CONTROL DATA

1604/1604-A COMPUTER

1604/1604-A

CDM3/LINEAR PROGRAMMING

CONTROL DATA 1604/1604-A COMPUTER

CDM3/LINEAR PROGRAMMING

CONTROL DATA CORPORATION
8100 34th Avenue South
Minneapolis 20, Minnesota

March, 1963
PUB. NO. 526

CONTENTS

CONTENTS (Continued)

INTRODUCTION

The Control Data Mathematical Programming System 3 (CDM3) is a linear programming system designed for use with the Control Data 1604 and 1604-A computers, and is operated under control of the CO-OP MONITOR SYSTEM. In addition to solving systems of linear equations, CDM3 solves certain systems of non-linear equations using the separable programming algorithm described in Chapter 1. CDM3 is an outgrowth of CDM2 and includes the features of that system.

In solving systems of equations, the system uses the Revised Simplex Method, the inverse being in product form. All arithmetic operations are performed in single precision, floating point format. CDM3 provides for multiple objective functions, multiple right hand sides and parametric programming, as well as a composite algorithm. It can handle problems up to 400 rows, 999 (non-artificial) variables and 69 sets of special variables (which are employed in the separable programming algorithm). Provision is made for up to 4000 non-zero matrix entries and up to 10,000 transformation entries. The data format conforms with the SHARE standard format and is similar to LP90 format.

Control of the CDM3 system is flexible; the user may select a variety of outputs and can exercise considerable control over the operation of the algorithm.

CDM3 is designed to operate with any machine configuration that conforms to the standard input-output requirements of the CO-OP MONITOR SYSTEM. Non-standard machine configurations may be used by establishing special control parameters and sense switches that are recognized by the CDM3 system and by the CO-OP MONITOR.

Chapter 1

PROBLEM FORMULATION

FORMULATION OF THE LINEAR PROBLEM

In its usual statement, the linear programming problem is posed as that of minimizing the linear form[1]

$$(1) \quad \sum_{J=1}^{N} C(J)\, X(J)$$

subject to the linear constraints

$$(2) \quad X(J) \geq 0, \; J = 1, \ldots\ldots, N$$

and

$$(3) \quad \sum_{J=1}^{N} A(I,J)\, X(J) = B(I), \; I = 1, \ldots\ldots, M.$$

The N quantities C(J) and the M times N quantities A(I,J) are arbitrary real numbers. The M quantities B(I) are assumed to be non-negative. This is no loss in generality since any of the equations in (3) may be multiplied by (-1) without changing the problem.

If the constraints (3) take the form of linear inequalities, they can normally be put into the correct form by the following device.

[1] Here and elsewhere in this manual the "FORTRAN-type" representation of variables and their subscripts will be used as a matter of typing convenience.

A linear inequality of the form:

$$\Sigma\ A(I,J)\ X(J) \leq B(I)$$

may be replaced by

$$\Sigma\ A(I,J)\ X(J) + S(N+1) = B(I).$$

The non-negativity of the new variable S(N+1) makes this relation equivalent to the old. Similarly, "greater than or equal" inequalities may be handled by subtracting the new variable from the left-hand side.

Any X(J) for which the constraints (2) do not apply (i.e., X(J) may assume negative as well as non-negative values) may be replaced by the difference X(N+1) - X(N+2) of two new non-negative variables.

COMPUTATIONAL FORMULATION

For computational purposes it is convenient to have the C(J) and A(I,J) of (1) and (3) together in a common format. The C(J) will be taken to be the top row of an M-rowed matrix A, that is, A(1,J) = C(J) for J = 1, . . .,N. The problem data can now be displayed as an M-rowed, N column matrix, with an additional column for the right-hand side.

$$(4)\quad \begin{matrix} A(1,1) & A(1,2) & . & . & . & A(1,N) & \\ A(3,1) & A(3,2) & . & . & . & A(3,N) & B(3) \\ A(4,1) & A(4,2) & . & . & . & A(4,N) & B(4) \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ A(M,1) & A(M,2) & . & . & . & A(M,N) & B(M) \end{matrix}$$

The row below the cost form is left blank because it performs a special role in the computation. Briefly, the computation is initiated by adjoining to the problem M-2 artificial variables in terms of which an immediate artificial starting solution satisfying (2) and (3) can be found. Unit costs are effectively entered for these variables into row 2, and the simplex method is used to reduce their total to zero, at which time a proper solution of the constraints is at hand. This initializing process is called Phase One of the simplex method. The subsequent minimization of the true cost form is referred to as Phase Two.

Problems can be formulated using several cost forms with the idea that after minimizing one form the program will proceed to minimize another, using the solution of the former as a point of departure. In this case, the first form to be minimized occurs directly above the blank row separating the cost forms from the equation constraints, the next cost form is directly above the first, and so on. The blank row would be displaced downward and M increased by all this.

Example -- A Breakfast Food Problem

A certain gentleman is confronted with the problem of choosing a breakfast which will satisfy the right requirements dictated by his physician. His choice from the four cereals listed below must yield exactly 150 calories, no more than 0.2 grams of sodium and at least 3 grams of protein. He is also interested in achieving this end as economically as possible.

He prepares a table in the form of the matrix (4) giving the cost, number of calories, and weights of sodium and protein for each of the four foods, and writes his requirements on the right. Realizing that two of the constraints are not equalities, he introduces two variables into the problem besides the four which will stand for the amounts of the breakfast foods to be used. The resulting table is

	CRISPIES	CRUNCHIES	CRACKLES	CHORTLES	VAR 5	VAR 6	REQMTS.
COST	4.0	7.0	8.0	6.0			
CALORIE	150.0	140.0	170.0	160.0			150.0
SODIUM	0.1	0.1	0.3	0.3	1.0		0.2
PROTEIN	2.0	4.0	5.0	3.0		-1.	3.0

Parameterizing

Suppose this gentleman hears that the cost of CRISPIES will be steadily increasing while the cost of CRACKLES will be correspondingly decreasing in the coming months, with the cost of CRUNCHIES and CHORTLES expected to remain the same.

He would then like to determine how much to change the menu as the cost varies and when to drop one cereal in favor of another. As the cost of CRISPIES increases he should consume less of them until finally he will eliminate them and choose a more economical cereal. While the cost decreases on CRACKLES he will consume more until it would not be practical to eat more.

Therefore there will be another row added to the matrix, above the cost row.

CFCN	1.0	0.0	-1.0	0.0

Chapters Two and Three will explain how to proceed from this point to final solution of the problem. The remainder of this chapter discusses the separable programming formulation.

FORMULATION OF THE SEPARABLE PROBLEM

Certain non-linear functions may be introduced into the set of constraints by means of the separable programming algorithm.

These functions are subject to the following restrictions:

Each non-linear function must be a function of only one variable or a linear combination of such functions (i.e., it is separable).

Each such function must be polygonal or replaceable by a polygonal approximation to it.

In addition, one must be willing to settle for a local minimum of the objective form (if the problem is convex, this is no restriction).

To introduce a non-linear function, $Y = F(X)$, into the system, the following procedure is used:

Suppose the function can be replaced by a piece-wise linear approximation to it. That is, there are a finite number of points P(J) with coordinates (X(J), Y(J)), J = 0, . . .,K on (or near) the graph of F(X) and linear interpolation between adjacent points will do. This relation between X and Y is described by introducing so-called special variables:

L(0), L(1), . . . ,L(K), and forming the constraints:

$$(5) \quad \sum_{J=0}^{K} L(J) = 1$$

$$(6) \quad \sum_{J=0}^{K} X(J)\, L(J) - X = 0$$

$$(7) \quad \sum_{J=0}^{K} Y(J)\, L(J) - Y = 0$$

Example -- The Sphere Problem

Consider the problem of maximizing Z, subject to:

$$(10) \qquad X^2 + Y^2 + Z^2 = 1$$

Restated, (10) becomes

$$(11) \qquad U + V + W = 1, \text{ where}$$

$$(11.1) \qquad U = X^2$$

$$(11.2) \qquad V = Y^2$$

$$(11.3) \qquad W = Z^2$$

replacing (11.1) by polygonal approximation gives:

$$(12.1) \qquad \begin{aligned} R00 + \quad R01 + \ . \ . \ . + \quad R05 &= 1.0 \\ -X + .0R00 + \ .2R01 + \ . \ . \ . + 1.0R05 &= 0 \\ -U + .0R00 + .04R01 + \ . \ . \ . + 1.0R05 &= 0 \end{aligned}$$

Similar sets of equations, and special variables (say SJJ and TJJ), replace (11.2) and (11.3). This results in a table in the form of matrix (4) as follows:

Row ID \ Col. ID	X	Y	Z	U	V	W	R00	R01	R02	R03	R04	R05	S00	S01	. .	T04	T05	B
COST			-1.															
SUMART																		
SPHERE				1.	1.	1.												1.
SUMR							1.	1.	1.	1.	1.	1.						1.
X	-1.							.2	.4	.6	.8	1.						
XSQ				-1.				.04	.16	.36	.64	1.						
SUMS													1.	1.	. .			1.
Y		-1.												.2	. .			
YSQ					-1.									.04	. .			
SUMT			-1.												. .	1.	1.	1.
Z						-1.									. .	.8	1.	
ZSQ															. .	.64	1.	

This sphere problem will be used as the example in the following chapters which describe input/output formats and operating procedures.

Chapter 2

INPUT/OUTPUT FORMATS

INPUT FORMATS -- INTRODUCTION

The basic input data for a problem are, of course, the elements of the array (4) of Chapter One (page 2). Each row and column of this array is given a unique identifier. The elements and their identifying information are prepared on punched cards as described in the following paragraphs.

Row Name Data

Each row of the computational matrix must be given a name. Associated with this name is its row number and an indication to the program of the constraint type. As indicated in Chapter One, the constraints of the computational formulation are equality constraints. Inequality constraints are handled by adding or subtracting slack variables. CDM2 will, if desired, automatically handle the addition of slack variables to the constraint matrix. This is done through the use of an indicator character in the row name data card:

Columns	Item	Remarks
1 - 11	Blank	
12	Row Type	The character in this column is the slack indicator; it is + if a positive slack is to be entered on this row; it is - if a minus slack is desired; it is blank if no slack is desired.

Columns	Item	Remarks
13 - 18	Row Name	Six alphanumeric characters.
19 - 24	Row Number	An integer, right adjusted.
25 - 80	Unrestricted	

Matrix Elements and Right-Hand Side Entries

The elements are punched one per card. Only non-zero elements need be punched. The card format is as follows:

Columns	Item	Remarks
1 - 6	Blank	
7 - 12	Column Name	Six alphanumeric characters (6 zeros are not allowed).
13 - 18	Row Name	Must appear in a format identical to that on the row name card.
19 - 30	Data Entry	Sign is + or blank for positive, - for negative. If decimal point is not punched, it is assumed between columns 24 and 25.
31-80	Unrestricted	

It is required that all entries for a given column be adjacent, although the ordering of the elements within a column is unrestricted. It is further required that all normal columns precede all special columns. (Special columns are those associated with the special variables of the separable programming formulation). Note the following: the last two characters of any special column name must be numeric and, further, these must be 00 for the first column and only the first column of each special set.

Columns 7-12 are irrelevant for the right-hand side entries.

Indicator Cards

There are also certain indicator cards which must be in the input deck. These are:

Title Card — Columns 1-66 contain identifying information. Columns 1-6 appear on the short output and in the identification portion of the restart punchouts. Columns 67-72 contain a row number increment.

The program assumes the rows are numbered such that the first objective is on row 3 and the first constraint on row 4 (MC = 3, MF = 4). If the row numbers on the row name card do not conform to this, then the increment necessary for conformity must be given as a six-digit, right-adjusted integer (bias).

ROW ID XX — in the card preceding the row identifiers, ROW ID in column 1-6
XX in columns 8-9 is the logical I/O unit from which the identifiers will be read.

MATRIX XX — in the card preceding the matrix entries MATRIX in columns 1-6
XX in columns 8-9 is the logical I/O unit from which matrix entries will be read.

SEPVAR XX — appears before the special columns
SEPVAR in columns 1-6
XX in columns 8-9 is the logical I/O unit from which the special columns will be read. This indicator card should not appear in the deck if there are no special columns.

Indicator Cards	FIRST BXX	appears in a card preceding the right-hand side entries FIRST B in columns 1-7 or RHS XX with RHS in columns 1-3 XX in columns 8-9 is the logical I/O unit from which the right hand side entries will be read.
	EOR	in columns 1-3 after the data following each indicator card used (see data listing).
	EOF or END	in columns 1-3 in the card immediately following the EOR after the last data entries.
	BASIS XX	when the basis or part of the basis is read in as part of the input data. BASIS in columns 1-6 XX in columns 8-9 is the logical I/O unit from which the basis data will be read.
	NEWRHS XX or SECOND XX	when reading in new right-hand side elements. NEWRHS or SECOND in columns 1-6 XX in columns 8-9 is the logical I/O unit from which the new righthand side elements will be read.
	ENDRHS	appears in column 1-6 after the last new right hand side elements following the NEWRHS or SECOND card outputs message NEW RIGHT HAND ELEMENTS READ IN on standard output. A new control card will then be read. NOTE: If XX in columns 8-9 are blanks or zero, it is assumed the preceding data is on logical tape 3 as used in standard procedure.

```
SPHERE PROBLEM - OR - FINDING THE NORTH POLE                    2
ROW ID
            COST      00
            SPHERE    02
            SUMR      03
            XEQU      04
            XSQEQU    05
            SUMS      06
            YEQU      07
            YSQEQU    08
            SUMT      09
            ZEQU      10
            ZSQEQU    11

EOR
MATRIX
        X   XEQU  -1.0
        Y   YEQU  -1.0
        Z   COST  -1.0
        Z   ZEQU  -1.0
        U   SPHERE 1.0
        U   XSQEQU-1.0
        V   SPHERE 1.0
        V   YSQEQU-1.0
        W   SPHERE 1.0
        W   ZSQEQU-1.0

EOR
SEPARABLE
        R000SUMR    1.0
        R010SUMR    1.0
        R010XEQU    0.2
        R010XSQEQU 0.04
        R020SUMR    1.00
        R020XEQU    0.40
        R020XSQEQU 0.16
        R030SUMR    1.00
        R030XEQU    0.60
        R030XSQEQU 0.36
        R040SUMR    1.00
        R040XEQU    0.80
        R040XSQEQU 0.64
        R050SUMR    1.00
        R050XEQU    1.00
        R050XSQEQU 1.00
        S000SUMS    1.00
        S010SUMS    1.00
        S010YEQU    0.20
        S010YSQEQU 0.04
        S020SUMS    1.00
        S020YEQU    0.40
        S020YSQEQU 0.16
        S030SUMS    1.00
        S030YEQU    0.60
        S030YSQEQU 0.36
        S040SUMS    1.00
        S040YEQU    0.80
        S040YSQEQU 0.64
        S050SUMS    1.00
        S050YEQU    1.00
        S050YSQEQU 1.00
        T000SUMT    1.00
        T010SUMT    1.00
        T010ZEQU    0.20
        T010ZSQEQU 0.04
        T020SUMT    1.00
        T020ZEQU    0.40
        T020ZSQEQU 0.16
        T030SUMT    1.00
        T030ZEQU    0.60
        T030ZSQEQU 0.36
        T040SUMT    1.00
        T040ZEQU    0.80
        T040ZSQEQU 0.64
        T050SUMT    1.00
        T050ZEQU    1.00
        T050ZSQEQU 1.00

EOR
FIRST B
            SPHERE 1.00
            SUMR   1.00
            SUMS   1.00
            SUMT   1.00

EOR
EOF
```

Listing of the input deck for the sphere problem of chapter one.

Sample Deck

OUTPUT FORMATS--INTRODUCTION

Three printed output reports are available in CDM3. The short output is used mostly for monitoring purposes. The long output is normally used to present the problem solution. The horizontal output gives more complete information than the long output and is used primarily when troubles occur.

Short Output

The short output is as follows:

IDENT	COND	FORM	ITER	OBJECTIVE	TRN ENTRIES
SPHERE	2	3	11	-.000000	34

IDENT -- First six characters of the problem identification card.

COND -- Condition number. An integer indicating one of five conditions:

1. Normal cycle; or a local optimum has occurred while parameterizing and program is seeking a new optimum.
2. Phase One completed.
3. Phase Two completed.
4. No feasible solution.
5. No pivot -- an infinite solution.

FORM -- Row number of the form being minimized.

ITER -- Current iteration number.

OBJECTIVE -- Current value of form being minimized.

TRN ENTRIES -- The current number of non-zero entries in the product form of the inverse.

J	X(J)	I	Y(I)	P(I)
Cost	1.000000	2	.200000	1.000000
Phase 1	.000000	3	.000000	.000000
Z	1.000000	4	.200000	.000000
X	.000000	5	-1.800000	.000000
U	.000000	6	-.360000	-.000000
R000	1.000000	7	9.000000	.000000
S000	1.000000	8	.000000	.000000
V	.000000	9	.000000	-.000000
S010	.000000	10	.000000	.000000
T050	1.000000	11	1.000000	1.000000
W	1.000000	12	.360000	-1.000000
R010	.000000	13	-9.000000	.000000

J -- Name of column (variable) in the basis.

X(J) -- Value of the variable J.

I -- Position of the variable in the basis. Note that the long output is sequenced on I.

Y(I) -- Transformed representation of last vector to enter the basis.

P(I) -- The negative of the shadow price on constraint I.

Horizontal Output

J	X(J)	DELTA(J)	POSITION
X	.000000	.111111	0
Y	.000000	.111111	0
Z	1.000000	.000000	11
U	.000000	.000000	3
V	.000000	.000000	6
W	1.000000	.000000	9
R000	1.000000	-.000000	5
R010	.000000	.000000	4
R020	.000000	.044444	0
R030	.000000	.133333	0
R040	.000000	.266667	0
R050	.000000	.444444	0
S000	1.000000	.000000	7
S010	.000000	.000000	8
S020	.000000	.044444	0
S030	.000000	.133333	0
S040	.000000	.266667	0
S050	.000000	.444444	0
T000	.000000	.444444	0
T010	.000000	.266667	0
T020	.000000	.133333	0
T030	.000000	.044444	0
T040	.000000	.000000	10
T050	1.000000	.000000	12

J -- Column (variable) name. Note that this output is ordered in the manner that the columns appear in the input data deck.

X(J) -- Value of variable J.

DELTA(J) -- Reduced cost coefficient on column J.

POSITION -- Position of variable J in the basis.

Chapter 3

STANDARD OPERATING PROCEDURES

INTRODUCTION

CDM3 provides operational flexibility to the user through parameters and monitoring controls. This flexibility is most useful in trouble problems and in experimenting with the revised simplex algorithm used here. The Standard Procedure described here is a very simple mode of operation for straightforward problems.

MACHINE CONFIGURATION

The machine configuration must include the minimum requirements for the CO-OP MONITOR SYSTEM. The procedure which follows assumes the monitor standard input and output units. The standard procedure will produce all output on the monitor standard output unit.

INPUT DATA

Problem data is prepared on cards as described in Chapter 2. If the card reader is the standard input unit (specified on MCS card), the complete input deck will consist of the following cards:

1) $\overset{7}{9}$BEGIN JOB XXX

2) $\overset{7}{9}$MCS Control Card

3) $\overset{7}{9}$EXECUTE, ,50,0.

4) $\frac{0}{\overset{7}{9}}$RELOCOM.

5) CDM3 on binary cards with two TRA (Transfer Address) cards at the end

* 6)	12	READY FOR NEW PROBLEM
* 7)	05	READ DATA
8)	Problem Data (on cards)	
* 9)	06	INITIATE NEW PROBLEM
*10)	07	COMPUTE
*11)	15	EXIT PROGRAM

BEGIN JOB CARD

The BEGIN JOB card is generally inserted by the operator to identify each job in a monitor run. This card is required even if the CDM3 program is run as a single job. It has a 7-9 punch in column 1 and BEGIN JOB starting in column 2.

The MCS control card which communicates programmer and accounting information to the Master Control System has a 7-9 punch in column 1 with eight fields starting in column 2. The fields are of variable length and must not include commas or periods. The significance of these fields is described in the CO-OP MONITOR SYSTEM Programmer's Manual. The fields on the MCS card are listed below:

1. Subordinate control routine name
2. Accounting identification
3. Programmer's name
4. Input-Output list
5. Time Limit
6. Line Limit
7. Recovery Key
8. Comments

---

* CDM3 control cards for the standard procedure. Only columns one and two are read.

Example of an MCS card:

$^{7}_{9}$COOP,24008-01,UBL,S/48/E/3=50/2=51/49=48,35,1000,5,CDM3.

The last field on the card must be terminated by a period.

LOGICAL UNIT NUMBERS

The following logical unit numbers are used by CDM3:

TAPE 2:	alternate output
TAPE 3:	problem data (ID, matrix, etc.)
TAPE 48:	punchout of basis headings
TAPE 49:	restart tape

Usually logical unit 49 is equated to logical unit 48.

The logical unit number and its program function is specified in the input/output list field on the MCS card. This list is interpreted by the Equipment-Assignment Routine in the Monitor and a table of assignments is constructed. In the above example, the data input is assigned as standard input (unit 50). Logical tape number 3 is assigned to standard input since the problem deck is not on a separate tape but part of the program deck. Alternate output is equated to standard output (unit 51). The logical tapes 49 and 48 are restart input and output units respectively. For further information about equipment assignment, see CO-OP MONITOR SYSTEM Programmer's Manual.

RELOCOM CARD

If the RELOCOM card is not used, an overflow of core will occur. The RELOCOM card will cause the starting location of numbered COMMON to be relocated to the start of the Relocatable Loader in the Monitor. This will provide a larger area for the program. The format of the RELOCOM card is: column 1 is zero minus seven and nine rows punched and starting in column 2 the word RELOCOM followed by a period in column 9. The RELOCOM card must be the first card read by the loader.

STANDARD OUTPUT

The following output appears on the Monitor standard output unit under the standard procedure. (Sample printouts are in Appendix I).

1. "CLEARED" -- This indicates that the "12" control card has been read, COMMON has been cleared, and the standard procedure set up.

2. "M mmm, N nnn, ENTRIES kkkkkk, SETS sss"
   mmm -- number of rows
   nnn -- number of columns
   kkkkkk -- number of non-zero matrix entries
   sss -- number of sets of special vectors
   This printout appears after the input data has been read.

3. "REINVERTING AFTER 0TH ITERATION. 0 TRANSFORMATIONS WITH 0 ENTRIES."
   This printout appears as a result of control card 06 -- initiate new problem.

4. At the end of phase one, the short output will be given.

5. At the end of phase two, the short and long output will be given.

   In addition, a check will be made on Ax - b (which should be zero), and the maximum error and sum of errors in the resulting vector will be printed:
   "MAX ERROR ON ROW mmm, ±d.dddddE±ee. SUM, ±d.dddddE±ee."
   A check is also made on the basic portion of pA, where p is the vector of shadow prices. It should be zero. Again the max error and sum are printed as above, except that the column number appears instead of the row number.

Also, at the end of phase two, restart punchouts are produced on logical unit 48 (normally the paper punch). These punchouts are described under restart procedures in Chapter Four.

6. If no feasible solution exists, or if no pivot can be found (infinite solution), then short, long, and horizontal outputs will be given along with the error checks mentioned above.

Chapter 4

OPERATIONAL CONTROLS

CDM3 is controlled primarily by cards. The following is a complete list of the control cards.

TABLE OF CONTROLS

01 READ NEW FIXED POINT PARAMETERS--3 through 9
02 READ NEW FLOATING POINT PARAMETERS--1 through 7
03 READ OUTPUT CONTROLS--6 STANDARD, 6 ALTERNATE
04 START PARAMETERIZING
05 READ DATA
06 INITIATE NEW PROBLEM
07 COMPUTE
08 FORM INVERSE
09 BUFFER OUT COMMON
10 BUFFER IN COMMON
11 READ NEW RIGHT HAND SIDES
12 READY FOR NEW PROBLEM
13 CHECK SOLUTION
14 CALL INPUT1
15 EXIT PROGRAM
16 SPECIAL OUTPUT--CALL QOT(6)
17 END FILE ON ALTERNATE OUTPUT
18 PAUSE 302
19 PAUSE 303
20 PAUSE 304
21 READ AND PRINT MESSAGE
22 PUNCH RESTART TAPE

Blank or "00" control cards will cause a "PAUSE 300" if they are read.

Again it should be noted that only the control number in columns 1 and 2 is interrogated.

Some use is also made of sense switches. The second section of this chapter gives a summary of the sense switch controls. The last section of this chapter discusses the use of certain controls in restart procedures.

FIXED POINT PARAMETERS

An array of 24 fixed point quantities is kept in common. This array is designated KM. KM(3), KM(4), KM(5), KM(6), KM(7), KM(8), and KM(9) are parameters which may be changed by reading the control card

01 READ NEW FIXED, etc.,

followed by a card with format 7I5 on which are punched KM(3) through KM(9) in that order. The items in KM are:

KM(1)-- Problem Identification Word -- the first six columns of the problem ID card.

KM(2)-- Current iteration number.

KM(3)-- Inversion frequency -- the number of iterations between calls of subroutines VER which re-inverts using the current basis headings. In the standard procedure this is set to zero -- i.e., VER will never be called because of the iteration count.

KM(4)-- Number of phases one to be done. In the standard procedure this is set to 1.

KM(5)-- Total number of forms to be minimized (total phases one and phases two). This is set to 2 in the standard procedure.

KM(6)-- Row number of the first objective form. Set to 2 in the standard procedure and to 3 when parameterizing costs.

KM(7)-- Number of iterations between calls of subroutine ERR which produces sum and maximum of errors in Ax-b and pA. It is zero in the standard procedure.

KM(8)-- Number of iterations between punchings of restart punchouts. Zero in the standard procedure.

KM(9)-- The iteration number after which a control card is to be read. Zero is the standard procedure.

KM(10)- Column dropped from basis.

KM(11)- Pivot row this iteration.

KM(12)- Number of negative delta-jays.

KM(13)- Column brought into basis.

KM(14)- Off-line page eject control.

KM(15)- On-line page eject control.

KM(16)- Condition number for this iteration.

KM(17)- Iterations to go until next error calculation. This count is initialized to KM(2).

KM(18)- Number of non-zero matrix entries.

KM(19)- Number of transformation entries.

KM(20)- Set to 1 for parameterizing costs with short output and to 1+IA for parameterizing costs with long output.

KM(21)- Non-zero if there is any form of off-line output, zero otherwise.

KM(22)- Column number of best delta-jay so far during a pricing pass.

KM(23)- Iterations to go until punchouts. This count is initialized to KM(8).

KM(24)- Iterations to go until reinversion. This count is initialized to KM(3).

The following example shows the use of the "01" control where it is desired to change KM(3)-inversion frequency, KM(7)-error checking frequency, and KM(8)-punchout frequency. Note that since every field on the card is read and its value assigned to the appropriate KM(J), it is necessary to punch all fields even though they are not different from the standard values.

```
01 READ FIXED POINT PARAMETERS

 50    1    2    2   20   40    0
```

In this example re-inversion will occur every 50 iterations, the next three parameters are standard, ERR checking will occur every 20 iterations, and punching will occur every 40 iterations.

FLOATING POINT PARAMETERS

An array of 16 floating point quantities is kept in common. This array is designated Z, and Z(1) through Z(7) are parameters which may be changed by reading the control card

02 READ NEW FLOATING POINT PARAMETERS

followed by a card with format 7E10.3 on which are punched Z(1) through Z(7) in that order.

The array Z contains:

Z(1) - A small positive number. In choosing the pivot row in subroutine row, denominators Y(I) less than this are deemed non-positive. In the standard procedure it is set to 0.00001, which seems to be a good general value.

Z(2) - A small positive number. Phase one is declared finished when the total infeasibility falls below this. A good general value is 10^{-6} times the sum of the right-hand side entries. Set 0.001 in the standard procedure.

Z(3) - A small negative number. Reaching of a minimum is declared when all the delta-jays are greater than this. A good general value should be about -10^{-6} times the largest cost-row entry. Set to -0.001 in the standard procedure.

Z(4) - Mixing parameter. In phase one the prices P(I) used in selecting the next basic column are formed as the sum of the phase one-form prices and this quantity times the prices from the next form. This is set to zero in the standard procedure. General value to use is about one-tenth the reciprocal of the largest entry in the first cost form.

Z(5) - In subroutine JMY, columns whose pivot-row entries are smaller than this are skipped. Set to zero in the standard procedure.

Z(6) - In subroutine PIV 3, transformation elements whose magnitudes are less than this are skipped. Set to zero in the standard procedure.

Z(7) - In subroutine CYC 3, the vector selected to enter the basis is rejected if its pivot is smaller in magnitude than this. Set to zero in the standard procedure.

Z(8) - Current theta (pivot ratio).

Z(9) - Determinant of current basis.

Z(10)- Current objective value.

Z(11)- Sum of absolute row or column errors.

Z(12)- Maximum absolute row or column error.

Z(13)- Best delta-jay so far during pricing pass.

Z(14)- Most negative right-hand side found during inversion.

Z(15)- Contains cost parameter.

Z(16)- Contains current value of cost parameter.

The following is an example of an "02" control card for changing Z(4) and Z(7) from those used in the standard procedure. Note that all seven fields are read and therefore must be punched even if some of them are the standard values.

```
02 READ NEW FLTING POINT PARAMETERS - 3 THROUGH 9. FORMAT 7E10.3

   1.0E-05   1.0E-03   -1.0E-03   1.0E-01   0.0E 00   0.0E 00   1.0E-02
```

OUTPUT CONTROL

Output in CDM3 is controlled by assigning the control numbers 0-4 as entries in the internally stored two-dimensional array KP(K,J). The first subscript, K, of this array is the condition number of the problem. It is an integer from 1 to 6 denoting respectively: normal cycle, end of phase one, end of phase two, no feasible solution, infinite solution, or special condition. The second subscript, J, is 1 for alternate controls or 2 for standard controls. When the output subroutine QOT(K) is called, alternate output will be governed by the value of KP(K,1), and standard output will be governed by the value of KP(K,2). The values 0-4 that KP(K,J) may be given have the following meaning

0 (or blank)	No output
1	Short
2	Short and long
3	Short and horizontal
4	Short, long, and horizontal.

(See Chapter 2 for a description of these outputs.)

Standard Output Control

The standard program uses the control characters 00000 for alternate output (thus there is none) and 01244 for standard output. This results in no output during normal cycling, short output at the end of phase one, short and long output at the end of phase two, and short, long, and horizontal output in case of no feasible solution or infinite solution.

Non-standard Output Control

Non-standard control is effected by reading in new values for the array KP(K,J) with the 03 control card. The six alternate controls are punched in columns 1-6 of the card following the control card, and the six standard controls are punched in columns 7-12 of that same card.

03 READ OUTPUT CONTROL, 6 ALTERNATE, 6 STANDARD 12444 01111

This control will give alternate output; short every iteration, long at the end of phase one, and short, long, and horizontal for conditions 3, 4, and 5. It will give short standard output for everything but normal iterations.

04 SET UP COST PARAMETERS reads card which contains the maximum value of the cost parameter to be used, in columns 1-20 in a F format. This is temporarily stored in Z(16). Columns 21-22 contain a positive integer or zero (or blank).

05 READ DATA reads the problem data from the monitor standard input unit.

06 INITIATE NEW PROBLEM will call subroutine NEW which forms the initial basis headings to start the problem. When control returns from NEW, subroutine VER is called to form the original inverse. (VER will automatically attach artificials if less than M basis headings were provided by NEW.)

07 COMPUTE transfers control to the main routine to start simplex cycling. It is used after the 06 control and after any interruption of the main routine if more cycling is to be done.

08 FORM INVERSE forms the inverse from the current basis headings (i.e., it calls subroutine VER).

09 BUFFER OUT COMMON calls subroutine RESTART (1), which will buffer out all of COMMON storage on logical tape number 1. This tape may be used to restart the calculation from this point at any later time.

10 BUFFER IN COMMON calls subroutine RESTART (2), which will buffer in all of COMMON storage from logical tape number 1.

11 READ IN NEW RIGHT HAND SIDE calls subroutine TAP(2), which will read new right hand side.

12 READY FOR NEW PROBLEM clears the data region (numbered common) and sets up for the standard procedure. If non-standard parameters have been set they will be reset to standard by this control.

13 CHECK SOLUTION calls subroutine ERR to compute and print error checks.

14 CALL INPUT1 calls subroutine INPUT1 which reads basis headings from logical unit 49, which were produced by the restart punchout subroutine. (See page 32 for use of this control in restart procedures.)

Read Restart Punchouts

15 EXIT PROGRAM calls subroutine EXIT which will go to EXIT* in the COOP MONITOR and terminate the program normally.

16 SET CONDITION 6 AND OUTPUT calls QOT -- the output control routine with condition 6 set. The value of K(6,1) and K(6,2) will determine the type of output given alternate and standard. Since the standard output controls assign the value zero to K(6,J) the 16 control card will produce nothing unless these values have been changed by control card 03.

Special Output

17 WRITE END OF FILE ON OFF-LINE OUTPUT

18 PAUSE 302

19 PAUSE 303

20 PAUSE 304

} executes the indicated FORTRAN PAUSE operation.

21 READ AND PRINT MESSAGE reads the next card and prints it in columns 1-72 on standard output. If alternate output has been indicated, the message will also be written on the alternate output tape.

22 PUNCH RESTART TAPE outputs on logical unit 48 the current basis headings. The FORMAT of this output is given under restart procedures.

SENSE SWITCH CONTROLS

Switch 1. When switch 1 is set, a new control card from the standard input unit is read at the end of each cycle. This switch is used in conjunction with control cards to interrupt the program to get solution checks and printouts, to change output controls, and so forth.

Switch 2. When switch 2 is set, a short output occurs every cycle.

Switch 3. When CDM3 is solving separable problems, it normally curtains the special vectors during pricing. That is, it prices them first, and if one or more has a sufficiently negative delta-jay, the regular vectors will not be priced. Thus, even though one of the regular vectors might have a smaller delta-jay than the most eligible special vector, it would not be entered into the basis. Setting switch 3 suppresses this curtaining and the vector with the smallest delta-jay from among all vectors, special and regular, enters the basis.

Switch 4. Not used.

Switch 5. For alternate output, this switch is set before calling CDM3.

RESTART PROCEDURES

Stop/restart is effected in CDM3 by the basis headings which are kept in COMMON. Subroutine PUNCH, which may be called at the end of any cycle, produces these headings on logical unit 48, in a form suitable for

listing. This same information is used for restarting with 49 as logical unit number. The format is as follows:

Basis Headings Format

The identification section contains the words basis headings followed by the one word ID, current iteration number, and current condition number. The basis headings are eight characters each, six headings per line.

Getting Off The Machine

If the operator must leave the machine in the middle of a run, the following control cards may be used:

```
22 WRITE RESTART TAPE
17 END FILE ON OUTPUT TAPE
15 EXIT PROGRAM
```

Upon setting switch 1, the program will read the 22 control at the end of the current iteration, output the restart information, write an end-of-file on the output unit and exit.

Save restart information, standard output, and alternate output.

For a somewhat more elaborate procedure, consider the following control cards.

```
21 MESSAGE
I AM BEING SUMMARILY THROWN OFF
03 READ OUTPUT CONTROLS
000002000000
16 CALL QOT(6)
13 CHECK SOLUTION
17 END FILE ON OUTPUT TAPE
22 WRITE RESTART TAPE
15 EXIT PROGRAM
```

In addition to the restart tape, there will be documentary proof of having been removed from the machine, short and long output of the solution at the end of the last iteration, and a solution check.

Restart punchouts may also be obtained at a predetermined frequency by use of the 01 control card. This control card can be used at the start of the problem to set KM(8) so that restart punchouts are given every n iterations. Example: n = 50

```
12 READY FOR NEW PROBLEM
01 READ FIXED POINT PARAMETERS
    0   1   2   2   50  50
05 READ DATA
06 INITIATE
07 COMPUTE
15 EXIT PROGRAM
```

The standard procedure is called for except a solution check and restart tape are called for every 50 iterations.

Restarting

Machine setup is exactly as in standard operations. Restart information must be placed on logical unit 49 and following control cards are used.

```
12 READY FOR NEW PROBLEM
05 READ DATA
14 READ RESTART TAPE
08 FORM INVERSE
07 COMPUTE
15 EXIT PROGRAM
```

Buffer Out Common

Instead of using the restart procedure it is possible to buffer out common by using a control card with 09 in columns 1-2.

```
09 BUFFER OUT COMMON
```

For example, if the program is running near the time limit, set sense 1, which will cause the reading of a new control card. Read an 09 to buffer out common on logical unit 1, then a 15 control card to exit program.

Buffer In Common

To buffer in common, from logical unit 1, a control card with 10 in columns 1-2 is used. The advantage of buffering in common is that the data deck need not be read in and converted along with the basis headings, and no inversion is required.

The following control cards could be used for a restart:

```
10 BUFFER IN COMMON
08 FORM INVERSE
07 COMPUTE
15 EXIT PROGRAM
```

Read In Basis

The basis or part of the basis can be read in together with regular input data. The first card must be the control card BASIS XX. The basis headings are identified by row and column number. Card format: blanks in columns 1-6, column name in 7-12, and row name in columns 13-18.

NOTE: The columns and row names use an A-format; therefore they must appear in the same format as in the ROW ID and MATRIX data. If the formats are not identical, an error message will be printed out.

Read In New Right Hand Sides

After an optimum solution is reached, new right hand side elements can be read in by using a control with 11 in columns 1-2.

The input deck to solve a problem and find a new solution with new right hand sides could consist of the following control cards:

```
12 ZERO COMMON
05 READ DATA
06 NEW PROBLEM
07 COMPUTE
11 READ NEW RIGHT HAND SIDE
08 FORM INVERSE
07 COMPUTE
15 EXIT PROGRAM
```

New Right-Hand Sides

New right hand side elements can be read in after an optimal solution has been reached by using a control card with 11 in columns 1-2. The program will install slacks if needed, and print out:

NEGATIVE R.H.S. FROM INVERTING. XXXE YY

where XXXEYY is the value of the right hand side.

The input deck control cards could be as follows:

12 ZERO COMMON
05 READ DATA
06 NEW PROBLEM
07 COMPUTE
11 READ NEW RIGHT HAND SIDE
08 FORM INVERSE
07 COMPUTE
15 EXIT PROGRAM

COST PARAMETER-IZING

Cost parameters may be used to change the effective cost on vectors by adding a ϕCf_j to the original cost on that vectors, where ϕ is the value of the cost parameter and Cf_j is an element j of the cost function row to be added to the corresponding C_j of the original cost row.

When an optimum is reached and cost parameters desired, the following set of control cards could be used:

12 ZERO COMMON
05 READ DATA
06 NEW PROBLEM
07 COMPUTE
(Optimum reached)
04 PARAMETERIZE COST
XXX·····XXYY
15 EXIT PROGRAM

Where XXX····XX in columns 1-20 is the maximum value of ϕ to be used. YY in columns 21-22 is zero or blank for the short form of output or a positive integer for the long form of output.

If the value of ϕ is zero or blank then ϕ will be set to 1.0E+10.

The output will consist of the value of the parameter, the total cost of the newly created cost row, the vector which has just entered the basis, as well as the regular short or long output.

This information will be output after each new change in the basis, until no further change will occur in the basis for ϕ increased to the maximum ϕ. Output will then be as follows:

MAXIMUM PHI = XXXX.XXXXXX REACHED.
NO NEW CHANGE IN BASIS UP TO THIS VALUE

Local Optimum

When using cost parameters on a non-linear (separable) problem and a local optimum is reached, the program will output:

LOCAL OPTIMUM DETECTED

then it will find the new optimum and begin parameterizing.

Chapter 5

PROGRAM ORGANIZATION

INTRODUCTION

CDM3 is composed of many independent subroutines. Most of the communication between these subroutines is by means of data in COMMON. Because of this, modification of the algorithm and its controls is made easier. Operation of the system is governed primarily by control cards. These cards are read and interpreted by subroutine INP. New controls may be introduced to the system by adding the appropriate subroutines and making minor modifications of INP.

The over-all flow within CDM3 is:

1. The main program sets up for the standard procedure.
2. The main program then calls INP which retains control until an " 07" control card is read. INP then returns control to the main program.
3. The main program calls subroutine CYC which executes one simplex step.
4. The main program tests the condition number as a result of the last step. If the condition is 1, 2, or 6, it will normally return to step 3. For the other condition numbers, it will perform certain special operations and return to step 2.

Figure 1 shows the flow within the main program. Figure 2 shows the flow within subroutine CYC. Figure 3 shows the control functions of subroutine INP. The remainder of this chapter shows the memory usage by the program.

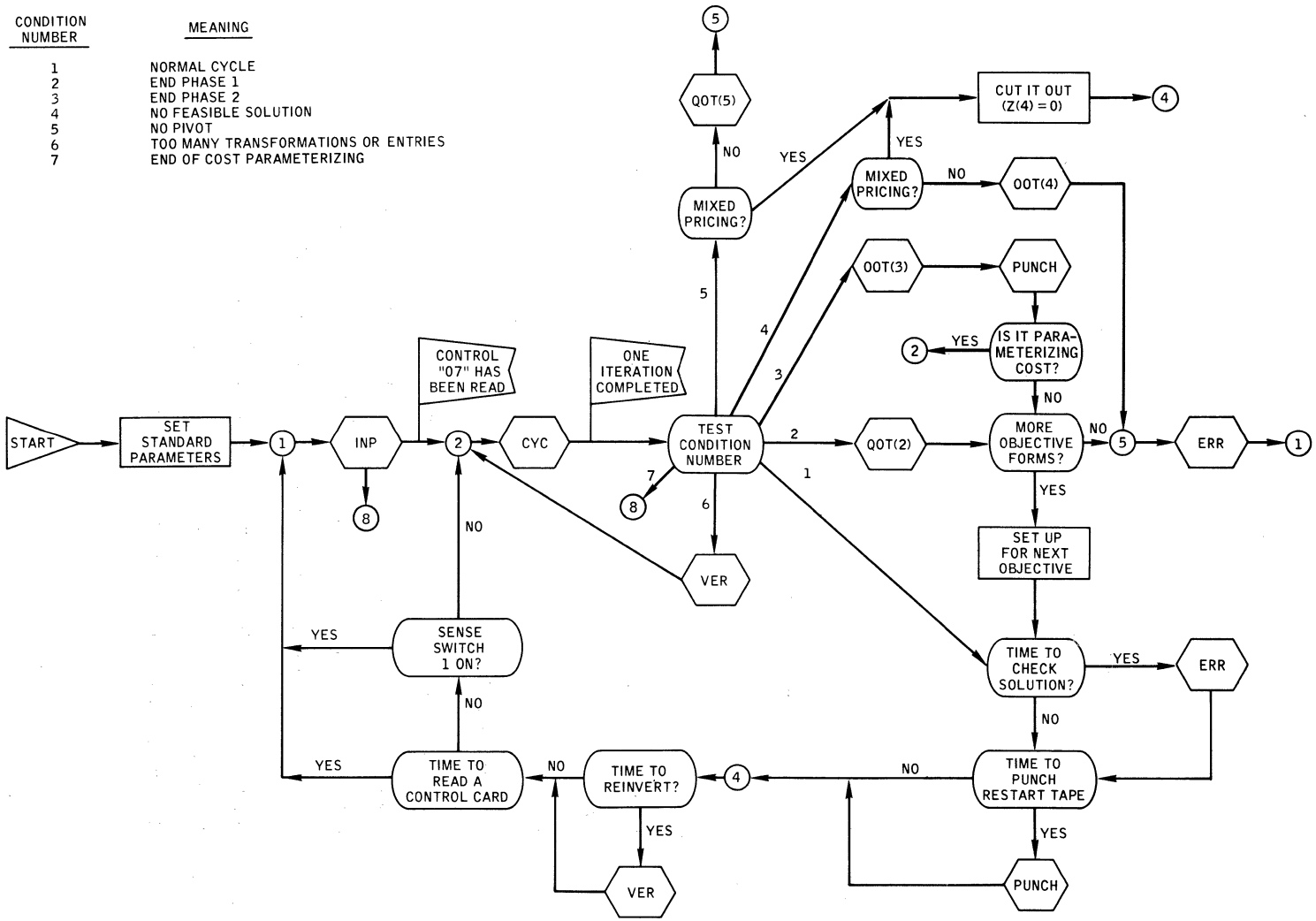

Figure 1. Main Program

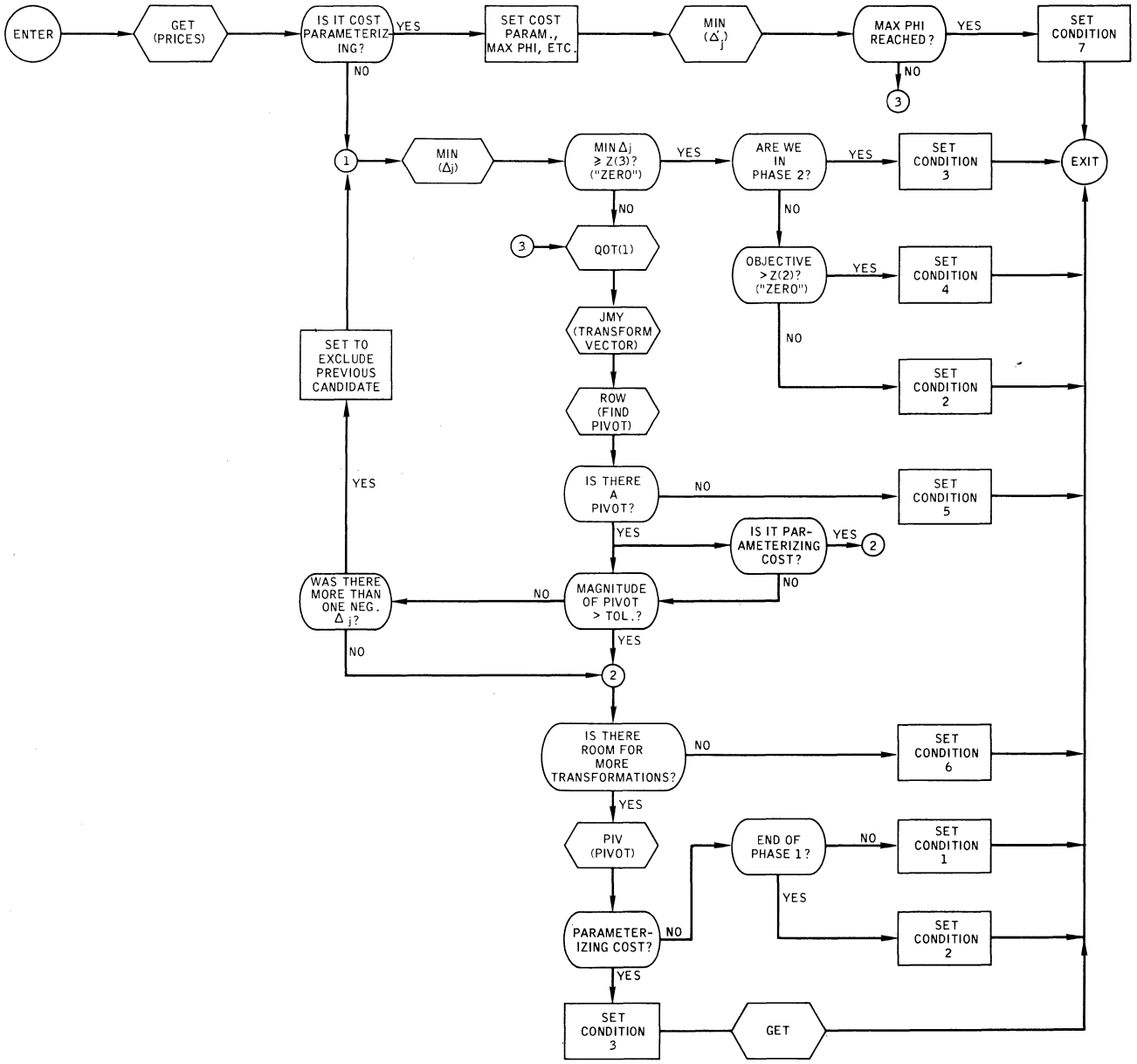

Figure 2. Subroutine CYC

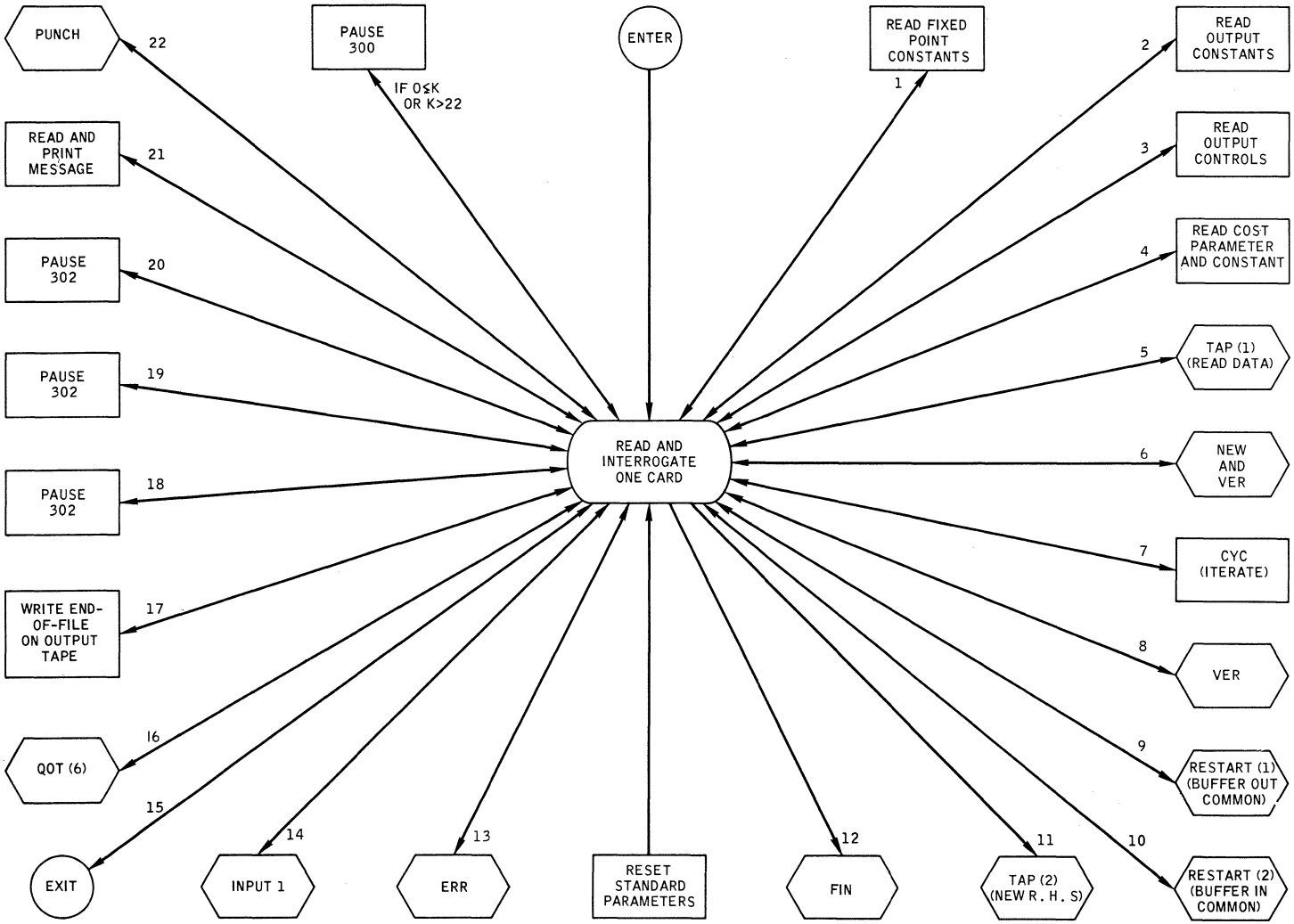

Figure 3. Subroutine INP

NAME	DIMENSION	DESCRIPTION	COMMON ALLOCATION
N	---	Number of columns.	
M	---	Number of rows.	
MA	---	Number of special vector sets.	
MB	---	(Not used)	
MC	---	Row number of current objective.	
MD	---	Row number of first non-negative variable.	
ME	---	Current number of transformations.	
MF	---	Row number of first constraint.	
KB	1000	Entry j is the row number of the pivot element of column j (for basic columns); zero for non-basic columns.	
KL	1000	Entry j is the index (in the matrix array A) of the first non-zero element of column j.	
KM	24	Fixed point miscellany.	
KN	1000	List of column names.	
KP	8,2	Output controls.	
IE	2500	Row numbers of transformation entries. These numbers are packed six per word.	
IK	1000	Row numbers of matrix entries; packed six per word.	
IRN	400	List of row names.	
IP	300	Row numbers of transformation pivots.	
LE	300	Contains the index of the first entry (in the array E) for each transformation.	

NAME	DIMENSION	DESCRIPTION
NSPV	70	Contains the column number of the first column in each set of special columns.
JH	400	"Basis heading" -- Entry i is the column number of the column in the basis on row i.
A	400	Original matrix entries.
B	400	Right-hand side entries.
P	400	Entry i is the price on constraint i.
T	10	Temporary storage.
X	400	Entry i is the value of the variable in the basis on row i.
Y	400	Holds the transformed representation of a column. Also holds π's for cost function row temporarily when parameterizing.
Z	16	Floating point miscellany.
E	10000	Transformation entries.

APPENDIX I

EXAMPLES

This appendix presents four sample runs using CDM3. The first three runs were made with the input deck shown on page 44. This input deck contains the data for two problems -- the sphere problem presented in Chapter 1, Section 5, and the diet problem presented in Chapter 1, Section 3. The deck for the fourth run, using cost parameters in the diet problem, is on page 46.

The first run illustrates the standard procedure for standard input and output. It solves both problems on the input tape.

The second run uses non-standard fixed point parameters and output controls. Re-inversion occurs every 10 iterations and solution checks occur every five iterations. Short output is given on standard output for conditions 2, 3, 4, and 5 (only conditions 2 and 3 occur in the run as shown). Alternate output has no output for condition 1; short and long output for condition 2; and short, long, and horizontal for conditions 3, 4, and 5.

The third run illustrates the restart procedure. Part 1 of this run shows one procedure for getting off the machine -- the operator sets switch 1 during iteration 11; a new control card is read at that point, and the procedure for getting off is initiated. Part 2 of the run shows the procedure necessary to restart the problem.

The fourth run illustrates the use of cost parameters with standard input and output.

For each run, the following are shown:

(1) Control Cards

(2) Standard Output

(3) Alternate Output

(4) Output on LTN 48

DATA DECK (REGULAR PROCEDURE)

```
SPHERE PROBLEM - OR - FINDING THE NORTH POLE                    2
ROW ID
             COST       00
             SPHERE     02
             SUMR       03
             XEQU       04
             XSQEQU     05
             SUMS       06
             YEQU       07
             YSQEQU     08
             SUMT       09
             ZEQU       10
             ZSQEQU     11

EOR
MATRIX
         X   XEQU    -1.0
         Y   YEQU    -1.0
         Z   COST    -1.0
         Z   ZEQU    -1.0
         U   SPHERE   1.0
         U   XSQEQU -1.0
         V   SPHERE   1.0
         V   YSQEQU -1.0
         W   SPHERE   1.0
         W   ZSQEQU -1.0

EOR
SEPARABLE
         R000SUMR     1.0
         R010SUMR     1.0
         R010XEQU     0.2
         R010XSQEQU   0.04
         R020SUMR     1.00
         R020XEQU     0.40
         R020XSQEQU   0.16
         R030SUMR     1.00
         R030XEQU     0.60
         R030XSQEQU   0.36
         R040SUMR     1.00
         R040XEQU     0.80
         R040XSQEQU   0.64
         R050SUMR     1.00
         R050XEQU     1.00
         R050XSQEQU   1.00
         S000SUMS     1.00
         S010SUMS     1.00
         S010YEQU     0.20
         S010YSQEQU   0.04
         S020SUMS     1.00
         S020YEQU     0.40
         S020YSQEQU   0.16
         S030SUMS     1.00
         S030YEQU     0.60
         S030YSQEQU   0.36
         S040SUMS     1.00
         S040YEQU     0.80
```

DATA DECK (CONTINUED)

```
        S040YSQEQU 0.64
        S050SUMS   1.00
        S050YEQU   1.00
        S050YSQEQU 1.00
        T000SUMT   1.00
        T010SUMT   1.00
        T010ZEQU   0.20
        T010ZSQEQU 0.04
        T020SUMT   1.00
        T020ZEQU   0.40
        T020ZSQEQU 0.16
        T030SUMT   1.00
        T030ZEQU   0.60
        T030ZSQEQU 0.36
        T040SUMT   1.00
        T040ZEQU   0.80
        T040ZSQEQU 0.64
        T050SUMT   1.00
        T050ZEQU   1.00
        T050ZSQEQU 1.00
EOR
FIRST B
             SPHERE 1.00
             SUMR   1.00
             SUMS   1.00
             SUMT   1.00
EOR
FOF
FATMAN A BREAKFAST FOOD PROBLEM
ROW ID
             COST     01
             CALRIE   03
            +SODIUM   04
            -PRTEIN   05
EOR
MATRIX
      CRSPIECOST     4.0
      CRSPIECALRIE 150.0
      CRSPIESODIUM   0.1
      CRSPIEPRTEIN   2.0
      CRUNCHCOST     7.0
      CRUNCHCALRIE 140.0
      CRUNCHSODIUM   0.1
      CRUNCHPRTEIN   4.0
      CRKLESCOST     8.0
      CRKLESCALRIE 170.0
      CRKLESSODIUM   0.3
      CRKLESPRTEIN   5.0
      CHRTLECOST     6.0
      CHRTLECALRIE 160.0
      CHRTLESODIUM   0.3
      CHRTLEPRTEIN   3.0
EOR
FIRST B
             CALRIE 150.0
             SODIUM   0.2
             PRTEIN   3.0
EOR
FOF
```

DATA DECK (FOR PARAMETERIZING COST IN DIET PROBLEM)

```
FATMAN  A BREAKFAST FOOD PROBLEM                                    1
ROW ID
            CFCN         0
            COST        01
            CALRIE      03
           +SODIUM      04
           -PRTEIN      05
EOR
MATRIX
      CRSPIECFCN        1.0
      CRSPIECOST        4.0
      CRSPIECALRIE    150.0
      CRSPIESODIUM      0.1
      CRSPIEPRTEIN      2.0
      CRUNCHCOST        7.0
      CRUNCHCALRIE    140.0
      CRUNCHSODIUM      0.1
      CRUNCHPRTEIN      4.0
      CRKLESCFCN       -1.0
      CRKLESCOST        8.0
      CRKLESCALRIE    170.0
      CRKLESSODIUM      0.3
      CRKLESPRTEIN      5.0
      CHRTLECOST        6.0
      CHRTLECALRIE    160.0
      CHRTLESODIUM      0.3
      CHRTLEPRTEIN      3.0
EOR
FIRST B
            CALRIE    150.0
            SODIUM      0.2
            PRTEIN      3.0
```

RUN #1 CONTROL CARDS

```
12  READY FOR NEW PROBLEM
05  READ DATA
06  INITIATE NEW PROBELM
07  COMPUTE
12  READY FOR NEW PROBLEM
05  READ DATA
06  INITIATE NEW PROBLEM
07  COMPUTE
15  EXIT PROGRAM
```

CLEARED

RUN #1 OUTPUT ON STANDARD UNIT

```
M  12, N  24,ENTRIES    58,SETS  3

REINVERTING AFTER     0TH ITERATION.      0TRANSFORMATIONS WITH    0 ENTRIES.

IDENT   COND  FORM   ITER      OBJECTIVE    TRN ENTRIES
SPHERE     2     3     11          -.000000         69


IDENT   COND  FORM   ITER      OBJECTIVE    TRN ENTRIES
SPHERE     3     2     21        -1.000000        156
     J          X(J)                 I          Y(I)              P(I)

  Cost          1.000000            2           .200000          1.000000
  Phase1         .000000            3           .000000           .000000
  Z             1.000000            4           .200000           .000000
  X              .000000            5         -1.800000           .000000
  U              .000000            6          -.360000          -.000000
  R000          1.000000            7          9.000000           .000000
  S000          1.000000            8           .000000           .000000
  V              .000000            9           .000000          -.000000
  S010           .000000           10           .000000           .000000
  T050          1.000000           11          1.000000          1.000000
  W             1.000000           12           .360000         -1.000000
  R010           .000000           13         -9.000000           .000000

MAX ERROR ON ROW   10, -.43656E-09.  SUM,  .54879E-09

MAX ERROR ON COL   10, -.84401E-09.  SUM,  .19360E-08

CLEARED

M   5, N   6,ENTRIES    18,SETS  0

REINVERTING AFTER     0TH ITERATION.      0 TRANSFORMATIONS WITH    0 ENTRIES.

IDENT   COND  FORM   ITER      OBJECTIVE    TRN ENTRIES
FATMAN     2     3     2           -.000000         15

IDENT   COND  FORM   ITER      OBJECTIVE    TRN ENTRIES
FATMAN     3     2     2           5.268293         15
     J          X(J)                 1          Y(I)              P(I)

Cost         -5.268293             2          -.009756          1.000000
Phase1         .000000             3          1.000000           .000000
CRSPIE         .585366             4           .012195          -.009756
S+SODIU        .031707             5           .000244           .000000
CRKLES         .365854             6          -.004878         -1.268293

MAX ERROR ON ROW    3, -.18626E-08. SUM,  .18626E-08
MAX ERROR ON COL    3,  .00000E+00. SUM,  .00000E+00
```

NO OUTPUT ON ALTERNATE UNIT

NO OUTPUT ON LOGICAL TAPE 48

RUN #2 CONTROL CARDS

```
12  READY FOR NEW PROBLEM
01  READ FIXED POINT PARMS  3 THROUGH 9
   10    1    2    2    5
03  READ OUTPUT CONTROLS  6 FOR OFFLINE,6 FOR ONLINE
12444 01111
05  READ DATA
06  INITIATE NEW PROBLEM
07  COMPUTE
17  ENDFILE ON OFFLINE OUTPUT TAPE
18  PAUSE  302
```

RUN #2 STANDARD OUTPUT

```
CLEARED

M  12, N  24,ENTRIES    58,SETS  3

REINVERTING AFTER    0TH ITERATION.    0 TRANSFORMATIONS WITH    0 ENTRIES.

MAX ERROR ON ROW   3,  .00000E+00. SUM,  .00000E+00

MAX ERROR ON COL   6,  .27285E-11. SUM,  .54570E-11

MAX ERROR ON ROW   5,  .29104E-10. SUM,  .29104E-10

MAX ERROR ON COL   6, -.34925E-09. SUM,  .76216E-09

REINVERTING AFTER   10TH ITERATION.   20 TRANSFORMATIONS WITH   64 ENTRIES.

IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
SPHERE    2     3    11        -.000000        34

MAX ERROR ON ROW   3, -.29104E-10. SUM,  .13097E-09

MAX ERROR ON COL  10,  .83819E-10. SUM,  .18219E-09

MAX ERROR ON ROW  10,  .43656E-09. SUM,  .71850E-09

MAX ERROR ON COL  11,  .14370E-09. SUM,  .24647E-09
REINVERTING AFTER   20TH ITERATION.   20 TRANSFORMATIONS WITH  108 ENTRIES.

IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
SPHERE    3     2    21       -1.000000        38
MAX ERROR ON ROW  10,  .29104E-10. SUM,  .58208E-10

MAX ERROR ON COL   5, -.45475E-12. SUM,  .90949E-12
```

RUN #2 ALTERNATE OUTPUT

```
SPHERE PROBLEM - OR - FINDING THE NORTH POLE                              1
M  12, N  24,ENTRIES    58,SETS  3
REINVERTING AFTER    0TH ITERATION.    0 TRANSFORMATIONS WITH    0 ENTRIES.
IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
SPHERE    1     3     0        1.000000        17
IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
SPHERE    1     3     1        1.000000        21
IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
SPHERE    1     3     2        1.000000        25
IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
SPHERE    1     3     3        1.000000        29
IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
SPHERE    1     3     4        1.000000        34
MAX ERROR ON ROW   3,  .00000E+00. SUM,  .00000E+00
MAX ERROR ON COL   6,  .27285E-11. SUM,  .54570E-11
IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
SPHERE    1     3     5        1.000000        39
IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
SPHERE    1     3     6        1.000000        44
IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
SPHERE    1     3     7         .960000        49
IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
SPHERE    1     3     8         .840000        54
IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
SPHERE    1     3     9         .640000        59
MAX ERROR ON ROW   5,  .29104E-10. SUM,  .29104E-10
MAX ERROR ON COL   6, -.34925E-09. SUM,  .76216E-09
REINVERTING AFTER   10TH ITERATION.   20 TRANSFORMATIONS WITH   64 ENTRIES.
IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
SPHERE    1     3    10         .360000        29

IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
SPHERE    2     3    11        -.000000        34
     J         X(J)              I         Y(I)              P(I)


Phase 1          .000000          3          .360000          1.000000
ARTFCIAL         .000000          4         -.360000         -1.000000
  R050          1.000000          5         1.000000           .640000
  X             1.000000          6          .200000           .000000
  U             1.000000          7          .360000         -1.000000
  S000          1.000000          8          .000000           .000000
  S010           .000000          9          .000000           .200000
  V              .000000         10          .000000         -1.000000
  T000          1.000000         11          .000000           .000000
  T010           .000000         12          .000000           .200000
  W              .000000         13          .000000         -1.000000
```

RUN #2 ALTERNATE OUTPUT (CONTINUED)

```
 IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
 SPHERE    1     2    11       -.000000        34
 IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
 SPHERE    1     2    12       -.000000        40
 IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
 SPHERE    1     2    13       -.200000        48
 IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
 SPHERE    1     2    14       -.400000        56
 MAX ERROR ON ROW   3, -.29104E-10. SUM,  .13097E-09
 MAX ERROR ON COL  10,  .83819E-10. SUM,  .18219E-09
 IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
 SPHERE    1     2    15       -.600000        64
 IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
 SPHERE    1     2    16       -.600000        72
 IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
 SPHERE    1     2    17       -.800000        81
 IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
 SPHERE    1     2    18       -.800000        90
 IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
 SPHERE    1     2    19       -.911111        99
 MAX ERROR ON ROW  10,  .43656E-09. SUM,  .71850E-09
 MAX ERROR ON COL  11,  .14370E-09. SUM,  .24647E-09
 REINVERTING AFTER   20TH ITERATION.   20 TRANSFORMATIONS WITH  108 ENTRIES.
 IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
 SPHERE    1     2    20       -.977778        30

 IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
 SPHERE    3     2    21      -1.000000        38
      J         X(J)              I         Y(I)              P(I)

   Cost          1.000000          2          .111111          1.000000
   Phase 1        .000000          3          .000000           .000000
   U              .000000          4         -.200000           .555556
   R010           .000000          5        -5.000000          -.000000
   R000          1.000000          6         5.000000          -.111111
   V              .000000          7          .000000           .555556
   S000          1.000000          8          .000000           .000000
   S010           .000000          9          .000000          -.111111
   W             1.000000         10          .200000           .555556
   T040           .000000         11         -.555556           .444444
   Z             1.000000         12          .111111         -1.000000
   T050          1.000000         13          .555556           .555556
0     J         X(J)              DELTA(J)      POSITION
```

RUN #2 ALTERNATE OUTPUT (CONTINUED)

X	.000000	.111111	0
Y	.000000	.111111	0
Z	1.000000	.000000	11
U	.000000	.000000	3
V	.000000	.000000	6
W	1.000000	.000000	9
R000	1.000000	-.000000	5
R010	.000000	.000000	4
R020	.000000	.044444	0
R030	.000000	.133333	0
R040	.000000	.266667	0
R050	.000000	.444444	0
S000	1.000000	.000000	7
S010	.000000	.000000	8
S020	.000000	.044444	0
S030	.000000	.133333	0
S040	.000000	.266667	0
S050	.000000	.444444	0
T000	.000000	.444444	0
T010	.000000	.266667	0
T020	.000000	.133333	0
T030	.000000	.044444	0
T040	.000000	.000000	10
T050	1.000000	.000000	12

MAX ERROR ON ROW 10, .29104E-10. SUM, .58208E-10
MAX ERROR ON COL 5, -.45475E-12. SUM, .90949E-12

NO OUTPUT ON LOGICAL TAPE 48

RUN #3 PART I CONTROL CARDS

```
12  READY FOR NEW PROBLEM
05  READ DATA
06  INITIATE NEW PROBLEM
07  COMPUTE
21  READ AND PRINT MESSAGE
I AM GETTING OFF THE MACHINE FOR LATER RESTART
03  READ OUTPUT CONTROLS
000002
16  SPECIAL OUTPUT  (  QOT(6)   )
13  CHECK SOLUTION
22  PUNCH RESTART TAPE
15  EXIT PROGRAM
```

RUN #3 PART I STANDARD OUTPUT

```
CLEARED

M  12, N  24,ENTRIES    58,SETS  3

REINVERTING AFTER    0TH ITERATION.    0 TRANSFORMATIONS WITH    0 ENTRIES.

IDENT  COND  FORM   ITER    OBJECTIVE   TRN ENTRIES
SPHERE    2     3    11        -.000000        69
I AM GETTING OFF THE MACHINE FOR LATER RESTART

MAX ERROR ON ROW   2,  .58208E-10. SUM,  .14552E-09

MAX ERROR ON COL   2,  .00000E+00. SUM,  .00000E+00
```

RUN #3 PART I ALTERNATE OUTPUT

IDENT	COND	FORM	ITER	OBJECTIVE	TRN ENTRIES
SPHERE	6	2.	11	-.000000	69

J	X(J)	I	Y(I)	P(I)
Phase 1	.000000	3	.360000	1.000000
ARTFCIAL	.000000	4	-.360000	-1.000000
X	1.000000	5	.200000	.640000
U	1.000000	6	.360000	-.000000
R050	1.000000	7	1.000000	-1.000000
S000	1.000000	8	.000000	.000000
V	.000000	9	.000000	.200000
S010	.000000	10	.000000	-1.000000
T000	1.000000	11	.000000	.000000
W	.000000	12	.000000	.200000
T010	.000000	13	.000000	-1.000000

MAX ERROR ON ROW 2, .58208E-10. SUM, .14552E-09
MAX ERROR ON COL 2, .00000E+00. SUM, .00000E+00

RUN #3 PART I OUTPUT ON LTN 48

basis headings sphere p 11 6

artfcialartfcialartfcial X U R050
S000 V S010 T000 W T010

RUN #3 PART 2 CONTROL CARDS

```
21  READ AND PRINT MESSAGE
THIS IS A RESTART OF THE SPHERE PROBLEM
12  READY FOR NEW PROBLEM
05  READ DATA
14  CALL INPUT1   (RESTART READ)
08  FORM INVERSE
07  COMPUTE
18  PAUSE  302
15  EXIT PROGRAM
```

RUN #3 PART 2 STANDARD OUTPUT

```
THIS IS A RESTART OF THE SPHERE PROBLEM

CLEARED


M  12, N   24,ENTRIES    58,SETS  3

REINVERTING AFTER     0TH ITERATION.     0 TRANSFORMATIONS WITH     0 ENTRIES.

IDENT   COND  FORM    ITER    OBJECTIVE    TRN ENTRIES
SPHERE     2     3       1        -.000000          34


IDENT   COND  FORM    ITER    OBJECTIVE    TRN ENTRIES
SPHERE     3     1      11       -1.000000         132
     J          X(J)                I          Y(I)                P(I)

  Cost          1.000000            2           .555556           1.000000
  Phase 1        .000000            3           .000000            .000000
  Y              .000000            4           .000000            .555556
  T040           .000000            5         -2.777778            .000000
  X              .000000            6         -5.000000            .000000
  R000          1.000000            7         25.000000           -.000000
  S000          1.000000            8           .000000            .000000
  S010           .000000            9          -.000000            .000000
  Z             1.000000           10           .555556            .000000
  T050          1.000000           11          2.777778            .444444
  R010           .000000           12        -25.000000          -1.000000
  W             1.000000           13          1.000000            .555556


  MAX ERROR ON ROW   10,  .43656E-09. SUM,    .62573E-09

  MAX ERROR ON COL    6,  .36380E-09. SUM,    .10262E-08
```

NO ALTERNATE OUTPUT

NO OUTPUT ON LTN 48

RUN #4 CONTROL CARDS

```
12  READY FOR NEW PROBLEM
05  READ DATA
06  INITIATE NEW PROBLEM
07  COMPUTE
04  READ COST PARAMETERS
^^^^^^8.0 ^^^^^^^^^^^^2              (^ = spaces)
15  EXIT PROGRAM
```

RUN #4 OUTPUT

```
M   6, N 6, ENTRIES      20, SETS  0

REINVERTING AFTER     0TH ITERATION.     0 TRANSFORMATIONS WITH     0 ENTRIES.

IDENT  COND  FORM    ITER    OBJECTIVE    TRN ENTRIES
FATMAN    2     3       2        -.000000          17


IDENT  COND  FORM    ITER    OBJECTIVE    TRN ENTRIES
FATMAN    3     2       2        5.268293          17

     J         X(J)              I         Y(I)              P(I)

COST           -5.268293           2        -.009756          1.000000
PHASE 1          .000000           3        1.000000           .000000
CRSPIE           .585366           4         .012195          -.009756
S+SODIUM         .031707           5         .000244           .000000
CRKLES           .365854           6        -.004878         -1.268293


MAX ERROR ON ROW   4,1.86265E-009.  SUM,1.86265E-009

MAX ERROR ON COL   4, .00000E 000.  SUM, .00000E 000

IDENT  COND  FORM    ITER    OBJECTIVE    TRN ENTRIES
FATMAN    3     2       3        5.857143          22


      PARAMETER          COST       OUT        IN
       1.625000        5.625000  S+SODIUM  S-PRTEIN

     J         X(J)              I         Y(I)              P(I)

PARAM            .142857           1       11.428571           .000000
COST           -5.857143           2      -18.571429          1.000000
PHASE 1          .000000           3        -.000000           .000000
CRSPIE           .392857           4       -6.071429          -.014286
S-PRTEIN         .464286           5       14.642857         -18.571429
CRKLES           .535714           6        5.357143          -.000000
```

RUN #4

IDENT	COND	FORM	ITER	OBJECTIVE	TRN ENTRIES
FATMAN	3	2	4	7.240000	27

PARAMETER	COST	OUT	IN
3.666667	5.333333	CRSPIE	CRUNCH

J	X(J)	I	Y(I)	P(I)
PARAM	.520000	1	.960000	.000000
COST	-7.240000	2	-3.520000	1.000000
PHASE 1	.000000	3	-.000000	.000000
CRUNCH	.440000	4	1.120000	-.052000
S-PRTEIN	1.360000	5	2.280000	2.800000
CRKLES	.520000	6	-.040000	.000000

MAXIMUM PHI= 8.000000 REACHED.
NO NEW CHANGE IN BASIS UP TO THIS VALUE

NO OUTPUT ON ALTERNATE UNIT

NO OUTPUT ON LTN 48

APPENDIX II

COMPILATION NOTES

SUBROUTINE DESCRIPTIONS

There are 30 subroutines other than standard library subroutines which make up CDM3. The following 23 of these subroutines are entirely FORTRAN.

AAA -- Main program.

CHECK (IDUM) -- Tests columns for singletons.

CYC (K) -- Executes one cycle.

DEL (J,D) -- Prices out one column.

DEL1 (J) -- Finds next Δj and tests against last best.

ERR -- Computes and prints solution checks.

GET -- Computes negative of shadow prices.

HOT (KA) -- Horizontal output routine.

INP (L) -- Reads and interrogates control cards.

INPUT 1 -- Reads restart tape.

JMY -- Transforms a column.

LOT -- Long output routine.

MIN -- Finds minimum Δj.

NEW -- Sets up initial basis headings.

PIV (IR) -- Performs pivot step.

PUNCH -- Writes basis headings on LTN 48.

QOT (K) -- Output control routine.

RESTART (L) -- Buffers common in and out.

ROW (IR) -- Chooses column to leave basis.

SOT (LK) -- Short output routine.

TAP (KK) -- Reads data.

NTEST (IROW, NROW, IFLG) -- Tests row names of data entries.

VER -- Inversion routine.

WOT -- Message writer.

There are seven subroutines written entirely in CODAP.

FIN -- Clears common.

IPAKRK (KI, II) -- Packs row numbers into array IK, 4 per word.

IPAKRE (KI, II) -- Packs row numbers into array IE, 4 per word.

IUNPKK (KI) -- Unpacks row number from IK.

IUNPKE (KI) -- Unpacks row number from IE.

EXIT -- Gives a normal exit.

SLACKS -- Set slacks if any.

APPENDIX III

PROGRAM STOPS

In addition to the stops of the input/output routines within the FORTRAN system, CDM3 has the following stops. Note that all stops in CDM3 are PAUSE stops.

PAUSE TAG		EXPLANATION
No tag	--	The problem is too large or the data deck is in error. A message on standard output unit accompanies this stop. Get off.
200	--	In subroutine VER, the largest available pivot is less than Z(1). Accompanied by a message on standard output giving row number, column name, and value of pivot. Get off.
277	--	In subroutine VER, the number of transformation entries has exceeded available storage. Accompanied by a message on standard output. Get off.
300	--	In subroutine INP, an illegal control card number has been read. Press start to continue to next control card.
302 303 304	--	These are normal pauses produced by control cards. Press start to read next control card.
332	--	Subroutine WOT(K,AA,AB) has been called with an illegal value for K. K indicates the type of standard output WOT was to have made. This pause is accompanied by a standard output indicating erroneous value of K. To continue, press start.
1111	--	During restart punchouts, a negative basis heading (which is presumably impossible) has been encountered. Get off.

Publications concerning programming and programming systems for the Control Data Corporation 1604 and 1604-A Computers are:

Manuals

1604 Fortran System (Fortran-60)	#087A
PERT	#133
1604 Programming Manual	#167B
1604-A Programming Manual	#245
Fortran Auto Tester	#186A
Satellite Programming	#187
Fortran-62 Reference Manual	#506
CO-OP Monitor/Programming Guide	#508
CO-OP Monitor/Operator's Guide	#509
CODAP-1 Reference Manual	#510
CDM2 Linear Programming System	#511
Fortran 63/General Information	#514
CO-OP Monitor/Library Subroutines	#516
CDM3/Linear Programming	#526
Fortran 63/Reference Manual (Vol. 1)	#527

Programming Systems Bulletins

Conversion Guide/Fortran-60 to Fortran-62	#PSB-AS06621
CO-OP Monitor/Usage	#PSB-AE01
CO-OP Monitor/Manuals Supplement	#PSB-AE02
CO-OP Monitor/LIBEDIT Routine	#PSB-AE04

PROGRAMMING TRAINING CENTERS

3330 Hillview Ave.
Palo Alto, California

●

8100 - 34th Ave.
Minneapolis, Minnesota

●

11428 Rockville Pike
Rockville, Maryland

●

Room 223, Terminal Building
Newark Airport
Newark, New Jersey

CONTROL DATA SALES OFFICES

ALBUQUERQUE, NEW MEXICO
BEVERLY HILLS, CALIFORNIA
BIRMINGHAM, ALABAMA
BOSTON, MASSACHUSETTS
CHICAGO, ILLINOIS
CLEVELAND, OHIO
DALLAS, TEXAS
DAYTON, OHIO
DENVER, COLORADO
DETROIT, MICHIGAN
HONOLULU, HAWAII
HOUSTON, TEXAS
HUNTSVILLE, ALABAMA
ITHACA, NEW YORK
LOS ALTOS, CALIFORNIA
MINNEAPOLIS, MINNESOTA
NEWARK, NEW JERSEY
NORFOLK, VIRGINIA
ORLANDO, FLORIDA
PALO ALTO, CALIFORNIA
SAN DIEGO, CALIFORNIA
SAN FRANCISCO, CALIFORNIA
TULSA, OKLAHOMA
WASHINGTON, D.C.
WILMINGTON, DELAWARE

INTERNATIONAL OFFICES

BAD HOMBURG, GERMANY
LUCERNE, SWITZERLAND
MELBOURNE, AUSTRALIA
STOCKHOLM, SWEDEN
ZURICH, SWITZERLAND

CONTROL DATA
CORPORATION

8100 34TH AVENUE SOUTH, MINNEAPOLIS 20, MINNESOTA

PUB. NO. 526