOL
IF (LABL(J) .NE. BLNK) ILBL = 2
1290 CCNTINUE
MPRT = (NCOL + 9)/10
DO 1301 JPRT = 1,MPRT
M2 = 10 * JPRT
M1 = M2 - 9
M2 = MIN0(M2,NCOL)
ITBL = 1
WRITE(6,6026) (I, I = M1,M2)
IF(ILBL.EQ.2) WRITE(6,6016) (LABL(J), J = M1,M2)
6016 FORMAT(19X,10(6X,A4))
6101 WRITE(6,6101) (A(NROW1,J),J=M1,M2)
FORMAT(' OBJ FUNCT ',10X,10F10.4)
DO 1300 I = 1,NROW
WRITE(6,6017) I,A(I,NCCL1),(A(I,J),J=M1,M2)
6017 FORMAT(1X,12,8X,11F10.4)
1300 CCNTINUE
1301 CCNTINUE
C
C QUERY FOR INPUT CHANGES
1310 REWIND 5
WRITE(6,6018)
6018 FORMAT('O THIS IS YOUR INPUT. DO YOU HAVE'
1, ' CHANGES TO MAKE? YES, NO, OR RESTART. ')
READ(5,5004,END=1310) NRES
IF (NRES .EQ. YES) GO TO 1190
IF (NRES .EQ. NO) GO TO 1320
IF (NRES .EQ. REST) GO TO 1040
IF (NRES .EQ. STOP) GO TO 2130
WRITE(6,6020)
6020 FORMAT('OSAY AGAIN...')
GO TO 1310
C
C WRITE DATA FOR RESTART
1320 CCNTINUE

```

```

LP102010
LP102020
LP102030
LP102040
LP102050
LP102060
LP102070
LP102080
LP102090
LP102100
LP102110
LP102120
LP102130
LP102140
LP102150
LP102160
LP102170
LP102180
LP102190
LP102200
LP102210
LP102220
LP102230
LP102240
LP102250
LP102260
LP102270
LP102280
LP102290
LP102300
LP102310
LP102320
LP102330
LP102340
LP102350
LP102360
LP102370
LP102380
LP102390
LP102400
LP102410
LP102420
LP102430
LP102440
LP102450
LP102460
LP102470
LP102480
LP102490
LP102500

```

```

REWIND 4
WRITE(4,5003) NROW,NCOL
DO 1330 J = 1,NCOL
TST = A(NROW1,J)
IF (TST.EQ. 0.DO .AND. ILBL.EQ. 1) GO TO 1330
WRITE(4,5001) J,TST,LABL(J)
CONTINUE
1330 WRITE(4,7704)
7704 FORMAT(80X)
DO 1350 I = 1,NROW
DO 1340 J = 1,NCOL
TST = A(I,J)
IF (TST.EQ. 0.DO) GO TO 1340
WRITE(4,5002) I,J,TST
CONTINUE
1340 WRITE(4,7704)
1350 FORMAT(80X)
DO 1360 I = 1,NROW
TST = A(I,NCOL1)
IF (TST.EQ. 0.DO) GO TO 1360
WRITE(4,5001) I,TST
CONTINUE
1360 END FILE 4
6021 WRITE(6,6021)
1370 FORMAT(:ORESTART FILE CREATED')
REWIND 5

      INITIALIZE OPTIONS
      MRATIO = 1 FOR NO PIVOT VIOLATION CHECKING ONLY
      MRATIO = 2 FOR PIVOT VIOLATION CHECKING ONLY
      MRATIO = 3 FOR AUTOMATIC PRIMAL PIVOT COMPUTATION
      NOPRT = 1 FOR FULL TABLEAU PRINTING
      NOPRT = 2 FOR XB & ZJ-CJ PRINTING ONLY
      MRATIO = 1
      NOPRT = 1
      ISSENS = 1
      CHECK PRIMAL PIVOTING?
      WRITE(6,6022)
6022 FORMAT(:000 YOU WANT YOUR PIVOT CHECKED FOR MIN-RATIO VIOLATION?')
READ(5,5004,END=1370) NRES
IF (NRES.EQ. NO) GO TO 1410
IF (NRES.EQ. YES) GO TO 1390
IF "AUTO" SETS MRATIO = 3 & NOPRT = 1
IF (NRES.EQ. AUTO) GO TO 1380
"NOTBL" SETS MRATIO = 3 & NOPRT = 2
IF (NRES.EQ. NOTBL) GO TO 1379
IF (NRES.EQ. STOP) GO TO 2130
WRITE(6,6020)
GO TO 1370
1379 NOPRT = 2

```

```

1380 MRATIO = 3
GO TO 1400
1390 MRATIC = 2
1400 WRITE(6,6023)
6023 FORMAT('00K. I WILL RESTRICT YOUR PIVOTING TO THE PROPER ROW.')
```

1410 GC TO 1420
CONTINUE

6024 WRITE(6,6024)
FORMAT('OTHER - HERE WE GO. JUST BE CAREFUL OF YOUR
1, PIVOTING CRITERIA.')

1420 CCNTINUE
NTBL = TABLEAU NUMBER
IART = 2 IF NO ARTIFICIAL VECTORS ARE BASIC. IART = 1 OTHERWISE.
IART = 1
NTBL = 0

INITIALIZE WORKING MATRIX

1430 DO 1440 I = 1, NROW
1440 IBV(I) = I + 70
1450 CONTINUE

DO 1460 J = 1, NCOLM
T(NROW1, J) = -A(NRGW1, J)
T(NROW2, J) = 0.00
DO 1460 I = 1, NROW
T(I, J) = A(I, J)
1460 CCNTINUE

IF (MRATIO .EQ. 1) GO TO 1490
DO 1480 I = 1, NROW
IF (T(I, NCOL1) .GE. 0.00) GO TO 1480
DO 1470 J = 1, NCOLM
T(I, J) = -T(I, J)
1470 CCNTINUE
1480 CONTINUE
1490 START TEST FOR UNIT VECTORS

DO 1510 J = 1, NCOL
IFL = 0
DO 1500 I = 1, NRGW
IF (T(I, J) .EQ. 0.00) GO TO 1500
IF (T(I, J) .NE. 1.00) GO TO 1510
IFL = IFL + 1
IF (IFL .GT. 1) GO TO 1510
II = I
CONTINUE

1500 IF (IFL .EQ. 0) GO TO 1510
IF (T(NROW1, J) .NE. 0.00) GO TO 1510
IBV(II) = J
CONTINUE

1510 END TEST FOR UNIT VECTORS

LP103010
LP103020
LP103030
LP103040
LP103050
LP103060
LP103070
LP103080
LP103090
LP103100
LP103110
LP103120
LP103130
LP103140
LP103150
LP103160
LP103170
LP103180
LP103190
LP103200
LP103210
LP103220
LP103230
LP103240
LP103250
LP103260
LP103270
LP103280
LP103290
LP103300
LP103310
LP103320
LP103330
LP103340
LP103350
LP103360
LP103370
LP103380
LP103390
LP103400
LP103410
LP103420
LP103430
LP103440
LP103450
LP103460
LP103470
LP103480
LP103490
LP103500


```

C          START SETUP OF M-COEFFICIENTS
          IF (LPYPE.EQ. MIN) GO TO 1530
          DO 1520 J = 1, NCOL1
          DO 1520 I = 1, NROW
          IF (IBV(I) .LT. 71) GO TO 1520
          T(NROW2,J) = T(I,J)
          CCNTINUE
1520      GC TO 1550
1530      DO 1540 J = 1, NCOL1
          DO 1540 I = 1, NROW
          IF (IBV(I) .LT. 71) GO TO 1540
          T(NROW2,J) = T(I,J)
          CCNTINUE
1540      C
C          OUTPUT SECTION
C
1550      WRITE(6,6025) NTBL
6025      FORMAT(//,0 TABLEAU NUMBER ',I4)
          DO 1555 I = 1, NROW
          IF (IBV(I) .LT. 71) GO TO 1555
          GO TO 1556
1555      CCNTINUE
1556      IART = 2
          CCNTINUE
          DO 1560 I = 1, NROW1
          DO 1560 J = 1, NCOL1
          IPRT1 = ICHK(I,J)
          GO TO (1560,1590,1610), IPRT1
          CCNTINUE
1560      MPRT = (NCOL + 9)/10
          DO 1583 JPRT = 1, MPRT
          M2 = 10 * JPRT
          M1 = M2 - 9
          M2 = MIN0(M2, NCOL)
          WRITE(6,6026) (I, I = M1, M2)
          IF (ILBL.EQ. 2) WRITE(6,6016)(LABL(J), J=M1, M2)
          FORMAT(0,16X,'B',10I10)
          IF (NOPRT.EQ. 2) GO TO 1571
          DO 1570 I = 1, NROW
          WRITE(6,6027) I, IBV(I), T(I, NCOL1), (T(I, J), J=M1, M2)
          FGRMAT(1X, I2, 1X, 'X(', I2, ', ', 2X, 11F10.4)
          CCNTINUE
1570      CCNTINUE
1571      WRITE(6,6028) T(NROW1, NCOL1), (T(NROW1, J), J=M1, M2)
6028      FORMAT(0,16X,'B',10I10)
          GO TO (1580, 1583), IART
1580      WRITE(6,6029) T(NROW2, NCOL1), (T(NROW2, J), J=M1, M2)
6029      FGRMAT(0,16X,'B',10I10)
1583      CCNTINUE
1581      IF (NOPRT.NE. 2) GO TO 1660

```

```

LP103510
LP103520
LP103530
LP103540
LP103550
LP103560
LP103570
LP103580
LP103590
LP103600
LP103610
LP103620
LP103630
LP103640
LP103650
LP103660
LP103670
LP103680
LP103690
LP103700
LP103710
LP103720
LP103730
LP103740
LP103750
LP103760
LP103770
LP103780
LP103790
LP103800
LP103810
LP103820
LP103830
LP103840
LP103850
LP103860
LP103870
LP103880
LP103890
LP103900
LP103910
LP103920
LP103930
LP103940
LP103950
LP103960
LP103970
LP103980
LP103990
LP104000

```



```

5006 FORMAT(I2)
    IF(IPIVCO .GE. 1 .AND. IPIVCO .LE. NCOL) GO TO 1670
    WRITE(6,6038)IPIVCO
6038 FORMAT(: YOU DON'T HAVE A COLUMN ',I2,',' TRY AGAIN')
1670 GO TO 1660
    CONTINUE
    IF (MRATIO .EQ. 3) GO TO 1690
    WRITE(6,6039)
    FORMAT(: XX - PIVOT ROW?)
6039 READ(5,5006,END=1800) IPIVRO
    IF (IPIVRO .GE. 1 .AND. IPIVRO .LE. NROW) GO TO 1680
    WRITE(6,6040)IPIVRO
6040 FORMAT(: SORRY, ROW ',I2,' DOESN'T EXIST')
    GO TO 1670
1680 DIV = 1/(IPIVRO,IPIVCO)
    GO TO (1750,1710,1690), MRATIO
1690 IPIVRO = 0
    TMIN = 1.D50
    DO 1700 I = 1,NROW
    DTST = T(I,IPIVCO)
    IF (DTST .LT. 1.D-10) GO TO 1700
    TST = T(I,NCOLL)/DTST
    IF (TST .GE. TMIN) GO TO 1700
    IPIVRO = I
    DIV = DTST
    TMIN = TST
1700 CONTINUE
    IF (IPIVRO .NE. 0) GO TO 1760
    WRITE(6,6041)
6041 FORMAT(: YOUR PIVOT COLUMN DOES NOT HAVE A POSITIVE ELEMENT.',
1, ' UNBOUNDED MAYBE?')
    GO TO 1660
1710 IF (DIV .GT. 0.D0) GO TO 1720
    WRITE(6,6042)
6042 FORMAT(: YOUR PIVOT IS NOT POSITIVE')
    GO TO 1660
1720 RTST = T(IPIVRO,NCOLL)/DIV
    DO 1730 I = 1,NROW
    IF (I .EQ. IPIVRO) GO TO 1730
    TT = T(I,IPIVCO)
    IF (TT .LE. 0.D0) GO TO 1730
    IF (T(I,NCOLL)/TT .LT. RTST) GO TO 1740
1730 CONTINUE
    GO TO 1750
1740 WRITE(6,6043)
6043 FORMAT(: YOU HAVE VIOLATED THE MIN-RATIO REQUIREMENT')
    GO TO 1660
1750 CONTINUE
    IF (DABS(DIV) .GE. 1.D-10) GO TO 1760
    WRITE(6,6044)

```

LPI04510
 LPI04520
 LPI04530
 LPI04540
 LPI04550
 LPI04560
 LPI04570
 LPI04580
 LPI04590
 LPI04600
 LPI04610
 LPI04620
 LPI04630
 LPI04640
 LPI04650
 LPI04660
 LPI04670
 LPI04680
 LPI04690
 LPI04700
 LPI04710
 LPI04720
 LPI04730
 LPI04740
 LPI04750
 LPI04760
 LPI04770
 LPI04780
 LPI04790
 LPI04800
 LPI04810
 LPI04820
 LPI04830
 LPI04840
 LPI04850
 LPI04860
 LPI04870
 LPI04880
 LPI04890
 LPI04900
 LPI04910
 LPI04920
 LPI04930
 LPI04940
 LPI04950
 LPI04960
 LPI04970
 LPI04980
 LPI04990
 LPI05000

```

6044 FORMAT('OPIVOT ELEMENT CLOSE TO OR EQUAL TC ZERC. TRY AGAIN..')
C
C
C      TRANSFORM TO NEW TABLEAU
1760 CCNTINUE
      IBV(IPIVRO) = IPIVCO
      DO 1770 J = 1, NCOLM
      T(IPIVRO, J) = T(IPIVRO, J)/DIV
1770 DO 1790 I = 1, NROW2
      IF (I.EQ. IPIVRO) GO TO 1790
      CM = -T(I, IPIVCO)
      IF (CM.EQ. 0.00) GO TO 1790
      DO 1780 J = 1, NCOLM
      T(I, J) = T(I, J) + T(IPIVRO, J)*CM
1780 CCNTINUE
1790 CCNTINUE
C
C      END TRANSFORMATION TO NEW TABLEAU
      NTBL = NTBL + 1
      GO TO 1550
1800 REWIND 5
      IF (ISENS.EQ. 2) GO TO 1890
      WRITE(6, 6045)
      FORMAT('YOU INDICATE THAT YOU'RE THROUGH PIVOTING..')
1804 1. ARE YOU QUITE SURE?')
      READ(5, 5004, END=1800) NRES
      IF (NRES.EQ. YES) GO TO 1810
      IF (NRES.EQ. NO) GO TO 1660
      IF (NRES.EQ. STOP) GO TO 2130
      WRITE(6, 6020)
      GO TO 1800
1810 REWIND 5
      WRITE(6, 6046)
      FORMAT('OTHER ARE NUMEROUS OPTIONS THAT YOU MAY PERFORM AT THIS
1. TIME. DO YOU WISH TO USE ONE OR MORE OF THEM?')
      READ(5, 5004, END=1810) NRES
      IF (NRES.EQ. YES) GO TO 2100
      IF (NRES.EQ. NO) GO TO 1820
      IF (NRES.EQ. STOP) GO TO 2130
      WRITE(6, 6020)
      GO TO 1810
1820 REWIND 5
      WRITE(6, 6047)
      FORMAT('DO YOU WANT YOUR OPTIONS ITEMIZED ALONG WITH THEIR
1. KEYWORD?')
      IF NO, THEN YOU MAY JUST TYPE THE KEYWORD..')
      READ(5, 5004, END=1820) NRES
      IF (NRES.EQ. STOP) GO TO 2120
      IF (NRES.EQ. YES) GO TO 1840

```

LPI05010
 LPI05020
 LPI05030
 LPI05040
 LPI05050
 LPI05060
 LPI05070
 LPI05080
 LPI05090
 LPI05100
 LPI05110
 LPI05120
 LPI05130
 LPI05140
 LPI05150
 LPI05160
 LPI05170
 LPI05180
 LPI05190
 LPI05200
 LPI05210
 LPI05220
 LPI05230
 LPI05240
 LPI05250
 LPI05260
 LPI05270
 LPI05280
 LPI05290
 LPI05300
 LPI05310
 LPI05320
 LPI05330
 LPI05340
 LPI05350
 LPI05360
 LPI05370
 LPI05380
 LPI05390
 LPI05400
 LPI05410
 LPI05420
 LPI05430
 LPI05440
 LPI05450
 LPI05460
 LPI05470
 LPI05480
 LPI05490
 LPI05500


```

6C48 WRITE(6,6048)
      FORMAT('BELOW ARE LISTED THE KEYWORDS FOLLOWED BY THEIR ',
1      'FUNCTION.',/
2      'RESTART - REWORK THIS SAME PROBLEM FROM THE START.',/
3      'MODIFY - MODIFY ONE OR MORE INPUTS TO THIS PROBLEM ',
4      ' & START AFRESH.',/
5      'DUAL - IF YOU CAN'T FIND THE DUAL VARIABLES IN YOUR ',
6      'TABLEAU, THEY WILL BE PRINTED.',/
7      'INVERSE - THE INVERSE OF YOUR CURRENT BASIS WILL BE PRINTED.',/
8      'RATIO - REMOVE THE MIN-RATIO VIOLATION CHECK & CONTINUE.',/
9      'SENSITIVITY - PERFORM SENSITIVITY ANALYSIS OPTIONS.',/
1     'DONE - WHEN YOU WISH TO LEAVE THIS SECTION OR START A NEW',
2     'PROBLEM.',)

C 1830 REWIND 5
      READ(5,5004,END=1820) NRES
1840 IF (NRES .EQ. REST) GO TO 1050
      IF (NRES .EQ. MODIFY) GO TO 1170
      IF (NRES .EQ. DUAL) GO TO 1850
      IF (NRES .EQ. INVERS) GO TO 1860
      IF (NRES .EQ. RATIO) GO TO 1870
      IF (NRES .EQ. SENS) GO TO 1880
      IF (NRES .EQ. NO) GO TO 1830
      IF (NRES .EQ. DONE) GO TO 2100
      IF (NRES .EQ. STOP) GO TO 2120
      WRITE(6,6020)
      GO TO 1820
1850 CONTINUE
      CALL PRNT1
      GO TO 1830
1860 CONTINUE
      CALL PRNT2
      GO TO 1830
1870 MRATIO = 1
      NOPRT = 1
      GO TO 1660
1880 CONTINUE
      ISENS = 2
6049 WRITE(6,6049)
      FORMAT('YOU MAY DO SENSITIVITY ANALYSIS ON THE OBJECTIVE ',
1      'FUNCTION AND/OR THE RIGHT-HAND-SIDE ELEMENTS.',/
2      'YOU MAY CHANGE ONLY ONE ELEMENT AT A TIME.',/
3      'WHEN QUERIED, TYPE IN .OBJ. OR .RHS., THE ELEMENT',/
4      'NUMBER (ROW), & THE AMOUNT THAT THE ELEMENT IS',/
5      'TO BE INCREASED (NEGATIVE FOR DECREASE).',/
6      'I WILL PRINT THE UPDATED ELEMENTS & THE NEW OBJ. FCTN. ROW',/
7      '(ZJ - CJ), OR THE NEW X.S.',/
8      'YOU WILL BE QUERIED "PIVOT?", TO WHICH YOU MAY RESPOND ONE',/
9      'OF THE FOLLOWING KEYWORDS--',/
1     'YES - YOU WISH TO MAKE AN ITERATION',/

```

LPI05510
 LPI05520
 LPI05530
 LPI05540
 LPI05550
 LPI05560
 LPI05570
 LPI05580
 LPI05590
 LPI05600
 LPI05610
 LPI05620
 LPI05630
 LPI05640
 LPI05650
 LPI05660
 LPI05670
 LPI05680
 LPI05690
 LPI05700
 LPI05710
 LPI05720
 LPI05730
 LPI05740
 LPI05750
 LPI05760
 LPI05770
 LPI05780
 LPI05790
 LPI05800
 LPI05810
 LPI05820
 LPI05830
 LPI05840
 LPI05850
 LPI05860
 LPI05870
 LPI05880
 LPI05890
 LPI05900
 LPI05910
 LPI05920
 LPI05930
 LPI05940
 LPI05950
 LPI05960
 LPI05970
 LPI05980
 LPI05990
 LPI06000

```

2  ! NO - YOU WANT TO CHANGE ANOTHER "B" CR "C" ELEMENT, /
3  ! DUAL - PRINT THE DUAL VARIABLES NOW, THEN RE-QUERY, /
4  ! INVE - PRINT THE INVERSE MATRIX NOW, THEN RE-QUERY, /
5  ! QUIT - YOU WANT TO LEAVE THE SENS. ANAL. MODE & DO SOMETH.,
6  ! INQ ELSE, /
7  ! STOP - YOU'VE HAD IT FOR THE DAY, /
1890 REWIND 5
6050 WRITE(6,6050)
1900 FORMAT(,00BJ,/, 'RHS NO ELEMENT',/ 'XXX XX XXXXXXXX.')
```

```

5007 READ(5,5007,END=1890) NRES,IX,TEMP
      FORMAT(A3,IX,I2,IX,F7.0)
      IF (NRES.EQ.OBJ) GO TO 1980
      IF (NRES.EQ.RHS) GO TO 1910
      IF (NRES.EQ.QUI) GO TO 2090
      IF (NRES.EQ.STG) GO TO 2130
      WRITE(6,6020)
      GO TO 1900
```

```

C
C
C  RHS - VARIATIONS
1910 CONTINUE
      IB = 2
      IF (1.LE. IX .AND. IX .LE. NROW) GO TO 1920
      WRITE(6,6008)
      GO TO 1900
1920 CONTINUE
      A(IX,NCOLL) = A(IX,NCOLL) + TEMP
      IPRT = 1
      ICOL = IX + NCOL1
      DO 1930 I = 1, NROW1
      TT = T(I,NCOLL) + T(I,ICOL)*TEMP
      T(I,NCOLL) = TT
      IF (ICLK(TT).NE. 1) IPRT = 2
1930 CONTINUE
      IF (IPRT.EQ.2) GO TO 1960
      DO 1940 I = 1, NROW
      IF (ICLK(A(I,NCOLL)).NE. 1) GO TO 1960
1940 CONTINUE
      WRITE(6,6051)
      FORMAT(,0,12X,'X-VECTOR B-VECTOR')
```

```

6051 DO 1950 I = 1, NROW
      WRITE(6,6027) I,IBV(I),T(I,NCOLL),A(I,NCOLL)
1950 CONTINUE
      WRITE(6,6052) T(NROW1,NCOLL)
      FORMAT(,0,VALUE OF OBJ. FCTN. = ,F10.4)
6052 GO TO 2060
1960 WRITE(6,6051)
      DO 1970 I = 1, NROW
      WRITE(6,6034) I,IBV(I),T(I,NCOLL),A(I,NCOLL)

```

```

LPI06010
LPI06020
LPI06030
LPI06040
LPI06050
LPI06060
LPI06070
LPI06080
LPI06090
LPI06100
LPI06110
LPI06120
LPI06130
LPI06140
LPI06150
LPI06160
LPI06170
LPI06180
LPI06190
LPI06200
LPI06210
LPI06220
LPI06230
LPI06240
LPI06250
LPI06260
LPI06270
LPI06280
LPI06290
LPI06300
LPI06310
LPI06320
LPI06330
LPI06340
LPI06350
LPI06360
LPI06370
LPI06380
LPI06390
LPI06400
LPI06410
LPI06420
LPI06430
LPI06440
LPI06450
LPI06460
LPI06470
LPI06480
LPI06490
LPI06500

```

```

1570 CONTINUE
      WRITE(6,6056) T(NROW1,NCOL1)
      GO TO 2060
C
C COST VARIATIONS
C
1580 CONTINUE
      IF (1.LE. IX .AND. IX .LE. NCOL) GO TO 1990
      WRITE(6,6038) IX
      GO TO 1900
C
1590 CONTINUE
      A(NROW1,IX) = A(NROW1,IX) + TEMP
      DO 2000 I = 1,NROW
        IIX = I
        IF (IBV(I) .EQ. IX) GO TO 2010
        CONTINUE
        T(NROW1,IX) = T(NROW1,IX) - TEMP
        GO TO 2030
      DO 2020 J = 1,NCOLM
        IF (J.EQ. IX) GO TO 2020
        T(NROW1,J) = T(NROW1,J) + TEMP*T(IIX,J)
        CONTINUE
C
2030 IPRT = 1
      DO 2040 J = 1,NCOL
        IF (ICLK(T(NROW1,J))) .NE. 1) GO TO 2050
        IF (ICLK(A(NROW1,J))) .NE. 1) GO TO 2050
      CONTINUE
      IF (ICLK(T(NROW1,NCOL1))) .NE. 1) GO TO 2050
      WRITE(6,6053)
      FORMAT('O J',5X,'C(J)',7X,'Z(J)-C(J)')
      WRITE(6,6054) {(J,A(NROW1,J)),T(NROW1,J),LABEL(J)},J=1,NCOL
      FORMAT(1X,I2,2F12.4,3X,A4)
      WRITE(6,6052) T(NROW1,NCOL1)
      GO TO 2060
C
2050 CONTINUE
      WRITE(6,6053)
      WRITE(6,6055) {(J,A(NROW1,J)),T(NROW1,J),LABEL(J)},J=1,NCOL
      FORMAT(1X,I2,1P2E12.3,3X,A4)
      WRITE(6,6056) T(NROW1,NCOL1)
      FORMAT('OVALUE OF OBJ. FCTN. = ',1PE12.3)
C
C QUERY
C
2060 REWIND 5
      WRITE(6,6057)
      FORMAT('PIVOT?',)
      READ(5,5004,END=2060) NRES
      IF (NRES .EQ. STOP) GO TO 2130

```

```

LPI06510
LPI06520
LPI06530
LPI06540
LPI06550
LPI06560
LPI06570
LPI06580
LPI06590
LPI06600
LPI06610
LPI06620
LPI06630
LPI06640
LPI06650
LPI06660
LPI06670
LPI06680
LPI06690
LPI06700
LPI06710
LPI06720
LPI06730
LPI06740
LPI06750
LPI06760
LPI06770
LPI06780
LPI06790
LPI06800
LPI06810
LPI06820
LPI06830
LPI06840
LPI06850
LPI06860
LPI06870
LPI06880
LPI06890
LPI06900
LPI06910
LPI06920
LPI06930
LPI06940
LPI06950
LPI06960
LPI06970
LPI06980
LPI06990
LPI07000

```

LPI07010
LPI07020
LPI07030
LPI07040
LPI07050
LPI07060
LPI07070
LPI07080
LPI07090
LPI07100
LPI07110
LPI07120
LPI07130
LPI07140
LPI07150
LPI07160
LPI07170
LPI07180
LPI07190
LPI07200
LPI07210
LPI07220
LPI07230
LPI07240
LPI07250
LPI07260
LPI07270
LPI07280
LPI07290
LPI07300
LPI07310
LPI07320
LPI07330
LPI07340
LPI07350
LPI07360
LPI07370
LPI07380
LPI07390
LPI07400
LPI07410
LPI07420
LPI07430
LPI07440
LPI07450
LPI07460
LPI07470
LPI07480
LPI07490
LPI07500

```

IF (NRES .EQ. NO) GO TO 1890
  .EQ. QUIT) GO TO 2090
IF (NRES .EQ. YES) GO TO 2080
  .EQ. DUAL) GO TO 2071
IF (NRES .EQ. INVERS) GO TO 2072
  .EQ. INVERS) GO TO 2072
WRITE(6,6020)
GO TO 2070
2071 CALL PRNT1
GO TO 2060
2072 CALL PRNT2
GO TO 2060
2080 IF (MRATIO .EQ. 1) GO TO 1550
  IF (IB .EQ. 1) GO TO 1550
MRATIO = 1
NOPRT = 1
WRITE(6,6058)
FORMAT(101) WILL NO LONGER CHECK YOUR PIVCTING FOR VIOLATIONS ',
1 , SINCE YOU ARE CHANGING B-VECTOR ELEMENTS & YOU',/
2 , WILL PROBABLY NEED TO USE DUAL SIMPLEX PIVOTING.')
```

```

2090 CONTINUE = 1
GO TO 1800
ISENS = 1
*****CONTINUE INPUT HERE*****
2100 REWIND 5
WRITE(6,6059)
FORMAT(101) ABOUT STARTING A NEW PROBLEM THEN?')
READ(5,5004,END=2100) NRES
IF (NRES .EQ. YES) GO TO 1020
IF (NRES .EQ. STOP) GO TO 2120
IF (NRES .EQ. NO) GO TO 2110
WRITE(6,6020)
GO TO 2100
2110 REWIND 5
WRITE(6,6060)
FORMAT(101) ABOUT ALL THAT IS LEFT IS TO CALL IT A DAY. ',
1 , DO YOU WANT TO QUIT,? IF NO, THEN WE WILL REVIEW',
2 , THE OPTIONS.')
```

```

2120 REWIND 5
READ(5,5004,END=2110) NRES
IF (NRES .EQ. YES) GO TO 2120
IF (NRES .EQ. NO) GO TO 1800
IF (NRES .EQ. STOP) GO TO 2120
WRITE(6,6020)
GO TO 2110
2120 CONTINUE = 1
REWIND 5
WRITE(6,6061)
FORMAT(101) GUESS I SHOULD HAVE ASKED YOU THAT IN THE FIRST PLACE.',
1 , SORRY ABOUT THAT....',/ IT HAS BEEN A PLEASURE SERVING YOU.')
```



```

6062 WRITE(6,6062)
C      FORMAT('O...HAVE A GOOD DAY ...')
C      THE FOLLOWING CALL WILL KILL THE SUMMARY ERROR TYPING, IF ANY.
C      KTT IS A NPS ROUTINE FOR CP/CMS
C 2140 CCNTINUE
C      CALL KTT
C      STOP
C      END

```

```

LPI07510
LPI07520
LPI07530
LPI07540
LPI07550
LPI07560
LPI07570
LPI07580

```

```

C
SUBROUTINE PRNT1
IMPLICIT REAL*8 (A-H,O-Z)

COMMON /TABL$/A(22,51),T(22,51),IBV(20)
COMMON /STAT$/NROW,NCOL,NRCW1,NROW2,NCOL1,NCOL2,NCOLM

WRITE(6,6025)
FORMAT('O DUAL VARIABLES.'/)
DO 10 J = NCOL2,NCOLM
IF (ICLK(T(NROW1,J)).NE. 1) GO TO 40
10 CONTINUE
WRITE(6,6026) (I,I = 1,NROW)
FORMAT(8X,11I10)
6026 WRITE(6,6027) (T(NROW1,J),J=NCOL2,NCOLM)
6027 FORMAT(11X,11F10.4)
RETURN
40 CONTINUE
WRITE(6,6028) (I,I = 1,NROW)
FORMAT(8X,10I11)
6028 WRITE(6,2007) (T(NROW1,J),J = NCCL2,NCOLM)
2007 FORMAT(11X,1P10E11.3)
RETURN
C
ENTRY PRNT2
WRITE(6,6029)
FORMAT('O INVERSE MATRIX.'/)
DO 50 I = 1,NROW
DO 50 J = NCOL2,NCOLM
IF (ICLK(T(I,J)).NE. 1) GO TO 70
50 CONTINUE
WRITE(6,6026) (I,I=1,NROW)
DO 60 I = 1,NROW
WRITE(6,2005) I, (T(I,J),J = NCCL2,NCOLM)
2005 FORMAT(1X,I2,8X,11F10.4/(11X,11F10.4))
60 CONTINUE
RETURN
70 CONTINUE
WRITE(6,6028) (I, I = 1,NROW)
DO 80 I = 1,NROW
WRITE(6,2006) I, (T(I,J),J = NCCL2,NCOLM)
2006 FORMAT(1X,I2,8X,1P10E11.3/(11X,1P10E11.3))
80 CONTINUE
RETURN
END

```

```

LP107600
LP107610
LP107620
LP107630
LP107640
LP107650
LP107660
LP107670
LP107680
LP107690
LP107700
LP107710
LP107720
LP107730
LP107740
LP107750
LP107760
LP107770
LP107780
LP107790
LP107800
LP107810
LP107820
LP107830
LP107840
LP107850
LP107860
LP107870
LP107880
LP107890
LP107900
LP107910
LP107920
LP107930
LP107940
LP107950
LP107960
LP107970
LP107980
LP107990
LP108000
LP108010
LP108020
LP108030

```

```

FUNCTION ICHK(A)
IMPLICIT REAL*8(A-H,O-Z)
ICHK=1
B=CABS(A)
IF (A .GE. 1.D5 .OR. A .LE. -1.D4) ICHK=2
IF (B .LT. 1.D-4 .AND. B .GE. 1.D-8) ICHK=3
RETURN
END

```

```

LP108050
LP108060
LP108070
LP108080
LP108090
LP108100
LP108110
LP108120

```

KT T

```
CSECT      *,R4,04)
USING      (14,R15
SAVE       R13,SAVAR+4
ST         R13,SAVAR
LA         562,X,40,
OI         R13,SAVAR+4
LM         R14,R4,12(R13)
LA         R15,0
BR         R14
CMSREG     18F
DS         END
SAVAR
```

KT T 00010
KT T 00020
KT T 00030
KT T 00040
KT T 00050
KT T 00060
KT T 00070
KT T 00080
KT T 00090
KT T 00100
KT T 00110
KT T 00120
KT T 00130
KT T 00140

Initial Distribution

	# of copies
Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
Dean of Research (Code 023) Naval Postgraduate School Monterey, California 93940	1
Library (Code 0212) Naval Postgraduate School Monterey, California 93940	2
Dean of Continuing Education (Code 024)	1
Office of Continuing Education (Code 500)	2
Naval Postgraduate School Monterey, California 93940	
Professor D. A. Schrady, Chairman (Code 55) Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
Professor D. G. Williams, Director (Code 0211) W. R. Church Computer Center Naval Postgraduate School Monterey, California 93940	1
Dr. William F. Brown, Director Computer Center Ball State University Muncie, Indiana 47306	1
Library, (Code 55) Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	3
Library (Code 0211) W. R. Church Computer Center Naval Postgraduate School Monterey, California 93940	3
Professors A. F. Andrus D. R. Barr D. C. Boger	1 each

G. H. Bradley
G. G. Brown
R. W. Butterworth
P. M. Carrick
R. S. Elster
R. N. Forrest
D. P. Gaver
J. K. Hartman
G. T. Howard
J. P. Hynes
C. R. Jones
M. B. Kline
H. J. Larson
P. A. W. Lewis
G. F. Lindsay
K. T. Marshall
A. W. McMasters
P. R. Milch
G. L. Musgrave
S. H. Parry
W. M. Raike
R. A. Stephen
J. G. Taylor
A. R. Washburn
B. R. Webster
H. H. Zweig

Department of Operations Research
and Administrative Sciences
Naval Postgraduate School
Monterey, California 93940

Professor G. L. Barksdale
" V. R. Kodres
Department of Mathematics
Naval Postgraduate School
Monterey, California 93940

1
1

Professor R. H. Shudde
Department of Operations Research
and Administrative Sciences
Naval Postgraduate School
Monterey, California 93940

U165507

DUDLEY KNOX LIBRARY - RESEARCH REPORTS



5 6853 01057752 1

U16550