LINEAR PROGRAMMING AND RELATED COMPUTATIONS

A GUIDE TO USDA LP/90

RICHARD H. DAY

SEPTEMBER 1964

FARM PRODUCTION ECONOMICS DIVISION/ECONOMIC RESEARCH SERVICE/U.S. DEPARTMENT OF AGRICULTURE

Richard H. Day, Agricultural Economist, Farm Production Economics Division, Economic Research Service, U.S. Department of Agriculture; now Assistant Professor, Department of Economics, University of Wisconsin, Madison

CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

This handbook was initially conceived as a set of examples describing--or rather illustrating--the use of a very powerful computer program for linear programming and other computations. It is still that, but in the writing it has become much more. First, it was clear that to understand the examples, it was necessary to describe the general character of the computer program, or alternatively, to refer the reader to an outside source. The former seemed more convenient to the student and so Part III on the Design and Use of USDA LP/90 was included. Second, a handbook on a computer program for linear programming seemed incomplete without at least a resume of the theory of linear programming itself. Thus, it was decided to present in Part II such a section. However, it too grew beyond its initial bounds and became rather an elementary introduction to linear programming and several closely related subjects.

The result, then, is this five-part handbook including a general Introduction (Part I), An Introduction to Linear Programming and Related Computations (Part II), The Design and Use of USDA LP/90 (Part III), Sample Computations with USDA LP/90 (Part IV) and finally, A Reference Guide to the Agendum Routines (Part V). The study as a whole is part textbook and part reference manual. As such it should be useful both to beginners and experts.

It is impossible to present the main ideas of linear programming (beyond the most superficial level) without the aid of some mathematics. However, applied economists do not have to prove the mathematical theorems of linear programming nor be able to derive explicit solution techniques. The mathematicians have already done the former and the computer program removes any need for the latter. Yet, to use linear programming effectively as an empirical tool, the economist must understand the logical meaning of the theorems, and their economic significance. This can be done by studying examples rather than mathematical proofs. This has been the approach taken here, though a certain amount of mathematical notation must be used even for this purpose.

Unorthodox is the omission of all but the briefest comment on the Simplex Method. It is the author's opinion that for beginners time spent on that subject might much more profitably be allocated to learning the use of an effective computer program. Parts III, IV and V are designed to serve just such an heuristic purpose. To put the matter bluntly, understanding the theory of linear programming is essential for constructing models and interpreting their solutions. Understanding the mathematics of computing solutions is not essential so long as one has a good computer program at hand to "do the dirty work." The Simplex Method and its variants can, in short, be left to the specialist.

The support of two organizations in the preparation of this guide needs particularly to be acknowledged. The first is the National Cotton Council which supported the author with a part time research stipend while he was on leave from the U.S. Department of Agriculture from November 1960 to July 1962. It was during this period that the author explored the use of the LP/90 computer system and developed specifications for the special USDA features described in the text. The second organization is CEIR, Inc. which made the documentation of the LP/90 system available to the author before its public release. In addition to these, acknowledgment is due Eli Hellerman and Hugh Powell of CEIR, Inc. who programmed the USDA features and who served as patient instructors in the art of using the LP/90 system, and to Tony Round of the same organization for assistance in computing the examples in Part IV.

Two pieces of the present volume draw heavily on material about LP/90 previously published by CEIR, Inc. They are Part III, Sections 3 and 4 and Part V, Sections 1 and 2. The Social Systems Research Institute of the University of Wisconsin provided secretarial and clerical help in the preparation of the manuscript. Finally, the author should like to acknowledge the encouragement and help of his many colleagues at the U.S. Department of Agriculture, particularly Burton L. French and W. Neill Schaller.

PART I INTRODUCTION

1. BACKGROUND

a. The Significance of Linear Programming in Agricultural Economics Research

It is remarkable that linear programming (LP) has become one of the stocks in trade of agricultural economists in a span of time that covers little more than a decade. The first application of linear programming to agriculture per se was described by Hildreth and Reiter in their paper "On the Choice of a Crop Rotation Plan" [7].* Since that publication in 1951 the literature of Agricultural Economics has contained LP contributions to many fields, from production economics and farm management, to marketing economics and spatial competition.

That this should have come to pass is no mere accident. It is surely because linear programming approximates two essential features of many economic problems. (1) Economic choices in practice are made among a finite set of competing alternatives, and (2) the best choice among available alternatives is, in practice, severely constrained by a variety of economic, technological and social factors.

While linear programming does not make possible a satisfactory solution to all research problems, it does make possible useful answers to many compelling questions, and--as the art of model building develops--the range of questions to which it contributes continually widens. The following list of problems is only indicative:

(a) The least-cost mix of feeds that meet certain nutritional standards (the feed-mix problem).

(b) The best production plans for farmers with different tenure position given their price expectations.

(c) The optimal flow of agricultural commodities among various economic regions.

(d) Long-run equilibrium outputs in various agricultural areas.

(e) The alternative effects of various federal price supports and production controls on regional agricultural production.

---

* Numbers in brackets refer to Selected References on page 273.

b. The Utility of Linear Programming and the Art of Computation

While the utility of linear programming for applied economics depends on the inherent realism of its logical structure, it depends equally on technical breakthroughs in computer science. Without the latter the subject would have been important theoretically, but of as little relevance for empirical work as many another economic concept.

As computer "hardware" has developed, computer programs enabling larger and larger models to be solved have come into being. At this particular writing it is possible in several hours time to solve linear programming models with as many as five thousand variables and a thousand constraints. It is likely that models of this size would contain from fifty thousand to two hundred fifty thousand parameters.[1/] Because so many variables interact in actual economic life, these developments have had everything to do with the extensive application of linear programming.

c. What USDA LP/90 Is

"USDA LP/90" is the name of one computer program designed to solve linear programming problems with up to a thousand constraints and more than five thousand "activities" or variables.

Many other computer programs for many different kinds of computers are currently available for solving linear programming problems. Each has its own particular advantages--there is probably no one best computer program. Which one is to be used in a particular case will depend on the problem size, the type of computer available, etc. USDA LP/90 is available to the research workers employed by the USDA and to its research cooperators in Universities and State Experiment Stations. Other researchers can also arrange to use the system--or its equivalent (see Section 3 below).

Because of the USDA LP/90's speed it will usually be quite economical. While the cost of computer time in dollars per hour increases with the memory size and internal computimg speed of the computer,

[1/] Assuming on LP Matrix Density of 1 to 5 percent.

the cost of solving linear programming problems in dollars per problem, usually falls. This increase in efficiency is analogous to the familiar economies of scale, though here "scale" is measured in time--in micro (or even nano) seconds.[2/]

USDA LP/90 has other advantages over many alternative systems. Briefly, these all lie (1) in the convenience of its input data format, (2) in the external control the user has over the program's operation and (3) in the variety of related computations the user can perform.

The system was partly designed for special USDA uses and was acquired to facilitate research in two specific areas, (1) recursive linear programming models which deal with regional production changes over time and (2) interregional competition models which have very many variables and constraints.

d. Our Purpose in Presenting This Handbook

Because experience has shown the great usefulness and economy of USDA LP/90 in agricultural economics research, it seems highly desirable to make a description of it available to economists everywhere. That in fact is the purpose of this handbook. So as to make the handbook useful to as wide a variety of readers as possible, it has been made rather more comprehensive than was first planned. For this reason some readers will find only parts of the handbook useful. The readers who know the theory of linear programming but not LP/90 will be interested only in Parts III to V. On the other hand, those who wish to remind themselves of the basic theory will wish to read Part II before tackling the description of the program.

It is also possible that the handbook will serve as a useful text for a one-semester course in linear programming, such as now being taught in many agricultural economics departments. Part II provides an introduction to the theory, and the remainder a guide to computational technique. While the mathematics of the latter are not presented, they are

---

[2/] A micro-second = 1/1,000,000 second. A nano-second = 1/1,000,000,000 second.

really of secondary interest to the applied linear programmer. As long as USDA LP/90 or some other program can do the computations for him, the economist has little need to know its methods. This explains the only passing reference to the Simplex Method and its variants. On the other hand, a theoretical knowledge of the subject is indispensable both for structuring models and for interpreting model solutions.

A principle that explains the length of the book is that for many people concrete examples facilitate the learning of any subject. The section summarizing the theory contains illustrations of each major concept. Part IV consists entirely of six examples designed to show how the computer system works on a variety of linear programming and related tasks.

Before the latter sections are reached there is a section which describes the main features of the computer program and how it is used. In view of the foregoing principle it would have been appropriate to include illustrations in this section, too. However, for various reasons all of the computation examples were gathered together in Part IV.

USDA LP/90 contains so many possible options that it would have been impossible to illustrate them all. On the other hand, once the reader understands the use of a few of them, he can easily apply the remainder when desired. Thus, Part IV illustrates the use of the system with only a few possible options. Part V then provides a reference guide for <u>all</u> of the various features.

To summarize, the handbook serves the three-fold purpose of:

(1) Introducing and summarizing the theory of linear programming and some related computations;

(2) Describing and illustrating the use of the USDA LP/90 computer system;

(3) Providing a user's reference manual for the possible program options.

2. A BRIEF HISTORY OF USDA LP/90

a. Predecessors of USDA LP/90

The evolution of USDA LP/90 began in the latter part of 1952 when William Orchard-Hays, then of the RAND Corporation, became associated with George Dantzig, originator of the Simplex Method. Orchard-Hays' task was to aid in designing computer programs for Dantzig's solution technique. Their collaboration led to a series of computer programs for several different machines. Each new program incorporated more efficient computation procedures based on new mathematical developments and on the experience garnered from preceding versions [8]. Orchard-Hays' efforts were continued at CEIR, Inc. along with other collaborators, where the SCROL LP system was developed for the SHARE[3/] organization with the financial support of several American business corporations.

b. The SHARE LP/90 System

With the advent of still more powerful computers the LP/90 system was designed. A basic version of this program was made publicly available through the SHARE organization. By this time the system had become extremely efficient and flexible from the user's point of view. Those who worked on the system included Orchard-Hays, David Smith, Nancy Dilenbeck, Eli Hellerman and Hugh Powell.

The basic computer system released to the SHARE organization will, in this handbook, be called SHARE LP/90. Its basic virtue from the user's point of view is (1) its ability to perform many LP related computations in addition to basic LP and (2) its ability to accommodate new, special purpose features without modifying the already existing system (except, of course, to add to it).

[3/] SHARE Distribution Agency, 590 Madison Avenue, New York 22, New York.

c. The Proprietary Version

Primarily under the direction of Eli Hellerman a great variety of special purpose features have been added to the basic SHARE system but whose use is reserved by the Corporation (CEIR) for competitive business reasons. The latter system we shall call "PROP LP/90." It contains all the features of the former and, as mentioned, much else besides. This system can be acquired by commercial arrangement with the Corporation or its services purchased at an hourly rate.

d. The USDA Version

In the spring of 1961 the Farm Production Economics Division contracted with CEIR to add several special features to the LP/90 system that would aid current research in recursive linear programming and other regional programming projects. These features enable the user to compute an LP matrix from more basic data matrices and to compute solutions to a set of recursively interdependent linear programming problems during a single machine "run."

Because the CEIR proprietary features were found to be of great value in the Farm Production Economics Division work, it was decided to acquire them in addition to the basic SHARE system and the special USDA features. <u>In order to respect the proprietary interests of the contractor, the USDA agreed to make these features available only for USDA work or work conducted on a cooperative basis with the USDA</u>. Any user of the SHARE LP/90 system, however, can acquire the special USDA features by arrangement with CEIR, Inc. The use of the PROP features can be obtained by commercial arrangement with that Corporation.

Much of this manual applies equally to all three versions; parts of it apply only to the PROP or USDA versions. The latter cases are clearly indicated in the text.

e. Future Work

No doubt development of computer programs for linear programming will continue uninterrupted. The USDA LP/90 program really captures the computing system at one point in time. It is felt, however, that it is so advanced, that it will remain a useful tool for research even after still better programs for still faster machines are designed.

3. HOW YOU CAN USE THE PROGRAM

a. The Various Versions of LP/90

In the preceding section three versions of LP/90 were discussed. Actually another version with presently existing features could be envisioned. This would be the SHARE System augmented by the special USDA features, but not including the Proprietary features. We shall call such a version PUBLIC LP/90. Thus we have the following tabulation summarizing the available or potentially available versions:

LP/90 VERSION	FEATURES INCLUDED IN THE PROGRAM			
	SHARE	PROPRIETARY	USDA	NEW FEATURES AS DEVELOPED
SHARE	Yes	No	No	No
PUBLIC	Yes	No	Yes	No
USDA	Yes	Yes	Yes	No
PROP	Yes	Yes	Yes	Yes

The fourth row is provided to indicate that only the Proprietary version of LP/90 incorporates all new features as developed. We shall discuss only briefly each version in turn.

b. How to Use SHARE LP/90

The SHARE version of LP/90 is available to any researcher with access to a suitable computer.[4] Information about it can be obtained by writing directly to the SHARE organization.[5] A great deal of this manual is applicable to this version of LP/90. Where special PROP or USDA features are discussed, the text indicates this.

[4] See Part III, Section 4.

[5] See note 3 above.

c. How to Use PUBLIC LP/90

PUBLIC LP/90, as already remarked, is the SHARE version augmented with the special USDA features. Such a version may be acquired by any researcher by sending a copy of the SHARE program master tape to CEIR, Inc.[6] The latter organization will add the USDA features to the system for a nominal fee.

d. How to Use USDA LP/90

Because it contains the Proprietary features (and for that reason the proprietary interests of their owners must be protected), USDA LP/90 can be used only for U.S. Department of Agriculture research, or for research done on a cooperative basis with that public agency. Arrangements for this can be made directly with the Farm Production Economics Division. Alternatively, the user can purchase directly from CEIR, Inc. the <u>services</u> of this program, though in this case the computing must be done by that company.

e. How to Use PROP LP/90

This version of LP/90 is the only one that can be depended upon to contain all of the latest "bells and whistles" and for that reason in the future it may have certain advantageous options that various users would like to exploit. Because of the proprietary nature of this system, however, arrangements for using it must be made directly with CEIR, Inc. As the special USDA features are public property, that company--like any other--is free to use them commercially.

---

[6] Arlington, Virginia.

PART II. A REVIEW OF LINEAR PROGRAMMING AND SOME RELATED COMPUTATIONS

1. THE PRIMAL LINEAR PROGRAMMING PROBLEM

a. Structure

Linear programming is the problem of maximizing (or minimizing) a linear function of several variables which are subject to linear inequality or equation constraints. The algebraic notation for the problem may be expressed as follows:

(1) $$\pi^* = \max_{x_1, \ldots, x_m} (z_1 x_1 + \ldots + z_m x_m)$$

subject to

(2) $$\begin{array}{llll} b_{11} x_1 + \ldots + b_{1m} x_m \leqq c_1, & x_1 \geqq 0 \\ \vdots \quad\quad\quad\quad \vdots & \vdots & \vdots \\ b_{k1} x_1 + \ldots + b_{km} x_m \leqq c_k, & x_m \geqq 0. \end{array}$$

This notation says, "choose the values of the variables $x_1, \ldots, x_m$ which gives π^*, the largest value of the objective function, and which are consistent with the "less than or equal" inequality and/or equation constraints represented by relations 2." The "objective function parameters" $z_1, \ldots, z_m$, the "constraint coefficients," $b_{11}, \ldots, b_{km}$, and the "right-hand side" (RHS) coefficients, $c_1, \ldots, c_k$, are the data or parameters for the problem.

The objective function parameters are often referred to as the "cost row." The columns of constraint coefficients are usually called <u>processes</u> or <u>activities</u>. The optimal solution to the primal problem (or primal solution) shall be denoted $x_1^*, \ldots, x_m^*$; these values give the largest value of π and still satisfy all the constraints. Any set of values that satisfy the constraints is called a <u>feasible solution</u>.

b. Example

To illustrate the primal problem we shall consider a model so simple that its solution can be found readily with only a few elementary computations. Even so, it is a model seriously put forward as a tool for forecasting agricultural land utilization [6].

Let the variables and the objective function parameters be as shown in Table 1.

Table 1: Definition of Variables and Objective Function Parameters for LP Illustration

Column Index j	Meaning of x_j	Profit ($/Acre) z_j
1	Acreage of rice	66.67
2	" " cotton	65.22
3	" " soybean	15.92
4	" " oats	14.10
5	" " corn	2.19

The objective function is then

(3) $$\pi^* = \max_{x_1, \ldots, x_5} (66.67\, x_1 + 65.22\, x_2 + 15.92\, x_3 + 14.10\, x_4 + 2.19\, x_5)$$

Also, consider a total of eleven (k = 11) constraints on the five (m = 5) variables as follows:

(4)
$$\begin{aligned}
x_1 + x_2 + x_3 + x_4 + x_5 &\leqq 1791 \\
x_1 &\leqq 90 \\
x_2 &\leqq 986 \\
x_3 &\leqq 504 \\
x_4 &\leqq 303 \\
x_5 &\leqq 181 \\
-x_1 &\leqq -74 \\
-x_2 &\leqq -681 \\
-x_3 &\leqq -356 \\
-x_4 &\leqq -230 \\
-x_5 &\leqq -127
\end{aligned}$$

The first of these is an overall land restriction, the second five are upper bounds on the maximum acreage allowed in each crop, and the last five are lower bounds on the minimum acreage in each crop. These last ten constraints might be called "flexibility constraints" for they specify limits on the decision maker's flexibility in altering output patterns. They may exist because of a desire to distribute risk, because of irrational or habitual production goals.

Table 2: Summary of Constraints for LP Illustration

Row Index i	Meaning of Constraint	Magnitude of RHS Coefficient c_i (1000's of Acres)
1	Total land availability	1791
2	Upper bound on rice	90
3	" " " cotton	986
4	" " " soybeans	504
5	" " " oats	303
6	" " " corn	181
7	Lower " " rice	- 74
8	" " " cotton	-681
9	" " " soybeans	-356
10	" " " oats	-230
11	" " " corn	-127

The role of the latter five constraints as lower bounds is more clearly revealed by writing them in the equivalent form:

$$
(5) \qquad \begin{array}{lllll} x_1 & & & & \geqq 74 \\ & x_2 & & & \geqq 681 \\ & & x_3 & & \geqq 356 \\ & & & x_4 & \geqq 230 \\ & & & & x_5 \geqq 127 \end{array}
$$

The solution of the model may proceed as follows. Let

$$x_1 = 74, \quad x_2 = 681, \quad x_3 = 356, \quad x_4 = 230, \quad x_5 = 121,$$

That is, let each variable equal its lower bound.

Adding up the total we see that

$$x_1 + x_2 + x_3 + x_4 + x_5 = 1468,$$

so that the overall land restriction is satisfied with plenty to spare. Obviously the upper bounds are satisfied and the lower bounds are of course satisfied as equalities. Thus we have a feasible solution.

Next, increase the acreage of the most profitable crop, rice, to its upper bound, 90. Now

$$x_1 + \ldots + x_5 = 1484,$$

still less than the total land available. Therefore we increase the acreage of the second most profitable crop (cotton) to its upper bound (986) and find that

$$x_1 + \ldots + x_5 = 1789.$$

Now all but two of the available acres are left, so that the acreage of the third most profitable crop can be increased by just exactly that amount. This given the final optimal solution

$$x_1^* = 90, \quad x_2^* = 986, \quad x_3^* = 358, \quad x_4^* = 230, \quad x_5^* = 127.$$

It just satisfies the overall land restriction, all of the other constraints are satisfied, and there is no way to increase profit without violating one of the constraints. The total value of the program is $z_1 x_1^* + \ldots + z_5 x_5^* = \pi^*$ or \$79,527.71.

c. Matrix Representation

In matrix notation the primal problem may be expressed as

(6) $$\pi^* = \max_x z^T x$$

subject to

(7) $$Bx \leq c, \quad x \geq 0$$

where $z = (z_1, \ldots, z_m)^T$, $x = (x_1, \ldots, x_m)^T$, $c = (c_1, \ldots, c_k)^T$,[7/]
and where

$$B = \begin{bmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & & \vdots \\ b_{ki} & \cdots & b_{km} \end{bmatrix}$$

In the numerical example

$$z = \begin{bmatrix} 66.67 \\ 65.22 \\ 15.92 \\ 14.10 \\ 2.19 \end{bmatrix}, \quad x^* = \begin{bmatrix} 90 \\ 986 \\ 358 \\ 230 \\ 127 \end{bmatrix}, \quad c = \begin{bmatrix} 1791 \\ 90 \\ 986 \\ 504 \\ 303 \\ 181 \\ -74 \\ -681 \\ -356 \\ -230 \\ -127 \end{bmatrix}$$

and, finally,

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

7/ "T" means "transpose." The column vector x is defined to be the transpose of the row vector $(x_1, \ldots, x_m)$. Thus, the small unsubscripted Latin letters always refer to column vectors.

d. Additional Remarks on the Primal Problem

Note again that the use of less than or equal inequality signs in (2) is purely a notational convenience. The lower bounds naturally expressed in (4) as "greater than or equal" constraints--were converted into the form used in (2) simply by multiplying through both sides of the statements by a minus one (and reversing the inequality sign).

The meaning of the processes or activities, that is, the column of b coefficients comprising the B (constraint) matrix, depends on the particular LP problem at hand. In some cases the coefficients will represent input coefficients (as in the first row of our example). In other cases some of the coefficients may be negative output coefficients. In some cases, as is in the flexibility constraint rows of our example, there is no natural interpretation of the coefficients as input or output coefficients. The important point is that <u>any</u> problem that can be put into the form of relation 1 and 2 (or of 6 and 7) is a legitimate LP problem.

A final arbitrary feature of relations (1) and (2) is its casting as a maximizing problem. We might just as well have written (1) as

(1') $$\underset{-}{\pi}^{*} = \min_{x_1, \ldots, x_m} (z_1 x_1 + \ldots + z_m x_m).$$

The two problems are really equivalent if we understand that the z-coefficients in (1') must be the negatives of those in (1). The latter representation is convenient for representing cost minimization problems, while the former more naturally express profit maximizing problems.

The computer system LP/90 actually is based on (1'). For that reason, the objective function coefficients for profit maximizing problems must be entered in the input data for that program after being pre-multiplied by minus one. Positive profits will appear in the data as minus numbers, while negative profits (losses) will appear as positive numbers. Minimizing such a transformed objective function will in fact maximize the desired profit function.[8/]

[8/] An optional way of converting an objective function to be maximized into one to be minimized by LP/90 is available. See discussion of SCALE, Part V.

2. THE DUAL LINEAR PROGRAMMING MODEL

a. Structure

To every primal linear programming problem there corresponds a dual linear programming problem with the following structure:

(8) $$\rho^* = \min_{r_1, \ldots, r_m} (r_1 c_1 + \ldots + r_k c_k)$$

subject to

(9) $$\begin{aligned} b_{11} r_1 + \ldots + b_{k1} r_k &\geq z_1, & r_1 \geq 0 \\ \vdots \qquad\qquad & \quad \vdots \\ b_{1m} r_1 + \ldots + b_{km} r_k &\geq z_m, & r_k \geq 0. \end{aligned}$$

This notation says, "choose the values of the (dual) variables $r_1, \ldots, r_k$ which gives ρ^*, the smallest value of the dual objective function (8) consistent with the constraints represented by 9. We note that the rows of b coefficients correspond to the columns of the B matrix in the primal problem; that the objective function coefficients are the RHS elements of the primal problem; that the RHS elements are the objective function coefficients of the primal problem; and finally, that the weak inequality sign in (8) are "greater than or equal" rather than "less than or equal" inequality signs. A feasible solution is any set of values $r_1, \ldots, r_k$ that satisfies the relation of (9), while an optimal solution is any feasible solution $r_1^*, \ldots, r_k^*$ that yields ρ^*.

b. Example

The dual of the problem in the example of Section 1b is as follows:

$$\rho^* = \min_{r_1, \ldots, r_{11}} 1791\, r_1 + 90\, r_2 + 968\, r_3 + 504\, r_4 + 303\, r_5$$

$$+ 181\, r_6 - 74\, r_7 - 681\, r_8 - 356\, r_9 - 230\, r_{10} - 127\, r_{11}.$$

Subject to the constraints:

$$
\begin{array}{llllllllllll}
r_1 & +r_2 & & & & & -r_7 & & & & & \geqq 66.67 \\
r_1 & & +r_3 & & & & & -r_8 & & & & \geqq 65.22 \\
r_1 & & & +r_4 & & & & & -r_9 & & & \geqq 15.92 \\
r_1 & & & & +r_5 & & & & & -r_{10} & & \geqq 14.10 \\
r_1 & & & & & +r_6 & & & & & -r_{11} & \geqq 2.19
\end{array}
$$

The solution of this dual problem is not quite so obvious as is that of the primal problem. Once one knows a bit more of the theory of linear programming, however, the solution reduces to comparably simple calculations. This will be shown in Section 3, below. We shall see there that the solution is

$$
\begin{aligned}
r_1^* &= z_3 = 15.92 \\
r_2^* &= 66.67 - r_1^* = z_1 - z_3 = 50.75 \\
r_3^* &= 65.22 - r_1^* = z_2 - z_3 = 49.30 \\
r_4^* &= 0 \\
r_5^* &= 0 \\
r_6^* &= 0 \\
r_7^* &= 0 \\
r_8^* &= 0 \\
r_9^* &= 0 \\
r_{10}^* &= r_1^* - 14.10 = z_3 - z_4 = 1.82 \\
r_{11}^* &= r_1^* - 2.19 = z_3 - z_5 = 13.73.
\end{aligned}
$$

The value of the dual program $\rho^* = \$79,527.71$.

c. Matrix Representation

In matrix notation the dual problem is expressed as

$$
\rho^* = \min_r r^T c, \tag{10}
$$

subject to

$$(11) \qquad B^T r \geqq z, \qquad r \geqq 0,$$

where $r = (r_1, \ldots, r_k)$, and B means the transpose of B (rows and columns interchanged). Thus in our example,

$$B^T = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

3. SOME THEORETICAL PROPERTIES OF DUAL LINEAR PROGRAMS

In this section the theory of dual linear programs is highlighted. Proofs are not given. These may be found in the works cited in the References included at the end of the handbook.

a. Existence of Solutions

A necessary and sufficient condition that dual linear programs have optimal solutions is that they both have feasible solutions. It is not sufficient if the primal but not the dual (or vice versa) has a feasible solution. In our examples feasible solutions are easily found by inspection.

b. Duality Theorem

If there exist optimal solutions to the dual linear programming problems, then the optimal values of their objective functions are equal, that is, $\pi^* = \rho^*$. Conversely, feasible solution vectors x and r such that $z^T x = r^T c$ are optimal. This property provides a check on the dual solutions of any empirical problem. We note that in the example of Section 1b and 2b, $\pi^* = \rho^*$ so that the dual solution found must indeed be optimal.

c. Meaning of the Dual Variables

The optimal dual variables, r_1^*, ..., r_k^* are the marginal values, respectively, of the RHS constants c_1, ..., c_k. That is, they show how much the objective function would be increased by a small increase in the values of the right-hand-side constants. In calculus notation

$$(12) \qquad r_i^* = \frac{\partial \pi^*}{\partial c_1}, \qquad i = 1, \ldots, k.$$

Thus, r_i^* is the partial derivative of the optimal value of objective function with respect to the i^{th} RHS coefficient.

Turning back to the example of Section 2b, we find that if the total amount of land available is increased one acre, the increase in profit will be \$15.92. This is because all crops are being grown at their maximum or minimum bounds except soybeans. The latter can be increased one acre without violating any constraints. Of course, oats or corn could also be increased one acre, but as they are less profitable than soybeans, that would not be the optimal response.

It will also be seen that if the upper bound on rice is increased one acre, the increase in the total value of the program is \$50.75. As it turns out, this is because rice can be increased one acre grossing an additional \$66.67, but this must be done at the expense of one acre of soybean production. Thus, we get $z_1 - z_3 = r_2^*$. Cotton acreage could have been reduced one acre instead of soybeans, but that would not have been optimal.

The reader may similarly interpret the remainder of the dual variables in the example.

d. "Tight" (Equated) and "Loose" (Slack) Constraints

A "tight" constraint is one that is satisfied as an equality by the optimal solution, a "loose" constraint is one that is satisfied as an inequality. A tight primal constraint would mean that the corresponding "resource" is exhausted by the optimal primal solution. A loose primal constraint would mean that some of the corresponding input is unused by the optimal production plan.

If in the optimal solution, a given primal constraint is loose, then the corresponding dual variable is zero. If a given constraint is tight, the corresponding dual variable will be positive. There will be as many positive dual variables as there are primal equated constraints.[9] Conversely, there will always be one zero activity level for every loose dual constraint and a positive one for every tight dual constraint. Thus, there will be as many positive primal variables as there are dual equated constraints. The tight constraints are frequently called equated constraints and the loose constraints are frequently called slack constraints.

In our numerical example the optimal primal solution satisfied five constraints as equalities, constraints 1, 2, 3, 10, and 11 (the land, rice and cotton upper bounds and oats and corn lower bounds). These correspond to the five positive dual variables r_1, r_2, r_3, r_{10}, and r_{11}. All of the other primal constraints were loose and their corresponding dual variables were zero.

In this example, because of the positive lower bound on each activity level, all five primal variables must be positive. We know in advance therefore that there must be five equated primal constraints with five positive dual variables, though which five we do not know. As there are exactly five dual constraints and as we know that all five primal variables will be positive, we know that all five dual constraints will be tight or equated.

A device for keeping track of the tight and loose constraints will be convenient in what follows. Let $K_1 = \left\{ i_1, \ldots, i_{k'} \right\}$ be the set of row indexes corresponding to the tight primal constraints (and to the positive dual variables). Let $K_2 = \left\{ i_{k'+1}, \ldots, i_k \right\}$ be the set of row indexes associated with the loose primal constraints (and zero dual variables). Also, let $M_1 = \left\{ j_1, \ldots, j_{m'} \right\}$ be the column indexes corresponding to the tight dual constraints (positive primal variables) and $M_2 = \left\{ j_{m'+1}, \ldots, j_m \right\}$ be the set of column indexes corresponding to the loose dual constraints (zero primal variable). Thus,

---

[9] This will not be true if the tight constraints are not all linearly independent. In such a case some of the associated dual variables may be zero. The same qualification holds, *mutatis mutandis*, for the following two sentences.

$$a_{i1}\, x_1 + \ldots + a_{im}\, x_m = c_i, \qquad r_i \geq 0; \qquad i \in K_1$$

$$a_{i1}\, x_1 + \ldots + a_{im}\, x_m < c_i, \qquad r_i = 0; \qquad i \in K_2$$

$$a_{1j}\, r_1 + \ldots + a_{kj}\, r_k = z_j, \qquad j \in M_1$$

$$a_{1j}\, r_1 + \ldots + a_{kj}\, r_k > z_j, \qquad x_j = 0; \qquad j \in M_2.$$

e. Systems of Equated Constraints

Let us form from among primal and dual systems of weak inequality constraints 2 and 6 the sets of primal and dual equated constraints. For any primal variable that is zero (none in this example) we can omit the corresponding column of B matrix coefficients, and for any dual variable that is zero we can omit the corresponding row of B matrix coefficients. We have then the primal system of equated constraints

(13)
$$\begin{aligned}
x_1 + x_2 + x_3 + x_4 + x_5 &= c_1 = 1791 \\
x_1 + x_2 + x_3 + x_4 + x_5 &= c_2 = 90 \\
x_2 &= c_3 = 986 \\
- x_4 &= c_{10} = -230 \\
- x_5 &= c_{11} = -127
\end{aligned}$$

and the dual system of equated constraints

(14)
$$\begin{aligned}
r_1 + r_2 &= 66.67 = z_1 \\
r_1 + r_3 &= 65.22 = z_2 \\
r_1 &= 15.92 = z_3 \\
r_1 - r_{10} &= 14.10 = z_4 \\
r_1 - r_{11} &= 2.19 = z_5 \\
r_4 = r_5 = r_6 = r_7 = r_8 = r_9 &= 0.
\end{aligned}$$

Having divided the variables into positive and zero classes, we can represent the positive primal and dual variables as solutions to the primal and dual systems of equated constraints, respectively.

An even more convenient way of representing the solution is in matrix form. Thus, 13 may be represented as

$$(13') \quad \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_{10} \\ c_{11} \end{bmatrix} = \begin{bmatrix} 1791 \\ 90 \\ 986 \\ -230 \\ -127 \end{bmatrix}$$

and 14 as

$$(14') \quad \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_{10} \\ r_{11} \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{bmatrix} = \begin{bmatrix} 66.67 \\ 65.22 \\ 15.92 \\ 14.10 \\ 2.19 \end{bmatrix}$$

The solution to these are then

$$(15) \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1791 \\ 90 \\ 986 \\ -230 \\ -127 \end{bmatrix}$$

$$(16) \quad \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_{10} \\ r_{11} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 66.67 \\ 65.22 \\ 15.92 \\ 14.10 \\ 2.19 \end{bmatrix}$$

The matrix in (15) is the inverse of the matrix in (13') and that in (16) is the inverse of that in (14'). Moreover, the matrix in (14') is the transpose of that in (13') and the matrix in (16) the transpose of that in (15)!

Recalling now our index sets M_1, M_2, K_1, K_2 we can form the following matrix and vectors:

$$B' = [\, b_{ij} \,] \qquad i \in K_1, \quad j \in M_1$$

$$x' = (x_j), \quad z' = (z_j) \qquad i \in M_1$$

$$x'' = (x_j), \quad z'' = (z_j) \qquad j \in M_2$$

$$r' = (r_i), \quad c' = (c_i) \qquad i \in K_1$$

$$r'' = (r_i), \quad c'' = (c_i) \qquad i \in K_2.$$

(Note that ' does not denote transposition in this manual.)
Thus our dual system of equated constraints may be written as

(17a) $$B' x' = c', \qquad x'' = 0$$

(17b) $$B'^T r' = z', \qquad r'' = 0.$$

Thus B' is comprised of the columns and rows of B that correspond to tight primal and dual constraints.

The preceding heuristic discussion illustrates the following principles. Feasible vectors x^* and r^* are optimal, if and only if, they satisfy dual systems of equated constraints. Further, if B' is non-singular, i.e. it has an inverse $(B')^{-1}$ then 17 is called a non-singular dual system of equated constraints. A solution satisfying a non-singular dual system of equated constraints is called an extreme vector, the non-zero elements of which can be written

(18a) $$(x^*)' = (B')^{-1} \, c' \qquad (x^*)'' = 0$$

(18b) $$(r^*)' = (B'^T)^{-1} \, z' \qquad (r^*)'' = 0.$$

Now (18b) can be re-expressed as

(19) $$(r^*)' = (B'^T)^{-1} \, z' = (B'^{-1})^T \, z' ,$$

so that once the B'^{-1} is found which gives $(x^*)'$, $(r^*)'$ can be found by a matrix multiplication. As our example of sections 1.b and 2.b had an extreme solution, we found ρ^* in this manner.

The solution to dual linear programs are not necessarily unique. By this we mean that several different primal and dual solution vectors may be found that satisfy several different nonsingular dual systems of equated constraints. That is, there may be several extreme optimal vectors to any problem. However, the following theorem relates all solution to the extreme solutions.

Let $x^1, \ldots, x^p; r^1, \ldots, r^p$ be p optimal extreme solutions of a given LP problem (all having the same values $\pi^1 = \ldots = \pi^p = \rho^1 = \ldots = \rho^p$). Then primal and dual vectors x^* and r^* are solutions if and only if there exist scalers $\lambda_1, \ldots, \lambda_p$ not all zero such that

(20a) $$x^* = \lambda_1 x^1 + \ldots + \lambda_p x^p$$

(20b) $$r^* = \lambda_1 r^1 + \ldots + \lambda_p r^p.$$

f. Comparative Cost Factors (the d_j's) and Small Primal Changes

Noted above is the interpretation of the dual variables as partial derivatives or marginal values. From the dual variables a second set of variables can be derived that indicate the relative costs of using activities with zero process levels in the optimal solution. If we let b_j be the j^{th} column of the original constraint matrix then the relations

(20-1) $$b_j^T r^* \geqq z_j$$

are satisfied by the optimal dual variables. If we let

(20-2) $$d_j = z_j - b_j^T r^*$$

then d_j is the relative cost per unit of using the j^{th} activity when its inputs are valued by their optimal marginal values. For columns in the equated constraint matrix (20-1) is an equality so that the corresponding d_j is zero. For columns not in the equated constraint matrix (20-1) is satisfied, in general, as an inequality, implying a positive d_j.

A small d_j suggests that the corresponding process could be substituted for an optimal one with only a small sacrifice in profit, while a large d_j suggests the converse. For this reason many linear programming routines, including LP/90, provide for computing these coefficients after the attainment of an optimal solution.

It is also possible to show that the elements of the inverted equated constraint matrix can be interpreted as partial derivatives. More specifically, if b'_{ij} is the ij^{th} element of the matrix $(B')^{-1}$ then

$$\frac{\partial x_j^*}{\partial c_i} = b'_{ij}. \qquad j \in M_1, \quad i \in K_1 \tag{20-3}$$

$$\frac{\partial x_j^*}{\partial c_i} = 0, \qquad j \notin M_1, \quad i \notin K_1 \tag{20-4}$$

In words, this means that the elements of the inverse of the equated constraint matrix show how much the optimal primal variables would change with a marginal change in the right-hand side constants, given that the system of equated constraints is not changed.

4. PARAMETRIC PROGRAMMING

Very frequently the researcher is not interested merely in a single LP solution. Rather he wishes to know how the optimal solution of a given LP model changes as one or more of its parameters are modified. As there are three basic sets of data or coefficients in an LP model, viz, the c-vector, the z-vector, and the B-matrix, there are three (major) corresponding types of parametric programming operations. The first of these is denoted (in LP/90 lingo) "PLP" and is frequently called "parametric right-hand-side" or "resource mapping." The second is denoted (again in LP/90 language) "PCR" and is more commonly known as "parametric objective function," "parametric cost row," or "price mapping." The third, does not (as far as the present author knows) have a commonly accepted name, though by analogy with the other types of parametric programming it might be called "parametric constraint coefficient" or "technical coefficient mapping," or, paraphrasing Simon [12], "Trigger Effect Analysis." Each of these will be briefly described in turn.

a. Parametric Right-Hand-Side or Resource Mapping (PLP)

Suppose $x^* = (x_1^*, \ldots, x_m^*)^T$ is the optimal primal solution and $r^* = (r_1^*, \ldots, r_k^*)$ is the optimal dual solution of a given LP problem. We wish now to see how r^* and x^* change as the RHS coefficients are changed in some fixed proportion. Let $v_1, \ldots, v_k$ be some constants positive or negative (or zero). The PLP problem may be written as follows:

(21) $$\pi^* (\theta) = \max_{x_1, \ldots x_m} (z_1 x_1 + \ldots + z_m x_m),$$

subject to

(22) $$\begin{matrix} b_{11} x_1 + \ldots + b_{1m} x_m \leqq c_1 + \theta v_1 & x_1 \geqq 0 \\ \vdots \qquad \vdots \quad \vdots \quad \vdots \qquad \vdots & \\ b_{k1} x_1 + \ldots + b_{km} x_m \leqq c_k + \theta v_k & x_m \geqq 0 \end{matrix}$$

for all $0 \leqq \theta \leqq \hat{\theta}$.

This problem always starts after solving the basic linear programming problem with $\theta = 0$. Then θ is increased continuously up to $\hat{\theta}$. As it increases π^*, x^* and r^* will change. Usually the constraints which are tight or loose (and therefore the system of dual equated constraints) will change status as θ is increased. These changes are made in discrete jumps (or in continuous linear segments), however, in between which a linear interpolation may be made for any desired solution variable (π^*, $x_1^*, \ldots, x_m^*$, or $r_1^*, \ldots, r_k^*$). The value of $\hat{\theta}$ may be infinite. The v-coefficients are sometimes called <u>change coefficients</u> and the vector comprised of them, the <u>change vector</u>.

In matrix notation the PLP problem is written

(21') $$\pi^* (\theta) = z^T x$$

subject to

$$(22') \qquad B\,x \leqq c + \theta\,v, \qquad x \geqq 0$$

where $\theta\,v = (v_1, \ldots, v_k)^T$. We may also note that the dual to this problem is

$$\rho^* (\theta) = r^T (c + \theta\,v)$$

subject to

$$B^T\,r \geqq z.$$

The optimal values π^* and ρ^* and the optimal dual solution vectors x^* and r^* may be regarded as being defined as implicit functions of the single parameter θ. The object of PLP algorithms, such as the one used by LP/90, have the task of determining these magnitudes as explicit functions of θ.

b. Example

In order to illustrate PLP we invoke our original example in this new context. Everything is the same as before except for defining a change vector. We shall consider a change vector with $v_1 = 1.0$ and all other elements zero, that is, we shall consider a continuous change of a single RHS element, the overall land availability. The linear inequalities of (22) now become

$$(25) \qquad \begin{array}{rcl} x_1 + x_2 + x_3 + x_4 + x_5 & \leqq & 1791 + \theta\,1.0 \\ x_1 & \leqq & 90 \\ x_2 & \leqq & 986 \\ x_3 & \leqq & 504 \\ x_4 & \leqq & 303 \\ x_5 & \leqq & 181 \\ -x_1 & \leqq & -74 \\ -x_2 & \leqq & -681 \\ -x_3 & \leqq & -356 \\ -x_4 & \leqq & -230 \\ -x_5 & \leqq & -127 \end{array}$$

The objective function (relation 3) is the same as before. The solution for $\theta = 0$ is already known and is $x_1^* = 90$, $x_2^* = 986$, $x_3^* = 358$, $x_4^* = 230$, $x_5^* = 127$. Of these optimal solution values x_1^* and x_2^* are at their upper bounds so that further increase is not possible. x_3^* however, the crop (soybeans) next in line in profitability can be increased to its upper bound of 504, or an increase of 504 - 358 = 146. Thus θ may increase from 0 to 146 without changing the dual system of equated constraints that describe the solution. At $\theta = 146$, however, a second dual system of equated constraints will describe the solution as well as the initial system [see Section 3.e, equations (13) and (14)]. The solution is, in this case,

$$x_1^* = 90, \quad x_2^* = 986, \quad x_3^* = 504, \quad x_4^* = 230, \quad x_5^* = 127,$$

and the second dual system of equated constraints would be

$$(26) \quad \begin{array}{lllllcll} x_1 & + x_2 & + x_3 & + x_4 & + x_5 & = & 1937 & = 1791 + \theta\ 1.0 \\ x_1 & & & & & = & 90 & \\ & x_2 & & & & = & 986 & \\ & & x_3 & & & = & 504 & \\ & & & & -x_5 & = & -127 & \end{array}$$

$$(27) \quad \begin{array}{lllllcl} r_1 & r_2 & & & & = & 66.67 \\ r_1 & & r_3 & & & = & 65.22 \\ r_1 & & & r_4 & & = & 15.92 \\ r_1 & & & & & = & 14.10 \\ r_1 & & & & -r_{11} & = & 2.19 \end{array}$$

As θ is increased beyond 146, dual system (26) and (27) will describe the solution until (and including) $\theta = 219$. At this point the optimal solution is

$$x_1^* = 90, \quad x_2^* = 986, \quad x_3^* = 504, \quad x_4^* = 303, \quad x_5^* = 127,$$

and, as θ increased from 146 to 219, the acreage of oats--the most profitable crop that is not already at its upper bound--increases continuously.

Again, at $\theta = 219$ a second dual system [other than (26)-(27)] will describe the optimal solution:

$$
(28)\qquad
\begin{array}{lllllcl}
x_1 & + x_2 & + x_3 & + x_4 & + x_5 & = & 2010 \quad = 1791 + \theta\ 1.0 \\
x_1 & & & & & = & 90 \\
 & x_2 & & & & = & 986 \\
 & & x_3 & & & = & 504 \\
 & & & x_4 & & = & 303
\end{array}
$$

and

$$
(29)\qquad
\begin{array}{lllllcl}
r_1 & r_2 & & & & = & 66.67 \\
r_1 & & r_3 & & & = & 65.22 \\
r_1 & & & r_4 & & = & 15.92 \\
r_1 & & & & r_5 & = & 14.10 \\
r_1 & & & & & = & 2.19.
\end{array}
$$

Like the first instance the values of any of the solution variables may be interpolated linearly for the values $146 < \theta < 219$.

In a similar manner we find that from $\theta = 219$ to $\theta = 273$ a linear interpolation can be made between the values at 219 and 273. For the latter value of θ the solution is

$$x_1^* = 90,\quad x_2^* = 986,\quad x_3^* = 504,\quad x_4^* = 303,\quad x_5^* = 181.$$

At this point no change in solution is effected through a rise in θ, for with each variable at its upper bound, none can be increased. Consequently, though θ can be increased indefinitely beyond this point without affecting the feasibility of the program, there is no point in doing so. With a little forethought we might have anticipated this result, and planned at the beginning for a range for PLP of $0 \leqq \theta \leqq 273$ (i.e. $\hat{\theta} = 273$). We may note that a second dual system of equated constraints can represent the solution at $\theta = 273$ in addition to (28) - (29).
It is

$$
(30)\qquad
\begin{array}{lllllcl}
x_1 & & & & & = & 90 \\
 & x_2 & & & & = & 986 \\
 & & x_3 & & & = & 504 \\
 & & & x_4 & & = & 303 \\
 & & & & x_5 & = & 181
\end{array}
$$

and

$$
(31)\qquad
\begin{array}{llllllcl}
r_2 & & & & & & = & 66.67 \\
 & r_3 & & & & & = & 65.22 \\
 & & r_4 & & & & = & 15.92 \\
 & & & r_5 & & & = & 14.10 \\
 & & & & r_6 & & = & 2.19
\end{array}
$$

The results of this exercise in PLP are summarized in Tables 3 and 4 and in Figures 1 to 3. Figure 1 shows the program value ($\pi^* = p^*$) for each value of θ; Figure 2 shows the values of each primal variable as functions of θ; finally, Figure 3 shows the values of each dual variable as functions of θ. Note that π^* changes at each value of θ until $\hat{\theta} = 273$ is reached, but at a decreasing rate. However, the change in π^* occurs in discrete linear segments each one of which is associated with a unique dual system of equated constraints. Also, note, both the discrete jumps and sloping linear segments of the acreage response curves, and the purely discrete jumps in the dual variables. These results of course depend on the particular structure of this LP example.

Table 3: Summary of Primal Solutions for PLP Example

Primal Index j	Objective Coefficients z_j	θ=0 x_j^*	θ=146 x_j^*	θ=219 x_j^*	θ=273 x_j^*
1	66.67	90	90	90	90
2	65.22	986	986	986	986
3	15.92	358	504	504	504
4	14.10	230	230	303	303
5	2.19	127	127	127	181
Primal Value	π^*	79,528	81,852	82,881	83,000

Table 4: Summary of Dual Solutions for PLP Example

Dual Variable Index i	θ = 0 RHS Elem. $c_i + \theta v_i$	θ = 0 Dual Var. r_i^*	θ = 146 RHS Elem. $c_i + \theta v_i$	θ = 146 Dual Var. r_i^*
1	1791	15.92	1937	14.10
2	90	50.75	90	52.57
3	986	49.30	986	51.12
4	504	0.00	504	1.82
5	303	0.00	303	0.00
6	181	0.00	181	0.00
7	- 74	0.00	- 74	0.00
8	-681	0.00	-681	0.00
9	-356	0.00	-356	0.00
10	-230	1.82	-230	0.00
11	-127	13.73	-127	11.91
Dual Value	ρ^* =	79,528	ρ^* =	81,852

Dual Variable Index i	θ = 219 RHS Elem. $c_i + \theta v_i$	θ = 219 Dual Var. r_i^*	θ = 273 RHS Elem. $c_i + \theta v_i$	θ = 273 Dual Var. r_i^*
1	2010	2.19	2064	0.00
2	90	64.48	90	66.67
3	986	63.03	986	65.22
4	504	13.73	504	15.92
5	303	11.91	303	14.10
6	181	0.00	181	2.19
7	- 74	0.00	- 74	0.00
8	-681	0.00	-681	0.00
9	-356	0.00	-356	0.00
10	-230	0.00	-230	0.00
11	-127	0.00	-127	0.00
Dual Value	ρ^* =	82,881	ρ^* =	83,000

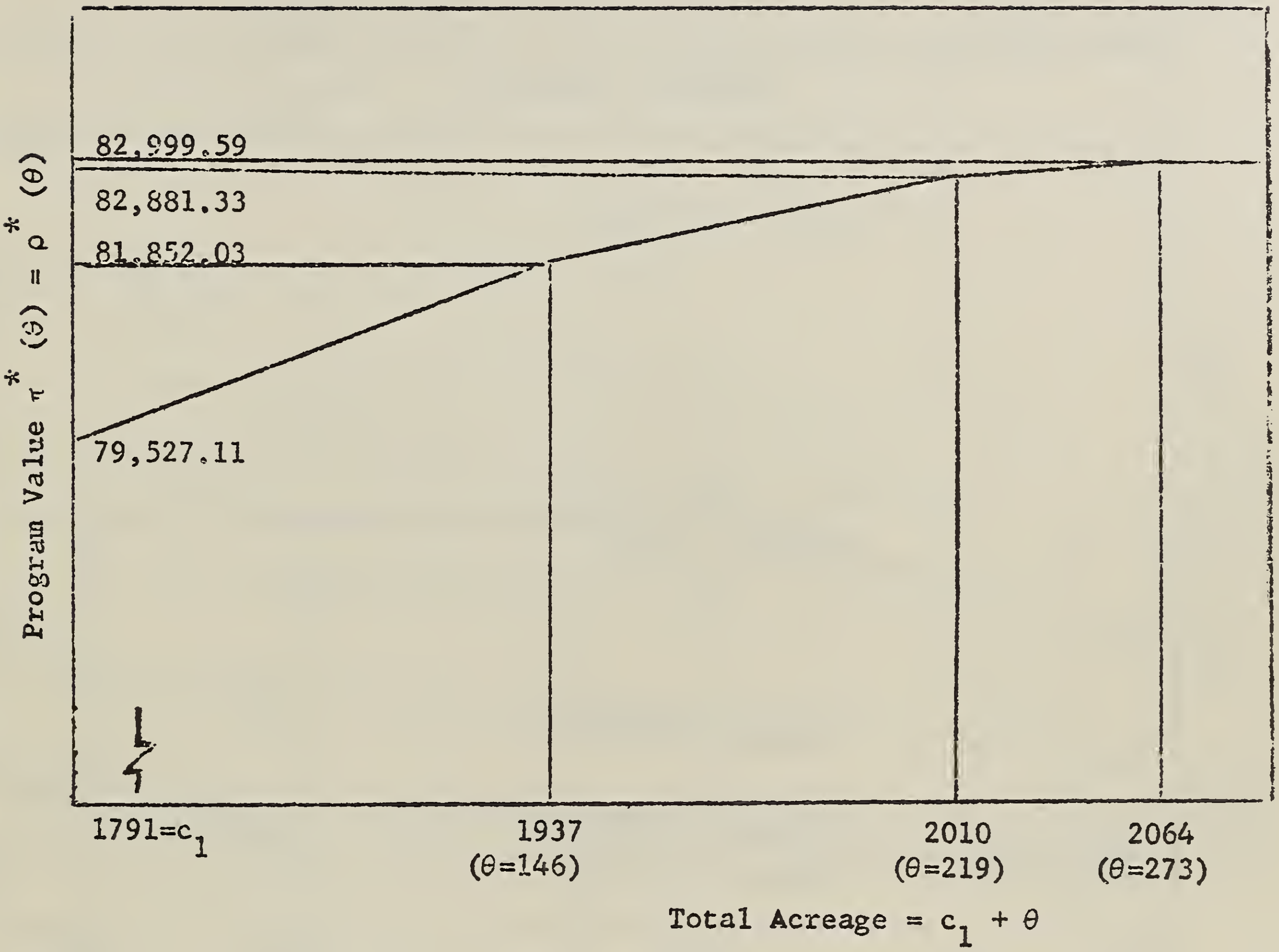


FIGURE 1

OPTIMAL VALUE OF EXAMPLE PROGRAM AS A FUNCTION OF TOTAL LAND (= $c_1 + \theta$)

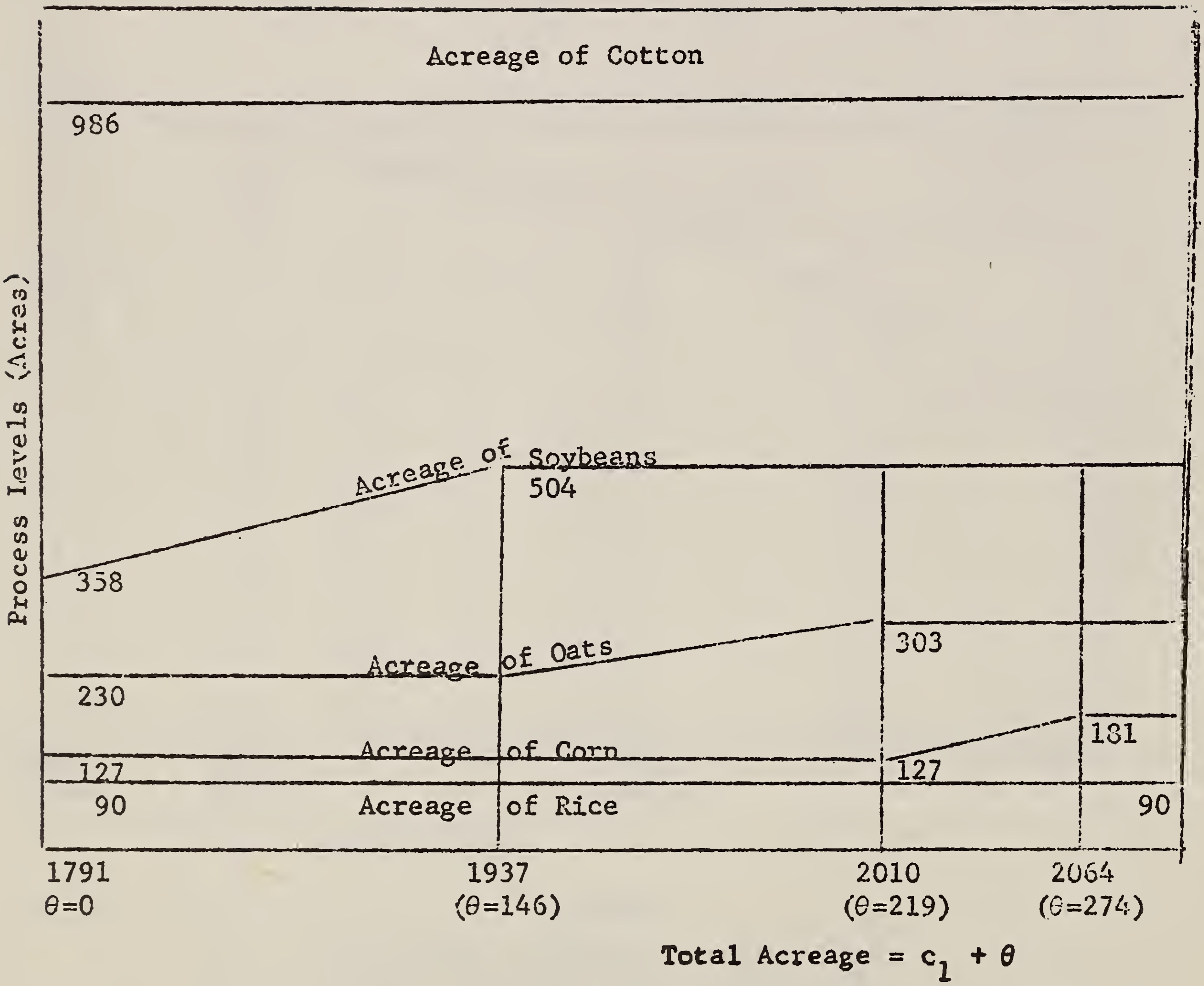


FIGURE 2

OPTIMAL PROCESS LEVELS OF EXAMPLE PROGRAM AS FUNCTIONS OF TOTAL LAND (= $c_1 + \theta$)

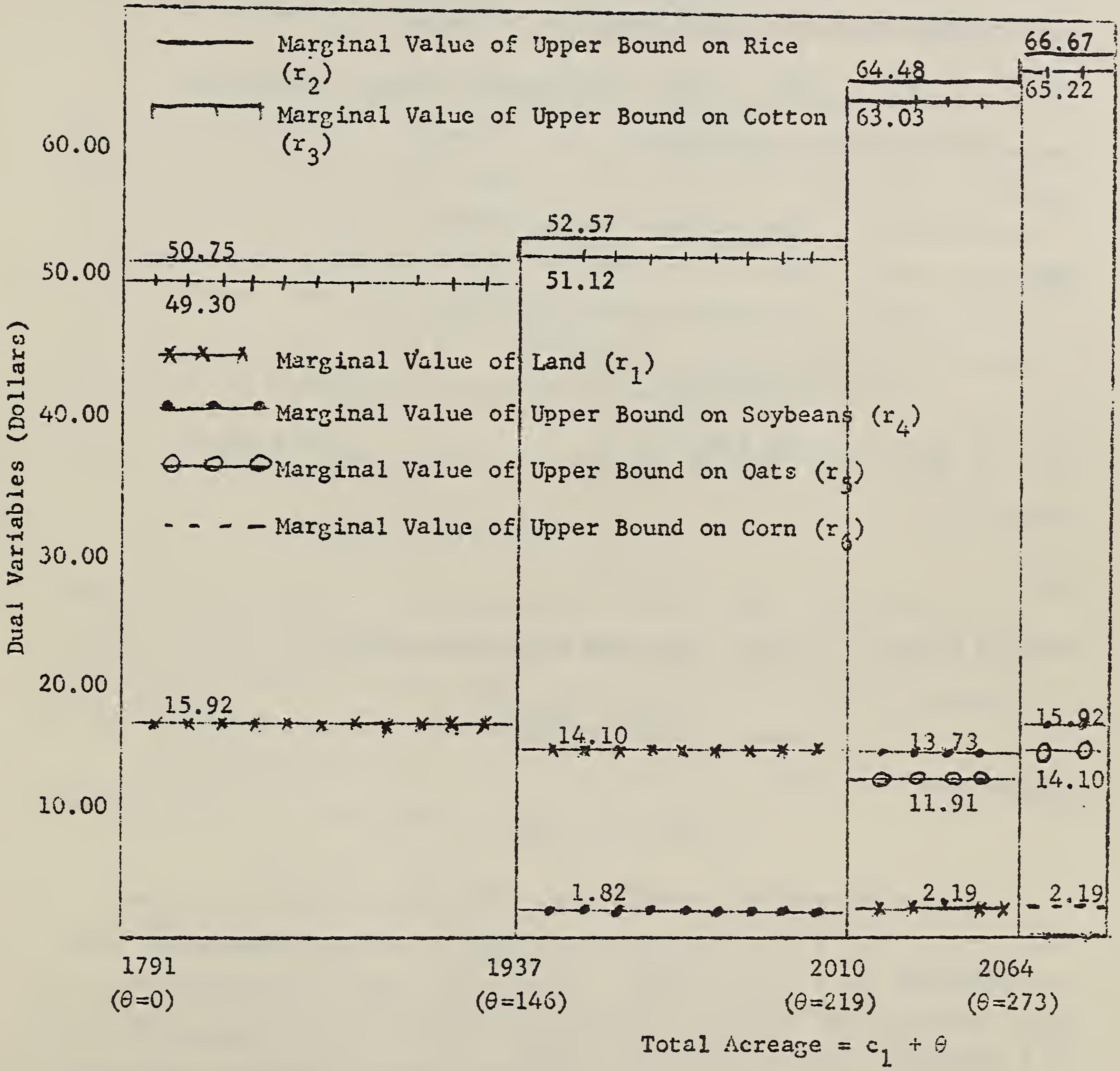


FIGURE 3: OPTIMAL DUAL VARIABLES OF EXAMPLE PROGRAM AS FUNCTIONS OF TOTAL LAND (= $c_1 + \theta$)

c. Parametric Objective Function of Price Mapping (PCR)

We wish now to see how the optimal solution changes as the <u>objective function parameters</u> are changed in some fixed proportion. Let $a_1, \ldots, a_m$ be some constants, positive or negative (or zero). The PCR problem may be written as follows.

$$(32) \quad \pi^* (\emptyset) = \max_{x_1,\ldots,x_m} [(z_1 + \emptyset a_1)x_1 + \ldots + (z_m + \emptyset a_m)x_m]$$

for all $0 \leqq \emptyset \leqq \hat{\emptyset}$, subject to

$$(33) \quad \begin{matrix} b_{11}x_1 + \ldots + b_{1m}x_m \leqq c_1 \\ \vdots \qquad\qquad \vdots \qquad \vdots \\ b_{k1}x_1 + \ldots + b_{km}x_m \leqq c_k. \end{matrix}$$

In matrix notation

$$(34) \quad \pi^* (\emptyset) = \max_x \ (z + \emptyset a)^T x, \qquad 0 \leqq \emptyset \leqq \hat{\emptyset}$$

subject to

$$(35) \quad B\,x \leqq c.$$

where $a = (a_1, \ldots, a_m)^T$. The dual problem is written

$$p^* (\emptyset) = \min_r r^T c$$

subject to

$$B^T r = z + \emptyset a, \quad r \geqq 0.$$

This problem always starts after solving the basic LP problem with $\emptyset = 0$. Then $\emptyset$ is increased continuously. As it increases x^*, and, of course, π^* and r^* will change. Usually the constraints which are tight or loose (the system of equated constraints) will change status as $\emptyset$ increases. As with PLP the changes are made in discrete jumps, or in continuous linear segments--as the case may be. In the latter case linear interpolation between the critical solutions yield the value of any desired solution variable. The value of $\emptyset$ may be infinite. The a-coefficients are sometimes called <u>change coefficients</u> and the vector comprised of them, the <u>change vector</u>.

The primary significance of PCR from the economists point of view, is the fact that it can be used to derive supply functions of the neo-classical variety. This will be illustrated in the next paragraph.

d. Example of Parametric Objective Function of Price Mapping (PCR)

Our original sample problem will continue to serve adequately our need for illustration. It needs to be modified only in the objective function to account for the change coefficients. We shall let the change coefficient for the first process be the negative yield of cotton lint in hundred weight (cwt) per acre, and suppose its value to be -4.00, that is, $a_2 = -4.00$. The remaining change coefficients are assumed equal to zero, $(a_1 = a_3 = a_4 = a_5 = 0)$. Consequently, the parametric objective function, as a function of the change parameter $\emptyset$, is

$$(36) \qquad \pi^*(\emptyset) = \max_{x_1,\ldots,x_5} 66.67\, x_1 + (65.22 - \emptyset\, 4.00)\, x_2 + 15.92\, x_3 + 14.10\, x_4 + 2.19\, x_5$$

subject to the original constraints (4). As we shall see shortly the results obtained by continuously changing $\emptyset$ will yield a neo-classical cotton supply curve. We note that the increase in $\emptyset$ causes a decline in the net return per acre of the cotton producing process z_2.

The solution with $\emptyset = 0$ is, of course, the solution to the original problem, which we already know to be the following:

$$x_1^* = 90, \quad x_2^* = 986, \quad x_3^* = 358, \quad x_4^* = 230, \quad x_5^* = 127.$$

Remembering that the positive increase in $\emptyset$ causes a decline in z_2, the profitableness of producing cotton, we realize that x_1^*, the optimal acreage of rice will not change, for it will remain the most profitable of the five crops. On the other hand, cotton will loose its position as the second ranking crop as soon as $\emptyset$ becomes large enough. Until that point the values of the primal variables will not change, (though the value of the primal solution π^* will). The value of $\emptyset$ that will make $z_2 = z_3$ is easily found by solving the equation:

$$z_2 - \phi\, a_2 = z_3$$

or $$65.22 - \phi\, 4.00 = 15.92$$

for ϕ. Thus $\phi = \frac{z_2 - z_3}{4.00} = 12.325.$

At this point, the solution can be represented by the same dual system of equated constraints as when $\phi = 0$ and the same primal variable values as before. However, equally good is the solution $x_1^* = 90$, $x_2^* = 840$, $x_3^* = 504$, $x_4^* = 230$, $x_5^* = 127$ and the dual system of equated constraints:

$$(37)\quad \begin{array}{lcr} x_1 + x_2 + x_3 + x_4 + x_5 & = & 1791 \\ x_1 & = & 90 \\ x_3 & = & 504 \\ -x_4 & = & -230 \\ -x_5 & = & -127 \end{array}$$

$$(38)\quad \begin{array}{lcl} r_1 + r_2 & = & 66.67 \\ r_1 & = & 15.92 = 65.22 - \phi\, 4.00 \\ r_1 \quad r_4 & = & 15.92 \\ r_1 \quad -r_{10} & = & 14.10 \\ r_1 \quad -r_{11} & = & 2.19 \end{array}$$

As ϕ is increased beyond 12.325 this second system will describe the solution [only the dual variables will change as shown in equation (38)] until ϕ exceeds the value determined by the following equation:

$$z_2 - \phi\, 4.00 = z_4$$

or $$\phi = \frac{z_2 - z_4}{4.00} = 12.78.$$

As before the solution at this point can be represented by the preceding dual system of equated constraints, but also equally well by the additional systems.

$$(39)\quad \begin{array}{lcr} x_1 + x_2 + x_3 + x_4 + x_5 & = & 1791 \\ x_1 & = & 90 \\ x_3 & = & 504 \\ x_4 & = & 303 \\ x_5 & = & -127 \end{array}$$

and

$$
(40)\qquad
\begin{array}{lllll}
r_1 + r_2 & & & = & 66.67 \\
r_1 & & & = & 14.10 \;=\; 65.22 - \not{0}\, 4.00 \\
r_1 & r_4 & & = & 15.92 \\
r_1 & & r_5 & = & 14.10 \\
r_1 & & \quad -r_{11} & = & 2.19
\end{array}
$$

with the primal solution variables of

$$x_1^* = 90, \quad x_2^* = 767, \quad x_3^* = 504, \quad x_4^* = 303, \quad x_5^* = 127.$$

Again $\not{0}$ is increased until it exceeds the value determined by the equations

$$z_2 - \not{0}\, 4.00 = z_5$$

or

$$\not{0} = 15.7575$$

at which point the primal solution is the same as before, or alternatively--and equally as good--$x_1^* = 90$, $x_2^* = 713$, $x_3^* = 504$, $x_4^* = 303$, $x_5^* = 181$ with the dual system of equated constraints:

$$
(41)\qquad
\begin{array}{lllllll}
x_1 + x_2 + x_3 + x_4 + x_5 & & & & = & 1791 \\
x_1 & & & & = & 90 \\
& x_3 & & & = & 504 \\
& & x_4 & & = & 803 \\
& & & x_5 & = & 181
\end{array}
$$

and

$$
(42)\qquad
\begin{array}{lllll}
r_1 + r_2 & & & = & 66.67 \\
r_1 & & & = & 2.19 \;=\; 65.22 - \not{0}\, 4.00 \\
r_1 & r_4 & & = & 15.92 \\
r_1 & & r_5 & = & 14.10 \\
r_1 & & \quad r_6 & = & 2.19.
\end{array}
$$

At this point no process level can be increased further at the expense of the cotton process, because all of the other processes are at their upper bounds. A further decline in the cotton activity therefore must be accompanied by allowing some land to go idle. But this can be optimal only if cotton is produced at a loss. The value of $\not{0}$ beyond which this will occur is found by solving the equation

$$z_2 - \emptyset\, 4.00 = 0.0$$

or

$$\emptyset = 16.305.$$

At this point the preceding dual system of equated constraints will describe the solution, or, equally as well, the systems

$$
(43) \quad
\begin{array}{rcr}
x_1 & = & 90 \\
x_3 & = & 504 \\
x_4 & = & 303 \\
x_5 & = & 181 \\
-x_2 & = & -681
\end{array}
$$

and

$$
(44) \quad
\begin{array}{rcl}
r_1 & = & 66.67 \\
-r_3 & = & 0.00 = 65.22 - \emptyset\, 4.00 \\
r_4 & = & 15.92 \\
r_5 & = & 14.10 \\
r_6 & = & 2.19
\end{array}
$$

with primal solution variables

$$x_1^* = 90, \quad x_2^* = 681, \quad x_3^* = 504, \quad x_4^* = 303, \quad x_5^* = 181.$$

No further change can be made in the system of equated constraints no matter how much larger $\emptyset$ becomes. Therefore, the PLP computations are completed and linear interpolation will produce all variables changing with $\emptyset$.

Let us now see how these computations can be transformed to produce the cotton supply curve. The net-return of cotton per acre (unit process level) is computed as $p_2\, y_2 - c_2 = z_2$ where p_2 is the price of cotton per cwt lint (we neglect the value of cotton seed for simplicity), y_2 the yield of cotton in cwt/acre, and c_2 the per-acre variable cost of producing cotton. Now as we have let a_2, the change coefficient, be the (negative of the) yield of cotton ($a_2 = -4.00$) we could write this expression as $p_2\, a_2 - c_2 = z_2 = p_2\, 4.00 - c_2$.
As a consequence the parametric objective function 36 could be written alternatively

$$\pi^*(\emptyset) = \max_{x_1,\ldots,x_5} \; 66.67\, x_1 + (p_2\, 4.00 - c_2 - \emptyset\, 4.00)x_2$$
$$+ 15.92\, x_3 + 14.10\, x_4 + 2.19\, x_5$$

or

$$\pi^*(\emptyset) = \max_{x_1,\ldots,x_5} \; 66.67\, x_1 + [(p_2-\emptyset)\, 4.00 - c_2]\, x_2 + 15.92\, x_3$$
$$+ 14.10\, x_4 + 2.19\, x_5.$$

We see then that $\emptyset$ plays the role of a price reduction in cotton, so that $p_2 - \emptyset$ is the parametrically modified cotton price. If we suppose that p_2 = \$35.00 and c_2 = \$74.78, then z_2 = 35.00 x 4.00 - 74.78 = 65.22 the original net return for cotton (with $\emptyset = 0$). Now we may write the parametric objective function as,

(45) $$\pi^*(\emptyset) = \max_{x_1,\ldots,x_5} \; 66.67\, x_1 + [(35.00-\emptyset)\, 4.00 - c_2]\, x_2$$
$$+ 15.92\, x_3 + 14.10\, x_4 + 2.19\, x_5.$$

At each value of $\emptyset$ now we can readily calculate the corresponding price of cotton as \$35.00 - $\emptyset$. Thus we have the following tabulation:

$\emptyset$	p_2
0	\$35.00
12.3250	22.68
12.7800	22.22
15.7575	19.24
16.3050	18.70

Our results are now summarized in Tables 5 and 6 and in Figures 4, 5, and 6. Figure 4 shows the optimal value of the parametric program as a function of p_2 (and of θ), Figure 5 shows the derived supply curve of cotton; and Figure 6 shows the values of several of the dual variables as a function of p_2. We may note that negative values of p_2 are meaningless. Therefore, we might have specified $\hat{\emptyset}$ = 35.00, the value of $\emptyset$ beyond which the implied price of cotton ($p_2 - \emptyset$) becomes negative.

Table 5: Summary of Primal Solutions for PCR Example

p_2	\$35.00		22.68		22.22		19.24		18.70	
ϕ	0.00		12.32		12.78		15.76		16.30	
Variable Index j	$z_j = p_j y_j - c_j$	x_j^*	$z_j = p_j y_j - c_j$	x_j^*	$z_j = p_j y_j - c_j$	x_j^*	$z_j = p_j y_j - c_j$	x_j^*	$z_j = p_j y_j - c_j$	x_j^*
1	66.67	90	66.67	90	66.67	90	66.67	90	66.67	90
2	65.22	986	15.92	840	14.10	767	2.19	713	0.00	681
3	15.92	358	15.92	504	15.92	504	15.92	504	15.92	504
4	14.10	230	14.10	230	14.10	303	14.10	303	14.10	303
5	2.19	127	2.19	127	2.19	127	2.19	181	2.19	181
π^*	79,527.71		30,917.91		29,389.11		20,254.14		18,692.67	

Table 6: Summary of Dual Solutions for PCR Example

	Price p_2	\$35.00	22.68	22.22	19.24	18.70
	ϕ	0.00	12.32	12.78	15.76	16.30
Index i	RHS Coefficients c_i	Dual Variables r_i^*	Dual Variables r_i^*	Dual Variables r_i^*	Dual Variables r_i^*	Dual Variables r_i^*
1	1791	15.92	15.92	14.10	2.19	0.00
2	90	50.75	50.75	52.57	64.48	66.67
3	986	49.30	0.00	0.00	0.00	0.00
4	504	0.00	0.00	1.82	13.73	15.92
5	303	0.00	0.00	0.00	11.91	14.10
6	181	0.00	0.00	0.00	0.00	2.19
7	- 74	0.00	0.00	0.00	0.00	0.00
8	-681	0.00	0.00	0.00	0.00	0.00
9	-356	0.00	0.00	0.00	0.00	0.00
10	-230	1.82	1.82	0.00	0.00	0.00
11	-127	13.73	13.73	11.91	0.00	0.00
	ρ^* =	79,527.71	30,917.91	29,389.11	20,254.14	18,692.67

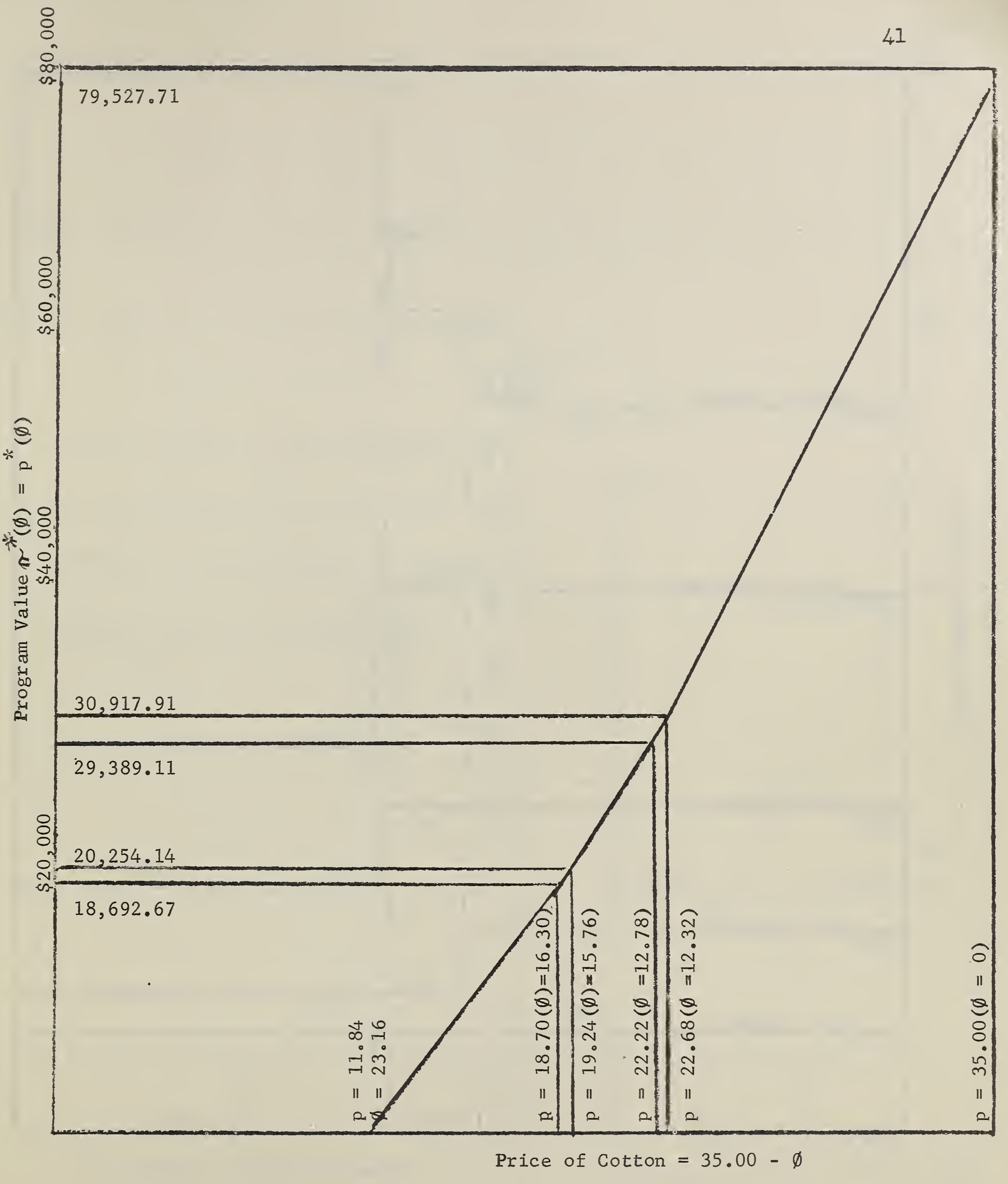


FIGURE 4

OPTIMAL VALUE OF EXAMPLE PROGRAM AS A FUNCTION OF COTTON PRICE (= p_2 - ϕ)

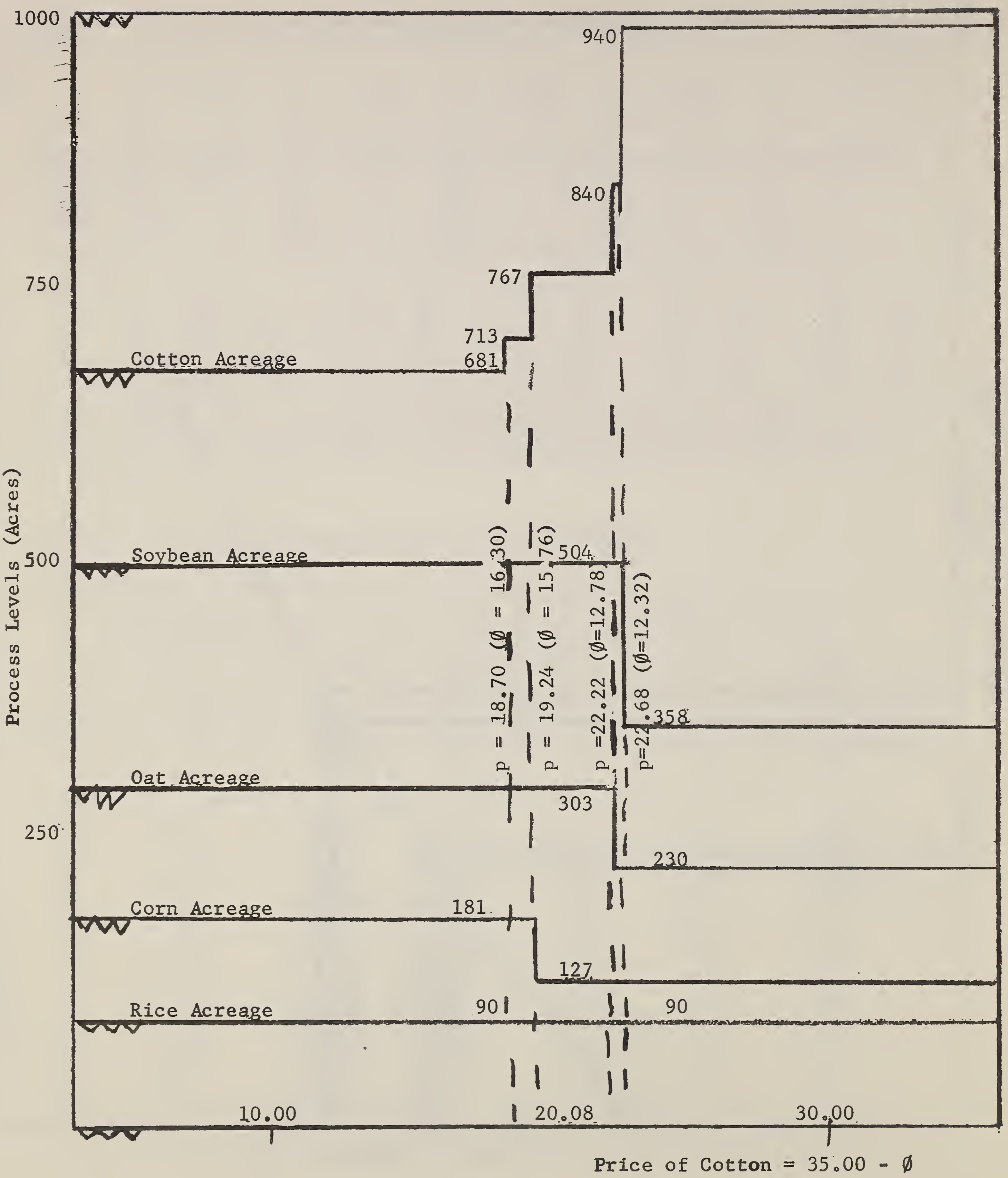


FIGURE 5

OPTIMAL PROCESS LEVELS OF EXAMPLE PROBLEM AS FUNCTIONS OF COTTON PRICE (= p_2 - Ø)

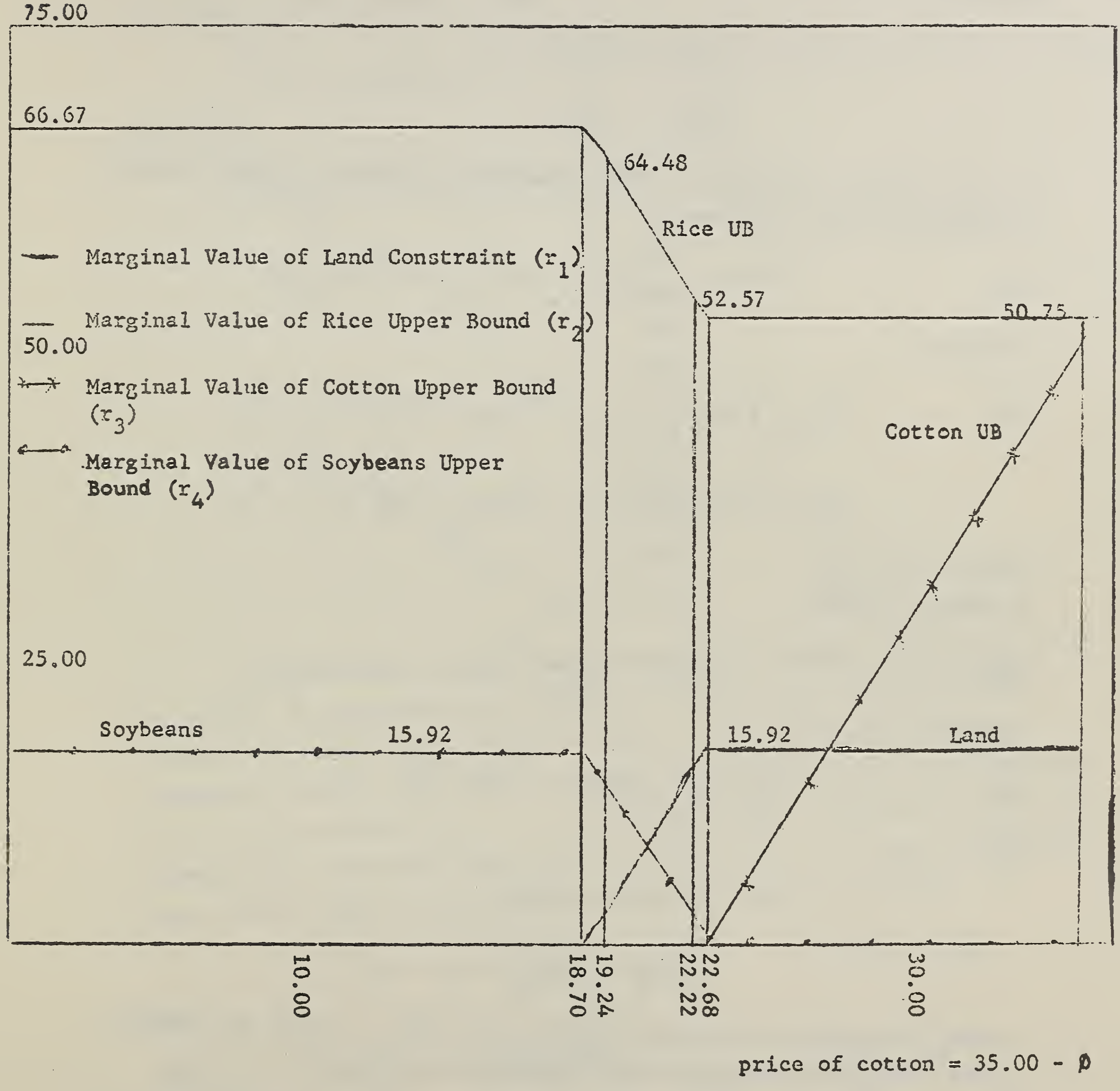


FIGURE 6: OPTIMAL DUAL VARIABLES (MARGINAL VALUES) OF EXAMPLE PROGRAM AS FUNCTIONS OF COTTON PRICE ($= p_2 - \emptyset$)

e. Trigger Analysis on Parametric Constraint Matrix

Here the objective is to determine the effect upon the optimal solution of changing the constraint matrix parameters in some fixed proportions. Let

$$\begin{matrix} h_{11}, & \dots, & h_{1m} \\ \vdots & & \vdots \\ h_{k1}, & \dots, & h_{km} \end{matrix}$$

be k × m change coefficients. The parametric programming problem now under consideration may be written

(46) $$\pi^*(\psi) = \max_{x_1, \dots, x_m} (z_1 x_1 + \dots + z_m x_m)$$

subject to

(47) $$\begin{matrix} (b_{11} + \psi h_{11}) x_1 + \dots + (b_{1m} + \psi h_{1m}) x_m & \leqq & c_1 \\ \vdots \qquad\qquad\qquad\qquad \vdots & \vdots & \vdots \\ (b_{k1} + \psi h_{k1}) x_1 + \dots + (b_{km} + \psi h_{km}) x_m & \leqq & c_k \end{matrix}$$

for all $0 \leqq \psi \leqq \hat{\psi}$.

In matrix notation

(48) $$\pi^*(\psi) = \max_{x} z^T x$$

subject to

(49) $$(B + \psi H) x \leqq c, \qquad 0 \leqq \psi \leqq \hat{\psi}$$

where

$$H = \begin{bmatrix} h_{11}, & \dots, & h_{1m} \\ \vdots & & \vdots \\ h_{k1}, & \dots, & h_{km} \end{bmatrix}.$$

As with the earlier cases this problem begins with $\psi = 0$ and then changes in the solution are recorded as ψ is increased continuously. As it increases, x^*, r^*, and π^* and the dual systems of equated constraints will change, as before in discrete linear segments.

Consider our familiar example from Section 1, now augmented with the change matrix coefficients

$$h_{11} = h_{12} = \dots = h_{1m} = 1.0$$
$$h_{12} = \dots \qquad h_{km} = 0.0.$$

For this case we have the problem

$$\pi^{*}(\psi) = \max_{x_1,\dots,x_5} 66.67\, x_1 + 65.22\, x_2 + 15.92\, x_3 + 14.10\, x_4 + 2.19\, x_5$$

Subject to

$$\begin{array}{lllllll}
(1-\psi)\, x_1 & + \dots + & & & (1-\psi)x_5 & \leqq & 1791 \\
x_1 & & & & & \leqq & 90 \\
& x_2 & & & & \leqq & 986 \\
& & x_3 & & & \leqq & 504 \\
& & & x_4 & & \leqq & 303 \\
& & & & x_5 & \leqq & 181 \\
-x_1 & & & & & \leqq & -74 \\
& -x_2 & & & & \leqq & -681 \\
& & -x_3 & & & \leqq & -356 \\
& & & -x_4 & & \leqq & -230 \\
& & & & -x_5 & \leqq & -127\ .
\end{array}$$

Although this example has no "natural" economic interpretation as it stands, it illustrates the class of problems under consideration. Further, the example can readily be shown to be equivalent to the PLP example of Section 4.b. Thus, dividing both sides of the top constraint by $(1 - \psi)$ we obtain $x_1 + \dots + x_5 \leqq (\frac{1}{1 - \psi})\, 1791$. Knowing (from the PLP example) that there are four critical values of θ to be found (including zero) we can find the equivalent values of ψ from the equation

$$1791 + \theta_i = (\frac{1}{1 - \psi})\, 1791 \qquad i = 1, \dots, 4.$$

Thus

(50) $$\psi_i = \frac{\theta_i}{1791 + \theta_i} \qquad i = 1, \dots, 4.$$

By substituting the θ values of Tables 3 and 4 into the above expression the PLP example could be transformed to the Trigger Analysis problem of this section.

The conversion of PLP to Trigger Analysis and vice versa is a very special property of our particular examples. In general they would be quite distinct in character. While this type of parametric programming is of less general interest, and certainly much less widely practiced than the others, we mention it here for completeness, and because for certain problems of economic analysis--such as the analysis of changes in technical efficiency--it is of immediate relevance.

It might also be mentioned in passing that one could consider parametric LP problems in which all three sets of data changed as linear functions of a single parameter, or in which combinations of two sets of data changed as linear function of a single parameter. An exegesis of these more general problems should not be needed by the reader. In any case they would be relevant only to highly specialized applications.

5. SLACK VARIABLES AND BASES

a. Slack Variables and Disposal Activities

The primal LP problem can be transformed into an alternative form which is particularly instructive when one faces the problem of computing solutions to arbitrary LP problems. We proceed by introducing (up to) k dummy variables $y_1, \ldots, y_k$ and converting the inequality expressions 2 (or 7) into equation constraints. Thus, we have, instead of 1 and 2, the problem

(51) $$\pi^* = \max z_1 x_1 + \ldots + z_m x_m$$

subject to

(52) $$\begin{array}{lll} b_1 x_1 + \ldots + b_{1m} x_m + y_1 & = c_1, & x_1 \geqq 0, \; y_1 \geqq 0 \\ \vdots & \vdots & \vdots \\ b_{k1} x_1 + \quad + b_{km} x_m \quad + y_k & = c_k, & x_m \geqq 0, \; y_k \geqq 0. \end{array}$$

(Notice that there are now k equations in m + k unknowns.)

In matrix notation the converted problem is,

(53) $$\pi^* = \max_x z^T x$$

subject to

(54) $$Bx + Iy = [B, I] \binom{x}{y} = c, \qquad \binom{x}{y} \geqq 0.$$

(We shall in the sequel refer to the j^{th} column of the matrix [B, I] as P_j.) The dummy variables $y_1, \ldots, y_k$ are frequently called slack or disposal variables. The columns of the I matrix appearing in expression (54) are frequently called disposal activities.

b. Relation of Slack Variables to the "Loose" Constraints

The slack variables indicate the amount by which each inequality constraint is unsatisfied as an equality, and, in the case in which constraint involves an economic input, the amount of the input not used in the production process. Thus, if $x_1^*, \ldots, x_m^*$ are the optimal primal solution variables then

$$
\begin{aligned}
y_1^* &= c_1 - b_1 x_1^* - \ldots - b_{1m} x_m^* \\
&\vdots \\
y_k^* &= c_k - b_{k1} x_1^* - \ldots - b_{km} x_m^*
\end{aligned}
\tag{55}
$$

are the optimal slack variables.

From the discussion of Section 3.d and e, we see an immediate correspondence between the zero or nonzero status of the slack variables and the "looseness" or "tightness" of a constraint. Thus, if the i^{th} constraint is tight, $y_i^* = 0$; if the i^{th} constraint is loose, $y_i^* > 0$.

c. Optimal Bases and Dual Systems of Equated Constraints

An arbitrary non-zero vector of k dimensions, say $c = (c_1, \ldots, c_k)^T$ can always be expressed as a linear combination of k (linearly independent) vectors.[10] Thus the RHS vector c can be expressed as a linear combination of k processes and disposal activities. Such a set of vectors is called a basis. Let $P_1, \ldots, P_k$ be a set of k such columns. Then we may write $c = x_1 P_1 + \ldots + x_k P_k$ or in vector notation $Px = c$. Corresponding to the optimal solution of an LP problem is an optimal basis, and as will

---

[10] A set of k vectors, say $a_1, \ldots, a_k$, is said to be linearly independent, if and only if, the vector equation $\lambda_1 a_1 + \ldots + \lambda_k a_k = 0$, where $\lambda_1, \ldots, \lambda_k$ are scalars, implies $\lambda_1 = \lambda_2 = \ldots = \lambda_k = 0$.

be immediately evident, there is a one-to-one correspondence between optimal bases and the dual systems of equated constraints.

Returning (as always) to our original example, we find that the RHS vector may be written as follows:

$$\begin{bmatrix} 1791 \\ 90 \\ 986 \\ 504 \\ 303 \\ 181 \\ -74 \\ -681 \\ -356 \\ -230 \\ -127 \end{bmatrix} = 90 \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 986 \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 358 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \end{bmatrix} + 230 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix} + 127 \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$

$$+ 146 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 73 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 54 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 16 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 305 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

This may be readily verified by the reader. The first five vectors are the columns of the B matrix that correspond to the positive primal variables (all of them in this example). The next six columns are the disposal activities corresponding to the rows that are <u>not</u> equated. The scalars by which these vectors are multiplied are respectively the positive primal variables and the non-zero slack or disposal variables. The vectors shown are the optimal basis.

Our reliable example has again illustrated a general principle. We remember now that

M_1 = the set of M column indexes corresponding to the non-zero primal variables

$M_2 = M - M_1$ = the set of M'' column indexes corresponding to the zero primal variables

K_1 = the set of k' row indexes corresponding to the tight primal constraints (with non-zero dual variables)

$K_2 = K - K_1$ = the set of k" row indexes corresponding to the loose (or linearly dependent) tight constraints (with zero dual variables).

The basic matrix P bears a definite relation to the equated constraint matrix B'. Let us form the matrix P^* from the matrix P by placing the rows with indexes in K_1 at the top and those with indexes in K_2 at the bottom. That is, we place all equated constraint rows at the top of the matrix. P^* can then be written

$$P^* = \begin{array}{c} \begin{matrix} M_1 & K_2 \end{matrix} \\ \left[\begin{array}{c|c} P_{11} & P_{12} \\ \hline P_{21} & P_{22} \end{array}\right] \end{array} \begin{array}{l} \\ K_1 = M_1 \\ K_2 \end{array}$$

The dimensions of these submatrices are as follows:

Submatrix	Dimension
P_{11}	$m' \times m'$
P_{12}	$m' \times k''$
P_{21}	$k'' \times m'$
P_{22}	$k'' \times k''$.

The elements of P_{11} are p_{ij} $i \in K_1$, $j \in M_1$ so that $P_{11} = B'$ (the equated constraint matrix), $P_{21} = B''$, $P_{12} = 0$ and $P_{22} = I_{k''}$ is a $k'' \times k''$ identity matrix. Thus,

$$(56) \qquad P^* = \begin{array}{c} \begin{matrix} m' & k'' \end{matrix} \\ \left[\begin{array}{c|c} B' & 0 \\ \hline B'' & I \end{array}\right] \end{array} \begin{array}{l} \\ m' \\ k'' \end{array}$$

There are k disposal variables with indexes $K = \{1, \ldots, k\}$. Thus, the optimal basis is composed of the m' vectors P_i, $i \in M_1$ and the disposal activities P_j, $j \in K_2$ where "$\in$" means "belongs to," and where I_j is the j^{th} column of the k-dimensioned identity matrix I. The RHS vectors may now be written as

$$(57) \qquad \sum_{j \in M_1} P_j x_j + \sum_{i \in K_2} P_i y_i = c$$

where "$\sum_{j \in M_1}$" means summing over the set of indexes M_1 and similarly for "$\sum_{i \in K_2}$."

If now we define P to be a $k \times k$ matrix comprised of the vectors, P_j, $j \in M_1$, and P_j, $j \in K_2$ then we have

$$(58) \qquad \begin{bmatrix} x' \\ y'' \end{bmatrix} = P^{-1} c$$

where x' is comprised of elements of x with indexes in M_1 and y'' is composed of elements of y with indexes in K_2. Therefore, we have

$$(59) \qquad \begin{bmatrix} x' \\ y'' \end{bmatrix} = P^{*-1} \begin{bmatrix} c' \\ c'' \end{bmatrix} = \left[\begin{array}{c|c} B'^{-1} & 0 \\ \hline B'' B'^{-1} & I \end{array}\right] \begin{bmatrix} c' \\ c'' \end{bmatrix}.$$

It turns out that the form (57) or alternatively (58) is computational the most useful but (59) shows that an optimal basis is equivalent to a nonsingular dual system of equated constraints.

d. A Remark in the Simplex Method

The art of computing solutions to LP problem is a highly developed branch of applied mathematics. From the economists point of view, it is not of crucial interest, for--as we have seen--the theory (and economic sense) of linear programming can be understood without any knowledge of it. And it has been possible to illustrate the theory with a model so simple that a computation method is intuitively obvious. While the solution of general LP problem of any size is far from obvious, it really need not concern the user of USDA LP/90. The latter has been designed by specialists and it has been thoroughly tested in a great variety of problems. All we need is a sufficient understanding of our own LP problem to properly prepare the data and to interpret LP/90's output. We can always check the solution to make sure that the program has done its job: that we know how to do from the theorems of Section 3.b.

There is an additional reason for omitting a detailed discussion of computation technique, and that is simply because there are many excellent discussions of it readily available. However, before passing to other subjects we may make one or two comments that will relate the best known computation procedure to the discussion so far. The latter is the justly famous simplex method and its variants.

This method is comprised of two parts: first, that of finding a feasible solution and second, that of finding an optimal solution. It proceeds by finding a basis, such that

$$\sum_{j \in M_1} P_j^o x_j^o + \sum_{j \in K_2} P_j^o y_j^o = P^o$$

where P_o is set equal to c and such that $x_j^o, y_j^o > 0$ for all j in M_1 and K_2 respectively. This is a feasible solution and the P_j^o columns are the corresponding feasible basis. Next the method proceeds to find basis after basis

$$P^o, P^1, \ldots, P^i$$

where P^i is the matrix of basis vectors in the i^{th} basis, until the optimal basis is reached. The procedure by which a given basis is derived from the proceeding basis leads to feasible solutions with successively larger values of the program value (π^*), until, after a finite number of computations, the optimal basis is found. Each successive basis is obtained by dropping one of the columns already in the basis and replacing it with one not previously in it. The simplex and related methods are techniques for choosing efficiently the vectors to be dropped from and added to the basis and for inverting the resulting basis matrix to obtain the successive feasible solution vectors.

6. RECURSIVE LINEAR PROGRAMMING (RLP)

a. The Basic Idea

Recursive linear programming (RLP) is a sequence of LP problems in which the objective function, constraint matrix, and/or the right hand side parameters depend upon the primal and/or dual solution variables of the preceding LP problem (or problems) in the sequence. In current agricultural applications[1, 2,3,10],each problem in the sequence is associated with a given production year, and its solution is interpreted as a "prediction" (or explanation) of the given years' expected production, income, costs and input utilization. Consequently, the model describes how current plans are related to past expectations and performance. In addition to a dependence on preceding LP solutions, the current problem data may depend on various exogenous or predetermined variables. One might summarize the meaning of this class of models as the description of optimizing over a limited time horizon on the basis of knowledge gained from past experience. They express the manner in which economic plans are reformulated as each period's experience is accumulated.

Formulated with this degree of generality RLP is a bit clumsy to express mathematically. Instead in what follows we shall describe several special cases of RLP.

b. Type I RLP: Recursive RHS, Diagonal Primal Dependence

This model can be expressed as follows

(60) $$\pi^*(t) = \max_{x_1(t),\ldots,x_m(t)} \hat{z}_1(t)\, x_1(t) + \ldots + \hat{z}_m(t)\, x_m(t)$$

subject to

$$\begin{array}{ccc} b_{11}\, x_1\,(t) + \ldots + b_{1m}\, x_m\,(t) & \leqq & c_1\,(t) \\ \vdots & & \vdots \\ b_{k1}\, x_1\,(t) + \ldots + b_{km}\, x_m\,(t) & \leqq & c_k\,(t) \end{array}$$

where

(62) $$\begin{array}{l} c_1\,(t) = \gamma_1\,[b_{11}\, x_1^*(t-1) + \ldots + b_{1m}\, x_m^*(t-1)] + u_1\,(t) \\ \vdots \\ c_k\,(t) = \gamma_k\,[b_{k1}\, x_1^*(t-1) + \ldots + b_{km}\, x_m^*(t-1)] + u_k\,(t) \end{array}$$

where $u_1(t)$, ..., $u_k(t)$ are "exogenous" variables (i.e., are not recursively generated). In matrix notation,

(63) $$\pi^*\,(t) = \max_{x\,(t)}\ \hat{z}\,(t)^T\, x\,(t)$$

subject to

(64) $$Bx(t) \leqq c(t)$$

where

(65) $$c(t) = \Gamma\ Bx^*(t-1) + u(t),$$

where $u(t) = [u_1(t), \ldots, u_k(t)]^T$, where $\hat{z}(t) = [\hat{z}_1(t), \ldots, z_m(t)]$ and finally where

$$\Gamma = \begin{bmatrix} \gamma_1 & 0 \\ 0 & \gamma_k \end{bmatrix}$$

Relation (64) and (65) can be recombined to show more clearly the recursive structure, thus

(64') $$Bx^*(t) \leqq \Gamma\ Bx^*(t-1) + u(t)$$

The objective function parameters $\hat{z}_i(t)$ have the "hat" to indicate that they involve expected net returns (gross profits) rather than realized gross profits. From the econometric point of view it is convenient (and often a close representation of actual fact) to represent these variables as dependent entirely upon their past values. Thus, for example, we might have

(66) $$\hat{z}(t) = \lambda\ z(t-1) + \lambda^2\ z(t-2) + \ldots + \lambda^h\ z(t-h)$$

where

(63') $$\sum_{i=1}^{h} \lambda^i = 1.$$

The simplest example of such a net-return expectation model is when $\lambda = 1$ and $h = 1$. Then, $\hat{z}(t) = z(t-1)$. This is in fact the case that has been studied to date. However, current research includes the study of models with $h = 3$. In general, we could write the case under consideration; using (64') and (66) as

$$\underline{\pi}^*(t) = \max_{x(t)} \left[\sum_{i=1}^{h} \lambda^i\ z(t-i)\right]^T x(t)$$

subject to

(64') $$Bx(t) \leqq \Gamma\ Bx^*(t-1) + u(t).$$

Diagonality of Γ explains the name for this case. While the $z(t)$ vector depends recursively on past (realized) values of net-returns, they are exogenously determined (and predetermined) in the sense that they do not depend on the lagged values of the solution variables $x^*(t)$ and $r^*(t)$. In addition to these predetermined variables the u(t) vectors exert an exogenous influence on the RHS elements. In some cases the γ_i may equal zero in which case the corresponding RHS element would be determined in a completely exogenous manner.

We must now inquire as to how such a model may be started up. The answer is, of course, the same as for other classes of dynamic models. One finds <u>initial conditions</u> from which a given year's data can be derived, and from which succeeding solutions may proceed. There are two alternative forms in which the initial data for the case being studied may be cast, depending on whether one uses the form (63), (64), (65) or, alternatively, (63'), (64'). We shall assume that--either way--$\hat{z}(t)$ is obtained from (66).

Consider the first form: for some base year (say for t = 1) we must know (or have estimates of) c(1), z(0), z(-1), ..., z(1-h). Then the first LP problem in the sequence can be solved, giving $x^*(1) = [x_1^*(1), \ldots, x_m^*(1)]^T$. Then this last solution vector can be plugged in the expression (65), giving the RHS for the next year, t = 2. This process is repeated as long as one has data for z(t) and $u(t) = [u_1(t), \ldots, u_k(t)]^T$. The latter (as noted already) are values which do not depend on the preceding year's solution.

In the second form we again need to know or have estimates of z(0), z(-1), ..., z(1-h), but now, instead of c(1) we presuppose a knowledge (or estimate) of $x^*(0)$, that is, an initial vector of process levels. Using (66) the former data give $\hat{z}(1)$. By plugging the latter into (62) [or alternatively (65) or (64')], gives c(1) so that $x^*(1)$ may be obtained as the solution to the resulting LP problem. Now this vector may be inserted into (62) or (65) again, giving the RHS for period 2, c(2), and so on, until the exogenous information is exhausted.

c. An Example of Type I RLP: Process and Flexibility Adjustment

The tried and true example with which the illustrations of LP began can be readily generalized to serve as an example of Type I RLP. Elsewhere, that simple model has been referred to as the Henderson Model and its generalization to Type I RLP, as the Dynamic Henderson Model [2]. Table 7 contains the initial conditions and the γ_i coefficients while Table 8 contains the objective function variables. Finally we assume that $c(1) = \ldots = c(6) = 1761$. That is, we assume the supply of land remains constant throughout.

Using a notation analogous to that of Section 1.b the model may be written

$$\pi^*(t) = \max_{x_1(t),\ldots,x_5(t)} z_1(t-1)\, x_1(t) + \ldots + z_5(t-1)\, x_5(t) \qquad t = 1, \ldots, 6. \tag{67}$$

$$\begin{aligned}
x_1(t) + x_2(t) + x_3(t) + x_4(t) + x_5(t) &\leqq 1791 \\
x_1(t) &\leqq 1.13\, x_1^*(t-1) \\
x_2(t) &\leqq 1.13\, x_2^*(t-1) \\
x_3(t) &\leqq 1.20\, x_3^*(t-1) \\
x_4(t) &\leqq 1.16\, x_4^*(t-1) \\
x_5(t) &\leqq 1.20\, x_5^*(t-1) \\
-x_1(t) &\leqq -.92\, x_1^*(t-1) \\
-x_2(t) &\leqq -.78\, x_2^*(t-1) \\
-x_3(t) &\leqq -.84\, x_3^*(t-1) \\
-x_4(t) &\leqq -.87\, x_4^*(t-1) \\
-x_5(t) &\leqq -.84\, x_5^*(t-1).
\end{aligned} \tag{68}$$

Now under the assumption that we know $c(1)$, as shown in Table 7, and using $z(0)$ from Table 8 we have an LP problem exactly the same as that in Section 1.b with exactly the same solution. That solution (now augmenting it with the time variable) is

$$\begin{aligned}
x_1^*(1) &= 90 \\
x_2^*(1) &= 986 \\
x_3^*(1) &= 358 \\
x_4^*(1) &= 230 \\
x_5^*(1) &= 127.
\end{aligned}$$

Table 7: Additional Parameters for Type I RLP Example

Row Index i	Gamma Coefficients (γ_i)	RHS Coefficients $c_i(0)$
1	0	1791
2	1.30	90
3	1.13	986
4	1.20	504
5	1.16	303
6	1.20	181
7	.92	- 74
8	.78	-681
9	.84	-356
10	.87	-230
11	.84	-127

Table 8: Objective Function Coefficients for Type I RLP Example

Year	$z_1(t)$	$z_2(t)$	$z_3(t)$	$z_4(t)$	$z_5(t)$
0	66.67	65.22	15.92	14.10	2.19
1	64.00	68.00	13.00	15.00	4.00
2	63.00	69.00	12.00	16.00	6.00
3	65.00	66.00	14.00	15.00	5.00
4	68.00	64.00	16.00	14.00	3.00
5	69.00	63.00	17.00	13.00	2.00

Plugging these values into the right hand part of (68) we find c(2). This gives the right hand side shown in Table 9 from which the new LP solution $x^*(2)$ is derived. The latter, shown in Table 10, is derived in the same manner as in all the preceding examples, because, for a given year, the model is an LP problem of exactly the same kind. Table 9 gives the recursively generated RHS's and Table 10 the primal solution variables for t = 1, ..., 6. Notice that because the exogenous net-return information Table 8) is available only up to t = 5, the model can be solved only up to t = 6. In short, the model under consideration is a one year forecaster.

Figure 7 represents the progress of the primal variables through time.

d. Type II RLP: Recursive RHS, Linear Primal Dependence

This model differs from its Type I relative in two regards, first, the RHS dependence on the preceding primal solution is a general linear transformation; second, the linear dependence is extended to the slack variables. Relations (60) and (61) are still applicable but should be rewritten as

(69)
$$\begin{aligned} c_1(t) &= \phi_{11}\, x_1^*(t-1) + \dots + \phi_{1m}\, x_m^*(t-1) + \phi_{1,m+1}\, y_1^*(t-1) + \dots \\ &\qquad + \phi_{1m,m+k}\, y_k^*(t-1) + u_1(t) \\ &\vdots \\ c_k(t) &= \phi_{k1}\, x_1^*(t-1) + \dots + \phi_{km}\, x_m^*(t-1) + \phi_{k,m+1}\, y_1^*(t-1) + \dots \\ &\qquad + \phi_{k,m+k}\, y_k^*(t-1) + u_k(t). \end{aligned}$$

Here, $y_1^*(t-1)$, ..., $y_k^*(t-1)$ are the values of the proceeding year's optimal slack variables. If we recall that

$$\begin{aligned} y_1^*(t) &= c_1(t) - [b_{11}\, x_1^*(t) + \dots + b_{1m}\, x_m^*(t)] \\ &\vdots \\ y_k^*(t) &= c_k(t) - [b_{k1}\, x_1^*(t) + \dots + b_{km}\, x_m^*(t)] \end{aligned}$$

then (69) can be re-expressed as

$$c_1(t) = \alpha_{11}x_1^*(t-1) + \ldots + \alpha_{1m}x_m^*(t-1) + \alpha_{1,m+1}c_1(t-1) + \ldots + \alpha_{1,m+k}c_k(t-1)$$

(70) $\vdots \qquad \vdots$

$$c_k(t) = \alpha_{k1}x_1^*(t-1) + \ldots + \alpha_{km}x_m^*(t-1) + \alpha_{k,m+1}c_1(t-1) + \ldots + \alpha_{k,m+k}c_k(t-1)$$

where $\alpha_{ij} = \emptyset_{ij} - \sum_{i=1}^{k} \emptyset_{i,m+1}\, b_{ij}$ $\qquad i = 1, \ldots, k_i$

$\qquad j = 1, \ldots, m$

and where $\alpha_{i,m+j} = \emptyset_{i,m+j}$ $\qquad i = 1, \ldots, i$

$\qquad j = 1, \ldots, k.$

In matrix notation (69) can be reduced to the following

(71) $$c(t) = \Phi' \, x^*(t-1) + \Phi'' \, y^*(t-1) + u(t)$$

when $\Phi = \begin{bmatrix} \emptyset_{11}, & \ldots, & \emptyset_{1m} \\ \vdots & & \vdots \\ \emptyset_{k1}, & \ldots, & \emptyset_{km} \end{bmatrix}$ and $\Phi'' = \begin{bmatrix} \emptyset_{1,m+1}, & \ldots, & \emptyset_{1,m+k} \\ \vdots & & \vdots \\ \emptyset_{k,m+1}, & \ldots, & \emptyset_{k,m+k} \end{bmatrix}$

and (70) to

(72) $$c(t) = \Omega \, x^*(t-1) + \Phi'' \, c(t-1) + u(t)$$

where

(73) $$\Omega = \Phi' - \Phi'' B = \begin{bmatrix} \alpha_{11}, & \ldots, & \alpha_{1m} \\ \vdots & & \vdots \\ \alpha_{k1}, & \ldots, & \alpha_{k1} \end{bmatrix} .$$

As in the Type I case, the objective function for Type II models would ordinarily be derived from an expression like (66).

e. An Example of Type II RLP: Investment and Depreciation

The significance of Type II RLP can be best illustrated by a simple investment model extracted from our familiar example in its original form. Let us retain the first, overall land restriction and the five upper bounds on each crop. Instead of interpreting these upper bounds

Table 9: Recursively Generated Right Hand Sides for Type I RLP Example

Row Index i	$c_i(1)^*$	$c_i(2)^{**}$	$c_i(3)^{**}$	$c_i(4)^{**}$	$c_i(5)^{**}$	$c_i(6)^{**}$
1	1791	1791	1791	1791	1791	1791
2	90	117	108	99	91	118
3	986	1114	1245	1357	1451	1502
4	504	605	359	300	252	211
5	303	351	323	202	175	152
6	181	217	128	108	91	77
7	- 74	- 83	- 76	- 70	- 64	- 84
8	-681	-779	-860	-937	-1001	-1037
9	-356	-299	-250	-210	-176	-148
10	-230	-200	-174	-151	-131	-114
11	-127	-107	- 90	- 76	- 64	- 54

*Note that this column gives the initial conditions.

**RHS elements rounded to nearest integer.

Table 10: Primal Solutions for Type I RLP Example

Year (t)	$x_1^*(t)$	$x_2^*(t)$	$x_3^*(t)$	$x_4^*(t)$	$x_5^*(t)$
1	90	986	358	230	127
2	83	1102	299	200	107
3	76	1201	250	174	90
4	70	1284	210	151	76
5	91	1329	176	131	64
6	118	1357	148	114	54

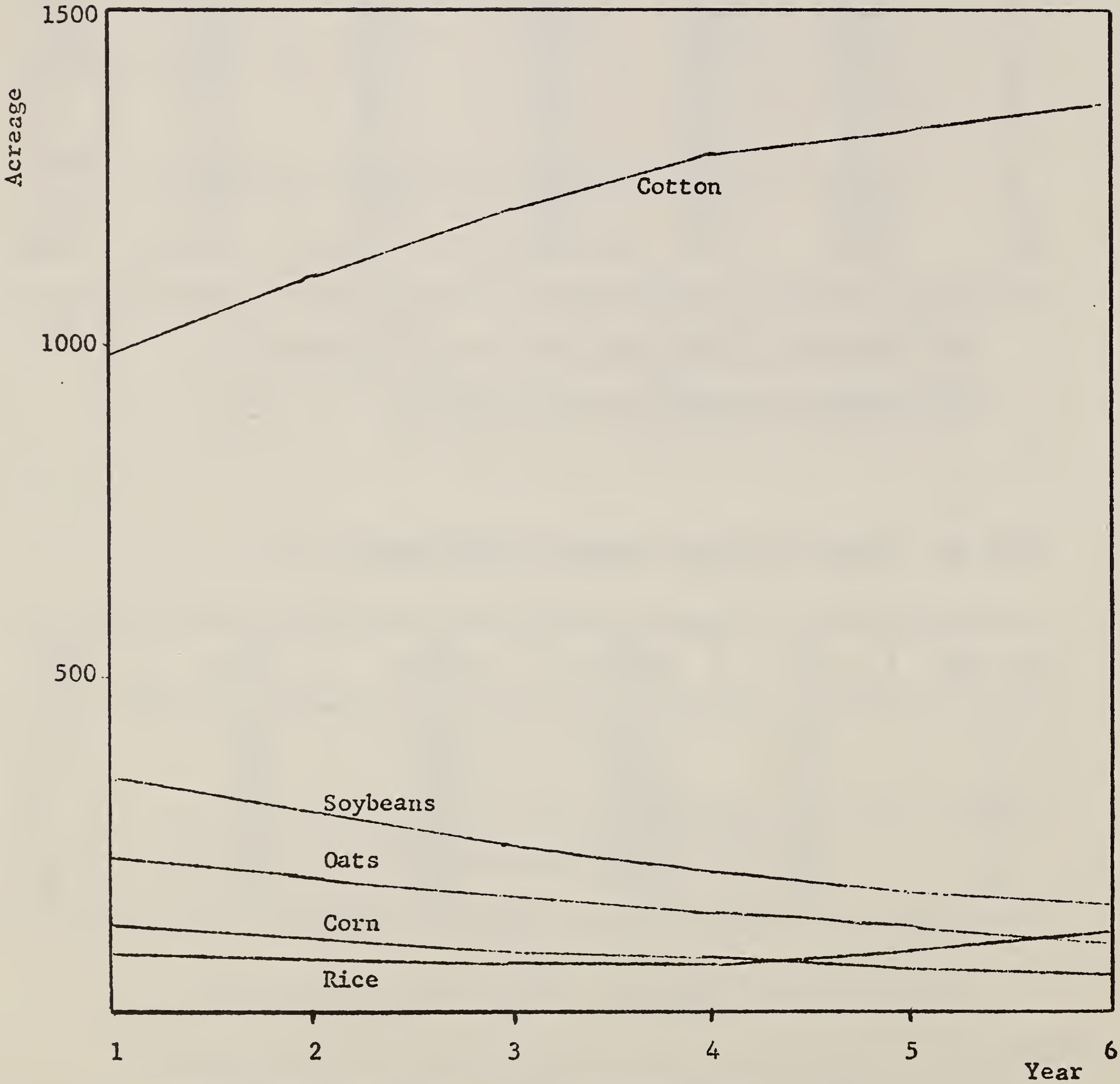


FIGURE 7: TYPE I RLP EXAMPLE: PROGRESS OF THE PRIMAL VARIABLES THROUGH TIME

as flexibility constraints, let us consider them to be capacity constraints, so that $c_2(t)$ is capacity in acres of producing rice, ..., $c_6(t)$ is capacity in acres of producing corn. Let us not worry in this exercise about the problem of defining and measuring capacity, and further let us drop the remaining five "lower" flexible constraints. Finally, let us set $\hat{z}_1(t) = z_1(t-1)$, ..., $\hat{z}_5(t) = z_5(t-1)$, i.e., the expected net return coefficient equal to lagged net returns, and use as data the items shown in Table 8. With this treatment our objective function is the same as (67).

Table 11: PHI Coefficients for Type II RLP Example

i	$\phi_{i+1,i}$	$\phi_{i+1,\ 5+i}$
1	.5	.9
2	.2	.9
3	.5	.9
4	.5	.9
5	.2	.9

Our model, written out in full is

$$\max z_1(t-1)\, x_1(t) + \ldots + z_5(t-1)\, x_5(t)$$

subject to

$$\begin{aligned}
x_1(t) + x_2(t) + x_3(t) + x_4(t) + x_5(t) &\leqq 1791 \\
x_1(t) &\leqq \phi_{21}x_1^*(t-1) + \phi_{26}y_1^*(t-1) \\
x_2(t) &\leqq \phi_{32}x_1^*(t-1) + \phi_{37}y_2^*(t-1) \\
x_3(t) &\leqq \phi_{43}x_3^*(t-1) + \phi_{48}y_3^*(t-1) \\
x_4(t) &\leqq \phi_{54}x_4^*(t-1) + \phi_{59}y_4^*(t-1) \\
x_5(t) &\leqq \phi_{65}x_5^*(t-1) + \phi_{6,10}y_5^*(t-1)
\end{aligned}$$

Using the data of Table 11, the constraints are as follows [expressed in the form (72)].

$$
(74) \quad
\begin{aligned}
x_1(t) + \ldots + x_5(t) &\leqq 1791 \\
x_1(t) &\leqq .5\, x_1^*(t-1) + .9\, c_1(t-1) \\
x_2(t) &\leqq .2\, x_2^*(t-1) + .9\, c_2(t-1) \\
x_3(t) &\leqq .5\, x_3^*(t-1) + .9\, c_3(t-1) \\
x_4(t) &\leqq .5\, x_4^*(t-1) + .9\, c_4(t-1) \\
x_5(t) &\leqq .2\, x_5^*(t-1) + .9\, c_5(t-1)\ .
\end{aligned}
$$

These constraints state, in turn, (i) the total amount of land used in any year must be less than 1791 acres, the total available, (ii) the capacity used of rice (in acres) cannot be greater than a 50% increase over that actually utilized the preceding year plus ninety per cent of the preceding year's capacity, and so on. We are saying, that new investment is related to the actual amount of capacity used the preceding year, plus the amount of capacity available the preceding year allowing for 10% depreciation. The coefficient in the first column of Table 11 thus gives a gross investment coefficient, while that in the second column gives one minus a depreciation coefficient (the latter being 10%). In the matrix form of relation (72) we would have

$$
B = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \qquad
\Omega = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ .5 & 0 & 0 & 0 & 0 \\ 0 & .2 & 0 & 0 & 0 \\ 0 & 0 & .5 & 0 & 0 \\ 0 & 0 & 0 & .5 & 0 \\ 0 & 0 & 0 & 0 & .2 \end{bmatrix},
$$

$$
\Phi'' = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ .9 & 0 & 0 & 0 & 0 \\ 0 & .9 & 0 & 0 & 0 \\ 0 & 0 & .9 & 0 & 0 \\ 0 & 0 & 0 & .9 & 0 \\ 0 & 0 & 0 & 0 & .9 \end{bmatrix}, \quad
u(t) = \begin{bmatrix} 1791 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

and, noting that from (73), we have $\Phi' = \Omega + \Phi'' B$, then

$$
\Phi' = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1.4 & 0 & 0 & 0 & 0 \\ 0 & 1.1 & 0 & 0 & 0 \\ 0 & 0 & 1.4 & 0 & 0 \\ 0 & 0 & 0 & 1.4 & 0 \\ 0 & 0 & 0 & 0 & 1.1 \end{bmatrix}
$$

and Φ'', B, and $u(t)$ are as shown for the relation (72) version of the constraint

Let us suppose that we start with Initial Condition as follows

$$c(0) = \begin{bmatrix} 1791 \\ 90 \\ 986 \\ 504 \\ 303 \\ 181 \end{bmatrix} \qquad x^*(0) = \begin{bmatrix} 90 \\ 986 \\ 358 \\ 230 \\ 127 \end{bmatrix}$$

Using the right side of the relation in (74) we find that

$$c_1(1) = 1791$$
$$c_2(1) = .5\ (90) + .9\ (90) = 126$$
$$c_3(1) = .2\ (986) + .9\ (986) = 1085$$
$$c_4(1) = .5\ (358) + .9\ (504) = 633$$
$$c_5(1) = .5\ (230) + .9\ (303) = 388$$
$$c_6(1) = .2\ (127) + .9\ (181) = 188$$

Using $\hat{z}(1) = z(0)$ from Table 8 as the objective function we now obtain the optimal solution

$$x_1^*(1) = 126$$
$$x_2^*(1) = 1085$$
$$x_3^*(1) = 580$$
$$x_4^*(1) = 0$$
$$x_5^*(1) = 0.$$

Next we find that

$$c_1(2) = 1791$$
$$c_2(2) = .5\ (126) + .9\ (126) = 176$$
$$c_3(2) = .2(1085) + .9\ (1085) = 1194$$
$$c_4(2) = .5\ (580) + .9\ (633) = 850$$
$$c_5(2) = .5\ (0) + .9\ (388) = 349$$
$$c_6(2) = .2\ (0) + .9\ (188) = 169$$

so that using $z(2) = z(1)$ as the objective function we get

$$x_1^*(2) = 176$$
$$x_2^*(2) = 1194$$
$$x_3^*(2) = 72$$
$$x_4^*(2) = 349$$
$$x_5^*(2) = 0.$$

In a similar way we find the solutions for the years $t = 3, 4, 5$ and 6. These are summarized in Tables 12 and 13, and the primal variables are plotted in figure 8.

Table 12: RHS or Capacity Coefficients for Type II RLP Example

Variable	Year (t)					
	1	2	3	4	5	6
$c_1(t)$	1791	1791	1791	1791	1791	1791
$c_2(t)$	126	176	264	370	507	710
$c_3(t)$	1085	1194	1313	1444	1588	1558
$c_4(t)$	633	860	810	729	656	589
$c_5(t)$	388	349	489	547	492	443
$c_6(t)$	183	169	152	137	123	111

Table 13: Optimal Primal Solutions for Type II RLP Example

$x_1(t)$	126	176	264	347	507	710
$x_2(t)$	1085	1194	1313	1444	1284	1081
$x_3(t)$	580	72	0	0	0	0
$x_4(t)$	0	349	214	0	0	0
$x_5(t)$	0	0	0	0	0	0

The attentive reader will notice that while this particular model describes capacity in a reasonably satisfactory way when capacity is either fully utilized, or--at the other extreme--not utilized at all. It does not appear satisfactory in the positive but underutilized case. A moment's reflection will lay the cause to the purely linear dependence of investment and depreciate on the actual and potential capacity utilization of the preceding year.

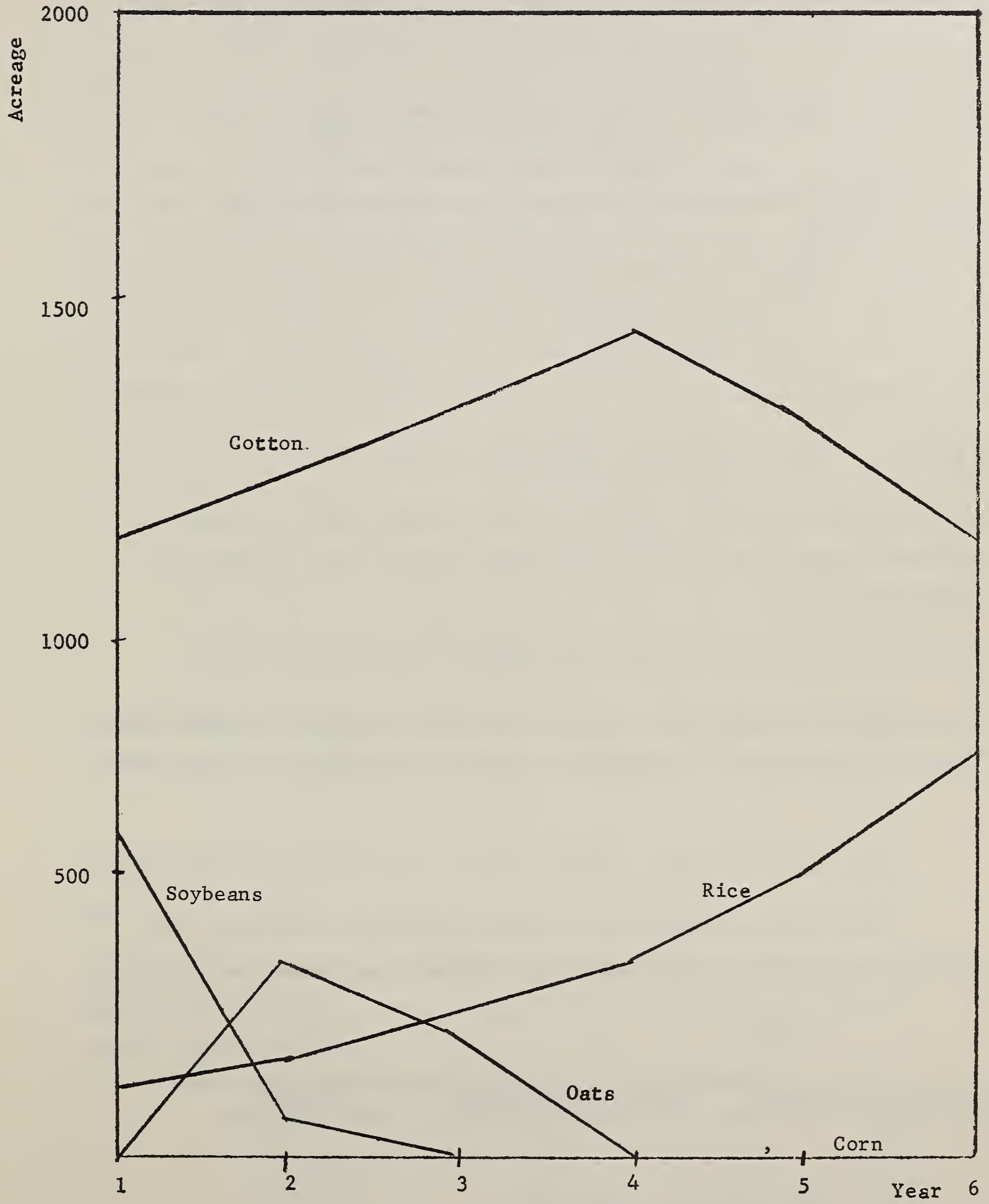


FIGURE 8: TYPE II RLP EXAMPLE: PROGRESS OF PRIMAL VARIABLES THROUGH TIME

The resolution of this difficulty cannot be taken up here. More general RLP models could be invoked for the purpose but that would carry us far into the theory of investment and dynamic economic models and beyond our immediate purpose of illustrating a particular type of RLP.

f. Type III RLP: Recursive RHS, Linear Dual Dependence

Here we shall let the recursive generation of the RHS vector depend on the dual variables as well as the lagged RHS vector. Thus,

(75) $$c_i(t) = \alpha_i r_1^*(t-1) + \ldots + \alpha_{ik} r_k^*(t-1) + \alpha_{i,k+1} c_1(t-1) + \ldots + \alpha_{i,2k} c_k(t-1)$$

or in matrix notation

(76) $$c(t) = \Omega_1 r^*(t-1) + \Omega_2 c(t-1) + u(t).$$

By recalling that $Bx^*(t) + y^*(t) = c(t)$, the model can be written as a relation between the current RHS and the lagged primal solution and slack vector, i.e.,

(77) $$c(t) = \Omega_1 r^*(t-1) + \Omega_2 Bx^*(t-1) + \Omega_2 y^*(t-1) + u(t).$$

This model will enable us to express investment in specific quasi fixed factors as function of the marginal value productivities or quasi-rents.

g. Example of Type III RLP: More on Investment and Depreciation

Under certain highly restrictive assumptions which suit our present needs, the "present" value of a capital good may be expressed as

$$V = \frac{r}{K+d}$$

where K is the interest rate, d an uncertainty discount factor and r the quasi-rent. Let these values be subscripted so that the terms

$$\frac{1}{K+d_2} r_2^*(t), \ldots, \frac{1}{K+d_k} r_k^*(t)$$

give the capital values of each fixed factor based on current quasi-rents. If we let investment depend on lagged capital values of marginal increments, we might write for our last investment model,

$$
(78)\qquad
\begin{aligned}
c_2(t) &= \frac{\beta_2}{K+d_2} r_2^*(t-1) + (1-\delta_2)\ c_2(t-1) \\
&\ \vdots \\
c_k(t) &= \frac{\beta_k}{K+d_k} r_k^*(t-1) + (1-\delta_k)\ c_k(t-1)
\end{aligned}
$$

where the first term measures gross investment and the second term measures previous capital stock less depreciation. These terms are linear in the model variables. If we suppose that the interest rate K and the prices of capital goods are reasonably constant over time we may proceed. The last expressions are already in the form (75) and can be placed in the form (76) by letting

$$
\Omega_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha_6 \end{bmatrix}, \quad
\Omega_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha_8 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_9 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_{10} & 0 & 0 \\ 0 & 0 & 0 & 0 & {}_{11} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0_{12} \end{bmatrix}
$$

and $u(t) = \begin{bmatrix} c_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

where $\alpha_i = \frac{\beta_i}{K+d_i} \quad i = 2, \ldots, 6 \qquad \alpha_i = 1-\delta_{i-6} \quad i = 8, \ldots, 12$

Let us suppose--as before--that $\delta_2 = \ldots\ \delta_6 = .10$. Also, let us suppose that $K = .05$ and $\beta_2 = \ldots = \beta_6 = .25$ and that the uncertainty discount factors may be written

Capacity	Discount Coefficient	Investment Coefficient
Rice	.15	2.50
Cotton	.15	1.25
Soybeans	.20	1.00
Oats	.05	2.50
Corn	.20	1.00

Rewriting (78) with these data we have

$$
\begin{aligned}
c_2(t) &= 2.50\ r^*_2(t-1) + .9\ c_2(t-1) \\
c_3(t) &= 1.25\ r^*_3(t-1) + .9\ c_3(t-1) \\
c_4(t) &= 1.00\ r^*_4(t-1) + .9\ c_4(t-1) \\
c_5(t) &= 2.50\ r^*_5(t-1) + .9\ c_5(t-1) \\
c_6(t) &= 1.00\ r^*_6(t-1) + .9\ c_6(t-1).
\end{aligned} \tag{78'}
$$

Let us now use the same lagged objective function values as before (Table 8) and use as initial conditions z(0) and c(1) with

$$
\begin{aligned}
c_1(1) &= 1791 \\
c_2(1) &= 90 \\
c_3(1) &= 986 \\
c_4(1) &= 504 \\
c_5(1) &= 303 \\
c_6(1) &= 181.
\end{aligned}
$$

The primal solution is

$$
\begin{aligned}
x^*_1(1) &= 90 \\
x^*_2(1) &= 986 \\
x^*_3(1) &= 504 \\
x^*_4(1) &= 211 \\
x^*_5(1) &= 0
\end{aligned}
$$

and the positive dual variables are

$$
\begin{aligned}
r^*_1(1) &= 14.10 \\
r^*_2(1) &= 52.57 \\
r^*_3(1) &= 51.12 \\
r^*_4(1) &= 1.82.
\end{aligned}
$$

Then, using (78') we find that

$c_1(2) = 1791$
$c_2(2) = 212$
$c_3(2) = 941$
$c_4(2) = 456$
$c_5(2) = 273$
$c_6(2) = 163.$

Using objective function z(1) we find that

$x_1^*(2) = 212$ $\quad r_1^*(2) = 13.00$
$x_2^*(2) = 941$ $\quad r_2^*(2) = 51.00$
$x_3^*(2) = 365$ $\quad r_3^*(2) = 55.00$
$x_4^*(2) = 273$ $\quad r_5^*(2) = 2.00$
$x_5^*(2) = 0$

In a similar manner we find the remaining solutions included in Tables 14 and 15, the primal variables of which are charted in Figure 9.

Table 14: Recursive RHS Vectors and Optimal Dual Variables For RLP Type III Example

Year Constant	1*		2		3	
	c(1)	r*(1)	c'(2)	r*(2)	c(3)	r*(3)
1	1791	14.10	1791	13.00	1791	12.00
2	90	52.57	212	51.00	309	51.00
3	986	51.12	941	55.00	916	57.00
4	504	1.82	456	0	410	0
5	303	0	273	2.00	246	4.00
6	181	0	163	0	147	0

Year Constant	4		5		6	
	c(4)	r*(4)	c(5)	r*(5)	c(6)	r*(6)
1	1791	14.00	1791	14.00	1791	13.00
2	406	51.00	493	54.00	579	56.00
3	895	52.00	871	50.00	846	50.00
4	369	0	332	0	299	4.00
5	231	1.00	211	2.00	195	
6	132	0	119	0	107	

*This is the initial condition year where c(1) is assumed known.

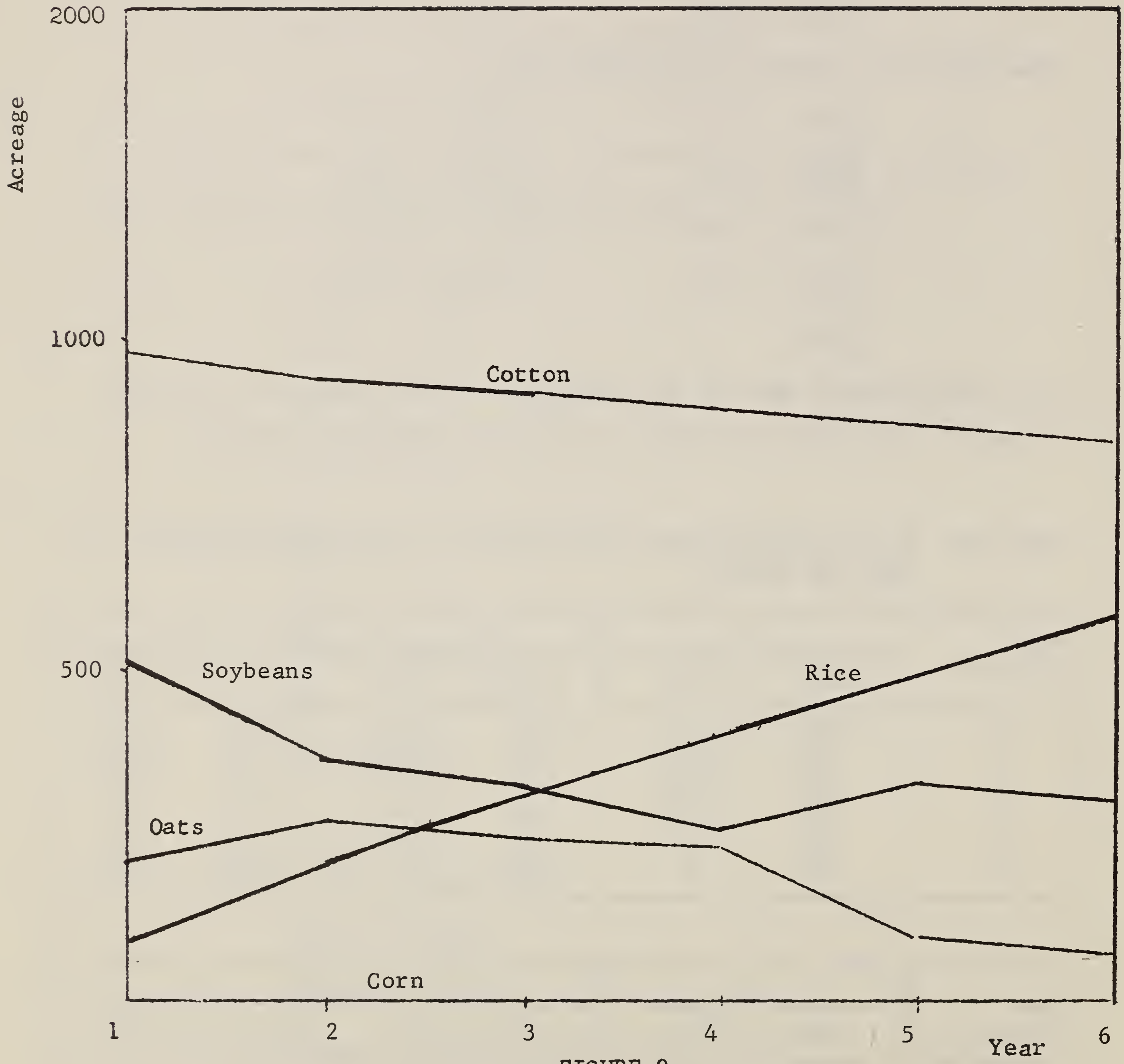


FIGURE 9

TYPE III RLP EXAMPLE: PROGRESS OF PRIMAL VARIABLES THROUGH TIME

Table 15: Optimal Process Levels for Type III RLP Example

Variable	Year (t)					
	1*	2	3	4	5	6
$x_1(t)$	90	212	309	406	493	579
$x_2(t)$	986	941	916	895	871	846
$x_3(t)$	504	365	320	259	332	299
$x_4(t)$	211	273	246	231	95	67
$x_5(t)$	0	0	0	0	0	0

*Initial condition year.

h. Other Kinds of RLP

More and more general RLP models can be constructed, but, as our discussion has already (in the last case) carried us beyond the current computing capacity of USDA LP/90, further elaboration is left to the reader. It is perhaps useful only to remark in this place on one or two RLP types that have yet to be explored but that appear to hold much promise.

Ever since Hicks' justly famous masterpiece Value and Capital appeared (1938) economists have been concerned with optimal planning over time. The field of dynamic programming is the class of mathematical methods devoted to this problem. It is quite possible that RLP models should be structured to account for planning over time. This would be done by including future time periods' (as well as present) activities and constraints. Such dynamic linear programming models could still be imbedded in a recursive structure like those presented here. This idea has been sketched out elsewhere [2, 3].

Another area begging for RLP treatment would seem to be that of spatial competition. The so-called spatial competition models presently available for agriculture suggest adjustment goals, but not rates of adjustment toward them. Without the latter, who knows if the so-called normative goals can be reached? In this field two types of recursiveness need to be built in, (1) adjustment in capacity--more or less like the models illustrated above--and (2) the dependence of actual prices on aggregate outputs and hence the dependence of expected prices on aggregate outputs. The latter type of structuring introduces a different kind of RLP, namely Recursive Objective Function, in which the coefficients of the latter would depend (through demand functions) on linear combinations of the lagged optimal process levels.

It is hoped that future analysis will include work along both of the above lines.

PART III: THE DESIGN AND USE OF LP/90

1. PHILOSOPHY

a. The Need for a General System

Agricultural economists, like other practitioners of linear programming, construct LP models that commonly call for a variety of closely related computations. A researcher may wish to maximize several different objective functions, each of whose variables are subject to identical constraints. The solutions to a number of quite independent problems may be needed. Parametric programming (price or resource mapping) on several different basic solutions may be desired.

In addition to the many LP and LP-related computations currently in use, model builders continually find new applications of LP models which require quite new computations, either for setting up an LP problem, analyzing its solution, or modifying a basic model. Sometimes the work required merely to set up the initial LP matrix is so taxing in itself, that a method of computing it from more basic data is desirable. At the other end of the spectrum, the work of analyzing a number of related or independent LP solutions may be so burdensome that some method of automatically performing the analysis and preparing final tables is required to make a study feasible.

The extent and variety of these related computations may reduce the solution of a single linear programming problem to a minor portion of the total job to be done. For this reason a working system that can accommodate a great variety of needs is a powerful asset to the user. Flexibility and generality of this kind has been the goal in the development of LP/90.

b. The System and Operation Concepts

To achieve the goal, a comprehensive computer <u>system</u> was created, containing a <u>compiler</u> and <u>library routines</u> on a special <u>Systems Tape</u>. The compiler enables new computation subprograms to be added to the library routines, so that the new uses desired by model builders can be accommodated with relatively minor additions to the entire system. It is this feature that enabled the USDA to add several entirely new subprograms of great power and flexibility to the system at a small cost.

The LP/90 systems tape on which is maintained the system compiler and all the currently available "library routines" is much like the FORTRAN system. However, actual computations are performed with an Operations Tape. The LP/90 Systems Tape is used to create an LP/90 Operations Tape. The latter contains an operation monitor or executive routine and all of the library routines including any new ones added during a compile. For a given LP/90 run, this Operations Tape is mounted on one of the computer's tape units. Its operation is controlled by the user through the use of special control cards. At the beginning of the computer run when the Load Tape Button is pushed, the operation monitor is read into the computer's core memory and executed. The first step in the execution is to read from the machine's card reader a control card that specifies which library routine is desired. The latter is then read into core and executed. When that computation is completed, a new control card is read. After all the control cards have been read, the run is over and control is returned to the machine operators for the next job, whatever it may be. In this manual we will be concerned exclusively with the use of the Operations Tape.

To summarize, it is the compiler and systems tape that enable new computation subprograms to be added to the operations tape. It is the control cards and library routines that enable the user to tailor the specific computations to his particular needs.

c. The Run Agenda

A series of LP and LP related computations performed at one time with an LP/90 operation tape will be called an LP/90 run, or run for short. The specified series of computations is called an Agenda. The Agenda describes each item to be performed in the run. Because of the flexibility of the system, an endless variety of Agenda is possible.

The library routines which exist on the Operations Tape are called Agendum Routines. The control cards which cause these Agendum Routines to be executed are called Agendum Control Cards, or Agendum Cards for short. Thus, a series of Agendum Cards in the desired order define an Agenda.

It is best to illustrate these concepts with a short example. The input data for a single LP problem are usually punched on cards and then read onto tape "offline."[11/] The resulting tape is called the I-tape. This I-tape and the LP/90 Operations Tape are placed on the appropriate tape units of the computer. The user has prepared three cards with the following appearance:[12/]

Column #	1	2	3	4	5	6
1st card	I	N	P	U	T	
2nd card	N	O	R	M	A	L
3rd card	O	U	T	P	U	T

The small numbers above the letters indicate the card columns into which the agendum name letters are punched. These three cards describe the agenda for this run. The first card says in effect, "Read the I-tape in BCD,[13/] convert it into binary[14/] data and store the results on the A-tape." The A-tape thus contains the original problem data, except in binary form, and is ready for use by the computer. The second card says, "Minimize the objective function subject to the constraints. That is, solve the linear programming problem described on the I-tape and now in binary on the A-tape." The third card says finally, "Prepare the solution of the problem just completed for printing on the O-tape."

At the end of this simple run the machine operator dismounts the O-tape and sends it to the printer where it is "listed." The user then obtains this listing, which gives the solution to the desired problem.

[11/] This might be done on a smaller computer such as the IBM 1401, or on a special purpose machine for transferring card images to tape.

[12/] Actually several additional items are required on the first of these. For the exact appearance of this card the reader may consult Parts IV or V.

[13/] BCD means Binary Coded Decimal, the standard language for recording alphabetic and numeric information on cards.

[14/] The binary "language" is used in the computer's internal operations.

One may ask, "Why do we need three special cards for this run? In any case we will need input and output. Why should we bother to tell the machine about it? Wouldn't it have been better just to push the start button and have the machine automatically do all three items?"

The answer to this question will not be fully appreciated until Parts IV and V of this manual have been studied. Yet it is possible to indicate even with this simple example, why the control cards are worthwhile. Suppose that some days after our run has been completed we wish to solve essentially the same problem as before, but with new right-hand-side elements. Would it not be desirable to avoid changing our original BCD cards, then doing a card-to-tape, and finally repeating the same Agenda when only a small part of the data need changing? As we shall see, it will be possible to read the new right-hand-side elements with a special Agendum Card and so avoid the time consuming INPUT Agendum. By mounting the A-tape from our original run, using the special Agendum Card for reading a new right-hand-side, we may proceed to NORMAL and OUTPUT and save some expensive computing time in the process.

In Part IV below a series of sample problems are presented which illustrate, in ascending order of complexity, the use of the Agendum Control Cards. There the reader may follow step by step the preparation of data and Agendum Cards and see the computational results for a variety of possible runs. The reader may wish to proceed to that part now. Part V contains a reference guide to all of the Agendum routines and their corresponding control cards for USDA LP/90. The Detailed Contents for Part V gives a list of all Agendum Routines in alphabetic order. In the remaining sections of this Part the reader will find a detailed description of how to prepare input data, how to use the Operations Tape and how to interpret the Output Information.

It will be difficult no doubt for the beginner to use the remaining pages of this part effectively until he has studied the examples of Part IV. After that they may be used together with Part V as a useage reference manual. Paragraphs and lettered subsections preceded by an asterisk need not be read by beginners. In some cases an understanding of them would require collateral readings [see References at the end of the handbook].

2. PREPARING THE INPUT DATA

a. The Job Concept

The input data for an LP computation or for a series of closely related LP computations will be called a job. The data format has been designed, like the Agendum Control Cards, for maximum flexibility and convenience to the user. Three basic characteristics of the format contribute to this goal. First is the ability to include multiple objective functions, various multiple right-hand-side vectors, and multiple "change vectors" for parametric programming. This enables the user to store compactly data for many LP and parametric LP computations that use the same "structural" or "constraint" matrix coefficients. Second, a series of such "jobs" may be stored consecutively on the I-tape. Thus, many computations for problems with different "structural" or "constraint" matrix coefficients (and different dimensions) may be solved during the same run. Third, rows and columns of the LP matrix are identified by six character mnemonic names, rather than sequential numbers. This greatly facilitates preparing model data, checking them for errors, and reading solution print-outs.

The job data are usually punched on cards from special format sheets and the resulting card deck "card to tape" on off-line equipment. The resulting tape as noted above is called the I-tape. The rules for preparing the data are now described in detail.

b. The Title or Remarks Card

A title card that identifies the job data to follow is placed at the beginning of the deck. An asterisk must be punched in card column 1. The title may follow in columns 2-72. For example, the following hypothetical title card might be used:

*LP MATRIX FOR SMALL DELTA FARMS WITH PRESENT TECHNOLOGY AND CLASS I SOILS.

Further title or remarks cards may be placed anywhere among the job data in order to identify portions of the input.

The title cards that precede the ROW ID cards (see below) will be printed when the input data is read by the INPUT Agendum.

c. Row Identification

The row names identify the objective functions, the change vectors and the active constraint rows of the LP matrix. The row names are preceded by an announcement card with ROW ID punched in card columns 1-6. The row names are comprised of six alphabetic or numeric characters punched in card columns 13-18. The first of these characters must be blank or numeric (1-9) but the remaining five may be alphabetic (or numeric or blank). Of course, each row name must be distinct.

Column 12 of the row identification cards contains an indicator which identifies whether the row is (1) an objective function row or objective function change vector for parametric programming, (2) an equation constraint (one that must be satisfied as an equality by the solution), (3) an "equal or less than equal" inequality constraint or, (4) a "greater than or equal" inequality constraint. The character to be punched in each case is as follows:

	BCD Character Punched in Column 12	Meaning of Character
(1)	Blank	Row is an objective function or change vector row.
(2)	Blank	Row is an equality constraint.
(3)	+	Row is an "equal or less than equal" inequality constraint.
(4)	-	Row is a "greater than or equal" inequality constraint.

All of the type (1) rows, that is, the objective function or change rows, must have their row identification cards placed first in the list. However, the type (2), (3),and (4) type rows, that is, the active constraint rows, may come in any order after the type (1) rows.

The purpose of the indicator column in the row identification card is to enable the INPUT Agendum to insert the appropriate "slack" vectors during the conversion to binary so that the problem will be properly set up for solution by the NORMAL Agendum. By means of the indicator column the user avoids having to input slack vectors at all.

*The user may optionally place up to five row identifications per card. This is done by punching the desired number of row identifications to be read per card in card column 6. In this case card column 24 would be the indicator column for the second row identification which would appear in card column 25-30. Card column 36 would be the indicator column for the third row identification, and so on.[15]

*The user need not specify all rows for which coefficients exist in the body of the matrix. By including row names for just those rows that make up the desired problem, the user can select a subset of rows from a very large basic matrix to form a given LP problem for solution.

*A further option is available. To use it the announcement card must be punched ROW ID, cu where "c" is a tape channel designation (A or B) and "u" is the tape unit number (0-9), or if desired it is punched ROW ID, CR where "CR" indicates the card reader. In this case, the INPUT Agendum will read from the specified input unit the row identification card records and basis records if any. The latter must be preceded by the ROW ID announcement card record and terminated by an additional announcement card record containing MATRIX.

*d. Basis Headings

The basis headings provide an optional feature which will be of primary interest to experts. The beginning user of LP/90 can omit this section altogether.

The INPUT Agendum automatically creates a basis consisting entirely of the slack vectors. The NORMAL Agendum, which usually follows INPUT, begins computations from this initial all slack basis. However, from previous runs or other considerations the user may have a very good idea of a basis that will be optimal or nearly optimal. By prespecifying this basis he can force NORMAL to begin computations with it and in this

---

[15] A paragraph or subsection preceded by an asterisk may be omitted on the first reading.

way he may diminish the number of iterations required to find the optimal solution. Unless this basis is cleverly chosen, however, the user might increase the number of iterations. Therefore, this option must be used with caution. Two additional Agendum routines also enable an initial basis to be specified and for most purposes these will probably be preferred. (See NEW.BASIS and LDHREG in Part V.)

The basis heading cards are preceded by an announcement card with BASIS punched in columns 1-5. The matrix column names are punched in card columns 7-12 and the slack names (defined by active constraint row names) which they are to replace are punched in card columns 13-18. Only slack vectors to be removed from the initial all slack basis need be mentioned. The pairing of column and row names on given basis heading cards is not unique, and the pairing chosen will no doubt be changed during the inversion which follows.

e. Matrix Elements

Following the row identification cards (or, optionally, the basis heading cards) the user places the non-zero LP matrix elements. The matrix element cards are preceded by an announcement card with MATRIX punched in card columns 1-6. The non-zero elements are punched one per card in the following format:

Card Columns	Contents
1-6	Blank
7-12	Column name
13-18	Row name
19	Sign of element (blank ≡ + .)
20-23[16]	Integer part of element (must be less than 9999)
24	Decimal point must be punched
25-30	Fractional part of element (trailing zeros may be left blank)
31-80	Any identifying comment or blank.

[16] Columns 20-30 may vary from this recommended format. See LP/90 Reference Manual, CEIR, Inc., Arlington, Virginia.

The columns can be in any order, but all the elements of one column must be together. The order of rows within a given column is arbitrary. In any case the rows will be ordered to match the order of the row identification cards. Any element with a row name not given in the row identification list is omitted by the INPUT Agendum.

*An optional feature is the splitting of matrix elements into groups of vectors indicated by the announcement cards PARTITION or CURTAIN. However, the usage of this option has largely been superseded by the special Agendum Routine FLAGS. For this reason these options will not be described here. For reference see Part V, Section 1, INPUT.

*In order to facilitate checking the input data provision is made for including count or sum cards at the end of the element cards for a given vector. A card containing SUM in card columns 4-6 has the arithmetic sum of the element values in the vector in card columns 19-30. A card containing CNT in card columns 4-6 has the number of non-zero elements in the vector in card columns 20-23 with a decimal in card column 24.

f. Right-Hand-Side or Constraint Vector Elements

The right-hand-side or constraint vector elements are preceded by an announcement card punched FIRST B in card columns 1-7. (A blank must be left in column 6.) Column names may be punched for the user's convenience, but they are never used by the system. The row names, non-zero elements, and the arbitrary remarks format is identical to that of the matrix element cards (subsection e above).

Additional right-hand-side elements can be added. The non-zero elements for each such right-hand-side must be preceded by an announcement card punched NEXT B,N where N is an integer greater than 1 and less than 9999. N gives the number of the right-hand-side. The right-hand-sides must be numbered in an ascending but not necessarily consecutive order.

*At the user's option the FIRST B announcement card may be replaced with the announcement card FIRST B,cu where "c" and "u" describe the tape channel and tape unit respectively, or are punched CR for the card reader. The INPUT Agendum will then seek the specified input unit to obtain the right-hand-side elements (including multiple right-hand-sides). In this case the specified right-hand-side elements must be preceded by the FIRST B announcement card and followed by an EOF card.

g. End of a Job

All of the above data and their respective announcement cards are followed by a card punched EOF in card columns 1-3. Columns 4-6 must be blank. The first such EOF card may be followed by any number of additional complete jobs as described in Sections b-f above. Each such job is terminated by an EOF card.

h. Summary of Input (I-Tape) Data

Figure 10 shows the input deck arrangement as just described. The reader may turn to Part IV, Section 1, Table 18, for a listing of a sample input deck. Figure 11 shows a schematic diagram of the data for an LP problem. In everything that follows we shall call the data in this form (or on I-Tape) an "LP Matrix."

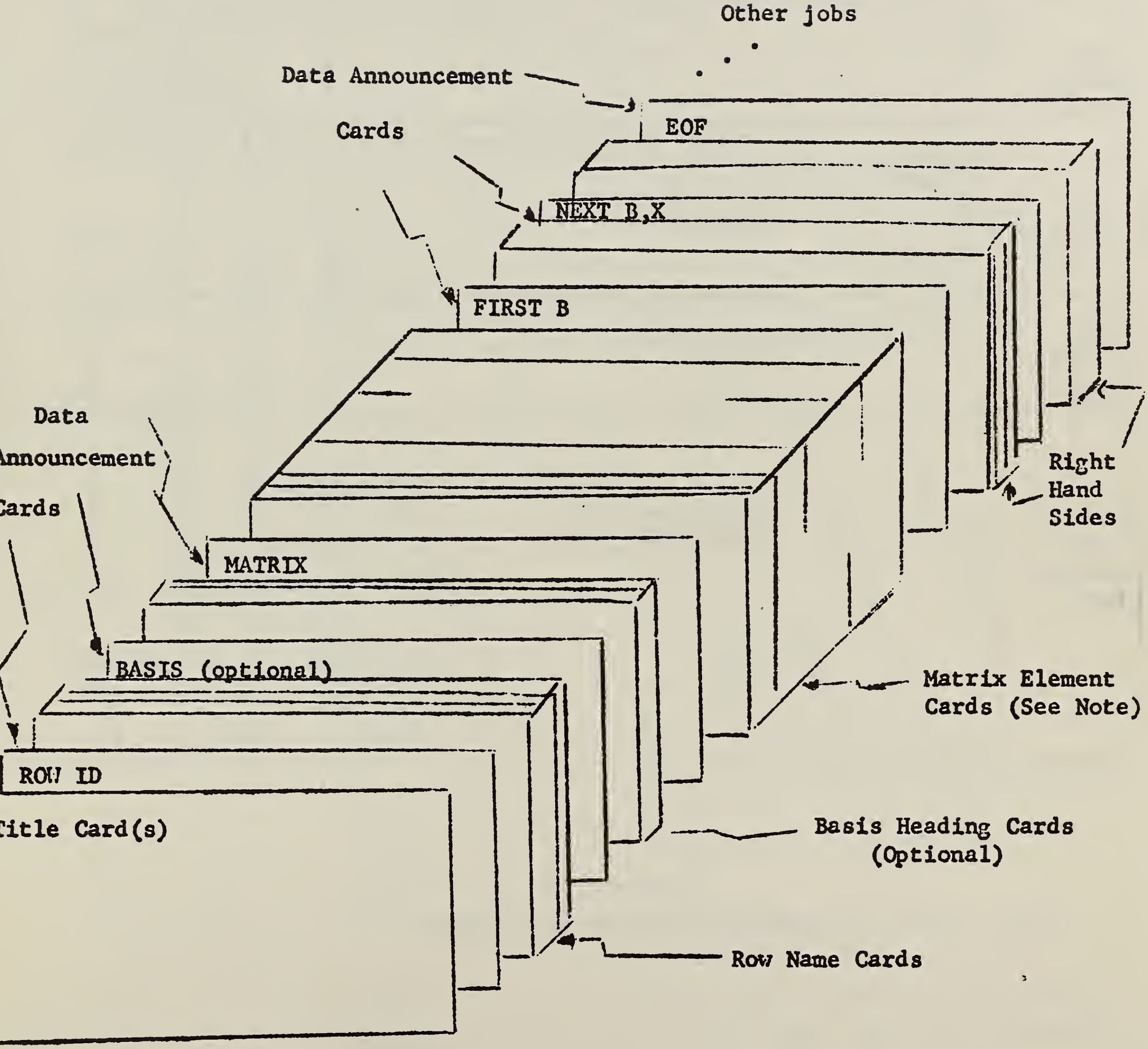


FIGURE 10

INPUT CARD ORDER

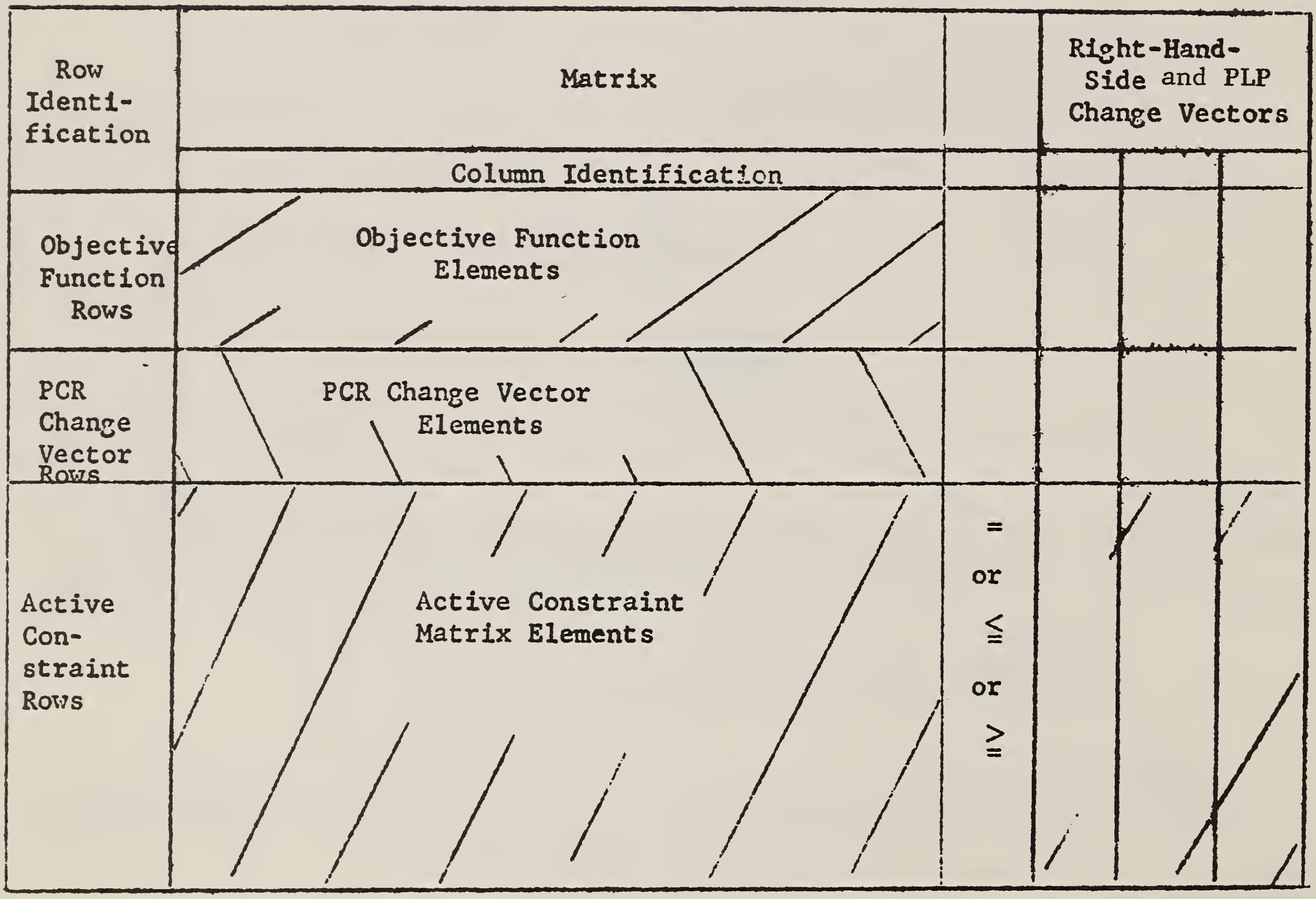


FIGURE 11

SCHEMATIC DIAGRAM OF LP MATRIX

3. OPERATING THE LP/90 SYSTEM

a. Machine Requirements

USDA LP/90 can be used with the IBM 709, 7090, or 7094 computers. A special system called LP/94 is being written for the 7094 computer. The latter will be more efficient, but LP/90 is still very fast, even when used on the 7094, and has the further utility of being compatible with the 709 as well as 7090. Similar systems (at the time of this writing) are in preparation for the IBM 7070, 7074, and 7040 computers.

USDA LP/90 requires an on line card reader (for reading the Agenda), an on line printer (for printing progress and error messages) at least five tape units on channel A and at least four tape units on channel B. An on line card punch is ordinarily required, though a version of the system can be employed that does not require this particular output unit.

b. Tape Assignments

LP/90 uses tapes on two input-output channels, A and B. The usual assignments are as follows:

Channel	Unit	LP/90 Symbol	Content
A	1	S-Tape	LP/90 Operations Tape
A	2	A-Tape	Binary LP Matrix (Created by INPUT, TRANSFORM, REVISE or RESTART)
A	3	B-Tape	Backward Eta (Used by NORMAL)
A	4	M-Tape	Binary or Mediary Output Tape
A	5	U-Tape	Scratch
A	6-0	None	Available for any optional input or output
B	1	I-Tape	BCD LP Matrix (Original Input Data)
B	2	O-Tape	BCD Output Tape (Created by OUTPUT)
B	3	F-Tape	Forward Eta (Used by NORMAL)
B	4	V-Tape	Scratch
B	5-0	None	Available for any optional input or output.

*If a 709 is being used, all of these tapes must be Low Density tapes. If a 7090, or 7094 is being used MOD IV, Hi-Density tapes are preferred for all units except, possibly, the I- and O-tapes. The choice of Hi- or Low-Density tapes for the I- or O-tapes depends on how the off-line card to tape for the I-tape, and the tape listing of the O-tape is performed.

c. Normal Operation

The USDA LP/90 Operations Tape is placed on tape unit A1 and set to load point. The I-tape is loaded on unit B1 and set to load point. Finally, the M-tape (A4) should be set to load point. All other tapes are automatically rewound. Remember that the I-tape has been card-to-taped off-line.

The Agenda, made up of the Agendum Control Cards prepared by the user, is placed in the on-line card reader.

The "Clear" and "Load Tape" buttons are pushed in succession. From this point on LP/90 is in command, reading Agendum Cards one after the other, and executing the specified Agendum Routines in turn. If no trouble is encountered during the run, the final OUTPUT Agendum will convert the binary (mediary) M-tape to BCD on the O-tape for printing. At the run's termination the latter is dismounted and a listing obtained off-line.

*d. Sense Switch Controls

The operation of LP/90 can be controlled, not only by means of the Agendum Control Cards, but also by means of the computer's Sense Switches. Placing a Sense Switch down during operation permits selected monitoring of the computations or aids in executing appropriate responses to error conditions. The possible uses are as follows:

Sense Switch	Use

#1 . . . . If placed down during the operation of INTRODUCE, NORMAL, DO.PLP or DO.PCR the GETOFF Agendum will be executed at the termination of the current iteration. This enables a long run to be interrupted so that other higher priority programs can be run without wasting the computations already completed. (See Part V GETOFF.)

#2 . . . . If placed down during operation of INTRODUCE, NORMAL, DO.PLP, or DO.PCR, the current iteration will be completed, the Agendum terminated and the card reader consulted for the next Agendum. If placed down before exercising GETOFF, the latter Agendum will use the B-tape (Backward Eta) for the GETOFF output.

#3 . . . . If placed down at beginning of run, a "by-pass" feature is initiated that will automatically cause the OUTPUT Agendum to be executed if certain unexpected stops occur. These stops are as follows:

Stop In	Message
NORMAL	NO FEASIBLE SOLUTION UNBOUNDED SOLUTION
DO.PLP	THETA UNBOUNDED
DO.PCR	PHI UNBOUNDED.

#4 . . . . When placed in down position during NORMAL, an optimal solution will be followed automatically by a d_j check in double precision. This will precede the reading of the next Agendum card. If all $d_j \geq 0$ without a tolerance and if $|d_j|$ < tolerance for all j in the basis headings, operation of the run will proceed normally. If any $d_j < 0$ or if any $|d_j|$ > tolerance for j in the basis headings, those j together with d_j will be output to the mediary tape and a halt will occur after an on-line message. If sense switch

Sense Switch	Use
	2 is now placed in the down position and the computer started, the next Agendum card will be read. If the computer is started with sense switch 2 up, control will revert to NORMAL. See references [11].
#5	a) If placed in the down position during INTRODUCE, NORMAL, INVERT, DO.PLP or DO.PCR, a "cycle print" will be output on-line at the end of the current iteration. If the switch is restored to up position at any time during printing (after the headings), printing will terminate. Note that switch 5 must be restored "on the fly" after printing.
	b) If placed down at the Matrix Singular stop in INVERT, a full printout containing α_s^i will be written on the mediary tape. (Not on-line.) See [11].
#6	If placed in the down position during INTRODUCE, NORMAL, INVERT, DO.PLP or DO.PCR, the solution will be checked at the end of the current iteration. If any error exceeds the tolerance in use, then a halt occurs. If switch 6 is left down and the computer started, a "check solution" print occurs on-line followed by the same halt. If the switch is put up and the computer started, normal operation resumes. Note that switch 6 must be restored "on the fly" after the check if all errors are within the tolerance. Printing will terminate if the switch is restored to up position at any time during printing (after headings).

*e. Summary of Error Halts and Operator Response

The normal termination of all Agenda except GETOFF, RESTART, and OUTPUT is to read the next Agendum Card. However, certain halts may occur because logical errors have been made in structuring the model, or because of faulty input data. Before the machine "halts" an error message is printed. These follow along with a brief interpretation and suggested response.

i) "MAXIMUM ERROR ON ROW ----"

This may occur at a check solution during INTRODUCE, NORMAL, INVERT, DO.PLP, or DO.PCR. If the error is small press START to continue. (INVERT may clear up digital trouble.) If the error is large it may be due to machine malfunction, to wrong input data, or more probably to a bad Agendum sequence such as LDBINARY NORMAL.

ii) "MONOTONICITY CHECK ----"

This will occur during NORMAL when the solution is feasible if the change in current value is greater than tolerance in the "wrong" direction. Press START continues. If the error is large suspect machine trouble.

iii) "SELECTED VECTOR IN BASIS ----"

This will occur in NORMAL, PLP, or PCR if the vector chosen is already in the basis. It is ordinarily due to digital error exceeding the DJ tolerance. Press START ignores the vector and continues; however, the algorithm usually terminates with no further iteration. If it terminates in NORMAL with an infeasible solution, the message NO FEASIBLE SOLUTION is printed; if it terminates NORMAL and the solution is feasible, an OPTIMAL termination results; if it terminates in PLP, a THETA AT MAX termination results; if in PCR, a PHI AT MAX termination results.

iv) "MATRIX SINGULAR . . . . . VECTOR DELETED ----"

This will occur in INVERT if no pivot element greater than 8×10^{-6} can be found. Press START continues and ignores the vector. (See Part V, INVERT for other options.)

v) "NO FEASIBLE SOLUTION"

This can occur in NORMAL due to faulty problem formulation, or occasionally it may be due to digital trouble. Press START twice goes to read the next Agendum.

vi) "UNBOUNDED SOLUTION"

This may occur in NORMAL due to faulty formulation. There is also the remote possibility that a legitimate pivot (when the solution is feasible) has been rejected and no other can be found. Continuation is possible by pushing START to read the next Agendum card.

vii) "THETA UNBOUNDED"

This occurs in PLP when no limiting row (vector leaving) can be found. Press START reads the next Agendum. This does not mean that the problem is unbounded, but only that one can extrapolate to any finite value of theta without a change of basis.

viii) "PHI UNBOUNDED"

This occurs in PCR when no vector to enter the basis can be found. Press START reads the next Agendum. This does not mean that the problem is unbounded, but only that one can extrapolate to any finite value of PHI without a change of basis.

ix) "CAN'T FIND VECTOR ON A-TAPE"

This may occur in the subroutine CHECK used by NORMAL, INTRODUCE, PLP, PCR, INVERT. Press START tries again, trouble in this case would probably be due to machine error.

x) "CAN'T FIND XXX BASIS VECTORS ON A-TAPE"

This may occur in INVERT when some name(s) is (are) specified in the basis headings which cannot be found on the A-tape. The number of vectors not found is given as XXX. Press START continues with INVERT; the final basis will have artificial variables corresponding to the vectors not found.

xi) "CAN'T FIND RIGHT-HAND-SIDE"

This may occur in NORMAL, INTRODUCE, PLP, PCR, and INVERT. Press START reads the next Agendum. Input data should be checked. Also check parameter used on Agendum Card.

xii) "INPUT ERRORS"

This occurs in INPUT. Press START reads the next Agendum. Sense switch 3 goes to OUTPUT. Print O-tape for more detail; Check I-tape carefully.

xiii) "REDUNDANCY"

This may occur in any routine; see the detailed operating instructions [11] for possible continuation.

xiv) "GETOFF COMPLETE START WITH SS2 DOWN READS NEXT AGENDUM"

If Sense Switch 2 is up the GETOFF is duplicated on the B-tape.

xv) "RESTART COMPLETE START WITH SS2 DOWN READS NEXT AGENDUM"

If Sense Switch 2 is up the last Agendum (prior to GETOFF) is actuated.

xvi) "INFEASIBLE SOLUTION CANNOT DO PLP (PCR)"

Press START reads next Agendum.

There are miscellaneous other stops without on-line messages. They usually indicate machine trouble. Other stops with on-line messages are possible, but these will usually be self-explanatory.

4. OUTPUT FORMAT

a. Operation Comments and the Run History

LP/90 output is printed on-line or written on a tape for subsequent printing (0-tape) or is in the form of restart information on tape or punchout cards. The progress of a run may be followed on-line where comments and important data are printed. Most of the results are for off-line printing. By means of the CONTROLS Agendum Card the user can control the amount of different kinds of output. (See Part V.)

The output data and operation comments are intermixed in general. With but few exceptions, all output printed on-line is also printed off-line. Thus the off-line output contains a nearly complete history of the run.

b. Types of Computation Output

Computation results are output in four styles:

1. Single-line summaries in tabular form with occasional captions printed.

2. Full listings of the current solution with appropriate headings, captions, etc.

3. Full tables of special output as, for example, the d_j quantities produced by the DO.D/J Agendum, or the matrix element output produced by PICTURE.

4. Cycle prints, a special form of full output for on-line use only. The special output as well as operation comments (Items 3 and 4) are more or less self-explanatory.

A full listing of the current solution consists of headings, captions, a one-line summary and five columns of m + 1 entries, where m is the number of rows in the LP matrix.

c. Output Summary Line

The headings for the summary are as follows:

Heading	Item Printed Below Heading
TOTAL ITERS	The total number of simplex iterations since the start of the job. (Does not count inversion iterations or transformation made in input.)
NO. ETAS	The total number of eta-transformations now in effect. One is produced by each simplex iteration and each inversion iteration. This starts over with an INVERT instruction. (Counts transformations made in INPUT or REVISE.)
ETA	Number of eta records (zero for in core problems).
ROW INDENT.	The row name of the objective function now being minimized.
SUM OF INFEAS.	This is a measure of the infeasibility (sum of the negative beta values) still in the system. It may not decrease monotonically. When feasibility occurs, it is replaced by CURRENT VALUE. 17/
CURRENT VALUE	When the solution is feasible, this is value of objective function. It should increase monotonically during NORMAL when the scale factor is positive. During PCR it is the current value of the composite objective function.
CHOSEN VECTOR	The name of the variable entering the basic solution during the current iteration.
VECTOR REMVD	The name of the variable being replaced in the solution during the current iteration. This is not the row position on which the pivot occurred.
RHS NO.	The numerical identification of the right-hand-side currently in use.
C/V NO.	Change vector number or change row number currently in use.
NO. OF INFE	The number of non-zero infeasibilities remaining. Replaced by CURRENT D/J when solution is feasible. 17/
CURRENT D/J THETA/PHI	If feasible and in NORMAL, value is the current D/J; of vector entering basis, i.e., the min. D/J. If in DO.PLP, or DO.PCR, value is the current value of Theta or Phi.

17/ The headings may not change immediately when feasibility occurs.

d. Full Print-Out of Solution

Of the five columns in a full print-out, the first three are standard and are headed J(H), BETA(H), and ROW(I). The various columns are described below:

(1) Basic Solution: The column headed J(H) is the list of variables in the current basic solution. Artificial variables have no name and are printed as 0 00000 in the list. All profit rows are considered artificial and always appear in every basic solution. At least one other artificial variable, usually SIGMA, will remain through the run if SIGMA was required.

(2) Values of Basic Variables: The column headed BETA(H) contains the values of the basic variables listed under J(H) in the same order. This order is unpredictable as is the set of basic variables chosen. Hence, the identification under J is an essential part of the solution listing.

(3) Row Names: The column headed ROW(I) is merely a list of the row names as specified by the input. It is provided for convenience and is placed in the middle of the page because the fourth and fifth columns are in proper row order.

(4) Printing of Dual Variables (Shadow Prices): The fourth column, headed PI(I), is the "pricing" vector used in the composite algorithm. This column is blank for INVERSION print-outs. If the solution is feasible and optimal, this is the solution of the dual problem, the so-called "shadow prices." During PCR this is the PI vector corresponding to the composite objective function.

(5) Printing of Right-Hand-Side: The fifth column, when headed B(I), is the given right-hand-side. During DO.PLP, solution prints are labeled CURRENT RT.H.SIDE and this column is headed $B(I) + T^{*}C(I)$ where

$B(I) = B_i$ = Original right-hand-side values

$T^{*}C(I) = \theta$ = Current value of Theta

$C(I) = C_i$ = Change vector values.

At PHI UNBOUNDED the 5th column is headed PI(SIGMA); it is then the row of the inverse corresponding to the change cost row. It can be used to extrapolate to any desired level of PHI by adding the appropriate delta times PI(SIGMA) to PI. The Beta values do not change. The new Current Value will be the sum over the functional rows only of PI(composite) values times the corresponding BETA values.

The THETA UNBOUNDED the 5th column is headed GAMMA(H); it is then the negative of the change vector (RHS), expressed in terms of the current basis. The effect of using any desired (positive) amount of this vector can be found by subtracting the appropriate amount times GAMMA from BETA.

A cycle print on-line (via sense switch 5) has a sixth column labeled ETA. This is the current transformation.

A check solution on-line (via sense switch 6) has a sixth column labeled ERR. This is B- A*BETA.

The special prints for INPUT, DO.D/J, TABLEAU, and sense switch 5 during INVERT are covered under the description of those Agendum Routines. (See Part V.)

PART IV: SAMPLE COMPUTATIONS WITH USDA LP/90

INTRODUCTION

The old saw has it that the proof of the pudding is in the eating. To the user the proof of a computer program is surely in the computing. In this section USDA LP/90 is put to work, computing a series of six sample problems. They are placed in order of ascending complexity, each example illustrating several specific Agendum routines. In this way the principles of the system are illustrated concretely and one may derive a very specific understanding of them. Further, it is hoped that the examples will indeed show that USDA LP/90 is a powerful tool for agricultural economics research, just as its other versions have proved to be powerful tools in many other fields.

At the completion of this Part the prospective user will be familiar with some of the most powerful of the SHARE, PROP, and USDA Agendum routines.[18/] It would have been impossible to include an illustration of each and every Agendum routine available for users of USDA LP/90. However, armed with a specific knowledge of those presented here, one can confidently exploit on his own the full list briefly described in Part IV.

The discussion of each example is divided into five sections that take up in turn the purpose, data, agenda, output and interpretation of the sample problem. A small linear programming model is used as the common basis for all the examples, though it is progressively extended as more complex Agenda are illustrated. The reader will see that it is only a modest generalization of the illustrative model used in Part II above. The model is designed to serve as a vehicle for illustrating the Agenda concept and use of LP/90. As a description of real farm production for any particular time or place it would be inadequate. But that is quite beside the point.

Most of the Agendum control cards have several optional forms. In discussing the sample Agenda, only the options actually used are described here. It is only in Part V that all of these details are presented.

To facilitate reading, the tables are placed together at the end of the lettered subsection in which they are first mentioned.

18/ For an explanation of this threefold classification of the Agendum routines, see Part I, INTRODUCTION.

1. A SINGLE LP PROBLEM

a. Purpose

To begin, a very simple example is considered. Five Agendum routines will be needed and a single LP problem solved. However, it will take some space to develop the data for the example. This same data will also serve for the more complicated examples which follow. Because of these multiple uses it will be well in the beginning to describe it in some detail.

b. Data

Our sample model consists of several components. First, seventeen production processes or activities are included. They describe the production of five crops on two different soil types at two different levels of fertilization. One crop, cotton, is assumed to have two different technical "stages" of production, one involving hand harvesting, the other, machine picking. The character of these processes is summarized in Table 16.

Table 17 briefly describes the constraints on the process levels. They include the amount of land in each of two soil classes, the supply of labor during the harvest season, the supply of fertilizer, acreage allotments on cotton and rice producing activities, upper and lower bounds on the total acreage devoted to each crop (flexibility constraints), and finally, mechanical cotton picker capacity.

Next, Table 18 gives the LP Matrix in LP/90 input form.[19/] The first row of the table gives the objective function. As LP/90 is designed to minimize a linear function, the objective row contains negative coefficients indicating positive profits to be maximized.[20/]

In short, the linear programming model for this example is to minimize (= maximize the negative) objective function in the row marked "bZ1bbb"[21/] subject to the constraints whose coefficients are described in the following rows of the table.

19/ See Part II, Section 7.

20/ See Part II, Section 1.d.

21/ Lower case letter "b" means blank.

The coefficients of the lower bound constraints marked LBCŌT, LBŌAT,[22/] etc., are minimal limits on the sum of the process levels for each commodity, respectively. That is, they could have been written as "greater than or equal to" constraints. In that case, the appropriate input coefficients would be plus ones instead of negative ones, and their respective row identification cards would have minus signs punched in indicator column 12. Either treatment is equally valid and the choice between them immaterial.[23/]

The input data were transcribed from Table 18 onto special LP/90 Matrix Element format sheets for key punching. Table 19 shows the first page of this input to the key punchers. After keypunching, the cards were listed. Table 20 shows this listing. These cards were then written onto tape (the I-tape) by the "off line" equipment. The card listing of Table 20 appears just as a listing of the I-tape would appear, except that the pages have been trimmed and pasted together to conserve space.

22/ The bar over the letter "Oh" is used to distinguish it from the number zero.

23/ See above, Part III, Section 2.c.

Table 16: Summary of Activities or Processes for the Sample Model

Column Symbol	Crop	Technology Stage	Soil	Fertilizer Level
1COT11	Cotton	1	1	1
1COT12	"	1	1	2
1COT21	"	1	2	1
1COT22	"	1	2	2
2COT11	"	2	1	1
2COT12	"	2	1	2
2COT21	"	2	2	1
2COT22	"	2	2	2
1OAT11	Oats	1	1	1
1OAT12	"	1	1	2
1OAT21	"	1	2	1
1OAT22	"	1	2	2
1SBG11	Soybeans	1	1	1
1SBG21	"	1	2	1
1ALF11	Alfalfa	1	1	1
1RIC21	Rice	1	2	1
1RIC22	"	1	2	2

Table 17: Summary of Constraints for Sample Model

Row Symbol	Type of Constraint	
SOIL 1	Soil Class 1	
SOIL 2	Soil Class 2	
LABP4	Unskilled Labor Fourth Quarter (Oct-Dec)	
FERT	Fertilizer	
LOTCT	Acreage Allotment	Cotton
LOTRC	" "	Rice
UBCOT	Upper Bound	Cotton
UBOAT	" "	Oats
UBSBG	" "	Soybean Grain
UBALF	" "	Alfalfa
UBRIC	" "	Rice
LBCOT	Lower Bound	Cotton
LBOAT	" "	Oats
LBSBG	" "	Soybean Grain
LBALF	" "	Alfalfa
LBRIC	" "	Rice
S2COT	Stage 2	Cotton Capacity

TABLE 18

LP MATRIX FOR SINGLE LP

Row Symbol	Matrix Column Symbol											
	1COT11	1COT12	1COT21	1COT22	2COT11	2COT12	2COT21	2COT22	1OAT11	1OAT12	1OAT21	1OAT22
Z1	- 35.16	-114.79	- 36.26	- 57.05	- 42.13	-144.33	- 42.52	- 71.60	- 3.80	- 30.63	- 2.09	- 21.45
SOIL1	1.00	1.00			1.00	1.00			1.00	1.00		
SOIL2			1.00	1.00			1.00	1.00			1.00	1.00
LABP4	5.80	13.50	5.50	8.30								
FERT		4.00		4.00		4.00		4.00		4.00		4.00
LOTCT	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00				
LOTRC												
UBCOT	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00				
UBOAT									1.00	1.00	1.00	1.00
UBSBG												
UBALF												
UBRIC												
LBCOT	- 1.00	- 1.00	- 1.00	- 1.00	- 1.00	- 1.00	- 1.00	- 1.00				
LBOAT									1.00	- 1.00	- 1.00	- 1.00
LBSBG												
LBALF												
LBRIC												
S2COT					2.79	6.51	2.73	4.89				

TABLE 18-Continued

Row Symbol	Matrix Column Symbol					Con-straint Type	Right-Hand-Side FIRST B
	1SBG11	1SBG21	1ALF11	1RIC21	1RIC22		
Z1	- 51.07	- 24.72	- 23.66	- 60.91	-144.81		
SOIL1	1.00		1.00			≤	500
SOIL2		1.00		1.00	1.00	≤	1500
LABP4	1.10	1.10	0.70			≤	9900
FERT					4.00	≤	1525
LOTCT						≤	700
LOTRC				1.00	1.00	≤	80
UBCOT						≤	880
UBOAT						≤	520
UBSBG	1.00	1.00				≤	650
UBALF			1.00			≤	230
UBRIC				1.00	1.00	≤	150
LBCOT						≤	- 640
LBOAT						≤	- 296
LBSBG	- 1.00	- 1.00				≤	- 370
LBALF			- 1.00			≤	- 152
LBRIC				- 1.00	- 1.00	≤	- 79
S2COT						≤	175

Form B

Table 19

KEYPUNCHING FORM FOR MATRIX ELEMENTS

SHARE STANDARD

JOB NO. 9021
DATE 18 June 1963
NAME R. H. DAY
SHEET 1 OF 4

KEYPUNCH: COL. 24 PUNCH DEC. PT.

ANNOUNCEMENT (1–6)	COLUMN NAME (7–12)	ROW NAME (13–18)	VALUE (19–30)	CARD I.D. (73–80)
ROW ID			.	EXAMPLE 1
		Z1	.	
	+	SOIL1	.	
	+	SOIL2	.	
	+	LABP4	.	
	+	FERT	.	
	+	LOTCT	.	
	+	LOTRC	.	
	+	UBCOT	.	
	+	UBOAT	.	
	+	UBSBG	.	
	+	UBALF	.	
	+	UBRIC	.	
	+	LBCOT	.	
	+	LBOAT	.	
	+	LBSBG	.	
	+	LBALF	.	
	+	LBRIC	.	
	+	S2COT	.	
MATRIX			.	
	1COT11	Z1	-35.16	
		SOIL1	1.00	
		LABP4	5.80	
		LOTCT	1.00	
		UBCOT	1.00	
		LBCOT	-1.00	
	1COT12	Z1	-114.79	
		SOIL1	1.00	
		LABP4	13.50	
		FERT	4.00	
		LOTCT	1.00	
		UBCOT	1.00	
		LBCOT	-1.00	
	1COT21	Z1	-36.26	
		SOIL2	1.00	

CD-32

Table 20: Listing of I-Tape for Single LP Example

(Listing Sheet 1)

```
* USDA LP/90   INPUT  LP MATRIX FOR  EXAMPLE  1
ROW ID
          Z1
        + SOIL1
        + SOIL2
        + LABP4
        + FERT
        + LOTCT
        + LOTRC
        + UBCOT
        + UBOAT
        + UBSBG
        + UBALF
        + UBRIC
        + LBCOT
        + LBOAT
        + LBSBG
        + LBALF
        + LBRIC
        + S2COT
MATRIX
    1COT11 Z1    -  35.16
    1COT11 SOIL1     1.00
    1COT11 LABP4     5.80
    1COT11 LOTCT     1.00
    1COT11 UBCOT     1.00
    1COT11 LBCOT-    1.00
    1COT12 Z1    - 114.79
    1COT12 SOIL1     1.00
    1COT12 LABP4    13.50
    1COT12 FERT      4.00
    1COT12 LOTCT     1.00
    1COT12 UBCOT     1.00
    1COT12 LBCOT-    1.00
    1COT21 Z1    -  36.26
    1COT21 SOIL2     1.00
    1COT21 LABP4     5.50
    1COT21 LOTCT     1.00
    1COT21 UBCOT     1.00
    1COT21 LBCOT-    1.00
    1COT22 Z1    -  57.05
    1COT22 SOIL2     1.00
    1COT22 LABP4     8.30
    1COT22 FERT      4.00
    1COT22 LOTCT     1.00
    1COT22 UBCOT     1.00
    1COT22 LBCOT-    1.00
    2COT11 Z1    -  42.13
    2COT11 SOIL1     1.00
    2COT11 LOTCT     1.00
    2COT11 UBCOT     1.00
    2COT11 LBCOT-    1.00
    2COT11 S2COT     2.79
    2COT12 Z1    - 144.33
    2COT12 SOIL1     1.00
    2COT12 FERT      4.00
    2COT12 LOTCT     1.00
    2COT12 UBCOT     1.00
    2COT12 LBCOT-    1.00
    2COT12 S2COT     6.51
    2COT21 Z1    -  42.52
```

(Listing Sheet 2)

```
2COT21 SOIL2     1.00
2COT21 LOTCT     1.00
2COT21 UBCOT     1.00
2COT21 LBCOT-    1.00
2COT21 S2COT     2.73
2COT22 Z1    -  71.60
2COT22 SOIL2     1.00
2COT22 FERT      4.00
2COT22 LOTCT     1.00
2COT22 UBCOT     1.00
2COT22 LBCOT-    1.00
2COT22 S2COT     4.89
1OAT11 Z1    -   3.80
1OAT11 SOIL1     1.00
1OAT11 UBOAT     1.00
1OAT11 LBOAT-    1.00
1OAT12 Z1    -  30.63
1OAT12 SOIL1     1.00
1OAT12 FERT      4.00
1OAT12 UBOAT     1.00
1OAT12 LBOAT-    1.00
1OAT21 Z1    -   2.09
1OAT21 SOIL2     1.00
1OAT21 UBOAT     1.00
1OAT21 LBOAT-    1.00
1OAT22 Z1    -  21.45
1OAT22 SOIL2     1.00
1OAT22 FERT      4.00
1OAT22 UBOAT     1.00
1OAT22 LBOAT-    1.00
1SBG11 Z1    -  51.07
1SBG11 SOIL1     1.00
1SBG11 LABP4     1.10
1SBG11 UBSBG     1.00
1SBG11 LBSBG-    1.00
1SBG21 Z1    -  24.72
1SBG21 SOIL2     1.00
1SBG21 LABP4     1.10
1SBG21 UBSBG     1.00
1SBG21 LBSBG-    1.00
1ALF11 Z1    -  23.66
1ALF11 SOIL1     1.00
1ALF11 LABP4     0.70
1ALF11 UBALF     1.00
1ALF11 LBALF-    1.00
1RIC21 Z1    -  60.91
1RIC21 SOIL2     1.00
1RIC21 LOTRC     1.00
1RIC21 UBRIC     1.00
1RIC21 LBRIC-    1.00
1RIC22 Z1    - 144.81
1RIC22 SOIL2     1.00
1RIC22 FERT      4.00
1RIC22 LOTRC     1.00
1RIC22 UBRIC     1.00
1RIC22 LBRIC-    1.00
FIRST B
  C1     SOIL1  500.
  C1     SOIL2 1500.
  C1     LABP4 9900.
  C1     FERT  1525.
```

(Listing Sheet 3)

```
  C1     LOTCT   700.
  C1     LOTRC    80.
  C1     UBCOT   880.
  C1     UBOAT   520.
  C1     UBSBG   650.
  C1     UBALF   230.
  C1     UBRIC   150.
  C1     LBCOT-  640.
  C1     LBOAT-  296.
  C1     LBSBG-  370.
  C1     LBALF-  152.
  C1     LBRIC-   79.
  C1     S2COT   175.
EOF
```

c. Agenda

The Agenda for this example consist of five Agendum cards as follows, with punching starting in card column one.

```
HEADING                  USDA LP/90      EXAMPLE 1    SINGLE LP PROBLEM
INPUT,1,,COUNTS,REWIND
CONTROLS                                        30
NORMAL
OUTPUT
```

The first of these is a simple instruction to print the comments in card columns 13 to 72 at the top of each output page. This message actually is printed in BCD on the O-tape in the appropriate places. But when the latter is listed under "program control" the heading will appear on each page of the output listing.

The second Agendum card instructs LP/90 to (1) read the I-tape (as shown in Table 20 above), (2) convert it to binary, (3) prepare a special analysis on the elements of the matrix which is helpful for checking purposes (COUNTS), and (4) rewind the I-tape when done. The "1" following "INPUT," indicates that a single objective function row has been input in the LP matrix. The two commas following the "1" indicate that the first job on the I-tape is to be processed. If a number were placed between these commas, that number of complete sets of input (= to the number of EOF cards) would be skipped before processing.

The CONTROLS card is used to determine the amount of printed output, the frequency of reinversion, and certain other control information. Reinversion may be thought of simply as a round-off error correction and speed procedure in the NORMAL algorithm. We have specified that this round-off error correction take place every 30 iterations. By leaving the rest of the card blank we have specified the minimum of output: namely, an on line message every 10th iteration, an off line message <u>every</u> iteration and a full iteration print on the <u>final</u> iteration (which gives the optimal solution). The reader may consult Part V for other optional uses of this card.

The NORMAL card calls for the solution of the LP problem prepared by INPUT from the I-tape and now in binary on the A-tape. NORMAL does not use the I-tape, but the binary A-tape, prepared by the preceding INPUT Agendum. Finally, OUTPUT instructs LP/90 to prepare the solution information in BCD on the O-tape for off line listing.

These Agendum cards are placed in the card reader and the I-tape mounted on tape unit B1. The machine operator then pushes the "Load Tape" button and LP/90 proceeds to follow the instructions as specified, producing at the end the desired O-tape.

All of these Agendum items used in this example are on the SHARE LP/90 System, as well as on USDA LP/90.

d. Output

The listing of the O-tape yields four pages of output. These are reproduced exactly in Table 21, except that the first three pages of output, which consume only a few lines each, have been overlapped so as to consume less space. The page numbers are at the right-most part of the heading line. The heading read in by the HEADING Agendum does indeed appear at the beginning of each new page, as advertised.

Consider now the output page by page. The two lines following the heading line are meaningless for the two first pages. Their use should be obvious in the discussion below of output pages 3 and 4. The fourth print line on page 1 is merely the second Agendum card as read from the card reader. The fifth line gives the time that the INPUT Agendum began processing. This time was 780.89 minutes since midnight. The title card on the I-tape is printed next.

Output page 2 presents the output produced by the COUNTS option on the INPUT Agendum. This output gives (1) the number of non-zero elements in each row of the LP matrix (exclusive of the right-hand-side), and (2) the number of non-zero elements in each column of the LP matrix

Table 21: O-Tape Listing for Single LP Example

(Listing Sheet 1)

```
        USDA LP/90    EXAMPLE  1   SINGLE LP PROBLEM                                PAGE     1

 TOTAL  NO. ETA  ROW      CURRENT  CHOSEN   VECTR  NEG.  C/V CURRENT D/J
 ITERS ETAS REC IDENT.     VALUE   VECTOR   REMVD  DJ-S  NO.  THETA/PHI

     INPUT,1,,COUNTS,REWIND

      TIME 780.89

     * USDA LP/90   INPUT  LP MATRIX FOR  EXAMPLE  1
```

(Listing Sheet 2)

```
        USDA LP/90    EXAMPLE  1   SINGLE LP PROBLEM                                PAGE     2

  NUMBER OF ELEMENTS IN EACH ROW EXCLUSIVE OF B(I)S.

 COUNT  ROW  COUNT  ROW  COUNT  ROW  COUNT  ROW  COUNT  ROW  COUNT  ROW  COUNT  ROW  COUNT  ROW  COUNT  ROW
  17   Z1       9  SOIL1   10  SOIL2    8  LABP4    8  FERT     9  LOTCT    3  LOTRC    9  UBCOT    5  UBOAT
   3  UBSBG     2  UBALF    3  UBRIC    9  LBCOT    5  LBOAT    3  LBSBG    2  LBALF    3  LBRIC    5  S2COT

  VECTORS IN PROBLEM WITH ELEMENT COUNTS

 CNT   NAME  CNT   NAME  CNT   NAME  CNT   NAME  CNT   NAME  CNT   NAME  CNT   NAME  CNT   NAME  CNT   NAME
   6  1COT11   7  1COT12   6  1COT21   7  1COT22   6  2COT11   7  2COT12   6  2COT21   7  2COT22   4  1OAT11

   5  1OAT12   4  1OAT21   5  1OAT22   5  1SBG11   5  1SBG21   5  1ALF11   5  1RIC21   6  1RIC22   1 + SOIL1

   1 + SOIL2   1 + LABP4   1 + FERT    1 + LOTCT   1 + LOTRC   1 + UBCOT   1 + UBOAT   1 + UBSBG   1 + UBALF

   1 + UBRIC   1 + LBCOT   1 + LBOAT   1 + LBSBG   1 + LBALF   1 + LBRIC   1 + S2COT

  DENSITY OF PROBLEM IN PER CENT        18.464052

  THIS PROBLEM HAS   113 ELEMENTS,      34 VECT. AND      18 ROWS.
```

(Listing Sheet 3)

```
        USDA LP/90    EXAMPLE  1   SINGLE LP PROBLEM                                PAGE     3

 TOTAL  NO. ETA  ROW      CURRENT  CHOSEN   VECTR  NEG.  C/V CURRENT D/J
 ITERS ETAS REC IDENT.     VALUE   VECTOR   REMVD  DJ-S  NO.  THETA/PHI

     CONTROLS                                30

     NORMAL

      TIME 781.02

     SOLUTION FEASIBLE
     6    6   0  Z1      41828.610  1SBG21 + LBSBG    2      -1.0000000
```

(Listing Sheet 4)

USDA LP/90   EXAMPLE 1   SINGLE LP PROBLEM   PAGE 4

TOTAL ITERS	NO. ETAS	ETA REC	ROW IDENT.	CURRENT VALUE	CHOSEN VECTOR	VECTR REMVD	RHS NO.	C/V NO.	CURRENT D/J THETA/PHI	OPTIMAL PRIMA-DUAL	TIME 781.06
16	16	0	Z1	83187.851			1		* * * *		

J(H)	BETA(H)	ROW(I)	PI(I)	B(I)
0 00000	83187.85102139	Z1	1.00000000	.
1COT12	274.36827957	+ SOIL1	28.44000000	500.00000000
+ LBOAT	122.00000000	+ SOIL2	2.09000000	1500.00000000
+ LABP4	3181.50322580	+ LABP4	.	9900.00000000
1OAT21	418.00000000	+ FERT	13.04500000	1525.00000000
+ LBCOT	60.00000000	+ LOTCT	34.17000000	700.00000000
+ LBRIC	1.00000000	+ LOTRC	90.54000000	80.00000000
+ UBCOT	180.00000000	+ UBCOT	.	880.00000000
+ UBOAT	102.00000000	+ UBOAT	.	520.00000000
+ LBSBG	280.00000000	+ UBSBG	22.63000000	650.00000000
+ UBALF	78.00000000	+ UBALF	.	230.00000000
+ UBRIC	70.00000000	+ UBRIC	.	150.00000000
1COT21	398.75000000	+ LBCOT	.	640.00000000-
1SBG11	46.75000000	+ LBOAT	.	296.00000000-
1SBG21	603.25000000	+ LBSBG	.	370.00000000-
1ALF11	152.00000000	+ LBALF	4.78000000	152.00000000-
1RIC22	80.00000000	+ LBRIC	.	79.00000000-
2COT12	26.88172043	+ S2COT	4.53763441	175.00000000

and in the proper slack vectors created automatically by INPUT (as specified on the row identification cards). Including the cost row, there are 18 rows in the LP matrix. Including the slack vectors, there are 34 vectors (the slacks are given names corresponding to the active constraint rows). In all there are 612 (= 18 x 34) possible entries, but only 113 non-zero elements. The ratio of the non-zero elements to the total number of possible entries gives the matrix "density" of 18%. As LP problems go this is a fairly "dense" problem, densities of about 5% being common.

On page 3 the CONTROLS and NORMAL Agendum cards are printed as read, and the time when the NORMAL Agendum began printed. The time elapsed from the beginning of INPUT to this point is .13 minutes, or 7.80 seconds! The last line shows information about the iteration on which the feasible solution was found: a feasible solution was found on the sixth iteration, the value of this feasible solution is $41,828.610. During this iteration column 1SBG21 replaced the slack vector + LBSBG in the basis.[24]

On page 4 the results of the final optimal solution are displayed. The optimal solution was achieved on the 16th iteration and the value of the optimal solution is $83,187.85. The computation was completed at time 781.06 with the elapse of .04 minutes, or 2.40 seconds!

The full solution print follows the iteration information. The J(H) column identifies the variable (column) name in the optimal basis and the BETA(H) column gives the value of each non-zero primal variable in the optimal (basic) solution. The ROW(1) column gives the row identification for each row in the matrix, the PI(I) column gives the optimal dual variable (marginal value) corresponding to each constraint element in the right-hand-side, and finally the B(I) column gives the right-hand-side elements or constraint vector for the problem. (The number one found opposite the objective function row in the PI(I) column indicates not a marginal value or dual variable, but merely the row of the LP matrix that was minimized.

Table 22 summarizes the solution (rounded to hundreds) including the implied zero values for the non-basis variables. There were eight non-zero process levels, corresponding to exactly eight non-zero marginal values. The slack variables give the amount of unused capacity. A positive

---

[24] For a discussion of bases see Part II, Section 5.d.

slack means a "loose constraint" and zero dual variable. A zero slack means a tight constraint and a positive dual variable. The reader can easily verify that the value of the primal solution equals the value of the dual solution by multiplying the dual variables by their respective right-hand-side elements and adding cumulatively. We then see confirmed in this example the duality theorem summarized in Part II above and we are left with no doubt that LP/90 has indeed given a desired optimal solution.[25]

---

[25] Where the optimal solution is nonunique, LP/90 gives only one such solution.

Table 22: Summary of the Optimal Solution of Single LP Example

Process Name	Acres Planted (Process Level)	Constraint Name (Slack Name)	Marginal Value (Dual Variable)	Amount Unused (Slack Var)
1COT11	0	SOIL 1	28.44	0
1COT12	274.37	SOIL 2	2.09	0
1COT21	398.75	LABP 4	0	3181.50
1COT22	0	FERT	13.04	0
2COT11	0	LOTCT	34.17	0
2COT12	26.88	LOTRC	90.54	0
2COT21	0	UBCOT	0	180.00
2COT22	0	UBOAT	0	102.00
1OAT11	0	UBSBG	22.63	0
1OAT12	0	UBALF	0	78.00
1OAT21	418.00	UBRIC	0	70.00
1OAT22	0	LBCOT	0	60.00
1SBG11	46.75	LBOAT	0	122.00
1SBG21	603.25	LBSBG	0	280.00
1ALF11	152.00	LBALF	4.78	0
1RIC21	0	LBRIC	0	1.00
1RIC22	80.00	S2COT	4.54	0

e. Remarks

While the real speed of USDA LP/90 can be appreciated fully only in the computation of large problems, it is still impressive to observe that the entire computation time for this run was about ten seconds. This includes both the conversion of the I-tape to binary and the problem solution. The fact that sixteen iterations took only 2.40 seconds, or .15 seconds per iteration, is indicative of what the user can expect on larger problems.

This time could have been further reduced by omitting the COUNTS option from the INPUT Agendum control card. It was used here for illustrative purposes. The information produced by this option (page 2 of the output listing) is very useful for "debugging" a model and its data. For example, a solution might have been obtained that was patently ridiculous. Such a result might have occurred simply because a single matrix element card was inadvertently omitted when the I-tape was prepared. This error could be immediately discerned by examining the COUNTS output.

2. MULTIPLE LP PROBLEMS

a. Purpose

The solution of several LP problems during the same computer run is illustrated in this example. A PROP Agendum routine that allows modification of an existing I-tape is used.

b. Data

The data for this example include all of the data used in Example 1, plus an additional objective function, and two additional RHS or Constraint Vectors. Table 23 shows the augmented LP matrix to be used. This table is identical to Table 18 except that the new objective function row bZ2bbb has been inserted and two new right-hand-sides appended at the right.

TABLE 23

LP MATRIX FOR MULTIPLE LP EXAMPLE

Row Symbol	Matrix Column Symbol											
	1COT11	1COT12	1COT21	1COT22	2COT11	2COT12	2COT21	2COT22	1OAT11	1OAT12	1OAT21	1OAT22
Z1	- 35.16	-114.79	- 36.26	- 57.05	- 42.13	-144.33	- 42.52	- 71.60	- 3.80	- 30.63	- 2.09	- 21.45
Z2	- 41.86	-119.59	- 42.04	- 64.38	- 46.93	-138.98	- 47.05	- 74.64	- 6.81	- 36.12	- 5.11	- 30.17
SOIL1	1.00	1.00			1.00	1.00			1.00	1.00		
SOIL2			1.00	1.00			1.00	1.00			1.00	1.00
LABP4	5.80	13.50	5.50	8.30								
FERT		4.00		4.00		4.00		4.00		4.00		4.00
LOTCT	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00				
LOTRC												
UBCOT	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00				
UBOAT									1.00	1.00	1.00	1.00
UBSBG												
UBALF												
UBRIC												
LBCOT	- 1.00	- 1.00	- 1.00	- 1.00	- 1.00	- 1.00	- 1.00	- 1.00				
LBOAT									- 1.00	- 1.00	- 1.00	- 1.00
LBSBG												
LBALF												
LBRIC												
S2COT					2.79	6.51	2.73	4.89				

TABLE 23 -Continued

Row Symbol	Matrix Column Symbol					Constraint Type	Right-Hand-Sides		
	1SBG11	1SBG21	1ALF11	1RIC21	1RIC22		FIRST B	NEXT B,2	NEXT B,3
Z1	- 51.07	- 24.72	- 23.66	- 60.91	-144.81				
Z2	- 65.87	- 33.27	- 34.52	- 55.32	-117.93				
SOIL1	1.00		1.00			≤	500	1500	1500
SOIL2		1.00		1.00	1.00	≤	1500	500	500
LABP4	1.10	1.10	0.70			≤	9900	9900	9900
FERT					4.00	≤	1525	1525	1525
LOTCT						≤	700	700	5000
LOTRC				1.00	1.00	≤	80	80	5000
UBCOT						≤	880	880	880
UBOAT						≤	520	520	520
UBSBG	1.00	1.00				≤	650	650	650
UBALF			1.00			≤	230	230	230
UBRIC				1.00	1.00	≤	150	150	150
LBCOT						≤	- 640	- 640	- 640
LBOAT						≤	- 296	- 296	- 296
LBSBG	- 1.00	- 1.00				≤	- 370	- 370	- 370
LBALF			- 1.00			≤	- 152	- 152	- 152
LBRIC				- 1.00	- 1.00	≤	- 79	- 79	- 79
S2COT						≤	175	175	175

c. Agenda

Four separate LP problems will be solved in this run. First, the Z1 objective function is maximized using the 2nd RHS. Second, the Z2 objective function is maximized using the 2nd RHS. Third, the Z1 objective function is maximized using the 3rd RHS. Finally, the Z2 objective function is maximized using the 3rd RHS. The complete Agenda, including the I-tape "correction" routine is listed in Table 24. Each of its Agendum control cards will now be discussed in turn.

The first of these is already familiar. The second, CORRECT.BCD, is a PROP routine. It enables us to ammend the I-tape created for the preceding run (Section 1) during the computer run, saving the need to re-card-to-tape the augmented data. The "B7" following the Agendum name (note the decimal in the latter) informs the computer that it will find the I-tape to be "corrected" on tape unit B7. The new augmented I-tape is to be written on tape unit B1. The corrections are read from the card reader (CR). We shall avoid interpreting the entry "4" until we come to Section 5. The REWIND causes all of the tapes to be rewound. The original I-tape is unloaded and can be dismounted during the run. This Agendum card is followed by a series of cards which it uses as input. If the "CR" entry on the Agendum card had contained, let us say, A7 the correction cards would have been read from tape unit A7.

The first card of the group following CORRECT.BCD is MATCH,7,18. This card tells the Agendum, that in searching for changes in the I-tape it should compare only card columns 7 through 18. (These include the row and column name fields.) The next card says search until you find a card with Z1 in column 14 and 15. The next card says insert the new record found on the card following. After the Z2 card which will be inserted as a row identification for the new objective function to be inserted, comes a match card. When this card record is found on the I-tape the card following the INSERT, 1 card is inserted, and so on.

In general, the number following the INSERT, gives the number of card images to be inserted on the new I-tape following the preceding card used for matching.

By comparing the data for CORRECT.BCD against Table 20, the reader can see that we have effected the entry of the new objective function elements in each column of input data, and the new RHS elements after the original Right-Hand-Side. Table 25 shows the listing of the new I-tape created by this Agendum from the old I-tape and the new card data.

Following CORRECT.BCD is the INPUT Agendum control card. Its appearance is the same as in Example 1 except that a "2" replaces "1" following the Agendum name. This is because our new I-tape contains two objective function rows.

The CONTROLS Agendum is the same as before. Note that it appears only once in the Agenda even though four separate problems are solved. This is because the control parameters set by it remain the same until a new CONTROLS Agendum or a new INPUT Agendum is encountered. The latter always resets the control parameters to their programmed values so that CONTROLS should always follow INPUT if control parameters different than the programmed values are desired.

The asterisk in the NORMAL cards indicates that objective function row names rather than numbers are to be used. Optionally the NORMAL cards could have been punched as follows with exactly the same result.

```
NORMAL,0,2  or  NORMAL,,2
NORMAL,1,2
NORMAL,0,3
NORMAL,1,3.
```

The row number on an LP matrix begins with zero. The first objective function is given a row index of zero, the second row index of one, and so on. In Example 1, no row or right-hand-side index was given because the first (and only) one was used.

Note that all of the Agendum routines in this example -- except CORRECT.BCD -- are SHARE routines. The exception is a PROP or Proprietary Routine.

Table 24: Agenda for Multiple LP Example

(Listing Sheet 1)

```
HEADING      USDA LP/90    EXAMPLE  2   MULTIPLE LP PROBLEMS
CORRECT.BCD,B7,B1,CR,,4,,REWIND
MATCH,7,18
          Z1
INSERT,1
          Z2
      1COT11 Z1   -  35.16                        EXAMPLE1
INSERT,1
      1COT11 Z2   -  41.86
      1COT12 Z1   - 114.79                        EXAMPLE1
INSERT,1
      1COT12 Z2   - 119.59
      1COT21 Z1   -  36.26                        EXAMPLE1
INSERT,1
      1COT21 Z2   -  42.04
      1COT22 Z1   -  57.05                        EXAMPLE1
INSERT,1
      1COT22 Z2   -  64.38
      2COT11 Z1   -  42.13                        EXAMPLE1
INSERT,1
      2COT11 Z2   -  46.93
      2COT12 Z1   - 144.33                        EXAMPLE1
INSERT,1
      2COT12 Z2   - 138.98
      2COT21 Z1   -  42.52                        EXAMPLE1
INSERT,1
      2COT21 Z2   -  47.05
      2COT22 Z1   -  71.60                        EXAMPLE1
INSERT,1
      2COT22 Z2   -  74.64
      1OAT11 Z1   -   3.80                        EXAMPLE1
INSERT,1
      1OAT11 Z2   -   6.81
      1OAT12 Z1   -  30.63                        EXAMPLE1
INSERT,1
      1OAT12 Z2   -  36.12
      1OAT21 Z1   -   2.09                        EXAMPLE1
INSERT,1
      1OAT21 Z2   -   5.11
      1OAT22 Z1   -  21.45                        EXAMPLE1
INSERT,1
      1OAT22 Z2   -  30.17
      1SBG11 Z1   -  51.07                        EXAMPLE1
INSERT,1
      1SBG11 Z2   -  65.87
      1SBG21 Z1   -  24.72                        EXAMPLE1
INSERT,1
      1SBG21 Z2   -  33.27
      1ALF11 Z1   -  23.66                        EXAMPLE1
INSERT,1
      1ALF11 Z2   -  34.52
      1RIC21 Z1   -  60.91                        EXAMPLE1
INSERT,1
      1RIC21 Z2   -  55.32
      1RIC22 Z1   - 144.81                        EXAMPLE1
INSERT,1
      1RIC22 Z2   - 117.93
       C1    S2COT  175.
INSERT,36
```

(Listing Sheet 2)

```
NEXT B,2
       C2     SOIL1 1500.
       C2     SOIL2  500.
       C2     LABP4 9900.
       C2     FERT  1525.
       C2     LOTCT  700.
       C2     LOTRC   80.
       C2     UBCOT  880.
       C2     LBCAT  520.
       C2     UBSBG  650.
       C2     UBALF  230.
       C2     UBRIC  150.
       C2     LBCOT- 640.
       C2     LBCAT- 296.
       C2     LBSBG- 370.
       C2     LBALF- 152.
       C2     LBRIC-  79.
       C2     S2COT  175.
NEXT B,3
       C3     SOIL1 1500.
       C3     SOIL2  500.
       C3     LABP4 9900.
       C3     FERT  1525.
       C3     LOTCT 5000.
       C3     LOTRC 5000.
       C3     UBCOT  880.
       C3     LBCAT  520.
       C3     UBSBG  650.
       C3     UBALF  230.
       C3     UBRIC  150.
       C3     LBCOT- 640.
       C3     LBCAT- 296.
       C3     LBSBG- 370.
       C3     LBALF- 152.
END***
INPUT,2,,COUNTS,REWIND
CONTROLS                                   30
TITLE        USDA LP/90    EXAMPLE  2     PROBLEM 1
NORMAL,* Z1    ,2
TITLE        USDA LP/90    EXAMPLE  2     PROBLEM 2
NORMAL,* Z2    ,2
TITLE        USDA LP/90    EXAMPLE  2     PROBLEM 3
NORMAL,* Z1    ,3
TITLE        USDA LP/90    EXAMPLE  2     PROBLEM 4
NORMAL,* Z2    ,3
OUTPUT
```

Table 25: Listing of I-Tape Created by Correct. BCD

(Listing Sheet 1)

```
* USDA LP/90    INPUT   LP MATRIX FOR   EXAMPLE   1
ROW ID
              Z1
              Z2
            + SOIL1
            + SOIL2
            + LABP4
            + FERT
            + LOTCT
            + LOTRC
            + UBCOT
            + UBOAT
            + UBSBG
            + UBALF
            + UBRIC
            + LBCOT
            + LBOAT
            + LBSBG
            + LBALF
            + LBRIC
            + S2COT
MATRIX
        1COT11 Z1    -  35.16
        1COT11 Z2    -  48.16
        1COT11 SOIL1     1.00
        1COT11 LABP4     5.80
        1COT11 LOTCT     1.00
        1COT11 UBCOT     1.00
        1COT11 LBCOT-    1.00
        1COT12 Z1    - 114.79
        1COT12 Z2    - 119.59
        1COT12 SOIL1     1.00
        1COT12 LABP4    13.50
        1COT12 FERT      4.00
        1COT12 LOTCT     1.00
        1COT12 UBCOT     1.00
        1COT12 LBCOT-    1.00
        1COT21 Z1    -  36.26
        1COT21 Z2    -  42.40
        1COT21 SOIL2     1.00
        1COT21 LABP4     5.50
        1COT21 LOTCT     1.00
        1COT21 UBCOT     1.00
        1COT21 LBCOT-    1.00
        1COT22 Z1    -  57.05
        1COT22 Z2    -  64.38
        1COT22 SOIL2     1.00
        1COT22 LABP4     8.30
        1COT22 FERT      4.00
        1COT22 LOTCT     1.00
        1COT22 UBCOT     1.00
        1COT22 LBCOT-    1.00
        2COT11 Z1    -  42.13
        2COT11 Z2    -  46.93
        2COT11 SOIL1     1.00
        2COT11 LOTCT     1.00
        2COT11 UBCOT     1.00
        2COT11 LBCOT-    1.00
        2COT11 S2COT     2.79
        2COT12 Z1    - 144.33
        2COT12 Z2    - 138.98
```

(Listing Sheet 2)

```
2COT12 SOIL1     1.00
2COT12 FERT      4.00
2COT12 LOTCT     1.00
2COT12 UBCOT     1.00
2COT12 LBCOT-    1.00
2COT12 S2COT     6.51
2COT21 Z1    -  42.52
2COT21 Z2    -  47.05
2COT21 SOIL2     1.00
2COT21 LOTCT     1.00
2COT21 UBCOT     1.00
2COT21 LBCOT-    1.00
2COT21 S2COT     2.73
2COT22 Z1    -  71.60
2COT22 Z2    -  74.64
2COT22 SOIL2     1.00
2COT22 FERT      4.00
2COT22 LOTCT     1.00
2COT22 UBCOT     1.00
2COT22 LBCOT-    1.00
2COT22 S2COT     4.89
1OAT11 Z1    -   3.80
1OAT11 Z2    -   6.81
1OAT11 SOIL1     1.00
1OAT11 UBOAT     1.00
1OAT11 LBOAT-    1.00
1OAT12 Z1    -  30.63
1OAT12 Z2    -  36.12
1OAT12 SOIL1     1.00
1OAT12 FERT      4.00
1OAT12 UBOAT     1.00
1OAT12 LBOAT-    1.00
1OAT21 Z1    -   2.09
1OAT21 Z2    -   5.11
1OAT21 SOIL2     1.00
1OAT21 UBOAT     1.00
1OAT21 LBOAT-    1.00
1OAT22 Z1    -  21.45
1OAT22 Z2    -  30.17
1OAT22 SOIL2     1.00
1OAT22 FERT      4.00
1OAT22 UBOAT     1.00
1OAT22 LBOAT-    1.00
1SBG11 Z1    -  51.07
1SBG11 Z2    -  65.87
1SBG11 SOIL1     1.00
1SBG11 LABP4     1.10
1SBG11 UBSBG     1.00
1SBG11 LBSBG-    1.00
1SBG21 Z1    -  24.72
1SBG21 Z2    -  33.27
1SBG21 SOIL2     1.00
1SBG21 LABP4     1.10
1SBG21 UBSBG     1.00
1SBG21 LBSBG-    1.00
1ALF11 Z1    -  23.66
1ALF11 Z2    -  34.52
1ALF11 SOIL1     1.00
1ALF11 LABP4     0.70
1ALF11 UBALF     1.00
1ALF11 LBALF-    1.00
```

Table 25: Listing of I-Tape Created by Correct. BCD--Continued

(Listing Sheet 3)

```
     1RIC21 Z1    -  60.91
     1RIC21 Z2    -  55.32
     1RIC21 SOIL2     1.00
     1RIC21 LOTRC     1.00
     1RIC21 UBRIC     1.00
     1RIC21 LBRIC-    1.00
     1RIC22 Z1    - 144.81
     1RIC22 Z2    - 117.93
     1RIC22 SOIL2     1.00
     1RIC22 FERT      4.00
     1RIC22 LOTRC     1.00
     1RIC22 UBRIC     1.00
     1RIC22 LBRIC-    1.00
FIRST B
      C1     SOIL1  500.
      C1     SOIL2 1500.
      C1     LABP4 9900.
      C1     FERT  1525.
      C1     LOTCT  700.
      C1     LOTRC   80.
      C1     UBCOT  880.
      C1     UBOAT  520.
      C1     UBSBG  650.
      C1     UBALF  230.
      C1     UBRIC  150.
      C1     LBCOT- 640.
      C1     LBOAT- 296.
      C1     LBSBG- 370.
      C1     LBALF- 152.
      C1     LBRIC-  79.
      C1     S2COT  175.
NEXT B,2
      C2     SOIL1 1500.
      C2     SOIL2  500.
      C2     LABP4 9900.
      C2     FERT  1525.
      C2     LOTCT  700.
      C2     LOTRC   80.
      C2     UBCOT  880.
      C2     UBOAT  520.
      C2     UBSBG  650.
      C2     UBALF  230.
      C2     UBRIC  150.
      C2     LBCOT- 640.
      C2     LBOAT- 296.
      C2     LBSBG- 370.
      C2     LBALF- 152.
      C2     LBRIC-  79.
      C2     S2COT  175.
NEXT B,3
      C3     SOIL1 1500.
      C3     SOIL2  500.
      C3     LABP4 9900.
      C3     FERT  1525.
      C3     LOTCT 5000.
      C3     LOTRC 5000.
      C3     UBCOT  880.
      C3     UBOAT  520.
      C3     UBSBG  650.
      C3     UBALF  230.
      C3     UBRIC  150.
```

(Listing Sheet 4)

```
      C3     LBCOT- 640.
      C3     LBOAT- 296.
      C3     LBSBG- 370.
      C3     LBALF- 152.
      C3     LBRIC-  79.
      C3     S2COT  175.
EOF
```

d. Output

The output for this example is like that for Example 1, except there is more of it. An abridged listing of the output appears on the next four pages (Table 26). The first of these presents page 3 of the Output listing containing the results of the COUNTS option. Note that the new row appears with its 17 elements, and that each column (except the slack) has an additional element. Thus, COUNTS gives us a quick check on whether or not CORRECT.BCD inserted the correct number of cards.

The Agendum routine TITLE causes pages to be renumbered starting with one, so that the page numbers of the full solution print-outs are not in ascending order. The four Output pages containing the four full LP solutions are reproduced two per page in Table 26. The solutions are summarized in Table 27 . The first problem required 15 iterations to gain the optimal solution, problem 2 required only one iteration. This is because the second NORMAL begins with the existing solution in core, and the optimal basis is the same for both problems. The optimal solution for the 3rd problem was obtained in 11 iterations and the fourth in one, for the same reason as in problem 2.

Table 26: Abridged O-Tape Listing for Multiple LP Example

(Listing Sheet 1)

USDA LP/90 EXAMPLE 2 MULTIPLE LP PROBLEMS PAGE 3

NUMBER OF ELEMENTS IN EACH ROW EXCLUSIVE OF B(I)S.

COUNT	ROW	COUNT	ROW	COUNT	ROW	COUNT	ROW	COUNT	ROW	COUNT	ROW	COUNT	ROW	COUNT	ROW	COUNT	ROW
17	Z1	17	Z2	9	SOIL1	10	SOIL2	8	LABP4	8	FERT	9	LOTCT	3	LOTRC	9	UBCOT
5	UBOAT	3	UBSBG	2	UBALF	3	UBRIC	9	LBCOT	5	LBOAT	3	LBSBG	2	LBALF	3	LBRIC
5	S2COT																

VECTORS IN PROBLEM WITH ELEMENT COUNTS

CNT	NAME	CNT	NAME	CNT	NAME	CNT	NAME	CNT	NAME	CNT	NAME	CNT	NAME	CNT	NAME	CNT	NAME
7	1COT11	8	1COT12	7	1COT21	8	1COT22	7	2COT11	8	2COT12	7	2COT21	8	2COT22	5	1OAT11
6	1OAT12	5	1OAT21	6	1OAT22	6	1SBG11	6	1SBG21	6	1ALF11	6	1RIC21	7	1RIC22	1 +	SOIL1
1 +	SOIL2	1 +	LABP4	1 +	FERT	1 +	LOTCT	1 +	LOTRC	1 +	UBCOT	1 +	UBOAT	1 +	UBSBG	1 +	UBALF
1 +	UBRIC	1 +	LBCOT	1 +	LBOAT	1 +	LBSBG	1 +	LBALF	1 +	LBRIC	1 +	S2COT				

DENSITY OF PROBLEM IN PER CENT 20.123839

THIS PROBLEM HAS 130 ELEMENTS, 34 VECT. AND 19 ROWS.

Table 26: Abridged O-Tape Listing for Multiple LP Example--Continued

(Listing Sheet 2)

```
          USDA LP/90   EXAMPLE  2     PROBLEM 1                                                          PAGE      2

 TOTAL  NO. ETA  ROW     CURRENT  CHOSEN   VECTR   RHS C/V  CURRENT D/J
 ITERS ETAS REC IDENT.    VALUE   VECTOR   REMVD   NO. NO.   THETA/PHI  OPTIMAL PRIMA-DUAL             TIME 809.98
   15   15   0  Z1      101311.01                   2          * * * *

   J(H)          BETA(H)        ROW(I)      PI(I)              B(I)
 0 00000   101311.01102137       Z1        1.00000000            .
 0 00000   115779.94905896       Z2         .                    .
 + LBSBG      280.00000000    + SOIL1      3.80000000      1500.00000000
  1OAT21       21.25000000    + SOIL2      2.09000000       500.00000000
 + LABP4     3126.90322580    + LABP4       .              9900.00000000
  1COT12      274.36827957    + FERT      19.20500000      1525.00000000
 + LBCOT       60.00000000    + LOTCT     34.17000000       700.00000000
 + LBRIC        1.00000000    + LOTRC     65.90000000        80.00000000
 + UBCOT      180.00000000    + UBCOT       .               880.00000000
 + UBOAT      180.00000000    + UBOAT       .               520.00000000
  1COT21      398.75000000    + UBSBG     47.27000000       650.00000000
 + LBALF       78.00000000    + UBALF     19.86000000       230.00000000
 + UBRIC       70.00000000    + UBRIC       .               150.00000000
 + LBOAT       44.00000000    + LBCOT       .               640.00000000-
  1OAT11      318.75000000    + LBOAT       .               296.00000000-
  1SBG11      650.00000000    + LBSBG       .               370.00000000-
  1ALF11      230.00000000    + LBALF       .               152.00000000-
  1RIC22       80.00000000    + LBRIC       .                79.00000000-
  2COT12       26.88172043    + S2COT      4.53763441       175.00000000
```

(Listing Sheet 3)

```
          USDA LP/90   EXAMPLE  2     PROBLEM 2                                                          PAGE      2

 TOTAL  NO. ETA  ROW     CURRENT  CHOSEN   VECTR   RHS C/V  CURRENT D/J
 ITERS ETAS REC IDENT.    VALUE   VECTOR   REMVD   NO. NO.   THETA/PHI  OPTIMAL PRIMA-DUAL             TIME 810.06
   16   16   0  Z2      116839.15                   2          * * * *

   J(H)          BETA(H)        ROW(I)      PI(I)              B(I)
 0 00000   100744.61102138       Z1         .                    .
 0 00000   116839.14905895       Z2        1.00000000            .
 + LBSBG      280.00000000    + SOIL1      6.81000000      1500.00000000
  1OAT21      101.25000000    + SOIL2      5.11000000       500.00000000
 + LABP4     2486.90322580    + LABP4       .              9900.00000000
  1COT12      354.36827957    + FERT      18.96250000      1525.00000000
 + LBCOT       60.00000000    + LOTCT     36.93000000       700.00000000
 + LBRIC        1.00000000    + LOTRC     50.21000000        80.00000000
 + UBCOT      180.00000000    + UBCOT       .               880.00000000
 + UBOAT      180.00000000    + UBOAT       .               520.00000000
  1COT21      318.75000000    + UBSBG     59.06000000       650.00000000
 + LBALF       78.00000000    + UBALF     27.71000000       230.00000000
 + UBRIC       70.00000000    + UBRIC       .               150.00000000
 + LBOAT       44.00000000    + LBCOT       .               640.00000000-
  1OAT11      238.75000000    + LBOAT       .               296.00000000-
  1SBG11      650.00000000    + LBSBG       .               370.00000000-
  1ALF11      230.00000000    + LBALF       .               152.00000000-
  1RIC21       80.00000000    + LBRIC       .                79.00000000-
  2COT12       26.88172043    + S2COT      2.97849462       175.00000000
```

Table 26: Abridged O-Tape Listing for Multiple LP Example--Continued

(Listing Sheet 4)

USDA LP/90 EXAMPLE 2 PROBLEM 3 PAGE 3

TOTAL ITERS	NO. ETAS	ETA REC	ROW IDENT.	CURRENT VALUE	CHOSEN VECTOR	VECTR REMVD	RHS NO.	C/V NO.	CURRENT D/J THETA/PHI	OPTIMAL PRIMA-DUAL	TIME 810.20
27	27	0	Z1	105671.96			3		* * * *		

J(H)	BETA(H)	ROW(I)	PI(I)	B(I)
0 00000	105671.96352134	Z1	1.00000000	.
0 00000	117805.81905897	Z2	.	.
+ LBSBG	280.00000000	+ SOIL1	35.16000000	1500.00000000
1COT12	204.36827957	+ SOIL2	36.26000000	500.00000000
+ LABP4	3404.27822580	+ LABP4	.	9900.00000000
1RIC22	150.00000000	+ FERT	19.90750000	1525.00000000
+ LOTCT	4248.00000000	+ LOTCT	.	5000.00000000
+ LBRIC	71.00000000	+ LOTRC	.	5000.00000000
+ UBCOT	128.00000000	+ UBCOT	.	880.00000000
+ UBOAT	224.00000000	+ UBOAT	.	520.00000000
1COT21	350.00000000	+ UBSBG	15.91000000	650.00000000
+ UBALF	78.00000000	+ UBALF	.	230.00000000
2COT12	26.88172043	+ UBRIC	28.92000000	150.00000000
+ LBCOT	112.00000000	+ LBCOT	.	640.00000000-
1OAT11	296.00000000	+ LBOAT	31.36000000	296.00000000-
1SBG11	650.00000000	+ LBSBG	.	370.00000000-
1ALF11	152.00000000	+ LBALF	11.50000000	152.00000000-
+ LOTRC	4850.00000000	+ LBRIC	.	79.00000000-
1COT11	170.75000000	+ S2COT	4.53763441	175.00000000

(Listing Sheet 5)

USDA LP/90 EXAMPLE 2 PROBLEM 4 PAGE 2

TOTAL ITERS	NO. ETAS	ETA REC	ROW IDENT.	CURRENT VALUE	CHOSEN VECTOR	VECTR REMVD	RHS NO.	C/V NO.	CURRENT D/J THETA/PHI	OPTIMAL PRIMA-DUAL	TIME 810.29
28	28	0	Z2	120073.82			3		* * * *		

J(H)	BETA(H)	ROW(I)	PI(I)	B(I)
0 00000	105031.46352137	Z1	.	.
0 00000	120073.81905896	Z2	1.00000000	.
+ LBSBG	280.00000000	+ SOIL1	41.86000000	1500.00000000
1COT12	354.36827957	+ SOIL2	42.04000000	500.00000000
+ LABP4	2249.27822580	+ LABP4	.	9900.00000000
1RIC21	150.00000000	+ FERT	19.43250000	1525.00000000
+ LOTCT	4248.00000000	+ LOTCT	.	5000.00000000
+ LBRIC	71.00000000	+ LOTRC	.	5000.00000000
+ UBCOT	128.00000000	+ UBCOT	.	880.00000000
+ UBOAT	224.00000000	+ UBOAT	.	520.00000000
1COT21	350.00000000	+ UBSBG	24.01000000	650.00000000
+ UBALF	78.00000000	+ UBALF	.	230.00000000
2COT12	26.88172043	+ UBRIC	13.28000000	150.00000000
+ LBCOT	112.00000000	+ LBCOT	.	640.00000000-
1OAT11	296.00000000	+ LBOAT	35.05000000	296.00000000-
1SBG11	650.00000000	+ LBSBG	.	370.00000000-
1ALF11	152.00000000	+ LBALF	7.34000000	152.00000000-
+ LOTRC	4850.00000000	+ LBRIC	.	79.00000000-
1COT11	20.75000000	+ S2COT	2.97849462	175.00000000

Table 27: Summary of the Four Optimal Solutions
(Primal Variables Only)

Process Name	Problem 1	Problem 2	Problem 3	Problem 4
1COT11	0	0	170.75	20.75
1COT12	274.37	354.37	204.37	354.37
1COT21	398.75	318.75	350.00	350.00
1COT22	0	0	0	0
2COT11	0	0	0	0
2COT12	26.88	26.88	26.88	26.88
2COT21	0	0	0	0
2COT22	0	0	0	0
1OAT11	318.75	238.75	296.00	296.00
1OAT12	0	0	0	0
1OAT21	21.25	101.25	0	0
1OAT22	0	0	0	0
1SBG11	650.00	650.00	650.00	650.00
1SBG21	0	0	0	0
1ALF11	230.00	230.00	152.00	152.00
1RIC21	0	80.00	0	150.00
1RIC22	80.00	0	150.00	0
Value	101,311.01	116,839.15	105,671.96	120,073.82

e. Remarks

The running time for the complete example was about a minute. Of this, CORRECT.BCD consumed a little over half. INPUT consumed about 10 seconds. The first solution required 1.8 seconds, the second .72, the third 2.88 and the fourth .72 seconds. The balance of the time was devoted to OUTPUT and reading of Agendum routines from the system tape.

The proportionately long time devoted to CORRECT.BCD illustrates a basic principle of LP/90 (as well as other scientific computer programs) computation. The internal computing speed is far faster than tape processing speed. Even though this Agendum has very little computing per se to do, it does have quite a bit of tape to read. At a cost of (approximately) $500.00 per machine hour the cost of CORRECT.BCD in this example was about $5.00. The entire run cost about $8.00. In some cases it will be preferable to make the corrections, insertions or deletions from the original card deck, and to use the off-line card to tape to create the new I-tape. Ordinarily this would be the least expensive of the two options. On the other hand, under some conditions it will be a great convenience and time saver to use the CORRECT.BCD Agendum to create the new I-tape. Thus, the user must balance cost against time saved when using CORRECT.BCD [note also that many corrections can be made by REVISE].

In general, there are certain economics of scale in LP/90. The search for and reading of Agendum routines from the Operations tape are really an overhead cost. As the size of the problem increases this subsidiary processing consumes less and less time relative to the time spent on simplex iterations. Correspondingly, <u>all</u> of the input processing and output processing Agendum routines consume a disproportionately large amount of time on tiny LP problems such as those included in this study. On problems with hundreds of rows and columns their use involves a negligible cost relative to the LP computations themselves. The latter do of course increase more or less exponentially with the number of rows.

3. WRITING REPORTS

a. Purpose

By now the use of the Agendum concept should be familiar. The Agendum Routine is simply a special computer sub-program stored on the Operations tape and called into play--at the user's option--by an Agendum control card. The particular LP and LP related computations have been organized in the form of Agendum routines in LP/90. And, as we saw in the preceding example where CORRECT.BCD was employed, any specialized computation can be adapted to this kind of usage.

It is the purpose of this example to illustrate the use of two of the most ingenious and generally useful of all the Agendum routines. They are the PROP (Proprietary) routines COMPILE and REPORT. Like CORRECT.BCD these routines do not perform LP computations. The former was useful primarily to facilitate timely and convenient correction of input data for LP computations. On the other hand, COMPILE and REPORT are designed to facilitate interpretation and convenient summarization of the output from LP computations.

b. Data

The data for this example is identical to the I-tape created in Example 2. We shall in fact perform a duplicate of Example 2 except that COMPILE will be used at the beginning of the run and REPORT after each use of NORMAL. Thus, there will be four LP problems solved and the solution output will be identical to that output in Example 2. However, we shall assume that the actual magnitude of the right-hand-sides is just 1000 times as great as the RHS elements on the I-tape. We could then regard the I-tape model as a typical firm possessing 1/1000 of a given region's resources. The Agendum routine REPORT will be made to scale our solution variable by 1000, so as to convert the data to regional totals.

The new I-tape created by the use of CORRECT.BCD in the last example provides the I-tape for this problem. This input corresponds of course to the LP matrix illustrated in Table 23, and the I-tape listing in Table 25.

c. Agenda

The complete Agenda for this example is listed in Table 31. The Agenda for this example is just like that for example 2, except that COMPILE,CR,B8,1 appears after the HEADING card and is followed by program cards up to and including END***, and the Agendum, REPORT,B8,1, follows each of the four NORMAL Agendum control cards. Also, as the new I-tape from example 2 is used, the CORRECT.BCD Agendum is deleted.

The COMPILE,CR,B8,1 instructs LP/90 to (1) read the cards in the card reader (CR) up to and including the END*** card. (These cards constitute a report writing program.) (2) Compile the program in machine language. (3) Store it on tape unti B8 as report writing program number 1. The REPORT,B8,1 card says to (1) read in report writing program number 1 on tape unit B8 (as placed there by the preceding COMPILE); (2) process and format the last solution on the mediary output tape (M-tape in binary) and place the resulting report on the O-tape for printing. The B8 tape can be saved for later runs.

In order to fully understand the use of these Agendum routines, we shall insert a special section dealing with the rules of their application next. Then we may return to the usual section on output.

COMPILE and REPORT are PROP LP/90 routines. The remainder belong to the SHARE LP/90 system.

d. How to Use COMPILE

The rules for coding[26/] a report writing program are really quite simple. They can in fact be summarized on a single page as shown in Table 28. A special COMPILE card format sheet is used to code the program. Table 30 shows a photographic reproduction of the first page of the report coding form used in this example, while Table 31 shows the complete listing. Rather than discuss the COMPILE card rules in general, we shall instead explain the particular COMPILE cards used in this example. In doing so we shall illustrate the steps which one would usually pursue in preparing the report writing program.

Step 1: Report Layout. As in report writing in general, the LP/90 COMPILE-REPORT user must decide (1) just what he wants to report, and (2) the form or appearance of his report. In a great many cases, the LP analyst does not really want the detailed optimal solution as produced by OUTPUT. Instead he wants the values of various variables that are based on or derived from the optimal solution. Let us suppose that as agricultural economists we do not really need to know how much cotton, oats, etc., would be produced by each and every process in the optimal solution. Rather, we are interested in (1) the total amount of each crop produced, (2) the amounts produced on each soil type and (3) the amounts produced at each level of fertilization. We could think of other equally interesting kinds of information, but these will serve to illustrate the procedure.

Having decided on the information we wish to derive from the optimal solution, the report layout may be prepared. This task is greatly facilitated by using a special printer spacing chart which enables the report to be specified exactly as it will appear when printed from the O-tape. As we shall see, the spacing chart also facilitates the coding of our report writing program.

The spacing chart used in this example is duplicated in Table 29. The report contains three tables as indicated above, and in our case they will all fit nicely on a single output page. Furthermore, the tables have

---

26/ By coding is meant the writing down of a series of instructions to be converted to machine language by a computer compiler or assembler.

been spaced so that when the output is received the sheets can be trimmed for binding in a standard government size ring binder.

Step 2: Coding the Report Writing Subprogram. Rather than itemize the application of the coding rules to each card in the Report Writing Program, we shall instead list for each of the 23 cards the effective instruction that corresponds to the punched statement. By comparing this list against Table 28, which presents the general COMPILE Card Format rules, the reader will see how the report layout is converted into a series of program statements.

Compile Card	Effective Instruction
Card 1	Scale each primal solution variable (column names 1COT11, ..., 1R1C22) and each slack variable by 1000.
Card 2	Begin a new page of the report with the solution heading (read in by HEADING or TITLE Agenda) and number page on the first line.
Card 3	Place the name of the report after double spacing.
Card 4	Place the column headings of the first report in the designated columns.
Card 5	Continue card 4.
Card 6	Double space and print name of first row (COTTON LINT) of table.
Cards 7-14	Compute the cumulative total of production for cotton, by multiplying each yield (constant) by the solution variable for the process (column name) with that yield; and, having done that, place the answer to the left of card column 40; place a comma in between every three digits to the left of column 40; and suppress the decimal point. Do not space (i.e., place on same print line as COTTON LINT).

Compile Card	Effective Instruction
Card 15	Compute total acreage in cotton as the upper bound for cotton (UBCOT) minus the slack variable for this row (÷UBCOT), and place answer to the left of column 60. Insert commas and suppress the decimal point.
Card 16	Obtain the average yield for all cotton processes by dividing total production by total acreage and place the answer with two places to right of decimal point in card column 80.
Card 17	Place name of second row of table after double spacing.
Cards 18-23	Compute the cumulative production for cotton seed by multiplying each cotton seed yield by the corresponding solution variable, etc.

With these examples, the reader may return to the listing of the complete Agenda and see how the program is continued.

Step 3: Checking the Coding for Errors. It is important for the coder to check the report writing program for errors. (1) He should first check his coding sheet carefully, making sure that exactly the right symbol is indicated in the proper column and that all LP Matrix data referred to are identified in the coding by exactly the same row and column names as used on the I-tape. (b) The punched cards should be carefully checked against the coding forms. A common error is the punching of "ohs" for "zeros" or vice versa. In column 1 of the COMPILE Card Format a zero is used to indicate double spacing. An "oh" in this column will cause a COMPILE error message.

While this exegesis is not exhaustive, the reader can, by studying the material presented here, gain a basic understanding of how report writing programs are prepared.

TABLE 28

COMPILE CARD FORMAT

Col. 1   Printer Control

1 ≡ New Page   b = Single Space (b = blank)
0 ≡ Double Space   + = No Space

Col. 2   Card Type

H ≡ Hollerith text, col. 3-72 are in print line.
C ≡ Continuation card, limited to one continuation card if Hollerith text in which case col. 13-60 continues the text.
T ≡ Title; begin new page, problem heading and page number on first line.

Col. 3   Blank

Col. 4   Z ≡ Suppress line if all computed values are zero. Should be used only on the first card calling for computation in a line.

Col. 1-6   END*** terminates loading of Compile Cards.

Col. 5-9   Format

Col. 5-9 blank means do not print.
Col. 5-7 decimal point position in print line.
Col. 8 decimal point or comma. If comma, commas will be inserted to mark the thousands place in values.

Col. 9 positions to the right of decimal point. (If 0, the decimal is suppressed.)

Col. 10-15   New Variable Name

Any alphanumeric symbol allowed, up to 6 characters. The symbol is used as in Fortran, e.g., XI = C 5. followed by XI = C 17. is legal. XI will have the value 5 between the first and second defining statements.

Col. 16, 25, etc.   Operator

+ ≡ add   / ≡ divide
- ≡ subtract   = = equals (may be used in col. 16 only)
* ≡ multiply   S = Scale all variables of kind indicated in type field. The scale factor is written in Constant format in the name field following the op. More than one scale can be given on one card. It must be the very first card of deck if used.

TABLE 28: COMPILE CARD FORMAT

Col. 17, 26, etc. Operand Type

X ≡ Beta (Primal Variable, use col. name)
P ≡ Pi (Dual Variable, use row name)
B ≡ Right-Hand Side
D ≡ D/J (will be computed during REPORT)
A ≡ a_j^i from matrix
C ≡ Constant
N ≡ New variable previously defined in column 10-15 (at its current value)
M ≡ Objective function values. (From Beta region identified by row name.)

Col. 18, 27, etc. Slack type for indicating slack variables. For betas this column must be +,-, or blank. For all other types of operands this must be blank. Exception: Constant values must begin in this column.

Col. 19-24, 28-33, etc. Name of Operand; if this is a new variable it must have been defined sometime previously in col. 10-15.

Note 1: Arithmetic operations are of the form: $(((=X_1)opX_2)opX_3)$ where quantities on left are operated on by the next succeeding quantity on the right and are thereby redefined for the next succeeding operation.

Note 2: If type field is A, then the corresponding name field is the column name and the next contiguous (cannot be on next card) name field is the row name defining the element.

Table 29: PRINT SPACING CHART

Line	1	2	3	5	7	Commonly Used Depths
2			OUTPUT SUMMARY			
4	CROP	/	PRODUCTION	ACREAGE	AVE. YIELD	
6	COTTON LINT					
8	COTTON SEED					
10	OATS					
12	SOYBEANS					
14	ALFALFA					
16	RICE					
18		FIELD CROP ACREAGE BY SOIL TYPE				
20	CROP	/	SOIL 1	SOIL 2	TOTAL	3⅓"
21						3½"
22	COTTON					3⅔"
24	OATS					
26	SOYBEANS					
28	ALFALFA					
30	RICE					
32		FIELD CROP ACREAGE BY FERTILIZER LEVEL				
33						5½"
34	CROP	/	LEVEL 1	LEVEL 2	TOTAL	
36	COTTON					6
38	OATS					
40	RICE					
42						7"
44						7⅓"
51						8½"

DUAL SPEED CARRIAGE
Two punches in same channel should be more than eight spaces apart.
Only one punch per line.

FORM WIDTHS	MEASUREMENTS
Minimum 3½ inches	Horizontal spacing 1/10 inch
Maximum 18¾ inches	Vertical spacing 1/6 inch

NOTE: This form is subject to inaccuracies from variations in humidity. Dimensions of bill should be calculated from measurements shown and not scaled on the form.

Table 30: REPORT CODING FORM

LINEAR PROGRAMMING OUTPUT REPORT
INPUT FORM (COMPILE CARDS)
LP-90 - INPUT FORM C

Job No. 9020
Date 15 June 1963
Pg. 1 of 4
Prog: DAY

	CNT	HCT	FORMAT	NEW VARIABLE	1 (OP TYPE b NAME)	2 (OP TYPE b NAME)	3 (OP TYPE b NAME)	4 (OP TYPE b NAME)	5 (OP TYPE b NAME)	6 (OP TYPE b NAME)	70–72	73–80
1					SX01000.0	SB01000.0						EXAMPLE 3
2		T										
3	O	H					ŌUTPUT	SUMMARY				
4	O	H		CRŌ	P /	PRŌDU	CTIŌN		ACREAGE		AVE	
5		C		• Y	IELD							
6	O	H		CŌTTŌN	LINT							
7	+			SUM	=C0003.10	*X 1CŌT11						
8	+			PCŌTLT	=C0007.23	*X 1CŌT12	+N SUM					
9	+			SUM	=C0002.96	*X 1CŌT21	+N PCŌTLT					
10	+			PCŌTLT	=C0004.43	*X 1CŌT22	+N SUM					
11	+			SUM	=C0002.79	*X 2CŌT11	+N PCŌTLT					
12	+			PCŌTLT	=C0006.51	*X 2CŌT12	+N SUM					
13	+			SUM	=C0002.66	*X 2CŌT21	+N PCŌTLT					
14	+		40, 0	PCŌTLT	=C0003.99	*X 2CŌT22	+N SUM					
15	+		60, 0	ACOTON	=B UBCŌT	-X+ UBCŌT						
16	+		80, 2		=N PCŌTLT	/N ACŌTŌN						
17	O	H		CŌTTŌN	SEED							
18	+			SUM	=C0005.29	*X 1CŌT11						
19	+			PCŌTSD	=C0012.32	*X 1CŌT12	+N SUM					
20	+			SUM	=C0005.04	*X 1CŌT21	+N PCŌTSD					
21	+			PCŌTSD	=C0007.56	*X 1CŌT22	+N SUM					
22	+			SUM	=C0004.67	*X 2CŌT11	+N PCŌTSD					
23	+			PCŌTSD	=C0011.09	*X 2CŌT12	+N SUM					

Table 31: Agenda for Report Writing Example

(Listing Sheet 1)

```
HEADING     USDA LP/90    EXAMPLE 3    USE OF REPORT
COMPILE,CR,B8,1
               SX01000.0SB01000.0
 T                                                                      EXAMPLE3
OH                                  OUTPUT  SUMMARY                     EXAMPLE3
OH          CROP              PRODUCTION              ACREAGE        AVEEXAMPLE3
 C           .  YIELD                                                   EXAMPLE3
OH       COTTON LINT                                                    EXAMPLE3
+        SUM   =C0003.10*X 1COT11
+        PCOTLT=C0007.23*X 1COT12+N SUM                                 EXAMPLE3
+        SUM   =C0002.96*X 1COT21+N PCOTLT                              EXAMPLE3
+        PCOTLT=C0004.43*X 1COT22+N SUM                                 EXAMPLE3
+        SUM   =C0002.79*X 2COT11+N PCOTLT                              EXAMPLE3
+        PCOTLT=C0006.51*X 2COT12+N SUM                                 EXAMPLE3
+        SUM   =C0002.66*X 2COT21+N PCOTLT                              EXAMPLE3
+    40,0PCOTLT=C0003.99*X 2COT22+N SUM                                 EXAMPLE3
+    60,0ACOTON=B  UBCOT-X+ UBCOT                                       EXAMPLE3
+    78,2      =N PCOTLT/N ACOTON                                       EXAMPLE3
OH       COTTON SEED                                                    EXAMPLE3
+        SUM   =C0005.29*X 1COT11
+        PCOTSD=C0012.32*X 1COT12+N SUM                                 EXAMPLE3
+        SUM   =C0005.04*X 1COT21+N PCOTSD                              EXAMPLE3
+        PCOTSD=C0007.56*X 1COT22+N SUM                                 EXAMPLE3
+        SUM   =C0004.67*X 2COT11+N PCOTSD                              EXAMPLE3
+        PCOTSD=C0011.09*X 2COT12+N SUM                                 EXAMPLE3
+        SUM   =C0004.54*X 2COT21+N PCOTSD                              EXAMPLE3
+    40,0PCOTSD=C0006.80*X 2COT22+N SUM                                 EXAMPLE3
+    60,0      =N ACOTON                                                EXAMPLE3
+    78,2      =N PCOTSD/N ACOTON                                       EXAMPLE3
OH       OATS                                                           EXAMPLE3
+        SUM   =C0014.40*X 1OAT11                                       EXAMPLE3
+        POATS =C0065.10*X 1OAT12+N SUM                                 EXAMPLE3
+        SUM   =C0012.90*X 1OAT21+N POATS                               EXAMPLE3
+    40,0POATS =C0058.40*X 1OAT22+N SUM                                 EXAMPLE3
+    60,0AOATS =B  UBOAT-X+ UBOAT                                       EXAMPLE3
+    78,2      =N POATS /N AOATS                                        5X147353
OH       SOYBEANS                                                       EXAMPLE3
+        SUM   =C0029.00*X 1SBG11                                       EXAMPLE3
+    40,0PSBG  =C0017.00*X 1SBG21+N SUM                                 EXAMPLE3
+    60,0ASBG  =B  UBSBG-X+ UBSBG                                       EXAMPLE3
+    78,2      =N PSBG  /N ASBG                                         EXAMPLE3
OH       ALFALFA                                                        EXAMPLE3
+    40,0      =C0002.07*X 1ALF11                                       EXAMPLE3
+    60,0      =X 1ALF11
+    78,2      =C0002.07                                                EXAMPLE3
OH       RICE                                                           EXAMPLE3
+        SUM   =C0017.60*X 1RIC21                                       EXAMPLE3
+    40,0PRICE =C0035.60*X 1RIC22+N SUM                                 EXAMPLE3
+    60,0ARICE =B  UBRIC-X+ UBRIC                                       EXAMPLE3
+    78,2      =N PRICE /N ARICE                                        EXAMPLE3
OH                    FIELD  CROP  ACREAGE  BY  SOIL  TYPE              EXAMPLE3
OH          CROP                SOIL 1               SOIL 2            TEXAMPLE3
 C          OTAL                                                        EXAMPLE3
OH       COTTON                                                         EXAMPLE3
+    40,0SUM   =X 1COT11+X 1COT12+X 2COT11+X 2COT12                     EXAMPLE3
+    60,0ADD   =X 1COT21+X 1COT22+X 2COT21+X 2COT22                     EXAMPLE3
+    80,0      =N SUM   +N ADD                                          EXAMPLF3
OH       OATS                                                           EXAMPLE3
+    40,0SUM   =X 1OAT11+X 1OAT12                                       EXAMPLE3
+    60,0ADD   =X 1OAT21+X 1OAT22                                       EXAMPLE3
+    80,0      =N SUM   +N ADD                                          EXAMPLE3
```

(Listing Sheet 2)

```
0H        SOYBEANS                                                          EXAMPLE3
+     40,0        =X 1SBG11                                                 EXAMPLE3
+     60,0        =X 1SBG21                                                 EXAMPLE3
+     80,0        =X 1SBG11+X 1SBG21                                        EXAMPLE3
0H        'ALFALFA                                                          EXAMPLE3
+     40,0        =X 1ALF11                                                 EXAMPLE3
+     80,0        =B  UBALF-X+ UBALF                                        EXAMPLE3
0H        RICE                                                              EXAMPLE3
+     60,0        =X 1RIC21+X 1RIC22                                        EXAMPLE3
+     80,0        =B  UBRIC-X+ UBRIC                                        EXAMPLE3
0H                   FIELD  CROP  ACREAGE  BY  FERTILIZER  LEVEL            EXAMPLE3
0H           CROP    /          LEVEL 1                 LEVEL 2            TEXAMPLE3
 C           OTAL                                                           EXAMPLE3
0H        COTTON                                                            EXAMPLE3
+     40,0SUM     =X 1COT11+X 1COT21+X 2COT11+X 2COT21                      EXAMPLE3
+     60,0ADD     =X 1COT12+X 1COT22+X 2COT12+X 2COT22                      EXAMPLE3
+     80,0        =N SUM   +N ADD                                           EXAMPLE3
0H        OATS                                                              EXAMPLE3
+     40,0SUM     =X 1OAT11+X 1OAT21                                        EXAMPLE3
+     60,0ADD     =X 1OAT12+X 1OAT22                                        EXAMPLE3
+     80,0        =N SUM   +N ADD                                           EXAMPLE3
0H        RICE                                                              EXAMPLE3
+     40,0        =X 1RIC21                                                 EXAMPLE3
+     60,0        =X 1RIC22                                                 EXAMPLE3
+     80,0        =B  UBRIC-X+ UBRIC                                        EXAMPLE3
END***
INPUT,2,,REWIND
CONTROLS                                  30
TITLE        USDA LP/90    EXAMPLE 3     PROBLEM  1
NORMAL,* Z1    ,2
REPORT,B8,1
TITLE        USDA LP/90    EXAMPLE 3     PROBLEM  2
NORMAL,* Z2    ,2
REPORT,B8,1
TITLE        USDA LP/90    EXAMPLE 3     PROBLEM  3
NORMAL,* Z1    ,3
REPORT,B8,1
TITLE        USDA LP/90    EXAMPLE 3     PROBLEM  4
NORMAL,* Z2    ,3
REPORT,B8,1
OUTPUT
```

e. Output

The reports produced by each use of the REPORT Agendum are found first on the O-tape. After the reports come the usual output created by the OUTPUT Agendum. As the latter is (for this example) exactly like that produced in example 2, we shall not present it here. The reports, however, are reproduced in Table 32. They are reproduced exactly as printed from the O-tape except that they have been trimmed to a more convenient size.

Table 32: OUTPUT FROM REPORT AGENDUM

(Listing Sheet 1)

USDA LP/90 EXAMPLE 3 PROBLEM 1

OUTPUT SUMMARY

CROP /	PRODUCTION	ACREAGE	AVE YIELD
COTTON LINT	3,338,983	700,000	4.77
COTTON SEED	5,688,035	700,000	8.13
OATS	4,864,125	340,000	14.31
SOYBEANS	18,850,000	650,000	29.00
ALFALFA	476,100	230,000	2.07
RICE	2,848,000	80,000	35.60

FIELD CROP ACREAGE BY SOIL TYPE

CROP /	SOIL 1	SOIL 2	TOTAL
COTTON	301,250	398,750	700,000
OATS	318,750	21,250	340,000
SOYBEANS	650,000		650,000
ALFALFA	230,000		230,000
RICE		80,000	80,000

FIELD CROP ACREAGE BY FERTILIZER LEVEL

CROP /	LEVEL 1	LEVEL 2	TOTAL
COTTON	398,750	301,250	700,000
OATS	340,000		340,000
RICE		80,000	80,000

Table 32: OUTPUT FROM REPORT AGENDUM--Continued

(Listing Sheet 2)

USDA LP/90 EXAMPLE 3 PROBLEM 2

OUTPUT SUMMARY

CROP /	PRODUCTION	ACREAGE	AVE YIELD
COTTON LINT	3,680,583	700,000	5.26
COTTON SEED	6,270,435	700,000	8.96
OATS	4,744,125	340,000	13.95
SOYBEANS	18,850,000	650,000	29.00
ALFALFA	476,100	230,000	2.07
RICE	1,408,000	80,000	17.60

FIELD CROP ACREAGE BY SOIL TYPE

CROP /	SOIL 1	SOIL 2	TOTAL
COTTON	381,250	318,750	700,000
OATS	238,750	101,250	340,000
SOYBEANS	650,000		650,000
ALFALFA	230,000		230,000
RICE		80,000	80,000

FIELD CROP ACREAGE BY FERTILIZER LEVEL

CROP /	LEVEL 1	LEVEL 2	TOTAL
COTTON	318,750	381,250	700,000
OATS	340,000		340,000
RICE	80,000		80,000

Table 32: OUTPUT FROM REPORT AGENDUM--Continued

(Listing Sheet 3)

USDA LP/90 EXAMPLE 3 PROBLEM 3

OUTPUT SUMMARY

CROP /	PRODUCTION	ACREAGE	AVE YIELD
COTTON LINT	3,217,908	752,000	4.28
COTTON SEED	5,483,203	752,000	7.29
OATS	4,262,400	296,000	14.40
SOYBEANS	18,850,000	650,000	29.00
ALFALFA	314,640	152,000	2.07
RICE	5,340,000	150,000	35.60

FIELD CROP ACREAGE BY SOIL TYPE

CROP /	SOIL 1	SOIL 2	TOTAL
COTTON	402,000	350,000	752,000
OATS	296,000		296,000
SOYBEANS	650,000		650,000
ALFALFA	152,000		152,000
RICE		150,000	150,000

FIELD CROP ACREAGE BY FERTILIZER LEVEL

CROP /	LEVEL 1	LEVEL 2	TOTAL
COTTON	520,750	231,250	752,000
OATS	296,000		296,000
RICE		150,000	150,000

Table 32: OUTPUT FROM REPORT AGENDUM--Continued

(Listing Sheet 4)

USDA LP/90 EXAMPLE 3 PROBLEM 4

OUTPUT SUMMARY

CROP /	PRODUCTION	ACREAGE	AVE YIELD
COTTON LINT	3,837,408	752,000	5.10
COTTON SEED	6,537,703	752,000	8.69
OATS	4,262,400	296,000	14.40
SOYBEANS	18,850,000	650,000	29.00
ALFALFA	314,640	152,000	2.07
RICE	2,640,000	150,000	17.60

FIELD CROP ACREAGE BY SOIL TYPE

CROP /	SOIL 1	SOIL 2	TOTAL
COTTON	402,000	350,000	752,000
OATS	296,000		296,000
SOYBEANS	650,000		650,000
ALFALFA	152,000		152,000
RICE		150,000	150,000

FIELD CROP ACREAGE BY FERTILIZER LEVEL

CROP /	LEVEL 1	LEVEL 2	TOTAL
COTTON	370,750	381,250	752,000
OATS	296,000		296,000
RICE	150,000		150,000

f. Remarks

The time used by COMPILE was about 25 seconds at a cost of about $3.50. Each use of report took about 9 seconds at a marginal cost of about $1.25 each. Even on the tiny, heuristic problem used here, this is surely a smaller cost than having a clerk process the output solution tape into comparable table form and having a typist complete the job. Of course, it took some time to code the report writing program, to key punch and verify it. Yet, in the very frequent type of LP run, with many, perhaps dozens, of individual problems run, each of which may have hundreds of columns and rows, this cost and effort will be amply repaid.

The fact that LP output can be processed into tables of direct interest for economic analysis in a way that short-circuits time-consuming and error producing transformations, intermediate computation, and table formating, greatly enhances the usefullness of LP/90. It makes model results immediately available for analysis or decision making. When the number and size of problems are very large, COMPILE and REPORT may reduce a clerically burdensome and expensive analysis into a modest task of photographing and suitably duplicating the final tables.

COMPILE and REPORT are particularly convenient for use in Recursive Linear Programming runs, when a sequence of recursively dependent LP problems are solved. This kind of application will be reviewed in Section 6 below.

4. PARAMETRIC PROGRAMMING

a. Purpose

Parametric objective function (price mapping) and parametric right-hand-side (resource mapping) computations are among the most frequently used of the LP-related computations. In this example several different parametric programming computations are performed. The use of price mapping to obtain (comparative static) output supply and variable input demand functions is illustrated. Also, resource mapping for proportional changes among several right-hand-side elements is exemplified. We also take this opportunity to introduce two additional PROP input modifying Agenda, one for adding new right-hand-side elements, and one for reading an initial basis from which to begin iterations.

b. Data

The input data with which we began in example 1 was augmented in example 2 by an additional objective function and two additional RHS vectors. For this example we augment the original data still further by the addition of four objective function "change vectors" and two right-hand-side "change vectors."

The augmented matrix is shown in Table 33. In this example the four additional matrix rows are added to the I-tape created in example 2 by again using CORRECT.BCD. After its use the I-tape will appear as shown in Table 34. The two additional right-hand-sides to be used for Parametric LP will be added directly to the A-tape (binary input tape) after the use of INPUT by means of the Agendum ADDRHS.

The new rows added to the data are (1) a cotton lint yield coefficient row 1COTLT, giving the yield of cotton per unit process level (per acre) for each cotton growing process; (2) an unskilled labor coefficient row 1UNSLB giving the hours of unskilled labor used per unit process level (per acre) for each process; (3) the negative of the cotton yield row, 2COTLT and (4) the negative of the unskilled labor row, 2UNSLB. The first and third addition will enable us to derive a supply function for cotton; the second and fourth item will enable us to derive a demand function for unskilled labor.

The new columns will not appear on the I-tape but will be placed on the A-tape as if they had been identified as NEXT B,4 and NEXT B,5 on the I-tape. These columns are (1) the values of the RHS elements for the upper and lower bounds (rows beginning with UB or LB) used in the first RHS vector and (2) the negatives of those values. All of the other elements are zeros.

The first of the two in effect will enable us to raise the upper and lower bounds proportionally while the latter will enable us to diminish them proportionally leaving the resource stocks and acreage allotments unchanged. While this type of change might be economically artificial it is presented not for its inherent economic interest but to serve as an illustration of PLP on several RHS elements simultaneiously.

TABLE 33

LP MATRIX FOR PARAMETRIC LP EXAMPLE

Row Symbol	Matrix Column Symbol											
	1COT11	1COT12	1COT21	1COT22	2COT11	2COT12	2COT21	2COT22	1OAT11	1OAT12	1OAT21	1OAT22
Z1	- 35.16	-114.79	- 36.26	- 57.05	- 42.13	-144.33	- 42.52	- 71.60	- 3.80	- 30.63	- 2.09	- 21.45
Z2	- 41.86	-119.59	- 42.04	- 64.38	- 46.93	-138.98	- 47.05	- 74.64	- 6.81	- 36.12	- 5.11	- 30.17
1COTLT	3.10	7.23	2.96	4.43	2.79	6.51	2.66	3.99				
1UNSLB	87.00	165.00	86.00	113.00	16.60	16.60	12.80	12.80	0.50	2.40	0.50	2.20
2COTLT	- 3.10	- 7.23	- 2.96	- 4.43	- 2.79	- 6.51	- 2.66	- 3.99				
2UNSLB	- 87.00	-165.00	- 86.00	-113.00	- 16.60	- 16.60	- 12.80	- 12.80	- 0.50	- 2.40	- 0.50	- 2.20
SOIL1	1.00	1.00			1.00	1.00			1.00	1.00		
SOIL2			1.00	1.00			1.00	1.00			1.00	1.00
LABP4	5.80	13.50	5.50	8.30								
FERT		4.00		4.00		4.00		4.00		4.00		4.00
LOTCT	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00				
LOTRC												
UBCOT	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00				
UBOAT									1.00	1.00	1.00	1.00
UBSBG												
UBALF												
UBRIC												
LBCOT	- 1.00	- 1.00	- 1.00	- 1.00	- 1.00	- 1.00	- 1.00	- 1.00				
LBOAT									- 1.00	- 1.00	- 1.00	- 1.00
LBSBG												
LBALF												
LBRIC												
S2COT					2.79	6.51	2.73	4.89				

TABLE 33--Continued

Row Symbol	Matrix Column Symbol					Con-straint Type	Right-Hand-Sides				
	1SBG11	1SBG21	1ALF11	1RIC21	1RIC22		C1	C2	C3	1CHFLX	2CHFLX
Z1	- 51.07	- 24.72	- 23.66	- 60.91	-144.81						
Z2	- 65.87	- 33.27	- 34.52	- 55.32	-117.93						
1COTLT											
1UNSLB	1.10	1.10	2.80								
2COTLT											
2UNSLB	- 1.10	- 1.10	- 2.80								
SOIL1	1.00		1.00			≤	500	1500	1500		
SOIL2		1.00		1.00	1.00	≤	1500	500	500		
LABP4	1.10	1.10	0.70			≤	9900	9900	9900		
FERT					4.00	≤	1525	1525	1525		
LOTCT						≤	700	700	5000		
LOTRC				1.00	1.00	≤	80	80	5000		
UBCOT						≤	880	880	880	880	-880
UBOAT						≤	520	520	520	520	-520
UBSBG	1.00	1.00				≤	650	650	650	650	-650
UBALF			1.00			≤	230	230	230	230	-230
UBRIC				1.00	1.00	≤	150	150	150	150	-150
LBCOT						≤	- 640	- 640	- 640	- 640	640
LBOAT						≤	- 296	- 296	- 296	- 296	296
LBSBG	- 1.00	- 1.00				≤	- 370	- 370	- 370	- 370	370
LBALF			- 1.00			≤	- 152	- 152	- 152	- 152	152
LBRIC				- 1.00	- 1.00	≤	- 79	- 79	- 79	- 79	79
S2COT						≤	175	175	175		

Table 34: I-Tape Listing for Parametric LP Example

(Listing Sheet 1)

```
* USDA LP/90    INPUT   LP MATRIX FOR   EXAMPLE  1
ROW ID
              Z1
              Z2
             1COTLT
             1UNSLB
             2COTLT
             2UNSLB
           + SOIL1
           + SOIL2
           + LABP4
           + FERT
           + LOTCT
           + LOTRC
           + UBCOT
           + UBOAT
           + UBSBG
           + UBALF
           + UBRIC
           + LBCOT
           + LBOAT
           + LBSBG
           + LBALF
           + LBRIC
           + S2COT
MATRIX
        1COT11 Z1     -   35.16
        1COT11 Z2     -   41.86
        1COT111COTLT       3.10
        1COT111UNSLB      87.00
        1COT112COTLT-      3.10
        1COT112UNSLB-     87.00
        1COT11 SOIL1       1.00
        1COT11 LABP4       5.80
        1COT11 LOTCT       1.00
        1COT11 UBCOT       1.00
        1COT11 LBCOT-      1.00
        1COT12 Z1     -  114.79
        1COT12 Z2     -  119.59
        1COT121COTLT       7.23
        1COT121UNSLB     165.00
        1COT122COTLT-      7.23
        1COT122UNSLB-    165.00
        1COT12 SOIL1       1.00
        1COT12 LABP4      13.50
        1COT12 FERT        4.00
        1COT12 LOTCT       1.00
        1COT12 UBCOT       1.00
        1COT12 LBCOT-      1.00
        1COT21 Z1     -   36.26
        1COT21 Z2     -   42.04
        1COT211COTLT       2.96
        1COT211UNSLB      86.00
        1COT212COTLT-      2.96
        1COT212UNSLB-     86.00
        1COT21 SOIL2       1.00
        1COT21 LABP4       5.50
        1COT21 LOTCT       1.00
```


(Listing Sheet 2)

```
1COT21 UBCOT       1.00
1COT21 LBCOT-      1.00
1COT22 Z1     -   57.05
1COT22 Z2     -   64.38
1COT221COTLT       4.43
1COT221UNSLB     113.00
1COT222COTLT-      4.43
1COT222UNSLB-    113.00
1COT22 SOIL2       1.00
1COT22 LABP4       8.30
1COT22 FERT        4.00
1COT22 LOTCT       1.00
1COT22 UBCOT       1.00
1COT22 LBCOT-      1.00
2COT11 Z1     -   42.13
2COT11 Z2     -   46.93
2COT111COTLT       2.79
2COT111UNSLB      16.60
2COT112COTLT-      2.79
2COT112UNSLB-     16.60
2COT11 SOIL1       1.00
2COT11 LOTCT       1.00
2COT11 UBCOT       1.00
2COT11 LBCOT-      1.00
2COT11 S2COT       2.79
2COT12 Z1     -  144.33
2COT12 Z2     -  138.98
2COT121COTLT       6.51
2COT121UNSLB      16.60
2COT122COTLT-      6.51
2COT122UNSLB-     16.60
2COT12 SOIL1       1.00
2COT12 FERT        4.00
2COT12 LOTCT       1.00
2COT12 UBCOT       1.00
2COT12 LBCOT-      1.00
2COT12 S2COT       6.51
2COT21 Z1     -   42.52
2COT21 Z2     -   47.05
2COT211COTLT       2.66
2COT211UNSLB      12.80
2COT212COTLT-      2.66
2COT212UNSLB-     12.80
2COT21 SOIL2       1.00
2COT21 LOTCT       1.00
2COT21 UBCOT       1.00
2COT21 LBCOT-      1.00
2COT21 S2COT       2.73
2COT22 Z1     -   71.60
2COT22 Z2     -   74.64
2COT221COTLT       3.99
2COT221UNSLB      12.80
2COT222COTLT-      3.99
2COT222UNSLB-     12.80
2COT22 SOIL2       1.00
2COT22 FERT        4.00
2COT22 LOTCT       1.00
2COT22 UBCOT       1.00
```

(Listing Sheet 3)

```
2COT22 LBCOT-    1.00
2COT22 S2COT     4.89
1OAT11 Z1    -   3.80
1OAT11 Z2    -   6.81
1OAT111UNSLB      .50
1OAT112UNSLB-     .50
1OAT11 SOIL1     1.00
1OAT11 UBOAT     1.00
1OAT11 LBOAT-    1.00
1OAT12 Z1    -  30.63
1OAT12 Z2    -  36.12
1OAT121UNSLB     2.40
1OAT122UNSLB-    2.40
1OAT12 SOIL1     1.00
1OAT12 FERT      4.00
1OAT12 UBOAT     1.00
1OAT12 LBOAT-    1.00
1OAT21 Z1    -   2.09
1OAT21 Z2    -   5.11
1OAT211UNSLB      .50
1OAT212UNSLB-     .50
1OAT21 SOIL2     1.00
1OAT21 UBOAT     1.00
1OAT21 LBOAT-    1.00
1OAT22 Z1    -  21.45
1OAT22 Z2    -  30.17
1OAT221UNSLB     2.20
1OAT222UNSLB-    2.20
1OAT22 SOIL2     1.00
1OAT22 FERT      4.00
1OAT22 UBOAT     1.00
1OAT22 LBOAT-    1.00
1SBG11 Z1    -  51.07
1SBG11 Z2    -  65.87
1SBG111UNSLB     1.10
1SBG112UNSLB-    1.10
1SBG11 SOIL1     1.00
1SBG11 LABP4     1.10
1SBG11 UBSBG     1.00
1SBG11 LBSBG-    1.00
1SBG21 Z1    -  24.72
1SBG21 Z2    -  33.27
1SBG211UNSLB     1.10
1SBG212UNSLB-    1.10
1SBG21 SOIL2     1.00
1SBG21 LABP4     1.10
1SBG21 UBSBG     1.00
1SBG21 LBSBG-    1.00
1ALF11 Z1    -  23.66
1ALF11 Z2    -  34.52
1ALF111UNSLB     2.80
1ALF112UNSLB-    2.80
1ALF11 SOIL1     1.00
1ALF11 LABP4     0.70
1ALF11 UBALF     1.00
1ALF11 LBALF-    1.00
1RIC21 Z1    -  60.91
1RIC21 Z2    -  55.32
```


(Listing Sheet 4)

```
            1RIC21 SOIL2     1.00
            1RIC21 LOTRC     1.00
            1RIC21 UBRIC     1.00
            1RIC21 LBRIC-    1.00
            1RIC22 Z1    - 144.81
            1RIC22 Z2    - 117.93
            1RIC22 SOIL2     1.00
            1RIC22 FERT      4.00
            1RIC22 LOTRC     1.00
            1RIC22 UBRIC     1.00
            1RIC22 LBRIC-    1.00
FIRST B
            C1     SOIL1  500.
            C1     SOIL2 1500.
            C1     LABP4 9900.
            C1     FERT  1525.
            C1     LOTCT  700.
            C1     LOTRC   80.
            C1     UBCOT  880.
            C1     UBOAT  520.
            C1     UBSBG  650.
            C1     UBALF  230.
            C1     UBRIC  150.
            C1     LBCOT- 640.
            C1     LBOAT- 296.
            C1     LBSBG- 370.
            C1     LBALF- 152.
            C1     LBRIC-  79.
            C1     S2COT  175.
NEXT B,2
            C2     SOIL1 1500.
            C2     SOIL2  500.
            C2     LABP4 9900.
            C2     FERT  1525.
            C2     LOTCT  700.
            C2     LOTRC   80.
            C2     UBCOT  880.
            C2     UBOAT  520.
            C2     UBSBG  650.
            C2     UBALF  230.
            C2     UBRIC  150.
            C2     LBCOT- 640.
            C2     LBOAT- 296.
            C2     LBSBG- 370.
            C2     LBALF- 152.
            C2     LBRIC-  79.
            C2     S2COT  175.
NEXT B,3
            C3     SOIL1 1500.
            C3     SOIL2  500.
            C3     LABP4 9900.
            C3     FERT  1525.
            C3     LOTCT 5000.
            C3     LOTRC 5000.
            C3     UBCOT  880.
            C3     UBOAT  520.
            C3     UBSBG  650.
            C3     UBALF  230.
```

PRINTED IN U.S.A
FORM 1413
MOORE BUSINESS FORMS, INC.

(Listing Sheet 5)

```
            C3     UBRIC  150.
            C3     LBCOT- 640.
            C3     LBOAT- 296.
            C3     LBSBG- 370.
            C3     LBALF- 152.
            C3     LBRIC-  79.
            C3     S2COT  175.
EOF
```

c. Agenda

The control cards comprising the Agenda for the run are listed in Table 35. The first Agendum after HEADING is CORRECT.BCD with whose operation we are already familiar (example 2). The cards following MATCH,7,18 to END*** are used by this routine along with the old I-tape (from example 2) mounted on unit B7, to create a new, augmented I-tape on B1. After the correction--in this case the insertions of the new cost rows--the new I-tape is rewound to load point. The INPUT Agendum that follows may now operate on the new I-tape converting it to binary on the A-tape.

The PROP Agendum ADDRHS,CR follows. This Agendum has not been considered heretofore but its operation is straightforward. The right-hand-side elements to be added are read from the card reader (CR). These elements follow the Agendum card and include the END card. They are read, converted to binary and inserted in the appropriate place on the binary A-tape. The A-tape will now appear just as if it had been created solely by INPUT from the I-tape when the latter contained the newly added right-hand-sides.

Parametric programming is always performed after a given optimal solution has been found. For this example we use the solution of example 1 as the basic solution. Rather than perform a NORMAL using the first objective function and first RHS, we can read in the optimal basis as printed on the final solution output for 1 (Table 20). This is done with the PROP Agendum routine NEW.BASIS. The twelve cards following this Agendum card contain the names of the columns [J(H) column in Table 20 and the names of rows ROW(I) column in Table 20] in the optimal basis. Not included are those slack columns entered opposite the row of the same name.

The reading of these basis heading cards is terminated by the next Agendum, which <u>must be</u> INVERT. This Agendum causes the inverse of the basis just read in to be computed and the corresponding solution to be derived. As the latter is an optimal solution, the use of the following NORMAL is (in this case) a check only, but <u>must be present</u> in order to use PLP or PCR.

The CONTROLS card is inserted to provide a full solution print every iteration during the parametric programming computations.

The GETOFF,A7 is an Agendum which we have not encountered before. Its use here is to save the optimal solution we just reconstructed so that we can start each of six independent parametric programming computations with it. The routine stores on tape unit A7 all information necessary to reconstruct the current solution. This information is called back by the Agendum RESTART,A7. Its effect is to restore the machine to its condition following the NORMAL Agendum.

The RESET Agendum that follows sets the change parameter (θ or $\emptyset$) used in parametric programming to zero.

DO.PCR,1COTLT performs "price mapping" using the 1COTLT row as a change vector. Similarly for the next three applications of DO.PCR, mutatis mutandis. DO.PLP,4 and DO.PLP,5 perform "resource" mapping using the fourth and fifth right-hand-sides (change vectors) found on the A-tape respectively. RESET AND RESTART are used prior to each parametric programming exercise to insure beginning at the original optimal solution.

The Agendum Routines CORRECT.BCD and ADDRHS are PROP routines. The remainder are SHARE routines.

Table 35: Agenda for Parametric LP Example

(Listing Sheet 1)

```
HEADING      USDA LP/90    EXAMPLE 4       PARAMETRIC PROGRAMMING
CORRECT.BCD,B7,B1,CR,,1,,REWIND
MATCH,7,18
               Z2
INSERT,4
              1COTLT
              1UNSLB
              2COTLT
              2UNSLB
        1COT11 Z2    -   41.86
INSERT,4
        1COT111COTLT      3.10
        1COT111UNSLB     87.00
        1COT112COTLT-     3.10
        1COT112UNSLB-    87.00
        1COT12 Z2    -  119.59
INSERT,4
        1COT121COTLT      7.23
        1COT121UNSLB    165.00
        1COT122COTLT-     7.23
        1COT122UNSLB-   165.00
        1COT21 Z2    -   42.04
INSERT,4
        1COT211COTLT      2.96
        1COT211UNSLB     86.00
        1COT212COTLT-     2.96
        1COT212UNSLB-    86.00
        1COT22 Z2    -   64.38
INSERT,4
        1COT221COTLT      4.43
        1COT221UNSLB    113.00
        1COT222COTLT-     4.43
        1COT222UNSLB-   113.00
        2COT11 Z2    -   46.93
INSERT,4
        2COT111COTLT      2.79
        2COT111UNSLB     16.60
        2COT112COTLT-     2.79
        2COT112UNSLB-    16.60
        2COT12 Z2    -  138.98
INSERT,4
        2COT121COTLT      6.51
        2COT121UNSLB     16.60
        2COT122COTLT-     6.51
        2COT122UNSLB-    16.60
        2COT21 Z2    -   47.05
INSERT,4
        2COT211COTLT      2.66
        2COT211UNSLB     12.80
        2COT212COTLT-     2.66
        2COT212UNSLB-    12.80
        2COT22 Z2    -   74.64
INSERT,4
        2COT221COTLT      3.99
        2COT221UNSLB     12.80
        2COT222COTLT-     3.99
        2COT222UNSLB-    12.80
        1OAT11 Z2    -    6.81
INSERT,2
        1OAT111UNSLB       .50
        1OAT112UNSLB-      .50
```

(Listing Sheet 2)

```
        1OAT12 Z2    -  36.12
   INSERT,2
        1OAT121UNSLB     2.40
        1OAT122UNSLB-    2.40
        1OAT21 Z2    -   5.11
   INSFRT,2
        1OAT211UNSLB      .50
        1OAT212UNSLB-     .50
        1OAT22 Z2    -  30.17
   INSFRT,2
        1OAT221UNSLB     2.20
        1OAT222UNSLB-    2.20
        1SBG11 Z2    -  65.87
   INSFRT,2
        1SBG111UNSLB     1.10
        1SBG112UNSLB-    1.10
        1SBG21 Z2    -  33.27
   INSERT,2
        1SBG211UNSLB     1.10
        1SBG212UNSLB-    1.10
        1ALF11 Z2    -  34.52
   INSERT,2
        1ALF111UNSLB     2.80
        1ALF112UNSLB-    2.80
   END***
   INPUT,6,,COUNTS
   CONTROLS                             30
   ADDRHS,CR
   NEXT B,4                .
        1CHFLX UBCOT  880.
        1CHFLX UBOAT  520.
        1CHFLX UBSBG  650.
        1CHFLX UBALF  230.
        1CHFLX UBRIC  150.
        1CHFLX LBCOT- 640.
        1CHFLX LBOAT- 296.
        1CHFLX LBSBG- 370.
        1CHFLX LBALF- 152.
        1CHFLX LBRIC-  79.
   NEXT B,5                .
        2CHFLX UBCOT- 880.
        2CHFLX UBOAT- 520.
        2CHFLX UBSBG- 650.
        2CHFLX UBALF- 230.
        2CHFLX UBRIC- 150.
        2CHFLX LBCOT  640.
        2CHFLX LBOAT  296.
        2CHFLX LBSBG  370.
        2CHFLX LBALF  152.
        2CHFLX LBRIC   79.
   EOF
   TITLE       USDA LP/90   EXAMPLE 4      OBTAIN BASIC SOLUTION
   NEW.BASIS
        1COT12 SOIL1
        1OAT21 FERT
        1COT21 LBCOT
        1SBG11 LBOAT
        1SBG21 LBSBG
        1ALF11 LBALF
        1RIC22 LBRIC
        2COT12 S2COT
```

Table 35: Agenda for Parametric LP Example--Continued

(Listing Sheet 3)

```
        +LBOAT+SOIL2
        +LBCOT+LOTCT
        +LBRIC+LOTRC
        +LBSBG+UBSBG
 INVERT
 NORMAL,* Z1    ,1
 CONTROLS              1
 GETOFF,A7
 RESTART,A7
 TITLE        USDA LP/90    EXAMPLE 4        POS  BRANCH COT SUPPLY FUNCTION
 RESET
 DO.PCR,*1COTLT
 RESTART,A7
 TITLE        USDA LP/90    EXAMPLE 4        NEG  BRANCH COT SUPPLY FUNCTION
 RESET
 DO.PCR,*2COTLT
 RESTART,A7
 TITLE        USDA LP/90    EXAMPLE 4      POS BRANCH UNSLB DEMAND FUNCTION
 RESET
 DO.PCR,*1UNSLB
 RESTART,A7
 TITLE        USDA LP/90     EXAMPLE 4  NEG BRANCH UNSLAB DEMAND FUNCTION
 RESET
 DO.PCR,*2UNSLB
 RESTART,A7
 TITLE        USDA LP/90    EXAMPLE 4      POS BRANCH FLEXIBILITY RESPONSE
 RESET
 DO.PLP,4
 RESTART,A7
 TITLE        USDA LP/90    EXAMPLE 4      NEG BRANCH FLEXIBILITY RESPONSE
 RESET
 DO.PLP,5
 OUTPUT
```

d. Output

The output for this series of six parametric programming exercises is quite extensive, and, as the reader is by now already familiar with the iteration and full solution print-outs, it would serve no new purpose to reproduce it here. Instead the solution print-outs for each iteration are summarized in Tables 36, 37, and 38. The first of these tables gives the results of the first two (price mapping) exercises. The first three iterations give the negative branch of a cotton supply function and were computed by the Agendum DO.PCR,*ICOTLT. The next three iterations, computed by DO.PCR,*2COTLT, provide the positive branch. The computations in each case were terminated by the message "PHI UNBOUNDED" which means that the results of the last iteration can be extrapolated for all greater meaningful values of Ø.

Table 37 shows corresponding results for the next two parametric LP Agendum cards. These give a "demand function" for unskilled labor. Finally, Table 38 gives the last two exercises.

Figures 12 and 13 give the resulting supply curve of cotton and demand curve for unskilled labor respectively, while figure 14 shows graphically the results of the "resource mapping" exercise.

Table 36: Iteration Summary Giving Cotton Supply Function

Iteration Number	Value of Ø	Corresponding Cotton Price ($/cwt.)	Total Profit ($)	Cotton Acreage (Acres)	Cotton Production (cwt.)	Marginal Value Cotton Allot. ($/acre)
Negative Branch*						
0	0	35.00	83,188	700	3338.98	34.17
1	7.69	27.31	57,524	700	2167.43	11.42
2	11.12	23.88	50,081	700	2052.77	1.25
3	11.54	23.46	49,211	640	1875.17	0.00
Positive Branch**						
1	7.43	42.43	107,992	700	3538.61	56.16
2	41.03	76.03	226,886	700	3557.96	155.61
3	42.93	77.93	233,661	700	3606.84	161.25

*Cotton price = 35.00 - Ø.
**Cotton price = 35.00 + Ø.

Table 37: Iteration Summary Giving Unskilled Labor Demand Function

Iteration Number	Value of Ø	Corresponding Labor Price ($/hour)	Total Profit ($)	Labor Demand (Hours)
Positive Branch*				
0	.00	.34	83,188	81,359
1	.40	.74	50,673	76,229
2	.42	.76	48,772	55,020
3	.46	.80	46,907	52,239
4	4.18	4.52	-147,503	52,151
5	9.75	10.09	-437,984	52,139
6	11.39	11.73	-523,401	51,636
7	22.47	22.81	-1,095,763	51,328
Negative Branch**				
1	.20	.14	99,383	85,348
2***	.40	-.06	116,663	89,041
3	2.34	-2.00	289,037	89,939
4	117.64	-117.30	10,659,277	89,939

*Labor price =.34 + Ø. **Labor price = .34 - Ø. ***Iteration including and after this one are meaningless for a labor price cannot be negative.

Table 33: Iteration Summary Giving "Flexibility Response" Analysis

Iteration Number	Value of θ	Total Profit	Cotton Production	Cotton Acreage
Positive Branch				
0	0	83,188	3338.98	700
0*	.013	83,364	2228.98	700
Negative Branch				
1	.077	82,109	3338.98	700
2	.205	79,976	3338.98	700
3	.467	67,222	2656.21	469
4	.630	57,015	2335.66	326
5	.703	49,993	1871.97	262
6	.716	48,664	1788.15	250
7	.754	43,846	1545.59	216
8	.781	39,676	1375.91	193
9**	.969	6,210	175.00	27

*Infeasibility occurs at θ = .01265823 without change in original optimal basis. However, profit increases because of increase on soybean acreage at expense of alfalfa.

**Infeasibility occurs at θ = 1.00 without changing optimal basis at iteration 9.

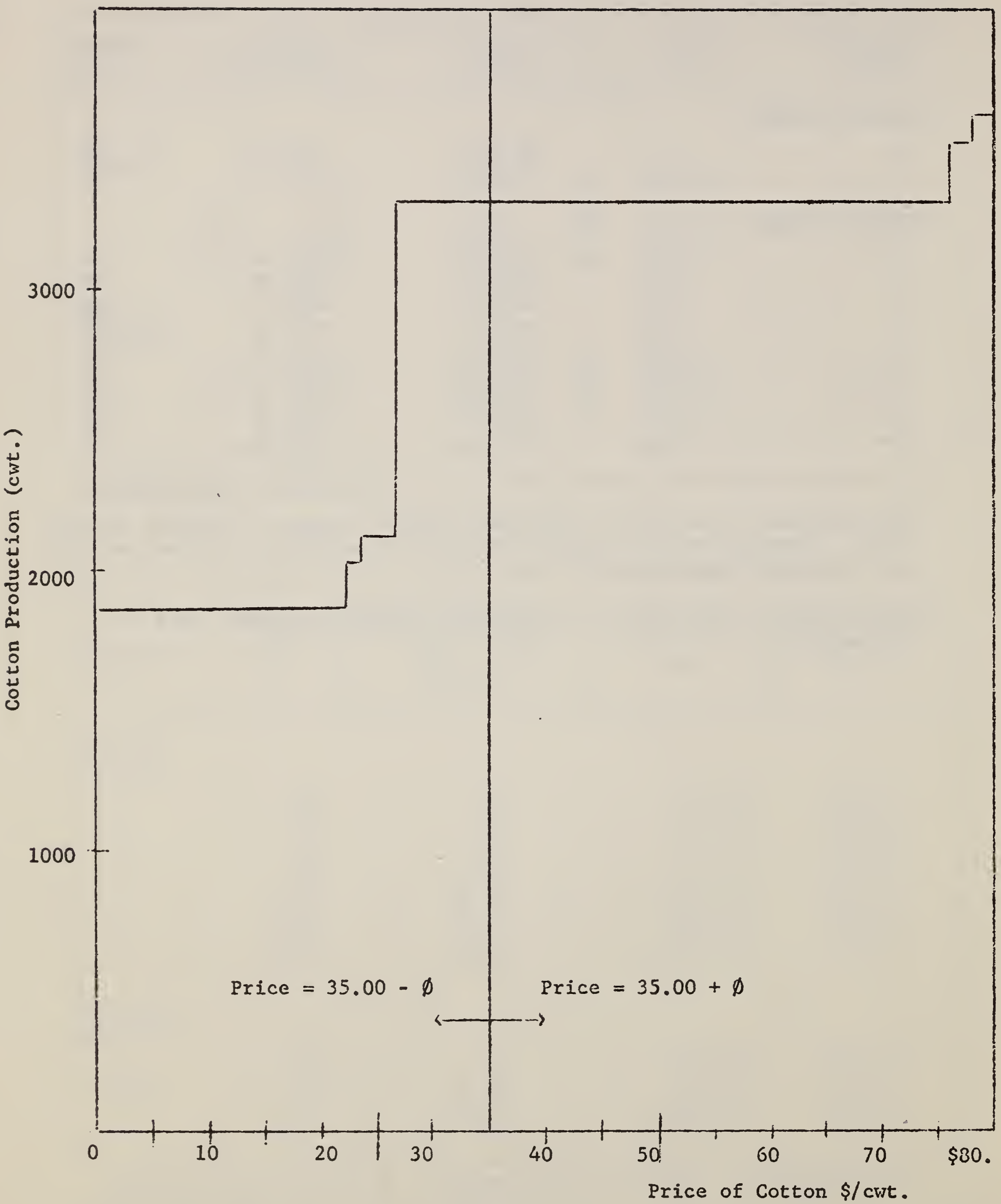


FIGURE 12:
SUPPLY FUNCTION FOR COTTON

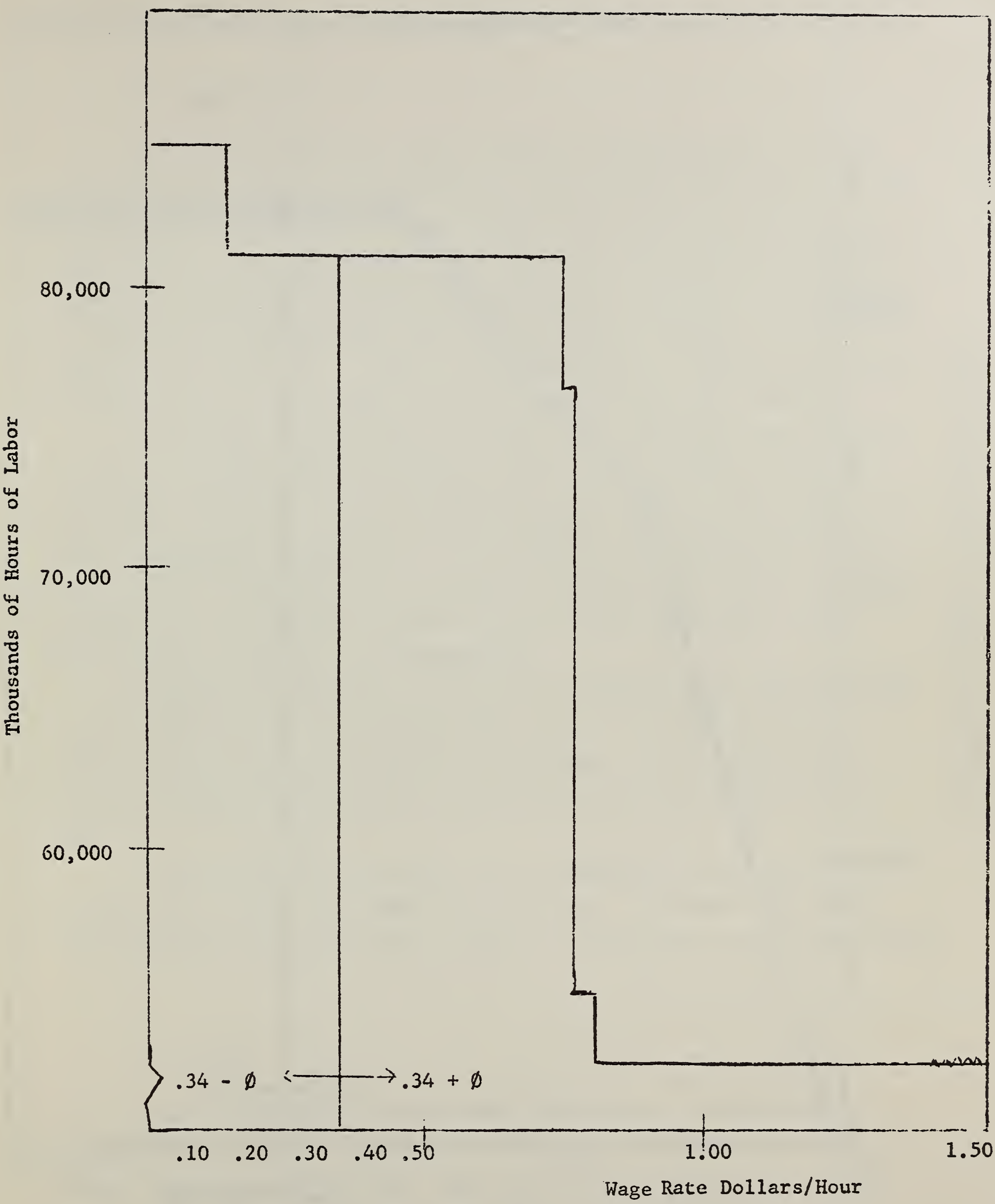


FIGURE 13:
DEMAND FUNCTION FOR LABOR

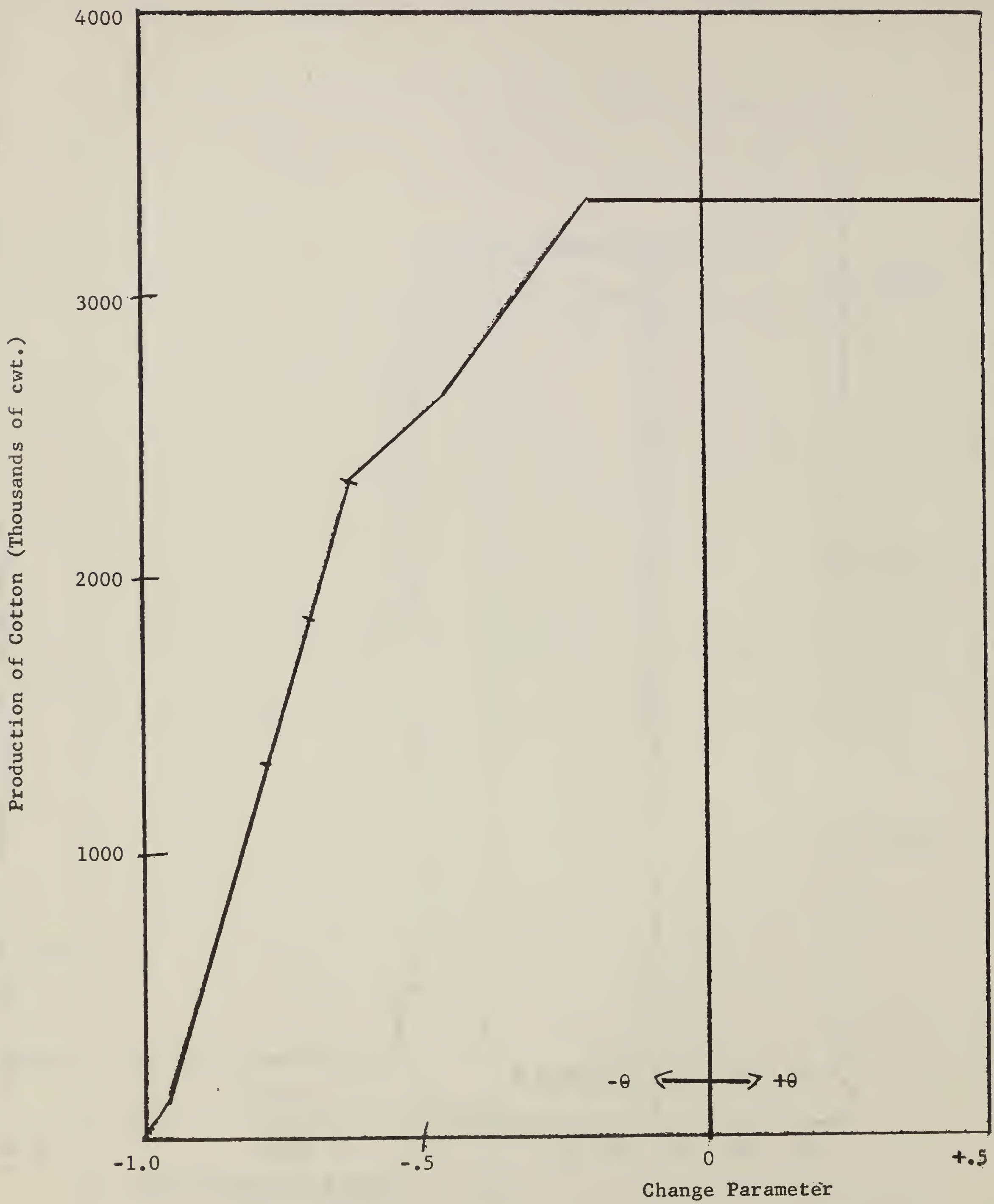


FIGURE 14: PRODUCTION OF COTTON AS AFFECTED BY PROPORTIONAL CHANGE IN THE FLEXIBILITY CONSTRAINTS

5. LP MATRIX TRANSFORMATION

a. Purpose

In the case of large scale LP models the preparation of the LP matrix data may constitute an excessive amount of time and money. As the various objective function, change vector, and constraint matrix coefficients are usually formed from more basic price and technical information, the natural question to be asked is, "Why not have a program to compute these coefficients from the basic data as well as have one to compute the LP solution." That is, why shouldn't the computer participate in preparing data for a model as well as provide the model solution?

So-called "matrix generators" have been developed whose purposes have been precisely that of computing the LP matrix elements from more basic data. The USDA version of LP/90 does not contain any of these "matrix generators." However, it does contain an Agendum routine which performs part of the function of a matrix generator, and which can be thought of as a particular variety of such a program species. This USDA Agendum routine is called TRANSFORM.

TRANSFORM multiplies two matrices in LP/90 I-tape format to form a third LP/90 A-tape. It thus provides a way of constructing an LP matrix from two rather more basic data matrices, or of modifying an existing LP matrix by scaling, recombining or deleting certain of its rows.

The purpose of this example is to illustrate one of the various possible uses of this new USDA Agendum routine. In doing so we shall create the input data for use in Example 6 on Recursive Linear Programming.

b. Data

Just as we did in the preceding exercises, we shall begin by augmenting the original (Example 1) model data. Two matrices, shown in Tables 39 and 40 will be used. The first is the G Matrix, the second, the T Matrix. During the execution of the Agendum TRANSFORM a third matrix, the H Matrix, will be created. It is the latter that will constitute an input matrix for LP solution. The former two, while in LP matrix (I-tape) form will not constitute meaningful LP matrix data by themselves.

The G Matrix: The G Matrix includes two types of coefficients: (1) the technical flow coefficients that describe how much of each output is expected per acre for each activity, and how much of each variable input is required per acre of each activity; and (2) the constraining coefficients that describe the active constraints which the process levels must satisfy. The latter have been fully described in section 1 above and shown in Table 18.

The former are arranged in rows corresponding to each output and variable input included in the model. In each row there will be one coefficient for each process (column) though some of these will be zero. Table 39 shows the structure of the G Matrix comprised of the two sets of coefficients. The flow coefficient (upper) portion will be called the A Matrix, the constraint coefficient (lower) portion will be called the B Matrix. The former has 23 rows, the latter 17 rows. Both have 17 columns of course. Thus G has 40 rows and 17 columns and may be written

$$G = \left[\frac{A}{B}\right] \ldots$$

The T Matrix: The T Matrix also contains two coefficient types: (1) several sets of prices for the inputs and outputs described in the G Matrix, each set corresponding to a given year, and (2) coefficients for carrying rows of the A or B matrices onto the new H Matrix.

The prices are arranged in rows, with one row for each complete set (or year) and one column for each row of the A Matrix. The row names are P1950, ..., P1958 and the column names are the same as the row names of the A Matrix. This 9 x 17 block we shall call the P^T Matrix.

The arrangement of the complete T Matrix is as follows:

$$T = \left[\begin{array}{c|c} P^T & 0 \\ \hline J_A & 0 \\ \hline 0 & J_B \end{array}\right] .$$

We can see then that TG = H yields

$$H = \begin{bmatrix} P^T A \\ \hdashline J_A A \\ \hdashline J_B B \end{bmatrix} = \begin{bmatrix} Z \\ \hdashline A' \\ \hdashline B' \end{bmatrix} .$$

The $P^T A = Z$ operation yields a 9 x 17 matrix, each column of which corresponds to a process and each row of which corresponds to a year. Its contents give the net return for each process in each year. The J_A is simply a matrix with one column for each row of the A Matrix and one row for each row of the A Matrix which we wish to carry onto the H Matrix. The complete T Matrix is shown in Table 40 and the H Matrix resulting from the transformation is shown in subsection d below in Table 42. The listing of the I-tape containing the G and H matrices is shown in Table 41. Both matrices are placed on the same I-tape as files one and two respectively. This I-tape also contains a third matrix in LP/90 format as a third file. The latter will be used in Example 6 and will be described there.

TABLE 39

G-MATRIX FOR MATRIX TRANSFORMATION

Row Symbol	Matrix Column Symbol											
	1COT11	1COT12	1COT21	1COT22	2COT11	2COT12	2COT21	2COT22	1OAT11	1OAT12	1OAT21	1OAT22
COTLT	3.10	7.23	2.96	4.43	2.79	6.51	2.66	3.99				
COTSD	5.29	12.32	5.04	7.56	4.67	11.09	4.54	6.80				
OATYD									14.40	65.10	12.90	58.40
SBGYD												
ALFYD												
RICYD												
SKLAB	8.30	10.50	8.20	9.80	13.34	16.14	13.04	14.44	3.53	5.90	5.00	7.10
UNSLB	87.00	165.00	86.00	113.00	16.60	16.60	12.80	12.80	0.50	2.40	0.50	2.20
MTRAC												
LTRAC	6.30	6.70	6.20	6.60	6.80	7.20	6.70	7.10	2.50	2.90	3.50	3.90
MMACH												
LMACH	6.30	6.30	6.20	6.60	6.80	7.20	6.70	7.10	2.50	2.90	3.50	3.90
PIKER					2.24	2.24	2.24	2.24				
COMB									0.40	0.40	0.40	0.40
SDCOT	45.00	45.00	33.00	33.00	90.00	90.00	60.00	60.00				
SCOAT									60.00	60.00	60.00	60.00
SDSBG												
SDALF												
SDRIC												
NITRO		4.00		4.00		4.00		4.00		4.00		4.00
DRY												
GIN	3.10	7.23	2.96	4.43	2.79	6.51	2.66	3.99				
WATER												

TABLE 39--Continued

Row Symbol	Matrix Column Symbol											
	1COT11	1COT12	1COT21	1COT22	2COT11	2COT12	2COT21	2COT22	10AT11	10AT12	1AOT21	10AT22
SOIL1	1.00	1.00			1.00	1.00			1.00	1.00		
SOIL2			1.00	1.00			1.00	1.00			1.00	1.00
LABP4	5.80	13.50	5.50	8.30								
FERT		4.00		4.00		4.00		4.00		4.00		4.00
LOTCT	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00				
LOTRC												
UBCOT	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00				
UBOAT									1.00	1.00	1.00	1.00
UBSBG												
UBALF												
UBRIC												
LBCOT	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00				
LBOAT									-1.00	-1.00	-1.00	-1.00
LBSBG												
LBALF												
LBRIC												
S2COT					2.79	6.51	2.73	4.89				

TABLE 39--Continued

Row Symbol	Matrix Column Symbol 1SBG11	1SBG21	1ALF11	1RIC21	1RIC22	Constraint Type	Dummy RHS C1
COTLT							
COTSD							
OATYD							
SBGYD	29.00	17.00					
ALFYD			2.07				
RICYD				17.60	35.60		
SKLAB	4.70	6.20	9.90	15.57	17.90		
*UNSLB	1.10	1.10	2.80				
LTRAC	3.00	4.40	0.70	6.20	7.70		
MMACH			9.20				
LMACH	3.00	4.40	0.70	6.20	7.70		
PIKER							
COMB	0.40	0.40		0.50	0.50		
SDCOT							
SDOAT							
SDSBG	60.00	60.00					
SDALF			15:00				
SDRIC				100.00	100.00		
NITRO							
DRY				17.60	35.60		
GIN							
WATER				1.00	1.00		
*MTRAC			6.40				

*Row MTRAC follows row UNSLB.

TABLE 40 --Continued

Row Symbol	Matrix Column Symbol					Con-straint Type	Dummy RHS
	1SBG11	1SBG21	1ALF11	1RIC21	1RIC22		C1
SOIL1	1.00		1.00			≤	500
SOIL2		1.00		1.00	1.00	≤	1500
LABP4	1.10	1.10	0.70			≤	
FERT					4.00	≤	
LOTCT						≤	
LOTRC				1.00	1.00	≤	
UBCOT						≤	
UBOAT						≤	
UBSBG	1.00	1.00				≤	
UBALF			1.00			≤	
UBRIC				1.00	1.00	≤	
LBCOT						≤	
LBOAT						≤	
LBSBG	- 1.00	- 1.00				≤	
LBALF			- 1.00			≤	
LBRIC				- 1.00	- 1.00	≤	
S2COT						≤	

TABLE 40

T-MATRIX FOR MATRIX TRANSFORMATION EXAMPLE

Row Symbol	Matrix Column Symbol											
	COTLT	COTSD	OATYD	SBGYD	ALFYD	RICYD	SKLAB	UNSLB	MTRAC	LTRAC	MMACH	LMACH
P1950	- 40.25	- 3.93	- 1.00	- 2.21	- 26.94	- 5.20	0.37	0.30	0.54	0.59	0.18	0.22
P1951	- 39.34	- 3.49	- 1.03	- 2.59	- 29.46	- 4.80	0.39	0.40	0.55	0.61	0.18	0.23
P1952	- 35.36	- 3.58	- 1.02	- 2.74	- 31.34	- 5.90	0.47	0.38	0.58	0.61	0.19	0.24
P1953	- 33.82	- 2.69	- 0.93	- 2.60	- 26.54	- 4.90	0.46	0.37	0.57	0.62	0.19	0.24
P1954	- 34.48	- 3.11	- 0.87	- 2.46	- 26.40	- 4.50	0.42	0.34	0.57	0.63	0.19	0.24
P1955	- 33.63	- 2.27	- 0.76	- 2.10	- 23.60	- 5.03	0.43	0.35	0.56	0.62	0.19	0.24
P1956	- 32.86	- 2.59	- 0.72	- 2.14	- 20.94	- 5.02	0.48	0.39	0.55	0.59	0.20	0.25
P1957	- 28.30	- 2.52	- 0.69	- 2.03	- 21.92	- 5.31	0.49	0.40	0.58	0.63	0.22	0.27
P1958	- 26.88	- 2.39	- 0.76	- 1.95	- 22.94	- 5.20	0.48	0.39	0.56	0.60	0.22	0.27
1COTLT	1.00											
1COTSD												
1DATYD			1.00									
1SBGYD				1.00								
1ALFYD					1.00							
1RICYD						1.00						
1SKLAB							1.00					
1UNSLB												
1MTRAC												
1LTRAC												
1MMACH												
1LMACH												
1PIKER												
1COMB												
1SDCOT												
1SDOAT												

TABLE 40 --Continued

Row Symbol	Matrix Column Symbol											
	COTLT	COTSD	OATYD	SBGYD	ALFYD	RICYD	SKLAB	UNSLB	MTRAC	LTRAC	MMACH	LMACH
1SDSBG												
1SDALF												
1SDRIC												
1NITRO						ALL ZEROS						
1DRY												
1GIN												
1WATER												
2COTLT	- 1.00											
2COTSD												
2OATYD			- 1.00									
2SBGYD				- 1.00								
2ALFYD					- 1.00							
2SKLAB						- 1.00						
2UNSLB												
2MTRAC												
2LTRAC												
2MMACH												
2LMACH												
2PIKER												
2COMB												
2SDCOT												
2SDOAT												

TABLE40 --Continued

Row Symbol	Matrix Column Symbol											
	COTLT	COTSD	OATYD	SBGYD	ALFYD	RICYD	SKLAB	UNSLB	MTRAC	LTRAC	MMACH	LMACH
2SDSBG												
2SDALF												
2SDRIC												
2NITRO						ALL ZEROS						
2DRY												
2GIN												
2WATER												
SOIL1												
SOIL2												
LABP4												
FERT												
LOTCT												
LOTRC												
UBCOT												
UBOAT						ALL ZEROS						
UBSBG												
UBALF												
UBRIC												
LBCOT												
LBOAT												
LBSBG												
LBALF												
LBRIC												
S2COT												

TABLE 40--Continued

Row Symbol	Matrix Column Symbol										
	PIKER	COMB	SDCOT	SDOAT	SDSBG	SDALF	SDRIC	NITRO	DRY	GIN	WATER
P1950	1.78	2.02	0.09	0.06	0.07	0.49	0.09	2.47	0.33	2.49	7.20
P1951	1.85	2.09	0.14	0.06	0.08	0.56	0.08	2.84	0.33	2.63	7.49
P1952	1.90	2.14	0.11	0.05	0.09	0.60	0.10	2.85	0.33	2.64	8.74
P1953	1.90	2.16	0.09	0.05	0.09	0.45	0.09	2.88	0.33	2.70	8.14
P1954	1.92	2.17	0.08	0.05	0.08	0.44	0.08	2.75	0.33	2.78	7.91
P1955	1.91	2.15	0.09	0.04	0.09	0.37	0.09	2.55	0.33	2.78	7.85
P1956	1.94	2.14	0.07	0.04	0.07	0.37	0.09	2.49	0.33	2.97	7.96
P1957	2.08	2.30	0.07	0.05	0.07	0.36	0.09	2.26	0.33	3.13	8.34
P1958	2.05	2.24	0.12	0.05	0.08	0.37	0.09	2.41	0.33	3.15	8.01
1COTLT											
1COTSD											
1DATYD											
1SBGYD											
1ALFYD											
1RICYD											
1SKLAB											
1UNSLB						ALL ZEROS					
1MTRAC											
1LTRAC											
1MMACH											
1LMACH											
1PIKER											
1COMB											
1SDCOT											
1SDOAT											

TABLE 40--<u>Continued</u>

Row Symbol	Matrix Column Symbol										
	PIKER	COMB	SDCOT	SDOAT	SDSBG	SDALF	SDRIC	NITRO	DRY	GIN	WATER
1SDSBG											
1SDALF											
1SDRIC											
INITRO						ALL ZEROS					
1DRY											
1GIN											
1WATER											
2COTLT											
2COTSD											
2OATYD											
2SBGYD											
2ALFYD											
2RICYD											
2SKLAB						ALL ZEROS					
2UNSLB											
2MTRAC											
2LTRAC											
2MMACH											
2LMACH											
2PIKER											
2COMB											
2SDCOT											
2SDOAT											

TABLE 40--Continued

Row Symbol	Matrix Column Symbol										
	PIKER	COMB	SDCOT	SDOAT	SDSBG	SDALF	SDRIC	NITRO	DRY	GIN	WATER
2SDSBG											
2SDALF											
2SDRIC											
2NITRO						ALL ZEROS					
2DRY											
2GIN											
2WATER											
SOIL1											
SOIL2											
LABP4											
FERT											
LOTCT											
LOTRC											
UBCOT											
UBOAT											
UBSBG											
UBALF						ALL ZEROS					
UBRIC											
LBCOT											
LBOAT											
LBSBG											
LBALF											
LBRIC											
S2COT											

TABLE 40-Continued

Row Symbol	Matrix Column Symbol											
	SOIL1	SOIL2	LABP4	FERT	LOTCT	LOTRC	UBCOT	UBOAT	UBSBG	UBALF	UBRSC	LBCOT
P1950												
P1951												
P1952												
P1953												
P1954												
P1955						ALL ZEROS						
P1956												
P1957												
P1958												
1COTLT												
1COTSD												
1OATYD												
1SBGYD												
1ALFYD												
1RICYD												
1SKLAB												
1UNSLB						ALL ZEROS						
1MTRAC												
1LTRAC												
1MMACH												
1LMACH												
1PIKER												
1COMB												
1SDCOT												
1SDOAT												

TABLE 40 --Continued

Row Symbol	Matrix Column Symbol											
	SOIL1	SOIL2	LABP4	FERT	LOTCT	LOTRC	UBCOT	UBOAT	UBSBG	UBALF	UBRIC	LBCOT
1SDSBG												
1SDALF												
1SDRIC						ALL ZEROS						
1NITRO												
1DRY												
1GIN												
1WATER												
2COTLT												
2COTSD												
2OATYD												
2SBGYD												
2ALFYD												
2RICYD												
2SKLAB												
2UNSLB												
2LTRAC						ALL ZEROS						
2MMACH												
2LMACH												
2PIKER												
2COMB												
2SDCOT												
2SDOAT												

TABLE 40 --Continued

Row Symbol	Matrix Column Symbol											
	SOIL1	SOIL2	LABP4	FERT	LOTCT	LOTRC	UBCOT	UBOAT	UBSBG	UBALF	UBRIC	LBCOT
2SDSBG												
2SDALF												
2SDRIC												
2NITRO						ALL ZEROS						
2DRY												
2GIN												
2WATER												
SOIL1	1.00											
SOIL2		1.00										
LABP4			1.00									
FERT				1.00								
LOTCT					1.00							
LOTRC						1.00						
UBCOT							1.00					
UBOAT								1.00				
UBSBG									1.00			
UBALF										1.00		
UBRIC											1.00	
LBCOT												1.00
LBOAT												
LBSBG												
LBALF												
LBRIC												
S2COT												

TABLE 40--Continued

Row Symbol	Matrix Column Symbol					Con-straint Type	Right-Hand-Sides	
	LBOAT	LBSBG	LBALF	LBRIC	S2COT		FIRST B	NEXT B,50 C 1950
P1950								
P1951								
P1952								
P1953								
P1954				ALL ZEROS				
P1955								
P1956								
P1957								
P1958								
1COTLT								
1COTSD								
1DATYD								
1SBGYD								
1ALFYD				ALL ZEROS				
1RICYD								
1SKLAB								
1UNSLB								
1MTRAC								
1LTRAC								
1MMACH								
1LMACH								
1PIKER								
1COMB								
1SDCOT								
1SDOAT								

TABLE 40,--Continued

Row Symbol	Matrix Column Symbol					Con-straint Type	Right-Hand-Sides	
	LBOAT	LBSBG	LBALF	LBRIC	S2COT		FIRST B	NEXT B,50 C 1950
1SDSBG								
1SDALF								
1SDRIC								
1NITRO				ALL ZEROS				
1DRY								
1GIN								
1WATER								
2COTLT								
2COTSD								
2OATYD								
2SBGYD								
2ALFYD								
2RICYD								
2SKLAB								
2UNSLB				ALL ZEROS				
2MTRAC								
2LTRAC								
2MMACH								
2LMACH								
2PIKER								
2COMB								
2SDCOT								
2SDOAT								

TABLE 40--Continued

Row Symbol	Matrix Column Symbol					Con-straint Type	Right-Hand-Sides	
	LBOAT	LBSBG	LBALF	LBRIC	S2COT		FIRST B	NEXT B,50 C 1950
2SDSBG								
2SDALF								
2SDRIC								
2NITRO				ALL ZEROS				
2DRY								
2GIN								
2WATER								
SOIL1						≤		1500
SOIL2						≤		500
LABP4						≤		9900
FERT						≤		1525
LOTCT						≤		700
LOTRC						≤		5000
UBCOT						≤		
UBOAT						≤		
UBSBG						≤		
UBALF						≤		
UBRIC						≤		
LBCOT						≤		
LBOAT	1.00					≤		
LBSBG		1.00				≤		
LBALF			1.00			≤		
LBRIC				1.00		≤		
S2COT					1.00	≤		

Table 41: I-Tape Listing for Matrix Transformation Example

(Listing Sheet 1)

```
*           USDA LP/90    EXAMPLE   5    GMATRIX
ROW ID
              COTLT
              COTSD
              OATYD
              SBGYD
              ALFYD
              RICYD
              SKLAB
              UNSLB
              MTRAC
              LTRAC
              MMACH
              LMACH
              PIKER
              COMB
              SDCOT
              SDOAT
              SDSBG
              SDALF
              SDRIC
              NITRO
              DRY
              GIN
              WATER
            + SOIL1
            + SOIL2
            + LABP4
            + FERT
            + LOTCT
            + LOTRC
            + UBCOT
            + UBOAT
            + UBSBG
            + UBALF
            + UBRIC
            + LBCOT
            + LBOAT
            + LBSBG
            + LBALF
            + LBRIC
            + S2COT
MATRIX
       1COT11 COTLT      3.10
       1COT11 COTSD      5.29
       1COT11 SKLAB      8.30
       1COT11 UNSLB     87.00
       1COT11 LTRAC      6.30
       1COT11 LMACH      6.30
       1COT11 SDCOT     45.00
       1COT11 GIN        3.10
       1COT11 SOIL1      1.00
       1COT11 LABP4      5.80
       1COT11 LOTCT      1.00
       1COT11 UBCOT      1.00
       1COT11 LBCOT-     1.00
       1COT12 COTLT      7.23
       1COT12 COTSD     12.32
```

(Listing Sheet 2)

```
1COT12 SKLAB     10.50
1COT12 UNSLB    165.00
1COT12 LTRAC      6.70
1COT12 LMACH      6.30
1COT12 SDCOT     45.00
1COT12 NITRO      4.00
1COT12 GIN        7.23
1COT12 SOIL1      1.00
1COT12 LABP4     13.50
1COT12 FERT       4.00
1COT12 LOTCT      1.00
1COT12 UBCOT      1.00
1COT12 LBCOT-     1.00
1COT21 COTLT      2.96
1COT21 COTSD      5.04
1COT21 SKLAB      8.20
1COT21 UNSLB     86.00
1COT21 LTRAC      6.20
1COT21 LMACH      6.20
1COT21 SDCOT     33.00
1COT21 GIN        2.96
1COT21 SOIL2      1.00
1COT21 LABP4      5.50
1COT21 LOTCT      1.00
1COT21 UBCOT      1.00
1COT21 LBCOT-     1.00
1COT22 COTLT      4.43
1COT22 COTSD      7.56
1COT22 SKLAB      9.80
1COT22 UNSLB    113.00
1COT22 LTRAC      6.60
1COT22 LMACH      6.60
1COT22 SDCOT     33.00
1COT22 NITRO      4.00
1COT22 GIN        4.43
1COT22 SOIL2      1.00
1COT22 LABP4      8.30
1COT22 FERT       4.00
1COT22 LOTCT      1.00
1COT22 UBCOT      1.00
1COT22 LBCOT-     1.00
2COT11 COTLT      2.79
2COT11 COTSD      4.67
2COT11 SKLAB     13.34
2COT11 UNSLB     16.60
2COT11 LTRAC      6.80
2COT11 LMACH      6.80
2COT11 PIKER      2.24
2COT11 SDCOT     90.00
2COT11 GIN        2.79
2COT11 SOIL1      1.00
2COT11 LOTCT     .1.00
2COT11 UBCOT      1.00
2COT11 LBCOT-     1.00
2COT11 S2COT      2.79
2COT12 COTLT      6.51
2COT12 COTSD     11.09
2COT12 SKLAB     16.14
```

(Listing Sheet 3)

```
2COT12 UNSLB     16.60
2COT12 LTRAC      7.20
2COT12 LMACH      7.20
2COT12 PIKER      2.24
2COT12 SDCOT     90.00
2COT12 NITRO      4.00
2COT12 GIN        6.51
2COT12 SOIL1      1.00
2COT12 FERT       4.00
2COT12 LOTCT      1.00
2COT12 UBCOT      1.00
2COT12 LBCOT-     1.00
2COT12 S2COT      6.51
2COT21 COTLT      2.66
2COT21 COTSD      4.54
2COT21 SKLAB     13.04
2COT21 UNSLB     12.80
2COT21 LTRAC      6.70
2COT21 LMACH      6.70
2COT21 PIKER      2.24
2COT21 SDCOT     60.00
2COT21 GIN        2.66
2COT21 SOIL2      1.00
2COT21 LOTCT      1.00
2COT21 UBCOT      1.00
2COT21 LBCOT-     1.00
2COT21 S2COT      2.73
2COT22 COTLT      3.99
2COT22 COTSD      6.80
2COT22 SKLAB     14.44
2COT22 UNSLB     12.80
2COT22 LTRAC      7.10
2COT22 LMACH      7.10
2COT22 PIKER      2.24
2COT22 SDCOT     60.00
2COT22 NITRO      4.00
2COT22 GIN        3.99
2COT22 SOIL2      1.00
2COT22 FERT       4.00
2COT22 LOTCT      1.00
2COT22 UBCOT      1.00
2COT22 LBCOT-     1.00
2COT22 S2COT      4.89
1OAT11 OATYD     14.40
1OAT11 SKLAB      3.53
1OAT11 UNSLB      0.50
1OAT11 LTRAC      2.50
1OAT11 LMACH      2.50
1OAT11 COMB       0.40
1OAT11 SDOAT     60.00
1OAT11 SOIL1      1.00
1OAT11 UBOAT      1.00
1OAT11 LBOAT-     1.00
1OAT12 OATYD     65.10
1OAT12 SKLAB      5.90
1OAT12 UNSLB      2.40
1OAT12 LTRAC      2.90
1OAT12 LMACH      2.90
```


PRINTED IN U S A
FORM 1413
MOORE BUSINESS FORMS, INC.

(Listing Sheet 4)

```
10AT12 COMB      0.40
10AT12 SDOAT    60.00
10AT12 NITRO     4.00
10AT12 SOIL1     1.00
10AT12 FERT      4.00
10AT12 UBOAT     1.00
10AT12 LBOAT-    1.00
10AT21 OATYD    12.90
10AT21 SKLAB     5.00
10AT21 UNSLB     0.50
10AT21 LTRAC     3.50
10AT21 LMACH     3.50
10AT21 COMB      0.40
10AT21 SDOAT    60.00
10AT21 SOIL2     1.00
10AT21 UBOAT     1.00
10AT21 LBOAT-    1.00
10AT22 OATYD    58.40
10AT22 SKLAB     7.10
10AT22 UNSLB     2.20
10AT22 LTRAC     3.90
10AT22 LMACH     3.90
10AT22 COMB      0.40
10AT22 SDOAT    60.00
10AT22 NITRO     4.00
10AT22 SOIL2     1.00
10AT22 FERT      4.00
10AT22 UBOAT     1.00
10AT22 LBOAT-    1.00
1SBG11 SBGYD    29.00
1SBG11 SKLAB     4.70
1SBG11 UNSLB     1.10
1SBG11 LTRAC     3.00
1SBG11 LMACH     3.00
1SBG11 COMB      0.40
1SBG11 SDSBG    60.00
1SBG11 SOIL1     1.00
1SBG11 LABP4     1.10
1SBG11 UBSBG     1.00
1SBG11 LBSBG-    1.00
1SBG21 SBGYD    17.00
1SBG21 SKLAB     6.20
1SBG21 UNSLB     1.10
1SBG21 LTRAC     4.40
1SBG21 LMACH     4.40
1SBG21 COMB      0.40
1SBG21 SDSBG    60.00
1SBG21 SOIL2     1.00
1SBG21 LABP4     1.10
1SBG21 UBSBG     1.00
1SBG21 LBSBG-    1.00
1ALF11 ALFYD     2.07
1ALF11 SKLAB     9.90
1ALF11 UNSLB     2.80
1ALF11 MTRAC     6.40
1ALF11 LTRAC     0.70
1ALF11 MMACH     9.20
1ALF11 LMACH     0.70
```


(Listing Sheet 5)

```
          1ALF11 SDALF    15.00
          1ALF11 SOIL1     1.00
          1ALF11 LABP4     0.70
          1ALF11 UBALF     1.00
          1ALF11 LBALF-    1.00
          1RIC21 RICYD    17.60
          1RIC21 SKLAB    15.57
          1RIC21 LTRAC     6.20
          1RIC21 LMACH     6.20
          1RIC21 COMB      0.50
          1RIC21 SDRIC   100.00
          1RIC21 DRY      17.60
          1RIC21 WATER     1.00
          1RIC21 SOIL2     1.00
          1RIC21 LOTRC     1.00
          1RIC21 UBRIC     1.00
          1RIC21 LBRIC-    1.00
          1RIC22 RICYD    35.60
          1RIC22 SKLAB    17.90
          1RIC22 LTRAC     7.70
          1RIC22 LMACH     7.70
          1RIC22 COMB      0.50
          1RIC22 SDRIC   100.00
          1RIC22 DRY      35.60
          1RIC22 WATER     1.00
          1RIC22 SOIL2     1.00
          1RIC22 FERT      4.00
          1RIC22 LOTRC     1.00
          1RIC22 UBRIC     1.00
          1RIC22 LBRIC-    1.00
FIRST B
         C1      SOIL1  500
         C1      SOIL2 1500
EOF
*              USDA LP/90    EXAMPLE  5    TMATRIX
ROW ID
                  P1950
                  P1951
                  P1952
                  P1953
                  P1954
                  P1955
                  P1956
                  P1957
                  P1958
                1COTLT
                1COTSD
                1OATYD
                1SBGYD
                1ALFYD
                1RICYD
                1SKLAB
                1UNSLB
                1MTRAC
                1LTRAC
                1MMACH
                1LMACH
                1PIKER
```


(Listing Sheet 6)

```
                 1COMB
                 1SDCOT
                 1SDOAT
                 1SDSBG
                 1SDALF
                 1SDRIC
                 1NITRO
                 1DRY
                 1GIN
                 1WATER
                 2COTLT
                 2COTSD
                 2OATYD
                 2SBGYD
                 2ALFYD
                 2RICYD
                 2SKLAB
                 2UNSLB
                 2MTRAC
                 2LTRAC
                 2MMACH
                 2LMACH
                 2PIKER
                 2COMB
                 2SDCOT
                 2SDOAT
                 2SDSBG
                 2SDALF
                 2SDRIC
                 2NITRO
                 2DRY
                 2GIN
                 2WATER
                + SOIL1
                + SOIL2
                + LABP4
                + FERT
                + LOTCT
                + LOTRC
                + UBCOT
                + UBOAT
                + UBSBG
                + UBALF
                + UBRIC
                + LBCOT
                + LBOAT
                + LBSBG
                + LBALF
                + LBRIC
                + S2COT
MATRIX
         COTLT P1950-   40.25
         COTLT P1951-   39.34
         COTLT P1952-   35.36
         COTLT P1953-   33.82
         COTLT P1954-   34.48
         COTLT P1955-   33.63
         COTLT P1956-   32.86
```

Table 41: I-Tape Listing for Matrix Transformation Example--Continued

(Listing Sheet 7)

COTLT P1957-	28.30
COTLT P1958-	26.88
COTLT1COTLT	1.00
COTLT2COTLT-	1.00
COTSD P1950-	3.93
COTSD P1951-	3.49
COTSD P1952-	3.58
COTSD P1953-	2.69
COTSD P1954-	3.11
COTSD P1955-	2.27
COTSD P1956-	2.59
COTSD P1957-	2.52
COTSD P1958-	2.39
OATYD P1950-	1.00
OATYD P1951-	1.03
OATYD P1952-	1.02
OATYD P1953-	.93
OATYD P1954-	.87
OATYD P1955-	.76
OATYD P1956-	.72
OATYD P1957-	.69
OATYD P1958-	.76
OATYD1OATYD	1.00
OATYD2OATYD-	1.00
SBGYD P1950-	2.21
SBGYD P1951-	2.59
SBGYD P1952-	2.74
SBGYD P1953-	2.60
SBGYD P1954-	2.46
SBGYD P1955-	2.10
SBGYD P1956-	2.14
SBGYD P1957-	2.03
SBGYD P1958-	1.95
SBGYD1SBGYD	1.00
SBGYD2SBGYD-	1.00
ALFYD P1950-	26.94
ALFYD P1951-	29.46
ALFYD P1952-	31.34
ALFYD P1953-	26.54
ALFYD P1954-	26.40
ALFYD P1955-	23.60
ALFYD P1956-	20.94
ALFYD P1957-	21.92
ALFYD P1958-	22.94
ALFYD1ALFYD	1.00
ALFYD2ALFYD-	1.00
RICYD P1950-	5.20
RICYD P1951-	4.80
RICYD P1952-	5.90
RICYD P1953-	4.90
RICYD P1954-	4.50
RICYD P1955-	5.03
RICYD P1956-	5.02
RICYD P1957-	5.31
RICYD P1958-	5.20
RICYD1RICYD	1.00
RICYD2RICYD-	1.00
SKLAB P1950	.37

(Listing Sheet 8)

SKLAB P1951	.39
SKLAB P1952	.47
SKLAB P1953	.46
SKLAB P1954	.42
SKLAB P1955	.43
SKLAB P1956	.48
SKLAB P1957	.49
SKLAB P1958	.48
SKLAB1SKLAB	1.00
SKLAB2SKLAB-	1.00
UNSLB P1950	.30
UNSLB P1951	.40
UNSLB P1952	.38
UNSLB P1953	.37
UNSLB P1954	.34
UNSLB P1955	.35
UNSLB P1956	.39
UNSLB P1957	.40
UNSLB P1958	.39
MTRAC P1950	.54
MTRAC P1951	.55
MTRAC P1952	.58
MTRAC P1953	.57
MTRAC P1954	.57
MTRAC P1955	.56
MTRAC P1956	.55
MTRAC P1957	.58
MTRAC P1958	.56
LTRAC P1950	.59
LTRAC P1951	.61
LTRAC P1952	.61
LTRAC P1953	.62
LTRAC P1954	.63
LTRAC P1955	.62
LTRAC P1956	.59
LTRAC P1957	.63
LTRAC P1958	.60
MMACH P1950	.18
MMACH P1951	.18
MMACH P1952	.19
MMACH P1953	.19
MMACH P1954	.19
MMACH P1955	.19
MMACH P1956	.20
MMACH P1957	.22
MMACH P1958	.22
LMACH P1950	.22
LMACH P1951	.23
LMACH P1952	.24
LMACH P1953	.24
LMACH P1954	.24
LMACH P1955	.24
LMACH P1956	.25
LMACH P1957	.27
LMACH P1958	.27
PIKER P1950	1.78
PIKER P1951	1.85
PIKER P1952	1.90

(Listing Sheet 9)

PIKER P1953	1.90
PIKER P1954	1.92
PIKER P1955	1.91
PIKER P1956	1.94
PIKER P1957	2.08
PIKER P1958	2.05
COMB P1950	2.02
COMB P1951	2.09
COMB P1952	2.14
COMB P1953	2.16
COMB P1954	2.17
COMB P1955	2.15
COMB P1956	2.14
COMB P1957	2.30
COMB P1958	2.24
SDCOT P1950	.09
SDCOT P1951	.14
SDCOT P1952	.11
SDCOT P1953	.09
SDCOT P1954	.08
SDCOT P1955	.09
SDCOT P1956	.07
SDCOT P1957	.07
SDCOT P1958	.12
SDOAT P1950	.06
SDOAT P1951	.06
SDOAT P1952	.05
SDOAT P1953	.05
SDOAT P1954	.05
SDOAT P1955	.04
SDOAT P1956	.04
SDOAT P1957	.05
SDOAT P1958	.05
SDSBG P1950	.07
SDSBG P1951	.08
SDSBG P1952	.09
SDSBG P1953	.09
SDSBG P1954	.08
SDSBG P1955	.09
SDSBG P1956	.07
SDSBG P1957	.07
SDSBG P1958	.08
SDALF P1950	.49
SDALF P1951	.56
SDALF P1952	.60
SDALF P1953	.45
SDALF P1954	.44
SDALF P1955	.37
SDALF P1956	.37
SDALF P1957	.36
SDALF P1958	.37
SDRIC P1950	.09
SDRIC P1951	.08
SDRIC P1952	.10
SDRIC P1953	.09
SDRIC P1954	.08
SDRIC P1955	.09
SDRIC P1956	.09

PRINTED IN U.S.A. FORM 1413 MOORE BUSINESS FORMS, INC.

(Listing Sheet 10)

```
SDRIC P1957      .09
SDRIC P1958      .09
NITRO P1950     2.47
NITRO P1951     2.84
NITRO P1952     2.85
NITRO P1953     2.88
NITRO P1954     2.75
NITRO P1955     2.55
NITRO P1956     2.49
NITRO P1957     2.26
NITRO P1958     2.41
DRY   P1950      .33
DRY   P1951      .33
DRY   P1952      .33
DRY   P1953      .33
DRY   P1954      .33
DRY   P1955      .33
DRY   P1956      .33
DRY   P1957      .33
DRY   P1958      .33
GIN   P1950     2.49
GIN   P1951     2.63
GIN   P1952     2.64
GIN   P1953     2.70
GIN   P1954     2.78
GIN   P1955     2.78
GIN   P1956     2.97
GIN   P1957     3.13
GIN   P1958     3.15
WATER P1950     7.20
WATER P1951     7.49
WATER P1952     8.74
WATER P1953     8.14
WATER P1954     7.91
WATER P1955     7.85
WATER P1956     7.96
WATER P1957     8.34
WATER P1958     8.01
SOIL1 SOIL1     1.00
SOIL2 SOIL2     1.00
LABP4 LABP4     1.00
FERT  FERT      1.00
LOTCT LOTCT     1.00
LOTRC LOTRC     1.00
UBCOT UBCOT     1.00
UBOAT UBOAT     1.00
UBSBG UBSBG     1.00
UBALF UBALF     1.00
UBRIC UBRIC     1.00
LBCOT LBCOT     1.00
LBOAT LBOAT     1.00
LBSBG LBSBG     1.00
LBALF LBALF     1.00
LBRIC LBRIC     1.00
S2COT S2COT     1.00
FIRST B
NEXT B,50
C1950 SOIL1 1500.
```

(Listing Sheet 11)

```
            C1950 SOIL2  500.
            C1950 LABP4 9900.
            C1950 FERT  1525.
            C1950 LOTCT  700.
            C1950 LOTRC 5000.
EOF
* USDA LP/90  DUMMY CHANGE MATRIX AND EXOGENOUS VECTORS FOR RLP EXAMPL 6
ROW ID
                   P1950
                   P1951
                   P1952
                   P1953
                   P1954
                   P1955
                   P1956
                   P1957
                   P1958
                  1COTLT
                  1COTSD
                  1OATYD
                  1SBGYD
                  1ALFYD
                  1RICYD
                  1SKLAB
                  1UNSLB
                  1MTRAC
                  1LTRAC
                  1MMACH
                  1LMACH
                  1PIKER
                  1COMB
                  1SDCOT
                  1SDOAT
                  1SDSBG
                  1SDALF
                  1SDRIC
                  1NITRO
                  1DRY
                  1GIN
                  1WATER
                  2COTLT
                  2COTSD
                  2OATYD
                  2SBGYD
                  2ALFYD
                  2RICYD
                  2SKLAB
                  2UNSLB
                  2MTRAC
                  2LTRAC
                  2MMACH
                  2LMACH
                  2PIKER
                  2COMB
                  2SDCOT
                  2SDOAT
                  2SDSBG
                  2SDALF
```

Table 41: I-Tape Listing for Matrix Transformation Example--Continued

(Listing Sheet 12)

```
              2SDRIC
              2NITRO
              2DRY
              2GIN
              2WATER
            + SOIL1
            + SOIL2
            + LABP4
            + FERT
            + LOTCT
            + LOTRC
            + UBCOT
            + UBOAT
            + UBSBG
            + UBALF
            + UBRIC
            + LBCOT
            + LBOAT
            + LBSBG
            + LBALF
            + LBRIC
            + S2COT
MATRIX
        1COT11 SOIL1     1.0
FIRST B
         C1950 FERT  2000.
NEXT B,51
         C1951 SOIL1 1500.
         C1951 SOIL2  500.
         C1951 FERT  2000.
         C1951 LOTCT 5000.
         C1951 LOTRC 5000.
NEXT B,52
         C1952 SOIL1 1500.
         C1952 SOIL2  500.
         C1952 FERT  2000.
         C1952 LOTCT 5000.
         C1952 LOTRC 5000.
NEXT B,53
         C1953 SOIL1 1500.
         C1953 SOIL2  500.
         C1953 FERT  2000.
         C1953 LOTCT 5000.
         C1953 LOTRC 5000.
NEXT B,54
         C1954 SOIL1 1500.
         C1954 SOIL2  500.
         C1954 FERT  2000.
         C1954 LOTCT 1000.
         C1954 LOTRC  150.
         C1954 LBCOT-2000.
         C1954 LBRIC-2000.
NEXT B,55
         C1955 SOIL1 1500.
         C1955 SOIL2  500.
         C1955 FERT  2000.
         C1955 LOTCT  950.
         C1955 LOTRC  125.
```


(Listing Sheet 13)

```
                     LBCOT 2000.
                     LBRIC 2000.
NEXT B,56
               C1956 SOIL1 1500.
               C1956 SOIL2  500.
               C1956 FERT  2000.
               C1956 LOTCT  925.
               C1956 LOTRC  115.
                     LBCOT 2000.
                     LBRIC 2000.
NEXT B,57
               C1957 SOIL1 1500.
               C1957 SOIL2  500.
               C1957 FERT  2000.
               C1957 LOTCT  920.
               C1957 LOTRC  110.
                     LBCOT 2000.
                     LBRIC 2000.
NEXT B,58
               C1958 SOIL1 1500.
               C1958 SOIL2  500.
               C1958 FERT  2000.
               C1958 LOTCT  900.
               C1958 LOTRC  100.
                     LBCOT 2000.
                     LBRIC 2000.
NEXT B,59
               C1959 SOIL1 1500.
               C1959 SOIL2  500.
               C1959 FERT  2000.
               C1959 LOTCT  925.
               C1959 LOTRC  130.
                     LBCOT 2000.
                     LBRIC 2000.
EOF
```

c. The Agenda

The Agenda for this example consists of the following eight Agendum control cards

```
1           13          26          41
HEADING     USDA LP/90  EXAMPLE 5   LP MATRIX  GENERATION
INPUT,23,,COUNTS,MATRIXG,75
GETOFF,A8
INPUT,54,,COUNTS,MATRIXT
TRANSFORM,A8
PICTURE
A.TO.I,B9,REWIND
OUTPUT.
```

The second card, INPUT,23, etc., causes the following tasks to be performed. (1) Read the first file from the I-tape containing the G Matrix. (2) Convert it to binary. (3) Place it on the A-tape. The following GETOFF,A8, causes the G Matrix, now in binary and initialized properly, to be stored on tape unit A8. The succeeding card, INPUT,54, etc., causes the second file of the I-tape containing the T Matrix to be read, and to be converted to binary and stored on the A-tape. It thus replaces the G Matrix on the A-tape.

TRANSFORM,.A8 now takes the T Matrix on the A-tape, multiplies it times the G Matrix on the A8 tape, and produces the H Matrix which replaces the T Matrix on the A-tape. After TRANSFORM, the A-tape and the computer's core memory are in precisely the condition that would result if the H Matrix had been read in from the I-tape by INPUT rather than having been created by TRANSFORM. Thus, at this point a NORMAL or any other LP or LP-related Agendum could be applied using the H Matrix as input. In the interest of simplicity we have not done this. Instead, we introduce two PROP Agendum routines. The first is PICTURE, and the second, A.TO.1.

PICTURE prepares a schematic or stylized representation of the H Matrix (or any LP matrix created by INPUT or TRANSFORM on the A-tape) which is quite useful for model debugging, just as is the COUNTS option of INPUT.

A.TO.I,B9, REWIND converts the binary A-tape to BCD or I-tape format. This can be printed and, in this case, gives us in I-tape format the results of the TRANSFORM computation.

The MATRIXG and MATRIXT options of INPUT and TRANSFORM are USDA Agenda. PICTURE and A.TO.I are PROP Agenda. The rest belong to the SHARE system.

d. Output

The Output consists of two parts: (1) the result of PICTURE, (2) the result of A.TO.I. The former is found on the O-tape after OUTPUT. The latter is found by listing off-line the tape on unit B9 at the completion of the run.

The results of PICTURE come first on the O-tape listing. Then comes the output in the order of the Agendum cards. In Table 43, the results of PICTURE are shown. Column names are listed on the top of the matrix with letters printed vertically. (The column 1COT11 appears as

1
C
O
T
1
1

and so forth.)

This schematic picture of the matrix reveals whether all desired coefficients got computed, or carried over to the H Matrix. It also shows the numbers in decimal magnitude so that the very common errors of misplaced decimal points are quickly revealed.

Table 44 presents a listing of the I-tape (on B9) produced by A.TO.I. It is simply Table 42 in I-tape format.

TABLE 42

H-MATRIX CREATED BY TRANSFORM

Row Symbol	H-Matrix Column Symbol											
	1COT11	1COT12	1COT21	1COT22	2COT11	2COT12	2COT21	2OAT22	1OAT11	1OAT12	1OAT21	1OAT22
P1950	- 99.52	-248.77	- 94.75	-141.26	- 96.19	-250.65	- 94.80	-143.18	- 6.51	- 45.56	- 3.66	- 37.67
P1951	- 82.63	-215.12	- 78.82	-118.46	- 84.42	-230.60	- 85.12	-129.58	- 6.72	- 45.56	- 3.76	- 37.43
P1952	- 73.10	-191.09	- 69.46	-103.83	- 75.49	-207.14	- 75.74	-114.96	- 6.86	- 44.99	- 3.79	- 36.82
P1953	- 65.23	-171.02	- 61.78	- 91.71	- 68.90	-188.79	- 68.84	-103.80	- 5.57	- 39.06	- 2.64	- 31.49
P1954	- 72.57	-186.66	- 68.79	-102.02	- 74.30	-199.67	- 73.68	-110.94	- 4.83	- 35.95	- 2.04	- 28.82
P1955	- 64.16	-168.83	- 60.83	- 91.22	- 66.90	-184.49	- 66.84	-101.85	- 3.84	- 30.14	- 1.21	- 23.75
P1956	- 60.00	-159.98	- 56.53	- 85.40	- 66.25	-182.43	- 65.84	-100.48	- 3.12	- 27.45	- 0.50	- 21.29
P1957	- 43.67	-123.77	- 40.90	- 63.26	- 51.74	-150.78	- 51.99	- 81.08	- 1.84	- 25.50	- 0.82	- 19.47
P1958	- 37.41	-110.86	- 35.46	- 55.08	- 43.18	-135.47	- 45.10	- 71.40	- 2.98	- 29.65	- 0.27	- 23.19
1COTLT	3.10	7.23	2.96	4.43	2.79	6.51	2.66	3.99				
1COTSD	5.29	12.32	5.04	7.56	4.67	11.09	4.54	6.80				
1OATYD									14.40	65.10	12.90	58.40
1SBGYD												
1ALFYD												
1RICYD												
1SKLAB	8.30	10.50	8.20	9.80	13.34	16.40	13.04	14.40	3.53	5.90	5.00	7.10
1UNSLB	87.00	165.00	86.00	113.00	16.60	16.60	12.80	12.80	0.50	2.40	0.50	2.20
1MTRAC												
1LTRAC	6.30	6.70	6.20	6.60	6.80	7.20	6.70	7.10	2.50	2.90	3.50	3.90
1MMACH												
1LMACH	6.30	6.30	6.20	6.60	6.80	7.20	6.70	7.10	2.50	2.90	3.50	3.90
1PIKER					2.24	2.24	2.24	2.24				
1COMB									0.40	0.40	0.40	0.40
1SDCOT	45.00	45.00	33.00	33.00	90.00	90.00	60.00	60.00				
SDOAT									60.00	60.00	60.00	60.00

TABLE 42--Continued

Row Symbol	H-Matrix Column Symbol											
	1COT11	1COT12	1COT21	1COT22	2COT11	2COT12	2COT21	2COT22	1OAT11	1OAT12	1OAT21	1OAT22
1SDSBG												
1SDALF												
1SDRIC												
1NITRO		4.00		4.00		4.00		4.00		4.00		4.00
1DRY												
1GIN	3.10	7.23	2.96	4.43	2.79	6.51	2.66	3.99				
1WATER												
2COTLT	- 3.10	- 7.23	- 2.96	- 4.43	- 2.79	- 6.51	- 2.66	- 3.99				
2COTSD	- 5.29	- 12.32	- 5.04	- 7.56	- 4.67	- 11.09	- 4.54	- 6.80				
2OATYD									- 14.40	- 65.10	- 12.90	- 58.40
2SBGYD												
2ALFYD												
2RICYD												
2SKLAB	- 8.30	- 10.50	- 8.20	- 9.80	- 13.34	- 16.14	- 13.04	- 14.44	- 3.53	- 5.90	- 5.00	- 7.10
2UNSLB	- 87.00	-165.00	- 86.00	-113.00	- 16.60	- 16.60	- 12.80	- 12.80	- 0.50	- 2.40	- 0.50	- 2.20
2MTRAC												
2LTRAC	- 6.30	- 6.70	- 6.20	- 6.60	- 6.80	- 7.20	- 6.70	- 7.10	- 2.50	- 2.90	- 3.50	- 3.90
2MMACH												
2LMACH	- 6.30	- 6.30	- 6.20	- 6.60	- 6.80	- 7.20	- 6.70	- 7.10	- 2.50	- 2.90	- 3.50	- 3.90
2PIKER					- 2.24	- 2.24	- 2.24	- 2.24				
2COMB									- 0.40	- 0.40	- 0.40	- 0.40
2SDCOT	- 45.00	- 45.00	- 33.00	- 33.00	- 90.00	- 90.00	- 60.00	- 60.00				
2SDOAT									- 60.00	- 60.00	- 60.00	- 60.00

TABLE 42--Continued

Row Symbol	H-Matrix Column Symbol										
	COT11	1COT12	1COT21	1COT22	2COT11	2COT12	2COT21	2COT22	1OAT11	1OAT12	1OAT21	1OAT22
2SDSBG												
2SDALF												
2SDRIC												
2NITRO		- 4.00		- 4.00		- 4.00		- 4.00		- 4.00		- 4.00
2DRY												
2GIN	- 3.10	- 7.23	- 2.96	- 4.43	- 2.79	- 6.51	- 2.66	- 3.99				
2WATER												
SOIL1	1.00	1.00			1.00	1.00			1.00	1.00		
SOIL2			1.00	1.00			1.00	1.00			1.00	1.00
LABP4	5.80	13.50	5.50	8.30								
FERT		4.00		4.00		4.00		4.00		4.00		4.00
LOTCT	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00				
LOTRC												
UBCOT	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00				
UBOAT									1.00	1.00	1.00	1.00
UBSBG												
UBALF												
UBRIC												
LBCOT	- 1.00	- 1.00	- 1.00	- 1.00	- 1.00	- 1.00	- 1.00	- 1.00				
LBOAT									- 1.00	- 1.00	- 1.00	- 1.00
LBSBG												
LBALF												
LBRIC												
S2COT					2.79	6.51	2.73	4.89				

TABLE 42--Continued

Row Symbol	H-Matrix Column Symbol					Con- straint Type	Right-Hand-Sides	
	1SBG11	1SBG21	1ALF11	1RIC21	1RIC22		FIRST B	NEXT B,51
P1950	- 54.58	- 26.37	- 38.23	- 57.72	-143.30			
P1951	- 64.68	- 31.84	- 41.84	- 50.86	-129.15			
P1952	- 68.03	- 33.25	- 44.10	- 65.63	-163.52			
P1953	- 63.99	- 30.89	- 36.60	- 49.72	-129.62			
P1954	- 60.72	- 29.35	- 36.93	- 44.46	-117.24			
P1955	- 49.65	- 22.60	- 32.13	- 52.77	-135.08			
P1956	- 51.80	- 24.22	- 26.00	- 51.83	-133.87			
P1957	- 48.31	- 21.95	- 27.64	- 55.95	-143.10			
P1958	- 45.56	- 20.22	- 29.87	- 54.71	-139.95			
1COTLT								
1COTSD								
1OATYD								
1SBGYD	29.00	17.00						
1ALFYD			2.07					
1RICYD				17.60	35.60			
1SKLAB	4.70	6.20	9.90	15.57	17.90			
1UNSLB	1.10	1.10	2.80					
1MTRAC			6.40					
1LTRAC	3.00	4.40	0.70	6.20	7.70			
1MMACH			9.20					
1LMACH	3.00	4.40	0.70	6.20	7.70			
1PIKER								
1COMB	0.40	0.40		0.50	0.50			
1SDCOT								
1SDOAT								

TABLE 42 --Continued

Row Symbol	H-Matrix Column Symbol					Con-straint Type	Right-Hand-Sides	
	1SBG11	1SBG21	1ALF11	1RIC21	1RIC22		FIRST B	NEXT B,51
1SDSBG	60.00	60.00						
1SDALF			15.00					
1SDRIC				100.00	100.00			
1NITRO								
1DRY				17.60	35.60			
1GIN								
1WATER				1.00	1.00			
2COTLT								
2COTSD								
2OATYD								
2SBGYD	- 29.00	- 17.00						
2ALFYD			- 2.07					
2RICYD				- 17.60	- 35.60			
2SKLAB	- 4.70	- 6.20	- 9.90	- 15.57	- 17.90			
2UNSLB	- 1.10	- 1.10	- 2.80					
2MTRAC			- 6.40					
2LTRAC	- 3.00	- 4.40	- 0.70	- 6.20	- 7.70			
2MMACH			- 9.20					
2LMACH	- 3.00	- 4.40	- 0.70	- 6.20	- 7.70			
2PIKER								
2COMB	- 0.40	- 0.40		- 0.50	- 0.50			
2SDCOT								
2SDOAT								

TABLE 42--Continued

Row Symbol	H-Matrix Column Symbol					Con- straint Type	Right-Hand-Sides	
	1SBG11	1SBG21	1ALF11	1RIC21	1RIC22		FIRST B	NEXT B,51
2SDSBG	- 60.00	- 60.00						
2SDALF			- 15.00					
2SDRIC				-100.00	-100.00			
2NITRO								
2DRY				- 17.60	- 35.60			
2GIN								
2WATER				- 1.00	- 1.00			
SOIL1	1.00		1.00			≤		1500
SOIL2		1.00		1.00	1.00	≤		500
LABP4	1.10	1.10	0.70			≤		9900
FERT					4.00	≤		1525
LOTCT						≤		700
LOTRC				1.00	1.00	≤		5000
UBCOT						≤		
UBOAT						≤		
UBSBG	1.00	1.00				≤		
UBALF			1.00			≤		
UBRIC				1.00	1.00	≤		
LBCOT						≤		
LBOAT						≤		
LBSBG	- 1.00	- 1.00				≤		
LBALF			- 1.00			≤		
LBRIC				- 1.00	- 1.00	≤		
S2COT						≤		

Table 43: OUTPUT FROM PICTURE AGENDUM

(Listing Sheet 1)

1LMACH

LESS THAN OR = TO	SYMBOL	GREATER THAN
9999	W	1000
1000	Z	100
100	Y	10
10	X	1

LESS THAN	SYMBOL	GREATER THAN OR = TO
1	A	.1
.1	B	.01
.01	C	.001
.001	D	.0001
.0001	E	.00001
.00001	F	.000001

Table 43: OUTPUT FROM PICTURE AGENDUM--Continued

(Listing Sheet 2)

```
       USDA LP/90   EXAMPLE  5     LP MATRIX GENERATION
                                                                                   (    1,    1)
                                           + + + + + + + + + + + + + + + + +    B
         1 1 1 1 2 2 2 2 1 1 1 1 1 1 1 1 1
         C C C C C C C C O O O O S S A R R S S L F L L U U U J J L L L L L S
         O O O O O O O O A A A A B B L I I O O A E O O B B B B B B B B B 2
         T T T T T T T T T T T T G G F C C I I B R T T C O S A R C O S A R C
         1 1 2 2 1 1 2 2 1 1 2 2 1 2 1 2 2 L L P T C R O A B L I O A B L I O
         1 2 1 2 1 2 1 2 1 2 1 2 1 1 1 1 2 1 2 4   T C T T G F C T T G F C T    1

 P1950  -Y-Z-Y-Z-Y-Z-Y-Z-X-Y-X-Y-Y-Y-Y-Y-Z
 P1951  -Y-Z-Y-Z-Y-Z-Y-Z-X-Y-X-Y-Y-Y-Y-Y-Z
 P1952  -Y-Z-Y-Z-Y-Z-Y-Z-X-Y-X-Y-Y-Y-Y-Y-Z
 P1953  -Y-Z-Y-Y-Y-Z-Y-Z-X-Y-X-Y-Y-Y-Y-Y-Z
 P1954  -Y-Z-Y-Z-Y-Z-Y-Z-X-Y-X-Y-Y-Y-Y-Y-Z
 P1955  -Y-Z-Y-Y-Y-Z-Y-Z-X-Y-X-Y-Y-Y-Y-Y-Z
 P1956  -Y-Z-Y-Y-Y-Z-Y-Z-X-Y-A-Y-Y-Y-Y-Y-Z
 P1957  -Y-Z-Y-Y-Y-Z-Y-Y-X-Y A-Y-Y-Y-Y-Y-Z
 P1958  -Y-Z-Y-Y-Y-Z-Y-Y-X-Y-A-Y-Y-Y-Y-Y-Z
1COTLT   X X X X X X X X
1COTSD
1DATYD                   Y Y Y Y
1SBGYD                           Y Y
1ALFYD                               X
1RICYD                                 Y Y
1SKLAB   X Y X X Y Y Y Y X X X X X X X Y Y
1UNSLB
1MTRAC
1LTRAC
1MMACH
1LMACH
1PIKER
1COMB
1SDCOT
1SDDAT
1SDSBG
1SDALF
1SDRIC
1NITRO
1DRY
1GIN
1WATER
2COTLT  -X-X-X-X-X-X-X-X
2COTSD
2DATYD                  -Y-Y-Y-Y
2SBGYD                          -Y-Y
2ALFYD                              -X
2RICYD                                -Y-Y
2SKLAB  -X-Y-X-X-Y-Y-Y-Y-X-X-X-X-X-X-X-Y-Y
2UNSLB
2MTRAC
2LTRAC
2MMACH
2LMACH
2PIKER
2COMB
2SDCOT
2SDDAT
2SDSBG
```

(Listing Sheet 3)

USDA LP/90 EXAMPLE 5 LP MATRIX GENERATION

2, B (1)

	1COT11	1COT12	1COT21	1COT22	2COT11	2COT12	2COT21	2COT22	1OAT11	1OAT12	1OAT21	1OAT22	1SBG11	1SBG21	1ALF11	1RIC21	1RIC22	+ SOIL1	+ SOIL2	+ LABP4	+ FERT	+ LOTCT	+ LOTRC	+ UBCOT	+ UBOAT	+ UBSBG	+ UBALF	+ UBRIC	+ LBCOT	+ LBOAT	+ LBSBG	+ LBALF	+ LBRIC	+ S2COT	B
																																			1
2SDALF																																			
2SDRIC																																			
2NITRO																																			
2DRY																																			
2GIN																																			
2WATER																																			
+ SOIL1	1	1			1	1			1	1			1		1			1																	
+ SOIL2			1	1			1	1			1	1		1		1	1		1																
+ LABP4	X	Y	X	X									X	X	A					1															
+ FERT		X		X		X		X		X		X					X				1														
+ LOTCT	1	1	1	1	1	1	1	1														1													
+ LOTRC																1	1						1												
+ UBCOT	1	1	1	1	1	1	1	1																1											
+ UBOAT									1	1	1	1													1										
+ UBSBG													1	1												1									
+ UBALF															1												1								
+ UBRIC																1	1											1							
+ LBCOT	-1	-1	-1	-1	-1	-1	-1	-1																					1						
+ LBOAT									-1	-1	-1	-1																		1					
+ LBSBG													-1	-1																	1				
+ LBALF															-1																	1			
+ LBRIC																-1	-1																1		
+ S2COT					X	X	X	X																										1	

Table 44: New I-Tape Listing H-Matrix Output From A. to I. Agendum

(Listing Sheet 1)

```
*     USDA LP/90    EXAMPLE  5     LP MATRIX GENERATICN
ROW IC
   5      P1950        P1951       P1952       P1953       P1954
   5      P1955        P1956       P1957       P1958       1CCTLT
   5     1COTSD       1CATYD      1SBGYD      1ALFYC      1RICYC
   5     1SKLAB       1UNSLB      1MTRAC      1LTRAC      1MMACH
   5     1LMACH       1PIKER      1COMB       1SDCCT      1SCOAT
   5     1SDSBG       1SDALF      1SDRIC      1NITRC      1CRY
   5     1GIN         1WATER      2COTLT      2CCTSC      2CATYC
   5     2SBGYD       2ALFYD      2RICYC      2SKLAB      2UNSLB
   5     2MTRAC       2LTRAC      2MMACH      2LMACH      2PIKER
   5     2COMB        2SDCOT      2SDOAT      2SDSBG      2SDALF
   5     2SDRIC       2NITRO      2CRY        2GIN        2WATER
   5   + SOIL1      + SOIL2     + LABP4     + FERT      + LOTCT
   5   + LOTRC      + LBCOT     + UBOAT     + UBSBG     + UBALF
   5   + UBRIC      + LBCOT     + LBOAT     + LBSBG     + LBALF
   2   + LBRIC      + S2COT
MATRIX
      1COT11 P1950-   99.5217
      1COT11 P1951-   82.6341
      1COT11 P1952-   73.1C42
      1COT11 P1953-   65.2261
      1COT11 P1954-   72.5749
      1CCT11 P1955-   64.1563
      1COT11 P1956-   6C.0C41
      1COT11 P1957-   43.6708
      1COT11 P1958-   37.4111
      1COT111COTLT     3.1
      1COT111SKLAB     8.3
      1COT112COTLT-    3.1
      1COT112SKLAB-    8.3
      1COT11 SOIL1     1.
      1COT11 LABP4     5.8
      1COT11 LOTCT     1.
      1COT11 UBCOT     1.
      1COT11 LBCOT-    1.
      1COT12 P1950-  248.7684
      1COT12 P1951-  215.1191
      1COT12 P1952-  191.0872
      1COT12 P1953-  171.0224
      1COT12 P1954-  186.6632
      1COT12 P1955-  168.8309
      1COT12 P1956-  159.9855
      1COT12 P1957-  123.7685
      1COT12 P1958-  11C.8617
      1COT121COTLT     7.23
      1COT121SKLAB    10.5
      1COT122COTLT-    7.23
      1COT122SKLAB-   10.5
      1COT12 SOIL1     1.
      1COT12 LABP4    13.5
      1COT12 FERT      4.
      1COT12 LOTCT     1.
      1COT12 UBCCT     1.
      1COT12 LBCCT-    1.
      1COT21 P1950-   94.75C8
      1COT21 P1951-   78.8252
      1COT21 P1952-   69.4604
```

(Listing Sheet 2)

```
1COT21 P1953-   61.7788
1COT21 P1954-   68.7884
1CGT21 P1955-   60.8288
1COT21 P1956-   56.534
1CCT21 P1957-   40.896
1COT21 P1958-   35.4564
1CCT211COTLT     2.96
1COT211SKLAB     8.2
1COT212COTLT-    2.96
1COT212SKLAB-    8.2
1COT21 SOIL2     1.
1COT21 LABP4     5.5
1COT21 LOTCT     1.
1COT21 UBCOT     1.
1COT21 LBCOT-    1.
1COT22 P1950-  141.2656
1COT22 P1951-  118.4637
1CCT22 P1952-  103.8284
1COT22 P1953-   91.714
1COT22 P1954-  102.0246
1COT22 P1955-   91.2167
1COT22 P1956-   85.4C51
1CCT22 P1957-   63.2623
1COT22 P1958-   55.0763
1CCT221COTLT     4.43
1CCT221SKLAB     9.8
1COT222COTLT-    4.43
1COT222SKLAB-    9.8
1COT22 SOIL2     1.
1CCT22 LABP4     8.3
1COT22 FERT      4.
1COT22 LOTCT     1.
1COT22 UBCOT     1.
1COT22 LBCOT-    1.
2COT11 P1950-   96.1925
2COT11 P1951-   84.4206
2COT11 P1952-   75.4936
2CCT11 P1953-   68.9C47
2COT11 P1954-   74.3C31
2COT11 P1955-   66.8998
2COT11 P1956-   66.2536
2COT11 P1957-   51.7369
2COT11 P1958-   43.1828
2COT111COTLT     2.79
2CCT111SKLAB    13.34
2COT112COTLT-    2.79
2COT112SKLAB-   13.34
2COT11 SOIL1     1.
2CCT11 LOTCT     1.
2CCT11 UBCOT     1.
2COT11 LBCOT-    1.
2COT11 S2COT     2.79
2CCT12 P1950-  250.6503
2CCT12 P1951-  230.5996
2COT12 P1952-  207.1396
2COT12 P1953-  188.7889
2COT12 P1954-  199.6693
2COT12 P1955-  184.4872
```

(Listing Sheet 3)

```
2COT12 P1956- 182.4322
2COT12 P1957- 150.7757
2COT12 P1958- 135.47C2
2COT121COTLT    6.51
2COT121SKLAB   16.14
2COT122COTLT-   6.51
2COT122SKLAB-  16.14
2COT12 SOIL1    1.
2COT12 FERT     4.
2COT12 LOTCT    1.
2COT12 UBCOT    1.
2COT12 LBCOT-   1.
2COT12 S2COT    6.51
2COT21 P1950-  94.8C48
2COT21 P1951-  85.1156
2COT21 P1952-  75.7446
2COT21 P1953-  68.8394
2COT21 P1954-  73.6828
2COT21 P1955-  66.8392
2COT21 P1956-  65.8412
2COT21 P1957-  51.9942
2COT21 P1958-  45.1CC2
2COT211COTLT    2.66
2COT211SKLAB   13.04
2COT212COTLT-   2.66
2COT212SKLAB-  13.04
2COT21 SOIL2    1.
2COT21 LOTCT    1.
2COT21 UBCOT    1.
2COT21 LBCOT-   1.
2COT21 S2COT    2.73
2COT22 P1950- 143.1854
2COT22 P1951- 129.5853
2COT22 P1952- 114.955
2COT22 P1953- 103.8C04
2COT22 P1954- 11C.9364
2COT22 P1955- 101.8539
2COT22 P1956- 1C0.48C3
2COT22 P1957-  81.0795
2COT22 P1958-  71.4C25
2COT221COTLT    3.99
2COT221SKLAB   14.44
2COT222COTLT-   3.99
2COT222SKLAB-  14.44
2COT22 SOIL2    1.
2COT22 FERT     4.
2COT22 LOTCT    1.
2COT22 UBCOT    1.
2COT22 LBCOT-   1.
2COT22 S2COT    4.89
1OAT11 P1950-   6.5109
1OAT11 P1951-   6.7193
1OAT11 P1952-   6.8579
1OAT11 P1953-   5.5692
1OAT11 P1954-   4.8324
1OAT11 P1955-   3.8411
1OAT11 P1956-   3.1226
1OAT11 P1957-   1.8363
```

(Listing Sheet 4)

```
1OAT11 P1958-   2.9836
1OAT111OATYD   14.4
1OAT111SKLAB    3.53
1OAT112OATYD-  14.4
1OAT112SKLAB-   3.53
1OAT11 SOIL1    1.
1OAT11 UBOAT    1.
1OAT11 LBOAT-   1.
1OAT12 P1950-  45.56
1OAT12 P1951-  45.56
1OAT12 P1952-  44.996
1OAT12 P1953-  39.063
1OAT12 P1954-  35.952
1OAT12 P1955-  30.145
1OAT12 P1956-  27.452
1OAT12 P1957-  25.498
1OAT12 P1958-  29.649
1OAT121OATYD   65.1
1OAT121SKLAB    5.9
1OAT122OATYD-  65.1
1OAT122SKLAB-   5.9
1OAT12 SOIL1    1.
1OAT12 FERT     4.
1OAT12 UBOAT    1.
1OAT12 LBOAT-   1.
1OAT21 P1950-   3.657
1OAT21 P1951-   3.761
1OAT21 P1952-   3.787
1OAT21 P1953-   2.638
1OAT21 P1954-   2.04
1OAT21 P1955-   1.2C9
1OAT21 P1956-    .497
1OAT21 P1957     .819
1OAT21 P1958-    .268
1OAT211OATYD   12.9
1OAT211SKLAB    5.
1OAT212OATYD-  12.9
1OAT212SKLAB-   5.
1OAT21 SOIL2    1.
1OAT21 UBOAT    1.
1OAT21 LBOAT-   1.
1OAT22 P1950-  37.666
1OAT22 P1951-  37.431
1OAT22 P1952-  36.824
1OAT22 P1953-  31.494
1OAT22 P1954-  28.817
1OAT22 P1955-  23.747
1OAT22 P1956-  21.29
1OAT22 P1957-  19.467
1OAT22 P1958-  23.189
1OAT221OATYD   58.4
1OAT221SKLAB    7.1
1OAT222OATYD-  58.4
1OAT222SKLAB-   7.1
1OAT22 SOIL2    1.
1OAT22 FERT     4.
1OAT22 UBOAT    1.
1OAT22 LBOAT-   1.
```

Table 44: New I-Tape Listing H-Matrix Output From A. to I. Agendum--Continued

(Listing Sheet 5)

```
1SBG11 P1950-   54.583
1SBG11 P1951-   64.681
1SBG11 P1952-   68.C27
1SBG11 P1953-   63.987
1SBG11 P1954-   60.714
1SBG11 P1955-   49.654
1SBG11 P1956-   51.799
1SBG11 P1957-   48.3C7
1SBG11 P1958-   45.559
1SBG111SBGYD    29.
1SBG111SKLAB     4.7
1SBG112SBGYD-   29.
1SBG112SKLAB-    4.7
1SBG11 SOIL1     1.
1SBG11 LABP4     1.1
1SBG11 UBSBG     1.
1SBG11 LBSBG-    1.
1SBG21 P1950-   26.374
1SBG21 P1951-   31.84
1SBG21 P1952-   33.252
1SBG21 P1953-   30.893
1SBG21 P1954-   29.346
1SBG21 P1955-   22.6C5
1SBG21 P1956-   24.223
1SBG21 P1957-   21.952
1SBG21 P1958-   20.221
1SBG211SBGYD    17.
1SBG211SKLAB     6.2
1SBG212SBGYD-   17.
1SBG212SKLAB-    6.2
1SBG21 SOIL2     1.
1SBG21 LABP4     1.1
1SBG21 UBSBG     1.
1SBG21 LBSBG-    1.
1ALF11 P1950-   38.2338
1ALF11 P1951-   41.8372
1ALF11 P1952-   44.1C18
1ALF11 P1953-   36.5998
1ALF11 P1954-   36.933
1ALF11 P1955-   32.131
1ALF11 P1956-   26.CC38
1ALF11 P1957-   27.6374
1ALF11 P1958-   29.8748
1ALF111ALFYD     2.07
1ALF111SKLAB     9.9
1ALF112ALFYD-    2.07
1ALF112SKLAB-    9.9
1ALF11 SOIL1     1.
1ALF11 LABP4      .7
1ALF11 UBALF     1.
1ALF11 LBALF-    1.
1RIC21 P1950-   57.7191
1RIC21 P1951-   50.8567
1RIC21 P1952-   65.6341
1RIC21 P1953-   49.7178
1RIC21 P1954-   44.4636
1RIC21 P1955-   52.7679
1RIC21 P1956-   51.8324
```


(Listing Sheet 6)

```
           1RIC21 P1957-   55.9487
           1RIC21 P1958-   54.7144
           1RIC211RICYD    17.6
           1RIC211SKLAB    15.57
           1RIC212RICYD-   17.6
           1RIC212SKLAB-   15.57
           1RIC21 SOIL2     1.
           1RIC21 LOTRC     1.
           1RIC21 UBRIC     1.
           1RIC21 LBRIC-    1.
           1RIC22 P1950-  143.3C2
           1RIC22 P1951-  129.148
           1RIC22 P1952-  163.524
           1RIC22 P1953-  129.616
           1RIC22 P1954-  117.24
           1RIC22 P1955-  135.076
           1RIC22 P1956-  133.874
           1RIC22 P1957-  143.097
           1RIC22 P1958-  139.951
           1RIC221RICYD    35.6
           1RIC221SKLAB    17.9
           1RIC222RICYD-   35.6
           1RIC222SKLAB-   17.9
           1RIC22 SOIL2     1.
           1RIC22 FERT      4.
           1RIC22 LOTRC     1.
           1RIC22 UBRIC     1.
           1RIC22 LBRIC-    1.
SLACK             SOIL1
SLACK             SOIL2
SLACK             LABP4
SLACK             FERT
SLACK             LOTCT
SLACK             LOTRC
SLACK             UBCOT
SLACK             UBOAT
SLACK             UBSBG
SLACK             UBALF
SLACK             UBRIC
SLACK             LBCOT
SLACK             LBOAT
SLACK             LBSBG
SLACK             LBALF
SLACK             LBRIC
SLACK             S2COT
FIRST B
NEXT B,00C50
           B050   SOIL1 1500.
           B050   SOIL2  500.
           B050   LABP4 9900.
           B050   FERT  1525.
           B050   LOTCT  700.
           B050   LOTRC 5000.
EOF
```

6. RECURSIVE LINEAR PROGRAMMING

a. Purpose

A recursive linear programming model (RLP) consists of a sequence of recursively dependent linear programming problems. "Recursive dependence" means that for a given LP problem the right-hand-side elements and/or the objective function parameters, and/or the active constraint matrix coefficients depend, in a well defined way, upon the solution of the preceding LP problem or problems. The purpose of this sixth and final example is to illustrate how a special class of RLP models may be solved automatically by USDA-LP/90.

This class of RLP models has the following structure. The objective function elements are "exogenous," that is, they are pre-determined data, not dependent on the preceding problem's solution. The matrix elements are constant over time. Finally, the right-hand-side elements are found by multiplying the preceding problem's solution by a change matrix and adding to the resulting vector a dummy right-hand-side vector of exogenous right-hand-side elements. It is the Type I RLP discussed in Part II above.

As with any dynamic system, the progress of the RLP model from one period to the next depends upon initial conditions. Two options are available for introducing initial conditions for an RLP run. This example discusses one such option, the introduction of an initial "solution" vector. An alternative, the introduction of an initial right-hand-side is discussed in sub-section e below.

The RLP features contain two options for the "change matrix" used to compute the right-hand-sides of succeeding problems. A general change matrix may be used directly (Type II RLP) or, if the change matrix has a very special form, it may be computed automatically (Type I, RLP). The latter option is illustrated here.

The example is based on the algebraic discussion of Part II, Section 6. The reader may wish to review that text before continuing with the discussion of this example.

b. Data

The input data for this problem consist of (1) the LP matrix (H Matrix) created in Example 5 (note that it contains nine objective function rows!); (2) a diagonal matrix (Γ matrix) of change coefficients for use in creating the change matrix; (3) an initial "solutiou" vector for the based year, after which production decisions are to be forecast or explained; (4) the vectors of exogenously determined right-hand-side elements.

The first item is shown in Table 42 above and a description of it need not be repeated here. We merely remind ourselves that its nine objective function rows are based on nine successive years' prices. For example, the P1950 row gives the net returns for each process using 1950 prices, and so on.

The second item (Γ matrix) shown in Table 45 below, is multiplied times the active constraint portion of the LP matrix (the "B" matrix) to create the change matrix (Φ matrix). The first six rows of the Γ matrix are all zero, for these are exogenously given rows. That is, the amounts of both soil types, of labor and fertilizer, and the magnitudes of the cotton and rice allotments are exogenously determined. They are not related to the preceding year's solution. The next rows give the "upper flexibility coefficients (plus one)" for the total acreage of the five crops, while the next five rows give the "lower flexibility coefficients" (subtracted from one). Finally the last row gives the "investment coefficient" (plus one) for the cotton stage II processes.

The initial conditions are also shown in Table 45. They give the estimated actual acreage planted for the base year, which is, in this case, 1950. This vector will enable us to compute the right-hand-side for the first LP problem in the sequence.

Finally, the exogenous vectors are shown in Table 46. These are placed on the change matrix as dummy right-hand-sides (we will see how this is done below) and are added to the vector which results from multiplying the change matrix times the preceding solution. In this way the exogenously determined RHS elements are created for a given year's LP problem. The negative entries in the LBCOT and LBRIC rows are designed to prevent an infeasibility from occurring during years with acreage allotments.

Table 45 : Diagonal Elements of Γ-Matrix And Initial Process Levels

Row and Column Symbol	Element Value	Column Symbol	Process Level
SOIL1	.00	1COT11	0.00
SOIL2	.00	1COT12	354.37
LABP4	.95	1COT21	318.75
FERT	.75	1COT22	0.00
LOTCT	.00	2COT11	0.00
LOTRC	.00	2COT12	26.88
UBCOT	1.10	2COT21	0.00
UBOAT	1.30	2COT22	0.00
UBSGB	1.30	1OAT11	238.75
UBALF	1.15	1OAT12	0.00
UBRIC	1.50	1OAT21	101.25
LBCOT	.80	1OAT22	0.00
LBOAT	.74	1SBG11	650.00
LBSBG	.74	1SBG21	0.00
LBALF	.76	1ALF11	230.00
LBRIC	.90	1RIC21	80.00
S2COT	1.25	1RIC22	0.00

Table 46 : Exogenous RHS-Vectors

Row Symbol	Column Symbol								
	C1951	C1952	C1953	C1954	C1955	C1956	C1957	C1958	C1959
SOIL1	1500	1500	1500	1500	1500	1500	1500	1500	1500
SOIL2	500	500	500	500	500	500	500	500	500
LABP4									
FERT	2000	2000	2000	2000	2000	2000	2000	2000	2000
LOTCT	5000	5000	50000	1000	950	925	920	900	925
LOTRC	5000	5000	5000	150	125	115	110	100	130
UBCOT									
UBOAT									
UBSBG									
UBALF									
UBRIC									
LBCOT				-2000	-2000	-2000	-2000	-2000	-2000
LBOAT									
LBSBG									
LBALF									
LBRIC				-2000	-2000	-2000	-2000	-2000	-2000
S2COT									

c. The Agenda

The Agenda is shown in Table 47.

The first Agendum card is our familiar INPUT. It reads the BCD tape created by A.TO.I in Example 5. We could have used the binary A-tape created by TRANSFORM in Example 5 instead and in that way avoided using INPUT.

The second Agendum card is a new one designed especially for RLP runs with USDA LP/90. Its function is to read the change (Φ) matrix and the exogenous vectors. The change matrix is in LP format and must have exactly the same row ID as the basic LP matrix. In this case, the change matrix is a "dummy" one and the Agendum is used only to read the exogenous vectors. This is because the change matrix for this example is to be computed from the Γ-matrix and the B matrix portion of the LP matrix.

The third Agendum, GAMMAB, CR, (1) reads from the card reader (as designated by the CR) the Γ-matrix coefficients, (2) multiplies them times the "B" portion of the LP matrix and (3) stores the resulting change matrix (Φ) on the change matrix tape in place of the dummy one read in.

The Agendum GAMMAB,CR,INITIATE reads from the card reader the initial solution vector (initial process levels). The next Agendum, CREATE,51,51, INITIATE (1) multiplies the change matrix (Φ) times the initial solution vector just read in by the preceding Agendum, (2) adds the exogenous vector numbered 51 and, (3) stores the result as RHS number 51 on the A-tape or binary LP-matrix. The next Agendum routines are already familiar to us. In groups of three, they provide for (1) solving the LP problem based on the objective function for the preceding year, the RHS just created from the solution of the preceding year, and the exogenous RHS elements for the current year; (2) writing a report of the solution; (3) creating the new RHS for the succeeding problem. Finally, at the very end comes OUTPUT with which we are already familiar.

The REPORT Agendum uses the same report writing program compiled in Example 3. As our process names are precisely the same as in that example, and as the yield coefficients are unchanged, this usage is possible.

Table 47: AGENDA FOR RLP EXAMPLE

(Listing Sheet 1)

```
HEADING      USDA LP/90   EXAMPLE 6    RECURSIVE LINEAR PROGRAMMING
INPUT,55,,COUNTS,REWIND
READ,CHANGE MATRIX,55,2,COUNTS
GAMMAB,CR
       GAMMA LABP4      .95
       GAMMA FERT       .75
       GAMMA UBCOT     1.10
       GAMMA UBOAT     1.30
       GAMMA UBSBG     1.30
       GAMMA UBALF     1.15
       GAMMA UBRIC     1.50
       GAMMA LBCOT      .80
       GAMMA LBOAT      .74
       GAMMA LBSBG      .74
       GAMMA LBALF      .76
       GAMMA LBRIC      .90
       GAMMA S2COT     1.25
EOF
GAMMAB,CR,INITIATE
      1OAT21         101.25
      1COT12         354.368280
      1COT21         318.75
      1OAT11         238.75
      1SBG11         650.
      1ALF11         230.
      1RIC21          80.
      2COT12          26.881720
EOF
TITLE        USDA LP/90  EXAMPLE 6  RLP FOR 1951
CREATE,51,51,INITIATE
NORMAL,* P1950,51
REPORT,B8,1
TITLE        USDA LP/90  EXAMPLE 6  RLP FOR 1952
CREATE,52,52
NORMAL,* P1951,52
REPORT,B8,1
TITLE        USDA LP/90  EXAMPLE 6  RLP FOR 1953
CREATE,53,53
NORMAL,* P1952,53
REPORT,B8,1
TITLE        USDA LP/90  EXAMPLE 6  RLP FOR 1954
CREATE,54,54
NORMAL,* P1953,54
REPORT,B8,1
TITLE        USDA LP/90  EXAMPLE 6  RLP FOR 1955
CREATE,55,55
NORMAL,* P1954,55
REPORT,B8,1
TITLE        USDA LP/90  EXAMPLE 6  RLP FOR 1956
CREATE,56,56
NORMAL,* P1955,56
REPORT,B8,1
TITLE        USDA LP/90  EXAMPLE 6  RLP FOR 1957
CREATE,57,57
NORMAL,* P1956,57
REPORT,B8,1
TITLE        USDA LP/90  EXAMPLE 6  RLP FOR 1958
CREATE,58,58
NORMAL,* P1957,58
REPORT,B8,1
TITLE        USDA LP/90  EXAMPLE 6  RLP FOR 1959
```

(Listing Sheet 2)

```
CREATE,59,59
NORMAL,* P1958,59
REPORT,B8,1
OUTPUT
```

d. Output

The listing of the O-tape for this run contains first the tables prepared by the REPORT Agendum. These are shown in Table 48. After the reports comes the run history including the Agendum cards, the timing comments, and the full solution print-outs. As the reader is already familiar with the run history listing for preceding examples, this is omitted here.

However, to show the nature of the solution of an RLP model, several graphs are presented in Figures 15 through 17. These show respectively, the acreages of the five crops, the dual variables of the five crops and the program value for each year. The first two were taken from the tables computed by REPORT while the latter were taken directly from the full solution print-outs (not shown).

Table 48: OUTPUT FROM REPORT AGENDUM

(Listing Sheet 1)

USDA LP/90 EXAMPLE 6 RLP FOR 1951

OUTPUT SUMMARY

CROP /	PRODUCTION	ACREAGE	AVE YIELD
COTTON LINT	3,410,896	560,000	6.09
COTTON SEED	5,811,543	560,000	10.38
OATS	14,708,020	251,600	58.46
SOYBEANS	24,505,000	845,000	29.00
ALFALFA	462,438	223,400	2.07
RICE	4,272,000	120,000	35.60

FIELD CROP ACREAGE BY SOIL TYPE

CROP /	SOIL 1	SOIL 2	TOTAL
COTTON	416,274	143,726	560,000
OATS	15,326	236,274	251,600
SOYBEANS	845,000		845,000
ALFALFA	223,400		223,400
RICE		120,000	120,000

FIELD CROP ACREAGE BY FERTILIZER LEVEL

CROP /	LEVEL 1	LEVEL 2	TOTAL
COTTON	143,726	416,274	560,000
OATS	1,936	249,664	251,600
RICE		120,000	120,000

Table 48: OUTPUT FROM REPORT AGENDUM--Continued

(Listing Sheet 2)

USDA LP/90 EXAMPLE 6 RLP FOR 1952

OUTPUT SUMMARY

CROP /	PRODUCTION	ACREAGE	AVE YIELD
COTTON LINT	3,278,806	457,683	7.16
COTTON SEED	5,586,992	457,683	12.21
OATS	18,688,000	320,000	58.40
SOYBEANS	25,303,455	872,533	29.00
ALFALFA	351,453	169,784	2.07
RICE	6,408,000	180,000	35.60

FIELD CROP ACREAGE BY SOIL TYPE

CROP /	SOIL 1	SOIL 2	TOTAL
COTTON	457,683		457,683
OATS		320,000	320,000
SOYBEANS	872,533		872,533
ALFALFA	169,784		169,784
RICE		180,000	180,000

FIELD CROP ACREAGE BY FERTILIZER LEVEL

CROP /	LEVEL 1	LEVEL 2	TOTAL
COTTON		457,683	457,683
OATS		320,000	320,000
RICE		180,000	180,000

Table 48: OUTPUT FROM REPORT AGENDUM--Continued

(Listing Sheet 3)

USDA LP/90 EXAMPLE 6 RLP FOR 1953

OUTPUT SUMMARY

CROP /	PRODUCTION	ACREAGE	AVE YIELD
COTTON LINT	3,153,714	441,427	7.14
COTTON SEED	5,373,800	441,427	12.17
OATS	13,874,680	236,800	58.59
SOYBEANS	26,759,380	922,737	29.00
ALFALFA	267,104	129,036	2.07
RICE	9,612,000	270,000	35.60

FIELD CROP ACREAGE BY SOIL TYPE

CROP /	SOIL 1	SOIL 2	TOTAL
COTTON	441,427		441,427
OATS	6,800	230,000	236,800
SOYBEANS	922,737		922,737
ALFALFA	129,036		129,036
RICE		270,000	270,000

FIELD CROP ACREAGE BY FERTILIZER LEVEL

CROP /	LEVEL 1	LEVEL 2	TOTAL
COTTON		441,427	441,427
OATS		236,800	236,800
RICE		270,000	270,000

Table 48: OUTPUT FROM REPORT AGENDUM--Continued

(Listing Sheet 4)

USDA LP/90 EXAMPLE 6 RLP FOR 1954

OUTPUT SUMMARY

CROP /	PRODUCTION	ACREAGE	AVE YIELD
COTTON LINT	3,023,665	424,747	7.12
COTTON SEED	5,152,155	424,747	12.13
OATS	17,977,856	307,840	58.40
SOYBEANS	29,055,116	1,019,346	28.50
ALFALFA	202,999	98,067	2.07
RICE	5,340,000	150,000	35.60

FIELD CROP ACREAGE BY SOIL TYPE

CROP /	SOIL 1	SOIL 2	TOTAL
COTTON	424,747		424,747
OATS		307,840	307,840
SOYBEANS	977,186	42,160	1,019,346
ALFALFA	98,067		98,067
RICE		150,000	150,000

FIELD CROP ACREAGE BY FERTILIZER LEVEL

CROP /	LEVEL 1	LEVEL 2	TOTAL
COTTON		424,747	424,747
OATS		307,840	307,840
RICE		150,000	150,000

Table 48: OUTPUT FROM REPORT AGENDUM--Continued

(Listing Sheet 5)

USDA LP/90 EXAMPLE 6 RLP FOR 1955

OUTPUT SUMMARY

CROP /	PRODUCTION	ACREAGE	AVE YIELD
COTTON LINT	2,986,524	421,243	7.09
COTTON SEED	5,088,814	421,243	12.08
OATS	21,900,000	375,000	58.40
SOYBEANS	29,122,539	1,004,226	29.00
ALFALFA	154,279	74,531	2.07
RICE	4,450,000	125,000	35.60

FIELD CROP ACREAGE BY SOIL TYPE

CROP /	SOIL 1	SOIL 2	TOTAL
COTTON	421,243		421,243
OATS		375,000	375,000
SOYBEANS	1,004,226		1,004,226
ALFALFA	74,531		74,531
RICE		125,000	125,000

FIELD CROP ACREAGE BY FERTILIZER LEVEL

CROP /	LEVEL 1	LEVEL 2	TOTAL
COTTON		421,243	421,243
OATS		375,000	375,000
RICE		125,000	125,000

Table 48: OUTPUT FROM REPORT AGENDUM--Continued

(Listing Sheet 6)

USDA LP/90 EXAMPLE 6 RLP FOR 1956

OUTPUT SUMMARY

CROP /	PRODUCTION	ACREAGE	AVE YIELD
COTTON LINT	2,961,794	419,865	7.05
COTTON SEED	5,046,610	419,865	12.02
OATS	22,484,000	385,000	58.40
SOYBEANS	29,681,238	1,023,491	29.00
ALFALFA	117,252	56,644	2.07
RICE	4,094,000	115,000	35.60

FIELD CROP ACREAGE BY SOIL TYPE

CROP /	SOIL 1	SOIL 2	TOTAL
COTTON	419,865		419,865
OATS		385,000	385,000
SOYBEANS	1,023,491		1,023,491
ALFALFA	56,644		56,644
RICE		115,000	115,000

FIELD CROP ACREAGE BY FERTILIZER LEVEL

CROP /	LEVEL 1	LEVEL 2	TOTAL
COTTON		419,865	419,865
OATS		385,000	385,000
RICE		115,000	115,000

Table 48: OUTPUT FROM REPORT AGENDUM--Continued

(Listing Sheet 7)

USDA LP/90 EXAMPLE 6 RLP FOR 1957

OUTPUT SUMMARY

CROP /	PRODUCTION	ACREAGE	AVE YIELD
COTTON LINT	2,983,094	425,364	7.01
COTTON SEED	5,082,826	425,364	11.95
OATS	22,776,000	390,000	58.40
SOYBEANS	29,916,005	1,031,586	29.00
ALFALFA	89,112	43,049	2.07
RICE	3,916,000	110,000	35.60

FIELD CROP ACREAGE BY SOIL TYPE

CROP /	SOIL 1	SOIL 2	TOTAL
COTTON	425,364		425,364
OATS		390,000	390,000
SOYBEANS	1,031,586		1,031,586
ALFALFA	43,049		43,049
RICE		110,000	110,000

FIELD CROP ACREAGE BY FERTILIZER LEVEL

CROP /	LEVEL 1	LEVEL 2	TOTAL
COTTON		425,364	425,364
OATS		390,000	390,000
RICE		110,000	110,000

Table 48: OUTPUT FROM REPORT AGENDUM--Continued
(Listing Sheet 8)

USDA LP/90 EXAMPLE 6 RLP FOR 1958

OUTPUT SUMMARY

CROP /	PRODUCTION	ACREAGE	AVE YIELD
COTTON LINT	3,058,932	439,045	6.97
COTTON SEED	5,211,955	439,045	11.87
OATS	23,360,000	400,000	58.40
SOYBEANS	29,818,890	1,028,238	29.00
ALFALFA	67,725	32,717	2.07
RICE	3,560,000	100,000	35.60

FIELD CROP ACREAGE BY SOIL TYPE

CROP /	SOIL 1	SOIL 2	TOTAL
COTTON	439,045		439,045
OATS		400,000	400,000
SOYBEANS	1,028,238		1,028,238
ALFALFA	32,717		32,717
RICE		100,000	100,000

FIELD CROP ACREAGE BY FERTILIZER LEVEL

CROP /	LEVEL 1	LEVEL 2	TOTAL
COTTON		439,045	439,045
OATS		400,000	400,000
RICE		100,000	100,000

Table 48: OUTPUT FROM REPORT AGENDUM--Continued

(Listing Sheet 9)

USDA LP/90 EXAMPLE 6 RLP FOR 1959

OUTPUT SUMMARY

CROP /	PRODUCTION	ACREAGE	AVE YIELD
COTTON LINT	3,200,187	462,571	6.92
COTTON SEED	5,452,530	462,571	11.79
OATS	21,608,000	370,000	58.40
SOYBEANS	29,364,338	1,012,563	29.00
ALFALFA	51,471	24,865	2.07
RICE	4,628,000	130,000	35.60

FIELD CROP ACREAGE BY SOIL TYPE

CROP /	SOIL 1	SOIL 2	TOTAL
COTTON	462,571		462,571
OATS		370,000	370,000
SOYBEANS	1,012,563		1,012,563
ALFALFA	24,865		24,865
RICE		130,000	130,000

FIELD CROP ACREAGE BY FERTILIZER LEVEL

CROP /	LEVEL 1	LEVEL 2	TOTAL
COTTON		462,571	462,571
OATS		370,000	370,000
RICE		130,000	130,000

HEADING USDA LP/90 EXAMPLE 6 RECURSIVE LINEAR PROGRAMMING

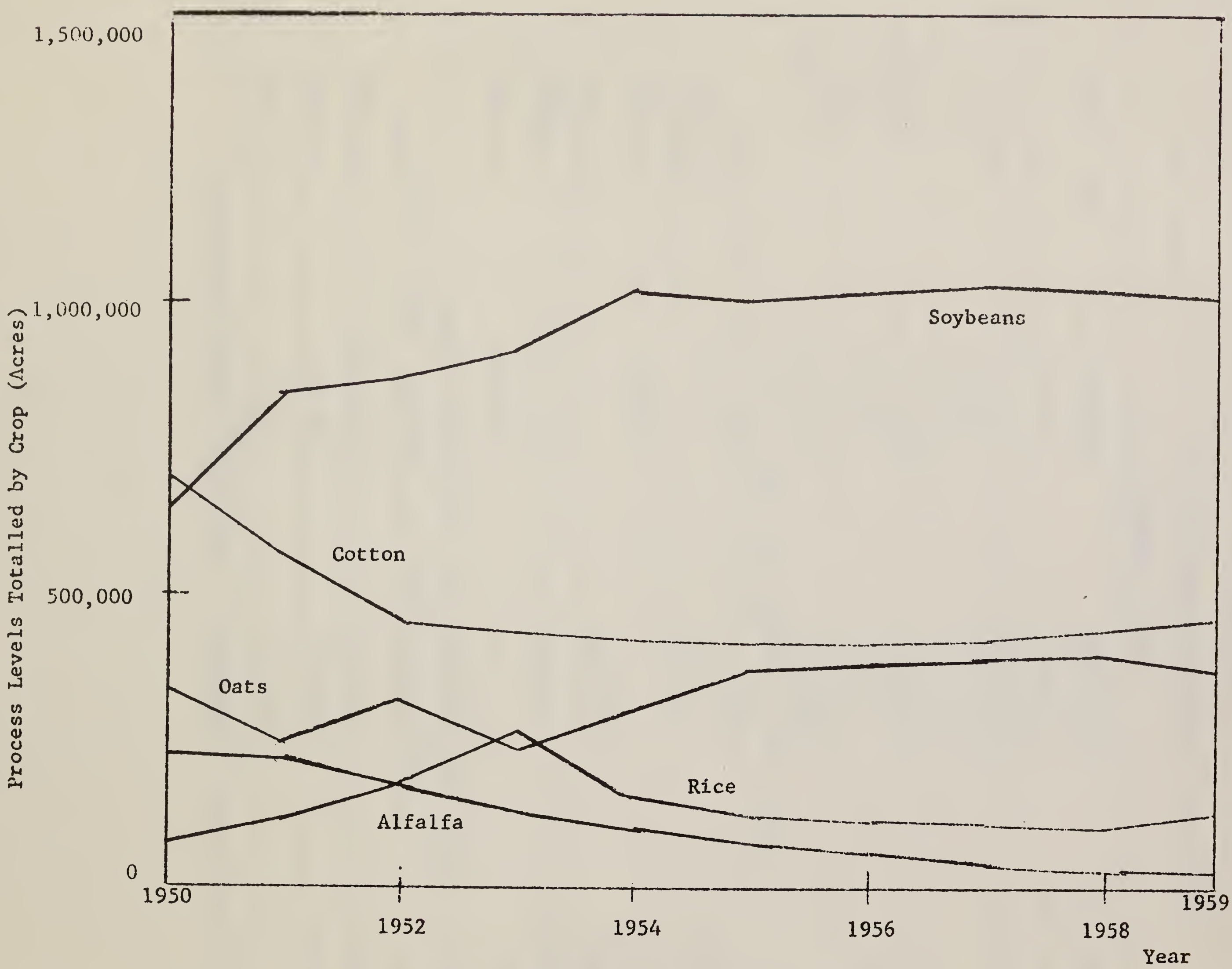


FIGURE 15: PROCESS LEVELS TOTALLED BY CROP

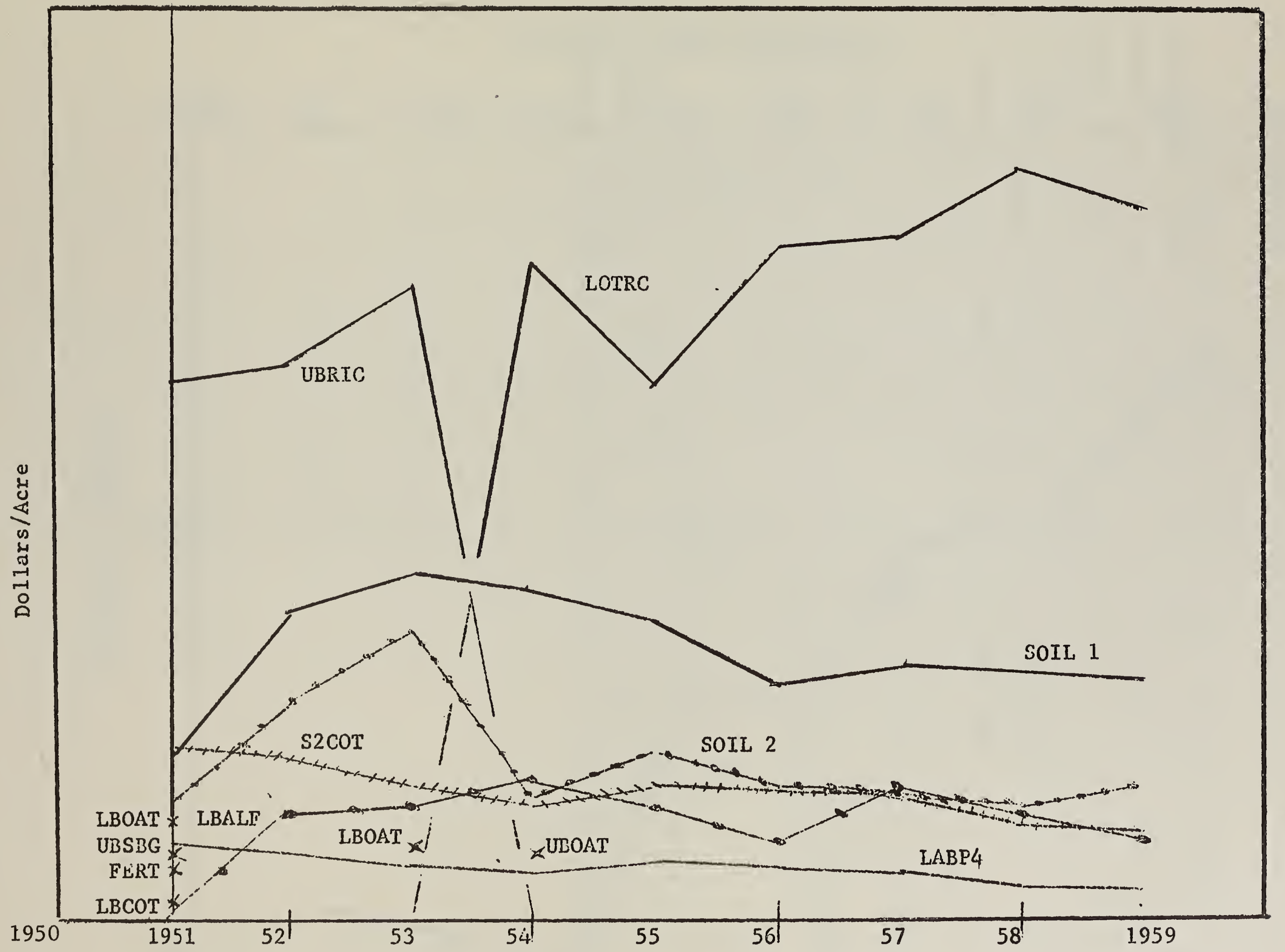


FIGURE 16: MARGINAL VALUES OF THE CONSTRAINTS

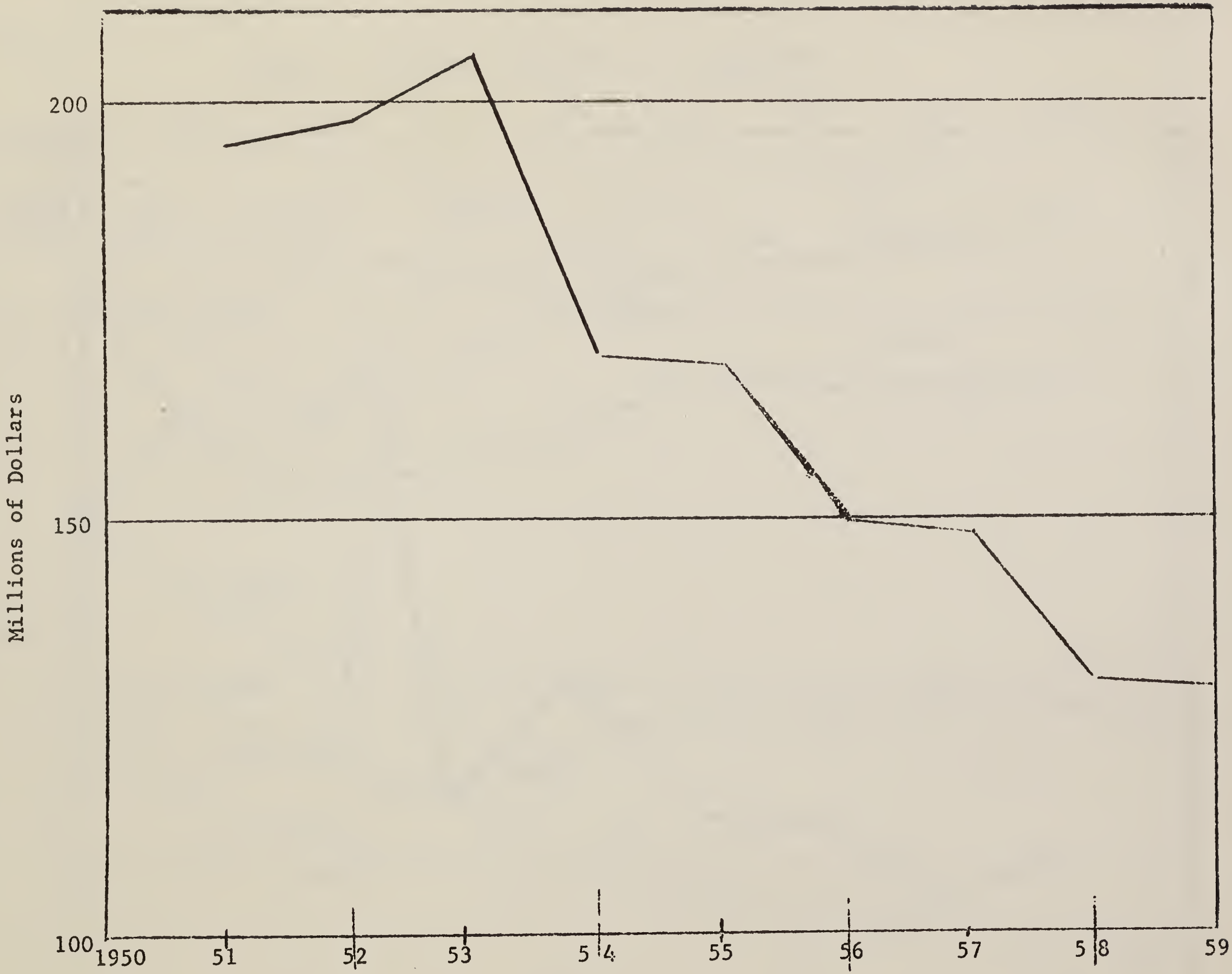


FIGURE 17: TOTAL NET RETURNS

e. Remarks

The entire run, including all of the input preparation Agendum routines, the REPORTS and OUTPUT tock .0832 hours: a shade under five minutes! As in the preceding examples, this is a remarkable performance even on the simple problem used for illustration, and, as before, almost the entire time was consumed reading the Agendum cards, reading the specified Agendum routines into the core from the Operation Tape, printing the on-line messages and writing the O-tape data.

The RLP model reported in [2] contained one hundred processes and 37 active constraints. It led to seventeen LP problems corresponding to seventeen years from 1941-1958. These were executed in about 12 minutes! While even that model is small as LP problems go, the performance is remarkable--at least by present day standards. (Who knows how slow this will appear in several years' time?)

In fact it can be said that TRANSFORM and the Agendum routines illustrated in this example make RLP a workable new tool for econometric analysis. Without them a few minutes calculations would be compounded into weeks of combined desk calculating and computer running.

Detailed Contents

Agendum Name	Brief Description	Detailed Description Found on
DO.PCR,Q,P,D	Parametric programming on cost row after optimal solution is reached. Q = Change cost row number (or name if * in col 8) P = Max. Phi D = Delta Phi for Printing	232
FIX.VEC,B,K	The vectors on the following cards are multiplied by the given scalar quantity and subtracted from RHS B; the resulting vector is output as RHS K.	233
GETOFF	Writes restart information on Mediary (Binary) Output Tape and B Tape if desired	234
GETOFF,cu	Writes restart information on channel c tape unit u. Follow by OUTPUT or RESTART, cu	234
GETOFF,PRINT	Print output after doing a GETOFF	234
HEADING OR HEADING,SAVE	Starts new problem. Identification punched in columns 13-72 will print on each output page.	234
INPUT,q,n,COUNTS,REWIND,PUNCH	Used for initial matrix assembly when sorted by columns. Sets R = q-1, B = 1 for NORMAL. q = integer number of functional rows. Since numbering starts at zero it is also the internal numerical index of the first restraint row. n = number of problems to skip. LIST = COUNTS	235
INTRODUCE,R,B	Used to enter vectors into basis. Special data cards follow. R = Row Number (functional) B = Right hand side	238
INVERT	Start reinversion of current basis	239
INVERT,NO.PUNCH	Same as INVERT except that punchouts are not obtained.	239
INVERT,NOT.INCORE	Same as INVERT except that problem will be set for tape operation after inversion.	239
LDBINARY	Loads absolute binary restart cards produced by PUNCH. Binary cards must be followed by blank card. The last six cards for the communications region should usually be removed when LDBINARY immediately follows INPUT. Cols. 13-72 for problem identification when not blank.	240

Agendum Name	Brief Description	Detailed Description Found on
LDHREG	Same as LDBINARY except that it loads punch-outs symbolically into H-Region; use only after INPUT or REVISE; remove Communications Regions cards. LDHREG must be followed by INVERT.	241
LP4290,V,X,REWIND,ALL	Converts 704 SCROL input on A6 to LP/90 input on I-tape. V: SCROL Format. 0 = fixed, 1 = variable X: Row ID Source. 0 = SCROL UP's, 1 = card reader	242
*MDUMP,CODES,REGNS,BUFFS	Dumps the designated portions of memory; the communication region is always included	242
NEW.BASIS OR NEW.BASIS,MOD	Loads basis heading cards through the card reader to overlay the basis in the machine.	243
NORMAL,R,B	Start main composite algorithm. Algorithm uses previous cost row if R is not given and previous right hand side if B is not given. R = Row Number (or Name if * in col. 8) of functional B = Right Hand Side Number	244
OUTPUT	End-of-Files the mediary tape, converts the binary output for printing off-line and terminates run.	244
OUTPUT,OLD	Same as OUTPUT but does not end-of-file so that a rewound mediary tape can be processed	244
*PATCH	For coders only, see CEIR, Inc. Reference Manual	
PUNCH	Punches binary cards for restarting. The last six cards are the communications region.	245
RESET	Resets theta, phi, change vector and change cost row numbers to zero. Required after DO.PLP or DO.PCR unless continuing with the same algorithm on the same computation.	245
RESTART	Restarts from GETOFF (on Mediary Tape), also functions as ASSIGN.	245
RESTART,cu	Restarts from GETOFF, cu where c is channel and u is tape unit. Must be preceded by CONTROLS if starting a new mediary tape.	245

Agendum Name	Brief Description	Detailed Description Found on
REVISE	Rows, columns, and/or revised elements can be inserted or substituted in the binary matrix A-tape. All new slack will appear at the end. Col. ID, Curtains, partitions and slack cards may not be used. CNT and SUM cards are ignored.	246
REVISE,q,n,PUNCH,REWIND,LIST	Like INPUT, puts the binary matrix on the A-tape. The differences from input are: Limited number (5,000) of elements to A-tape, element order is immaterial, Col ID's can be used like Row ID's to select and order columns, curtains and partitions. Slack cards cannot be used, CNT and SUM cards are ignored.	246
RHS.RANGING	Obtain the range over which the right hand side elements can be varied, one at a time, without requiring a change of basis.	251
RHS.RANGING,CLASS,W,F	Same as RHS.RANGING except that a set of contiguous columns of length W is set up starting in column F+1 of the column name. The following cards define the class characters.	251
SCALE,S	Introduces Scale Factor, S, which must have decimal point and may be negative. (Precede number by minus punch.) Saves old scale factor.	253
STATUS	Prints on and off-line the current status of all tolerances and control parameters.	253
TABLEAU	Expresses entire matrix in terms of current basis.	254
TABLEAU,FROMYAAAAA	Same as TABLEAU except it starts processing at the vector given.	254
TABLEAU,INVERSE	Gives the complete explicit inverse of the current basis. (Except for artificial columns)	254
TABLEAU,NAMES	Expresses those columns given on following cards in terms of the current basis.	254
UNSCRAMBLE	Sorts the primal solution to the vector order of the input tape.	255

2. PROPRIETARY AGENDA

Agendum Name	Brief Description	Detailed Description Found on
ADDRHS,cu	Adds the RHS described in the cards following (NEXT B,N...EOF) to the A-tape.	256
A.TO.I,cu,REWIND,PACK,ROWS,RHS,Ni....Nm.	Produces an LP/90 format input tape tape from an LP/90 A-tape and current H-region in core.	256
CARD/TAPE,cu,LOW	Reads the following cards (until END***) from the card reader and puts the card images on tape u Channel c with the density given (bbb = HIGH)	257
COMPILE,cu,cu,N,REWIND	Compiles a report writing program from Compile Data Cards on unit designated by first cu and outputs to unit designated by second cu with label N.	257
CORRECT,BCD,cu,cu,cu,n,J,Y,REWIND,LOW,GENERAL	Corrects a BCD tape by inserting, deleting, or changing any number of records.	260
DIFFERENCE,B1,B2,K	Computes a new right-hand-side, K, as B1-B2 and places it on A-tape.	262
EPSILON,B,K_1,K_2,E	The zero elements of RHS B are modified by placing E in all = and $\leq$ type rows and -E in all $\geq$ rows. The modified RHS is output to the A-tape as K_1, and B-K_1 is output as K_2.	263
EXTSOL,EXTSOL,K	Multiplies each column of the A-matrix by BETA and outputs the individual extensions in TABLEAU format. The flag "K" means that only those vectors on the following cards will be included.	263
FLAGS,mode,specs	mode = IGNORE: Marks specified vectors to prohibit them from entering the solution and/or to remove them from the present basis. mode = RESTORE: Remove the suppression flag from vectors on A-tape.	263
PICTURE,HELP	Outputs the A-tape matrix and RHS's in stylized, condensed format for off-line printing. The parameter HELP causes only those vectors having entries in infeasible rows to be included.	265

Agendum Name	Brief Description	Detailed Description Found on
REPORT,cu,N	Outputs LP answers in a formal, card controlled, report format using report writing program N. Reference may be made to any beta, pi, RHS, objective value, d_j or Aij. (See COMPILE)	266
SENSIT, SENSIT,K	Gives the rate of change in the objective function for a small change in the non-zero coefficients of the optimal basis.	267
STACK,REPORT,cu,cu	Stacks compiled report writing programs on a single tape for later use by the REPORT agendum.	267
TITLE	The information in columns 13-72 of the agendum card replaces the page heading on all subsequent printout.	267

3. USDA AGENDA

Agendum Name	Brief Description	Detailed Description Found on
CREATE,C,K	Generates a new right-hand-side of multiplying the "change matrix" times the current primal solution in core and adding the dummy RHS (Exogenous vector) numbered K. New RHS is given number C on the A-Tape.	268
CREATE,C,K,INITIATE	Same as CREATE,C,K except used after GAMMAB,cu,INITIATE.	268
GAMMAB,cu	Creates a "change matrix" for recursive Linear Programming by premultiplying the constraint matrix by a diagonal matrix	268
GAMMAB,cu,INITIATE	Reads from specified unit initial primal solution variables for beginning a Recursive Linear Programming Run.	269
INPUT,...,MATRIXG	Modification of SHARE INPUT Agendum for inputting basic data matrix to be used by TRANSFORM.	269
INPUT,...,MATRIXT	Modification of SHARE INPUT Agendum for inputting transformation matrix for use by TRANSFORM.	269

Agendum Name	Brief Description	Detailed Description Found on
READ,CHANGE,MATRIX,Q,N,COUNTS	Reads a "change matrix" and exogenous vector in LP/90 I-Tape format for Recursive Linear Programming run.	270
TRANSFORM,cu	Prelmultiplies a matrix (G) by a second matrix (T), both in I-Tape format to give a third matrix (H) in LP/90 in any format on the A-Tape.	270

INTRODUCTION

In the following pages all of the Agendum Routines are briefly described. To avoid confusion, each Agendum Routine is given the name used on the Agendum Control Card that calls it.[27/] Also, to facilitate the use of this section as a reference guide, each such routine is listed alphabetically within three categories. The latter correspond to classification of the routines as being SHARE, Proprietary (PROP) or USDA routines. The reader is reminded of the text in Part I which explains these distinctions and describes the various versions of LP/90 on which they appear. Briefly, USDA LP/90 contains all of these features; the SHARE version contains only the SHARE routines; a public version (PUBLIC LP/90) could be provided that would contain the SHARE and USDA features while a proprietary version (PROP LP/90) is available that, like the USDA version, contains all three. The advantage of PROP LP/90 is that new Agendum Routines are added to it when they are written, while USDA LP/90 contains only those features available circa Spring 1963.

It is quite likely that the material presented here will not be sufficient in all cases to enable a prospective user to go ahead on his own--even if he has studied the examples of Part IV. The selected references, with which the study concludes, contain further material on LP/90 which may fill in the necessary gaps.

The material in sections 1 and 2 (SHARE and PROP Agenda) is a highly edited version of descriptions previously published by CEIR, Inc. (see references). Section 3 (USDA Agenda) is based on the original programming specifications prepared by the author in collaboration with Eli Hellerman and Hugh Powell.

An asterisk preceding an agendum name indicates a routine for programmers, operators, or other advanced users.

[27/] The Agendum Control Card is punched with the Agendum name beginning in card column one.

1. SHARE AGENDA

CONTROLS

Certain parameters which control the function of NORMAL and the solution printouts are set by HEADING. CONTROLS enables the user to reset these paramters later in a run. Only those parameters being reset are changed. The parameters to be reset are entered on the Agendum Control Card according to the following rules.

All values must be left justified or have leading zeros.
All parameters are decimal integers.

The tolerance, E_{max}, is a decimal fraction or fixed point number. It must have a decimal point and not more than 8 digits following. Any punching other than blank, a digit or a decimal point will cause this field to be ignored.

Blank or zero for a parameter or tolerance will give the initial value shown in following table.

PARAMETERS	Card Columns	Symbol	Initial Value	Normal Value
Frequencies				
On-line log (iteration prints) and clock test frequency.	11-15	F_{cp}	*	10
Off-line log (iteration prints)	16-20	F_{ct}	*	1
Off-line full solution print	21-25	F_s	*	*
Punchout of K-H regions	26-30	F_p	*	*
No. of iterations before stop	31-35	T_{max}	*	*
No. of etas before first reinversion	36-40	T_i	*	E
Index Number[28]				
Starting Curtain	41-45	I_{cs}	0	
Ending Curtain	46-50	I_{ce}	*	
Starting Partition	51-55	I_{ps}	0	
Ending Partition	56-60	I_{pe}	*	

---

[28] For an explanation of "curtains" and "partitions" see write up of INPUT below.

Tolerance for Solution Check	Cols.	Symbol	Initial Value
Maximum Allowable Error	61-70	E_{max}	**

---

* Maximum = 32767

** 2^{-26} = 1.5×10^{-8}

E = estimated number, often 1/2 m to m.

Frequencies not equal to 32767 for F_s and F_p will suppress printing and/or punching during an automatic invert is controlled by the clock after the first inversion has occurred.

Partition indices set by a CONTROLS card are effective only in NORMAL, DO.PLP, DO.PCR, and TABLEAU. The starting and ending partitions define the set of vectors which may enter the basis. The partition codes have no ability to insure the removal of vectors not within the active set. Thus it is possible to solve a sequence of related problems, which may be nested, by the use of partitions. A terminal printout (OPTIMAL, UNBOUNDED, INFEASIBLE, THETA or PHI AT MAXIMUM) is given when the vector set is exhausted. It is not possible to have two pairs of partitions simultaneously active, but one can alternate between two separated sets of vectors by the use of CONTROLS cards (and possible sense switch 2). For example, if a problem has partitions 1 and 2 in the matrix, then there are 6 sets of vectors that can be defined by appropriate use of CONTROLS.

These possibilities are shown in the following table.

Set	Vectors Between Announcement Cards	
A	MATRIX	PARTITION,1
B	MATRIX	PARTITION,2
C	MATRIX	FIRST B
D	PARTITION,1	PARTITION,2
E	PARTITION,1	FIRST B
F	PARTITION,2	FIRST B

The sets can be called for in any order; SETS MAY BE REPEATED, IT IS NOT NECESSARY TO USE ALL SETS.

It is almost always necessary to use slack cards in the original data when Partitions are to be made active. Otherwise an infeasible solution is very likely to result.[29]

Curtain indices set by a CONTROLS card are also effective only in NORMAL, DO.PLP, and DO.PCR. However, instead of giving a terminal print when the vector set is exhausted, the LP/90 code "peeks" under the curtains to find the best vector outside. The curtains are restored on the next and subsequent iterations. The activation and numbering of curtains is completely independent of the partitions.

Curtains are used to influence the course of iterations by favoring the choice of vectors from the selected set; in a sparse problem with a very large number of vectors they may also reduce the time required for the pricing sequence in the composite algorithm. Curtains should not be active during PLP and PCR since they may cause dual infeasibilities to occur.

COST.RANGING

COST.RANGING provides the range over which the objective function elements of the basis vectors can be changed (one at a time) without changing the optimal basis. The ranges for non-basic vectors are given by the D/J values and infinity (see DO.D/J below). This routine is essentially the first step of PCR and should be used only after an optimal solution.

The printout includes the vector name, the BETA (primal solution) value, the original objective function element, the minimum objective function element, the maximum objective function element, the vector entering the basis at the minimum, and the vector entering the basis at the maximum.

---

[29] See INPUT below.

COST.RANGING,NAMES

This option of COST.RANGING has the ability to select a subset of the problem vectors for analysis. The punching ,NAMES indicates that ranges are to be obtained for the vectors with names punched in successive 6 character fields starting in column 7. Any punching in cols. 1-6 is taken as the next agendum. No more than m names may be given with one Ranging card. If the vector requested is not in the basis it is ignored. Cost Ranges on slack vectors may be obtained by overpunching the first column of the name with the slack designator ("+" or "-").

COST.RANGING,CLASS,W,F

This option is like its predecessor except that it allows one to specify a group or "class" of vectors to be analyzed each member of which is identified by common characters. "W" indicates the number of such classes; "F" indicates the number characters identifying the classes. This is accomplished as follows. The card (or cards) following the Agendum Card contains W six character vector name fields. The last 6-F characters in each field define the vector classes. For example, consider the following:

Card Column	1 7 21
Card 1	COST.RANGING,CLASS,2,3
Card 2	111COT111OAT

Cost ranging would, in this example, be performed on all vectors in the optimal basis whose names ended in either the three character COT or OAT.

DO.D/J,R,gN,gN

This routine recomputes the d_j's for all the vectors (processes or activities) in the LP-matrix including slacks. The d_j's are the relative or "opportunity" cost factors associated with each activity or process. They should be "almost" zero (i.e., within the standard tolerance) for all activities in the optimal basis. They should be positive for those vectors not in the optimal basis. For the meaning of these factors after an optimal solution, see Part II, this handbook.

They are computed by pre-multiplying the LP matrix by the R^{th} row (π_R vector) of the inverted basis. Thus, in general, they are the R^{th} row of the LP matrix expressed in terms of the current basis. If the parameter R (numerical index only) is specified, the π-vector is recomputed as the R^{th} row of the inverse. Otherwise, the existing vector in P-region is used. The D/J routine is most often employed after an optimal solution; however, it may be used at any time to give a row of the current tableau. When the solution is infeasible, R should be specified if the objective function row is desired, since otherwise the infeasibility rows will be combined with the functional. During, or at the end of PCR, the parameter R should not be specified so that the composite objective function row will be obtained.

The indicator-parameter sets shown as gN are for blocking out a certain part of the A-matrix for computing the d_j. They rely on curtains (g = C) and partitions (g = P) being in the matrix but are independent of the beginning and ending ones specified on a CONTROLS card. Either a partition or a curtain may be used either as the beginning or ending mark. The N represents the partition or curtain number and may be 1 to 5 digits.

If the d_j for the basis headings exceeds the standard tolerance, two asterisks will be printed alongside the vector name.

This routine may also be called in automatically from NORMAL if sense switch 4 is on when an optimal solution is obtained. All the d_j are recomputed in full double precision and checked to be non-negative for all j and zero for j in basis headings (within standard tolerances). If such is not the case for some j, the d_j is computed again in the reduced precision mode used in normal pricing. If it still fails the test, the d_j is printed with asterisks. If this occurs, a halt is made before continuing to the next agendum routine (after appropriate comment). If sense switch 2 is put on, the next call card is read; otherwise NORMAL is called back in. If no errors were found, no printing is done, no halt occurs, and the next call card is read.

Sense switches (other than 2 noted above) and the Pickup routine are not effective in DO.D/J.

DO.PLP,C,T,D,K,SPEC or DO.PLP,C,T,D,K,COMP

DO.PLP performs parametric programming on a right hand side. It may be used only after an optimal solution. The last right hand side optimized is the basic right hand side (specified on the last NORMAL), and the right hand side specified on the DO.PLP card is the change vector; the last objective function is used.

This parametric programming feature provides for calculating a whole sequence of related optimal solutions. In effect, the restraints of the system are progressively changed (in a specified ratio) until some vector must leave the basis to maintain feasibility. At this point a feasible vector is found to enter the basis which will maintain optimality, the change of basis is made, solution printing may occur, and the restraints are modified further until another change of basis is required. This process is repeated until either no further change can be made and maintain feasibility or a specified value is reached (either event gives a THETA AT MAXIMUM message), or until an ulimited change in requirements can be made with no further change of basis. (THETA UNBOUNDED message). Intermediate solutions can be obtained by linear interpolation.

Nomenclature

C = Change vector number ($\leq$ 9999)

T = Maximum value of Theta. If given it must contain a decimal point.

(Maximum = 7 digits plus decimal point. Positive value only.)

If T is omitted Theta is set at infinity.

D = The parameter D (punched with decimal point) is an (optional) delta theta for printing. Whenever the accumulated change in theta exceeds D, a new solution will occur and the accumulation, for this purpose, starts over. This is independent of the other printing controls except that the accumulation is set to zero when printing occurs according to F_s (off-line solution print) in CONTROLS.

K = Right hand side number to be used for new B when Theta reaches a maximum. If K is non-zero B will go to the binary matrix A-tape and be so identified. If K = 0 or blank, nothing goes to A-tape. (Maximum K $\leq$ 9999).

SPEC = A command to output K at the value of theta given by T even though theta has gone unbounded.

COMP = A command to output K at the value of theta where theta went unbounded.

bbbb = An ordinary "theta unbounded" halt will occur and K will not be output.

The solution printout and the new right hand side will be at the exact theta specified in the event that a change of basis would occur if the code were to proceed. If a maximum value of theta is found prior to the value specified, the solution print and new right hand side are computed for the value THETA MAXIMUM.

The capacity to save a composite right hand side makes convenient complicated trees of parametric runs. However, certain precautions are necessary. First, the value of "K" must not duplicate some already existing RHS. (Duplicates are given the next available RHS number.) Second, if the solution goes unbounded the new right hand side will not be produced unless the card is punched ,SPEC or ,COMP.

One very effective use of PLP where the intermediate answers are of little interest is in correcting a model. Suppose that an internal coefficient must be changed for some vector in the basis. One can multiply the level of this vector by the change in coefficient giving the change in RHS which will yield the same BETA levels as previously. One can do NORMAL against this "pseudo" RHS and then DO.PLP,C,1.0 using the negative of the change in NEXT B,C. This technique will ordinarily take <u>much less</u> time than using the composite algorithm to correct the infeasibilities introduced by changing the matrix.

In general, when using PLP one should be very certain of the formulation. For example, it is common practice to use less-than-or-equal relations for rows that are to be met as equalities when it is "known" that the profit function will drive the slack out. However, during PLP a previous condition that is "known" may prove to be violated. Thus, unless the row in question is changed to an equality, slack may come into the basis. Sometimes the easiest way to do this sort of thing involves overlaying or patching of the restart cards from a previous run.

Solution printing during PLP takes not only the usual time to write the Mediary tape, but also requires time to bring in and update the right-hand-side to the current set of requirements.

On a Theta Unbounded printout there is a column labelled GAMMA. This is the negative of the change vector in terms of the current basis. All elements in the restraint area will be negative or zero, since no positive ratio could be found against the right hand side. If the answer for a larger value of theta is desired, subtract (GAMMA)x(the increase in theta) from the Beta column.

DO.PCR,Q,P,D

This routine operates the same as PLP would applied to the dual problem; that is, the algorithm provides parametric variation of the current objective function row. It may be used only after an optimal solution; the basic objective function and the right hand side are those previously used. The indication "Q" specifies the change row either by numerical index or by its name preceded by * as in NORMAL.

This use of parametric programming also sweeps out a whole sequence of optimal solutions. The objective function elements of the problem are progressively changed (in a specified ratio) until some vector must enter the basis to maintain optimality. Solution printing may occur at each change of basis, or at the desired maximum change; intermediate answers can be obtained by linear interpolation. The algorithm terminates when either the incomimg vector is all negative, indicating an unbounded solution since no vector to remove can be found, or when the specified maximum change is reached (PHI at MAXIMUM). The other termination is PHI UNBOUNDED: this occurs whenever an unlimited modification to the functional can occur without requiring a change of basis. (The $d_{j's}$ for the change cost row will all be positive.

Nomenclature

Q = Change row numerical index or name preceded by asterisk as in NORMAL (can be any row). If Q is omitted, the previously used Q is in effect. If one was never specified, Q = 0.

P = Maximum value of Phi. If given, it must contain a decimal point (maximum = 7 digits plus decimal point. positive value only). If P is omitted (e.g. ,,), Phi is set at infinity.

D = "Delta Phi" for solution print when the accumulated change in Phi since the last print equals or exceeds given value. (Maximum = 7 digits plus decimal point. Positive only.)

The same remarks as given under PLP apply to the use of PCR. However, there is no capability to construct a composite cost row and save it on the binary matrix tape.

All switches (except 4), Pickup, "in-core" problems, etc., work just as in NORMAL. As in PLP no curtains should be active (within the partitions in use)and the active partitions must not be changed to include more vectors during the run.

FIX.VEC,B,K

The FIX.VEC agendum produces a modified RHS vector with index K. This new RHS is the original RHS with index B minus the LP matrix vectors multiplied by the levels specified on the following cards in SHARE standard format. Thus if B is the RHS, K the new RHS, A the LP matrix, and x the vector of specified constants then K = B - Ax. A feasible solution with the resulting composite right hand side then gives the minimum process levels of the vectors as the values specified rather than zero as in an ordinary RHS. Thus it is possible to have vectors essentially unrestricted as to sign (by using a negative value) or fixed at some particular level (by leaving them behind a partition). No more than m vectors can be specified with one FIX.VEC agendum.

GETOFF or GETOFF,PRINT or GETOFF,cu

This routine copies all necessary restart information to the mediary tape (M-tape) in GETOFF and GETOFF,PRINT. In GETOFF,cu restart information is copied onto channel c, tape drive number u. GETOFF,PRINT also goes directly to the output routine for printing onto the O-tape the M-tape information for listing. In this case the run is terminated. Sense switch 1 may be used in place of the GETOFFb card.

The GETOFF's can be used effectively in four ways: (1) as an emergency measure to free up the machine (GETOFFb); (2) as a problem termination (GETOFF,PRINT); (3) to save the problem status at a desired point for later use in a complicated run agenda (GETOFF,cu); (4) to provide an extra copy of the matrix tape for a long run with extensive input data by using GETOFF,cu immediately after INPUT or REVISE.

Tapes are not rewound either before or after the GEOFF. A GETOFF "clobbers" a portion of core so that it should be followed by the corresponding RESTART before going on to the next agendum.

HEADING or HEADING,SAVE

This Agendum is used to begin a new problem. It causes all control parameters to be set to their programmed values (see CONTROLS above), and records the problem identification punched in columns 13-72 for later printing as a header for each page of output on the O-tape. If HEADING,SAVE is used the machine will print a message and stop between problems to enable the operator to make tape unit changes as required.

TITLE performs much the same function as HEADING except that it does not set the control parameters (see TITLE below).

INPUT,q,n,COUNTS,REWIND,PUNCH or INPUT,q or INPUT,q,,LIST,,ETC.

This agendum call card initiates reading of the input data, which are converted to binary, packed, and written back out on the binary matrix tape (A-tape). Slack vectors are created according to the indicators on the ROW ID cards (see Part III). The slacks will be placed at the end of the matrix unless otherwise positioned by SLACK card. Elements having a row name not in the ROW ID list are suppressed (not output to the A-tape). The communication region in core is also set up for the size of the given problem with proper initial conditions.

The letter q represents an integer which must always be provided by the user. It is the number of objective functions, all of which must appear at the "top" of the matrix. For a single objective function, q = 1. All functional rows must have a ROW ID indicator of 0 or b, indicating equality (see Part III).

The remaining items, n,LIST,REWIND, and PUNCH, are <u>optional</u> controls. The letter n represents an integer which is used for getting to a particular problem on an input tape on which several problems have been stacked. It is the number of problems to pass over on the I-tape, i.e., the number of EOF cards to pass, before reading the information.

The presence of the actual punching LIST or COUNTS is an instruction to print out lists of row names and column vector names in input order together with the counts of non-zero elements in each row and column, excluding the right hand side and sum row. These lists are developed automatically in one of the edit checks but are not output unless requested. If it is desired to list without skipping any problems the card can be punched either

INPUT,q,,LIST or INPUT,q,0,LIST

The presence of the actual punching REWIND is an instruction to rewind the input tape (I-tape only) when the EOF card is encountered. Normally the tape is left sitting ready to read more data. Again REWIND without n or LIST can be punched

INPUT,q,,REWIND or INPUT,q,0,REWIND.

However, do not use 0 to indicate the omission of LIST.

PUNCH is an instruction to punch out initial restart cards. This is useful only on large problems where the risk of reconverting and re-editing a large amount of input data in the event of machine failure justifies the punching time.

On the Agendum Cards only the positions of q and n are fixed. COUNTS or LIST, REWIND, and PUNCH can be entered on the card in any order after n (or its implied omission by a double comma).

INPUT reads the I-tape, at whatever position it is sitting, and expects to find a ROW ID indicator card (or a remarks card with an asterisk in col. 1) first. The ROW ID card can call for another tape unit or the card reader.[30] If so, a ROW ID card is expected there, also. Reading continues until a MATRIX card is encountered. If reading of the ROW ID was from a special tape or card reader, it will revert to I-tape, which will be passed until a MATRIX card is encountered there also. Reading then continues until a FIRST B card is read. The FIRST B card may also call for a special tape or card reader. If so, reading is switched and another FIRST B card expected. Reading continues until an EOF indicator card is found which ends input for the current job. If on an alternate unit, reading reverts to I-tape, which is spaced up until an EOF card is also read there. If more than one reel of input tape is required, an on-line message is output when a physical end-of-file is encountered. (Do not use an EOF card to end a reel.)

The matrix is read in column by column, and each column is checked for duplicate row names. The entire matrix is checked for duplicate column names at the end. It is possible to "split" a vector without giving a duplicate vector check if a split portion contains only objective function elements. If more than about 7500 columns are present, the duplicate vector check will be incomplete; a message is written giving the name of the last vector checked.

Columns 1-30 are checked on each card for characters not meaningful in the card type expected. The sequences of curtains,

---

[30] For example, ROW ID,A6 specifies tape unit 6 on channel A.ROW ID,B9 specifies tape unit 9 on channel B, ROW ID,CR specifies card reader.

partitions, and right-hand sides are checked independently for ascending order. When an indicator card in non-ascending order is encountered it is changed to the next legitimate number and a message written. The code attempts to process all data even though errors are found. When any errors have been detected the code halts at the end of INPUT; press START will go to the next agendum. Start with sense switch 3 goes directly to OUTPUT.

There are also several optional input checking procedures. The element cards are counted for each column. This count is then checked against the value given on the CNT card, if present (see Part III). The individual descrepancies are output off-line and the data error signal is turned on. Note that these checks are made against the I tape data, whether or not some rows have been omitted from the ROW ID list.

If a sum card is present the elements on the binary matrix are summed and compared to the value on the SUM card. A discrepancy does not necessarily indicate an error since this check does not account for rows left out of the ROW ID list. A separate on-line message is given and the individual descrepancies are output off-line if any were found.

The use of LIST or COUNTS on the INPUT card causes output on the O-tape of the following information:

(1) A list of the assembled rows on the A-tape, including the kind of row-relation, together with the number of non-zero elements (not including the right hand sides).

(2) A list of rows not used.

(3) A list of vectors not used.

(4) A list of the structural vectors, slacks, curtains, and partitions which were output to the A-tape, together with the count of the number of non-zero elements in each vector (not including the sum row).

(5) The total number of rows, vectors, and non-zero elements in the problem, and its density in per-cent of non-zero elements to total possible elements.

INTRODUCE,R,B or INTRODUCE or INTRODUCE,R or INTRODUCE,,B

INTRODUCE is an agendum which will bring specified column vectors into the basis; it is followed by data cards containing the names of vectors to be introduced. The symbols "R" and "B" have the same meaning and use as in NORMAL,R,B except that names are not recognized for "R". That is, "R" is the index of the functional row and "B" is the index of the right hand side. Since optimality is disregarded during INTRODUCE, the only effect of "R" is to designate the row to use for CURRENT VALUE on iteration prints.

The use of INTRODUCE is related to the use of basis headings with INVERT or the use of curtains in a matrix. The use of basis headings in the input data requires a preliminary INVERT. A curtain, when activated, will cause vectors ahead of it to be introduced if they price out profitably, maintaining feasibility or improving feasibility and/or optimality. Vectors brought in by INTRODUCE, with no regard to optimality, do maintain feasibility if it exists or do it the "least" violence otherwise. There is no guarantee that all vectors to be introduced will remain at the conclusion of the agendum because one of the later vectors may replace an earlier one. Thus the order of the vectors influences the final basis obtained. Some of the requested vectors may not appear if no "pivot" of appropriate sign and magnitude can be found. A "CAN'T USE VECTOR" message is given.

The data cards have the following format:

Columns	Vector Name
1-6	(Blank)
7-12	YAAAAA
13-18	YAAAAA
19-24	"
25-30	"
37-42	"
43-48	"
49-54	"
55-60	"
61-66	"
67-72	"

where Y = Digit 1-9 or blank only for structural vectors; for slack vectors Y is overpunched with the row-type indicator.

A = Alphanumeric including blanks.

Blank fields are ignored in the reading of a card so that less than 11 vectors can be supplied per card; however, this means that no comments should be punched on these car

All sense switches are operative during INTRODUCE, but the "iterations to stop" and the "etas to invert" from CONTROLS are ignored.

The operation of the code is about twice as fast as NORMAL. It is therefore useful when an initial feasible solution is known in advance to be "near" the optimal solution.

INVERT or INVERT,NO.PUNCH or INVERT, NOT.INCORE

This agendum inverts the current basis. It must be used when a pre-selected non-slack basis (partial or full) is specified, i.e., in the data matrix, or by LDBINARY (if followed by NORMAL). Punching of restart information occurs when INVERT is used. INVERT,NO.PUNCH suppresses this punching. INVERT disregards curtains and partitions so that vectors outside the active partitions may be in the basis.

INVERT can be used for reducing the number of tranformations in the product form of the inverse and to reduce roundoff errors. As vectors are brought into the basis in the order in which they are found on the A-tape, the order of vectors in the matrix seriously affects the density of the inverse.

The INVERT algorithm can be called automatically (without using the Agendum Card) after a fixed number of iterations by use of the CONTROLS Agendum (See above). The optimum inversion frequency occurs when the average time per iteration, including the time for inversion, reaches a minimum. After the first INVERT, this frequency is maintained by testing the clock at each on-line print and inverting when the average time per iteration becomes more than 1% greater than the minimum average time computed.

INVERT,NOT.INCORE changes an "in core" problem to an "out of core" problem and inverts the current basis. This agendum should be used if excessive inverts are required for a problem that would ordinarily be solved "in core."

INVERT cannot be interrupted and then restarted. Only sense switches 2, 5 & 6 are recognized during INVERT to be abandoned and another call card read, but the job will not be salvageable except by reloading the punch-outs produced at the beginning. Switch 5 will cause on-line "cycle printing" as usual, but the headings are incorrect; it is useful in debugging system or machine troubles. The output format is fully described below. Sense switch 6 enables the user to obtain an on line solution print for checking in the event an error is found in the solution.

INVERT can fail if a singular basis is somehow specified in the basis headings. When a column that is a linear combination of the columns already introduced is found, the caption "MATRIX SINGULAR S VECTOR DELETED-----" is printed on-line and the machine halts. If sense switch 5 is put down and START is hit the particular linear combination which forms the singular column which is listed on the O-tape. If sense switch 5 is up and START is hit the code ignored this vector and goes on to the next.

INVERT can also fail to find vectors for all basis headings. In this case a halt and message occur. The missing vectors can be ignored by pressing start. They are listed off-line.

LDBINARY

This agendum loads the binary punch-out cards from a previous run of the same or similar problem. The binary punch-out must have the leading card removed. (This card contains BCD descriptive information.) The last six binary cards contain problem constants from the Communication Region. They include the right-hand side in use, q, the number of rows in the problem, and the number of records on the A-tape as well as the previous setting of the control parameters. The last card of the punch-out is often blank in cols. 1-72; if not, one blank card must be supplied to terminate the loading of the binary cards.

Hollerity information punched in columns 13-72 of the LDBINARY Card will replace the problem ID (from a prior HEADING). If the LDBINARY card is blank in cols. 13-72, the old ID is retained. Thus one special use of LDBINARY is to retitle a run; in this case the agendum can be followed immediately by one blank card.

If only the matrix or right-hand side data change, but not the ROW ID's, LDBINARY can follow INPUT,q or REVISE,q. In this event it may be necessary to remove the last 6 binary cards due to their being inconsistent with the changes. If changes in the action parameters are needed use CONTROLS after LDBINARY.

If no changes are required and the A-tape (binary matrix) exists, the run should be started with LDBINARY instead of INPUT,q or REVISE,q. In this case the last six cards must be retained.

If REVISE is to be used to add ROW ID's, and an old basis is desired, use LDBINARY prior to REVISE.

LDHREG

This agendum loads the binary punchout cards followed by one blank card from a previous run symbolically into H-Region. Only bais headings may be loaded with this agendum since it matches row names with those obtained from the ROW ID cards. If a match occurs, the corresponding vector name is taken from the card. If no match is found the vector is ignored.

This agendum should be preceded by INPUT or REVISE to establish the Communication Region and the ROW ID region.

One special use of this agendum is to merge the optimal bases of two or more subproblems to give a starting solution for the overall problem. A precaution that must be observed to obtain a feasible solution is that the active constraint rows of each subproblem must be distinct.

LP4290,V,X,REWIND,ALL

This converts 704 SCROL input on tape A6 to LP/90 input on I-tape. Negative slacks are not recognized and will be carried as ordinary vectors with the corresponding rows handled as equalities. Slack cards will replace UP's as encountered on the SCROL tape. The ROW ID's for LP/90 are created from the SCROL Basis when X = 0 so that this option will not work unless all UP's are in the SCROL Basis Headings.

Nomenclature

V = 0	means fixed field SCROL data
V = 1	means variable format SCROL data
X = 0	means ROW ID's from SCROL UP's in Basis
X = 1	means ROW ID's from card reader
REWIND	causes a rewind of the SCROL tape
ALL	is needed to process multiple SCROL jobs to multiple LP/90 jobs.

A corresponding code, CONVERT, has been distributed by SHARE for the 704 EDPM that takes LP/90 input to 704 SCROL format.

*MDUMP,CODES,REGNS,BUFFS

This routine dumps the specified portions of memory. The punching,,CODES calls for the Actuator, System Routines, and current agendum with mnemonic operation symbols. The punching ,REGNS calls for H-region in BCD and the other regions in octal. The Communication Region is always given in octal. MDUMP,bbbbb gives only the communication region. The code operates in two phases when called for; first, the designated portion of core are put in binary on the mediary tape, then at OUTPUT, these regions are translated as above. MDUMP, leaves core unchanged so that the run may be continued with any agendum. The order of the words CODES,REGNS,BUFFS is immaterial.

NEW.BASIS or NEW.BASIS,MOD

This agendum may be used for two purposes: (1) to insert the names of vectors that would comprise a full basis (except for slacks); (2) to overwrite some particular column name or names without disturbing the balance of the current basis.

Option (1) is called for by the agendum having only NEW.BASIS punched in col. 1-9. The code will then clear all basis column names and restore only slack variables to their proper positions in much the same manner as they would be after INPUT,q. The cards following are punched in the same manner as the BASIS cards for INPUT,q, i.e., up to 5 pairs per card, the number of pairs in col. 6, and a pair consists of 12 columns the first 6 of which are the vector name and the balance the row name. (Cols. 1-5 must be blank). If a slack is to be replaced, it is necessary to specify a vector to come into that row position. If no row name is specified with a vector name (the row name field being blank), the vector will be assigned to the first artificial position available.

In option (2) the agendum card must be punched NEW.BASIS,MOD in columns 1 to 11. This does not disturb the current basis except to overlay the headings corresponding to the row positions specified in the following cards with the vector names given. The format of the cards that follow is exactly as in option (1).

The next agendum following the basis cards in either option must be INVERT.

Option (1) may be used to make a transition from SCROL to LP/90, or to specify an advanced basis in Hollerith card form because the binary start cards are not available. A use for option (2) is the switching of one or two vector names in a basis before inversion.

NORMAL or NORMAL,R,B $(R \geq 0, B \geq 1)$

(NORMAL,R or NORMAL,,B are also possible.)

This main composite algorithm tries to find a feasible solution, if not provided, and then an optimal solution. If an optimal solution is reached, the next agendum card is called for automatically. If "no feasible solution" exists, or an "unbounded solution" is found, processing is stopped for operator action. On "no feasible solution" the next agendum will be read by pressing START twice.

NORMAL,R,B uses the functional with numerical index R and the right hand side with numerical index B. The name of the row can be substituted for the index by preceding the name with *, as in NORMAL,*1COSTS,2 if either R or B or both are blank, the R and/or B from the previous NORMAL is/are used. INPUT,q, and REVISE,q, put the highest numbered functional in R and the first right hand side in B. REVISEb and LDBINARY leave existing values unchanged; however, note that the last 6 binary cards of a punch-out also contain this information and if loaded by LDBINARY reset it accordingly.

NORMAL is required prior to PLP or PCR runs to set the current right-hand-side and functional indices. In most cases it is also needed to insure the optimal solution required to start PLP or PCR.

OUTPUT or OUTPUT,OLD

This agendum processes the binary information on the Mediary Output Tape CM-tape into a final BCD output tape for printing off-line or on the 1401 computer. It is followed by an irretrievable halt.

The OUTPUT,OLD option was designed primarily in event of trouble to convert the binary Mediary output tape to BCD output. It assumes the Mediary tape has been End-of-Filed and rewound, whereas the OUTPUTb agendum writes the EOF and rewinds the tape.

PUNCH

PUNCH punches out restart information about the current basis on binary cards to be loaded behind LDBINARY.

The initial card is Hollerith and contains iteration count (columns 2-5), eta count (columns 6-9), and heading identification (columns 13-72). In subsequent cards, the iteration count appears in columns 73-76 and the eta count is in columns 77-80. Note that these fields are dependent on the Punch Board wiring.

The next cards are absolute loading, row-binary, 23-word cards containing basis heading names and row identification (H-region). The last 6 cards contain parameter, buffer size, and other information (K-region). For use of the punch-out cards see LDBINARY.

RESET

Reset theta (for PLP), phi (for PCR) the right hand side, change vector number, and change cost row to zero. It must be used after DO.PLP or DO.PCR unless the latter has been interrupted by the use of CONTROLS, INVERT, etc., and the same parametric computation is to be resumed.

RESTART or RESTART,cu

The two forms of RESTART correspond to the several ways of doing a GETOFF. The indicators "cu" stand for channel and tape unit. The following procedures may be used:

1. GETOFF

   RESTART (Also functions as ASSIGN)

   This sequence is primarily an operator option to make the machine available to another user. After RESTART the Mediary tape is positioned correctly at the end of the last record before the GETOFF.

2. GETOFF,PRINT

   CONTROLS (may be blank)
   RESTART,cu (also acts as ASSIGN)

   (Note that the mediary tape from GETOFF,PRINT may be mounted on an alternate unit for RESTART,cu.)

   This sequence may be called for on a long job where the first answers are of interest. The purpose of the CONTROLS card is merely to insure that the new mediary tape is started properly. This will not occur if RESTART is first. A heading card is not necessary unless it is desired to change the problem ID, in which case it comes first, followed by CONTROLS. (The old heading will print on-line, the new one off-line).

3. GETOFF,cu

   CONTROLS (may be blank)
   RESTART,cu (also functions as ASSIGN)

   This sequence may be called for when a long complicated run is not completed in a single pass on the machine. The comments for case 2 also apply here since the run was broken.

4. GETOFF,cu

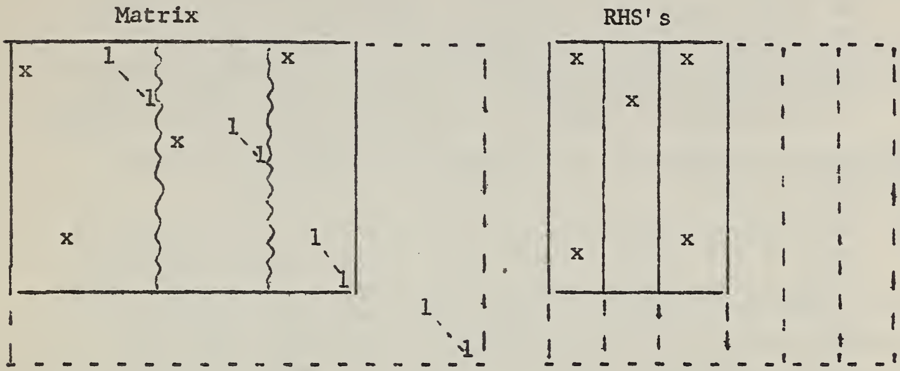
   RESTART,cu (also functions as ASSIGN)
   (subsequent agenda)
   RESTART,cu (also functions as ASSIGN)

   This sequence is primarily for formulator use in setting up a complicated agenda. Leave columns 13-18 blank so that the ASSIGN option is not inadvertantly used. Neither HEADING nor CONTROLS should immediately precede the RESTART since they will be ineffective.

REVISE or REVISE,q

The REVISE agendum is a second input routine and takes the same formats as INPUT with one or two exceptions. It has three primary functions and these should not be confused.

Function A: REVISEb modifies and/or extends the matrix and/or RHS's on an existing A-tape. In the diagram below the solid lines represent an existing model. The X's indicate element changes and the dotted lines indicate new rows and columns. The wavy lines indicate existing curtains or partitions. The "ones" indicate slack vectors.



In this use of REVISE, a Communications Region setup consistent with the A-tape to be revised is assumed and the following restrictions should be noted:

(1) Existing rows cannot be deleted nor re-named, nor can their slack indicators be changed.

(2) Existing curtains and partitions cannot be removed nor can new columns be inserted ahead of them.

(3) Existing slacks cannot be removed and all new columns will be added after them. Therefore an active ending partition can seldom be used.

(4) New slacks required by new ROW ID's (if any) will be at the end of the new matrix.

(5) Old RHS's cannot be removed but their elements can be changed in any way. New RHS's will follow old ones and hence must have higher ID numbers. NEXT Bbb is <u>not</u> allowed. If a new RHS with index N is out of sequence then precede the NEXT B,N card with NEXT B,N-1.

(6) New curtains and partitions cannot be defined, even among the new columns. Existing curtains and partitions cannot be changed or removed. Of course, they can be made inactive by a subsequent CONTROLS card.

(7) The ordering of new columns (which must in any event follow all old ones) can only be controlled by their order of input, i.e., a column name is assigned the next position when it is first encountered.

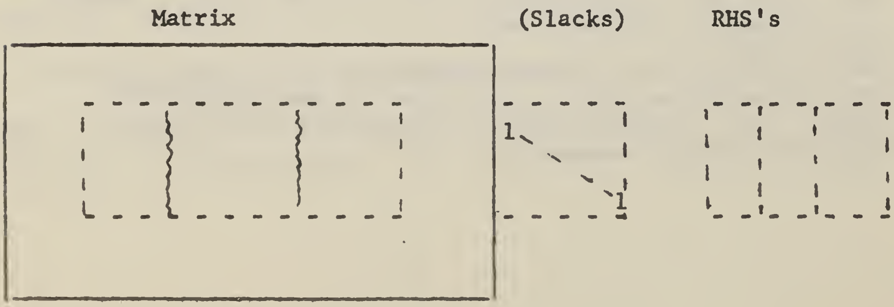
(8) If a BASIS card is used, the following Basis Headings can refer to either old or new ROW ID's and old or new columns. However, it is strongly recommended that Basis Headings refer only to new rows and new columns. This function of REVISE must be followed by INVERT and the user should be able to guarantee non-singularity of the new basis, if possible. If Basis Headings are not supplied, new basis positions will be filled with slack or artificial names, as appropriate. BASIS is not allowed if no new rows added.

(9) In particular, the COL ID option must not be used with Function A.

(10) If existing non-zero matrix elements are to be changed to zero, a zero element card must be supplied and the zero element will occupy space on the A-tape. This will cause no difficulty except in the event that all constraint elements in the column become zero, and even then only if the objective function elements are negative. Of course, there is always the danger of revising the existing basis in such a way as to make it singular, whether by zero elements or not.

(11) If all the elements of a vector are to be revised to zero the vector can be deleted by placing a DELETE card followed by cards with the name (names) of the vector(s) to be deleted, one per card, in cols. 7-12 immediately prior to the MATRIX card. This option is only effective if at least one additional matrix change is being made.

Function B: REVISE,q selects rows and columns of an existing problem on I-tape, furnishes RHS's, and creates the A-tape for a particular run, ordered in a specified way. In the diagram below the solid lines represent an existing comprehensive matrix in BCD form with elements in any order. The dotted lines indicate the selected ordered matrix (with curtains or partitions placed as desired) and the RHS's furnished. If desired, the RHS's can be standard and follow the matrix on I-tape in the usual fashion. They cannot be selected by REVISE as columns but elements corresponding to specified rows will be selected.



In this use of REVISE, the following restrictions should be noted:

1. An existing A-tape, mounted on the machine will be overwritten.

2. The I-tape or alternate input tape, must not contain CURTAIN or PARTITION cards in the body of the matrix. (They are intermixed with column ID's.)

3. All slacks are placed in order at the end of the matrix. They cannot be placed ahead of curtains or partitions. Therefore active partitions are of little use.

Function C: To put together an A-matrix and RHS's whose elements are furnished in random order, either with or without column order specified (but with row order specified, whether important or not.) This is clearly a special case of Function B but may be an important one for processing BCD output of other routines, such as a matrix generator. The following restrictions should be noted. (They apply to all Functions):

1. Row ID's must be entered first.
2. All matrix elements must be between the MATRIX AND FIRST B indicators.
3. Each right hand side must be entered as a batch in the order desired. The order of elements within a batch is immaterial.

In all three functions the following remarks are relevant.

1. The write-up on INPUT (page 7) down to and including section 5 (page 10) applies to REVISE also. In addition there is an indicator card punched COLbID in card columns 1-6 which REVISE recognizes as announcing column ID cards. These latter cards are similar to ROW ID cards except the column name is punched in columns 7-12 and columns 13-18 are ignored. The column ID's specify the desired set and order of matrix columns desired, just as ROW ID's do for rows. Note that column ID's may not be used for function A. They are optional in any event.

2. PARTITION and CURTAIN cards have the same format and function as with INPUT but may be used only with column ID's. Simply insert the cards between column names as desired. They must not appear among matrix elements.

3. The option for reading 5 pairs of names per card applied only to Basis Headings. Column ID and ROW ID cards must contain only one row name per card and must be blank in card columns 1-6 for REVISE. Card columns 19-80 are ignored.

4. The CNT and SUM cards will be passed but ignored by REVISE. Similarly, no split vector check is possible since they are eliminated by the sort in REVISE. If two element cards have the same row and column names, the last one will be used, i.e., it will overwrite preceding ones. This is true for any RHS also.

5. SLACK cards must not be used with REVISE.

6. The option to use alternate input units applies to REVISE in exactly the same way as with INPUT.

7. Function A is distinguished from Functions B and C in the following way. If the agendum card is punched REVISE in card columns 1-6 with column 7 blank, this indicates Function A. The existing matrix on A-tape will be revised according to BCD information on I-tape, which may in turn call for an alternate unit. REVISE expects to find either a remarks card (* in col. 1) or a ROW ID card ready to be read on I-tape. Remarks cards preceding the ROW ID card are not printed. If no ROW ID card is used, the first indicator must be DELETE, MATRIX, FIRST B or NEXT B. In the latter two cases, the matrix is merely copied from A-tape. The new matrix and RHS's are written on B-tape. DELETE is always followed by MATRIX since a matrix change is required. Finally, the unit designations for A-tape and B-tape are interchanged in the communications region. The communication region and all storage assignments are updated. The RHS number, however, is unchanged and existing basis headings in core are not disturbed. The next major

agendum card must be INVERT. Exception: CONTROLS, SCALE, etc., can intervene. Control parameters are not reset by REVISEb. The "FIRST B" RHS is left in B region.

8. Functions B and C are distinguished from Function A in the following way. If the agendum card is punched

REVISE,q

in columns 1 to 8 (or 9 if q $\geq$ 10), this indicates Function B or C. The parameter n and indicators REWIND, PUNCH, LIST, can also be used in this case just as with INPUT,q. However, the output resulting from LIST is much less elaborate with REVISE,q, consisting merely of the column names in the order used. (COUNTS is not recognized and causes a halt.) REVISE,q expects to find a remarks card or a ROW ID card ready to be read from I-tape. In this case, ROW ID must be present. If COL ID is used, it follows the Basis Headings, if any, or else ROW ID's. In any case DELETE may not be used in Function B and C. The matrix and RHS's are written on A-tape and core is left as it would be by INPUT. (The first RHS is left in core.) Control parameters are reset by REVISE,q.

9. Where not contradicted by the foregoing, further details found in the write-up of INPUT apply to REVISE also.

10. There is an upper limit to the total number of new elements which REVISE can handle. This is imposed by the sort routine and is in the range of 4800-5000 elements, depending on the number of rows.

RHS.RANGING or RHS.RANGING,NAMES or RHS.RANGING,CLASS,W,F

RHS.RANGING provides the range over which the values of the RHS vector can be changed (one at a time) without requiring a change of basis to maintain feasibility. This routine is essentially the first step of PLP and should be used only after an optimal solution. Like COST.RANGING it has the ability to select a subset of the problem (rows) for analysis.

The print-out includes the row name, the PI-value (dual variable), the original right-hand-side element, the minimum RHS element value, the maximum RHS element value, the vector leaving the basis at the minimum, and the vector leaving the basis at the maximum.

Nomenclature

The punching ,NAMES indicates that ranges are to be obtained for the Row Names punched on the following cards in INPUT-INTRODUCE format. (Row names in successive 6 character fields starting in column 7.) Any punching in col. 1-6 is taken as the next agendum. No more than m names may be given with one Ranging card. If the row requested is not above q in the basis it is ignored.

,CLASS,W,F sets up a group of w columns starting in column F + 1 of the name field. (That is, F is the number of columns to skip.) The following cards then define, starting in column 7, successive 6 character fields containing in the group w the class characters. All rows in the basis having the designated class characters in the corresponding columns are used.

Example:

RHS.RANGING,CLASS,2,3

ABCDEFG 1b1b1b1b --- b

All rows with either CDE or b1b in positions 345 of the row name will be chosen if they are in the basis. Other vectors will be ignored.

SCALE,S

This in effect multiplies the objective function row elements by the value specified as S. Any number other than zero must be punched with a decimal point. A blank terminates reading. If the number is negative, the minus sign comes first. Examples:

SCALE,.001	i.e.	.XXX
SCALE,-1.	i.e.	-X.
SCALE,0 or SCALE,		
(Dec. Pt. not necessary for zero.)		

Negative scale factors are used to reverse the sense of optimization (see Part II of this manual). A scale factor of .001 effectively makes the D/J tolerance greater; that is, the solution will be said to be optimal after fewer iterations. On the other hand, a scale factor of 1000 effectively makes the tolerance smaller so that more iterations will be required to reach "optimum."

The current scale factor in use is printed in the profit row of the PI column on output.

*Another use of SCALE,S is to over-ride the initial setting of the scale factor by INPUT. Thus, a composite weighting of infeasible constraints and the objective function row can be obtained. The NORMAL composite algorithm will set the scale factor to zero if the objective function dominates the infeasibilities so that no more vectors can be chosen. Therefore, scale factor should be set to the largest value possible that will <u>not</u> be changed to zero by the composite algorithm. For many problems a typical value of .1 or .01 will result in appreciably fewer iterations required to reach optimal.

STATUS

STATUS gives on-line and off-line the status of all the parameters set by CONTROLS, the current value of all TOLERANCES, the current and saved scale factors, and the current RHS number. The tolerances are given as a power of two since in most cases they are carried as a floating point characteristic with no mantissa.

TABLEAU or TABLEAU,NAMES or TABLEAU,INVERSE or TABLEAU,FROMYAAAAA

This routine transforma the entire L.P. matrix or a specified subset of it into terms of the current basis. Printing then occurs six columns at a time excluding the current basis (which is merely the identity matrix after transformation). The process can be quite time-consuming even on moderate problems, so that an entire tableau cannot be recommended as a standard operating practice. For this reason several modifiers have been included to restrict TABLEAU to certain specified subsets, i.e.,

TABLEAU — This multiplies the inverted basis times each column not in the basis. If we let P^{-1} be the inverted basis then each column of the tableau say a_j', will be

$$a_j' = P^{-1} a_j .$$

Where a_j is a column not in the basis. The tableau columns, a_j', give the vectors that enable one to express the original LP matrix columns as a linear combination of the columns in the basis. Thus $a_j = Pa_j'$.

TABLEAU,FROMYAAAAA

This gives the same information as TABLEAU, except that the tableau is computed only for the columns beginning with that named YAAAAA and ending with the last one in the currently active partition. If YAAAAA is a slack vector name place an over punch (+ or - depending on the slack type) over the first character Y (even if it is blank).

TABLEAU,NAMES — Same as TABLEAU except tableau is computed only for the columns (not in the basis) identified on the following cards. The vector names are punched in contiguous fields of six characters each <u>beginning</u> in column 7. No more than 2M+2 columns are permitted. Slack vectors are treated as in TABLEAU,FROMYAAAAA.

TABLEAU,INVERSE or TABLEAU,I

This usage outputs the non-slack columns of the inverse of the current basis.

TABLEAU has a special output format. There are seven columns on each page. The first three are headed by column names except for TABLEAU,INVERSE when they are headed by row names. The fourth column is headed by J(H) (see OUTPUT) and the last three are again vector names.

TABLEAU or its options may be called at any time. When used after REVISE or REVISE,q it is the only way to obtain a complete, exact record of the problem actually solved.

This Agendum is particularly hard on tapes and on large problems tape errors are common. If failure occurs TABLEAU is terminated and the list of six vectors in possible error pointed. Output will be obtained on the already processed correct portion so that the problem can be completed by using TABLEAU,FROMYAAAAA starting with the first bad vector.

UNSCRAMBLE

This re-orders the basis headings and corresponding beta values on the M-tape to the order of the ROW ID.

2. PROPRIETARY (PROP) AGENDA

ADDRHS,cu

The purpose of this routine is to provide an alternative to REVISE for getting additional right-hand-sides directly to the A-tape. Thus entry of change vectors for PLP can be made from the card-reader (CR) or from a specified tape (u) on channel (c). The data format is the same as that for I-tape input; the first card is NEXT B,N and the last is EOF.

It is the programmer's responsibility to specify right-hand-side numbers which are larger than those already on the A-tape since no checking for duplicates is made.

A.TO.I,cu,REWIND,PACK,ROWS,RHS,Ni,...,Nm,

This agendum processes a binary A-tape and the current H-region in core to a BCD tape in LP/90 input format. The parameters REWIND,PACK,ROWS may be omitted or placed in any order. If used, the parameter RHS must be last.

Nomenclature

cu	=	tape unit for output, if omitted B2 is used
REWIND	=	write EOF and rewind output tape at conclusion
PACK	=	pack output (approx. 33 lines/record) (suitable for printing only)
ROWS	=	output ROW ID's and BASIS only
RHS	=	output RHS (which can be specified by number in trailing parameters) only.

No options result in unpacked ROW ID, BASIS, MATRIX,RHS.

EXAMPLES		OUTPUT	
Agendum	ROW ID	Matrix	RHS No's
A.TO.I,cu	yes	yes	all
A.TO.I,cu,ROWS	yes	no	no
A.TO.I,cu,ROWS,RHS	yes	no	all
A.TO.I,cu,ROWS,RHS,1,7	yes	no	1,7
A.TO.I,cu,RHS	no	no	all
A.TO.I,cu,RHS,1,7	no	no	1,7

If the output tape is printed it must be listed single space, not under program control.

CARD/TAPE,cu,LOW

This agendum routine reads all cards following in the card-reader, converts them to BCD and writes them on the tape designated as cu (c = channel, u = unit) in high density (unless ,LOW is punched) until a card punched END*** in cols. 1-6 is encountered. When the END*** card is read an end-of-file-mark is put on the tape and the tape is rewound. The card following may be another agendum.

The routine is mostly a convenience routine to allow loading small problems on-line if the 1401 or other card to tape equipment is not available. However, it may also be used for loading small I-tapes for use by REVISE.

COMPILE,cu,cu,N,REWIND

This agendum compiles a report writing program labelled with the integer N. The compiled report can be called for by number by using the Agendum REPORT. The compile data cards are read from the unit designated by the first cu, (c = channel, u = unit) which may be the card reader. The report program is written on the second designated unit (which must be a tape unit). The COMPILE agendum destroys the initial conditions in core. Therefore, it must be followed by INPUT, RESTART, etc. prior to using other agenda.

The compile data card format description and the Input Card Form are given on succeeding pages.

The report is labelled 1 if the parameter N is missing or zero. The final punching REWIND causes the input tape to be rewound after it is processed. If this parameter is left blank the tape is in position to read the record following END*** and thus is ready to do INPUT, another report, etc.

See REPORT below.

COMPILE CARD FORMAT

Col. 1 Printer Control

1 ≡ New Page b = Single Space (b = blank)
0 ≡ Double Space + = No Space

Col. 2 Card Type

H ≡ Hollerith text, col. 3-72 are in print line.
C ≡ Continuation card, limited to one continuation card if Hollerith text in which case col. 13-60 continues the text.
T ≡ Title; begin new page, problem heading and page number on first line.

Col. 3 Blank

Col. 4 Z ≡ Suppress line if all computed values are zero. Should be used only on the first card calling for computation in a line.

Col. 1-6 END*** terminates loading of Compile Cards.

Col. 5-9 Format

Col. 5-9	blank means do not print.
Col. 5-7	decimal point position in print line.
Col. 8	decimal point or comma. If comma, commas will be inserted to mark the thousands place in values.
Col. 9	positions to the right of decimal point. (If 0, the decimal is suppressed.)

Col. 10-15 New Variable Name

Any alphanumeric symbol allowed, up to 6 characters. The symbol is used as in Fortran, e.g., XI = C 5. followed by X1 = C 17. is legal. X will have the value 5 between the first and second defining statements.

Col. 16, 25, etc. Operator

+ ≡ add	/ ≡ divide
- ≡ subtract	= ≡ equals (may be used in col. 16 only)
* ≡ multiply	S ≡ Scale all variables of kind indicated in type field. The scale factor is written in Constant format in the name field following the op. More than one scale can be given on one card. It must be the very first card of deck if used.

Col. 17, 26, etc. Operand Type

- X ≡ Beta (Primal Variable, use col. name)
- P ≡ Pi (Dual Variable, use row name)
- B ≡ Right-Hand Side
- D ≡ D/J (will be computed during REPORT)
- A ≡ a^i_j from matrix
- C ≡ Constant
- N ≡ New variable previously defined in column 10-15 (at its current value)
- M ≡ Objective function values. (From Beta region identified by row name.)

Col. 18, 27, etc. Slack type for indicating slack variables. For betas this column must be +,-, or blank. For all other types of operands this must be blank. Exception: Constant values must begin in this column.

Col. 19-24, 28-33, etc. Name of Operand; if this is a new variable it must have been defined sometime previously in col. 10-15.

Note 1: Arithmetic operatons are of the form: $(((=X_1)opX_2)opX_3)$ where quantities on left are operated on by the next succeeding quantity on the right and are thereby redefined for the next súcceeding operation.

Note 2: If type field is A, then the corresponding name field is the column name and the next contiguous (cannot be on next card) name field is the row name defining the element.

CORRECT,BCD,cu,cu,cu,n,J,Y,REWIND,LOW,GENERAL

This agendum corrects a BCD tape by inserting, deleting, or changing any number of records. The last record must be an EOF card unless the punching ,GENERAL is included on the agendum card.

While this program was written primarily to correct LP/90 input tapes (with EOF cards) any BCD tape may be corrected by operating in the GENERAL mode. To obtain maximum speed extensive buffering has been used; however, this requires either EOF cards or that the last record to be written on the new tape be part of the data input.

Nomenclature

The first cu gives the old I-tape. The second cu gives the new I-tape and the third cu gives the unit on which the correction records are to be found. Only the latter may be the card reader (CR). As usual, c designates the channel and u designates the unit.

n	=	number of EOF cards (jobs) to be read and copied prior to tape correction.
J	=	total number of jobs (EOF cards) to be output. This includes both the jobs to be corrected and those only to be copied. The number of jobs on I-tape could be greater than the specified J. Unless specified, J is assumed to be 1.
L	=	word length of records output, if not specified, it is set to 14 words.
REWIND or 0	=	The former causes rewinding of all tapes; the Input tape is also unloaded. The punching "0" will inhibit rewinding all tapes.
LOW or 0	=	The density of the output tape is set to high unless LOW is specifically requested.

GENERAL causes program to operate in the GENERAL mode. Zero or blank orient the program to LP/90 for counting of EOF cards.

Control Cards Used by CORRECT.BCD

The correction unit may contain 10 types of control cards in addition to the data cards that are to be written on the output tape. These are briefly described below. The control card names are punched beginning in card column one. A control card without a name is a "comparison card."

MATCH,B,E - Compare cols. B through E inclusive on the following "comparison" card with the same column fields on I-tape. Do not compare any columns outside of this field. This range of columns remains effective until another MATCH card is encountered. Any number of contiguous columns on a card may be used for the comparison field. Neither B nor E may be greater than 72 if the correction unit is the card reader.

After a "comparison card" is read the input tape is copied onto the output tape until a matching record is found on input. (See MATCH,B,E below.) The correction unit is then read for the next control card.

INSERT,N - Insert the next N cards that follow the INSERT card after the matched record.

DELETE,N - Delete N records beginning with the matched record.

CHANGE,N - Replace the N records <u>beginning with</u> the matched record with the N cards following the CHANGE card. CHANGE,0 is legal; it is used in the GENERAL mode.

$$$$$$ - This card may be used instead of a comparison card. No match is performed; the next instruction card is read immediately. This permits multiple operations at one point.

ADDROW - (For LP/90 only) This causes the program to insert all the following matrix element cards (no other cards are permitted) into the correct vectors on the input tape. Only additions to previously existing vectors or RHS's may be made. Once an ADDROW card is found, no other control card is permitted except END***.

CHGROW - (For LP/90 only) Is similar to ADDROW; it is used to <u>replace</u> existing matrix elements instead of adding elements. Once a CHGROW card is found, no other control card is permitted except END***.

UNTIL,N - This operation is used only in the GENERAL mode; it defines the end card. All columns of the card immediately following the UNTIL card are matched against the input records. The tape is copied (no further corrections are allowed) until N of the cards are found. After copying the last card that was found, the program terminates.

END*** - This card must always be the last card in the correction unit. This is not used in the GENERAL mode.

Record length on all tapes must be no greater than 14 words.

Input and Output tapes should be on different channels for greatest speed.

DIFFERENCE,B1,B2,K

Many times the path from an optimal solution with one right-hand-side to an optimal solution with a different right-hand-side is shortened by using PLP than by using NORMAL. However, for right-hand-sides with many elements it may be impractical to pre-punch the change vectors. DIFFERENCE is designed to overcome this difficulty.

DIFFERENCE will subtract the right-hand-side numbered B2 from right-hand-side numbered B1; the difference will be written on the A-tape as right-hand-side number K.

Illustration of how to use a DIFFERENCE follow:

```
DIFFERENCE,2,1,3
NORMAL,0,1
SET THETA
DO.PLP,3,1.0
SET THETA
NORMAL,0,2

  ETC.

NORMAL,0,1
DIFFERENCE,2,1,3
SET THETA
DO.PLP,3,1.0
SET THETA
NORMAL,0,2

  ETC.
```

EPSILON,B,K_1,K_2,E

This routine generates two new RHS's numbered K_1 and K_2, and places them on the A-tape. RHS K_1 equals the existing RHS B plus a constant E in all less than or equal and equality rows and minus the constant E in all greater than or equal rows for which the corresponding element of RHS B is zero. The RHS K_2 equals the RHS B minus the newly created RHS K_1. E is a positive decimal fraction and must contain a decimal point.

This routine is useful for problems that have right-hand-sides with a high percentage of zero elements. Better results can usually be obtained by using RHS K_1 in NORMAL and then DO.PLP with the change vector K_2.

EXTSOL,K

Each vector of the constraint matrix (including slacks) in the current basis is multiplied by the corresponding element of the BETA vector (primal variables). The results are written in TABLEAU format. If the flag "K" is punched only those vectors given on the following cards in INTRODUCE format are used. (See INTRODUCE Section 1 of this part.)

FLAGS,mode,space

FLAGS, is an LP/90 agendum routine which may be used to suppress certain matrix columns from the solution. The use is similar to that of partitions <u>except</u> that (1) the suppressed columns need not be contiguous, and (2) any flagged column is removed from the basis even if it is in the initial basic solution. The original A-tape is re-written using the U-tape as an intermediate scratch tape. If a column is removed from the basis, INVERT is automatically called at the end of the routine.

This agendum may also be used to remove the suppression bits from specified columns making them available to enter the solution.

Nomenclature

mode = IGNORE if columns are to be suppressed, (that is removed from and prevented from entering the solution).

= RESTORE if columns are to be restored; that is, the IGNORE bit is removed. If all columns are to be restored, the parameter "specs" may be omitted.

specs ≡ NAMESb -- the following cards will contain the names of variables (columns) that are to be flagged. Column names are punched up to 11 per card in successive fields of 6 col., beginning in col. 7. Blank fields are skipped. The last name must be followed by END*** in the next field. Col. 1-6 must be blank.

≡ MASK,------b
the six characters after the comma are a "mask" and correspond by position to the characters of the column names and may contain zeros or asterisks. A zero indicates that the character is not to be considered in the class definition, an asterisk indicates that it will be considered. The following cards define, starting in col. 7, successive six character fields containing any member of the class to be flagged. The last class must be followed by END***.

= CLASS,F,L An alternate way of specifying classes of vectors. (See Cost Ranging description.) It is recommended that MASK be used instead.

= FROM,XXXXXX All vectors beginning with the vector named XXXXXX through the end of the matrix will be flagged.

= THRU,XXXXXX All vectors from the beginning of the matrix through vector XXXXXX will be flagged.

= FROM,XXXXXX,THRU,YYYYYY All vectors beginning with vector XXXXXX through vector YYYYYY will be flagged.

EXAMPLES:

Card 1 FLAGS,IGNORE,NAMES

Card 2 12345A22345AEND***

This will cause the vectors named 12345 and 22345A to be ignored.

Card 1 FLAGS,IGNORE,MASK,*000

Card 2 12345A22345AEND***

This will cause all vectors that have a 1 or a 2 in the first column of their names and an A in the sixth column of their names to be ignored.

Card 1 FLAGS,RESTORE,FROM,12345A,THRU,22345A

All activities physically between vectors 12345A and 22345A inclusive will be restored to the matrix.

Restriction: The U-tape must be available.

NOTE: Flags comes back to the system "clean." That is, an inversion is done if needed before control is relinquished to the system.

PICTURE or PICTURE,HELP

The two forms of PICTURE are used under different conditions. The first condition is

PICTURE This may be used after INPUT or between any two agenda for the purpose of translating the A-tape into pictorial form on the O-tape. All numbers other than 1 are converted to alphabetic symbols which indicate magnitude. The first page of output defines these symbols. Fifty-five vectors and sixty lines are output for each page and the pages are numbered in matrix notation for easy identification.

The program is effective in showing the analysit the structure of his matrix and may help pick out incorrectly placed or missing coefficients.

PICTURE,HELP ... This program should be used only when a "NO FEASIBLE SOLUTION" is reached in NORMAL. The effect of the punching ,HELP is to extract that sub-set of vectors in the matrix that have a first order interaction with the infeasible conditions. The method of extraction is as follows:

1) For non-zero artificials - all vectors that have non-zero coefficients in the row positions corresponding to the artificial positions will be extracted.

2) For negative activities - all vectors having coefficients in the same row positions as non-zero elements in vectors with negative values will be extracted.

For the ,HELP portion of picture the U-tape must be available to the program.

The output is made directly to the O-tape, bypassing the mediary tape. Therefore PICTURE output will, in general, be the first item on the O-tape, and will be followed by the contents of the M-tape translated by OUTPUT.

REPORT,cu,N

This agendum calls in the report writing program (identified by N and previously prepared by COMPILE) from tape unit, cu. It may be used after NORMAL, DO.PLP, and DO.PCR to prepare a formal report. If blank or zero, N is set to 1. The machine contents are saved on U-tape when REPORT is called in and restored before going to the next agendum. V-tape is used for scratch, and the output is written directly on O-tape. A typical agendum sequence might be:

```
HEADING                     USE COMPILE-STACK REPORT
COMPILE,B1,B5,1
COMPILE,B1,A2,2
STACK,REPORT,A2,B5
INPUT,1
NORMAL,1,1
REPORT,B5,1
DO.D/J
NORMAL,1,2
REPORT,B5,2
PUNCH
OUTPUT
```

If used after DO.PLP or DO.PCR only the final iteration will be reported.

SENSIT or SENSIT,K

SENSIT is designed to give the rate of change in the objective function for a small change in the non-zero coefficient of the optimal basis. That is,

$$\partial\pi/\partial a_{ij} \quad \text{for all } a_{ij} \neq 0 \text{ in the optimal basis.}$$

The output is in TABLEAU form for easy identification of the elements. Activities with zero values are not output.

If all vectors of the basis are desired the agendum card is punched SENSITb. If the flag "K" is punched only those vectors given on the following cards (in INTRODUCE format) are used.

STACK,REPORT,cu,cu

This agendum will stack compiled report writing programs on a single tape for later use by the REPORT agendum. The order of units is channel and unit of report to be added, then channel and unit of stack tape. The punching ",REPORT," is <u>required</u>. The number of the incoming report to be added is checked against the existing reports on the stack tape and renumbered if necessary. Thus, there must already exist at least one report on the stack tape prior to using this agendum.

TITLE

The information in columns 13-72 of the agendum replaces the page heading on all subsequent printout. It is used in preference to HEADING when it is desired not to reset the control paramters to their programmed values.

3. USDA AGENDA

CREATE,C,K

The purpose of this agendum is to create a new right-hand-side on the LP A-tape. It does this by multiplying the current solution in core by a "change matrix" in A-tape format (but called the C-tape) and adding a dummy right-hand-side vector on the change matrix tape (C-tape). The new right-hand-side is given the index C and stored on the A-tape. Thus, we have

$$c = \Phi x^{*} + k,$$

where c is the new RHS on the A-tape; Φ is the change matrix on the C-tape, k is the dummy RHS or "exogenous" vector on the C-tape, and where x^{*} is the solution vector in core (excluding slacks). CREATE,C,K is used after a NORMAL. The indexes C and K must be greater than 0 and less than 1000.

CREATE,C,K,INITIATE

This routine functions exactly like CREATE,C,K except that it assumes that the primal solution in core was put there by GAMMAB,cu,INITIATE. Therefore it should be used after the latter agendum. See below.

GAMMAB,cu

This agendum creates a change matrix in A-tape format (but puts it on the C-tape for use by the Agendum CREATE. It does this in effect by multiplying a diagonal matrix, say Γ, times the LP constraint matrix (not including slack vectors). Thus if Φ is the change matrix and B the constraint matrix then $\Phi = \Gamma B$.

The diagonal elements of Γ are read into core from the input unit specified on the agendum card (c = channel, u = tape number, CR = card reader). The B portion of the LP matrix is obtained from the A-tape created by a previous INPUT,TRANSFORM etc. The new matrix is placed on the C-tape. If any exogenous vectors are desired on the C-tape they must already be on it. The C-tape must be in A-tape format and must contain all announcement records as required by INPUT (ROW ID, MATRIX, FIRST B, etc., END) even though there are no matrix elements.

The diagonal elements of Γ have the same format as the RHS elements, and the row names must correspond to the active constraint row names on the A-tape. Only non-zero elements need be input. Reading is terminated by a card with EOF in columns 1-3.

See READ,CHANGE MATRIX below.

GAMMAB,cu,INITIATE

This agendum is used to read in an initial solution vector in core for use by CREATE,C,K,INITIATE. The element cards are now process levels (primal variables or BETA values) and are punched the same as the Γ-elements in GAMMAB,cu except that the column names are punched in card columns 7-12. Only non-zero process levels are input.

INPUT,q,n,A....A,MATRIXG

The input for this Agendum is a basic data matrix in I-tape format called the "G" Matrix. It is used by TRANSFORM. The parameters "q" and "n" have the same meaning as in INPUT. The word "A....A" may be REWIND, COUNTS, OR PUNCH with the same meaning as in the SHARE INPUT Agendum. See TRANSFORM for usage.

INPUT,q,n,A....A,MATRIXT

This Agendum is the same as INPUT, but is used to input the "T" Matrix for use by the Agendum TRANSFORM. The T Matrix is in I-tape format. All parameters are the same as in the SHARE INPUT Agendum. See TRANSFORM for usage.

READ, CHANGE MATRIX,Q,N,COUNTS

This routine is used to read the change matrix and exogenous vectors from the I-tape and convert them to binary on the C-tape. The input tape format is exactly the same as that used by INPUT. The rows and columns of the change matrix must have exactly the same names as the LP-matrix on the I-tape. Hence, the ROW ID list must be the same for INPUT, and READ,CHANGE MATRIX. Sometimes the user may desire to read in the exogenous vectors only with this routine and compute the change matrix with GAMMAB. In this case a dummy change matrix must be read in. This may consist of only one vector element which is located in a constraint row. The parameters Q,N, etc., have the same meaning as INPUT.

TRANSFORM,cu

This routine creates an LP/90 matrix from a basic data matrix and a second transforming matrix; or, it may be used to modify an existing LP/90 matrix by means of matrix multiplication.

If we let G be the basic data or original LP/90 matrix and T the tranforming matrix, the TRANSFORM performs the operation H = TG. The matrix H is in binary on the A-tape at the termination of this agendum and is ready for use by NORMAL etc.

Example 1 Let P be a price matrix, A a flow coefficient matrix, B a stock coefficient matrix, I an identity matrix, then TRANSFORM performs the following:

$$\begin{bmatrix} P^T & 0 \\ I & 0 \\ -I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} Z \\ A \\ -A \\ B \end{bmatrix}$$

This operation creates a set of objective function Z, a double set of change rows for a complete PCR, A and -A, and carries over the active constraint rows.

Example 2

$$\begin{bmatrix} 0 & P^T & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} Z \\ A \\ B \end{bmatrix} = \begin{bmatrix} Z' \\ A \\ B \end{bmatrix}$$

This operation replaces the old set of objective functions with a new set.

This agendum routine is called by a card punched as follows, starting in column 1.

TRANSFORM,cu

Just prior to reading this card the following conditions must exist:

(1) The G matrix must be on a "GETOFF" tape on channel c, tape unit u.

(2) The T matrix must be on the ATAPE with matrix parameters set in the Communications Region.

The G matrix must be input with the following card (Punching begins in column 1):

INPUT,q,n,REWIND,MATRIXG, or
INPUT,q,n,COUNTS,MATRIXG, or
INPUT,q,n,PUNCH,MATRIXG,

where the parameters q and n have the same meaning as in INPUT, and similarly for the options REWIND,COUNTS and PUNCH. The parameter p must be greater than or equal to the total number of rows in the G matrix or less than or equal to 1024.

The value p is used in calculating the size of the G matrix records. The smaller the p the larger the records. A small value is desirable.

The T matrix must be input with the following card:

INPUT,q,n,REWIND,MATRIXT
INPUT,q,n,COUNTS,MATRIXT
INPUT,q,n,PUNCH,MATRIXT.

The ROW ID list of matrix T is used for matrix H. H slacks also are generated in accordance with this list. All H right-hand-sides must be input with matrix T. All vectors flagged in the G matrix will be flagged in the H matrix. (See FLAGS).

The vectors of T must match the rows of G both in name and order. Any curtains or partitions found in either matrix T or G are ignored; they are not put in matrix H. Any H vector with all zeros in the constraint rows will be deleted. All right-hand-sides for the H matrix must be input with the T matrix. An initial, all-slack basis exists at the completion of TRANSFORM.

An example of deck arrangement is:

```
HEADING             DECK SETUP
INPUT,1,,COUNTS,MATRIXG,350
GETOFF,B6
INPUT,5,,COUNTS, REWIND,MATRIXT
TRANSFORM,B6
NORMAL,4,3
OUTPUT
```

The B6 tape could be saved and used as long as matrix T had 350 rows or less.

SELECTED REFERENCES

1. Day, Richard H., "An Approach to Production Response," Agricultural Economics Research, October 1962. Reprint No. 35, Social Systems Research Institute, University of Wisconsin 1963.

2. Day, Richard H., Recursive Programming and Production Response, North-Holland, Amsterdam, 1963.

3. Day, Richard H., "Recursive Programming and Supply Prediction," Reprint 40, Social Systems Research Institute, University of Wisconsin, 1963.

4. Dorfman, Samuelson and Solow, Linear Programming and Economic Analysis, McGraw-Hill, 1958.

5. Hellerman, Eli, Anthony P. Round, and Hugh W. Powell, LP/90/94 Supplemental Usage Manual, CEIR, Inc., Washington, D. C., October 11, 1963.

6. Henderson, James M., "The Utilization of Agricultural Land," Review of Economics and Statistics, August 1959.

7. Hildreth, Clifford and Stanley Reiter, "On the Choice of a Crop Rotation Plan," in Koopmans (editor) Activity Analysis of Production and Allocation, Wiley, 1951.

8. Orchard-Hays, William, "Evolution of Computer Codes for Linear Programming," P-810, The RAND Corporation, Santa Montica, California, March 14, 1956.

9. Orchard-Hays, William, Matrices, Elimination, and the Simplex Method, CEIR, Inc., October 1961.

10. Schaller, W. Neill, A Recursive Programming Analysis of Regional Production Response, unpublished doctoral dissertation, University of California, 1962.

11. LP/90 Usage Manual: A Comprehensive Computing System for Linear Programming, CEIR, Inc., Washington, D. C., July 1, 1961.

12. Simon, Herbert A., "Effects of Technological Change in a Linear Model," in Koopmans, Tjalling C., Activity Analysis of Production and Allocation (John Wiley & Sons, Inc., 1951).