MICROCOMPUTER APPLICATIONS TO AGRICULTURAL POLICY ANALYSIS IN DEVELOPING COUNTRIES

By

ELTON LI

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

1976

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF PHILOSOPHY
May, 1987

MICROCOMPUTER APPLICATIONS TO AGRICULTURAL POLICY ANALYSIS IN DEVELOPING COUNTRIES

Thesis Approved:

Daryll E. Ray

Thesis Adviser

James N. Trapp

Michael R. Edgmand

Harry P. Mapp

Luther G. Tweeten

Norman N. Durham

Dean of the Graduate College

ACKNOWLEDGEMENTS

I wish to express my sincerest thanks and appreciation to my major adviser Dr. Daryll E. Ray. His influence on me will be deeply felt.

Thanks is also expressed to Dr. Luther Tweeten, Dr. James N. Trapp, Dr. Harry P. Mapp, Jr., and Dr. Michael R. Edgmand, members of my graduate committee.

I am indebted to Dr. James E. Osborn, Head of the Department of Agricultural Economics, and to the Department of Agricultural Economics for arranging financial assistance during my graduate program.

The Agricultural Policy Analysis Project, co-directed by Dr. Dean F. Schreiner and Dr. Luther Tweeten at Oklahoma State University, funded part of this research.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

This work is about a number of issues in microcomputers and agricultural policy analysis in developing countries. The thesis is organized around a number of essays. Each relates directly or indirectly to microcomputers and policy analysis. It does not cover all the issues. Because modern microcomputers can handle a majority of chores also handled by mainframe computers, a complete coverage of microcomputer applications would require a voluminous treatise on computer methods in economics in general, which is outside the scope of this work.

The underlying theme of these essays is instead applications to policy analysis most readily adapted to microcomputers. What makes microcomputers unique is difficult to pinpoint. But simplicity is obviously an important ingredient. To illustrate, let us examine the graph in Figure 1. The graph shows the hypothetical relationships between computer expertise (CE) required and the complexity of the policy analysis methods (CPA) that can be handled at a particular level of CE. The curve representing mainframe computers is marked by B, the curve for micros by M. Thus for relatively simple tasks, CE required for micros is much less than that of mainframes. As complexity of the policy work increases, CE of micros approaches that of mainframes. At high CPA, like beyond Px, CE for micros actually overtakes that of mainframes. This last point is plausible since it is actually more difficult, say, to solve an agricultural sector model of 7,000 equations on a micro than on a mainframe due to the extra skill required to overcome the severe limitations in storage and processing speed of a microcomputer. Px and beyond is certainly outside the scope of this thesis.

An advanced agency probably has staff with computer expertise of C1. At that point, they could and usually do use micros for CPA of less than P1; beyond P1 they must use mainframes. P2 is the limit of complexity the agency can handled. If computer expertise in an agency is below Cb, mainframe computers cannot be used. This is the situation in many agencies in developing countries. Below Cm, no computers can be used.

The maximum computer expertise many smaller agencies can realistically reach and more importantly, maintain or sustain, is around, say, C3. Maintenance or sustainence means that the computer expertise is generally available in the agency and not subject to evaporation with the departure of a few personnels. Before microcomputers were available, an expertise level of C3 was not sufficient for any computer methods to be used, even if access to computers was not a problem. Thus such agencies had to resort to manual operations, and settle for P0. With microcomputers, C3 of computer expertise can now handle policy work of complexity of up to P3.

Surprisingly, methods of complexity below P3 were not frequently emphasized on mainframe computers even before micros arrived, and thus have become somewhat of a lost art. One reason is computer availability. And when they are available, the marginal CE for no mainframes can be used. At high fixed and relatively low incremental expenses, incentive exists to pursue more complexity. But 'sub-P3' methods are important for many developing countries because they often are also the maximum supportable by data availability and other limitations.

Note that the curves on these graphs are not static. As microcomputer hardware and software evolves, curve M should shift outward, meaning more complexities can be handled with the same levels of computer expertise. Careful choice of hardware and software should also shift curve M of an individual agency.

Objectives

Simply stated, the objective of this thesis is to investigate agricultural policy analysis that can be performed with a minimal level of microcomputer expertise. 'Minimal' is difficult to quantify, but a good target is the equivalent of intermediate spreadsheet skills. Three viewpoints will be taken: user, tool-maker and trainer. The user is the analyst himself. His interests are the microcomputer analysis and informational handling methods that he can understand, use, and maintain. A tool-maker, on the other hand, builds tools to extend analysts' microcomputing capabilities, without elevating the requirements in computer expertise. From a trainer's point of view, of interest are the appropriate ingredients of effective training programs on microcomputers for policy analysts in developing countries.

More specifically, the objective of this thesis is to:

1. Identify simple microcomputer techniques that are useful for small agencies in developing countries and illustrate how these techniques can be used.

2. In particular, one illustration will be an extension of the framework of analysis of impacts of government price intervention policies using consumer and producer surplus to a multicommodity setting. The extension must strike a balance between theoretical soundness and simplicity. The target is an implementation suitable for a spreadsheet and easily understandable, maintainable, extendible, and adaptable.

3. Identify and discuss the difficulty and design issues in developing software which takes only a minimal amount of computer expertise to operate.

4. Identify the suitable ingredients of microcomputer training programming for policy analysts in developing countries.

Organization of the Thesis

The remainder of the thesis is organized into six chapters:

Chapter 2 begins with an introduction to the concept of a computer. It then provides a brief history of computers and microcomputers. The discussion then turns to spreadsheet programs and illustrations of how they can be used to handle policy analysis chores.

Chapter 3 is on the economic background for policy analysis work. It basically is a brief history of economic thought on questions of welfare and utility.

Chapter 4 provides some insights on software development on microcomputers. It is an illustration of designing and building tools for increasing analysts' microcomputing capability without elevating the requirement for computer expertise.

Chapter 5 is on short-term microcomputer training for policy analyst. It contains points to consider when a training program is being planned. The discussion is in fact a summary of experiences by the author in conducting courses of this nature.

Chapter 6 discusses an implementation of partial welfare analysis using consumer and producer surplus on a microcomputer spreadsheet program. A multicommodity setting is assumed. The implementation is sensitive to theoretical issues concerning consumer and producer surplus. It also demonstrates some unusual spreadsheet techniques useful for modeling simultaneous economic relationships.

Chapter 7 summarizes and concludes the thesis.

The appendices are mainly illustrations for discussion in the text. However, they might be interesting and useful in their own right. Appendix A, for example, contains a listing of source code of a LP package with an interface to Lotus 1-2-3. It contains some of the finer points about programming the the IBM PC computer. The interface to Lotus 1-2-3 (Lotus, 1985) has wide applicability as well.

Appendix B is an example of self-guided tutorial suitable for use in short term training programs.

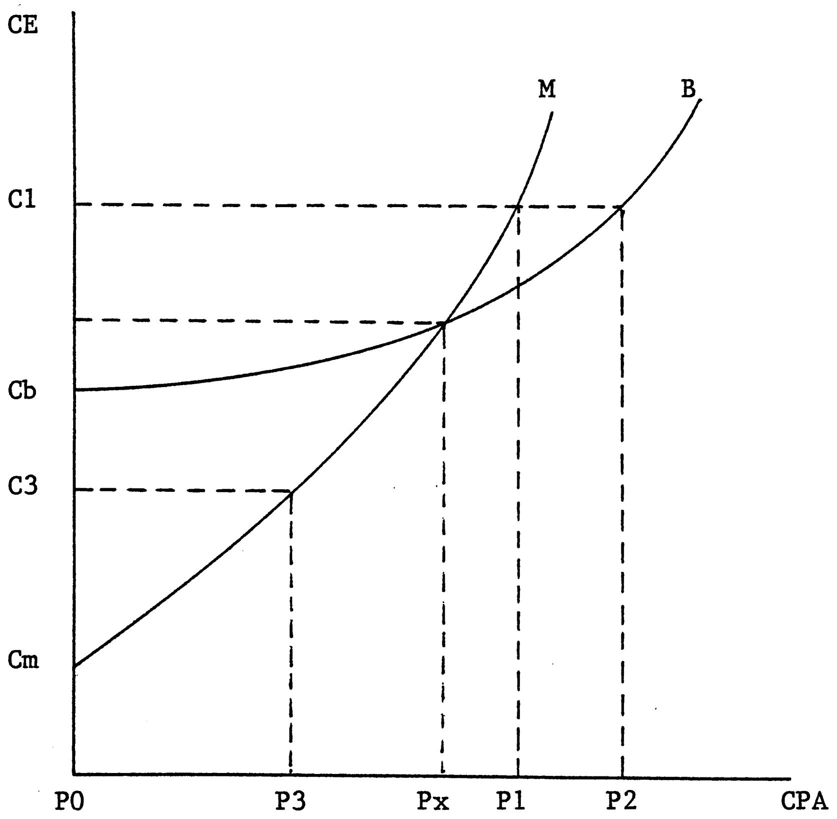


Figure 1. Relationships Between Computer Expertise (CE) and Complexity of Policy Analysis (CPA)

CHAPTER II

COMPUTERS, MICROCOMPUTERS, ELECTRONIC SPREADSHEETS, AND APPLICATIONS TO POLICY ANALYSIS

A modern computer is a synthesis of two of mankind's great achievements in the past few decades: the theory of computations and solid-state electronic technology. The former clarifies, using formal mathematical logic, what a computer ought to be and delimits the class of problems suitable for solution by this idealized machine. The latter makes possible an accurate, dependable and practical implementation.

At its native level, a computer can only operate according to a series of 1's and 0's. Software designs hide the complexity of this native level and instead present to the users high-level metaphors suitable for solving problems. Some of these metaphors useful for policy analysts include spreadsheets, business graphics, data base systems, statistical packages, optimization packages and project management packages (Li and Norton, 1985).

Essence of a Computer

Electronics aside, the primitive operations performed by an idealized computer is perhaps no more complicated than those of a mechanical clock. It is the possibility for specifying more complex tasks in terms of (huge) cumulations of these primitives plus the blinding speed at which these operations can be carried out that turns a computer into a powerful tool. Computer theorists, abstracting from implementation practicalities, conceptualize a computer as a machine consisting of, first, some storage locations capable of storing numbers. These locations are addressed sequentially with numbers for identification. Facilities are provided for humans to initially insert data and numerically coded instructions into these locations. The machine repeats a cycle of fetching a stored instruction, decoding it, and executing the instruction: this is usually called a machine cycle.

These instructions are usually operations to be applied to the stored data. Each instruction typically consists of an operation code and an address field. The operation code informs the machine what operation to perform, the address field identifies the location(s) whose contents is to be operated on. These are very simplistic operations only: among the more complex ones are instructions to add the contents of two locations and leaving the result in another location, instructions to move data from one location to another, or instructions to transfer data between the data storage locations and some external devices.

At the beginning of a machine cycle, the machine by default fetches the instruction stored in the location immediately after the one previously executed. This sequential execution of instruction is suspended, however, if the previously executed instruction is a "branch". A branch instruction instructs the machine to fetch and execute the instruction contained in the cell specified in the branch instruction's address field instead of the defaulted next sequential instruction. Branches can also be made conditional to the contents of selected cells, for instance, branching can be contingent upon a certain cell containing a zero.

These branch instructions allow iterative procedures to be implemented. Conditional branches provide additional problem solving flexibility by altering the actual sequence of instructions executed according to the changing values in some selected locations.

The Computer as an Extension to the Human Brain

It is apparent that such a machine has a "mental capacity" equivalent to the ability of following a finite set of instructions as described above in an effective, deterministic, step-by-step manner in addition to a perfect recall of information. This is both more and less than the function of a human brain. It is less since (most) human brains can function much beyond monotonously following primitive instructions and recalling information: creativity, insights, experiences and intuitions are certainly among functions that cannot be such described. It is more because the human brain is in fact a very poor performer in repeating monotonous instructions and recalling information in an error-free and speedy manner.

It is precisely for this reason that the computer is such a valuable tool. It excels very well indeed, but only in a minute area where the human brain performs relatively poorly. Just like a hammer is a valuable tool for driving nails when used as an extension to the flesh and blood of a human hand but is by itself motionless and useless, a computer is thoughtless -- it cannot make any decisions unplanned by the analyst. Only when used as a complement to the human brain can it extend intelligence beyond that reachable by the human brain alone.

Implementation of the Abstract Machine

Such is the logical essence of a computer as an abstract machine. The rest is technology. Since the simplistic nature of a computer's primitive operations necessarily implies that even the simplest useful task, such as recalculating an electronic spreadsheet or formatting a section of text with a word processor, must take astronomical machine cycles to complete, these primitive operations must be performed at near idealized speed and accuracy in a successful implementation.

A mechanical implementation of a machine with very similar ideas as described above was attempted as early as 1823 by the Englishman Charles Babbage (1792-1871). The "Analytical Engine", as the machine was called, was to operate with mechanical gears and cranks powered by steam. The machine was never perfected. The speed and accuracy demanded by a computer was simply too much for a mechanical implementation. Babbage died broke after attempting to continue the venture when the Royal Society discontinued its funding. His idea was ahead of its time. The supporting technology he needed was not to arrive for another century.

A hundred years later (1943), the first modern computer was built by Howard Aiken and International Business Machines Corporation (IBM) for a cost of a million dollars. The Mark I was constructed of electromagnetic components: a machine fifty one feet long and eight feet high. Aiken's effort was quickly duplicated in 1946 by J.P. Eckert and John Mauchly at the University of Pennsylvania. The ENIAC (Electronic Numerical Integrator and Calculator) had electronic instead of electromagnetic components and hence was several hundred times faster than the Mark I. It, however, consisted of nineteen thousand vacuum tubes and weighed thirty tons.

These early machines were haunted with reliability problems. The short life span of vacuum tubes plus the huge number of tubes used caused high break down rates. Running the machines required a large inventory of spare tubes. When a tube burned out, operation of the machine was interrupted until the offending tube was located amongst thousands and replaced.

Semiconductor technology quickly replaced vacuum tubes in computer designs. The last of the vacuum tube machines was IBM's model 709 (1958). By 1961, IBM began the design and by 1964 launched a new series of computers call the system/360. The system/370 followed in 1970. The 360/370 and their direct descendant 308xD remain the industry standard for mainframe computers today (Baer, 1980).

Enter the Micros

Technological advances allow the implantation of thousands of vacuum tubes worth of computing power of yesteryears on a "chip". The same technology which first appeared in electronic calculators blossomed into one which squeezes the computing power of near 30 tons of ENIAC, into a space of a desk-top.

Equally impressive as this increase in computing power per cubic inch is the increase in computing power per dollar. Computer usage is no longer monopsonized by rich corporations and agencies, but is now made affordable to many. With the right software, many moderately priced microcomputers offer to many small corporations and agencies computing capacity that was possible only a decade before with inflexible and often inaccessible mainframe computers.

Yet, a microcomputer is more than just a poor man's substitute for a mainframe computer. Among the largest purchasers of microcomputers are large corporations that can afford (and own) mainframe computers. Indeed, the importance and usefulness of microcomputers lie in the revolutionary concept they brought about in computing. Whereas mainframe computers were meant to be operated by persons with specialized training, microcomputers and most of their software packages are designed to be used by persons with minimal training in computers. In fact, many find it easier to communicate with the microcomputer directly instead of through "computer experts" with limited knowledge of the subject matter.

Furthermore, the low cost and high accessibility of microcomputers mean that applications can be extended to a much broader range of tasks. Many tasks are simple applications not conventionally associated with computers. One good example is word processing. The prohibitive cost of doing word processing on a mainframe or a dedicated word processing system had restricted many to the "cut-and-paste" methods of producing documents with a typewriter and photocopy machine. Excessive burden on making corrections can lead to compromise in style and substance. Improvements in both the appearance and contents of documents are now achieved with affordable word-processing software on a microcomputer. This also applies to activities related to policy work such as data tabulation, manipulation, and business graphics. Although these activities may not be considered by some as policy analysis per se, they are, doubtlessly, required as part of the policy analysis process. In practice, policy workers in smaller agencies often must perform part or all these tasks themselves, manually or otherwise. Microcomputers can therefore increase the effectiveness of an analyst by simultaneously increasing the quality of inputs to, and by freeing up more time for the central analytical process.

The microcomputer revolution began when the first commercial microcomputer -- the MITS Altair -- was launched around 1973. Apple computers soon followed. When IBM introduced its series of personal computers, the IBM PC in 1981, the revolution was ready to be mopped up.

Electronic Spreadsheets

In the "stone age" of personal computing (circa 1980), most microcomputers were bought for one of two reasons: video games and electronic spreadsheet. Although a multitude of application programs exist for microcomputers today, spreadsheet software continues to top software best-selling lists and is perhaps the most often used microcomputer software among policy analysts.

Interestingly, spreadsheet software is the only category of software that does not have a mainframe ancestry. Data bases or statistical software, for example, had long been implemented on mainframe computers before the micros came along. In fact, many microcomputer packages are adapted versions of well established mainframe packages.

In essence, an electronic spreadsheet is a replacement of the traditional way of solving problems using a pencil, a large sheet of paper (or spreadsheet), a calculator and, for most of us, an eraser, a pair of scissors, and some transparent tape. In this solution process, the paper is divided into columns with optional column or row labelling. For example, accountants would use this apparatus for developing budgets, cash flows, and projections. A moment's reflection reveals many chores of the policy analyst or his staff are performed this way as well. These chores include but are not limited to data collection, tabulation and aggregation, accounting and financial procedures (e.g. net present value and internal rate of return calculations), cost-benefit analysis, and design of linear programming matrices.

Typically, raw data are first recorded onto a sheet of paper. Other figures on the sheet are calculated directly or indirectly from these raw data. The eraser, pair of scissors, and scotch tape come in when modifications must be made.

Needless to say, these are error-prone, tedious, and boring tasks best delegated to machines. Indeed, the world's first electronic spreadsheet was invented by a Harvard Business School student motivated by the boredom and exhaustion of the necessary calculations and recalculations in case studies for his business and finance class-work. Dan Bricklin, together with Robert Frankston as the programmer, published a program called VisiCalc (Visible Calculator) in 1979 (Lammers 1986). The program ran on the Apple II computer. Not only was the program an instant hit, many attribute the success of the Apple II computer to the program. VisiCalc or VisiCalc work-alikes were quickly implemented on other machines.

One of the first published reviews of spreadsheet software appeared in the August 1979 issue of Byte Magazine. In this article (Helmero, 1979), the newly introduced VisiCalc was discussed in the context of artificial intelligence. But fundamentally, an electronic spreadsheet such as VisiCalc or LOTUS 1-2-3 is an interactive screen oriented piece of software that makes the memory of the computer a logical "blackboard" where data are remembered along with relationships. The key element of the electronic spreadsheet is this last phrase, "along with relationships". Once a set of relationships is defined, it serves as a template for a similar set of data without reentry of the formulae. In addition, the spreadsheet offers many electronic "cut and paste" operations similar to those on a word processor. In particular, blank columns or rows can be inserted. Blocks of data can be moved or copied to other locations of the spreadsheet. The relationships defined among cells are automatically updated relative to these "cut and paste" operations.

Significantly, with the ease of updating and restructuring, analysts need not have the entire design perfected on paper before translating it into a spreadsheet implementation. This would be the old fashion way of using a computer -- prominent in the mainframe era. Rather, the spreadsheet itself should be viewed as a design tool. A "tool for thought" whereby ideas can be jotted down, tried out, and successively refined into better versions. This exploratory approach to problem solving encouraged by microcomputers is an important advantage vis-a-vis mainframe computers or the manual approach.

Specifically, the electronic spreadsheet presents to the user a two-dimensional matrix of displayable, interrelated storage areas called cells. An individual cell can be empty, or contains a data value, text, or formula involving data values and contents of other cells. When a cell contains a data value or text, its content is displayed as is. Whereas a cell containing a formula would display the value of the formula instead of the formula itself. Each cell has a display format and is referred to by its coordinate within the matrix. This coordinate is called the cell address. Usually, columns are identified by letters and rows by numbers. Thus C2 is the cell in the third column and second row, AA3 is at the 27th column and third row.

If the formula 2 + A1 + B1 is inserted in cell C2, say, the cell would display a value according to the current values of cell A1 and B1. If cells A1 and B1 were 20 and 30 respectively, then 50 would be displayed in cell C2. Cells A1 and B2 can themselves contain formulae. Any change affecting the values of A1 or B1 automatically updates the value of cell C2 as specified by the formula.

The cell formulae adjust automatically and intuitively relative to any "cut and paste" operations. If a new column is inserted after the first column in the above example, thus C2 and B1 now become D2 and B2 respectively. The formula in the "old" cell C2 now appear in cell D2, and is adjusted to 2 + A1 + C1 as expected.

The few examples below should clarify these concepts and illustrate the use of electronic spreadsheets in many situations in policy analysis.

Use of Spreadsheet Programs in Policy Analysis:
Some Examples

Some examples of applications of electronic spreadsheets are now presented. Release 2 of Lotus 1-2-3 (Lotus, 1985) is used for the illustrations but other spreadsheet programs could have been used.

Example 1. Data Tabulation

Figures 2a and 2b are tabulations of data collected on the monthly sales quantities and wholesale prices of various commodities at the Waterside Market in Liberia. The data was obtained from a survey conducted by the Ministry of Agriculture of Liberia.

Wholesales margins presented in Figure 1c are computed as the differences between wholesale prices (Figure 2b) and and the sum of farm-gate prices and transport costs (not shown). At all levels, annual prices for each commodity or aggregated prices for a commodity group per month are unobservable but are computed as weighted averages of individual prices using quantity data in Figure 2a as weights.

The annual averages, standard deviations, variances and the coefficients of variation are also computed. The coefficient of variation serves as an comparative measure of the monthly variations. These calculations are simple, but laborious when a manual approach is used. The necessary training for acquiring the skill for this type of data tabulation on an electronic spreadsheet is minimal. Even for a novice, the time and effort invested in producing a data tabulation of this kind with a microcomputer is not more than what would be required by a manual method using a calculator. The time invested will be well paid off by future time savings. In addition, using a microcomputer to perform price tabulation of this kind offers the following benefits.

1. It is more accurate. Although the numbers are chosen to display with two decimals places, they are actually stored and carried in computations with 16 significant digits. Inaccuracy due to human errors and premature rounding are minimized.

2. A presentable copy can be obtained with minimal effort. Using the pencil-paper-calculator method, a final copy must be typed up for presentation or publication. With microcomputers, a publishable copy can be obtained easily at any time.

3. Changes in cell contents are more easily made. Changes in cell contents are inevitable when information is proved to be erroneous or when missing data become available. With the manual method, change in just one cell content can necessitate a whole multitude of recalculations. With an electronic spreadsheet, corrections made to the raw data automatically cause the appropriate changes to all numbers calculated directly or indirectly from the entries being altered. The relationships defined in the worksheet are permanently remembered and are always in effect.

4. Since the relationships or formulae of the spreadsheet are always remembered, they can be used as templates for future years. In other words, when a new survey is done for subsequent years, the worksheet does not have to be redone since the formulas for the calculations remain the same - only the raw data entries need to be updated with the new data. Whereas with the manual method, all calculations must be repeated for a new set of survey data, with an electronic spreadsheet, once programmed, the same worksheet can be used for years to come.

5. Not only the cell contents, but the structure and the relationships stored in an electronic spreadsheet can be altered easily as well. All electronic spreadsheets include commands to insert or to delete rows or columns, duplicate or move sections of the worksheet from one location to another. The manual alternative, short of starting

afresh, involves massive amount of erasers or covering material, scotch tape, scissors and calculator batteries.

6. The data are already stored in computer-readable media. With the manual approach, additional hand coding and data entry must be done to prepare the data for use with, say, a regression package. On the contrary, once data are stored as lotus worksheets, they can be manipulated into forms suitable as input to other programs via computerized means. Lotus worksheets can serve as a centralized database from which data can be obtained for other analytical procedures.

Example 2. Linear Programming Matrix Design

Not much imagination is required to come up with useful applications of electronic spreadsheets. Another activity in policy analysis which requires the use of large sheets of paper is LP matrix design. Figure 3 offers an electronic alternative. The electronic spreadsheet implementation is as intuitive as the manual approach. In cell A1, the name of the problem, in this case NIMBA, is inserted. B1 contains the word MAXIMIZE: a reminder that this is a maximization problem. Cell C1 is the name of the right hand side, or constraint levels. The rest of the first rows are column (or activity) names. The second row contains information for the objective function. Cell A2 is its name, in this case B. Cell D2 is the coefficient of the objective function for the variable RICEOK. Column A from row 3 onwards contains the names of the constraints. The corresponding entry in column B indicates whether the constraint is a less than constraint (L), a greater than constraint (G) or an equality constraint (E). The corresponding entry in column C is the constraint level. Thus in row 3, MLJAN is required to be less than or equal to 50 units. The rest of the entries are the Aij's.

In addition to obtaining printouts as in Figure 2, an electronic spreadsheet implementation offers the following advantages. First of all, cutting and pasting are now replaced with spreadsheet operations such as MOVE or COPY. As new activities or constraints are added, new rows and columns can be inserted. Thus the new activities and constraints can be put where they logically belong, and not at the end of the tableau as is usually done. Moreover, the spreadsheet COPY command, which allows a block of numbers or formulae to be copied from one spreadsheet location to another, can be very useful for developing LP problems having blocks of similar structure, e.g. multi-period problems.

Last but not least, the cells representing coefficients may be changed easily through formulae, if these are made dependent on some other entries. For example, in a transportation model the shipping costs between pairs of points depend on the distance and a unit cost per mile (usually a function of gas price). If gas price is stored in a separate cell and is used in computing the coefficients of the objective function, only one cell needs to be changed to derive a new LP problem.

But how can the LP problem be solved once the tableau is designed? Even if one has to recode the tableau manually to suit the input requirements of the particular LP package used, this approach still will have made the design easier. But once any information is electronically recorded, the possibility exists to translate the information to any format via computerized means. Many standard file formats exist. For

LP, many microcomputer or mainframe packages accept input in IBM's MPSX (Mathematical Programming System Extended) format, which is an industry standard (Beneke and Winterboer, 1973). A program named ToMpsx, developed by the author, is available for translating the spreadsheet tableau to MPSX format suitable as input to most LP solution packages (Epplin and Li 1986).

In fact this translation is not required at all if a program called Musah86 is used. This program directly reads in an LP tableau coded with Lotus 1-2-3, perform the optimization, and output the solution and final tableau in a format directly readable by Lotus 1-2-3 (Li, 1984, Epplin and Li, 1985). Thus the solution and final tableau can be examined and/or printed from within the Lotus 1-2-3 package. The program is discussed in more detail in Chapter 4.

Example 3. Record Keeping

Figure 4 is an example of keeping records of the monthly rice stock and flows for a parastatal marketing agency. It keeps track of the opening stock, accounts for the inflow and outflow, and computes the closing stock.

Example 4. Loan Amortization

A spreadsheet layout for computing loan amortization is shown in Figure 5. Given a loan amount, annual interest rate, number of years of the loan and number of payments per year, the amortization table displays the appropriate payment per period and separates out the payment on interest and payment on principal. The beginning principal and remaining balance are also computed and displayed for each period.

Example 5. Cost and Returns of Coffee Production

Figure 6 contains a worksheet for computing the costs, returns, net cash flows and internal rate of return of coffee production. Using labor requirements, wage rates, and operating costs, total costs of production are computed for each of 25 periods. Likewise, for each year, revenue is computed as the estimated production times the anticipated unit price. Net cash flow is computed as revenue less cost per period. The internal rate of return, a difficult measure to compute manually, can be requested easily with the @irr() function in Lotus 1-2-3. Most electronic spreadsheets have a complete list of financial functions.

Example 6. Applications in Project Appraisal

Figure 7 contains an example of discounting calculations that typically arise from project appraisal. In this analysis, the user needs to supply only the most probable incremental costs and incremental benefits series. The rest of the numbers in the tables are generated by cell formulae in the worksheet.

The alternative outcome differs from the most probable in that it incorporates the specified percent cost overrun, percent benefit shortfall, and benefit delay. Insertion of a 2 as benefit delay, for example, would automatically shift the incremental benefit column of the alternative outcome down two rows. Sensitivity analysis, which in practice is often not done when a manual approach is used, can now be performed as easily as new parameters can be inserted in the appropriate cells. Various discounted measures of project worth are also computed. More detailed discussion and the implementation specifics of this worksheet can be found in Appendix B as an example of tutorial material used for training programs.

Chapter Summary

This chapter provides some background of how computers work logically (but not physically). A brief history of computers and microcomputers was also given. Six examples of use of the electronic spreadsheet in policy analysis were demonstrated.

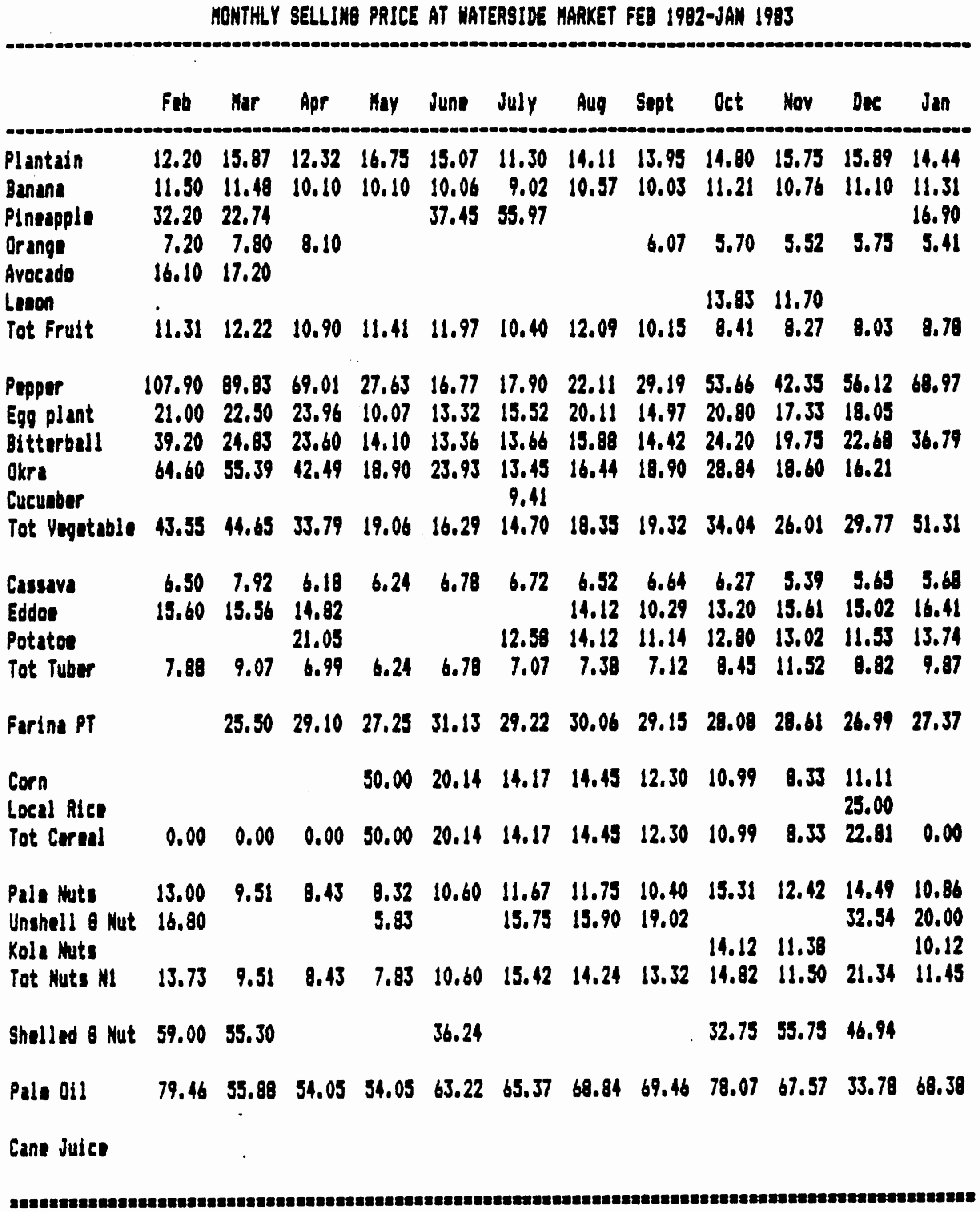
MONTHLY WEIGHTS IN POUNDS FEB 1982-JAN 1983

	Feb	Mar	Apr	May	June	July	Aug	Sept	Oct	Nov
Plantain	37,355	24,618	42,061	9,975	25,745	24,583	28,156	19,963	15,032	27,224
Banana	14,707	24,020	69,507	40,755	47,354	50,076	37,433	43,458	29,272	46,110
Pineapple	822	507	-		418	1,035				
Orange	14,200	17,870	1,530					17,350	76,993	125,360
Avocado	1,116	279								
Lemon									5,645	7,878
Tot Fruit	68,200	67,294	113,098	50,730	73,517	75,694	65,589	80,771	126,942	206,572
Pepper	2,615	11,157	13,500	20,175	30,256	31,746	33,975	15,957	12,670	6,480
Egg plant	2,280	400	6,655	5,170	6,600	11,095	2,640	1,670	3,673	2,250
Bitterball	45,799	26,268	41,880	25,320	36,780	63,120	44,280	29,526	15,240	7,140
Okra	3,908	2,382	1,925	5,280	14,740	28,315	11,880	12,818	9,613	5,610
Cucumber						1,700				
Tot Vegetable	54,602	40,207	63,960	55,945	88,376	135,976	92,775	59,971	41,196	21,480
Cassava	34,306	60,527	81,855	36,460	57,885	90,990	54,300	50,525	22,140	11,600
Eddoe	6,110	10,734	6,375				5,270	2,380	2,546	12,178
Potatoe			1,140			5,695	1,700	4,130	8,290	14,315
Tot Tuber	40,416	71,261	89,370	36,460	57,885	96,685	61,270	57,035	32,976	38,093
Farina PT		900	8,460	33,090	5,370	8,815	22,905	23,639	11,261	5,400
Corn				1,240	22,855	26,730	28,765	18,840	7,389	1,200
Local Rice										
Tot Cereal	0	0	0	1,240	22,855	26,730	28,765	18,840	7,389	1,200
Palm Nuts	12,960	9,213	16,560	8,650	14,490	5,120	31,120	23,195	8,003	3,640
Unshelled G.Nut	3,095			2,100		57,630	46,680	11,880		
Kola Nuts									5,525	28,560
Tot Nuts N1	16,055	9,213	16,560	10,750	14,490	62,750	77,800	35,075	13,528	32,200
Shelled G.Nut N1	2,548	1,504			13,095				5,517	5,865
Palm Oil	3,663	4,921	555	814	3,589	7,511	6,771	3,996	6,623	3,256
Cane Juice										

Figure 2a. Example of Data Tabulation with a Spreadsheet: Monthly Quantities.

	Dec	Jan	total
Plantain	20,164	24,713	299,589
Banana	44,520	47,033	494,245
Pineapple		5,889	8,671
Orange	129,600	90,843	473,746
Avocado			1,395
Lemon			13,523
Tot Fruit	194,284	168,478	1,291,169
Pepper	11,515	5,795	195,841
Egg plant	3,135		45,568
Bitterball	11,840	7,048	354,241
Okra	13,480		109,951
Cucumber			1,700
Tot Vegetable	39,970	12,843	707,301
Cassava	48,060	37,334	585,982
Eddoe	18,750	15,498	79,841
Potatoe	13,180	14,195	62,645
Tot Tuber	79,990	67,027	728,468
Farina PT	17,717	16,243	153,800
Corn	630		107,649
Local Rice	3,360		3,360
Tot Cereal	3,990	0	111,009
Palm Nuts	13,950	13,057	159,958
Unshelled G.Nut	8,535	2,400	132,320
Kola Nuts		9,605	43,690
Tot Nuts Nl	22,485	25,062	335,968
Shelled G.Nut Nl	10,440		38,969
Palm Oil	1,850	3,663	47,212
Cane Juice			

Figure 2a. (Cont.)



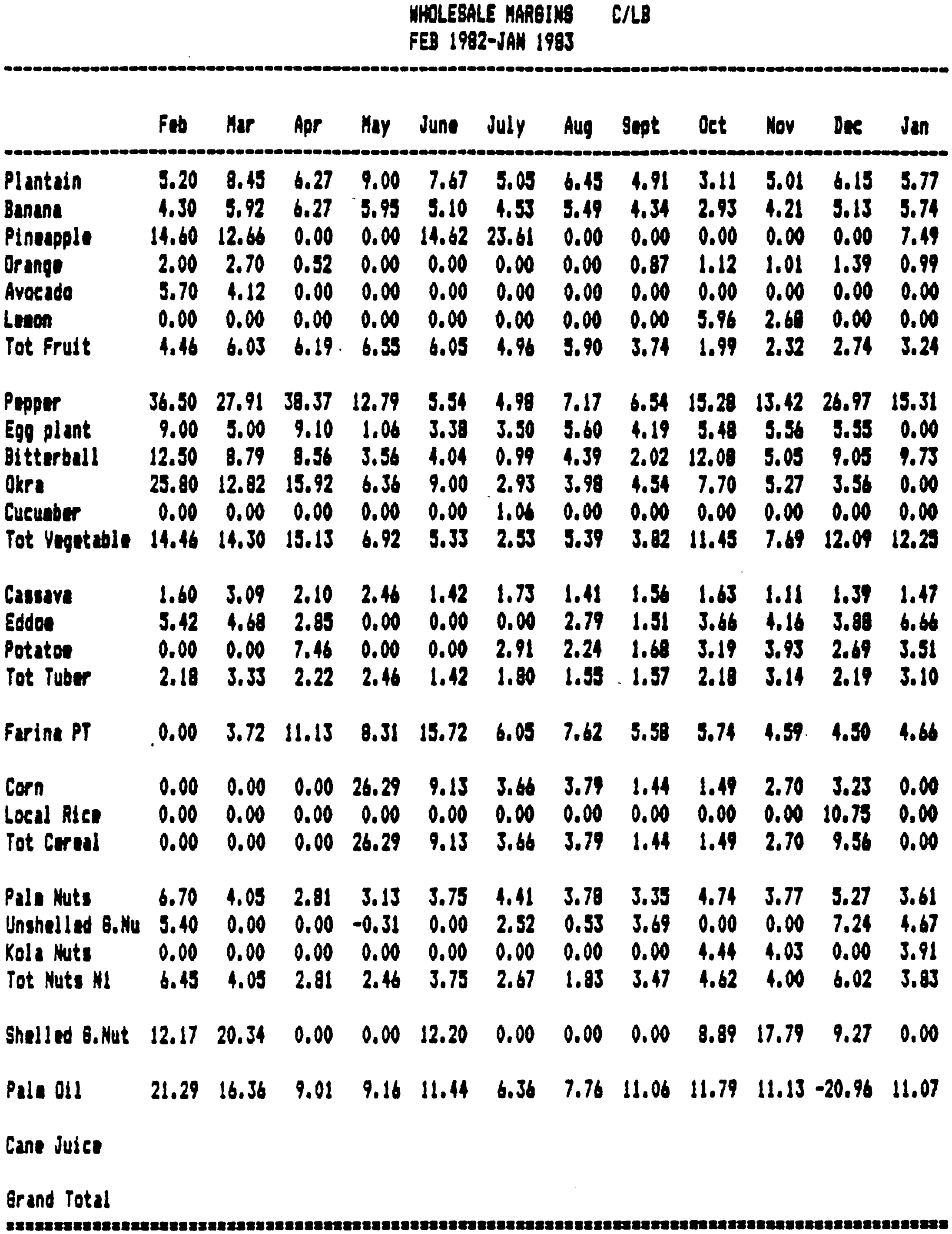
MONTHLY SELLING PRICE AT WATERSIDE MARKET FEB 1982-JAN 1983

	Feb	Mar	Apr	May	June	July	Aug	Sept	Oct	Nov	Dec	Jan
Plantain	12.20	15.87	12.32	16.75	15.07	11.30	14.11	13.95	14.80	15.75	15.89	14.44
Banana	11.50	11.48	10.10	10.10	10.06	9.02	10.57	10.03	11.21	10.76	11.10	11.31
Pineapple	32.20	22.74			37.45	55.97						16.90
Orange	7.20	7.80	8.10					6.07	5.70	5.52	5.75	5.41
Avocado	16.10	17.20										
Lemon	.								13.83	11.70		
Tot Fruit	11.31	12.22	10.90	11.41	11.97	10.40	12.09	10.15	8.41	8.27	8.03	8.78
Pepper	107.90	89.83	69.01	27.63	16.77	17.90	22.11	29.19	53.66	42.35	56.12	68.97
Egg plant	21.00	22.50	23.96	10.07	13.32	15.52	20.11	14.97	20.80	17.33	18.05	
Bitterball	39.20	24.83	23.60	14.10	13.36	13.66	15.88	14.42	24.20	19.75	22.68	36.79
Okra	64.60	55.39	42.49	18.90	23.93	13.45	16.44	18.90	28.84	18.60	16.21	
Cucumber						9.41						
Tot Vegetable	43.55	44.65	33.79	19.06	16.29	14.70	18.35	19.32	34.04	26.01	29.77	51.31
Cassava	6.50	7.92	6.18	6.24	6.78	6.72	6.52	6.64	6.27	5.39	5.65	5.68
Eddoe	15.60	15.56	14.82				14.12	10.29	13.20	15.61	15.02	16.41
Potatoe			21.05			12.58	14.12	11.14	12.80	13.02	11.53	13.74
Tot Tuber	7.88	9.07	6.99	6.24	6.78	7.07	7.38	7.12	8.45	11.52	8.82	9.87
Farina PT		25.50	29.10	27.25	31.13	29.22	30.06	29.15	28.08	28.61	26.99	27.37
Corn				50.00	20.14	14.17	14.45	12.30	10.99	8.33	11.11	
Local Rice											25.00	
Tot Cereal	0.00	0.00	0.00	50.00	20.14	14.17	14.45	12.30	10.99	8.33	22.81	0.00
Palm Nuts	13.00	9.51	8.43	8.32	10.60	11.67	11.75	10.40	15.31	12.42	14.49	10.86
Unshell G Nut	16.80			5.83		15.75	15.90	19.02			32.54	20.00
Kola Nuts									14.12	11.38		10.12
Tot Nuts Nl	13.73	9.51	8.43	7.83	10.60	15.42	14.24	13.32	14.82	11.50	21.34	11.45
Shelled G Nut	59.00	55.30			36.24				32.75	55.75	46.94	
Palm Oil	79.46	55.88	54.05	54.05	63.22	65.37	68.84	69.46	78.07	67.57	33.78	68.38
Cane Juice												

Figure 2b. Example of Data Tabulation with a Spreadsheet: Wholesale Prices.

	Annual	N	AVG	VAR	S.D.	C.V.
Plantain	14.02	12	14.37	2.85	1.69	11.75%
Banana	10.46	12	10.60	0.58	0.76	7.17%
Pineapple	24.35	5	33.05	228.13	15.10	45.70%
Orange	5.76	8	6.44	1.18	1.09	16.85%
Avocado	16.32	2	16.65	0.60	0.78	4.67%
Lemon	12.59	2	12.77	2.27	1.51	11.80%
Tot Fruit	9.68	12	10.33	2.49	1.58	15.29%
Pepper	36.08	12	50.12	874.11	29.57	58.99%
Egg plant	17.09	11	17.97	17.76	4.21	23.46%
Bitterball	20.64	12	21.87	76.08	8.72	39.88%
Okra	21.26	11	28.89	304.67	17.45	60.43%
Cucumber	9.41	1	9.41	NA	NA	NA
Tot Vegetable	24.76	12	29.24	153.04	12.37	42.31%
Cassava	6.51	12	6.37	0.44	0.66	10.39%
Eddoe	15.22	9	14.51	3.39	1.84	12.69%
Potatoe	12.85	8	13.75	9.71	3.12	22.67%
Tot Tuber	8.01	12	8.10	2.32	1.52	18.79%
Farina PT	28.39	11	28.41	2.49	1.58	5.55%
Corn	15.30	8	17.69	182.52	13.51	76.39%
Local Rice	25.00	1	25.00	NA	NA	NA
Tot Cereal	15.59	8	19.15	317.33	17.81	93.03%
Palm Nuts	11.25	12	11.40	4.77	2.18	19.16%
Unshell G Nut	17.12	7	17.98	62.58	7.91	44.00%
Kola Nuts	11.45	3	11.87	4.18	2.05	17.22%
Tot Nuts NI	13.59	12	12.68	13.63	3.69	29.11%
Shelled G Nut	43.77	6	47.66	121.13	11.01	23.09%
Palm Oil	66.76	12	63.18	153.97	12.41	19.64%
Cane Juice	0.00	0	NA	NA	NA	NA

Figure 2b. (Cont.)



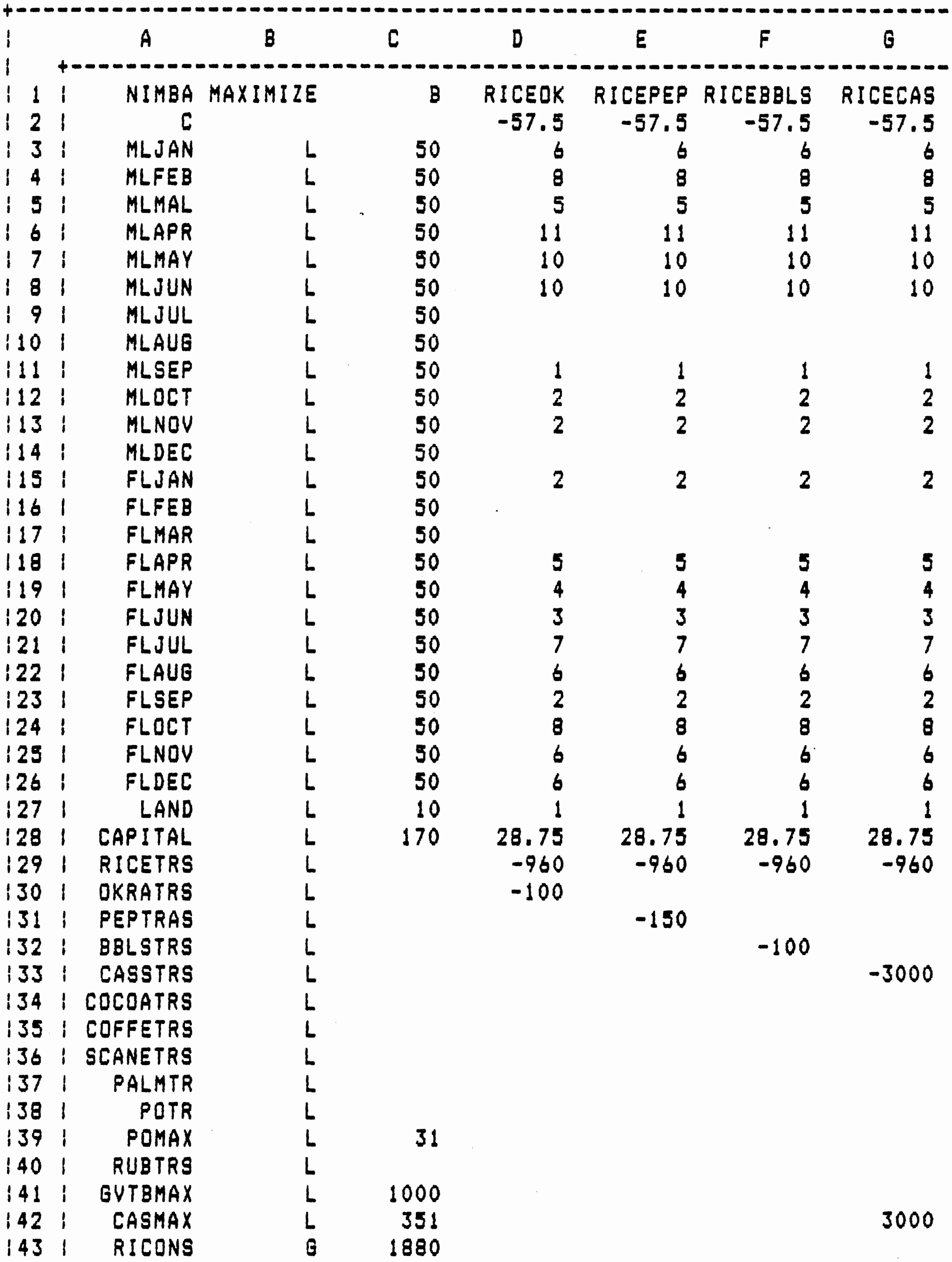
WHOLESALE MARGINS C/LB
FEB 1982-JAN 1983

	Feb	Mar	Apr	May	June	July	Aug	Sept	Oct	Nov	Dec	Jan
Plantain	5.20	8.45	6.27	9.00	7.67	5.05	6.45	4.91	3.11	5.01	6.15	5.77
Banana	4.30	5.92	6.27	5.95	5.10	4.53	5.49	4.34	2.93	4.21	5.13	5.74
Pineapple	14.60	12.66	0.00	0.00	14.62	23.61	0.00	0.00	0.00	0.00	0.00	7.49
Orange	2.00	2.70	0.52	0.00	0.00	0.00	0.00	0.87	1.12	1.01	1.39	0.99
Avocado	5.70	4.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Lemon	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.96	2.68	0.00	0.00
Tot Fruit	4.46	6.03	6.19	6.55	6.05	4.96	5.90	3.74	1.99	2.32	2.74	3.24
Pepper	36.50	27.91	38.37	12.79	5.54	4.98	7.17	6.54	15.28	13.42	26.97	15.31
Egg plant	9.00	5.00	9.10	1.06	3.38	3.50	5.60	4.19	5.48	5.56	5.55	0.00
Bitterball	12.50	8.79	8.56	3.56	4.04	0.99	4.39	2.02	12.08	5.05	9.05	9.73
Okra	25.80	12.82	15.92	6.36	9.00	2.93	3.98	4.54	7.70	5.27	3.56	0.00
Cucumber	0.00	0.00	0.00	0.00	0.00	1.06	0.00	0.00	0.00	0.00	0.00	0.00
Tot Vegetable	14.46	14.30	15.13	6.92	5.33	2.53	5.39	3.82	11.45	7.69	12.09	12.25
Cassava	1.60	3.09	2.10	2.46	1.42	1.73	1.41	1.56	1.63	1.11	1.39	1.47
Eddoe	5.42	4.68	2.85	0.00	0.00	0.00	2.79	1.51	3.66	4.16	3.88	6.66
Potatoe	0.00	0.00	7.46	0.00	0.00	2.91	2.24	1.68	3.19	3.93	2.69	3.51
Tot Tuber	2.18	3.33	2.22	2.46	1.42	1.80	1.55	1.57	2.18	3.14	2.19	3.10
Farina PT	0.00	3.72	11.13	8.31	15.72	6.05	7.62	5.58	5.74	4.59	4.50	4.66
Corn	0.00	0.00	0.00	26.29	9.13	3.66	3.79	1.44	1.49	2.70	3.23	0.00
Local Rice	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.75	0.00
Tot Cereal	0.00	0.00	0.00	26.29	9.13	3.66	3.79	1.44	1.49	2.70	9.56	0.00
Palm Nuts	6.70	4.05	2.81	3.13	3.75	4.41	3.78	3.35	4.74	3.77	5.27	3.61
Unshelled G.Nu	5.40	0.00	0.00	-0.31	0.00	2.52	0.53	3.69	0.00	0.00	7.24	4.67
Kola Nuts	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.44	4.03	0.00	3.91
Tot Nuts Nl	6.45	4.05	2.81	2.46	3.75	2.67	1.83	3.47	4.62	4.00	6.02	3.83
Shelled G.Nut	12.17	20.34	0.00	0.00	12.20	0.00	0.00	0.00	8.89	17.79	9.27	0.00
Palm Oil	21.29	16.36	9.01	9.16	11.44	6.36	7.76	11.06	11.79	11.13	-20.96	11.07
Cane Juice												
Grand Total												

Figure 2c. Example of Data Tabulation with a Spreadsheet: Wholesale Prices.

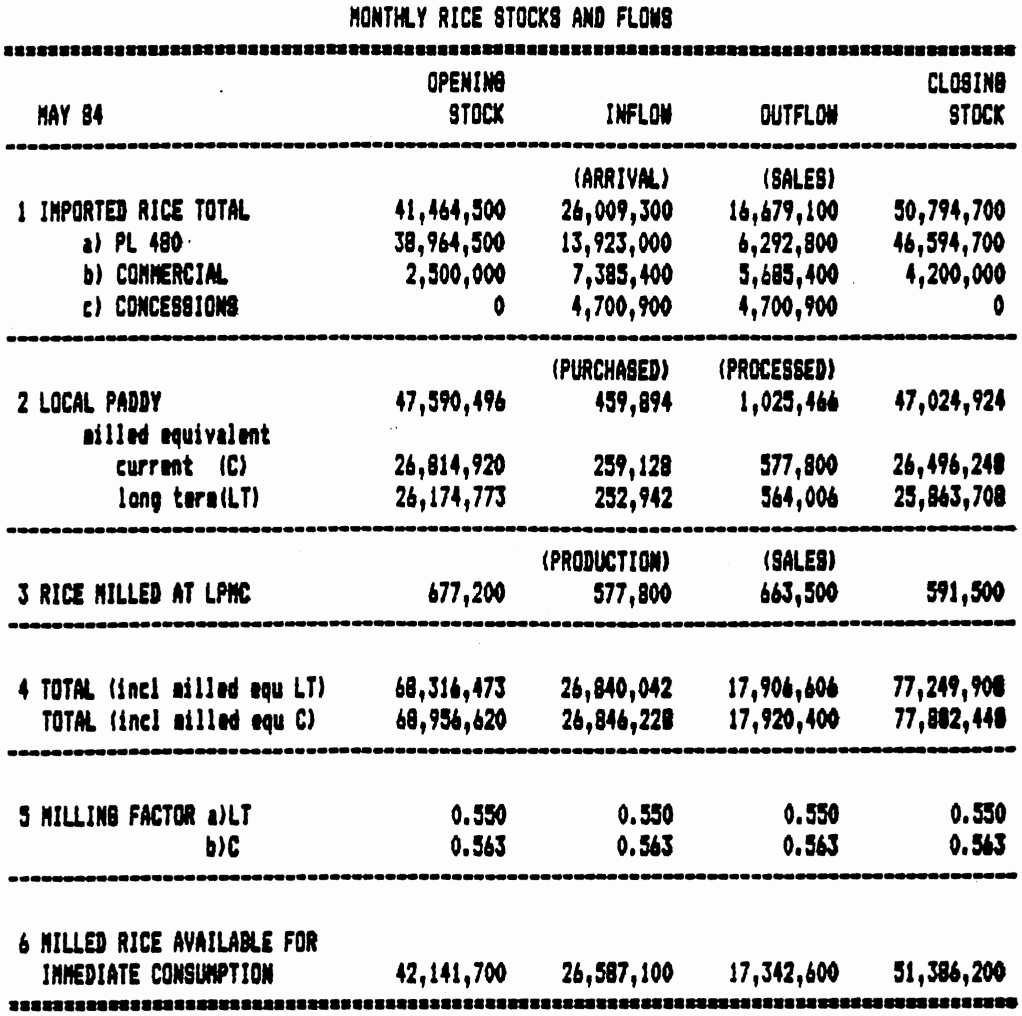
	Total	N	AVG	VAR	S.D.	C.V.
Plantain	6.03	12	6.09	2.74	1.65	27.18%
Banana	5.11	12	4.99	0.93	0.96	19.32%
Pineapple	10.73	5	14.60	189.22	13.76	94.24%
Orange	1.21	8	1.33	1.16	1.08	81.39%
Avocado	5.38	2	4.91	41.43	6.44	131.09%
Lemon	4.05	2	4.32	36.48	6.04	139.82%
Tot Fruit	3.92	12	4.51	2.76	1.66	36.79%
Pepper	12.95	12	17.57	142.70	11.95	68.01%
Egg plant	4.86	11	5.22	7.95	2.82	54.02%
Bitterball	5.96	12	6.73	14.96	3.87	57.48%
Okra	6.08	11	8.90	55.13	7.42	83.44%
Cucumber	1.06	1	1.06	NA	NA	NA
Tot Vegetable	7.83	12	9.28	20.16	4.49	48.39%
Cassava	1.82	12	1.75	0.30	0.55	31.52%
Eddoe	4.46	9	3.96	6.74	2.60	65.60%
Potatoe	3.25	8	3.45	7.66	2.77	80.19%
Tot Tuber	2.24	12	2.26	0.42	0.65	28.58%
Farina PT	6.90	11	7.06	17.36	4.17	59.05%
Corn	4.57	8	6.47	85.90	9.27	143.33%
Local Rice	10.75	1	10.75	NA	NA	NA
Tot Cereal	4.75	8	7.26	89.20	9.44	130.13%
Palm Nuts	4.02	12	4.11	1.13	1.06	25.78%
Unshelled G.Nu	2.29	7	3.39	12.79	3.58	105.47%
Kola Nuts	4.06	3	4.13	19.24	4.39	106.28%
Tot Nuts N1	3.34	12	3.83	1.89	1.37	35.86%
Shelled G.Nut	12.10	6	13.44	129.99	11.40	84.81%
Palm Oil	10.01	12	8.79	103.42	10.17	115.71%
Cane Juice	0.00	0	NA	NA	NA	NA
Grand Total	0	0	NA	NA	NA	NA

Figure 2c. (Cont.)



```
+-------------------------------------------------------------------
|          A        B        C        D        E        F        G
|    +--------------------------------------------------------------
| 1 |    NIMBA MAXIMIZE        B   RICEOK  RICEPEP RICEBBLS  RICECAS
| 2 |        C                      -57.5    -57.5    -57.5    -57.5
| 3 |    MLJAN        L       50        6        6        6        6
| 4 |    MLFEB        L       50        8        8        8        8
| 5 |    MLMAL        L       50        5        5        5        5
| 6 |    MLAPR        L       50       11       11       11       11
| 7 |    MLMAY        L       50       10       10       10       10
| 8 |    MLJUN        L       50       10       10       10       10
| 9 |    MLJUL        L       50
|10 |    MLAUG        L       50
|11 |    MLSEP        L       50        1        1        1        1
|12 |    MLOCT        L       50        2        2        2        2
|13 |    MLNOV        L       50        2        2        2        2
|14 |    MLDEC        L       50
|15 |    FLJAN        L       50        2        2        2        2
|16 |    FLFEB        L       50
|17 |    FLMAR        L       50
|18 |    FLAPR        L       50        5        5        5        5
|19 |    FLMAY        L       50        4        4        4        4
|20 |    FLJUN        L       50        3        3        3        3
|21 |    FLJUL        L       50        7        7        7        7
|22 |    FLAUG        L       50        6        6        6        6
|23 |    FLSEP        L       50        2        2        2        2
|24 |    FLOCT        L       50        8        8        8        8
|25 |    FLNOV        L       50        6        6        6        6
|26 |    FLDEC        L       50        6        6        6        6
|27 |     LAND        L       10        1        1        1        1
|28 |  CAPITAL        L      170    28.75    28.75    28.75    28.75
|29 |  RICETRS        L               -960     -960     -960     -960
|30 |  OKRATRS        L               -100
|31 |  PEPTRAS        L                        -150
|32 |  BBLSTRS        L                                 -100
|33 |  CASSTRS        L                                         -3000
|34 | COCOATRS        L
|35 | COFFETRS        L
|36 | SCANETRS        L
|37 |   PALMTR        L
|38 |     POTR        L
|39 |    POMAX        L       31
|40 |   RUBTRS        L
|41 |  GVTBMAX        L     1000
|42 |   CASMAX        L      351                                  3000
|43 |   RICONS        G     1880
```

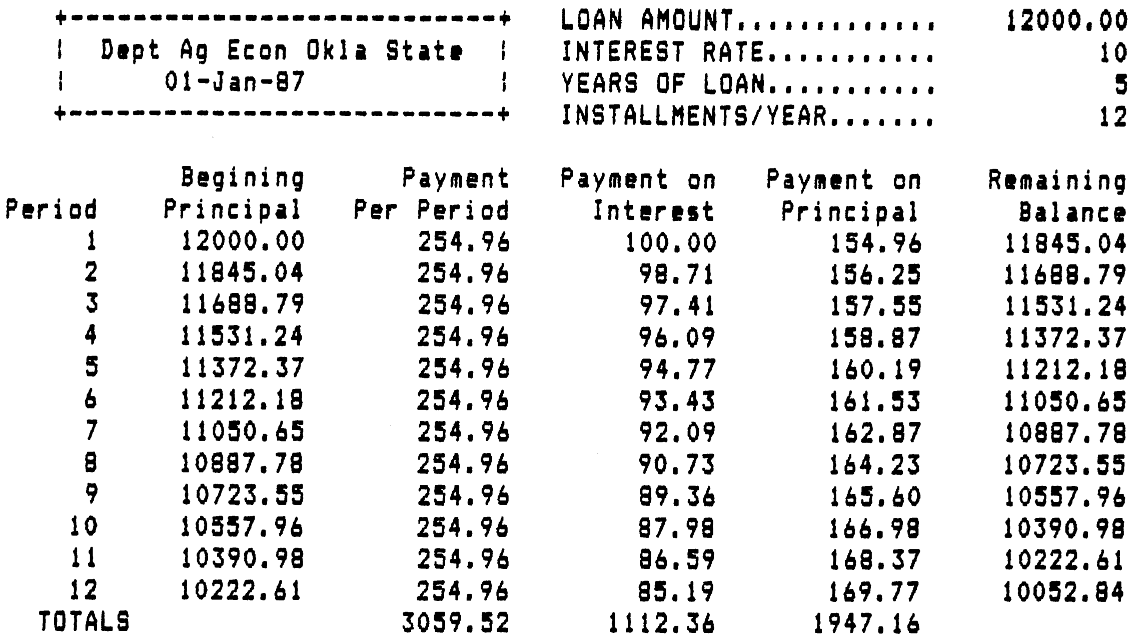
Figure 3. Linear Programming Matrix Design with a Spreadsheet.



MONTHLY RICE STOCKS AND FLOWS

MAY 84	OPENING STOCK	INFLOW	OUTFLOW	CLOSING STOCK
		(ARRIVAL)	(SALES)	
1 IMPORTED RICE TOTAL	41,464,500	26,009,300	16,679,100	50,794,700
a) PL 480	38,964,500	13,923,000	6,292,800	46,594,700
b) COMMERCIAL	2,500,000	7,385,400	5,685,400	4,200,000
c) CONCESSIONS	0	4,700,900	4,700,900	0
		(PURCHASED)	(PROCESSED)	
2 LOCAL PADDY	47,590,496	459,894	1,025,466	47,024,924
milled equivalent				
current (C)	26,814,920	259,128	577,800	26,496,248
long term(LT)	26,174,773	252,942	564,006	25,863,708
		(PRODUCTION)	(SALES)	
3 RICE MILLED AT LPMC	677,200	577,800	663,500	591,500
4 TOTAL (incl milled equ LT)	68,316,473	26,840,042	17,906,606	77,249,908
TOTAL (incl milled equ C)	68,956,620	26,846,228	17,920,400	77,882,448
5 MILLING FACTOR a)LT	0.550	0.550	0.550	0.550
b)C	0.563	0.563	0.563	0.563
6 MILLED RICE AVAILABLE FOR IMMEDIATE CONSUMPTION	42,141,700	26,587,100	17,342,600	51,386,200

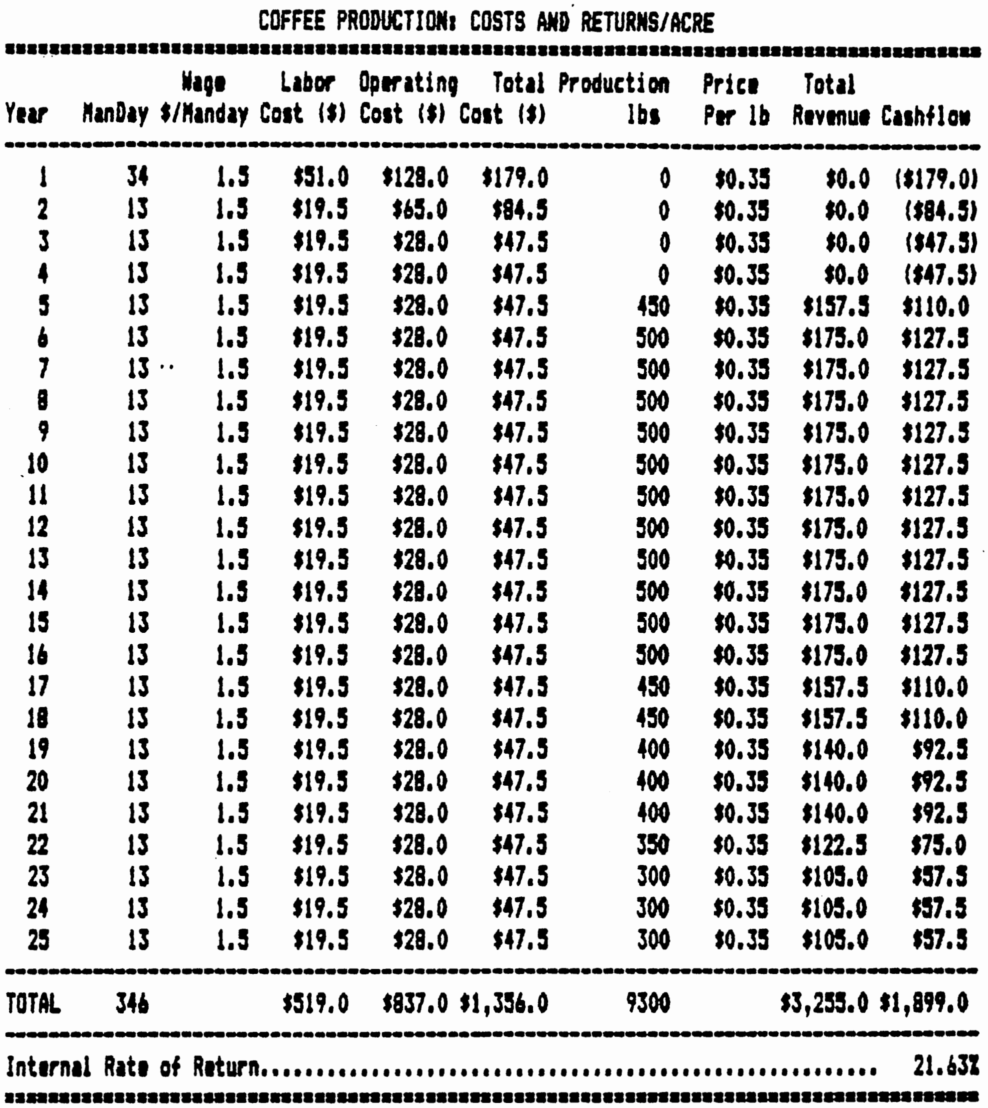
Figure 4. Record Keeping with a Spreadsheet.



```
+--------------------------+  LOAN AMOUNT.............     12000.00
|  Dept Ag Econ Okla State |  INTEREST RATE...........           10
|      01-Jan-87           |  YEARS OF LOAN...........            5
+--------------------------+  INSTALLMENTS/YEAR.......           12

          Begining     Payment  Payment on  Payment on    Remaining
Period   Principal  Per Period    Interest   Principal      Balance
     1    12000.00      254.96      100.00      154.96     11845.04
     2    11845.04      254.96       98.71      156.25     11688.79
     3    11688.79      254.96       97.41      157.55     11531.24
     4    11531.24      254.96       96.09      158.87     11372.37
     5    11372.37      254.96       94.77      160.19     11212.18
     6    11212.18      254.96       93.43      161.53     11050.65
     7    11050.65      254.96       92.09      162.87     10887.78
     8    10887.78      254.96       90.73      164.23     10723.55
     9    10723.55      254.96       89.36      165.60     10557.96
    10    10557.96      254.96       87.98      166.98     10390.98
    11    10390.98      254.96       86.59      168.37     10222.61
    12    10222.61      254.96       85.19      169.77     10052.84
  TOTALS               3059.52     1112.36     1947.16
```

Figure 5. Loan Amortization with Spreadsheets.

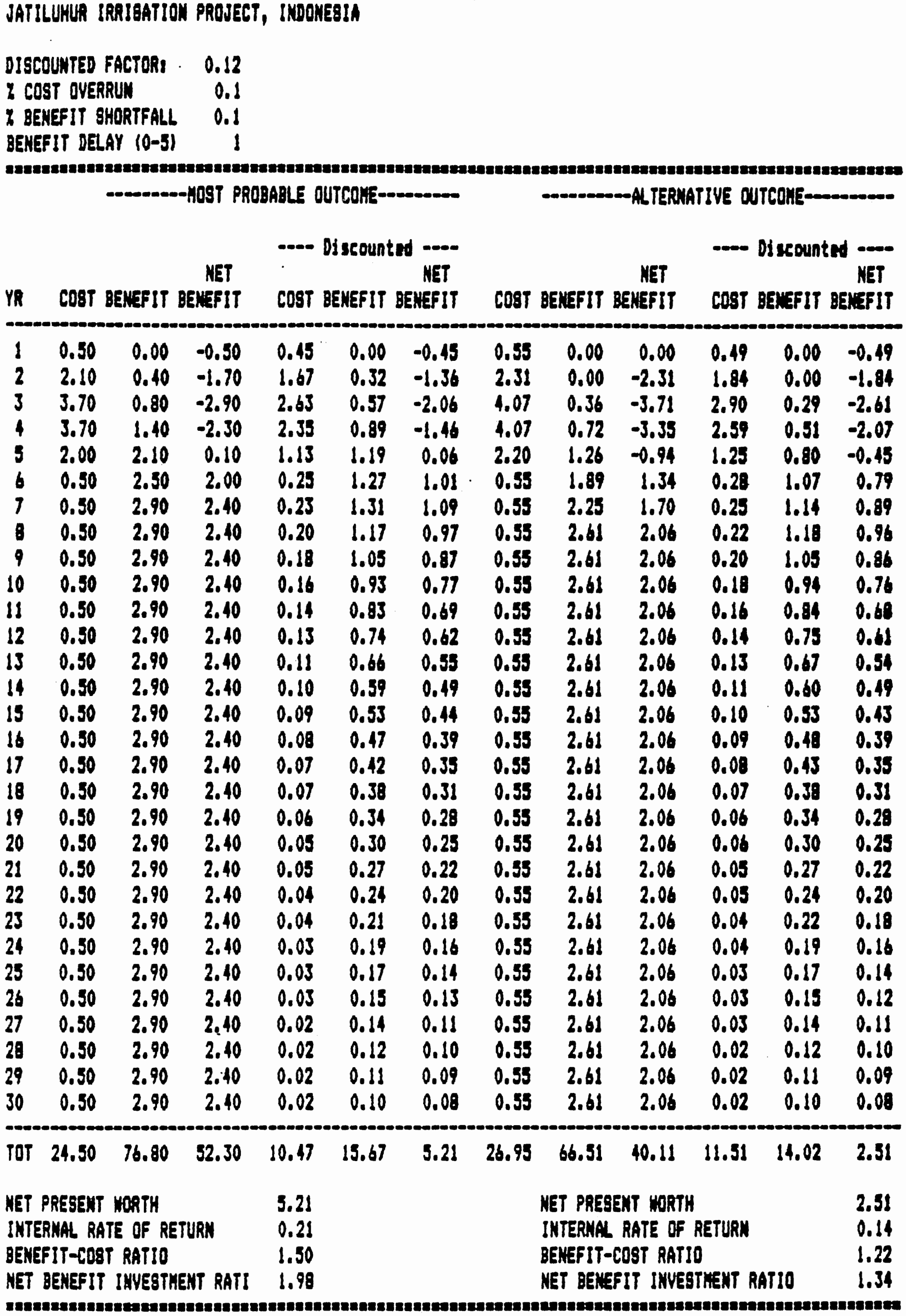


COFFEE PRODUCTION: COSTS AND RETURNS/ACRE

Year	ManDay	Wage $/Manday	Labor Cost ($)	Operating Cost ($)	Total Cost ($)	Production lbs	Price Per lb	Total Revenue	Cashflow
1	34	1.5	$51.0	$128.0	$179.0	0	$0.35	$0.0	($179.0)
2	13	1.5	$19.5	$65.0	$84.5	0	$0.35	$0.0	($84.5)
3	13	1.5	$19.5	$28.0	$47.5	0	$0.35	$0.0	($47.5)
4	13	1.5	$19.5	$28.0	$47.5	0	$0.35	$0.0	($47.5)
5	13	1.5	$19.5	$28.0	$47.5	450	$0.35	$157.5	$110.0
6	13	1.5	$19.5	$28.0	$47.5	500	$0.35	$175.0	$127.5
7	13	1.5	$19.5	$28.0	$47.5	500	$0.35	$175.0	$127.5
8	13	1.5	$19.5	$28.0	$47.5	500	$0.35	$175.0	$127.5
9	13	1.5	$19.5	$28.0	$47.5	500	$0.35	$175.0	$127.5
10	13	1.5	$19.5	$28.0	$47.5	500	$0.35	$175.0	$127.5
11	13	1.5	$19.5	$28.0	$47.5	500	$0.35	$175.0	$127.5
12	13	1.5	$19.5	$28.0	$47.5	500	$0.35	$175.0	$127.5
13	13	1.5	$19.5	$28.0	$47.5	500	$0.35	$175.0	$127.5
14	13	1.5	$19.5	$28.0	$47.5	500	$0.35	$175.0	$127.5
15	13	1.5	$19.5	$28.0	$47.5	500	$0.35	$175.0	$127.5
16	13	1.5	$19.5	$28.0	$47.5	500	$0.35	$175.0	$127.5
17	13	1.5	$19.5	$28.0	$47.5	450	$0.35	$157.5	$110.0
18	13	1.5	$19.5	$28.0	$47.5	450	$0.35	$157.5	$110.0
19	13	1.5	$19.5	$28.0	$47.5	400	$0.35	$140.0	$92.5
20	13	1.5	$19.5	$28.0	$47.5	400	$0.35	$140.0	$92.5
21	13	1.5	$19.5	$28.0	$47.5	400	$0.35	$140.0	$92.5
22	13	1.5	$19.5	$28.0	$47.5	350	$0.35	$122.5	$75.0
23	13	1.5	$19.5	$28.0	$47.5	300	$0.35	$105.0	$57.5
24	13	1.5	$19.5	$28.0	$47.5	300	$0.35	$105.0	$57.5
25	13	1.5	$19.5	$28.0	$47.5	300	$0.35	$105.0	$57.5
TOTAL	346		$519.0	$837.0	$1,356.0	9300		$3,255.0	$1,899.0

Internal Rate of Return.. 21.63%

Figure 6. Costs and Returns of Coffee Production.



```
JATILUHUR IRRIGATION PROJECT, INDONESIA

DISCOUNTED FACTOR:     0.12
% COST OVERRUN          0.1
% BENEFIT SHORTFALL     0.1
BENEFIT DELAY (0-5)       1
==================================================================================================
         ---------MOST PROBABLE OUTCOME---------          ----------ALTERNATIVE OUTCOME----------

                              ---- Discounted ----                            ---- Discounted ----
                   NET                         NET                     NET                     NET
YR   COST BENEFIT BENEFIT   COST BENEFIT BENEFIT   COST BENEFIT BENEFIT   COST BENEFIT BENEFIT
--------------------------------------------------------------------------------------------------
 1   0.50    0.00   -0.50   0.45    0.00   -0.45   0.55    0.00    0.00   0.49    0.00   -0.49
 2   2.10    0.40   -1.70   1.67    0.32   -1.36   2.31    0.00   -2.31   1.84    0.00   -1.84
 3   3.70    0.80   -2.90   2.63    0.57   -2.06   4.07    0.36   -3.71   2.90    0.29   -2.61
 4   3.70    1.40   -2.30   2.35    0.89   -1.46   4.07    0.72   -3.35   2.59    0.51   -2.07
 5   2.00    2.10    0.10   1.13    1.19    0.06   2.20    1.26   -0.94   1.25    0.80   -0.45
 6   0.50    2.50    2.00   0.25    1.27    1.01   0.55    1.89    1.34   0.28    1.07    0.79
 7   0.50    2.90    2.40   0.23    1.31    1.09   0.55    2.25    1.70   0.25    1.14    0.89
 8   0.50    2.90    2.40   0.20    1.17    0.97   0.55    2.61    2.06   0.22    1.18    0.96
 9   0.50    2.90    2.40   0.18    1.05    0.87   0.55    2.61    2.06   0.20    1.05    0.86
10   0.50    2.90    2.40   0.16    0.93    0.77   0.55    2.61    2.06   0.18    0.94    0.76
11   0.50    2.90    2.40   0.14    0.83    0.69   0.55    2.61    2.06   0.16    0.84    0.68
12   0.50    2.90    2.40   0.13    0.74    0.62   0.55    2.61    2.06   0.14    0.75    0.61
13   0.50    2.90    2.40   0.11    0.66    0.55   0.55    2.61    2.06   0.13    0.67    0.54
14   0.50    2.90    2.40   0.10    0.59    0.49   0.55    2.61    2.06   0.11    0.60    0.49
15   0.50    2.90    2.40   0.09    0.53    0.44   0.55    2.61    2.06   0.10    0.53    0.43
16   0.50    2.90    2.40   0.08    0.47    0.39   0.55    2.61    2.06   0.09    0.48    0.39
17   0.50    2.90    2.40   0.07    0.42    0.35   0.55    2.61    2.06   0.08    0.43    0.35
18   0.50    2.90    2.40   0.07    0.38    0.31   0.55    2.61    2.06   0.07    0.38    0.31
19   0.50    2.90    2.40   0.06    0.34    0.28   0.55    2.61    2.06   0.06    0.34    0.28
20   0.50    2.90    2.40   0.05    0.30    0.25   0.55    2.61    2.06   0.06    0.30    0.25
21   0.50    2.90    2.40   0.05    0.27    0.22   0.55    2.61    2.06   0.05    0.27    0.22
22   0.50    2.90    2.40   0.04    0.24    0.20   0.55    2.61    2.06   0.05    0.24    0.20
23   0.50    2.90    2.40   0.04    0.21    0.18   0.55    2.61    2.06   0.04    0.22    0.18
24   0.50    2.90    2.40   0.03    0.19    0.16   0.55    2.61    2.06   0.04    0.19    0.16
25   0.50    2.90    2.40   0.03    0.17    0.14   0.55    2.61    2.06   0.03    0.17    0.14
26   0.50    2.90    2.40   0.03    0.15    0.13   0.55    2.61    2.06   0.03    0.15    0.12
27   0.50    2.90    2.40   0.02    0.14    0.11   0.55    2.61    2.06   0.03    0.14    0.11
28   0.50    2.90    2.40   0.02    0.12    0.10   0.55    2.61    2.06   0.02    0.12    0.10
29   0.50    2.90    2.40   0.02    0.11    0.09   0.55    2.61    2.06   0.02    0.11    0.09
30   0.50    2.90    2.40   0.02    0.10    0.08   0.55    2.61    2.06   0.02    0.10    0.08
--------------------------------------------------------------------------------------------------
TOT 24.50   76.80   52.30  10.47   15.67    5.21  26.95   66.51   40.11  11.51   14.02    2.51

NET PRESENT WORTH            5.21                NET PRESENT WORTH                    2.51
INTERNAL RATE OF RETURN      0.21                INTERNAL RATE OF RETURN              0.14
BENEFIT-COST RATIO           1.50                BENEFIT-COST RATIO                   1.22
NET BENEFIT INVESTMENT RATI  1.98                NET BENEFIT INVESTMENT RATIO         1.34
==================================================================================================
```

Figure 7. Applications in Project Appraisal.

CHAPTER III

BACKGROUND FOR POLICY ANALYSIS

Introduction

With or without computers, successful application of idealized quantitative techniques and economic theories to real-life economic problems in a form usable by decision makers is both a science and an art. The ultimate objective of policy is the optimal attainment of goals by groups (including the society as a whole). This chapter provides some background on the use of the positive science of economics as a scientific critique of policy decisions.

The Function of an Economic System

An economic system must simultaneously perform five closely related functions: organize production, distribute products efficiently for consumption, determine what to produce, provide a mechanism for rationing products in the very short run, and properly maintain and expand its productive capacity (Leftwich, 1979).

Efficient Organization of Production

Resources used in production are limited, versatile and can be combined in varying proportions to produce different commodities. Production organization is a technical consideration of optimal input use for a desired mix of output. Resource usage is said to be Pareto optimal when the same level of resource inputs cannot produce more of any one good without producing less of another. This requires the marginal rate of technical substitution (MRTS) of any one resource for any other resources (which measures the comparative contribution of each resource to a production process) be the same for all production processes for which these resources can be used.

Efficient Distribution of Output for Consumption

The concept of Pareto optimality is also applicable in efficient distribution of output for consumption. Products yield utility or satisfaction when consumed. An output distribution is said to be Pareto optimal if no one's satisfaction can be raised without reducing the satisfaction of another. Pareto optimality in consumption thus requires the relative satisfaction of an additional unit of any good as measured by the marginal rate of substitution (MRTS) be the same among individuals. Otherwise, incentive for trade exists. Unless restrained, individuals trade what each feels is relatively less important for what each considers will yield more satisfaction to increase utility until a Pareto optimality is reached.

Determining What to Produce

Determining what to produce involves selecting from a collection of product mixes which are Pareto optimal both in production and consumption, one that maximizes the welfare or utility of the economy. This requires that the subjective value (or utility) of consuming an additional unit of good x in relation to that of any other good y be equal to the opportunity cost of producing an extra unit of x instead of

y. If this subjective value exceeds the opportunity cost, then incentive to produce the additional unit exists since the relative cost of producing this additional unit is more than justified by the relative value of this unit to the consumers in terms of utility creation. In other words, the marginal rate of substitution (MRS) of any one product for any other product must be the same as the marginal rate of transformation (MRS) of the products. Any deviation from this equality indicates that an alternative feasible product-mix yields a higher satisfaction level for the society.

Rationing in the Very Short Run

An economic system must make provision for rationing commodities over the market period or the very short run when supplies cannot be changed. For instance, supply of agricultural products harvested only once per year must be stretched in an orderly manner from one harvest period to the next.

Economic Maintenance and Growth

Economic growth is usually defined as secular increases in per capita real income. One necessary condition for growth in the economy is the proper maintenance and expansion of its productive capacity, using resources that could otherwise be used to produce goods for current consumption. An economic system should provide the mechanisms to (1) allocate the appropriate fraction of resources to investments that could otherwise be used to generate products for current consumption, (2) direct the investment of the allocation profitably, and (3) induce the necessary social transformation, in consistence with the society's growth requirements or objectives.

The Market Price System

A Purely Price Competitive Model

The competitive market price system assumes impersonal competition based on price alone in all resource and product markets. Buyers and sellers of each homogeneous product are assumed too small to influence the price bidding process. Resources and products are perfectly divisible and free to be moved to more profitable uses. No artificial restraint is put on price levels and trading activities. Prices are therefore left to perform their functions as resource and output allocators.

In any market, the market price serves as a rationing device for buyers and as a profit motive for sellers. Too high a price induces excessive production and inhibits the incentive to buy. The resulting surplus depresses price. The lowered price motivates reduction in production and increase in purchases putting upward pressure on price. This price oscillation stabilizes to an equilibrium level when neither a surplus nor shortage exists in a market. At equilibrium price, the quantities of the good producers wish to supply coincides with the amount buyers demand. No incentive for change exists. The market is said to be in equilibrium.

Partial and General Equilibrium

General equilibrium of the economy is said be attained when all markets have simultaneously reached their own (partial) equilibrium.

Allocation of resources and output of the economy is complete until further disturbed.

Any economic disturbance to the system first impacts one or a few markets. Some of these markets may quickly return to an equilibrium -- called a partial equilibrium because not all markets are without incentives for more adjustments. Due to the interrelationship among markets, the first round adjustments to new partial equilibrium positions dislocate the economy from its old general equilibrium. Movement to a new general equilibrium in turn requires further adjustments in partial equilibrium positions. The adjustment process finally gravitates to a new general equilibrium.

Competitive General Equilibrium and Pareto Optimality

When perfect competition prevails in the economy, the price system leads to a general equilibrium which is Pareto optimal in consumption and production.

The size insignificance of each buyer or seller implies the absence of monopoly or monopsony. Resource price ratios then are true indexes of marginal productivities. Profit maximizing firms equate the ratio of marginal productivities of two resources in each productive process these resources can be used to the ratio of their prices. This fulfills the Pareto conditions of equal marginal rates of technical substitution between any two resources in all production processes.

Similarly, the free market price ratio of any two products is an index of their relative marginal utilities. A utility maximizing individual consumes each product until the ratios of marginal utilities equal the price ratios. The existence of only one set of prices implies the marginal rate of substitution between two goods are the same for all individuals -- the condition of Pareto optimality in consumption.

Simultaneous with efficient production and consumption organization, the perfectly competitive market price system also determines the mix of products to be produced. Facing the same set of product prices, the profit maximizing firms and the utility maximizing consumers respectively equate the relative marginal opportunity cost of two products (marginal rate of transformation) and the relative marginal benefit of consumption of the two products (marginal rate of substitution) to the products' price ratios. All resources are used appropriately in their ultimate role of utility creation.

Consequences of the Price System

Mathematical derivations of the existence and implications of competitive equilibrium involve concepts of point set theory and fixed point theorems (Hildenbrand and Kirman, 1976, Nikaido, 1970) but in logical essence a reinforcement of the classical belief in the efficiency of competition as a mechanism for allocating resources and output in production and consumption. Each individual, as Adam Smith described in The Wealth of Nations, "intends only his own gain, but is in this ... led by an invisible hand to promote an end which was no part of his intentions." Under the usual neoclassical assumptions, research work in general equilibrium theory arrives at the following conclusions (Quirk and Saposnik, 1968):

1. There exists more than one set of allocations (or prices) in the economy where Pareto optimality in both consumption and production are attained. That is, Pareto optimality positions are not unique.

2. Whereas individuals can always move from a non-Pareto optimal position to an optimal one by cooperation, there are other Pareto optimal positions not reachable by cooperation for a particular initial resource endowment and income distribution.

3. For any given resource endowment and income distribution, competitive equilibrium exists and necessarily leads to Pareto optimality both in consumption and production. The competitive equilibrium position is unique if slightly more restrictive (but plausible) postulates can be made.

4. For a given resource endowment and income distribution, there are Pareto optimality positions not reachable by the competitive price system. Other ways to reach Pareto optimal positions are possible. Indeed, Pareto optimality can be attained even if monopoly and monopsony exist in the economy or when the economy is centrally planned. But when the economy's resource endowment and/or income distribution are allowed to vary accordingly, then any Pareto optimality position can be reached by the competitive price system.

Critique of the Price System

The theory of general equilibrium is positivistic. Its implications arrive inescapably and indisputably as purely logical and mathematical realities. Critique of the price system can come in two forms. The first form accepts the postulates but questions the adequacy of the price system in fulfilling the function of an economic system. The second form questions the realism of the postulates themselves.

The Adequacy of the Price System

Based on of the implications of the competitive price system and the function of an economic system, the ability of the price system to achieve production and consumption efficiency is unquestionable in theory. Pareto optimality is reached. The fact that the particular Pareto optimality is reached by perfect competition without interference by 'authority' adds to its appeal as socially acceptable allocation. But some may be starving while others are satiated with goods and services under a Pareto optimum.

Indeed, competitive equilibrium is 'ideal' in the sense that resources are allocated to uses such that the marginal opportunity cost is justified by the marginal utility or benefit of the product. In a national economy, this implies national income is maximized subjected to the initial distribution of resources (Silberberg, 1978). If welfare of the economy depends on the size of the national income alone, then the competitive price system undoubtedly fulfills the third function of an economic system: it picks among the set of efficient allocations the one that maximizes welfare or utility.

This utility maximization position, however, is qualified: it is the utility maximization only for a given initial resource endowment and income distribution in the economy. Moreover, if the society's welfare is based on more than the size of the income alone, or equivalently, if the marginal utility of money differs among individuals, then the contention that this position maximizes utility is further objectionable. Maximization of total income in this case is neither necessary nor sufficient for utility maximization.

The price system, however, can lead to maximum utility if the society is willing to modify its resource endowment and distribution of income. Ironically, such endeavors in themselves impair the workings of the price system and can initially lead to an sub-Pareto optimal position. Nevertheless, with the right policies, this sub-Pareto optimal position can be one that yields higher utility than the former position. Further improvement in utility should always be possible if the price system is again left to lead the economy to a new Pareto optimality.

Realism of the Price System Postulates

The assumptions of the price system are seldom met in practice. Most resources are lumpy in nature and not perfectly divisible thus preventing the marginal conditions for optimality from being met. Each buyer and each seller in most markets are seldom insignificant in size. Influential sellers pursuing profits can adjust their price to do so. The invisible hand in this case cannot prevent misaligned profits and costs. Most products vary in quality, few are homogeneous. Much price behavior and consequences can be explained only when the non-homogeneity of products are taken into account.

Moreover, the assumption of perfect knowledge of input and output relationships and prices is obviously not realistic, especially for subsistence farmers in developing countries. Aversion to risk can cause individuals to accept sub-optimal positions. Violations of the assumptions of the price system is especially magnified when public goods are considered and when costs and benefits are measured in social rather than private terms (Tweeten, 1980).

Measurements of Welfare and Objective in Economics

Classical Economics and Utilitarism

The search for meaningful measurements of economic status or welfare is almost as old as economics itself. Classical economists, typically accepting the utilitarian moral philosophy, spoke of "utils" as an cardinal measure of satisfaction. The implicit assumption of cardinality means that the consumer not only can rank his/her preference, but can also assign a meaningful absolute index to his/her level of satisfaction. The index is meaningful and absolute in the sense that interpersonal comparison is valid. If A claims that he derives 4 utils from consuming 1 unit of some commodity and B claims that she derives 8 utils from consuming a unit of the same commodity, then the commodity is worth more to B than to A and by two times. The utility scale is assumed to be unique for one and all individuals. Any action which could increases the utility of a society (i.e. sum of individual utilities) is a necessary and sufficient condition for its approval.

Rejection of Cardinal Utility and the Pareto Criterion

The Neoclassical economists, being more concerned with efficiency than equity of allocations, and perhaps excessively reacting to the then new-found scientific status of economics, frowned on any value judgment by an economic analyst including interpersonal comparison of utility. Jevons explicitly suggested that, as far as he could see, no meaning could be attached to comparisons between the utility experienced by one man and that experienced by another. These were states of mind and, in Jevons' opinion, forever inscrutable (Walsh, 1970, p. 95).

Neo-classical theory typically regards utility measures as ordinal. Individuals are assumed capable of ranking preferences in a consistent manner and assigning higher ranking for a better preferred bundle. The fact that 5 utils is assigned to bundle A and 10 utils for bundle B means bundle B is preferred to bundle A but does not imply the preference for bundle B is twice that of A. Any other scale which assigns a higher numerical value to bundle B would suffice as well. The utility scale is thus not unique for each individual or among individuals. The non-uniqueness of utility scales necessarily invalidates interpersonal comparisons. Recommendations can only be made for policies which make some people better off without making anybody else worse off. No statement can be made of policies which make even just one individual worse off by, say, taking a dollar from him even if it clearly improves the well-being of a million others. For the ordinalist, whether the loss of one dollar's worth of utility of the individual is justified by the benefits of million of others involves value judgments outside the realm of science.

Some economists, notably Allen, Robbins and Hicks (in his earlier works) even reject the very concept of utility itself and consider its usage, implicit or explicit, an unnecessary acceptance of the utilitarian philosophy. In Value and Capital, Hicks (1957, p. 18) wrote: "If one is a utilitarian in philosophy, one has a perfect right to be a utilitarian in one's economics. But if one is not (and few people are utilitarians nowadays), one also has the right to an economics free of utilitarian assumption." According to him, maximization of utility, cardinal or ordinal, is a utilitarian assumption neither appropriate nor necessary for explaining market behavior. Thus the principle of Occam's razor alone is strong enough justification for bypassing the assumption of utility maximization and the use of indifference curves as a starting point for his analysis instead.

This rejection of cardinal utility, among other things, invalidates any meaning to the concept of marginal utility: if total utility is arbitrary, so is marginal utility (Hicks, 1957, p. 19). In particular (and more seriously), the principle of diminishing marginal utility is threatened: if marginal utility has no exact sense, diminishing marginal utility can have no exact sense either (Hicks, 1957, p 20). Only ratios of marginal utilities can have precise meaning. As a result. the principle of diminishing marginal utility is replaced by the (weaker) principle of diminishing marginal rate of substitution in "modern" demand theory. More importantly, an acceptable welfare theory must not only be void of all interpersonal comparison of utility, but must not even utter the term "utility" itself. A logical consequence of this stance is that policy recommendations must only resort to the Pareto criterion.

Need for Value Axioms

Whereas the Pareto criterion is an indisputably elegant piece of pure science of choice theory, some feel that it is useless in generating policy recommendations unless some value judgment, such as the acceptance of utilitarism, is injected as axioms, according to Vivian Walsh (1970):

> ...a successful welfare theory should, in certain logical respects, resemble a sausage. To get sausages you must feed sausage meat into a sausage machine. Even if you have the most perfect, the most efficient, and the most elegant sausage machine in the world, you will not get sausages out of it unless you put sausage meat in... A system of welfare theory is based either explicitly or implicitly on an axiom system. If it is a contemporary welfare theory, it is likely to be based, like most recent economy theory, on an explicit system... If this axiom system includes no value axioms, whether explicit or implicit, the so-called welfare theory will not generate results that contain any rich welfare recommendations. It will simply repeat the results of pure choice theory, refurbished and offered in a welfare theoretical language that makes them sound as if in fact they were rich policy recommendations, which, of course, they are not and cannot be. The meat of a welfare theory consists of the information its gives about how some people could be made better off in some sense, which cannot be done unless some assumption is made initially as to what constitutes 'better off.' (p. 97-98)

The basic argument is that some value judgment are, in fact, not so difficult to make. An example is that an additional dollar of income is obviously more important to a subsistence family than to a millionaire. To further illustrate some value judgments which, according to him, are not difficult, he went on to say:

> There are many places in the world where most of the children who are born simply died of malnutrition. It is not a daring moral hypothesis to suggest that it would be a better world if they lived. (Walsh, 1970, p. 99)

Indeed, implicit in even the most "objective" economics is the assumption of the validity of income, and income alone, as a welfare measure. Higher income results in higher welfare -- a direct implication from the model of an economic man who prefers more to less. If a society can exclusively be divided into two groups A and B, then any policy which increases either group's income without diminishing that of the other group is worth undertaking according to the Pareto criterion. Also implicitly accepted is the independence axiom: the satisfaction of one group should be independent of income of the other group.

But the Pareto criterion is not functional if any losers are involved; even in the trivial situation where millions benefit by millions of dollars at the cost of one dollar to one individual. Needless to say, scenarios where the Pareto criterion is applicable are rare since most policy involves at least some cost to taxpayers -- someone is made worse off by a policy to benefit society.

Recognizing the infertility of the Pareto criterion as a tool for policy analysis, proponents of the "new welfare economics" resorted to the compensation principle. The doctrine can be stated wholly in indifference curve terms, without even mentioning the notions of quantities of utility. Simply stated, a policy is worth undertaking if winners can potentially compensate losers. In indifference curve terms, if the gainers can compensate the losers by offering them something to move them back to their previous indifference curve and still themselves stay at a higher indifference curve than before, then the economic change can be described as an "increase in welfare". Nothing is said here about quantities of satisfaction, and more importantly, no interpersonal comparisons of utility have been made.

But let us examine whether the compensation principle is as innocent in not making interpersonal comparisons as its proponents claim. Let society consist of two individual A and B and a policy results in them gaining 10,000 and losing 8,000 dollars per annum respectively. Clearly if A transfers 8,000 dollars to B he will still be 2,000 dollars ahead. The compensation principle is this case seem plausible -- this simplified society experienced an increase in welfare.

But if compensation is not actually carried out, then assessing the welfare change is impossible unless one is willing to resort to interpersonal comparisons. If B happens to be a subsistence farmer whereas A is a millionaire, then it is hard to accept that the welfare of the society had increased. One may claim that the conclusion does follow if we assume that one dollar is worth the same to every individual at the margin. This claim is interpersonal utility comparison in disguise, and extremely misleading not only because it is disguised, but also most likely erroneous.

Difficulties exist even if compensation is actually carried out. Seldom is the suitable amount of compensation as obvious as the simplified example given above, and the calculation of this suitable amount must often require interpersonal comparisons.

This argument of utility and objectivity in economics is still alive and well and the day when a general consensus emerges is hard to envision. Economic policy analysis is a rational scientific aid to decision making, albeit not coldly objective. A scientific approach requires the explication of one's concealed value postulates and behavioral assumptions and subjecting them to open criticism.

What Can Be Expected From Policy Analysis

According to Quade (1982, pp. 11-12), what one can expect from policy analysis should be rather modest. First, it can frequently reduce the complexity of problems to manageable proportions by identifying and clarifying those elements about which information exists or can be found. Second, it can eliminate from considerations the demonstrably inferior alternatives and sometimes find one that all interested parties can accept even though they are not fully satisfied. Third, it can counter the purely subjective approach on the part of advocates of a program by forcing them to defend their line of arguments and talk about the specifics of the situation rather than merely expressing their personal opinion with statements of noble purpose, thereby raising the quality of public discussion.

The major contribution of policy analysis is to yield insights, particularly with regard to the dominance and sensitivity of the parameters. It is no more than an adjunct, although a powerful one, to the judgment, intuition, and experience of decision-makers.

Summary

The objective of policy, as stated in the beginning of the chapter, is the optimal attainment of goals. The goal of policy analysis is to help a policy-maker make a better decision than he otherwise would have made. The price system is used as an idealized norm by which performance of an economic system is evaluated.

Not all goals can be expressed in monetary terms. Self-sufficiency, food security, preservation of family farming, or elimination of malnutrition and rural poverty, are sometimes perused at the expense of potential monetary gains. One role of policy analysis is to provide decision makers information on how well their goals are being accomplished by their policies. Another is to provide information on the economic consequences of alternative policies that influence the well-being of society.

CHAPTER IV

PROGRAMMING ON A MICROCOMPUTER: ISSUES AND CASE STUDY

Chapter Objective

The objective of this chapter is to discuss, and illustrate through a case study, some issues concerning computer programming with a microcomputer. The subject of the illustration is a computer program called Musah86 -- a linear programming package with an easy interface to a popular spreadsheet program. This linear programming package takes as input an LP matrix built with a spreadsheet in a format discussed in Example 2 in Chapter II, solves the LP problem, and outputs the solution in a form suitable for use with a spreadsheet. The operation of the package only requires few intuitive steps thus the package is suitable for analysts who are new to microcomputers or are casual users.

Introduction

Computer programming is not a necessary tool in a policy analyst's tool box. And programming per se, traditionally the basics of learning about computers, is not necessary either for learning or using microcomputers. Indeed much of the intention of the thesis is to demonstrate how little about computers one needs to know to produce something useful with a microcomputer. However, it is the tool-maker's careful and thoughtful design of tools which makes it simple for the user. Generally, simplicity at the user level is at the expense of complexity at the tool maker level.

Uses of computer programming can roughly be classified into two categories: as a direct tool and as a tool to develop other tools. If a series of random numbers is needed, and the analyst writes a simple computer program to generate and print the series, then computer programming is used as a direct tool. The computer program is not likely to be used by other than its author. On the other hand, developing application software such as a generalized package for linear programming, a LP matrix generator or report writer, or a simulation model of an economy that will be updated and reused are examples of using computer programming for development of other tools.

Computer programming is quickly diminishing in importance as a direct tool. Most situations formerly requiring knowledge of computer programming can now be handled by many special-purposed or general-purposed application packages and programs available for microcomputers. The discussion will henceforth concentrate on developing computer programs intended to be used by others. In particular, the users are targeted to be analysts with 'minimal' computer expertise as assumed throughout this thesis.

The essence of custom developing software is to tailor a program to specific needs. The goal is to transform the computer, the machine, into a metaphor suitable for handling the problem at hand. A metaphor that can be used as a tool for solution or model for understanding of a problem. Building a computer programming from the ground up using a programming language is but one means to this end. One should also consider the alternatives:

-- Buying an existing program.

-- Modifying or extending an existing program.

-- Using a general purpose software such as spreadsheet or database without resorting to programming per se.

-- Using a general purpose tool but supplementing it with some computer programming.

Evaluating the alternatives requires first the identification of needs, technical expertise in the subject matter, knowledge of computer and, in some alternatives, computer programming. All these skills need not be possessed by the same person.

The disadvantages of computer programming are its cost and complexity. Programming an economic simulator, say, on a computer requires not only understanding of the economic model, but also the correct communication of the model to the computer via a programming language. Correctness of a computer program is extremely difficult to verify. "Bugs" in computer programs are very subtle and can remain latent for a long time before showing up (otherwise they would not be "bugs"). A simple neglect of detail in the computer program can severely affect the precision and accuracy of the results. It is not sufficient to view the computer program as a black box and verify its correctness simply by looking at its output for given sets of input, since no testing can span the whole space of possible inputs. The complete verification process must also include opening up the "black box" and inspect whether it is correctly constructed inside.

Before microcomputers were generally available, computer programming with mainframe computers was the only resort when specialized simulation models, linear programming matrix generators, record systems, statistical procedures, management tools and operations research algorithms were required. Many of these applications can now be implemented on microcomputers by using higher level tools such as spreadsheets programs or database packages without programming with FORTRAN or BASIC. These tools are actually themselves powerful computer programs with prescribed sets of instructions allowing users to tailor their use to specific needs. The customization can be done easily but are, however, very limited compared to those attainable with computer programming.

Computer Programming

Computer programming is not as precise a term as one might think. The more modern concept of computer programming includes any description of the solution of a problem in a form intelligible to the computer -- the description would be a computer program, and the process of producing it would be "computer programming". This definition would regard spreadsheet design, and indeed, preparing documents with word processors as computer programming. The more traditional and narrower definition (the one used here) restricts the description of the the problem solution to the computer (i.e. the computer program) be in terms of not what needs to be done, but rather in terms of a well-defined sequence of instructions for the computer to follow. This sequence of instructions is the algorithm. Usually, a general-purpose higher-level computer language such as FORTRAN or BASIC is used to express the algorithm. The FORTRAN or BASIC program is then translated into machine language, i.e. the 1's and 0's which the computer can understand. This translation can be done in two ways. With the compiler method, the whole program is translated at once. If successful, the resulting machine code becomes independent of the compiler and can be used without further help by the compiler. With the interpreter method, the translation is done one statement at a time. The translated statement is then executed before another statement is processed. Unlike the compiler method, the interpreter controls not just the translation, but also the execution of the program and must be present every time the program is run. Compilers and interpreters are themselves computer programs. Microcomputer compilers and interpreters are available for many high-level languages including BASIC, Pascal, FORTRAN, Forth, Modula-2, C, PL/1, COBOL and Ada. On the more popular microcomputers such as the IBM PC series, the variety of computer languages and other programming tools available is actually better than for many mainframe computers or minicomputers.

Difference between Programming on the Microcomputer and Mainframe

Many of the differences between programming on the mainframe computer and microcomputers stem from the differences in the nature of the hardware. Inherently, microcomputer hardware is much weaker in raw processing power but allows more interactive communication with the user. The lack in raw processing power means that efficiency issues become more prominent when a program is running on a microcomputer. Often, algorithms and implementation strategies must be carefully selected for programs to run within the limited memory of the microcomputer at acceptable speed. Many time-efficiency issues also arise because of the need for interacting with the user in real-time, not in batch mode as is usually the case with mainframes. Overcoming these issues often requires the programmer to take direct control of the hardware and operating system resources beyond which is typically needed when programming on mainframes. Thus, whereas knowledge of the hardware and operating system is often not necessary to produce a "good" mainframe computer program, this knowledge is essential to produce "good" microcomputer software.

Whether a piece of software is good depends on the perception of the user. Mainframe software users are typically more computer literate than their microcomputer counterparts. When problems arise, an expert, who is available (and required) at almost all mainframe computer sites, can be consulted. This scenario is certainty not applicable for microcomputers. Microcomputer users typically are more computer naive and have limited or no access to experts. Thus a program which terminates abnormally with just a cryptic error code may be acceptable in a mainframe environment but unacceptable when used on a microcomputer. Thus an important quality of a microcomputer programmer is the ability to anticipate user error. When recovery cannot be made, the program should at least explain, in clear English, what the error is and how to avoid it. Anticipating and recovering from user error can be a great demand on the programmer's skill and resources. Often, the concern for a good user interface and graceful error recovery dominates the way the program is designed and accounts for the major part of the coding and programmers' effort.

Microcomputer as a Programming Environment

The programming environment on a microcomputer such as the IBM PC series is actually an improvement over those available on many mainframe or minicomputers. Mainframes typically constrain programmers with charge, time, and access restrictions. Consequently, programmers must often, due to necessity, consider the minimization of the number of trial-runs as the major design objective! This perhaps is the rationale for the old school which insists that programmers take the specifications, do the design, refine the design, then code the program and get it running with a only few trial-runs. The free access and interactive nature of the microcomputer environment instead encourages an exploratory or experimental style of computer programming where the programmer can have more freedom in trying out new ideas, and to fine-tune until the program not only works correctly, but also "feels" right.

Programming tools available on the micro has reached a very matured stage compared to just a few years ago and they are getting better. Many high quality program editors, interpreters, debuggers, compilers and assemblers are available on the market. These tools, like other microcomputer software, are more user-friendly and forgiving than their mainframe counterparts. There are also more to choose from and are available at affordable prices.

The tide has turned from the early days of the micro revolution when much microcomputer software was developed on the mainframe and then adapted to the micro. Many programmers now instead prefer to develop even mainframe programs on the micros. For the size of programming typically encounter in policy analysis, the microcomputer is not a restrictive programming environment and is in fact in many ways superior to mainframe computers.

A Case Study

A case study of developing customized application program on a microcomputer is now presented. It serves to demonstrate the design decisions and issues that go into implementing a medium-size program, in this case a programming package with an interface to a Lotus 1-2-3 spreadsheet (Li, 1984).

Setting the Scene

The users of this program are assumed to be analysts with very little experience on microcomputers and practical experience in linear programming. The objective is to encourage the use of linear programming in their work by providing them with the capability of solving linear programming on the microcomputer. Counter to this objective and discouraging microcomputer use would be a program which requires considerable learning time on the operation of the computer and the program.

The program would be used to demonstrate LP concepts and tableau design. As an important design objective, the amount of time needed to explain the operation of the program should be negligible, since a complex program would divert attention from the main point of the training -- LP principles and not the operation of any particular software package. Nevertheless, the program should have sufficient capability to handle problems of realistic sizes in order to encourage continual usage of the program and LP in actual policy work after the training. Thus, the program should accurately handle tableaus of size up to about 100 equations and 200 variables. Input procedures of the LP tableau must be intuitive and easy. Both the input and output of the program should as closely resemble a "text-book" style LP tableau.

Many commercial LP packages were reviewed. None, however, provided a satisfactory trade-off between simplicity and capability. Modifying an existing package was impossible because the better commercial packages are usually not released in source code form.

Through interviewing potential users, it was clear that the most intuitive procedure for inputting LP tableaus is through an electronic spreadsheet using a layout as presented in figure 3 in Chapter II. The best approach, then, was to provide some linkage between a widely-used electronic spreadsheet program such as Lotus 1-2-3 and an existent LP package. This linkage can be established in many ways, but to satisfy the objective of operation simplicity, the LP package should take a Lotus worksheet as layout in Figure 3 in Chapter II directly, solve the tableau, and output the solution and final tableau also into a Lotus 1-2-3 worksheet for examination or further manipulation. No existing LP package could be extended or supplemented to fulfill these requirements without asking users to perform awkward steps. Musah86 was conceived.

Choosing the Solution Algorithm and Implementation Strategy

The choice of an algorithm is the single most important factor that affects the performance of the program. The performance difference between good and poor choices of algorithms usually overwhelms the difference between good and poor implementation of an algorithm, or the difference between implementations by different languages. The appropriate choice depends on circumstances. In this particular circumstance, the following considerations were given:

1. The algorithm must be simple and easy to implement because of the limited time and manpower resources available.

2. It must be time efficient, i.e. it should solve the tableau within a tolerable amount of time.

3. It must be accurate.

4. The algorithm should perform 'reasonably' well even on machines that are only modestly equipped. For example, except for inputting and outputting the tableau, the algorithm should solve without any additional disk access to avoid slowing down the solution process and complicating the program's operation especially on machines without a hard disk.

5. It must use storage efficiently, so that large tableaus can be solved.

Several methods was considered: simplex, revised simplex, and dual simplex. The simplex method was chosen because it was judged to yield the best trade-off among the considerations listed above. In particular, the simplex method is simple to implement and is efficient in it use of storage since only one copy of the tableau needs to be stored. Thus complex storage management strategies, which complicate the program and slow down its execution, can be avoided.

The simplex method, however, tends to be slow and more vulnerable to the cumulation of round-off errors which affects accuracy. It was felt, however, that this should not create a problem for the size of problems under consideration and can be circumvented somewhat by exercising some care in the implementation phase.

At any rate, it was decided that the program should be designed in a modular manner so that in case the solution algorithm proved to be unsatisfactory, a new solution module can be substituted easily without affecting the rest of the program.

Choosing the Implementation Language

Several languages were considered. The language should be:

1. Easy to use, debug and provide good diagnostics on programming errors.

2. Perform arithmetic efficiently because the simplex algorithm (or any LP algorithm in general) is very computational intensive.

3. It must be easy to deal with the lower-level issues such as reading and decoding the Lotus 1-2-3 template, and error handling.

4. Have good readability to make it easier for other programmers who might maintain or extend the program.

5. Have quality compilers available.

Pascal was chosen because it provided the best trade-off among the considerations above and it was one of the languages the author was familiar with. The Pascal compiler used was developed by Borland International (1985). Realistically, in short-term programming projects such as this one (less than one man-month), the choice of programming language is usually a moot point, since most programmers have their favorite language and it is difficult to become familiar with another in short periods of time. In longer-term projects such as those requiring (say) more than one man year, there is more latitude in the choice of the right implementation language.

Program Commentary

A listing of Musah86 is included in Appendix A.

The main program is in line 1627 to 1659. It consists of calls to subprograms SetupInput, ReadTableau, SetupLpTableau, SetupOutput, OutputInitialTableau, SolveTableau, and OutputFinalTableau, in that order. Descriptive names are chosen for the subprograms to make their functions obvious.

The subprogram SetupInput (line 538) prompts the user for the file name of the Lotus 1-2-3 file in which the LP tableau is stored. It checks whether the file is on the diskette and makes sure that it is indeed a Lotus 1-2-3 file. A valid Lotus file starts with a header coded with 002064, this is checked in line 598. Much effort is put into ensuring graceful error recoveries, even for trivial situations such as when the user forgets to close the drive door or inserts an unformatted disk.

ReadLPTableau (Line 603) then reads the Lotus tableau, interpret it and transform it into an internal tableau ready for solution. Line 603 to 750 is the sections where the Lotus file is decoded.

SetupLPTableau (Line 754) sets up the initial LP tableau by adding slacks and artificial variables. Less-than, greater-than, and equality constraints are handled differently in each case by subprograms GreaterThan, EqualTo and LessThan respectively.

SetupOutput (Line 1050) prompts the user for an output file name and readies the file for output.

OutputInitialTableau (Line 1113) outputs the initial tableau into the Lotus file.

SolveTableau (Line 1325) solves the LP problem using a simplex method. The heart of the simplex algorithm spans line 1584 to 1618. Subprograms RowIn and ColumnOut are used to determine the incoming row and the outgoing column (hence the pivoted element). Using row elimination, the pivot element is turned to one whereas the rest of the element in the same column is turned into zero. For each iteration, subprogram UpdateScreen is called to display summary information concerning the iteration. This summary information includes the new value of the objective function, and the activities incoming and outgoing from the basis.

OutputFinalTableau (Line 1157) outputs the final tableau to the user specified Lotus file after the solution is reached.

More than 1600 lines of code were written; only about one fourth of which deals with actually solving the tableau. Much of the rest of the code deals with decoding the input tableau read from Lotus and encoding the output tableau to output as a Lotus files.

Afterthoughts on Musah86

Extensive use of Musah86 by a wide variety of people has revealed some good and bad judgments in the design phase. The idea of using an electronic spreadsheet as input and output device is much praised. However, in hindsight, a more complex but faster algorithm should have been chosen. This would allow bigger tableaus to be solved at a more 'reasonable' amount of time and at higher accuracy. Although the performance of the current implementation is by no mean unrespectable --

a 50 by 60 matrix requires less than two minutes to solve -- users quickly upward adjust their definition of 'reasonable'.

The decision to output the solution in a 'text-book' style final tableau is much welcomed by users who use the program for educational purposes. For actual policy work, an option should have been provided to suppress some of the output. Parametric programming and range analysis should have been implemented.

Coding the algorithm with a generally more efficient language such as the C programming language instead of Pascal would potentially have speeded up the program by about 30 percent even if the algorithm remains unchanged.

Chapter Summary

Some issues on programming the microcomputer was discussed, illustrated by a case study of a medium size program. Like the rest of the thesis, this chapter is about simple tools for policy analysts. The chapter, however, emphasized the point that simple tools are more difficult to build.

All things equal, programming on the microcomputer is actually easier than on the mainframe. The difference lies in user expectations and levels of their computer literacy. The microcomputer programmer usually must work harder than a mainframe programmer to deliver a program his users can be comfortable with.

CHAPTER V

ON SHORT-TERM MICROCOMPUTER TRAINING FOR POLICY ANALYSTS IN DEVELOPING COUNTRIES

This chapter suggests a format and approach for providing microcomputer training for policy analysts. The discussion is relevant for training sessions of about 3 to 6 weeks, and small group formats, about 6 to 15, offered to policy professionals with little prior training in microcomputers. The subject matter is application of microcomputers for policy analysis in developing countries. The chapter is in fact a summary of experiences on actual short courses on microcomputer applications for policy analysts from developing countries, conducted as part of the thesis research.

Why training?

Why require training when microcomputers, according to claims of hardware and software vendors, are designed to need no special training; and that many successful users of microcomputers are self taught? Usually, novices find microcomputers overwhelming. Understandably, it is not a static discipline like classical Newtonian mechanics or basic economic theory. Microcomputer hardware is changing rapidly, and software even faster. This rapid change creates a source of confusion and a stumbling block for learning. Even finding out what to learn can be confusing. The appropriate amount and type of computer knowledge depend on the types of analysis an analyst usually handles. And even if the suitable type and amount of computer knowledge are identified, the cost of attaining such knowledge can be prohibitive if a trial-and-error approach is used. Training is therefore offered to the analysts as a cost and time effective way for overcoming these learning stumbling blocks, and to initiate a learning process for fostering the necessary confidence and skill to apply microcomputers productively in daily analysis.

Participants' own learning objectives vary, often expressed in terms of desired topics. Representative samples are:

1. How to use microcomputers in one's daily work.

2. Microcomputer fundamentals and terminologies.

3. Suitable hardware setup for one's working environment.

4. Types of software appropriate for one's work.

5. Physical operation of microcomputer including disk operating system.

6. Operations of specific application software packages.

7. Examples of situations where a particular software package can be applied.

8. Hardware and software compatibility issues.

9. How to perform specific quantitative or economic modeling techniques with microcomputers.

10. Explanation of the quantitative technique and/or economic model being applied.

11. Examples of different types of policy analysis using microcomputers.

Learning objectives are more specific and technical for participants' with more familiarity of microcomputers, and change as their experiences grow. Objectives (1), (2), (3), and (4) above are typical for complete beginners.

Minimum End-Results

Objectives are wishes. End results are pragmatic assessments of desirable achievements of the participants at the end of the course. What is the minimum that a participant would have accomplished at the end of the training? The following are offered as an example of what each participant should achieve:

1. An understanding and appreciation of how microcomputers can be used in agricultural policy analysis in developing countries.

2. How to operate a microcomputer and be able to distinguish and evaluate various types of pheriperals.

3. Know the major software categories and their typical and potential applications to policy analysis work, and have hands-on experience with each.

4. Have assessed different software packages' power, weaknesses, and ease of learning and use.

5. Know which categories of software are most suitable to one's analysis and information requirements and achieve competency in operating these packages. Each participant should at the minimum be able to proficiently operate a spreadsheet.

6. Have applied the software in (5) to realistic problems and data preferably taken from situations encountered in the participant's own job.

7. Have at least one of the applications above polished into an operational model ready for immediate use. The participant must be able to operate the model and to perform non-trivial modifications.

Issue on Methods and Course Design

Teaching Microcomputer Skills

Teaching microcomputers is, in many aspects, like teaching a craft. It encompasses teaching of rules and facts, and also involves intuition and creativity. Operations of the computer, disk operating system, specific software packages and programming languages, are examples of topics which, like grammar, center around rules or facts. Instruction requires first knowing the facts well, separating the useful from the less relevant, and presenting them in an effective manner. Presentation is usually done in lecture form in a classroom setting.

On the other hand, aspects such as formulating problems and applying software appropriately to arrive at solutions, or intuition required in trouble-shooting, for instance, are dimensions that are difficult to teach in a lecture format. Here the role of the teaching staff is no less vital and difficult, only different. Instead of presenting rules and facts, he/she must provide guidelines, offer demonstrations of his/her own skill, function as an involved critic, and be the source of information about the process in which the student is involved. Teaching is done by example. And learning is done, and demonstrated, through doing and practice -- effective only in a laboratory atmosphere where each participant has sole and unlimited access to a machine. Little formal lecturing is done. Instead, most of the time is spent discussing topics raised spontaneously as the actual implementation problems and design decisions are encountered.

Parallel Training in Policy Analysis Techniques

The purpose of the training is application of microcomputers to policy analysis, not microcomputers for their own sake. Versatility in microcomputer is necessary but insufficient for effective application. Understanding of underlying economic and quantitative concepts must precede microcomputer implementation. Refresher lectures in areas on policy modeling, econometrics and other quantitative methods are not only beneficial in their own rights, but also clear the way for discussions on technical implementation issues. Great efficiency in instruction can be achieved if lab materials are designed to fulfill the dual objective of solidifying the concepts discussed and improvement of microcomputer skills. In the ideal, not only is the microcomputer used as a vehicle for teaching these concepts, but also vice versa. Appendix B is an example of lab material designed with this objective.

Meeting Participants' Diverse Learning Objectives

Beginners' interest in microcomputers are typically narrowly focused on applications in existing situations from their job duties. Their short-run assessments of the training are understandably based on perceptions of how well these needs are addressed. Early fulfillment of these needs are important and often compulsory motivational devices for the higher objectives of expanding analytical perspectives and more advanced and creative applications of microcomputers. Attaining the depth of knowledge required to handle practical situations demands specialization. Instruction must recognize the fact that effective use of microcomputer requires good knowledge of a small number of the proper set of tools instead of superficial knowledge of many.

What tools are right depends on each analyst. Expected are differences in analysts' backgrounds and job duties, hence also their learning capacities and interests; creating severe uncertainty in the appropriate instructional materials to prepare and also logistical difficulties when group instruction are offered. Private tutoring, where instructional materials are not preset but custom-made and delivered individually, is certainly the most effective but costly. Nonetheless, any alternative course design and delivery approach should build in sufficient flexibility to tailor to individuals' microcomputer needs.

Teaching General Purpose Tools versus Special Purpose Tools

A general-purpose tool is one which can be tailored to perform different applications. Since most programs can be customized to some extent, generality is a matter of degree. Special purpose programs with no or restrictive customization potential are often called "canned" programs. Example of a general purpose tool is a spreadsheet, in contrast to special purpose programs written for specific situations, such as computing break-even discount rates.

The main advantage of teaching a general purpose tool is that it promotes and in fact demands deeper understanding of problem-solving with microcomputers plus the underlying quantitative methods being applied -- unlike "canned" programs which are usually "black boxes". More than just a teaching device, the general purpose tool, once mastered, equips the student to tackle a wide class of problems. A general purpose tool is also the appropriate one to teach to a group with diverse interests. The skills developed from learning to construct a break-even analysis on a spreadsheet retain their usefulness even if the analyst never has the occasion to perform such analysis in reality, since many of the same skills are applicable to other types of analysis.

The price for generality are steeper learning curve and the increased effort needed to produce something useful even after the learning curve is overcome. Indeed, for this precise reason, coverage of computer programming -- one of the most general purpose of tools -- cannot be recommended within a 3 to 6 week time frame.

On the other hand, a "canned" program which fits an analyst's special needs certainly deserves coverage. But "canned" programs are more vulnerable to obsolescence as situations changes. A training program which bequeaths the students the need of more training when faced with a different machine or software is of limited value. The ability to find out by oneself how to operate software and hardware through consultation of appropriate documentation is the relevant ability to develop. Operation of specific programs should hence be covered, not just for their own sake, but also as a case study for the deeper instructional objective of developing the participants' capability for self-learning.

A Modular Course Design

The concern for flexibility precludes the use of a rigid syllabus. The course is instead structured by a set of modules which span a relatively wide range. With the aid of the instructor, participants select modules which suit their individual backgrounds, interests, needs, and aspirations. Modules selected by majority of participants, usually the introductory modules, can be delivered with more teaching staff involvement, both in terms of lectures and labs. The less popular modules, typically the advanced ones, can be delivered as a package containing reading materials, self-guided tutorial, supplemented by small group sessions, and over-the-shoulder lecturing and discussion with teaching staff during lab sessions.

Example module topics are:

1. Introduction to microcomputers and their functions in agricultural agencies.
2. Survey of microcomputer software.
3. Survey of microcomputer hardware.
4. Introduction to disk operating system.
5. Introduction to electronic spreadsheets.
6. Word processing.
7. Statistical concepts.
8. Presentation of data.
9. Introduction to data management.
10. Advanced computer business graphics.
11. Advanced spreadsheets.
12. Linear programming.
13. Econometric analysis.
14. Time series analysis and forecasting.
15. Simulation.
16. Analysis of cost and benefits of government intervention.
17. Project management.

18. Advanced data base techniques.

19. Microcomputer programming and software design.

Chapter Summary

An approach and format for providing microcomputer training was suggested. Issues which must be considered when designing such a training session were raised, although not all the answers were provided. Training programs should aim for providing practical skills which can immediately be applied to individual's everyday analysis work; and also stir curiosities, provide background, and build confidence for further self-guided learning on microcomputers. True evaluations of the success of the training is only possible in the long run.

CHAPTER VI

A FRAMEWORK FOR MEASURING WELFARE IMPACTS OF GOVERNMENT PRICE POLICIES TO CONSUMERS AND PRODUCERS

Chapter Objective

Described in this chapter is a framework for calculating welfare impacts of government price policies to consumers and producers. The economic tool employed is based on concepts of consumer and producer surplus. The microcomputer tool used is an electronic spreadsheet.

This type of analysis is already popular for a one commodity case (see Tweeten, 1984 for example). The data requirements are modest: only elasticity estimates and prices and quantities observations are needed. The underlying economic concepts and numerical calculations are easily understandable. Thus, the expertise to not just operate but also to comprehend the model is widely available in many agencies. Moreover, since almost all microcomputer users own and can operate spreadsheet software, a spreadsheet implementation of the model allows analysts to customize to individual policy situations not only by changing parameter values but also by adjusting the model structure when appropriate. In addition to simplicity, the analysis provides practical illustrations to decision makers on how prices impact on the welfare of consumers and producers.

The challenge here is to extend this type of analysis to a multi-commodity situation and yet retain its major strength of simplicity. This extension is necessary. Setting a higher producer price in one commodity market affects welfare of other commodity producers both because of changes in other output prices and shifts in supplies. Likewise, consumers react to a higher price of one good by increasing demand for its substitutes, bidding up their prices and thus starting an additional round of welfare losses in addition to that caused directly by the price increase of the first good.

Specifically, the objective of this chapter is two-fold. The first is to explicate some of the controversies of welfare analyses which use consumer and producer surplus, especially when the analysis is done for multi-markets. We will attempt to demonstrate that welfare measures are meaningful, albeit difficult to calculate exactly; then offer a means for approximation. The second objective is to illustrate the microcomputer spreadsheet techniques needed for implementing this type of analysis. A generic approach is used. In other words, the demonstration is not specific to any set of commodities, nor is it specific to any administrative settings. The aim is to describe a machine that computes consumer and producer surplus in a multi-commodity setting; and in a manner that is sensitive to the theoretical concerns of producer and consumer surplus and yet simple enough for a spreadsheet implementation.

Measurement of Consumer Welfare

Existence of an Objective Measure of Consumer Welfare

Although few would dispute that consumers experience welfare changes when the product price varies, the measurement of this welfare change has long been controversial in the economic literature. Since welfare is ultimately related to the consumer's utility function, some argue that acceptance of the existence of welfare measures is an implicit acceptance of cardinal utility and interpersonal comparisons of utility, and thus must be laden with value judgments.

But counter arguments can be provided. In the consumer demand curve in Figure 8, suppose initially q0 is consumed at price p0. As price falls to p1, consumption is expected to increase to q1 but the consumer now only need to spend p1q0 instead of p0q0 for q0 units. The saving (p0-p1)xq0 is the amount a consumer would be willing to pay for the price decrease. This amount can be considered as a monetary measure of the welfare gain, derived with only indirect reference to the consumer's utility function through the consumer's demand curve: an observable consequence of the consumer's (ordinal) utility function.

This measure of welfare change is not without problems, however. A price increase from p1 back to p0 would leave a consumer worse off by the amount (p0-p1)xq1: the additional expense needed to continue consumption of q1. The welfare loss of this price increase more than offsets the welfare gain of a price decrease of equal magnitude and thus is intuitively unsatisfactory. However, the two amounts can be reconciled if one considers that price change is realized in a series of small steps (Figure 9). Thus at the limit the welfare change becomes the area enclosed by the two prices and the demand curve (Figure 10). This geometric area, as discussed below, can be given a different but related interpretation as the change in consumer surplus when price varies.

Dupuit's Interpretation of Consumer Surplus

The term consumer surplus was coined by the French engineer Dupuit in 1844. Viewing a consumer's ordinary demand curve as a marginal willingness to pay curve, in Figure 11 the consumer is willing to pay a maximum price of p1 for the first unit, p2 for the second unit and so on. Since the consumer only pays p0 for q0 units, a "surplus" of pi-p0 is realized for the i-th unit consumed. If the commodity is perfectly divisible, consumers surplus for consumption of q0 units is the area above the price line and below the demand curve (Figure 12). As price changes from p0 to p1, the same shaded area in Figure 10 represents the increase in consumer surplus: this is the apparatus most often used in empirical work to measure consumer welfare.

At any quantity of consumption, consumer surplus is always greater than the total expenditure consumer spend on the product. The significance of consumer surplus as a welfare measure is that market situations deemed privately unprofitable may potentially have a more profitable trade-off from a public point of view when welfare is considered instead of revenue gain.

Problem of Consumer Surplus as a Welfare Measure

The above provides an intuitive introduction to the concept and usefulness of consumer surplus, defined as the "Dupuit's triangle" --

the triangular area below an ordinary demand curve and above the price line. Intuition can be deceiving, however. When put under the scrutiny of the neo-classical consumer utility maximization framework, this area is shown to be neither (1) well-defined nor (2) a meaningful monetary measure of utility change, except under very restrictive situations not supported by the bulk of empirical evidence (Just, 1982; Silberberg, 1978).

Path Dependency

"Well-defined" refers to whether alternative but equivalent methods of measurement yield unique or consistent results. Consider the case of two rival commodities 1 and 2. In Figure 13a and 13b, suppose the demand curve of 1 and 2 are represented by D1 and D2 respectively. Initial quantities consumed are q1 and q2 at prices p1 and p2. When prices falls to p1' and p2', D1 and D2 shift inward to D1' and D2' due to substitution, quantities consumed increase to q1' and q2' respectively. To compute consumer surplus, pick any quantity q1" between q1 and q1' and ask for the maximum price the consumer wishes to pay for this unit. Determining this price requires knowledge of the precise location of demand curve D1 which is shifting as p2 is also changing. In other words, as price of commodities 1 falls from p1 to p1' we would need to know where the price of commodities 2 is at each point. Mere knowledge of initial and final prices is insufficient to unambiguously determine the maximum price the consumer is willing to pay at each point. Consumer surplus, the sum of areas under these prices, depends on the adjustment paths of prices, even if the final prices are the same. Different assumptions on price paths need not yield the same consumer surplus value.

It can be shown that if the income elasticities of the commodities are equal, this ambiguity does not occur (Just, 1982). However, equality of income elasticities is a restrictive assumption difficult to justify in many cases.

Problem with Utilitarian Interpretation

The next difficulty of consumer surplus is largely caused by imposing the interpretation of "monetary measure of utility" -- an interpretation beyond that of "willingness to pay" as originally intended by Dupuit, and beyond which is necessary for applied welfare economics. In neoclassical microeconomics, consumers are assumed to maximize an ordinal utility function subject to a given income. First order conditions for constrained maximization requires equating ratios of marginal utilities to price ratios. In particular, if money (with price of one) is used as the numeraire good, then the price of any good can be expressed as the ratio between the marginal utilities of the good and money on the ordinary demand curve. Thus the area under a consumer's demand curve is a monetary measure of utility only if the marginal utility of money -- the scale of measurements -- remains constant as quantities vary. With aggregation, the constancy of the scale of measurements (marginal utility of money) must further hold among the categories. And this condition holds if and only if all income elasticities are constant and equal. These stronger restrictions are less likely to hold than equal (but not necessarily constant) income elasticities required previously for path independence (Just, 1982).

Compensation Criterion, Pareto Optimality and "Willingness to Pay"

Since a strict utility interpretation of welfare change is possible only under conditions not likely to hold in practice, policies are often assessed by the simpler but plausible compensation criterion.

The compensation criterion is closely related to Pareto optimality and "Willingness to pay". An allocation y is said to Pareto dominate x if every one prefers y to x. When some prefer x while others prefers y, but we can reallocate y by appropriately compensating losers and winners, so that the new allocation z Pareto dominates x. Then y is 'superior' to x even though the reallocation of y to z is not actually carried out (Walsh, 1980).

For example, suppose the economy consists of group As and B and C and, policy p is being assessed. Suppose group A as a whole is willing to pay $100,000 to have x implemented, whereas group B is willing to pay $50,000 to avoid x. Thus A prefers x while B does not. But both A and B would prefer x if a compensation of $75,000 is made from A to B, since this position clearly Pareto dominates the initial one. However, the compensation principle still judges the final position as superior even if the $75,000 payment is not made. Only allocative efficiency, not distribution, is of concern here. An allocation which is "bigger" (in monetary terms) but not necessarily "better" in (utility terms) than the original one is picked, although one can in principle reshuffle a "bigger" allocation into "better" by actual monetary compensation. The question of actual compensation, some proponents of the compensation principle argue, is one of income distribution. According to welfare economic theory, the question of income distribution can be made separately from the question of allocative efficiency, and requires different instruments such as redistributive taxation (Varian, 1984).

Equivalent and Compensating Variations

Denote the amount of income the consumer would need at price p1 to be as well off as facing price p and income y by w(p';p,y). Two measurements of compensation are possible:

$$EV(p0,\ y0;\ p1,\ y1) = \qquad (1)$$

$$w(p0;\ p1,\ y1) - w(p0;\ p0,\ y0) = w(p0;\ p1,\ y1) - y0$$

$$CV(p0,\ y0;\ p1,\ y1) = \qquad (2)$$

$$w(p1;\ p1,\ y1) - w(p1;\ p0,\ y0) = y1 - w\ (p1;\ p0,\ y0)$$

Where p and y denote vector of prices and income and 0 and 1 denotes respectively before and after policy positions. EV(p0, y0; p1, y1) and CV(p0, y0; p1, y1) denote EV and CV as prices and income change respectively from p0, y0 to p1, y1. In equivalent variation (EV), the status quo price is used as the base to measure the income change that would be equivalent to the proposed change. Compensating variation (CV) uses new prices as the base and asks what income change would be necessary to compensate the consumer after the price change. Both are reasonable measures of the welfare effect of a price change. Their magnitudes will generally differ since the dollar's value depends on reigning prices. However, their sign will always be the same since they both measure utility difference.

To clarify how these amounts are measured, consider the consumer's utility contour in Figure 14. The axis are quantities consumed. Initially, the consumer is maximizing his utility subject to his income and attains U0 on his utility scale. Denote the initial price of A and B as p_A and p_B respectively. Thus the utility maximizing position is point A at a cost of living of $C(P_A,P_B,U0)$. As price of A is lowered from p_A to p_A', the consumer is supposed to readjust his commodity bundle and attains a higher utility level U1. At each set of prices p and utility U, the consumer minimizes cost of living. The minimized cost can be expressed as a function of P and U, i.e. C(p,U). We can alternatively express CV and EV as the results of these cost minimizations:

$$CV = C(p_A,\ p_B,\ U0) - C(p_A',\ p_B,\ U0) \tag{3}$$

$$EV = C(p_A,\ p_B,\ U1) - C(p_A',\ p_B,\ U1) \tag{4}$$

For a welfare gain, CV is the amount the consumer will be willing to pay for the change; EV is the amount he would need to forego the change. For a welfare loss, CV is minus the amount the consumer would need to receive as compensation for the change; EV is the amount he would be willing to pay to avert the change. Both measures are expressed as difference in consumer's total cost, where total cost of living is a function of prices and desired level of utility. These can be denoted as area under the consumer's marginal cost curves, i.e. the integration of:

$$CV = MC(p_A \mid p_B',\ U0) \tag{5}$$

$$EV = MC(p_A \mid p_B',\ U1) \tag{6}$$

with respect to p_A over the interval of the change in price of A. The marginal cost curve expresses additional cost to the consumer for a small rise in the price of A to maintain the original utility level. At the margin, the cost to return to the original utility level is the "cost" of the last unit lost, derivable directly from the individual's demand function. Mathematically, when the envelope theorem is applied to the indirect cost function, the first derivative with respect to own price is precisely equal to the price of the last quantity consumed, this allows the expression of CV and EV as the integration of:

$$EV = q_A(p_A \mid p_B', U0) \qquad (7)$$

$$EV = q_A(p_A \mid p_B', U1) \qquad (8)$$

with respect to p_A over the interval of price change in A. Where $q_A()$ is the demand function. CV is measured with reference to the original utility level U0, whereas EV is measured with reference to the utility level after the policy change.

Thus after re-examining and adjusting our interpretation of welfare measures in terms of the compensating principle, we have again expressed welfare measures as areas bound by price lines and the consumer's demand curves.

But these are not ordinary demand functions derived from first order conditions in the primal utility maximization model given prices and income -- these demands are not functions of prices and income. Instead, these demand functions are derived from first order conditions of the consumer's dual problem of cost minimization for given levels of prices and utility. Unlike ordinary demand curves along which income is held constant, here cost (required income) is allowed to vary by a conceptual income compensation to arrive at the given level of utility. These are referred to as (Hicksian) compensated demand curve. Figure 15 shows the relationships between ordinary and compensated demand curves, and EV, CV, and consumer surplus (CS).

For goods with no income effect, CV and EV are equal to each other and to consumer surplus. For non-inferior good, CV <= CS <= EV. For inferior goods EV <= CS <= CV. For any good, CV of a move from state A to B equals minus EV of a move from B to A.

There is no real answer to whether CV or EV should be preferred. If one considers the ultimate problems of social choice can only be solved in principle by allowing for distributional judgments, then neither EV nor CV could make these judgment easier. However, if compensation does not alter the structure of relative prices, then the compensation criterion amounts to requiring the sum of CV of all losers and gainers to be at least zero. This requirement arise because CV, unlike EV, is defined with reference to the original level of utility. For this reason CV has been preferred by economists and we shall henceforth concentrate on it instead of EV.

Welfare Measure for Income and Price Change

Consider first the case of income change alone. In Figure 16, the consumer's initial demand is represented by D(y0) and consumption is at point (p, q). As income decreases by say, 100 dollars, the demand curve shifts inward to D(y1) for non-inferior goods: the consumer is now willing to pay less for an additional unit at each quantity. A compensation of 100 dollars would bring him back to his original bundle and thus his initial utility. Hence the consumer's CV or EV loss is trivially 100 dollars.

This point can be further illustrated by applying an income increase from y0 to y1 but holding p at p0 in equations (1) and (2):

$$EV(p0,\ y0;\ p0,\ y1) = \qquad (9)$$

$$w(p0;\ p0,\ y1) - w(p0;\ p0,\ y0) = y1 - y0$$

$$CV(p0,\ y0;\ p0,\ y1) = \qquad (10)$$

$$w(p0;\ p0,\ y1) - w(p0;\ p0,\ y0) = y1 - y0$$

expressing both EV and CV change precisely as the change in income.

Now consider a simultaneous change in price and income from p0, y0 to p1, y1:

$$EV(p0,\ y0;\ p1,\ y1) \qquad (11)$$

$$= w(p0;\ p1,\ y1) - w(p0;\ p0,\ y0)$$

$$= w(p0;\ p1,\ y1) - w(p1;\ p1,\ y1)$$

$$+ w(p1;\ p1,\ y1) - w(p0;\ p0,\ y0)$$

$$= EV(p0\ y1;\ p1,\ y1) + y1 - y0$$

$$CV(p0,\ y0;\ p1,\ y1) = \qquad (12)$$

$$= w(p1;\ p1,\ y1) - w(p1;\ p0,\ y0)$$

$$= w(p1;\ p1,\ y1) - w(p0;\ p0,\ y0)$$

$$+ w(p0;\ p0,\ y0) - w(p1;\ p0,\ y0)$$

$$= y1 - y0 + CV(p0,\ y0;\ p1,\ y1)$$

Thus to estimate EV of a simultaneous price and income change, the effects of the price change should be evaluated at the terminal income level y1 and then add that effect to the change in income, i.e. y1 - y0. On the other hand, for CV, the effects of the price change should be evaluated at the initial income level and then add to that effect the change in income. Thus in Figure 17, loss in CV for a decrease in both income and price from y0, p0 to y1 p1 is y1 - y0 + (areas a + b). The EV change is y1 - y0 + area a.

Multi-market Considerations

Price changes in one market are expected to affect related markets. In Figure 18, suppose a consumer faces perfectly elastic supply curves for product X and Y and initially consuming Qx" and Qy" at prices Px" and Py" respectively. As the price of X falls to Px', consumption of X increases to Qx'. CV for this price change is the area Px" F G Qx'. The demand curve for Y, assuming it is a rival of X, shifts inward. A lower quantity of Y is consumed, resulting in an apparent loss of CV in the area H I J K. It is tempting to subtract this loss of CV from the CV gain in Figure 18a to obtain the net CV from the fall of Px.

But this is not the case: as the consumer is moving from his compensated demand curve from F to G in Figure 18a, prices of other goods remain unchanged, but he is free to alter his expenditures on all other goods in the way he deem most advantageous to him. At Qx", he is willing to pay Px" for an additional unit of X, but only provided that he could freely redistributed his expenditures on other goods; or else he would not be willing to pay quite Px". Thus Px" can be considered as the exact measure of his gain in CV if this additional unit of X is given to him at no charge, if he is free to reshuffle his bundle of goods according to his preference. In particular, having this additional unit of X would at the same time reduce his consumption of any rival good and make him less willing to pay for any unit of it: having an additional pound of coffee per week reduces one's consumption of tea and weakens the willingness to pay for it. But this reduction in willingness to pay should not be counted as a reduction in the consumer's welfare.

Continuing the same argument, when the price of X falls from Px" to Px', Px" F G Qx' is the largest sum he will pay for this price fall, if adjustment of expenditures on other goods is also possible, in particular reducing expenditure on substitute good Y. Thus as price of a good change but other prices and income can be assumed constant, CV change is captured entirely in the demand for X despite the shifts of other demands.

If price of Y now falls to Py', the gain in CV should naturally be made with reference to the demand curve Dy'Dy' which is the appropriate curve when the price of X has already fallen to Px'. CV for a "simultaneous" fall in the prices of both X an Y is therefore the sum of the two shaded areas in Figure 18.

Note that "simultaneous" is put in quotes since the price changes actually occurred sequentially, Px before Py. If instead change in Py is considered to precede Px, then CV gain is the sum of the two shaded areas similar to, but not the same as, those in Figure 18. The two measures need not be equal if ordinary demand curves are used, but must be equal with compensated demand curves. It can be shown that assumptions of other price paths also yield a unique measure of CV: the path dependence problem does not exist (Just, 1982). For the demand system $q_i(p_1, p_2, \ldots)$, where the q_i are quantities, p_j's are prices, path independence is guaranteed mathematically by symmetry of the cross partials:

$$\frac{dq_i}{dp_j} = \frac{dq_j}{dq_i} \qquad (13)$$

A condition which holds along an indifference curve. Since compensated demand curves hold utility constant, path independence holds for compensated demand. Thus a major criticism of consumer surplus, path dependency, is circumvented by using instead CV (or EV).

Clarification of Terminology

We have thus far discussed the concepts of consumer surplus (CS), equivalent variation (EV), compensation variation (CV) and willingness to pay and how these concepts are related. For the rest of the chapter, we will adopt the following convention. CV, the monetary amount needed to compensate losers or taken away from gainers after a policy change for them to be indifferent to the change, is considered to be the same as "willingness to pay". We will thus use these two terms interchangeably. EV will seldom be used. Consumer surplus is used to refer to the usual area under an ordinary demand curve, it is, however, interpreted as a pure geometry area void of any welfare meaning. In reality, however, ordinary demand curves, not compensated demand curves, are usually observed. We will interpret the change in consumer surplus (after some adjustment to be discussed later) as an approximation of CV or willingness to pay. And this is what will be used as our measure of welfare change.

Adjusting Consumer Surplus to Approximate CV

Referring again to Figure 15 where price is initially p0 and falls to p1, consumer surplus gain is the area a+b -- the area under the ordinary demand curve D(p,q0,y0). The compensated curve, at the initial utility level (when price is p0) and at the final utility level (when price is p1), intersect DD respectively when price are p0 and p1 respectively. CV, as discussed in a previous section, should be measured under the compensated curve at the initial utility level, i.e. D(p,q0,U0). In this case CV is area a. Thus the gain in CS overstates CV by area b. This overstatement is expected to be small when the income effect is small since then the ordinary curve and the compensated curves tend to coincide. The area b however, is itself impossible to calculate using information from ordinary demand alone but can be shown to be approximately equal to (Just, 1982):

$$n * (\text{CS change})^2 / 2m \qquad (14)$$

Where n is the income elasticity of demand, and m is the initial income level. Thus CV can be approximately calculated as:

$$(\text{CS change}) - n * (\text{CS change})^2/2m \qquad (15)$$

This is the basis for Willig's (1976) argument that consumer surplus can be used without apology since the adjustment factor is expected to be small when income elasticity is small or when the change in CS is minute when compared to income (Willig 1976). Thus change in CS, which many consider an "unsound" welfare measure, is actually a close approximation to "willingness to pay" which is a well-defined concept. When income elasticity is large or when the change in CS is large relative to income (likely for subsistence farming), the adjustment should be made since it would yield a closer approximation to the true willingness to pay. We will always make this adjustment for the analysis below.

Measurement of Producer Welfare

Following the spirit of consumer welfare measures described above, an acceptable measurement of producer welfare might be: "The excess of the gross receipts which a producer gets for any of his commodities over their prime cost -- that is, over the extra cost which he incurs in order to produce those things and which he could have escaped if he had not produced them". This in fact, is Marshall's definition of producer surplus -- the device commonly used to measure producer welfare.

The traditional measure of producer surplus is symmetric to that of consumer surplus: the area above the supply curve and below the price line. Since the industry supply curve is a marginal cost curve, this area is thus equivalent to receipts less total variable cost which is also the usual definition of quasi-rent. (Stigler, 1952).

Difficulty with Producer Surplus

The concept of producer surplus is not without ambiguities and controversies. But unlike those of consumer surplus, which mainly arise because of the income effect, the ambiguities and controversies of producer surplus mainly stem from the ambiguity of the supply function as length of run varies. Consider Figure 19. At price p1, producer surplus is represented by the area ABp1. Suppose now price is set to p2. Three measures of producer surplus are possible. First of all, assuming prices of the factors of production is fixed thus the supply or the marginal cost curve remains unchanged, producer surplus is now AEp2. However, if eventually the general price of the factors of production adjusts upward, so that the short run marginal cost curve shifts to S2, and in doing so, a longer run supply curve represented by S' is traced out. Now producer surplus becomes ambiguous. Is it CDp2, the area above the new short run supply curve? Or is EDp2, the area above the long(er) run supply curve?

We will avoid the controversies by emphasizing the word "impact" in the title of this chapter. In other words, we assume the first case where the prices of all variable factors of productions are fixed. Thus the area AEp2 is a "surplus" which accrues to the owners of firms in their production and sale of the product resulting from the ownership of the fixed factors of production. In this sense, the terms "producer surplus" and "quasi-rent" are equivalent. This has led some economist, notably E. J. Mishan (1968), to consider "producer surplus" as an unnecessary jargon. We will interpret producer surplus, or quasi-rent, as the maximum amount producers would be willing to pay for the price increase of the product.

Multi-market Considerations

Change in producer price in one market is expected both to change the price and shift supply of a related product. If we interpret the inward shift of S1 to S2 in Figure 19 as due to price increase of another commodity, and assume the curve has attained its equilibrium position with respect to the rest of the system, then producer surplus is area CDp2 after price is increased to p2. Since this area is now the relevant gross receipts over variable costs. And this area, less area ABp1 (gross receipts over variable cost before the situation changed), is the change in producer surplus in this market. The sum of these differences in all markets after equilibrium is reached is considered as the welfare change of producers. This total amount is interpreted as the maximum amount producers are willing to pay to face the new market situation. The producers are assumed to be willing to pay exactly the total gains in gross receipts less variable costs.

A Two Commodity Example

Introduction

Figure 20 illustrates a spreadsheet layout for calculating the welfare effects of government price policies. Release 2.0 of Lotus 1-2-3 spreadsheet program (Lotus, 1985) is used as the implementation vehicle. But attempts were made to restrict ourselves to features that are available to many spreadsheet programs. For example, although 1-2-3 provides facilities for matrix inversion, and this analysis could have taken advantage of this feature, we avoided this feature since this is not available for most other spreadsheet programs. The calculations incorporate the theoretical considerations discussed in earlier sessions. Cell formulas for selected cells are listed in Figure 21.

The required input data for the analysis are shown in the Figure 20 as underlined. These are to be provided by the user. A base scenario, and three alternatives are included. The base scenario is built using data that are actually observed. In the figure, producer and consumer prices and quantities, are needed to build the scenario. In addition, a set of demand and supply elasticities are required. In the figure, line 5 to 42 represent the figures for commodity 1. Own and cross price elasticity of supply are required in cells B8 and B9 respectively. Likewise, cells B12, B13 and B14 contain own price, cross price and income elasticities of demand for the first commodity. Lines 44 to 81 pertain to commodity 2. Information from line 83 onwards are not commodity specific. Line 83, for example, contains consumer income.

General Approach

A simultaneous linear supply and demand system is assumed. The coefficients of the system are solved from the given elasticities and quantities in the base scenario. These coefficients are listed from cells B16 to B25 for commodity one. The analysis uses these derived coefficients instead of the specified elasticities directly. These coefficients are assumed to remain valid for the other scenarios. Thus the corresponding entries for elasticities and coefficients need not be, and indeed must not be respecified for the alternatives scenarios. These entries are thus marked as -- in the worksheet.

The alternative scenarios are provided for answering the question "what if?" Economic changes in one commodity market are assumed to be simultaneously linked to other markets. Welfare changes are accessed after equilibrium of the economic system is simulated.

Exogenous variables

Each alternative scenario allows specification of three potential exogenous variables for each commodity. They are consumer prices, producer prices and desired excess supply. Excess supply is the amount by which production exceeds consumption. A negative value denotes deficit. This analysis does not yield information concerning the measurements of the benefit of positive excess supply nor the cost of acquisition of deficits. Deficits, for example, can be overcome with imports, commercial or concession, or stock depletion. The exact ways and costs whereby deficits can be overcome are usually institution specific and depend on how controls are administered. For instance, decision makers can associate a high per unit cost when stock are depleted below a 'secure' level. On the other hand, concession imports may be below world price or free, whereas imports beyond a certain level can be costly from a financial and/or a social viewpoint. Likewise, positive excess supply means possibility of export, or additions to stocks. But even additions to stocks may be a bane or boon depending on the availability of storage facilities. We assume the decision maker can independently assign subjective costs or benefits of these deficits or surplus for weighing against our computed impacts to consumers and producers.

If an exogenous variable is not set, the system attempts to calculate it endogenously. Exogenous values not specified are denoted by NA in the figure. This is entered into the Lotus worksheet using the @NA function to distinguish it from a number or a character string, which most spreadsheets interpret as a numeric zero. Obviously some minimal amount of exogenous information is needed. If this minimal amount is not met, the simulated economic system as implemented would nonetheless supply some default value, usually the base scenario value. For example, if none of the three exogenous variables are set, excess supply is assumed to be the base line value, and producer price is forced to be equal to the consumer price, the values of which are determined endogenously. This is necessary since even if producer price and consumer price are forced to be equal, an infinite number of combinations of prices and excess supply are still possible.

On the other hand, too much exogenous information can be supplied. For example, if both the consumer price and producer price are controlled, then the level of excess supply must be allowed to gravitate to a level consistent with these prices. In our simulator, if all three exogenous variable are set, the excess supply setting is not honored and realized excess supply is determined by the system to be consistent with the controlled producer and consumer prices. Of course, in reality, depending on how the policy is enforced, some or all of the three may deviate from the set values.

Specifically, the following combinations of exogenous variable settings are possible:

Case	Producer Price	Consumer Price	Excess Supply
1.	xx	xx	NA
2.	NA	xx	xx
3.	xx	NA	xx
4.	xx	NA	NA
5.	NA	xx	NA
6.	NA	NA	xx
7.	xx	xx	xx
8.	NA	NA	NA

where "xx" denotes a set value. Cases 1, 2, and 3 create no difficulty. Two out of the three possible exogenous variables are set, and our simulator can uniquely determine the value of the other item endogenously. Only one price is set in cases 4 and 5. In these cases, excess supply is assumed to be the same as the base scenario and the economic system endogenously determines the other price. In case 6, excess supply is given, the system assumes producer price and consumer price are equal and determined them endogenously. In case 7, where all three instruments are set, our simulated system must leave at least one setting un-honored: in this case the excess supply. Thus this case is identical to case 1. In case 8, no instrument is set. The system must assume excess supply to be the same as base and prices to be equal.

Thus this case is the same as case 6 with excess supply set to the base scenario value.

Building the Alternative Scenarios

Referring back to Figure 20, in the base scenario, suppose prices of commodity 1 are controlled at the base value at 6.00 and 5.00 (cell B33 and B34). Production and consumption levels are respectively 100 and 150, resulting in a deficit of 50. Commodity 2, on the other hand, is not controlled. Both consumption and production occurred at world price of 5.00. Production and consumption levels are respectively 300 and 200, and 100 units are exported.

What would be the impact to consumers and producers if the price of commodity 1 is not controlled? In scenario I, suppose world price of commodity 1 is 5.50. We thus insert 5.50 into C28 and C29 for consumer price and producer price. @NA is entered in the excess supply field. Thus the excess supply will be calculated endogenously. For commodity 2, we continue to assume the world price of 5.00. As expected, both consumption and production of commodity 1 is reduced. Excess supply is now -47. Compared to the base scenario, the deficit is reduced by 3 units. This reduction can mean a loss to commercial importers, or alternatively, a slow down of stock depletion which policy makers may consider as a benefit under the objective of self-sufficiency. Assignment of exact cost or benefit figures for this decrease in deficit requires intimate knowledge of the institutional setting and/or policy objective; and is beyond the scope of this work.

The economic changes in commodity 1 also affects the market of commodity 2. Even if prices were not changed from the base scenario, as we have left them, both production and consumption are reduced somewhat. Also reduced is the excess supply.

Line 86 and 87 summarize the welfare impact of this price change. Producers and consumers need to be compensated 31.39 and 103.45 respectively for them to be as well off as facing the base scenario.

Scenario II is a "self sufficiency" scenario for commodity 1. Thus excess supply is set to zero to denote no imports. Consumer price and producer prices are left to find their own levels. Setting for commodity 2 is the same as the previous scenario. In this case, realized consumer price and producer price are forced to be equal to 6.81 by our simulated economic system. Both production and consumption of the commodity occurred at 112.08. Even if prices in commodity 2 remains untouched, production and excess supply reduced and consumption increased. As shown in line 86 and 87, this policy benefits producer but heavily penalizes consumers.

Scenario III is also a self sufficient scenario for commodity 1. But unlike scenario II which heavily penalizes consumers and reduces exports (i.e. positive excess supply) of commodity 2, we choose now to set consumer prices of commodity 1 and 2 to respectively 5.50 and 5.00. Excess supply for commodity 1 is again set to zero but we insist excess supply to continue at level of 100.00. According to the simulator, this can occur only if producer prices for commodity 1 and 2 are supported at 8.63 and 5.90 respectively. Compared to the base scenario, producers gained 496.19 whereas consumers lost 73.68 for a net gain of 422.50.

Design of the Simulator

The simulator is designed to be operated by individuals with only casual experience with microcomputer spreadsheet programs. After loading the worksheet, the users are required to fill in the underlined values which are the input parameters to the analysis. Since simultaneous relationships exist in the spreadsheet, more than the usual one pass recalculation is needed to achieve equilibrium. The user must therefore repeatedly force recalculation by pressing the recalculation key until equilibrium is reached.

But how can one tell when equilibrium is accomplished? Usually, this necessitates monitoring the values of the endogenous variables in successive iterations until they differ by less than a required tolerance. Few spreadsheet programs provide this monitoring automatically and naturally. Most implementation of iterative algorithms on spreadsheets thus requires users to visually determine when the endogenous variables stop changing as more recalculations are forced. This is a workable approach only if the number of endogenous variables is small.

An intuitive explanation of the procedure used in the simulator to solve simultaneous relationships follows. Simultaneous relationships exist when both price and quantity of any commodity must be determined together. Prices are calculated as functions of own quantity and other prices. Producer price (Pp), for example is calculated by:

$$Pp = Qp - (a1 + b1xPp') / b2 \qquad (16)$$

where Pp' is the price of the other commodity and a1, b1 and b2 are coefficients of the linear supply curve. Thus this is simply a rearrangement of terms of the the supply equation so that producer price is now the dependent variable. Instead of directly using Qp in the equation, however, we insert 0.5*(Qp+Qc+ES), where ES is the excess supply equal to Qp-Qs. Note that 0.5*(Qp+Qc+ES) is identically equal to Qp when equilibrium is reached.

For each iteration, prices are calculated by application of equation (16). Prices such calculated are in turn used to derive Qp and Qc by simple applications of the supply and demand equations. Excess supply is then calculated as a difference of these quantities. In other words, at each iteration, the difference between production and consumption need not equal the excess supply, but must be when convergence is reached. This approach was used instead of the traditional checking of successive iterations as a condition for convergence. (See the formulae in the worksheet in Figure 21 for additional details.)

As discussed so far, our approach would still require the user to visually inspect whether the production quantities, consumption quantities, and excess supply add up for all commodities. To eliminate the need for this visual inspection, a Lotus 1-2-3 macro is implemented whereby recalculation continues until the above stated condition holds. This macro is listed in Figure 22. Tha macro is simply an implementation of a loop which continues as long as production less consumption for any commodity in any scenario differs from excess supply for more than a set tolerance, in this case .01 (the value of cell F22). The loop will nonetheless terminate after a set number of iterations (30 or the value of cell F23) even if convergence is not achieved to avoid infinite looping in unusual occasions. In this case the user is notified with a message.

Thus to operate the model, the user need only to fill in the input parameters and press the Alt-A key. This invokes the Lotus 1-2-3 macro which monitors the iteration.

A More Elaborate Example

Introduction

Figure 23 displays a more elaborate example of a spreadsheet layout for calculating the welfare impacts of government price policies to consumers and producers. This implementation follows the spirit of the former two commodity example but differs in that 3 commodities, rice, cassava and coffee are considered. Line 1 to Line 70 contain the information for rice. Information for cassava and coffee begin at line 72 and line 139 respectively.

In addition, this analysis yields information on marketed surplus, defined as rural production less consumption. This is usually the amount of domestic production available for urban consumption. Among other uses, this figure often reflects the amount of the good the government must handle in intervention policies.

The Rural Sector

In this analysis the word "consumers" alone refers to urban consumers. "Producers" actually refers to the rural sector which of course also engages in consumption. However, different demand curves are assumed for rural and urban consumers. Urban demand curves express the amounts of urban consumption as a linear function of own and cross consumer prices, and urban income. Rural demand curves, on the other hand, use producer price and urban income as independent variables.

As producer price increases, production increases but not all of this additional production results in increase in marketed surplus. With increase in production and hence income, rural consumption also is expected to increase due to an income effect. The exact change in rural income due to the change in production is usually difficult to measure, but is approximated in this analysis by the change in producer surplus. Thus rural demand is in effect a function of consumer prices and the change in producer surplus. With this setting, the market surplus curve need not be positively sloped throughout its range. However, in practice the marketed surplus curve is usually positively sloped in the relevant range since the gain in producer surplus is usually small when expressed as a percentage of income. Producer welfare is calculated as the net welfare change in the rural market of consumption and production.

Exogenous Variables

As in the two commodity example, four possible exogenous variables are allowed for each commodity, namely producer price, consumption price, desired marketed surplus and desired excess supply. If desired market surplus or excess supply is marked with @NA (i.e. not set), then they are calculated endogenously as the difference between simulated production and rural consumption, and total consumption and total production respectively. If in addition consumer price or producer price is not set, then both marketed surplus and excess supply is set to the base scenario value to make possible the endogenous determination of these prices.

Building the Scenarios

In Figure 23, a base scenario and two alternatives are provided. The three scenarios differ mainly in the rice section. The base line prices for rice are 500 and 510. In the first alternative, both producer price and consumer price for rice are set to 480. Desired excess supply and marketed surplus are not set, thus they are to be calculated endogenously. In the second scenario, the consumer price and desired marketed surplus are set to 540 and 20000 respectively. Neither the producer price nor the excess supply are set. All prices for other commodities are set at the base level, and with both marketed surplus and excess supply marked as @NA (not set).

As expected, the low producer price in scenario I discouraged production of rice and increased production of cassava and coffee. However, welfare loss due to the low producer price is more than offset by consumption gain in the rural sector, yielding a net 3,272.72 of rural gain. The expanded production in cassava and coffee in addition yields a gain of 52,922+37,922. Urban consumer benefited 3,061,860 due to the lowered consumer price. The net gain of this policy to the rural sector and urban sector as a whole is 3,061,766.28.

In scenario II, the system endogenously set the producer price of rice to be 550 in order to realize 30,000 of marketed surplus and at a consumer price of 550. Production of rice increased, depressing the production of both cassava and coffee. But nonetheless, the welfare gain due to production is a net 7,628,963. Rural consumers and urban consumers lose 7,664,889.58 and 35,926.36 respectively, resulting in a net gain of 5259 to the rural and urban sector as a whole.

Chapter Summary

We have in this chapter extended the framework for analyzing the impact of price policies to consumers and producers to a multicommodity setting. The solution offered is one which takes into considerations the controversies surrounding consumer and producer surplus, and one suitable for implementation with an electronic spreadsheet.

The precise meaning and conceptual difficulties of consumer surplus, especially in a multicommodity setting, were carefully examined. We settled for willingness to pay, or CV, in order to bypass the problems of a meaningful utilitarian measure and path dependency. Consumer surplus is nevertheless still a useful geometric concept and with a simple adjustment, provides us with reasonable estimates of CV. Producer surplus by comparison created few conceptual difficulties if input cost structure can be assumed to be unchanged.

Also demonstrated is the modeling of simultaneous economic relationships with the Lotus 1-2-3 spreadsheet program. The implementation, however, is somewhat convoluted, particularly in the more elaborate example: an indication that a spreadsheet may not be the right tool for such modeling. A spreadsheet is, nevertheless, a tool which many analysts own and know. A framework for expressing simultaneous economic relationships on a spreadsheet is therefore an useful addition to the analyst's repertoire of spreadsheet techniques.

This analysis merely computes impacts to consumers and producers, and falls short of a complete accounting of the costs and benefits of government policies. We have factored out and discussed and implemented the part of the analysis which can be done without an intimate understanding of the institutional settings. Although our results are useful in their own rights, our developments should more appropriately be viewed as a module ready to be fit into a more full-blown analysis of costs and benefits of government policies.

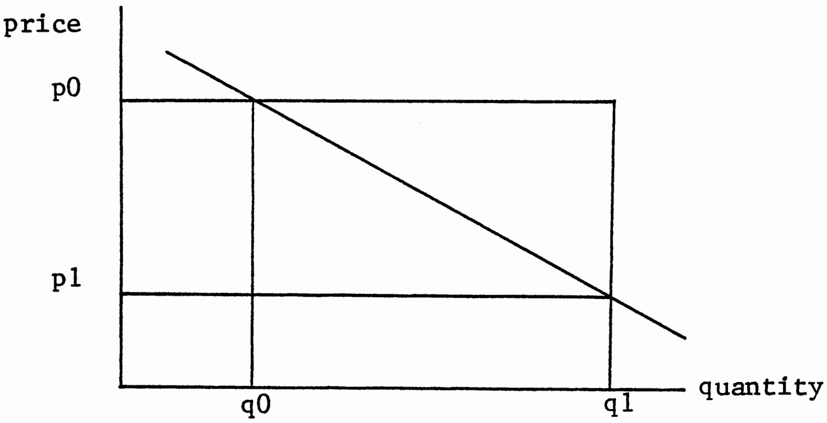


Figure 8. Change in Welfare as Change in Required Expenditures.

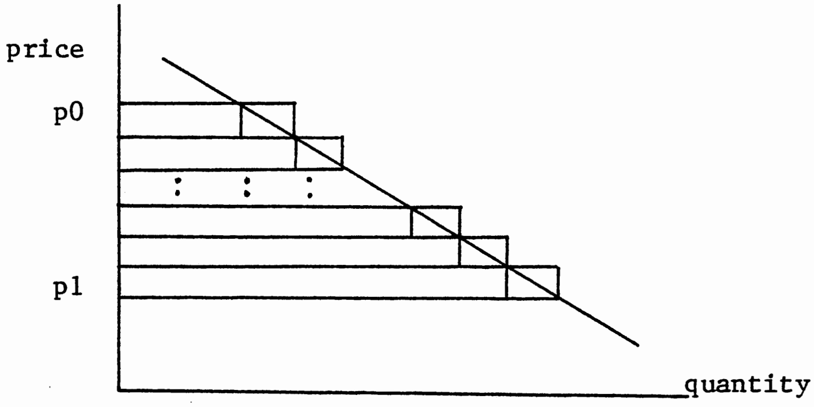


Figure 9. Change in Required Expenditures When Price Changes are Small.

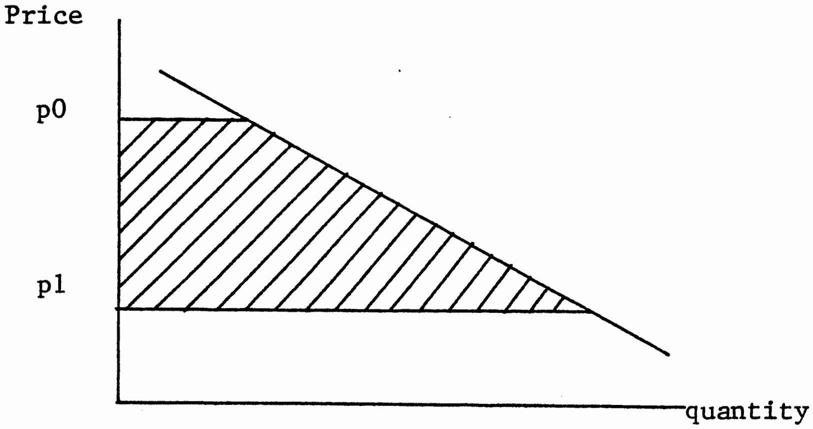


Figure 10. Change in Consumer Surplus when Price Changes

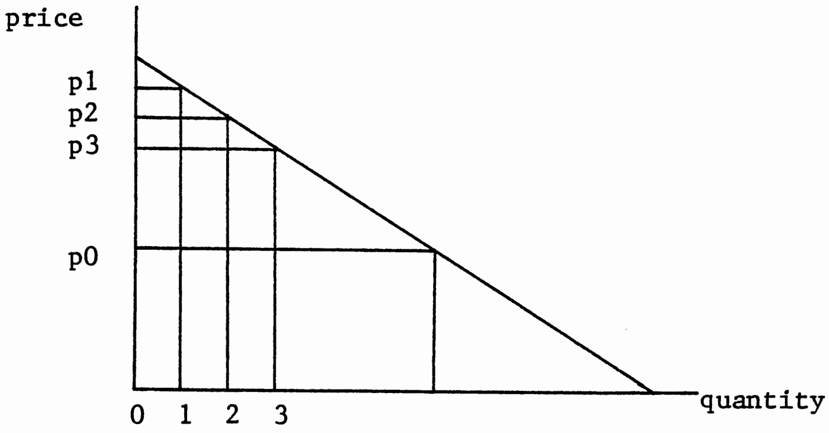


Figure 11. Willingness to Pay vs Actual Amount Paid for Each Unit of Comsumption.

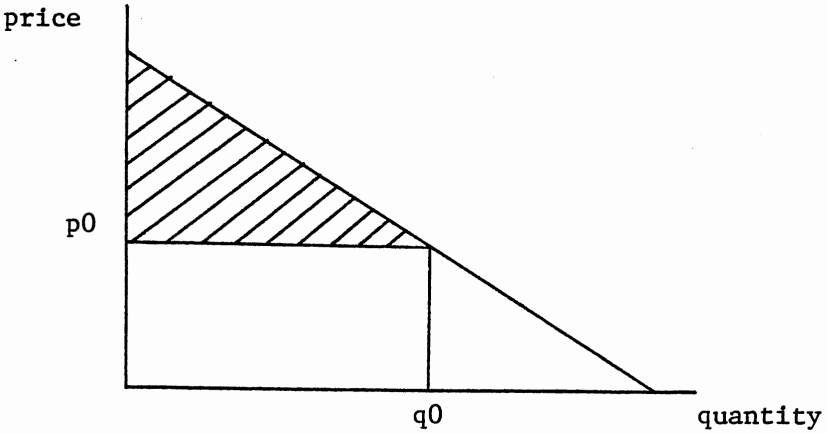


Figure 12. Dupuit's Triangle.

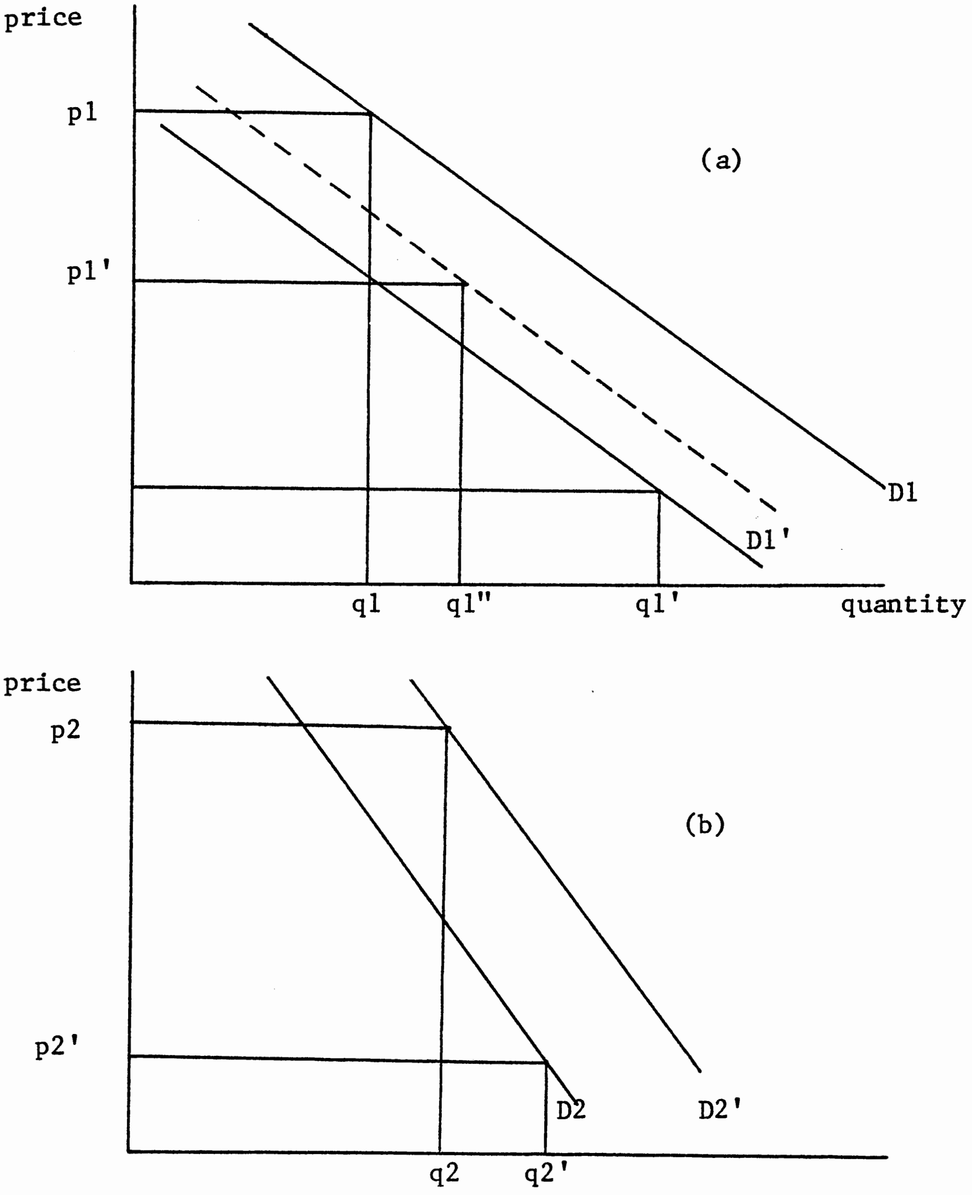


Figure 13. Path Dependency.

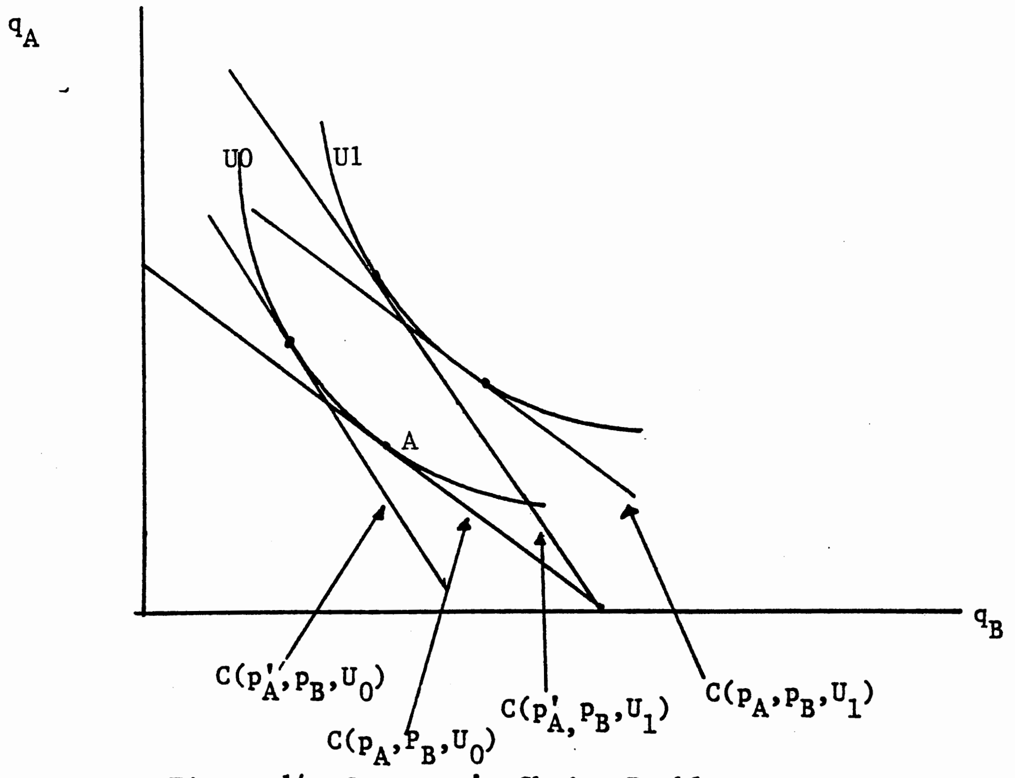


Figure 14. Consumer's Choice Problem

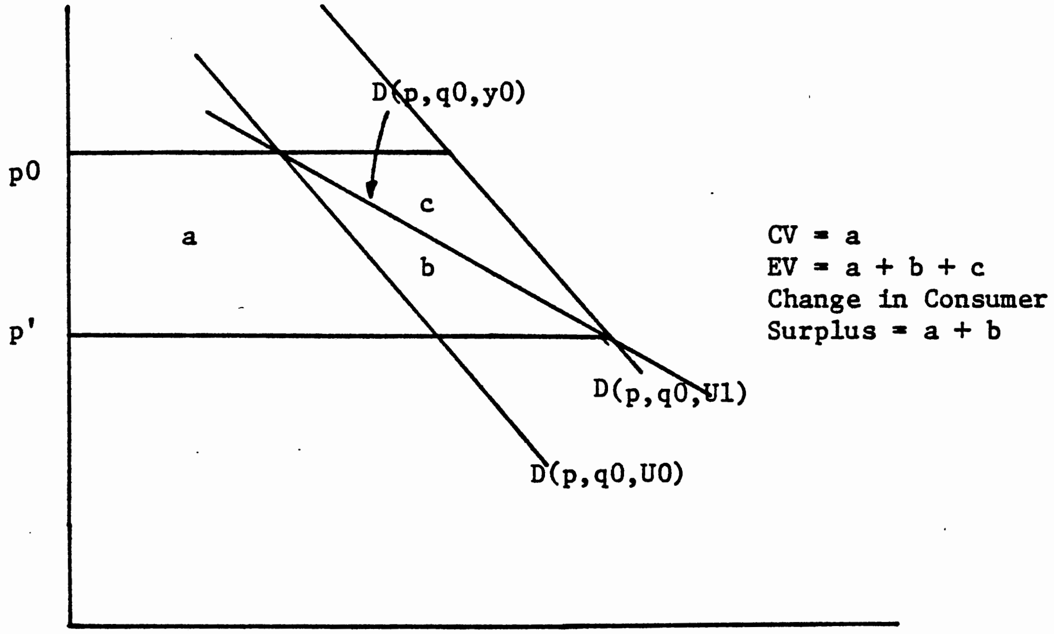


Figure 15. Relation Between CV, EV and Change in Consumer Surplus.

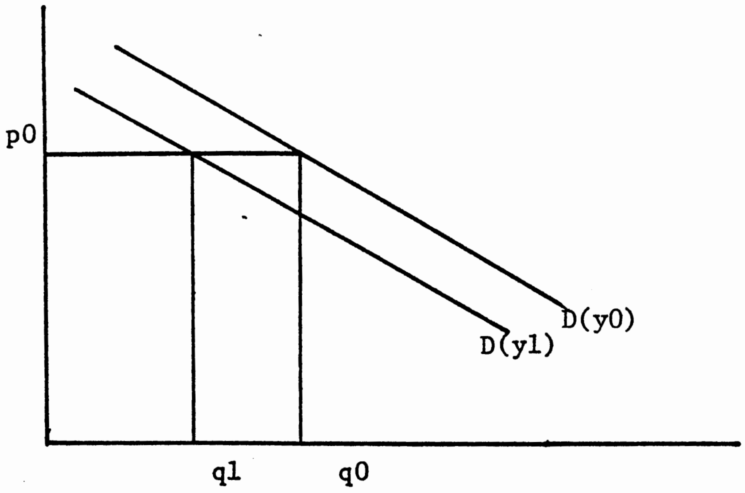
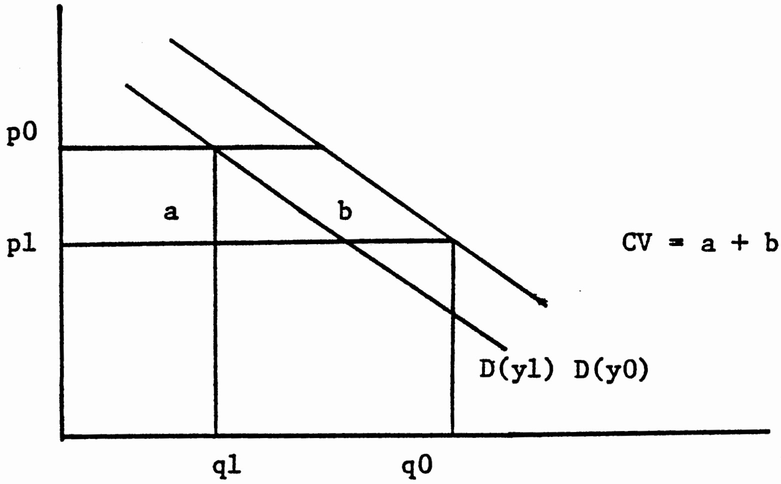


Figure 16. CV for Income Change Alone



$CV = a + b + (y1 - y0)$

Figure 17. CV for Price and Income Decrease.

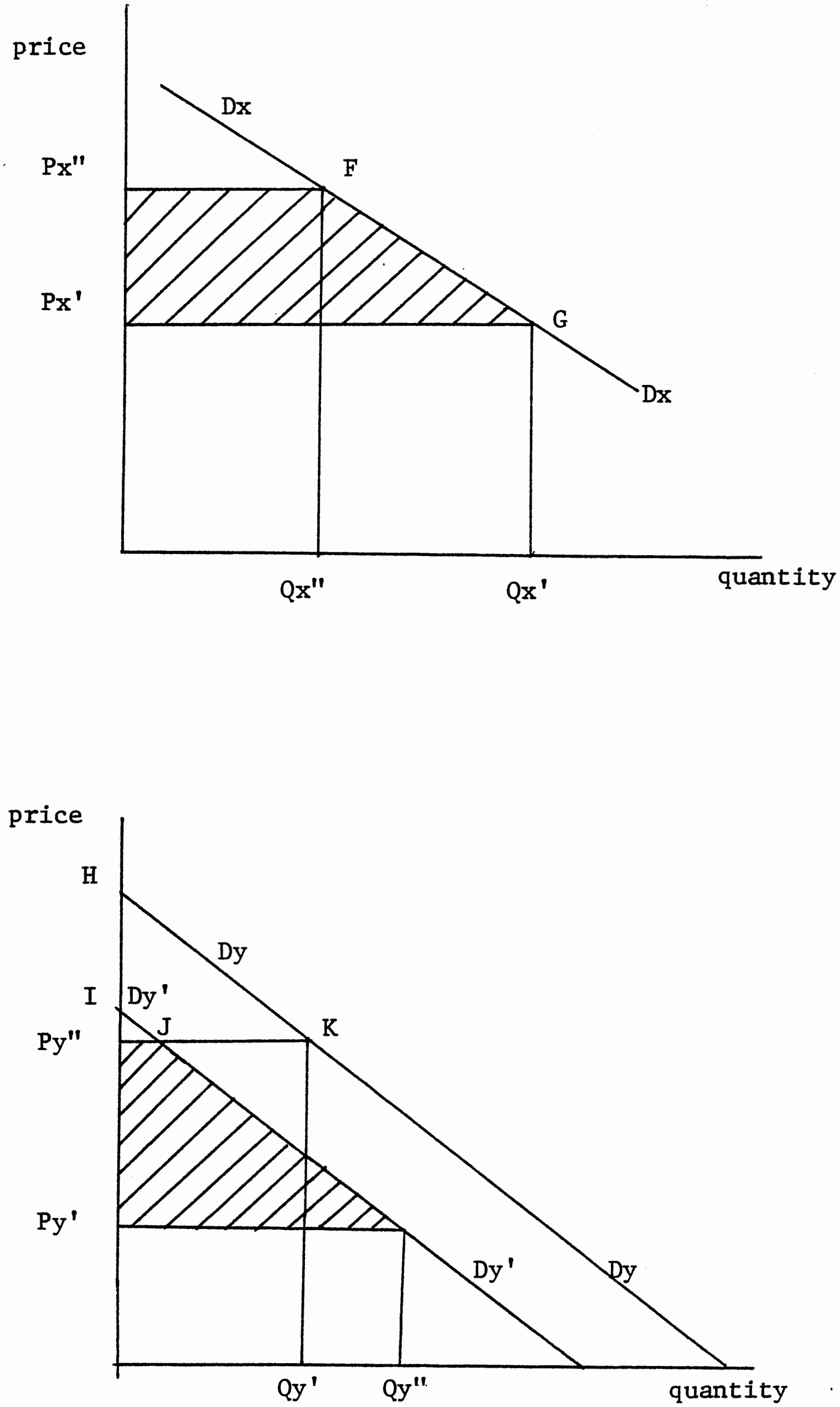


Figure 18. CV for Simultaneous Price Changes.

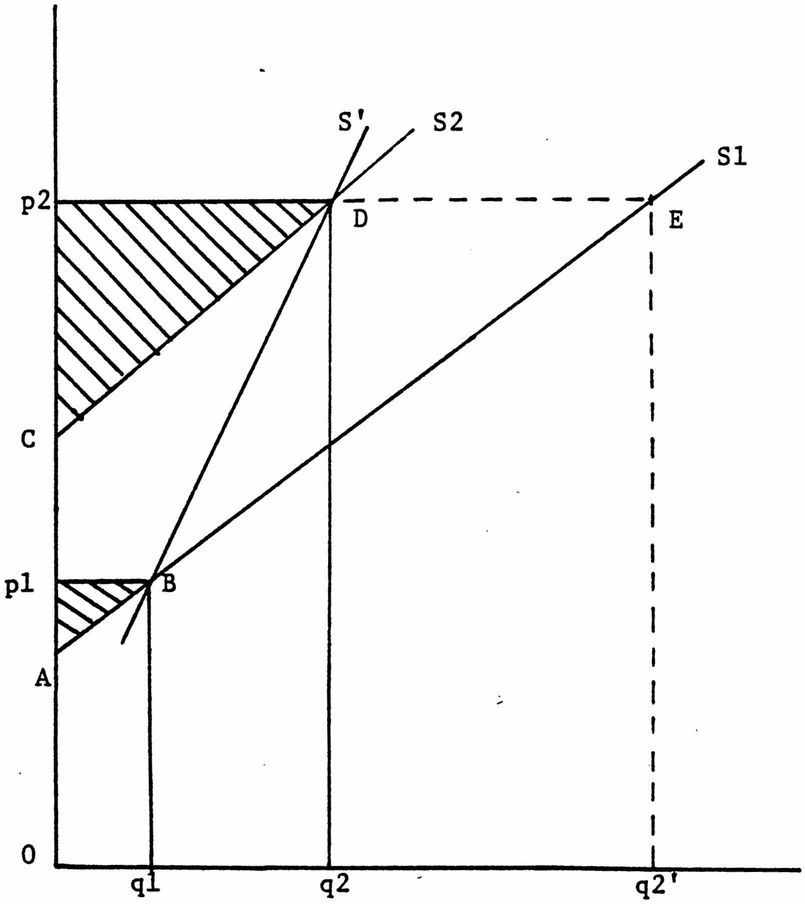
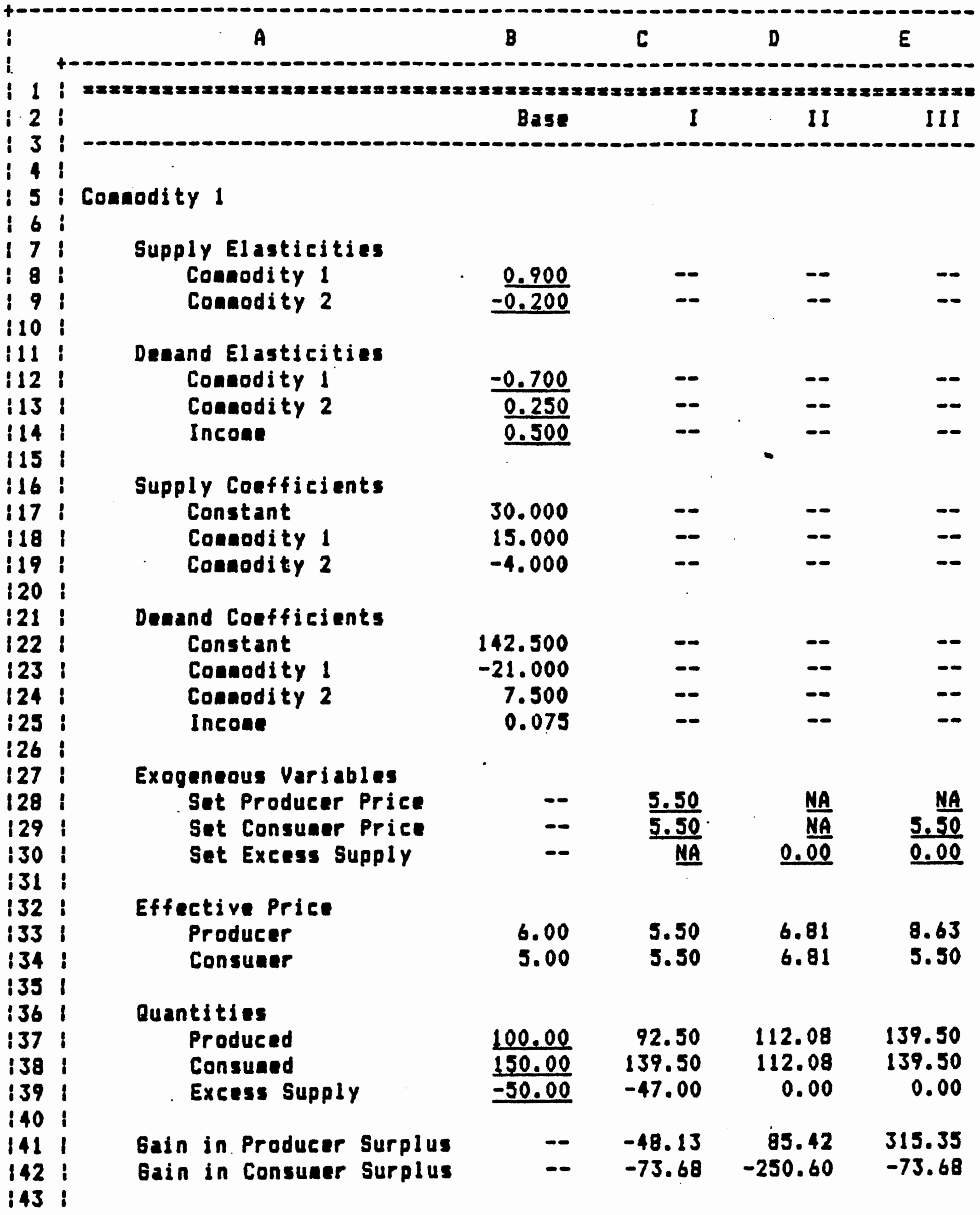


Figure 19. Producer Surplus



```
+----------------------------------------------------------------------------
|                 A                         B          C          D          E
|    +------------------------------------------------------------------------
| 1  | ======================================================================
| 2  |                                      Base          I         II        III
| 3  | ----------------------------------------------------------------------
| 4  |
| 5  | Commodity 1
| 6  |
| 7  |     Supply Elasticities
| 8  |         Commodity 1                  0.900        --         --         --
| 9  |         Commodity 2                 -0.200        --         --         --
|10  |
|11  |     Demand Elasticities
|12  |         Commodity 1                 -0.700        --         --         --
|13  |         Commodity 2                  0.250        --         --         --
|14  |         Income                       0.500        --         --         --
|15  |
|16  |     Supply Coefficients
|17  |         Constant                    30.000        --         --         --
|18  |         Commodity 1                 15.000        --         --         --
|19  |         Commodity 2                 -4.000        --         --         --
|20  |
|21  |     Demand Coefficients
|22  |         Constant                   142.500        --         --         --
|23  |         Commodity 1                -21.000        --         --         --
|24  |         Commodity 2                  7.500        --         --         --
|25  |         Income                       0.075        --         --         --
|26  |
|27  |     Exogeneous Variables
|28  |         Set Producer Price               --      5.50         NA         NA
|29  |         Set Consumer Price               --      5.50         NA       5.50
|30  |         Set Excess Supply                --        NA       0.00       0.00
|31  |
|32  |     Effective Price
|33  |         Producer                       6.00      5.50       6.81       8.63
|34  |         Consumer                       5.00      5.50       6.81       5.50
|35  |
|36  |     Quantities
|37  |         Produced                     100.00     92.50     112.08     139.50
|38  |         Consumed                     150.00    139.50     112.08     139.50
|39  |         Excess Supply                -50.00    -47.00       0.00       0.00
|40  |
|41  |     Gain in Producer Surplus             --    -48.13      85.42     315.35
|42  |     Gain in Consumer Surplus             --    -73.68    -250.60     -73.68
|43  |
```

Figure 20. A Two Commodity Example.

```
+--------------------------------------------------------------------------
!                      A                     B        C        D        E
!   +----------------------------------------------------------------------
!   !
!44 ! Commodity 2
!45 !
!46 !     Supply Elasticities
!47 !         Commodity 1          -0.100       --       --       --
!48 !         Commodity 2           0.750       --       --       --
!49 !
!50 !     Demand Elasticities
!51 !         Commodity 1           0.300       --       --       --
!52 !         Commodity 2          -0.600       --       --       --
!53 !         Income                1.500       --       --       --
!54 !
!55 !     Supply Coefficients
!56 !         Constant            105.000       --       --       --
!57 !         Commodity 1          -5.000       --       --       --
!58 !         Commodity 2          45.000       --       --       --
!59 !
!60 !     Demand Coefficients
!61 !         Constant            -40.000       --       --       --
!62 !         Commodity 1          12.000       --       --       --
!63 !         Commodity 2         -24.000       --       --       --
!64 !         Income                0.300       --       --       --
!65 !
!66 !     Exogeneous Variables
!67 !         Set Producer Price       --     5.00     5.00       NA
!68 !         Set Consumer Price       --     5.00     5.00     5.00
!69 !         Set Excess Supply        --       NA       NA   120.00
!70 !
!71 !     Effective Price
!72 !         Producer               5.00     5.00     5.00     5.90
!73 !         Consumer               5.00     5.15     5.00     5.00
!74 !
!75 !     Quantities
!76 !         Produced             300.00   302.50   295.97   326.00
!77 !         Consumed             200.00   202.50   221.67   206.00
!78 !         Excess Supply        100.00   100.00    74.31   120.00
!79 !
!80 !     Gain in Producer Surplus     --    16.74   -26.67   180.00
!81 !     Gain in Consumer Surplus     --   -29.48     0.00     0.00
!82 !
!83 ! Income                      1000.00  1000.00  1000.00  1000.00
!84 !
!85 ! Change in Willingness to Pay
!86 !     Producer                     --   -31.39    58.75   496.19
!87 !     Consumer                     --  -103.45  -250.60   -73.68
!88 !     Net                          --  -134.83  -191.85   422.50
!89 !
!90 !
```

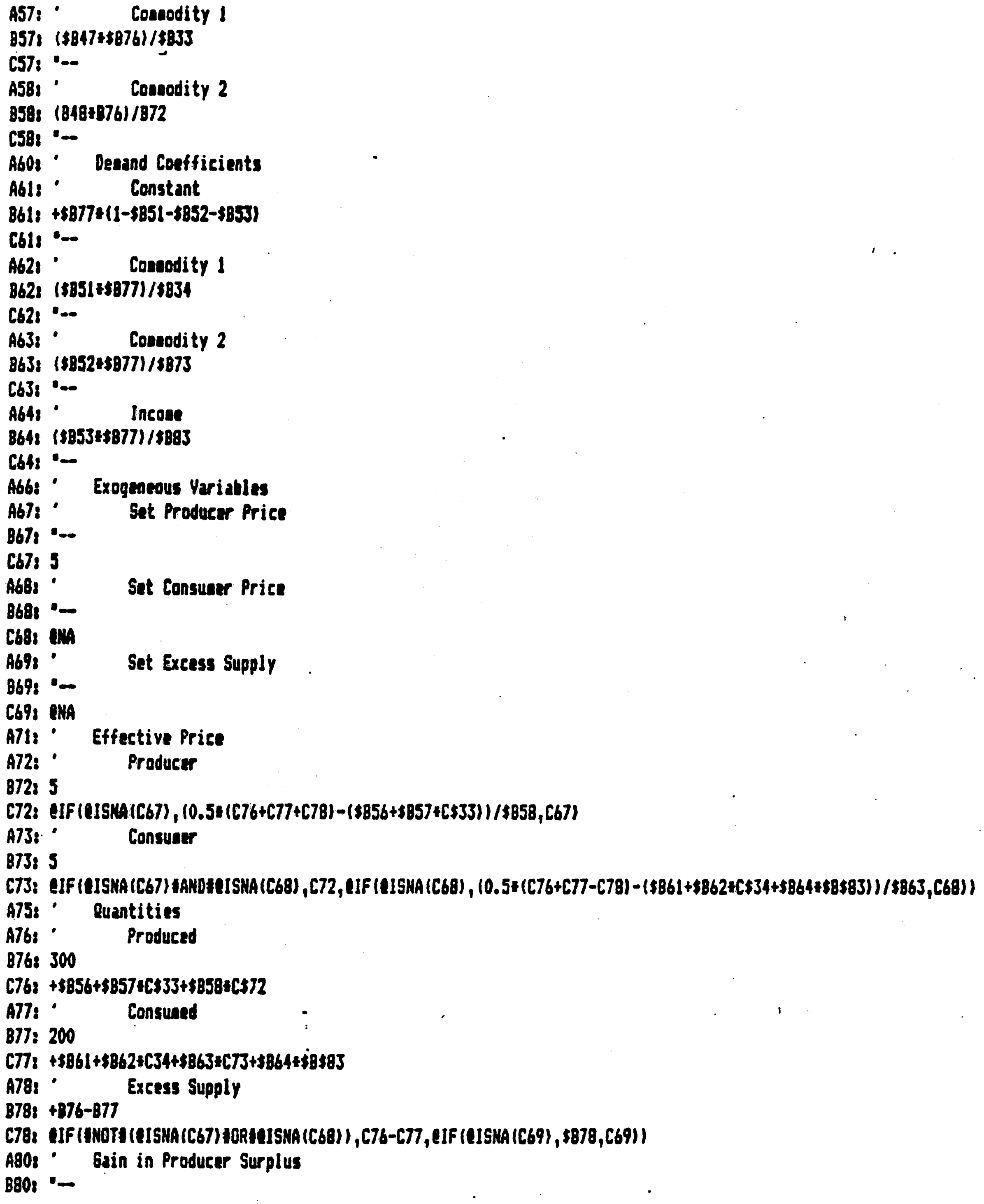
Figure 20. (Cont.)

```
A5: 'Commodity 1
A7: '    Supply Elasticities
A8: '         Commodity 1
B8: 0.9
C8: "--
A9: '         Commodity 2
B9: -0.2
C9: "--
A11: '    Demand Elasticities
A12: '         Commodity 1
B12: -0.7
C12: "--
A13: '         Commodity 2
B13: 0.25
C13: "--
A14: '         Income
B14: 0.5
C14: "--
A16: '    Supply Coefficients
A17: '         Constant
B17: +$B37*(1-$B8-$B9)
C17: "--
A18: '         Commodity 1
B18: ($B8*$B37)/$B33
C18: "--
A19: '         Commodity 2
B19: ($B9*$B37)/$B72
C19: "--
A21: '    Demand Coefficients
A22: '         Constant
B22: +$B38*(1-$B12-$B13-$B14)
C22: "--
A23: '         Commodity 1
B23: (B12*B38)/B34
C23: "--
A24: '         Commodity 2
B24: ($B13*$B38)/$B73
C24: "--
A25: '         Income
B25: ($B14*$B38)/$B83
C25: "--
A27: '    Exogeneous Variables
A28: '         Set Producer Price
B28: "--
C28: 5.5
A29: '         Set Consumer Price
B29: "--
C29: 5.5
```

Figure 21. Cell Listings of the Two Commodity Example.

```
A30: '       Set Excess Supply
B30: "--
C30: @NA
A32: '    Effective Price
A33: '       Producer
B33: 6
C33: @IF(@ISNA(C28),(0.5*(C37+C38+C39)-($B17+$B19*C$72))/$B18,C28)
A34: '       Consumer
B34: 5
C34: @IF(@ISNA(C28)#AND#@ISNA(C29),C33,@IF(@ISNA(C29),(0.5*(C37+C38-C39)-($B22+$B24*C$73+$B25*$B$83))/$B23,C29))
A36: '    Quantities
A37: '       Produced
B37: 100
C37: +$B17+$B18*C$33+$B19*C$72
A38: '       Consumed
B38: 150
C38: +$B22+$B23*C34+$B24*C73+$B25*C$83
A39: '       Excess Supply
B39: +B37-B38
C39: @IF(#NOT#(@ISNA(C28)#OR#@ISNA(C29)),C37-C38,@IF(@ISNA(C30),$B39,C30))
A41: '    Gain in Producer Surplus
B41: "--
C41: 0.5*((C33+($B17+$B19*C72)/$B18)*C37-($B33+($B17+$B19*$B72)/$B18)*$B37)
A42: '    Gain in Consumer Surplus
B42: "--
C42: -0.5*(C34-$B34)*(C38+$B38)-((-0.5*(C34-$B34)*(C38+$B38))^2/(2*$B$83))*$B14+C$83-$B$83
A44: 'Commodity 2
A46: '    Supply Elasticities
A47: '       Commodity 1
B47: -0.1
C47: "--
A48: '       Commodity 2
B48: 0.75
C48: "--
A50: '    Demand Elasticities
A51: '       Commodity 1
B51: 0.3
C51: "--
A52: '       Commodity 2
B52: -0.6
C52: "--
A53: '       Income
B53: 1.5
C53: "--
A55: '    Supply Coefficients
A56: '       Constant
B56: +$B76*(1-$B47-$B48)
C56: "--
```

Figure 21. (Cont.)



```
A57: '         Commodity 1
B57: ($B47*$B76)/$B33
C57: "--
A58: '         Commodity 2
B58: (B48*B76)/B72
C58: "--
A60: '    Demand Coefficients
A61: '         Constant
B61: +$B77*(1-$B51-$B52-$B53)
C61: "--
A62: '         Commodity 1
B62: ($B51*$B77)/$B34
C62: "--
A63: '         Commodity 2
B63: ($B52*$B77)/$B73
C63: "--
A64: '         Income
B64: ($B53*$B77)/$B83
C64: "--
A66: '    Exogeneous Variables
A67: '         Set Producer Price
B67: "--
C67: 5
A68: '         Set Consumer Price
B68: "--
C68: @NA
A69: '         Set Excess Supply
B69: "--
C69: @NA
A71: '    Effective Price
A72: '         Producer
B72: 5
C72: @IF(@ISNA(C67),(0.5*(C76+C77+C78)-($B56+$B57*C$33))/$B58,C67)
A73: '         Consumer
B73: 5
C73: @IF(@ISNA(C67)#AND#@ISNA(C68),C72,@IF(@ISNA(C68),(0.5*(C76+C77-C78)-($B61+$B62*C$34+$B64*$B$83))/$B63,C68))
A75: '    Quantities
A76: '         Produced
B76: 300
C76: +$B56+$B57*C$33+$B58*C$72
A77: '         Consumed
B77: 200
C77: +$B61+$B62*C34+$B63*C73+$B64*$B$83
A78: '         Excess Supply
B78: +B76-B77
C78: @IF(#NOT#(@ISNA(C67)#OR#@ISNA(C68)),C76-C77,@IF(@ISNA(C69),$B78,C69))
A80: '    Gain in Producer Surplus
B80: "--
```

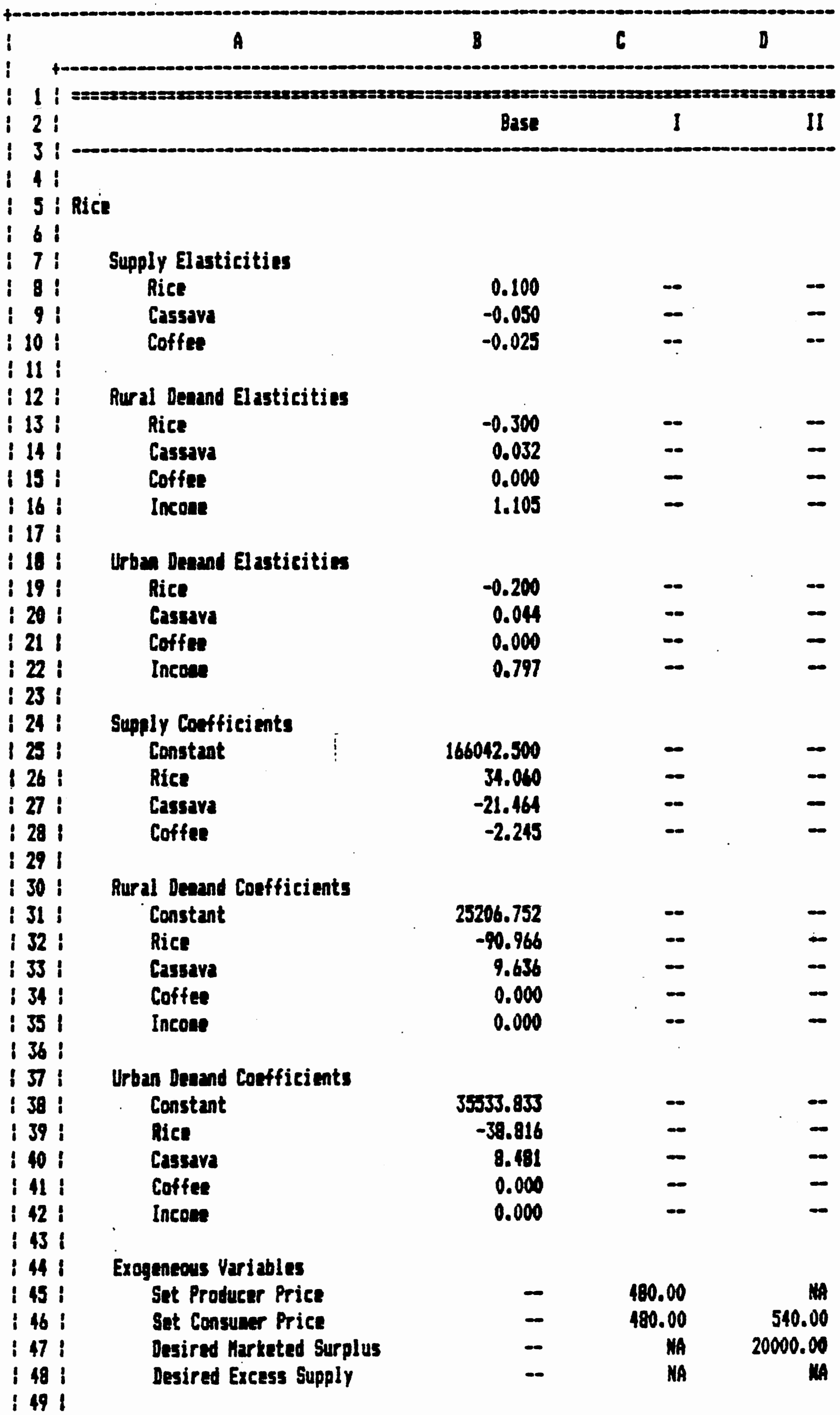
Figure 21. (Cont.)

```
C80: 0.5*((C72+($B56+$B57*C33)/$B58)*C76-($B72+($B56+$B57*$B33)/$B58)*$B76)
A81: '     Gain in Consumer Surplus
B81: "--
C81: -0.5*(C73-$B73)*(C77+$B77)-(-0.5*(C73-$B73)*(C77+$B77))^2/(2*$B83)*$B64+C$83-$B$83
A83: 'Income
B83: 1000
C83: 1000
A85: 'Change in Willingness to Pay
A86: '     Producer
B86: "--
C86: +C41+C80
A87: '     Consumer
B87: "--
C87: +C42+C81
A88: '     Net
B88: "--
C88: +C86+C87
```

Figure 21. (Cont.)

```
+-----------------------------------------------------------------
¦                                 F
¦    +------------------------------------------------------------
¦ 1 ¦ {breakon}
¦ 2 ¦ {let f18,1}
¦ 3 ¦ {if f18>f23}{branch f17}
¦ 4 ¦ {let f18,f18+1}
¦ 5 ¦ {Calc}
¦ 6 ¦ {if @abs(c37-c38-c39)>f22}{branch f3}
¦ 7 ¦ {if @abs(d37-d38-d39)>f22}{branch f3}
¦ 8 ¦ {if @abs(e37-e38-e39)>f22}{branch f3}
¦ 9 ¦ {if @abs(c76-c77-c78)>f22}{branch f3}
¦10 ¦ {if @abs(d76-d77-d78)>f22}{branch f3}
¦11 ¦ {if @abs(e76-e77-e78)>f22}{branch f3}
¦12 ¦
¦13 ¦
¦14 ¦
¦15 ¦
¦16 ¦
¦17 ¦ {beep}{getlabel "Convergence failed, press a key to continue
¦18 ¦                                                             4
¦19 ¦
¦20 ¦
¦21 ¦
¦22 ¦                                                          0.01
¦23 ¦                                                            30
```

Figure 22. Lotus Macro to Monitor Convergence.



```
+------------------------------------------------------------------------------
!                       A                         B             C           D
!    +-------------------------------------------------------------------------
!  1 ! ========================================================================
!  2 !                                           Base            I          II
!  3 ! ------------------------------------------------------------------------
!  4 !
!  5 ! Rice
!  6 !
!  7 !     Supply Elasticities
!  8 !         Rice                              0.100          --          --
!  9 !         Cassava                          -0.050          --          --
! 10 !         Coffee                           -0.025          --          --
! 11 !
! 12 !     Rural Demand Elasticities
! 13 !         Rice                             -0.300          --          --
! 14 !         Cassava                           0.032          --          --
! 15 !         Coffee                            0.000          --          --
! 16 !         Income                            1.105          --          --
! 17 !
! 18 !     Urban Demand Elasticities
! 19 !         Rice                             -0.200          --          --
! 20 !         Cassava                           0.044          --          --
! 21 !         Coffee                            0.000          --          --
! 22 !         Income                            0.797          --          --
! 23 !
! 24 !     Supply Coefficients
! 25 !         Constant                     166042.500          --          --
! 26 !         Rice                             34.060          --          --
! 27 !         Cassava                         -21.464          --          --
! 28 !         Coffee                           -2.245          --          --
! 29 !
! 30 !     Rural Demand Coefficients
! 31 !         Constant                      25206.752          --          --
! 32 !         Rice                            -90.966          --          --
! 33 !         Cassava                           9.636          --          --
! 34 !         Coffee                            0.000          --          --
! 35 !         Income                            0.000          --          --
! 36 !
! 37 !     Urban Demand Coefficients
! 38 !         Constant                      35533.833          --          --
! 39 !         Rice                            -38.816          --          --
! 40 !         Cassava                           8.481          --          --
! 41 !         Coffee                            0.000          --          --
! 42 !         Income                            0.000          --          --
! 43 !
! 44 !     Exogeneous Variables
! 45 !         Set Producer Price                  --      480.00          NA
! 46 !         Set Consumer Price                  --      480.00      540.00
! 47 !         Desired Marketed Surplus            --          NA    20000.00
! 48 !         Desired Excess Supply               --          NA          NA
! 49 !
```

Figure 23. A More Elaborate Example.

```
¦ 50 ¦     Effective Price
¦ 51 ¦         Producer                         500.00        480.00        550.00
¦ 52 ¦         Consumer                         510.00        480.00        540.00
¦ 53 ¦
¦ 54 ¦     Quantities (m tons)
¦ 55 ¦         Production                    170300.00     169618.80     172003.00
¦ 56 ¦         Rural Consumption             154642.65     154464.83     152003.00
¦ 57 ¦         Urban Consumption              98980.04     100144.51      97815.57
¦ 58 ¦         Total Consumption             253622.69     254609.33     249818.57
¦ 59 ¦         Marketed Surplus               15657.35      15153.97      20000.00
¦ 60 ¦         Realized Excess Supply        -83322.69     -84990.53     -77815.57
¦ 61 ¦
¦ 62 ¦     Rural
¦ 63 ¦         Gain from Production                 --   -3058588.00    7706075.00
¦ 64 ¦         Gain from Consumption                --    3061860.72   -7664889.58
¦ 65 ¦         Net Gain                             --       3272.72      41185.42
¦ 66 ¦
¦ 67 ¦     Urban Consumers
¦ 68 ¦         Gain from Consumption                --    2967649.09   -2970706.14
¦ 69 ¦
¦ 70 ¦     Net Gain                                 --    2970921.82   -2929520.72
¦ 71 ¦
¦ 72 ¦ Cassava
¦ 73 ¦
¦ 74 ¦     Supply Elasticities
¦ 75 ¦         Rice                             -0.050            --            --
¦ 76 ¦         Cassava                           0.180            --            --
¦ 77 ¦         Coffee                           -0.050            --            --
¦ 78 ¦
¦ 79 ¦     Rural Demand Elasticities
¦ 80 ¦         Rice                              0.100            --            --
¦ 81 ¦         Cassava                          -0.320            --            --
¦ 82 ¦         Coffee                            0.000            --            --
¦ 83 ¦         Income                            1.105            --            --
¦ 84 ¦
¦ 85 ¦     Urban Demand Elasticities
¦ 86 ¦         Rice                              0.100            --            --
¦ 87 ¦         Cassava                          -0.500            --            --
¦ 88 ¦         Coffee                            0.000            --            --
¦ 89 ¦         Income                            0.626            --            --
¦ 90 ¦
¦ 91 ¦     Supply Coefficients
¦ 92 ¦         Constant                      61364.000            --            --
¦ 93 ¦         Rice                             -6.670            --            --
¦ 94 ¦         Cassava                          30.263            --            --
¦ 95 ¦         Coffee                           -1.759            --            --
¦ 96 ¦
¦ 97 ¦     Rural Demand Coefficients
¦ 98 ¦         Constant                       5023.412            --            --
¦ 99 ¦         Rice                              8.736            --            --
¦100 ¦         Cassava                         -35.234            --            --
¦101 ¦         Coffee                            0.000            --            --
¦102 ¦         Income                            0.000            --            --
¦103 ¦
```

Figure 23. (Cont.)

```
!104 !    Urban Demand Coefficients
!105 !        Constant                      17809.094          --          --
!106 !        Rice                              4.512          --          --
!107 !        Cassava                         -22.403          --          --
!108 !        Coffee                            0.000          --          --
!109 !        Income                            0.000          --          --
!110 !
!111 !    Exogeneous Variables
!112 !        Set Producer Price                  --        396.72      396.72
!113 !        Set Consumer Price                  --        513.53      513.53
!114 !        Desired Marketed Surplus            --            NA          NA
!115 !        Desired Excess Supply               --            NA          NA
!116 !
!117 !    Prices
!118 !        Producer                        396.72        396.72      396.72
!119 !        Consumer                        513.53        513.53      513.53
!120 !
!121 !    Quantities (m tons)
!122 !        Production                    66700.00      66833.40    66366.50
!123 !        Rural Consumption             43681.85      43096.58    45153.02
!124 !        Urban Consumption             23009.17      22873.82    23144.51
!125 !        Total Consumption             66691.01      65970.39    68297.54
!126 !        Marketed Surplus              23018.15      23736.82    21213.48
!127 !        Realized Excess Supply            8.99        863.01    -1931.04
!128 !
!129 !    Rural
!130 !        Gain from Production                --      52922.45   -32306.12
!131 !        Gain from Consumption               --          0.00        0.00
!132 !        Net Gain                            --      52922.45   -32306.12
!133 !
!134 !    Urban Consumers
!135 !        Gain from Consumption               --          0.00        0.00
!136 !
!137 !    Net Gain                                --      52922.45   -32306.12
!138 !
!139 ! Coffee
!140 !
!141 !    Supply Elasticities
!142 !        Rice                             -0.050          --          --
!143 !        Cassava                          -0.050          --          --
!144 !        Coffee                            0.120          --          --
!145 !
!146 !    Rural Demand Elasticities
!147 !        Rice                              0.000          --          --
!148 !        Cassava                           0.000          --          --
!149 !        Coffee                            0.000          --          --
!150 !        Income                            0.000          --          --
!151 !
!152 !    Urban Demand Elasticities
!153 !        Rice                              0.000          --          --
!154 !        Cassava                           0.000          --          --
!155 !        Coffee                            0.000          --          --
!156 !        Income                            0.000          --          --
!157 !
```

Figure 23. (Cont.)

```
!158 !    Supply Coefficients
!159 !        Constant                      9800.000             --             --
!160 !        Rice                            -1.000             --             --
!161 !        Cassava                         -1.260             --             --
!162 !        Coffee                           0.633             --             --
!163 !
!164 !    Rural Demand Coefficients
!165 !        Constant                         0.000             --             --
!166 !        Rice                             0.000             --             --
!167 !        Cassava                          0.000             --             --
!168 !        Coffee                           0.000             --             --
!169 !        Income                           0.000             --             --
!170 !
!171 !    Urban Demand Coefficients
!172 !        Constant                         0.000             --             --
!173 !        Rice                             0.000             --             --
!174 !        Cassava                          0.000             --             --
!175 !        Coffee                           0.000             --             --
!176 !        Income                           0.000             --             --
!177 !
!178 !    Exogeneous Variables
!179 !        Set Producer Price                  NA        1896.10        1896.10
!180 !        Set Consumer Price                  NA        1900.00        1900.00
!181 !        Desired Marketed Surplus                           NA             NA
!182 !        Desired Excess Supply               NA             NA             NA
!183 !
!184 !    Effective Price ($ per a ton)
!185 !        Producer                       1896.10        1896.10        1896.10
!186 !        Consumer                       1900.00        1900.00        1900.00
!187 !
!188 !    Quantities (m tons)
!189 !        Production                    10000.00       10020.00        9950.00
!190 !        Rural Consumption                 0.00           0.00           0.00
!191 !        Urban Consumption                 0.00           0.00           0.00
!192 !        Total Consumption                 0.00           0.00           0.00
!193 !        Marketed Surplus              10000.00       10020.00        9950.00
!194 !        Realized Excess Supply        10000.00       10020.00        9950.00
!195 !
!196 !    Rural
!197 !        Gain from Production               --        37922.02      -44805.06
!198 !        Gain from Consumption              --            0.00           0.00
!199 !        Net Gain                           --        37922.02      -44805.06
!200 !
!201 !    Urban Consumers
!202 !        Gain from Consumption              --            0.00           0.00
!203 !
!204 !    Net Gain                               --        37922.02      -44805.06
!205 !
!206 !
!207 ! Rural Income                  359605000.00   356546412.00   367311075.00
!208 ! Urban Income                  739926000.00   739926000.00   739926000.00
!209 !
```

Figure 23. (Cont.)

```
!210 ! Rural
!211 !     Gain from Production                 -2967743.53    7628963.22
!212 !     Gain from Consumption                 3061860.72   -7664889.58
!213 !     Net Gain                                94117.19     -35926.36
!214 !
!215 ! Urban Consumers
!216 !     Gain from Consumption                 2967649.00    -2970606.14
!217 !
!218 ! Net Gain                                  2789232.76    -2475298.47
```

Figure 23. (Cont.)

CHAPTER VII

SUMMARY AND CONCLUSIONS

Objectives

The objective of this thesis was to investigate the 'simple things' in agricultural policy analysis that can be done with a minimal level of microcomputer expertise. Three viewpoints were taken: user, tool-maker and trainer. The user is the analyst himself. His interests are the microcomputer analysis and informational handling methods that he can not only use, but understand, build and maintain. A tool-maker, on the other hand, is concerned with building tools to extend analysts' microcomputing capabilities, without elevating the requirements in computer expertise. From a trainer's point of view, of interest is the appropriate ingredients of effective training programs on microcomputers for policy analysts in developing countries.

More specifically, the objectives of this thesis were to:

1. Identify simple microcomputer techniques that are useful for small agencies in developing countries and illustrate how these techniques can be used.

2. In particular, one illustration will be an extension of the framework of analysis of impacts of government price intervention policies using consumer and producer surplus to a multicommodity setting. The extension must strike a balance between theoretical soundness and simplicity. The target is an implementation suitable for

a spreadsheet and easily understandable, maintainable, extendible, and adaptable.

3. Identify and discuss the difficulties and design issues in developing software which requires only a minimal amount of computer expertise to operate.

4. Identify the suitable ingredients of microcomputer training programing for policy analysts in developing countries.

Findings

Even with only a minimal amount of computer expertise -- equivalent to about intermediate spreadsheet skills -- agricultural analysis in developing countries can be greatly enhanced with a microcomputer. Among the many useful things in agricultural policy analysis that can be done on microcomputers with only a minimal amount of expertise are data tabulation, linear programming matrix design, financial calculations, and applications in project appraisal.

'Simplicity', as emphasized throughout the dissertation, does not preclude the possibility of elaborate economic modeling. Even the modeling of simultaneous economic relationships, and the computation of welfare impacts to consumers and producers in a multi-commodity setting, can be performed without requiring microcomputer techniques out of reach for typical small agencies.

Simplicity to the user is the result of thoughtful tool making. Simple tools require clever designs and are usually technically demanding to build. The microcomputer programmer usually must work harder than a mainframe programmer to deliver a program with which his users can be comfortable.

Properly conducted short-term microcomputer training can be a cost and time effective way for analysts to overcome learning stumbling blocks. Training programs should aim for providing practical skills which can immediately be applied to each analyst's everyday analysis work; and also stir curiosities, provide background, and build confidence for further self-guided learning on microcomputers.

Limitations and Needs for Further Research

This study is not a scientific survey on microcomputer methods for agricultural policy analysts in developing countries. The microcomputer methods discussed are by no means exhaustive. Data used in some models are hypothetical, aiming for illustration only. The calculation of welfare impacts with consumer surplus and producer surplus falls short of a complete accounting of costs and benefits to society as a whole. Only impacts to consumers and producers are accounted for. Further research is needed to devise a uniform and generally applicable method for computing government costs of agricultural price policies.

In a field which is only at most five years old, this thesis must draw conclusions from limited experiences. The discussion on short-term training presented an approach which was proven effective by actual applications, but other approaches may be effective as well. The tool making effort described in chapter 5 is modest compared to what is possible on a micro. Much research is needed on the overall question of how microcomputers can be made more useful for agricultural policy analysts in developing countries.

Use of Microcomputer in Agricultural Policy Analysis in Developing Countries: Concluding Notes

At present, the technology of microcomputers is undergoing a very rapid evolution. Hardware is changing fast, and software is changing faster. Nevertheless, several stable trends are emerging both in terms of software packages and hardware, and the way software is being applied to agricultural policy analysis in developing countries.

In terms of hardware, most agencies seem to have settled on the IBM PC or compatible machines. Although more powerful machines are already in the market or just around the corner, future machines used by most agencies would at least be downward compatible with the IBM PC.

The most popular software package used are spreadsheet software, in particular Lotus 1-2-3. The thesis demonstrated the type of analysis that can be done with a spreadsheet and their limitations. In the future, more software will likely be spreadsheet-like or spreadsheet-based.

Tools developed for policy analysis by tool-makers in universities or advanced agencies should probably use spreadsheet as a base. If not all the analysis can be performed within the limitations of a spreadsheet, then combine a "black-box" with the spreadsheet as a linkage to the outside world. This methodology is illustrated by Musah86, where the black box is the LP solution algorithm. A listing of Musah86 is provided in Appendix A showing in detail how this can be done. Whichever methodology is used, user-friendliness and easy of operation should be the prime consideration in the development of tools for policy analysis.

I will close this thesis with an optimistic note from professor R. D. Norton (Li and Norton 1985):

> As a profession, we are reaching the stage where time and expertise needed for good policy analysis no longer are bottlenecks. The most important bottleneck now is an inherent and timeless one: our ability to conceptualize a problem in the most useful framework, and to conceive of possible solutions. Machines have evolved sufficiently that we once again are face to face with human possibilities and limitations, which is a very appropriate state of affairs. (p. 9)

REFERENCES

Agrawal, R. C. and Earl O. Heady. Operation Research Methods for Agricultural Decisions. Iowa State University Press, 1972.

Arden, Bruce W. What can be Automated? MIT Press, 1981.

Baer, Jean-Loup. Computer Systems Architecture. Computer Science Press, 1980.

Bardhan, P. K. and K. Bardhan. "Price Response of Marketed Surplus of Food Grains." Oxford Economic Papers. July 1970.

Beneke, R. R. and R. D. Winterboer. Linear Programming Applications to Agriculture. Iowa State University Press, 1973.

Booch, Crady. Software Engineering with ADA. California: Benjamin/Cummings, 1983.

Borland International. Turbo Pascal Version 3.0 Reference Manual. Borland International Inc., 1985.

Brodie, Leo. Thinking Forth. Prentice Hall 1984.

Currie, J. M., J. A. Murphy and A. Schmitz, "The Concept of Economic Surplus and Its Use in Economic Analysis." Economic Journal, 81, 741-749.

Draves, William A. How to Teach Adults. The Learning Resource Network, 1984.

Epplin, Francis and Elton Li. "Introduction to Musah86." Department of Agricultural Economics, Oklahoma State University, 1986.

Feuer, Alan and Narain Gehani. Comparing and Assessing Programming Languages. Bell Laboratories, 1984.

Gittinger, J. Price. Economic Analysis of Agricultural Projects. 2nd Edition. John Hopkins University Press, 1982.

Ghatak, S. and K. A. Igersent. Agriculture and Economic Development. John Hopkins University Press, 1984.

Helmero, Carl. "Returning to the Tower of Babel, or... Some Notes about LISP, Languages and Other Topics...." Byte Magazine, August 1979.

Hicks, J. R. "The Four Consumers' Surplus." Review of Economic Studies, XI, no 1, 1943.

Hicks, J. R. "The Rehabilitation of Consumer's Surplus." Review of Economic Studies, 8, 112.

Hicks, J. R. Value and Capital. Oxford: The Clarendon Press, 1946.

Hildenbrand, W. and A. P. Kirman. Introduction to Equilibrium Analysis. New York: American Elsevier Publishing Co., Inc., 1976.

Jackson, Philip W. The Practice of Teaching. Teachers College Press, 1986.

Johnson, S. R., Zuhair A. Hassan and Richard Green. Demand Systems Application. Iowa State University Press, 1984.

Just, R. E., D. L. Hueth and Andrew Schmitz. Applied Welfare Economics and Public Policy. Prentice Hall, 1982.

Kemeny, John G. and Thomas E. Kurtz. Back to BASIC, Addison-Wesley, 1985.

Krishna, R. "Farm Supply Response in India and Pakistan: A Case Study of the Punjab Regions." Economic Journal, vol. 73, 1963.

Lammers, Susan. Programmers at Work. MicroSoft Press, 1986.

Leftwich, Richard H. The Price System and Resource Allocation. 7th edition. The Dryden Press, 1979.

Lenz, Elinor. The Art of Teaching Adults. CBS College Publishing, N.Y. 1982.

Li, Elton. "Tutorial Introduction to Musah86." APAP, Department of Agricultural Economics, Oklahoma State University, 1984.

Li, Elton and Daryll E. Ray. "Using Electronic Spreadsheets for Designing LP Tableaus and for Transmission to LP Programs." Dept. of Agricultural Economics, Oklahoma State University. A.E. 8591.

Li, Elton and Roger D. Norton. "The Role of Microcomputers in Agricultural Policy Analysis in Developing Countries." Department of Agricultural Economics, Oklahoma State University, Nov., 1985.

Lotus. Lotus 1-2-3 Release 2 Reference Manual. Lotus Corporation, 1985.

Marshall, Alfred. Principles of Economics. London: McMillan, 1930.

McWilliams, Peter A. The Personal Computer in Business Book. New York: Quantum Press, 1984.

Mishan, E. J. "What is Producers' Surplus?" American Economic Review. 58(1968):1279-82.

Mishan, E. J. Cost Benefit Analysis, New and Expanded Edition. Praeger Publishers, 1976.

Mishan, E. J. Introduction to Normative Economics. Oxford Univ. Press, 1981.

Quade, E. S. Analysis for Public Decisions. 2nd ed., New York: Elsevier Science Publishing Co., Inc., 1982.

Quirk, James and Rubin Saposnik. Introduction to General Equilibrium Theory and Welfare Economics. McGraw-Hill, 1968

Reutlinger, Shlomo and Keith Knapp. Food Security in Deficit Countries. World Bank Staff Working paper No. 393, June 1980.

Sharda, Ramesh. "Linear Programming on Microcomputers: A Survey." Interfaces 14:6 Dec., 1984.

Sharda, Ramesh. "Optimization Using Spreadsheets on a Microcomputer." Anuals of Operations Research, Vol. 5, pp. 599-612, 1986.

Sharda, Ramesh. "State-of-the-Art Linear Programming on Personal Computers." College of Business Adminstration, Oklahoma State University. 1986.

Sharpe, William F. "Inside 1-2-3 Worksheet Files." PC Tech Journal, Vol. 2, No. 4, Oct, 1984.

Simpson, Henry. Design of User-Friendly Programs for Small Computers. McGraw-Hill 1985.

Scandizzo, P. L. and Colin Bruce. Methodologies for Measuring Agricultural Price Intervention Effects. World Bank Staff Working Paper No. 394.

Snark, B. Hungtington. "Diverse Approaches to Teaching Programming: Three Reports." In W. M. Turski (eds) Programming Teaching Techniques. New York: American Elsevier 1973.

Snyder, Robert E. Guide to Teaching Techniques for Adult Classes. New York: Prentice-Hall 1972.

Timmer, C. P., W. P. Falcon and S. R. Pearson. Food Policy Analysis. Baltimore: John Hopkins University Press, 1983.

Tweeten, Luther. Foundations of Farm Policy. Lincoln: University of Nebraska Press, 1979.

Tweeten, Luther and Boima Rogers. "Costs, Benefits, and Income Redistribution from Liberian Rice Policies." Proceedings of the National Agricultural Policy Seminar held March 26-29, 1985 at Yakepa, Liberia. Stillwater: APAP, Department of Agricultural Economics, Oklahoma State University 1985.

Tweeten, Luther. "Introduction to Agricultural Policy Analysis: The Analytical Program and Tools." Report B-20. Department of Agricultural Economics, Oklahoma State University, June 1986.

Varian, H. R. Microeconomic Analysis. 2nd Edition. Norton, 1984.

Walsh, V. C. Introduction to Contemporary Microeconomics. McGraw-Hill, 1970.

Willig, R. D. "Consumer's Surplus Without Apology." American Economic Review. 66, no. 4, 589-97.

APPENDIX A

LISTING OF MUSAH86

```
{$K-,V-,R-,C+}

{=============================================================================
|
|    Musah86 v2.0
|    by Elton Li
|
+----------------------------------------------------------------------------}

program Musah86;
  const
    SIGNITURE =
      'Copyright 1985.  Elton Li, Dept Ag Econ, Okla St Univ';

    ConLimit       =  255;            { Max # of constraints              }
    RealVarLimit   =  255;            { Max # of non-slack/artifical vars. }
    VarLimit       =  770;            { Max # of variables                }
    Tolerence      =  1.0E-10;        { How small must a number be to be 0 }
    MinusTolerence = -1.0E-10;        { Negative of tolerence             }
    Space          =  ' ';            { One space                         }
    NullString     =  '';             { Null string                       }
    BigM           =  1.0E+09;        { Big M used for artifical var cost  }
    MinusBigM      = -1.0E+09;        { Negative of BigM                  }
    BufferLimit    =  128;            { Length of I/O buffer              }
    criticalError: boolean = false;  { Has critical i/O error occurred?  }
  type
    VarRange      = 1..VarLimit;
    VarRange1     = 0..VarLimit;
    ConRange      = 1..ConLimit;
    ConRange1     = 0..ConLimit;
    RealVarRange  = 1..RealVarLimit;
    RealVarRange1= 0..RealVarLimit;
    SString       = string[30];
    LString       = string[255];
    MatrixCol     = array[ConRange] OF real;
    Matrix        = array[VarRange] OF ^Matrixcol;
    VarPtrs       = array[VarRange] OF integer;
    ConPtrs       = array[ConRange] OF integer;
    TableauRow    = array[VarRange] OF real;
    TableauCol    = array[ConRange] OF real;
  var
    realLength:     integer;       { 6 bytes for turbo, 8 for turbo w 8087    }
    infile:         FILE;          { input Lotus wk1 file                     }
    outFile:        FILE;          { output Lotus wk1 file                    }
    inFileName:     SString;       { Name of input lotus file                 }
    outFileName:    SString;       { Name of output lotus file                }
    basisNo:        ConPtrs;       { Number basis                             }
    cost:           TableauRow;    { Cost vector                              }
    R:              TableauCol;    { right hand side of tableau               }
    rhs:            TableauCol;    { original right hand side of tableau      }
    A:              Matrix;        { A matrix of tableau                      }
    no8087:         boolean;       { True if 8087 version is NOT used         }
    outputError:    boolean;       { True if output error had occured         }
    inputError:     boolean;       { True if input error had occured          }
    endOfWks:       boolean;       { True if end of worksheet while input     }
    maximize:       boolean;       { True if problem is maximize, false if min}
    objLevel:       real;          { Value of the objective function          }
    probName:       SString;       { Name of the problem                      }
    objName:        SString;       { Name of the objective function           }
    rhsName:        SString;       { Name of the right hand side              }
    objective:      SString;       { Char string holding Maximize or Minimize }
    numRealAct:     RealVarRange1;{ # real (not slack or artifical) act.     }
    numNonArtVar:   VarRange1;     { # non-artifical variables                }
    numAct:         VarRange1;     { Total # of activities                    }
    numCon:         ConRange1;     { Total # of constraints                   }
    numLessThan:    ConRange1;     { # of less than constraints               }
    numArtVar:      ConRange1;     { # of artifical variables                 }
    numEqual:       ConRange1;     { # of equality constraints                }
    numGreaterThan: ConRange1;     { # of greater than constraints            }
    wksCol:         integer;       { # of Cols in the input spreadsheet       }
  wksRow:         integer;      { # of Rows in the input spreadsheet        }
  z:              TableauRow;   { Reduced cost (Zj) row                     }
  shadow:         TableauRow;   { Shadow price (Cj-Zj) row                  }
  basis:          VarPtrs;      { Array indicating order of basis           }
  bCount:         integer;      { Count for basis                           }
  offset:         integer;      { offset used in outputing tableau          }
  ioErrorCode:    integer;      { I/O error code                            }
  another:        char;         { Y or N answer for wanting another tab.    }
  actName:                      { Array of activity names                   }
    array[VarRange] OF SString;
  conType:                      { Array of constranit types L, G, E         }
    array[ConRange] OF SString;
  finalRow:                     { Index array to order row in final tab.    }
    array[ConRange] OF integer;
  finalCol:                     { Indes array to order col in final tab.    }
    array[RealVarRange] OF integer;
  buffer:                       { I/O buffer                                }
    array[1..128] OF byte;
  outPutIndex:    byte;         { Output buffer index                       }
  inputIndex:     byte;         { Input buffer index                        }
  heapPtr:        ^integer;     { Heap ptr for dynamic managment of A mat.  }

  intRec:
    record
      case boolean OF
        true:  (bite: array[1..2] OF byte);
        false: (int: integer)
    end;

  floatRec:
    record
      case boolean OF
        true:  (bite: array[1..8] OF byte);
        false: (float: real)
    end;


{============================================================================
|
|   Miscelleneous Global Procedures
|
+---------------------------------------------------------------------------}

function ToUpper(strg: LString): LString;
  {
  Converts strg from lower case to upper case if necessary
  }
  var
    i: integer;
    TempStrg: LString;
  begin
    TempStrg := '';
    for i:=1 to length(strg) do
      TempStrg := TempStrg + UpCase(Copy(strg,i,1));
    ToUpper := TempStrg
  end { ToUpper };

function Bell: char;
  {
  Produce a "pleasant" bell tone.
  }
  procedure Ring(soundTime,soundDelay: integer);
    begin
      Sound(soundTime);
      Delay(soundDelay);
      NoSound
    end { Ring };

  begin { Bell }
    Ring(660,15);
    Ring(330,90);
    Ring(165,1);
```

```
      Ring(330,90);
      Ring(660,15);
      Ring(1760,1);
      Bell := Space
    end { Bell };

  procedure FatalError(Message: LString);
    begin
      ClrScr;
      Write(Message);
      Halt(9)
    end;

  procedure StoreA(i,j: integer; num: real);
    {
    Store num into (i,j)th element of A matrix.  In order to overcome the 64K
    limit of Turbo Pascal, the A matrix is allocated on the heap.  The (i,j)th
    element of A is A[j]^[i].  Thus A is an array of pointers to an array of
    real.  Thus each element of A[j] points to a row of the matrix.  The whole
    col is allocated when the first element is stored.
    }
    var
      MemoryAvailable: real;
    begin
      if A[j] = nil then
        begin
          MemoryAvailable := MaxAvail;
          if MemoryAvailable < 0 then
            MemoryAvailable := MemoryAvailable + 65536.0;
          MemoryAvailable := MemoryAvailable * 16;
          if realLength*numCon >= MemoryAvailable then
            FatalError('Insufficient memory for tableau, program terminated!');
          GetMem(A[j],realLength*numCon);
          FillChar(A[j]^[1],realLength*numCon,$00)
        end;
      A[j]^[i] := num
    end { StoreA };

  procedure DFrame(bright: boolean);
    {
    Paint Screen
    }
    var i: integer;
    begin
      LowVideo;
      GotoXY(1,1); Write('(C) 1985, Dept Agricultural Economics, ');
      Write('Oklahoma State Univ. Stillwater, OK, USA.');
      HighVideo;
      GotoXY(1,2);  for i:=1 to 80 do Write('=');
      GotoXY(1,21); for i:=1 to 80 do Write('-');
      if bright then HighVideo else LowVideo;
      GotoXY(05,04); Write('Problem Name........');
      GotoXY(05,05); Write('Objective...........');
      GotoXY(05,06); Write('Objective Name......');
      GotoXY(05,07); Write('Rhs Name............');
      GotoXY(05,09); Write('Total Columns.......');
      GotoXY(05,10); Write('  Real..............');
      GotoXY(05,11); Write('  Slack.............');
      GotoXY(05,12); Write('  Artificial........');
      GotoXY(05,13); Write('Total Constraints...');
      GotoXY(05,14); Write('  Less Than.........');
      GotoXY(05,15); Write('  Greater Than......');
      GotoXY(05,16); Write('  Equality..........');
      GotoXY(05,18); Write('Input File..........');
      GotoXY(05,19); Write('Output File.........');
      if bright then LowVideo else HighVideo;
      GotoXY(45,04); Write('Iteration...........');
      GotoXY(45,05); Write('Solution............');
      GotoXY(45,06); Write('Activity In.........');
      GotoXY(45,07); Write('Activity Out........');
      GotoXY(45,08); Write('Objective Value.....');
      HighVideo
  end { DFrame };

function FileName(fName: SString): SString;
  {
  Strip spaces and extension off fName and append wk1 as extension
  }
  var
    i:    integer;
    quit: boolean;
    strg: SString;
    ch: char;

  procedure Initialize;
    begin
      i     := 0;
      strg  := '';
      quit  := false;
      fname := fname + Space
    end { Initialize };

  procedure GetFileNameChar;
    begin
      i  := succ(i);
      ch := copy(fname,i,1);
      if ch = '.' then
        quit := true
      else if ch <> Space then
        strg := strg + ch;
      if i >= length(fname) then quit := true
    end   { GetFileNameChar };

  begin { FileName }
    Initialize;
    while not quit do GetFileNameChar;
    FileName := strg
  end { FileName };

procedure CleanWindow;
  {
  Clear message area of screen.
  }
  begin
    GotoXY(1,23); ClrEol;
    GotoXY(1,22); ClrEol
  end { CleanWindow };

procedure HandleIOError;
  {
  Instead of letting dos handle the critical io error (DOS 2.x), set a
  global flag and handle the error in the program instead.
  }
  begin
    inline( $5D/           { POP  BP ; pop twice to bypass turbo           }
            $5D/           { POP  BP ; procedure interface !               }
            $FB);          { STI                                           }
    criticalError := true; { Signal critical i/o err had occured           }
    inline( $58/           { POP  AX ; CS:IP and Flag of Int 24H caller    }
            $58/           { POP  AX ;   so that IRET will be to original  }
            $58/           { POP  AX ;   int 21H caller                    }
            $58/           { POP  AX ; restore original int 21H            }
            $5B/           { POP  BX ; caller's registers                  }
            $59/           { POP  CX                                       }
            $5A/           { POP  DX                                       }
            $5E/           { POP  SI                                       }
            $5F/           { POP  DI                                       }
            $5D/           { POP  BP                                       }
            $1F/           { POP  DS                                       }
            $07/           { POP  ES                                       }
            $80/$FF/       { MOV  AL,FF                                    }
            $CF);          { IRET                                          }
  end { HandleIOError };
```

```
procedure SetInterruptVector;
  {
  Point DOS interrupt vector for INT 24H to my own error handler routine
  }
  type
    regPack = record
                ax,bx,cx,dx,bp,di,si,ds,es,flags: integer
              end;
  var
    recPack: regPack;
  begin
    with recPack do
      begin
        ax := $2524; ds := CSeg; dx := Ofs(HandleIOError);
        msDos(recPack)
      end
  end { SetInterruptVector };

procedure GetSpace;
  {
  Wait for user to press space bar
  }
  var ch: char;
  begin
    repeat
      Write(bell,chr(8));
      read(KBD,ch)
    until ch=Space
  end { GetSpace };

procedure Initialize;
  {
  Program level initialization.
  }
  var
    i,j: integer;
  begin
    ClrScr; DFrame(true); GotoXY(1,22); Write('One Moment Please...');
    for j := 1 to VarLimit do
      begin
        actName[j] := '?';
        cost[j]    := 0.0
      end;
    for i := 1 to ConLimit do
      begin
        conType[i] := 'L';
        R[i]       := 0.0;
        rhs[i]     := 0.0;
        A[i]       := nil
      end;
    probName       := 'ProbName?';
    objName        := 'ObjName?';
    rhsName        := 'RhsName?';
    Objective      := 'Maximize';
    realLength     := SizeOf(real);
    no8087         := (realLength=6);
    outputIndex    := 0;
    inputIndex     := 128;
    ioErrorCode    := 0;
    outPutError    := false;
    numArtVar      := 0;
    numEqual       := 0;
    numGreaterThan := 0;
    numLessThan    := 0;
    inFileName     := '';
    outFileName    := '';
    offset         := 0;
  end { Initialize };

procedure ComputeShadowPrices;
  {
  Compute Zj and Zj-Cj rows
  }
  var
    i, j: integer;
    sum:  real;
  begin
    for j := 1 to numAct do
      if basis[j] = 0 then
        begin
          sum := 0.0;
          for i := 1 to numCon do
            sum := sum + A[j]^[i] * cost[basisNo[i]];
          z[j] := sum;
          shadow[j] := sum - cost[j]
        end
      else
        begin
          z[j] := cost[j];
          shadow[j] := 0.0
        end
  end { ComputeShadowPrices };

procedure DFrame1;
  {
  Put on the screen information prior to iteration.
  }
  begin
    GotoXY(25,04); Write(probName);
    GotoXY(25,05); Write(objective);
    GotoXY(25,06); Write(objName);
    GotoXY(25,07); Write(rhsName);
    GotoXY(25,09); Write(numAct);
    GotoXY(25,10); Write(numRealAct);
    GotoXY(25,11); Write(numLessThan+numGreaterThan);
    GotoXY(25,12); Write(numArtVar);
    GotoXY(25,13); Write(numCon);
    GotoXY(25,14); Write(numLessThan);
    GotoXY(25,15); Write(numGreaterThan);
    GotoXY(25,16); Write(numEqual);
    GotoXY(25,18); Write(inFileName);
    GotoXY(25,19); Write(outFileName);
    HighVideo
  end { DFrame1 };


{==========================================================================
|
|   Common utilities for inputing and setting up tableau
|
+-------------------------------------------------------------------------}

function GetByte: byte;
  {
  Read in a byte from lotus wk1 file
  }
  var bite: byte;

  procedure HandleCriticalError;
    {
    Override dos' handling of drive not ready
    }
    begin
      CleanWindow; Write('Disk Drive is not ready ... ',
        'hit <space> key to continue...');
      GetSpace;
      criticalError := false;
      EndOfWks   := true;
      inputError := true
    end { HandleCriticalError };

  procedure HandleIOError;
    {
    Handle "non-critical" Input Error
    }
    begin
      inputError := true;
      EndOfWks   := true;
      CleanWindow; Write('Input error, hit <space> key to continue...');
      GetSpace
    end { HandleIOError };

  begin { GetByte }
    if not inputError then
      begin
        {
        if buffer exhaused read in another, otherwise return next byte
        in buffer.
        }
        if inputIndex = BufferLimit then
          begin
            inputIndex := 1;
            {$I-} blockread(inFile,buffer,1); {$I+}
            ioErrorCode := IOResult;
            if criticalError then
              HandleCriticalError
            else if ioErrorCode <> 0 then
              HandleIOError
          end
        else
          inputIndex := succ(inputIndex);
        GetByte := buffer[inputIndex]
      end
  end { GetByte };

function GetInt: integer;
  {
  Get an integer from lotus wk1 file by calling GetByte twice
  }
  begin
    intRec.bite[1] := GetByte;
    intRec.bite[2]:= GetByte;
    GetInt := intRec.int
  end { GetInt };

function GetFloat: real;
  {
  Get an real number from lotus tableau, resolve imcapatibility if
  necessary.
  }
  var i: integer;
  begin
    for i := 1 to 8 do FloatRec.bite[i] := GetByte;
    {
    If non 8087 version, then must convert 6 byte real number
    representation to the IEEE format required by Lotus.
    }
    if no8087 then
      inline(
        $B1/$05/          {           MOV  CL,5      }
        $B5/$00/          {           MOV  CH,0      }
        $BE/$06/$00/      {           MOV  SI,6      }
        $8B/$84/floatrec/ {           MOV  AX,FR[SI] }
        $D1/$E0/          {           SHL  AX,1      }
        $73/$02/          {           JAE  POS       }
        $B5/$80/          {           MOV  CH,128    }
        $D3/$E8/          { POS:      SHR  AX,CL     }
        $3D/$00/$00/      {           CMP  AX,0      }
        $75/$13/          {           JNZ  NOTZERO   }
        $A3/floatrec/     {           MOV  FR,AX     }
        $BF/$02/$00/      {           MOV  DI,2      }
        $89/$85/floatrec/ {           MOV  FR,AX     }
        $BF/$04/$00/      {           MOV  DI,4      }
        $89/$85/floatrec/ {           MOV  FR,AX     }
        $EB/$4B/          {           JMP  DONE      }
        $2D/$7E/$03/      { NOTZERO: SUB  AX,894    }
        $A2/floatrec/     {         MOV  FR,AL     }
        $BF/$01/$00/      {         MOV  DI,1      }
        $8B/$85/floatrec/ {         MOV  AX,FR[DI] }
        $D3/$E8/          {         SHR  AX,CL     }
        $88/$85/floatrec/ {         MOV  FR[DI],AL }
        $BF/$02/$00/      {         MOV  DI,2      }
        $8B/$85/floatrec/ {         MOV  AX,FR[DI] }
        $D3/$E8/          {         SHR  AX,CL     }
        $88/$85/floatrec/ {         MOV  FR[DI],AL }
        $BF/$03/$00/      {         MOV  DI,3      }
        $8B/$85/floatrec/ {         MOV  AX,FR[DI] }
        $D3/$E8/          {         SHR  AX,CL     }
        $88/$85/floatrec/ {         MOV  FR[DI],AL }
        $BF/$04/$00/      {         MOV  DI,4      }
        $8B/$85/floatrec/ {         MOV  AX,FR[DI] }
        $D3/$E8/          {         SHR  AX,CL     }
        $88/$85/floatrec/ {         MOV  FR[DI],AL }
        $BF/$05/$00/      {         MOV  DI,5      }
        $8B/$85/floatrec/ {         MOV  AX,FR[DI] }
        $D3/$E8/          {         SHR  AX,CL     }
        $24/$7F/          {         AND  AL,127    }
        $0A/$C5/          {         OR   AL,CH     }
        $88/$85/floatrec);{         MOV  FR[DI],AL }
                          { DONE:                  }
    GetFloat := FloatRec.float
  end { GetFloat };



{==========================================================================
|
|   Input and build initial Lp tableau from Lotus worksheet file
|
+-------------------------------------------------------------------------}

procedure SetupInput;
  {
  This procedure opens the lotus file and checks whether its header is a
  valid lotus header.
  }
  procedure GetInfileName;
    begin { GetInfileName }
      repeat
        GotoXY(25,18); ClrEol;
        CleanWindow; Write('Please specify the name of the input file-->',bell);
        readln(inFileName); inFileName := FileName(inFileName);
        GotoXY(25,18); Write(inFileName);
        CleanWindow; Write('Reading ',inFileName,'...')
      until inFileName <> NullString
    end { GetInfileName };

  procedure HandleCriticalError;
    {
    Handle critical open error.
    }
    begin
      CleanWindow; Write('Disk Drive is not ready, ',
         'hit <space> key to continue...');
      GetSpace; criticalError := false
    end { HandleCriticalError };

  procedure HandleIOError;
    {
    Handle non critical open error.
    }
    begin
      CleanWindow; Write('Error ',ioErrorCode,': ',
         'Cannot find file ',inFileName,
         ', hit <space> key to continue...');
      GetSpace
    end { HandleIOError };

    procedure NotLotus;
      {
      Display error if not a valid lotus file.
      }
      begin
        inputError := true;
        CleanWindow; Write(inFileName,' is not a Lotus 1-2-3 worksheet file, ',
          'hit any key to continue',bell); GetSpace
      end { NotLotus };

    begin
      repeat
        InputIndex := BufferLimit;
        GetInfileName;
        Assign(Infile,inFileName+'.WK1');
        {$I-} reset(Infile); {$I+}
        ioErrorCode := IOresult;
        inputError := (ioErrorCode <> 0) or criticalError;
        if criticalError then
          HandleCriticalError
        else if inputError then
          HandleIOError;
        if not inputError then
          if ((getInt<>0) or (getInt<>2) or (getByte<>6) or (getByte<>4)) then
            NotLotus
      until not inputError
    end { SetupInput };

  procedure ReadTableau;
    {
    Input LP tableau from Lotus file
    }
    var
      i:         integer;
      bite:      byte;
      recType:   integer;
      recFormat: byte;
      recLength: integer;
      fromRow:   integer;
      fromCol:   integer;
      toRow:     integer;
      toCol:     integer;
      row, col:  integer;

    procedure error(no:integer);
      {
      Report error in tableau
      }
      begin
        CleanWindow;
        WriteLn('error at row: ',row,', column: ',col,
          Space,recType,Space,recFormat);
        Write('  Hit <space> to continue...',bell);  GetSpace
      end { error };

    function StoreNumber(int: boolean): boolean;
      {
      Store a real number into the tableau, return false if disk error.
      }
      var
        num: real;
      begin
        StoreNumber := true;
        if not inputError then
          begin
            if int then num := getInt else num := getFloat;
            if (col > 3) and (row > 2) then      { Element of Aij }
              StoreA(row - 2,col - 3,num)
            else if (col > 3) and (row = 2) then { Element of Cj  }
              cost[col - 3] := num
            else if (col = 3) and (row > 2) then { Element of RHS }
              begin
            R[row - 2] := num;
            rhs[row-2] := num
          end
        else StoreNumber := false
      end
  end { StoreNumber };

function StoreString: boolean;
  {
  Store a string into the tableau, return false if disk error
  }
  var
    strg:  SString;
    i:     integer;
    bite:  byte;
    ch:    char;
  begin { StoreString }
    strg := NullString;
    bite := GetByte;
    if inputError then
      StoreString := true
    else
      begin
        for i := 1 to recLength - 7 do
          begin
            ch := chr(GetByte);
            if ch <> Space then strg := strg + ch
          end;
        bite:= GetByte;
        if  (row <= wksRow) and (col <= wksCol) then
          begin
            if row = 1 then
              case col OF
                1:   probName       := strg; { cell A1 is problem name   }
                2:   objective      := strg; { cell B1 is min or max     }
                3:   rhsName        := strg; { cell C1 is RHS name       }
                else actName[col-3] := strg  { Rest of Row1 is act name  }
              end
            else if (row = 2) and (col = 1) then
              objName := strg                { cell B1 is obj name       }
            else if col = 1 then             { rest of col is con. name  }
              actName[numRealAct + row - 2 ] := strg
            else if (col = 2) and (row <> 2) then
              conType[row - 2] := strg;      { col B is constrant type   }
            storeString := true
          end
        else
          storeString := false
      end
  end { storeString };

begin
  EndOfWks := false;
  repeat
    recType   := GetInt;
    recLength := GetInt;
    case RecType OF
      1:  EndOfWks := true;               { End wksheet marker found     }
      6:
        begin                             { Range record is type 6       }
          fromCol := getInt;              { Get upper left coodinate     }
          fromRow := getInt;              {   of worksheet               }
          toCol   := getInt;              { Get lower right coodinate    }
          toRow   := getInt;              {   of worksheet               }
          wksCol  := toCol - fromCol + 1; { Number of cols in wksheet    }
          wksRow  := toRow - fromRow + 1; { Number of rows in wksheet    }
          numRealAct := wksCol - 3;       { # real acts. in LP tableau   }
          numCon  := wksRow -2;           { # constraints in LP tab.     }
          numAct  := numRealAct + numCon; { Total number of activities   }
          if numRealAct > realVarLimit then
            FatalError('Too many columns in tableau, program terminated!');
          if numCon > conLimit then
```

```
                  FatalError('Too many rows in tableau, program terminated!');
              end;                              {   before adding artificals   }
            13,14:                              { Integer or Reals are types   }
              begin                             {   13 and 14 respectively     }
                recFormat := GetByte;           { Skip record format           }
                col := GetInt + 1;              { Column coordinate            }
                row := GetInt + 1;              { Row coordinate               }
                if not storeNumber(recType = 13){ Store number & signal error  }
                  then error(0)                 {   if location not supposed   }
              end;                              {   to have number             }
            15:
              begin                             { Char Strings is type 15      }
                recFormat := GetByte;           { Ignore record format         }
                col := GetInt + 1;              { Get column coordinate        }
                row := GetInt + 1;              { Get row coordinate           }
                if not storeString              { Signal error if location not }
                  then error(0)                 {   supposed to have string    }
              end;
            16:
              begin                             { Formula is type 16           }
                recFormat := GetByte;           { Ignore record format         }
                col := GetInt + 1;              { Get column coordinate        }
                row := GetInt + 1;              { Get row coordinate           }
                if not storeNumber(false)       { Signal error if location not }
                  then  error(0);               {   supposed to have number    }
                for i := 1 to recLength - 13 do { We just need the value of    }
                  bite := getByte               {   the formula and not the    }
              end;                              {   formula itself, so skip it }
            else
              for i := 1 to recLength do        { Don't need any other types   }
                bite := getByte                 {   of record, so skip it      }
          end { case }
        until EndOfWks
      end { ReadTableau };

    procedure SetupLpTableau;
      {
      Setup the internal LP tableau after reading in from lotus
      }
      var
        strg: SString;
        i,j: integer;

      procedure GreaterThan;
        {
        Handle GreaterThan or equal to constraints
        }
        begin
          numArtVar      := succ(numArtVar);
          numGreaterThan := succ(numGreaterThan);
          conType[i]     := 'G';
          StoreA(i,numRealAct + i,-1.0);              { Slack variable     }
          StoreA(i,numAct + numGreaterThan,1.0);      { Artifical variable }
          cost[numRealAct + i] := 0.0;                { Cost for slack     }
          if maximize then                            { Cost for artifical }
            cost[numAct + numGreaterThan] := MinusBigM
          else
            cost[numAct + numGreaterThan] := BigM;
          actName[numAct + numGreaterThan] := 'ARTIFICAL';
          bCount := succ(bCount);                     { Set up basis      }
          basisNo[bCount] := numAct + numGreaterThan;
          basis[numAct + numGreaterThan] := bCount
        end { GreaterThan };

      procedure EqualTo;
        {
        Handle equality constraints
        }
        begin
          numEqual   := succ(numEqual);
          numArtVar  := succ(numArtVar);
          conType[i] := 'E';
        if maximize then
          Cost[numRealAct + i] := MinusBigM
        else
          Cost[numRealAct + i] := BigM;
        StoreA(i,numRealAct + i,1.0);
        bCount := succ(bCount);
        basisNo[bCount] := numRealAct + i;
        basis[numRealAct + i] := bCount
      end { EqualTo };

  procedure LessThan;
    {
    Handle Less than or equal to constraints
    }
    begin
      numLessThan := succ(numLessThan);
      conType[i] := 'L';
      Cost[numRealAct + i] := 0.0;
      bCount := succ(bCount);
      basisNo[bCount] := numRealAct + i;
      StoreA(i,numRealAct + i,1.0);
      basis[numRealAct + i] := bCount
    end { LessThan };

  begin
    CleanWindow; Write('One Moment Please...');
    {
    different set up for min or max
    }
    strg := objective;
    if pos('MIN',ToUpper(strg)) = 0 then
      begin { Maximization assumed }
        objective := 'MAXIMIZE';
        maximize := TRUE
      end
    else
      begin
        objective := 'MINIMIZE';
        maximize := false
      end;
    numGreaterThan := 0; bCount:= 0;
    for j := 1 to VarLimit do basis[j] := 0; { basis indicator }
    for i := 1 to numCon do
      begin
        strg := ToUpper(conType[i]);
        if (pos('G',strg)<>0) then
          GreaterThan
        else if (pos('E',strg)<>0) and (pos('L',strg)=0) then
          EqualTo
        else
          LessThan
      end;
    numNonArtVar := numAct;
    numAct := numAct + numGreaterThan; { Total # activities includes     }
                                       {   artificials for >= constraints}
    ComputeShadowPrices;
    if maximize then
      objLevel := MinusBigM * BigM
    else
      objLevel := BigM * BigM;
    DFrame1
  end { SetupLpTableau };



{=============================================================================
!
!   Common utilities for outputing initial and final tableaus
!
+----------------------------------------------------------------------------}

procedure HandleOutputError(n: integer);
```

```
    {
    Handle non critical output error
    }
    begin
      case n OF
        240:
          Write('Insufficient space on disk, output file (',
            outFileName,') is not stored!!!');
        241:
          Write('Disk directory is full, output file (',
            outFileName,') is not stored!!!');
        else
          Write('I/O error ',ioErrorCode,', output file (',
            outFileName,') is not stored!!!');
      end { case };
      close(outFile);
      outputError := true;
      Write('  Hit <space> key to continue...'); GetSpace
    end { HandleOutputError };

  procedure PutByte(bite: byte);
    {
    Output a byte to lotus file.
    }
    begin
      if not outputError then
        if outputIndex = BufferLimit then
          begin
            {$I-} blockwrite(outfile,buffer,1); {$I+}
            ioErrorCode := IOresult;
            if ioErrorCode <> 0 then HandleOutPutError(ioErrorCode);
            outputIndex := 1; buffer[1] := bite
          end
        else
          begin
            outputIndex := succ(outputIndex);
            buffer[outputIndex] := bite
          end;
    end { PutByte };

  procedure PutInt(intg: integer);
    {
    Output an integer to Lotus file.
    }
    begin
      intRec.int := intg;
      PutByte(intRec.bite[1]);
      PutByte(intRec.bite[2])
    end { PutInt };

  procedure PutString(row,col: integer; strg: LString);
    {
    Output a string record to Lotus file.
    }
    var i,recLength,strgLength : byte;
    begin
      strgLength := Length(strg);
      recLength := strgLength + 6;
      PutInt(15);
      PutInt(recLength);
      PutByte(255);
      PutInt(col-1);
      PutInt(row-1);
      for i := 1 to strgLength do PutByte(ord(copy(strg,i,1)));
      PutByte(0)
    end { PutString };

  procedure PutNumber(row,col: integer; num: real);
    {
    Output a real number record to Lotus file.
    }
    begin
```

```
PutInt(14);
PutInt(13);
PutByte(255);
PutInt(col-1);
PutInt(row-1);
floatRec.float := num;
{
If not 8087 version, then convert 6 byte turbo real to 8 byte IEEE
format required by Lotus
}
if no8087 then
  inline (
    $A0/floatrec/        {           MOV  AL,FR      }
    $3C/$00/             {           CMP  AL,0       }
    $75/$1C/             {           JNZ  NOTZERO    }
    $B4/$00/             {           MOV  AH,0       }
    $A3/floatrec/        {           MOV  FR,AX      }
    $BF/$02/$00/         {           MOV  DI,2       }
    $89/$85/floatrec/    {           MOV  FR[DI],AX  }
    $BF/$04/$00/         {           MOV  DI,4       }
    $89/$85/floatrec/    {           MOV  FR[DI],AX  }
    $BF/$06/$00/         {           MOV  DI,6       }
    $89/$85/floatrec/    {           MOV  FR[DI],AX  }
    $EB/$6C/             {           JMP  DONE       }
    $B1/$05/             { NOTZERO:  MOV  CL,5       }
    $B4/$00/             {           MOV  AH,0       }
    $05/$7E/$03/         {           AND  AX,894     }
    $D3/$E0/             {           SHL  AX,CL      }
    $D1/$E8/             {           SHR  AX,1       }
    $BE/$05/$00/         {           MOV  SI,5       }
    $8A/$9C/floatrec/    {           MOV  BL,FR[SI]  }
    $B7/$00/             {           MOV  BH,0       }
    $D0/$E3/             {           SHL  BL,1       }
    $73/$02/             {           JMP  POS        }
    $B7/$80/             {           MOV  BH,128     }
    $D0/$EB/             {           SHR  BL,1       }
    $B1/$03/             {           MOV  CL,3       }
    $D2/$EB/             {           SHR  BL,CL      }
    $0B/$C3/             {           OR   AX,BX      }
    $BF/$06/$00/         {           MOV  DI,6       }
    $89/$85/floatrec/    {           MOV  FR[DI],AX  }
    $BE/$04/$00/         {           MOV  SI,4       }
    $BF/$05/$00/         {           MOV  DI,5       }
    $8B/$84/floatrec/    {           MOV  AX,FR[SI]  }
    $D3/$E8/             {           SHR  AX,CL      }
    $88/$85/floatrec/    {           MOV  FR[DI],AL  }
    $4E/                 {           DEC  SI         }
    $4F/                 {           DEC  DI         }
    $8B/$84/floatrec/    {           MOV  AX,FR[SI]  }
    $D3/$E8/             {           SHR  AX,CL      }
    $88/$85/floatrec/    {           MOV  FR[DI]     }
    $4E/                 {           DEC  SI         }
    $4F/                 {           DEC  DI         }
    $8B/$84/floatrec/    {           MOV  AX,FR[DI]  }
    $D3/$E8/             {           SHR  AX,CL      }
    $88/$85/floatrec/    {           MOV  FR[DI],AL  }
    $4E/                 {           DEC  SI         }
    $4F/                 {           DEC  DI         }
    $8B/$84/floatrec/    {           MOV  AX,FR[SI]  }
    $D3/$E8/             {           SHR  AX,CL      }
    $88/$85/floatrec/    {           MOV  FR[DI],AL  }
    $4F/                 {           DEC  DI         }
    $8A/$A5/floatrec/    {           MOV  AH,FR[DI]  }
    $B1/$05/             {           MOV  CL,5       }
    $D2/$E4/             {           SHL  AH,CL      }
    $B4/$00/             {           MOV  AH,0       }
    $88/$26/floatrec);   {           MOV  FR,AH      }
                         { DONE:                     }
with floatRec do
  begin
    PutByte(bite[1]); PutByte(bite[2]); PutByte(bite[3]);
    PutByte(bite[4]); PutByte(bite[5]); PutByte(bite[6]);
```

```
        PutByte(bite[7]); PutByte(bite[8])
      end
    end { PutNumber };

procedure PutHead;
  {
  Put lotus wk1 file header 002064
  }
  begin
    PutInt(0); PutInt(2); PutByte(6); PutByte(4);
  end { PutHead };

procedure PutRange(fromCol,fromRow,toCol,toRow: integer);
  {
  Output range of output tableau to lotus wk1 file
  }
  begin
    {
    Range record is type 6 with length 8
    }
    PutInt(6); PutInt(8);
    PutInt(fromCol-1); PutInt(fromRow-1);
    PutInt(toCol-1);   PutInt(toRow-1)
  end { PutRange };

procedure PutEnd;
  {
  Output end of lotus worksheet file marker
  }
  begin
    {
    End record is type 1 with length 0
    }
    PutInt(1); PutInt(0)
  end { PutEnd };


{==========================================================================
|
|   Procedures to output initial and final tableaus
|
+--------------------------------------------------------------------------}

procedure SetupOutput;
  {
  Ready Lotus file for output.
  }
  var
    outputOK: boolean;

  procedure HandleCriticalError;
    begin
      criticalError := false;
      outputOK := false;
      CleanWindow; WriteLn('Attempt to output to Write-Protected disk ',
        ' OR  disk drive is not ready,');
      Write('  Hit <space> key to continue...'); GetSpace
    end { HandleCriticalError };

  procedure HandleIOError;
    {
    Signal output error had occured.
    }
    begin
      outPutOK := false;
      HandleOutputError(ioErrorCode)
    end { HandleIOError };

  procedure GetFileName;
    {
    Prompt for and input a valid file name from the console.
    }
    begin
      GotoXY(25,19);
      ClrEol;
      CleanWindow;
      Write('Please specify the name of the output file-->',bell);
      Read(outFileName);
      outFileName := FileName(outFileName);
      GotoXY(25,19);
      Write(outFileName)
    end { GetFileName };

  procedure OpenFile;
    {
    Open output Lotus file.
    }
    begin
      CleanWindow; Write('Writing initial tableau to ',outFileName,' ...');
      ASSIGN(outFile,outFileName + '.WK1');
      {$I-} rewrite(outFile); {$I+}
      ioErrorCode := IOresult
    end { OpenFile };

  begin
    repeat
      outputOK := true;
      GetFileName;
      OpenFile;
      if criticalError then
        HandleCriticalError
      else if ioErrorCode <> 0 then
        HandleIOError
    until outPutOK
  end { SetupOutput };

procedure OutputInitialTableau;
  {
  Format and output initial LP tableau.
  }
  var
    i,j: integer;
    row,col: integer;
    index: integer;
    Aij: real;
  begin
    PutHead;
    PutString(1,2,'''I n i t i a l   T a b l e a u');
    for j := 1 to numRealAct+4 do
      begin
        PutString(2,j,'\=');
        PutString(5,j,'\-');
        PutString(numCon+6,j,'\-')
      end;
    PutString(3,2,''''+probName);
    PutString(3,3,''''+objective);
    PutString(3,4,'"'+rhsName);
    PutString(4,2,''''+objName);
    PutString(4,4,'"(RHS)');
    for j := 1 to numRealAct  do
      begin PutString(3,j+4,'"'+actName[j]); PutNumber(4,j+4,cost[j])
      end;
    i := 0;
    while (not outputError) AND (i < numCon) do
      begin
        i := succ(i);
        PutString(i+5,2,''''+actName[numRealAct + i]);
        PutString(i+5,3,'^'+conType[i]);
        PutNumber(i+5,4,R[i]);
        j := 0;
        while (not outputError) and (j < numRealAct) do
          begin
            j := succ(j);
            Aij := A[j]^[i];
```

```
              if abs(Aij) > Tolerence then PutNumber(i+5,j+4,Aij)
            end
        end;
      offset := numCon + 10
    end { OutputInitialTableau };

  procedure OutputFinalTableau;
    {
    Format and output final LP tableau.
    }
    var
      i, j, k, n, index: integer;
      temp: real;

    function StoreSolution: boolean;
      {
      Check if user desires to store solution
      }
      var ch: char;
      begin { StoreSolution }
        repeat
          CleanWindow; Write('Store solution? (Y/N)',bell); Read(kbd,ch)
        until Upcase(ch) in ['Y','N'];
        if Upcase(ch) = 'Y' then
          storeSolution := true
        else
          begin
            storeSolution := false;
            putRange(1,1,numRealAct+4,numCon+10);
          end
      end { StoreSolution };

    procedure Initialize;
      {
      Rearrange tableau items before output.
      }
      var
        i, k, n: integer;
      begin { Initialize }
        k := 0; n := 0;
        for i := 1 to numAct - numGreaterThan do
          if basis[i] = 0 then
            begin
              k := succ(k);
              FinalCol[k] := i
            end
          else
            begin
              n := succ(n);
              FinalRow[n] := i
            end
      end { Initialize };

    procedure PutFrame;
      {
      Put the window dressing of the final tableau.
      }
      var
        j: integer;
      begin { PutFrame }
        CleanWindow; Write('Writing solution to ',outFileName,' ...');
        putString(offset+1,2,'''S o l u t i o n');
        putString(offset+2,2,'''OPTIMAL');
        putString(offset+3,2,'''function Value:');
        putNumber(offset+3,4,objLevel);
        putString(offset+10+numCon,1,'''Z');
        if maximize then
          putString(offset+11+numCon,1,'''Shadow Price')
        else
          putString(offset+11+numCon,1,'''Reduced Cost');
        j := 0;
        while (not outputError) and (j < numRealAct + 4) do
        begin
          j := succ(j);
          putString(offset+4,j,'\=');
          putString(offset+8,j,'\-');
          putString(offset+9+numCon,j,'\-');
          putString(offset+12+numCon,j,'\=')
        end;
      if maximize then
        putString(offset+7,1,'"Returns')
      else
        putString(offset+7,1,'"Cost');
      putString(offset+7,2,'^Name');
      putString(offset+7,3,'^Type');
      putString(offset+7,4,'^Level')
    end { PutFrame };

  begin
    if StoreSolution then
      begin
        Initialize;
        PutFrame;
        k := 0;
        while (not outputError) and (k < numRealAct) do
          begin
            k := succ(k);
            j := FinalCol[k];
            putString(offset+5,k+4,'"'+actName[j]);
            putNumber(offset+6,k+4,cost[j]);
            if j <= numRealAct then
              putString(offset+7,k+4,'"real')
            else
              putString(offset+7,k+4,'"slack');
            putNumber(offset+10+numCon,k+4,z[j]);
            putNumber(offset+11+numCon,k+4,shadow[j])
          end;
        n := 0;
        while (not outputError) and (n < numCon) do
          begin
            n := succ(n);
            i := FinalRow[n];
            putNumber(offset+8+n,1,cost[i]);
            putString(offset+8+n,2,''''+actName[i]);
            if i <= numRealAct then
              putString(offset+8+n,3,'^real')
            else
              putString(offset+8+n,3,'^slack');
            putNumber(offset+8+n,4,R[basis[i]]);
            k := 0;
            while (not outputError) and (k < numRealAct) do
              begin
                k := succ(k);
                j := FinalCol[k];
                {
                Output A[basis[i],FinalCol[k]]
                }
                temp := A[j]^[basis[i]];
                if abs(temp) > Tolerence then putNumber(offset+8+n,k+4,temp)
              end
          end;
        PutRange(1,1,numRealAct+4,offset+numCon+12)
      end
  end { OutputFinalTableau };

procedure CloseOutputFile;
  {
  Close output Lotus file.
  }
  begin
    PutEnd;
    if not outPutError then blockwrite(outFile,buffer,1);
    close(outFile)
  end { CloseOutputFile };
```

```
{=========================================================================
|
|   Solve the tableau using simplex method
|
+------------------------------------------------------------------------}

function SetZ(x: real; tolerence: real): real;
  {
  Return 0 if absolute value of x is less than tolerence.
  }
  begin
    if (abs(x) < tolerence) then
      SetZ := 0
    else
      SetZ := x
  end { SetZ };

function SetP(x: real; Tolerence: real): real;
  {
  Return 0 if x (assumed positive) is less than tolerence.
  }
  begin
    if x < Tolerence then
      SetP := 0
    else
      SetP := x
  end { SetP };

procedure SolveTableau;
  var
    outC:       integer; { Out going column                  }
    inR:        integer; { In coming row                     }
    i:          integer; { Loop index                        }
    j:          integer; { Loop index                        }
    pivot:      real;    { Index of pivoting col             }
    divisor:    real;    { Temp value                        }
    itnum:      integer; { Iteration Number                  }
    quit:       boolean; { User might want to quit           }
    ch:         char;    { Temp char value                   }
    infeasible: boolean; { True if solution still infeasible }

  function ColumnOut: integer;
    {
    Find outgoing column
    }
    var
      i:        integer;
      mostNegI: integer;
      mostPosI: integer;
      mostNegX: real;
      mostPosX: real;
      temp:     real;

    procedure FindMostPositive;
      begin { FindMostPositive }
        mostNegI := 0;
        mostNegX := BigM;
        for i := 1 to numAct - numGreaterThan do
          if basis[i] = 0 then
            begin
              temp := shadow[i];
              if temp < 0.0 then
                if temp <= mostNegX then
                  begin
                    mostNegX := temp;
                    mostNegI := i
                  end
            end;
        ColumnOut := mostNegI
      end { FindMostPositive };

  procedure FindMostNegative;
    begin
      mostPosI := 0; mostPosX := -BigM;
      for i := 1 to numAct - numGreaterThan do
        if basis[i] = 0 then
          begin
            temp := shadow[i];
            if temp > 0.0 then
              if temp >= mostPosX then
                begin
                  mostPosX := temp;
                  mostPosI := i
                end
          end;
      ColumnOut := mostPosI
    end { FindMostNegative };

  begin
    if maximize then
      FindMostPositive
    else
      FindMostNegative
  end { ColumnOut };

function RowIn: integer;
  {
  Find incoming row.
  }
  var
    i:       integer;
    mini:    integer;
    minx:    real;
    temp:    real;
    divisor: real;
  begin
    minx := BigM;
    mini := 0;
    if outC <> 0 then
      for i := 1 to numCon do
        begin
          divisor := A[outC]^[i];
          if divisor > 0.0 then
            begin
              temp := R[i]/divisor;
              if temp <= minx then
                begin
                  mini := i;
                  minx := temp
                end
            end
        end;
    RowIn := mini
  end { RowIn };

procedure SolveInit;
  {
  Initialize before iteration begin.
  }
  begin
    outC       := ColumnOut;      { Outgoing column }
    inR        := RowIn;          { incoming row    }
    itnum      := 0;
    quit       := false;
    infeasible := true
  end { SolveInit };

function Obj: real;
  {
  Compute Objective value
  }
  var
```

```
    i: integer;
    sum: real;
  begin { Obj }
    sum := 0.0;
    for i := 1 to numCon do sum := sum + R[i] * cost[basisNo[i]];
    Obj := sum
  end { Obj };

procedure StartScreen;
  {
  Display information before iteration begins
  }
  begin
    DFrame(false); LowVideo; DFrame1;
    GotoXY(65,5);  Write('infeasible');
    GotoXY(66,04); Write(itnum);
    CleanWindow;   Write('Solving...')
  end { StartScreen };

procedure UpDateScreen;
  {
  Update the screen after each iteration
  }
  begin
    GotoXY(66,4); Write(itnum);
    GotoXY(66,6); Write(actName[outC]); ClrEol;
    GotoXY(66,7); Write(actName[basisNo[inR]]); ClrEol;
    GotoXY(65,8);
    Write(objLevel:14);
    if infeasible then
      if maximize then
        begin
          if objLevel > MinusBigM then infeasible := false
        end
      else if objLevel < BigM
        then infeasible := false;
      if not infeasible then
        begin
          GotoXY(65,05);
          Write('feasible   ')
        end;
    GotoXY(11,22)
  end { UpDateScreen };

procedure CheckOptimal;
  {
  See if solution is optimal and display status on screen.
  }
  begin
    GotoXY(65,5);
    if inR <> 0 then
      WriteLn('OPTIMAL   ')
    else if outC <> 0 then
      WriteLn('UNBOUNDED  ')
    else
      WriteLn('OPTIMAL     ');
    GotoXY(65,8);
    Write(objLevel:14)
  end { CheckOptimal };

procedure SetPivot;
  {
  Set the pivot row after pivot element is found.
  }
  begin
    {
    A[inR,outC] is the pivot, set pivot and make A[inR,outC] one
    }
    pivot         := A[outC]^[inR];
    A[outC]^[inR] := 1.0;
    {
    Vector inR goes out of basis, vector outC comes into basis
    }
    basis[basisNo[inR]] := 0;
    basis[outC]   := inR;
    basisNo[inR]  := outC
   end { SetPivot };

  procedure DivideRow(i: integer; number: real);
   {
   Divide row of LP tableau by a real number.
   }
   var
    j: integer;
   begin { DivideRow }
    R[i] := R[i]/number;
    for j := 1 to numAct do
    {
    only elements not in basis need be divided, basis elements
    are zero here
    }
    if basis[j] = 0 then a[j]^[i] := a[j]^[i]/number
   end { DivideRow };

  procedure MultiplyRow(i: integer; number: real);
   {
   Multiply Row of LP tableau by a real number
   }
   var
    j: integer;
   begin { Multiply Row }
    R[i] := R[i] * number;
    for j := 1 to numAct do
    {
    only elements not in basis need be divided, basis elements
    are zero here
    }
    if basis[j] = 0 then a[j]^[i] := a[j]^[i] * number
  end { Multiply };

  procedure RowEliminate(i, inR: integer; number: real);
   {
   Performs row elimination on row i of tableau.
   }
   var
    j:     integer;
    temp:  real;
    temp2: real;
   begin
    if number <> 0.0 then
     begin
      R[i] := SetP(R[i] - (number * R[inR]),Tolerence);
      for j:=1 to numAct do
       if basis[j] = 0 then
        A[j]^[i] := SetZ(A[j]^[i]-number * A[j]^[inR],Tolerence)
     end
   end { RowEliminate };

  procedure VerifyQuit;
   {
   See if user wants to abort solution after pausing it.
   }
   begin
    repeat
     CleanWindow; ClrEol; Write('Abort Solution? (Y/N)',bell);
     Read(kbd,ch)
    until UpCase(ch) in ['Y','N'];
    if UpCase(ch) = 'N' then
     begin
      CleanWindow;
      Write('Solving...');
      quit := false
     end
   end { VerifyQuit };

  begin
    SolveInit;
    StartScreen;
    {
    Iterate not optimal, infeasible, unbounded, or user abort
    }
    while (inR <> 0) and (outC <> 0) and (not quit) do
      begin { iteration }
        itnum := succ(itnum);
        quit := Keypressed; { Pause if user hit any key }
        UpDateScreen;
        SetPivot;
        quit := Keypressed;
        {
        if pivot is not one divide the pivot row by the pivot element
        }
        if pivot <> 1.0 then  DivideRow(inR,pivot);
        {
        Make the pivot column except the pivot element zero
        }
        for i := 1 to numCon do
          if i <> inR then
            begin
              RowEliminate(i,inR,A[outC]^[i]);
              A[outC]^[i] := 0.0
            end;
        ComputeShadowPrices;
        objLevel := Obj;
        quit     := keyPressed;
        outC     := ColumnOut;
        inR      := RowIn;
        if quit then VerifyQuit
      end;
      if not quit then CheckOptimal
  end { SolveTableau };


{==========================================================================
!
!   Main Program
!
+-------------------------------------------------------------------------}

begin
  SetInterruptVector;
  repeat
    Mark(heapPtr);
    Initialize;
    repeat
      SetupInput;
      ReadTableau;
    until not inputError;
    SetupLpTableau;
    SetupOutput;
    OutputInitialTableau;
    if not outputError then
      begin
        SolveTableau;
        OutputFinalTableau
      end;
    CloseOutputFile;
    Release(heapPtr);
    repeat
      CleanWindow; Write('Solve another Problem? (Y/N)',bell);
      Read(kbd,another)
    until UpCase(another) in ['Y','N'];
  until UpCase(another) = 'N';
  CleanWindow;
  Write('Musah86 is Developed by Elton Li');
  Write('Dept of Agricultural Economics');
  Write('Oklahoma State University');
  Write('Stillwater, Oklahoma 74078');
  Write('U.S.A.');
  Write('Thanks for using this program.  Bye Bye!');
  Delay(1500); LowVideo; GotoXY(1,1); ClrScr
end.
```

APPENDIX B

EXAMPLE OF TUTORIAL MATERIAL USED IN TRAINING

AGRICULTURAL PROJECT INVESTMENT ANALYSIS WITH ELECTRONIC SPREADSHEETS: A CASE STUDY

by

Elton Li, Suki Kang and Dean F. Schreiner

Scope of Case Study

In many public decisions, especially those involving a resource allocation, decisions must be made on whether or not a given undertaking is worth the cost. The most common approach is to express the benefits and costs associated with each alternative in dollars as a function of time. The future benefits and costs are discounted at some appropriate rate, and then the alternatives are compared on the basis of the present value of the net benefits, or on the basis of internal rate of return.

Figure 1 displays a format commonly used to compute various discounted measures of project worth. The example is adapted from chapter 10 of Gittinger's "Economic Analysis of Agricultural Projects". With an economic life of 30 years and at a 12 percent discount rate, the project is shown to yield a net present worth of 5.21 Indonesian rupiahs (Rp), a benefit-cost ratio of 1.50 and a net benefit-investment ratio of 1.98. The internal rate of return of the project is 21 percent.

Many fundamental difficulties exist in performing a cost-benefit analysis. Among these are the ambiguity of projecting and expressing in dollars terms both the cost and benefit at each point in time, the distributional impacts of the project, and choosing a suitable discount rate. These issues are beyond the scope of this case study (see Gittinger, Little, Brown). Suffice it to say that guesses of circumstances must often be made. Reliability of the analysis requires reworking it with different assumptions to see what happens under other likely circumstances.

This case study demonstrates the use of a microcomputer spreadsheet program for performing computations that commonly arise in the cost-benefit analysis of public decisions, including agricultural projects. Emphasis is placed on how sensitivity analysis is facilitated by use of a spreadsheet program.

The Spreadsheet Template (First Try)

Lotus 1-2-3 is used for this case study although most other electronic spreadsheets could be used. Implementation of figure 1 on an electronic spreadsheet is relatively straightforward. In figure 1, column A from cell A11 to A40 is the time period and can be filled in by the Data Fill command. The corresponding cells in column B and C represent respectively the undiscounted incremental cost and benefit of the project for the year; these are part of the input values required for this analysis. Column D, net benefit, is computed as the difference between column B and C. This relationship, for year 1, say, is expressed by typing in the formula + C11 - B11 in cell D11. Similar formulas are required for the other 29 cells in column D. These can be inserted quickly by using the Copy command to replicate the formula in cell D11 to cells D12 to D40.

The discounted incremental cost in column E is computed by the formula:

incremental cost / (1 + discount factor) ^ year

where ^ denotes exponentiation. In our template, the discount factor is stored in cell D3, year is stored in column A, and the incremental cost is stored in column B. Thus the appropriate formula for year 1 is:

+B11 / (1 + D3) ^ A11

and the formula for year 2 is:

+B12 / (1 + D4) ^ A12

Note that, if the formula in cell E11 is copied by the copy command to cell E12 the resulting formula would be

+D12 / (1 + D4) ^ A12

which is incorrect since cell D4 does not contain the discount rate. By default, the copy command uses relative addresses. It adjusts the cell references in the formula to be consistent with the location difference between the original cell and the destination cell. This adjustment is generally desirable: in column D, the formula + C11 - B11 is appropriately adjusted to + C12 - B12 for year 2, +C13 - B13 for year 3 and so on. However, for the discounted rate (cell D3) this automatic adjustment is undesirable, since it would cause cell D4 to be used as the discount rate in cell E12 after the copy operation. Different electronic spreadsheets have different means to "fix" absolute cell references while copying or replicating. In 1-2-3, absolute cell references are indicated with dollar signs. Thus, the formula in cell E11 should be entered as:

+B11 / (1 + D3) ^ A11

The $ sign in front of D in D3 prevents the column coordinate from being adjusted and the $ sign in front of 3 prevents the row coordinate from being adjusted. Thus a dollar sign in front of both "D" and "3" prevents both the row and column coordinate from being adjusted when copying occurs. Operation of the formulas are not affected by the $ signs. A dollar sign is also placed before "A" in A11 to prevent "A" from being adjusted to "B" when the formula in cell E11 is copied to column F. To continue completion of figure 2, the formula in cell E11 is copied to the range E1. F40. Folumn G is computed as the difference between column F and column E. The table is then formatted to display 2 decimal places by the Range Format command. Row 42 represents the sum of the various quantities. In column B, the sum of incremental cost can be computed by using the @SUM function in 1-2-3. The appropriate formula for cell B42 is:

@SUM (B11 . .B40)

which says the cell B42 is the sum of the entries from B11 to B40. Once this is entered, the other totals can quickly be inserted by the Copy command. No suppression of the automatic adjustment in the Copy command is required here.

Discounted Measures of Project Worth

Various common discounted measures are discussed by Gittinger. The formulation of these measures are:

Net present worth:

$$\sum_{t=i}^{t=n} \frac{B_t - C_t}{(1+i)^t}$$

Internal rate of return: the discount rate such that

$$\sum_{t=i}^{t=n} \frac{B_t - C_t}{(1+i)^t} = 0$$

Benefit-cost ratio:

$$\sum_{t=i}^{t=n} \frac{B_t}{(1+i)^t} \Big/ \sum_{t=i}^{t=n} \frac{C_t}{(1+i)^t}$$

Net benefit-investment (N/K) ratio:

$$\sum_{t=i}^{t=n} \frac{N_t}{(1+i)^t} \;/\; \sum_{t=i}^{t=n} \frac{K_t}{(1+i)^t}$$

where:

B_t = benefit in each year

C_t = cost in each year

N_t = incremental net benefit in each year after stream has turned positive

K_t = incremental net benefit in initial years when stream is negative

t = 1, 2, 3, ..., n

n = number of years

i = interest (discount) rate

In figure 1, net present worth in E44 is the sum of the column of discounted incremental net benefits (column G). Cell E44 thus is defined as an absolute cell reference to G42, which contains the column total for column G.

The benefit cost ratio in cell E46 is the quotion of the sum of the discounted incremental benefits and the sum of the discounted incremental costs. The net investment ratio in cell E47 is computed by dividing the sum of the discounted incremental net benefit after the stream has turned positive by the sum of the discounted incremental net benefit in initial years when stream is negative. To compute this, column I is defined to be equal to the corresponding row element of column G if the discounted incremental net benefit is negative, zero otherwise. This is implemented by the @IF statement. In cell I11, fcr example, the appropriate formula is:

```
@IF (G11 < 0, G11, 0)
```

which says: if cell G11 is negative, insert the value of G11 into I11, otherwise insert 0 there. With the help of this "working" column, cell E47 is defined as:

(+ G42 - @SUM(I11. I40)) / (-@SUM (I11. I40)

The internal rate of return is the discount rate which results in a zero discounted net present worth. This is typically computed by some systematic search algorithm such as Newton's method which is cumbersome to perform by hand. Several short-cut methods (see Gittinger, Brown) are more appropriate for hand calculation at the sacrifice of precision. Most electronic spreadsheets have an internal rate of return function. Referring to figure 2, the cell E45 is defined as @IRR (.16, D11 . D40). D11 to D40 is the cash flow series from which the IRR is computed. The .16 is an initial guess, required by 1-2-3, used as the starting point of the search for the correct IRR.

Effect of Different Discount Rates

Figure 2 contains a table of discount project worth measures at various discount rates obtained by changing the discount rate cell of the template just described. The attractiveness of the project diminishes as a larger discount rate is used. A discount rate equal to the rate of return drives the net present worth to zero and both the benefit-cost ratio and the net benefit-investment ratio to one. Figure 3 is a graph of the B-C ratio and the N-K ratio of the project under various discount rates.

Sensitivity Analysis

Reworking an analysis to see what happens under changed circumstances is called sensitivity analysis. The above template can be modified to perform sensitivity analysis involving cost overrun, benefit shortfall, and benefit delay.

The table shown in figure 4 is derived from figure 1. The information in the "most probable outcome" section is identical to figure 1. The alternative outcome section allows for a percentage cost overrun, a percentage of benefit shortfall, and a delay of benefit of up to five years from the most probable scenario.

Referring to figure 4, the percentage cost overrun, percentage benefit shortfall, and benefit delay are input parameters to the analysis and are recorded in cells D4, D5 and D6 respectively. Column H, the incremental cost of the alternative outcome section, is computed as a function of the corresponding incremental cost of the most probable outcome section and the percent of cost overrun recorded in cell D4. The appropriate formula for cell H15 is therefore:

+ B15 • (1 + D4)

The incremental benefit column is more complicated in order to accomodate both a benefit shortfall and a benefit delay. In cell I15, for example, the appropriate formula is:

(1 - D5) • @CHOOSE (D6, C15, C14, C13, C12, C11, C10)

The @CHOOSE function chooses among cells C15, C14, C13, C12, C11, and C10 according to the contents of cell D6. If cell D6 is 0, i.e. no delay of benefit, then the most probable incremental benefit of the same year (cell C15) is chosen. If cell D6 is 1, which means the benefit is delayed for one year, the @CHOOSE statement would cause cell C14, which is the most probable incremental benefit of the previous year, to be chosen. After the most probable incremental benefit is chosen, it is then adjusted by the percentage of benefit shortfall to represent the alternative incremental benefit for the year.

The formula in cell I15 can be copied to later years. For earlier years, say year 1, the following formula is used instead:

(1 - D5) • @CHOOSE (D6, C15, 0, 0, 0, 0, 0)

The insertion of zeros instead of cell references as parameters of the choose statements is to prevent the selection of cells beyond the first year which is meaningless.

The remaining entries of figure 4 are similar to that of figure 1. In figure 4, with a 10 percent cost overrun, a 10 percent benefit shortfall, and a benefit delay of 1 year, the net present worth of the project dropped from 5.21 to 2.51 at 12 percent discount rate, the internal rate of return decreased to 13 percent from 21 percent, and the benefit-cost ratio and net benefit-investment ratio at 12 percent discount rate to 1.22 and 1.34 from 1.50 and 1.98, respectively. Other alternative outcomes can be obtained by changing cells D3, D4, D5, and D6.

Figure 5 shows the effect of benefit delays on the various discounted measures of project worth. Figure 6 is a chart showing the effect of benefit delay on the internal rate of return of the project.

```
             !    A         B         C         D         E         F         G         H         I
-------------!----------------------------------------------------------------------------------------
  1    !   JATILUHUR IRRIGATION PROJECT, INDONESIA
  2    !
  3    !   DISCOUNTED FACTOR:                 0.12
  4    !   ==============================================================================
  5    !                                          ----- Discounted -----
  6    !                               INCRE-                        INCRE-
  7    !            INCRE-    INCRE-   MENTAL    INCRE-    INCRE-    MENTAL
  8    !            MENTAL    MENTAL   NET       MENTAL    MENTAL    NET
  9    !   YEAR     COST      BENEFIT  BENEFIT   COST      BENEFIT   BENEFIT
 10    !   ------------------------------------------------------------------------------
 11    !          1      0.50      0.00     -0.50      0.45      0.00     -0.45              -0.45
 12    !          2      2.10      0.40     -1.70      1.67      0.32     -1.36              -1.36
 13    !          3      3.70      0.80     -2.90      2.63      0.57     -2.06              -2.06
 14    !          4      3.70      1.40     -2.30      2.35      0.89     -1.46              -1.46
 15    !          5      2.00      2.10      0.10      1.13      1.19      0.06                  0
 16    !          6      0.50      2.50      2.00      0.25      1.27      1.01                  0
 17    !          7      0.50      2.90      2.40      0.23      1.31      1.09                  0
 18    !          8      0.50      2.90      2.40      0.20      1.17      0.97                  0
 19    !          9      0.50      2.90      2.40      0.18      1.05      0.87                  0
 20    !         10      0.50      2.90      2.40      0.16      0.93      0.77                  0
 21    !         11      0.50      2.90      2.40      0.14      0.83      0.69                  0
 22    !         12      0.50      2.90      2.40      0.13      0.74      0.62                  0
 23    !         13      0.50      2.90      2.40      0.11      0.66      0.55                  0
 24    !         14      0.50      2.90      2.40      0.10      0.59      0.49                  0
 25    !         15      0.50      2.90      2.40      0.09      0.53      0.44                  0
 26    !         16      0.50      2.90      2.40      0.08      0.47      0.39                  0
 27    !         17      0.50      2.90      2.40      0.07      0.42      0.35                  0
 28    !         18      0.50      2.90      2.40      0.07      0.38      0.31                  0
 29    !         19      0.50      2.90      2.40      0.06      0.34      0.28                  0
 30    !         20      0.50      2.90      2.40      0.05      0.30      0.25                  0
 31    !         21      0.50      2.90      2.40      0.05      0.27      0.22                  0
 32    !         22      0.50      2.90      2.40      0.04      0.24      0.20                  0
 33    !         23      0.50      2.90      2.40      0.04      0.21      0.18                  0
 34    !         24      0.50      2.90      2.40      0.03      0.19      0.16                  0
 35    !         25      0.50      2.90      2.40      0.03      0.17      0.14                  0
 36    !         26      0.50      2.90      2.40      0.03      0.15      0.13                  0
 37    !         27      0.50      2.90      2.40      0.02      0.14      0.11                  0
 38    !         28      0.50      2.90      2.40      0.02      0.12      0.10                  0
 39    !         29      0.50      2.90      2.40      0.02      0.11      0.09                  0
 40    !         30      0.50      2.90      2.40      0.02      0.10      0.08                  0
 41    !   ------------------------------------------------------------------------------
 42    !   TOTAL        24.50     76.80     52.30     10.47     15.67      5.21              -5.33
 43    !
 44    !   NET PRESENT WORTH                          5.21
 45    !   INTERNAL RATE OF RETURN                    0.21
 46    !   BENEFIT-COST RATIO                         1.50
 47    !   NET BENEFIT INVESTMENT RATIO               1.98
 48    !   ==============================================================================
 49    !
```

FIGURE 1

DISCOUNT RATE	NPW	B/C RATIO	N/K RATIO
0.07	13.67	2.00	3.25
0.08	11.34	1.88	2.92
0.09	9.39	1.77	2.63
0.10	7.75	1.67	2.38
0.11	6.37	1.58	2.17
0.12	5.21	1.50	1.98
0.13	4.22	1.42	1.81
0.14	3.37	1.35	1.67
0.15	2.65	1.29	1.54
0.16	2.03	1.23	1.42
0.17	1.49	1.18	1.32
0.18	1.03	1.13	1.23
0.19	0.64	1.08	1.14
0.20	0.29	1.04	1.07
0.21	0.00	1.00	1.00
0.22	-0.26	0.96	0.94
0.23	-0.49	0.93	0.88
0.24	-0.68	0.90	0.83

FIGURE 2

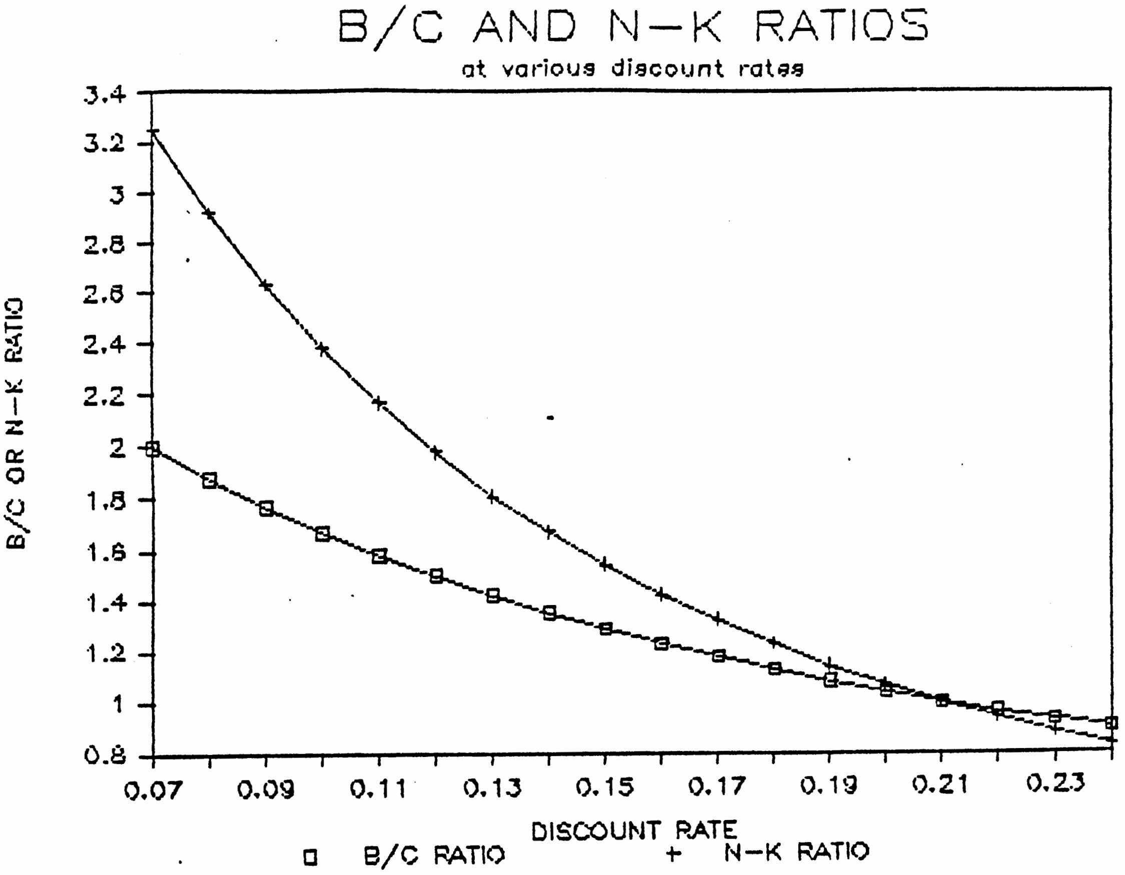

B/C AND N-K RATIOS
at various discount rates
B/C OR N-K RATIO
3.4
3.2
3
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.07
0.09
0.11
0.13
0.15
0.17
0.19
0.21
0.23
DISCOUNT RATE
□ B/C RATIO
+ N-K RATIO

FIGURE 3

JATILUHUR IRRIGATION PROJECT, INDONESIA

DISCOUNTED FACTOR:	0.12
% COST OVERRUN	0.1
% BENEFIT SHORTFALL	0.1
BENEFIT DELAY (0-5)	1

	----MOST PROBABLE OUTCOME----						----ALTERNATIVE OUTCOME----							
				----- Discounted -----						----- Discounted -----				
YEAR	INCREMENTAL COST	INCREMENTAL BENEFIT	INCREMENTAL NET BENEFIT-	INCREMENTAL COST	INCREMENTAL BENEFIT	INCREMENTAL NET BENEFIT	INCREMENTAL COST	INCREMENTAL BENEFIT	INCREMENTAL NET BENEFIT	INCREMENTAL COST	INCRE- . MENTAL BENEFIT	INCREMENTAL NET BENEFIT		
1	0.50	0.00	-0.50	0.45	0.00	-0.45	0.55	0.00	-0.55	0.49	0.00	-0.49	-0.45	-0.491
2	2.10	0.40	-1.70	1.67	0.32	-1.36	2.31	0.00	-2.31	1.84	0.00	-1.84	-1.36	-1.841
3	3.70	0.80	-2.90	2.63	0.57	-2.06	4.07	0.36	-3.71	2.90	0.29	-2.61	-2.06	-2.609
4	3.70	1.40	-2.30	2.35	0.89	-1.46	4.07	0.72	-3.35	2.59	0.51	-2.07	-1.46	-2.074
5	2.00	2.10	0.10	1.13	1.19	0.06	2.20	1.26	-0.94	1.25	0.80	-0.45	0	-0.447
6	0.50	2.50	2.00	0.25	1.27	1.01	0.55	1.89	1.34	0.28	1.07	0.79	0	
7	0.50	2.90	2.40	0.23	1.31	1.09	0.55	2.25	1.70	0.25	1.14	0.89	0	
8	0.50	2.90	2.40	0.20	1.17	0.97	0.55	2.61	2.06	0.22	1.18	0.96	0	
9	0.50	2.90	2.40	0.18	1.05	0.87	0.55	2.61	2.06	0.20	1.05	0.86	0	
10	0.50	2.90	2.40	0.16	0.93	0.77	0.55	2.61	2.06	0.18	0.94	0.76	0	
11	0.50	2.90	2.40	0.14	0.83	0.69	0.55	2.61	2.06	0.16	0.84	0.68	0	
12	0.50	2.90	2.40	0.13	0.74	0.62	0.55	2.61	2.06	0.14	0.75	0.61	0	
13	0.50	2.90	2.40	0.11	0.66	0.55	0.55	2.61	2.06	0.13	0.67	0.54	0	
14	0.50	2.90	2.40	0.10	0.59	0.49	0.55	2.61	2.06	0.11	0.60	0.49	0	
15	0.50	2.90	2.40	0.09	0.53	0.44	0.55	2.61	2.06	0.10	0.53	0.43	0	
16	0.50	2.90	2.40	0.08	0.47	0.39	0.55	2.61	2.06	0.09	0.48	0.39	0	
17	0.50	2.90	2.40	0.07	0.42	0.35	0.55	2.61	2.06	0.08	0.43	0.35	0	
18	0.50	2.90	2.40	0.07	0.38	0.31	0.55	2.61	2.06	0.07	0.38	0.31	0	
19	0.50	2.90	2.40	0.06	0.34	0.28	0.55	2.61	2.06	0.06	0.34	0.28	0	
20	0.50	2.90	2.40	0.05	0.30	0.25	0.55	2.61	2.06	0.06	0.30	0.25	0	
21	0.50	2.90	2.40	0.05	0.27	0.22	0.55	2.61	2.06	0.05	0.27	0.22	0	
22	0.50	2.90	2.40	0.04	0.24	0.20	0.55	2.61	2.06	0.05	0.24	0.20	0	
23	0.50	2.90	2.40	0.04	0.21	0.18	0.55	2.61	2.06	0.04	0.22	0.18	0	
24	0.50	2.90	2.40	0.03	0.19	0.16	0.55	2.61	2.06	0.04	0.19	0.16	0	
25	0.50	2.90	2.40	0.03	0.17	0.14	0.55	2.61	2.06	0.03	0.17	0.14	0	
26	0.50	2.90	2.40	0.03	0.15	0.13	0.55	2.61	2.06	0.03	0.15	0.12	0	
27	0.50	2.90	2.40	0.02	0.14	0.11	0.55	2.61	2.06	0.03	0.14	0.11	0	
28	0.50	2.90	2.40	0.02	0.12	0.10	0.55	2.61	2.06	0.02	0.12	0.10	0	
29	0.50	2.90	2.40	0.02	0.11	0.09	0.55	2.61	2.06	0.02	0.11	0.09	0	
30	0.50	2.90	2.40	0.02	0.10	0.08	0.55	2.61	2.06	0.02	0.10	0.08	0	
TOTAL	24.50	76.80	52.30	10.47	15.67	5.21	26.95	66.51	39.56	11.51	14.02	2.51	-5.33	-7.

NET PRESENT WORTH	5.21	NET PRESENT WORTH	2.51
INTERNAL RATE OF RETURN	0.21	INTERNAL RATE OF RETURN	0.13
BENEFIT-COST RATIO	1.50	BENEFIT-COST RATIO	1.22
NET BENEFIT INVESTMENT RATIO	1.98	NET BENEFIT INVESTMENT RATIO	1.34

FIGURE 4

DELAY	NPW (12 %)	IRR	B/C (12 %)	N-K (12 %)
0	5.21	0.21	1.50	1.98
1	3.44	0.17	1.33	1.52
2	1.87	0.14	1.18	1.25
3	0.46	0.13	1.04	1.06
4	-0.80	0.11	0.92	0.90
5	-1.92	0.10	0.82	0.78

FIGURE 5

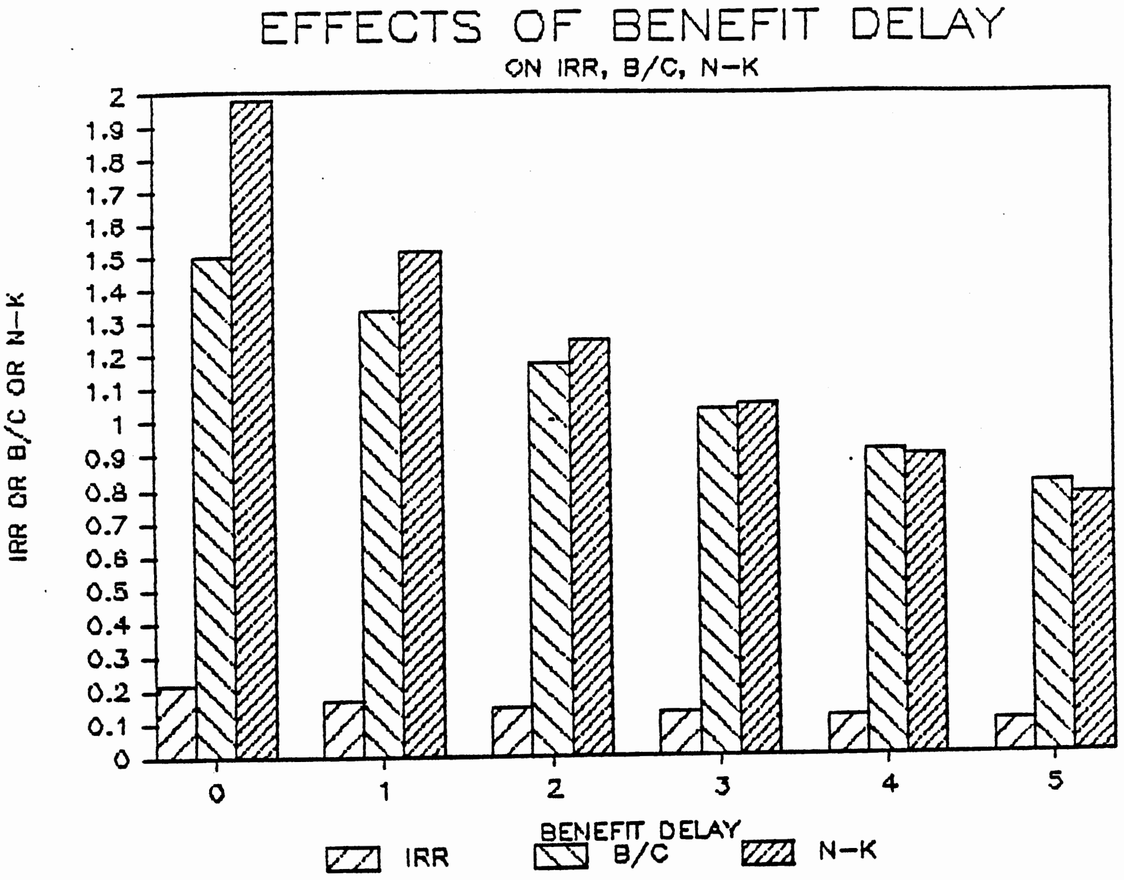

EFFECTS OF BENEFIT DELAY
ON IRR, B/C, N-K
IRR OR B/C OR N-K
2
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1
2
3
4
5
BENEFIT DELAY
IRR
B/C
N-K

FIGURE 6

VITA

Elton Li

Candidate for the Degree of

Doctor of Philosophy

Thesis: MICROCOMPUTER APPLICATIONS TO AGRICULTURAL POLICY ANALYSIS IN DEVELOPING COUNTRIES

Major Field: Agricultural Economics

Biographical:

Personal Data: Born in Hong Kong, December 15, 1954, the son of Mr. Lee King Yeung and Madam Tsui Yuk Yu.

Education: Attended Northeastern Oklahoma A&M College, 1973-1974; attended University of Houston, summer 1974; received Bachelor of Science Degree in Mathematics from Oklahoma State University in July, 1976; completed requirements for the Doctor of Philosophy degree at Oklahoma State University in May, 1987.

Professional Experience: Mathematics Tutor, Department of Mathematics, Northeastern Oklahoma A&M College, 1973 to 1974; Computer Programmer, Department of Agricultural Economics, Oklahoma State University, 1976 to 1980; Graduate Assistant, Department of Agricultural Economics, Oklahoma State University, 1980 to present.