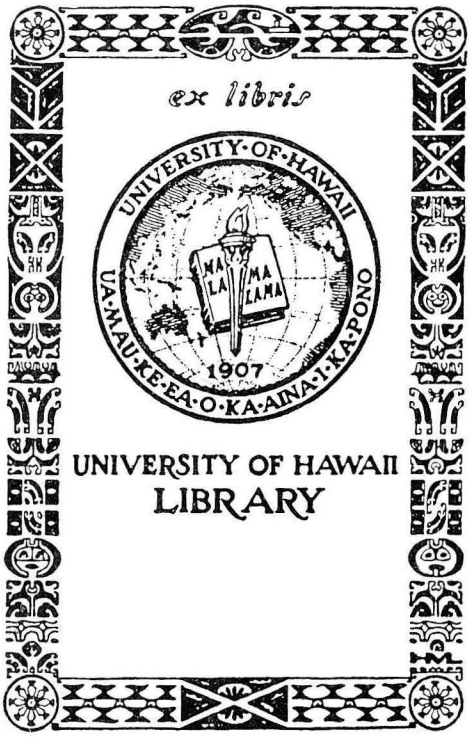
Agricultural Economics Report 81
Hawaii Agricultural Experiment Station
University of Hawaii / October 1968

LINEAR PROGRAMMING WITH ITERATIVE MODIFICATION OF THE OBJECTIVE FUNCTION AND RESTRAINTS VECTOR

ARNOLD B. LARSON
HOWARD C. HOGG


ex libris
UNIVERSITY·OF·HAWAII
NA LA MA LAMA
1907
UA·MAU·KE·EA·O·KA·AINA·I·KA·PONO
UNIVERSITY OF HAWAII
LIBRARY

LINEAR PROGRAMMING WITH ITERATIVE MODIFICATION OF THE OBJECTIVE FUNCTION AND THE RESTRAINTS VECTOR

Arnold B. Larson[1/] and Howard C. Hogg[2/]

INTRODUCTION

This manual supplements "An Iterative Linear Programming Procedure for Estimating Patterns of Agricultural Land Use," by the present authors and published in the January 1968 issue of Agricultural Economics Research (3). It is intended to accomplish three objectives. First, it presents a complete printout of the computer program used to find the equilibrium pattern of land use on land development projects in which production on project lands affects product prices. This was the problem dealt with in the paper referred to above. The makeup of the program deck, the reading in of data and parameters, and interpretation of output are explained in detail. Secondly, program modifications required to permit the solution of closely analogous problems are explicitly set forth. Finally, the power of the method to solve problems similar in some respects but requiring different convergence techniques and different criteria of optimality is discussed. We hope that by presenting these three progressively more general areas of application we will make the method we have developed more accessible to other potential users.

In addition to possible adaptations of the basic method, we show by an example how the method, in one of its adaptations, is used in a wider context to solve related economic problems. The example used is the derivation of a demand curve for water in a project area by determining the optimum pattern of land use and, by implication, the optimum level of water use on project lands at various levels of the price of water.

USE OF THE PRESENT COMPUTER PROGRAM

In this section we explain the use of the computer program we have developed for its intended purpose--to estimate the pattern of land use in a land development project when some product prices will be affected by the production on project land. The explanation will be based on the examples used in reference (3). It is suggested that users of the program first run the program using this example, to gain familiarity before proceeding to other uses.

1/ Associate Agricultural Economist, Hawaii Agricultural Experiment Station, and Associate Professor, Department of Agricultural Economics, University of Hawaii.

2/ Agricultural Economist, Natural Resources Economics Division, Economic Research Service, U. S. Department of Agriculture.

The program printout is reproduced as Appendix A. A number of comment cards are interspersed in the program to assist the reader in following the method. In addition, a glossary of the names of variables is given in Appendix B. The program consists of a main program and a Simplex subroutine. The main program prepares an initial coefficient matrix, levels vector or restraints vector, and coefficients for an objective function. Criteria of optimality are introduced in the main program, specifying that rents earned by all crops grown on a land class must be equal and that all land earning rent must be used. The activities contemplated are raising each crop on each land class, so elements in the objective function are the net revenues from each crop on each land class. There are two types of restraints; the first is the acreage in each land productivity class and the second is the demand for each product at a given level of product price.

An initial solution is obtained for each crop grown on all project land on which it can earn a rent. In this solution, quantities produced are necessarily too large and rents are necessarily too low. The Simplex subroutine is entered and an optimum pattern of activities is determined. Results from the Simplex table are then evaluated in the main program to see if they meet the additional criteria of optimality, i.e., that demand equals supply for each crop or no quantity of crop is in disposal and that no land in disposal earns rent. If a quantity of a crop is in disposal, it means that more of the crop is demanded than is supplied at the price of that crop specified for the current iteration, so price must be raised. If land of a given class is in disposal (i.e., if a portion of the land in that class is unused) and earns rent, it means that the prices of all crops grown on that land class are too high. If either criterion is not met, the objective function is modified by raising or lowering crop prices, and the Simplex subroutine is reentered. Prices are changed in such a way as to retain equality of rents earned by all crops sharing a land class. Adjustments in prices are progressively smaller, and the solution converges to a competitive equilibrium within predetermined levels of tolerance.

The problem to be solved by the computer program is embodied in Table 1, which shows linear demand and supply curves for each of the five crops that might be grown on a project, and in Table 2, which shows yields and costs of production for each crop on each of three land classes. There are 550 acres of land class 1 and 100 acres each in land classes 2 and 3.

Each application of the program requires a new set of parameter specification cards, subroutine argument cards, and data cards. The parameter specifications are read in as data, but the Simplex subroutine arguments must be inserted at statement position 0168. Arguments depend on the dimensions of the Simplex matrix and on whether the problem is one of maximizing or minimizing. There are seven arguments in the CALL SYMPLX statement. They are as follows:

1. The first argument is the number of rows in the matrix or the number of restraints imposed. In our example there are three land classes with restricted acreage plus five crops grown with restricted demand aat any level of price, so the total number of restraints is 8.

2. The second argument is the number of columns in the Simplex matrix. In our example there are 3 x 5 = 15 productive activities plus 8 slack activities, so the number of columns is 23.

Table 1. Market demand functions and supply functions for existing producers*

	Demand functions		Supply functions	
	Intercept	Slope	Intercept	Slope
Crop 1	250	-.003	- 1125.	0.25
Crop 2	300	-.05	-33330.	17.0
Crop 3	400	-.08	- 1143.	1.4
Crop 4	250	-.03	- 1500.	2.0
Crop 5	300	-.03	0 2500.	1.0

* Prices are estimated in dollars per 1,000 pounds and quantities are in 1,000-pound units. In this example the demand and supply curves are linear.

Table 2. Production costs and yields by land class*

	Production costs per acre			1,000 pounds		
	Land class 1	Land class 2	Land class 3	Land class 1	Land class 2	Land class 3
	Dollars			1,000 pounds		
Crop 1	7,000	6,500	6,000	50	45	40
Crop 2	5,000	4,500	4,000	55	50	45
Crop 3	7,000	6,500	6,000	40	30	20
Crop 4	10,000	9,500	9,000	100	90	80
Crop 5	10,500	10,000	9,500	95	90	85

* In this example, land classes 1, 2, and 3 show progressively lower yields for all crops, but this is not a requirement of the program.

4. The fourth argument is the number of slack variables, which in our example is 8.

5. The fifth argument is the number of surplus activities, which in our example is 0.

6. The sixth argument is the number of artificial activities, which in our example is 0.

7. The seventh argument is a c-value for the artificial activities. Although these are not needed in our example they are set at 999. The last three arguments relate to restrictions of the greater than or equal to and of the exact equality types. These types of restrictions will not normally be encountered in problems of the sort dealt with by the iterative method, but they are included in the versatile Simplex subroutine used.

All other special features of a particular problem are accommodated by data cards. The cards required for the sample we have used are shown in Table 3. The data are printed in the correct column to conform to the format specifications currently in the program as printed; but, of course, these can be changed at will. The explanation of each line of data is as follows:

(1) Parameter card specifying number of crops and land classes being considered, printout option to be used, and tolerance levels for acreage of a land class in disposal and for quantity of a crop in disposal.

(2)-(6) Per-acre production costs for crops one through five, respectively, grown on land productivity classes 1, 2, and 3.

(7)-(11) Per-acre yields for crops one through five, respectively, grown on land productivity classes 1, 2, and 3.

(12) Available acreage of land productivity classes 1, 2, and 3.

(13) Intercept terms for product demand functions.

(14) Slope of product demand functions.

(15) Intercept terms for existing producer supply functions.

(16) Slope of supply functions.

(17)-(31) Coefficients for the main body of the Simplex matrix. Cards (17), (22), and (27), for example, give the coefficients for land and yield associated with the production of crop one on land classes 1, 2, and 3. The numbers represent the per-acre land requirements and yields in thousands of pounds per year. The coefficients for the slack variables are automatically supplied by the SYMPLX subroutine.

(32) Levels of restraints vector for Simplex showing the available quantities of each land class. Demand restraints are computed and read in from the program.

Table 3. Data cards for example of 5 crops and 3 land classes

```
        1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
 1)     5    3    1  01  1
 2)  7000      6500      6000
 3)  5000      4500      4000
 4)  7000      6500      6000
 5) 10000      9500      9000
 6) 10500     10000      9500
 7)    50        45        40
 8)    55        50        45
 9)    40        30        20
10)   100        90        80
11)    95        90        85
12)   550       100       100
13)    25        30        40       25
14)         3         50        80        3         3
15) -1125    -33330    -1143    -1500    -2500
16)      25        17        14        20        10
17)       1                            50
18)       1                                      55
19)       1                                                40
20)       1                                                         100
21)       1                                                                  95
22)                 1                  45
23)                 1                            50
24)                 1                                      30
25)                 1                                               90
26)                 1                                                        90
27)                           1        40
28)                           1                  45
29)                           1                            20
30)                           1                                      80
31)                           1                                               85
32)   550       100       100
```

The option parameter, which is the third item punched on the first card, specifies if intermediate results are to be printed, in which case option is set at 1, or suppressed, in which case option is set at zero. Since many iterations may be required for a solution, it will probably be desirable to suppress printing of intermediate results once the program has been checked thoroughly.

The solution to the example problem is shown in Tables 4 and 5. The equilibrium pattern of land use and per-acre rents earned by each type of land are shown in Table 4. The equilibrium prices and the quantities produced by project producers and by all producers including competing producers on non-project lands are shown in Table 5. These results are all clearly labelled on the printout for the final iteration of the computer program.

The deck is assembled as follows: first the main program, then the Simplex subroutine, and then the data cards, in the order shown. In addition system control cards will be required as specified by the computing facility used.

MODIFICATIONS REQUIRED FOR CLOSELY ANALOGOUS PROBLEMS

It is very difficult to understand a computer program written by another programmer so thoroughly that the program can be adjusted to encompass problems differing even slightly from those for which the program was written. However, it was partly in the hope that this is possible that we have explained the original program in such detail.

It would appear that many problems are closely related to the one illustrated, in which the interative linear programming method should prove useful. Land classification is to some extent a fiction or an abstraction, since there are many more gradations, perhaps even a continuum of land qualities, yields, and land-related cost factors, than are usually contemplated. Nevertheless, planning and land use in most cases proceed on the assumption that relatively few classes are economically significant. Hence there might be occasion to estimate the pattern of land use and agricultural production in an entire region or national economy, while taking cognizance of the amount of land available in a number of land classes. In such cases, the dichotomy of project and non-project producers may be irrelevant or inappropriate. This case can be encompassed by simply setting supply by existing producers equal to zero, i.e., by setting the supply curve intercept equal to zero and giving the curve a vertical slope.

The prices of some products may not be affected by production on land within the project area. This calls for a very simple modification also, in which the intercept of the demand curve is set equal to average price and the slope is set equal to zero.

A similar type of modification is suggested by the situation facing agriculture in Hawaii, where much of the supply of many commodities is imported for extended periods in each year. In this case the supply curve of importers and the demand curve for local producers are approximately coincidental, except for dynamic factors of market coordination and quality differences that may exist for some commodities. Once again, with a suitable choice of intercept and slope parameters, this case appears to be adapted to analysis by the method.

Table 4. Final acreages and rents for five crop three land class problem

	Acreage			Per-acre rents		
	Land class 1	Land class 2	Land class 3	Land class 1	Land class 2	Land class 3
	Acres			Dollars		
Crop 1	467.2	100.0	6.9	514.15	262.73	11.32
Crop 2			50.0			11.32
Crop 3	42.5			514.15		
Crop 4	40.3			514.15		
Crop 5			43.0			11.32
Total available	550	100	100			

Table 5. Equilibrium product prices and quantities

	Product prices	Quantity supplied by project producers	Total quantity supplied
	Dollars/1,000 pounds	Pounds	Pounds
Crop 1	150.28	23,137,882	35,185,186
Crop 2	89.14	2,251,360	4,222,222
Crop 3	187.85	1,701,219	2,812,500
Crop 4	105.14	4,026,047	5,000,000
Crop 5	111.90	3,658,173	6,315,789

POSSIBLE USES IN OTHER PROBLEM AREAS

At the risk of being glib, one could argue that the basic method used in this computer program might be applicable to a wide range of more or less related problems. Land classes as used in this application would ordinarily be geographically interspersed with one another, but in other respects they are analogous to regions in studies of interregional competition. Economic land classes are distinguished by differing costs of production, by differing crop yields, or by both, and the same might be said for regions. Especially where the problem is to determine the regions supplying a single market, it would seem that a variant of the present method might be used in interregional competition studies.

The program can be used to estimate factor demand curves where the production function of each crop is known. Irrigation water will be used to illustrate this application, and several different methods will be considered.

Linear programming becomes a useful tool for estimating factor demand curves when more than one crop is involved. In the multiple crop case with fixed product prices a linear program can be employed which incorporates the different production functions. In a formulation of this type the objective function would consist of the sum of the net returns for each activity where there would be an activity for each intensity of irrigation for each crop, and production costs would include the cost of water. The number of activities would equal the number of crops multiplied by the number of intensities being considered. Program output would specify the pattern of activities and, consequently, irrigation levels for each crop. The total quantity of water used at the budgeted cost is a single point on a demand relationship for irrigation water. Additional solutions at alternative water prices would be required to generate the entire function. The procedure would be similar to that employed by Hartman and Whittelsey (1).

When the product prices depend on output, one of two variants of the present method can be employed to determine demand for irrigation water. Each may have advantages in a particular problem, and they require somewhat different modifications of the present program. In the first method, an irrigation intensity is initially assumed for each crop. The cost of irrigation water at this intensity of use is included in the budgeted production costs, and the level of water use affects yield as determined from a production function. The present program is then employed to estimate a land use pattern for crops irrigated at these intensities. This solution provides a set of product prices which can be used to determine the points on the production functions of the individual crops, where the marginal revenue of water is equal to its price. If one or more of these optimum levels of irrigation water application differs from the assumed level, the level is adjusted and the process is repeated until the levels of intensity are optimum for all crops. Steps similar to those used in the present program can be used to assure convergence, for continuous production functions. The total amount of water used in the final solution is a point of a derived aggregate demand curve for irrigation water. Similar solutions at several prices of water would trace out a water demand curve. It should perhaps be emphasized that the marginal revenue of input (water) is made to depend on the realized prices of all outputs, a result not achieved by ordinary methods.

A second method which is only approximate but a good deal faster than the above can be developed by a slightly greater modification of the basic procedure. A number of irrigation intensities are specified for each crop, hopefully straddling the optimum level at a given price of water. The coefficient matrix of the Simplex subroutine is modified to show an activity for each crop on each land class irrigated at each intensity. Yields, costs, and net returns are determined for each productive activity and the optimum pattern of land use is determined. The water input will be earning its marginal cost at each level of use appearing in the solution, within the level of accuracy induced by the number of water intensity levels used. As in the previous method, the price of water can be varied and total quantity of water used can be determined to obtain a number of points on a water demand curve. The computer program would need to be altered in a number of ways, principally in regard to the reading in of the coefficients matrix.

Both of the methods shown for finding a derived demand curve for irrigation water depend on knowing the production functions for all crops being grown in the project area. Since these are usually not known, it may be necessary to assume a fixed level of water application and a fixed yield for each crop on each land class. A price-quantity relationship for irrigation water can be established by simply estimating a series of land use patterns, and resultant water use totals, corresponding to different water prices, at the given intensity of water use (2). This calls for repeated use of the present computer program, without modification.

REFERENCES

(1) Hartman, L. M., and Norman Whittelsay. Marginal Values of Irrigation Water. Technical Bulletin 70, Colorado State University, Fort Collins, Colorado.

(2) Hogg, Howard C. The Diversified Crop Demand for Molokai Irrigation Project Water, 1965-1970. Agricultural Economics Report No. 72, University of Hawaii, Honolulu, Hawaii.

(3) Hogg, Howard C., and Arnold B. Larson. "An Iterative Linear Programming Procedure for Estimating Patterns of Agricultural Land Use," in Agricultural Economics Research, January 1968.

APPENDIX A
PROGRAM PRINTOUT

(SYSTEM CARDS)

MAIN

```
0001              DIMENSION COST(10,10),YIELD(10,10),ACRES(10),
                 1AA(10),BB(10),CC(10),DD(10),CUNIT(10,10),P(10),
                 2COSTU(10,10),Q(10),C(100),A(20,121),
                 3X(10,10),AAA(20,121),QTY(20),QUANT(10),POLD(10),
                 4JCB(50),OPNET(10),COMP(10),EQPR(10),RNET(10,10)
0002              DOUBLE PRECISION A, AAA
0003              COMMON A, C, QTY, NOITER,P,Q,RNET,QUANT,MC,ML,
                 1TRACE,LA,NAA,AAA,OPTION,X
0004              NOITER=0
                C READ IN DATA
                C M= NUMBER OF CROPS
                C N= NUMBER OF LAND CLASSES
                C OPTION= ONE IF COMPLETE PRINT-OUT DESIRED
                C TOLERA= TOLERANCE FOR LAND -ACRES
                C TOLERC= TOLERENCE FOR CROPS -1000 POUNDS
0005              READ(5,1) M,N,OPTION,TOLERA,TOLERC
0006            1 FORMAT(2I5,F5.0,2F5.3)
0007              LA=M
0008              NAA=N
0009              KN = N+1
                C READ IN COST, YIELD, ACRES,AND DEMAND AND SUPPY
0010              DO 10 I=1,M
0011             10 READ(5,11)(COST(I,J), J=1,N)
0012             11 FORMAT(8F10.4)
0013              DO 12 I=1,M
0014             12 READ(5,13)(YIELD(I,J), J=1,N)
0015             13 FORMAT(8F10.4)
0016              READ(5,14)(ACRES(J), J=1,N)
0017             14 FORMAT(8F10.4)
0018              READ(5,15)(AA(I),I=1,M)
0019              READ(5,15)(BB(I), I=1,M)
0020              READ(5,15)(CC(I),I=1,M)
0021              READ(5,15)(DD(I),I=1,M)
0022             15 FORMAT(8F10.4)
                C READ IN MATRIX OF COEFFICIENTS
                C COEFFICIENT MATRIX- C1LC1, C2LC1, ETC.
0023              ML=M+N
0024              MC=M*N
0025              MRE=ML+MC+1
0026              DO 80 J=1,MC
0027             80 READ(5,81)(AAA(K,J), K=1,ML)
0028             81 FORMAT(8D10.4)
0029              READ(5,84)(AAA(J,MRE),J=1,ML)
0030             84 FORMAT(8D10.4)
0031              FACTOR=10.
0032              TRACK=2.
                C FIND PER UNIT COST BY CROP AND LAND CLASS
0033              DO 16 I=1,M
```

MAIN

```
0034            DO 16 J=1,N
0035          16 CUNIT(I,J)=COST(I,J)/YIELD(I,J)
0036            DO 420 I=1,M
0037         420 EQPR(I)=(DD(I)*AA(I)+BB(I)*CC(I))/(BB(I)+DD(I))
              C SOLVE FOR EQUILIBRIUM FOR CROP I
0038         200 DO 100 I=1,M
0039            DO 111 IC=1,M
0040            DO 111 J=1,N
0041         111 COSTU(IC,J)=COST(IC,J)/YIELD(IC,J)
              C FIND LOWEST COST PER UNIT LAND FOR CROP I
0042            CLOW = COSTU(I,1)
0043            KK=1
0044            KNTLND= 1
0045            SMPROS = 0.
0046            DO 3 K=2,N
0047            DIF = COSTU(I,K)-CLOW
0048            IF(DIF.GE.0.) GO TO 3
0049            CLOW = COSTU(I,K)
0050            KK=K
0051          3 CONTINUE
              C TEST TO SEE IF ANY OF CROP I CAN BE GROWN ON PROJECT LAND
0052            DEMQ = (AA(I)-CLOW)/BB(I)
0053            SUPQ = (CLOW-CC(I))/DD(I)
0054            IF(SUPQ.GT.0.) GO TO 4
0055            SUPQ = 0.
0056          4 CONTINUE
0057            DIF = DEMQ-SUPQ
0058            IF(DIF.LE.0.) GO TO 30
0059            GO TO 2
              C AT 30 THERE IS NO PRODUCTION ON PROJECT LAND
0060         30 P(I) = (DD(I)*AA(I)+BB(I)*CC(I))/(BB(I)+DD(I))
0061            POLD(I)=P(I)
0062            Q(I) = (AA(I)-P(I))/BB(I)
0063            QUANT(I)=0.
0064            GO TO 100
              C FIND PROJECT SUPPLY ON LOWEST COST LAND
0065          2 SUPLO = SUPQ
0066            PROSUP = ACRES(KK)*YIELD(I,KK)
0067            SMPROS = SMPROS+PROSUP
0068            SUPHI = SUPLO+PROSUP
0069            DIFF = DEMQ - SUPHI
0070            IF(DIFF.GT.0.) GO TO 6
0071            P(I) = CLOW
0072            POLD(I)=P(I)
0073            Q(I) = (AA(I)-CLOW)/BB(I)
0074            SMPROS = Q(I)-SUPLO
0075            QUANT(I)=SMPROS
0076            SMPROS=0.
```

MAIN

```
0077            GO TO 100
              C FIND NEXT LOWEST COST LAND
0078          6 KNTLND = KNTLND+1
0079            IF(KNTLND.LT.KN) GO TO 31
              C PRICE IS DETERMINED AFTER ALL PROJECT LAND HAS BEEN USED
0080            Q(I) = (AA(I)-CC(I)+DD(I)*SMPROS)/(BB(I)+DD(I))
0081            P(I) = AA(I)-BB(I)*Q(I)
0082            POLD(I)=P(I)
0083            QUANT(I)=SMPROS
0084            SMPROS=0.
0085            GO TO 100
              C THERE IS MORE LAND
0086         31 COSTU(I,KK) = 10000.
0087            CLOW = COSTU(I,1)
0088            KK=1
0089            DO 32 K=2,N
0090            DEF = COSTU(I,K)-CLOW
0091            IF(DEF.GE.0.) GO TO 32
0092            CLOW = COSTU(I,K)
0093            KK = K
0094         32 CONTINUE
              C CLOW IS PRICE ON THE NEXT LOWEST LAND CLASS
              C FIND QUANTITY SUPPLIED BY EXISTING PRODUCERS AT NEW PRICE
0095            SUPQ = (CLOW-CC(I))/DD(I)
0096            DEMQ = (AA(I)-CLOW)/BB(I)
0097            IF(SUPQ.GT.0.) GO TO 33
0098            SUPQ = 0.
0099         33 SUPLO = SUPQ + SMPROS
0100            IF(DEMQ.GT.SUPLO) GO TO 34
              C SUPPLY EQUALS DEMAND ON SLANTING PORTION OF RISER
0101            DEMPR = AA(I)-BB(I)*SUPHI
0102            CP = CC(I)
0103            RNET(I,KK)=0.
0104            IF(DEMPR.GT.CP) GO TO 35
0105            P(I) = DEMPR
0106            POLD(I)=P(I)
0107            Q(I) = SUPHI
0108            QUANT(I)=SMPROS
0109            SMPROS=0.
0110            GO TO 100
0111         35 Q(I)=(AA(I)+DD(I)*SMPROS-CC(I))/(BB(I)+DD(I))
0112            P(I) = AA(I)-BB(I)*Q(I)
0113            POLD(I)=P(I)
0114            QUANT(I)=SMPROS
0115            SMPROS=0.
0116            GO TO 100
              C SEE IF SUPPLY EQUALS DEMAND ON FLAT OF STEP
0117         34 PROSUP = ACRES(KK)*YIELD(I,KK)
```

MAIN

```
0118            SMPROS = SMPROS + PROSUP
0119            SUPHI = SUPLO+ PROSUP
0120            IF(DEMQ.GT.SUPHI) GO TO 6
              C SUPPLY EQUALS DEMAND ON FLAT OF STEP
0121            P(I) = CLOW
0122            POLD(I)=P(I)
0123            Q(I) = (AA(I)-CLOW)/BB(I)
0124            SUPEXP=(P(I)-CC(I))/DD(I)
0125            SMPROS=Q(I)-SUPEXP
0126            QUANT(I)=SMPROS
0127            SMPROS=0.
0128            GO TO 100
0129          100 CONTINUE
              C COMPUTE NET REVENUE FOR EACH CROP ON EACH LAND CLASS
0130            DO 101 I=1,M
0131            DO 101 J=1,N
0132            X(I,J)=0.
0133          101 RNET(I,J)=P(I)*YIELD(I,J)-COST(I,J)
0134            MC=M*N
0135            ML=M+N
0136            DO 102 JJJ=1,N
0137            DO 102 III=1,M
0138            J=III+(JJJ-1)*M
0139          102 C(J)=RNET(III,JJJ)
0140            DO 112 KP=1,MC
0141            ERROR=ABS(C(KP))
0142            IF(ERROR.GT. .001) GO TO 112
0143            C(KP)=.001
0144          112 CONTINUE
0145          113 DO 105 I=1,M
0146            IN=I+N
0147            QUANT(I)=(AA(I)-P(I))/BB(I)-(P(I)-CC(I))/DD(I)
0148            IF(QUANT(I).GT.0.) GO TO 105
0149            QUANT(I)=0.
0150          105 AAA(IN,MRE)=QUANT(I)
0151            NOITER=NOITER+1
0152            DO 120 J=1,MC
0153            DO 120 K=1,ML
0154          120 A(K,J)=AAA(K,J)
0155            DO 119 K=1,ML
0156          119 A(K,MRE)=AAA(K,MRE)
0157            MREE=ML+MC
0158            MEE=MC+1
0159            DO 135 J=MEE,MREE
0160            DO 135 K=1,ML
0161          135 A(K,J)=0.
0162            IF(OPTION.EQ.0.)GO TO 951
0163            WRITE(6,592)
```

MAIN

```
0164          592 FORMAT(1H1)
0165          275 WRITE(6,274) NOITER
0166          274 FORMAT(23H NUMBER OF ITERATIONS =,I5///)
0167          951 CONTINUE
0168          300 CALLSYMPLX(8,23,15,8,0,0,999.)
              C RETURN FROM SYMPLX WITH X,S
              C CHECK TO SEE IF SOLUTION IS OPTIMUM
              C SEE IF QUANTITY OF ANY CROP IS IN DISPOSAL
0169          593 IF(TRACK.EQ.1.) GO TO 351
0170          800 TRACE=0.
0171              BIG=0.
0172              DO 271 I=1,M
0173              IK=I+N
0174              IF(QTY(IK).LT.TOLERC) GO TO 271
0175              IF(QTY(IK).LT.BIG) GO TO 271
0176              BIG=QTY(IK)
0177              ITEM=I
0178              TRACE=1.
0179          271 CONTINUE
0180              IF(TRACE.EQ.1.) GO TO 811
0181              DO 720 J=1,N
0182              IF(QTY(J).EQ.0.) GO TO 720
0183              DO 721 I=1,M
0184              IF(RNET(I,J).LT..001) GO TO 721
0185              GO TO 350
0186          721 CONTINUE
0187          720 CONTINUE
0188              RECUR=TRACE+TRACK
0189              IF(RECUR.EQ.0.)GO TO 998
              C CROP ITEM IS IN DISPOSAL
              C INCREASE PRICE FOR CROP IDENTIFIED BY SUBSCRIPT ITEM
0190          811 PREV=P(ITEM)+.001
0191              P(ITEM)=P(ITEM)+(EQPR(ITEM)-POLD(ITEM))/FACTOR
0192              PRIN=(EQPR(ITEM)-POLD(ITEM))/FACTOR
0193              IF(OPTION.EQ.0.)GO TO 960
0194              WRITE(6,731) ITEM,PRIN
0195          731 FORMAT(24H PRICE INCREASE FOR CROP,I3, 7H EQUALS,F8.3)
0196          960 CONTINUE
              C ARE THERE POTENTIAL NOT ACTUAL COMPETITORS FOR CROP ITEM
0197              SUM=0.
0198              TIC=0.
0199              DO 722 J=1,N
0200          722 SUM=SUM+X(ITEM,J)
0201              DO 723 J=1,N
0202              DO 723 I=1,M
0203              IF(ITEM.EQ.I) GO TO 723
0204              ERR=RNET(I,J)-RNET(ITEM,J)
0205              ERRX=ABS(ERR)
```

MAIN

```
0206            IF(ERRX.GT..0005) GO TO 723
                C IF SO, PRETEND THAT ITEM COMPETES NOW
0207            TIC=1.
0208            X(ITEM,J)=5.
0209            IF(X(I,J).NE.0.) GO TO 723
0210            X(I,J)=5.
0211          723 CONTINUE
                C IS PRICE HIGHER THAN PRODUCTION COST ON IDLE LAND
0212          276 DO 310 J=1,N
0213            IF(QTY(J).EQ.0.) GO TO 310
0214            COSTU(ITEM,J)=COST(ITEM,J)/YIELD(ITEM,J)
0215            IF(COSTU(ITEM,J).LE.PREV) GO TO 310
0216            IF(COSTU(ITEM,J).GT.P(ITEM)) GO TO 310
0217            PRINC=COSTU(ITEM,J)-P(ITEM)
0218            P(ITEM)=COSTU(ITEM,J)
0219            IF(OPTION.EQ.0.)GO TO 961
0220            WRITE(6,731) ITEM,PRINC
0221          961 CONTINUE
0222          310 CONTINUE
0223            ST=SUM+TIC
0224            IF(ST.EQ.0.) GO TO 100
                C ADJUST PRICES OF COMPETING CROPS
0225            DO 272 I=1,M
0226          272 COMP(I)=0.
0227            COMP(ITEM)=P(ITEM)
                C RAISE PRICES OF ALL CROPS THAT COMPETE WITH CROP ITEM
0228            DO 280 J=1,N
0229            OPNET(J)=P(ITEM)*YIELD(ITEM,J)-COST(ITEM,J)
0230            IF(X(ITEM,J).EQ.0.) GO TO 280
0231            DO 277 I=1,M
0232            IF(I.EQ.ITEM) GO TO 277
0233            IF(X(I,J).EQ.0.) GO TO 277
                C CROP I AND CROP ITEM ARE BOTH GROWN ON LAND CLASS J
0234            CONDP2=(OPNET(J)+COST(I,J))/YIELD(I,J)
0235            IF(CONDP2.GT.EQPR(I)) GO TO 278
0236            IF(CONDP2.LT.P(I)) GO TO 290
0237            P(I)=CONDP2
0238            GO TO 290
0239          278 P(I)=EQPR(I)
0240          290 COMP(I)=P(I)
0241          277 CONTINUE
0242          280 CONTINUE
                C SEE IF ANY CROP DOES NOT COMPETE DIRECTLY WITH CROP ITEM
0243            DO 273 I=1,M
0244            IF(I.EQ.ITEM) GO TO 273
0245            IF(COMP(I).GT.0.) GO TO 273
                C CROP I DOES NOT COMPETE DIRECTLY
                C SEE IF IT COMPETES INDIRECTLY
```

MAIN

```
0246                DO 701 JK=1,N
0247                IF(X(I,JK).EQ.0.) GO TO 701
0248                DO 702 IK=1,M
0249                IF(I.EQ.IK) GO TO 702
0250                IF(X(IK,JK).EQ.0.) GO TO 702
                C CROPS I AND IK ARE BOTH GROWN ON LAND CLASS JK
                C DID CROP IK COMPETE DIRECTLY WITH CROP ITEM
0251                IF(COMP(IK).EQ.0.) GO TO 702
0252                OPNET(JK)=P(IK)*YIELD(IK,JK)-COST(IK,JK)
0253                CONDP3=(OPNET(JK)+COST(I,JK))/YIELD(I,JK)
0254                IF(CONDP3.GT.EQPR(I)) GO TO 704
0255                IF(CONDP3.LT.P(I)) GO TO 702
0256                P(I)=CONDP3
0257                GO TO 703
0258              704 P(I)=EQPR(I)
0259              703 COMP(I)=P(I)
0260              702 CONTINUE
0261              701 CONTINUE
0262              273 CONTINUE
0263                GO TO 100
                C DOES LAND IN DISPOSAL EARN RENT
0264              350 DO 812 I=1,M
0265              812 POLD(I)=P(I)
0266                FACTOR=FACTOR*2.
0267              351 TRACK=0.
0268                BIG=0.
0269                DO 260 J=1,N
0270                IF(QTY(J).LT.TOLERA) GO TO 260
0271                DO 710 I=1,M
0272                IF(RNET(I,J).LT..001) GO TO 710
0273                IF(RNET(I,J).LE.BIG) GO TO 710
0274                BIG=RNET(I,J)
0275                ITEM=I
0276                TRACK=1.
0277              710 CONTINUE
0278              260 CONTINUE
0279                IF(TRACK.EQ. 0.) GO TO 800
                C CROP ITEM EARNS HIGHEST RENT ON A LAND CLASS IN DISPOSAL
                C REDUCE PRICE OF CROP ITEM
0280                P(ITEM)=P(ITEM)-(EQPR(ITEM)-POLD(ITEM))/FACTOR
0281                PRDE=(EQPR(ITEM)-POLD(ITEM))/FACTOR
0282                IF(OPTION.EQ.0.)GO TO 962
0283                WRITE(6,732) ITEM, PRDE
0284              732 FORMAT(24H PRICE DECREASE FOR CROP, I3, 7H EQUALS,F8.3)
0285              962 CONTINUE
                C REDUCE PRICES OF COMPETING CROPS
0286                DO 872 I=1,M
0287              872 COMP(I)=0.
```

MAIN

```
0288            COMP(ITEM)=P(ITEM)
              C REDUCE PRICES OF ALL CROPS THAT COMPETE DIRECTLY WITH ITEM
0289            DO 880 J=1,N
0290            OPNET(J)=P(ITEM)*YIELD(ITEM,J)-COST(ITEM,J)
0291            IF(X(ITEM,J).EQ.0.) GO TO 880
0292            DO 877 I=1,M
0293            IF(X(I,J).EQ.0.) GO TO 877
              C CROP I AND CROP ITEM ARE BOTH GROWN ON LAND CLASS J
0294            CONDP4=(OPNET(J)+COST(I,J))/YIELD(I,J)
0295            IF(CONDP4.GT.P(I)) GO TO 890
0296            P(I)=CONDP4
0297          890 COMP(I)=P(I)
0298          877 CONTINUE
0299          880 CONTINUE
              C SEE IF ANY CROP DOES NOT COMPETE DIRECTLY WITH CROP ITEM
0300            DO 873 I=1,M
0301            IF(COMP(I).GT.0.) GO TO 873
              C CROP I DOES NOT COMPETE DIRECTLY
              C SEE IF IT COMPETES INDIRECTLY
0302            DO 901 JK=1,N
0303            IF(X(I,JK).EQ.0.) GO TO 901
0304            DO 902 IK=1,M
0305            IF(X(IK,JK).EQ.0.) GO TO 902
              C CROPS I AND IK ARE BOTH GROWN ON LAND CLASS JK
              C DID CROP IK COMPETE DIRECTLY WITH CROP ITEM
0306            IF(COMP(IK).EQ.0.) GO TO 902
0307            OPNET(JK)=P(IK)*YIELD(IK,JK)-COST(IK,JK)
0308            CONDP5=(OPNET(JK)+COST(I,JK))/YIELD(I,JK)
0309            IF(CONDP5.GT.P(I)) GO TO 902
0310            P(I)=CONDP5
0311          902 CONTINUE
0312          901 CONTINUE
0313            COMP(I)=P(I)
0314          873 CONTINUE
0315            GO TO 100
0316          998 IF(OPTION.EQ.1.)GO TO 999
0317            OPTION=1.
0318            GO TO 100
0319          999 STOP
0320            END
```

SYMPLX

```
0001              SUBROUTINE SYMPLX(NB,NT,NV,NSL,NSU,NA,BM)
0002              DIMENSION A(20,121),C(100),Z(100),CB(50),JCB(50),
                 1X(10,10),QTY(20),P(10),Q(10),RNET(10,10),QUANT(10),
                 2AAA(20,121),QTYCP(20)
0003              DOUBLE PRECISION A,  D,ANM, AAA
0004              COMMON A, C, QTY, NOITER,P,Q,RNET,QUANT,MC,ML,
                 1TRACE,LA,NAA,AAA,OPTION,X
0005              IF(OPTION.EQ.0.)GO TO 952
0006              WRITE(6,1) NB,NT,NV,NSL,NSU,NA,BM
0007           1  FORMAT(21HOSYMPLX ENTERED WITH,,/10X,
                 -32H NUMBER OF              EQUATION =,I5,/10X,
                 -32H NUMBER OF       TOTAL VARIABLE =,I5,/10X,
                 -32H NUMBER OF       BASIC VARIABLE =,I5,/10X,
                 -32H NUMBER OF       SLACK VARIABLE =,I5,/10X,
                 -32H NUMBER OF    SURPLUS VARIABLE =,I5,/10X,
                 -32H NUMBER OF ARTIFICIAL VARIABLE =,I5,10X,
                 -17H ASSIGNED PRICE =,F15.0)
0008          952 CONTINUE
0009              IF(NB.EQ.NSL+NA.AND.NT.EQ.NSL+NSU+NA+NV) GO TO 4
0010              WRITE(6,5)
0011           5  FORMAT(34HOARGUMENT ERROR IN SYMPLX, PROGRAM
                 110H ABANDONED)
0012              STOP
0013          100 DO 189 IK=1,NB
0014              DO 189 IJ=1,NT
0015          189 A(IK,IJ)=AAA(IK,IJ)
0016           4  NV1=NV+1
0017              JA=NT-NA+1
0018              J1=JA-1
0019              NVL=NV+NSL
0020              NLU=NSU+NSL
0021              NT1=NT+1
0022              NB1=NB+1
0023              DO 3 K=1,NT1
0024              Z(K)=0.
0025           3  CONTINUE
0026              MRE=MC+ML
0027              MCC=MC+1
0028              A(NB1,NT1)=0.
0029              IF(NLU.EQ.0) GO TO 101
0030              DO 11 K=1,NLU
0031              J=K+NV
0032              C(  J)=0.
0033          11  CONTINUE
0034         101  IF(NA.EQ.0) GO TO 102
0035              DO 12 J=JA,NT
0036              C(  J)=-BM
0037          12  CONTINUE
```

SYMPLX

```
0038        102   IF(OPTION.EQ.0.)GO TO 953
0039              WRITE(6,15)
0040         15   FORMAT(35HOCOEFFICIENTS OF OBJECTIVE FUNCTION)
0041          953 CONTINUE
0042              DO 400 I=1,NT,10
0043              II=MINO(I+9,NT)
0044              IF(OPTION.EQ.0.)GO TO 954
0045              WRITE(6,16)(J,J=I,II)
0046         16   FORMAT(1X,11H VAR. INDEX,10(I8,4X))
0047              WRITE(6,17)(C(J ),J=I,II)
0048         17   FORMAT(1X,6H PRICE,5X,10F12.3)
0049          954 CONTINUE
0050        400   CONTINUE
0051              IF(NSL.EQ.0) GO TO 105
0052              DO 22 J=1,NSL
0053              K=NV+J
0054              A(J,K)=1.
0055         22   CONTINUE
0056        105   IF(NSU.EQ.0) GO TO 106
0057              DO 23 J=1,NSU
0058              K=NSL+J
0059              L=NVL+J
0060              A(K,L)=-1.
0061         23   CONTINUE
0062        106   IF(NA.EQ.0) GO TO 107
0063              DO 24 J=1,NA
0064              K=NSL+J
0065              L=J1+J
0066              A(K,L)=1.
0067         24   CONTINUE
0068        107   DO 220 K=1,NB
0069              IF(OPTION.EQ.0.)GO TO 955
0070              WRITE(6,221) K
0071        221   FORMAT(25HOCOEFFICIENTS OF EQUATION,I4)
0072          955 CONTINUE
0073              DO 410 I=1,NT,10
0074              II=MINO(I+9,NT)
0075              IF(OPTION.EQ.0.)GO TO 956
0076              WRITE(6,16)(J,J=I,II)
0077              WRITE(6,8)(A(K,J),J=I,II)
0078           8  FORMAT(12H COEFFICIENT,10F12.3)
0079          956 CONTINUE
0080        410   CONTINUE
0081        220   CONTINUE
0082              IF(OPTION.EQ.0.)GO TO 963
0083              WRITE(6,25)
0084         25   FORMAT(9HOVECTOR B)
0085          963 CONTINUE
```

SYMPLX

```
0086              DO 28 I=1,NB,10
0087              II=MINO(I+9,NB)
0088              IF(OPTION.EQ.0.)GO TO 957
0089              WRITE(6,26)(K,K=I,II)
0090        26    FORMAT(1X,10H ROW INDEX,1X,10(I8,4X))
0091              WRITE(6,27)(A(K,NT1),K=I,II)
0092        27    FORMAT(1X,9H VECTOR B, 10F12.4)
0093         957  CONTINUE
0094        28    CONTINUE
0095              IF(NSL.EQ.0) GO TO 103
0096              DO 13 J=1,NSL
0097              CB(J)=0.
0098              JCB(J)=NV+J
0099        13    CONTINUE
0100       103    IF(NA.EQ.0) GO TO 104
0101              DO 14 J=1,NA
0102              K=NSL+J
0103              CB(K)=-BM
0104              JCB(K)=J1+J
0105        14    CONTINUE
0106       104    IF(OPTION.EQ.0.)GO TO 958
0107              WRITE(6,225)
0108       225    FORMAT(37HOVARIABLES IN BASIS AND CORRESPONDING
                 1 7H PRICES)
0109         958  CONTINUE
0110              DO 226 I=1,NB,10
0111              II=MINO(I+9,NB)
0112              IF(OPTION.EQ.0.)GO TO 964
0113              WRITE(6, 26)(K,K=I,II)
0114              WRITE(6, 16)(JCB(K),K=I,II)
0115              WRITE(6, 17)(CB(K),K=I,II)
0116         964  CONTINUE
0117       226    CONTINUE
0118              DO 31 K=1,NT
0119              DO 30 J=1,NB
0120              Z(K)=Z(K)+CB(J)*A(J,K)
0121        30    CONTINUE
0122              A(NB1,K)=Z(K)-C(  K)
0123        31    CONTINUE
0124              DO 33 J=1,NB
0125              A(NB1,NT1)=A(NB1,NT1)+CB(J)*A(J,NT1)
0126        33    CONTINUE
0127       300    N=0
0128              ZMIN=1.E20
0129              RMIN=1.E20
0130              DO 32 K=1,NT
0131              IF(A(NB1,K).GE.ZMIN) GO TO 32
0132              ZMIN=A(NB1,K)
```

SYMPLX

```
0133            M=K
0134         32 CONTINUE
0135            IF(ZMIN.GE.(-1.E-9)) GO TO 600
0136            DO 40 J=1,NB
0137            IF(A(J,M).LE.1.E-6) GO TO 40
0138            R=A(J,NT1)/A(J,M)
0139            IF(R.GE.RMIN) GO TO 40
0140            RMIN=R
0141            N=J
0142         40 CONTINUE
0143            IF(N.GT.0) GO TO 41
0144            WRITE(6,44)
0145         44 FORMAT(19HOUNBOUNDED SOLUTION)
0146            STOP
0147           41 CONTINUE
0148            IF(OPTION.EQ.0.)GO TO 965
0149            WRITE(6,42) JCB(N),N,M,N,M,A(N,M)
0150         42 FORMAT(9HOVARIABLE,I4,7H IN ROW,I4,15H IS REPLACED BY,
               -I4,30X,3H A(,I3,1H,,I3,3H) =,E15.4)
0151          965 CONTINUE
0152            JCB(N)=M
0153            DO 51 I=1,NB1
0154            IF(I.EQ.N) GO TO 51
0155            D=A(I,M)/A(N,M)
0156            DO 50 J=1,NT1
0157            A(I,J)=A(I,J)-A(N,J)*D
0158            IF(J.EQ.M) A(I,M)=0.
0159         50 CONTINUE
0160         51 CONTINUE
0161            ANM=A(N,M)
0162            DO 52 J=1,NT1
0163            A(N,J)=A(N,J)/ANM
0164         52 CONTINUE
0165            GO TO 300
0166          600 CONTINUE
0167            IF(OPTION.EQ.0.)GO TO 959
0168            WRITE(6,60)
0169         60 FORMAT(17HOOPTIMAL SOLUTION/)
0170          959 CONTINUE
0171            MC=LA*NAA
0172            ML=LA+NAA
0173            DO 350 IZ=1,ML
0174          350 QTY(IZ)=0.
0175            DO 61 I=1,NB,10
0176            II=MINO(I+9,NB)
0177            IF(OPTION.EQ.0.)GO TO 966
0178            WRITE(6,26)(K,K=I,II)
0179            WRITE(6,16)(JCB(K),K=I,II)
```

SYMPLX

```
0180              WRITE(6,64)(A(K,NT1),K=I,II)
0181           64 FORMAT(1X,9H SOLUTION, 2X, 10F12.4/)
0182          966 CONTINUE
              C CONVERT SOLUTION VECTOR TO X(I,J)
0183              DO 601 K=I,II
0184              JC = JCB(K)
0185              IF(JC. LE. MC) GO TO 621
0186              ID=JC-MC
0187              QTY(ID) = A(K,NT1)
0188          621 INDEX = 1
0189              MMM=1
0190              NN=1
0191          604 IF(JC.EQ.INDEX) GO TO 602
0192              INDEX = INDEX+1
0193              IF(MMM.EQ.LA) GO TO 603
0194              MMM=MMM+1
0195              GO TO 604
0196          603 MMM=1
0197              NN=NN+1
0198              GO TO 604
0199          602 X(MMM,NN)=A(K,NT1)
0200          601 CONTINUE
0201           61 CONTINUE
0202              IF(OPTION.EQ.0.)GO TO 967
0203              WRITE(6,65) A(NB1,NT1)
0204           65 FORMAT(//17H FUNCTION VALUE =,F19.3//)
0205          500 WRITE(6,501)(P(I),I=1,LA)
0206          501 FORMAT(15H CROP PRICES   ,8F12.3//)
0207          502 WRITE(6,503)(Q(I),I=1,LA)
0208          503 FORMAT(15H TOTAL QUANTITY,8F12.3//)
0209              WRITE(6,504)(QUANT(I),I=1,LA)
0210          504 FORMAT(15H PROJECT SUPPLY,8F12.3//)
0211              WRITE(6,506)(QTY(I),I=1,NAA)
0212          506 FORMAT(17H LAND IN DISPOSAL,8F12.3//)
0213              DO 520 I=1,LA
0214              NC=I+NAA
0215          520 QTYCP(I)=QTY(NC)
0216              WRITE(6,507)(QTYCP(I),I=1,LA)
0217          507 FORMAT(17H CROP IN DISPOSAL,8F12.3)
0218              WRITE(6,510)
0219          510 FORMAT(//1HO,20H NET RETURNS BY CROP)
0220              DO 512 J=1,NAA
0221          512 WRITE(6,505)J,(RNET(I,J),I=1,LA)
0222          505 FORMAT(11H LAND CLASS,I5,8F12.3)
0223              WRITE(6,509)
0224          509 FORMAT(//1HO,33H ACRES IN EACH CROP BY LAND CLASS)
0225              DO 508 J=1,NAA
0226          508 WRITE(6,511)J,(X(I,J),I=1,LA)
```

```
0227          511 FORMAT(11H LAND CLASS,I5,8F12.3//)
0228          967 CONTINUE
0229          551 RETURN
0230              END
```

APPENDIX B

GLOSSARY OF NAMES OF VARIABLES IN ORDER OF APPEARANCE

M = number of crops.

N = number of land classes.

ØPTIØN = selector for printout of intermediate results.

TØLERA = level of tolerance for acreage of a land class which can be in disposal without demanding correction.

TØLERC = level of tolerance of quantity of a crop which can be in disposal (unsatisfied demand) without demanding correction.

LA = M for use in SYMPLX subroutine where M has other meaning.

NAA = N for use in SYMPLX subroutine where N has other meaning.

CØST(I,J) = per-acre cost of production of Ith crop on Jth land class.

YIELD(I,J) = per-acre yield of Ith crop on Jth land class.

ACRES(J) = acres of Jth land class available.

AA(I) = intercept for linear demand curve for Ith crop.

BB(I) = slope for linear demand curve for Ith crop (read as positive).

CC(I) = intercept for linear supply curve for Ith crop for existing, non-project producers.

DD(I) = slope for linear supply curve for Ith crop for existing producers.

ML = number of rows in matrix of coefficients for Simplex table.

MC = number of real activities in Simplex table.

MRE = number of columns in Simplex table.

AAA(K,J) = element in Kth row and Jth column of matrix of coefficients.

AAA(J,MRE) = Jth element in the levels or restraints vector of the Simplex table.

FACTØR = factor to regulate adjustment of prices toward equilibrium values.

TRACK = indicator that land is in disposal and earns rent, so all crops grown on that land must have their prices reduced.

CUNIT(I,J) = cost per unit of output of crop I grown on land class J.

EQPR(I) = price of crop I if there were no production of that crop on project land.

CØSTU(I,J) = same as CUNIT(I,J).

SMPRØS = quantity of crop currently being considered which is grown on project land.

P(I) = price of Ith crop.

PØLD(I) = price of Ith crop in previous iteration.

Q(I) = total quantity of Ith crop produced.

QUANT(I) = quantity of Ith crop produced on project land.

CLØW = per-unit cost of production (including rent earned by other crops) which is lower than that on any other available land.

RNET(I,J) = net revenue earned by Ith crop on Jth land class.

PRØSUP = production of the crop currently being considered on a particular land class.

X(I,J) = acreage of Ith crop on Jth land class as determined by current SYMPLX printout.

C(J) = Jth element in objective function for Simplex table.

ERRØR = indicator that a crop may be a potential competitor of crops currently being produced, and signal that the C-value for this crop on some land class should be set equal to a small positive number (.001) so that it might enter the Simplex solution on the next iteration.

AAA(IN,MRE) = element in the restraints vector giving the current level of quantity of a crop demanded at the current price. These elements occupy positions N + 1 to N + M, since the first N positions are occupied by acreage restrictions.

A(I,J) = AAA(I,J) for use in Simplex table.

NØITER = number of iterations.

TRACE = indicator that at least one crop has quantity demanded in disposal, so its price must be increased.

QTY(I) = element in the solution vector from SYMPLX corresponding to a slack activity. If QTY(I) is greater than 0 and I is in the range of 1 to N, a land class is in disposal. If I is in the range of N + 1 to N + M, a quantity of crop demanded is in disposal.

ITEM = subscript denoting crop with largest amount in disposal, thus making it the one whose price is increased at the current iteration.

RECUR = indicator that both tolerance levels are being observed, so final solution has been obtained.

PRIN = price increase to reduce quantity of crop ITEM in disposal.

ERR = indicator that crop I competes with crop ITEM on land class J.

PRINC = second adjustment of price of crop ITEM to ensure that price is not raised above price needed to make the crop profitable in a currently unused land class.

CØMP(I) = price of crop I needed to make it compete with crop ITEM after the price of crop ITEM has been raised.

ØPNET(J) = net revenue earned by crop ITEM on land class J. All crops grown on this land class must be able to earn this amount of rent.

CØNDP2 = tentative value of new price of crop ITEM, which will become the price at this iteration unless price already exceeds this level.

CØNDP3 = same as CØNDP2, but for crops competing with crop ITEM.

PRDE = price reduction for crop ITEM, made to reduce or eliminate land in disposal on which crop ITEM is being grown and earning highest rent.

CØNDP4 = tentative value of new price of crop ITEM, which will become the price unless current price is already lower.

CØNDP5 = same as CØNDP4, but for crop competing with crop ITEM.

QTYCP(I) = quantity of crop I in disposal.

NB = number of equations in Simplex table.

NT = number of columns in Simplex table, exclusive of levels vector.

NV = number of real activities.

NSL = number of slack variables.

NSU = number of surplus variables.

NA = number of artificial activities.

BM = objective function value for artificial activities.

Note: All other variable names (not here defined) should not normally be involved in modifications of the existing program or contribute to an understanding of the method if explained, and are of very transitory elements in the computation process. Some variable names were omitted because their meaning is explicitly given when they are introduced and their purpose is clear, either from the context or by reference to the comment cards interspersed in the program.

UNIVERSITY OF HAWAII
COLLEGE OF TROPICAL AGRICULTURE
HAWAII AGRICULTURAL EXPERIMENT STATION
HONOLULU, HAWAII

ROBERT W. HIATT
Acting President of the University

C. PEAIRS WILSON
Dean of the College and
Director of the Experiment Station

G. DONALD SHERMAN
Associate Director of the Experiment Station