Optimization of Stormwater Management Practices and Processes

Mark Michael Calabrese
University of Central Florida
OPTIMIZATION OF STORMWATER MANAGEMENT PRACTICES AND PROCESSES

BY

MARK MICHAEL CALABRESE
B.S.E., Florida Technological University, 1973

THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering in the Graduate Studies Program of the College of Engineering of the University of Central Florida at Orlando, Florida

Spring Quarter
1979
OPTIMIZATION OF STORMWATER MANAGEMENT PRACTICES AND PROCESSES

BY

MARK MICHAEL CALABRESE

ABSTRACT

In recent years, stormwater has been found to be a major source of pollution to receiving waters. Major research efforts have been directed in this area, primarily as a result of the Federal Water Pollution Control Act and Public law 92-500, the 1972 Amendments to the act. Yet, a need remains for more data in the field of stormwater management. Such needs include cost-performance data and planning methodologies to optimally select best management practices (BMP's).

The research culminating in this report addresses these needs. A computer program, "MANAGE", has been written to generate cost/efficiency curves, and uses these curves to optimally select a combination of management practices. The program was written in FORTRAN language and was run on the IBM 360/370 computer system. It can analyze up to 3 management practices per subwatershed and up to 20 subwatersheds in a given watershed. The optimization routine of the program utilizes a piece-wise linear approximation method in its analysis.
ACKNOWLEDGEMENTS

The author wishes to extend his sincere appreciation to Dr. Martin Paul Wanielista for his direction, help, and friendship throughout this study. Special thanks is also extended to Dr. Chris Bauer, Dr. Michael Muiga, and especially Mr. Thomas Peeples, who provided valuable guidance and help in this research effort.

Sincere appreciation is expressed to my parents for their help, both financially and spiritually, and for their inspiration throughout my graduate program and my life.

Gratitude is expressed to Miss Susie Shigley for her faith and to Mrs. Kathy Caldes for her assistance in typing.

My thanks to the friends and faculty of the University of Central Florida who made my five years seem so short, yet so fulfilled.
# TABLE OF CONTENTS

ACKNOWLEDGMENTS ................................................... iii

Chapter

I. INTRODUCTION .................................................. 1

Objectives ............................................................ 3

II. MATHEMATICAL FORMULATION .................................. 5

  Linear Programming General Form of The Lake Eola
  Stormwater Management Problem .................................. 7
  One Subwatershed Linear Programming Formulation .......... 8
    Variables .......................................................... 8
    Constraints ....................................................... 10
    Nodal Equations ................................................ 10
    Generalized Form .............................................. 10
  Two Subwatershed Linear Programming Formulation ........ 11
    Variables .......................................................... 11
    Constraints ....................................................... 11
    Nodal Equations ................................................ 11
    Generalized Form .............................................. 14
    Generalized Formulation - All Treatment .................. 14

III. LAKE EOLA REVIEW .............................................. 16

IV. STORMWATER MANAGEMENT PRACTICES .......................... 18

  Diversion/Percolation ........................................... 21
  Detention ........................................................... 22
  Swales .............................................................. 24
  Dutch Drains ..................................................... 25
  Porous Pavement - Asphalt ..................................... 26
  Street Sweeping .................................................. 26
  Catchbasin Cleaning ............................................ 32
    Manual Cleaning ................................................ 33
    Bucket Cleaning ................................................ 35
    Eductor Cleaning .............................................. 35
    Vacuum Cleaning ............................................... 35
    Cleaning Frequencies .......................................... 36
    Storage Basin ................................................... 36
    Fabric Bags ..................................................... 36

iv
Storage Treatment ........................................ 38
Storage ................................................... 39
Roof-top ................................................... 39
Microstraining .......................................... 40
Swirl Concentrators .................................... 41

V. COMPUTER PROGRAM "MANAGE" DISCUSSION ........... 42

Main Selector Routine .................................... 42
Main Selector Routine (Variable Descriptions) .......... 45
Catchbasin Cleaning Subroutine - Subroutine
CATCH .................................................... 46
Street Sweeping Subroutine - Subroutine SWEEP ....... 48
Diversion/Percolation Subroutine - Subroutine
BASIN ................................................... 50
Subroutine PLOT .......................................... 58
Subroutine SLOPE ....................................... 58
Subroutine TABLO ....................................... 58
Subroutine LPGOGO ...................................... 59
Lake Eola Linear Programming Formulation ............. 59
LPGOGO Output Interpretation .......................... 66
Subroutine TABLOX ..................................... 67
Subroutine LPGOGX ...................................... 67

VI. LAKE EOLA DATA FOR COMPUTER PROGRAM .......... 68

Loading Rates ........................................... 68
Soil Type ............................................... 68

VII. EXAMPLE PROBLEM ....................................... 70

Data for Subwatershed #1 ................................ 70
Catchbasin .............................................. 70
Street Sweeping ........................................ 70
Diversion/Percolation ................................... 71
Sizing and Costing of BMP's ............................ 72

VIII. CONCLUSIONS ........................................... 84

APPENDIX A - COMPUTER PRINTOUT ..................... 88

LIST OF REFERENCES ..................................... 188
# LIST OF TABLES

1. Stormwater Management Comparative Data/Impervious Acre  
   (Land Cost Not Included) (Central Florida Data)........ 28
2. Efficiency Data for Brush Type Street Sweepers .......... 30
3. Broom Sweeping Efficiencies Low Versus Average Ability .. 31
4. Frequency of Catchbasin Cleaning in Various North  
   American Cities.............................................. 37
5A. Pond Volume for Type D Soils .............................. 52
5B. Pond Volume for Type A Soils .............................. 53
6. Efficiencies of Diversion/Percolation (First Flush  
   Phenomena)..................................................... 54
7. Efficiencies of Diversion/Percolation (No First Flush  
   Phenomena)..................................................... 55
8. Land Use Data for Lake Eola.................................. 69
LIST OF FIGURES

1. Convex Cost/Efficiency Curve & Piecewise Linear Approximation
   ........................................................................................................ 6
2. Sketch of One Subwatershed ................................................................. 9
3. Nodal Diagram of One Subwatershed .................................................... 9
4. Sketch of Dual Subwatersheds ............................................................... 12
5. Nodal Diagram of Dual Subwatersheds ............................................... 12
6. Management Methods ........................................................................... 19
7. Diversion/Percolation Structure ............................................................ 23
8. Detention Basin ................................................................................... 27
9. Swale Cross Sections ........................................................................... 27
10. Dutch Drain and Porous Asphalt Devices ............................................ 27
11. Representative Catchbasin Design in United States and Canada ................. 34
12. Computer Program "MANAGE" Flow Chart ......................................... 43
13. Graph #1 - Catchbasin Cleaning (nitrogen graph) ............................ 61
14. Graph #2 - Street Sweeping (nitrogen graph) ...................................... 62
15. Graph #3 - Diversion/Percolation (nitrogen graph) ............................ 63
16. Catchbasin Cleaning Data Output ....................................................... 73
17. Nitrogen Convex Cost/Efficiency Curve ............................................. 74
18. Initial Tableau for Subwatershed #1 .................................................... 75
19. Linear Programming (LPGOGO) Solution Output for Subwatershed #1 .... 76
20. Street Sweeping Data Output ............................................................... 77
21. Diversion/Percolation Data Output ..................................................... 78
22. Subwatershed Summary Data Output .................................................... 79
23. Initial Tableau for All Subwatersheds (Total Watershed) .................... 80
24. Linear Programming (LPGOGX) Solution Output for Watershed .......... 81
25. Watershed Summary Data Output ....................................................... 82
CHAPTER I

INTRODUCTION

Public law 92-500, the 1972 Amendments to the Federal Water Pollution Control Act was enacted to eliminate the discharge of pollutants into the navigable waters of the United States by 1985 and, in the interim, obtain water quality which provides for the protection and propagation of fish, shellfish and wildlife, and provides for water related recreation by July 1, 1983 (Searcy, n.d.). The act provides certain tools to accomplish these goals, including Section 208 which allows for Areawide Waste Treatment Management Plans along the following lines (Heaney 1977):

1. Identify the problems in meeting the 1983 goals of the act
2. Identify all constraints and priorities pertaining to the 208 planning area
3. Identify all possible solutions to the problem
4. Develop alternative plans to meet the statutory requirements
5. Analyze the alternative plans for technologic and economic feasibility
6. Select an areawide plan
7. Seek approval for the plan
8. Periodically update the plan

The selection of a specific plan should be based on cost effectiveness, which implies engineering feasibility, and public acceptance.
In recent years, stormwater has been found to be a major source of pollution to receiving waters. Major research efforts have been directed in this area, primarily as a result of the water pollution acts and amendments. Yet, there remains a need for more data in this field of stormwater management, as stated by Mr. Richard Field (1978) of the U.S. Environmental Protection Agency. Mr. Field stated: "There is a crying need for (1) cost-performance data for full-scale nonstructural urban stormwater pollution control measures or best management practices (BMPs), and (2) for the appropriate planning methodology to optimally integrate nonstructural with structural control on an area-wide basis..." Nonstructural control is one requiring source control of runoff. It minimizes or prevents land pollution. If the pollutants can be reduced before transport by stormwaters to receiving water bodies, then the stormwater will be managed to a specified degree. Structural control requires treatment onsite or at the point of discharge without a pollution permit. It requires complete or partial physical modifications of the source, transport or discharge stormwater system. This research was developed to aid in providing answers for the needs outlined by Mr. Field.

It is felt that a great deal of knowledge and useful data can be obtained by implementing a stormwater management plan in an area of Central Florida, one abundant with lakes and waterways. The area chosen as the study site for this project was Lake Eola, located in downtown Orlando, Florida. At the present time, the tro-
Phic state of Lake Eola is eutrophic, as determined by planning models developed by Vollenweider, Dillon, and Larsen-Mercier (Wanielista 1979). Nonpoint sources have been identified as potential major contributors to lake degradation in the State of Florida (Wanielista 1977). Pollutants include suspended solids, $\text{BOD}_5$, nitrogen, and phosphorus. Lake quality will not likely improve until these sources of pollution are managed.

Past efforts in the Orange-Seminole-Osceola area have defined the efficiencies and costs for diversion/percolation basins, poorly designed detention basins, swales, underdrains, and vacuum sweeping nonpoint source management methods (Wanielista 1977). In the highly impervious urban areas, the cost of land is expensive, and land intensive activities (detention and retention basins) are sometimes not aesthetically pleasing. Thus, street sweeping, diversion with retention underground, and catchbasin cleaning appears probable for urban areas. Diversion of the first flush, a portion of the first runoff waters for treatment, is practiced with percolation into the ground. Dutch drains, roof-top storage, coagulation, filtration and concentrators are other management methods under investigation.

Objectives

The objectives of this research are to develop cost-effective data on various management practices, and use this data to find optimal combinations of stormwater management practices for storm-sewer systems. The optimal combination of structural and nonstruc-
tural management practices for wet-weather pollution control is determined using linear programming methods.

This is one of the most critical elements of stormwater management research because stormwater removals must be related to lake productivity and the best combination of management practices must be specified to achieve desired water quality improvement. Combinations of computer analyses and mathematical programming are used to analyze the data.

Cost efficiency curves (present value dollars versus removal quantities) are developed in this work for each subwatershed of the Lake Eola watershed. Removal efficiencies from the literature and the local 208 programs are used. These formed the basis for the determination of optimal combinations of practices. The total expansion path or combination of management practices at least cost was developed. This combination was accomplished utilizing a computer program written for this work. A linear programming network routing model was incorporated. The cost-efficiency curves were approximated by "piecewise" linear approximations. This reduces the problem to a linear programming problem.
The cost-efficiency curves generated by the computer program are non-linear and convex. This is illustrated in Figure 1. Axis $X_j$ is that of efficiency, or pounds removed per year, and axis $f_j(x_j)$ is present value cost.

Suppose that one is to solve the following problem:

Minimize:

$$\sum_{j=1}^{n} f_j(x_j),$$

Subject to:

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i=1, \ldots, m \tag{II-1}$$

and $$x_j \geq 0, \quad j=1, \ldots, n$$

First reduce those $f_j(x_j)$ that are nonlinear to their piece-wise linear approximations by fitting linear segments to the original curve as indicated in Figure 1. This approximation can be carried out to any degree of accuracy by dividing the curve into a suitable number of intervals for linear representation. Now assume that each $f_j(x_j)$ is divided into $p_j$ intervals marked by $x_j = 0, d_{j1}, d_{j2}, \ldots, d_{jp_j}$, and that the slopes of the linear segments occupying these intervals are correspondingly $s_{j1}, \ldots, s_{jp_j}$. Since $f_j(x_j)$ is convex by assumption, it follows that $s_{j1} < s_{j2} \ldots < s_{jp_j}$, which makes it possible for us to substitute:
\[ f_j(x_j) \]  
(present value cost)

original non-linear cost/efficiency curve

piece-wise linear approximation

Fig. 1. Convex cost/efficiency curve & piecewise linear approximation
\[ x_j = \sum_{t=1}^{\infty} x_{jt}, \text{ where } 0 \leq x_{jt} \leq d_{jt} - d_j, t-1, \]  

\[ \text{into equation II-1 and obtain the following equivalent problem:} \]

\[
\begin{align*}
\text{Minimize:} & \quad f(x) = \sum_{j=1}^{n} \sum_{t=1}^{\infty} s_{jt} x_{jt} \\
\text{subject to:} & \quad \sum_{j=1}^{n} a_{ij} \left( \sum_{t=1}^{\infty} x_{jt} \right) > b_i, \\
& \quad x_{jt} \leq d_{jt} - d_j, t-1 \\
& \quad x_{jt} > 0 \quad i = 1, \ldots, m, \quad j = 1, \ldots, n, \quad t = 1, \ldots, p_j.
\end{align*}
\]

This now gives a bounded-variable problem with \( m \) constraints and \( \sum_{j=1}^{n} p_j \) variables. It can be solved by the simplex method.

**Linear Programming General Form of the Lake Eola Stormwater Management Problem**

Minimize: \[ \sum_{i,j} C_{ij} R_{ij} \] for all pollutants and management practices.

Subject to:

1. **Mass Flow Constraints**
   
   \[ Q_i - \sum_j R_{ij} = F_i \]

2. **Minimum Removal Constraints**
   
   \[ \sum_i R_{ij} \geq Y_{min} \]
(3) Non-negativity constraints

$$\sum_{i} Q_i; \sum_{i} \sum_{j} R_{ij}; \text{ all } \geq 0$$

where: $C_{ij} = \text{slope of the "jth" segment of the "ith" total cost curve or management practices ($/lb)}$

$R_{ij} = \text{removal quantity of the "jth" segment of the "ith" total cost curve (lb)}$

$Q_i = \text{mass flow to the management practice "i" described by total cost curve}$

$F_i = \text{mass flow from the management practice}$

$$F_i = X_i - \sum_{j} R_{ij} \text{ (lb)}$$

$Y_{\min} = \text{minimum removal of pollutant (lbs)}$

This is demonstrated for a one subwatershed case, a two subwatershed case, and a generalized form for "k" subwatersheds.

**One Subwatershed Linear Programming Formulation**

Figure 2 shows a sketch of one subwatershed. Street patterns are included, representing street sweeping operations, while the catchbasin site represents catchbasin cleaning technology. Diver-sion/percolation and storage treatment may also be included in the subwatershed. Figure 3 illustrates the nodal diagram representative of the one subwatershed.

The formulation of the linear programming problem for one subwatershed is as follows:

**Variables**

$Q_i = \text{lbs of pollutant "i"/yr (for all pollutant i)}$

$R_i = \text{lbs of pollutant "i" removed/yr (for all pollutant i)}$
Fig. 2. Sketch of one subwatershed

Fig. 3. Nodal diagram of one subwatershed
\( F_i = \text{mass flow of } "i" \text{ remaining (} F_i = Q_i - R_i \) \)

**Constraints**

\( R_i \geq 0; \quad R_i \leq Q_i \)

**Nodal Equations**

\[
\begin{align*}
Q_{CB} + F_{SWCB} - R_{CB} &= F_{CB} + S_{CB} \\
Q_{SW} - F_{SWCB} - F_{SWDI} - R_{SW} &= F_{SW} + S_{SW} \\
Q_{DI} + F_{SWDI} - R_{DI} &= F_{DI} + S_{DI} \\
F_{CB} + F_{SW} + F_{DI} &= F_1 \\
S_{CB} + S_{SW} + S_{DI} &= F_{ST} + R_{ST} \\
F_{ST} + R_{ST} &= F_{MAX} \\
\end{align*}
\]

Thus,

\[
\begin{align*}
R_{CB} + R_{SW} + R_{DI} + F_{CB} + F_{SW} + F_{DI} + (S_{CB} + S_{SW} + S_{DI}) - F_{SWCB} + F_{SWDI} + F_{SWCB} - F_{SWDI} &= Q_i \\
F_1 &= F_{CB} + F_{SW} + F_{DI} + F_{ST} \\
R_{CB} + R_{SW} + R_{DI} + F_1 + R_{ST} &= Q_i \\
\end{align*}
\]

Finally,

\[
R_{CB} + R_{SW} + R_{DI} + R_{ST} \geq Q_i - F_1; \quad "\forall i" \\
\]

**Generalized Form (One Subwatershed-All Treatments)**

\[
\sum_{j=1}^{n} R_{ij} \geq Q_i - F_1; \quad "\forall i" \\
\]

where: \( i = \text{pollutant type} \)

\( j = \text{management practice} \)

\( n = \# \text{ of different management practices} \)

and:

\( F_1 \leq F_{MAX} \)

A statement for the generalized form is expressed as: The
quantities of pollutants removed \( (R_{ij}) \) must equal or exceed the difference between available runoff mass and allowable mass discharged into the receiving water body \( (Q_i - F_i) \). In addition, there is a constraint on the amount of pollutants discharged into the receiving water body \( (F_1 \leq F_{\text{max}}) \). Also, the quantity of pollutant removed must not exceed that available in the runoff \( (R_i \leq Q_i) \).

**Two Subwatershed Linear Programming Form**

Figure 4 shows a sketch of two subwatersheds. Street patterns are included, representing street sweeping operations, while the catchbasin sites represent catchbasin technology. Again, diversion/percolation and storage treatment may also be included in the subwatersheds. Figure 5 illustrates the nodal diagram representative of the two subwatersheds.

The formulation of the linear programming problem for the two subwatersheds is as follows:

**Variables**

\[
Q_i = \text{lbs of pollutant "i"/yr (for all pollutant i)} \\
R_i = \text{lbs of pollutant "i" removed/yr (for all pollutant i)} \\
F_i = \text{mass flow of "i" remaining (F_i = Q_i - R_i)}
\]

**Constraints**

\[
R_i \geq 0, \quad R_i \leq Q_i
\]

**Nodal Equations**

\[
Q_{\text{CB1}} - R_{\text{CB1}} + F_{\text{SWD1}} = F_{\text{CB1}} + S_{\text{CB1}} \\
Q_{\text{SW1}} - R_{\text{SW1}} - F_{\text{SWCB1}} - F_{\text{SWD1}} = F_{\text{SW1}} + S_{\text{SW1}}
\]
Fig. 4. Sketch of dual subwatersheds

\[ q_{i1} = q_{CB1} + q_{SW1} + q_{DI1} \]

\[ q_{i2} = q_{CB2} + q_{SW2} + q_{DI2} \]

Fig. 5. Nodal diagram of dual subwatersheds

\[ F_1 = F_{CB1} + F_{SW1} + F_{DI1} + F_{ST1} \]

\[ F_2 = F_{CB2} + F_{SW2} + F_{DI2} + F_{ST2} \]

LAKE
From Nodal Diagram:

\[
Q_{DI1} - R_{DI1} + F_{SWD11} = F_{DI1} + S_{DI1}
\]
\[
F_{CB1} + F_{SW1} + F_{DI1} = F_1
\]
\[
S_{CB1} + S_{SW1} + S_{DI1} = F_{ST1} + R_{ST1}
\]

and

\[
Q_{CB2} - R_{CB2} + F_{SWCB2} = F_{CB2} + S_{CB2}
\]
\[
Q_{SW2} - R_{SW2} - F_{SWCB2} - F_{SWD12} = F_{SW2} + S_{SW2}
\]
\[
Q_{DI2} - R_{DI2} + F_{SWD12} = F_{DI2} + S_{DI2}
\]
\[
F_{CB2} + F_{SW2} + F_{DI2} = F_2
\]
\[
S_{CB2} + S_{SW2} + S_{DI2} = F_{ST2} + R_{ST2}
\]
\[
F_2 + F_1 \leq F_{max}
\]

Thus,

\[
F_{ST1} + R_{ST1}
\]
\[
R_{CB1} + R_{SW1} + R_{DI1} + F_{CB1} + F_{SW1} + F_{DI1} + \left\{ S_{CB1} + S_{SW1} + S_{DI1} \right\} -
F_{SWCB1} + F_{SWCB1} + F_{SWD11} - F_{SWD11} = Q_{11}
\]
\[
F_{ST2} + R_{ST2}
\]
\[
R_{CB2} + R_{SW2} + R_{DI2} + F_{CB2} + F_{SW2} + F_{DI2} + \left\{ S_{CB2} + S_{SW2} + S_{DI2} \right\} -
F_{SWCB2} + F_{SWCB2} + F_{SWD12} - F_{SWD12} = Q_{12}
\]
\[
F_1 = F_{CB1} + F_{SW1} + F_{DI1} + F_{ST1}
\]
Finally,

\[ R_{CB1} + R_{SW1} + R_{DI2} + R_{ST1} \geq Q_{i1} - F_{i1}, \quad "\psi_i" \]

\[ R_{CB2} + R_{SW2} + R_{DI2} + R_{ST2} \geq Q_{i2} - F_{i2}, \quad "\psi_i" \]

**Generalized Form (Two Subwatersheds - All Treatment)**

\[ \sum_{j=1}^{n} R_{ij1} + \sum_{j=1}^{n} R_{ij2} \geq Q_{ik} - F_{ik}, \quad "\psi_{i,k}\]" \]

**Generalized Form: (All Treatment)**

For "k" subwatersheds the generalized form of the linear programming problem is:

\[ \sum_{j=1}^{n} \sum_{k=1}^{l} R_{ijk} \geq Q_{ik} - F_{ik}, \quad "\psi_{i,k}\]"

where:

- \( i \) = pollutant type
- \( j \) = management practice
- \( k \) = subwatershed
- \( l \) = total sub-watersheds
- \( n \) = total possible management practices in a subwatershed

and:

\[ \sum F_{ik} \leq Q_{ik}, \quad "\psi_{ik}\]" and \[ \sum F_{ik} \leq F_{\text{max}} \]

The generalized form states that the quantity of pollutants removed from all subwatersheds \( R_{ijk} \) cannot exceed the difference between available runoff mass and allowable mass discharged into the receiving water body \( Q_{ik} - F_{ik} \). Note also that there are constraints which state:
1. The amount of pollutant flow into the receiving water body \( F_{ik} \) must be less than, or equal to, the total pollutant flow (influent) to the subwatersheds.

2. The amount of pollutant flow into the receiving water body \( F_{ik} \) must be less than, or equal to a maximum tolerance level that the water body can sustain \( (F_{ik} \leq F_{max}) \).
CHAPTER III

LAKE EOLA REVIEW

Lake Eola was chosen as a test site because:

1. historical water quality records were available
2. it is a landlocked lake with no point sources
3. the watershed area and physical picture was well defined

Lake Eola is in an active area of downtown Orlando. It is often used for parades, musical festivities, and simple strolling. Lake Eola has a surface area of 28.75 acres and an approximate volume of 140 million gallons. The lake receives stormwater through storm sewers from the commercial and residential areas surrounding it (Wanielista 1978). A fountain in the lake is the major attraction to residents and tourists, with no swimming, motor boating or fishing being allowed.

The normal water level of Lake Eola is between 87.5 and 88.5 feet above sea level. The level of the lake is controlled by two drainage wells. These wells drain into the artesian aquifer and can be used for raising or lowering the lake level. The maximum water depth of Lake Eola is 22 feet, with a gradual decrease to a shore depth of 2 to 3 feet.

The principle sources of pollution are through the 12 active street drains servicing the urban watershed surrounding the lake.
Trash catchers on eight of the twelve drains are the only form of treatment at this time.

A restoration project of Lake Eola was undertaken in 1972. The lake was lowered exposing 40% of the bottom. The bottom was cleaned and dredged, sand was placed, and the stormdrains were extended into the lake. Gradually, the water quality has declined. Today algae are present and increasing. Fish and duck kills have been documented. It is this degradation that has prompted federal, state, and local assistance in a restoration project of Lake Eola.
CHAPTER IV

STORMWATER MANAGEMENT PRACTICES

There are a number of management practices available for the abatement, control, and treatment of stormwater runoff. These practices, or Best Management Practices (BMPs) have been researched and can be classified as structural or nonstructural. Lynard and Finnemore (1978) define these practices as "Best Management Practices". "These nonstructural and low-structural source controls offer considerable promise as the first line of defense to control urban runoff pollution. For developing areas, BMPs are implemented through planning, legislation, and enforcement with goals of maximizing detention/percolation, avoidance of over-development or land misuse, and minimizing impacts of construction activities. For developed areas, sound maintenance and operating practices are required for (1) litter and chemical use control, (2) street cleaning and repair, (3) catchbasin and collection systems maintenance, (4) runoff flow controls, and (5) public support and involvement. BMPs have decided benefits over structural alternatives including lower cost, earlier results, and an improved and cleaner neighborhood environment. The greatest difficulty, however, is that the action-impact relationships are almost totally unqualified." Figure 6 illustrates the different management methods, separating them as structural
Fig. 6. Management methods

and non-structural.

The management practices selected for review in this thesis are:

1. diversion/percolation
2. detention
3. swales
4. dutch drains
5. porous pavement - asphalt
6. street sweeping
7. catchbasin cleaning
8. fabric bags
9. storage/treatment
   a. storage
   b. roof-top storage
   c. microstraining
   d. swirl concentrators

A short analysis and explanation of each practice is given.

For the purposes of the optimization study, three of the above technologies were considered: (1) diversion/percolation, (2) street sweeping, and (3) catchbasin cleaning. A computer program was written for these three practices, generating the cost/efficiency curves of each. These curves were reduced to their slope and intercept data which was used in a linear programming, piece-wise approximation program. The program and explanation is available in a future chapter of this report.
Diversion/Percolation

Diversion/Percolation or retention areas are natural or man-made holding areas providing storage of surface runoff water resulting in evapotranspiration and percolation. In these areas, soil and cover crops provide "natural treatment" for many pollutants, including suspended solids, BOD$_5$, nitrogen, phosphorus and metals found in stormwater runoff. In the final evaluation, the use of these retention areas depends on removal efficiencies and cost. Removal efficiencies are defined as the yearly average surface water mass of pollutants removed by treatment through soil percolation. Both cost and efficiencies are a function of the size of basins. The size of basins are a function of watershed size, quality of diverted water, percent imperviousness of land area, amount of pollutant removal (efficiency) required, and permeability of the soil. Further reading from Wanielista and Calabrese is offered (1977).

If first flush surface runoff is diverted into retention areas, and the first inch of stormwater runoff can be stored and treated, then the quantity of pollutants removed from direct surface discharge to adjacent lands and water bodies is about 99% of yearly runoff mass. If conduits are used to transport a number of first flushes from various areas at different times, the concept of first flush is no longer valid because the pollution concentrations are random. In this case, the efficiencies must be calculated from cumulative runoff distributions for associated rational runoff co-
efficients. Both these cases are provided for in the computer program. Efficiencies and cost data are presented in Table 1.

Figure 7 offers a sketch of a typical percolation pond with a common diversion structure. Varying the height of the diversion baffle will alter the diversion volume discharged to the pond. Diversion of the first flush of pollutants to treatment is a popular practice because of the belief that the first flush contains a greater percentage of pollutants than remaining runoff waters. The quantification of the first flush is usually in terms of the first volume of runoff waters from a watershed. Diversion with percolation (retention) was specified as the most cost-effective alternative for the Central Florida area (Wanielista 1977).

Detention

Detention, or sedimentation systems are another popular management practice because hydrograph peaks are reduced and water bodies are created (Figure 8). However, pollution efficiencies are not as great as diversion/percolation systems (see Table 1).

The purpose of detention basins are to retain or detain runoff in order to increase the time of concentration, or to reduce the maximum discharge rate of runoff from an urban area. A detention basin is a most effective technique for reducing the peak flow at a point immediately downstream of the impoundment (Turbier 1974).

The advantages of detention facilities are their ability to catch a large portion of suspended solids loading from the watershed. In some cases, these basins may provide sedimentation.
**Fig. 7.** Diversion/percolation structure

Another advantage is the aesthetic and recreational benefits of detention ponds. Detention is one viable management practice in areas where a pleasing environment is desired, if they are designed carefully. Such an advantage is required in the Lake Eola watershed.

Successful variations of detention that take advantage of facilities primarily used for other purposes are ponding on parking lots, plazas, recreation and park areas, and ponding on roof-tops. The fundamental approach is the same as for other forms of detention, but low cost is implied. Dual purpose basins used for storage and athletics when dry are also employed.

**Swales**

Swales, or grass channels are another management practice. They are aesthetically pleasing and are often designed for residential locations. They are also utilized quite often along roadways and highways to control runoff from the paved surfaces.

The swales can be built in parabolic, trapezoidal, or V-shaped cross sections (Figure 9). The soil type should be characteristic of well drained conditions. Side slopes should be approximately 4 horizontal to 1 vertical or less. Suitable grade favorable to vegetative growth must be found or favorably designed. Flow velocities should not exceed 4-6 feet per second (Wanielista 1977). Swales reduce the velocities of runoff and considerably reduce the energy, and consequently the erosive capacity of runoff. They also allow percolation of runoff, thus improving water quality. In general, swales allow for the advantages of detention and diver-
It is important to avoid excessive compaction during construction by earthmoving machinery which will result in an inferior percolation capacity. Between the time of seeding the cover crop and the actual establishment, the waterway will be unprotected and subject to damage. Provisions should be made to divert flows during this period (Wanielista 1977).

Cost and efficiency data for the central Florida area were estimated by Wanielista and Calabrese (1978) and are reproduced in Table 1.

**Dutch Drains**

Dutch drains are a management practice used to reduce the volume of storm runoff and reduce flood peaks by increasing ground infiltration. They prevent concentration of runoff by intercepting it, as compared to percolation basins.

In the process of reducing the volume of runoff, dutch drains allow the advantages of enhancing ground water supplies, improve the quality of vegetation on site by increasing available water in the ground, and result in a reduction in the size of storm drains required downstream of these devices (Tourbier 1974). This is all accomplished by allowing runoff to percolate into the ground before a "sheet" buildup can occur. All these benefits point toward advantageous use in the Lake Eola watershed. With the excellent soils available in the watershed, dutch drains or some form of percolation should be considered.
Illustrations of typical installations are shown in Figure 10.

**Porous Paving - Asphalt**

Porous paving is another management practice used to reduce the volume of storm runoff by increasing infiltration.

Although still experimental, porous paving has been found to reduce the total volume of runoff from paved areas. It also enhances ground water supplies, can result in savings by eliminating need for storm sewers and curbing, improves skid resistance by minimizing water on road surfaces, provides benefits for pedestrians by eliminating puddles, and preserves the natural urban draining patterns of an area. (Tourbier 1974). These benefits point toward consideration of porous paving on the roads and parking lots of newly developed or planned watersheds. However, the Lake Eola watershed is developed and the use of porous pavement would be cost prohibitive.

Costs here are again variant, yet low with only slight ORM costs resulting from vacuum sweeping to keep surface porosity as great as possible.

**Street Sweeping**

Street Sweeping is the management practice of removing particles and debris from impervious street areas with the use of mechanical cleaners or by manual means. The mechanical cleaners come in 2 categories. One is a simple brush type device which sweeps the
Fig. 8. Detention basin

Fig. 9. Swale cross section

Fig. 10. Dutch drain and porous asphalt devices
### TABLE 1

**STORMWATER MANAGEMENT COMPARATIVE DATA/IMPERVIOUS ACRE (LAND COST NOT INCLUDED)
(CENTRAL FLORIDA DATA)**

<table>
<thead>
<tr>
<th>Management Practices</th>
<th>Overall Efficiency (%)</th>
<th>ORM $/AC-mon</th>
<th>Average Cost ($/ac/% removal) Capital</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diversion/Percolation b</td>
<td>99</td>
<td>10.00</td>
<td>25.00</td>
</tr>
<tr>
<td>Percolation Pond c</td>
<td>99+</td>
<td>35.00</td>
<td>36.00</td>
</tr>
<tr>
<td>Swales with Percolation d</td>
<td>92</td>
<td>30.00</td>
<td>28.40</td>
</tr>
<tr>
<td>Residential Swales d</td>
<td>80</td>
<td>20.00</td>
<td>26.08</td>
</tr>
<tr>
<td>Sedimentation e</td>
<td>64</td>
<td>29.00</td>
<td>19.20</td>
</tr>
<tr>
<td>Fabric Bag f</td>
<td>44</td>
<td>26.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Advanced Sweeping g</td>
<td>68</td>
<td>26.00</td>
<td>30.40</td>
</tr>
</tbody>
</table>


- aYearly average of BOD$_5$, N, P, and SS not discharged to surface waters
- bDesigned 1-in. runoff diversion
- cDesigned for 4-in. runoff diversion
- dEighty percent of the rainwater percolates
- eDesigned for 0.65 in. of runoff water
- fFabric bag replacement every two years
- gAssumed 60% nitrogen in particulate form
pollutant mass from the street into a holding area on board for later disposal. The other type of cleaner is the vacuum street cleaner which utilizes a powerful suction to remove and store debris until later disposal. The efficiencies of both devices differ and are estimated by manufacturers. Some efficiency data is shown in Table 2 for brush type sweepers at various particle size ranges (Heaney 1977). Note the overall efficiency of about 50% and the fact that coarser materials are more efficiently removed.

APWA reports that vacuum type cleaners have achieved efficiencies of greater than 95% (Heaney 1977). This increase in efficiency over brush type sweepers results in a substantially higher cost.

There are both advantages and disadvantages to street sweeping. Advantages include:

1. control of pollutants at their source
2. sweeping is a pollution control practice, and it also improves the aesthetics of an area

Disadvantages include:

1. removals include only the gutter length area of the street, not the entire street
2. inconveniences to car owners who must use alternate side of the street parking
3. low efficiencies of pollutant removals as compared to other management practices (see Table 3)

Street sweeping in itself may not be an extremely efficient management practice, yet it could be utilized with other manage-
### Table 2

**Efficiency Data for Brush Type Street Sweepers**

<table>
<thead>
<tr>
<th>Particle Size (microns)</th>
<th>Sweeper Efficiency (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>79</td>
</tr>
<tr>
<td>840 - 2000</td>
<td>66</td>
</tr>
<tr>
<td>246 - 840</td>
<td>60</td>
</tr>
<tr>
<td>104 - 246</td>
<td>48</td>
</tr>
<tr>
<td>43 - 104</td>
<td>20</td>
</tr>
<tr>
<td>&lt; 43</td>
<td>15</td>
</tr>
<tr>
<td>overall</td>
<td>50</td>
</tr>
</tbody>
</table>

### Table 3

**Broom Sweeping Efficiencies Low Versus Average Ability**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Efficiencies (%)</th>
<th>Low (Altamonte Mall)</th>
<th>National Average Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solids</td>
<td>83</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>BOD$_5$</td>
<td>2</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>5</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

ment practices making it an attractive and viable alternative. Recent studies in the area of street sweeping indicate that a portion of the material found on the streets (and therefore a potential pollution source during runoff events) may be removed by a conscientious sweeping program (Heaney 1977).

The reader is asked to review the references for further, more in-depth studies of street sweeping.

**Catchbasin Cleaning**

Control of stormwater is a problem of increasing importance in the field of water quality management. Over the past 70 years, there has been some extensive use of catchbasins for coarse material removal from stormwater runoff (Lager 1977).

Catchbasins were historically designed to prevent sewer clogging by removing larger, coarse debris before they entered the sewer system. Today, however, catchbasins are thought to be a stormwater management practice of some merit by their ability to retain solids carried in stormwater runoff. A catchbasin is defined as a chamber or well, usually built at the curbline of a street, for the admission of surface water to a sewer or subdrain, having at its base a sediment sump designed to retain grit and detritus below the point of overflow. Because some communities call any device that receives stormwater a catchbasin, the distinction is made between those devices that intentionally trap sediment and those that do not. The device that traps sediment is called a catchbasin and the device that does not is called an inlet.
The entrance to either the catchbasin or the inlet is through a grate and/or a curb opening; or in the case of a catchbasin not connected directly to the street but supplied from one or more inlets, the entrance is through an inlet pipe (Lager 1977).

Catchbasins serve two main purposes: to prevent sewer gases from escaping through the inlet gratings and to prevent solid matter from the street from entering the sewers. The trapping of sewer gases is accomplished by water seals of different types. The retention of solids is achieved by providing a sump or settling basin in which the heavy solids settle to the bottom while the light solids float on top. The water drains to the sewers through the inlet of a trap, which is generally a few inches below the water surface (Lager 1977).

Although a universally accepted catchbasin design has not yet been agreed upon Figure 11 shows representative catchbasin designs in the U.S. and Canada.

The required catchbasin cleaning is achieved by four categories of cleaning methods: manual cleaning, bucket cleaning, eductor cleaning, and vacuum cleaning.

Manual Cleaning

In this method, bailing the water out of the sump and dipping out the material accumulated below using long-handled right angle spoons or dippers is utilized. The solid material is then piled onto the street surface for removal by truck. A fire hose then refills the basin with clean water. This is not the most desirable
Fig. 11. Representative Catchbasin Designs in United States and Canada.

method of cleaning due to odor and unsightliness problems.

Bucket Cleaning

This method consists of lowering a standard or specially designed orange peel or clamshell bucket into the catchbasin, lifting the full bucket to the surface, and emptying it into a dump truck. This method is fairly effective in removing all the basin contents, yet the buckets do spill quantities of water onto the street surface causing odor and aesthetic problems.

Eductor Cleaning

The vacuum effect of an eductor is used in this method to draw up the catchbasin contents. The contents are separated to solids and water in the eductor truck tank compartment with the water being recycled to operate the eductor. This method is a sound approach to catchbasin cleaning and should be considered for new catchbasin designs or for modification of existing basins (Lager 1977).

Vacuum Cleaning

This is a very similar method to eductor cleaning, except that an air blower is used to create the vacuum, not water, and the air-solids-liquid separation is accomplished in the unit by gravity separation and baffles (Lager 1977). Usually, larger pieces of debris can be removed by the vacuum unit than with an eductor unit.
Cleaning Frequencies

Cleaning frequency must be cleaned as a minimum requirement often enough to prevent the buildup of solids to such a depth that they might block the outlet to the sewer. This frequency is a function of several parameters, such as sump capacity, quantity of accumulated street solids, antecedent dry period, meteorological conditions, street cleaning methods and practices, surrounding land use, topography, and to some extent, the type of surface soil adjacent to the street. Table 4 shows the frequencies of catchbasin cleaning in various north American cities.

Storage Basin

Sizing a catchbasin must be determined by site specific studies. Review of J. A. Lager's *Catchbasin Technology Overview and Assessment* (1977) will give excellent design and hydraulic criteria, as well as cost and efficiency data.

It is concluded from a review of the literature that properly designed and maintained catchbasins can be very efficient in removing medium to very coarse solids from stormwater runoff. Coupled with an active street sweeping campaign, urban stormwater runoff can be cost-effectively managed.

Fabric Bags

Fabric bags are devices placed inside stormsewers, preventing solids from entering the receiving waters. They consist of a fabric filter placed below the influent flow of stormwater, causing runoff
### TABLE 4

**FREQUENCY OF CATCHBASIN CLEANING IN VARIOUS NORTH AMERICAN CITIES**

<table>
<thead>
<tr>
<th>Frequency</th>
<th>No. of Cities</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1973</td>
</tr>
<tr>
<td>As needed</td>
<td>...</td>
</tr>
<tr>
<td>Once in 4 years</td>
<td>3</td>
</tr>
<tr>
<td>Once in 3 years</td>
<td>7</td>
</tr>
<tr>
<td>Once in 2 years</td>
<td>13</td>
</tr>
<tr>
<td>Twice in 3 years</td>
<td>1</td>
</tr>
<tr>
<td>1 time per year</td>
<td>142</td>
</tr>
<tr>
<td>1.5 times per year</td>
<td>1</td>
</tr>
<tr>
<td>2 times per year</td>
<td>81</td>
</tr>
<tr>
<td>2.5 times per year</td>
<td>...</td>
</tr>
<tr>
<td>3 times per year</td>
<td>21</td>
</tr>
<tr>
<td>3.5 times per year</td>
<td>...</td>
</tr>
<tr>
<td>4 times per year</td>
<td>13</td>
</tr>
<tr>
<td>5 times per year</td>
<td>2</td>
</tr>
<tr>
<td>6 times per year</td>
<td>5</td>
</tr>
<tr>
<td>7-8 times per year</td>
<td>1</td>
</tr>
<tr>
<td>9 times per year</td>
<td>3</td>
</tr>
<tr>
<td>10-15 times per year</td>
<td>2</td>
</tr>
<tr>
<td>20-26 times per year</td>
<td>1</td>
</tr>
<tr>
<td>31 times per year</td>
<td>1</td>
</tr>
<tr>
<td>45 times per year</td>
<td>1</td>
</tr>
<tr>
<td>52 times per year</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>299</td>
</tr>
<tr>
<td>Median annual frequency</td>
<td>1</td>
</tr>
<tr>
<td>Mean annual frequency</td>
<td>2.3</td>
</tr>
<tr>
<td>Mean of middle 80% of cities</td>
<td>1.5</td>
</tr>
</tbody>
</table>

to pass through the filter before discharge. Overall removal efficiencies; overall efficiency defined as the average of BOD$_5$, nitrogen, phosphorus, and suspended solids efficiencies, is found to be 44% (Wanielista 1978). This comes from central Florida data. Capital and ORM costs in this area for fabric bags can be seen in Table 1.

**Storage Treatment**

The remaining method of controlling stormwater pollution involves storage and/or treatment of the collected runoff. Storage-treatment facilities operate in series with the management practices. A variety of storage and treatment technologies are available. Examples of storage include:

1. in-line storage
2. tanks
3. lagoons
4. tunnels

Treatment methods include:

1. sedimentation
2. chemical treatment
3. physical treatment

Physical treatment may include:

1. microstraining
2. swirl concentrators

and others. Physical treatment methods will be discussed.
Storage

Storage is perhaps the most cost-effective method available for reducing pollution resulting from urban stormwater runoff (Field 1977). Storage facilities are frequently used to alternate peak flows, thus reducing the size of facilities required for further treatment.

The concept of storage systems is to capture stormwater runoff quantities and slowly discharge it to treatment during dry weather periods when the treatment process is not at capacity.

Storage facilities possess many favorable attributes (Field 1977): (1) they are basically simple in structural design and operation; (2) they respond without difficulty to intermittent and random storm behavior; (3) they are relatively unaffected by flow and quality changes; and (4) they are capable of providing flow equalization. Finally, storage facilities are non-mechanical and free from severe operating problems.

Storage facilities may be designed for in-line or off-line duty, and they can be open or closed systems. Some storage systems involve the use of underground tanks to hold quantities of stormwater. These are closed systems. Open systems may involve detention ponds or roof-top storage.

Roof-Top

To lower the hydrograph peaks by delaying runoff, roof-top storage is sometimes utilized.

An advantage allowed from roof-top storage is the possible
reduction in the size of storm drainage facilities over the watershed. This is a benefit in the Lake Eola watershed. However, in the Florida area, roof-tops are not designed for heavy loads (snow); stabilization and reinforcement may be required.

Mechanics of roof-top storage are illustrated in other references (Tourbier 1974). Roof-top runoff flows slowly through a set of holes on the rainfall detention ponding ring. From here it flows through a strainer and down a vertical leader. Percolation into the surrounding grounds may be the final destination of this runoff, or other treatment practices can be incorporated.

Microstraining

Microstrainers, or microscreens are conventionally designed for polishing secondary sewage plant effluent at an optimum rate of approximately 10 gpm/ft². They are also used in water treatment systems, and recently have gained acceptance in stormwater management.

The microscreen unit consists of a motor driven rotating drum mounted horizontally in a rectangular chamber. A fine screening media, called the microfabric, covers the drum. Feedwater enters the drum interior through the open end and passes radially through the screen since the other end of the drum is closed. This allows the deposit of solids associated with the feedwater onto the inner surface of the screen. As the drum rotates, the solids remain fixed to the screen and are rotated out of the water towards the top of the microscreening device. At the top of the drum, pressure
jets of effluent water are directed onto the screen to remove the mat of deposited solids. The dislodged solids, together with the portion of backwash effluent stream which penetrates the screen are captured in a waste hopper.

Microscreening is an alternative that was reviewed in the Lake Eola watershed. Removal efficiencies indicate suspended solids and BOD removals of close to 80%. Operating costs are low, approximately 1.5 dollars per million gallon treated; yet capital and maintenance costs are high.

Swirl Concentrator

The swirl concentrator device is a physical management practice providing outstanding potential for both quality and quantity control.

The basic conventional methodology of primary clarification usually involves the gravimetric separation of settleable solids, together with floatable materials, in flow-through chambers which provide detention periods and lowered hydraulic velocities that produce deposition and rise of solids and the resultant discharge of a clarified effluent. The swirl concentrator achieves clarification of solids-laden liquid flows, not by simple gravimetric separation under quiescent settling conditions, but by introducing secondary motion flows. The concentrator involves the establishment of long-path circular separation within a cylindrical tank (Sullivan 1978).

The absence of moving parts in the swirl concentrator reduces maintenance and operating costs of the unit.
CHAPTER V

COMPUTER PROGRAM "MANAGE" DISCUSSION

This chapter proceeds with a simplified flow diagram of the computer program (Figure 12). It is written in Fortran language and run on the IBM 360/370 series computer system. It analyzes 3 management practices (catchbasin cleaning, street sweeping, and diversion/percolation) chosen as representative practices. Others can be added by expanding the program.

Main Selector Routine

The main selector program contains the "read" statements for all subwatershed data. Separate data for each management practice is found in the various subroutines.

In the selector routine, the user can choose which of the three management practices he wishes to be analyzed in the optimization routine. Any of seven combinations listed below are possible:

1. catchbasin cleaning - street sweeping - diversion/percolation
2. catchbasin cleaning - street sweeping
3. catchbasin cleaning
4. catchbasin cleaning - diversion/percolation
5. street sweeping - diversion/percolation
6. diversion/percolation
Fig. 12. Computer program "MANAGE" flow chart
7. street sweeping

After the selection of management practices is made, a subroutine is called to set-up the completed data into a special form for subroutine LPGOGO (the linear programming optimization step). This subroutine will be either TABLO or TABLO1 through TABLO4, depending on the combination of management practices selected. TABLO stands for "tableau", the initial data tableau of any linear programming problem.

From one of the TABLO subroutines, LPGOGO is called and run. LPGOGO is executed for five levels of pollutant removals for each watershed using predetermined percent removals. These percent removals should be spread out to range from 0% to 100%, so that the cost-efficiency curves are estimated with minimum error. This program utilizes PER values of 40%, 65%, 80%, 90% and 95%. It is worthy to note that the "X" in the PYEEX represents either suspended solids ("S"), BOD₅ ("B"), nitrogen ("N"), or phosphorus ("P"), depending on which pollutant is selected.

\[
\text{PYRRS}(I) = \text{PREM}(I) \times \text{SSRT} \times \text{AR}
\]

where: \( \text{PYRRX}(I) = \) pollutant removals for suspended solids (lb/yr)
\( \text{PREM}(I) = \) percent removals for all subwatersheds, (%)
\( = \) PER1 through PER5
\( \text{SSRT} = \) suspended solids loading rate (lb/ac - yr)
\( \text{AR} = \) area of subwatershed (acres)

Summary data statistics for each subwatershed are printed after LPGOGO. This includes the subwatershed identification, the
pollutant selected for the optimization analysis, the management practices selected by the linear programming subroutine, along with associated pounds removed and cost, and data such as area of subwatershed, and loading rates. Calculations and subroutines described above will be run for each successive subwatershed, up to 20 subwatersheds. A linear programming optimization step is next run on the "n" subwatersheds to find an optimal solution for the entire watershed. A maximum pollutant load removal efficiency before discharge to receiving waters determines the level of treatment desired in the watershed.

Variable descriptions for the main selector routine are presented. Appendix A contains a copy of the actual computer program for review.

Main Selector Routine Variable Descriptions

N = number of rows in matrix A, or the number of points for the cost/efficiency graphs
M = number of columns in matrix A, or the number of pollutants plotted on one graph
NO = chart number (3 digits maximum)
ISLC = subroutine CATCH selector
ISLS = subroutine SWEEP selector
ISLB = subroutine BASIN selector

If ISLC, ISLS, or ISLB = 1, that practice is selected

ISSS = pollutant selector for suspended solids
ISBD = pollutant selector for BOD$_5$
ISNI = pollutant selector for nitrogen
ISPH = pollutant selector for phosphorus

If any of the four selectors is 0, that pollutant is chosen.

Note: only one of the four pollutant selectors ISSS, ISBD, ISNI, and ISPH can be 0 at a time.

NL = number of lines in the plot (y axis size)

If NL = 0, 50 lines are used.

AR = area of subwatershed - acres

SSRT = suspended solids loading rate - lb/ac - yr

BODRT = BOD loading rate - lb/ac - yr

TRT = nitrogen loading rate - lb/ac - yr

PRT = phosphorus loading rate - lb/ac - yr

XMAX = the maximum pollutant load removed before being discharged to receiving waters

NWAT = number of subwatersheds to be analyzed

PER(I) = percent removals for all subwatershed, a total of five to be read, lowest value read first

Catchbasin Cleaning Subroutine - Subroutine CATCH

Subroutine CATCH takes the efficiencies of catchbasin cleaning at five different frequencies of cleaning and multiplies them by the loadings of each pollutant. This gives the pounds per year removed.

\[ QBYRS(I) = YRLS \times EFS(I) \]  \hspace{1cm} (V-2)

where: \( QBYRS = \text{pounds per year removed of pollutant suspended solids} - (\text{lb/yr}) \)

\( YRLS = \text{yearly loading of suspended solids} - (\text{lb/yr}) \)

\( EFS = \text{SSRT} \times \text{AR} \)
SSRT = suspended solids loading rate - (lb/ac - yr)
AR = subwatershed area (acres)

EFFS(I) = efficiency of catchbasin cleaning at "I" different frequencies

Next calculated are the ORM, capital, and present value costs of catchbasin cleaning at the five different frequencies of cleaning.

\[ \text{COSTCB}(I) = \text{NB} \times \text{CCB} \times \text{XCB}(I) \]  
\[(V-3)\]

where: \( \text{COSTCB}(I) = \text{ORM} \) cost of catchbasin cleaning - $
\text{NB} = \text{number of catchbasins in a given area}
\text{CCB} = \text{cost of cleaning catchbasins} - $/catchbasin
\text{XCB}(I) = \text{frequency of cleaning catchbasins} - /year

\[ \text{CAPIT}(I) = \text{COST} \times \text{FREQC}(I) \]  
\[(V-4)\]

where: \( \text{CAPIT}(I) = \text{capital cost of catchbasin cleaning} - $
\text{COST} = \text{capital cost of inlet conversion to catchbasin} - $
\text{FREQC}(I) = \text{frequency of cleaning catchbasin} - /month
\text{XCB}(I)/12.

\[ \text{PV}(I) = \text{CAPIT}(I) + \text{ORM}(I) \]  
\[(V-5)\]

where: \( \text{PV}(I) = \text{present value of catchbasin cleaning}, (20\text{yr}, 7\%) -$
\text{CAPIT}(I) = \text{capital cost of catchbasin cleaning} - $
\text{ORM}(I) = \text{ORM} \text{ cost spread over ENY years at EINT interest rate}
\text{COSTCB}(I) = \text{ORM} \text{ cost of catchbasin cleaning} - $
\text{EINT} = \text{interest rate for present value analysis} \text{ fraction)
ENY = number of years in present value analysis

Data for these calculations are either read in for the subroutine on a separate data card, or carried through the program from the main selector routine data card. Appendix A will be useful to consult if misunderstandings arise.

From this compiled data, cost/efficiency curves are plotted. This is accomplished by transferring CATCH data to subroutine PLOT. Following plot of the curves, slope and intercept data is calculated by subroutine SLOPE, which is later used in the piece-wise linear programming subroutine LPGOGO for an optimization analysis. All subroutines can be seen in Appendix A of this report.

Street Sweeping Subroutine - Subroutine SWEEP

Subroutine SWEEP begins by computing the gutter length of a given area in curb - mi/ac (Graham 1974).

\[
GL = 0.0782 - (0.0668 \times (0.839 ^ {PD}))
\]  \hspace{1cm} (V-6)

where: GL = gutter length - curb - mi/ac

PD = population density of a given area - persons/ac

Like subroutine CATCH, SWEEP takes the efficiencies of street sweeping at five different frequencies of cleaning and multiplies them by the loadings of each pollutant. This gives the pounds per year removed by this management practice. By utilizing the default features of the subroutine, one can simulate the functions of either brush or vacuum type street cleaners. Inherent to the program are brush type street sweeper data.

\[
QBYRS(I) = YRLS \times EFFS(I)
\]  \hspace{1cm} (V-7)
where:  

QBYRS = pounds per year removed of pollutant suspended solids - lb/yr

YRLS = yearly loading of suspended solids on street surface - lb/yr

= SSRT * GL * AR

SSRT = suspended solids loading rate - lb/ac - yr

GL = gutter length of given area - curb mi/ac

AR = subwatershed area - acres

EFFS(I) = efficiency of street sweeping at "I" different frequencies

Next calculated are the ORM, capital and present value costs of street sweeping at the five different frequencies of cleaning.

COSTSW(I) = 365. * CSW * GL * AR * XSW(I)  \hspace{1cm} (V-8)

where:  

COSTSW(I) = ORM cost of street sweeping - $

365. = 365 day/yr conversion factor

CSW = cost of street sweeping - $/curb-mi

GL = gutter length of given area - curb-mi/ac

AR = subwatershed area - acres

XSW(I) = frequency of street sweeping - /yr

CAPSW(I) = COSTS * XSW(I) * GL * AR/26.  \hspace{1cm} (V-9)

where:  

CAPSW(I) = capital cost of street sweeping - $

COSTS = capital cost of a street sweeper - $

XSW(I) = frequency of cleaning streets - /yr

GL = gutter length of given area - curb-mi/ac

AR = subwatershed area - acres

26. = assumption that a sweeper can cover 26 curb-miles/day
\[ PVS(I) = \text{CAPSW}(I) + \text{ORMSW}(I) \]  
where: \( PVS(I) \) = present value of street sweeping, (20ys, 7\%) - $
\text{CAPSW}(I) = \) capital cost of street sweeping - $
\text{ORMSW}(I) = \) ORM cost spread over ENYS years at EINTS interest rate

\[ \text{ORMSW}(I) = \text{COSTSW}(I) \times \left( \frac{(1 + \text{EINTS})^{\text{ENYS}} - 1}{(1 + \text{EINTS})^{\text{ENYS}}} \right) \]

where: \( \text{COSTSW}(I) = \) ORM cost of street sweeping - $
\text{EINTS} = \) interest rate for present value analysis (fraction)
\( \text{ENYS} = \) number of years in present value analysis

From the computed data, cost/efficiency curves are plotted through subroutine PLOT. This is accomplished by transferring SWEEP data to subroutine PLOT. Following this subroutine, slopes and intercepts are calculated through subroutine SLOPE. This data is later used in the piece-wise linear programming subroutine LPGOGO for an optimization analysis. All subroutines can be seen in Appendix A of this report.

**Diversion/Percolation Subroutine - Subroutine BASIN**

Subroutine BASIN proceeds by first calculating the yearly loadings of four pollutants (suspended solids, BOD\textsubscript{5}, nitrogen, and phosphorus). This is accomplished by multiplying the loading rates of each pollutant by the area of the given watershed. This is the same procedure followed in subroutines CATCH and SWEEP.

Next, volumes of basins are calculated, sized depending on soil types available in the watershed. Soil type options are TYPE
soils possess a low runoff potential. These soils have a high infiltration rate, even when thoroughly wetted, and consist of deep, well to excessively drained sands or gravels.

Type D soils possess high runoff potential. These soils have a very slow infiltration rate when thoroughly wetted and consist chiefly of clay soils with a high swelling potential, soils with a permanent high water table, soils with a claypan or clay layer at or near the surface, and shallow soils over nearly impervious material. Although there are also the additional SCS soil types of B and C, it is felt that type B was similar in characteristics to soil type A and type C is similar in characteristics to soil type D. This reduced the requirements and complexity of the computer program.

When a soil type is selected through the variable IS (IS = 0 for soil type A, IS = 1 for soil type D), volumes of percolation basins are calculated by the equations in Table 5.

Pollutant removals are next calculated in the program. If first flush phenomena are observed, efficiencies used in the program are as shown in Table 6. If first flush phenomena are not observed, efficiencies can be determined by the rational runoff coefficients. This is accomplished by subroutine CVALU which interpolates efficiencies between values of rational runoff coefficient C. This can be seen in Table 7.

Subroutine CVALU interpolates values of efficiency between rational runoff coefficient values of 0 to 1. C value efficienc-
## TABLE 5A

POND VOLUME FOR TYPE D SOILS

<table>
<thead>
<tr>
<th>Diversion Volume (in.)</th>
<th>Pond Volume, ac-ft</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3-ft-Deep Pond, Impervious Watershed</td>
</tr>
<tr>
<td></td>
<td>$V_1 = 0.02(A)^{1.31}$</td>
</tr>
<tr>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>0.50</td>
<td>$V_1 = 0.05(A)^{1.24}$</td>
</tr>
<tr>
<td>0.75</td>
<td>$V_1 = 0.13(A)^{1.11}$</td>
</tr>
<tr>
<td>1.00</td>
<td>$V_1 = 0.20(A)^{1.07}$</td>
</tr>
<tr>
<td>1.25</td>
<td>$V_1 = 0.29(A)^{1.04}$</td>
</tr>
</tbody>
</table>

where:

- $V_m$ = minimum basin volume, ac-ft
- $A$ = contributing watershed area, ac
- $DI$ = diversion volume, in.
- $12$ = conversion factor, in/ft
- $V_D$ = volume of basin at depth "D" in Type "D" soil, ac-ft
- $D$ = depth of basin, ft
- $CN$ = composite curve number
- $V_3$ = basin volume at 3-ft depth, ac-ft
- $V_1$ = basin volume for impervious area, 3-ft depth, ac-ft

For D Soils in Percolation Pond Area (Minimum Percolation Rate = 0.25 in/hr)

TABLE 5B

POND VOLUME FOR TYPE A SOILS

<table>
<thead>
<tr>
<th>Diversion Volume (in.)</th>
<th>5-ft-Deep Pond, Impervious Watershed</th>
<th>5-ft-Deep Pond, Composite Land Use</th>
<th>1 ft &lt; Pond Depth &lt; 5 ft, Composite Land Use</th>
<th>Pond Depth = 1 ft</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>( V_1 = 0.016(A) )</td>
<td>( V_5 = V_1 (0.59 + 0.37 \frac{CN}{100}) )</td>
<td>( V^A_D = V_m + \frac{V_5 - V_m(D - 1)}{4} )</td>
<td>( V_m = \frac{A \times DI}{12} )</td>
</tr>
<tr>
<td>0.50</td>
<td>( V_1 = 0.046(A) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.75</td>
<td>( V_1 = 0.09(A) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>( V_1 = 0.14(A) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.25</td>
<td>( V_1 = 0.20(A) )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

where:
- \( V_m \) = minimum basin volume, ac-ft
- \( A_m \) = contributing watershed area, ac
- \( DI \) = diversion volume, in.
- \( 12 \) = conversion factor, in./ft
- \( V^A_D \) = volume of basin at depth "D" in Type A soil, ac-ft
- \( D \) = depth of basin, ft
- \( CN \) = composite curve number
- \( V_5 \) = basin volume at 5 in. depth, ac-ft
- \( V_1 \) = basin volume for impervious area, 5-ft depth, ac-ft

For A Soils in Percolation Pond Area (Minimum Percolation Rate = 1.0 in./hr)

**TABLE 6**

**EFFICIENCIES OF DIVERSION/PERCOLATION**
(First Flush Phenomena)

<table>
<thead>
<tr>
<th>% Efficiencies</th>
<th>Diversion Volume, in.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secondary System</td>
<td></td>
</tr>
<tr>
<td>99.9</td>
<td>1.25</td>
</tr>
<tr>
<td>99</td>
<td>1.00</td>
</tr>
<tr>
<td>95</td>
<td>0.75</td>
</tr>
<tr>
<td>90</td>
<td>0.50</td>
</tr>
<tr>
<td>80</td>
<td>0.25</td>
</tr>
</tbody>
</table>

TABLE 7
EFFICIENCIES OF DIVERSION/PERCOALITION
(No First Flush Phenomena)

<table>
<thead>
<tr>
<th>% Efficiencies Primary System</th>
<th>O=0</th>
<th>O=0.1</th>
<th>O=0.2</th>
<th>O=0.4</th>
<th>O=0.8</th>
<th>O=1.0</th>
<th>Diversion Volume, in.</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>97</td>
<td>96</td>
<td>95</td>
<td>90</td>
<td>90</td>
<td>82</td>
<td>1.25</td>
</tr>
<tr>
<td>96</td>
<td>95</td>
<td>93</td>
<td>82</td>
<td>72</td>
<td>72</td>
<td>60</td>
<td>1.00</td>
</tr>
<tr>
<td>95</td>
<td>93</td>
<td>90</td>
<td>72</td>
<td>60</td>
<td>60</td>
<td>40</td>
<td>0.75</td>
</tr>
<tr>
<td>93</td>
<td>90</td>
<td>82</td>
<td>60</td>
<td>40</td>
<td>40</td>
<td>20</td>
<td>0.50</td>
</tr>
<tr>
<td>90</td>
<td>82</td>
<td>60</td>
<td>40</td>
<td>20</td>
<td>20</td>
<td>0.25</td>
<td></td>
</tr>
</tbody>
</table>
cies are shown in Table 7.

The equations used to interpolate are:

\[
\text{FRACT} = \frac{1}{\text{right "C" value - read in "C" value}} \quad \text{(V-11)}
\]

\[
\text{FRACT} = \frac{\text{difference between left and right interpolated "C" values}}{\text{right "C" value - read in "C" value}}
\]

FRACT determines the "distance" to go between the left and right "C" values when given a read in "C" value.

\[
\text{FFF} = - \left[ \frac{\text{difference between right and left "C" value efficiencies}}{\text{FRACT}} \right] + \quad \text{(V-12)}
\]

\[
\text{FFF} = - \left[ \frac{\text{difference between right and left "C" value efficiencies}}{\text{FRACT}} \right] + \quad \text{left most "C" value efficiency}
\]

FFF computes the value of the efficiency at the "distance" between the left and right "C" values. The efficiencies are used in subroutine BASIN.

Cost equations are computed within the program and are added together to get a total cost for basins. The cost equations are as follows:

\[
\text{CCLD}(I) = \text{CCLAN} \times 1.2 \times \frac{\text{VC}(I)}{\text{D}} \quad \text{(V-13)}
\]

where:  \( \text{CCLD}(I) = \text{land cost - $} \)

\( \text{CCLAN} = \text{cost of land - $/acre} \)

\( 1.2 = 20\% \text{ safety margin} \)

\( \text{VC}(I) = \text{volume of basin - (ac - ft)} \)

\( \text{D} = \text{depth of basin - ft} \)

\[
\text{CCEX}(I) = \text{EXCOS} \times \frac{\text{VC}(I)}{27} \times \frac{43560.}{27} \quad \text{(V-14)}
\]

where:  \( \text{CCEX}(I) = \text{excavation cost - $} \)

\( \text{EXCOS} = \text{cost of excavation - $/yd}^3 \)
VC(I) = volume of basin - (ac - ft)
43560. = conversion factor, acres to ft²
27. = conversion factor, ft³ to yd³

CCCC(I) = COCOST * 1.2 * VC(I) * 4840./D  (V-15)

where: CCCC(I) = cover crop cost - $
COCOST = cost of putting down a cover crop - $/yd²
1.2 = 20% safety margin
VC(I) = volume of basin - (ac - ft)
4840. = conversion factor, ft² to yd²
D = depth of basin - ft

CCST(I) = (CCLD(I) + CCEX(I) + CCCC(I) + COIOUT) * 1.15  (V-16)

where: CCST(I) = total cost - $
CCLD(I) = land cost - $
CCEX(I) = excavation cost - $
CCC(I) = cover crop cost - $
COIOUT = inlet/outlet cost - $
1.15 = 15% engineering fees

TCCOST(I) = CCST(I) * VC(I)  (V-17)

where: TCCOST(I) = total cost - $
CCST(I) = total cost - $
VC(I) = volume of basin - (ac - ft)

With the cost of percolation basin and known efficiency, a plot of cost versus efficiency is graphed. There are five points on the graph, one for each diversion volume, and four graphs, one for each pollutant. Subroutine SLOPE is called to compute the slope
and intercept data for each graph.

**Subroutine PLOT**

This subroutine uses the data from the subroutines CATCH, SWEEP, and BASIN and plots cost/efficiency curves of present value dollars (y axis) versus efficiency or pounds removed per year (x axis). If variable NL = 0, 50 subdivisions of present value dollars are used. If anything other than 0 is used, that number of subdivisions will appear on the y axis. Appendix A will allow the reader to review the mechanics of subroutine PLOT.

**Subroutine SLOPE**

Subroutine SLOPE uses the equations:

\[
SL = \frac{\Delta y}{\Delta x} \quad \text{and,} \quad (V-18)
\]

\[
B = Y - MX \quad (V-19)
\]

for slope and intercept calculations, respectively. These are later used in the linear programming optimization subroutine LPGOGO.

**Subroutine TABLO**

These subroutines arrange the computed data into a useful form for subroutine LPGOGO.

First, constraint names are required along with the right hand sides (RHS) values of these constraint equations. An array is set up for each requirement.

Next, variable names and objective function values are required. Objective function values are merely the slopes as calcu-
lated in subroutine SLOPE. Arrays are set up for these requirements. All data is then passed into subroutine LPGOGO for an optimization analysis; that is, the best solution at minimum and maximum pollutant removal. Subroutine LPGOGO and subroutines TABLO can be seen in Appendix A.

Subroutine LPGOGO

Subroutine LPGOGO is taken from the pages of Daellenbach and Bell's text on linear programming and has been modified to run in IBM 370 Fortran IV (1970). The significant code changes were to provide double precision storage for the 8 character variable and constraint names.

LPGOGO is a maximizing linear programming code. It uses the two phase, full tableau form of the simplex method, requires all RHS parameters to be nonnegative, and starts from a fully artificial basis. It assumes that all constraints have been converted to equations. To minimize a problem in LPGOGO, all objective function values must be multiplied by -1.

The objective function and linear programming phase one coefficients are stored as the \((M + 1)\)st and \((M + 2)\)nd rows of the array which also stores the inverse of the basis in its last \(m\) columns.

Lake Eola Linear Programming Formulation

The data computed in subroutines PLOT and SLOPE are used in subroutine LPGOGO. This can be seen in the following example.

If the three management practices of catchbasin cleaning,
street sweeping, and diversion/percolation are selected for analysis in the linear programming optimization subroutine of LPCOGO, and the pollutant selected is nitrogen, the following formulation is set up.

From Figure 13, the catchbasin cleaning cost efficiency curve for nitrogen, slopes of the linear segments from the origin to the fifth point are computed. These slopes (SL) are the marginal costs of the segments of the curve. Also computed are the distances on the x axis between the five points (BK). The same is done for the cost efficiency curves of nitrogen for street sweeping (Figure 14) and diversion/percolation (Figure 15). The linear programming formulation now takes the form:

Minimize: (objective function)

\[ Z = SL(11)X_{11} + SL(12)X_{12} + SL(13)X_{13} + SL(14)X_{14} + SL(15)X_{15} \]
\[ + SL(21)X_{21} + SL(22)X_{22} + SL(23)X_{23} + SL(24)X_{24} + SL(25)X_{25} \]
\[ + SL(31)X_{31} + SL(32)X_{32} + SL(33)X_{33} + SL(34)X_{34} + SL(35)X_{35} \]

Subject To: (computer variable name = SEGSUM)

\[ X_{11} + X_{12} + X_{13} + X_{14} + X_{15} + X_1 + X_{22} + X_{23} + X_{24} + X_{25} \]
\[ + X_{31} + X_{32} + X_{33} + X_{34} + X_{35} \geq \text{PYRRN} \]

(computer variable names = SGIPRI to SG5PR3)

\[ X_{11} \leq BK(11) - BK(\text{origin}) \]
\[ X_{12} \leq BK(12) - BK(11) \]
\[ X_{13} \leq BK(13) - BK(12) \]
\[ X_{14} \leq BK(14) - BK(13) \]
Fig. 13. Graph #1 - catchbasin cleaning (nitrogen graph)
Fig. 14. Graph #2 - street sweeping (nitrogen graph)
Fig. 15. Graph #3 - diversion/percolation (nitrogen graph)
\[ X_{15} \leq BK(15) - BK(14) \]
\[ X_{21} \leq BK(21) - BK(20) \]
\[ X_{22} \leq BK(22) - BK(21) \]
\[ X_{23} \leq BK(23) - BK(22) \]
\[ X_{24} \leq BK(24) - BK(23) \]
\[ X_{25} \leq BK(25) - BK(24) \]
\[ X_{31} \leq BK(31) - BK(30) \]
\[ X_{32} \leq BK(32) - BK(31) \]
\[ X_{33} \leq BK(33) - BK(32) \]
\[ X_{34} \leq BK(34) - BK(33) \]
\[ X_{35} \leq BK(35) - BK(34) \]

\[ X_{11}, X_{12}, X_{13}, X_{14}, X_{15}, X_{21}, X_{22}, X_{23}, X_{24}, X_{25}, X_{31}, X_{32}, X_{33}, X_{34}, X_{35} \geq 0 \]

where:
\[ X_{11} = \text{point one on graph one (catchbasin cleaning)} \]
\[ X_{21} = \text{point one on graph two (street cleaning)} \]
\[ X_{31} = \text{point one on graph three (diversion/percolation)} \]

\[ \text{PYRRN} = \text{pounds of pollutant nitrogen which have to be removed from the subwatershed (lb/yr)} \]

Setting up phase one of the linear programming problem involves converting non-equalities to equalities. Also, since LPGOGO is a maximizing program, the objective function variables must be multiplied by -1. Thus:

Maximize:
\[ Z = -SL(11)X_{11} - SL(12)X_{12} - SL(13)X_{13} - SL(14)X_{14} - SL(15)X_{15} \]
\[ - SL(21)X_{21} - SL(22)X_{22} - SL(23)X_{23} - SL(24)X_{24} \]
- SL(25)X_{25} - SL(31)X_{31} - SL(32)X_{32} - SL(33)X_{33} - SL(34)X_{34} - SL(35)X_{35}

Subject To:

\[ x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{31} + x_{32} + x_{33} + x_{34} + x_{35} - \text{SURPLUS 1} = \text{PYRRN} \]

\[ x_{11} + \text{SLACK 1} = \text{BK}(11) - \text{BK}(\text{origin}) \]

\[ x_{12} + \text{SLACK 2} = \text{BK}(12) - \text{BK}(11) \]

\[ x_{13} + \text{SLACK 3} = \text{BK}(13) - \text{BK}(12) \]

\[ x_{14} + \text{SLACK 4} = \text{BK}(14) - \text{BK}(13) \]

\[ x_{15} + \text{SLACK 5} = \text{BK}(15) - \text{BK}(14) \]

\[ x_{21} + \text{SLACK 6} = \text{BK}(21) - \text{BK}(\text{origin}) \]

\[ x_{22} + \text{SLACK 7} = \text{BK}(22) - \text{BK}(21) \]

\[ x_{23} + \text{SLACK 8} = \text{BK}(23) - \text{BK}(22) \]

\[ x_{24} + \text{SLACK 9} = \text{BK}(24) - \text{BK}(23) \]

\[ x_{25} + \text{SLACK 10} = \text{BK}(25) - \text{BK}(24) \]

\[ x_{31} + \text{SLACK 11} = \text{BK}(31) - \text{BK}(\text{origin}) \]

\[ x_{32} + \text{SLACK 12} = \text{BK}(32) - \text{BK}(31) \]

\[ x_{33} + \text{SLACK 13} = \text{BK}(33) - \text{BK}(32) \]

\[ x_{34} + \text{SLACK 14} = \text{BK}(34) - \text{BK}(33) \]

\[ x_{35} + \text{SLACK 15} = \text{BK}(35) - \text{BK}(34) \]

\[ x_{ij} \geq 0 \]

This formulates the piece-wise linear approximation problem which is utilized by the computer program to optimize stormwater management practices cost effectively.
LPGOGO Output Interpretation

Consider the numbers listed under the heading DELTAJ. These numbers are called the reduced objective function coefficients. For nonbasic variables they are nonpositive, with most of them usually negative. If the DELTAJ is equal to zero for a nonbasic variable, then this variable could be assigned an arbitrary positive value within a certain range, with corresponding changes in the present basic variables without affecting the value of the objective function. Since the latter is at its optimum, this means letting the variable in question assume positive values and we obtain alternate optimal solutions.

Consider now a non-basic variable with a negative DELTAJ. If the original objective function coefficient of this nonbasic variable would be larger by an amount equal to the negative of the corresponding DELTAJ, then this variable would become a candidate to assume positive values in alternate optimal solutions.

Consider the part of the computer solution that deals with the constraints. The numbers listed under the heading value gives the change in the optimal value of the objective function for a unit increase in the value of the RHS parameter of the corresponding constraint, all other inputs remaining unchanged. For a constraint that has slack in the optimal solution, this number is zero. These numbers are called imputed values. The numbers listed under decrease and increase indicate by how much the original RHS parameter for the constraint in question may be decreased or increased, respectively,
without affecting the imputed value for that constraint, all other inputs remaining unchanged.

**Subroutine TABLOX**

Subroutine TABLOX is exactly the same as the other TABLO subroutines described previously. However, this differs from the others in that it sets up the data, into proper form for LPGOGX, from each subwatershed instead of from each management practice. This allows an optimization analysis of a series of subwatersheds, in order to find the cost-effective solution in an entire watershed. In essence, it breaks up a large problem into "n" smaller problems, solves them, and then uses this data to solve once more. The limit at this time for subwatersheds is 20. This can be expanded later by changing the dimension statements and minor other statements, yet it must be understood that major round-off errors might become commonplace.

**Subroutine LPGOGX**

This is exactly the LPGOGO subroutine described previously, but now optimizes "n" subwatersheds instead of "n" management practices.
CHAPTER VI

LAKE EOLA DATA FOR COMPUTER PROGRAM "MANAGE"

Through studies preceding and continuing through this research, data was compiled on the subwatersheds of the Lake Eola watershed. The data were accumulated through research at the site, laboratory analyzes, and planimeter studies. Data are seen below and in Table 8.

Loading Rates

The loading rates of the four pollutants are as follows:
1. suspended solids = 1195 lbs/ac-yr
2. BOD = 175 lbs/ac-yr
3. nitrogen = 60 lbs/ac-yr
4. phosphorous = 4.2 lbs/ac-yr

Soil Type

The soil types in the Lake Eola watershed vary between SCS soil classification types "A" and "D". These indicate an ability of the soils to allow percolation and infiltration.

Additional land use data are found in Table 8.
### Table 8

**Land Use Data for Lake Eola**

<table>
<thead>
<tr>
<th>Subwater Shed #</th>
<th>Contrib Area (Acres)</th>
<th>Land Use**</th>
<th>% Impervious W.O. Street</th>
<th>% Streets Lots, etc</th>
<th>% Pervious</th>
<th>Composite Curve # Including Streets, Lots</th>
<th>Max Length of Drainage ft</th>
<th>tc* (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>29.56</td>
<td>Comm</td>
<td>69</td>
<td>29</td>
<td>2</td>
<td>75.0</td>
<td>1700</td>
<td>14.17</td>
</tr>
<tr>
<td>2</td>
<td>22.81</td>
<td>Comm</td>
<td>70</td>
<td>26</td>
<td>4</td>
<td>75.0</td>
<td>1340</td>
<td>11.17</td>
</tr>
<tr>
<td>3</td>
<td>14.18</td>
<td>Resid</td>
<td>38</td>
<td>25</td>
<td>37</td>
<td>75.0</td>
<td>1560</td>
<td>13.0</td>
</tr>
<tr>
<td>4</td>
<td>0.27</td>
<td>Resid</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>98.0</td>
<td>100</td>
<td>5.0</td>
</tr>
<tr>
<td>5</td>
<td>0.20</td>
<td>Resid</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>98.0</td>
<td>80</td>
<td>5.0</td>
</tr>
<tr>
<td>6</td>
<td>6.21</td>
<td>Resid</td>
<td>38</td>
<td>34</td>
<td>28</td>
<td>75.0</td>
<td>600</td>
<td>5.0</td>
</tr>
<tr>
<td>7</td>
<td>25.02</td>
<td>Resid</td>
<td>37</td>
<td>23</td>
<td>40</td>
<td>75.0</td>
<td>1700</td>
<td>14.17</td>
</tr>
<tr>
<td>8</td>
<td>5.95</td>
<td>Resid</td>
<td>48</td>
<td>24</td>
<td>28</td>
<td>75.0</td>
<td>600</td>
<td>5.0</td>
</tr>
<tr>
<td>9</td>
<td>6.13</td>
<td>Resid</td>
<td>49</td>
<td>38</td>
<td>13</td>
<td>75.0</td>
<td>1480</td>
<td>12.33</td>
</tr>
<tr>
<td>10</td>
<td>25.79</td>
<td>Comm</td>
<td>66</td>
<td>29</td>
<td>5</td>
<td>75.0</td>
<td>1600</td>
<td>13.33</td>
</tr>
</tbody>
</table>

**Total**

- W.O./Lake 136.12

**Total**

- W/Lake 164.87 not including park land

*Assuming 2 fps average velocity

**Comm = commercial use
Resid = residential use
CHAPTER VII

EXAMPLE PROBLEM

The following example problem is given to aid the reader in understanding the computer program. There are 3 subwatersheds in the watershed.

Data For Subwatershed #1

Practices chosen as possible for subwatershed one are Catchbasin Cleaning, Street Sweeping, and Diversion/Percolation. The pollutant selected for the optimization analysis is nitrogen. Subwatershed one has an area of 57.00 acres and loading rate of:

- Suspended solids: 1195.0 lb/ac-yr
- BOD: 175.0 lb/ac-yr
- Nitrogen: 60.0 lb/ac-yr
- Phosphorus: 4.2 lb/ac-yr

Catchbasin

The number of stormwater inlets found in this subwatershed is 72. The capital cost of converting an inlet to a catchbasin is $100. The ORM cost of cleaning a catchbasin is $150/catchbasin.

Street Sweeping

The population density in this subwatershed is 20 persons/acre. The ORM and capital cost of sweepers are $7.00/curb mile
swept and $44000, respectively.

**Diversion/Percolation**

The soil type found in this subwatershed is type "D". The curve number was found to be 85. Cost of land is $2000/acre, with excavation cost being $1.50/cubic yard. Cover crop cost is 12¢/square yard. The inlet/outlet cost is $500. A runoff coefficient "C" was found to be 0.05.

From this and other read in data, the program generated the following output:

1. Figure 16 - Catchbasin Technology Statistics (as indicated in section "Data For Subwatershed #1") including removal (efficiency) and cost results

2. Figure 17 - Nitrogen Cost/Efficiency Curve (Note: also generated were cost/efficiency curves for suspended solids, BOD₅, and phosphorus; not shown)

3. Figure 18 - Street Sweeping Technology Statistics (as indicated in section "Data For Subwatershed #1") including removal (efficiency) and cost results. Note: all graphs and slope & intercept data is generated for street sweeping, but not shown

4. Figure 19 - Diversion/Percolation Statistics (as indicated in section "Data For Subwatershed #1") including removal (efficiency) and cost results. Again, all graphs and slope & intercept data are generated for diversion/percolation, but not shown
5. Figures 20 and 21 - the linear programming solution to the subwatershed #1 problem at a given efficiency of removal. This is done for 5 efficiencies of removal; not all shown. Detailed accounting of the output is shown in previous chapters.

6. Figure 22 - Subwatershed #1 Summary Data. This shows the pollutant chosen for analysis, the management practices chosen for analysis, and pertinent subwatershed data. Also produced, and most importantly, are the efficiencies of removal (in fraction form), the actual removals in pounds of the pollutant, the cost to remove this mass of pollutant (as determined by the linear programming solution), and the practices selected to remove this mass cost-effectively. The breakdown per practice of pollutant mass in pounds per year is also produced. This is the complete analysis of subwatershed #1.

Similar results described above are produced for the other subwatersheds depending on imputed data and selected practices. These will not be shown. Instead, an actual example utilizing all given figures, including Figures 23-25, the total watershed cost-efficient solution, will be demonstrated. This, in actuality is an explanation of the subwatershed and total watershed summary pages.

**Sizing and Costing of BMP's**

From Figures 16-25 one can determine what combination of practices is optimal. Sizing and costing of these practices is also possible.

From Figure 25 it is seen that 9653.76 pounds of pollutant
CATCH BASIN CLEANING DATA

AREA OF THE WATERSHED = 5,740.00 ACRES

INM COST OF CLEANING CATCH BASIN = 150.00 /CATCH BASIN

NUMBER OF CATCHBASINS = 12

CAPITAL COST OF CONVERSION = 100,000

SULFUR LOADING RATE = 119.50 LB/AC-yr

NITROGEN LOADING RATE = 7.50 LB/AC-yr

PHOSPHORUS LOADING RATE = 4.00 LB/AC-yr

<table>
<thead>
<tr>
<th>FREQUENCY</th>
<th>SULFUR</th>
<th>NITROGEN</th>
<th>PHOSPHORUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>1350.53</td>
<td>685.27</td>
<td>342.01</td>
</tr>
<tr>
<td>1.0</td>
<td>2603.06</td>
<td>1360.52</td>
<td>689.05</td>
</tr>
<tr>
<td>2.0</td>
<td>5160.35</td>
<td>2653.14</td>
<td>1311.07</td>
</tr>
<tr>
<td>3.0</td>
<td>7817.29</td>
<td>2992.50</td>
<td>1660.19</td>
</tr>
<tr>
<td>4.0</td>
<td>5508.54</td>
<td>3191.49</td>
<td>1128.59</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FREQUENCY</th>
<th>OWN</th>
<th>CAPITAL</th>
<th>PRESENT VALUE (20 yr, 7%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>3000.00</td>
<td>7200.00</td>
<td>64467.52</td>
</tr>
<tr>
<td>1.0</td>
<td>1800.00</td>
<td>4200.00</td>
<td>121615.10</td>
</tr>
<tr>
<td>2.0</td>
<td>2100.00</td>
<td>4200.00</td>
<td>230630.20</td>
</tr>
<tr>
<td>3.0</td>
<td>3200.00</td>
<td>6400.00</td>
<td>350445.40</td>
</tr>
<tr>
<td>4.0</td>
<td>4300.00</td>
<td>1200.00</td>
<td>648650.50</td>
</tr>
</tbody>
</table>

The frequency of catchbasin cleaning is a function of several local parameters, such as soil leachability, quantity of accumulated street sulfates, antecedent dry period, meteorological conditions, street cleaning methods and practices, surrounding land use, topography, and to some extent, the type of soil adjacent to the street:

- FREQUENCY 0.5 = ONCE IN 2 YEARS
- FREQUENCY 1.0 = ONCE A YEAR
- FREQUENCY 2.0 = 2 TIMES PER YEAR
- FREQUENCY 3.0 = 3 TIMES PER YEAR
- FREQUENCY 4.0 = 4 TIMES PER YEAR

Fig. 16. Catchbasin cleaning data output
**P.V. LOT 7**

<table>
<thead>
<tr>
<th>Pounds Removed</th>
<th>Nitrogen Chart</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>242.41</td>
<td>31.54</td>
<td>419.94</td>
</tr>
</tbody>
</table>

**Pounds per Year Removed**

Fig. 17. Nitrogen convex cost/efficiency curve
**CONTRAINT NAME** | **"RHS" VALUE**
---|---
S61FR1 | 0.000000E+00
S62FR1 | 0.000000E+00
S63FR1 | 0.000000E+00
S64FR1 | 0.000000E+00
S65FR1 | 0.000000E+00
S61FR2 | 0.000000E+00
S62FR2 | 0.000000E+00
S63FR2 | 0.000000E+00
S64FR2 | 0.000000E+00
S65FR2 | 0.000000E+00
S61FR3 | 0.000000E+00
S62FR3 | 0.000000E+00
S63FR3 | 0.000000E+00
S64FR3 | 0.000000E+00
S65FR3 | 0.000000E+00
SEG001 | 0.000000E+00

**VARIABLE NAME** | **OBJECTIVE FUNCTION VALUE**
---|---
X11 | 0.000000E+00
X12 | 0.000000E+00
X13 | 0.000000E+00
X14 | 0.000000E+00
X15 | 0.000000E+00
X21 | 0.000000E+00
X22 | 0.000000E+00
X23 | 0.000000E+00
X24 | 0.000000E+00
X25 | 0.000000E+00
X31 | 0.000000E+00
X32 | 0.000000E+00
X33 | 0.000000E+00
X34 | 0.000000E+00
X35 | 0.000000E+00
SURFL1 | 0.000000E+00
SLK1 | 0.000000E+00
SLK2 | 0.000000E+00
SLK3 | 0.000000E+00
SLK4 | 0.000000E+00
SLK5 | 0.000000E+00
SLK6 | 0.000000E+00
SLK7 | 0.000000E+00
SLK8 | 0.000000E+00
SLK9 | 0.000000E+00
SLK10 | 0.000000E+00
SLK11 | 0.000000E+00
SLK12 | 0.000000E+00
SLK13 | 0.000000E+00
SLK14 | 0.000000E+00
SLK15 | 0.000000E+00

Fig. 18. Initial tableau for subwatershed #1
**LINEAR PROGRAMMING SOLUTION**

**EFFICIENCY OF REMOVAL (FRACTION) = 0.40**

**POLUTANT IS NITROGEN**

**NUMBER EQUATIONS = 10**
**NUMBER VARIABLES = 31**

**SOLUTION, OPTIMAL AFTER 30 ITERATIONS**

**MAXIMAL OBJECTIVE = -0.113436E 00**

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>STATUS</th>
<th>VALUE</th>
<th>DELTA</th>
<th>DECREASE</th>
<th>INCREASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A11</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A12</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A13</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A14</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A15</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A16</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A17</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A18</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A19</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A20</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A21</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A22</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A23</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A24</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A25</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A26</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A27</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A28</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A29</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A30</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A31</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A32</td>
<td>cacl</td>
<td>0.00000VE</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CONSTRAINT STATUS**

<table>
<thead>
<tr>
<th>CONSTRAINT</th>
<th>STATUS</th>
<th>VALUE</th>
<th>DECREASE</th>
<th>INCREASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>slaca</td>
<td>0.00000VE</td>
<td>0.00000VE 0.242642E 03</td>
<td>OPEN</td>
</tr>
<tr>
<td>C2</td>
<td>slaca</td>
<td>0.00000VE</td>
<td>0.00000VE 0.242642E 03</td>
<td>OPEN</td>
</tr>
<tr>
<td>C3</td>
<td>slaca</td>
<td>0.00000VE</td>
<td>0.00000VE 0.242642E 03</td>
<td>OPEN</td>
</tr>
<tr>
<td>C4</td>
<td>slaca</td>
<td>0.00000VE</td>
<td>0.00000VE 0.242642E 03</td>
<td>OPEN</td>
</tr>
<tr>
<td>C5</td>
<td>slaca</td>
<td>0.00000VE</td>
<td>0.00000VE 0.242642E 03</td>
<td>OPEN</td>
</tr>
<tr>
<td>C6</td>
<td>slaca</td>
<td>0.00000VE</td>
<td>0.00000VE 0.242642E 03</td>
<td>OPEN</td>
</tr>
<tr>
<td>C7</td>
<td>slaca</td>
<td>0.00000VE</td>
<td>0.00000VE 0.242642E 03</td>
<td>OPEN</td>
</tr>
<tr>
<td>C8</td>
<td>slaca</td>
<td>0.00000VE</td>
<td>0.00000VE 0.242642E 03</td>
<td>OPEN</td>
</tr>
<tr>
<td>C9</td>
<td>slaca</td>
<td>0.00000VE</td>
<td>0.00000VE 0.242642E 03</td>
<td>OPEN</td>
</tr>
<tr>
<td>C10</td>
<td>slaca</td>
<td>0.00000VE</td>
<td>0.00000VE 0.242642E 03</td>
<td>OPEN</td>
</tr>
</tbody>
</table>

**Fig. 19. Linear programming (LP/GOGO) solution output for subwatershed #1**
**Street Sweeping Data**

- Population density: 2400 persons/acre
- Daily cost per linear mile: $3.75
- Capital cost for truck: $24,000
- Capital cost of sweepers: $4000
- Annual maintenance cost: $110,000
- Annual labor cost: $115,000
- Capital cost of loaders: $20,000
- Phosphorus loading rate: 1.60 lbs/acre
- Nitrates loading rate: 1.24 lbs/acre

---

**Pounds per Year Removed**

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Sodium</th>
<th>Phosphorus</th>
<th>Nitrogen</th>
<th>Phosphorus</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.143</td>
<td>1557.01</td>
<td>209.02</td>
<td>10.75</td>
<td>0.2517</td>
</tr>
<tr>
<td>0.200</td>
<td>1057.22</td>
<td>294.58</td>
<td>15.12</td>
<td>0.3592</td>
</tr>
<tr>
<td>0.330</td>
<td>5101.19</td>
<td>425.07</td>
<td>21.84</td>
<td>0.5108</td>
</tr>
<tr>
<td>0.500</td>
<td>5669.84</td>
<td>494.09</td>
<td>25.81</td>
<td>0.5929</td>
</tr>
<tr>
<td>1.000</td>
<td>4515.95</td>
<td>6000.15</td>
<td>31.24</td>
<td>0.7297</td>
</tr>
</tbody>
</table>

---

**Cost of Street Sweeping per Year**

<table>
<thead>
<tr>
<th>Frequency</th>
<th>0.143</th>
<th>0.200</th>
<th>0.330</th>
<th>0.500</th>
<th>1.000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capital</td>
<td>1051.16</td>
<td>1472.10</td>
<td>2472.76</td>
<td>3072.40</td>
<td>7350.82</td>
</tr>
<tr>
<td>Present Value (20 yr, 7%)</td>
<td>17864.06</td>
<td>25948.70</td>
<td>41224.76</td>
<td>62481.76</td>
<td>129923.50</td>
</tr>
</tbody>
</table>

---

**Figure 20. Street Sweeping Data Output**
### Diversion/Percolation Data

<table>
<thead>
<tr>
<th>Diversion Volume (luc)</th>
<th>Water</th>
<th>Flushing</th>
<th>Phosphorus</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>2442.470</td>
<td>4942.470</td>
<td>1072.990</td>
</tr>
<tr>
<td>0.50</td>
<td>4884.940</td>
<td>5942.470</td>
<td>1710.450</td>
</tr>
<tr>
<td>0.75</td>
<td>7327.410</td>
<td>6942.470</td>
<td>2257.910</td>
</tr>
<tr>
<td>1.00</td>
<td>9770.380</td>
<td>7170.380</td>
<td>2715.370</td>
</tr>
<tr>
<td>1.25</td>
<td>12213.350</td>
<td>7412.700</td>
<td>3172.830</td>
</tr>
</tbody>
</table>

### Volume of Lucas with Variable Depth (luc) in Composite Luo using VAO (VAD)

<table>
<thead>
<tr>
<th>Diversion Volume (luc)</th>
<th>VAO (2x+1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>2.512500</td>
</tr>
<tr>
<td>0.50</td>
<td>4.712500</td>
</tr>
<tr>
<td>0.75</td>
<td>6.912500</td>
</tr>
<tr>
<td>1.00</td>
<td>9.112500</td>
</tr>
<tr>
<td>1.25</td>
<td>11.312500</td>
</tr>
</tbody>
</table>

### Total Cost of Danny

<table>
<thead>
<tr>
<th>Diversion Volume (luc)</th>
<th>Capital</th>
<th>Run</th>
<th>Present Value (2DA, 7%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>40133.4400</td>
<td>107.0400</td>
<td>63342.3500</td>
</tr>
<tr>
<td>0.50</td>
<td>80266.8800</td>
<td>171.0800</td>
<td>116680.6900</td>
</tr>
<tr>
<td>0.75</td>
<td>120400.3200</td>
<td>225.1200</td>
<td>169309.6400</td>
</tr>
<tr>
<td>1.00</td>
<td>160532.7600</td>
<td>279.1600</td>
<td>221956.0100</td>
</tr>
<tr>
<td>1.25</td>
<td>200665.2000</td>
<td>333.2000</td>
<td>274603.5600</td>
</tr>
</tbody>
</table>

Fig. 21. Diversion/percolation data output
The pollutant selected for the optimization analysis is nitrogen.

Note: If any of the three selector equals 1, that practice is possible (analyzed in the linear programming routine).

Catchbasin_1 selector ISLC = 1
Infiltration_1 selector ISLC = 1
Diversions_percolation_1 selector ISLC = 1

The area of this subwatershed is = 57.00 acres
The subwatershed loading rate of this subwatershed is = 114.00 lb/ac-yr
The nonloading rate of this subwatershed is = 17.00 lb/ac-yr
The nitrogen loading rate of this subwatershed is = 0.00 lb/ac-yr
The phosphorus loading rate of this subwatershed is = 0.00 lb/ac-yr

Efficiencies (Fraction) Removals (lb/yr) Cost ($) Selected Practice(s) lb/yr/practice removal
---
0.40 1368.00 11546.900000 * Diversions/Percolation * 1368.00
0.65 2225.00 504981.600000 * Catchbasin Technology * * Diversions/Percolation * 513.00 1710.00
0.40 2736.00 456688.400000 * Catchbasin Technology * * Diversions/Percolation * 937.08 1798.92
0.75 5018.00 555617.200000 * Catchbasin Technology * * Diversions/Percolation * 937.08 2140.92
0.75 5749.00 623012.500000 * Catchbasin Technology * * Diversions/Percolation * 937.08 2311.92

Fig. 22. Subwatershed summary data output
<table>
<thead>
<tr>
<th>CONSTRAINT NAME</th>
<th>&quot;RHS&quot; VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>S61A11</td>
<td>0.130000E+04</td>
</tr>
<tr>
<td>S62A11</td>
<td>0.650000E+03</td>
</tr>
<tr>
<td>S63A11</td>
<td>0.310000E+03</td>
</tr>
<tr>
<td>S64A11</td>
<td>0.541000E+03</td>
</tr>
<tr>
<td>S65A11</td>
<td>0.170000E+03</td>
</tr>
<tr>
<td>S61A12</td>
<td>0.194000E+04</td>
</tr>
<tr>
<td>S62A12</td>
<td>0.106000E+04</td>
</tr>
<tr>
<td>S63A12</td>
<td>0.712000E+03</td>
</tr>
<tr>
<td>S64A12</td>
<td>0.474100E+03</td>
</tr>
<tr>
<td>S65A12</td>
<td>0.233300E+03</td>
</tr>
<tr>
<td>S61A13</td>
<td>0.150000E+03</td>
</tr>
<tr>
<td>S62A13</td>
<td>0.975000E+03</td>
</tr>
<tr>
<td>S63A13</td>
<td>0.585000E+03</td>
</tr>
<tr>
<td>S64A13</td>
<td>0.299999E+03</td>
</tr>
<tr>
<td>S65A13</td>
<td>0.195000E+03</td>
</tr>
<tr>
<td>SUR1U</td>
<td>0.963376E+04</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>OBJECTIVE FUNCTION VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>X11</td>
<td>-0.044043E+07</td>
</tr>
<tr>
<td>X12</td>
<td>-0.231579E+03</td>
</tr>
<tr>
<td>X13</td>
<td>-0.270123E+03</td>
</tr>
<tr>
<td>X14</td>
<td>-0.310623E+03</td>
</tr>
<tr>
<td>X15</td>
<td>-0.394124E+03</td>
</tr>
<tr>
<td>X21</td>
<td>-0.526673E+05</td>
</tr>
<tr>
<td>X22</td>
<td>-0.100000E+09</td>
</tr>
<tr>
<td>X23</td>
<td>-0.100000E+09</td>
</tr>
<tr>
<td>X24</td>
<td>-0.100000E+09</td>
</tr>
<tr>
<td>X25</td>
<td>-0.100000E+09</td>
</tr>
<tr>
<td>X31</td>
<td>-0.641025E+05</td>
</tr>
<tr>
<td>X32</td>
<td>-0.100000E+09</td>
</tr>
<tr>
<td>X33</td>
<td>-0.100000E+09</td>
</tr>
<tr>
<td>X34</td>
<td>-0.100000E+09</td>
</tr>
<tr>
<td>X35</td>
<td>-0.100000E+09</td>
</tr>
<tr>
<td>SLK1</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>SLK2</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>SLK3</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>SLK4</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>SLK5</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>SLK6</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>SLK7</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>SLK8</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>SLK9</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>SLK10</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>SLK11</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>SLK12</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>SLK13</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>SLK14</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>SLK15</td>
<td>0.000000E+00</td>
</tr>
</tbody>
</table>

Fig. 23. Initial tableau for all subwatersheds (total watershed).
Fig. 24. Linear programming (LPCGDX) solution output for watersheds.
TOTAL WATERSHED SUMMARY DATA

THE POLLUTANT SELECTED FOR THE OPTIMIZATION ANALYSIS IS NITROGEN

OUTPUT INTERPRETATION FOR WATERSHED

<table>
<thead>
<tr>
<th>MAXIMUM</th>
<th>SELECTED</th>
<th>LBS/yr</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBS REMOVED</td>
<td>COST ($)</td>
<td>SUBWATERSHEDS</td>
</tr>
<tr>
<td>9653.758</td>
<td>1816066.0</td>
<td># 1 3249.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td># 2 3869.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td># 3 2535.0</td>
</tr>
</tbody>
</table>

HEADING "MAXIMUM LBS REMOVED" GIVES THE PRODUCT OF WASHOFF YEARLY LOADINGS TIMES REMOVAL AS A FRACTION (XMAX). HEADING "COST ($)" GIVES THE PRESENT VALUE COST FOR THE TOTAL WATERSHED. HEADING "SELECTED SUBWATERSHEDS" SHOWS IN WHICH OF THE SUBWATERSHEDS BEST MANAGEMENT PRACTICES ARE CHosen. THE ASSOCIATED REMOVALS PER SUBWATERSHED ARE GIVEN IN THE COLUMN HEADED "LBS/yr". THE SIZE OF PRACTICE IS THEN DETERMINED FOR EACH SUBWATERSHED BY USING THE LBS/yr COLUMN, AND RETURNING TO EACH SUBWATERSHED SUMMARY PAGE.

Fig. 25. Watershed summary data output
nitrogen is removed through the management practices in the watershed. The total present value cost of this removal for all subwatersheds is 1,816,066.0 dollars. This is the cost of the optimal solution which is the capital and operating cost over 20 years.

In the column labeled "selected subwatersheds", removal quantities for each subwatershed in the entire watershed can be determined. This value is in the "lbs/yr" column. Thus, it is seen that 3249.0 pounds of pollutant nitrogen is removed in subwatershed #1 by the management practices. Turning now to Figure 22, the subwatershed #1 summary data page, it is possible to determine what mix of practices produces the optimal cost-effective solution, and how much of the nitrogen mass is removed by each of these practices. Management practices Catchbasin Technology and Diversion/Percolation are selected for the 3249.0 pounds removal of nitrogen in subwatershed #1, with 937.08 pounds being removed by Catchbasin Technology and 2311.92 pounds being removed by Diversion/Percolation. This shows that these two practices were efficient enough to remove the desired nitrogen mass without the aid of street sweeping technology.

To determine what size and frequency of cleaning is required per practice selected, one must turn to the statistics of the practices selected, in this case, Catchbasin Cleaning Data (Figure 16), and Diversion/Percolation Data (Figure 19). From Figure 16, knowing that 937.08 pounds of nitrogen have been removed by Catchbasin Technology, one must go to the column "nitrogen" under heading "Pounds Per Year Removed" to determine the frequency of cleaning necessary to
cost-effectively remove the 937.08 pounds of the pollutant. At the value of 937.07980, one goes horizontally across to the left, to the "Frequency" column, to find the frequency of cleaning required. This value is 2.0. The last paragraph of Figure 16 states that a frequency of 2.0 means twice per year cleaning of the 72 catchbasins found in subwatershed #1.

From Figure 19, knowing that 2311.92 pounds of nitrogen have been removed by Diversion/Percolation, one must go to the column "nitrogen" under heading "Pounds Per Year Removed" to determine the size of basin required to cost-effectively remove the 2311.92 pounds of the pollutant. Interpolating between values 2257.199 and 2633.399 and proceeding horizontally (left) to column "Diversion Volume (in.)", it is found that a diversion volume of 0.76 inches must be diverted to percolation for desired removal. Going to the heading "Volume of Basin with Variable Depth (1<D<3) In Composite Land Use Areas (VAD)", under column "Diversion Volume (in.)", and interpolating between the 0.75 and 1.00 inch volumes of basins (column "VAD (Ac-ft)") it is found that a volume of 7.4 Ac-Ft is required for a 0.73 inch diversion volume, which is the size of basin required to cost-effectively remove the 2311.92 pounds of pollutant nitrogen from subwatershed #1.

Cost data is also available. From Figure 16, for a determined frequency of 2.0, ORM cost will be $21,600 per year; capital cost will be $7200 per year, and present value cost (20 year, 7% interest) will be $236,030.20 per year. Similarly, from Figure 19, at a diversion volume of 0.76 inches, capital cost will be $250,000 per year,
ORM cost will be $8850 per year, and present value cost (20 year, 7% interest) will be $345,000 per year, from interpolation.

The same procedure is followed for size, frequency, and cost data in subwatersheds #2 and #3.
CHAPTER VIII

CONCLUSIONS

The research culminating in the computer program "MANAGE" described in the body of this report reveals a methodology for the choice of the most optimal combination of stormwater management practices. However, more work is needed to estimate localized cost-efficiencies for stormwater management practices. Incorporating mathematical programming methods to this work saves time and money since it eliminates guesswork and allows one to select a combination of practices which will remove a maximum amount of pollutants at least cost.

The program, while very useful and badly needed can use some expanding. Additions to the program include the following:

1. Adding more than 3 practices to the program, thus allowing a better choice of combinations. This expansion will benefit the field of stormwater management by generating sorely needed cost efficiency data for many practices.

2. Increasing the number of subwatersheds allowed to be analyzed. This increase from the 20 now allowed will increase the efficiency of the program. By increasing the number of subwatersheds, smaller areas within a given watershed can be analyzed, thus allowing greater detail of the watershed.
3. Increasing the number of points on the cost-efficiency graphs. As stated in the body of this report, the linear programming approximation can be carried out to any degree of accuracy by dividing the curve into a suitable number of intervals for linear representation.

This program has been applied to the selection of management practices in a representative watershed, namely the Lake Eola watershed. It used site-specific data to determine the most cost-efficient combination of management practices. Further work will make this type program an integral part of stormwater management.
APPENDIX A

COMPUTER PROGRAM "MANAGE"
Appendix A contains the computer program "MANAGE". The entire program is shown, yet to conform to thesis publication format, it was printed out in lines of 60 characters. If a line exceeds 60 characters, it is truncated, indented, and continued on the next line. This allows the entire program to be shown with the necessary margins. Special thanks is extended to Mr. Tom Peeples of the UCF computer center for his "THESIS" program which made this printing format possible.
DATA SET IRMANAGE AT LEVEL 001 AS OF 05/22/79

THIS PROGRAM IS THE WORK OF MARK MICHAEL CALABRESE. IT IS FOR USE IN HIS OPTIMIZATION THESIS OF STORMWATER MANAGEMENT PRACTICES AND PROCESSES. THE PROGRAM WILL GENERATE CUST EFFICIENCY CURVES FOR THREE CHosen STORMWATER MANAGEMENT PRACTICES AND WILL OPTIMIZE THESE PRACTICES EFFICIENTLY IN ANY NUMBER OF WATERSHEDS.

MAIN SELECTOR ROUTINE

CARD GROUP # 1
VARIABLE DESCRIPTIONS

N = NUMBER OF ROWS IN MATRIX A, OR NUMBER OF POINTS FOR THE COST/EFFICIENCY GRAPHS (NOW EQUALS 5)
M = NUMBER OF COLUMNS IN MATRIX A, OR THE NUMBER OF AXES PLOTTED ON ONE GRAPH
NO = CHART NUMBER (3 DIGITS MAXIMUM)

ISLC = SUBROUTINE CATCH SELECTOR
ISLS = SUBROUTINE SWEEP SELECTOR
ISLB = SUBROUTINE BASIN SELECTOR

IF ISLC, ISLS, OR ISLB = 1, THEN SUBROUTINE IS CALLED
ISSS = PULUTANT SELECTOR FOR SUSPENDED SOLIUS
C ISBD = PULLUTANT SELECTOR FOR BOD-5
C ISNI = PULLUTANT SELECTOR FOR NITRUGEN
C ISP = PULLUTANT SELECTOR FOR PHOSPHORUS
C
C IF ANY OF THE FOUR SELECTORS IS 0, THAT PULLUTANT IS CHOOSEN.
C NOTE: ONLY ONE OF THE FOUR PULLUTANT SELECTORS
C ISSS, ISBD, ISNI, AND ISP CAN BE 0 AT A TIME
C
C NL = NUMBER OF LINES IN THE PLOT (Y AXIS SIZE)
C IF NL = 0, 50 LINES ARE USED
C AR = AREA OF SUBWATERSHED - ACRES
C SSRT = SUSPENDED SOLIDS LOADING RATE - LB/AC-YR
C BODRT = BOD LOADING RATE - LB/AC-YR
C
C TRI = NITRUGEN LOADING RATE - LB/AC-YR
C PRT = PHOSPHORUS LOADING RATE - LB/AC-YR
C
C XMAX = THE MAXIMUM PULLUTANT LOAD REMOVED BEFORE BEING DISCHARGED TO RECEIVING WATERS - FRACTION
C
C NSUL = SELECTOR DETERMINING WHETHER TO PRINT OUT TABLE-MATRIX
C IF NSUL = 0, NO MATRIX IS PRINTED
C
C NWAT = NUMBER OF SUBWATERSHEDS ANALYZED
C PER(1) = PERCENT REMOVALS FOR ALL SUBWATERSHEDS, A TOTAL OF FIVE TO BE READ, LOWEST VALUE READ FIRST.
C NRIG = GRAPH, SLOPE & INTERCEPT, AND LINEAR PROGRAM
C PRINT SELECTOR
C IF NBIG = 0, ONLY SUMMARY AND PRACTICE PRINTING DONE
C
C DIMENSION CQBYRP(6), SQBYRP(6), QUQBYRP(6), PV(6), PVS(6), PB(6), CUQBYRS
CALL CATCH(SSRT,BOURT,TRT,PRT,AR,N,M,NL,NO,CUBYRS,PV,C=UBYRB,CQBYKN)

CALL SWEET(SSRT,BOURT,TRT,PRT,AR,N,M,NL,NO,CUBYRS,PV,-

CALL BASIN(SSRT,BOURT,TRT,PRT,AR,N,M,NL,NO,DUBYRS,PV,-

CALL BASIN(SSRT,BOURT,TRT,PRT,AR,N,M,NL,NO,DUBYRS,PV,-

CALL BASIN(SSRT,BOURT,TRT,PRT,AR,N,M,NL,NO,DUBYRS,PV,-
11 CALL SWEEP(SSRT, BOURT, IRT, PRT, AR, N, M, NL, NO, SUBYRS, PVS, - SQBYRH, SUBYR
2N, SQBYRP, NOIG)
 NPRA=NPRA+1
 CALL TABLO2(NPRA, N, SUBYRS, SQBYRH, SUBYRN, SQBYRP, PVS, NSOL, L, ISSL, ISBU,
 21SNI, ISPH, PYKRS, PYKR, PYKR, PPREM, SSUM, DBFUN, A-, ZIMBO, NCNT
 3N, NPICK, NPICK1, NPICK2, ISLC, ISLS, ISLB, NPICK3, NPICK4, NOIG-
 311, SPLIT2)
 GO TO 14
12 CALL SWEEP(SSRT, BOURT, IRT, PRT, AR, N, M, NL, NO, SUBYRS, PVS, - SQBYRH, SUBYR
2N, SQBYRP, NOIG)
 NPRA=NPRA+1
 CALL BASIN(SSRT, BOURT, IRT, PRT, AR, N, M, NL, NO, SUBYRS, PVS, - DQBYRH, DQBYR
2N, DQBYRP, NOIG)
 NPRA=NPRA+1
 CALL TABLO3(NPRA, N, SUBYRS, SQBYRH, SUBYRN, SQBYRP, DQBYR, DQBYR, NSOL, L, ISSL, ISBU,
 21SNI, ISPH, PYKRS, PYKR, PYKR, PPREM, SSUM, DBFUN, A-, ZIMBO, NCNT
 5LL, PPREM, SSUM, DBFUN, A, ZIMBO, NCNTw, NPICK, NPICK1, NPICK2, - ISLC, ISLS, IS
 6LB, NPICK3, NPICK4, NOIG, SPLIT, SPLIT1, SPLIT1, SPLIT2)
 GO TO 14
13 CALL BASIN(SSRT, BOURT, IRT, PRT, AR, N, M, NL, NO, DQBYRS, PVS, - DQBYRH, DQBYR
2N, DQBYRP, NOIG)
 NPRA=NPRA+1
 CALL TABLO4(NPRA, N, DQBYRS, DQBYRH, DQBYRN, DQBYRP, PVS, NSOL, L, ISSL, ISBU,
 21SNI, ISPH, PYKRS, PYKR, PYKR, PPREM, SSUM, DBFUN, A-, ZIMBO, NCNT
 3N, NPICK, NPICK1, NPICK2, ISLC, ISLS, ISLB, NPICK3, NPICK4, NOIG-
 311, SPLIT2)
14 WRITE(6,15) NCNTW
15 FORMAT('1', 132('*'), '/132('*'), '/5X', 'SUBWATERSHED # - ', I2, ' SUMMAR
2Y DATA', '///)
 IF(ISSS.EQ.0) GO TO 53
 IF(ISBU.EQ.0) GO TO 52
 IF(ISNI.EQ.0) GO TO 51
 IF(ISPH.EQ.0) GO TO 50
53 WRITE(6,20)
20 FORMAT(5X, 'THE PULILLANT SELECTED FOR THE OPTIMIZATION-
ANALYSIS IS
1 SUSPENDED SOLIDS', //)
GO TO 24
24 WRITE(6,24)
FORMAT(5X,'THE PULLUANT SELECTED FOR THE OPTIMIZATION-
ANALYSIS IS
1 BOD', //)
GO TO 24
25 WRITE(6,25)
FORMAT(5X,'THE PULLUANT SELECTED FOR THE OPTIMIZATION-
ANALYSIS IS
2 NITROGEN', //)
GO TO 24
26 WRITE(6,26)
FORMAT(5X,'THE PULLUANT SELECTED FOR THE OPTIMIZATION-
ANALYSIS IS
3 PHOSPHORUS', //)
GO TO 24
27 WRITE(6,27)
FORMAT(5X,'THE PULLUANT SELECTED FOR THE OPTIMIZATION-
ANALYSIS IS
NOTE: IF ANY OF THE THREE SELECTORS EQUALS-
1, THAT PRA
2CTICE IS POSSIBLE (ANALYZED IN THE LINEAR PROGRAMMING-
ROUTINE)', //
GO TO 25
28 WRITE(6,28)
FORMAT(5X,'CATCHBASIN SELECTOR ISLC =', IC, //, 8X,'STREET SWEE-
PING SELECTOR ISLS =', IC, //, 8X,'DIVERSION/PERCOLATION SELECTOR ISL-
E =', IC, //, 5X,'THE AREA OF THIS SUBWATERSHED IS =', F7.2, ' ACRES',//
GO TO 29
29 WRITE(6,29)
FORMAT(5X,'THE BOD LOADING RATE OF THIS SUBWATERSHED IS =
5/AC-YR', //, 5X,'THE BOD LOADING RATE OF THIS SUBWATERSHED IS =', F7.2,' LB
5/AC-YR', //, 5X,'THE NITROGEN LOADING RATE OF THIS SUBWATERSH
ED IS =', F7.2,' LB/AC-YR', //, 5X,'THE PHOSPHORUS LOADING RATE OF THIS
SUBWATERSHED IS =', F7.2,' LB/AC-YR', //, 5X,'EFFICIENCY (FRAC
31 TON)', 5X,'REMOVALS (LB/YR)', 8X,'CUST $(S)', 11X,'SELECTED PRACTICE
2(S)', 11X,'LB/YR/PRACTICE REMOVAL')
GO 9876 I=1,N
ZIMBUY(I)=ZIMBUY(I)*(-1.)
9876 CONTINUE
IF(ISSE.EQ.0) GO TO 51
IF(ISBU.EQ.0) GO TO 52
IF(ITON.EQ.0) GO TO 53
IF(ISPH.EQ.0) GO TO 54
51 LL=1
WRITE(6,35) PER1, PYRHR(1), ZIMDOY(1)
55 FORMAT(//, 14X, F4.2, 16X, F9.2, 12X, F10.6)
IF(NPICK(LL).EQ.1) GO TO 77
44 IF(NPICK1(LL).EQ.1) GO TO 88
55 IF(NPICK2(LL).EQ.1) GO TO 99

2353 IF(NPICK3(LL).EQ.1) GO TO 2351
2354 IF(NPICK4(LL).EQ.1) GO TO 2352

GO TO 101
77 WRITE(0,78)SPLIT(LL)
78 FORMAT(+' ',77X,'* CATCHBASIN TECHNOLOGY *',15X,F9.2,/')

GO TO 44
88 WRITE(0,89)SPLIT1(LL)
89 FORMAT(+' ',77X,'* STREET SWEEPING TECHNOLOGY *',15X,F9.2,')

GO TO 55
999 WRITE(0,100)SPLIT2(LL)
100 FORMAT(+' ',77X,'* DIVERSION/PERCULATION *',15X,F9.2,')

GO TO 2353
2351 WRITE(0,2355)
2355 FORMAT(+' ',77X,'* SOLUTION UNBOUNDED *',20X,'--')

GO TO 2354
2352 WRITE(0,2356)SPLIT3
2356 FORMAT(+' ',77X,'* NO FEASIBLE SOLUTION *',16X,F9.2,')

101 LL=LL+1
102 IF(LL.EQ.2) GO TO 103
103 WRITE(0,35)PER2,PYRRS(2),ZIMBOY(2)

GO TO 68
104 WRITE(0,35)PER3,PYRRS(3),ZIMBOY(3)

GO TO 68
105 WRITE(0,35)PER4,PYRRS(4),ZIMBOY(4)

GO TO 68
106 WRITE(0,35)PER5,PYRRS(5),ZIMBOY(5)

GO TO 68
107 IF(NPICK(LL).EQ.1) GO TO 403
403 IF(NPICK1(LL).EQ.1) GO TO 484
406 IF(NPICK2(LL).EQ.1) GO TO 485

2355 IF(NPICK3(LL).EQ.1) GO TO 2357
2357 IF(NPICK4(LL).EQ.1) GO TO 2360

GO TO 66
403 WRITE(0,78)SPLIT(LL)

GO TO 486
486 WRITE(0,69)SPLIT1(LL)

GO TO 487
487 WRITE(0,100)SPLIT2(LL)

GO TO 2358
2358 WRITE(0,2355)

GO TO 2359
2360 WRITE(0,2356)SPLIT3

GO TO 68
32 LL=1
```
WRITE(0,55) PER1,PYRR4(1),Z1M041(1)
355 IF(UNPICK(LL),EN.1) GO TO 710
444 IF(UNPICK1(LL),EN.1) GO TO 680
555 IF(UNPICK2(LL),EN.1) GO TO 990
234 IF(UNPICK3(LL),EN.1) GO TO 302
235 IF(UNPICK4(LL),EN.1) GO TO 304
60 TU 1010
170 WRITE(0,18) SPLIT(LL)
60 TU 490
380 WRITE(0,100) SPLIT1(LL)
60 TU 390
390 WRITE(0,100) SPLIT2(LL)
60 TU 301
250 WRITE(0,105) SPLIT3
60 TU 2303
255 WRITE(0,105) SPLIT
60 TU 2303
1010 LL=LL+1
1020 IF(LL.EQ.2) GO TO 1030
1030 IF(LL.EQ.3) GO TO 1040
1040 IF(LL.EQ.4) GO TO 1050
1050 IF(LL.EQ.5) GO TO 1060
1060 WRITE(0,105) PER2,PYRK4(2),Z1M04(2)
60 TU 350
1070 WRITE(0,105) PER3,PYRK5(3),Z1M04(3)
60 TU 350
1080 WRITE(0,105) PER4,PYRK6(4),Z1M04(4)
60 TU 350
1090 WRITE(0,105) PER5,PYRK7(5),Z1M04(5)
60 TU 2305
2300 IF(UNPICK4(LL),EN.1) GO TO 2307
60 TU 23
2300 WRITE(0,18) SPLIT(LL)
60 TU 2304
2305 WRITE(0,100) SPLIT1(LL)
60 TU 2305
2306 WRITE(0,100) SPLIT2(LL)
60 TU 2306
2307 WRITE(0,105) SPLIT3
60 TU 2307
2310 WRITE(0,105) SPLIT
60 TU 2307
33 LL=1
34 WRITE(0,105) PER1,PYRK1(1),Z1M01(1)
160 IF(UNPICK4(LL),EN.1) GO TO 171
170 IF(UNPICK5(LL),EN.1) GO TO 180
180 IF(UNPICK6(LL),EN.1) GO TO 190
190 IF(UNPICK7(LL),EN.1) GO TO 309
2300 IF(UNPICK4(LL),EN.1) GO TO 2309
```
2370 IF(NPICK4(LL).EQ.1) GO TO 2371
   GO TO 1101
171 WRITE(6,78)SPLIT(LL)
   GO TO 144
180 WRITE(6,69)SPLIT1(LL)
   GO TO 155
199 WRITE(6,100)SPLIT2(LL)
   GO TO 2368
2369 WRITE(6,2355)
   GO TO 2370
2371 WRITE(6,2356)SPLIT3
1101 LL=LL+1
   IF(LL.EQ.2) GO TO 1103
   IF(LL.EQ.3) GO TO 1104
   IF(LL.EQ.4) GO TO 1105
   IF(LL.EQ.5) GO TO 1106
1103 WRITE(6,35)PER2,PYKRN(2),ZIMBOY(2)
   GO TO 168
1104 WRITE(6,35)PER3,PYKRN(3),ZIMBOY(3)
   GO TO 168
1105 WRITE(6,35)PER4,PYRRN(4),ZIMBOY(4)
   GO TO 168
1106 WRITE(6,35)PER5,PYKRN(5),ZIMBOY(5)
   IF(NPICK(LL).EQ.1) GO TO 70
   IF(NPICK1(LL).EQ.1) GO TO 71
   IF(NPICK2(LL).EQ.1) GO TO 72
2372 IF(NPICK3(LL).EQ.1) GO TO 2373
2374 IF(NPICK4(LL).EQ.1) GO TO 2375
   GO TO 66
70 WRITE(6,78)SPLIT(LL)
   GO TO 73
71 WRITE(6,69)SPLIT1(LL)
   GO TO 74
72 WRITE(6,100)SPLIT2(LL)
   GO TO 2372
2373 WRITE(6,2355)
   GO TO 2374
2375 WRITE(6,2356)SPLIT3
   GO TO 66
34 LL=1
   WRITE(6,35)PER1,PYKRP(1),ZIMBOY(1)
260 IF(NPICK(LL).EQ.1) GO TO 277
244 IF(NPICK1(LL).EQ.1) GO TO 280
255 IF(NPICK2(LL).EQ.1) GO TO 299
2376 IF(NPICK3(LL).EQ.1) GO TO 2377
2377 IF(NPICK4(LL).EQ.1) GO TO 2379
   GO TO 2101
271 WRITE(6,78)SPLIT(LL)
   GO TO 244
280 WRITE(6,69)SPLIT1(LL)
GO TO 255
299 WRITE(6,100) SPLIT2(LL)
GO TO 2376
2377 WRITE(6,2355)
GO TO 2378
2379 WRITE(6,2356) SPLIT3
2101 LL=LL+1
IF(LL.EQ.2) GO TO 2103
IF(LL.EQ.3) GO TO 2104
IF(LL.EQ.4) GO TO 2105
IF(LL.EQ.5) GO TO 2106
2103 WRITE(6,35) PER2, PYRRP(2), ZIMD0Y(2)
GO TO 266
2104 WRITE(6,35) PER3, PYRRP(3), ZIMD0Y(3)
GO TO 266
2105 WRITE(6,35) PER4, PYRRP(4), ZIMD0Y(4)
GO TO 266
2106 WRITE(6,35) PER5, PYRRP(5), ZIMD0Y(5)
IF(NPICK(LL).EQ.1) GO TO 2222
227 IF(NPICK1(LL).EQ.1) GO TO 226
229 IF(NPICK2(LL).EQ.1) GO TO 230
2360 IF(NPICK3(LL).EQ.1) GO TO 2301
2382 IF(NPICK4(LL).EQ.1) GO TO 2363
GO TO 66
2222 WRITE(6,78) SPLIT(LL)
GO TO 227
228 WRITE(6,59) SPLIT1(LL)
GO TO 229
230 WRITE(6,100) SPLIT2(LL)
GO TO 230
2381 WRITE(6,2355)
GO TO 2362
2383 WRITE(6,2356) SPLIT3
60 WRITE(6,200)
3003 FORMAT(/,132(''),/132(''))
SIGAK=SIGAK+AR
SIGAKS=SIGARS+(AR*SSRT)
SIGAKB=SIGARB+(AR*B0URT)
SIGAKN=SIGARN+(AR*TRI)
SIGAKP=SIGARP+(AR*PRT)
IF(NCNTW.EQ.WMAT)GO TO 1
GO TO 9
1 WRITE(6,44) SIGAK
644 FORMAT('1',25(132(''),4U(''),52X,4U(''),/4U('')-"
52X,4U('')/3/,4U(''),52X,4U(''),/4U(''),7X,'OPTIMIZATION FOR -"
8ATERSHED',0X,4U(''),/4U(''),11X,'TOTAL WATERSHED AR-
EA = ',F8.2,
312X,40(''),/40(''),52X,40(''),/40(''),52X,40('')-"
)/,40(''),)
352X, 40('?'), 125(152('?'))
100
1F(ISSS. EQ. 0) GO TO 300
1F(ISBU. EQ. 0) GO TO 400
1F(ISN1. EQ. 0) GO TO 500
1F(ISPH. EQ. 0) GO TO 600
300 PMAX5 = MAX*SIGARS
   GO TO 800
400 PMAX6 = MAX*SIGAR6
   GO TO 800
500 PMAXN = MAX*SIGARN
   GO TO 800
600 PMAXP = MAX*SIGARP
600 CALL TABLEA(PMAXS, PMAXB, PMAXP, PMAXN, SSUM, OBFUN, NCONW, ISSO, ISBD, ISN
   31, ISPH, NSOL, NBIG, ARKCM, IHELP, IHELP1, IHELP2, IHELP3, IHELP4, IHELP5, IH
   3ELP6, IHELP7, IHELP8, IHELP9, IHELPA, IHELPG, IHELPM, IHELPU, IHELPU,
   IHELPE, IHELPE, IHELPE, IHELPE, UG, UG1, UG2, UG3, UG4, UG5, UG6, UG7, UG8, U
   7G9, UG10, UG11, UG12, UG13, UG14, UG15, UG16, UG17, UG18, UG19)
   WRITE(0, 3006)
3006 FORMAT('1', 2(132('?')) '/ ', 5X, 'TOTAL WATERSHED SUMMARY-
   DATA', ' ')
   1F(ISSS. EQ. 0) GO TO 2789
   1F(ISBD. EQ. 0) GO TO 2790
   1F(ISN1. EQ. 0) GO TO 2791
   1F(ISPH. EQ. 0) GO TO 2792
2789 WRITE(6, 20)
   GO TO 3005
2790 WRITE(6, 21)
   GO TO 3005
2791 WRITE(6, 22)
   GO TO 3005
2792 WRITE(6, 23)
3005 WRITE(6, 3007)
3007 FORMAT('/ ', 5X, 'OUTPUT INTERPRETATION FOR WATERSHED', ' ', '7X, 'MAXIMUM'  
   3, 24X, 'SELECTED', '/ ', 5X, 'LBS REMOVED', '6X, 'CUST (S)', 6X, 'S-
   UBWATERSHEDS
   8', 5X, 'LBS/YR')
   1F(ISSS. EQ. 0) GO TO 306
   1F(ISBD. EQ. 0) GO TO 307
   1F(ISN1. EQ. 0) GO TO 306
   1F(ISPH. EQ. 0) GO TO 309
306 WRITE(6, 4001) PMAX5, ARKCM
   GO TO 310
307 WRITE(6, 4001) PMAX6, ARKCM
4001 FORMAT('/ ', 6X, 'FLU.S, 5X, F11.1')
   GO TO 310
308 WRITE (0, 401) PMAKX, ARKCM
GO TO 310
309 WRITE (0, 401) PMAKP, ARKCM
310 IF (HELP.EQ.1) GO TO 311
311 IF (HELP1.EQ.1) GO TO 313
312 IF (HELP2.EQ.1) GO TO 315
314 IF (HELP3.EQ.1) GO TO 317
316 IF (HELP4.EQ.1) GO TO 319
318 IF (HELP5.EQ.1) GO TO 321
320 IF (HELP6.EQ.1) GO TO 323
322 IF (HELP7.EQ.1) GO TO 325
324 IF (HELP8.EQ.1) GO TO 327
326 IF (HELP9.EQ.1) GO TO 329
328 IF (HELPa.EQ.1) GO TO 331
330 IF (HELPb.EQ.1) GO TO 333
332 IF (HELPc.EQ.1) GO TO 335
334 IF (HELPd.EQ.1) GO TO 337
336 IF (HELPe.EQ.1) GO TO 339
338 IF (HELPf.EQ.1) GO TO 341
340 IF (HELPg.EQ.1) GO TO 343
342 IF (HELPh.EQ.1) GO TO 345
344 IF (HELPi.EQ.1) GO TO 347
346 IF (HELPj.EQ.1) GO TO 349
GO TO 4708
311 WRITE (0, 401) UG
401 FORMAT ('+', 42X, '# 1', 7X, F8.1, /)
GO TO 312
313 WRITE (0, 402) UG
402 FORMAT ('+', 42X, '# 2', 7X, F8.1, /)
GO TO 314
315 WRITE (0, 403) UG
403 FORMAT ('+', 42X, '# 3', 7X, F8.1, /)
GO TO 316
317 WRITE (0, 404) UG
404 FORMAT ('+', 42X, '# 4', 7X, F8.1, /)
GO TO 318
319 WRITE (0, 405) UG
405 FORMAT ('+', 42X, '# 5', 7X, F8.1, /)
GO TO 320
321 WRITE (0, 406) UG
406 FORMAT ('+', 42X, '# 6', 7X, F8.1, /)
GO TO 322
323 WRITE (0, 407) UG
407 FORMAT ('+', 42X, '# 7', 7X, F8.1, /)
GO TO 324
325 WRITE (0, 408) UG
408 FORMAT ('+', 42X, '# 8', 7X, F8.1, /)
GO TO 326
327 WRITE (0, 409) UG
409 FORMAT ('+', 42X, '# 9', 7X, F8.1, /)
GO TO 328
329 WRITE(0,410)UG9
410 FORMAT('5',42X,'#10',7X,F8.1,/)  
GO TO 330
331 WRITE(0,411)UG10
411 FORMAT('5',42X,'#11',7X,F8.1,/)  
GO TO 332
333 WRITE(0,412)UG11
412 FORMAT('5',42X,'#12',7X,F8.1,/)  
GO TO 334
335 WRITE(0,413)UG12
413 FORMAT('5',42X,'#13',7X,F8.1,/)  
GO TO 336
337 WRITE(0,414)UG13
414 FORMAT('5',42X,'#14',7X,F8.1,/)  
GO TO 338
339 WRITE(0,415)UG14
415 FORMAT('5',42X,'#15',7X,F8.1,/)  
GO TO 340
341 WRITE(0,416)UG15
416 FORMAT('5',42X,'#16',7X,F8.1,/)  
GO TO 342
343 WRITE(0,417)UG16
417 FORMAT('5',42X,'#17',7X,F8.1,/)  
GO TO 344
345 WRITE(0,418)UG17
418 FORMAT('5',42X,'#18',7X,F8.1,/)  
GO TO 346
347 WRITE(0,419)UG18
419 FORMAT('5',42X,'#19',7X,F8.1,/)  
GO TO 348
349 WRITE(0,420)UG19
420 FORMAT('5',42X,'#20',7X,F8.1,/)  
GO TO 342
328 WRITE(0,4769)
4769 FORMAT(6X,'HEADING "MAXIMUM LBS REMOVED" GIVES THE-
PRODUCT OF
5ASHOFF YEARLY',//,5X,'LOADINGS TIMES REMOVAL AS A FRA-
TION (XMAX)'.
5 HEADING "CUST ($)"',//,5X,'GIVES THE PRESENT VALUE CO-
ST FOR THE T
6OTAL WATERSHED. HEADING',//,5X,'"SELECTED SUBWATERSHED-
S" SHOWS IN
6WHICH OF THE SUBWATERSHEDS BEST',//,5X,'MANAGEMENT PRACTI-
CLES ARE CH
6SEN. THE ASSOCIATED REMOVALS PER SUB-',//,5X,'WATERSH-
ED ARE GIVEN
7 IN THE COLUMN HEADED "LBS/YR". THE SIZE OF',//,5X,'PRACTICE IS TH
5EN DETERMINED FOR EACH SUBWATERSHED BY USING THE',//,5X-
,'LBS/YR CUL
STOP
END

SUBROUTINE CATCH CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE CATCH(SSRT, SOUT, ITI, PRI, AR, N, M, NL, NO, CUBYR, S, PV, CUBYR, B,
2CUBYR, CUBYRP, NBIG)

C CARD GROUP # 2
C VARIABLE DESCRIPTIONS

C ICB = THE OPTION TO READ IN OR USE GIVEN EFFICIENCIES
C IF ICB = 0 READ IN EFFICIENCIES AND FREQUENCIES
C IF ICB = 1, USE GIVEN EFFICIENCIES AND FREQUENCIES
C NB = THE NUMBER OF CATCHBASINS IN GIVEN AREA
C CCB = THE ORM COST OF CLEANING CATCHBASINS - $/CATCHBASIN
C COST = CAPITAL COST OF INLET CONVERSION TO CATCHBASINS - $
C EINT = INTEREST RATE FOR PRESENT VALUE ANALYSIS (FRACTION)
C ENY = NUMBER OF YEARS IN PRESENT VALUE ANALYSIS
C
C CCSCCCSCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C DIMENSION EFFS(6), EFFB(6), EFFH(6), EFFP(6), UBYRS(6), OBYRS(6), NBYRN(6),
C OBYRP(6), NBYRNP(6), XCB(6), COSTCB(6), CUBYR(6), CUBYRP(6)
C
C READ(5,33)ICB, NB, CCB, AA, BB, CC, DD, EE, FF, GG, HH, OU, PP, QU, KR, SS, TT, UU,
C 2V, WW, XX, YY, ZZ
C 33 FORMAT(I1,14,FS*2,20F3.2)
C READ(5,10)AAA, BBB, CCC, DDD, EEE
C 10 FORMAT(SF3.1)
C READ(5,111)CUST, EINT, ENY
C 111 FORMAT(F7.1, F3.2, F3.2)
C WRITE(5,888)
C WRITE(5,112)
112 FORMAT(/5X,'CATCH BASIN CLEANING DATA',/)
WRITE(6,333) AR,CCR,NO,COST,SSRT,DOUR,TRI,PR1
333 FORMAT(5X,'AREA OF THE WATERSHED = ',F7.2,' ACRE',/)
2 OF CLEANING CATCH BASIN = 5',F7.2,' /CATCHBASIN',/5X,'URM COST
6 CATCHBASINS = ',I5,5X,'CAPITAL CUST OF CONVERSION = ',F9.2,5X,
6,'SUSPENDED SOLIDS LOADING RATE = ',F7.2,' LB/AC-YR',/
3ADING RATE = ',F7.2,' LB/AC-YR',/5X,'NITROGEN LOADING RATE = ',F
27.2,' LB/AC-YR',/5X,'PHOSPHORUS LOADING RATE = ',F7.
1X,'/)
98 FORMAT(/',132('*'))
IF(IEB.EQ.0) GO TO 1
C EFFICIENCIES FOR CATCHBASIN CLEANING AT DIFFERENT C
FREQUENCIES AND DIFFERENT POLLUTANTS
EFFS(1)=0.196
EFFS(2)=0.591
EFFS(3)=0.75
EFFS(4)=0.6
EFFS(5)=0.62
EFFB(1)=0.69
EFFB(2)=0.139
EFFB(3)=0.26
EFFB(4)=0.3
EFFB(5)=0.32
EFFN(1)=0.071
EFFN(2)=0.143
EFFN(3)=0.274
EFFN(4)=0.51
EFFN(5)=0.33
EFFP(1)=0.01
EFFP(2)=0.031
EFFP(3)=0.06
EFFP(4)=0.08
EFFP(5)=0.09
C FREQUENCIES OF CLEANING
ACB(1)=0.5
ACB(2)=1.0
ACB(3)=2.0
ACB(4)=3.0
ACB(5)=4.0
GO TO 2
C READ IN EFFICIENCIES AND FREQUENCIES (DEFAULT VALUES)
1 EFFS(1)=AA
EFFS(2)=BB
WRITE(6,250)ACD(I),CUSD,B(1),CAPT,PV(I)
250 FORMAT(/33X,'FREQUENCY ',FS.1,3X,'1',1X,F15.5,3X,F15.5-
   ,3X,F15.5)
500 CONTINUE
WRITE(6,98)
WRITE(6,87)
81 FORMAT(/,'THE REQUIRED FREQUENCY OF CATCHBASIN C-
   LEANING IS A 1 FUNCTION OF SEVERAL LOCAL PARAMETERS','/,'22X,'SUCH AS SUMP CAPAC-
   TITY, QUANTITY OF ACCUMULATED STREET SOLIDS, ANTECEDENT-
   DRY PERIOD, 3 METEOROLOGICAL CONDITIONS, STREET CLEANING M-
   ETHODS AND PRACTICES, SURROUNDING LAND USE','/,'22X,'TOPOGRAPHY, AND TO SOME E-
   XTENT, THE TYPE OF SOIL ADJACENT TO THE STREET:',/,'49-
   X,'FREQUENCY 7 0.5 = ONCE IN 2 YEARS','/,'49X,'FREQUENCY 1.0 = ONCE A-
   YEAR','/,'49X,'FREQUENCY 2.0 = 2 TIMES PER YEAR','/,'49X,'FREQUENCY 3.0 = 0 = 3 TIMES
9PER YEAR','/,'49X,'FREQUENCY 4.0 = 4 TIMES PER YEAR'
)WRITE(6,98)
1F(NDIG.E0.0) GO TO 779
DO 7 I=1,N
A(I,1)=QBYKB(I)
A(I,2)=PV(I)
7 CONTINUE
WRITE(6,88)
686 FORMAT('1','/,'132('X'),//)
WRITE(6,79)
79 FORMAT(/5AX,'SUSPENDED SOLIDS')
CALL PLOT(NO,A,N,M,NL)
WRITE(6,753)
753 FORMAT(/,'56X,'SUSPENDED SOLIDS SLOPE & INTERCEPT-
DATA')
CALL SLOPE(A,N)
DO 8 I=1,N
A(I,1)=QBYKB(I)
A(I,2)=PV(I)
8 CONTINUE
NO=NU+1
WRITE(6,88)
WRITE(6,13)
13 FORMAT(/,'65X','SOU')
CALL PLOT(NO,A,N,M,NL)
WRITE(6,43)
43 FORMAT(/,'57X','SOU SLOPE & INTERCEPT DATA')
CALL SLOPE(A,N)
DO 9 I=1,N
A(I,1)=OBYRN(I)
A(I,2)=PV(I)
9 CONTINUE
NO=NO+1
WRITE(6,680)
WRITE(6,686)
FORMAT('/62X,'NITRIGEN')
CALL PLOT(NO,A,N,M,NL)
WRITE(6,232)
232 FORMAT('1',/63X,'NITRIGEN SLOPE & INTERCEPT DATA')
CALL SLOPE(A,N)
DO 20 I=1,N
A(I,1)=OBYRP(I)
A(I,2)=PV(I)
20 CONTINUE
NO=NO+1
WRITE(6,680)
WRITE(6,1111)
1111 FORMAT('/62X,'PHOSPHURUS')
CALL PLOT(NO,A,N,N,M,NL)
WRITE(6,555)
555 FORMAT('1',/63X,'PHOSPHURUS SLOPE & INTERCEPT DATA')
CALL SLOPE(A,N)
779 DO 6661 I=1,N
COBYRS(I)=UBYRS(I)
COBYRB(I)=UBYRB(I)
COBYRN(I)=UBYRN(I)
6661 COBYRP(I)=UBYRP(I)
NO =1
RETURN
END

CCCCCGCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCGCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE SWEEP CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE SWEEP(SSRT,HOURT,RT,PR,T,AR,N,M,NL,NO,SubYR=<
G=PV5,SOBYRP,NDIG)
CCCCCGCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCGCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

C CARD GROUP # 3
C VARIABLE DESCRIPTIONS
C
C 1P = THE OPTION TO READ IN OR USE GIVEN EFFICIENCIES.
C
C IF 1P = 0, USE THE GIVEN EFFICIENCIES AND FREQUENCIES.
IF IP = 1, READ IN EFFICIENCIES AND FREQUENCIES -

PD = THE POPULATION DENSITY FOR THE GIVEN AREA (PERSONS/AC) -

CSW = THE ORN COST OF STREET SWEETING ($/CURB-MILE) -

COSTS = CAPITAL COST OF A STREET SWEeper ($) -

EINTS = THE INTEREST RATE FOR PRESENT VALUE ANALYSIS (-)

ENYS = NUMBER OF YEARS IN PRESENT VALUE ANALYSIS -

DIMENSION QBYRS(6),QBYRP(6),QBYRP(6),EFFS(6),-

6(6),EFFP(6),A(5,2),CUSTSW(6),XSW(6),QBYRP(6),CAPSW(6)-

READ(5,19)IP,PD,CSW,AB,BL,LD,DE,EF,HL,AJ,AK,AL,AM,UP,A-

WRITE(6,115)

115 FORMAT('1',132('x'))

WRITE(6,228)

19 FORMAT(11,F3.0,F5.2,20F3.2)

READ(5,22)GGG,HHH,PPP,QQQ,RRR

WRITE(6,119)PD,CSW,GL,AR,CUSIS,SORT,SORT,RT,PT,

199 FORMAT(5X,'POPULATION DENSITY = ',F3.0,' PERSONS/ACRE' -

20ST PER CURB MILE SWEPT = ',F5.2,'/5X,'GUTTER LENGTH -

2 CURB-MI/AC,'/5X,'AREA OF THE WATERSHED = ',F7.2,' -

2 CAPITAL COST OF SWEeper = $',F9.2,'/5X,'SUSPENDED SOL- 

3 RATE = ',F7.2,' LB/AC-YR','/5X,'BOD LOADING RATE = ',- 

3L-YR','/5X,'NITROGEN LOADING RATE = ',F7.2,' LB/AC-YR -

3PHUS LOADING RATE = ',F7.2,' LB/AC-YR,'/

WRITE(6,116)

1F(1P.EQ.0)GO TO 11

EFFS(1)=AB
\[ \text{EFS}(2) = \theta C \]
\[ \text{EFS}(3) = LD \]
\[ \text{EFS}(4) = DE \]
\[ \text{EFS}(5) = EF \]
\[ \text{EFFB}(1) = HI \]
\[ \text{EFFB}(2) = AJ \]
\[ \text{EFFB}(3) = AK \]
\[ \text{EFFB}(4) = AL \]
\[ \text{EFFB}(5) = AM \]
\[ \text{EFFN}(1) = UP \]
\[ \text{EFFN}(2) = AO \]
\[ \text{EFFN}(3) = UR \]
\[ \text{EFFN}(4) = KS \]
\[ \text{EFFN}(5) = ST \]
\[ \text{EFFP}(1) = VN \]
\[ \text{EFFP}(2) = WX \]
\[ \text{EFFP}(3) = XY \]
\[ \text{EFFP}(4) = ZZ \]
\[ \text{EFFP}(5) = BA \]
\[ \text{XSW}(1) = GUG \]
\[ \text{XSW}(2) = HHH \]
\[ \text{XSW}(3) = PPP \]
\[ \text{XSW}(4) = QGQ \]
\[ \text{XSW}(5) = RRR \]

\text{GO TO 12}

\text{EFS}(1) = 0.299
\text{EFS}(2) = 0.421
\text{EFS}(3) = 0.609
\text{EFS}(4) = 0.707
\text{EFS}(5) = 0.87
\text{EFFB}(1) = 0.0275
\text{EFFB}(2) = 0.3875
\text{EFFB}(3) = 0.56
\text{EFFB}(4) = 0.65
\text{EFFB}(5) = 0.8
\text{EFFN}(1) = 0.0413
\text{EFFN}(2) = 0.058
\text{EFFN}(3) = 0.084
\text{EFFN}(4) = 0.0975
\text{EFFN}(5) = 0.12
\text{EFFP}(1) = 0.0158
\text{EFFP}(2) = 0.0194
\text{EFFP}(3) = 0.028
\text{EFFP}(4) = 0.0325
\text{EFFP}(5) = 0.04

\text{C}
\text{XSW}=1.0=\text{ONCE EVERY DAY FREQUENCY}
\text{C}
\text{XSW}=0.5=\text{ONCE EVERY 2 DAY FREQUENCY}
\text{C}
\text{XSW}=0.33=\text{ONCE EVERY 3 DAY FREQUENCY}
\text{C}
\text{XSW}=0.20=\text{ONCE EVERY 5 DAY FREQUENCY}
\text{C}
\text{XSW}=0.143=\text{ONCE EVERY 7 DAY FREQUENCY}
\text{XSN}(1) = 0.143 \\
\text{XSN}(2) = 0.2 \\
\text{XSN}(3) = 0.35 \\
\text{XSN}(4) = 0.5 \\
\text{XSN}(5) = 1.0 \\

\text{YRLS} = \text{SZT} \times \text{GL} \times \text{AR} \\
\text{YRLB} = \text{BDKT} \times \text{GL} \times \text{AR} \\
\text{YRLN} = \text{THT} \times \text{GL} \times \text{AR} \\
\text{YRLP} = \text{PHT} \times \text{GL} \times \text{AR} \\
\text{WRITE}(6, 111) \\
\text{111 FORMAT}(/ \text{55}x, \text{, 'PUUNDS PER YEAR REMOVED', /}, /, / \text{31x, 'FREQUENCY', /}, /, / \text{5x, 'SOLID', /}, /, / \text{5x, 'NITROGEN', /}, /, / \text{7x, 'PHOSPHORUS'}) \\
\text{DO I=1,N} \\
\text{UYRS(I)} = \text{YRLS} \times \text{EFFS(I)} \\
\text{UBYR(I)} = \text{YRLB} \times \text{EFFB(I)} \\
\text{UBYRN(I)} = \text{YRLN} \times \text{EFFN(I)} \\
\text{UBYRP(I)} = \text{YRLP} \times \text{EFFP(I)} \\
\text{WRITE}(6, 300) \text{XSN(I)}, \text{UBYRS(I)}, \text{UBYRo(I)}, \text{UBYRN(I)}, \text{UBYRP(I)} \\
\text{300 FORMAT}(/ \text{33}x, /, / \text{F5.3, 3x, 4F15.5}) \\
\text{1 CONTINUE} \\
\text{WRITE}(6, 11b) \\
\text{11b FORMAT}(/, / \text{132('*')})) \\
\text{WRITE}(6, 99b) \\
\text{99b FORMAT}(/, / \text{50x, 'COST OF STREET SWEETING PER YEAR', /}, /, / \text{59x-}, /, / \text{'URM', 14x, /}, /, / \text{5x, 'PRESENT VALUE(20 YR, 7%')}) \\
\text{DO 66 I=1,N} \\
\text{C FROM STORMWATER MANAGEMENT MODEL MANUAL, LEVEL I; PG. 35} \\
\text{CUSTSW(I)} = 365 \times \text{CSW} \times \text{GL} \times \text{AR} \times \text{XSN(I)} \\
\text{C ASUME SWEEPER CAN COVER 20 CURB-MI/DAY} \\
\text{CAPSW(I)} = \text{CUST} \times \text{XSN(I)} \times \text{GL} \times \text{AR} \times 26. \\
\text{URM} \times \text{SW(I)} = \text{CUSTSW(I)} \times (((1 + \text{EINTS}) \times \text{ENYS}) - 1) / ((1 - \text{+ EINTS}) \times \text{EN}) \\
\text{2YS))) \\
\text{PVd(I)} = \text{CAPSW(I)} + \text{URM} \times \text{SW(I)} \\
\text{WRITE}(6, 2000) \text{XSN(I)}, \text{CUSTSW(I)}, \text{CAPSW(I)}, \text{PVd(I)} \\
\text{200 FORMAT}(/ \text{33}x, /, / \text{F5.3, 3x, 'S', F15.5, 3x, F15.5, 3x-, /}, /, / \text{F15.5}) \\
\text{66 CONTINUE} \\
\text{WRITE}(6, 11b) \\
\text{WRITE}(6, 99b) \\
\text{999 FORMAT}(/, / \text{47x, 'FREQUENCY 0.143 = ONCE EVERY 7 DAYS', /}, /, / \text{47x, 'FREQUENCY 0.20 = ONCE EVERY 5 DAYS', /}, /, / \text{47x, 'FREQUENCY 0.33 = ONCE EVERY 2 DAYS', /}, /, / \text{47x, 'FREQUENCY 0.50 = ONCE EVERY DAY'})
WRITE(6,110)
IF(N0T.G.EQ.0) GO TO 779
DO 98 I=1,N
A(I,1)=QBYKS(I)
A(I,2)=PVS(I)
98 CONTINUE
WRITE(6,886)
886 FORMAT('1','132('*'),//')
WRITE(6,59)
59 FORMAT('SUSPENDED SOLIDS')
CALL PLOT(NO,A,N,M,NL)
WRITE(6,45)
45 FORMAT('SUSPENDED SOLIDS SLOPE & INTERCEPT DATA')
CALL SLOPE(A,N)
DO 88 I=1,N
A(I,1)=QBYKN(I)
A(I,2)=PVS(I)
88 CONTINUE
NO=NU+1
WRITE(6,888)
WRITE(6,77)
77 FORMAT('BOD')
CALL PLOT(NO,A,N,M,NL)
WRITE(6,72)
72 FORMAT('BOD SLOPE & INTERCEPT DATA')
CALL SLOPE(A,N)
DO 78 I=1,N
A(I,1)=QBYKN(I)
A(I,2)=PVS(I)
78 CONTINUE
NO=NU+1
WRITE(6,888)
WRITE(6,6)
6 FORMAT('NITROGEN')
CALL PLOT(NO,A,N,M,NL)
WRITE(6,64)
64 FORMAT('NITROGEN SLOPE & INTERCEPT DATA')
CALL SLOPE(A,N)
DO 3 I=1,N
A(I,1)=QBYKP(I)
A(I,2)=PVS(I)
3 CONTINUE
NO=NU+1
WRITE(6,888)
WRITE(6,61)
61 FORMAT('PHOSPHORUS')
CALL PLOT(NO,A,N,M,NL)
WRITE(6,66)
66 FORMAT('PHOSPHORUS SLOPE & INTERCEPT DATA')
56 FORMAT('PHOSPHORUS SLOPE & INTERCEPT DATA')
CALL SLHPEF(N,N)
179 on S051 1=1,
UNBYR(N)=UNBYR(1)
UNBYR(N)=UNBYR(1)
UNBYR(N)=UNBYR(1)
S051 UNBYRP(N)=UNBYRP(1)
RETURN
END

SUBROUTINE BASIN (S051, UNBYR, UNBYRP, N)

CARD GROUP # 4

VARIABLE DESCRIPTIONS

IS = SOIL TYPE OPTION. IF IS = 0 SOIL TYPE = A
IF IS = 1 SOIL TYPE = B

NRU = THE VARIABLE TO READ IN DIFFERENT EFFICIENCIES
IF NRU = 0 USE GIVEN EFFICIENCIES FEP (1-6)
IF NRU = 1 READ IN EFFICIENCIES R, M, E, AND F

CL = THE RUNOFF COEFFICIENT

CNU = THE CURVE NUMBER, RELATED TO LAND USE.

CLAN = LAND COST = $/AC

CLNO = EXCAVATION COST = $/CUBIC YARD

CLNOR = COVER CROP COST = $/SQUARE YARD

CLNUT = INLET/OUTLET COST = $

SD = DEPTH OF BASIN = FEET

DIMENSION FEP(6), V(5), N(4), UNBYR(6), UNBYRP(4), N

2LD(6), CCEX(6), CCGL(6), CCST(6), ICLUST(6), A(5,6), VA(6), VAM(6), DQBYYP
5(6), PVV(6), ORMB(6), DQBYRM(6), DQBYRP(6), DQBYRN(6), DVE(6), URM(6)
READ(5,41) IS, NRD, CN, CCCLN, EXCLS, LOCUS, CLOUT, P, B, R, G, C, F
2v FORMAT(211, F4.2, F4.0, F5.0, 2F5.2, F4.0, F3.1, 5F4.2)
READ(5, 14) CORM, EINTB, ENVY
19 FORMAT(F7.2, F3.2, F3.0)
WRITE(6, 110)
11b FORMAT('1', '12('*)
DIV(1) = 0.25
DIV(2) = 0.5
DIV(3) = 0.75
DIV(4) = 1.0
DIV(5) = 1.25
WRITE(6, 778)
77b FORMAT('/5X, 'DIVERSION BASSIN DATA' ,//)
IF(I3.EQ.1) GO TO 56
WRITE(6, 60)
0 FORMAT('/5X, 'SOIL TYPE = A'
GO TO 5
5b WRITE(6, 15)
15 FORMAT('/5X, 'SOIL TYPE = D'
GO TO 5
5 WRITE(6, 100) AR, CN, SSRT, BUDRT, TRT, PRT, CCLN, EXCLS, CCUS,
T, CLOUT
100 FORMAT('/5X, 'AREA OF AATERSHED = ', F7.2, ' ACRES',//, 5X,'-
CURVE NUMBER 4 = ', F4.0, '/5X, 'SUSPENDED SOLIDS LOADING RATE = ',F7.-
2, ' LB/AC-Y
2K', '/5X, 'BUD LOADING RATE = ', F5.1, ' LB/AC-YR', '/5X, 'N-
ITROGEN LOAD 3ING RATE = ', F5.1, ' LB/AC-YR', '/5X, 'PHOSPHORUS LOADING -=
RATE = ', F5.1
2, ' LB/AC-YR', '/5X, 'LAND CUST = $ ', F5.0, '/ACRE', '/5X-
EXCAVATION 2 CUST = $ ',F5.2, '/CUBIC YD', '/5X, 'COVER CROP CUST = -
2, F5.2, '/
2 SQUARE YD', '/5X, 'INLET/OUTLET COST = $ ', F4.0)
IF(C.GT.1) GO TO 5610
CALL CVALU(C, FFF)
WRITE(6, 34) C, FFF(1), FFF(2), FFF(3), FFF(4), FFF(5)
34 FORMAT('/5X, 'C VALUE = ', F5.3, '/5X, 'INTERPOLATED EFFICI-
ENCIES', '/5X
3X, 'FFF(1) = ', F5.3, '/5X, 'FFF(2) = ', F5.3, '/5X, 'FFF(3) = -
F5.3, '/5X,
6 FFF(4) = ', F5.3, '/5X, 'FFF(5) = ', F5.3, '/5X
WRITE(6, 115)
GO TO 7
IF(NRD.EQ.0) GO TO 10

C FFF(1)=R
C FFF(2)=H
C FFF(3)=G
C FFF(4)=E
C FFF(5)=F
C GO TO 7
C

C FFF(1-5) ARE THE EFFICIENCIES OF 0.25, 0.50, 0.75, 1.00, -
C AND 1.25 = 00980
C
C DIVERSION VOLUMES, RESPECTIVELY. (WANIELISTA ET AL)

10 FFF(1)=0.8
C FFF(2)=0.9
C FFF(3)=0.95
C FFF(4)=0.99
C FFF(5)=0.999
C
C YRLOX = THE YEARLY LOADING RATE IN LBS/YR FOR A GIVEN -
C AREA, WHERE 00995
C
7 YRLOD=SSRT*AR
C YRLOD=BOORT*AR
C YRLODN=INT*AR
C YRLODP=PR1*AR
C IF(IS.EQ.1) GO TO 49
C
C GO TO 657
C
C V(1-5) = THE VOLUMES OF BASINS 3 FEET DEEP AND IN AN I-
C MPERVIOUS 01003
C
49 V(1) = 0.02*(AR**1.31)
C V(2) = 0.05*(AR**1.24)
C V(3) = 0.15*(AR**1.11)
C V(4) = 0.20*(AR**1.07)
C V(5) = 0.24*(AR**1.04)
C
C GO TO 658
C
C V(1-5) = THE VOLUMES OF FONDS FIVE FEET DEEP AND IN AN I-
C MPERVIOUS 01011
C
657 V(1) = 0.016*(AR**1.28)
C V(2) = 0.046*(AR**1.18)
C V(3) = 0.09*(AR**1.11)
C V(4) = 0.14*(AR**1.07)
C V(5) = 0.2*(AR**1.04)
C
C UBYRX = THE POUNDUS PER YEAR OF POLLUTANT X WHICH WILL -
C BE REMOVED. 01018
C
658 WRITE(6,102)
C 102 FORMAT(/55,X,'POUNDS PER YEAR REMOVED',/11X,'DIVERSION-
C VOLUME (IN
C 1.)',/X,'SOLIDS',13X,'BOD',9X,'NITROGEN',7X,'PHOSPHURUS-
C ')
C 00 1 I=1,N
QBYSR(I) = FFF(I) * YRLOD
QBYRBI(I) = FFF(I) * YRLOD
QBYRN(I) = FFF(I) * YRLODN
QBYP(I) = FFF(I) * YRLODP
WRITE(6, 12) DIV(I), QBYSR(I), QBYRBI(I), QBYRN(I), QBYRP(I)
12 FORMAT(/20X, F4.2, 10X, 4F15.5)
1 CONTINUE
WRITE(6, 115)
115 IF(IS. EQ. 1) GO TO 444
GO TO 654
444 IF(D. LT. 3.) GO TO 14
C VC(J) COMPUTES THE VOLUME OF BASINS 3 FEET DEEP IN COMPOSITE 01034

C LAND USE AREAS.
WRITE(6, 110)
110 FORMAT(///36X, 'VOLUME OF BASIN 3 FEET DEEP IN COMPOSITE LAND USE A
1RES (VC) AND', //, 39X, 'VOLUME OF BASIN 3 FEET DEEP IN -
AN IMPERVIOUS
25 WATERSHED (V)', //, 30X, 'DIVERSION VOLUME (IN.)', 6X, 'V-'
C(AC-FI)', 6X
3, 'V(AC-FI)')
00 1V8 J=1, N
VC(J)=V(J)*((0.07+(0.92*(CN/100.)))
WRITE(6, 11) DIV(J), VC(J), V(J)
108 CONTINUE
WRITE(6, 115)
C CCLO = THE LAND COST EQUATION
C CCEX = THE EXCAVATION COST EQUATION
C CCCC = THE COVER CROP COST EQUATION
C CCO = THE CAPITAL COST EQUATION
C CTCOST = THE TOTAL CAPITAL COST
WRITE(6, 38)
00 2V 1=1, N
CCLO(I)=CLAN*1.2*VC(I)/3.
CCEX(I)=ECOS*VC(I)*4356U./27.
CCCC(I)=CCOST*1.2*VC(I)*4640U./3.
CCO(I)=(CCLO(I)+CCEX(I)+CCCC(I)+CUIUFI)*1.15
CTCOST(I)=CCST(I)*VC(I)
URM(I)=UERM*12.*(VC(I)/3.)
URMB(I)=URM(I)*(((1.*EINTD)**ENYB)-1.)/(EINTB*(((1.*EINTD)**ENYB))
1)
PVBI(I)=TCUSI(I)+ORMB(I)
WRITE(6, 17) DIV(I), TCCOS(T), URM(I), PVB(I)
c7 CONTINUE
WRITE(6, 115)
GO TO 23
14 WRITE(6, 43)
43 FORMAT(///47X, 'VOLUME OF BASIN WITH VARIABLE DEPTH (1<-
U<3)', //, 52X,
3'IN COMPOSITE LAND USE AREAS (VAU)',//65X,'DEPTH = ',F=
3.1,'//,37X,'
2DIVERSION VOLUME (IN.), 8X,'VAU (A-C-FT)',//' VM(I)=AR*0.25/12.
VM(2)=AR*0.50/12.
VM(3)=AR*0.75/12.
VM(4)=AR*1.00/12.
VM(5)=AR*1.25/12.
10 887 I=1,N
VC(I)=V(I)*((U.07+(0.92*(CN/100.)))
VAD(I)=VM(I)+(((VC(I)-VM(I))/2.5))*(U=0.5)
WRITE(5,13)DIV(I),VAU(I)
887 CONTINUE
WRITE(5,115)
WRITE(5,38)
10 67 I=1,N
CCCL(I)=CCLAN*1.2*VAU(I)/D
CCAX(I)=EXD*C*VAU(I)*43560./W
CCCC(I)=COCOST*1.2*VAD(I)*4840./W
CCST(I)=(CCLU(I)+CCAX(I)+CCCL(I)+CUJUT*)1.15
CCST(I)=CCST(I)*VAU(I)
URM(I)=URM*1.2*(VAD(I)/D)
URMB(I)=URM(I)*(((1.+EINTB)**ENYB)-1.)/(EINTB*(((1.+EINTB)**ENYB))
1).
PBV(I)=TCCUST(I)+OMB(I)
WRITE(5,17)DIV(I),TCCST(I),URM(I),PBV(I)
67 CONTINUE
GO TO 23
554 IF(D.LT.5.) GO TO 111
C VC(J) COMPUTES THE VOLUME OF PUNUS FIVE FEET DEEP IN C-
C LAND USES
WRITE(5,37)
37 FORMAT(///36X,'VOLUME OF BASIN 5 FEET DEEP IN COMPOSIT-
E LAND USE A
6AREAS (VC)',///,39X,'VOLUME OF BASIN 5 FEET DEEP IN AN I-
MPERVIOUS AA
6TERKED (V)',///,30X,'DIVERSION VOLUME (IN.)',6X,'VC(AC-
=FT)',6X,'V(
MAC-FT)').
1O 3 J=1,N
VC(J)=V(J)*((U.59+(U.37*(CN/100.)))
WRITE(5,11)DIV(J),VC(J),V(J)
11 FORMAT(///35X,3F15.5)
5 CONTINUE
WRITE(5,115)
C CCLD = THE LAND COST EQUATION
C CCEX = THE EXCAVATION COST EQUATION
C CCCC = THE COVER CROP COST EQUATION
CCST = THE CAPITAL COST EQUATION
WRITE(0,58)
38 FORMAT(4/5X,'TOTAL COST OF BASINS',//,33X,'DIVERSION -
VOLUME (IN.)
2',5X,'CAPITAL',1UX,'URM',1UX,'PRESENT VALUE (2UYK, 7%)-
')
DO 2 I=1,N
CCLD(I)=CCLAN*1.2*VC(I)/5.
CCEX(I)=EXCOS*VC(I)*43560./27.
CCCC(I)=CCOST*1.2*VC(I)*4840./5.
CCST(I)=(CCLUD(I)+CCEX(I)+CCCL(I)+CCLUD(I))*1.15
TCCOST(I)=CCST(I)*VC(I)
URM(I)=URM(I)*12. *(VCL(I)/5.)
URMB(I)=URM(I)*(((1.+EINTB)**ENYBJ-1.)/(EINTB*((1.+EIN-
TBJ)**ENYBJ))
8)
PVb(I)=TCCST(I)+URMB(I)
WRITE(6,17)DIV(I),TCCST(I),URM(I),PVb(I)
17 FORMAT(4/40X,F4.2,13X,F15.5,2X,F10.2,10X,F15.5)
2 CONTINUE
WRITE(6,115)
115 FORMAT(4/132('(''))
GO TO 9
111 WRITE(6,777)
777 FORMAT(4/47X,'VOLUME OF BASIN WITH VARIABLE DEPTH (1<-
D<5)',//,52X
6,'IN COMPOSITE LAND USE AREAS (VAD)',//,65X,'DEPTH = '
,F3.1,13X,37X
4,'DIVERSION VOLUME (IN.)',8X,'VAD (AC-FT)',//
VM(1)=AR*0.25/12.
VM(2)=AR*0.50/12.
VM(3)=AR*0.75/12.
VM(4)=AR*1.00/12.
VM(5)=AR*1.25/12.
DO 66 I=1,N
VC(I)=V(I)*(U.59+(U.37*(CN/100.)))
C = DEPTH OF BASIN REQUIRED - FEET
VAD(I)=VM(I)+(((VC(I)-VM(I))/4.))*(D-1.)
WRITE(6,13)DIV(I),VAD(I)
13 FORMAT(4/46X,F4.2,8X,F15.5)
66 CONTINUE
WRITE(6,115)
WRITE(6,58)
DO 22 I=1,N
CCLUD(I)=CCLUD(I)*1.2*VAD(I)/D
CCEX(I)=EXCOS*VAD(I)*43560./27.
CCCL(I)=CCOST*1.2*VAD(I)*4840./D
CCST(I)=(CCLUD(I)+CCEX(I)+CCCL(I)+CCLUD(I))*1.15
TCCOST(I)=CCST(I)*VAD(I)
ORN(I) = CUQ~' 12.·l V.DI IJ/u)
Ur M~ (IJ = UA ~ (I)·(t(ll .+~J ~ T~J"~ N ' R J-I.J/lEI N r A ·(ll .+ t\[­

L

1) PVb(I)=TCCUST(I)+Owmb(I)
WRITE(0,17)DIV(I),1CCOST(I),URM(I),PVb(I)
2 CONTINUE
3 WRITE(0,115)
L THESE ROUTINES ARRANGE THE CROSS AND BASE VARIABLES INTO MATRICES.

9 IF(NOIG.EQ.0) GO TO 779
DO 4 I=1,N
A(I,1)=06YKS(I)
A(I,2)=PVb(I)
4 CONTINUE
WRITE(0,888)
686 FORMAT('I',132('x'),//)
WRITE(0,59)
59 FORMAT('/58X,'SUSPENDED SOLIDS')
CALL PLOT(NO,A,N,M,NL)
WRITE(0,47)
47 FORMAT('I',50X,'SUSPENDED SOLIDS SLOPE & INTERCEPT DATA')
CALL SLOPE(A,N)
DO 77 I=1,N
A(I,1)=06YRK(I)
A(I,2)=PVb(I)
77 CONTINUE
NO=NU+1
WRITE(0,888)
WRITE(0,58)
58 FORMAT('/65X,'BOU')
CALL PLOT(NO,A,N,M,NL)
WRITE(0,46)
46 FORMAT('I',57X,'BOU SLOPE & INTERCEPT DATA')
CALL SLOPE(A,N)
DO 55 I=1,N
A(I,1)=06YKN(I)
A(I,2)=PVb(I)
55 CONTINUE
NO=NU+1
WRITE(0,888)
WRITE(0,57)
57 FORMAT('/62X,'NITRUGEN')
CALL PLOT(NO,A,N,M,NL)
WRITE(0,40)
40 FORMAT('I',53X,'NITRUGEN SLOPE & INTERCEPT DATA')
CALL SLOPE(A,N)
DO 35 1=1,N
A(I,1)=06YRP(I)
A(1,2)=PVR(I)
35 CONTINUE
   NO=NO+1
   WRITE(6,080)
   WRITE(6,53)
53 FORMAT(//52X,'PHOSPHURUS')
   CALL PLOT(NO,A,N,M,NL)
   WRITE(6,48)
40 FORMAT('1',53X,'PHOSPHURUS SLOPE & INTERCEPT DATA')
   CALL SLOPE(A,N)
   RETURN
DO 4441 I=1,N
   YBYR(I)=WBYRB(I)
   YBYRS(I)=WBYRS(I)
   YBYRN(I)=WBYRN(I)
4441 DOYRP(I)=WBYRP(I)
END
SUBROUTINE PLOT
SUBROUTINE PLOT(NO,A,N,M,NL)
C  CARD GROUP # 5
C  VARIABLE DESCRIPTIONS
C  A = MATRIX OF DATA TO BE PLOTEO. FIRST COLUMN REPRESENTS
C  BASE VARIABLE AND SUCCESSIVE COLUMNS ARE THE CROSS VARIABLES.
C  DIMENSION OUT(101),YPR(11),ANG(9),A(10),GRAPH(101,50),XPR(50)
DATA BLANK/' '/DATA ANG/'1','2','3','4','5','6','7','8','9'/
1 FORMAT(1X,'P.V. CUSI - S ',50X,'CHART ',15,//)
2 FORMAT(1H ,F13.2,5X,101A1)
7 FORMAT(1H ,14X,101H.
1.
6 FORMAT(1H0,9X,11F11.4)
   NLL=NL
   IF(NLL) 20,10,20
10 NLL=50
C  PRINT TITLE
20 WRITE(6,1)NO
XSCAL = (A(N) - A(1))/100.
M1 = N+1
YMIN = A(M1)
YMAX = YMIN
M2 = M*1
DO 40 J = M1, M2
   IF (A(J) - YMIN) < 28, 26, 26
26 IF (A(J) - YMAX) > 40, 40, 30
   YMIN = A(J)
   GO TO 40
30 YMAX = A(J)
40 CONTINUE
YSCAL = (YMAX - YMIN) / (FLOAT(NLL - 1))
YPK(1) = A(1)
DO 90 KN = 1, 9
90 YPK(KN+1) = YPK(KN) + XSCAL*10.
YPK(11) = A(N)
DO 101 I = 1, 101
   J = 1, 50
101 GRAPH(I, J) = BLANK
   XR = YMIN
   DO 102 I = 1, NLL
      AI = I - 1
   102 XPK(I) = XR + AI*XSCAL
   XPK(NLL) = YMAX
   MY = M - 1
   DO 103 I = 1, N
      J = 1, MY
103 GRAPH((A(I) - A(1)) / XSCAL + 1., (A(I*J*N) - YMIN) / YSCAL + 1.) = A-
NG(J)
   DO 104 II = 1, NLL
      I = NLL - II + 1
   104 WRITE (6, 2) XPK(I), (GRAPH(J, I), J = 1, 101)
   WRITE (6, 7)
   WRITE (6, 12) (YPK(IP), IP = 1, 11)
   WRITE (6, 2000)
2000 FORMAT (/58A, 'POUNDS PER YEAR REMOVED',/)
   WRITE (6, 115)
115 FORMAT (/6, 132(' '))
RETURN
END

SUBROUTINE SLOPE(CCCCCCCCCCODC01283

SUBROUTINE SLOPE(A,N)
CCC01284
CCC01285
C CARD GROUP # 6
C VARIABLE DESCRIPTIONS
CU1286
C CU1287
C        SL = SLOPE OF A GIVEN PORTION OF THE GRAPH
C        BS = INTERCEPT OF A GIVEN PORTION OF THE GRAPH
C
DIMENSION SL(6), BS(6), A(10)
C
SLOPE = SL = DY/DX
SL(I) = (A(I)-U*)/(A(I)-U*)
WRITE(6,4) SL(I)
4 FORMAT(//58X,'SLOPE(',I,I,'') = ',F15.5)
DO 7 I = 1, N
SL(I+1) = (A(N+1-I)-A(N+1))/A(I+1)-A(1)
IF(SL(I+1)<.LT.,SL(I)) GO TO 77
7 CONTINUE
88 BS(I) = A(N+1)- (SL(I)*A(I))
WRITE(6,11) BS(I), BS(I+1), SL(I+1)
11 FORMAT(//54X,'INTERCEPT(',I2,'') = ',F20.5, //,58X,'SLOPE(',I2,'') = ',F15.5)
7 CONTINUE
DO 5 I = 1, N
BS(I) = A(I) - (SL(I)*A(I))
WRITE(6,5) BS(I)
5 FORMAT(//54X,'INTERCEPT(',I2,'') = ',F20.5)
RETURN
END

SUBROUTINE CVALU(C, FFF)

DIMENSION FFF(6)
I F(C.EQ.U.) GO TO 6
IF(C.EQ.U.1) GO TO 5
IF(C.EQ.U.2) GO TO 4
IF(C.EQ.U.3) GO TO 3
IF(C.EQ.U.4) GO TO 2
IF(C.LT.U.1) GO TO 11
IF(C.LT.U.2) GO TO 10
IF(C.LT.U.3) GO TO 9
IF(C.LT.U.4) GO TO 8
IF(C.LT.U.5) GO TO 7
1 FFF(1) = 1.0
FFF(2) = 1.0
FFF(3) = 1.0
FFF(4) = 1.0
FFF(5) = 1.0
GO TO 22

2 FFF(5) = 0.97
FFF(4) = 0.96
FFF(3) = 0.95
FFF(2) = 0.93
FFF(1) = 0.9

GO TO 22

3 FFF(5) = 0.96
FFF(4) = 0.95
FFF(3) = 0.93
FFF(2) = 0.9
FFF(1) = 0.82

GO TO 22

4 FFF(5) = 0.95
FFF(4) = 0.93
FFF(3) = 0.9
FFF(2) = 0.82
FFF(1) = 0.6

GO TO 22

5 FFF(5) = 0.9
FFF(4) = 0.82
FFF(3) = 0.72
FFF(2) = 0.6
FFF(1) = 0.4

GO TO 22

6 FFF(5) = 0.82
FFF(4) = 0.72
FFF(3) = 0.6
FFF(2) = 0.4
FFF(1) = 0.2

GO TO 22

7 FRAC1 = 1.0 / ((1.0 - C) / 0.2)
FFF(5) = (((1.0 - U - 2.97) / FRAC1) + 1.0
FFF(4) = (((1.0 - U - 2.96) / FRAC1) + 1.0
FFF(3) = (((1.0 - U - 2.95) / FRAC1) + 1.0
FFF(2) = (((1.0 - U - 2.94) / FRAC1) + 1.0
FFF(1) = (((1.0 - U - 2.9) / FRAC1) + 1.0

GO TO 22

8 FRAC1 = 1.0 / ((0.8 - C) / 0.4)
FFF(5) = (((U - 2.97 - 0.96) / FRAC1) + 0.97
FFF(4) = (((U - 2.96 - 0.95) / FRAC1) + 0.96
FFF(3) = (((U - 2.95 - 0.93) / FRAC1) + 0.93
FFF(2) = (((U - 2.93 - 0.9) / FRAC1) + 0.90
FFF(1) = (((U - 2.90 - 0.82) / FRAC1) + 0.90

GO TO 22

9 FRAC1 = 1.0 / ((0.4 - C) / 0.2)
FFF(5) = (((U - 2.96 - 0.95) / FRAC1) + 0.96
FFF(4) = (((U - 2.95 - 0.93) / FRAC1) + 0.95
FFF(3) = (((U - 2.93 - 0.90) / FRAC1) + 0.93
FFF(2) = (((U - 2.90 - 0.82) / FRAC1) + 0.90

GO TO 22
FFF(1) = -((.82-.60)/FRACT)+.82
GO TO 22
10 FRACT = 1. /((0.2-C)/0.1)
FFF(5) = -((.95-.90)/FRACT)+.95
FFF(4) = -((.95-.82)/FRACT)+.93
FFF(5) = -((.90-.72)/FRACT)+.9
FFF(2) = -((.82-.60)/FRACT)+.82
FFF(1) = -((.60-.40)/FRACT)+.6
GO TO 22
11 FRACT = 1. /((0.1-C)/0.1)
FFF(5) = -((.90-.62)/FRACT)+.9
FFF(4) = -((.62-.72)/FRACT)+.62
FFF(3) = -((.72-.60)/FRACT)+.72
FFF(2) = -((.6-.4)/FRACT)+.6
FFF(1) = -((.4-.2)/FRACT)+.4
GO TO 22
22 RETURN
END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC SUBROUTINE TABL01
CCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE TABL01(NPRA,N,CQBYRS,DQBYRS,PR,PRB,CPQBYRS,CPQBYKB,
1,DQBYRD,DQBYRN,DQBYRP,NSUL,ISRS,ISOS,IS0G,ISNI,ISPQ,PYRKS,PYRK,B,PRYN,P
5YRKP,LL,PREM,SSUM,UBFUN,A,ZUMBOX,NCNTW,NPICK,NPICK1,NPICK2,ISLC,ISL
6,ISLB,NPICK3,NPICK4,NBRG,SPLIT,SPLIT1,SPLIT2)
DIMENSION BK(16),AK(18,46),CQBYRS(6),CQBYKB(6),CPQBYRS(6),
1,DQBYRS(6),DQBYRD(6),DQBYRN(6),DQBYRP(6),PR(6),PRB(6),
IBASIS(2V),S
2L(15),PYRKS(6),PYRK,B(6),PYRKN(6),PYRKP(6),PREM(6),ZUMBOX
5V(6),UBFUN(2V),A(5,2),NPICK(6),NPICK1(6),NPICK2(6),N
PIK3(6),NP1
3GK4(6),SPLIT(6),SPLIT1(6),SPLIT2(6)
REAL*8 IKOW(16)
REAL*8 JCOL(31)
REAL*8 JCOLM(31)
DATA IKOW/'SG1PR1','SG2PR1','SG3PR1','SG4PR1','SG5PR1','-SG1PR3',
'2G2PR3','SG3PR3','SG4PR3','SG5PR3','SEG0UM','SG1PR1',
DATA JCOL/'X11','X12','X13','X14','X15','X31','X32','X33','X34','X
135','SUPPL1','SLK1','SLK2','SLK3','SLK4','SLK5','SLK6','-SLK7','SLK
2d','SLK9','SLK10','JCENT=1
LL = 0
9905 MEQ = (5*NPR A) + 1
NVAR = ((5*NPR A) * 2) + 1
NM = NVAR + MEQ
MPLUS2 = MEQ + 2
00 95 I = 1, MPLUS2
DK(I) = 0.
I B A S I S ( I ) = 0
00 95 J = 1, NM
95 AK ( I , J ) = 0.
IF ( IS S , EQ . 0 ) GO TO 53
IF ( IS B . EQ . 0 ) GO TO 52
IF ( IS N I . EQ . 0 ) GO TO 51
IF ( IS P H . EQ . 0 ) GO TO 50
53 BK(1) = CBQYRS(1)
BK(2) = CBQYRS(2) - CBQYRS(1)
BK(3) = CBQYRS(3) - CBQYRS(2)
BK(4) = CBQYRS(4) - CBQYRS(3)
BK(5) = CBQYRS(5) - CBQYRS(4)
IF ( MEQ . EQ . 0 , 0 ) GO TO 49
BK(6) = DQBYRS(1)
BK(7) = DQBYRS(2) - DQBYRS(1)
BK(8) = DQBYRS(3) - DQBYRS(2)
BK(9) = DQBYRS(4) - DQBYRS(3)
BK(10) = DQBYRS(5) - DQBYRS(4)
IF ( JC NT . EQ . 1 ) GO TO 1
IF ( JC NT . EQ . 2 ) GO TO 2
IF ( JC NT . EQ . 3 ) GO TO 3
IF ( JC NT . EQ . 4 ) GO TO 4
IF ( JC NT . EQ . 5 ) GO TO 55
1 BK(MEQ) = PYRKS(1)
JC NT = JC NT + 1
LL = LL + 1
SSUM(NCNTW, LL) = PYRKS(1)
GO TO 445
2 BK(MEQ) = PYRKS(2)
JC NT = JC NT + 1
LL = LL + 1
SSUM(NCNTW, LL) = PYRKS(2)
GO TO 445
3 BK(MEQ) = PYRKS(3)
JC NT = JC NT + 1
LL = LL + 1
SSUM(NCNTW, LL) = PYRKS(3)
GO TO 445
4 BK(MEQ) = PYRKS(4)
JC NT = JC NT + 1
LL = LL + 1
SSUM(NCNTW, LL) = PYRKS(4)
GO TO 445
555 \( dK(\text{ME}Q)=\text{PYRS}(5) \)
\( JCN\text{T}=JCN\text{T}+1 \)
\( LL=LL+1 \)
\( S\text{SUM}(\text{NCN}\text{T}W,LL)=\text{PYRS}(5) \)
\( \text{GO TO 445} \)
409 \( \text{IRUN}(6)=\text{IRUN}(11) \)
445 IF(\( \text{ME}Q,\text{NE},6) \) \( \text{GO TO 55} \)
\( J\text{CUL}(6)=J\text{CUL}(11) \)
\( J\text{CUL}(7)=J\text{CUL}(12) \)
\( J\text{CUL}(8)=J\text{CUL}(13) \)
\( J\text{CUL}(9)=J\text{CUL}(14) \)
\( J\text{CUL}(10)=J\text{CUL}(15) \)
\( J\text{CUL}(11)=J\text{CUL}(16) \)
\( SL(1)=(PV(1)-0.)/(CQBYRS(1)-0.) \)
\( \text{DO 9981} \)
9981 \( SL(I+1)=(PV(I+1)-PV(I))/(CQBYRS(I+1)-CQBYRS(I)) \)
\( \text{DO 5} \)
5 \( I=1,5 \)
\( \text{AK(\( \text{ME}Q+1,I)=-\text{SL}(I) \)} \)
\( \text{DO 6} \) \( I=6,\text{NVAK} \)
6 \( \text{AK(\( \text{ME}Q+1,I)=0.} \)
\( \text{GO TO 60} \)
55 \( SL(1)=(PV(1)-0.)/(CQBYRS(1)-0.) \)
\( \text{DO 4242} \)
4242 \( I=1,4 \)
\( SL(I+1)=(PV(I+1)-PV(I))/(CQBYRS(I+1)-CQBYRS(I)) \)
\( \text{IF(SL(I+1),LI,SL(I)) \text{GO TO 765}} \)
\( \text{GO TO 4242} \)
765 \( SL(I+1)=SL(I)+1. \)
\( \text{4242 CONTINUE} \)
\( SL(5)=(PV(5)-0.)/(DQBYRS(5)-0.) \)
\( \text{DO 4243} \)
4243 \( I=6,9 \)
\( SL(I+1)=(PV(I-N+1)-PV(I-N))/(DQBYRS(I-N+1)-DQBYRS(I-N)) \)
\( \text{IF(SL(I+1),LI,SL(I)) \text{GO TO 654}} \)
\( \text{GO TO 4243} \)
654 \( SL(I+1)=SL(I)+1. \)
\( \text{4243 CONTINUE} \)
\( \text{GO TO 7779} \)
52 \( dK(1)=CQBYKB(1) \)
\( dK(2)=CQBYKB(2)-CQBYKB(1) \)
\( dK(3)=CQBYKB(3)-CQBYKB(2) \)
\( dK(4)=CQBYKB(4)-CQBYKB(3) \)
\( dK(5)=CQBYKB(5)-CQBYKB(4) \)
\( \text{IF(MEQ.EQ.,6)GO TO 901} \)
\( dK(6)=UQBYKB(1) \)
\( dK(7)=UQBYKB(2)-UQBYKB(1) \)
\( dK(8)=UQBYKB(3)-UQBYKB(2) \)
\( dK(9)=UQBYKB(4)-UQBYKB(3) \)
\( dK(10)=DQBYRD(5)-DQBYRD(4) \)
\( \text{IF(JCN\text{T},EQ,1) \text{GO TO 66}} \)
\( \text{IF(JCN\text{T},EQ,2) \text{GO TO 77}} \)
IF(JCNT.EQ.3) GO TO 68
IF(JCNT.EQ.4) GO TO 99
IF(JCNT.EQ.5) GO TO 10
66 BK(MEQ)=PYKRB(1)
       JCNT=JCNT+1
       LL=LL+1
       SSM(NCNT,LL)=PYKRB(1)
       GO TO 223
77 BK(MEQ)=PYKRB(2)
       JCNT=JCNT+1
       LL=LL+1
       SSM(NCNT,LL)=PYKRB(2)
       GO TO 223
88 BK(MEQ)=PYKRB(3)
       JCNT=JCNT+1
       LL=LL+1
       SSM(NCNT,LL)=PYKRB(3)
       GO TO 223
99 BK(MEQ)=PYKRB(4)
       JCNT=JCNT+1
       LL=LL+1
       SSM(NCNT,LL)=PYKRB(4)
       GO TO 223
10 BK(MEQ)=PYKRB(5)
       JCNT=JCNT+1
       LL=LL+1
       SSM(NCNT,LL)=PYKRB(5)
       GO TO 223
901 IRUM(6)=IRUM(11)
223 IF(MEQ . ME . 6) GO TO 903
       JCOL(6)=JCOL(11)
       JCOL(7)=JCOL(12)
       JCOL(8)=JCOL(13)
       JCOL(9)=JCOL(14)
       JCOL(10)=JCOL(15)
       JCOL(11)=JCOL(16)
       SL(1)=(PV(1)-0.)/(CYKRB(1)-V.)
60 904 I=1,4
904 SL(I+1)=(PV(I+1)-PV(1))/(CYKRB(I+1)-CYKRB(1))
60 905 I=1,5
905 AK(MEQ+1,I)=SL(I)
60 906 I=6,NVAR
906 AK(MEQ+1,I)=V.
GO TO 60
903 SL(1)=(PV(1)-0.)/(CYKRB(1)-V.)
60 907 I=1,4
907 SL(I+1)=(PV(I+1)-PV(1))/(CYKRB(I+1)-CYKRB(I))
IF(SL(I+1).LT.SL(I)) GO TO 376
GO TO 907
878 SL(I+1)=SL(I)+1.
\[ S_L(n) = \frac{(PVH(1) - v)}{\sum_{i=2}^{n}(PVH(i) - PVH(i-1))} \]

1. \( S_L(n+1) = \frac{(PVH(1 - n + 1) - PVH(1 - n))}{\sum_{i=2}^{n}(PVH(1 - n + 1) - PVH(i - 1))} \)

2. \( S_L(n+1) = S_L(n) \cdot \text{If } S_L(n+1) \geq 10.543 \text{ then } T = 90^\circ \)

3. \( S_L(n+1) = S_L(n) \cdot \text{If } S_L(n+1) < 10.543 \text{ then } T = 77^\circ \)

4. \( S_L(n+1) = \text{PYRN}(1) \)

5. \( S_L(n+1) = \text{PYRN}(2) \)

6. \( S_L(n+1) = \text{PYRN}(3) \)

7. \( S_L(n+1) = \text{PYRN}(4) \)

8. \( S_L(n+1) = \text{PYRN}(5) \)

9. \( S_L(n+1) = \text{PYRN}(6) \)

10. \( S_L(n+1) = \text{PYRN}(7) \)

11. \( S_L(n+1) = \text{PYRN}(8) \)

12. \( S_L(n+1) = \text{PYRN}(9) \)

13. \( S_L(n+1) = \text{PYRN}(10) \)

14. \( S_L(n+1) = \text{PYRN}(11) \)

15. \( S_L(n+1) = \text{PYRN}(12) \)

16. \( S_L(n+1) = \text{PYRN}(13) \)

17. \( S_L(n+1) = \text{PYRN}(14) \)

18. \( S_L(n+1) = \text{PYRN}(15) \)
GO TO 225
201 IRW(6)=IRW(11)
225 IF(MEQ,NE,6) GO TO 803
JCUL(6)=JCUL(11)
JCUL(7)=JCUL(12)
JCUL(8)=JCUL(13)
JCUL(9)=JCUL(14)
JCUL(10)=JCUL(15)
JCUL(11)=JCUL(16)
SL(1)=(PV(1)-0.)/(CWBYRN(1)-0.)
GO 804 I=1,4
004 SL(I+1)=(PV(I+1)-PV(I))/(CWBYRN(I+1)-CWBYRN(I))
GO 805 I=1,5
005 AK(MEQ+1,I)=-SL(I)
GO 806 I=6,NVAK
306 AK(MEQ+1,I)=0.
GO TU 60
805 SL(I)=(PV(I)-0.)/(CWBYRN(I)-0.)
GO 807 I=1,4
SL(I+1)=(PV(I+1)-PV(I))/(CWBYRN(I+1)-CWBYRN(I))
IF(SL(I+1).LT.SL(I)) GO TO 987
GO TU 807
987 SL(I+1)=SL(I)+1.
807 CONTINUE
SL(6)=(PV(1)-0.)/(DWBYRN(1)-0.)
GO 808 I=6,9
SL(I+1)=(PV(B(I-N)-PV(B(I-N)))/(DWBYRN(I-N+1)-DWBYRN(I-N))
IF(SL(I+1).LT.SL(I)) GO TO 432
GO TU 808
432 SL(I+1)=SL(I)+1.
608 CONTINUE
GO TU 7779
50 AK(1)=CWBYRP(1)
AK(2)=CWBYRP(2)-CWBYRP(1)
AK(3)=CWBYRP(3)-CWBYRP(2)
AK(4)=CWBYRP(4)-CWBYRP(3)
AK(5)=CWBYRP(5)-CWBYRP(4)
IF(MEQ,NE,6) GO TU 701
AK(6)=DWBYRP(1)
AK(7)=DWBYRP(2)-DWBYRP(1)
AK(8)=DWBYRP(3)-DWBYRP(2)
AK(9)=DWBYRP(4)-DWBYRP(3)
AK(10)=DWBYRP(5)-DWBYRP(4)
IF(JCNT.EQ.1) GO TU 16
IF(JCNT.EQ.2) GO TU 18
IF(JCNT.EQ.3) GO TU 119
IF(JCNT.EQ.4) GO TU 820
IF(JCNT.EQ.5) GO TU 222
16 AK(MEQ)=PYKRP(1)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNTW,LL)=PYRKP(1)
GO TO 503
18 dK(MEQ)=PYRKP(2)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNTW,LL)=PYRKP(2)
GO TO 503
119 dK(MEQ)=PYRKP(3)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNTW,LL)=PYRKP(3)
GO TO 503
620 dK(MEQ)=PYRKP(4)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNTW,LL)=PYRKP(4)
GO TO 503
222 dK(MEQ)=PYRKP(5)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNTW,LL)=PYRKP(5)
GO TO 503
701 IRUW(6)=IRUW(11)
503 IF(MEQ.NE.6) GO TO 703
JCUL(6)=JCUL(11)
JCUL(7)=JCUL(12)
JCUL(8)=JCUL(13)
JCUL(9)=JCUL(14)
JCUL(10)=JCUL(15)
JCUL(11)=JCUL(16)
SL(I)=(PV(I)-O.)/(CQBYRP(I)-U.)
DO 704 I=1,4
704 SL(I+1)=(PV(I+1)-PV(I))/(CQBYRP(I+1)-CWBYRP(I))
DO 705 I=1,5
705 AK(MEQ+1,I)=-SL(I)
DO 706 I=6,NVAR
706 AK(MEQ+1,I)=0.
GO TO 60
703 SL(I)=(PV(I)-O.)/(CQBYRP(I)-U.)
DO 707 I=1,4
SL(I+1)=(PV(I+1)-PV(I))/(CQBYRP(I+1)-CWBYRP(I))
IF(SL(I+1).LT.SL(I)) GO TO 1096
GO TO 707
1096 SL(I+1)=SL(I)+1.
707 CONTINUE
SL(I)=(PV(I)-O.)/(CWBYRP(I)-O.)
DO 708 I=1,N
SL(I+1)=(PV(I-N+1)-PV(I-N))/(DQBYRP(I-N+1)-DQBYRP(I-N))
IF(SL(I+1)=SL(I)*1) GO TO 321
GO TO 706
321 SL(I+1)=SL(I)*1.
706 CONTINUE
7779 DO 7 I=1,5
7 AK(MEQ+1,I)=-SL(I)
DO 8 I=6,10
8 AK(MEQ+1,I)=-SL(I)
DO 9 I=11,NVAR
9 AK(MEQ+1,I)=0.
60 MEQ1=MEQ-1
DO 25 II=1,MEQ1
AK(II,II)=1.
AK(II,II+MEQ)=1.
AK(MEQ,II)=1.
25 CONTINUE
AK(MEQ,MEQ)=-1.
MEQ3=MEQ+1
IF(NSOL.EQ.0)GO TO 400
WRITE(*,28)
28 FORMAT('1',132(''),//,'10X,'INITIAL TABLEAU FROM SUBRO-
UTINE TABLU')
1//)
DO 45 I=1,MEQ3
DO 45 J=1,NVAR
WRITE(*,20)II,J,AK(I,J)
40 FORMAT(55X,'AK(',I2,',',12,) = ',F11.2,/')
45 CONTINUE
WRITE(*,29)
29 FORMAT(1,132(''),//)
400 CALL LP6GUGU(MEQ,NVAR,IKOW,AK,HK,JCUH,SL,NM,MPLUS2,LL,IS-SS,ISB0,NBM
41,ISPH,PREM,SSUM,OFUN,NCNTW,A,N,ZIMB0X,NPICK,NPICK1,N-1
PICK2,ISLC,I
3SL3,ISL8,JCOLM,NPICK3,NPICK4,NBIG,SPLIT,SPLIT1,SPLIT2)
IF(JCN1.EQ.0)GO TO 21
GO TO 9905
21 RETURN
END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCC SUBROUTINE TABLU2 CCCCCCCCCCCCCCCCCCCCCC-
-CCCCCCCCCC01752
SUBROUTINE TABLU2(NPRA,N,SUBYR3,SBYKR,P,VS,NSOL,IS33
2,ISB0,ISMN,ISPH,PKRS,PKR0,PKR1,PKR2,LL,PREM,SSUM,OFUN,A,ZIMB0
2X,NCNTW,NPICK,NPICK1,NPICK2,ISLC,ISLS,ISL8,JCOLM,NPICK3,NPICK-
X4,NBIG,SPLIT
31,SPLIT1,SPLIT2)
DIMENSION UK(1d),AK(18,40),SUBYRP(0),SUBYRN(0),SUBYR0(0)
SUBYR0(0)
1, PV$5(6), 1BASIS(20), SL(15), PYK5(6), PYKH(6), PYKRN(6), PY-
YRNP(6), PREM
2(6), 1IMBOX(6), SSUM(20, 6), 06FUN(20, 6), A(5, 2), NPRICK(6), N-
PRICK1(6), NPRICK2(6), NPRICK3(6), NPRICK4(6), SPLIT(6), SPLIT1(6), SPLIT2(6-
)
REAL*8 IKOW(16)
REAL*8 JCOL(31)
REAL*8 JCOLM(31)
DATA IKOW/ 'SG1PR2', 'SG2PR2', 'SG3PR2', 'SG4PR2', 'SG5PR2' -
, 'SEGSUM', /2', ' /0
DATA JCOL/ 'X21', 'X22', 'X23', 'X24', 'X25', 'SURPL1', 'SLK1-
', 'SLK2', 'SLK4', 'SLK5', '/1', ' /0
JCNT=1
LL=0
120 MEW=(5*NPRA)+1
NVAR=((5*NPRA)*2)+1
NM=NVAR+MEW
MPLUS2=MEW+2
DO 95 I=1, MPLUS2
95 AK(I,J)=0.
IBASIS(I)=0
DO 95 J=1, NM
95 AK(I,J)=0.
IF(ISSS.EQ.0) GO TO 53
IF(ISBU.EQ.0) GO TO 52
IF(ISNI.EQ.0) GO TO 51
IF(ISPH.EQ.0) GO TO 50
53 AK(1)=SQBYRS(1)
AK(2)=SQBYRS(2)-SQBYRS(1)
AK(3)=SQBYRS(3)-SQBYRS(2)
AK(4)=SQBYRS(4)-SQBYRS(3)
AK(5)=SQBYRS(5)-SQBYRS(4)
IF(JCNT.EQ.1) GO TO 1
IF(JCNT.EQ.2) GO TO 2
IF(JCNT.EQ.3) GO TO 3
IF(JCNT.EQ.4) GO TO 4
IF(JCNT.EQ.5) GO TO 5
1 BK(MEQ)=PYK5(1)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNTW, LL) = PYK5(1)
GO TO 114
2 BK(MEQ)=PYK5(2)
JCNT=JCNT+1
LL=LL+1
SSUM(NCINTW,LL)=PYRKS(2)
GO TO 114
3 BK(MEO)=PYRKS(3)
  JCNT=JCNT+1
  LL=LL+1
  SSUM(NCINTW,LL)=PYRKS(3)
  GO TO 114
4 BK(MEO)=PYRKS(4)
  JCNT=JCNT+1
  LL=LL+1
  SSUM(NCINTW,LL)=PYRKS(4)
  GO TO 114
5 BK(MEO)=PYRKS(5)
  JCNT=JCNT+1
  LL=LL+1
  SSUM(NCINTW,LL)=PYRKS(5)
  GO TO 114
114 SL(I)=(PVS(I)-U.)/(SUBYRS(I)-O.)
DO 501 I=1,4
  SL(I+1) =(PVS(I+1)-PVS(I))/(SUBYRS(I+1)-SUBYRS(I))
  IF(SL(I+1)<SL(I)) GO TO 876
GO TO 501
076 SL(I+1)=SL(I)+1.
501 CONTINUE
DO 506 I=1,5
506 AK(MEO+1,I)=-SL(I)
DO 505 I=6,NVAR
505 AK(MEO+1,I)=U.
GO TO 60
52 BK(1)=SQBYKB(1)
  BK(2)=SQBYKB(2)-SQBYKB(1)
  BK(3)=SQBYKB(3)-SQBYKB(2)
  BK(4)=SQBYKB(4)-SQBYKB(3)
  BK(5)=SQBYKB(5)-SQBYKB(4)
  IF(JCNT.EQ.1) GO TO 0
  IF(JCNT.EQ.2) GO TO 7
  IF(JCNT.EQ.3) GO TO 6
  IF(JCNT.EQ.4) GO TO 9
  IF(JCNT.EQ.5) GO TO 10
6 BK(MEO)=PYRRB(1)
  JCNT=JCNT+1
  LL=LL+1
  SSUM(NCINTW,LL)=PYRRB(1)
  GO TO 123
7 BK(MEO)=PYRRB(2)
  JCNT=JCNT+1
  LL=LL+1
  SSUM(NCINTW,LL)=PYRRB(2)
  GO TO 123
8 BK(MEO)=PYRRB(3)
JCNT=JCNT+1
LC=LL+1
SSUM(NCNTW,LL)=PYRB(3)
GO TO 123
9 BK(MEQ)=PYRB(4)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNTW,LL)=PYRB(4)
GO TO 123
10 BK(MEQ)=PYRB(5)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNTW,LL)=PYRB(5)
GO TO 123
123 SL(1)=(PVS(1)-0.)/(SUBYRB(1)-0.)
DO 502 I=1,4
SL(I+1)=(PVS(I+1)-PVS(I))/(SUBYRB(I+1)-SUBYRB(I))
IF(SL(I+1).LT.SL(I)) GO TO 765
GO TO 502
765 SL(I+1)=SL(I)+1.
502 CONTINUE
DO 503 I=1,5
503 AK(MEQ+1,I)=-SL(I)
DO 504 I=6,NVAR
504 AK(MEQ+1,I)=0.
GO TO 60
51 BK(1)=SUBYRN(1)
BK(2)=SUBYRN(2)-SUBYRN(1)
BK(3)=SUBYRN(3)-SUBYRN(2)
BK(4)=SUBYRN(4)-SUBYRN(3)
BK(5)=SUBYRN(4)-SUBYRN(5)
IF(JCNT.EQ.1) GO TO 11
IF(JCNT.EQ.2) GO TO 12
IF(JCNT.EQ.3) GO TO 13
IF(JCNT.EQ.4) GO TO 14
IF(JCNT.EQ.5) GO TO 15
11 BK(MEQ)=PYRN(1)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNTW,LL)=PYRN(1)
GO TO 125
12 BK(MEQ)=PYRN(2)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNTW,LL)=PYRN(2)
GO TO 125
13 BK(MEQ)=PYRN(3)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNTW,LL)=PYRN(3)
14 0K(MEQ) = PYRN(4)
   JCNT=JCNT+1
   LL=LL+1
   SSUM(NCNTW,LL)=PYRN(4)
   GO TO 125
15 0K(MEQ) = PYRN(5)
   JCNT=JCNT+1
   LL=LL+1
   SSUM(NCNTW,LL)=PYRN(5)
   GO TO 125
125 SL(1)=(PVs(1)-U.)/(SUBYN(1)-0.)
   u0   5u7 I=1,4
   SL(I+1)=(PVs(I+1)-PVs(I))/(SUBYN(I+1)-SUBYN(I))
   IF(SL(I+1).LT.SL(I)) GO TO 967
   GO TO 507
987 SL(I+1)=SL(I)+1.
507 CONTINUE
   u0   5u8 I=1,5
508 AK(MEQ+1,I)=-SL(1)
   u0   5u9 I=6,NVAR
509 AK(MEQ+1,I)=0.
   GO TO 60
50 0K(1)=SQRKY(1)
   0K(2)=SQRK(2)-SQRKY(1)
   0K(3)=SQRKY(3)-SQRKY(2)
   0K(4)=SQRKY(4)-SQRK(3)
   0K(5)=SQRKY(5)-SQRK(4)
   IF(JCNT.EQ.1) GO TO 16
   IF(JCNT.EQ.2) GO TO 18
   IF(JCNT.EQ.3) GO TO 19
   IF(JCNT.EQ.4) GO TO 220
   IF(JCNT.EQ.5) GO TO 222
10 0K(MEQ) = PYRP(1)
   JCNT=JCNT+1
   LL=LL+1
   SSUM(NCNTW,LL)=PYRP(1)
   GO TO 127
13 0K(MEQ) = PYRP(2)
   JCNT=JCNT+1
   LL=LL+1
   SSUM(NCNTW,LL)=PYRP(2)
   GO TO 127
14 0K(MEQ) = PYRP(3)
   JCNT=JCNT+1
   LL=LL+1
   SSUM(NCNTW,LL)=PYRP(3)
   GO TO 127
220 0K(MEQ) = PYRP(4)
   JCNT=JCNT+1
LL=LL+1
"SSUM(NCNW,LL)=PRYRP(4)
GO TO 127
222 AK(MEQ)=PRYRP(5)
JCNT=JCNT+1
LL=LL+1
"SSUM(NCNW,LL)=PRYRP(5)
GO TO 127
127 SL(1)=(PVS(1)-0.)/(SUBYRP(1)-0.)
DO 510 I=1,4
SL(I+1)=(PVS(I+1)-PVS(I))/(SUBYRP(I+1)-SUBYRP(I))
IF(SL(I+1)<0.)*.LT.SL(I)) GO TO 1098
GO TO 510
1098 SL(I+1)=SL(I)+1.
510 CONTINUE
DO 511 I=1,5
511 AK(MEQ+1,I)=-SL(I)
DO 512 J=5,NVAR
512 AK(MEQ+1,I)=0.
60 MEQI=MEQ-1
DO 25 II=1,MEQ1
AK(II,II)=1.
AK(II,II+MEQ1)=1.
AK(MEQ,II)=1.
25 CONTINUE
AK(MEQ,MEQ)=-1.
MEQ3=MEQ+1
IF(NSOL*EQ.0) GO TO 400
WRITE(6,28)
28 FORMAT('1',132('*'),//'1UX,'INITIAL TABLEAU FROM SUBROUTINE TABLO'
1,//)
DO 45 I=1,MEQ3
DO 45 J=1,NVAR
WRITE(6,20),I,J,AK(I,J)
40 FORMAT(55X,'AK(',I2,',',I2,') = ',F11.2,/) 
45 CONTINUE
WRITE(6,29)
29 FORMAT(//'1',132('*'),//)
400 CALL LPGUGU(MEW,NVAR,IROW,AK,BK,JCUM,SL,NM,NPLUS2,LL,I-
SS,ISD,ISN
41,ISPH,PHEM,SSUM,OBFUN,NCNW,A,N,ZIMOx,NPICK,NPICK1,N-
PICK2,ISLC,I
3SL,SISL,JCUM,NPICK3,NPICK4,NOIG,SPLIT,SPLIT1,SPLIT2)
IF(JCNT.EQ.0.0) GO TO 21
GO TO 120
21 RETURN
END

CCCCCCCCCCCLLLLLLCLCCCLCCCLCCCLCE
CCCLLCLCCCLCCCLCCCLCCCLCCCLCCCLC
1997
6K(6)=UOBYRS(1)
6K(7)=UOBYRS(2)-UOBYRS(1)
6K(8)=UOBYRS(3)-UOBYRS(2)
6K(9)=UOBYRS(4)-UOBYRS(3)
6K(10)=UOBYRS(5)-UOBYRS(4)
IF(JCNT.EQ.1) GO TO 1
IF(JCNT.EQ.2) GO TO 2
IF(JCNT.EQ.3) GO TO 3
IF(JCNT.EQ.4) GO TO 4
IF(JCNT.EQ.5) GO TO 555
1 6K(MEO)=PYRRS(1)
   JCNT=JCNT+1
   LL=LL+1
   SSMU(NCNTW,LL)=PYRRS(1)
   GO TO 122
2 6K(MEO)=PYRRS(2)
   JCNT=JCNT+1
   LL=LL+1
   SSMU(NCNTW,LL)=PYRRS(2)
   GO TO 122
3 6K(MEO)=PYRRS(3)
   JCNT=JCNT+1
   LL=LL+1
   SSMU(NCNTW,LL)=PYRRS(3)
   GO TO 122
4 6K(MEO)=PYRRS(4)
   JCNT=JCNT+1
   LL=LL+1
   SSMU(NCNTW,LL)=PYRRS(4)
   GO TO 122
555 6K(MEO)=PYRRS(5)
   JCNT=JCNT+1
   LL=LL+1
   SSMU(NCNTW,LL)=PYRRS(5)
   GO TO 122
409 1RUN(6)=1RUN(11)
122 IF(MEO.NE.0) GO TO 55
   JCUL(6)=JCUL(11)
   JCUL(7)=JCUL(12)
   JCUL(8)=JCUL(13)
   JCUL(9)=JCUL(14)
   JCUL(10)=JCOL(15)
   JCUL(11)=JCUL(16)
   SL(1)=PV5(1)-PV5(1)/(SUBYRS(1)-0.)
   UD 9981 I=1.4
9981 SL(I+1)=(PV5(I+1)-PV5(1))/(SUBYRS(I+1)-SUBYRS(1))
   UD 5 I=1,5
3 AK(MEO+1,I)=-SL(1)
   UD 6 I=6,NVAR
6 AK(MEO+1,I)=0.
56 \text{SL}(1) = (P_{VS}(1) - v_1)/(S_{\text{WR}}(1) - 1) \\
\text{IF} (SL(1), L1, SL(1)) \text{ GO TO 705} \\
\text{GO To 4242} \\
\text{THO} SL(1) = SL(T) + 1. \\
\text{GO To 4242} \\
\text{CON} \text{INUE} \\
\text{SL}(2) = (P_{VS}(1) - v_2)/(S_{\text{WR}}(1) - 1) \\
\text{IF} (SL(2), L1, SL(2)) \text{ GO TO 876} \\
\text{GO To 4245} \\
\text{GO To 7779} \\
56 \text{BK}(1) = 0\text{BUYR}(1) \\
\text{BK}(2) = 0\text{BUYR}(2) - 0\text{BUYR}(1) \\
\text{BK}(3) = 0\text{BUYR}(3) - 0\text{BUYR}(2) \\
\text{BK}(4) = 0\text{BUYR}(4) - 0\text{BUYR}(3) \\
\text{BK}(5) = 0\text{BUYR}(5) - 0\text{BUYR}(4) \\
\text{IF} (\text{SUM}_E, 1) \text{ GO To 90} \\
\text{BK}(6) = 0\text{BUYR}(1) \\
\text{BK}(7) = 0\text{BUYR}(2) - 0\text{BUYR}(1) \\
\text{BK}(8) = 0\text{BUYR}(3) - 0\text{BUYR}(2) \\
\text{BK}(9) = 0\text{BUYR}(4) - 0\text{BUYR}(3) \\
\text{BK}(10) = 0\text{BUYR}(5) - 0\text{BUYR}(4) \\
\text{IF} (\text{SUM}_E, 1) \text{ GO To 6} \\
\text{IF} (\text{SUM}_E, 2) \text{ GO To 777} \\
\text{IF} (\text{SUM}_E, 3) \text{ GO To 99} \\
\text{IF} (\text{SUM}_E, 4) \text{ GO To 99} \\
\text{IF} (\text{SUM}_E, 5) \text{ GO To 110} \\
65 \text{BK}(10) = \text{PRYR}(1) \\
\text{JCN} = \text{JCN} + 1 \\
\text{LE} = \text{LE} + 1 \\
\text{SU}([\text{LE}], \text{LL}) = \text{PRYR}(1) \\
\text{GO To 210} \\
177 \text{BK}(10) = \text{PRYR}(2) \\
\text{JCN} = \text{JCN} + 1 \\
\text{LE} = \text{LE} + 1 \\
\text{SU}([\text{LE}], \text{LL}) = \text{PRYR}(2) \\
\text{GO To 210} \\
273 \text{BK}(10) = \text{PRYR}(3) \\
\text{JCN} = \text{JCN} + 1 \\
\text{LE} = \text{LE} + 1 \\
\text{SU}([\text{LE}], \text{LL}) = \text{PRYR}(3) \\
\text{GO To 210} \\
49 \text{BK}(10) = \text{PRYR}(4) \\
\text{JCN} = \text{JCN} + 1
LL = LL + 1
\textcolor{red}{\textbf{GO TO 215}}

\textcolor{blue}{110} \textcolor{red}{SK(MEQ) = PYRKB(5)}
\textcolor{red}{JCNT = JCNT + 1}
\textcolor{red}{LL = LL + 1}
\textcolor{red}{SSUM(NCNTW, LL) = PYRKB(5)}
\textcolor{red}{GO TU 215}

\textcolor{blue}{10} \textcolor{red}{IRUN(6) = IRUN(11)}
\textcolor{red}{LL = LL + 1}

\textcolor{blue}{215} \textcolor{red}{IF(MEQ . NE. 6) GU TO 61}
\textcolor{red}{JCUL(6) = JCUL(11)}
\textcolor{red}{JCUL(7) = JCUL(12)}
\textcolor{red}{JCUL(8) = JCUL(13)}
\textcolor{red}{JCUL(9) = JCUL(14)}
\textcolor{red}{JCUL(10) = JCUL(15)}
\textcolor{red}{JCUL(11) = JCUL(16)}
\textcolor{red}{SL(1) = (PV5(1) - 0.)/(SUbYRb(1) - 0.)}
\textcolor{red}{GO TO 79}

\textcolor{blue}{70} \textcolor{red}{SL(I+1) = \textcolor{red}{(PV5(I+1) - PV5(I))/(SUbYRb(I+1) - SUbYRb(I))}}
\textcolor{red}{GO TO 73}

\textcolor{blue}{73} \textcolor{red}{AK(MEQ+1, I) = -SL(1)}
\textcolor{red}{GO TO 76}

\textcolor{blue}{76} \textcolor{red}{AK(MEQ+1, I) = 0.}
\textcolor{red}{GO TO 60}

\textcolor{blue}{61} \textcolor{red}{SL(1) = (PV5(1) - 0.)/(SUbYRb(1) - 0.)}
\textcolor{red}{GO TO 68}

\textcolor{blue}{14} \textcolor{red}{SL(I+1) = SL(I) + 1.}

\textcolor{blue}{80} \textcolor{red}{CONTINUE}
\textcolor{red}{SL(6) = (PVb(1) - 0.)/(DwBYRb(1) - 0.)}
\textcolor{red}{GO TO 987}

\textcolor{blue}{987} \textcolor{red}{SL(I+1) = SL(I) + 1.}

\textcolor{blue}{33} \textcolor{red}{CONTINUE}
\textcolor{red}{GO TO 7779}

\textcolor{blue}{51} \textcolor{red}{SK(1) = SubYRN(1)}
\textcolor{red}{SK(2) = SubYRN(2) - SubYRN(1)}
\textcolor{red}{SK(3) = SubYRN(3) - SubYRN(2)}
\textcolor{red}{SK(4) = SubYRN(4) - SubYRN(3)}
\textcolor{red}{SK(5) = SubYRN(5) - SubYRN(4)}
\textcolor{red}{IF(MEQ . EQ. 6) GU TO 11}
\textcolor{red}{SK(6) = UbYRN(1)}
\textcolor{red}{SK(7) = UbYRN(2) - UbYRN(1)}
\textcolor{red}{SK(8) = UbYRN(3) - UbYRN(2)}
BK(9) = DIBYRN(4) - DDBYRN(3)
BK(10) = DIBYRN(5) - DDBYRN(4)
IF (JCNT.EQ.1) GO TO 111
IF (JCNT.EQ.2) GO TO 112
IF (JCNT.EQ.3) GO TO 113
IF (JCNT.EQ.4) GO TO 114
IF (JCNT.EQ.5) GO TO 115
111 BK(MEQ) = PYRRN(1)
   JCNT = JCNT + 1
   LL = LL + 1
   S$UM(NCNTW, LL) = PYRRN(1)
   GO TO 210
112 BK(MEQ) = PYRRN(2)
   JCNT = JCNT + 1
   LL = LL + 1
   S$UM(NCNTW, LL) = PYRRN(2)
   GO TO 210
113 BK(MEQ) = PYRRN(3)
   JCNT = JCNT + 1
   LL = LL + 1
   S$UM(NCNTW, LL) = PYRRN(3)
   GO TO 210
114 BK(MEQ) = PYRRN(4)
   JCNT = JCNT + 1
   LL = LL + 1
   S$UM(NCNTW, LL) = PYRRN(4)
   GO TO 210
115 BK(MEQ) = PYRRN(5)
   JCNT = JCNT + 1
   LL = LL + 1
   S$UM(NCNTW, LL) = PYRRN(5)
   GO TO 210
116 JRUN(6) = JRUN(11)
210 IF (MEQ .NE. b) GO TO 62
   JCUL(6) = JCUL(11)
   JCUL(7) = JCUL(12)
   JCUL(8) = JCUL(13)
   JCUL(9) = JCUL(14)
   JCUL(10) = JCUL(15)
   JCUL(11) = JCUL(16)
   SL(1) = (PVS(1) - U.) / (S$BYRN(1) - 0.)
   DO 71 I = 1, 4
171 SL(I+1) = (PVS(I+1) - PVS(I)) / (S$BYRN(I+1) - S$BYRN(I))
   DO 74 I = 1, 5
174 AK(MEQ+1, I) = SL(I)
   DO 77 I = 0, NVAR
177 AK(MEQ+1, I) = 0.
   GO TO 60
62 SL(1) = (PVS(1) - U.) / (S$BYRN(1) - 0.)
   DO 89 I = 1, 4
\[ S_L(I+1) = (P_V S(I+1) - P_V S(I)) / (S_W B_Y R_N(I+1) - S_W B_Y R_N(I)) \]

IF \( S_L(I+1) \cdot L.T. \cdot S_L(I) \) GO TO 1098

GO TO 89

1098 \( S_L(I+1) = S_L(I) + 1 \)

89 CONTINUE

\[ S_L(6) = (P_V B(I) - 0) / (D_W B_Y R_N(I) - 0) \]

GO 34 \( I = 6, 9 \)

\[ S_L(I+1) = (P_V B(I-N+1) - P_V B(I-N)) / (D_W B_Y R_N(I-N+1) - D_W B_Y R_N(I-N)) \]

IF \( S_L(I+1) \cdot L.T. \cdot S_L(I) \) GO TO 665

GO TO 34

665 \( S_L(I+1) = S_L(I) + 1 \)

34 CONTINUE

GO TO 7779

50 \( k(1) = S_Q b_Y R_P(1) \)

\( k(2) = S_Q b_Y R_P(2) - S_Q b_Y R_P(1) \)

\( k(3) = S_Q b_Y R_P(3) - S_Q b_Y R_P(2) \)

\( k(4) = S_Q b_Y R_P(4) - S_Q b_Y R_P(3) \)

\( k(5) = S_Q b_Y R_P(5) - S_Q b_Y R_P(4) \)

IF \( M_E Q.E Q .6 \) GO TO 12

\( k(6) = D_Q b_Y R_P(1) \)

\( k(7) = D_Q b_Y R_P(2) - D_Q b_Y R_P(1) \)

\( k(8) = D_Q b_Y R_P(3) - D_Q b_Y R_P(2) \)

\( k(9) = D_Q b_Y R_P(4) - D_Q b_Y R_P(3) \)

\( k(10) = D_W B_Y R_P(5) - D_W B_Y R_P(4) \)

IF \( J_C N T.E Q .1 \) GO TO 16

IF \( J_C N T.E Q .2 \) GO TO 18

IF \( J_C N T.E Q .3 \) GO TO 119

IF \( J_C N T.E Q .4 \) GO TO 220

IF \( J_C N T.E Q .5 \) GO TO 222

10 \( k(M_E Q) = P_Y R_K_P(1) \)

\( J_C N T = J_C N T + 1 \)

\( L_L = L_L + 1 \)

\( S_S U M(N_C N T_W, L_L) = P_Y R_K_P(1) \)

GO TO 217

18 \( k(M_E Q) = P_Y R_K_P(2) \)

\( J_C N T = J_C N T + 1 \)

\( L_L = L_L + 1 \)

\( S_S U M(N_C N T_W, L_L) = P_Y R_K_P(2) \)

GO TO 217

119 \( k(M_E Q) = P_Y R_K_P(3) \)

\( J_C N T = J_C N T + 1 \)

\( L_L = L_L + 1 \)

\( S_S U M(N_C N T_W, L_L) = P_Y R_K_P(3) \)

GO TO 217

220 \( k(M_E Q) = P_Y R_K_P(4) \)

\( J_C N T = J_C N T + 1 \)

\( L_L = L_L + 1 \)

\( S_S U M(N_C N T_W, L_L) = P_Y R_K_P(4) \)

GO TO 217
220 0K(MEQ)=PYRRP(5)
   JCNT=JCNT+1
   LL=LL+1
   SSUM(MCNTW,LL)=PYRRP(5)
   GO TO 217
12 IRUN(M)=IRUN(N)
217 IF(MEQ.NE.0) GO TO 65
   JCOL(5)=JCOL(11)
   JCOL(7)=JCOL(12)
   JCOL(8)=JCOL(13)
   JCOL(9)=JCOL(14)
   JCOL(10)=JCOL(15)
   JCOL(11)=JCOL(16)
   SL(I)=(PVS(I)-0.)/(SBYRP(I)-0.)
   DO 72 I=1,4
   72 SL(I+1)=(PVS(I+1)-PVS(I))/(SBYRP(I+1)-SBYRP(I))
   DO 75 I=5,9
5
   75 AK(MEQ+1,I)=-SL(I)
   DO 78 I=10,NVAR
   78 AK(MEQ+1,I)=0.
   GO TO 60
60 SL(I)=(PVS(I)-0.)/(SBYRP(I)-0.)
   DO 90 I=1,4
   90 CON TINUE
   90 SL(I)=(PVS(I)-0.)/(SBYRP(I)-0.)
   DO 95 I=5,9
   95 SL(I+1)=(PVS(I+1)-PVS(I))/(SBYRP(I+1)-SBYRP(I))
   IF(SL(I+1).LT.SL(I)) GO TO 4007
   GO TO 90
4007 SL(I+1)=SL(I)+1.
   90 CON TINUE
   SL(I)=(PVB(I)-0.)/(SBYRP(I)-0.)
   DO 100 I=1,5
   100 AK(MEQ+1,1)=-SL(I)
   DO 105 I=6,10
   105 AK(MEQ+1,1)=-SL(I)
   DO 110 I=11,NVAR
   110 AK(MEQ+1,1)=0.
   110 MEQ1=MEQ-1
   DO 120 I=1,MEQ1
   120 AK(I,II)=1.
   AK(I,II+MEQ)=1.
   AK(MEQ,II)=1.
   25 CON TINUE
   AK(MEQ,MEQ)=-1.
   MEQ3=MEQ+1
IF(NSOL.EQ.0) GO TO 400
WRITE(6,28)
28 FORMAT(1,132('*'),/,'//,10X,'INITIAL TABLEAU FROM SUBROUTINE TABLU'
1,//)
   DO 45 I=1,MEW
   DO 45 J=1,NVAR
   WRITE(6,20)I,J,AK(I,J)
   20 FORMAT(55X,'AK(',I2,','I2,') = ',F11.2,/) 
   45 CONTINUE
   WRITE(6,29)
   29 FORMAT(//'\132('*')//)
   400 CALL LPGUGU(MEW,NVAR,AK,BK,JCUL,SL,NM,MPLUS2,LL,ISS,ISBD,ISN
   ,ISPH,PREM,SSUM,OBFUN,NCNTW,A,N,ZIM0X,NPICK,NPICK1,NPI
   CK2,ISLC,I
   3LS,ISLA,JCOLM,NPICK3,NPICK4,NBIG,SPLIT1,SPLIT2
   IF(JCNT.EQ.6) GO TO 21
   GO TO 218
21 RETURN
END

SUBROUTINE TABLU4(CCCCLCCCCCCCCCCCCCCCCC
-CCCLCCCCCCCCCCCCCCCCCCCC
-CCCLCCCCCCCCCCCCC)

SUBROUTINE TABLU4(NPHA,N,DQBYR3,DQBYRb,DQBYRN,DQBYK3,DQBYK,
P=VR,NSOL,ISSS
2,ISBD,ISNI,ISPH,PYRRS,PYRRD,PYRRN,PYRKP,LL,PREM,SSUM,O=
BFUN,A,ZIM0X
3A,NCNTW,NPICK,NPICKK1,NPICK2,ISLC,ISLS,ISLB,NPICK3,NPICK
K2,NBIG,SPLI
3I,SPLIT1,SPLIT2)
DIMENSION BK(16),AK(18,48),DQBYR3(6),DQBYRb(6),DQBYRN(-
6),DQBYK3(6),DQBYK(6)
1,PVR(6),IBASIS(20),SL(15),PYRRS(6),PYRRD(6),PYRRN(6),P=
YRKP(6),PREM
6(6),ZIM0UX(6),OBFUN(20,6),SSUM(20,6),A(5,2),NPICK(6),NP=
ICK1(6),NPICK2(6)
2CK2(6),NPICK3(6),NPICK4(6),SPLIT1(6),SPLIT2(6-
)
REAL*8 IKOW(16)
REAL*8 JCOL(31)
REAL*8 JCOLM(31)
DATA IKOW/'SG1PR3','SG2PR3','SG3PR3','SG4PR3','SG5PR3-'
,'SEPSUM','
1','','','','','','','','','','','','','','','','','','
,','/'
DATA JCOL/'X51','X32','X53','X34','X55','SURPL1','SLK1-
1','SLK2','SL
1K3','SLK4','SLK5','','','','','','','','','','','','
4','','','','','','','','','','','','
JCNT=1
LL=0
117 MEW=(5*NPRA)+1
NVAR=((5*NPRA)*2)+1
NM=NVAR+MEQ
MPLUS2=MEQ+2
DO 95 I=1,MPLUS2
BK(I)=0.
IBASIS(I)=0
DO 95 J=1,NM
95 AK(I,J)=0.
IF(ISSS.EQ.0) GO TO 53
IF(ISBD.EQ.0) GO TO 52
IF(ISNI.EQ.0) GO TO 51
IF(ISPH.EQ.0) GO TO 50
53 BK(1)=DQBYS(R(1)
BK(2)=DQBYS(R(2) - DQBYS(R(1)
BK(3)=DQBYS(R(3) - DQBYS(R(2)
BK(4)=DQBYS(R(4) - DQBYS(R(3)
BK(5)=DQBYS(R(5) - DQBYS(R(4)
IF(JCNT.EQ.1) GO TO 11
IF(JCNT.EQ.2) GO TO 222
IF(JCNT.EQ.3) GO TO 333
IF(JCNT.EQ.4) GO TO 444
IF(JCNT.EQ.5) GO TO 5
11 BK(MEO)=PYRRS(1)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNTW,LL)=PYRRS(1)
GO TO 17
222 BK(MEO)=PYRRS(2)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNTW,LL)=PYRRS(2)
GO TO 17
333 BK(MEO)=PYRRS(3)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNTW,LL)=PYRRS(3)
GO TO 17
444 BK(MEO)=PYRRS(4)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNTW,LL)=PYRRS(4)
GO TO 17
5 BK(MEO)=PYRRS(5)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNTW,LL)=PYRRS(5)
GO TO 17
17 SL(1)=(PVb(1)-V.)/(DQBYS(R(1)-0.))
$u_0$ 9981  I=1,4
$SL(I+1) = (PV*B(I+1) - PV*B(I)) / (D*BY*RS(I+1) - D*BY*RS(I))$

IF($SL(I+1) \cdot LT \cdot SL(I)$) GO TO 1098

GO TO 9981

1098 $SL(I+1) = SL(I) + 1$

9981 CONTINUE

$u_0$ 3 I=1,5

3 AK(MEQ+1, I) = -SL(I)

$u_0$ 35 I=0,NVAR

33 AK(MEQ+1, I) = 0.

GO TO 60

52 aK(1) = D*BY*RB(1)

$u_0$ aK(2) = D*BY*RB(2) - D*BY*RB(1)

$u_0$ aK(3) = D*BY*RB(3) - D*BY*RB(2)

$u_0$ aK(4) = D*BY*RB(4) - D*BY*RB(3)

$u_0$ aK(5) = D*BY*RB(5) - D*BY*RB(4)

IF(JCNT.EQ.1) GO TO 6

IF(JCNT.EQ.2) GO TO 77

IF(JCNT.EQ.3) GO TO 8

IF(JCNT.EQ.4) GO TO 9

IF(JCNT.EQ.5) GO TO 10

6 aK(MEQ) = PY*RB(1)

$u_0$ JCNT = JCNT+1

$u_0$ LL = LL + 1

SSUM(NCNTW, LL) = PY*RB(1)

GO TO 114

77 aK(MEQ) = PY*RB(2)

$u_0$ JCNT = JCNT+1

$u_0$ LL = LL + 1

SSUM(NCNTW, LL) = PY*RB(2)

GO TO 114

8 aK(MEQ) = PY*RB(3)

$u_0$ JCNT = JCNT+1

$u_0$ LL = LL + 1

SSUM(NCNTW, LL) = PY*RB(3)

GO TO 114

9 aK(MEQ) = PY*RB(4)

$u_0$ JCNT = JCNT+1

$u_0$ LL = LL + 1

SSUM(NCNTW, LL) = PY*RB(4)

GO TO 114

10 aK(MEQ) = PY*RB(5)

$u_0$ JCNT = JCNT+1

$u_0$ LL = LL + 1

SSUM(NCNTW, LL) = PY*RB(5)

GO TO 114

114 $SL(I) = (PV*B(I) - PV*B(I)) / (D*BY*RS(I) - D*BY*RS(I))$

$u_0$ 9999  I=1,4

$SL(I+1) = (PV*B(I+1) - PV*B(I)) / (D*BY*RB(I+1) - D*BY*RB(I))$

IF($SL(I+1) \cdot LT \cdot SL(I)$) GO TO 987

GO TO 9999
CONTINUE

10 4 I=1,5
4  AK(ME0+1,T) = SL(1)
10 32 I=0,NVAR
32  AK(ME0+1,T) = 0.
GO TO 60
51  BK(1) = UQBYRN(1)
   BK(2) = UQBYRN(2) - UQBYRN(1)
   BK(3) = UQBYRN(3) - UQBYRN(2)
   BK(4) = UQBYRN(4) - UQBYRN(3)
   BK(5) = UQBYRN(5) - UQBYRN(4)
   IF(JCNT.EQ.1) GO TO 111
   IF(JCNT.EQ.2) GO TO 12
   IF(JCNT.EQ.3) GO TO 13
   IF(JCNT.EQ.4) GO TO 14
   IF(JCNT.EQ.5) GO TO 15
111  BK(ME0) = PYRN(1)
   JCNT = JCNT + 1
   LL = LL + 1
   SSUM(NCNTW, LL) = PYRN(1)
   GO TO 115
12  BK(ME0) = PYRN(2)
   JCNT = JCNT + 1
   LL = LL + 1
   SSUM(NCNTK, LL) = PYRN(2)
   GO TO 115
13  BK(ME0) = PYRN(3)
   JCNT = JCNT + 1
   LL = LL + 1
   SSUM(NCNTW, LL) = PYRN(3)
   GO TO 115
14  BK(ME0) = PYRN(4)
   JCNT = JCNT + 1
   LL = LL + 1
   SSUM(NCNTW, LL) = PYRN(4)
   GO TO 115
15  BK(ME0) = PYRN(5)
   JCNT = JCNT + 1
   LL = LL + 1
   SSUM(NCNTW, LL) = PYRN(5)
   GO TO 115
115  SL(1) = (PV(Q(1) - U) / (QBYRN(1) - 0.))
   DO 1 I=1,4
1   SL(I+1) = (PV(R(I+1) - PV(1)) / (QBYRN(I+1) - QBYRN(1))
   IF(SL(I+1).LT.SL(I)) GO TO 27
   GO TO 1
27  SL(I+1) = SL(I) + 1.
1  CONTINUE
   DO 7 I=1,5
7 AK(MEQ+1,I) = -SL(I)
   00 34 I=0,NVAR
34 AK(MEQ+1,I) = 0.
   60 TO 60
50 BK(1) = UQBYRP(1)
   BK(2) = UQBYRP(2) - UQBYRP(1)
   BK(3) = UQBYRP(3) - UQBYRP(2)
   BK(4) = UQBYRP(4) - UQBYRP(3)
   BK(5) = UQBYRP(5) - UQBYRP(4)
   IF(JCNT.EQ.1) GO TO 16
   IF(JCNT.EQ.2) GO TO 18
   IF(JCNT.EQ.3) GO TO 19
   IF(JCNT.EQ.4) GO TO 20
   IF(JCNT.EQ.5) GO TO 22
16 BK(MEQ) = PYKRK(1)
   JCNT = JCNT + 1
   LL = LL + 1
   SSUM(NCNTW,LL) = PYRK(1)
   GO TO 116
18 BK(MEQ) = PYRK(2)
   JCNT = JCNT + 1
   LL = LL + 1
   SSUM(NCNTW,LL) = PYRK(2)
   GO TO 116
19 BK(MEQ) = PYRK(3)
   JCNT = JCNT + 1
   LL = LL + 1
   SSUM(NCNTW,LL) = PYRK(3)
   GO TO 116
20 BK(MEQ) = PYRK(4)
   JCNT = JCNT + 1
   LL = LL + 1
   SSUM(NCNTW,LL) = PYRK(4)
   GO TO 116
22 BK(MEQ) = PYRK(5)
   JCNT = JCNT + 1
   LL = LL + 1
   SSUM(NCNTW,LL) = PYRK(5)
   GO TO 116
116 SL(I) = (PVB(I) - V) / (DQBYRP(I) - 0)
   u0 2 I=1,4
27 SL(I+1) = (PVB(I+1) - PVB(I)) / (DQBYRP(I+1) - DQBYRP(I))
   IF(SL(I+1) .LT. SL(I)) GO TO 277
   GO TO 2
277 SL(I+1) = SL(I) + 1.
   2 CONTINUE
   u0 7777 I = 1,5
7777 BK(MEQ+1,I) = -SL(I)
   u0 6666 I = 0, NVAR
6060 BK(MEQ+1,I) = 0.
60 MEQ1=MEQ-1
25 I=1, MEQ1
AK(I,I)=1.
AK(I, I+MEQ)=1.
AK(MEQ, I)=1.
CONTINUE
AK(MEQ, MEQ)=-1.
MEQ3=MEQ+1
IF(NSOL.EQ.0) GO TO 400
WRITE(6,28)
28 FORMAT(1',132(''),'/IUX,'INITIAL TABLEAU FROM SUBRO-
UTINE TABLU'
2,,//)
45 I=1, MEQ3
DO 45 J=1, NVAR
WRITE(6,20) I, J, AK(I, J)
20 FORMAT(55X,'AK('','I2','','I2') = ',F11.2,//)
CONTINUE
WRITE(6,29)
29 FORMAT(,,132(''),//)
400 CALL LPGUGU(MEQ, NVAR, INOW, AK, BK, JCOL, SL, NM, MPLUS2, LL, I-
S5, I850, ISN
41, ISPH, PREM, SSUM, OBUN, NCINT, AN, ZIMBOX, NPICK, NPICK1, NP-
ICK2, ISLC, 0
3SLS, ISLB, JCOLM, NPICK3, NPICK4, NDIG, SPLIT, SPLIT1, SPLIT2)
IF(JCNT.EQ.6) GO TO 21
GO TO 117
21 RETURN
END
SUBROUTINE TABLO CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC-
02594
SUBROUTINE TABLO(NPRA, N, CQBYK, SUBYR, UQBYK, PV, PVS, PV-
0, CUBYR, CUB)
1YN, CQBYRK, SUBYRB, SQBYRN, SUBYRP, UQBYKB, DUBYRN, UQBYRP, I-
S5, ISB0, ISN
11, ISPH, NSOL, PYRRS, PYRRP, PPRRN, PYRRRP, LL, PREM, SSUM, OBUN-
R, A, ZIMBOX, NC
2NTW, NPICK, NPICK1, NPICK2, ISLC, ISLS, ISLB, NPICK3, NPICK4, ND-
IG, SPLIT, SP
3LI11, SPLIT2)
DIMENSION UK(18), AK(18, 46), CUBYRP(6), SUBYRP(6), DUBYRP(-
0), PV5(6), PV
5(6), PVb(6), IBASIS(20), SL(15), CUBYRN(6), CUBYRB(6), CUBYR-
S(6), SUBYRS(6), SUBYRN(6), DUBYRS(6), DUBYRB(6), DUBYRN(6), P-
YRR(6), PYRR
60(6), PYRRN(6), PYRRP(6), PREM(6), ZIMBOX(6), SSUM(20, 6), UB-
FUN(20, 6), A(
63, 2), NPICK(6), NPICK1(6), NPICK2(6), NPICK3(6), NPICK4(6), S-
PLIT(6), SPL
NOTE: ONLY 5 GRAPH POINTS PER PRACTICE.
IF(MEQ,EQ,11) GO TO 408
B(11)=D(2)-D(1)
B(12)=D(3)-D(2)
B(13)=D(4)-D(3)
B(14)=D(5)-D(4)
GO TO 22
408 IROW(6)=IROW(16)
GO TO 22
409 IROW(11)=IROW(16)
22 IF(JCNT.EQ.1) GO TO 540
IF(JCNT.EQ.2) GO TO 533
IF(JCNT.EQ.3) GO TO 534
IF(JCNT.EQ.4) GO TO 533
IF(JCNT.EQ.5) GO TO 536
540 B(MEQ)=PYRS(1)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNT,LL)=PYRS(1)
GO TO 113
533 B(MEQ)=PYRS(2)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNT,LL)=PYRS(2)
GO TO 113
534 B(MEQ)=PYRS(3)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNT,LL)=PYRS(3)
GO TO 113
535 B(MEQ)=PYRS(4)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNT,LL)=PYRS(4)
GO TO 113
536 B(MEQ)=PYRS(5)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNT,LL)=PYRS(5)
GO TO 113
113 IF(MEQ.NE.6) GO TO 75
JCOL(6)=JCOL(6)
JCOL(7)=JCOL(17)
JCOL(8)=JCOL(16)
JCOL(9)=JCOL(19)
JCOL(10)=JCOL(20)
JCOL(11)=JCOL(21)
JCOL(12)=JCOL(22)
JCOL(13)=JCOL(23)
JCOL(14)=JCOL(24)
JCOL(15) = COL(25)
JCOL(16) = COL(26)
SL(1) = (PV(I+1) - PV(I)) / (CwByRS(I+1) - CwByRS(I))

DO 992 I = 1, 4
SL(I+1) = (PV(I+1) - PV(I)) / (CwByRS(I+1) - CwByRS(I))
IF(SL(I+1) .LT. SL(I)) Go to 27
GO TO 992

27 SL(I+1) = SL(I) + 1.
992 CONTINUE

DO 417 I = 1, 5
417 AK(MEQ+1, I) = SL(I)
DO 420 I = 6, NVAR
420 AK(MEQ+1, I) = 0.
GO TO 60

75 IF(MEQ .NE. 1) Go to 414
JCOL(11) = COL(16)
JCOL(12) = COL(17)
JCOL(13) = COL(18)
JCOL(14) = COL(19)
JCOL(15) = COL(20)
JCOL(16) = COL(21)
JCOL(17) = COL(22)
JCOL(18) = COL(23)
JCOL(19) = COL(24)
JCOL(20) = COL(25)
JCOL(21) = COL(26)
PRINT, JCOL

SL(I) = (PV(I) - 0.) / (CwByRS(I) - 0.)

DO 410 I = 1, 4
SL(I+1) = (PV(I+1) - PV(I)) / (CwByRS(I+1) - CwByRS(I))
IF(SL(I+1) .LT. SL(I)) Go to 27
GO TO 410

277 SL(I+1) = SL(I) + 1.
410 CONTINUE

SL(6) = (PV5(1) - 0.) / (SubYRS(6) - 0.)

DO 403 I = 6, 9
SL(I+1) = (PV5(I-N+1) - PV5(I-N)) / (SubYRS(I-N+1) - SubYRS(I-N))
IF(SL(I+1) .LT. SL(I)) Go to 277
GO TO 403

2777 SL(I+1) = SL(I) + 1.
403 CONTINUE

DO 77 I = 1, 5
77 AK(MEQ+1, I) = -SL(I)
DO 4000 I = 6, 10
4000 AK(MEQ+1, I) = -SL(I)
DO 7000 I = 11, NVAR
7000 AK(MEQ+1, I) = 0.
GO TO 60

414 SL(1) = (PV(I) - 0.) / (CwByRS(I) - 0.)
UN 5002 1=1,4
SL(I+1)=(PV(I+1)-PV(I))/(CVBYRS(I+1)-CVBYRS(I))
IF(SL(I+1).LT.SL(I)) GO TO 278
GO TO 5002
278 SL(I+1)=SL(I)+1.
5002 CONTINUE
SL(6)=(PVS(1)-V.)/(SUBYRS(1)-0.)
DD 8888 I=5,9
SL(I+1)=(PVS(I-N+1)-PVS(I-N))/(SUBYRS(I-N+1)-SUBYRS(I-N))
IF(SL(I+1).LT.SL(I)) GO TO 278
GO TO 8888
278 SL(I+1)=SL(I)+1.
8888 CONTINUE
SL(11)=(PVB(1)-O.)/(UQBYRS(1)-0.)
MEQ5=MEQ=2
DD 3333 I=11,MEQ5
SL(I+1)=(PVB(I-9)-PVB(I-10))/(UQBYRS(I-9)-UQBYRS(I-10))
IF(SL(I+1).LT.SL(I)) GO TO 1444
GO TO 3333
1444 SL(I+1)=SL(I)+1.
3333 CONTINUE
GO TO 7779
52 OK(1)=CVBYRS(1)
OK(2)=CVBYRS(2)-CVBYRS(1)
OK(3)=CVBYRS(3)-CVBYRS(2)
OK(4)=CVBYRS(4)-CVBYRS(3)
OK(5)=CVBYRS(5)-CVBYRS(4)
IF(MEQ.EQ.0) GO TO 1
OK(6)=QBYRK(1)
OK(7)=QBYRK(2)-QBYRK(1)
OK(8)=QBYRK(3)-QBYRK(2)
OK(9)=QBYRK(4)-QBYRK(3)
OK(10)=QBYRK(5)-QBYRK(4)
IF(MEQ.EQ.1) GO TO 9
OK(11)=QBYRb(1)
OK(12)=QBYRb(2)-QBYRb(1)
OK(13)=QBYRb(3)-QBYRb(2)
OK(14)=QBYRb(4)-QBYRb(3)
OK(15)=QBYRb(5)-QBYRb(4)
GO TO 5550
1 IRUW(6)=IRUW(16)
GO TO 5550
69 IRUW(11)=IRUW(16)
5350 IF(JCNT.EQ.1) GO TO 539
IF(JCNT.EQ.2) GO TO 211
IF(JCNT.EQ.3) GO TO 212
IF(JCNT.EQ.4) GO TO 213
IF(JCNT.EQ.5) GO TO 214
539  ak(keq) = pyrrb(1)
       jcnt = jcnt + 1
       ll = ll + 1
       ssu(m(ncniw, ll)) = pyrrb(1)
       go tu 989
211  ak(keq) = pyrrb(2)
       jcnt = jcnt + 1
       ll = ll + 1
       ssu(m(ncniw, ll)) = pyrrb(2)
       go tu 989
212  ak(keq) = pyrrb(3)
       jcnt = jcnt + 1
       ll = ll + 1
       ssu(m(ncniw, ll)) = pyrrb(3)
       go tu 989
213  ak(keq) = pyrrb(4)
       jcnt = jcnt + 1
       ll = ll + 1
       ssu(m(ncniw, ll)) = pyrrb(4)
       go tu 989
214  ak(keq) = pyrrb(5)
       jcnt = jcnt + 1
       ll = ll + 1
       ssu(m(ncniw, ll)) = pyrrb(5)
       go tu 989
989  if(keq .ne. 6) go 10 70
       jcul(6) = jcul(10)
       jcul(7) = jcul(17)
       jcul(8) = jcul(18)
       jcul(9) = jcul(19)
       jcul(10) = jcol(20)
       jcul(11) = jcol(21)
       jcul(12) = jcol(22)
       jcul(13) = jcol(23)
       jcul(14) = jcol(24)
       jcul(15) = jcol(25)
       jcul(16) = jcol(26)
       sl(i) = (pv(i+1) - pv(i)) / (cubyrb(i+1) - cubyrb(i))
       go 993 i=1,4
       sl(i+1) = (pv(i+1) - pv(i)) / (cubyrb(i+1) - cubyrb(i))
       if(sl(i+1) .lt. sl(i)) go 10 101
       go tu 993
101  sl(i+1) = sl(i) + 1.
993  continue
       go 418 i=1,5
410  ak(keq+1, i) = sl(i)
       go 416 i=6, nvar
416  ak(keq+1, i) = 0.
       go tu 00
70  if(keq .ne. 11) go tu 413
JCUL(11)=JCUL(16)
JCUL(12)=JCUL(17)
JCUL(13)=JCUL(18)
JCUL(14)=JCUL(19)
JCUL(15)=JCUL(20)
JCUL(16)=JCUL(21)
JCUL(17)=JCUL(22)
JCUL(18)=JCUL(23)
JCUL(19)=JCUL(24)
JCUL(20)=JCUL(25)
JCUL(21)=JCUL(26)
SL(1)=(PV(1)-0.)/(CQBYRB(1)-v.)
UD 401 I=1,4
SL(I+1)=(PV(I+1)-PV(1))/(CQBYRB(I+1)-CQBYRB(I))
IF(SL(I+1),LT,SL(I)) GO TO 402
GO TO 401
102 SL(I+1)=SL(I)+1.
401 CONTINUE
SL(6)=(PV(6)-0.)/(CQBYRB(6)-0.)
UD 404 I=6,9
SL(I+1)=(PV(I+1)-PV(I))/(CQBYRB(I+1)-CQBYRB(I))
IF(SL(I+1),LT,SL(I)) GO TO 403
GO TO 404
103 SL(I+1)=SL(I)+1.
404 CONTINUE
UD 76 I=1,5
76 AK(MEQ+1,I)=SL(I)
UD 5000 I=6,10
5000 AK(MEQ+1,I)=SL(I)
UD 8000 I=11,NVAK
8000 AK(MEQ+1,I)=0.
GO TO 60
413 SL(1)=(PV(1)-0.)/(CQBYRB(1)-v.)
UD 6002 I=1,4
SL(I+1)=(PV(I+1)-PV(I))/(CQBYRB(I+1)-CQBYRB(I))
IF(SL(I+1),LT,SL(I)) GO TO 419
GO TO 6002
419 SL(I+1)=SL(I)+1.
6002 CONTINUE
SL(6)=(PV(6)-0.)/(CQBYRB(6)-0.)
UD 7777 I=6,9
SL(I+1)=(PV(I+1)-PV(I))/(CQBYRB(I+1)-CQBYRB(I))
IF(SL(I+1),LT,SL(I)) GO TO 4040
GO TO 7777
4040 SL(I+1)=SL(I)+1.
7777 CONTINUE
SL(11)=(PV(1)-0.)/(CQBYRB(1)-v.)
MEWS=MEQ-2
DO 2222 I=11,MEO5
SL(I+1)=(PVH(I-9)-PVH(I-10))/(UOPVR(I-9)-UOPVR(I-10))
IF(SL(I+1),LT.,SL(I)) GO TO 5U5U
GO TO 2222
5U5U SL(I+1)=SL(I)+1.
2222 CONTINUE
GO TO 7779
51 dK(1)=CQBYKN(1)
dK(2)=CQBYKN(2)-CQBYKN(1)
dK(3)=CQBYKN(3)-CQBYKN(2)
dK(4)=CQBYKN(4)-CQBYKN(3)
dK(5)=CQBYKN(5)-CQBYKN(4)
IF(MEO.EQ.0) GO TO 2
dK(6)=CQBYKN(1)
dK(7)=CQBYKN(2)-CQBYKN(1)
dK(8)=CQBYKN(3)-CQBYKN(2)
dK(9)=CQBYKN(4)-CQBYKN(3)
dK(10)=CQBYKN(5)-CQBYKN(4)
IF(MEO.EQ.1) GO TO 70
dK(11)=CQBYKN(1)
dK(12)=CQBYKN(2)-CQBYKN(1)
dK(13)=CQBYKN(3)-CQBYKN(2)
dK(14)=CQBYKN(4)-CQBYKN(3)
dK(15)=CQBYKN(5)-CQBYKN(4)
GO TO 1234
21 RUW(6)=IRUN(16)
GO TO 1234
70 RUW(11)=IRUN(16)
1234 IF(JCNT.EQ.1) GO TO 530
IF(JCNT.EQ.2) GO TO 215
IF(JCNT.EQ.3) GO TO 210
IF(JCNT.EQ.4) GO TO 217
IF(JCNT.EQ.5) GO TO 210
530 dK(MEO)=PYRKN(1)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNIW,LL)=PYRKN(1)
GO TO 789
215 dK(MEO)=PYRKN(2)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNIW,LL)=PYRKN(2)
GO TO 789
210 dK(MEO)=PYRKN(3)
JCNT=JCNT+1
LL=LL+1
SSUM(NCNIW,LL)=PYRKN(3)
GO TO 789
217 dK(MEO)=PYRKN(4)
JCNT = JCNT + 1
LL = LL + 1
SSUM(NCNT4, LL) = PYRN(4)
GO TO 789
218 AK(MEQ) = PYRN(5)
JCNT = JCNT + 1
LL = LL + 1
SSUM(NCNT4, LL) = PYRN(5)
GO TO 789
789 IF(MEQ, NE, 6) GO TO 412
JCUL(6) = JCOL(16)
JCUL(7) = JCOL(17)
JCUL(8) = JCOL(18)
JCUL(9) = JCOL(19)
JCUL(10) = JCOL(20)
JCUL(11) = JCOL(21)
JCUL(12) = JCOL(22)
JCUL(13) = JCOL(23)
JCUL(14) = JCOL(24)
JCUL(15) = JCOL(25)
JCUL(16) = JCOL(26)
SL(I) = (PV(I+1) - PV(I)) / (CUBYRN(I+1) - CUBYRN(I))
GO 994 I = 1, 4
SL(I+1) = (PV(I+1) - PV(I)) / (CUBYRN(I+1) - CUBYRN(I))
IF(SL(I+1).LT., SL(I)) GO TO 6060
GO TO 994
6060 SL(I+1) = SL(I) + 1.
994 CONTINUE
GO 419 I = 1, 5
419 AK(MEQ+1, I) = -SL(I)
GO 415 I = 6, NVAK
415 AK(MEQ+1, I) = 0.
GO TO 60
412 IF(MEQ, NE, 11) GO TO 411
JCUL(11) = JCOL(16)
JCUL(12) = JCOL(17)
JCUL(13) = JCOL(18)
JCUL(14) = JCOL(19)
JCUL(15) = JCOL(20)
JCUL(16) = JCOL(21)
JCUL(17) = JCOL(22)
JCUL(18) = JCOL(23)
JCUL(19) = JCOL(24)
JCUL(20) = JCOL(25)
JCUL(21) = JCOL(26)
SL(I) = (PV(1) - 0.) / (CUBYRN(1) - 0.)
GO 402 I = 1, 4
SL(I+1) = (PV(I+1) - PV(I)) / (CUBYRN(I+1) - CUBYRN(I))
IF(SL(I+1).LT., SL(I)) GO TO 7070
GO TO 402
7070 SL(I+1)=SL(I)+1.
402 CONTINUE
   SL(5)=(PVS(I)-0.)/(SUBYRN(I)-0.)
  00 67 I=6,9
   SL(I+1)=(PVS(I-N+1)-PVS(I-N))/(SUBYRN(I-N+1)-SUBYRN(I-N))
   IF(SL(I+1).LT.SL(I)) GO TO 8080
  00 TU 67
8080 SL(I+1)=SL(I)+1.
  67 CONTINUE
  00 21 I=1,5
   AK(MEQ+1,I)=-SL(I)
  00 6000 I=0,10
  6000 AK(MEQ+1,I)=-SL(I)
  00 9000 I=11, IMF
  9000 AK(MEQ+1,I)=0.
  00 TU 60
411 SL(I)=(PV(1)-0.)/(CUoYRN(1)-0.)
  00 7002 I=1,4
   SL(I+1)=(PV(I+1)-PV(I))/(CUBYRN(I+1)-CUBYRN(I))
   IF(SL(I+1).LT.SL(I)) GO TO 9090
  00 TU 7002
9090 SL(I+1)=SL(I)+1.
  7002 CONTINUE
   SL(5)=(PVS(1)-0.)/(SUBYRN(1)-0.)
  00 5555 I=0,9
   SL(I+1)=(PVS(I-N+1)-PVS(I-N))/(SUBYRN(I-N+1)-SUBYRN(I-N))
   IF(SL(I+1).LT.SL(I)) GO TO 1011
  00 TU 5555
1011 SL(I+1)=SL(I)+1.
  5555 CONTINUE
   SL(11)=(PV(1)-0.)/(UQBYRN(1)-0.)
  00 1111 I=11, MEQ5
   MEWS=MEQ-2
  00 1111 I=11, MEQ5
   SL(I+1)=(PV(1)-PV(1-10))/(UQBYRN(I-9)-UQBYRN(I-10))
   IF(SL(I+1).LT.SL(I)) GO TO 6789
  00 TU 1111
6789 SL(I+1)=SL(I)+1.
  1111 CONTINUE
  00 TU 7779
50 B(1)=CUoYRP(1)
  00 B(2)=CUoYRP(2)-CUoYRP(1)
  00 B(3)=CUoYRP(3)-CUoYRP(2)
  00 B(4)=CUoYRP(4)-CUoYRP(3)
  00 B(5)=CUoYRP(5)-CUoYRP(4)
  00 B(MEQ,EU,0) GO TO 406
  00 B(6)=SQBYRK(1)
  00 B(7)=SQBYRK(2)-SQBYRK(1)
OK (8) = SQBYRP (3) - SQBYRP (2)
OK (9) = SQBYRP (4) - SQBYRP (3)
OK (10) = SUBTRP (5) - SQBYRP (4)
IF MEQ, EQ, 11) G0 TO 407
OK (11) = DUBYRP (1)
OK (12) = DUBYRP (2) - DUBYRP (1)
OK (13) = DUBYRP (3) - DUBYRP (2)
OK (14) = DUBYRP (4) - DUBYRP (3)
OK (15) = DUBYRP (5) - DUBYRP (4)
G0 T0 2345
406 I10W (6) = I10W (16)
G0 T0 2345
407 I10W (11) = I10W (16)
2345 IF (JCNT, EQ, 1) G0 TO 537
IF (JCNT, EQ, 2) G0 TO 219
IF (JCNT, EQ, 3) G0 TO 220
IF (JCNT, EQ, 4) G0 TO 341
IF (JCNT, EQ, 5) G0 TO 622
537 OK (MEQ) = PYRKP (1)
JCNT = JCNT + 1
LL = LL + 1
SSUM (NCNTW, LL) = PYRKP (1)
G0 T0 9987
219 OK (MEQ) = PYRKP (2)
JCNT = JCNT + 1
LL = LL + 1
SSUM (NCNTW, LL) = PYRKP (2)
G0 T0 9987
220 OK (MEQ) = PYRKP (3)
JCNT = JCNT + 1
LL = LL + 1
SSUM (NCNTW, LL) = PYRKP (3)
G0 T0 9987
341 OK (MEQ) = PYRKP (4)
JCNT = JCNT + 1
LL = LL + 1
SSUM (NCNTW, LL) = PYRKP (4)
G0 T0 9987
622 OK (MEQ) = PYRKP (5)
JCNT = JCNT + 1
LL = LL + 1
SSUM (NCNTW, LL) = PYRKP (5)
G0 T0 9987
9987 IF (MEQ, NE, 0) G0 TO 55
JCUL (6) = JCUL (16)
JCUL (7) = JCUL (17)
JCUL (8) = JCUL (16)
JCUL (9) = JCUL (17)
JCUL (10) = JCUL (20)
JCUL (11) = JCUL (21)
JCUL(12)=JCUL(22)
JCUL(13)=JCUL(23)
JCUL(14)=JCUL(24)
JCUL(15)=JCUL(25)
JCUL(16)=JCUL(26)
SL(1)=(PV(1)-0.)/(CuBYRP(1)-0.)
GO 9981 I=1,4
SL(I+1)=(PV(I+1)-PV(I))/(CuBYRP(I+1)-CUBYRP(I))
1F(SL(I+1),LI,SL(I)) GU TO 4001
GO TO 9981
4001 SL(I+1)=SL(I)+1.
9981 CONTINUE
GO 5 I=1,5
5 AK(MEQ+1,I)=-SL(1)
GO 6 I=6,NVAR
6 AK(MEQ+1,I)=0.
GO TO 00
55 IF(MEQ,NE,11) GO TO 56
JCUL(11)=JCUL(16)
JCUL(12)=JCUL(17)
JCUL(13)=JCUL(18)
JCUL(14)=JCUL(19)
JCUL(15)=JCUL(20)
JCUL(16)=JCUL(21)
JCUL(17)=JCUL(22)
JCUL(18)=JCUL(23)
JCUL(19)=JCUL(24)
JCUL(20)=JCUL(25)
JCUL(21)=JCUL(26)
SL(1)=(PV(1)-0.)/(CuBYRP(1)-0.)
GO 4242 I=1,4
SL(I+1)=(PV(I+1)-PV(I))/(CuBYRP(I+1)-CUBYRP(I))
1F(SL(I+1),LI,SL(I)) GU TO 1443
GO TO 4242
1443 SL(I+1)=SL(I)+1.
4242 CONTINUE
SL(I)=(PV(I)-0.)/(CuBYRP(I)-0.)
GO 4243 I=6,9
SL(I+1)=(PV(I-N+1)-PV(I-N))/(CuBYRP(I-N+1)-CUBYRP(I-N))
1F(SL(I+1),LI,SL(I)) GU TO 1442
GO TO 4243
1442 SL(I+1)=SL(I)+1.
4243 CONTINUE
GO 7 I=1,5
7 AK(MEQ+1,I)=-SL(1)
GO 8 I=6,10
8 AK(MEQ+1,I)=-SL(1)
GO 9 I=11,NVAR
9 AK(MEQ+1,I)=0.
GO TO 60

56  SL(1) = (PV(1) - 0) / (CSBYRP(1) - V.)
DO 4245 I = 1, 4
SL(I + 1) = (PV(I + 1) - PV(I)) / (CSBYRP(I + 1) - CSBYRP(I))
IF (SL(I + 1) .LT. SL(I)) GO TO 1441
GO TO 4245

1441 SL(I + 1) = SL(I) + 1.
4245 CONTINUE
SL(I) = (PV(I) - 0) / (CSBYRP(I) - 0.)
DO 4247 I = 6, 9
SL(I + 1) = (PV(I - N + 1) - PV(I - N)) / (CSBYRP(I - N + 1) - CSBYRP(I - N))
IF (SL(I + 1) .LT. SL(I)) GO TO 8410
GO TO 4247

8410 SL(I + 1) = SL(I) + 1.
4247 CONTINUE
SL(I) = (PV(I) - 0) / (CSBYRP(I) - 0.)
MEW5 = MEQ - 2
DO 4251 I = 11, MEQ5
SL(I + 1) = (PV(I - 9) - PV(I - 10)) / (CSBYRP(I - 9) - CSBYRP(I - 10))
IF (SL(I + 1) .LT. SL(I)) GO TO 8911
GO TO 4251

8911 SL(I + 1) = SL(I) + 1.
4251 CONTINUE
7779 DO 10 I = 1, 5
10 AK(MEQ + 1, I) = -SL(I)
DO 11 I = 6, 10
11 AK(MEQ + 1, I) = -SL(I)
DO 12 I = 11, 15
12 AK(MEQ + 1, I) = -SL(I)
DO 13 I = 16, NVAR
13 AK(MEQ + 1, I) = 0.
60 MEW1 = MEQ - 1
DO 25 II = 1, MEW1
AK(II, II) = 1.
AK(II, II + MEQ) = 1.
AK(MEQ, II) = 1.
25 CONTINUE
AK(MEQ, MEQ) = -1.
MEW3 = MEQ + 1
IF (NSOL .EQ. 0) GO TO 400
WRITE(*, 28)
28 FORMAT(1', 'IS2('X'),//, '10X,'INITIAL TABLEAU FROM SUBRO-
UTINE TABLU'
1,//)
DO 45 I = 1, MEW3
DO 45 J = 1, NVAR
WRITE(*, 20) I, J, AK(I, J)
20 FORMAT(52X, 'AK(', 'I2', ',', 'I2', ') = ', 'F11.2,/)
45 CONTINUE
   WRITE(6,29)
29 FORMAT(/,13x('*'),/)
400 CALL LPGUGU(MEQ,NVAR,INOW,AK,RK,JCUL,SL,NM,MPLUS2,LL,I-
   SSS,ISSD,ISN
   41,ISP,WK,SPR,SSUM,OBFUN,NCITW,A,N,ZINOK,NNPK,NNPK1,NN-
   PK2,ISLC,ISL,
3LS,ISLB,JCOLM,NNPK3,NNPK4,NDIG,SPLIT,SPLIT1,SPLIT2)
   DO 131 I=1,31
   JCUL(I)=JCULM(I)
131 CONTINUE
   IF(JCNT.EQ.6) GO TO 221
   GO TO 112
221 RETURN
END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE LPGUGU M=1,31
SUBROUTINE LPGUGU(MEQ,NVAR,INOW,AK,RK,JCUL,SL,NM,MPLUS-
2,LL,I-SS,ISSD,ISN
20D,ISN1,ISP,SPR,SSUM,OBFUN,NCITW,A,N,ZINOK,NNPK,NNPK1,NN-
PK2,ISLC,ISL,
2LS,ISLB,JCOLM,NNPK3,NNPK4,NDIG,SPLIT,SPLIT1,S-
PLIT2)
C THIS ROUTINE IS TAKEN FROM PAGES 109-113 OF DAELLENBAC-
H AND BELL'S U3202
C TEXT ON LINEAR PROGRAMMING AND HAS BEEN MODIFIED TO RU-
N IN IBM 370 U3203
C FORTRAN IV. THE SIGNIFICANT CODE CHANGES WERE TO PROVID-
E DOUBLE U3204
C PRECISION STORAGE FOR THE 8 CHARACTER VARIABLES AND
C CONSTRAINT NAMES.
C LPGUGO IS A MAXIMIZING LINEAR PROGRAMMING CODE. IT US-
ES THE TWO U3206
C PHASE, FULL TABLEAU FORM OF THE SIMPLEX METHOD, REQUI-
RES ALL RNS U3209
C PARAMETERS TO BE NONNEGATIVE, AND STARTS FROM A FULLY-
C ARTIFICIAL U3210
C BASIS. IT ASSUMES THAT ALL CONSTRAINTS HAVE BEEN CONV-
ERTED TO U3211
C EQUATIONS.
C THE OBJECTIVE FUNCTION AND PHASE ONE COEFFICIENTS ARE
C STORED AS THE (M+1)ST AND (M+2)ND ROWS OF THE ARRAY WH-
ICH
C ALSO STORES THE INVERSE OF THE BASIS IN ITS LAST M COL-
UMNS. U3216
C PROGRAM OUTPUT INTERPRETATION. REFERENCE PAGE 102
C CONSIDER THE NUMBERS LISTED UNDER THE HEADING DELTA, T-
HESE U3219
Numbers are called the reduced objective function coefficients. 0320
For nonbasic variables they are nonpositive, with most of them 0321
usually negative. If the Delta_j is equal to zero for a - 0322
nonbasic variable, then this variable could be assigned an arbitrary positive 0323
value within a certain range with corresponding changes - 0324
present basic variables without affecting the value of the objective function, since the latter is at its optimum, this means letting 0325
the variable in question assume positive values we obtain in alternate 0326
optimal solutions.
Consider now a non-basic variable with a negative Delta_j. 0327
If the original objective function coefficient of the is nonbasic 0328
candidate would be larger by an amount equal to the negative of the 0329
corresponding Delta_j, then this variable would become a - 0330
candidate to assume positive values in alternate optimal solutions. 0331
Consider the part of the computer solution that deals with the 0332
constraints. The numbers listed under the heading value 0333
give the change in the optimal value of the objective function 0334
for a unit increase in the value of the RHS parameter of the 0335
corresponding constraint, all other inputs remaining unchanged. 0336
For a constraint that has slack in the optimal solution, this number is zero. These numbers are called imputed values. 0337
The original RHS parameter for the constraint in question may be 0338
decreased or increased, respectively, without affecting the imputed 0339
value for that constraint, all other inputs remaining unchanged. 0340
THE PRESENT DIMENSIONS ACCOMMODATE MAX CONSTRAINS AND N-

VARIABLES (DECISION PLUS SLACK). THESE DIMENSIONS CAN BE-

CHANGED UNDER

\[ \text{DIMENSION } a(m+2, m+n), m(n+2), jcol(n), lrnn(m), lbasis(n), IT-

\text{ERENCE} \] 

WHERE \( m \) IS THE MAX NUMBER OF CONSTRAINTS AND \( n \) IS THE MAX-

\text{NUMBER OF VARIABLES DESIRED.} 

\begin{align*}
& \text{DOCURE PRECISION } \text{BASIC, INBASIC, TOLBASIC, JU-

\text{AL, JDBASIC, I} \\
& \text{Komencl, jclol}(31), \text{run}(10), jclolm}(31) \\
& \text{DIMENSION } a(10, 48), x(10), \text{baseline}(20), \text{calc}(n), \text{ J} \\
& \text{= lmb(x)(m), sou-

\text{r}(e-1, b), \text{uhr}(e-1, b), r(c, c), \text{artlk}(h), \text{npick}(1, h), \text{ npick}(2, h-

\text{= jcalcue}(h), \\
& \text{npick}(3(h), \text{npick}(4(h), \text{split}(3), \text{split}(4), \text{split}(5) \\
& \text{DATA } \text{BASIC/GHBASIC} / \\
& \text{DATA } \text{LOBASIC/GB} / \\
& \text{DATA } \text{MDASIC/GBASAC} / \\
& \text{DATA } \text{TSOLVE/GHSOLVE} / \\
\end{align*}

\text{ITER} = 0

\text{IF} (\text{ISTOP, EQU. 1}) \text{ GO TO 9000} \\
\text{npick}(1) = 0 \\
\text{npick}(1) = 0 \\
\text{npick}(2) = 0 \\
\text{npick}(3) = 0 \\
\text{npick}(4) = 0 \\
\text{split}(1) = 0 \\
\text{split}(2) = 0 \\
\text{split}(3) = 0 \\
\text{split}(4) = 0 \\
\text{split}(5) = 0 \\

\text{READ END NAMS AND NON NEGATIVE RMS PARAMETERS} \\
\text{IF } (\text{STOP, EQU. 1}) \text{ GO TO 20} \\
\text{WRITE (1, 1000)} \\
\text{WRITE (0, 1004)} \\
\text{1004 FORMAT (1X, 'CONSTRANS NAME', 8x, 'RMS VALUE', 7x)} \\
\text{20 WRITE (1, 1005)} \text{RKN}(1), RKN(1) \\
\text{WRITE (0, 1005)} \\
\text{1005 FORMAT (7x, 3x, 'VARIABLE NAME', 3x, 'OBJECTIVE FUNCTION VALUE', 7x)} \\
\text{20 WRITE (1, 1006))} \text{JCOL}(j), \text{AA(ME+1, j)} \\
\text{WRITE (0, 1107)} \\
\text{IF } (\text{STOP, EQU. 1}) \text{ GO TO 2701} \\
\text{WRITE (0, 1107)}
WRITE(*,99)
99 FORMAT(//,132('*'))

2101 K=2
   M1=NVAR+1
   DO 120 J=1,NVAR
   AK(MEQ+2,J)=0.
   DO 120 I=1,MEQ
120  AK(MEQ+2,J)=AK(MEQ+2,J)+AK(I,J)
   DO 110 I=1,MEQ
   NPLUSI=NVAR+1
   AK(I,NPLUSI)=1.
   IBASIS(I)=0
110  bK(MEQ+2)=bK(MEQ+2)+bK(I)
   UPS=0.
   NPLUSK=MEQ+K
400  DO 410 J=1,NVAR
405   IF(AK(MPLUSK,J)-UPS)410,410,420
420  UPS=AK(MPLUSK,J)
   JPIV=J
410  CONTINUE
   IF((OPS=1.0E-06)501,501,450
450  KATMIN=1.0E+06
   JPIV=MEQ+3
   DO 470 I=1,MEQ
   IF(AK(I,JPIV)).LE.1.0E-06)GO TO 470
   RATIO=bK(I)/AK(I,JPIV)
   IF(RATIO.GE.KATMIN)GO TO 470
   KATMIN=RATIO
   JPIV=I
470  CONTINUE
   IF(K.EQ.2)GO TO 476
   DO 475 I=1,MEQ
   IF(IBASIS(I),NE.0)GO TO 475
   IF(ABS(AK(I,JPIV)).LE.1.0E-06)GO TO 475
   JPIV=I
475  CONTINUE
476  PIVOT=AK(IPIV,JPIV)
   IBASIS(IPIV)=JPIV
1TER$=ITER$+1
   IF(IPIV.EQ.MEQ+3)GO TO 496
   DO 500 I=1,NPLUSK
   IF(I.EQ.IPIV)GO TO 500
   DO 490 J=1,NM
   IF(J.EQ.JPIV)GO TO 490
   AK(I,J)=AK(I,J)-AK(I,JPIV)*AK(IPIV,J)/PIVOT
490  CONTINUE
   bK(I)=bK(I)-AK(I,JPIV)*bK(IPIV)/PIVOT
   AK(I,JPIV)=0.
500  CONTINUE
   DO 495 J=1,NM

495 AK(IPIV,J)=AK(IPIV,J)/PIVOT
   &K(IPIV)=&K(IPIV)/PIVOT
   GO TO 599
496 NPICK3(LL)=1
   ZIMBUX(LL)=0.
   URFUN(NCNTW,LL)=100000000.
   IF(NBIG.EQ.0) GO TO 9001
   WRITE(6,1006)
9001 GO TO 571
501 IF(K.EQ.1) GO TO 510
   IF(&K(MEQ+2)=1.0E-03) 504, 504, 505
505 NPICK4(LL)=1
   ZIMBUX(LL)=0.
   URFUN(NCNTW,LL)=100000000.
   IF(NBIG.EQ.0) GO TO 9002
   WRITE(6,1007)
9002 GO TO 571
504 K=1
   GO TO 399
310 CONTINUE
   IF(NBIG.EQ.0) GO TO 2702
   WRITE(6,1000)
1000 FORMAT('1',152('*'),//)
   WRITE(6,1001)PREM(LL)
1001 FORMAT('/5X,'\LINEAR PROGRAMMING SOLUTION'//,5X,'\EFFICIENCY OF REMO\')
   600 VAL (FRACTION) = '1',F4.2)
      IF(ISSS.EQ.0) GO TO 2
      IF(ISBD.EQ.0) GO TO 3
      IF(ISNI.EQ.0) GO TO 4
      IF(ISPH.EQ.0) GO TO 5
   2 WRITE(6,6)
   6 WRITE(6,6)
      FORMAT('/5X,'POLLUTANT IS SUSPENDED SOLIDS')
   2 GO TO 101
   3 WRITE(6,7)
   7 FORMAT('/5X,'POLLUTANT IS BOD')
   2 GO TO 101
   4 WRITE(6,8)
   8 FORMAT('/5X,'POLLUTANT IS NITROGEN')
   2 GO TO 101
   5 WRITE(6,9)
   9 FORMAT('/5X,'POLLUTANT IS PHOSPHURUS')
C NUMBER OF EQUATIONS, NUMBER OF VARIABLES
   101 WRITE(6,1002)MEQ,NVAR
1002 FORMAT('/5X,'\NUMBER EQUATIONS =',I4,/,5X,'\NUMBER VARIABLES =',I4,/
   3)
      WRITE(6,1008)ITERS
2702 ZIMBUX(LL)=-&K(MEQ+1)
   CALCUL normalized LL)=ZIMBUX(LL)*(-1.)
URFUN(NCNzw,LL)=CALCUE(ML)
IF(NB16.EQ.0) GO TO 2703
WRITE(6,1010)Z1MB0A(ML)
WRITE(6,1011)
2702 DO 560 J=1,NVAR
JCOLJ=JCOL(J)
DELTJ=AK(MEU+1,J)
DO 520 I=1,MEO
II=I
IF(IBASIS(I),JU,J) GO TO 550
520 CONTINUE
X=V.
JBASIC=LDBASIC
GO TO 560
550 X=EU(I)
JBASIC=KBASIC
IF(ISLC.EQ.0.AND.ISLS.EQ.0) GO TO 11
IF(ISLC.EQ.1.AND.ISLS.EQ.1.AND.ISLB.EQ.0) GO TO 12
IF(ISLC.EQ.1.AND.ISLS.EQ.1.AND.ISLB.EQ.1) GO TO 13
IF(ISLC.EQ.0.AND.ISLS.EQ.1.AND.ISLB.EQ.0) GO TO 14
IF(ISLC.EQ.1.AND.ISLS.EQ.0.AND.ISLB.EQ.0) GO TO 15
IF(ISLC.EQ.1.AND.ISLS.EQ.1.AND.ISLB.EQ.0) GO TO 16
IF(JCOLJ.EQ.JCU(1).OR.JCOLJ.EQ.JCU(2).OR.JCOLJ.EQ.JCU(3)) OR(JC)
2LJ.EQ.JCU(4).OR.JCOLJ.EQ.JCU(5)) GO TO 560
560 IF(JCOLJ.EQ.JCU(6).OR.JCOLJ.EQ.JCU(7).OR.JCOLJ.EQ.JCU(8).OR(JC)
2LJ.EQ.JCU(9).OR.JCOLJ.EQ.JCU(10)) GO TO 560
560 IF(JCOLJ.EQ.JCU(11).OR.JCOLJ.EQ.JCU(12).OR.JCOLJ.EQ.JCU(13)) OR.
5JCOLJ.EQ.JCU(14).OR.JCOLJ.EQ.JCU(15)) GO TO 7000
50 TO 560
11 IF(JCOLJ.EQ.JCU(1).OR.JCOLJ.EQ.JCU(2).OR.JCOLJ.EQ.JCU(3).OR(JC)
3LJ.EQ.JCU(4).OR.JCOLJ.EQ.JCU(5)) GO TO 17
50 TO 560
17 NPK2(LL)=1
SPLIT2(LL)=SPLIT2(LL)+X
50 TO 560
12 IF(JCOLJ.EQ.JCU(1).OR.JCOLJ.EQ.JCU(2).OR.JCOLJ.EQ.JCU(3)) OR(JC)
3LJ.EQ.JCU(4).OR.JCOLJ.EQ.JCU(5)) GO TO 16
21 IF(JCOLJ.EQ.JCU(6).OR.JCOLJ.EQ.JCU(7).OR.JCOLJ.EQ.JCU(8).OR(JC)
2LJ.EQ.JCU(9).OR.JCOLJ.EQ.JCU(10)) GO TO 19
50 TO 560
16 NPK2(LL)=1
SPLIT1(LL)=SPLIT1(LL)+X
50 TO 21
19 NPK2(LL)=1
SPLIT2(LL) = SPLIT2(LL) + X
GO TO 560
13 IF(JCOLJ.EU.JCUL(1)).OR.JCOLJ.EU.JCUL(2)).OR.JCOLJ.EU.JCUL(3)).OR.JC0
2LJ.EU.JCUL(4)).OR.JCOLJ.EU.JCUL(5)). GU TO 22
23 IF(JCOLJ.EU.JCUL(6)).OR.JCOLJ.EU.JCUL(7)).OR.JCOLJ.EU.JCUL(8)).OR.JC0
2LJ.EU.JCUL(9)).OR.JCOLJ.EU.JCUL(10)). GU TO 24
GO TO 560
22 NPICK(LL) = 1
SPLIT(LL) = SPLIT(LL) + X
GO TO 23
24 NPICK2(LL) = 1
SPLIT2(LL) = SPLIT2(LL) + X
GO TO 560
14 IF(JCOLJ.EU.JCUL(1)).OR.JCOLJ.EU.JCUL(2)).OR.JCOLJ.EU.JCUL(3)).OR.JC0
3LJ.EU.JCUL(4)).OR.JCOLJ.EU.JCUL(5)). GU TO 25
GO TO 560
25 NPICK1(LL) = 1
SPLIT1(LL) = SPLIT1(LL) + X
GO TO 560
15 IF(JCOLJ.EU.JCUL(1)).OR.JCOLJ.EU.JCUL(2)).OR.JCOLJ.EU.JCUL(3)).OR.JC0
5LJ.EU.JCUL(4)).OR.JCOLJ.EU.JCUL(5)). GU TO 26
GO TO 560
26 NPICK(LL) = 1
SPLIT(LL) = SPLIT(LL) + X
GO TO 560
16 IF(JCOLJ.EU.JCUL(1)).OR.JCOLJ.EU.JCUL(2)).OR.JCOLJ.EU.JCUL(3)).OR.JC0
4LJ.EU.JCUL(4)).OR.JCOLJ.EU.JCUL(5)). GU TO 27
28 IF(JCOLJ.EU.JCUL(6)).OR.JCOLJ.EU.JCUL(7)).OR.JCOLJ.EU.JCUL(8)).OR.JC0
2LJ.EU.JCUL(9)).OR.JCOLJ.EU.JCUL(10)). GU TO 29
GO TO 560
27 NPICK(LL) = 1
SPLIT(LL) = SPLIT(LL) + X
GO TO 28
29 NPICK1(LL) = 1
SPLIT1(LL) = SPLIT1(LL) + X
GO TO 560
5000 NPICK(LL) = 1
SPLIT(LL) = SPLIT(LL) + X
GO TO 5600
6000 NPICK1(LL) = 1
SPLIT1(LL) = SPLIT1(LL) + X
GO TO 5601
7000 NPICK2(LL) = 1
SPLIT2(LL) = SPLIT2(LL) + X
560  IF(NBIG.EQ.0) GO TO 580
   WRITE(*,1009)ICOLJ,JBASEIC,X,DELTAJ
580  CONTINUE
561  IF(NBIG.EQ.0) GO TO 2704
   WRITE(*,6,1012)
2704  GO TO 570 I=1,MEQ
   JBASEIC=NBASEIC
   IRUMI=IRUM(I)
   NPLUSI=NVAR+I
   X=-AK(MEQ+1,NPLUSI)
   IF(ABS(X).LE.1.E-09)562,562,000
562  IF(IBASIS(I)).NE.4,5,64
563  JBASEIC=LBASEIC
   FLOWER=0.
   FUPPER=0.
   GO TO 569
564  JBASEIC=NBASEIC
600  FLOWER=1.0E+10
   FUPPER=1.0E+10
   DO 900 K=1,MEQ
      IF(AK(K,NPLUSI)).GT.01,900,008
501  QUOT=-BK(K)/AK(K,NPLUSI)
   IF(QUOT.GE.FUPPER) GO TO 10 900
   FUPPER=QUOT
   GO TO 900
605  QUOT=-BK(K)/AK(K,NPLUSI)
   IF(QUOT.LE.FLOWER) GO TO 10 900
   FLOWER=QUOT
900  CONTINUE
   FLOWER=FLOWER
   IF(FLOWER.LT.1.0E+10.AND.FUPPER.LT.1.0E+10) GO TO 574
   IF(FLOWER.LT.1.0E+10.AND.FUPPER.EQ.1.0E+10) GO TO 576
   IF(FLOWER.EQ.1.0E+10.AND.FUPPER.EQ.1.0E+10) GO TO 572
569  IF(NBIG.EQ.0) GO TO 2706
   WRITE(*,1013)IRUMI,JBASEIC,X,FLOWER,FUPPER
2706  GO TO 570
572  IF(NBIG.EQ.0) GO TO 2707
   WRITE(*,6,1017)IRUMI,JBASEIC,X
2707  GO TO 570
574  IF(NBIG.EQ.0) GO TO 2708
   WRITE(*,6,1018)IRUMI,JBASEIC,X,FUPPER
2708  GO TO 570
576  IF(NBIG.EQ.0) GO TO 570
   WRITE(*,6,1019)IRUMI,JBASEIC,X,FLOWER
570  CONTINUE
   IF(NBIG.EQ.0) GO TO 571
   WRITE(*,6,99)
571  ISOP=1
   GO TO 1
1006  FORMAT(19H SOLUTION UNBOUNDED)
169

1001 FORMAT(21H NO FEASIBLE SOLUTION)
1000 FORMAT(23H SOLUTION OPTIMAL AFTER,2X,I5,11H ITERATIONS-/
/)
1009 FORMAT(2X,A8,2X,A8,5X,E13.6,2X,E13.6)
1010 FORMAT(20H MAXIMAL OBJECTIVE =,E16.6,/
/)
1011 FORMAT(2X,8H VARIABLE,2X,6H STATUS,12X,SHVALUE,8X,8HELT-
AJ)
1012 FORMAT('UCONSTRAINT',3X,6H STATUS,11X,SHVALUE,9X,8HECR-
CREASE,9X,8MIN)
1013 FORMAT(4X,A8,2X,A8,4X,E13.6,4X,E13.6)
1017 FORMAT(4X,A8,2X,A8,4X,E13.6,4X,4HOPEN,12X,4HOPEN)
1018 FORMAT(4X,A8,2X,A8,4X,E13.6,8X,4HOPEN,8X,E13.6)
1103 FORMAT(1X,A8,12X,E13.6)
1104 FORMAT(1X,AD,2X,E13.6)
1105 FORMAT(1X,SHSOLVE)
9000 RETURN
END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCNNNT
SUBROUTINE TABLOX CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCNNNT
SUBROUTINE TABLOX(PMAXS,PMAXB,PMAXN,PMAXP,SSUM,OBFUN,N-
CITRT,ISSS,IS
5D0,ISNI,ISPH,NSOL,NB1G,AKCCM,IMHELP,IMHELP1,IMHELP2,IMHELP-
3,IMHELP4,THE
4LP3,IMHELP6,IMHELP7,IMHELP8,IMHELP9,IMHELPA,IMHELPB,IMHELP,
C
8,IMHELPF,IMHELPG,IMHELPH,IMHELPJ,UG,UG1,U2,UG3,UG4-
,UG5,UG6,UG7
3,UG8,UG9,UG10,UG11,UG12,UG13,UG14,UG15,UG16,UG17,UG18,
UG19)
DIMENSION BK(103),AK(103,302),SSUM(20,6),OBFUN(20,6),S-
L(100),IBASI
35(103)
REAL*8 IRUN(101)
REAL*8 JCOL(201)
DATA (IRUN(I),I=1,31)
6 /'SG1MAT1','SG2MAT1','SG3MAT1','SG4-
2T1','SG1MAT2','SG2MAT2','SG3MAT2','SG4MAT5','SG5MAT5',/
5,'SG5WA13','SG3WA13','SG4WA13','SG5WA13','SG6WA14','SG2WA-
51','SG4MAT4','SG5WA14','SG6WA15','SG3WA15','SG-
7MAT5','SG1WA16','SG2WA16','SG3WA16','SG4WA16','SG5WA16-
1A16','SG1WA7'/
DATA (IRUN(I),I=32,6U)
4 /'SG2MAT7','SG3MAT7','SG4MAT7','SG-
4WA7','SG1W
DATA (IRUN(I), I=01, 08)
2 /'G1WAT13', 'G2WAT13', 'G3WAT13', -
   'G4WAT13',
4 'G5WAT13', 'G1WAT14', 'G2WAT14', 'G3WAT14', -
   'G5WAT14',
4 'G1WAT15', 'G2WAT15', 'G3WAT15', 'G4WAT15', -
   'G5WAT15',
4 'G2WAT16', 'G3WAT16', 'G4WAT16', 'G5WAT16', -
   'G1WAT17',
4 'G5WAT17', 'G4WAT17', 'G5WAT17', 'G1WAT18', -
   'G2WAT18',
   'G3WAT18'/
DATA (IRUN(I), I=09, 101)
5 /'G4WAT18', 'G5WAT18', 'G1WAT19', -
   'G2WAT19',
2 'G3WAT19', 'G4WAT19', 'G5WAT19', 'G1WAT20', -
   'G2WAT20',
3 'G4WAT20', 'G5WAT20', 'G5WMum'/
DATA (JCAL(I), I=1, 50)
7 /'X11', 'X12', 'X13', 'X14', 'X15', 'X21-',
   'X22', 'X23'
2 '/X24', 'X25', 'X31', 'X32', 'X33', 'X34', 'X35', 'X41', 'X42-',
   'X43', 'X44'
7 '/X45', 'X51', 'X52', 'X53', 'X54', 'X55', 'X61', 'X62', 'X63-',
   'X64', 'X65'
3 '/X71', 'X72', 'X73', 'X74', 'X75', 'X81', 'X82', 'X83', 'X84-',
   'X85', 'X91'
3 '/X92', 'X93', 'X94', 'X95', 'X101', 'X102', 'X103', 'X104-',
   'X105'/
DATA (JCAL(T), I=51, 94)
2 '/X111', 'X112', 'X113', 'X114', 'X115-',
   'X116', 'X117'
3 'X122', 'X123', 'X124', 'X125', 'X131', 'X132', 'X133', 'X134-',
   'X135', 'X141'
3 '/X142', 'X143', 'X144', 'X145', 'X151', 'X152', 'X153', 'X15-',
   'X155', 'X161'
7 '/X162', 'X163', 'X164', 'X165', 'X171', 'X172', 'X173-',
   'X174', 'X175'
8 '/X181', 'X182', 'X183', 'X184', 'X185', 'X191', 'X192', 'X1-',
   'X193', 'X194'/
DATA (JCAL(I), I=95, 134)
5 '/X195', 'X201', 'X202', 'X203', 'X20-',
   'X204', 'X205', 'X8
DATA (JCOL(I), I=135, 172)
   /'SLK34', 'SLK35', 'SLK36', 'SLK37-', 'SLK38', 'SLK39', 'SLK40', 'SLK41', 'SLK42', 'SLK43', 'SLK44', 'SLK45', 'SLK46', 'SLK47', 'SLK48', 'SLK49', 'SLK50', 'SLK51', 'SLK52', 'SLK53', 'SLK54', 'SLK55'
   /'SLK56', 'SLK57', 'SLK58', 'SLK59', 'SLK60', 'SLK61', 'SLK62', 'SLK63', 'SLK64', 'SLK65', 'SLK66', 'SLK67', 'SLK68', 'SLK69', 'SLK70', 'SLK71'/

DATA (JCOL(I), I=173, 201)
   /'SLK72', 'SLK73', 'SLK74', 'SLK75-', 'SLK76', 'SLK77', 'SLK78', 'SLK79', 'SLK80', 'SLK81', 'SLK82', 'SLK83', 'SLK84', 'SLK85', 'SLK86', 'SLK87', 'SLK88', 'SLK89', 'SLK90', 'SLK91', 'SLK92', 'SLK93', 'SLK94', 'SLK95', 'SLK96', 'SLK97', 'SLK98', 'SLK99', 'SLK100'/

MEQX=(5*NCNTW)+1
NVARX=((5*NCNTW)*2)+1
NM=MEQX+NVARX
MPLUS2=MEQX+2
DO 95 I=1, MPLUS2
1BA51S(I)=0
AK(I)=0.
DO 95 J=1,NM
95 AK(I,J)=0.

AK(I,J)=SSUM(I,1)
AK(2)=SSUM(I,2)-SSUM(I,1)
AK(3)=SSUM(I,3)-SSUM(I,2)
AK(4)=SSUM(I,4)-SSUM(I,3)
AK(5)=SSUM(I,5)-SSUM(I,4)
NSUM=1
II=1
NP=0
MEQX4=6
IF(MEQX.EQ.6) GO TO 1
AK(6)=SSUM(2,1)
AK(7)=SSUM(2,2)-SSUM(2,1)
AK(8)=SSUM(2,3)-SSUM(2,2)
BK(9) = SUM(2,4) - SUM(2,3)  
BK(10) = SUM(2,5) - SUM(2,4)  
NSUB = 2  
IT = 6  
NP = 5  
MEUX4 = 16  
IF (MEOX.EQ.11) GO TO 1  
BK(11) = SUM(3,1)  
BK(12) = SUM(3,2) - SUM(3,1)  
BK(13) = SUM(3,3) - SUM(3,2)  
BK(14) = SUM(3,4) - SUM(3,3)  
BK(15) = SUM(3,5) - SUM(3,4)  
NSUB = 3  
IT = 11  
NP = 10  
MEUX4 = 26  
IF (MEOX.EQ.16) GO TO 1  
BK(16) = SUM(4,1)  
BK(17) = SUM(4,2) - SUM(4,1)  
BK(18) = SUM(4,3) - SUM(4,2)  
BK(19) = SUM(4,4) - SUM(4,3)  
BK(20) = SUM(4,5) - SUM(4,4)  
NSUB = 4  
IT = 16  
NP = 15  
MEUX4 = 36  
IF (MEOX.EQ.21) GO TO 1  
BK(21) = SUM(5,1)  
BK(22) = SUM(5,2) - SUM(5,1)  
BK(23) = SUM(5,3) - SUM(5,2)  
BK(24) = SUM(5,4) - SUM(5,3)  
BK(25) = SUM(5,5) - SUM(5,4)  
NSUB = 5  
IT = 21  
NP = 20  
MEUX4 = 46  
IF (MEOX.EQ.26) GO TO 1  
BK(26) = SUM(6,1)  
BK(27) = SUM(6,2) - SUM(6,1)  
BK(28) = SUM(6,3) - SUM(6,2)  
BK(29) = SUM(6,4) - SUM(6,3)  
BK(30) = SUM(6,5) - SUM(6,4)  
NSUB = 6  
IT = 26  
NP = 25  
MEUX4 = 56  
IF (MEOX.EQ.31) GO TO 1  
BK(31) = SUM(7,1)  
BK(32) = SUM(7,2) - SUM(7,1)  
BK(33) = SUM(7,3) - SUM(7,2)
\[ \begin{align*}
&\text{rk}(54) = \text{som}(7, 4) - \text{som}(7, 3) \\
&\text{rk}(53) = \text{som}(7, 3) - \text{som}(7, 4) \\
&\text{rk}(52) = \text{som}(6, 2) - \text{som}(6, 1) \\
&\text{rk}(51) = \text{som}(6, 1) - \text{som}(6, 2) \\
&\text{rk}(50) = \text{som}(5, 0) - \text{som}(5, 1) \\
&\text{rk}(49) = \text{som}(5, 1) - \text{som}(5, 0) \\
&\text{rk}(48) = \text{som}(4, 2) - \text{som}(4, 1) \\
&\text{rk}(47) = \text{som}(4, 1) - \text{som}(4, 2) \\
&\text{rk}(46) = \text{som}(4, 2) - \text{som}(4, 3) \\
&\text{rk}(45) = \text{som}(4, 3) - \text{som}(4, 2) \\
&\text{rk}(44) = \text{som}(4, 3) - \text{som}(4, 4) \\
&\text{rk}(43) = \text{som}(4, 4) - \text{som}(4, 3) \\
&\text{rk}(42) = \text{som}(4, 4) - \text{som}(4, 5) \\
&\text{rk}(41) = \text{som}(4, 5) - \text{som}(4, 4) \\
&\text{rk}(40) = \text{som}(4, 5) - \text{som}(4, 6) \\
&\text{rk}(39) = \text{som}(3, 2) - \text{som}(3, 1) \\
&\text{rk}(38) = \text{som}(3, 1) - \text{som}(3, 2) \\
&\text{rk}(37) = \text{som}(3, 2) - \text{som}(3, 3) \\
&\text{rk}(36) = \text{som}(3, 3) - \text{som}(3, 2) \\
&\text{rk}(35) = \text{som}(3, 3) - \text{som}(3, 4) \\
&\text{rk}(34) = \text{som}(3, 4) - \text{som}(3, 3) \\
&\text{rk}(33) = \text{som}(3, 4) - \text{som}(3, 5) \\
&\text{rk}(32) = \text{som}(3, 5) - \text{som}(3, 4) \\
&\text{rk}(31) = \text{som}(3, 5) - \text{som}(3, 6) \\
&\text{rk}(30) = \text{som}(3, 6) - \text{som}(3, 5) \\
&\text{rk}(29) = \text{som}(2, 2) - \text{som}(2, 1) \\
&\text{rk}(28) = \text{som}(2, 1) - \text{som}(2, 2) \\
&\text{rk}(27) = \text{som}(2, 2) - \text{som}(2, 3) \\
&\text{rk}(26) = \text{som}(2, 3) - \text{som}(2, 2) \\
&\text{rk}(25) = \text{som}(2, 3) - \text{som}(2, 4) \\
&\text{rk}(24) = \text{som}(2, 4) - \text{som}(2, 3) \\
&\text{rk}(23) = \text{som}(2, 4) - \text{som}(2, 5) \\
&\text{rk}(22) = \text{som}(2, 5) - \text{som}(2, 4) \\
&\text{rk}(21) = \text{som}(2, 5) - \text{som}(2, 6) \\
&\text{rk}(20) = \text{som}(2, 6) - \text{som}(2, 5) \\
&\text{rk}(19) = \text{som}(1, 2) - \text{som}(1, 1) \\
&\text{rk}(18) = \text{som}(1, 1) - \text{som}(1, 2) \\
&\text{rk}(17) = \text{som}(1, 2) - \text{som}(1, 3) \\
&\text{rk}(16) = \text{som}(1, 3) - \text{som}(1, 2) \\
&\text{rk}(15) = \text{som}(1, 3) - \text{som}(1, 4) \\
&\text{rk}(14) = \text{som}(1, 4) - \text{som}(1, 3) \\
&\text{rk}(13) = \text{som}(1, 4) - \text{som}(1, 5) \\
&\text{rk}(12) = \text{som}(1, 5) - \text{som}(1, 4) \\
&\text{rk}(11) = \text{som}(1, 5) - \text{som}(1, 6) \\
&\text{rk}(10) = \text{som}(1, 6) - \text{som}(1, 5) \\
&\text{rk}(9) = \text{som}(0, 2) - \text{som}(0, 1) \\
&\text{rk}(8) = \text{som}(0, 1) - \text{som}(0, 2) \\
&\text{rk}(7) = \text{som}(0, 2) - \text{som}(0, 3) \\
&\text{rk}(6) = \text{som}(0, 3) - \text{som}(0, 2) \\
&\text{rk}(5) = \text{som}(0, 3) - \text{som}(0, 4) \\
&\text{rk}(4) = \text{som}(0, 4) - \text{som}(0, 3) \\
&\text{rk}(3) = \text{som}(0, 4) - \text{som}(0, 5) \\
&\text{rk}(2) = \text{som}(0, 5) - \text{som}(0, 4) \\
&\text{rk}(1) = \text{som}(0, 5) - \text{som}(0, 6) \\
&\text{rk}(0) = \text{som}(0, 6) - \text{som}(0, 5) \\
\end{align*} \]
OKL5 = SUM(12, 4) - SUM(12, 3)
OK(60) = SUM(12, 5) - SUM(12, 4)
NSUB = 12
IT = 56
NP = 55
MEwX4 = 116
IF(MEX. EQ. 61) GU TO 1
OK(61) = SUM(13, 1)
OK(62) = SUM(13, 2) - SUM(13, 1)
OK(63) = SUM(13, 3) - SUM(13, 2)
OK(64) = SUM(13, 4) - SUM(13, 3)
OK(65) = SUM(13, 5) - SUM(13, 4)
NSUB = 13
IT = 61
NP = 60
MEwX4 = 126
IF(MEX. EQ. 66) GU TO 1
OK(66) = SUM(14, 1)
OK(67) = SUM(14, 2) - SUM(14, 1)
OK(68) = SUM(14, 3) - SUM(14, 2)
OK(69) = SUM(14, 4) - SUM(14, 3)
OK(70) = SUM(14, 5) - SUM(14, 4)
NSUB = 14
IT = 60
NP = 65
MEwX4 = 136
IF(MEX. EQ. 71) GU TO 1
OK(71) = SUM(15, 1)
OK(72) = SUM(15, 2) - SUM(15, 1)
OK(73) = SUM(15, 3) - SUM(15, 2)
OK(74) = SUM(15, 4) - SUM(15, 3)
OK(75) = SUM(15, 5) - SUM(15, 4)
NSUB = 15
IT = 71
NP = 70
MEwX4 = 146
IF(MEX. EQ. 76) GU TO 1
OK(76) = SUM(16, 1)
OK(77) = SUM(16, 2) - SUM(16, 1)
OK(78) = SUM(16, 3) - SUM(16, 2)
OK(79) = SUM(16, 4) - SUM(16, 3)
OK(80) = SUM(16, 5) - SUM(16, 4)
NSUB = 16
IT = 76
NP = 75
MEwX4 = 156
IF(MEX. EQ. 81) GU TO 1
OK(81) = SUM(17, 1)
OK(82) = SUM(17, 2) - SUM(17, 1)
OK(83) = SUM(17, 3) - SUM(17, 2)
OK(84) = S\text{SUM}(17,4) - S\text{SUM}(17,3)
OK(85) = S\text{SUM}(17,5) - S\text{SUM}(17,4)
I = 81
NSUB = 17
NP = 80
MEQX4 = 166
1F(MEQX1.EQ.8b) GU TO 1
OK(8b) = S\text{SUM}(18,1)
OK(87) = S\text{SUM}(18,2) - S\text{SUM}(18,1)
OK(88) = S\text{SUM}(18,3) - S\text{SUM}(18,2)
OK(89) = S\text{SUM}(18,4) - S\text{SUM}(18,3)
OK(90) = S\text{SUM}(18,5) - S\text{SUM}(18,4)
NSUB = 16
I = 8b
NP = 85
MEQX4 = 176
1F(MEQX1.EQ.91) GU TO 1
OK(91) = S\text{SUM}(19,1)
OK(92) = S\text{SUM}(19,2) - S\text{SUM}(19,1)
OK(93) = S\text{SUM}(19,3) - S\text{SUM}(19,2)
OK(94) = S\text{SUM}(19,4) - S\text{SUM}(19,3)
OK(95) = S\text{SUM}(19,5) - S\text{SUM}(19,4)
NSUB = 19
I = 91
NP = 90
MEQX4 = 186
1F(MEQX1.EQ.9b) GU TO 1
OK(9b) = S\text{SUM}(20,1)
OK(97) = S\text{SUM}(20,2) - S\text{SUM}(20,1)
OK(98) = S\text{SUM}(20,3) - S\text{SUM}(20,2)
OK(99) = S\text{SUM}(20,4) - S\text{SUM}(20,3)
OK(100) = S\text{SUM}(20,5) - S\text{SUM}(20,4)
NSUB = 20
I = 9b
NP = 95
MEQX4 = 196
1RUN(MEQX) = IRUN(101)
1F(ISSS.EQ.0) GO TO 53
1F(ISBU.EQ.0) GO TO 52
1F(ISUN.EQ.0) GO TO 51
1F(ISPH.EQ.0) GO TO 50
53 OK(MEQX) = PMAXS
GO TO 20
52 OK(MEQX) = PMAXB
GO TO 20
51 OK(MEQX) = PMAXN
GO TO 20
50 OK(MEQX) = PMAXP
20 GO 21 I = I, MEQX4
21 JCU(L(I+5)) = JCU(L(I+100)-NP)
\[ \text{SL}(1) = \text{SUM}(1, 1) / \text{SUM}(1, 1) \]

\[ \text{SL}(1+1) = (\text{SUM}(1, 1+1) - \text{SUM}(1, 1)) / \text{SUM}(1, 1+1) - \text{SUM}(1, 1) \]

IF (SL(1+1), 0, 0) GO TO 9994

\[ \text{SL}(1) = 100000000 \]

CONTINUE

IF (SL(1+1), 0, 0) GO TO 9997

\[ \text{SL}(1) = 0 \]

IF (SL(1+1), 0, 0) GO TO 9994

\[ \text{SL}(1+1) = 100000000 \]

CONTINUE

IF (SL(1+1), 0, 0) GO TO 9995

\[ \text{SL}(1) = 0 \]

IF (SL(1+1), 0, 0) GO TO 9992

\[ \text{SL}(1+1) = 100000000 \]

CONTINUE

IF (SL(1+1), 0, 0) GO TO 9992

\[ \text{SL}(1) = 0 \]

IF (SL(1+1), 0, 0) GO TO 9995

\[ \text{SL}(1+1) = 100000000 \]

CONTINUE

IF (SL(1+1), 0, 0) GO TO 9992

\[ \text{SL}(1) = 0 \]
GO TU 28
SL(I+20)=10000000.
20 CONTINUE
IF(MEQX.EQ.31) GO TO 2
SL(31)=0BFUN(7,1)/SSUM(7,1)
DO 29 I=1,4
SL(I+31)=(UBFUN(7,I+1)-0BUN(7,I))/(SSUM(7,I+1)-SSUM(7-I))
IF(SL(I+31).EQ.0.) GO TO 7000
GO TO 29
7000 SL(I+31)=10000000.
29 CONTINUE
IF(MEQX.EQ.36) GO TO 2
SL(36)=0BUN(8,1)/SSUM(8,1)
DO 30 I=1,4
SL(I+36)=(UBFUN(8,I+1)-0BUN(8,I))/(SSUM(8,I+1)-SSUM(8-I))
IF(SL(I+36).EQ.0.) GO TO 6999
GO TO 30
6999 SL(I+36)=10000000.
30 CONTINUE
IF(MEQX.EQ.41) GO TO 2
SL(41)=0BUN(9,1)/SSUM(9,1)
DO 31 I=1,4
SL(I+41)=(UBFUN(9,I+1)-0BUN(9,I))/(SSUM(9,I+1)-SSUM(9-I))
IF(SL(I+41).EQ.0.) GO TO 6998
GO TO 31
6998 SL(I+41)=10000000.
31 CONTINUE
IF(MEQX.EQ.46) GO TO 2
SL(46)=0BUN(10,1)/SSUM(10,1)
DO 32 I=1,4
SL(I+46)=(UBFUN(10,I+1)-0BUN(10,I))/(SSUM(10,I+1)-SSUM(10-I))
IF(SL(I+46).EQ.0.) GO TO 6995
GO TO 32
6995 SL(I+46)=10000000.
32 CONTINUE
IF(MEQX.EQ.51) GO TO 2
SL(51)=0BUN(11,1)/SSUM(11,1)
DO 33 I=1,4
SL(I+51)=(UBFUN(11,I+1)-0BUN(11,I))/(SSUM(11,I+1)-SSUM(11-I))
IF(SL(I+51).EQ.0.) GO TO 6997
GO TO 33
6997 SL(I+51)=10000000.
33 CONTINUE
IF(MEQX.EQ.56) GO TO 2
SL(56)=0BUN(12,1)/SSUM(12,1)
178

UN34 I=1,4
SL(I+56) = (UBFUN(12, I+1) - UBUFUN(12, I)) / (SSUM(12, I+1) - SSUM(12, I))

IF(SL(I+56) .EQ. 0.) GO TO 6996
GO TO 34

6996 SL(I+56) = 100000000.
34 CONTINUE
IF(MEQX .EQ. 61) GO TO 2
SL(61) = OBUFUN(13, I) / SSUM(13, I)
DO 35 I = 1, 4
SL(I+61) = (UBFUN(13, I+1) - UBUFUN(13, I)) / (SSUM(13, I+1) - SSUM(13, I))

IF(SL(I+61) .EQ. 0.) GO TO 6994
GO TO 35

6994 SL(I+61) = 100000000.
35 CONTINUE
IF(MEQX .EQ. 62) GO TO 2
SL(62) = OBUFUN(14, I) / SSUM(14, I)
DO 36 I = 1, 4
SL(I+66) = (UBFUN(14, I+1) - UBUFUN(14, I)) / (SSUM(14, I+1) - SSUM(14, I))

IF(SL(I+66) .EQ. 0.) GO TO 6993
GO TO 36

6993 SL(I+66) = 100000000.
36 CONTINUE
IF(MEQX .EQ. 71) GO TO 2
SL(71) = OBUFUN(15, I) / SSUM(15, I)
DO 37 I = 1, 4
SL(I+71) = (UBFUN(15, I+1) - UBUFUN(15, I)) / (SSUM(15, I+1) - SSUM(15, I))

IF(SL(I+71) .EQ. 0.) GO TO 6992
GO TO 37

6992 SL(I+71) = 100000000.
37 CONTINUE
IF(MEQX .EQ. 72) GO TO 2
SL(72) = OBUFUN(16, I) / SSUM(16, I)
DO 38 I = 1, 4
SL(I+76) = (UBFUN(16, I+1) - UBUFUN(16, I)) / (SSUM(16, I+1) - SSUM(16, I))

IF(SL(I+76) .EQ. 0.) GO TO 6991
GO TO 38

6991 SL(I+76) = 100000000.
38 CONTINUE
IF(MEQX .EQ. 81) GO TO 2
SL(81) = OBUFUN(17, I) / SSUM(17, I)
DO 39 I = 1, 4
SL(I+81) = (UBFUN(17, I+1) - UBUFUN(17, I)) / (SSUM(17, I+1) - SSUM(17, I))

IF(SL(I+81) .EQ. 0.) GO TO 6990
GO TO 39
SL(I+81) = 100000000.

CONTINUE
IF(MEQX.EQ.86) GO TO 2
SL(86) = 0 OFUN(18,1)/SSUM(18,1)
DO 40 I = 1,4
SL(I+86) = (UBFUN(18,1+1) - UBUF(18,1))/ (SSUM(18,1+1) - SSU-
M(18,1))
IF(SL(I+86).EQ.0.) GO TO 6989
GO TO 40

6989 SL(I+86) = 100000000.

CONTINUE
IF(MEQX.EQ.91) GO TO 2
SL(91) = 0 OFUN(19,1)/SSUM(19,1)
DO 41 I = 1,4
SL(I+91) = (UBFUN(19,1+1) - UBUF(19,1))/ (SSUM(19,1+1) - SSU-
M(19,1))
IF(SL(I+91).EQ.0.) GO TO 6986
GO TO 41

6986 SL(I+91) = 100000000.

CONTINUE
IF(MEQX.EQ.96) GO TO 2
SL(96) = 0 OFUN(20,1)/SSUM(20,1)
DO 42 I = 1,4
SL(I+96) = (UBFUN(20,1+1) - UBUF(20,1))/ (SSUM(20,1+1) - SSU-
M(20,1))
IF(SL(I+96).EQ.0.) GO TO 6987
GO TO 42

6987 SL(I+96) = 100000000.

CONTINUE
2 MEWX1 = MEWX1 - 1
DO 100 I = 1, MEQX1
100 AK(MEQX+1,1) = - SL(I)
DO 200 I = MEQX, NVARX
200 AK(MEQX+1,1) = 0.
DO 300 II = 1, MEWX1
AK(II,II) = 1.
AK(II,II+MEQX) = 1.
AK(MEQX,II) = 1.
300 CONTINUE
AK(MEQX,MEWX) = -1.
MEWX2 = MEWX + 1
IF(NSOL.EQ.0) GO TO 400
WRITE(6,228)
228 FORMAT(1',152(''),/,'INITIAL TABLEAU FROM SUBRO-
UTEIN TABLUX')
5',//)
DO 45 I = 1, MEW2
DO 45 J = 1, NVARX
WRITE(6,500) I, J, AK(I,J)
500 FORMAT(5X,'AK(',I5,')',//)

CONTINUE
WRITE(6,229)
229 FORMAT(/ '*'),/)
400 CALL LPGUX(MEQX,NVARX,IRON,AK,8K,JCUL,SL,NM,MPLUS2,NB-
IG,AKKCM,NSU
AD,HELP,HELP1,HELP2,HELP3,HELP4,HELP5,HELP6,HELP7,HELP8,IH
5ELP9,HELPA,HELPB,HELPC,HELPD,HELPF,HELPG,-
HELPH,HELPJ,UG6,UG1,UG2,UG3,UG4,UG5,UG6,UG7,UG8,UG9,UG10,U-
G11,UG12,UG1
53,UG14,UG15,UG16,UG17,UG18,UG19)
RETURN
END
SUBROUTINE LPGUX(MEQX,NVARX,IRON,AK,8K,JCUL,SL,NM,MPLUS2,NB-
IG,AKKCM,NSU
CM,NSUB,HELP,HELP1,HELP2,HELP3,HELP4,HELP5,HELP6,HELP7,HELP8,
HELP9,HELPA,HELPB,HELPC,HELPD,HELPF,HELPG,-
HELPH,HELPC,UG6,UG1,UG2,UG3,UG4,UG5,UG6,UG7,UG8,UG9,-
UG10,UG11,UG12,UG13,UG14,UG15,UG16,UG17,UG18,UG19)
DOUBLE PRECISION KBASIC,LBASIC,M BASIC,NBASIC,ISOLVE,JCL-
ULJ,BASIC,I
1KOWI,JCUL(201),IRON(101)
DIMENSION AK(103,302),8K(103),IBASIS(103),SL(100)
DATA KBASIC/0HBasic /
DATA LBASIC/0H /
DATA MBASIC/0HBinding /
DATA NBASIC/0HSlack /
DATA ISOLVE/0Hsolve /
1STOP=0
1 ITERS=0
IF(1STOP.GEQ.1) GO TO 9000
HELP=0
HELP1=0
HELP2=0
HELP3=0
HELP4=0
HELP5=0
HELP6=0
HELP7=0
HELP8=0
HELP9=0
HELPA=0
HELPB=0
HELPD=0
HELPF=0
HELPH=0
HELPJ=0
1HELPD=0
1HELPF=0
1HELPG=0
1HELPH=0
1HELPI=0
1HELPJ=0
UG=0.
UG1=0.
UG2=0.
UG3=0.
UG4=0.
UG5=0.
UG6=0.
UG7=0.
UG8=0.
UG9=0.
UG10=0.
UG11=0.
UG12=0.
UG13=0.
UG14=0.
UG15=0.
UG16=0.
UG17=0.
UG18=0.
UG19=0.
1F(NBIG.EQ.0) GO TO 50
WRITE(6,1000)
1000 FORMAT('1',132(' '),/)
WRITE(6,1004)
1004 FORMAT(1X,'CONSTRAINT NAME',6X,'RHS VALUE',/)
DO 10 I=1,MEGX
10 WRITE(6,1103)IRON(I),BA(I)
WRITE(6,1003)
1003 FORMAT(//,5X,'VARIABLE NAME',3X,'OBJECTIVE FUNCTION VALUE',/)
DO 20 J=1,NVARX
20 WRITE(6,1104)JCOL(J),AK(MEGX+1,J)
50 J2=0
1IF(NBIG.EQ.0) GO TO 2700
WRITE(6,1107)
WRITE(6,99)
99 FORMAT(//,132(' '))
2700 K=2
M1=NVARX+1
DO 120 J=1,NVARX
AK(MEQX+2,J)=0.
120 I=1,MEQX
120  \text{AK}([\text{MEQX}+2, J]) = \text{AK}(\text{MEQX}+2, J) + \text{AK}(I, J)
\text{DO} 110 \text{I}=1, \text{MEQX}
\text{NPLUS1}=\text{NVARX}+1
\text{AK}(I, \text{NPLUS1})=1.
\text{IBASIS}(I)=0
110 \text{BK}([\text{MEQX}+2]) = \text{BK}([\text{MEQX}+2]) + \text{BK}(I)
399 \text{UPS}=0.
\text{NPLUSK}=\text{MEQX}+K
400 \text{DO} 410 \text{J}=1, \text{NVARX}
405 \text{IF}(\text{AK}([\text{NPLUSK}, J])=\text{UPS}) \text{GO TO 410}
420 \text{UPS}=\text{AK}(\text{NPLUSK}, J)
\text{JPIV}=J
410 \text{CONTINUE}
\text{IF}(\text{UPS}=1.0 \cdot 10^{-06}) \text{GO TO 501}
450 \text{RATMIN}=1.0 \cdot 10^{-06}
\text{JPIV}=\text{MEQX}+5
\text{DO} 470 \text{I}=1, \text{MEQX}
\text{IF}(\text{AK}(I, \text{JPIV}) \cdot \text{LE} 1.0 \cdot 10^{-06}) \text{GO TO 470}
\text{RTOL}=\text{BK}(I) / \text{AK}(I, \text{JPIV})
\text{IF}(\text{RATIO} \cdot 6.0, \text{RATMIN}) \text{GO TO 470}
\text{RATMIN}=\text{RATIO}
\text{JPIV}=I
470 \text{CONTINUE}
\text{IF}(\text{KEW} \cdot 2) \text{GO TO 476}
\text{DO} 475 \text{I}=1, \text{MEQX}
\text{IF}(\text{IBASIS}(I) \cdot \text{NE} 0) \text{GO TO 475}
\text{IF}(\text{ABS}([\text{AK}(I, \text{JPIV})] \cdot \text{LE} 1.0 \cdot 10^{-06}) \text{GO TO 475}
\text{JPIV}=I
475 \text{CONTINUE}
476 \text{PIV0I}=\text{AK}(\text{JPIV}, \text{JPIV})
\text{IBASIS}(\text{JPIV})=\text{JPIV}
\text{ITERS}=1\text{ITERS}+1
\text{IF}(\text{IPIV} \cdot \text{EQ} \cdot \text{MEQX}+3) \text{GO TO 496}
\text{DO} 500 \text{I}=1, \text{NPLUSK}
\text{IF}(\text{KEW} \cdot \text{IPIV}) \text{GO TO 500}
\text{DO} 480 \text{J}=1, \text{NM}
\text{IF}(\text{KEW} \cdot \text{JPIV}) \text{GO TO 480}
\text{AK}(I, J)=\text{AK}(I, J) - \text{AK}(I, \text{JPIV}) \cdot \text{AK}(\text{IPIV}, J) / \text{PIVOI}
480 \text{CONTINUE}
\text{BK}(I) = \text{BK}(I) - \text{AK}(I, \text{JPIV}) \cdot \text{BK}(\text{IPIV}) / \text{PIVOI}
\text{AK}(I, \text{JPIV}) = 0.
500 \text{CONTINUE}
\text{DO} 495 \text{J}=1, \text{NM}
495 \text{AK}(\text{IPIV}, J)=\text{AK}(\text{IPIV}, J) / \text{PIVOI}
\text{BK}(\text{IPIV})=\text{BK}(\text{IPIV}) / \text{PIVOI}
\text{GO TO 599}
496 \text{RITE}(10, 1006)
\text{GO TO 571}
501 \text{IF}(\text{KEW} \cdot 1) \text{GO TO 510}
\text{IF}([\text{AK}(\text{MEQX}+2)] - 1.0 \cdot 10^{-03}) 504, 504, 505
505 IF(NBIG.EQ.0) GO TO 571
WRITE(6,1007)
GO TO 571
504 K=1
GO TO 599
510 CONTINUE
IF(NBIG.EQ.0) GO TO 9002
WRITE(6,1002)MEQX,NVARX
1002 FORMAT(//5x,'NUMBeR OF EQUATIONS =',I4,/,5x,'NUMBeR OF
VARIABLES =
2',I4,/,)
WRITE(6,1008)ITERS
9002 ZIMBOX=6K(MEQX+1)
ARKCM=ZIMBOX*(-1.)
IF(NBIG.EQ.0) GO TO 9003
WRITE(6,1010)ZIMBOX
WRITE(6,1011)
9003 DO 560 J=1,NVARX
JCOLJ=JCOL(J)
DELTAJ=AK(MEQX+1,J)
DO 520 I=1,MEQX
IT=I
IF(IBASIS(I).EQ.J) GO TO 550
520 CONTINUE
X=0.
JBASEIC=IBASIC
GO TO 560
550 X=AK(11)
JBASEIC=IBASIC
IF(JCOLJ.EQ.JCOL(1).OR.JCOLJ.EQ.JCOL(2).OR.JCOLJ.EQ.JCOL
(3).OR.JCOL
5LJ.EQ.JCOL(4).OR.JCOLJ.EQ.JCOL(5)) GO TO 5000
7005 IF(NSUB.EQ.1) GO TO 560
IF(JCOLJ.EQ.JCOL(6).OR.JCOLJ.EQ.JCOL(7).OR.JCOLJ.EQ.JCOL
(8).OR.JCOL
5LJ.EQ.JCOL(9).OR.JCOLJ.EQ.JCOL(10)) GO TO 5001
7006 IF(NSUB.EQ.2) GO TO 560
IF(JCOLJ.EQ.JCOL(11).OK.JCOLJ.EQ.JCOL(12).OR.JCOLJ.EQ.JCOL
(13).OK.
5JCOLJ.EQ.JCOL(14).OR.JCOLJ.EQ.JCOL(15)) GO TO 5002
7007 IF(NSUB.EQ.3) GO TO 560
IF(JCOLJ.EQ.JCOL(16).OK.JCOLJ.EQ.JCOL(17).OR.JCOLJ.EQ.JCOL
(18).OK.
5JCOLJ.EQ.JCOL(19).OR.JCOLJ.EQ.JCOL(20)) GO TO 5003
7008 IF(NSUB.EQ.4) GO TO 560
IF(JCOLJ.EQ.JCOL(21).OK.JCOLJ.EQ.JCOL(22).OR.JCOLJ.EQ.JCOL
(23).OK.
5JCOLJ.EQ.JCOL(24).OR.JCOLJ.EQ.JCOL(25)) GO TO 5004
7009 IF(NSUB.EQ.5) GO TO 560
IF(JCOLJ.EQ.JCOL(26).OK.JCOLJ.EQ.JCOL(27).OR.JCOLJ.EQ.JCOL
(28).OK.
7010 IF (NSUB . eq . 6) 60 tu 360
          JcULJ(33) . ok .
5JcULJ . eq . JcOLJ(34) . ur . JcOLJ . eq . JcULJ(35) . 60 tu 5006
7011 IF (NSUB . eq . 7) 60 tu 360
          JcULJ(38) . ok .
5JcULJ . eq . JcOLJ(39) . ur . JcOLJ . eq . JcULJ(40) . 60 tu 5007
7012 IF (NSUB . eq . 8) 60 tu 360
          JcULJ(43) . ok .
5JcULJ . eq . JcOLJ(44) . ur . JcOLJ . eq . JcULJ(45) . 60 tu 5008
7013 IF (NSUB . eq . 9) 60 tu 360
          JcULJ(48) . ok .
5JcULJ . eq . JcOLJ(49) . ur . JcOLJ . eq . JcULJ(50) . 60 tu 5009
7014 IF (NSUB . eq . 10) 60 tu 360
          JcULJ(53) . ok .
5JcULJ . eq . JcOLJ(54) . ur . JcOLJ . eq . JcULJ(55) . 60 tu 5010
7015 IF (NSUB . eq . 11) 60 tu 360
          JcULJ(58) . ok .
5JcULJ . eq . JcOLJ(59) . ur . JcOLJ . eq . JcULJ(60) . 60 tu 5011
7016 IF (NSUB . eq . 12) 60 tu 360
          JcULJ(63) . ok .
5JcULJ . eq . JcOLJ(64) . ur . JcOLJ . eq . JcULJ(65) . 60 tu 5012
7017 IF (NSUB . eq . 13) 60 tu 360
          JcULJ(68) . ok .
5JcULJ . eq . JcOLJ(69) . ur . JcOLJ . eq . JcULJ(70) . 60 tu 5013
7018 IF (NSUB . eq . 14) 60 tu 360
          JcULJ(73) . ok .
5JcULJ . eq . JcOLJ(74) . ur . JcOLJ . eq . JcULJ(75) . 60 tu 5014
7019 IF (NSUB . eq . 15) 60 tu 360
          JcULJ(78) . ok .
5JcULJ . eq . JcOLJ(79) . ur . JcOLJ . eq . JcULJ(80) . 60 tu 5015
7020 IF (NSUB . eq . 16) 60 tu 360
          JcULJ(83) . ok .
5JcULJ . eq . JcOLJ(84) . ur . JcOLJ . eq . JcULJ(85) . 60 tu 5016
7021 IF (NSUB . eq . 17) 60 tu 360
          JcULJ(88) . ok .
5JcULJ . eq . JcOLJ(89) . ur . JcOLJ . eq . JcULJ(90) . 60 tu 5017
7022 IF (NSUB . eq . 18) 60 tu 360
185

IF (JCOLJ.EQ.JCUL(91).OR.JCOLJ.EQ.JCUL(92).OR.JCOLJ.EQ.JCUL(93)) G0 TO 5018
5JCULJ.EQ.JCOLJ(94).OR.JCOLJ.EQ.JCUL(95)) G0 TO 5018
7025 IF (NSUB.EQ.19) G0 TO 360
IF (JCOLJ.EQ.JCUL(96).OR.JCOLJ.EQ.JCUL(97).OR.JCOLJ.EQ.JCUL(98)) G0 TO 5019
5JCULJ.EQ.JCOLJ(99).OR.JCOLJ.EQ.JCUL(100)) G0 TO 5019
7024 IF (NSUB.EQ.20) G0 TO 360
5000 IHELP=1
UG=UG+x
G0 Tu 7005
5001 IHELP1=1
UG1=UG1+x
G0 Tu 7006
5002 IHELP2=1
UG2=UG2+x
G0 Tu 7007
5003 IHELP3=1
UG3=UG3+x
G0 Tu 7008
5004 IHELP4=1
UG4=UG4+x
G0 Tu 7009
5005 IHELP5=1
UG5=UG5+x
G0 Tu 7010
5006 IHELP6=1
UG6=UG6+x
G0 Tu 7011
5007 IHELP7=1
UG7=UG7+x
G0 Tu 7012
5008 IHELP8=1
UG8=UG8+x
G0 Tu 7013
5009 IHELP9=1
UG9=UG9+x
G0 Tu 7014
5010 IHELP10=1
UG10=UG10+x
G0 Tu 7015
5011 IHELP11=1
UG11=UG11+x
G0 Tu 7016
5012 IHELP12=1
UG12=UG12+x
G0 Tu 7017
5013 IHELP13=1
UG13=UG13+x
G0 Tu 7018
'1H'("E=1
10 14 = 10' 14 + x
0 n T0 701
5015 '1H'("E=1

O 15 = 1015 + x
5016 '1H'("E=1
10 16 = 1016 + x
5017 '1H'("E=1
10 17 = 1017 + x
5018 '1H'("E=1
10 18 = 1018 + x
5019 '1H'("E=1
10 19 = 1019 + x
5020 IF (fltrg(e, 0)) S0 TO 580
WRITE(x, 100x) LUDL, JUDGIC, a, UELTAJ
580 UNTINUE
5021 IF (fltrg(e, 0)) S0 TO 5904
WRITE(x, 100x)
5904 ON 570 (I=1, 15x)
JUDGIC=RaopTC
10011=Rpua(l)
TPLUS1=NVAKX+1
X=AK(MEX+1, TPLUS1)
TFLAB(X-1, VE-0.5)SBC, SCH, BUC
5905 IF (fltrg(e, 5)) S0 TO 5004, 5003, 5002
5002 JUDGIC=RoopTC
FLOUER=0,
FUPPER=0,
S0 TO 589
5003 JUDGIC=RoopTC
5004 FLOUER=-1, VE+1v
FUPPER=-1, VE+1v
ON 900 K=1, 15x
1F (AK(X, TPLUS1))S0UT, 900, G00
5005 SUUT=-6K(X) /AK(K, TPLUS1)
1F (SUUT, LE, FUPPER) GU 10 900
FUPPER=0,01
900 GU 900
5006 SUUT=-6K(X) /AK(K, TPLUS1)
1F (SUUT, LE, FLOUER) GU 10 900
FLOUER=0,01
900 CONTINUE
5007 FLOUER=FLOUER
1F (FLOUER, Eh, +1, VE+1v, AND, FUPPER, EH, 1, VE+1v) GU TO 578
1F (FLOUER, Eh, +1, VE+1v, AND, FUPPER, Eh, 0, VE+1v) GU TO 578
1F (FLOUER, 0, +1, VE+1v, AND, FUPPER, 0, 1, VE+1v) GU TO 578
5008 IF (fltrg(e, 0)) GU TO 570
WRITE(x, 1013) FLTRG, JUDGIC, X, FLOUER, FUPPER
30 TO 370
5022 IF (fltrg(e, 0)) GU TO 370
WRITE(6,1017)IRORI,JBASIC,X
GO TO 570
574 IF(NBIG.EQ.0) GO TO 570
WRITE(6,1018)IRORI,JBASIC,X,UPPER
GO TO 570
576 IF(NBIG.EQ.0) GO TO 570
WRITE(6,1019)IRORI,JBASIC,X,FLOWER
570 CONTINUE
IF(NBIG.EQ.0) GO TO 571
WRITE(6,99)
571 1STOP=1
GO TO 1
1006 FORMAT(19H SOLUTION UNBOUNDED)
1007 FORMAT(21H NO FEASIBLE SOLUTION)
1008 FORMAT(23H SOLUTION OPTIMAL AFTER,2X,I5,11H ITERATIONS-,/)
1009 FORMAT(2X,A8,2X,A8,5X,E13.6,2X,E13.6)
1010 FORMAT(20H MAXIMAL OBJECTIVE =,E16.6,/)
1011 FORMAT(2X,OHVARIABLE,2X,OHSTATUS,12X,SHVALUE,8X,6HUELT-
AJ)
1012 FORMAT(1UCONSTRAINT',3X,OHSTATUS,11X,SHVALUE,9X,6HUECR-
CREASE,9X,5HIN
1013 FORMAT(4X,A8,2X,A8,4X,E13.6,4X,E13.6,4X,E13.6)
1017 FORMAT(4X,A8,2X,A8,4X,E13.6,6X,4HOPEN,12X,4HOPENJ
1018 FORMAT(4X,A8,2X,A8,4X,E13.6,6X,4HOPEN,8X,E13.6)
1019 FORMAT(4X,A8,2X,A8,4X,E13.6,4X,E13.6,8X,4HOPEN)
1102 FORMAT(1X,A8,12X,E13.6)
1104 FORMAT(11X,A8,2X,E13.6)
1107 FORMAT(1X,5HSOLVE)
9000 RETURN
END
LIST OF REFERENCES


