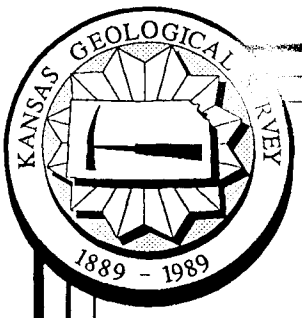


•100 Years of Science and Service•

Development and Testing

of a Linear-integer

Water-resources Planning (Screening) Model

(A Systems Approach to In-basin and Inter-basin Water Transfer)

M. Heidari, T. Maddock, III,
and N. Shaukat

Kansas Geological Survey
Open-File Report #89-13

June 1989

Kansas Geological Survey
Open-file Report

Disclaimer

The Kansas Geological Survey does not guarantee this document to be free from errors or inaccuracies and disclaims any responsibility or liability for interpretations based on data used in the production of this document or decisions based thereon. This report is intended to make results of research available at the earliest possible date, but is not intended to constitute final or formal publication.

Table of Contents

Table of Contents i

List of Figures iv

List of Tables v

I. Abstract 1

II. Water-resources Planning—A Systems-analysis Approach 2

A. What is Systems Analysis? 2

B. Systems Approach and Water-resources Planning 3

C. Water-resources Planning Process 5

1. Input-Output of Water Resources Planning 5

2. Data Requirement for Water-resources Planning 6

a. Classification of Demands 8
i. Water-quantity Demand Classification 8
ii. Water-quality Demand Classification 8

b. Identification of Planning Units 10

c. Identification of Demands 10
i. Existing Demands 10
ii. Future Demand Projections 10

3. Identification of Existing Water-resources Structures 11
a. Reservoirs 11
b. Aqueducts, Canals, and Tunnels 11
c. Ground-water Fields 12

4. Identification of Proposed or Potential Water-resources Structures 12
a. Reservoirs 12
b. Aqueducts, Canals, and Tunnels 12
c. Ground-water Fields 12
d. Water-treatment Plants 13

e. Waste-treatment Plants 13
f. Desalinization Plants. 13

5. Water Supply. 13
a. Surface Water 13
b. Ground Water. 13

III. Water-resources Planning—Methodology 13

A. Assumptions 13

B. Decision Variables. 16

C. Cost Coefficients 17

D. Model Formulation 20

1. Objective Function 20

2. Project Initiation Constraint 21

3. Project Limitation Constraint 22

4. Treated Water Production Constraint. 23

5. Total Water Demand Constraint 23

6. Streamflow Requirement Constraints 24

7. Bounds on Integer Variable Constraints 25

8. Non-negativity Constraint 25

IV. Example Problems. 25

A. Description of the System 25

B. Input Data Preparation and Execution Model. 29

C. Analysis of Results 31

D. Sensitivity Analysis. 36

V. Summary and Concluding Remarks. 40

VI. References 42

VII. Appendices 45

Appendix A 45

Appendix B 52

Appendix C.I 55

Appendix C.II 58

Appendix D.I 62

Appendix D.II 66

Appendix D.III 68

Appendix E 75

Appendix F.I 92

Appendix F.II 95

Appendix G 108

Appendix H 117

Appendix I 139

Appendix J 159

List of Figures

Figure 1. Schematic representation of a multi-basin water-resources planning project 7

Figure 2. Schematic representation of system in region 1 of figure 1 9

Figure 3. Major hydrologic basins of Kansas 15

Figure 4. Schematic representation of a water-resources planning project 26

Figure 5. Projected demand (MGD) vs. time (years) for treated water 28

Figure 6. Results of Planning Period 1 freshwater demand = 53.5 MGD 32

Figure 7. Results of Planning Period 2 freshwater demand = 61.0 MGD 33

Figure 8. Results of Planning Period 3 freshwater demand = 70.2 34

Figure 9. Results of Planning Period 4 freshwater demand = 79.8 MGD Minimum cost = $62.98 million 35

Figure 10. Optimal total cost of the project vs. the capital cost of the pipeline 38

List of Tables

Table 1. Components of the Existing System and Their Costs & Yields in Basin A 27

Table 2. Proposed Components for the System and Their Costs and Yields 27

I. Abstract

Water-resources planning is a multi-disciplinary and dynamic process. For large multi-component and multi-basin systems, this process becomes rather complicated because of the number of options that the planners must consider. Systems-analysis technology can play a major role in screening the number of options with which the planners have to deal and in making the planners capable of repeating the planning process consistently.

In this report, a linear-integer programming model is adopted for the preliminary screening of options for a multi-component and multi-basin water-resources planning effort. The model seeks to minimize the total cost [capital costs (CC) plus operation maintenance and replacement (OMR) costs] of the system over a given time horizon for supplying given functions for treated and untreated water demands subject to physical and minimum streamflow-requirement constraints.

The utility of the model is demonstrated by its application to a two-basin system with several water-production options. How the model may be used to conduct sensitivity analysis on different elements of the system is demonstrated. These results must be checked further using simulation models for both ground water and surface water to make sure that the constraints of the system are not violated over smaller time increments.

II. Water-resources Planning—A Systems-analysis Approach

A. What is Systems Analysis?

The term "Systems Analysis" originally stems from the mathematical analysis of a system of equations. It may be defined as an analytical study that helps a decision maker to identify and select the best course of action among several feasible alternatives. The approach is logical and systematic under well defined assumptions, objectives and criteria. It can significantly aid a decision maker to arrive at better decisions by broadening the information base, by providing a better understanding of the system and linkages of the various subsystems and by predicting the consequences of several alternate courses of actions. It uses all the information available and determines a course of action based on scientific procedures "best" suited to the objectives set by the planners and decision makers. Systems analysis has created new dimensions to the science of policy-planning and decision making.

Under systems-analysis approach, attempts are made to build a replica of a real-world system or situation with the objective of experimenting with the replica to gain insight into the real world problem. The whole system is represented by a series of mathematical expressions (called mathematical model) in such a way that the resulting relationships describe the phenomena. The mathematical model formulated in such a way should be one that can be solved, although the insoluble models have often provided the development of new solution techniques. The model includes the traditional descriptive goal called "objective function" and a series of technical and other side conditions called "constraints." A constraint is usually defined as a condition that can never be violated at any cost or for which there is no gain in overfulfilling it.

The computer revolution has made possible the solution of large and complex mathematical models. The increased capacity for information processing and problem solving in the current era has been a stimulating force in the application of systems analysis to a variety of problems in the field of environmental and resource management.

B. Systems Approach and Water-resources Planning

Since the late 1950's, when the importance of water-resources planning became apparent, a great deal has been achieved in the field of planning and management of these resources. One of the first products of these efforts was publication of the research done in the Harvard Water Program [Maass et al., 1962] and [Hufschmidt and Fiering, 1966]. This program in early stages emphasized the value of systems approach to planning. The analysis of a water resources system in the overall planning process goes through the following four stages [Maass et al., 1962]:

1) Identification of the objectives of system design bearing in mind the political realities,
2) Translation of objectives into measurable criteria,
3) Use of the criteria to devise plans for the development of specific water-resources systems that fulfill the criteria in the highest degree,
4) Evaluation of the consequences of the plans that have been developed.

The early approaches to water-resources planning utilized these steps and used simulation techniques for achieving the best solution possible. Lehigh Basin in Pennsylvania is among the first areas where simulation techniques were employed to achieve the objectives of water planning [Hufschmidt and Fiering, 1966]. Among other simulation models, the Delaware Basin Model [Jacoby and Louks, 1972], Hydrologic Engineering Center (HEC) Model [Beard et al., 1972], SIMLYD-II Model developed for Texas Water System [McBean et al., 1972], HISIM Model developed for Rio Colorado Basin of Argentina [Major and Lentin, 1979] may be mentioned. These models use straight simulation techniques or a combination of simulation and some form of optimization technique to achieve the objectives of planning. Their acceptability for a particular planning process depends on the objectives of the plan. Some of these models are quite elaborate and their application requires a substantial amount of data.

The applicability of the straight simulation models to water planning is somewhat hindered by many alternative schemes which the planners can propose for simulation. In order to reduce the time required to simulate many different alternatives, the use of mathematical programming techniques has been proposed [see for example: Major and Lenton (1979), O'Neil (1972), Moody et al., 1972, Moody et al. (1973), Hall and Dracup (1970), and Heidari, et al. (1988)]. Although these techniques cannot simulate the behavior of a system, they can provide a scheme or a set of schemes most likely to produce the "best" results and therefore make the simulation process less time consuming. These models are generally called the "screening models" because they permit determination of the most promising consideration of design elements on the basis of model sensitivity to objective functions, constraints, and parameter values. The "best" combination can then be used to refine management options, a procedure which greatly simplifies the subsequent simulation procedure.

Several authors have proposed the use of systems-analysis techniques in water-resources planning (Dorfman, 1965; Hufschmidt, 1965; Hufschmidt and Fiering, 1966; Fiering and Harrington, 1968; Hall and Dracup, 1970; Buras, 1972; deLucia et al., 1971; Biswas, 1975). From these publications one can conclude that water-resources planning is a multi-discipline and dynamic process. Several disciplines, such as economics, engineering, hydrology, social sciences, and others are directly involved in its different stages. It is also a dynamic process in that the results of a particular analysis must be checked and revised iteratively. Systems approach to problem solving using optimization, simulation and statistical analysis can make great contributions to the efficiency of this dynamic process.

Dorfman (1965) was one of the first scientists who realized the importance of systems analysis, particularly optimization techniques, in water-resources planning. Optimization techniques are used to arrive at a set of solutions best suited to the needs of the planners. These techniques become particularly important when the planners have to screen a large number of

alternative configurations of water-supply projects to determine the most efficient set of projects subject to physical and legal constraints of the system.

Mathematical-programming techniques such as linear programming, integer programming, dynamic programming, classical analysis using Lagrange multipliers, and quadratic programming (Hillier and Lieberman, 1968; Wagner, 1969) may be used to arrive at the solution of a systems-analysis approach to water-resources planning.

Optimization models by themselves may not be adequate to answer all the questions asked by the planners. Simulation models of different degrees of complexity may be used to further focus on the questions asked by the planners. For example, detailed analysis of the behavior of a system on a small time scale may only be analyzed using simulation techniques. Simulation techniques also have the ability to analyze system's behavior stochastically. By conducting many simulations of systems behavior, the planners can calculate the mean and variance of system's outputs such as the percent of time that a particular component fails to answer the target demand. Because the simulation models must be executed repeatedly, their cost is usually high. This cost may be justifiable if it is put in the context of the magnitude of the investments for a comprehensive water-resources planning process. The result of analysis from optimization models (screening models) can significantly reduce the number of alternatives which must be analyzed by simulation models.

C. Water-resources Planning Process

1. Input-Output of Water-resources Planning

Given a set of policies and goals governing the economic development of a region, water-resources planners must consider how best to distribute available resources among various sectors of the economy (e.g., agriculture, industry, power, etc.) to achieve the desired goals. Each development scheme imposes a set of spatial and temporal demands for various quantities and qualities of water depending upon its use. For each set of demands, the planners can design a

number of water-resources programs, consisting of structural and non-structural measures, which will transform available surface-water and ground-water resources into a set of desired outputs.

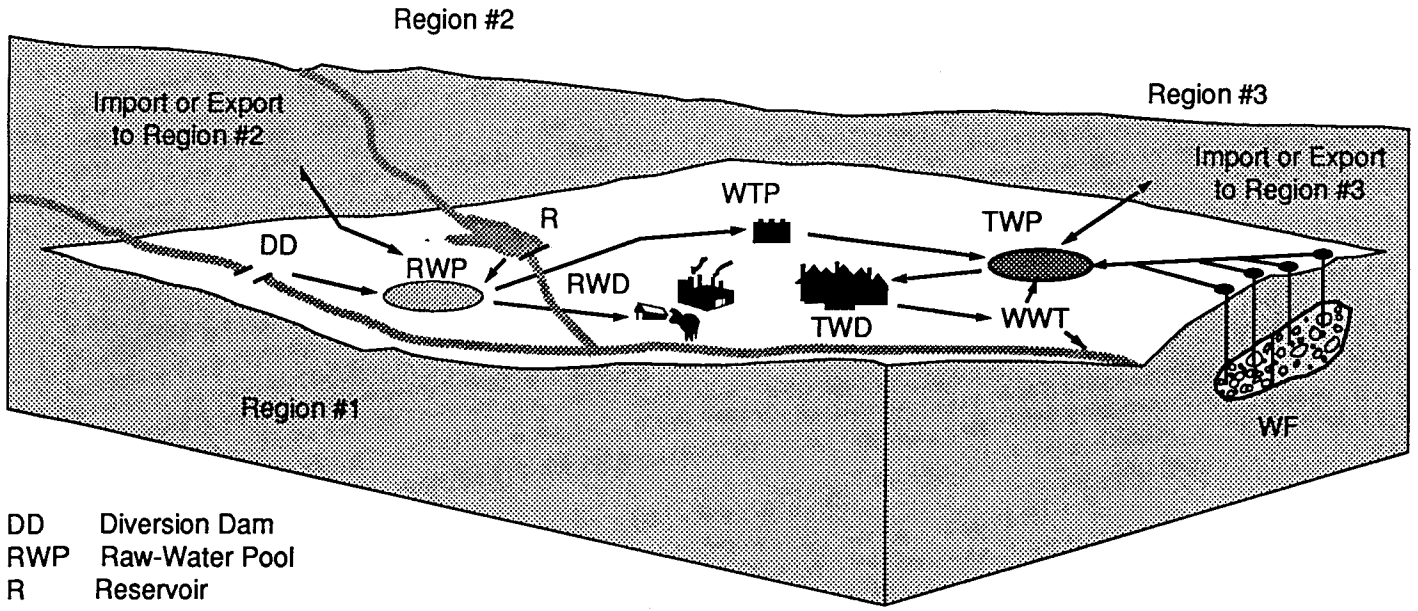
Perhaps the most important input into a water-resources planning effort is a sound policy based on scientific facts. A policy of development subject to physical, legal, and environmental constraints can be argued to be the basis of a long-term planning effort. Water-resources planners can never divorce their job from political issues. These issues in many cases supersede any scientifically sound water-resources planning policy, and water-resources planners have to be ready to take political considerations into account. To do this, their job is easier if their development policy is based on the analysis of a large number of alternatives using scientific procedures.

Other inputs into a comprehensive water-resources planning effort are physical and administrative structures and basic data to conduct the analysis. Physical structures such as reservoirs, diversion dams, well fields, desalinization plants, pipe lines, treatment plants, distribution facilities and administrative structures are all part of a comprehensive water-resources planning effort. In this report, the issues of administrative structures will not be addressed.

2. Data Requirements for Water-resources Planning

In the application of any planning procedure, several sets of data must be provided. These data provide input into the planning procedure. If a screening model is adopted for the preliminary stages of planning, these data must be quantified and supplied to the model in the formats required by the model.

Assume that the objective of a water-resources planning effort is to provide treated and raw waters for the treated-water demand center (TWD) and raw-water demand center (RWD) in Region 1 of Figure 1. These waters may be provided from reservoir (R), diversion dam (DD), ground-water well field (WF), or imported into Region 1 from Regions 2 or 3. For analysis purposes, it is assumed that Region 1 has a raw-water pool (RWP) and a treated-water pool (TWP) from which



DD Diversion Dam
RWP Raw-Water Pool
R Reservoir
RWD Raw-Water Demand
WTP Water-Treatment Plant
TWD Treated-Water Demand
TWP Treated-Water Pool
WWT Waste-Water Treatment
WF Well Field

Figure 1: Schematic representation of a multi-basin water-resources planning project

waters may be processed in water-treatment plants (WTP) and sent to TWP. The ground water may be of varying quality. Therefore, depending on the quality, it may be processed in WTP or sent to RWP. The planners searching for the most efficient way of supplying Region 1 as well as Regions 2 and 3 may decide to export raw and/or treated waters to Regions 2 and 3. Figure 2 shows a schematic representation of the system in Figure 1.

To conduct a water-resources planning study for such a system, the following data must be provided first.

a. Classification of Demands

Water demands in any area under consideration may be classified into quantity and quality demands:

i) Water-quantity demand classifications:

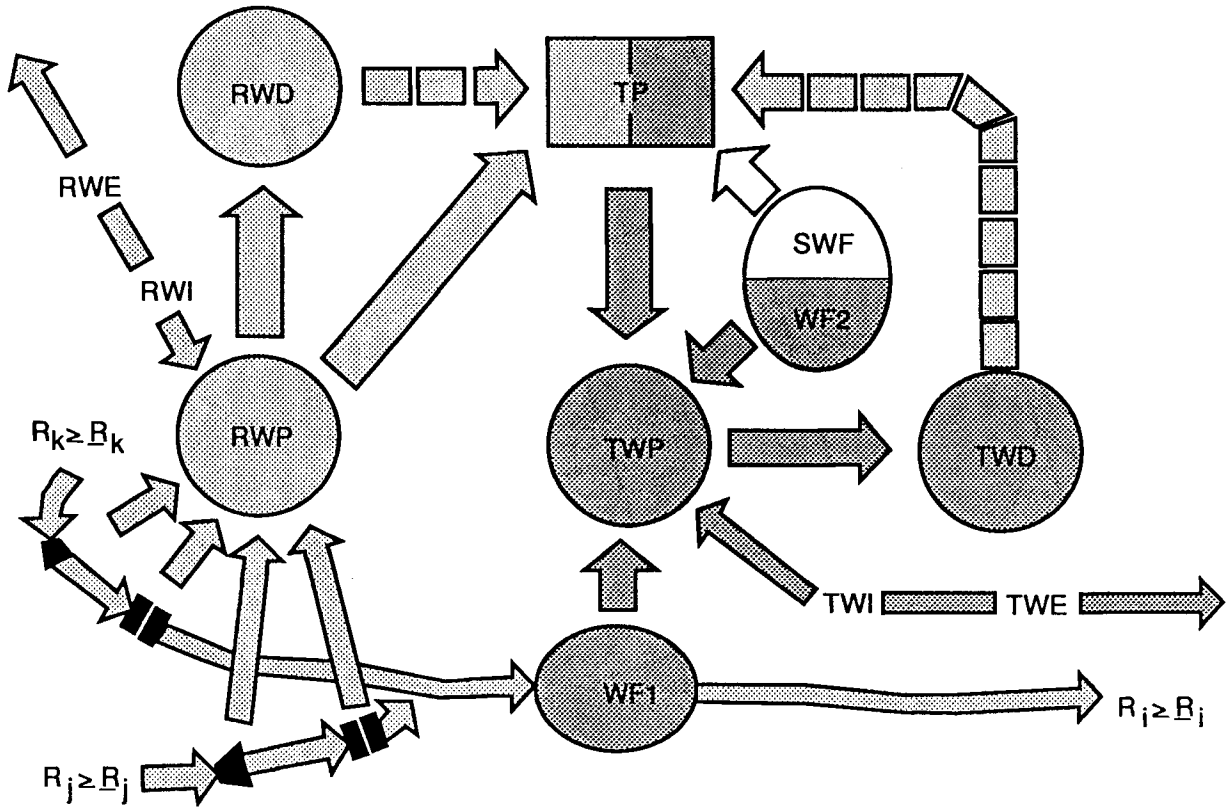
For planning purposes, the water demands may be classified as follows:

1. Residential (domestic)	Annual quantity (mgd)
2. Industrial	Annual quantity (mdg)
3. Agricultural (irrigation)	Annual quantity (mgd)
4. Cooling Water	Annual quantity (mgd)
5. Flood Control	Flood-flow bound
6. Environmental management	Low-flow requirements

Industrial water demands may be broken down by industry. Water used to process agricultural products may be assumed to be part of the industrial demand. Each demand may also be separated into private and public supplies.

ii) Water-quality Classifications:

The water quality reflects the various demands for water. For planning purposes, water quality may be classified as follows:



RWD: raw-water demand
RWP: raw-water pool
RWI: raw-water import
RWE: raw-water export
resrvoir

TWD: treated-water demand
TWP: treated-water pool
TWI: treated-water import
TWE: treated-water export
diversion dam

TP: treatment plant
WF: well field
R: streamflow
SWF: contaminated ground water

Figure 2: Schematic representation of system in region 1 of figure 1

1. Freshwater—suitable for residential or industrial use with minimum investment
2. Raw water—suitable for cooling water or irrigation without treatment. Use of residential or industrial purposes will require substantial treatment
3. Waste water—unsuitable for any use before treatment.

b. Identification of Planning Units

1. Basic planning units must be identified
2. Aggregation of basic units into planning units based on political and hydrologic boundaries

c. Identification of Demands

*i) **Existing Demands** (by basic planning unit):*

1. Domestic an residential demands
   a) municipal supply
   b) self supplied
2. Industrial demand
   a) municipal supply
   b) self supplied
3. Irrigation demands
   a) government supplies
   b) self supplied

*ii) **Future Demand Projections of Above Classes:***

1. Crude linear projection based on alternative features
2. Refined projections
3. Model projections

3. Identification of Existing Water-resources Structures

a. Reservoirs

1. Location—latitude-longitude, municipalities, river basins (show on map)
2. Type—water supply, drainage area, flood control, hydropower, head regulation, etc.
3. Service area—municipalities supplied, irrigated area, etc.

4a. Capacity—water supply, flood control, sediment storage, etc.

4b. Yield—percentage of dead storage already filled

5. Date constructed
6. Capital costs—break down into classes
7. Cost/capacity
8. Statistics of withdrawals—average, maximum, minimum, variance, etc.
9. OMR costs—breakdown into classes
10. Cost/unit of water delivered
11. Description of system
12. Owner and source of information

b. Aqueducts, canals, and tunnels

1. Location (as above)
2. Type—aqueduct, canal, tunnel, etc.
3. Service area (see above)
4. Capacity—optimal transmission rate
5. Date constructed
6. Capital costs—tunnel costs, pipeline costs, pumps, etc.
7. Cost/capacity
8. Statistics on water transfers
9. OMR costs

10. Cost/unit of water delivered
11. Description of system
12. Owner and source of information

c. Ground-water fields

1. Location of field (see above)—include aquifer name
2. Type—type of supply, quality of raw water, sources of water
3. Service area (see above)
4. Maximum capacity for raw-water delivery
5. Date constructed
6. Capital costs
7. Cost/capacity
8. Statistics of withdrawals (in flow)
9. OMR costs
10. Cost/unit of water delivered
11. Description of associated distribution system
12. Owner and source of information

4. <u>Identification of Proposed or Potential Water-resources Structures</u>

a. Reservoirs

1. Gross storage below spillway, sediment storage, dead storage, net storage
2. Flood discharge (100 yr.), spillway-design discharge
3. Design yield
4. Construction cost operation and maintenance costs (broken down into $/acre-ft. assuming whole reservoir is used; if OMR are annual costs, must consider length of time)

b. Aqueducts, canals and tunnels

c. Ground-water fields

d. **Water-treatment plants**

e. **Waste-treatment plants**

f. **Desalinization plants**

5. Water Supply

a. **Surface water** (by planning units)

1. Classify available records
2. Compute storage yield curves for different probabilities of failure
3. Calculate quantity of water available for development
4. Estimate quality
5. Identify data "deficient" areas

b. **Ground water** (by planning units)

1. Classify available records and studies
2. Estimate yields for each aquifer
3. Calculate quantity of water available for development
4. Estimate quality and identify possible pollution sources
5. Identify data "deficient" areas

The list of data requirements for water-resources planning given above is by no means considered exhaustive. There are data elements which become necessary for particular projects. The need for these data must be the decision of experienced water planners.

III. Water-resources Planning—Methodology

A. Assumptions

A screening model has been adopted for the initial stages of water-resources planning efforts for a system with several components. The model uses a mixed-integer program that provides the planners the ability to screen large numbers of water-resources projects proposed for possible construction and implementation. Given a scenario of future water demands, the model

selects and schedules for construction of the water-supply projects that will meet projected demands for water at minimum present-value of costs. The planning model is formulated using the following assumptions.

•1. The system is subdivided into I planning regions. The regions are distinct, cover the system and contain one or more river basins (see Figure 3 as an example of basins in Kansas, USA).

•2. The planning period of interest (e.g., 20 years) is subdivided into Ne time periods of equal length.

•3. There are two types of water demands: untreated water used for agriculture and treated water used for domestic and industrial use. The water demands are known and specified by the planners.

•4. There are three direct sources of water (treated and untreated). They are 1) in-stream diversions, 2) reservoirs and 3) groundwater well fields. In-stream diversions and reservoirs may provide only untreated water. Ground-water, provided there has been no contamination, may be used directly as treated water. In-stream diversion and reservoir waters must undergo treatment for use as domestic or industrial supplies.

•5. Water may be transported between regions either in treated or untreated form.

•6. Within the region i (i=1, ... ,I) there are $K_s(i)$ in-stream diversion projects, $K_R(i)$ reservoir projects, $K_G(i)$ ground-water projects, $K_T(i)$ water treatment projects, $K_{TI}(i,j)$ treated-water projects importing water from region j and $K_{UI}(i,j)$ untreated water projects importing water from region j. The K values will be zero if no such project types exist in a region or between regions.

•7. The various projects have been sited and their yields and capital costs determined by preliminary engineering studies.

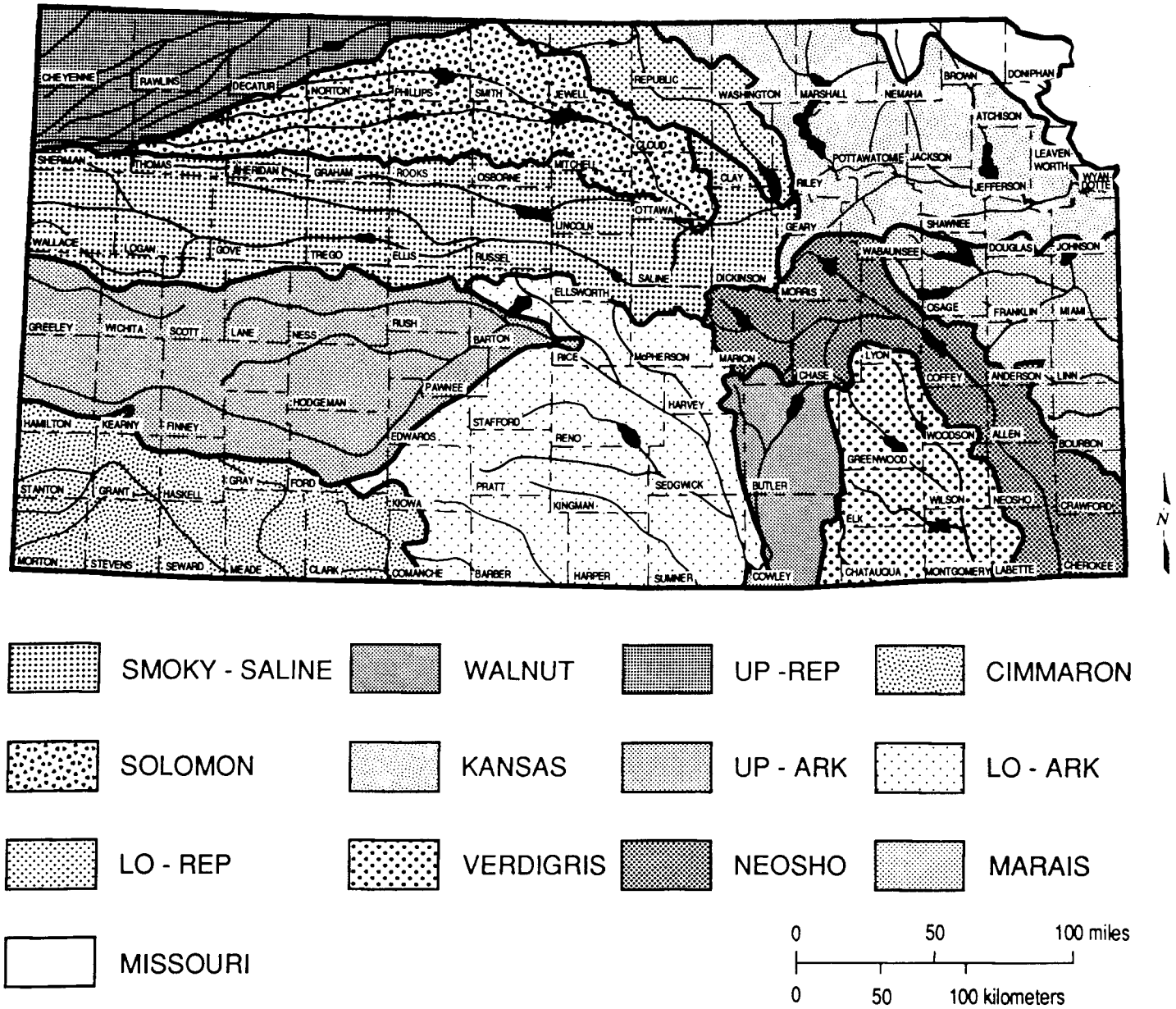


Figure 3: Major hydrologic basins of Kansas

•8. The project's operation, maintenance and replacement (OMR) costs are approximately linear with water produced, processed or transported during a time period (e.g., millions of gallons per year).

•9. Least-cost, summed over the planning period and discounted to present value, is a satisfactory criterion for evaluating alternative project configuration.

•10. The untreated- and treated-water demands are aggregated to demand points (pools) within a region. In general, there is only one demand point or water demand center for each type water, in each region.

B. Decision Variables

The unknowns in the model consist of two types of decision variables, continuous and zero-one integer. The continuous variables are

1. Q_S (n,i,k): quantity of water removed by the k-th in-stream diversion project in the n-th time period, in region i.
2. Q_R (n,i,k): quantity of water removed from the k-th reservoir project in the n-th time period, in region i.
3. Q_G (n,i,k): quantity of water removed from the k-th ground-water well-field project in the n-th time period, in region i.
4. Q_T (n,i,k): quantity of water treated in the k-th water-treatment project in the n-th time period, in region i.
5. $Q_{TI}(n,i,j,k)$: quantity of treated water imported to region i <u>from</u> region j by the k-th import project (j to i) in the n-th time period.
6. $Q_{UI}(n,i,j,k)$: quantity of untreated water imported to region i <u>from</u> region j by the k-th import project (j to i) in the n-th time period.

The zero-one integer variables are:

1. I_S (n,i,k): indicates whether or not the k-th in-stream diversion project is built in the n-th time period, in region i.

2. I_R (n,i,k): indicates whether or not the k-th reservoir project is built in the n-th time period, in region i.

3. I_G (n,i,k): indicates whether or not the k-th well field is built in the n-th time period, in region i.

4. I_T (n,i,k): indicates whether or not the k-th water treatment project is built in the n-th time period, in region i.

5. I_{TI} (n,i,j,k): indicates whether or not the k-th treated water import project (j to i) is built in the n-th time period.

6. I_{UI} (n,i,j,k): indicates whether or not the k-th untreated water import project (j to i) is built in the n-th time period.

The zero-one integer variables have the value one if the project is built, zero otherwise.

C. Cost Coefficients

The Operation Maintenance and Replacement (OMR) cost per unit water produced, transported or processed during a planning period are reduced to present value using a suitable interest rate. If the OMR cost per unit of water is given, by C_o', then;

$$C_o(n) = \frac{C_o'}{(1 + r)^n} \qquad (1)$$

is the present value of C_o' in the n-th time period given interest rate r. The OMR cost coefficients for the model are

C_{oS} (n,i,k) = present value of the OMR cost of producing a unit quantity of water by k-th in-stream diversion project , in region i, in the n-th time period.

C_{oR} (n,i,k) = present value of the OMR cost of producing a unit quantity of water from k-th reservoir project, in region i, in the n-th time period.

$C_{oG}(n,i,k)$ = present value of the OMR cost of producing a unit quantity of water from k-th well-field project, in region i, in the n-th time period.

$C_{oT}(n,i,k)$ = present value of the OMR cost of processing a unit quantity of water at water-treatment project k, in region i, in the n-th time period.

$C_{oTI}(n,i,j,k)$ = present value of the OMR cost of transporting a unit quantity of treated water from (j to i) k-th import project in the n-th time period.

$C_{oUI}(n,i,j,k)$ = present value of the OMR cost of transporting a unit quantity of untreated water from (j to i) k-th import project in the n-th time period.

The fixed cost (construction, land, engineering, administrative and legal) can be calculated in two ways. Either the entire fixed cost can be paid in the time period of construction or an annual fixed cost can be determined using an amortization rate.

If the entire fixed cost is assessed during the time period of construction as C_f' dollars then the fixed cost coefficient is given by

$$C_f(n) = \frac{C_f'}{(1+r)^n} \tag{2}$$

where r is the present value interest rate and n is the time period.

If an annual fixed cost is to be used, four factors are involved: 1) the total fixed cost of the project C_f', 2) N_p the economic life of the project for purposes of computing the debt service, 3) the amortization rate, r_a, and 4) the present value interest rate, r. The fixed cost coefficient is then calculated in two steps. In step 1 a non-discounted annual fixed cost, AFC, is determined:

$$AFC = \left[\frac{r_a(1+r_a)^{N_p}}{(1+r_a)^{N_p}-1}\right] C_f' \tag{3}$$

In step 2 the AFC is discounted to present value, using interest rate r, beginning at year n, the first year at which the project is built and ending at either the design period N_e or the project economic life period N_p, which ever is the less.

$$C_f(n) = \sum_{\tau=n}^{N_e \text{ or } N_p} \left[\frac{1}{(1+r)^{\tau}}\right] AFC \qquad (4)$$

The choice of an amortization rate, r_a reflects the cost to the state government of borrowing money to finance projects, while the present value interest rate, r, reflects society's feelings about providing for the future as opposed to present or current consumption. It should be noted that increasing the amortization rate, r_a, will increase the AFC, increasing the economic life of the project N_p, thus decreasing the AFC. Increasing the present value interest rate, r, will decrease the present value of both fixed costs and OMR costs.

Regardless of the method of calculation, the fixed-costs coefficients for the model are

$C_{fS}(n,i,k)$ = present value of fixed cost of k-th in-stream diversion project in region i, initiated in the n-th time period.

$C_{fR}(n,i,k)$ = present value of fixed cost of k-th reservoir project in region i, in the n-th time period.

$C_{fG}(n,i,k,i)$ = present value of fixed cost of k-th well-field project in region i, initiated in the n-th time period.

$C_{fIT}(n,i,k)$ = present value of fixed cost of k-th water-treatment project in region i, initiated in the n-th time period.

$C_{fTI}(n,i,j,k)$ = present value of fixed cost of k-th treated-water import project from other (j to i) regions initiated in the n-th time period.

$C_{fUI}(n,i,j,k)$ = present value of fixed cost of k-th untreated-water import project from other (j to i) regions initiated in the n-th time period.

D. Model Formulation

1. Objective Function

The objective function of the model is to minimize the present cost of meeting future water demands over the design period. This objective in mathematical form is

$$
\begin{aligned}
\text{Min} \sum_{n=1}^{N_e} \sum_{i=1}^{I} \Bigg\{ & \sum_{k=1}^{K_S(i)} \left[C_{oS}(n,i,k) Q_S(n,i,k) + C_{fS}(n,i,k) I_S(n,k,k) \right] \\
& \qquad \text{(In-stream diversion costs)} \\
& + \sum_{k=1}^{K_R(i)} \left[C_{oR}(n,i,k) Q_R(n,i,k) + C_{fR}(n,i,k) I_R(n,i,k) \right] \\
& \qquad \text{(Reservoir costs)} \\
& + \sum_{k=1}^{K_G(i)} \left[C_{oG}(n,i,k) Q_G(n,i,k) + C_{fG}(n,i,k) I_G(n,i,k) \right] \\
& \qquad \text{(Well-field costs)} \\
& + \sum_{k=1}^{K_T(i)} \left[C_{oT}(n,i,k) Q_T(n,i,k) + C_{fT}(n,i,k) I_T(n,i,k) \right] \\
& \qquad \text{(Water-treatment costs)} \\
& + \sum_{\substack{j=1 \\ i \neq j}}^{I} \sum_{k=1}^{K_{TI}(i,j)} \left[C_{oTI}(n,i,j,k) Q_{TI}(n,i,j,k) + C_{fTI}(n,i,j,k) I_{TI}(n,i,j,k) \right] \\
& \qquad \text{(Treated-water import costs)} \\
& + \sum_{\substack{j=1 \\ i \neq j}}^{I} \sum_{k=1}^{K_{UI}(i,j)} \left[C_{oUI}(n,i,j,k) Q_{UI}(n,i,j,k) + C_{fUI}(n,i,j,k) I_{UI}(n,i,j,k) \right] \\
& \qquad \text{(Untreated-water import costs)} \Bigg\}
\end{aligned}
\tag{5}
$$

The terms in the objective function (5) represent the operation and fixed costs of in-stream diversion, reservoir and well-field water production, treatment-plant water processing and treated- and untreated-water transportation. The objective function is subjected to the following constraints

2. **Project-initiation constraints**—no water may be produced, treated or transported until the project has been constructed;

a) In-stream diversion:

$$V_S(i,k)\sum_{n'=1}^{n} I_S(n',i,k) \geq Q_S(n,i,k);$$

$$n = 1,\ldots,N_e;$$
$$i = 1,\ldots,I;\ k = 1,\ldots,K_S(i) \qquad (6)$$

where $V_S(i,k)$ is the yield or capacity of the k-th in-stream diversion project in region i.

b) Reservoirs:

$$V_R(i,k)\sum_{n'=1}^{n} I_R(n',i,k) \geq Q_R(n,i,k);$$

$$n = 1,\ldots,N_e;$$
$$i = 1,\ldots,I;$$
$$k = 1,\ldots,K_R(i) \qquad (7)$$

where $V_R(i,k)$ is the yield or capacity of the k-th reservoir project in region i.

c) Well fields:

$$V_G(i,k)\sum_{n'=1}^{n} I_G(n',i,k) \geq Q_G(n,i,k);$$

$$n = 1,\ldots,N_e;$$
$$i = 1,\ldots,I;$$
$$k = 1,\ldots,K_G(i) \qquad (8)$$

where $V_G(i,k)$ is the yield of the k-th well-field project in region i.

d) Water treatment plants:

$$V_T(i,k)\sum_{n'=1}^{n} I_T(n',i,k) \geq Q_T(n,i,k);$$

$$n = 1,\ldots,N_e$$
$$i = 1,\ldots,I;$$
$$k = 1,\ldots,K_W(i) \qquad (9)$$

where $V_T(i,k)$ is the capacity of the k-th water-treatment project in region i.

e) Treated-water imports:

$$V_{TI}(i,j,k)\sum_{n'=1}^{n} I_{TI}(n',i,j,k) \geq Q_{TI}(n,i,j,k)$$
$$n = 1,\ldots,N_e;\ i = 1,\ldots,I;\ j = 1,\ldots,I \text{ but } i \neq j$$
$$k = 1,\ldots,K_{TI}(i,j) \qquad (10)$$

where $V_{TI}(i,j,k)$ is the capacity of the k-th treated-water transport system supplying water from region j to region i.

f) Untreated-water imports:

$$V_{UI}(i,j,k)\sum_{n'=1}^{n} I_{UI}(n',i,j,k) \geq Q_{UI}(n,i,j,k)$$
$$n = 1,\ldots,N_e;\ i = 1,\ldots,I;\ j = 1,\ldots,I \text{ but } i \neq j$$
$$k = 1,\ldots,K_{UI}(i,j) \qquad (11)$$

where $V_{UI}(i,j,k)$ is the capacity of the k-th untreated-water transport system supplying water from region j to region i.

3. **Project-limitation constraints**—a water production treatment, or transport project may be constructed once and only once during the design period, Ne.

a) In-stream diversion:

$$\sum_{n=1}^{N_e} I_S(n,i,k) \leq 1 \qquad i = 1,\ldots,I;\ k = 1,\ldots,K_S(i) \qquad (12)$$

b) Reservoirs:

$$\sum_{n=1}^{N_e} I_R(n,i,k) \leq 1 \qquad i = 1,\ldots,I;\ k = 1,\ldots,K_R(i) \qquad (13)$$

c) Ground water:

$$\sum_{n=1}^{N_e} I_G(n,i,k) \leq 1 \qquad i = 1,\ldots,I;\ k = 1,\ldots,K_G(i) \qquad (14)$$

d) Water treatment:

$$\sum_{n=1}^{N_e} I_T(n, i, k) \le 1 \qquad i = 1,\ldots,I;\ k = 1,\ldots,K_T(i) \tag{15}$$

e) Treated-water import:

$$\sum_{n=1}^{N_e} I_{TI}(n,i,j,k) \le 1 \qquad i = 1,\ldots,I; \qquad j = 1,\ldots,I \text{ but } i \ne j$$
$$k = 1,\ldots,K_{TI}(i,j) \tag{16}$$

f) Untreated-water import:

$$\sum_{n=1}^{N_e} I_{UI}(n, i, j,k) \le 1 \qquad i = 1,\ldots,I; \qquad j = 1,\ldots,I \text{ but } i \ne j$$
$$k = 1,\ldots,K_{UI}(i,j) \tag{17}$$

4. **Treated-water production constraint** — The quantity of treated water produced by water-treatment plant and uncontaminated well fields; and the quantity of treated water imported minus the quantity exported must be equal to or exceed the treated water demands in a planning region during a time period:

$$\sum_{k=1}^{K_G(i)} Q_G(n, i,k) + \sum_{k=1}^{K_T(i)} Q_T(n, i,k)$$
$$+ \sum_{\substack{j=1 \\ i \ne j}}^{I} \left[\sum_{k=1}^{K_{TI}(i,j)} Q_{TI}(n, i, j, k) - \sum_{k=1}^{K_{TI}(i,j)} Q_{TI}(n, j, i, k) \right] \ge T_D(n, i)$$
$$n = 1,\ldots,N_e; i = 1,\ldots,I, j = 1,\ldots,I \tag{18}$$

where $T_D(n,i)$ is the treated-water demand in the n-th time period for region i.

5. **Total water-demand constraint** — The total quantity of water produced (the output of in-stream diversion, reservoirs and well fields) and the imports minus exports of treated and

untreated water must be equal to or exceed the total water demands in a planning region during a time period:

$$\sum_{k=1}^{K_S(i)} Q_S(n,i,k) + \sum_{k=1}^{K_R(i)} Q_R(n,i,k) + \sum_{k=1}^{K_G(i)} Q_G(n,i,k)$$
$$\sum_{\substack{i=1 \\ i\neq j}}^{I} \left[\sum_{k=1}^{K_{UI}(j,i)} Q_{UI}(n,j,i,k) - \sum_{k=1}^{K_{UI}(j,i)} Q_{UI}(n,j,i,k) \right] \geq T_D(n,i) + R_D(n,i)$$
$$n = 1,\ldots,N_e;\ i = 1,\ldots,I \qquad (19)$$

where $R_D(n,i)$ is the untreated- or raw-water demand in region i during the n-th time period.

6. <u>Streamflow-requirement constraints</u> — There may be associated with each well field a stream segment with ground-water/surface-water interactions occurring. During a time period the discharge in the stream segment minus withdrawals from the stream system by diversions and reservoirs, and by leakage to the ground-water system, must be greater than a required streamflow discharge. For the p-th such segment

$$Q_A(n,i,p) - \sum_{k=1}^{K_S(i:p)} Q_S(n,i,k) - \sum_{k=1}^{K_R(i:p)} Q_R(n,i,k) -$$
$$\sum_{k'=1}^{K_G(n,i:p)} \sum_{t=1}^{n} p(i,k',n-t+1)\, Q_G(n,i,k') \geq R_q(n,i,p) \qquad (20)$$

where $(i:p)$ signifies the p-th stream segment portion of the i-th region; $K_S(i:p)$, the number of in-stream diversion projects on segment p; $K_R(i:p)$, the number of reservoir projects on segment p; $Q_A(n,i:p)$, the aggregated streamflow during time period n for the i-th region's p-th segment; $p(i,k,n-t+1)$ is the fraction of water diverted from the surface-water system to the ground-water system (Appendix A) in the p-th stream segment of region i during step n due to pumping of well field $Q_G(n,i,k')$; and $R_q(n,i:p)$ is the streamflow requirement for the n-th time period for stream

segment p of the i-th region. Maddock (1974) and Heidari et al. (1988) discuss the theoretical background for calculation of p in equation (20).

7. Bound on integer-variable constraints — All integer variables are bounded between zero and one. Thus

$$\begin{aligned}
&O < I_S(n,i,k) < 1 && \text{for all } n,i \text{ and } k \\
&O < I_R(n,i,k) < 1 && \text{for all } n,i \text{ and } k \\
&O < I_G(n,i,k) < 1 && \text{for all } n,i \text{ and } k \\
&O < I_W(n,i,k) < 1 && \text{for all } n,i \text{ and } k \\
&O < I_{TI}(n,i,k) < 1 && \text{for all } n,j,i \text{ and } k \\
&O < I_{UI}(n,i,k) < 1 && \text{for all } n,j,i \text{ and } k
\end{aligned} \qquad (21)$$

8. Non-negativity constraint — all continuous-decision variables are greater than or equal to zero.

The planning model described by equations (1) - (21) with the non-negativity conditions for continuous variables may be used for preliminary analysis of a variety of water-supply planning problems simply by changing the size or number of planning regions, the length of the planning period, the number of time periods in the planning horizon and the level of aggregation of water demands and water-supply projects.

IV. Example Problem

A. Description of the System

The state of Kansas consists of 13 major basins (see Figure 3). The system in Figure 4 is a representation of a water-resources planning project located in Lower Arkansas (LO-ARK) and Smoky Hill–Saline (SMOKY-SALINE) basins in Kansas.

The objective of the planning effort is to provide treated water to the demand center D in Figure 4 for the next 20 years at a minimum cost (Capital cost + OMR). The existing components of the system and their costs and yields are listed in Table 1. In the following input data, some of the data are assumed for demonstrative purposes. Therefore, it is not recommended that the results of this analysis be implemented without further investigation.

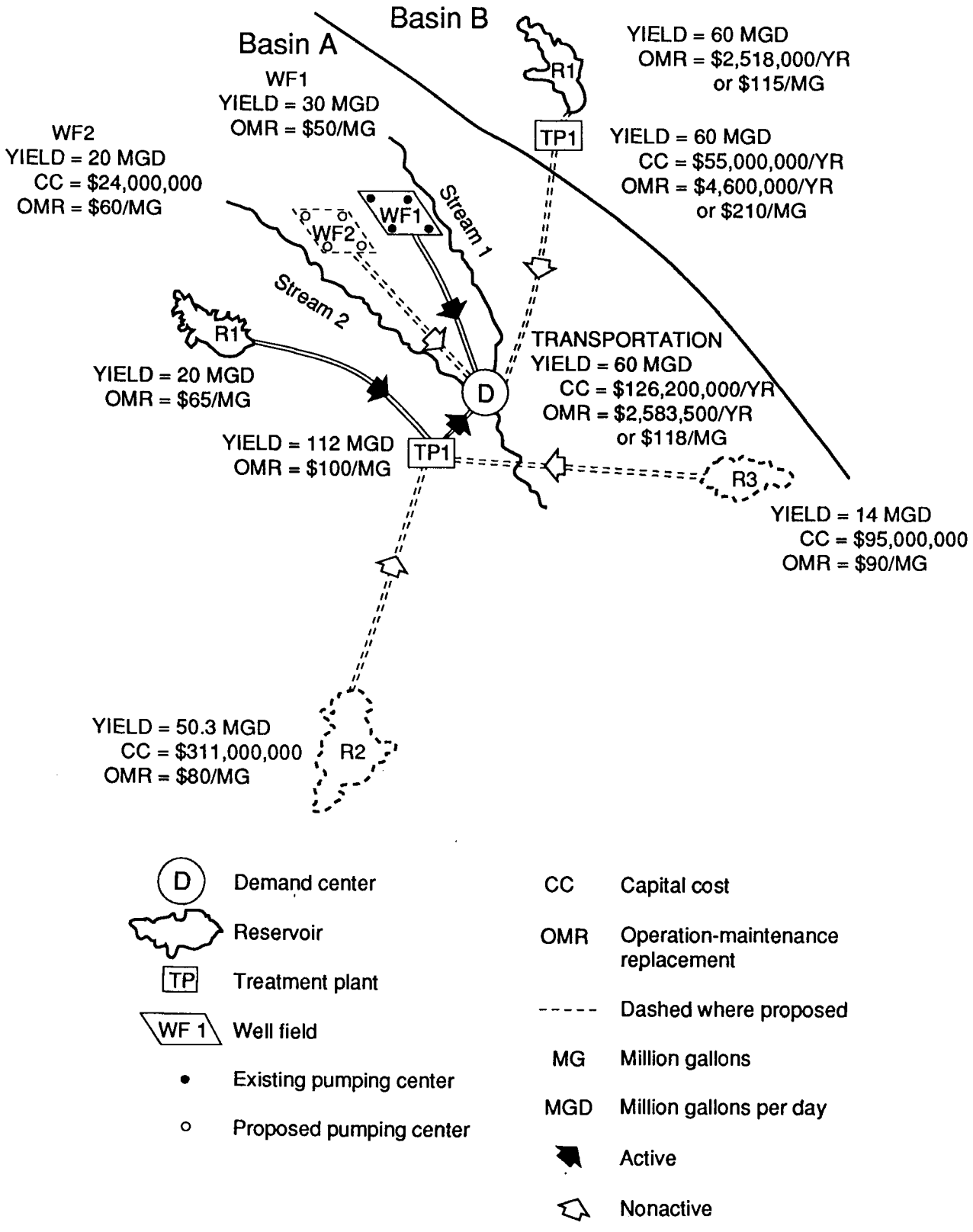


Figure 4: Schematic representation of a water-resources planning project

Table 1. Components of the Existing System and Their Costs & Yields in Basin A

Project	Yield (MGD)	OMR $/MG$_2$	Comments
Reservoir R1	20	65	
Well field WF1	30	50	4 pumping centers assumed
Treatment plant TP1	112	100	

A linear fit to the projected demand for D is presented in Figure 5. Because the demand is expected to exceed supply by 1985, several new components are proposed as additions to the system. These components and their characteristics are listed in Table 2 and schematically represented in Figure 4.

It must be mentioned that well fields WF1 and WF2 are located in the vicinity of streams 1 and 2 where minimum streamflow requirements are set for each stream by the planners. The mean streamflows are 181.7 and 384.0 cfs for streams 1 and 2 respectively. The assumed required minimum downstream flows are 12.9 and 61.4 cfs for streams 1 and 2.

Table 2. Proposed Components for the System and Their Costs and Yields

Basin	Component	Yield (MGD)	Capital Cost ($10^6)	OMR $/MG	Comments
A	Reservoir R2	50.3	311	80.00	
A	Reservoir R3	14.0	95	90.00	
A	Well Field WF2	20.0	24	60.00	4 pumping centers assumed
B	Reservoir R1	60.0	exists	115.00	
B	Treatment Plant TP1	60.0	55	210.00	
B to A	Pipeline	60.0	162	118.00	

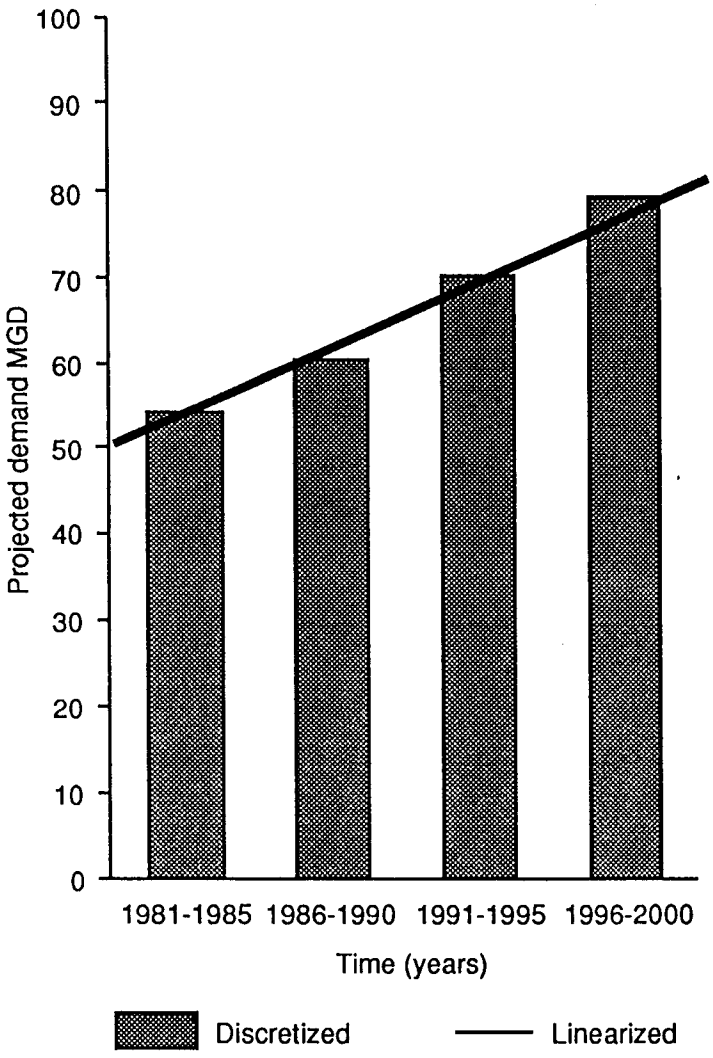


Figure 5: Projected demand (MGD) vs. time (years) for treated water

For the preliminary analysis, it was decided to conduct a four period analysis with each period being equal to 5 years. As was stated previously, water planning is a dynamic process. This means any information obtained from such a process for future decisions must be checked and upgraded periodically. Therefore, it was assumed that a planning period of 5 years will give the planner enough time to go over the results and update the cost coefficients and demand function for possible upgrading of the results.

B. Input-data Preparation and Program Executions

Stage I: In stage I, stream-aquifer interaction coefficients (P,s in Equation 20) are generated. This is done using a ground-water model (Maddock, 1975) specifically designed to generate these coefficients. This program, called PROG_BETA, requires data as specified in Appendices B and C-I. For the Example Problem, the input data for this stage are listed in Appendix D-I. These input data are called BETA-INDATA.

Once BETA-INDATA is prepared, program PROG_BETA (Appendix H) is executed to calculate stream-aquifer interaction coefficients. These coefficients are stored on file BETA_VALUES. Other output data from the PROG_BETA are stored on a file such as BETA_OUTDATA. In Appendices D-II and D-III, BETA_VALUES and BETA_OUTDATA are listed.

Stage II: In this stage, input data for the planning model, WATER_PLAN_MOD (Appendix J), are being prepared. Two methods are listed for preparation of these data (Appendix B). In method A data are prepared using PROG_PRE_PLAN (Appendix I). This program, which is written for Data General MV20000 and Dasher 210/211 terminals, may not be operational on other systems. It prepares input data in an interactive mode. In Appendix E, screens used by PROG_PRE_PLAN are listed. The user can generate an input-data file for WATER_PLAN_MOD such as the one listed in Appendix F-I by answering the questions in screens of Appendix E. If Dasher 210/211 terminals are not available, method B may be used to generate the input-data file.

The format of the input-data file to WATER_PLAN_MOD is given in Appendix C-II. In Appendix F-I the input data to this program for the Example Problem in Figure 4 are listed.

Stage III: Once the input data to the program WATER_PLAN_MOD (Appendix J) are prepared, this program can be executed to generate a matrix in the MPS format for the linear-integer mathematical programming package, PROG-LINDO.

If the coefficients in the generated matrix are too large, then the scaling option of WATER_PLAN_MOD must be used. When the scaling option of the WATER_PLAN_MOD is chosen, the program will ask for three scaling factors. These are:

CSF = column-scaling factor

RSF = row-scaling factor

ORSF = objective function—scaling factor

The output from this stage is an MPS file called PLAN_MOD_SCALED_MATRIX which is listed in Appendix F-II.

If we decide to bypass scaling of the matrix, then the output from stage III is called PLAN_MOD_OUT_MATRIX which is an MPS file ready for use by PROG_LINDO.

Stage IV: In this stage, program PROG-LINDO reads file PLAN_MOD_SCALED_ MATRIX or PLAN_MOD_OUT_MATRIX depending on whether or not the scaling procedure is chosen and solves it for its optimum solution. Note that scaling is a very important factor in obtaining a solution. Rows and columns must be divided by proper scaling factors, CSF, RSF and ORSF to obtain coefficients which are acceptable for solution by a linear-integer programming package such as PROG-LINDO. A rule of thumb is that there should be no nonzero coefficient whose absolute value is greater than 10^{5} or less than 10^{-4} (Schrage, 1984). For Example Problem CSF = 10,000, RSF = 10,000 and ORSF = 1,000,000 are appropriate. The user may wish to use different scaling factors for other input data.

Program PROG_LINDO is a linear-integer programming package (Schrage, 1984) under license agreement to the Kansas Geological Survey. It uses a branch and bound algorithm to find the solution to the input matrix. In Appendix G a sample output of this program is presented.

C. Analysis of Results

In Appendix G, section 5, the optimum solution (objective function and variables) is listed. The objective function, i.e. the present worth of all capital costs plus operation-maintenance and replacement (OMR) is $\$62.983 \times 10^6$. Note that because the coefficients of the objective function were scaled down by ORSF = 10^6, the objective function value in Table 5 must be multiplied by 10^6 to obtain the unscaled objective value. The zero-one integer values are self explanatory. For example, DTIABt01, t = 1,..., 4, are all equal to zero, indicating that the objective value does not require treated-water import from basin B to A using pipeline number 01 during time periods 1 through 4.

The continuous-variable section starting with QRSA101 and ending with QTPB401 can take on values from zero to the upper bound values listed in section 4 of Appendix G. The unit of these values in section 5 is (MG/Planning Period) x CSF. For example,

$$\text{QRSA101} = 3.474871 \times 10^4 \text{ MG/5 yrs}$$

$$= 19.04 \text{ MGD}$$

where CSF = 10^4.

In Figures 6 through 9, the decision variables for each new and existing component are listed. During period 1, R1, WF1 and WF2 are used to satisfy a demand of 53.5 MGD. WF2 is being exploited from one of the pumping centers for 5.0 MGD at a cost of $24,000,000. This clearly requires the planners to re-evaluate the demand and supply functions and make sure that there is absolutely no other way to either reduce the demand or increase the yields of the two existing components, i.e. R1 and WF1 in Figure 4.

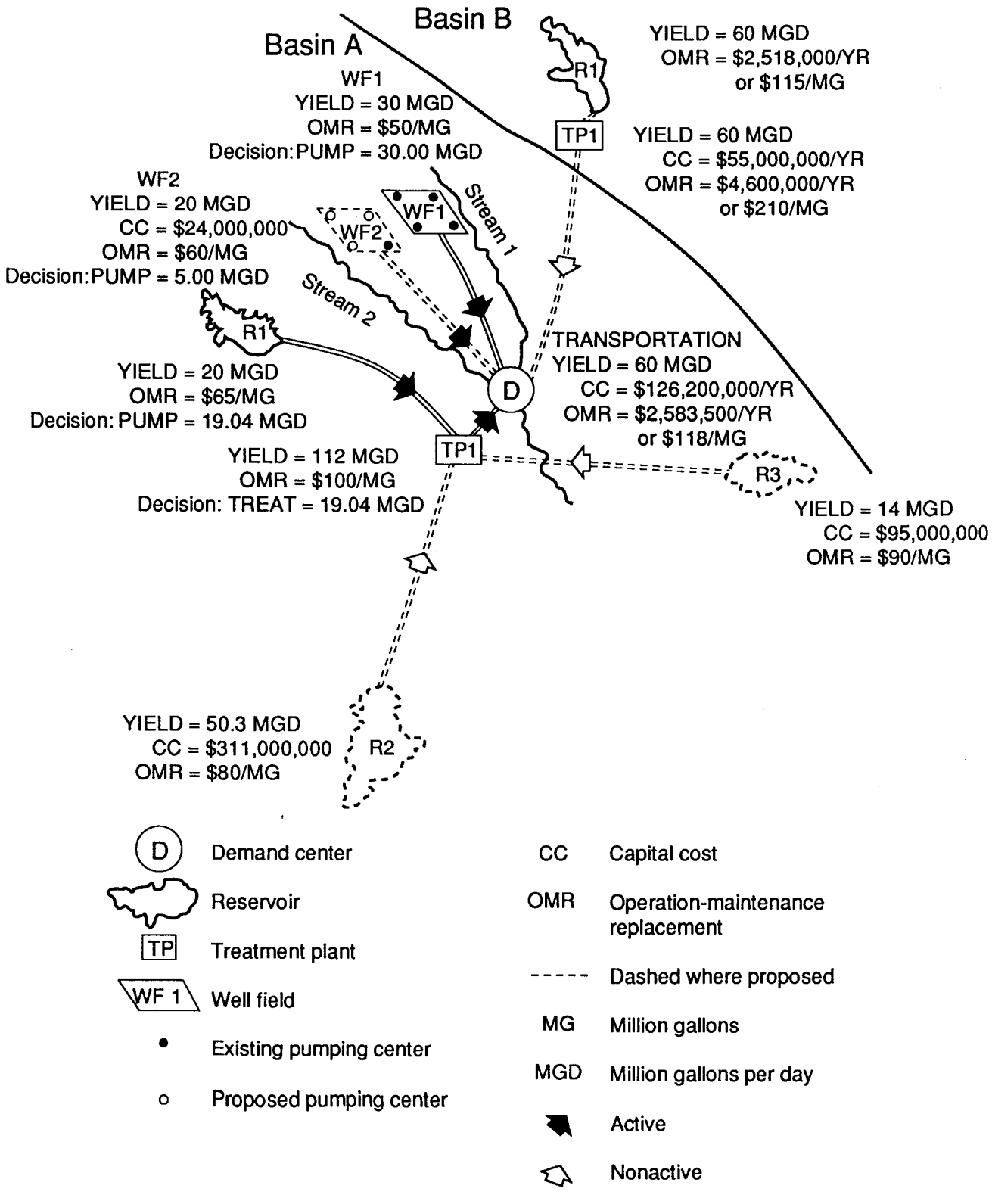


Figure 6: Results of Planning Period 1 freshwater demand = 53.5 MGD

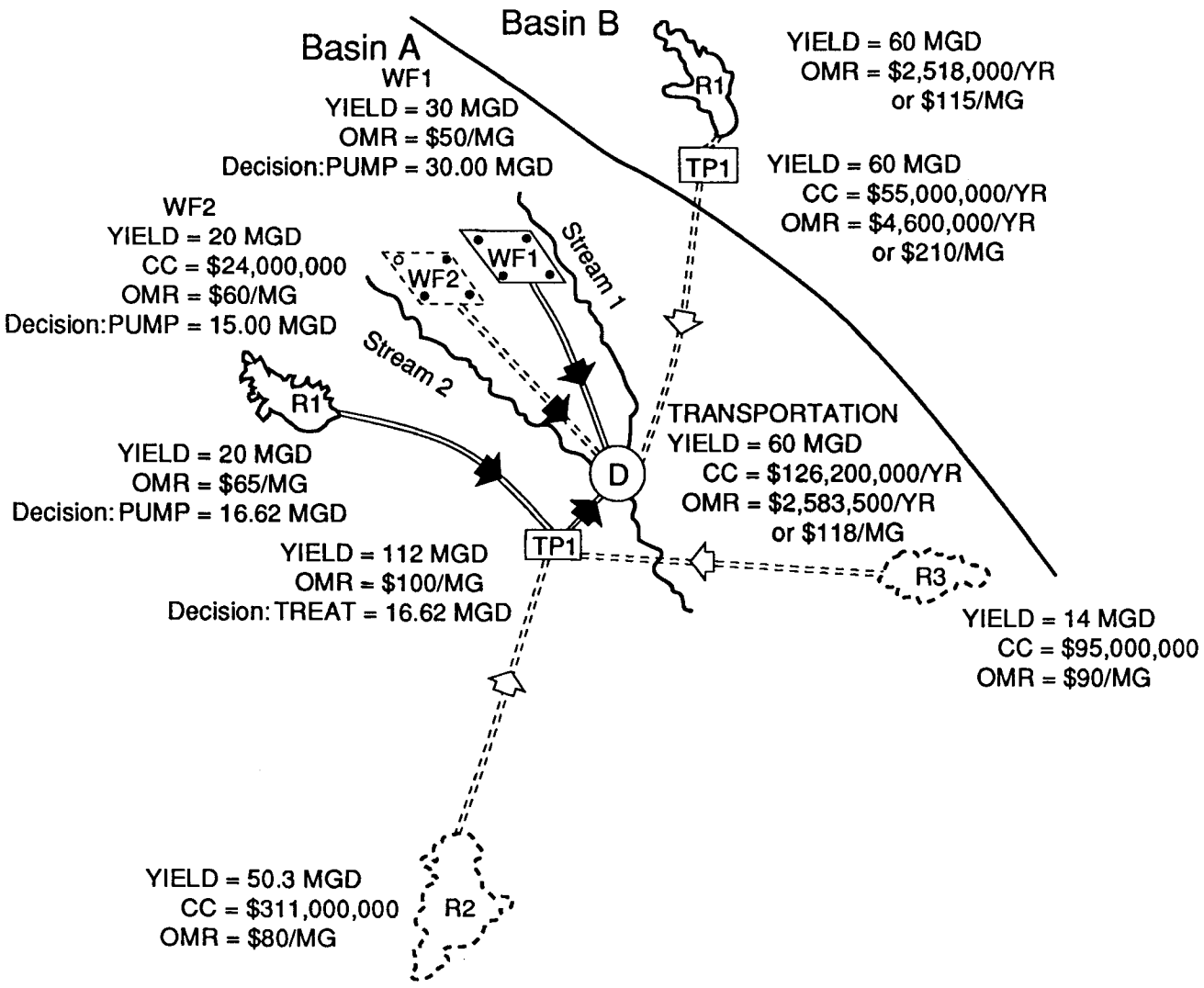


Figure 7: Results of Planning Period 2 freshwater demand = 61.0 MGD

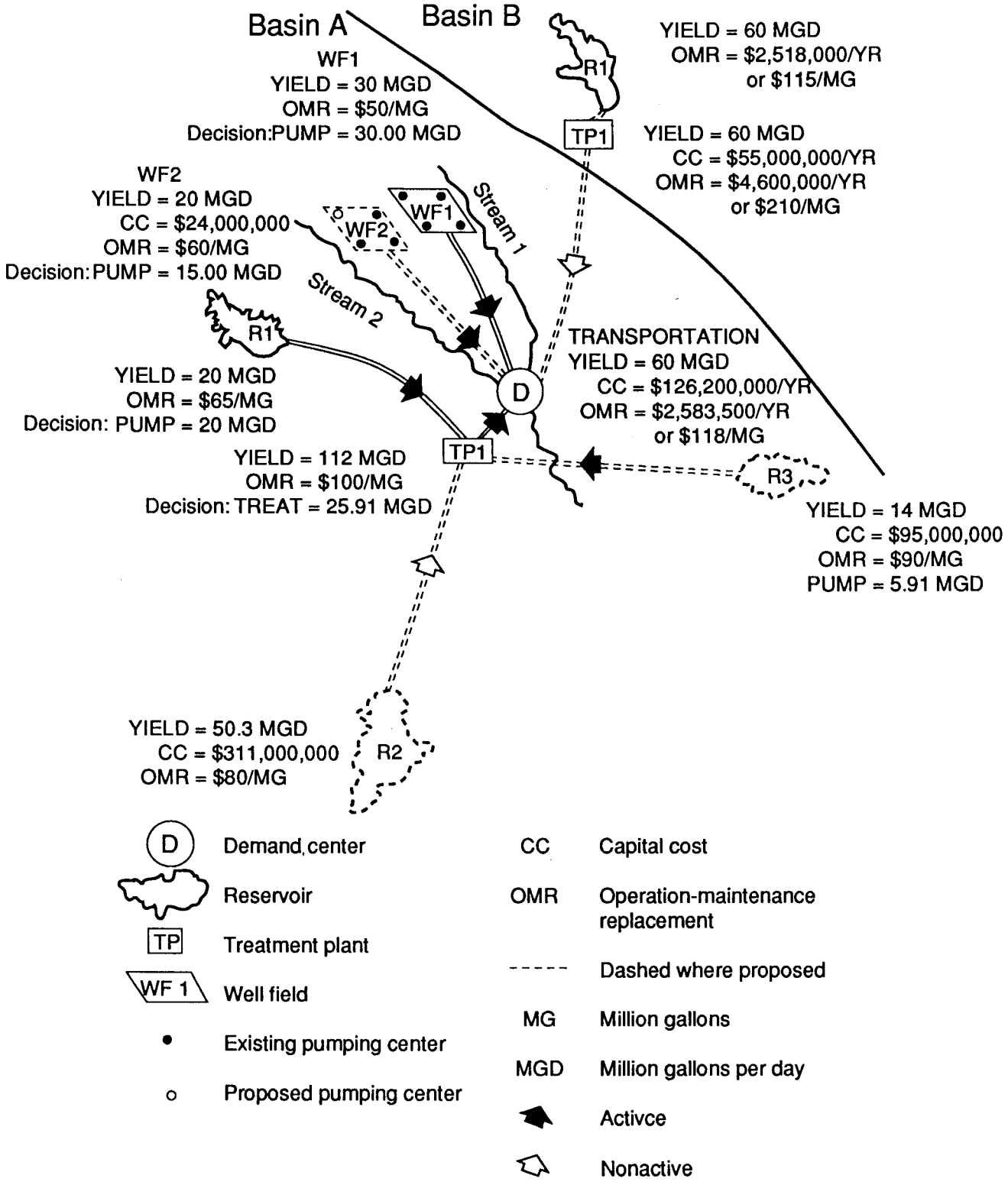


Figure 8: Results of Planning Period 3 fresh water demand = 70.2 MGD

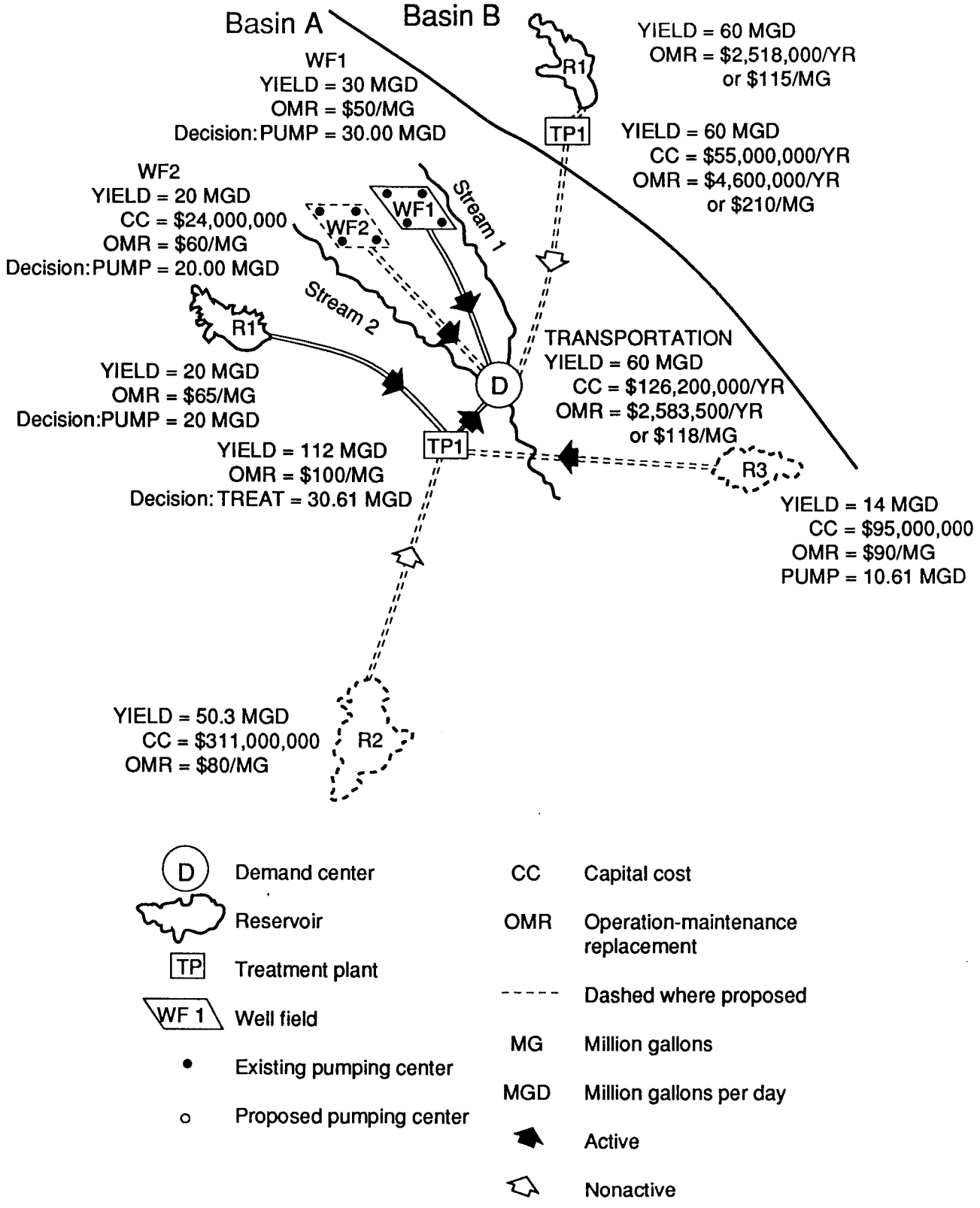


Figure 9: Results of Planning Period 4 fresh water demand = 79.8 MGD
Minimum cost = $62.98 million

During the second period (see Figure 7), the demand is again being supplied for R1, WF1 and WF2. Because the OMR of WF2 is less than that of R1 ($60/MG vs. $65/MG), the model shifts the demand to WF2 and reduces the demand from R1. Well field WF2 will have three pumping centers operational pumping at a rate of 3 (.9125 x 10^4 MG/5 years) = 15.0 MGD.

In the third time period (see Figure 8), reservoir R3 will have to be operational providing water at a rate of 5.91 MGD. Again, only three of the four pumping centers of WF2 are operational pumping a total of 15.00 MGD even though WF2 has a yield of 20.00 MGD. This is because if stress (pumpage) on WF2 is increased further, it will cause further contribution from stream 2 to the aquifer which will violate the minimum streamflow requirements by the end of the planning horizon, i.e. period 4. Since the capital cost of R3 is $95,000,000, it is advisable that the planners re-evaluate the supply and demand functions. Chances are by modification of these functions, the deployment of R3 can be postponed further.

Figure 9 shows the configuration of the system at the end of the fourth planning period. Here reservoirs R1 and R3 and well fields WF1 and WF2 are operational. Three other proposed projects, TP1 in Basin B, pipeline from Basin B to Basin A, and reservoir R2 in Basin A are not needed during the planning horizon. The four components, R1, R3, WF1, and WF2 supply water at a rate of 80.6 MGD, which is 1% above the projected demand of 79.8 MGD. The reason for the difference between the demand and supply is that it is assumed that 1% of the supply is lost in lines.

D. Sensitivity Analysis

Perhaps one of the most important advantages of the screening models using mathematical programming techniques such as the one proposed in equations (5) through (21) is that they provide the planners with a scientific tool to conduct sensitivity analysis on different components. Let us assume that for a set of data and an estimated cost of the pipeline from Basin B to Basin A, the model has chosen pipeline and treatment plant TP1 in Basin B over the development of

reservoirs R2 and R3 in Basin A. The planners are interested in knowing the upper limit of the cost of pipeline alone which is more economical than the development of either reservoirs R2 or R3. By varying the cost of the pipeline from \$162,200,000 up, it is observed that above a capital cost of $\$243.4 \times 10^6$, the development of R2 or R3 are more economical. This is graphically demonstrated in Figure 10. Above a capital cost of pipeline of $\$243.4 \times 10^6$, the model chooses to develop reservoir R3, and the total cost of the project is kept at $\$231.7 \times 10^6$. Clearly, if the planners want to develop the pipeline, a capital cost of $\$243.4 \times 10^6$ is the maximum they should consider. This kind of sensitivity analysis may be conducted on other components to determine their impacts on the total costs. A useful by-product of most mathematical programming packages is the opportunity or marginal cost figures. The interpretation of these figures can help the investigators greatly to identify the coefficients to which the objective function is most sensitive. Therefore, if plans are to improve the input data, resources may be diverted to gather more accurate information about these coefficients. For example, if it is quite obvious that a certain decision variable, when brought into the solution, will change the objective function noticeably, then it is only logical to allocate more funds to gather more information about the coefficients of that variable.

The REDUCED COST column in the COLUMN SECTION of section 5 of Appendix G contains valuable information on the sensitivity of the objective function to continuous variables. The reduced cost is the rate at which the objective function value will deteriorate if a variable currently at zero is arbitrarily forced to increase a small amount. In our example problem, the value of the objective function is in 10^6 dollars and the unit of the continuous variables is in 10^4MG/planning period. For example, the reduced costs of QRSA203 which is the diversion of water from reservoir 3 in basin A during the second planning period is $\$21.574799 \times 10^6/10^4$MG $= \$21.574799 \times 10^2$/MG. Thus, if one decides arbitrarily to divert water from reservoir 3 in Basin A during the second time period without first improving QRSA203's profitability, for each million gallons of water an increase in the objective function equal to $\$21.574799 \times 10^2$ will be realized.

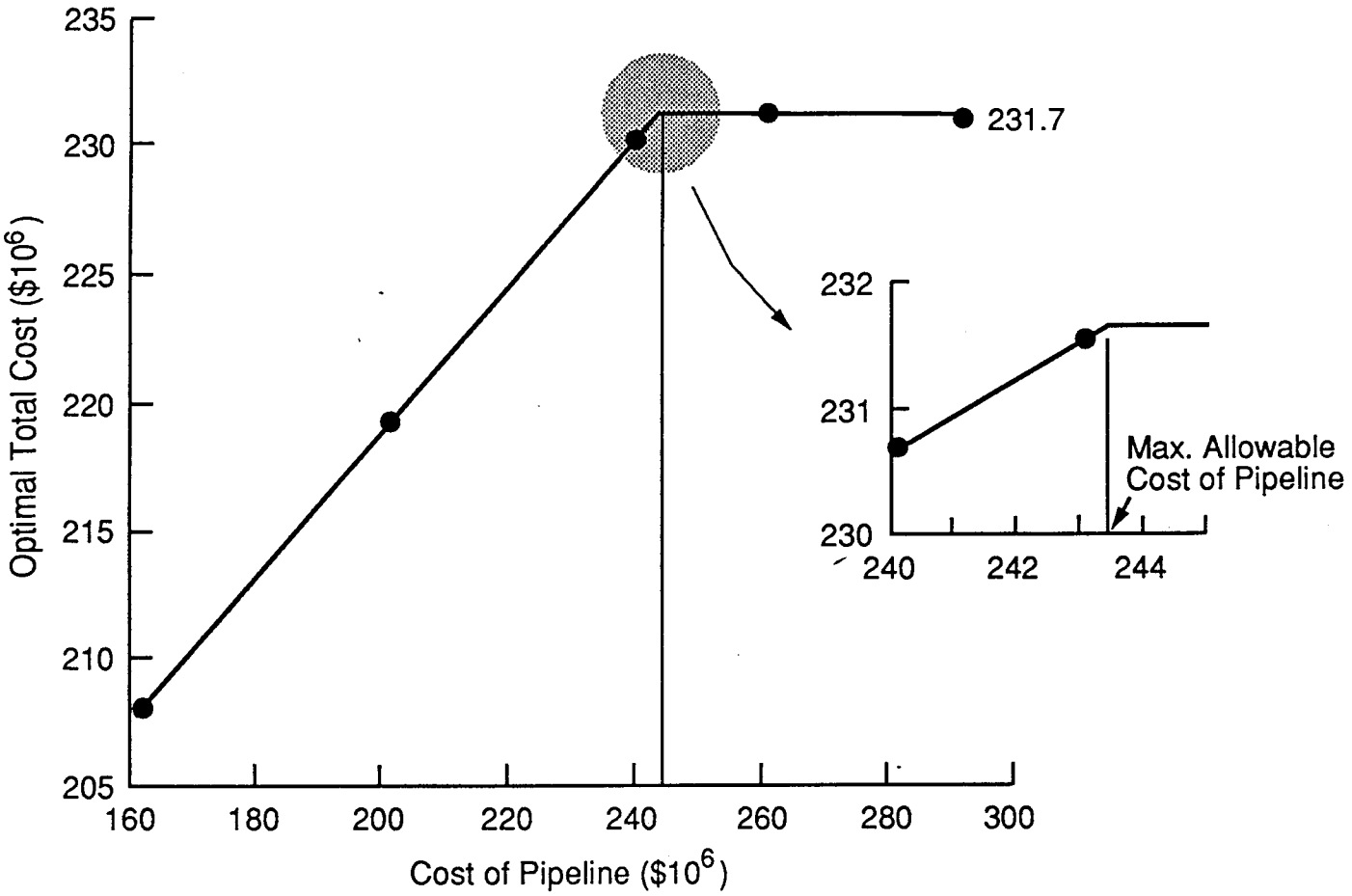


Figure 10: Optimal total cost of the project vs. the capital cost of the pipeline

The reduced cost associated with the variables already in optimal solution is either zero or negative. A zero reduced cost clearly means that the variables' impact on the objective function is already taken into account. A negative reduced cost means that the variable is binding by the upper bound, and a unit increase in the upper bound will hurt the objective function by the amount of the reduced cost.

From this definition of the reduced cost, if we were to spend more funds on collection of new data, then funds should be spent on coefficient of variables with the smallest reduced cost such as QRSA102 with a reduced cost of \$.10695 x 10^2/MG. Reduced costs for integer variables do not have meanings.

DUAL PRICES are another set of output which can be used for sensitivity analysis. These prices are listed in the ROW SECTION of section 5 of Appendix G. If the unit of the objective function is in 10^6 dollars and the units of a constraint row are 10^4 MG/planning period, then the units of the dual prices associated with that particular constraint are 10^6 dollar/10^4 MG/planning period. This represents the rate at which the objective function value will improve as the right-hand-side or constant term of the constraint is increased a small amount, say one unit. A positive dual price indicates that a unit change in the right-hand side of constraint will improve the value of the objective value, while a negative dual price indicates the reverse. A zero dual price indicates that a unit change in the right-hand side does not change the objective value. For example, the dual price of row DTWA1 is -.71299. This means if the right-hand-side of row DTWA1 in section 3 of Appendix G is changed from 9.86237 to 10.86237, the objective value will increase by \$.71299 x $10^6/10^4$ MG 71.29/MG.

V. Summary and Concluding Remarks

Water-resources planning is a multi-disciplinary and dynamic process. This process becomes rather complex when planning is being conducted for a multi-component and/or multi-basin system. In such cases, systems analysis can play a beneficial role to screen the most efficient components.

In this report, a linear integer programming model is adopted which seeks to minimize the present value of all costs (Capital + OMR) of a multi-component and multi-basin water-resources system while satisfying the treated- and raw-water demands in each basin. The use of the model is demonstrated by its application to a multi-component and multi-basin water-resources planning project. The results of this application show that certain proposed elements of the system may not need to be developed. Also demonstrated is how this model may be used to conduct sensitivity analysis on the different components of the system.

The success in obtaining an optimal or near-optimal solution to the proposed model is highly dependent upon the optimization package used. Recent software developments in linear-integer programming methods have made the solution of these models for reasonable-size systems possible. We found LINDO (Schrage, 1984) capable of solving the example problem (32 column integer variables, 60 continuous variables, and 64 rows) satisfactorily. For larger matrices, our experience is rather limited. Chances are a more powerful version of LINDO or another commercial package is needed to solve much larger problems.

The methodology for water-resources planning in this report must be used in the preliminary stages of planning. This procedure is designed to screen the most efficient elements of a system. The computation of stream-aquifer coefficients require a calibrated ground-water model. This by itself is a major element of the planning if data are not available. Since the procedure used in the calculation of the stream-aquifer coefficients is based on the linear systems theory, care must be taken to insure that this theory is basically upheld throughout the planning horizon. This may be done by simulation. Once the optimum aggregated pumpage from a well field is determined, it

is possible to apply a ground-water management model (Maddock, 1974; and Heidari, 1982) to the aquifer to calculate optimum transient or steady-state pumpages of individual wells subject to the optimum aggregate pumpage calculated by the planning model. In this stage, computational difficulties may arise due to the number of wells and time periods. Decomposition techniques (Heidari, 1985a, 1985b) may be used to reduce computational demands. Once the optimum pumpages for each well are calculated, quantity- and quality-simulation models with small time increments must be applied to the area to make sure that assumptions and constraints of the analysis are not violated to an unacceptable level.

A simulation model is also required to evaluate the impact of the planning model's optimum diversions from diversion dams and reservoirs on the system using small time increments. Here the objective is to find out how often the optimum diversion from dams and reservoirs can or cannot be provided. Simulation models developed for river basins (Lenton and Strzepek, 1977; Strzepek and Lenton, 1978; Strzepek et al., 1979; and Heidari and Jian, 1988) may be used to conduct such analysis.

A major factor in the water-resources planning process is the uncertainty associated with supplies and demands. Demand projections of municipal, industrial and agricultural demands are influenced by population growth, governmental policies for growth, market prices and stochastic nature of rainfall. Supplies are also influenced by a great many uncertainties. Not only do uncertainties exist in rainfall-runoff processes leading to stochastic recharge and stream flows, they also exist in the parameters of some of the elements of a system to deliver a certain amount of water. In this report these uncertainties are not addressed.

VI. References

Beard, L.R., Weise, A.O., and Austin, T., 1972. Alternative approaches to water resources system simulation, Proc. of International Symposium on Mathematical Modeling Techniques in Water Resources Systems, Ottawa, Canada, May 9-12.

Buras, N., 1972. Scientific allocation of water resources. American Elsevier, New York, 208 p.

Biswas, A.K., 1976. Systems approach to water management, McGraw-Hill, New York, 429 p.

deLucia, R.J., et al., 1971. Systems analysis in water resources planning. U.S. National Water Commission, Report no. NWC-EES-71-018, 408 p. (PB 208-374).

Dorfman, R., 1965. Formal models in the design of water resources systems. WRR. vol. 1, no. 3, p. 324-336.

Fiering, M.B., and Harrington, J.I., 1968. Systems analysis in river basin planning. Washington, D.C., Pan American Health Organization.

Hall, W.A., and Dracup, J.A., 1970. Water resources systems engineering, McGraw-Hill, New York, 372 p.

Heidari, M., 1982. Application of linear systems theory and linear programming to groundwater management in Kansas. Water Resources Bulletin, vol. 18, no. 6, p.1003-1012.

Heidari, M., 1985a. Optimal management of large aquifers for irrigation activities. IWRA Vth World Congress on Water Resources, June 9-15, Brussels, Belgium, p. 1003-1013.

Heidari, M., 1985b. Optimal management of large aquifers with uncertain demands using minicomputers. Proceedings of the Specialty Conference Hydraulics and Hydrology in the Small Computer Age, HY Div/ASCE, Aug. 12-17, Lake Buena Vista, Florida, p. 1358-1363.

Heidari, M., and Jian, X., 1988. Test and application of MIT river basin simulation model (MITSIM) to the Kansas River, Kansas Geological Survey, Open File Report 88-44, 68 p.

Heidari, M., Maddock, T., III, and Shaukat, N., 1988. Role of systems analysis in improving the cost effectiveness of multi-component water resources projects. Proceedings of Sixth IWRA World Congress on Water Resources, Ottawa, Canada. May 29-June 3.

Hillier, F.S., and Lieberman, G.J., 1968. Introduction to operation research, Holden-Day, San Francisco, 639 p.

Hufschmidt, M.M., 1965. Field level planning of water resources systems. WRR vol. 1, no. 2, p. 147-163.

Hufschmidt, M.M., and Fiering, M.B., 1966. Simulation techniques for design of water resources systems: Harvard University Press, Cambridge, Mass. 212 p.

Jacoby, H.D., and Louks, D.P., 1972. Combined use of optimization and simulation models in river basin planning, WRR, vol. 8, no. 6, p. 1401-1414.

Lenton, R.L., and Strzepek, K.M., 1977. Theoretical and practical characteristics of the MIT River Basin Simulation Model, MIT Dept. of Civil Engineering, Report No. 225, 131 p.

Maass, A., Hufschmidt, M.M., Dorfman, R., Thomas, H.A., Marglin, S.A., and Fair, G.M., 1962. Design of water resource systems, Harvard University Press, Cambridge, Mass., 620 p.

Maddock, T., III, 1974. The operation of a stream-aquifer system under stochastic demand. WRR vol. 10, no. 1, p. 1-10.

Maddock, T., III, 1975. A program to compute aquifer response coefficients, U.S. Geological Survey Open File Report 75-612.

Major, D.C., and Lenton, R.L., 1979. Applied water resource systems planning, Prentice-Hall, Englewood Cliffs, New Jersey, 248 p.

McBean, E.A., Lenton, R., Vicens, G., and Schaake, J.C., Jr., 1972. A general purpose simulation model for analysis of surface water allocation using large time increment, MIT, Ralph M. Parsons Laboratory, Report no. 160.

Moody, D.W., and Maddock, T., III, 1972. A planning model for preliminary network design. Proceedings of International Symposium on Uncertainties in Hydrologic and Water Resources Systems, Tucson, Arizona, U.S.A.

O'Neil, P.G., 1972. A mathematical programming model for planning a regional water resources system: Jour. Inst. Water Engineers, vol. 26, no. 1, p. 47-61.

Schrage, L., 1981. User's manual: linear, integer, and quadratic programming with LINDO, Scientific Press, 86 p.

Schrage, L., 1984. Linear, integer, and quadratic programming with LINDO, The Scientific Press, 274 p.

Strzepek, K.M., and Lenton, R.L., 1978. Analysis of multipurpose river basin systems: guidelines for simulation modeling, MIT, Ralph M. Parsons Laboratory for Water Resources and Hydrodynamics, Report No. 236, 137 p.

Strzepek, K.M., Rosenberg, M.S., Goodman, D.D., Lenton, R.L., and Marks, D.H., 1979. User's manual for MIT river basin simulation model, MIT, Ralph M. Parsons Laboratory for Water Resources and Hydrodynamics. Report No. 242, 363 p.

Wagner, H.M., 1969. Principles of operations research. Prentice-Hall, Englewood Cliffs, New Jersey, 937 p.

Appendix A

Evaluation of Stream-Aquifer Interaction Coefficients

(Paper presented at the National Conference on Engineering Hydrology,

ASCE, Williamsburg, VA, August 3–7, 1987)

EVALUATION OF SURFACE-WATER - GROUND-WATER INTERACTION COEFFICIENTS IN WATER-RESOURCES PLANNING MODEL

Nadeem Shaukat[1]
Manoutch Heidari[2]
Thomas Maddock, III[3]

INTRODUCTION

The planners of water-resources systems are concerned with the identification, selection and scheduling of projects which meet development objectives. A water-resources planning model formulated as a mixed-integer program is used to screen and sequence proposed projects such that the present cost of meeting future water demands is minimized. Surface-water - ground-water interactions are represented in the model by a set of constraints guaranteeing minimum surface-water flow which can be affected by diversions and well-field pumping. When well fields in a planning region are hydraulically connected to streams, consideration must be given to the impact of ground-water pumping on surface-water flows. The streamflow-requirement constraint in the planning model contains surface-water - ground-water interaction coefficients, which relate the quantity of water withdrawn from the stream to the quantity of water pumped by the well fields.

The length of the time period in the planning horizon plays an important role in the simulation of stream-aquifer interaction coefficients. Two approaches for the simulation of these coefficients are investigated. In one approach, long time periods are used, and in the other, shorter time periods are used and the results are aggregated to represent the long time periods. The results obtained from these two approaches are compared.

As expected, the use of stream-aquifer interaction coefficients produced for shorter time periods as compared with those produced for longer time periods showed better results in terms of using the maximum available stream water through well-field pumpage. Furthermore, when the stream-aquifer interaction coefficients were aggregated from smaller time periods of one year to time periods of the planning model, more stream water was shown to be used as compared with the results obtained with non-aggregated stream-aquifer interaction coefficients.

THEORY

When well fields are hydraulically connected with streams in a planning region, a surface-flow-requirement constraint in the water-resources planning model may be required. This constraint contains a surface-water - ground-water interaction coefficient, Ψ, which relates the quantity of water withdrawn from the stream during time

[1]Research Associate, Kansas Geological Survey, Lawrence, KS 66045
[2]M. ASCE, Sen. Sci., Kansas Geological Survey, Lawrence, KS 66045
[3]Professor of Hydrology, University of Arizona, Tucson, AZ 85721

period n and all prior time periods. The stream-aquifer interaction coefficients must be produced for the same time periods as those of the planning periods of the model.

Assume that the aquifer in a planning region possesses the characteristics described below:

1. the saturated thickness of the aquifer is large compared to any drawdown caused by pumpage, i.e. the aquifer's transmissivity is independent of head;
2. water is instantaneously released from storage in the aquifer, i.e. the aquifer's storage coefficient is independent of time;
3. pumping of wells is constant within a given time period, but it may vary from time to time period; and
4. the wells fully penetrate the aquifer, i.e. vertical movement in the aquifer is negligible.

Under these conditions, Maddock (1974) has shown that the stream's cumulative contribution to an aquifer with m pumping wells at time n is

$$F(n) = \sum_{j=1}^{m} \sum_{i=1}^{n} \Psi(j, n-i+1)\, Q(j,i)$$

where

- $F(n)$ = cumulative volume of water removed from stream as of the n-th time period
- m = number of wells
- n = number of time periods
- $\Psi(j,n-i+1)$ = fraction of well water supplied by induced flow from the stream to the j-th well in the i-th time period
- $Q(j,i)$ = volume of water pumped by the j-th well in the i-th time period.

The Ψ parameters are a function of

1. the river-boundary conditions,
2. the distance of wells to the river,
3. the distance between wells,
4. the well radii,
5. the spatial variability of the aquifer's transmissivity and storage coefficients, and,
6. boundary conditions at points other than the river.

AGGREGATED VS. NON-AGGREGATED INTERACTION COEFFICIENTS

The model shown in Figure 1 is the schematic presentation of a water-resources planning program in Kansas. Its mathematical structure, based on that of Maddock and Moody (1974), is a linear integer programming model seeking to minimize the total cost (capital cost + operation, maintenance and replacement cost) of supplying water to the demand area, D. The numbers in the figure represent the yields, capital costs (CC) and operational, maintenance and replacement costs (OMR) associated with each water resource. The water-planning efforts are underway for the demand area (D), the city of Wichita. The area has a certain demand for treated water which increases with time. The future demands for freshwater were developed from past records of demand and projected forward with consideration of the past and predicted further growth of the area.

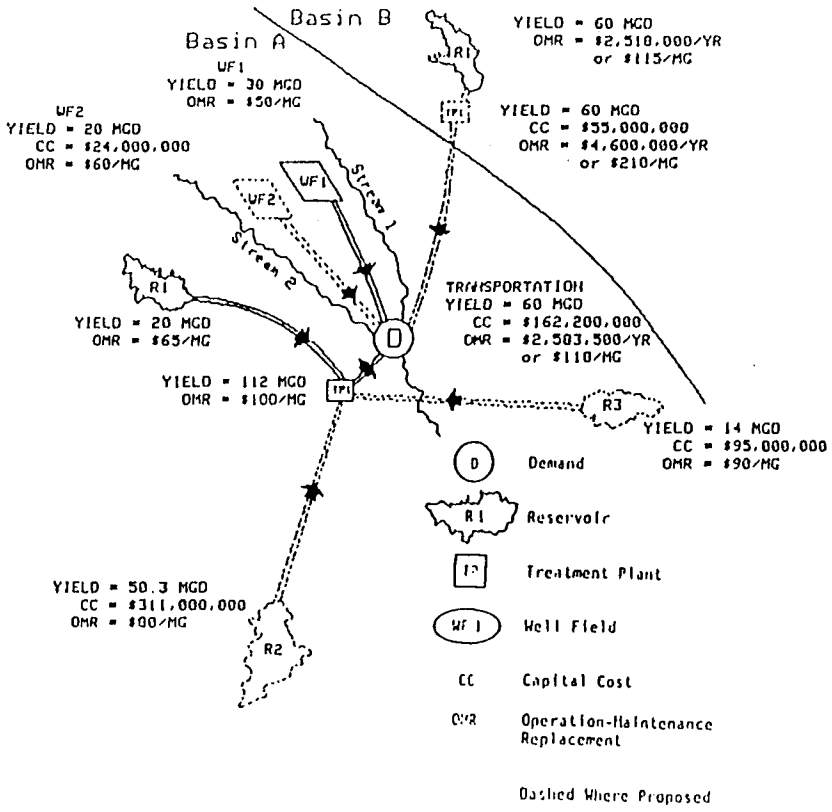


Figure 1: Schematic Presentation of a Water Planning Project.

Two streams, the Little Arkansas (stream 1) and Arkansas (stream 2) are part of the existing planning model. Four wells with yields of 7.5 million gallons per day (MGD) are located next to stream 1. Well field 2 (WF2), consisting of four wells with yields of 5 MGD, each is also proposed in the model for future demand of the area. The mean streamflows in streams 1 and 2 are 169 MGD and 323 MGD, respectively.

The execution of the model produces optimized pumpages from each water resource based on the demand function for the area, keeping the total project cost for the 20-year planning period at a minimum. A discussion of the planning results is beyond the scope of the paper. Here, only the effects of surface-water - ground-water interaction coefficients produced in two different approaches are compared and reported.

In the first approach, the interaction coefficients are produced for time periods of 5, 4 and 2 years. This was done by setting a time increment, Δt, for calculation of these coefficients equal to 5, 4, and 2 years in a mathematical model designed to calculate these coefficients (see Maddock, 1975; Heidari, 1982 a, b). As expected, the results show that the use of smaller time periods for stream-aquifer interaction coefficients are more responsive than those for longer time periods in terms of producing maximum stream water through well-field pumping. The cumulative water used with time periods of 2 years is 100% as compared to 51.66% for 5-year time periods in WF1 (Figure 2a). In WF2, the cumulative stream water used for 2-year time periods is 41.71%, as compared to 13.80% when time periods of 5 years were used (Figure 3a).

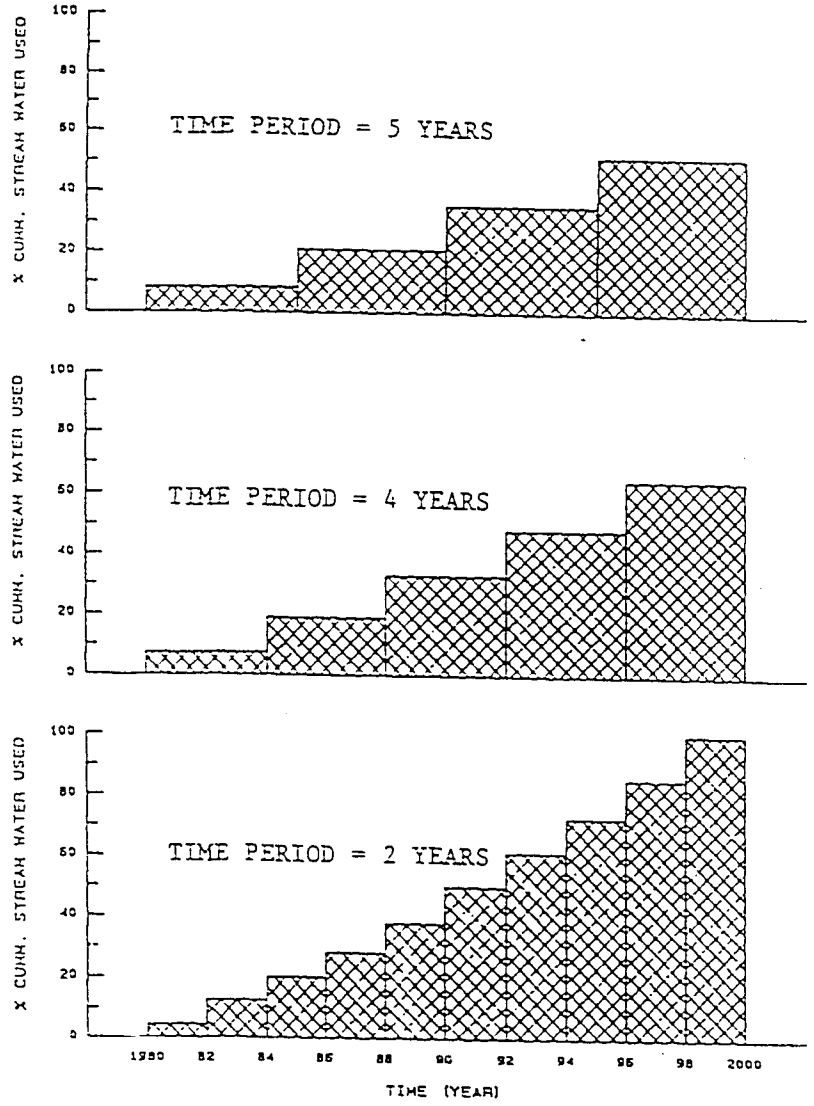


Figure 2a: Percent Cumulative Water Used From Stream 1 Using Non-Aggregated Coefficients.

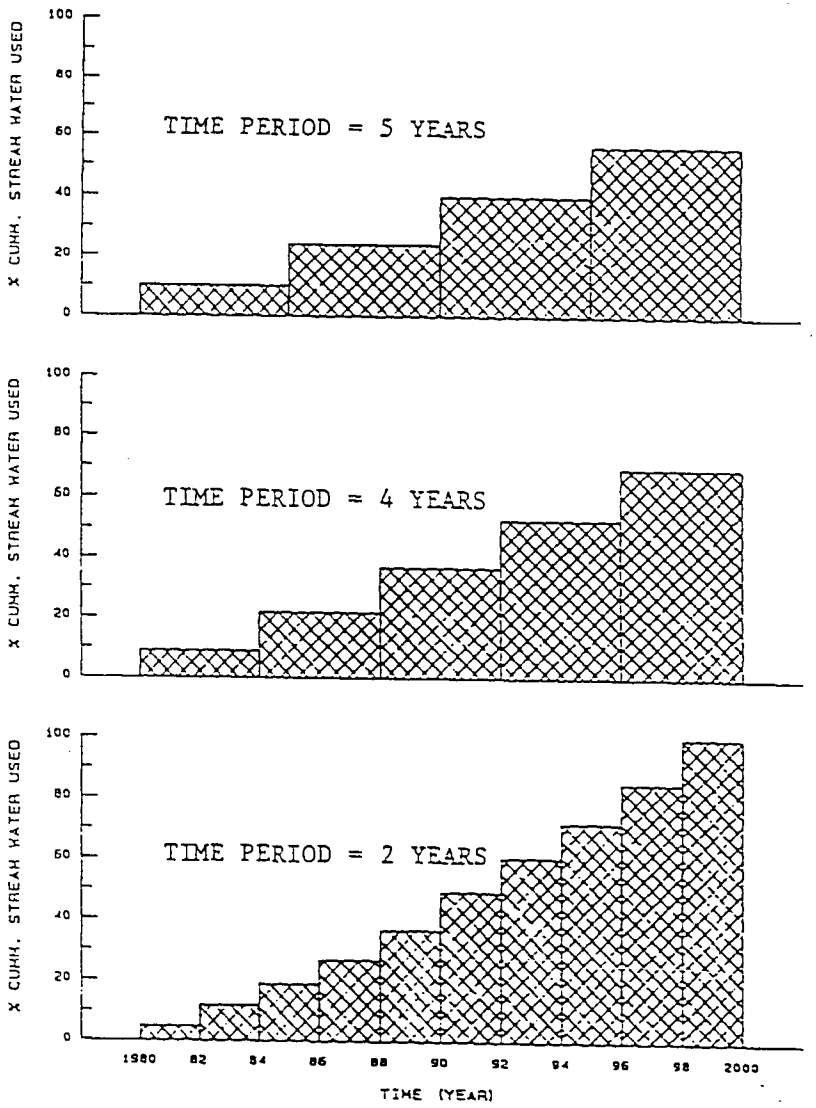


Figure 2b: Percent Cumulative Water Used From Stream 1 Using Aggregated Coefficients.

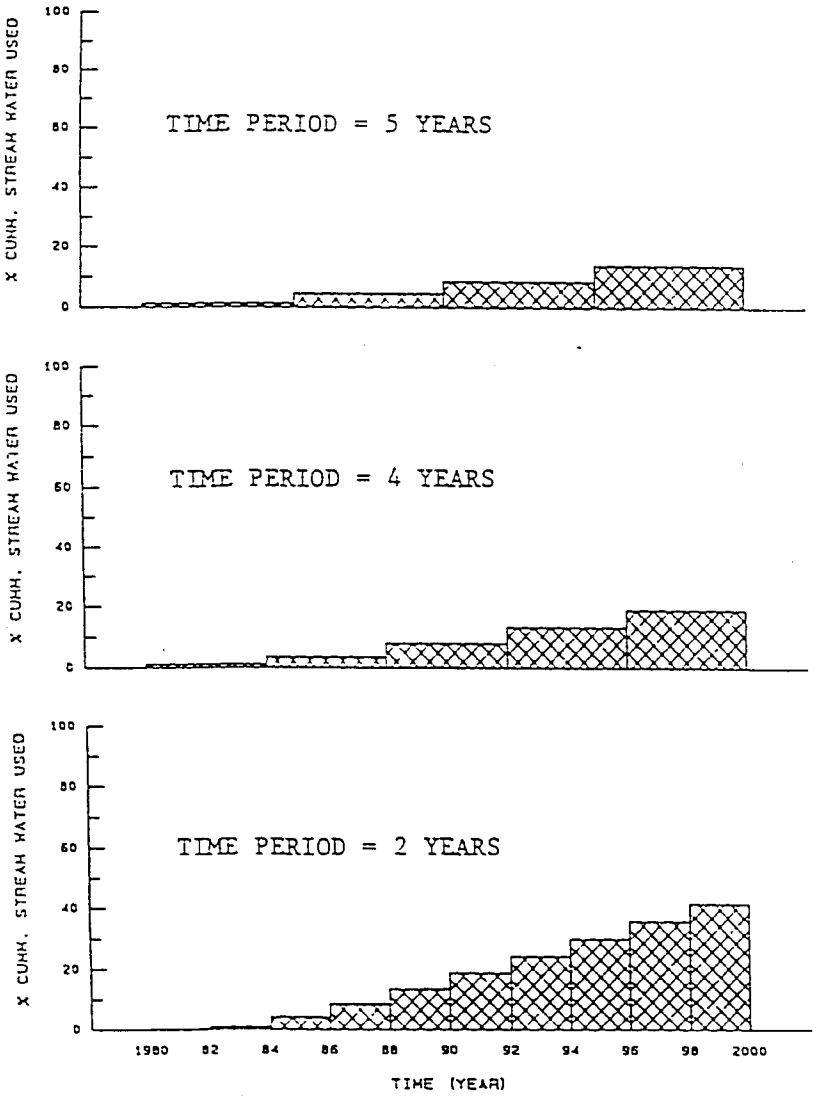


Figure 3a: Percent Cumulative Water Used From Stream 2 Using Non-Aggregated Coefficients.

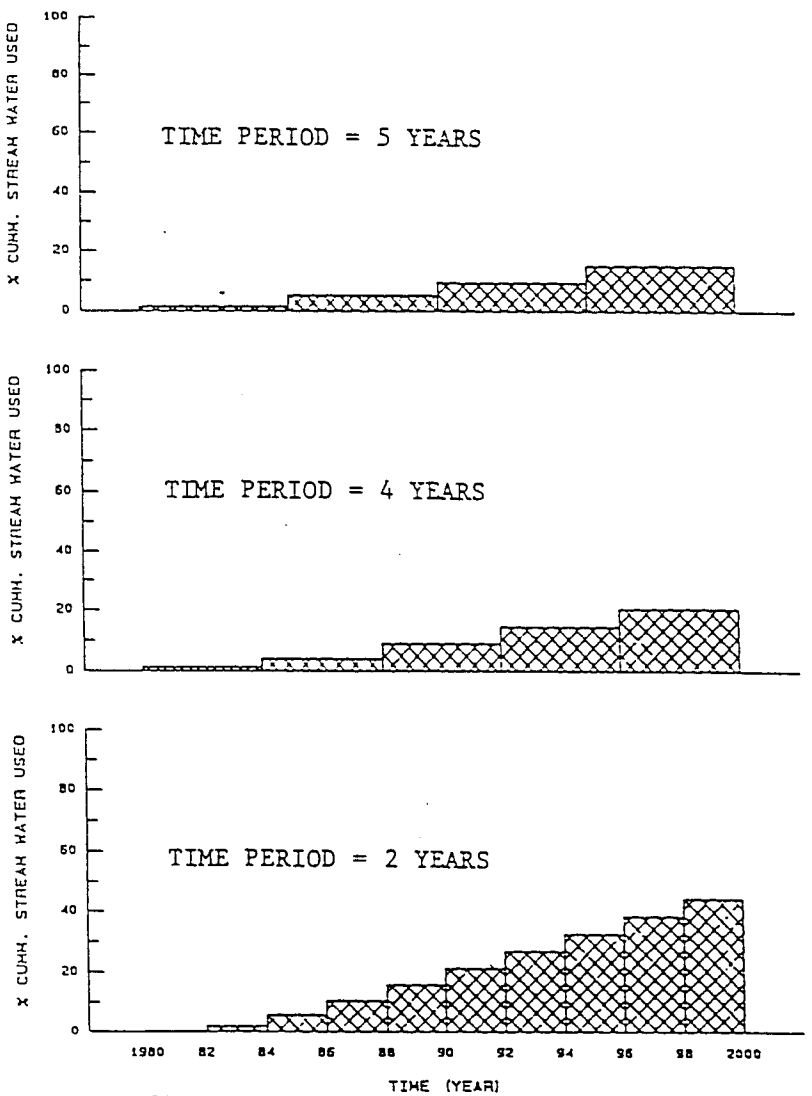


Figure 3b: Percent Cumulative Water Used From Stream 2 Using Aggregated Coefficients.

In the second approach, the interaction coefficients were first produced for 1-year time periods each and then aggregated to the required time periods of 2, 4 and 5 years for the planning model. This was done following the procedure explained by Maddock (1974) for aggregating the stream-aquifer interaction coefficients. The results obtained using these interaction coefficients show that the stream contribution to pumpages increased up to 5% in certain time periods for WF1, for time periods of 5 and 4 years each. No considerable increase in stream-water use was found with aggregated interaction coefficients for WF1 with a time period of 2 years. This is perhaps because a time period of 2 years is short enough, and aggregation of two 1-year stream-aquifer interaction coefficients does not increase the accuracy to any measureable degree. An increase up to 3% in stream-water use for WF2 also was noticed in time periods of 5, 4 and 2 years.

During this comparison study, each well was treated individually for interaction coefficients. The coefficients for all wells in each well field can also be aggregated depending on the contribution of each well in each time period to reduce the computing time. However, contribution of each well to the well field is hard to assign ahead of time during the planning process.

CONCLUSIONS

1. The stream-aquifer coefficients based on linear system's theory enable us to incorporate the stream contribution to an aquifer into a planning model.
2. The degree of accuracy of these coefficients is directly related to the time period used in the ground-water model for their generation.
3. Experience in this study shows that shorter time periods used in generation of these coefficients produce more accurate stream-aquifer coefficients. However, the short time periods may create a large planning model whose solution may require substantial computing efforts.
4. The stream-aquifer coefficients may be generated for short time periods and aggregated for long time periods. This study shows that such aggregation produces somewhat more accurate stream-aquifer coefficients using aggregated coefficients calculated from short time periods.

APPENDIX - REFERENCES

Heidari, M., "Application of linear system's theory and linear programming to groundwater management in Kansas," Water Resources Bulletin, 18(6), 1003-1012, December, 1982a.

Heidari, M., "Groundwater management options for Pawnee Valley of south-central Kansas": Kansas Geological Survey, Groundwater Series 4, 54 p., 1982b.

Maddock, T., III, "The operation of a stream-aquifer system under stochastic demands": Water Resources Research, 10(1), 1-10, 1974.

Maddock, T., III, and D. W. Moody, "Surface water and ground water interaction phenomena in planning models": presented at the symposium on use of computer techniques and automation for water resources systems, Washington, D. C., March 26-28, 1974.

Maddock, T., III, "A program to compute aquifer response coefficients": U.S. Geological Survey Open-File Report 75-612, 1975.

Appendix B

Instructions for Execution of the Water-resources Planning Model

I. Generate Stream-Aquifer Interaction Coefficients

1. Gather information about the physical properties of the area, i.e.,
   a. transmissivity, strativity, boundary conditions
   b. saturated thickness
   c. initial head values
   d. bedrock thickness
   e. leakance
   f. other information needed for a calibrated groundwater model.

2. Decide about the planning parameter i.e.,
   a. number and length of each planning period
   b. number of nodes in X– and Y–directions
   c. distance between each node
   d. number and location of constant head nodes
   e. number and location of observation wells
   f. number and location of pumping wells

3. Prepare a data file according to the format described in Appendix C-I, and name this file as **BETA_INDATA.**

4. Execute the program **PROG_BETA** to calculate the stream-aquifer interaction coeeficients (betas). The program uses an Implicit Alternate Direction scheme.

5. The program will generate two files. The file **BETA_VALUES** will contain the values of stream-aquifer interaction coefficeients to be used in planning model. The other file called **BETA_OUTDATA** (file name provided by user) will contain the information as it is read from the input data file.

II. Prepare Data for the Planning Model

Method A

If you have a Dasher 210/211 terminal from Data General, use the program **PROG_PRE_PLAN** to prepare data for the planning model. This program is a kind of fill-in-blanks for the input required by the model. The forms are saved in 11 files named F1 through F11 and these files must be present in the current directory when running this program. Follow the instructions below:

1. Type in **X PROG_PRE_PLAN** to execute the program.
2. Enter a number for a selected option.
3. Fill in the blanks for each form accordingly.
4. Type in the file name **BETA_VALUES** created in step A when asked for stream-aquifer interaction coefficients.
5. Save the data on a file by typing in any file name.

Method B

If you do not have a Dasher 210/211 terminal, prepare an input data file using any text editor according to the format specifications given in Appendix C-II.

III. Prepare Data for Optimizing Program

The optimization model **PROG_LINDO** accepts the data in a required matrix form. The program **WATER_PLAN_MOD** takes the data prepared in step B and arranges in an acceptable matrix format for **PROG_LINDO**. If you are dealing with big numerals, select the option to scale the matrix down, and provide appropriate scaling factors when asked by the program.

The output file name from this program depends on whether or not the scaling option is selected. If the scaling option is selected, the file name will be **PLAN_MOD_MATRIX_SCALED** otherwise the file name will be **PLAN_MOD_OUT_MATRIX.**

IV. Run the Optimizing Program

Execute the program **PROG_LINDO** and follow instructions from the LINDO manual.

Appendix C-I

Data Format for Program: PROG_BETA

Card#	Column	Format	Variable	Description
1	1	I1	IWRITE	Option for printing transmissivity, initial head, saturated thickness and storitivity maps. 1 = Yes, 0 = No
	2	I1	IDISK	Option to write out drawdown response functions on disk file. 1 = Yes, 0 = No
	3	I1	IVDSK	Option to write out velocity response functions on disk file. 1 = Yes, 0 = No
	4	I1	IMASS	Option to calculate mass balance. 1 = Yes, 0 = No
	5	I1	IVEL	Option to write out velocity field. 1 = Yes, 0 = No
	6	I1	IBETA	Option to write out drawdowns. 1 = Yes, 0 = No
	7	I1	IRAD	Option to perform well radius corrections. 1 = Yes, 0 = No
2	1	I1	ISTDY	Option to simulate under steady state or transient condition. 1 = Yes, 0 = No
	2	I1	IBOUND	Option to set boundary conditions. 1 = Constant head boundary 0 = No constant head boundary
	3	I1	ILEAK	Option to set leakance 1 = Leakage exists 0 = No leakage exists
3	1-2	I2	IDIM	Number of Nodes in X-direction
	3-4	I2	JDIM	Number of Nodes in Y-direction
	11-20	F10.2	DX	Distance between each node (feet)
4	1-2	I2	NUMKT	Number of time periods
	3-22	F20.2	DT	Length of each time period (seconds)
5	1-2	I2	LENG	Number of iterations
	3-22	F20.2	EPS	Convergence factor (0.01)
6	1-10	F10.2	POR	Porosity in fraction (e.g., 0.25)
7		15F5.2	DELX(I), [I=1,IDIM]	Grid scale in X-direction (e.g., 0 to 1)
8		15F5.2	DELY(J), [J=1,JDIM] [J=1,JDIM]	Grid scale in Y-direction (e.g., 0 to 1) (e.g., 0 to 1)
9	1-20	F20.5	SCALET	Scaling factor for transmissivity values

10	1-20 12	F10.0	T(I,J), [J=1,JDIM, I=1,IDIM]	Matrix of transmissivity values
11	1-20	F20.5	SCALEM	Scaling factor for saturated thickness values
12		15I5	ST(I,J), [J=1,JDIM, I=1,IDIM]	Matrix of saturated thickness values
13	1-20	F20.5	SCALEH	Scaling factor for initial head values
14		15I5	IH(I,J) [J=1,JDIM, I=1,IDIM]	Matrix of initial head values
15	1-20	F20.5	SCALES	Scaling factor for storage coefficient
16		15I5	S(I,J), [J=1,JDIM, I=1,IDIM]	Matrix of storage coefficient values
17	1-3	I3	NCLN	Number of no constant-head leaky nodes
18		20I2	IICL(K), JJCL(K), [K=1,NCLN]	Location of no constant-head leaky nodes
19		8E16.5	ZHC(K), [K=1,NCLN]	Hydraulic conductivity values of no constant-head leaky nodes
20		8F8.0	BD(K), [K=1,NCLN]	Matrix of bedrock thickness values
21	1-3	I3	NCHN	Number of constant head nodes
22		20I2	IICH(K), JJCH(K), [K=1,NCHN]	Location of constant head nodes
23	1-2	I2	NOBSW	Number of observation wells
24		20I2	IO(K),JO(K), [K=1,NOBSW]	Location of observation wells
25	1-2	I2	NWEL	Number of pumping wells
26		20I2	II(K),JJ(K), [K=1,NWEL]	Location of pumping wells

Appendix C-II

Data Format for Program: WATER_PLAN_MOD

The input data consists of eleven data sets. Each data set is preceeded by a description card (TLINE) which states which data set is to follow. The eleven data sets are (for examples, see APPENDIX F-I):

1. DATA SET II-E: READ TITLE OF PROBLEM
2. DATA SET II-F: READ NO. OF BASINS, NO. PROJ., ETC.
3. DATA SET II-G: READ NO. OF PROJECTS OF KIND J:
4. DATA SET II-H: READ NO. OF UNTREATED IMPORTS
5. DATA SET II-I: READ NO. OF TREATED IMPORTS
6. DATA SET II-J: READ PARAMETERS
7. DATA SET II-K: READ UNTREATED WATER IMPORT PARAMETERS
8. DATA SET II-L: READ TREATED WATER IMPORT PARAMETERS
9. DATA SET II-M: READ DEMANDS FOR TREATED AND UNTREATED
10. DATA SET II-N: READ SURFACE WATER PARAMETERS
11. DATA SET II-O: READ STREAM-AQUIFER PARAMETERS
12. END OF DATA

NOTE:

1) In the data description lines, the first four letters such as II-E, II-F, etc., must be entered.
The user may use the rest of the space on that line to enter an optional description of the data cards to follow (see Appendix F-1).

2) The user may skip data sets which are not involved in a particular problem. For example, in Appendix F-1, data-set II-K is skipped because there is no import of untreated water in the example problem.

Data Set	Column	Format	Variable	Description
II-E	1-80	20A4	TLINE	Description of data set II-E.
	1-80	20A4	TITLE	Title of the problem
II-F	1-80	20A4	TLINE	Description of data set II-F.
	1-5	I5	NBASIN	No. of basins in analysis.
	6-10	I5	MNPDK	Max. no. of projects of different kinds.
	11-15	I5	NPRODS	No. of time periods in analysis.
	16-20	F5.0	DELT	Length of each time period (years).
	21-25	F5.0	RATE	Interest rate.
	26-30	F5.0	AMRATE	Amortization rate.
II-G	1-80	20A4	TLINE	Description of data set II-G.
		IS	IB	Basin no.
		10IS	NPROJ(IB,J), J=1,MNPDK	No. of projects of kind J in basin IB.
II-H	1-80	20A4	TLINE	Description of data set II-H.
	1-5	I5	IB	No. associated with importing basin.
	6-10	I5	JB	No. associated with exporting basin.
	11-15	I5	NUWIP(IB,JB)	No. of projects to import untreated water from JB to IB.

Data Set	Column	Format	Variable	Description
II-I	1-80	20A4	TLINE	Temporary description of data set II-I.
	1-5	I5	IB	No. associated with importing basin.
	6-10	I5	JB	No. associated with exporting basin.
	11-15	I5	NTWIP(IB,JB)	No. of projects to import treated water from JB to IB.
II-J	1-80	20A4	TLINE	Description of data set II-J.
	1-5	I5	IB	Basin no.
	6-10	I5	K	Project kind.
	11-15	I5	N	N-th project of kind K.
	16-20	I5	IFLAG (IB,K,N)	A flag stating whether project exists or not. =1 project exists =0 project does not exist.
	21-25	I5	LIFE(IB,K,N)	Economic life of project(years).
	26-35	F10.0	YIELD(IB,K,N)	Yield of project(MGD).
	36-45	F10.0	CC(IB,K,N)	Capital cost of project($).
	46-55	F10.0	OMR(IB,K,N)	OMR cost of project($).
II-K	1-80	20A4	TLINE	Description of data set II-K.
	1-5	I5	IBI	No. associated with importing basin.
	6-10	I5	IBE	No. associated with exporting basin.
	11-15	I5	N	No. associated with project.
	16-20	I5	IFUIP(IBI, IBE,N)	A flag stating whether untreated water import project exists or not. =1 project exists. =0 project does not exist.
	21-25	I5	LFUIP(IBI, IBE,N)	Economic life of untreated water import project (years).
	26-35	F10.0	YLDUIP(IBI, IBE,N)	Yield of untreated water import project (MGD).
	36-45	F10.0	CCUIP(IBI, IBE,N)	Capital cost of untreated water import project ($).
	46-55	F10.0	OMUIP(IBI, IBE,N)	OMR cost of untreated water import project ($).
II-L	1-80	20A4	TLINE	Description of data set II-L.
	1-5	I5	IBI	NO. associated with importing basin.
	6-10	I5	IBE	No. associated with exporting basin.
	11-15	I5	N	No. associated with project.
	16-20	I5	IFTIP(IBI, IBE,N)	A flag stating whether treated water import project exists or not. =1 project exists. =0 project does not exist.
	21-25	I5	LFTIP(IBI, IBE,N)	Economic life of treated water import project(years).

Data Set	Column	Format	Variable	Description
	26-35	F10.0	YLDTIP(IBI, IBE,N)	Yield of treated water import project(MGD.
	36-45	F10.0	CCTIP(IBI, IBE,N)	Capital cost of treated water import project($).
	46-55	F10.0	OMTIP(IBI, IBE,N)	OMR cost of treated water import project($).
II-M	1-80	20A4	TLINE	Description of data set II-M.
	1-5	I5	IB	Basin no.
	6-10	I5	IT	Time period.
	11-15	F5.0	DTWATR(IB,IT)	Demand for treated water in basin IB during time period IT (MGD).
	16-20	F5.0	PLTWTR(IB,IT)	Percent line loss of treated water.
	21-25	F5.0	DUWATR(IB,IT)	Demand for untreated water in basin IB during time period IT(MGD).
	26-30	F5.0	PLUWTR(IB,IT)	Percent line loss of untreated water.
	31-35	F5.0	PTTWD(IB,IT)	Percent of total treated water demand for basin IB during time period IT to be satisfied.
	36-40	F5.0	PTUWD(IB,IT)	Percent of total untreated water demand for basin IB during time period IT to be satisfied.
II-N	1-80	20A4	TLINE	Description of data set II-N.
	1-5	I5	IB	Basin no.
	6-10	I5	NRMF(IB)	No. of stream reaches with minimum stream flow requirement in basin IB.
	1-5	I5	IB	Basin no.
	6-10	I5	NR	Stream reach no.
	11-20	F10.0	FLOWIN(IB,NR)	Flow into the reach(MGD).
	21-30	F10.0	FLOWOT(IB,NR)	Minimum flow requirement from the reach(MGD).
	31-35	I5	NPLNR(IB,NR,1)	1-st diversion dam no. in basin IB which diverts water from reach NR.
	36-40	I5	NPLNR(IB,NR,2)	2-nd diversion dam no. in basin IB which diverts water from reach NR.
	31-35	I5	NPLNR(IB,NR,3)	1-st reservoir no. in basin IB which diverts water from reach NR.
	31-35	I5	NPLNR(IB,NR,4)	2-nd reservoir no. in basin IB which diverts water from reach NR.
	31-35	I5	NPLNR(IB,NR,5)	1-st well filed no. in basin IB which diverts water from reach NR.
	31-35	I5	NPLNR(IB,NR,6)	last well1 field no. in basin IB which diverts water from reach NR.
II-O	1-80	20A4	TLINE	Description of data set II-O.
	1-5	I5	IB	Basin no.
	6-10	I5	NN	Well field no.
	11-80	10F6.0	BSA(IB,NN,IT), IT=1,NPRODS	Stream-aquifer interaction coefficients

Appendix D-I

Input Data to PROG_BETA (BETA_INDATA)

to Generate Stream-Aquifer Coefficients

```
0101000
011
6344     5280.00
 4       157680000.00
40                .01
          .25
   1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
   1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
   1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
   1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
   1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
   1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
   1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
   1.00  1.00  1.00

  1000000.00000
      0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    6541. 7231. 128916. 94875. 4398. 2083.  0.    0.  10581.
  11923. 16548.  8173.  0.    0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    1474. 6951. 137458. 105722. 11696. 8550. 12553. 8703. 10513.
  12294.  6463. 18560. 18388. 19085.  0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.  3534. 125319. 150611. 129763. 16868. 8696. 12307. 7247.
   6459. 15152. 13798. 13921. 12247. 13277.  0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0. 15851. 137791. 182111. 144638. 25997. 8306. 7525. 7856.
  20916. 13071.  8150. 12442. 10752. 11245.  0.   10611.  0.    0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0. 10719. 133166. 183361. 165458. 32606. 21194. 9411. 25740.
  25842. 13138. 14708. 15173. 15703. 12703.  7016.  6979.  7506.  0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.  3362. 140444. 196805. 178458. 196819. 118041. 7530. 19766.
  10435. 13229. 14793. 15585. 16106. 10995.  6196.  7173.  7828.  8347.  0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.  3976. 145541. 201333. 202749. 191027. 117319. 7818. 9071.
   8563. 14236. 14594. 18831. 12659. 11231.  6553.  8537.  8062.  8680.  9217.  0.
      0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.  1955.  4548. 135319. 204500. 232874. 220041. 193652. 9262. 9266.
   7855.  2628. 10497. 17043.  9979. 11099.  6520. 13240. 13953. 10037.  8451. 6962.
      0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.    0.    0.  2029. 117125. 192944. 213805. 172430. 165930. 9784. 7673.
```

5119. 7046. 15469. 13395. 9111. 10627. 11833. 13085. 9418. 9106. 8743. 11837.
8148. 9175. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 3664. 112416. 194833. 208416. 166458. 165430. 140597. 4959.
6527. 7337. 15113. 10773. 8171. 10138. 11321. 12884. 11564. 12331. 13009. 10259.
9870. 8594. 8423. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 4046. 100472. 168652. 237611. 195763. 168055. 100305. 61875.
2402. 10354. 14731. 11331. 9543. 7180. 6775. 10446. 13537. 14067. 11094. 9280.
5824. 7590. 11016. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 5084. 115958. 156097. 217875. 210708. 169402. 149916. 103666.
52777. 4764. 13687. 11541. 11578. 8777. 3928. 9960. 12935. 15571. 15851. 8108.
7259. 6099. 10053. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 2923. 130611. 198555. 253264. 127930. 128250. 136264. 117249.
58222. 5123. 14712. 9659. 10874. 4594. 5199. 6361. 7152. 16814. 17043. 17238.
13044. 11594. 8773. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 2592. 5892. 145166. 226736. 121444. 99986. 114069. 142750. 137430.
77736. 6192. 13895. 8525. 9976. 9314. 6037. 7199. 8064. 10780. 15601. 15861.
14268. 10416. 8662. 6886. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 94125. 145833. 205652. 120597. 112777. 119833. 155430. 101472.
55416. 4517. 14925. 9641. 9569. 11370. 4655. 8187. 8800. 11423. 13805. 11192.
10379. 8918. 9083. 7236. 5682. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 3480. 17523. 169388. 160152. 152374. 152514. 132555. 57541.
53444. 10076. 13476. 9398. 16303. 12071. 7381. 8893. 7240. 9750. 12004. 12069.
8634. 8131. 8643. 7518. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 3593. 9053. 27849. 233541. 219277. 163833. 94999. 95763.
49152. 8692. 12016. 10263. 12844. 10953. 8419. 7368. 10182. 14923. 14678. 10328.
7430. 8467. 8041. 7199. 5710. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 6775. 29372. 244624. 230527. 188791. 119500. 106319.
107652. 62166. 9960. 10942. 6974. 7685. 9680. 9358. 13351. 13402. 7805. 9192.
5268. 6451. 7233. 6759. 7854. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 4458. 158083. 241736. 241583. 199777. 171958. 158458.
132000. 107347. 7233. 12055. 3592. 7854. 9159. 11314. 11976. 7467. 3793. 6143.
5442. 5662. 6393. 6185. 7912. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 44347. 113735. 239055. 266583. 224652. 168750. 154680.
169486. 26620. 8558. 7861. 10722. 8342. 8495. 12023. 9256. 5932. 3097. 5266.
6912. 6745. 6129. 5277. 7708. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 70305. 139555. 236666. 263986. 207750. 172097. 150555.
165333. 23555. 24668. 21664. 16740. 7569. 12141. 12756. 7354. 4268. 6192. 9932.
8932. 5122. 7384. 4750. 6868. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 803. 96958. 165583. 241486. 268402. 218514. 203027. 146000.
132472. 20074. 25826. 22520. 16030. 10912. 11944. 8356. 6222. 3706. 8682. 10030.
11069. 7627. 8000. 3979. 6013. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 1345. 96583. 164458. 218819. 259111. 229430. 205902. 138944.
126333. 112458. 24171. 23451. 18002. 17182. 14303. 11131. 4527. 7916. 3865. 5009.
3979. 2567. 7108. 9590. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 1951. 5337. 149375. 245069. 257138. 226861. 195444. 133652.
121000. 107958. 109194. 18532. 17900. 19178. 19141. 15324. 6486. 2988. 3842. 4648.
3231. 3874. 5796. 9777. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 4398. 6018. 149458. 202722. 269624. 252833. 206916. 130097.
115666. 102569. 83027. 13865. 16675. 23534. 22189. 16016. 5002. 2069. 881. 3567.
6733. 5094. 4245. 6370. 5840. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 19574. 150694. 168180. 282764. 265708. 205555. 129888.
108347. 74389. 75611. 12219. 15259. 24664. 23173. 12196. 5291. 2488. 3115. 3840.
5409. 4666. 3363. 3240. 2425. 0. 0. 0.
0. 0. 0. 451388. 416666. 409722. 0. 0. 3009. 3472. 4398. 4398.
3240. 15046. 20370. 16247. 21259. 24743. 166027. 176180. 297236. 265333. 203666. 142680.
107874. 87875. 88791. 69805. 17037. 21060. 22041. 15569. 10958. 3259. 4974. 5159.
5937. 4979. 3384. 3969. 4893. 0. 0. 0.
0. 0. 395833. 424167. 391735. 398610. 427291. 267013. 2730. 3748. 4791. 4781.
5813. 4400. 23210. 26166. 31439. 29402. 181263. 274375. 319444. 266444. 153291. 149222.
129291. 106263. 109180. 99347. 15784. 19893. 18805. 14618. 9905. 4342. 5893. 7120.
6222. 5064. 3388. 2944. 0. 0. 0. 0.
0. 0. 743055. 637639. 537846. 437291. 403264. 378402. 9273. 9111. 3286. 4816.
5076. 5398. 19268. 23719. 27631. 29731. 238361. 302152. 320347. 244083. 159222. 162388.
156541. 123194. 120819. 96388. 17136. 12715. 3638. 8935. 10960. 7208. 4004. 4814.
6108. 5685. 7379. 5187. 0. 0. 0. 0.
0. 0. 1090278. 991806. 581319. 477013. 509999. 472639. 366597. 377570. 293958. 2565.
3837. 4224. 4334. 6295. 6345. 7631. 8512. 9949. 56115. 264000. 234111. 244624.
233986. 31467. 19523. 19935. 18886. 12705. 8782. 2680. 11696. 10069. 6458. 22208.
20986. 5395. 8106. 5745. 0. 0. 0. 0.
0. 0. 1104166. 1072499. 763888. 518888. 481667. 441874. 412222. 422846. 365278. 384375.
5439. 2403. 2814. 4082. 4717. 31960. 34837. 9631. 56564. 331777. 328444. 324638.
321000. 282944. 36189. 29527. 21988. 20939. 19449. 14851. 10175. 7659. 6354. 19763.
12763. 16722. 0. 0. 0. 0.

Appendix D-II

Stream-Aquifer Coefficients (BETA_VALUES)

for Example Problem

1	.2278	.4727	.6572	.7824
2	.3901	.6644	.8149	.9863
3	.5218	.7856	.8991	.9498
4	.7075	.9059	.9639	.9842
5	.4792	.7129	.8281	.8929
6	.6765	.8599	.9269	.9588
7	.7133	.8902	.9496	.9747
8	.7725	.9258	.9697	.9859

Appendix D-III

Output data from the PROG-BETA for the example problem

GROUNDWATER RESPONSE FUNCTION MODEL
PROBLEM:

PRINT OUT TRANSMISSIVITY, SATURATED THICKNESS, INITIAL HEAD AND STORATIVITY MAPS?	NO
WRITE OUT DRAWDOWN RESPONSE FUNCTIONS ON DISK ?	YES
WRITE OUT VELOCITY RESPONSE FUNCTIONS ON DISK ?	NO
CALCULATE MASS BALANCE ?	YES
WRITE OUT AREAL VELOCITY FIELDS ?	NO
WRITE OUT AREAL DRAWDOWNS ?	NO
PERFORM WELL RADIUS CORRECTIONS ?	NO

GRID INFORMATION

NUMBER OF GRID ROWS = 63
NUMBER OF GRID COLUMNS = 44
DISTANCE METRIC = 5280. FT

GRID SCALE IN X DIRECTION

1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0						

GRID SCALE IN Y DIRECTION

1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0							

TIME INFORMATION

TOTAL NUMBER OF TIME PERIODS = 4
TIME PERIOD LENGTH = .15768E+09 SEC.

ITERATION INFORMATION

NUMBER OF PARAMETERS = 40
CONVERGENCE EPSILON = .01000
ITERATION PARMETERS:
CONSTANT HEAD BOUNDARY
CONFINING LAYER WITH LEAKAGE

CONFINING LAYER DATA READ IN
CONSTANT HEAD BOUNDARY POINTS READ IN

OBSERVATION WELLS

WELL	ROW	COLUMN	INITIAL VELOCITY			
"	"	"	(X)	(Y)	(R)	(THETA)
1	37	27	.5473E-05	.1631E-05	.5711E-05	16.6
2	41	30	.6790E-05	.2105E-05	.7109E-05	17.2
3	43	32	.6053E-05	.1526E-05	.6242E-05	14.2
4	46	33	.5788E-05	.2421E-05	.6274E-05	22.7
5	40	21	.5315E-05	.4158E-05	.6748E-05	38.0
6	43	24	.2342E-04	.1158E-04	.2612E-04	26.3
7	45	28	.2868E-04	.1868E-04	.3423E-04	33.1
8	47	31	.2026E-04	.1237E-04	.2374E-04	31.4

PUMPING WELLS

WELL	ROW	COLUMN	INITIAL VELOCITY			
"	"	"	(X)	(Y)	(R)	(THETA)
1	37	27	.0000E+00	.0000E+00	.0000E+00	.0
2	41	30	.0000E+00	.0000E+00	.0000E+00	.0
3	43	32	.0000E+00	.0000E+00	.0000E+00	.0
4	46	33	.0000E+00	.0000E+00	.0000E+00	.0
5	40	21	.0000E+00	.3076E-77	.0000E+00	90.0
6	43	24	.0000E+00	.2375E-77	.0000E+00	90.0
7	45	28	.0000E+00	.0000E+00	.0000E+00	.0
8	47	31	.0000E+00	.0000E+00	.0000E+00	.0

PUMPING WELL " 1
NODE COORDINATE (37,27)
TIME LAG 1
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .7721919 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 22.7808075 PERCENT

PUMPING WELL " 1
NODE COORDINATE (37,27)
TIME LAG 2
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .5272946 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 47.2705383 PERCENT
PUMPING WELL " 1
NODE COORDINATE (37,27)
TIME LAG 3
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .3428131 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 65.7186890 PERCENT
PUMPING WELL " 1
NODE COORDINATE (37,27)
TIME LAG 4
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .2175591 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 78.2440948 PERCENT
PUMPING WELL " 2
NODE COORDINATE (41,30)
TIME LAG 1
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .6098942 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 39.0105896 PERCENT
PUMPING WELL " 2
NODE COORDINATE (41,30)
TIME LAG 2
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .3355878 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 66.4412231 PERCENT
PUMPING WELL " 2
NODE COORDINATE (41,30)
TIME LAG 3
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .1850856 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 81.4914398 PERCENT
PUMPING WELL " 2
NODE COORDINATE (41,30)
TIME LAG 4
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .1036884 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 89.6311646 PERCENT
PUMPING WELL " 3
NODE COORDINATE (43,32)
TIME LAG 1
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .4782084 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 52.1791687 PERCENT
PUMPING WELL " 3
NODE COORDINATE (43,32)
TIME LAG 2

PUMPING RATE ESTIMATED FROM STORAGE LOSS = .2135091 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 78.6490784 PERCENT
PUMPING WELL " 3
NODE COORDINATE (43,32)
TIME LAG 3
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .1009300 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 89.9069977 PERCENT
PUMPING WELL " 3
NODE COORDINATE (43,32)
TIME LAG 4
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .0502237 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 94.9776306 PERCENT
PUMPING WELL " 4
NODE COORDINATE (46,33)
TIME LAG 1
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .2924827 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 70.7517242 PERCENT
PUMPING WELL " 4
NODE COORDINATE (46,33)
TIME LAG 2
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .0940644 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 90.5935669 PERCENT
PUMPING WELL " 4
NODE COORDINATE (46,33)
TIME LAG 3
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .0361027 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 96.3897247 PERCENT
PUMPING WELL " 4
NODE COORDINATE (46,33)
TIME LAG 4
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .0158462 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 98.4153748 PERCENT
PUMPING WELL " 5
NODE COORDINATE (40,21)
TIME LAG 1
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .5208056 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 47.9194336 PERCENT
PUMPING WELL " 5
NODE COORDINATE (40,21)
TIME LAG 2
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .2870904 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 71.2909698 PERCENT

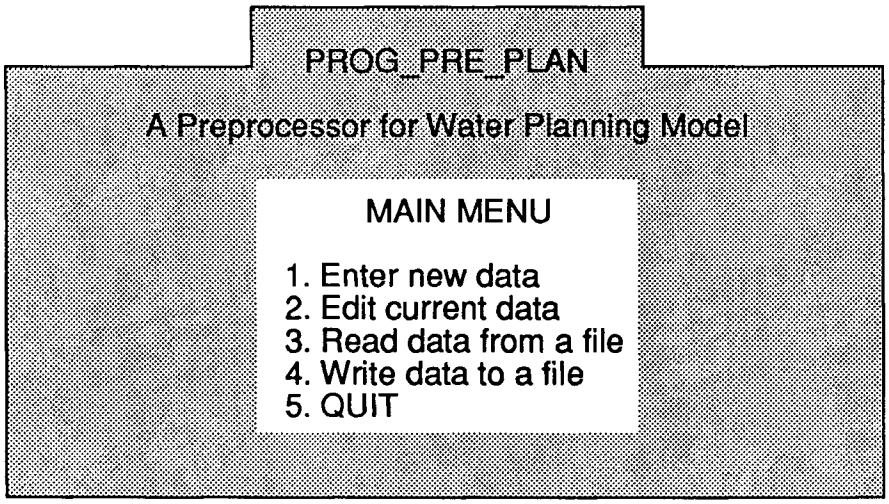
PUMPING WELL " 5
NODE COORDINATE (40,21)
TIME LAG 3
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .1718727 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 82.8127289 PERCENT
PUMPING WELL " 5
NODE COORDINATE (40,21)
TIME LAG 4
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .1070675 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 89.2932587 PERCENT
PUMPING WELL " 6
NODE COORDINATE (43,24)
TIME LAG 1
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .3235232 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 67.6476898 PERCENT
PUMPING WELL " 6
NODE COORDINATE (43,24)
TIME LAG 2
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .1401109 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 85.9889221 PERCENT
PUMPING WELL " 6
NODE COORDINATE (43,24)
TIME LAG 3
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .0730510 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 92.6949005 PERCENT
PUMPING WELL " 6
NODE COORDINATE (43,24)
TIME LAG 4
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .0411885 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 95.8811493 PERCENT
PUMPING WELL " 7
NODE COORDINATE (45,28)
TIME LAG 1
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .2866566 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 71.3343506 PERCENT
PUMPING WELL " 7
NODE COORDINATE (45,28)
TIME LAG 2
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .1098195 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 89.0180511 PERCENT
PUMPING WELL " 7
NODE COORDINATE (45,28)
TIME LAG 3

PUMPING RATE ESTIMATED FROM STORAGE LOSS = .0503595 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 94.9640503 PERCENT
PUMPING WELL " 7
NODE COORDINATE (45,28)
TIME LAG 4
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .0253116 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 97.4688416 PERCENT
PUMPING WELL " 8
NODE COORDINATE (47,31)
TIME LAG 1
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .2275206 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 77.2479401 PERCENT
PUMPING WELL " 8
NODE COORDINATE (47,31)
TIME LAG 2
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .0741882 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 92.5811768 PERCENT
PUMPING WELL " 8
NODE COORDINATE (47,31)
TIME LAG 3
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .0303262 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 96.9673767 PERCENT
PUMPING WELL " 8
NODE COORDINATE (47,31)
TIME LAG 4
PUMPING RATE ESTIMATED FROM STORAGE LOSS = .0140930 FT**3/SEC
ACTUAL PUMPING RATE = 1.0000000 FT**3/SEC
PERCENT DIFFERENCE = 98.5906982 PERCENT

Appendix E

Screens used by Program PROG_PRE_PLAN

01



ENTER OPTION NUMBER: 1

02

ENTER THE TITLE OF THE PROBLEM (MAXIMUM 80 CHARACTERS)

TEST OF WATER_PLAN_MOD VERSION 3.0

ENTER NEW LINE TO CONTINUE, 1 TO EDIT, 2 FOR MAIN MENU:

03

NUMBER OF BASINS	2
MAXIMUM NUMBER OF PROJECTS OF DIFFERENT KINDS	5
NUMBER OF THE PERIODS IN THIS ANALYSIS	4
LENGTH OF EACH TIME PERIOD (YEARS)	5.0
INTEREST RATE (DECIMAL)	.07
INTEREST RATE FOR AMORTIZATION (DECIMAL)	.08

ENTER: TAB for next field, CTRL-T for previous, NEW-LINE to exit

04

PROJECTS IN BASIN 1

Number of Diversion Dams	0
Number of Surface Reservoirs	3
Number of Well Fields	8
Number of Desalination Plants	0
Number of Treatment Plants	1

ENTER: TAB for next field, CTRL-T for previous field, NEW-LINE to exit

04

PROJECTS IN BASIN 2

Number of Diversion Dams	0
Number of Surface Reservoirs	1
Number of Well Fields	0
Number of Desalination Plants	0
Number of Treatment Plants	1

ENTER: TAB for next field, CTRL-T for previous field, NEW-LINE to exit

05

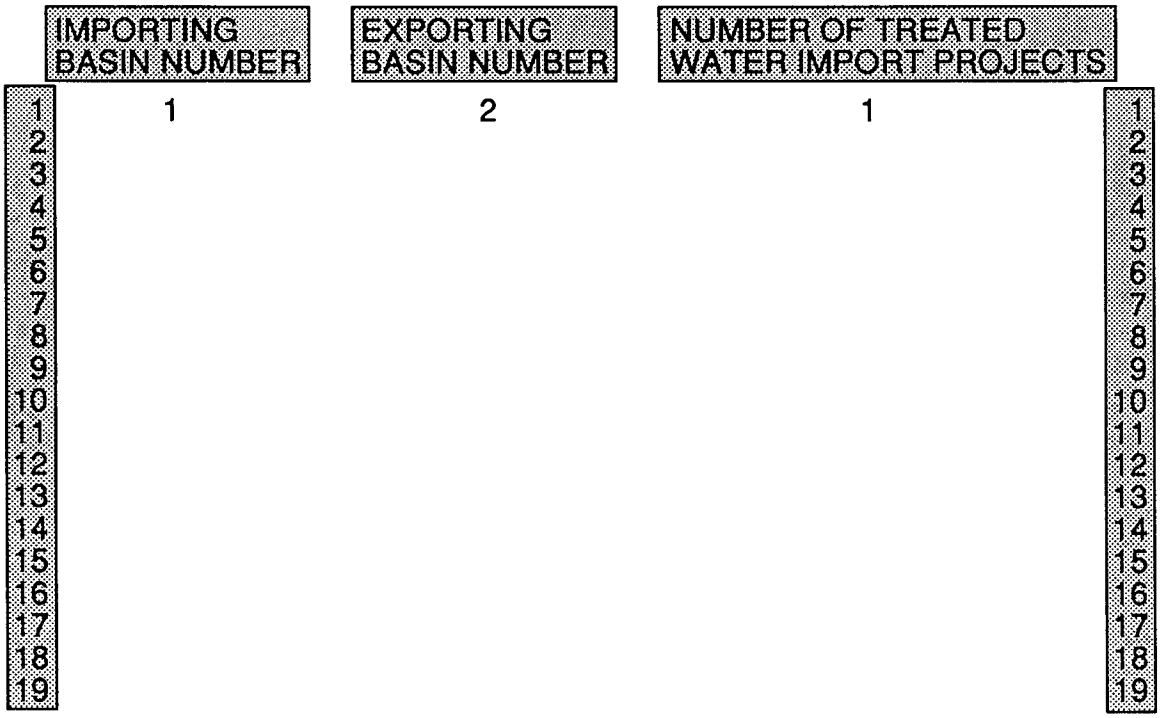
UNTREATED WATER IMPORT/EXPORT

IMPORTING BASIN NUMBER	EXPORTING BASIN NUMBER	NUMBER OF UNTREATED WATER IMPORT PROJECTS
1		1
2		2
3		3
4		4
5		5
6		6
7		7
8		8
9		9
10		10
11		11
12		12
13		13
14		14
15		15
16		16
17		17
18		18
19		19

ENTER: TAB for next field, CTRL-T for previous field, NEW-LINE to exit

06

TREATED WATER IMPORT/EXPORT



ENTER: TAB for next field, CTRL-T for previous field, NEW-LINE to exit

07

PARAMETERS OF WATER PRODUCING PROJECTS IN BASIN 1

	PROJECT KIND K	NO. ASSOC. WITH PROJ. KIND K	PROJECT EXIST?0/1	LIFE OF PROJECT (YEAR)	YIELD (MGD)	CAPITOL COST ($)	OMR ($/MG)
1	2	1	1	30	20.00	0.	65.00
2	2	2	0	30	50.30	311000000.	80.00
3	2	3	0	30	14.00	95000000.	90.00
4	3	1	1	30	7.50	0.	50.00
5	3	2	1	30	7.50	0.	50.00
6	3	3	1	30	7.50	0.	50.00
7	3	4	1	30	7.50	0.	50.00
8	3	5	0	30	5.00	6000000.	60.00
9	3	6	0	30	5.00	6000000.	60.00
10	3	7	0	30	5.00	6000000.	60.00
11	3	8	0	30	5.00	6000000.	60.00
12	5	1	1	30	112.00	0.	100.00
13							
14							
15							
16							
17							
18							
19							

ENTER: TAB for next field, CTRL-T for previous field, NEW-LINE to exit

07

PARAMETERS OF WATER PRODUCING PROJECTS IN BASIN 2

	PROJECT KIND K	NO. ASSOC. WITH PROJ. KIND K	PROJECT EXIST?0/1	LIFE OF PROJECT (YEAR)	YIELD (MGD)	CAPITOL COST ($)	OMR ($/MG)
1	2	1	1	30	60.00	0.	115.00
2	5	1	0	30	60.00	55000000.	210.00
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							

ENTER: TAB for next field, CTRL-T for previous field, NEW-LINE to exit

08

PARAMETERS FOR UNTREATED WATER IMPORT PROJECTS

	IMPORTING BASIN #	EXPORTING BASIN #	NO. ASSOC. WITH PROJ.	PROJECT EXIST?0/1	LIFE OF PROJECT (YEAR)	YIELD (MGD)	CAPITOL COST ($)	OMR ($/MG)
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								

ENTER: TAB for next field, CTRL-T for previous field, NEW-LINE to exit

09

PARAMETERS FOR TREATED WATER IMPORT PROJECTS

	IMPORTING BASIN #	EXPORTING BASIN #	NO. ASSOC. WITH PROJ.	PROJECT EXIST?0/1	LIFE OF PROJECT (YEAR)	YIELD (MGD)	CAPITOL COST ($)	OMR ($/MG)
1	1	2	1	0	30	60.00	162200000.	118.00
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								

ENTER: TAB for next field, CTRL-T for previous field, NEW-LINE to exit

10

DEMANDS FOR TREATED &UNTREATED WATER BASIN 1

	TIME PERIOD	TREATED WATER DEMAND (MGD)	% LOSS OF TREATED WATER (DECIMAL)	UNTREATED WATER DEMAND (MGD)	% LOSS OF UNTREATED WATER (DECIMAL)	TARGET DEMAND OF TREATED WATER (%)	TARGET DEMAND OF UNTREATED WATER (%)	
1	1	53.50	.01	.00	.00	100.00	.00	1
2	2	61.00	.01	.00	.00	100.00	.00	2
3	3	70.20	.01	.00	.00	100.00	.00	3
4	4	79.80	.01	.00	.00	100.00	.00	4
5								5
6								6
7								7
8								8
9								9
10								10
11								11
12								12
13								13
14								14
15								15
16								16

ENTER: TAB for next field, CTRL-T for previous field, NEW-LINE to exit

10

DEMANDS FOR TREATED &UNTREATED WATER BASIN 2

	TIME PERIOD	TREATED WATER DEMAND (MGD)	% LOSS OF TREATED WATER (DECIMAL)	UNTREATED WATER DEMAND (MGD)	% LOSS OF UNTREATED WATER (DECIMAL)	TARGET DEMAND OF TREATED WATER (%)	TARGET DEMAND OF UNTREATED WATER (%)	
1	1	.00	.00	.00	.00	.00	.00	1
2	2	.00	.00	.00	.00	.00	.00	2
3	3	.00	.00	.00	.00	.00	.00	3
4	4	.00	.00	.00	.00	.00	.00	4
5								5
6								6
7								7
8								8
9								9
10								10
11								11
12								12
13								13
14								14
15								15
16								16

ENTER: TAB for next field, CTRL-T for previous field, NEW-LINE to exit

11

SURFACE-WATER - GROUND-WATER INTERACTION PARAMETERS FOR BASIN 2

			PROJECT OF KIND K ASSOCIATED WITH THE REACH (1 = EFFECTING 0 = NOT EFFECTING)						
	REACH NUMBER	UPSTREAM FLOW (CFS)	DOWNSTREAM FLOW (CFS)	DIVERSION DAMS		SURFACE RESERVIORS		WELL FIELDS	
				#1	#2	#1	#2	#1	#2
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									

ENTER: TAB for next field, CTRL-T for previous field, NEW-LINE to exit

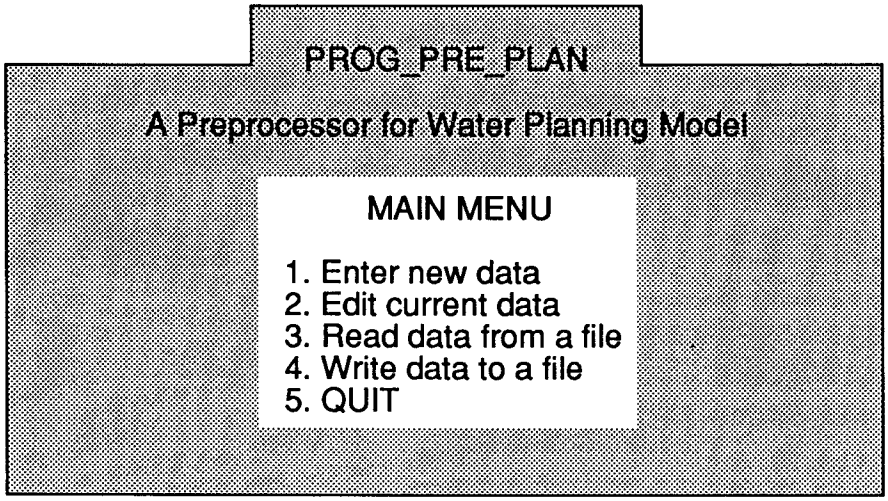
11

SURFACE-WATER - GROUND-WATER INTERACTION PARAMETERS FOR BASIN 2

	REACH NUMBER	UPSTREAM FLOW (CFS)	DOWNSTREAM FLOW (CFS)	PROJECT OF KIND K ASSOCIATED WITH THE REACH (1 = EFFECTING 0 = NOT EFFECTING) DIVERSION DAMS #1	DIVERSION DAMS #2	SURFACE RESERVIORS #1	SURFACE RESERVIORS #2	WELL FIELDS #1	WELL FIELDS #2	
1	1	181.68	12.93	0	0	0	0	1	4	1
2	2	384.06	61.42	0	0	0	0	5	8	2
3										3
4										4
5										5
6										6
7										7
8										8
9										9
10										10
11										11
12										12
13										13
14										14
15										15
16										16
17										17

ENTER: TAB for next field, CTRL-T for previous field, NEW-LINE to exit

01



ENTER OPTION NUMBER: 4

Enter the FILENAME containing the STREAM_AQUIFER_COEFFICIENTS = BETA_VALUES

Appendix F-I

Input Data to Program WATER_PLAN_MOD

for the Example Problem

II-E) READ TITLE OF THE PROBLEM
TEST OF WATER_PLAN_MOD VERSION 3.0
II-F) READ NO. OF BASINS, NO. PF PROJ., NO. OF PERIDS, INTEREST RATES

2	5	4	5.0	.07	.08

II-G) READ NO. OF PROJECTS OF KIND J:

1	0	3	8	0	1
2	0	1	0	0	1

II-H) READ NO. OF UNTREATED WATER IMPORT PROJECTS TO EACH BASIN

1	1	0
1	2	0
2	1	0
2	2	0

II-I) READ NO. OF TREATED WATER IMPORT PROJECTS TO EACH BASIN

1	1	0
1	2	1
2	1	0
2	2	0

II-J) READ PARAMETERS OF WATER PRODUCTION PROJECTS IN EACH BASIN

1	2	1	1	30	20.0	0.	65.0
1	2	2	0	30	50.3	311000000.	80.0
1	2	3	0	30	14.0	95000000.	90.0
1	3	1	1	30	7.5	0.	50.0
1	3	2	1	30	7.5	0.	50.0
1	3	3	1	30	7.5	0.	50.0
1	3	4	1	30	7.5	0.	50.0
1	3	5	0	30	5.0	6000000.	60.0
1	3	6	0	30	5.0	6000000.	60.0
1	3	7	0	30	5.0	6000000.	60.0
1	3	8	0	30	5.0	6000000.	60.0
1	5	1	1	30	112.0	0.	100.0
2	2	1	1	30	60.0	0.	115.0
2	5	1	0	30	60.0	55000000.	210.0

II-L) READ TREATED WATER IMPORT PROJECT PARAMETERS IN EACH BASIN

1	2	1	0	30	60.1	62200000.	118.

II-M) READ DEMANDS FOR TREATED & UNTREATED WATERS IN EACH BASIN

1	1	53.5	.01	.0	.0	100.	.0
1	2	61.0	.01	.0	.0	100.	.0
1	3	70.2	.01	.0	.0	100.	.0
1	4	79.8	.01	.0	.0	100.	.0
2	1	.0	.0	.0	.0	.0	.0
2	2	.0	.0	.0	.0	.0	.0
2	3	.0	.0	.0	.0	.0	.0
2	4	.0	.0	.0	.0	.0	.0

II-N) READ SURFACE WATER REQUIREMENTS PARAMETERS FOR EACH BASIN

1	2								
1	1	181.683	312.931	0	0	0	0	1	4
1	2	384.056	661.423	0	0	0	0	5	8
2	0								

II-O) READ STREAM-AQUIFER INTERACTION COEFFICIENTS

1	1	.2278	.4727	.6572	.7824
1	2	.3901	.6644	.8149	.8963
1	3	.5218	.7865	.8991	.9498
1	4	.7075	.9059	.9639	.9841
1	5	.4792	.7129	.8281	.8929
1	6	.6765	.8599	.9269	.9588
1	7	.7133	.8902	.9496	.9747
1	8	.7723	.9258	.9697	.9859

END OF INPUT DATA

Appendix F-II

Input Scaled MPS Matrix (PLAN_MOD_SCLAED_MATRIX)

to Program PROG_LINDO

for the Example Problem

```
NAME                  TEST OF
ROWS
    N MINCOST
    G IRSA102
    G IRSA103
    G IRSA202
    G IRSA203
    G IRSA302
    G IRSA303
    G IRSA402
    G IRSA403
    G IWFA105
    G IWFA106
    G IWFA107
    G IWFA108
    G IWFA205
    G IWFA206
    G IWFA207
    G IWFA208
    G IWFA305
    G IWFA306
    G IWFA307
    G IWFA308
    G IWFA405
    G IWFA406
    G IWFA407
    G IWFA408
    G ITIAB101
    G ITIAB201
    G ITIAB301
    G ITIAB401
    L LRSA02
    L LRSA03
    L LWFA05
    L LWFA06
    L LWFA07
    L LWFA08
    L LTIAB01
    G DTWA1
    G DTWA2
    G DTWA3
    G DTWA4
    G DTOA1
    G DTOA2
    G DTOA3
    G DTOA4
    L SFLA11
    L SFLA12
    L SFLA21
    L SFLA22
    L SFLA31
    L SFLA32
    L SFLA41
    L SFLA42
```

```
    G ITPB101
    G ITPB201
    G ITPB301
    G ITPB401
    L LTPB01
    G DTWB1
    G DTWB2
    G DTWB3
    G DTWB4
    G DTOB1
    G DTOB2
    G DTOB3
    G DTOB4
COLUMNS
    BEGIN     'MARKER'     .00000     'INTORG'
    DRSA102   MINCOST    292.66381
    DRSA102   IRSA102      9.17975
    DRSA102   IRSA202      9.17975
    DRSA102   IRSA302      9.17975
    DRSA102   IRSA402      9.17975
    DRSA102   LRSA02       1.00000
    DRSA103   MINCOST     89.39886
    DRSA103   IRSA103       2.55500
    DRSA103   IRSA203       2.55500
    DRSA103   IRSA303       2.55500
    DRSA103   IRSA403       2.55500
    DRSA103   LRSA03        1.00000
    DRSA202   MINCOST    179.39435
    DRSA202   IRSA202      9.17975
    DRSA202   IRSA302      9.17975
    DRSA202   IRSA402      9.17975
    DRSA202   LRSA02       1.00000
    DRSA203   MINCOST     54.79890
    DRSA203   IRSA203       2.55500
    DRSA203   IRSA303       2.55500
    DRSA203   IRSA403       2.55500
    DRSA203   LRSA03        1.00000
    DRSA302   MINCOST     98.63472
    DRSA302   IRSA302      9.17975
    DRSA302   IRSA402      9.17975
    DRSA302   LRSA02       1.00000
    DRSA303   MINCOST     30.12957
    DRSA303   IRSA303       2.55500
    DRSA303   IRSA403       2.55500
    DRSA303   LRSA03        1.00000
    DRSA402   MINCOST     41.05419
    DRSA402   IRSA402      9.17975
    DRSA402   LRSA02       1.00000
    DRSA403   MINCOST     12.54066
    DRSA403   IRSA403       2.55500
    DRSA403   LRSA03        1.00000
    DWFA105   MINCOST      5.64624
    DWFA105   IWFA105        .91250
    DWFA105   IWFA205        .91250
```

DWFA105	IWFA305	.91250
DWFA105	IWFA405	.91250
DWFA105	LWFA05	1.00000
DWFA106	MINCOST	5.64624
DWFA106	IWFA106	.91250
DWFA106	IWFA206	.91250
DWFA106	IWFA306	.91250
DWFA106	IWFA406	.91250
DWFA106	LWFA06	1.00000
DWFA107	MINCOST	5.64624
DWFA107	IWFA107	.91250
DWFA107	IWFA207	.91250
DWFA107	IWFA307	.91250
DWFA107	IWFA407	.91250
DWFA107	LWFA07	1.00000
DWFA108	MINCOST	5.64624
DWFA108	IWFA108	.91250
DWFA108	IWFA208	.91250
DWFA108	IWFA308	.91250
DWFA108	IWFA408	.91250
DWFA108	LWFA08	1.00000
DWFA205	MINCOST	3.46098
DWFA205	IWFA205	.91250
DWFA205	IWFA305	.91250
DWFA205	IWFA405	.91250
DWFA205	LWFA05	1.00000
DWFA206	MINCOST	3.46098
DWFA206	IWFA206	.91250
DWFA206	IWFA306	.91250
DWFA206	IWFA406	.91250
DWFA206	LWFA06	1.00000
DWFA207	MINCOST	3.46098
DWFA207	IWFA207	.91250
DWFA207	IWFA307	.91250
DWFA207	IWFA407	.91250
DWFA207	LWFA07	1.00000
DWFA208	MINCOST	3.46098
DWFA208	IWFA208	.91250
DWFA208	IWFA308	.91250
DWFA208	IWFA408	.91250
DWFA208	LWFA08	1.00000
DWFA305	MINCOST	1.90292
DWFA305	IWFA305	.91250
DWFA305	IWFA405	.91250
DWFA305	LWFA05	1.00000
DWFA306	MINCOST	1.90292
DWFA306	IWFA306	.91250
DWFA306	IWFA406	.91250
DWFA306	LWFA06	1.00000
DWFA307	MINCOST	1.90292
DWFA307	IWFA307	.91250
DWFA307	IWFA407	.91250
DWFA307	LWFA07	1.00000
DWFA308	MINCOST	1.90292

DWFA308	IWFA308	.91250	
DWFA308	IWFA408	.91250	
DWFA308	LWFA08	1.00000	
DWFA405	MINCOST	.79204	
DWFA405	IWFA405	.91250	
DWFA405	LWFA05	1.00000	
DWFA406	MINCOST	.79204	
DWFA406	IWFA406	.91250	
DWFA406	LWFA06	1.00000	
DWFA407	MINCOST	.79204	
DWFA407	IWFA407	.91250	
DWFA407	LWFA07	1.00000	
DWFA408	MINCOST	.79204	
DWFA408	IWFA408	.91250	
DWFA408	LWFA08	1.00000	
DTIAB101	MINCOST	152.63678	
DTIAB101	ITIAB101	10.95000	
DTIAB101	ITIAB201	10.95000	
DTIAB101	ITIAB301	10.95000	
DTIAB101	ITIAB401	10.95000	
DTIAB101	LTIAB01	1.00000	
DTIAB201	MINCOST	93.56190	
DTIAB201	ITIAB201	10.95000	
DTIAB201	ITIAB301	10.95000	
DTIAB201	ITIAB401	10.95000	
DTIAB201	LTIAB01	1.00000	
DTIAB301	MINCOST	51.44226	
DTIAB301	ITIAB301	10.95000	
DTIAB301	ITIAB401	10.95000	
DTIAB301	LTIAB01	1.00000	
DTIAB401	MINCOST	21.41154	
DTIAB401	ITIAB401	10.95000	
DTIAB401	LTIAB01	1.00000	
DTPB101	MINCOST	51.75723	
DTPB101	ITPB101	10.95000	
DTPB101	ITPB201	10.95000	
DTPB101	ITPB301	10.95000	
DTPB101	ITPB401	10.95000	
DTPB101	LTPB01	1.00000	
DTPB201	MINCOST	31.72568	
DTPB201	ITPB201	10.95000	
DTPB201	ITPB301	10.95000	
DTPB201	ITPB401	10.95000	
DTPB201	LTPB01	1.00000	
DTPB301	MINCOST	17.44344	
DTPB301	ITPB301	10.95000	
DTPB301	ITPB401	10.95000	
DTPB301	LTPB01	1.00000	
DTPB401	MINCOST	7.26038	
DTPB401	ITPB401	10.95000	
DTPB401	LTPB01	1.00000	
FINIS	'MARKER'	.00000	'INTEND'
QRSA101	MINCOST	.46344	
QRSA101	DTOA1	1.00000	

QRSA102	MINCOST	.57039
QRSA102	IRSA102	-1.00000
QRSA102	DTOA1	1.00000
QRSA103	MINCOST	.64169
QRSA103	IRSA103	-1.00000
QRSA103	DTOA1	1.00000
QRSA201	MINCOST	.33043
QRSA201	DTOA2	1.00000
QRSA202	MINCOST	.40668
QRSA202	IRSA202	-1.00000
QRSA202	DTOA2	1.00000
QRSA203	MINCOST	.45752
QRSA203	IRSA203	-1.00000
QRSA203	DTOA2	1.00000
QRSA301	MINCOST	.23559
QRSA301	DTOA3	1.00000
QRSA302	MINCOST	.28996
QRSA302	IRSA302	-1.00000
QRSA302	DTOA3	1.00000
QRSA303	MINCOST	.32620
QRSA303	IRSA303	-1.00000
QRSA303	DTOA3	1.00000
QRSA401	MINCOST	.16797
QRSA401	DTOA4	1.00000
QRSA402	MINCOST	.20674
QRSA402	IRSA402	-1.00000
QRSA402	DTOA4	1.00000
QRSA403	MINCOST	.23258
QRSA403	IRSA403	-1.00000
QRSA403	DTOA4	1.00000
QWFA101	MINCOST	.35649
QWFA101	DTWA1	1.00000
QWFA101	DTOA1	1.00000
QWFA101	SFLA11	.22780
QWFA101	SFLA21	.47270
QWFA101	SFLA31	.65720
QWFA101	SFLA41	.78240
QWFA102	MINCOST	.35649
QWFA102	DTWA1	1.00000
QWFA102	DTOA1	1.00000
QWFA102	SFLA11	.39010
QWFA102	SFLA21	.66440
QWFA102	SFLA31	.81490
QWFA102	SFLA41	.89630
QWFA103	MINCOST	.35649
QWFA103	DTWA1	1.00000
QWFA103	DTOA1	1.00000
QWFA103	SFLA11	.52180
QWFA103	SFLA21	.78650
QWFA103	SFLA31	.89910
QWFA103	SFLA41	.94980
QWFA104	MINCOST	.35649
QWFA104	DTWA1	1.00000
QWFA104	DTOA1	1.00000

QWFA104	SFLA11	.70750
QWFA104	SFLA21	.90590
QWFA104	SFLA31	.96390
QWFA104	SFLA41	.98410
QWFA105	MINCOST	.42779
QWFA105	IWFA105	-1.00000
QWFA105	DTWA1	1.00000
QWFA105	DTOA1	1.00000
QWFA105	SFLA12	.47920
QWFA105	SFLA22	.71290
QWFA105	SFLA32	.82810
QWFA105	SFLA42	.89290
QWFA106	MINCOST	.42779
QWFA106	IWFA106	-1.00000
QWFA106	DTWA1	1.00000
QWFA106	DTOA1	1.00000
QWFA106	SFLA12	.67650
QWFA106	SFLA22	.85990
QWFA106	SFLA32	.92690
QWFA106	SFLA42	.95880
QWFA107	MINCOST	.42779
QWFA107	IWFA107	-1.00000
QWFA107	DTWA1	1.00000
QWFA107	DTOA1	1.00000
QWFA107	SFLA12	.71330
QWFA107	SFLA22	.89020
QWFA107	SFLA32	.94960
QWFA107	SFLA42	.97470
QWFA108	MINCOST	.42779
QWFA108	IWFA108	-1.00000
QWFA108	DTWA1	1.00000
QWFA108	DTOA1	1.00000
QWFA108	SFLA12	.77230
QWFA108	SFLA22	.92580
QWFA108	SFLA32	.96970
QWFA108	SFLA42	.98590
QWFA201	MINCOST	.25418
QWFA201	DTWA2	1.00000
QWFA201	DTOA2	1.00000
QWFA201	SFLA21	.22780
QWFA201	SFLA31	.47270
QWFA201	SFLA41	.65720
QWFA202	MINCOST	.25418
QWFA202	DTWA2	1.00000
QWFA202	DTOA2	1.00000
QWFA202	SFLA21	.39010
QWFA202	SFLA31	.66440
QWFA202	SFLA41	.81490
QWFA203	MINCOST	.25418
QWFA203	DTWA2	1.00000
QWFA203	DTOA2	1.00000
QWFA203	SFLA21	.52180
QWFA203	SFLA31	.78650
QWFA203	SFLA41	.89910

QWFA204	MINCOST	.25418
QWFA204	DTWA2	1.00000
QWFA204	DTOA2	1.00000
QWFA204	SFLA21	.70750
QWFA204	SFLA31	.90590
QWFA204	SFLA41	.96390
QWFA205	MINCOST	.30501
QWFA205	IWFA205	-1.00000
QWFA205	DTWA2	1.00000
QWFA205	DTOA2	1.00000
QWFA205	SFLA22	.47920
QWFA205	SFLA32	.71290
QWFA205	SFLA42	.82810
QWFA206	MINCOST	.30501
QWFA206	IWFA206	-1.00000
QWFA206	DTWA2	1.00000
QWFA206	DTOA2	1.00000
QWFA206	SFLA22	.67650
QWFA206	SFLA32	.85990
QWFA206	SFLA42	.92690
QWFA207	MINCOST	.30501
QWFA207	IWFA207	-1.00000
QWFA207	DTWA2	1.00000
QWFA207	DTOA2	1.00000
QWFA207	SFLA22	.71330
QWFA207	SFLA32	.89020
QWFA207	SFLA42	.94960
QWFA208	MINCOST	.30501
QWFA208	IWFA208	-1.00000
QWFA208	DTWA2	1.00000
QWFA208	DTOA2	1.00000
QWFA208	SFLA22	.77230
QWFA208	SFLA32	.92580
QWFA208	SFLA42	.96970
QWFA301	MINCOST	.18122
QWFA301	DTWA3	1.00000
QWFA301	DTOA3	1.00000
QWFA301	SFLA31	.22780
QWFA301	SFLA41	.47270
QWFA302	MINCOST	.18122
QWFA302	DTWA3	1.00000
QWFA302	DTOA3	1.00000
QWFA302	SFLA31	.39010
QWFA302	SFLA41	.66440
QWFA303	MINCOST	.18122
QWFA303	DTWA3	1.00000
QWFA303	DTOA3	1.00000
QWFA303	SFLA31	.52180
QWFA303	SFLA41	.78650
QWFA304	MINCOST	.18122
QWFA304	DTWA3	1.00000
QWFA304	DTOA3	1.00000
QWFA304	SFLA31	.70750
QWFA304	SFLA41	.90590

QWFA305	MINCOST	.21747
QWFA305	IWFA305	-1.00000
QWFA305	DTWA3	1.00000
QWFA305	DTOA3	1.00000
QWFA305	SFLA32	.47920
QWFA305	SFLA42	.71290
QWFA306	MINCOST	.21747
QWFA306	IWFA306	-1.00000
QWFA306	DTWA3	1.00000
QWFA306	DTOA3	1.00000
QWFA306	SFLA32	.67650
QWFA306	SFLA42	.85990
QWFA307	MINCOST	.21747
QWFA307	IWFA307	-1.00000
QWFA307	DTWA3	1.00000
QWFA307	DTOA3	1.00000
QWFA307	SFLA32	.71330
QWFA307	SFLA42	.89020
QWFA308	MINCOST	.21747
QWFA308	IWFA308	-1.00000
QWFA308	DTWA3	1.00000
QWFA308	DTOA3	1.00000
QWFA308	SFLA32	.77230
QWFA308	SFLA42	.92580
QWFA401	MINCOST	.12921
QWFA401	DTWA4	1.00000
QWFA401	DTOA4	1.00000
QWFA401	SFLA41	.22780
QWFA402	MINCOST	.12921
QWFA402	DTWA4	1.00000
QWFA402	DTOA4	1.00000
QWFA402	SFLA41	.39010
QWFA403	MINCOST	.12921
QWFA403	DTWA4	1.00000
QWFA403	DTOA4	1.00000
QWFA403	SFLA41	.52180
QWFA404	MINCOST	.12921
QWFA404	DTWA4	1.00000
QWFA404	DTOA4	1.00000
QWFA404	SFLA41	.70750
QWFA405	MINCOST	.15505
QWFA405	IWFA405	-1.00000
QWFA405	DTWA4	1.00000
QWFA405	DTOA4	1.00000
QWFA405	SFLA42	.47920
QWFA406	MINCOST	.15505
QWFA406	IWFA406	-1.00000
QWFA406	DTWA4	1.00000
QWFA406	DTOA4	1.00000
QWFA406	SFLA42	.67650
QWFA407	MINCOST	.15505
QWFA407	IWFA407	-1.00000
QWFA407	DTWA4	1.00000
QWFA407	DTOA4	1.00000

QWFA407	SFLA42	.71330		
QWFA408	MINCOST	.15505		
QWFA408	IWFA408	-1.00000		
QWFA408	DTWA4	1.00000		
QWFA408	DTOA4	1.00000		
QWFA408	SFLA42	.77230		
QTPA101	MINCOST	.71299		
QTPA101	DTWA1	1.00000		
QTPA201	MINCOST	.50835		
QTPA201	DTWA2	1.00000		
QTPA301	MINCOST	.36245		
QTPA301	DTWA3	1.00000		
QTPA401	MINCOST	.25842		
QTPA401	DTWA4	1.00000		
QTIAB101	MINCOST	.84133		
QTIAB101	ITIAB101	-1.00000		
QTIAB101	DTWA1	1.00000	DTWB1	-1.0000
QTIAB101	DTOA1	1.00000	DTOB1	-1.0000
QTIAB201	MINCOST	.59985		
QTIAB201	ITIAB201	-1.00000		
QTIAB201	DTWA2	1.00000	DTWB2	-1.0000
QTIAB201	DTOA2	1.00000	DTOB2	-1.0000
QTIAB301	MINCOST	.42769		
QTIAB301	ITIAB301	-1.00000		
QTIAB301	DTWA3	1.00000	DTWB3	-1.0000
QTIAB301	DTOA3	1.00000	DTOB3	-1.0000
QTIAB401	MINCOST	.30494		
QTIAB401	ITIAB401	-1.00000		
QTIAB401	DTWA4	1.00000	DTWB4	-1.0000
QTIAB401	DTOA4	1.00000	DTOB4	-1.0000
QRSB101	MINCOST	.81994		
QRSB101	DTOB1	1.00000		
QRSB201	MINCOST	.58460		
QRSB201	DTOB2	1.00000		
QRSB301	MINCOST	.41682		
QRSB301	DTOB3	1.00000		
QRSB401	MINCOST	.29718		
QRSB401	DTOB4	1.00000		
QTPB101	MINCOST	1.49727		
QTPB101	ITPB101	-1.00000		
QTPB101	DTWB1	1.00000		
QTPB201	MINCOST	1.06754		
QTPB201	ITPB201	-1.00000		
QTPB201	DTWB2	1.00000		
QTPB301	MINCOST	.76114		
QTPB301	ITPB301	-1.00000		
QTPB301	DTWB3	1.00000		
QTPB401	MINCOST	.54268		
QTPB401	ITPB401	-1.00000		
QTPB401	DTWB4	1.00000		
RHS	.00000			
RHS01	LRSA02	1.00000		
RHS01	LRSA03	1.00000		
RHS01	LWFA05	1.00000		

RHS01	LWFA06	1.00000
RHS01	LWFA07	1.00000
RHS01	LWFA08	1.00000
RHS01	LTIAB01	1.00000
RHS01	DTWA1	9.86237
RHS01	DTWA2	11.24495
RHS01	DTWA3	12.94091
RHS01	DTWA4	14.71061
RHS01	DTOA1	9.86237
RHS01	DTOA2	11.24495
RHS01	DTOA3	12.94091
RHS01	DTOA4	14.71061
RHS01	SFLA11	30.79725
RHS01	SFLA21	30.79725
RHS01	SFLA31	30.79725
RHS01	SFLA41	30.79725
RHS01	SFLA12	58.88049
RHS01	SFLA22	58.88049
RHS01	SFLA32	58.88049
RHS01	SFLA42	58.88049
RHS01	LTPB01	1.00000
RHS01	DTWB1	.00000
RHS01	DTWB2	.00000
RHS01	DTWB3	.00000
RHS01	DTWB4	.00000
RHS01	DTOB1	.00000
RHS01	DTOB2	.00000
RHS01	DTOB3	.00000
RHS01	DTOB4	.00000
BOUNDS	.00000	
UP BND01	QRSA101	3.65000
UP BND01	QRSA102	9.17975
UP BND01	QRSA103	2.55500
UP BND01	QRSA201	3.65000
UP BND01	QRSA202	9.17975
UP BND01	QRSA203	2.55500
UP BND01	QRSA301	3.65000
UP BND01	QRSA302	9.17975
UP BND01	QRSA303	2.55500
UP BND01	QRSA401	3.65000
UP BND01	QRSA402	9.17975
UP BND01	QRSA403	2.55500
UP BND01	QWFA101	1.36875
UP BND01	QWFA102	1.36875
UP BND01	QWFA103	1.36875
UP BND01	QWFA104	1.36875
UP BND01	QWFA105	.91250
UP BND01	QWFA106	.91250
UP BND01	QWFA107	.91250
UP BND01	QWFA108	.91250
UP BND01	QWFA201	1.36875
UP BND01	QWFA202	1.36875
UP BND01	QWFA203	1.36875
UP BND01	QWFA204	1.36875

UP BND01	QWFA205	.91250
UP BND01	QWFA206	.91250
UP BND01	QWFA207	.91250
UP BND01	QWFA208	.91250
UP BND01	QWFA301	1.36875
UP BND01	QWFA302	1.36875
UP BND01	QWFA303	1.36875
UP BND01	QWFA304	1.36875
UP BND01	QWFA305	.91250
UP BND01	QWFA306	.91250
UP BND01	QWFA307	.91250
UP BND01	QWFA308	.91250
UP BND01	QWFA401	1.36875
UP BND01	QWFA402	1.36875
UP BND01	QWFA403	1.36875
UP BND01	QWFA404	1.36875
UP BND01	QWFA405	.91250
UP BND01	QWFA406	.91250
UP BND01	QWFA407	.91250
UP BND01	QWFA408	.91250
UP BND01	QTPA101	20.44000
UP BND01	QTPA201	20.44000
UP BND01	QTPA301	20.44000
UP BND01	QTPA401	20.44000
UP BND01	QRSB101	10.95000
UP BND01	QRSB201	10.95000
UP BND01	QRSB301	10.95000
UP BND01	QRSB401	10.95000
UP BND01	QTPB101	10.95000
UP BND01	QTPB201	10.95000
UP BND01	QTPB301	10.95000
UP BND01	QTPB401	10.95000
UP BND01	QTIAB101	10.95000
UP BND01	QTIAB201	10.95000
UP BND01	QTIAB301	10.95000
UP BND01	QTIAB401	10.95000
UP BND01	DRSA102	1.00000
UP BND01	DRSA103	1.00000
UP BND01	DRSA202	1.00000
UP BND01	DRSA203	1.00000
UP BND01	DRSA302	1.00000
UP BND01	DRSA303	1.00000
UP BND01	DRSA402	1.00000
UP BND01	DRSA403	1.00000
UP BND01	DWFA105	1.00000
UP BND01	DWFA106	1.00000
UP BND01	DWFA107	1.00000
UP BND01	DWFA108	1.00000
UP BND01	DWFA205	1.00000
UP BND01	DWFA206	1.00000
UP BND01	DWFA207	1.00000
UP BND01	DWFA208	1.00000
UP BND01	DWFA305	1.00000
UP BND01	DWFA306	1.00000

```
UP BND01        DWFA307         1.00000
UP BND01        DWFA308         1.00000
UP BND01        DWFA405         1.00000
UP BND01        DWFA406         1.00000
UP BND01        DWFA407         1.00000
UP BND01        DWFA408         1.00000
UP BND01        DTIAB101        1.00000
UP BND01        DTIAB201        1.00000
UP BND01        DTIAB301        1.00000
UP BND01        DTIAB401        1.00000
UP BND01        DTPB101         1.00000
UP BND01        DTPB201         1.00000
UP BND01        DTPB301         1.00000
UP BND01        DTPB401         1.00000
ENDATA
```

Appendix G

Partial Listing of Output from Program PROG_LINDO for the Example Problem

1. DESCRIPTION OF THE VARIABLES IN THE EXAMPLE PROBLEM

A- INTEGER VARIABLES:

DRSatn= 0-1 decision variable to build reservior n in basin a at time t.

DWFatn= 0-1 decision variable to develop well field n in basin a at time t.

DTIabtn=0-1 decision variable to build pipeline n to import treated water from basin b to basin a at time t.

DTPatn= 0-1 decision variable to build treatment plant n in basin a at time t.

B- CONTINEOUS VARIABLES:

QRSatn= decision variable to divert water from reservoir n in basin a during time period t.

QWFatn= decision variable to pump water from well field n in basin a during time period t.

QTIabtn=decision variable to import treated water from basin b to basin a during time

QTPatn= decision variable to treat water in the n-th treatment plant in basin a during time

2. OBJECTIVE FUNCTION OF THE EXAMPLE PROBLEM

MIN 292.66382 DRSA102 + 89.39886 DRSA103
+ 179.39435 DRSA202 + 54.7989 DRSA203 + 98.63472 DRSA302
+ 30.12956 DRSA303 + 41.05418 DRSA402 + 12.54066 DRSA403
+ 5.64624 DWFA105 + 5.64624 DWFA106 + 5.64624 DWFA107
+ 5.64624 DWFA108 + 3.46098 DWFA205 + 3.46098 DWFA206
+ 3.46098 DWFA207 + 3.46098 DWFA208 + 1.90292 DWFA305
+ 1.90292 DWFA306 + 1.90292 DWFA307 + 1.90292 DWFA308
+ 0.79204 DWFA405 + 0.79204 DWFA406 + 0.79204 DWFA407
+ 0.79204 DWFA408 + 152.63678 DTIAB101 + 93.5619 DTIAB201
+ 51.44226 DTIAB301 + 21.41154 DTIAB401 + 51.75723 DTPB101
+ 31.72568 DTPB201 + 17.44344 DTPB301 + 7.26038 DTPB401
+ 0.46344 QRSA101 + 0.57039 QRSA102 + 0.64169 QRSA103
+ 0.33043 QRSA201 + 0.40668 QRSA202 + 0.45752 QRSA203
+ 0.23559 QRSA301 + 0.28996 QRSA302 + 0.3262 QRSA303
+ 0.16797 QRSA401 + 0.20674 QRSA402 + 0.23258 QRSA403
+ 0.35649 QWFA101 + 0.35649 QWFA102 + 0.35649 QWFA103
+ 0.35649 QWFA104 + 0.42779 QWFA105 + 0.42779 QWFA106
+ 0.42779 QWFA107 + 0.42779 QWFA108 + 0.25418 QWFA201
+ 0.25418 QWFA202 + 0.25418 QWFA203 + 0.25418 QWFA204
+ 0.30501 QWFA205 + 0.30501 QWFA206 + 0.30501 QWFA207
+ 0.30501 QWFA208 + 0.18122 QWFA301 + 0.18122 QWFA302
+ 0.18122 QWFA303 + 0.18122 QWFA304 + 0.21747 QWFA305
+ 0.21747 QWFA306 + 0.21747 QWFA307 + 0.21747 QWFA308
+ 0.12921 QWFA401 + 0.12921 QWFA402 + 0.12921 QWFA403
+ 0.12921 QWFA404 + 0.15505 QWFA405 + 0.15505 QWFA406
+ 0.15505 QWFA407 + 0.15505 QWFA408 + 0.71299 QTPA101
+ 0.50835 QTPA201 + 0.36245 QTPA301 + 0.25842 QTPA401

+ 0.84133 QTIAB101 + 0.59985 QTIAB201 + 0.42769 QTIAB301
+ 0.30494 QTIAB401 + 0.81994 QRSB101 + 0.5846 QRSB201
+ 0.41682 QRSB301 + 0.29718 QRSB401 + 1.49727 QTPB101
+ 1.06754 QTPB201 + 0.76114 QTPB301 + 0.54268 QTPB401

3. CONSTRAINTS OF THE EXAMPLE PROBLEM

SUBJECT TO
IRSA102) 9.17975 DRSA102 - QRSA102 >= 0
IRSA103) 2.555 DRSA103 - QRSA103 >= 0
IRSA202) 9.17975 DRSA102 + 9.17975 DRSA202 - QRSA202 >= 0
IRSA203) 2.555 DRSA103 + 2.555 DRSA203 - QRSA203 >= 0
IRSA302) 9.17975 DRSA102 + 9.17975 DRSA202 + 9.17975 DRSA302 - QRSA302 >= 0
IRSA303) 2.555 DRSA103 + 2.555 DRSA203 + 2.555 DRSA303 - QRSA303 >= 0
IRSA402) 9.17975 DRSA102 + 9.17975 DRSA202 + 9.17975 DRSA302 + 9.17975 DRSA402 - QRSA402 >= 0
IRSA403) 2.555 DRSA103 + 2.555 DRSA203 + 2.555 DRSA303 + 2.555 DRSA403 - QRSA403 >= 0
IWFA105) 0.9125 DWFA105 - QWFA105 >= 0
IWFA106) 0.9125 DWFA106 - QWFA106 >= 0
IWFA107) 0.9125 DWFA107 - QWFA107 >= 0
IWFA108) 0.9125 DWFA108 - QWFA108 >= 0
IWFA205) 0.9125 DWFA105 + 0.9125 DWFA205 - QWFA205 >= 0
IWFA206) 0.9125 DWFA106 + 0.9125 DWFA206 - QWFA206 >= 0
IWFA207) 0.9125 DWFA107 + 0.9125 DWFA207 - QWFA207 >= 0
IWFA208) 0.9125 DWFA108 + 0.9125 DWFA208 - QWFA208 >= 0
IWFA305) 0.9125 DWFA105 + 0.9125 DWFA205 + 0.9125 DWFA305 - QWFA305 >= 0
IWFA306) 0.9125 DWFA106 + 0.9125 DWFA206 + 0.9125 DWFA306 - QWFA306 >= 0
IWFA307) 0.9125 DWFA107 + 0.9125 DWFA207 + 0.9125 DWFA307 - QWFA307 >= 0
IWFA308) 0.9125 DWFA108 + 0.9125 DWFA208 + 0.9125 DWFA308 - QWFA308 >= 0
IWFA405) 0.9125 DWFA105 + 0.9125 DWFA205 + 0.9125 DWFA305 + 0.9125 DWFA405 - QWFA405 >= 0
IWFA406) 0.9125 DWFA106 + 0.9125 DWFA206 + 0.9125 DWFA306 + 0.9125 DWFA406 - QWFA406 >= 0
IWFA407) 0.9125 DWFA107 + 0.9125 DWFA207 + 0.9125 DWFA307 + 0.9125 DWFA407 - QWFA407 >= 0
IWFA408) 0.9125 DWFA108 + 0.9125 DWFA208 + 0.9125 DWFA308 + 0.9125 DWFA408 - QWFA408 >= 0
ITIAB101) 10.95 DTIAB101 - QTIAB101 >= 0
ITIAB201) 10.95 DTIAB101 + 10.95 DTIAB201 - QTIAB201 >= 0
ITIAB301) 10.95 DTIAB101 + 10.95 DTIAB201 + 10.95 DTIAB301 - QTIAB301 >= 0
ITIAB401) 10.95 DTIAB101 + 10.95 DTIAB201 + 10.95 DTIAB301 + 10.95 DTIAB401 - QTIAB401 >= 0
LRSA02) DRSA102 + DRSA202 + DRSA302 + DRSA402 <= 1
LRSA03) DRSA103 + DRSA203 + DRSA303 + DRSA403 <= 1
LWFA05) DWFA105 + DWFA205 + DWFA305 + DWFA405 <= 1
LWFA06) DWFA106 + DWFA206 + DWFA306 + DWFA406 <= 1
LWFA07) DWFA107 + DWFA207 + DWFA307 + DWFA407 <= 1

```
LWFA08)      DWFA108 + DWFA208 + DWFA308 + DWFA408 <=          1
LTIAB01)     DTIAB101 + DTIAB201 + DTIAB301 + DTIAB401 <=      1
DTWA1)  QWFA101 + QWFA102 + QWFA103 + QWFA104 + QWFA105 + QWFA106 +
   QWFA107 + QWFA108 + QTPA101 + QTIAB101  >=      9.86237
DTWA2)  QWFA201 + QWFA202 + QWFA203 + QWFA204 + QWFA205 + QWFA206 +
   QWFA207 + QWFA208 + QTPA201 + QTIAB201 >=       11.24495
DTWA3)  QWFA301 + QWFA302 + QWFA303 + QWFA304 + QWFA305 + QWFA306 +
   QWFA307 + QWFA308 + QTPA301 + QTIAB301 >=       12.94091
DTWA4)  QWFA401 + QWFA402 + QWFA403 + QWFA404 + QWFA405 + QWFA406 +
   QWFA407 + QWFA408 + QTPA401 + QTIAB401  >=      14.71061
DTOA1)  QRSA101 + QRSA102 + QRSA103 + QWFA101 + QWFA102 + QWFA103 +
   QWFA104 + QWFA105 + QWFA106 + QWFA107 + QWFA108 + QTIAB101
   >= 9.86237
DTOA2)  QRSA201 + QRSA202 + QRSA203 + QWFA201 + QWFA202 + QWFA203 +
   QWFA204 + QWFA205 + QWFA206 + QWFA207 + QWFA208+ QTIAB201
   >= 11.24495
DTOA3)  QRSA301 + QRSA302 + QRSA303 + QWFA301 + QWFA302 + QWFA303 +
   QWFA304 + QWFA305 + QWFA306 + QWFA307 + QWFA308 + QTIAB301
   >= 12.94091
DTOA4)  QRSA401 + QRSA402 + QRSA403 + QWFA401 + QWFA402 + QWFA403 +
   QWFA404 + QWFA405 + QWFA406 + QWFA407 + QWFA408 + QTIAB401
   >= 14.71061
SFLA11)  0.2278 QWFA101 + 0.3901 QWFA102 + 0.5218 QWFA103+ 0.7075 QWFA104
   <= 30.79726
SFLA12)  0.4792 QWFA105 + 0.6765 QWFA106 + 0.7133 QWFA107 + 0.7723 QWFA108
   <= 58.88049
SFLA21)  0.4727 QWFA101 + 0.6644 QWFA102 + 0.7865 QWFA103 + 0.9059 QWFA104
   + 0.2278 QWFA201 + 0.3901 QWFA202 + 0.5218 QWFA203 + 0.7075 QWFA204
   <= 30.79726
SFLA22)  0.7129 QWFA105 + 0.8599 QWFA106 + 0.8902 QWFA107 + 0.9258 QWFA108
   + 0.4792 QWFA205 + 0.6765 QWFA206 + 0.7133 QWFA207 + 0.7723 QWFA208
   <= 58.88049
SFLA31)  0.6572 QWFA101 + 0.8149 QWFA102 + 0.8991 QWFA103 + 0.9639 QWFA104
   + 0.4727 QWFA201 + 0.6644 QWFA202 + 0.7865 QWFA203 + 0.9059 QWFA204 +
   0.2278 QWFA301 + 0.3901 QWFA302 + 0.5218 QWFA303 + 0.7075 QWFA304
   <= 30.79726
SFLA32)  0.8281 QWFA105 + 0.9269 QWFA106 + 0.9496 QWFA107
   + 0.9697 QWFA108 + 0.7129 QWFA205 + 0.8599 QWFA206
   + 0.8902 QWFA207 + 0.9258 QWFA208 + 0.4792 QWFA305
   + 0.6765 QWFA306 + 0.7133 QWFA307 + 0.7723 QWFA308
   <= 58.88049
SFLA41)  0.7824 QWFA101 + 0.8963 QWFA102 + 0.9498 QWFA103
   + 0.9841 QWFA104 + 0.6572 QWFA201 + 0.8149 QWFA202
   + 0.8991 QWFA203 + 0.9639 QWFA204 + 0.4727 QWFA301
   + 0.6644 QWFA302 + 0.7865 QWFA303 + 0.9059 QWFA304
   + 0.2278 QWFA401 + 0.3901 QWFA402 + 0.5218 QWFA403
   + 0.7075 QWFA404 <= 30.79726
SFLA42)  0.8929 QWFA105 + 0.9588 QWFA106 + 0.9747 QWFA107
   + 0.9859 QWFA108 + 0.8281 QWFA205 + 0.9269 QWFA206
   + 0.9496 QWFA207 + 0.9697 QWFA208 + 0.7129 QWFA305
   + 0.8599 QWFA306 + 0.8902 QWFA307 + 0.9258 QWFA308
   + 0.4792 QWFA405 + 0.6765 QWFA406 + 0.7133 QWFA407
   + 0.7723 QWFA408 <= 58.88049
```

```
    ITPB101)  10.95 DTPB101 - QTPB101 >=  0
    ITPB201)  10.95 DTPB101 + 10.95 DTPB201 - QTPB201 >=  0
    ITPB301)  10.95 DTPB101 + 10.95 DTPB201 + 10.95 DTPB301 - QTPB301 >=  0
    ITPB401)  10.95 DTPB101 + 10.95 DTPB201 + 10.95 DTPB301 + 10.95 DTPB401 -
      QTPB401 >=  0
    LTPB01)  DTPB101 + DTPB201 + DTPB301 + DTPB401 <=  1
    DTWB1) - QTIAB101 + QTPB101 >=  0
    DTWB2) - QTIAB201 + QTPB201 >=  0
    DTWB3) - QTIAB301 + QTPB301 >=  0
    DTWB4) - QTIAB401 + QTPB401 >=  0
    DTOB1) - QTIAB101 + QRSB101 >=  0
    DTOB2) - QTIAB201 + QRSB201 >=  0
    DTOB3) - QTIAB301 + QRSB301 >=  0
    DTOB4) - QTIAB401 + QRSB401 >=  0
END
INTEGER-VARIABLES=    32
```

4. BOUNDS ON VARIABLES

SUB	QRSA101	3.65
SUB	QRSA102	9.18
SUB	QRSA103	2.56
SUB	QRSA201	3.65
SUB	QRSA202	9.18
SUB	QRSA203	2.56
SUB	QRSA301	3.65
SUB	QRSA302	9.18
SUB	QRSA303	2.56
SUB	QRSA401	3.65
SUB	QRSA402	9.18
SUB	QRSA403	2.56
SUB	QWFA101	1.37
SUB	QWFA102	1.37
SUB	QWFA103	1.37
SUB	QWFA104	1.37
SUB	QWFA105	.91
SUB	QWFA106	.91
SUB	QWFA107	.91
SUB	QWFA108	.91
SUB	QWFA201	1.37
SUB	QWFA202	1.37
SUB	QWFA203	1.37
SUB	QWFA204	1.37
SUB	QWFA205	.91
SUB	QWFA206	.91
SUB	QWFA207	.91
SUB	QWFA208	.91
SUB	QWFA301	1.37
SUB	QWFA302	1.37
SUB	QWFA303	1.37
SUB	QWFA304	1.37
SUB	QWFA305	.91
SUB	QWFA306	.91

SUB	QWFA307	.91
SUB	QWFA308	.91
SUB	QWFA401	1.37
SUB	QWFA402	1.37
SUB	QWFA403	1.37
SUB	QWFA404	1.37
SUB	QWFA405	.91
SUB	QWFA406	.91
SUB	QWFA407	.91
SUB	QWFA408	.91
SUB	QTPA101	20.44
SUB	QTPA201	20.44
SUB	QTPA301	20.44
SUB	QTPA401	20.44
SUB	QTIAB101	10.95
SUB	QTIAB201	10.95
SUB	QTIAB301	10.95
SUB	QTIAB401	10.95
SUB	QRSB101	10.95
SUB	QRSB201	10.95
SUB	QRSB301	10.95
SUB	QRSB401	10.95
SUB	QTPB101	10.95
SUB	QTPB201	10.95
SUB	QTPB301	10.95
SUB	QTPB401	10.95

5. OPTIMUM SOLUTION FOUND BY PROG_LINDO

OBJECTIVE FUNCTION VALUE

1) 62.9831085

A. COLUMN SECTION:

VARIABLE	VALUE	REDUCED COST
DRSA102	.000000	113.269470
DRSA103	.000000	34.599960
DRSA202	.000000	.000000
DRSA203	.000000	.000000
DRSA302	.000000	.000000
DRSA303	1.000000	30.129562
DRSA402	.000000	40.816986
DRSA403	.000000	12.540660
DWFA105	.000000	4.169486
DWFA106	.000000	3.739534
DWFA107	.000000	3.739534
DWFA108	1.000000	3.739534
DWFA205	1.000000	2.667360
DWFA206	.000000	2.237409
DWFA207	1.000000	2.237409
DWFA208	.000000	2.237409
DWFA305	.000000	1.596365

DWFA306	.000000	1.166413
DWFA307	.000000	1.166413
DWFA308	.000000	1.166413
DWFA405	.000000	.485486
DWFA406	1.000000	.485486
DWFA407	.000000	.485486
DWFA408	.000000	.485486
DTIAB101	.000000	55.405533
DTIAB201	.000000	.000000
DTIAB301	.000000	27.173203
DTIAB401	.000000	.000000
DTPB101	.000000	20.031555
DTPB201	.000000	.000000
DTPB301	.000000	17.443436
DTPB401	.000000	7.260380
QRSA101	3.474871	.000000
QRSA102	.000000	.106950
QRSA103	.000000	.178250
QRSA201	3.032452	.000000
QRSA202	.000000	8.873835
QRSA203	.000000	21.574799
QRSA301	3.650000	-.090610
QRSA302	.000000	10.682735
QRSA303	1.078412	.000000
QRSA401	3.650000	-.064610
QRSA402	.000000	.000000
QRSA403	1.935612	.000000
QWFA101	1.368750	-.819940
QWFA102	1.368750	-.819940
QWFA103	1.368750	-.819940
QWFA104	1.368750	-.819940
QWFA105	.000000	.000000
QWFA106	.000000	.000000
QWFA107	.000000	.000000
QWFA108	.912500	.000000
QWFA201	1.368750	-.584600
QWFA202	1.368750	-.584600
QWFA203	1.368750	-.584600
QWFA204	1.368750	-.584600
QWFA205	.912500	.000000
QWFA206	.000000	.000000
QWFA207	.912500	.000000
QWFA208	.912500	.000000
QWFA301	1.368750	-.507430
QWFA302	1.368750	-.507430
QWFA303	1.368750	-.507430
QWFA304	1.368750	-.507430
QWFA305	.912500	-.471180
QWFA306	.000000	.000000
QWFA307	.912500	.000000
QWFA308	.912500	.000000
QWFA401	1.368750	-.361790
QWFA402	1.368750	-.361790
QWFA403	1.368750	-.361790

QWFA404	1.368750	-.361790
QWFA405	.912500	.000000
QWFA406	.912500	.000000
QWFA407	.912500	.000000
QWFA408	.912500	.000000
QTPA101	3.474871	.000000
QTPA201	3.032452	.000000
QTPA301	4.728412	.000000
QTPA401	5.585611	.000000
QTIAB101	.000000	.000000
QTIAB201	.000000	10.638645
QTIAB301	.000000	.000000
QTIAB401	.000000	2.609192
QRSB101	.000000	.819940
QRSB201	.000000	.000000
QRSB301	.000000	.416820
QRSB401	.000000	.000000
QTPB101	.000000	1.497270
QTPB201	.000000	.000000
QTPB301	.000000	.761140
QTPB401	.000000	.000000

B. ROW SECTION:

ROW	SLACK OR SURPLUS	DUAL PRICES
IRSA102)	.000000	.000000
IRSA103)	.000000	.000000
IRSA202)	.000000	-8.797585
IRSA203)	.000000	-21.447708
IRSA302)	.000000	-10.718976
IRSA303)	1.476588	.000000
IRSA402)	.000000	-.025840
IRSA403)	.619388	.000000
IWFA105)	.000000	-.748640
IWFA106)	.000000	-.748640
IWFA107)	.000000	-.748640
IWFA108)	.000000	-.748640
IWFA205)	.000000	-.533770
IWFA206)	.000000	-.533770
IWFA207)	.000000	-.533770
IWFA208)	.000000	-.533770
IWFA305)	.000000	.000000
IWFA306)	.000000	-.471180
IWFA307)	.000000	-.471180
IWFA308)	.000000	-.471180
IWFA405)	.000000	-.335950
IWFA406)	.000000	-.335950
IWFA407)	.000000	-.335950
IWFA408)	.000000	-.335950
ITIAB101)	.000000	-.335100
ITIAB201)	.000000	-6.328114
ITIAB301)	.000000	-.260960

ITIAB401)	.000000	-1.955392
LRSA02)	1.000000	.000000
LRSA03)	.000000	.000000
LWFA05)	.000000	.000000
LWFA06)	.000000	.000000
LWFA07)	.000000	.000000
LWFA08)	.000000	.000000
LTIAB01)	1.000000	.000000
DTWA1)	.000000	-.712990
DTWA2)	.000000	-.508350
DTWA3)	.000000	-.362450
DTWA4)	.000000	-.258420
DTOA1)	.000000	-.463440
DTOA2)	.000000	-.330430
DTOA3)	.000000	-.326200
DTOA4)	.000000	-.232580
SFLA11)	28.268906	.000000
SFLA12)	58.175766	.000000
SFLA21)	24.396027	.000000
SFLA22)	56.242813	.000000
SFLA31)	19.831116	.000000
SFLA32)	53.895142	.000000
SFLA41)	14.886364	.000000
SFLA42)	50.756042	.000000
ITPB101)	.000000	.000000
ITPB201)	.000000	-2.897323
ITPB301)	.000000	.000000
ITPB401)	.000000	.000000
LTPB01)	1.000000	.000000
DTWB1)	.000000	.000000
DTWB2)	.000000	-3.964863
DTWB3)	.000000	.000000
DTWB4)	.000000	-.542680
DTOB1)	.000000	.000000
DTOB2)	.000000	-.584600
DTOB3)	.000000	.000000
DTOB4)	.000000	-.297180

Appendix H

List of Program PROG_BETA

```
C
**************************************************************************
C       THIS PROGRAM CALCULATES STREAM-AQUIFER INTERACTION
COEFFICIENTS.
C
**************************************************************************
C       ADIP-ITERATIVE

C       BETA CALCULATIONS

C       AQUIFER MODEL

C       CONFINED AQUIFER OR UNCONFINED AQUIFER WITH CONSTANT

C       TRANSMISSIVITY IN TIME

C

C       THIS PROGRAM USES AN IMPLICIT ALTERNATING DIRECTION SCHEME.

C

        COMMON /C4/ DT
        COMMON /C8/ KT
        COMMON /C10/ NUMKT,NWEL,NOBSW
        COMMON /C11/ IBOUND,IWRITE,IDISK,ISTDY,ILEAK,IMASS
        COMMON /C23/ IDIVG,KOUNT
        COMMON /C24/ LENG
        COMMON /VEL/ IVEL,IVDSK,IBETA,IRAD
C
        CALL INF
        STRT=DT
        CALL PARAM
        CALL PRINT
        DO 30 KWEL=1,NWEL
        KW=KWEL
        CALL INF2(KW)
        WRITE(17,'(I5,$)')KWEL
        DO 20 KTIME=1,NUMKT
        KT=KTIME
        PRINT*,'Calculating BETA for WELL #',KWEL,', TIME PERIOD ',KT
        IF(KT.EQ.1) DT=STRT
 10     CALL ITRATE
        CALL ROW
        CALL COLUMN
        IF(KOUNT.LE.10*LENG.AND.IDIVG.EQ.1) GO TO 10
        IF(KOUNT.GT.10*LENG) WRITE(16,1000)
        IF(KOUNT.GT.10*LENG) GO TO 30
        IF(IDIVG.EQ.0) CALL BETA(KW)
        IF(ISTDY.EQ.1) DT=DT*10
 20     CONTINUE
        WRITE(17,'(A)')' '
 30     CONTINUE
        WRITE(*,*)' '
```

```
      WRITE(*,*)'BETA values are saved in file "BETA_VALUES" '
C
 1000 FORMAT(' THE SOLUTION FAILED TO CONVERGE')
      END
C
C
*************************************************************************
C
      SUBROUTINE INF
C
      COMMON /C1/ IDIM,JDIM,ILESS1,JLESS1
      COMMON /C2/ T(100,100),ZK(100,100)
      COMMON /C3/ G(100,100)
      COMMON /C4/ DT
      COMMON /C5/ P(100,100)
      COMMON /C6/ PI
      COMMON /C8/ KT
      COMMON /C9/ S(100,100)
      COMMON /C10/ NUMKT,NWEL,NOBSW
      COMMON /C11/ IBOUND,IWRITE,IDISK,ISTDY,ILEAK,IMASS
      COMMON /C13/ DX,DELX(100),DELY(100)
      COMMON /C14/ II(200),JJ(200),IO(200),JO(200)
      COMMON /C18/ SCALET,SCALES,SCALEH,SCALEM
      COMMON /C20/ HC(100,100),IH(100,100),POR
      COMMON /C21/ EPS,PARM,PG(100,100)
      COMMON /C22/ OMEGA(100)
      COMMON /C23/ IDIVG,KOUNT
      COMMON /C24/ LENG
      COMMON /C25/ BDTHK(100,100)
      COMMON /C50/ RAD(100,100),THETA(100,100)
      COMMON /VEL/ IVEL,IVDSK,IBETA,IRAD
C
      INTEGER P,S
C
      INTEGER*2 ST,DAY,YEAR
      CHARACTER*1 KEY
      CHARACTER*3 CHAR
      CHARACTER*64 NAMEP,DBETA,VBETA,NAMEF,NAMEO
      CHARACTER*40 FILEIN,FILEOUT
      DIMENSION BD(550),ZHC(550),IICH(550),JJCH(550),IICL(550),JJCL(550)
      DIMENSION ST(100,100),CHAR(2)
C
      DATA CHAR /'NO ','YES'/
C
C     SET UP TITLE PAGE ** PAGE 1
C
C
C     OPEN INPUT DATAFILE AND ASSIGN UNIT #15
C
      WRITE(*,*)' Enter the INPUT        file name : '
      READ(*,'(A40)')FILEIN
      OPEN(15,FILE=FILEIN,IOINTENT='INPUT',PAD='YES')
C
C     OPEN FILE FOR OUTPUT AND ASSIGN UNIT #16
```

```
C
      WRITE(*,*)' Enter the OUTPUT file name : '
      READ(*,'(A40)')FILEOUT
      OPEN(16,FILE=FILEOUT,IOINTENT='OUTPUT',PAD='YES')
C
C     FILE# 17 OPENED BY NADEEM
C     ************************
C
      OPEN(17,FILE='BETA_VALUES',IOINTENT='OUTPUT',PAD='YES')
C
      READ(15,1000)IWRITE,IDISK,IVDSK,IMASS,IVEL,IBETA,IRAD
      READ(15,1000) ISTDY,IBOUND,ILEAK
 1000 FORMAT(10I1)
C
      IF(IDISK.NE.0) THEN
 1010 FORMAT(A)
      OPEN(29,FILE='CASE1_BETA4_BETAS',IOINTENT='OUTPUT')
      ENDIF
C
      IF(IVDSK.NE.0) THEN
      OPEN(31,FILE='CASE1_BETA4_BVS',IOINTENT='OUTPUT')
      ENDIF
C
C
C
C
      PRINT*,'*** READING DATA NOW ***'
C
C     READ NODAL DIMENSION AND METRIC
C
      READ(15,1110)IDIM,JDIM,DX
 1110 FORMAT(2I2,F10.2)
C
C     INITIALIZE CONSTANTS
C
      ILESS1=IDIM-1
      JLESS1=JDIM-1
      PI=3.14159265359
C
C     READ TIME DATA
C
      READ(15,1130) NUMKT,DT
 1130 FORMAT(I2,F20.2,I2)
C
C     READ ITERATION DATA
C
      READ(15,1150) LENG,EPS
 1150 FORMAT(I2,F20.2)
C
C     READ POROSITY
C
      READ(15,1170) POR
 1170 FORMAT(F10.2)
C
```

```
C       GRID SCALES IN X AND Y DIRECTIONS
C
        READ(15,1220) (DELX(J),J=1,JDIM)
        READ(15,1220) (DELY(I),I=1,IDIM)
 1220 FORMAT(15F5.2)
C
C       READ SCALE FACTOR FOR TRANSMISSIVITY
C
        READ(15,1180) SCALET
 1180 FORMAT(F20.5)
C
C       READ IN TRANSMISSIVITY DATA
C
        DO 10 I=1,IDIM
        READ(15,1261) (T(I,J),J=1,JDIM)
 1261 FORMAT(12F10.0)
 1260 FORMAT(15I5)
   10   CONTINUE
C
C       READ SCALE FACTOR SATURATED THICKNESS
C
        READ(15,1180) SCALEM
C
C       READ IN SATURATED THICKNESS DATA
C
        DO 30 I=1,IDIM
        READ(15,1260) (ST(I,J),J=1,JDIM)
        30 CONTINUE
C
C       CALCULATE HYDRAULIC CONDUCTIVITY
C
        DO 50 I=1,IDIM
        DO 50 J=1,JDIM
        HC(I,J)=0.0
        IF(ST(I,J).NE.0) THEN
        HC(I,J)=( (T(I,J))/SCALET)/(FLOAT(ST(I,J))/SCALEM)
        ENDIF
        50 CONTINUE
C
C       READ SCALE FACTOR FOR INITIAL HEAD
C
        READ(15,1180) SCALEH
C
C       READ IN INITIAL HEAD DATA
C
        DO 60 I=1,IDIM
        READ(15,1260) (IH(I,J),J=1,JDIM)
        60 CONTINUE
C
C       READ SCALE FACTOR FOR STORATIVITY
C
        READ(15,1180) SCALES
C
C       READ IN STORAGE COEFFICIENT DATA
```

```
C
      DO 80 I=1,IDIM
      READ(15,1260) (S(I,J),J=1,JDIM)
      80 CONTINUE
C
C     READ IN CONFINING LAYER (IF ANY)
C
      DO 110 I=1,IDIM
      DO 110 J=1,JDIM
      BDTHK(I,J)=1.0
      110 ZK(I,J)=0.0
      IF(ILEAK.NE.0) THEN
      READ(15,1351) NCLN
 1351 FORMAT(I3)
      READ(15,1350) (IICL(K),JJCL(K),K=1,NCLN)
      READ(15,1361) (ZHC(K),K=1,NCLN)
 1361 FORMAT(8E16.5)
      READ(15,1360) (BD(K),K=1,NCLN)
 1350 FORMAT(20I2)
 1360 FORMAT(8F8.5)
      DO 120 K=1,NCLN
      I=IICL(K)
      J=JJCL(K)
      BDTHK(I,J)=BD(K)
      ZK(I,J)=ZHC(K)
      120    CONTINUE
      ENDIF
C
C     INITIALIZE PUMPING MATRIX
C
      DO 100 I=1,IDIM
      DO 100 J=1,JDIM
      100 P(I,J)=0
C
C     READ IN CONSTANT HEAD POINTS IF ANY
C
      IF(IBOUND.NE.0) THEN
      READ(15,1351) NCHN
      READ(15,1350) (IICH(K),JJCH(K),K=1,NCHN)
      DO 130 K=1,NCHN
      I=IICH(K)
      J=JJCH(K)
      P(I,J)=2
      130    CONTINUE
      ENDIF
C
C     READ IN OBSERVATION WELL POINTS
C
C     NOBSW INCLUDES SUPPLY,INTRCEPTION ANS VEL OBSEVATION WELLS
      READ(15,1350) NOBSW
      READ(15,1350) (IO(K),JO(K),K=1,NOBSW)
C
C     READ IN PUMPING WELL POINTS
C
```

```
      READ(15,1350) NWEL
 1355 FORMAT(3I4)
      READ(15,1350)(II(K),JJ(K),K=1,NWEL)
C
C     PRINT OUT INPUT DATA
C
C     PRINT OUT TITLE AND SURPRESSION CHOICES
C
      PRINT*,'BETA CALCULATIONS IN PROGRESS NOW '
      WRITE(16,1020) NAMEP
 1020 FORMAT(1H1,/' GROUNDWATER RESPONSE FUNCTION
      MODEL',/,'PROBLEM:',A64)
      WRITE(16,1030) CHAR(IWRITE+1)
 1030 FORMAT(' PRINT OUT TRANSMISSIVITY, SATURATED THICKNESS, ',/,
      14X,'INITIAL HEAD AND STORATIVITY MAPS?',T60,A3)
      WRITE(16,1040) CHAR(IDISK+1)
 1040 FORMAT(' WRITE OUT DRAWDOWN RESPONSE FUNCTIONS ON DISK ?',T60,
      2A3)
      WRITE(16,1050) CHAR(IVDSK+1)
 1050 FORMAT(' WRITE OUT VELOCITY RESPONSE FUNCTIONS ON DISK ?',T60,
      1A3)
      WRITE(16,1060) CHAR(IMASS+1)
 1060 FORMAT(' CALCULATE MASS BALANCE ?',T60,A3)
      WRITE(16,1070) CHAR(IVEL+1)
 1070 FORMAT(' WRITE OUT AREAL VELOCITY FIELDS ?',T60,A3)
      WRITE(16,1080) CHAR(IBETA+1)
 1080 FORMAT(' WRITE OUT AREAL DRAWDOWNS ?',T60,A3)
      WRITE(16,1090) CHAR(IRAD+1)
 1090 FORMAT(' PERFORM WELL RADIUS CORRECTIONS ?',T60,A3)
C
C     PRINT OUT NODAL DIMENSION AND METRIC
C
      WRITE(16,1120) IDIM,JDIM,DX
 1120 FORMAT(1H1,'GRID INFORMATION'//5X,
      1' NUMBER OF GRID ROWS',T30,'=',I3/5X,' NUMBER OF GRID',
      2' COLUMNS',T30,'=',I3/5X,' DISTANCE METRIC',T30,'=',F6.0,' FT')
C
C     PRINT OUT GRID SCALES
C
      WRITE(16,1230)
 1230 FORMAT(/5X,' GRID SCALE IN X DIRECTION'/)
      WRITE(16,1240) (DELX(J),J=1,JDIM)
 1240 FORMAT(2X,10F5.1)
      WRITE(16,1250)
 1250 FORMAT(/5X,' GRID SCALE IN Y DIRECTION'/)
      WRITE(16,1240) (DELY(I),I=1,IDIM)
C
C     PRINT OUT TIME DATA
C
      WRITE(16,1140)NUMKT,DT
 1140 FORMAT(/' TIME INFORMATION'//
      15X,' TOTAL NUMBER OF TIME PERIODS',T60,'=',I3/
      25X,' TIME PERIOD LENGTH',T60,'=',E12.5,' SEC')
C
```

```
C       PRINT OUT INPUT ITERATION PARAMETERS
C
        WRITE(16,1160) LENG,EPS
 1160   FORMAT(/' ITERATION INFORMATION'//5X,
        1' NUMBER OF PARAMETERS',T30,'=',I3/5X,
        2' CONVERGENCE EPSILON',T30,'=',F7.5/5X,
        3' ITERATION PARMETERS:')
C
C       PRINT OUT AVAILABLE CAPTURE SOURCES
C
        IF(IBOUND.EQ.1) THEN
        WRITE(16,1200)
 1200   FORMAT(' CONSTANT HEAD BOUNDARY')
        ENDIF
        IF(ILEAK.EQ.1) THEN
        WRITE(16,1210)
 1210   FORMAT(' CONFINING LAYER WITH LEAKAGE'//)
        ENDIF
C
C       PRINT OUT TRANSMISSIVITY
C
        IF(IWRITE.NE.0) THEN
        WRITE(16,1270)
 1270   FORMAT(1H1,30X,'TRANSMISSIVITY (FT*FT/SEC)'/)
        DO 20 I=1,IDIM
        WRITE(16,1261) (T(I,J),J=1,JDIM)
 1280   FORMAT(30I4)
        20      CONTINUE
        WRITE(16,1290) SCALET
 1290   FORMAT(/20X,' ACTUAL TRANSMISSIVITY = T(I,J)/(',E9.3,')')
        ENDIF
C
C       PRINT OUT SATURATED THICKNESS
C
        IF(IWRITE.NE.0) THEN
        WRITE(16,1300)
 1300   FORMAT(1H1,30X,'SATURATED THICKNESS (FT)'/)
        DO 40 I=1,IDIM
        WRITE(16,1280) (ST(I,J),J=1,JDIM)
        40      CONTINUE
        WRITE(16,1310) SCALEM
 1310   FORMAT(/20X,' ACTUAL SATURATED THICKNESS = ST(I,J)/(',E9.3,')')
        ENDIF
C
C       PRINT OUT INITIAL HEAD
C
        IF(IWRITE.NE.0) THEN
        WRITE(16,1315)
 1315   FORMAT(1H1,30X,'INITIAL HEAD (FT)'/)
        DO 70 I=1,IDIM
        WRITE(16,1282) (IH(I,J),J=1,JDIM)
 1282   FORMAT(20I5)
        70      CONTINUE
        WRITE(16,1320) SCALEH
```

```
 1320  FORMAT(/20X,' ACTUAL INITIAL HEAD = IH(I,J)/(',E9.3,')')
       ENDIF
C
C      PRINT OUT STORAGE COEFFICIENT
C
       IF(IWRITE.NE.0) THEN
       WRITE(16,1330)
 1330  FORMAT(1H1,30X,'STORAGE COEFFICIENT'/)
       DO 90 I=1,IDIM
       WRITE(16,1280) (S(I,J),J=1,JDIM)
       90     CONTINUE
       WRITE(16,1340) SCALES,POR
 1340  FORMAT(/20X,' ACTUAL STORAGE COEFFICIENT = SC(I,J)/(',E9.3,')'/
       1      21X,'POROSITY = ',E9.3/)
       ENDIF
C
C      LEAKAGE SIGNAL
C
       IF(ILEAK.NE.0) THEN
       WRITE(16,1370)
 1370  FORMAT(' CONFINING LAYER DATA READ IN')
       ENDIF
C
C      CONSTANT HEAD BOUNDARY SIGNAL
C
       IF(IBOUND.NE.0) THEN
       WRITE(16,1380)
 1380  FORMAT(' ***CONSTANT HEAD BOUNDARY POINTS READ IN***')
       ENDIF
C
C
C      CALCULATE INITIAL VELOCITY FIELD
C
       KT=0
       CALL INTVEL
C
       CLOSE (15)
C
       RETURN
       END
C
C
C**********************************************************************
C
C*************************************************

C
C      ***************************************************************
C
       SUBROUTINE PARAM
C
C      THIS SUBROUTINE COMPUTES ITERATION PARAMETERS
C
       COMMON /C1/ IDIM,JDIM,ILESS1,JLESS1
```

```
      COMMON /C6/ PI
      COMMON /C13/ DX,DELX(100),DELY(100)
      COMMON /C22/ OMEGA(100)
      COMMON /C24/ LENG
C
      OMAX=1.0
      OMIN=2.0
      ID2=IDIM-2
      JD2=JDIM-2
      XVAL=PI**2/(2.0*JD2**2)
      YVAL=PI**2/(2.0*ID2**2)
      DO 10 I=3,ID2
      DO 10 J=3,JD2
      XPART=XVAL*(1.0/(1.0+DELX(J)**2/DELY(I)**2))
      YPART=YVAL*(1.0/(1.0+DELY(I)**2/DELX(J)**2))
      OMIN=AMIN1(OMIN,XPART,YPART)
      10 CONTINUE
      BETA= EXP(LOG(OMAX/OMIN)/(FLOAT(LENG)-1.0))
      OMEGA(1)=OMIN
      DO 20 I=2,LENG
      20 OMEGA(I)=OMEGA(I-1)*BETA
      RETURN
C
      END
C
C     ****************************************************************
C
      SUBROUTINE INF2(KWEL)
C
      COMMON /C1/ IDIM,JDIM,ILESS1,JLESS1
      COMMON /C3/ G(100,100)
      COMMON /C5/ P(100,100)
      COMMON /C11/ IBOUND,IWRITE,IDISK,ISTDY,ILEAK,IMASS
      COMMON /C14/ II(200),JJ(200),IO(200),JO(200)
      COMMON /C23/ IDIVG,KOUNT
C
      INTEGER*4 P
C
C     INITIALIZE CONSTANTS
C
      IDIVG=0
C
C     SET INITIAL VALUE OF HEAD AND ZERO PUMPING
C
      DO 10 I=1,IDIM
      DO 9 J=1,JDIM
      IF(P(I,J).NE.2)  P(I,J)=0
      G(I,J)=0.0
      9 CONTINUE
      10 CONTINUE
C
C     SET UP PUMPING WELL
C
      I=II(KWEL)
```

```
      J=JJ(KWEL)
      P(I,J)=1
      RETURN
C
      END
C
C     ****************************************************************
C
      SUBROUTINE ITRATE
C
      COMMON /C1/ IDIM,JDIM,ILESS1,JLESS1
      COMMON /C3/ G(100,100)
      COMMON /C4/ DT
      COMMON /C10/ NUMKT,NWEL,NOBSW
      COMMON /C21/ EPS,PARM,PG(100,100)
      COMMON /C22/ OMEGA(100)
      COMMON /C23/ IDIVG,KOUNT
      COMMON /C24/ LENG
C
      IF(IDIVG.EQ.1) GO TO 20
      NTH=0
      KOUNT=0
      PARM=0.0
      DO 10 I=1,IDIM
      DO 9 J=1,JDIM
      PG(I,J)=G(I,J)
      9 CONTINUE
      10 CONTINUE
      RETURN
      20 KOUNT=KOUNT+1
      IF(MOD(KOUNT,LENG).GT.0) GO TO 30
      NTH=0
      30 NTH=NTH+1
      PARM=OMEGA(NTH)
      RETURN
C
      END
C
C     ****************************************************************
C
      SUBROUTINE ROW
C
      COMMON /C1/ IDIM,JDIM,ILESS1,JLESS1
      COMMON /C2/ T(100,100),ZK(100,100)
      COMMON /C3/ G(100,100)
      COMMON /C4/ DT
      COMMON /C5/ P(100,100)
      COMMON /C6/ PI
      COMMON /C8/ KT
      COMMON /C9/ S(100,100)
      COMMON /C11/ IBOUND,IWRITE,IDISK,ISTDY,ILEAK,IMASS
      COMMON /C13/ DX,DELX(100),DELY(100)
      COMMON /C18/ SCALET,SCALES,SCALEH,SCALEM
      COMMON /C21/ EPS,PARM,PG(100,100)
```

```
      COMMON /C23/ IDIVG,KOUNT
      COMMON /C25/ BDTHK(100,100)
C
      INTEGER P,S
      DIMENSION BE(550), G1(550), TEMP(550)
C
      JLESS1=JDIM-1
      JLESS2=JDIM-2
      ILESS1=IDIM-1
      ILESS2=IDIM-2
      RHO=DX**2/DT
      BE(1)=0.0
      G1(1)=0.0
      DO 70 I=2,IDIM
      DO 10 J=2,JLESS1
      BE(J)=0.0
      G1(J)=0.0
C
C     DETERMINE WHETHER NODE IS OUTSIDE AQUIFER BOUNDARY
C
      IF(T(I,J).EQ.0) GO TO 10
      IF(P(I,J).EQ.2) GO TO 10
C
C     CALCULATE AVERAGE VALUES OF T BETWEEN ADJACENT NODES.
C
C     NODE.       T1=LEFT,    T2=RIGHT,  T3=UPPER,  T4=LOWER
C
      T1=2.0*(T(I,J-1)*T(I,J))
      T1=T1/((T(I,J-1))*DELX(J)+(T(I,J))*DELX(J-1))
      T1=T1/(DELX(J)*SCALET)
      T2=2.0*(T(I,J+1)*T(I,J))
      T2=T2/((T(I,J+1))*DELX(J)+(T(I,J))*DELX(J+1))
      T2=T2/(DELX(J)*SCALET)
      T3=2.0*(T(I-1,J)*T(I,J))
      T3=T3/((T(I-1,J))*DELY(I)+(T(I,J))*DELY(I-1))
      T3=T3/(DELY(I)*SCALET)
      T4=2.0*(T(I+1,J)*T(I,J))
      T4=T4/((T(I+1,J))*DELY(I)+(T(I,J))*DELY(I+1))
      T4=T4/(DELY(I)*SCALET)
      GINCR=PARM*(T1+T2+T3+T4)
C
C     DEFINE BOUNDARY REQUIREMENTS OF CALCULATION PARAMETERS
C     A,B,C,Q, AND BE FOR ROWS OF MATRIX
C
C     CALCULATE VALUES      FOR PARAMETERS A,B,C,D
C
      Q=0.0
      IF(P(I,J).EQ.1.AND.KT.EQ.1) Q=1.0
      IF(P(I,J).EQ.1.AND.ISTDY.EQ.1) Q=1.0
      BDT=BDTHK(I,J)
      AREA=DX**2
      SC=FLOAT(S(I,J))/SCALES
      A=T1
      B=-T1-T2-RHO*SC-GINCR-ZK(I,J)*AREA/BDT/2.0
```

```
      C=T2
      D=-T3*G(I-1,J)-RHO*SC*PG(I,J)+(T3+T4-GINCR)*G(I,J)
      1-T4*G(I+1,J)-Q/(DELX(J)*DELY(I))+ZK(I,J)*AREA/BDT*G(I,J)/2.0
      W=B-A*BE(J-1)
      BE(J)=C/W
      G1(J)=(D-A*G1(J-1))/W
      10 CONTINUE
C
C     CALCULATE HEAD VALUES FOR ROWS OF MATRIX AND PLACE THEM IN
C     TEMPORARY LOCATION TEMP
C
      IF(I.GT.2) GO TO 40
      DO 30 K=1,JLESS2
      N=JDIM-K
      IF(T(I,N).NE.0.AND.P(I,N).NE.2) GO TO 20
      TEMP(N)=0.0
      GO TO 30
      20 TEMP(N)=G1(N)-BE(N)*TEMP(N+1)
      30 CONTINUE
      GO TO 70
      40 DO 60 K=1,JLESS2
      N=JDIM-K
      G(I-1,N)=TEMP(N)
      IF(T(I,N).NE.0.AND.P(I,N).NE.2) GO TO 50
      TEMP(N)=0.0
      GO TO 60
      50 TEMP(N)=G1(N)-BE(N)*TEMP(N+1)
      60 CONTINUE
      70 CONTINUE
      RETURN
C
      END
C
C     ****************************************************************
C
      SUBROUTINE COLUMN
C
      COMMON /C1/ IDIM,JDIM,ILESS1,JLESS1
      COMMON /C2/ T(100,100),ZK(100,100)
      COMMON /C3/ G(100,100)
      COMMON /C4/ DT
      COMMON /C5/ P(100,100)
      COMMON /C6/ PI
      COMMON /C8/ KT
      COMMON /C9/ S(100,100)
      COMMON /C11/ IBOUND,IWRITE,IDISK,ISTDY,ILEAK,IMASS
      COMMON /C13/ DX,DELX(100),DELY(100)
      COMMON /C18/ SCALET,SCALES,SCALEH,SCALEM
      COMMON /C21/ EPS,PARM,PG(100,100)
      COMMON /C23/ IDIVG,KOUNT
      COMMON /C25/ BDTHK(100,100)
C
      INTEGER P,S
      DIMENSION BE(550), G1(550), TEMP(550)
```

```
      JLESS1=JDIM-1
      JLESS2=JDIM-2
      ILESS1=IDIM-1
      ILESS2=IDIM-2
      RHO=DX**2/DT
      BE(1)=0.0
      G1(1)=0.0
C
      IDIVG=0
      DO 70 J=2,JDIM
      DO 10 I=2,ILESS1
      BE(I)=0.0
      G1(I)=0.0
C
C     DETERMINE WHETHER NODE IS OUTSIDE AQUIFER BOUNDARY
C
      IF(T(I,J).EQ.0) GO TO 10
      IF(P(I,J).EQ.2) GO TO 10
C
C     DEFINE BOUNDARY REQUIREMENTS OF CALCULATION PARAMETERS
A,B,C,BE
C     FOR COLUMN OF MATRIX
C
C     CALCULATE AVERAGE VALUE OF T BETWEEN ADJACENT NODES
C
      T1=2.0*(T(I,J-1)*T(I,J))
      T1=T1/((T(I,J-1))*DELX(J)+(T(I,J))*DELX(J-1))
      T1=T1/(DELX(J)*SCALET)
      T2=2.0*(T(I,J+1)*T(I,J))
      T2=T2/((T(I,J+1))*DELX(J)+(T(I,J))*DELX(J+1))
      T2=T2/(DELX(J)*SCALET)
      T3=2.0*(T(I-1,J)*T(I,J))
      T3=T3/((T(I-1,J))*DELY(I)+(T(I,J))*DELY(I-1))
      T3=T3/(DELY(I)*SCALET)
      T4=2.0*(T(I+1,J)*T(I,J))
      T4=T4/((T(I+1,J))*DELY(I)+(T(I,J))*DELY(I+1))
      T4=T4/(DELY(I)*SCALET)
      GINCR=PARM*(T1+T2+T3+T4)
C
C     CALCULATE VALUES FOR PARAMETERS A,B,C,D
C
      BDT=BDTHK(I,J)
      AREA=DX**2
      SC=FLOAT(S(I,J))/SCALES
      Q=0.0
      IF(P(I,J).EQ.1.AND.KT.EQ.1) Q=1.0
      IF(P(I,J).EQ.1.AND.ISTDY.EQ.1) Q=1.0
      A=T3
      B=-T3-T4-RHO*SC-GINCR-ZK(I,J)*AREA/BDT/2.0
      C=T4
      IF(ABS(G(I,J)).LE.1.0E-10) G(I,J)=0.
      D=-T1*G(I,J-1)-RHO*SC*PG(I,J)+(T1+T2-GINCR)*G(I,J)
      2-T2*G(I,J+1)-Q/(DELX(J)*DELY(I))+ZK(I,J)/BDT*AREA*G(I,J)/2.0
      W=B-A*BE(I-1)
```

```
      BE(I)=C/W
      G1(I)=(D-A*G1(I-1))/W
      10 CONTINUE
C
C     CALCULATE HEAD VALUES FOR COLUMNS OF MATRIX AND PLACE IN
TEMPORARY
C     LOCATION TEMP
C
      IF(J.GT.2) GO TO 40
      DO 30 K=1,ILESS2
      N=IDIM-K
      IF(T(N,J).NE.0.AND.P(N,J).NE.2) GO TO 20
      TEMP(N)=0.0
      GO TO 30
      20 TEMP(N)=G1(N)-BE(N)*TEMP(N+1)
      IF(ABS(TEMP(N)-G(N,J)).GT.EPS) IDIVG=1
      30 CONTINUE
      GO TO 70
      40 DO 60 K=1,ILESS2
      N=IDIM-K
      G(N,J-1)=TEMP(N)
      IF(T(N,J).NE.0.AND.P(N,J).NE.2) GO TO 50
      TEMP(N)=0.0
      GO TO 60
      50 TEMP(N)=G1(N)-BE(N)*TEMP(N+1)
      IF(ABS(TEMP(N)-G(N,J)).GT.EPS) IDIVG=1
      60 CONTINUE
      70 CONTINUE
      RETURN
C
      END
C
C     ******************************************************************
C
      SUBROUTINE BETA(KWEL)
C
C     THIS SUBROUTINE CALCULATES BETAS AND A MASS BALANCE.
C
      COMMON /C1/ IDIM,JDIM,ILESS1,JLESS1
      COMMON /C2/ T(100,100),ZK(100,100)
      COMMON /C3/ G(100,100)
      COMMON /C4/ DT
      COMMON /C6/ PI
      COMMON /C8/ KT
      COMMON /C9/ S(100,100)
      COMMON /C10/ NUMKT,NWEL,NOBSW
      COMMON /C11/ IBOUND,IWRITE,IDISK,ISTDY,ILEAK,IMASS
      COMMON /C13/ DX,DELX(100),DELY(100)
      COMMON /C14/ II(200),JJ(200),IO(200),JO(200)
      COMMON /C18/ SCALET,SCALES,SCALEH,SCALEM
      COMMON /C20/ HC(100,100),IH(100,100),POR
      COMMON /VEL/ IVEL,IVDSK,IBETA,IRAD
C
      INTEGER*4 T,S
```

```
      DIMENSION B(200,50),BVX(200,50),BVY(200,50),R(200),THETA(200)
C
      IF(IWRITE.EQ.0.AND.IMASS.EQ.0) GO TO 20
      IF(ISTDY.EQ.1.AND.KT.NE.NUMKT) GO TO 20
      WRITE(16,1000) KWEL,II(KWEL),JJ(KWEL),KT
 1000 FORMAT(1H1,' PUMPING WELL " ',I2/' NODE COORDINATE (',I2,',',
     1I2,')'/' TIME LAG ',I2)
C
C     MASS BALANCE
C
      IF(IMASS.EQ.0) GO TO 20
      SUM=0.0
      DO 10 I=1,IDIM
      DO 10 J=1,JDIM
      IF(T(I,J).EQ.0) GO TO 10
      AREA=DELX(J)*DELY(I)*DX**2
      RHO=AREA/DT
      SUM=SUM+G(I,J)*FLOAT(S(I,J))/SCALES*RHO
      10 CONTINUE
      DIFF=(1.0-SUM)*100.0
      WRITE(16,1010) SUM,DIFF
 1010 FORMAT(' PUMPING RATE ESTIMATED FROM STORAGE LOSS = ',
     1F10.7,1X,'FT**3/SEC'/' ACTUAL PUMPING RATE = 1.0000000 FT**3'
     2,'/SEC'/' PERCENT DIFFERENCE = ',F11.7,' PERCENT')
C**** THE FOLLOWING LINE ADDED BY NADEEM *******
      DIFFNEW=1.0-SUM
      WRITE(17,3500)DIFFNEW
 3500 FORMAT(F6.4,$)
C**** ******************************************
      20 CONTINUE
C
C     CALCULATE DRAWDOWN BETAS
C
      DO 30 KOUNT=1,NOBSW
      I=IO(KOUNT)
      J=JO(KOUNT)
      30 B(KOUNT,KT)=G(I,J)
C
C     CALCULATE THE VELOCITY BETAS
C
      DO 40 KOUNT=1,NOBSW
      I=IO(KOUNT)
      J=JO(KOUNT)
      BVX(KOUNT,KT)=-HC(I,J)
      BVX(KOUNT,KT)=BVX(KOUNT,KT)*GRAD(G(I,J-1),G(I,J+1),DX)/POR/DELX(J)
      IF(ABS(BVX(KOUNT,KT)).LT.0.1E-10) BVX(KOUNT,KT)=0.0
      IF(I.EQ.II(KWEL).AND.J.EQ.JJ(KWEL)) BVX(KOUNT,KT)=0.0
      BVY(KOUNT,KT)=-HC(I,J)
      BVY(KOUNT,KT)=BVY(KOUNT,KT)*GRAD(G(I-1,J),G(I+1,J),DX)/POR/DELY(I)
      IF(I.EQ.II(KWEL).AND.J.EQ.JJ(KWEL)) BVY(KOUNT,KT)=0.0
      IF(ABS(BVY(KOUNT,KT)).LT.0.1E-10) BVY(KOUNT,KT)=0.0
      40 CONTINUE
C
C     WRITE OUT RESPONSE FUNCTIONS
```

```
C
      IF(IWRITE.EQ.0) GO TO 60
      IF(ISTDY.EQ.1.AND.KT.NE.NUMKT) GO TO 60
      WRITE(16,1020)
 1020 FORMAT(/' OBSERVATION',5X,'NODE',11X,'DRAWDOWN',21X,'VELOCITY','
      1 RESPONSE'/'       WELL',8X,'COORDINATE',7X,'RESPONSE',26X,
      2'FUNCTIONS'/'     "',12X,'(I,J)',10X,'FUNCTIONS',9X,'(X)',9X,
      3'(Y)',10X,'(R)',8X,'(THETA)'/)
      DO 50 K=1,NOBSW
      R(K)=0.0
      THETA(K)=0.0
C     IF(BVX(K,KT).LT.1.0E-10) BVX(K,KT)=0.
C     IF(BVY(K,KT).LT.1.0E-10) BVY(K,KT)=0.0
      IF(BVX(K,KT).EQ.0.0.AND.BVY(K,KT).EQ.0.0) GO TO 50
      R(K)=SQRT(BVX(K,KT)**2+BVY(K,KT)**2)
      THETA(K)=57.29577951*ATAN2(BVY(K,KT),BVX(K,KT))
      IF(THETA(K).LT.0.0) THETA(K)=THETA(K)+360.0
      50 WRITE(16,1030) K,IO(K),JO(K),B(K,KT),BVX(K,KT),BVY(K,KT)
      1,R(K),THETA(K)
 1030 FORMAT(2X,I3,11X,'(',I2,',',I2,')',6X,E11.5,5X,E11.5,2X,E11.5,
      12X,E11.5,2X,F7.1)
      60 CONTINUE
      IF(IBETA.EQ.0) GO TO 65
      WRITE(16,1050) KWEL,II(KWEL),JJ(KWEL),KT
 1050 FORMAT(1H1,10X,'RESPONSE FUNCTIONS FOR WELL "',I3,
      1' AT NODE (',I2,',',I2,'), LAGGED',I3,' TIME PERIODS'//)
      DO 62 I=1,IDIM
      62 WRITE(16,1051) (G(I,J),J=1,JDIM)
 1051 FORMAT(25F5.1)
      IF(IRAD.EQ.0.OR.KT.NE.1) GO TO 65
      I=II(KWEL)
      J=JJ(KWEL)
      RC=SQRT(DELX(J)*DELY(I)*DX**2/PI)
      GCOR=G(I,J)+LOG(RC)/(2.0*PI*(T(I,J))/SCALET)
      WRITE(16,1052) KWEL,GCOR
 1052 FORMAT(/10X,'RESPONSE FUNCTION FOR WELL "',I3,' CORRECTED',
      1' FOR WELL RADIUS = ',F10.5)
      65 CONTINUE
      IF(IVEL.EQ.0) GO TO 70
C     CALL VELFLD(KWEL)
      70 CONTINUE
      IF(IDISK.EQ.1.AND.ISTDY.EQ.0) WRITE(29,1040)(B(KOUNT,KT
      1),KOUNT=1,NWEL)
      IF(IDISK.EQ.1.AND.ISTDY.EQ.1.AND.KT.EQ.NUMKT) WRITE(29,1040)
      1(B(KOUNT,KT),KOUNT=1,NWEL)
      IF(ISTDY.EQ.0.AND.IVDSK.EQ.1) WRITE(31,1040) (BVX(KOUNT,KT)
      1,KOUNT=(NWEL+1),NOBSW)
      IF(ISTDY.EQ.1.AND.KT.EQ.NUMKT.AND.IVDSK.EQ.1) WRITE(31,1040)
      1 (BVX(KOUNT,KT),KOUNT=(NWEL+1),NOBSW)
      IF(ISTDY.EQ.0.AND.IVDSK.EQ.1) WRITE(31,1040) (BVY(KOUNT,KT)
      1,KOUNT=(NWEL+1),NOBSW)
      IF(ISTDY.EQ.1.AND.KT.EQ.NUMKT.AND.IVDSK.EQ.1) WRITE(31,1040)
      1 (BVY(KOUNT,KT),KOUNT=(NWEL+1),NOBSW)
 1040 FORMAT(5E15.5)
```

```
1090 FORMAT(6E15.5)
      RETURN
C
      END
C
C     ****************************************************************
C
      SUBROUTINE PRINT
C
      COMMON /C1/ IDIM,JDIM,ILESS1,JLESS1
      COMMON /C2/ T(100,100),ZK(100,100)
      COMMON /C5/ P(100,100)
      COMMON /C10/ NUMKT,NWEL,NOBSW
      COMMON /C13/ DX,DELX(100),DELY(100)
      COMMON /C14/ II(200),JJ(200),IO(200),JO(200)
      COMMON /C20/ HC(100,100),IH(100,100),POR
      COMMON /C18/ SCALET,SCALES,SCALEH,SCALEM
C
      INTEGER*4 T,P
      DIMENSION GRAPH(200),CHAR(108),BVXO0(200),BVYO0(200),BVXP0(200)
      1,BVYP0(200),R(200),THETA(200)
C
      DATA CHAR/2H      ,2HCH,2HPW,2H L,2H+T,2H+0,2H 1,2H 2,2H 3,2H 4,2H 5,2
      1H 6,2H 7,2H 8,2H 9,2H10,2H11,2H12,2H13,2H14,2H15,2H16,2H17,2H18,2H
      219,2H20,2H21,2H22,2H23,2H24,2H25,2H26,2H27,2H28,2H29,2H30,2H31,2H3
      32,2H33,2H34,2H35,2H36,2H37,2H38,2H39,2H40,2H41,2H42,2H43,2H44,2H45
      4,2H46,2H47,2H48,2H49,2H50,2H51,2H52,2H53,2H54,2H55,2H56,2H57,2H58,
      52H59,2H60,2H61,2H62,2H63,2H64,2H65,2H66,2H67,2H68,2H69,2H70,2H71,2
      6H72,2H73,2H74,2H75,2H76,2H77,2H78,2H79,2H80,2H81,2H82,2H83,2H84,2H
      785,2H86,2H87,2H88,2H89,2H90,2H91,2H92,2H93,2H94,2H95,2H96,2H97,2H9
      88,2H99,2HWL,2HOW,2HOP/
C
C     WRITE(16,1030)(J,J=1,41)
C     WRITE(16,1030) (J,J=3,JDIM-2)C  C       C       C       C       C
      DO 10 K=1,NWEL
      I=II(K)
      J=JJ(K)
      10 P(I,J)=1
      DO 20 K=1,NOBSW
      I=IO(K)
      J=JO(K)
      IF(P(I,J).EQ.0) P(I,J)=3
      IF(P(I,J).EQ.1) P(I,J)=4
      20 CONTINUE
      DO 50 I=1,IDIM
      DO 40 J=1,JDIM
      IF(T(I,J).GT.0) GO TO 30
      GRAPH(J)=CHAR(1)
      GO TO 40
      30 K=T(I,J)
      KSIG=T(I,J)/100
      IF(KSIG.GT.0) K=T(I,J)/10
      K=K+6
      IF(K.GT.105) K=5
```

```
      GRAPH(J)=CHAR(K)
      IF(ZK(I,J).NE.0.0) GRAPH(J)=CHAR(4)
      IF(P(I,J).EQ.1) GRAPH(J)=CHAR(3)
      IF(P(I,J).EQ.2) GRAPH(J)=CHAR(2)
      IF(P(I,J).EQ.3) GRAPH(J)=CHAR(107)
      IF(P(I,J).EQ.4) GRAPH(J)=CHAR(108)
      IF(P(I,J).EQ.1.AND.ZK(I,J).NE.0.0) GRAPH(J)=CHAR(106)
      40 CONTINUE
C     WRITE(16,1040)I,(GRAPH(J),J=1,41)
C     WRITE(16,1040) I,(GRAPH(J),J=3,JDIM-2)C     C     C     C     C
      50 CONTINUE
C     WRITE(16,1031)(J,J=42,JDIM)
      DO 51 I=1,IDIM
C     WRITE(16,1042)(GRAPH(J),J=42,JDIM)
 1042 FORMAT(6X,41A3)
      51 CONTINUE
 1031 FORMAT(1X,41I3)
      WRITE(16,1000)
 1000 FORMAT(1H1,6X,'OBSERVATION WELLS'/2X,'WELL',7X,'ROW',7X,'COLUMN',
      117X,'INITIAL  VELOCITY'/3X,"",10X,"",10X,"",11X,'(X)',8X,'(Y)',
      28X,'(R)',7X,'(THETA)'/)
      DO 60 K=1,NOBSW
      R(K)=0.0
      THETA(K)=0.0
      I=IO(K)
      J=JO(K)
      HI1=FLOAT(IH(I,J))/SCALEH
      HI2=FLOAT(IH(I,J+1))/SCALEH
      HI3=FLOAT(IH(I+1,J))/SCALEH
      HI4=FLOAT(IH(I,J-1))/SCALEH
      HI5=FLOAT(IH(I-1,J))/SCALEH
      BVXO0(K)=HC(I,J) *GRAD(HI4,HI2,DX)/POR/DELX(J)
      BVYO0(K)=HC(I,J) *GRAD(HI5,HI3,DX)/POR/DELY(I)
C     IF(BVYO0(K).LT.1.0E-10) BVYO0(K)=0.0
C     IF(BVXO0(K).LT.0.0E-10) BVXO0(K)=0.0
      IF(BVXO0(K).EQ.0.0.AND.BVYO0(K).EQ.0.0) GO TO 60
      R(K)=SQRT(BVXO0(K)**2+BVYO0(K)**2)
      THETA(K)=57.29577951*ATAN2(BVYO0(K),BVXO0(K))
      IF(THETA(K).LT.0.0) THETA(K)=THETA(K)+360.0
      IF(K.LE.NWEL) GOTO 65
      65 CONTINUE
      60 WRITE(16,1010) K,IO(K),JO(K),BVXO0(K),BVYO0(K),R(K),THETA(K)
 1010 FORMAT(1X,I3,8X,I3,8X,I3,7X,E10.4,1X,E10.4,1X,E10.4,1X,F7.1)
      WRITE(16,1020)
 1020 FORMAT(/6X,'PUMPING  WELLS'/2X,'WELL',7X,'ROW',7X,'COLUMN',
      117X,'INITIAL  VELOCITY'/3X,"",10X,"",10X,"",11X,'(X)',8X,'(Y)',
      28X,'(R)',7X,'(THETA)'/)
      DO 70 K=1,NWEL
      R(K)=0.0
      THETA(K)=0.0
      I=II(K)
      J=JJ(K)
      HI1=FLOAT(IH(I,J))/SCALEH
      HI2=FLOAT(IH(I,J+1))/SCALEH
```

```
      HI3=FLOAT(IH(I+1,J))/SCALEH
      HI4=FLOAT(IH(I,J-1))/SCALEH
      HI5=FLOAT(IH(I-1,J))/SCALEH
      BVXP0(K)=BVXP0(K)*GRAD(HI4,HI2,DX)/POR/DELX(J)
      BVYP0(K)=BVYP0(K)*GRAD(HI5,HI3,DX)/POR/DELY(I)
C     IF(BVYP0(K).LT.1.0E-10)BVYP0(K)=0.0
C     IF(BVXP0(K).LT.1.0E-10)BVXP0(K)=0.0
      IF(BVXP0(K).EQ.0.0.AND.BVYP0(K).EQ.0.0) GO TO 70
      R(K)=SQRT(BVXP0(K)**2+BVYP0(K)**2)
      THETA(K)=57.29577951*ATAN2(BVYP0(K),BVXP0(K))
      IF(THETA(K).LT.0.0) THETA(K)=THETA(K)+360.0
      70 WRITE(16,1010) K,II(K),JJ(K),BVXP0(K),BVYP0(K),R(K),THETA(K)
      RETURN
C
 1030 FORMAT(1H1,25X,'REGION MAP'//1X,'NODE '/2X,' '',1X,41I3)
 1040 FORMAT(1X,I3,2X,41A3)
      END
C
C     ******************************************************************
C
      SUBROUTINE INTVEL
C
      COMMON /C1/ IDIM,JDIM,ILESS1,JLESS1
      COMMON /C13/ DX,DELX(100),DELY(100)
      COMMON /C20/ HC(100,100),IH(100,100),POR
      COMMON /C18/ SCALET,SCALES,SCALEH,SCALEM
C
      COMMON /C50/ IR(100,100),THETA(100,100)
C
      DO 10 I=1,IDIM
      DO 10 J=1,JDIM
      IR(I,J)=0.0
      THETA(I,J)=0.0
      IF(HC(I,J).EQ.0.0) GO TO 10
      HI1=FLOAT(IH(I,J))/SCALEH
      HI2=FLOAT(IH(I,J+1))/SCALEH
      HI3=FLOAT(IH(I+1,J))/SCALEH
      HI4=FLOAT(IH(I,J-1))/SCALEH
      HI5=FLOAT(IH(I-1,J))/SCALEH
      VX0=VX0*GRAD(HI4,HI2,DX)/POR/DELX(J)
      VY0=VY0*GRAD(HI5,HI3,DX)/POR/DELY(I)
      IF(VX0.LT.1.0E-10)VX0=0.0
      IF(VY0.LT.1.0E-10)VY0=0.0
      IF(VY0.EQ.0.0.AND.VX0.EQ.0.0) GO TO 10
      IR(I,J)=SQRT(VX0**2+VY0**2)*1.0E08
      THETA(I,J)=57.29577951*ATAN2(VY0,VX0)
      IF(THETA(I,J).LT.0.0) THETA(I,J)=THETA(I,J)+360.0
      10 CONTINUE
C     WRITE(16,1000)
 1000 FORMAT(1H1,10X,'INITIAL VELOCITY FIELD'//)
      DO 20 I=1,IDIM
C     WRITE(16,1010) (IR(I,J),J=1,JDIM)
C     WRITE(32,1010) (IR(I,J),J=1,JDIM)
C     WRITE(16,1020) (THETA(I,J),J=1,JDIM)
```

```
C       WRITE(33,1020) (THETA(I,J),J=1,JDIM)
        WRITE(16,1030)
        20 CONTINUE
 1010 FORMAT(25I5)
 1020 FORMAT(25F5.0)
 1030 FORMAT(/)
        RETURN
C
        END
C
C       ***************************************************************
C
        SUBROUTINE VELFLD(KWEL)
C
        COMMON /C1/ IDIM,JDIM,ILESS1,JLESS1
        COMMON /C3/ G(100,100)
        COMMON /C5/ P(100,100)
        COMMON /C8/ KT
        COMMON /C13/ DX,DELX(100),DELY(100)
        COMMON /C14/ II(200),JJ(200),IO(200),JO(200)
        COMMON /C20/ HC(100,100),IH(100,100),POR
C
        COMMON /C50/ IR(100,100),THETA(100,100)
        INTEGER*4 P
C
        DO 10 I=1,IDIM
        DO 10 J=1,JDIM
        IR(I,J)=0.0
        THETA(I,J)=0.0
        IF(HC(I,J).EQ.0.0) GO TO 10
        VX=-HC(I,J)
        VX=VX*GRAD(G(I,J-1),G(I,J+1),DX)/POR/DELX(J)
        VY=-HC(I,J)
        VY=VY*GRAD(G(I-1,J),G(I+1,J),DX)/POR/DELY(I)
        IF(P(I,J).EQ.0) GO TO 5
        VX=0.0
        VY=0.0
        5 CONTINUE
C       IF(VX.LT.1.0E-10)VX=0.0
C       IF(VY.LT.1.0E-10)VY=0.0
        IF(VX.EQ.0.0.AND.VY.EQ.0.0) GO TO 10
        IR(I,J)=SQRT(VX**2+VY**2)*1.0E08
        THETA(I,J)=57.29577951*ATAN2(VY,VX)
        IF(THETA(I,J).LT.0.0) THETA(I,J)=THETA(I,J)+360.0
        10 CONTINUE
        WRITE(16,1000) KWEL,II(KWEL),JJ(KWEL),KT
 1000 FORMAT(1H1,10X,'VELOCITY FIELD FOR WELL "',I3,' AT NODE',
        1' (',I2,',',I2,'), LAGGED',I3,' TIME PERIODS'//)
        DO 20 I=1,IDIM
        WRITE(16,1010) (IR(I,J),J=1,JDIM)
        WRITE(16,1020) (THETA(I,J),J=1,JDIM)
        WRITE(16,1030)
        20 CONTINUE
 1010 FORMAT(25I5)
```

```
1020 FORMAT(25F5.0)
1030 FORMAT(/)
      RETURN
C
      END
C
      FUNCTION GRAD(G1,G2,DX)
      GRAD=(G1-G2)/(2.0*DX)
      RETURN
      END
```

Appendix I

Listing of Program PROG_PRE_PLAN

```
C
*************************************************************************
C   THIS PROGRAM GENERATES SCREENS ON THE DATA GENERAL'S DASHER
C   D210/211 TERMINALS FOR PREPARING DATA FOR THE PLANNING MODEL.
C
*************************************************************************
C
C   LINKAGE: F77LINK FED FORMS77.LB UTILLIB.LB
C
C   WRITTEN BY:   NADEEM SHAUKAT & YIN TAN
C                 KANSAS GEOLOGICAL SURVEY
C                 1930 CONSTANT AVE., LAWRENCE, KS 66047,USA
C                 (913) 864-3965
C   *********************************************************************
C
      IMPLICIT NONE
      INTEGER FU_ACTION
      INTEGER FU_DISPLAYMESSAGE
      INTEGER JUNK          !DUMMY VARIABLE
      INTEGER N       !CHOICE NUMBER FOR MAIN MENU
      INTEGER CURSOR      !CURRENT SCREEN POSITION
      LOGICAL DONE          !FLAG TO EXIT PROGRAM
      LOGICAL EDIT  !TRUE IF WE HAVE PREVIOUSLY ENTERED THE EDIT
                    !MODE BUT HAVE NOT WRITTEN THE DATA TO A FILE.
      LOGICAL OK    !OK IF DATA HAS BEEN WRITTEN TO A FILE SUCCESFULLY
      LOGICAL FILE_INCLUDE !INCLUDE THE FILE CONTAINING THE
C                   !STREAM AQUIFER COEFFICIENTS?
      CHARACTER*80 REPLY        !STORE USER'S RESPONSE
C
C   THE FOLLOWING ARE THE DECLARATION OF THE BUFFERS USED TO STORE
C   INFORMATION OF EACH SCREEN WE HAVE EDITED. THESE BUFFERS WILL BE
      USED
C   LATER TO GENERATE THE TARGET OUTFILE. THEY ARE ALSO USED WHEN WE
      NEED
C   TO RE-EDIT THE SCREENS. EACH OF THESE BUFFERS HAS A NUMBER OF ITS
C   CORRESPONDING FORM. SOME OF THESE BUFFERS ARE SIMPLY CHARACTER
      STRINGS
C   BUT SOME ARE ARRAYS OF STRINGS. IF A BUFFER IS AN ARRAY(eg: BUF04(10))
C   IT STORES THE INFORMATION OF THE SAME FORM BUT FOR DIFFERENT BASINS.
C   FOR EXAMPLE:
C   BUF03    STORES THE INFORMATION OF SCREEN 03 (ie. FORM F3)
C   BUF04(2) STORES THE INFORMATION OF SCREEN 04.02, WHICH MEANS
C   FORM F4 AND BASIN #2.
C
C
      CHARACTER BUF02*80,BUF03*36, BUF04(10)*17, BUF05*171, BUF06*171,
     & BUF07(10)*781, BUF08*836, BUF09*836, BUF10(10)*720,
     & BUF11(10)*597
      COMMON/BUF/BUF02,BUF03,BUF04,BUF05,BUF06,BUF07,BUF08,BUF09,
     & BUF10,BUF11
C
      CALL FU_SETUP(' ','E')
      CALL DEFINE_FORM('F1')
      JUNK=FU_ACTION(10,0,127)
```

```
      DONE=.FALSE.
      EDIT=.FALSE.
      DO WHILE (.NOT. DONE)
10    CALL FU_ERASECAREA
      CURSOR=FU_DISPLAYMESSAGE(' ENTER OPTION NUMBER: ')
      CALL GET_MESSAGE(CURSOR,REPLY)
      IF (REPLY(1:1).EQ.'H') THEN
      CALL COPYSCREEN(1,0)
      GO TO 10
      END IF
      READ(REPLY,'(I1)',ERR=10) N
      IF (N.LT.1 .OR. N.GT.5) GO TO 10
      IF (N.EQ.1) THEN
      IF (EDIT) THEN
      CALL FU_BEEP
      CURSOR=FU_DISPLAYMESSAGE('***** NO WRITE SINCE THE LAST
     & EDIT.')
      CURSOR=FU_DISPLAYMESSAGE('***** TYPE "IGNORE" TO IGNORE THE
     & PREVIOUS EDIT.')
      CURSOR=FU_DISPLAYMESSAGE('***** ANY OTHER KEY TO RESUME.?')
      CALL GET_MESSAGE(CURSOR,REPLY)
      IF (REPLY .EQ. 'IGNORE') CALL ENTER_NEW_DATA
      ELSE
      CALL ENTER_NEW_DATA
      EDIT=.TRUE.
      END IF
      END IF
      IF (N.EQ.2) THEN
      CALL EDIT_CURRENT_DATA
      EDIT=.TRUE.
      END IF
      IF (N.EQ.3) THEN
      IF (EDIT) THEN
      CALL FU_BEEP
      CURSOR=FU_DISPLAYMESSAGE('***** NO WRITE SINCE THE LAST
     & EDIT!')
      CURSOR=FU_DISPLAYMESSAGE('***** TYPE "IGNORE" TO IGNORE
     & THE PREVIOUS EDIT.')
      CURSOR=FU_DISPLAYMESSAGE('***** ANY OTHER KEY TO
     & RESUME.?')
      CALL GET_MESSAGE(CURSOR,REPLY)
      IF (REPLY.EQ.'IGNORE') THEN
      EDIT=.FALSE.
      CALL READ_DATA_FROM_FILE
      END IF
      ELSE
      CALL READ_DATA_FROM_FILE
      END IF
      END IF

      IF (N.EQ.4) THEN
      CALL WRITE_DATA_TO_FILE(OK,FILE_INCLUDE)
      EDIT=(.NOT.OK)
      END IF
```

```
      IF (N.EQ.5) THEN
      DONE=.TRUE.
      IF (EDIT) THEN
      CALL FU_BEEP
      CURSOR=FU_DISPLAYMESSAGE('***** NO WRITE SINCE THE LAST
     & EDIT!')
      CURSOR=FU_DISPLAYMESSAGE('***** TYPE "QUIT" TO CONFORM
     & EXIT.')
      CURSOR=FU_DISPLAYMESSAGE('***** ANY OTHER KEY TO RESUME.?
     & ')
      CALL GET_MESSAGE(CURSOR,REPLY)
      IF (REPLY.NE.'QUIT') DONE=.FALSE.
      END IF
      END IF
      IF (.NOT.DONE.AND.N.NE.5) CALL DEFINE_FORM('F1')
      END DO
      END



      SUBROUTINE SKIPBLANKS(STR)
*
*     Eliminates leading blanks in STR.
*     Called by: GET_MESSAGE
*
      CHARACTER*(*) STR
      IF (STR.NE.' ') THEN
      I=1
      N=LEN(STR)
      DO WHILE(I.LE.N .AND. STR(I:I).EQ.' ')
      I=I+1
      END DO
      STR=STR(I:N)
      END IF
      END



      FUNCTION LENGTHOF(STR)
      CHARACTER*(*) STR
      INTEGER LENGTHOF,I
      I=LEN(STR)
      DO WHILE(I.GT.1 .AND. STR(I:I).EQ.' ')
      I=I-1
      END DO
      LENGTHOF=I
      RETURN
      END
```

```
      SUBROUTINE ENTER_NEW_DATA
*
*     Initialize blank buffers and allow users to edit brand new screen.
*     Calls: INIT_BUFFERS, EDIT_CURRENT_DATA
*     Called by: MAIN
*
      INTEGER CURSOR,FU_DISPLAYMESSAGE
      CHARACTER*80 FNAME
      CALL INIT_BUFFERS
      CALL EDIT_CURRENT_DATA
      END



      SUBROUTINE EDIT_CURRENT_DATA
*
*     Allows users to edit/re-edit whatever contained in the current
*     buffers. After completing the modifications, the new content of each
*     screens are stored back to their corresponding buffers.
*     Calls: DEFINE_FORM, PROMPT_EXIT, NOLOOP_EDIT, LOOP_EDIT, FU_PUTFORM,
*     FU_MODFORM, FU_POSITION, FU_GETFORM
*     CALLED BY: MAIN, ENTER_NEW_DATA
*
      CHARACTER BUF02*80,BUF03*36, BUF04(10)*17, BUF05*171, BUF06*171,
     & BUF07(10)*781, BUF08*836, BUF09*836,BUF10(10)*720,
     & BUF11(10)*597
      COMMON/BUF/ BUF02,BUF03,BUF04,BUF05,BUF06,BUF07,BUF08,BUF09,
     & BUF10,BUF11
      INTEGER N,I
      LOGICAL EXIT_MAIN

      CALL DEFINE_FORM('F2')
      CALL FU_PUTFORM(BUF02)
10    CALL FU_POSITION(1)
      CALL FU_MODFORM(I)
      CALL PROMPT_EXIT(N,2,0)
      IF (N.EQ.0) THEN
      CALL FU_GETFORM(BUF02)
      ELSE IF (N.EQ.1) THEN
      GO TO 10
      ELSE
      CALL FU_GETFORM(BUF02)
      GO TO 77
      END IF

      CALL NOLOOP_EDIT(3,EXIT_MAIN)
      IF (EXIT_MAIN) GO TO 77

      CALL LOOP_EDIT(4,EXIT_MAIN)
      IF (EXIT_MAIN) GO TO 77

      DO I=5,6
      CALL NOLOOP_EDIT(1*I,EXIT_MAIN)
      IF (EXIT_MAIN) GO TO 77
```

```
      END DO

      CALL LOOP_EDIT(7,EXIT_MAIN)
      IF (EXIT_MAIN) GO TO 77

      DO I=8,9
      CALL NOLOOP_EDIT(1*I,EXIT_MAIN)
      IF (EXIT_MAIN) GO TO 77
      END DO

      DO I=10,11
      CALL LOOP_EDIT(1*I,EXIT_MAIN)
      IF (EXIT_MAIN) GO TO 77
      END DO

77    END



      SUBROUTINE WRITE_DATA_TO_FILE(OK,FILE_INCLUDE)
*
*     Prompts users for the output filename and calls GENERATE_FILE to
*     generate the target outfile based on the latest information in the
*     buffers.
*     Calls: DEFINE_FORM, GET_MESSAGE, GENERATE_FILE, FU_DISPLAYMESSAGE,
*     FU_BEEP
*     Called by: MAIN
*
      LOGICAL FILE_INCLUDE,OK
      CHARACTER FN*80,CH,FN1*80,STR*132
      INTEGER CURSOR,FU_DISPLAYMESSAGE
      CALL DEFINE_FORM('F0')
32    CURSOR=FU_DISPLAYMESSAGE('Enter the FILENAME containing the STRE
     &AM_AQUIFER_COEFFICIENTS ')
      CURSOR=FU_DISPLAYMESSAGE(' = ')
      CALL GET_MESSAGE(CURSOR,FN)
      FILE_INCLUDE = (FN.NE.' ')
      IF (FN.EQ.' ') GO TO 34
      OPEN(19,FILE=FN,IOINTENT='INPUT',PAD='YES',ERR=33,STATUS='OLD')
      GO TO 34
33    CALL FU_BEEP
      CURSOR=FU_DISPLAYMESSAGE('***** FILE NOT FOUND!!!')
      GO TO 32
34    CURSOR=FU_DISPLAYMESSAGE(' ')
      CURSOR=FU_DISPLAYMESSAGE('ENTER THE OUTPUT FILENAME: ')
      CALL GET_MESSAGE(CURSOR,FN1)
      OPEN(77,FILE=FN1,IOINTENT='OUTPUT',PAD='YES')
      CALL GENERATE_FILE(FN1,OK,FILE_INCLUDE)
      CLOSE(UNIT=77)
      CLOSE(UNIT=19)
      END


      SUBROUTINE DEFINE_FORM(F)
```

```
*
*     Called by: MAIN, EDIT_CURRENT_DATA, WRITE_DATA_TO_FILE, LOOP_EDIT,
*     NOLOOP_EDIT, READ_DATA_FROM_FILE
*     CALLS: FU_INTERNALIZE, FU_CURRENT, FU_ERASESCREEN, FU_SCROLL
*
      CHARACTER*(*) F
      INTEGER FU_INTERNALIZE,FORM
      LOGICAL LJUNK,FU_SCROLL
      FORM=FU_INTERNALIZE(F)
      CALL FU_CURRENT(FORM,.TRUE.)
      CALL FU_ERASESCREEN
      LJUNK=FU_SCROLL(.TRUE.)
      END



      SUBROUTINE PROMPT_EXIT(N,FN,BN)
*
* THIS IS CALLED IMMEDIATELY AFTER EACH PHASE (ie: COMPLETION OF ANY ONE
* SCREEN).USERS ARE GIVEN CHOICES WHETHER 0:TO PROCEED TO THE NEXT
* SCREEN,    1: TO RE-EDIT CURRENT SCREEN, OR    2:TO EXIT TO THE MAIN MENU.
* THE INTEGER OF THE CHOICES IS RETURNED IN THE FIRST ARGUMENT N.
* OF THE CHOICES IS RETURNED IN THE FIRST ARGUMENT N.
* CALLS: GET_MESSAGE, COPYSCREEN, FU_ERASESCREEN, FU_BEEP
* CALLED BY: EDIT_CURRENT_DATA
*
      INTEGER N,CURSOR,FU_DISPLAYMESSAGE,BN,FN
      CHARACTER*5 REPLY
      CALL FU_ERASECAREA
10    CURSOR=FU_DISPLAYMESSAGE(
      &'ENTER NEW-LINE TO CONTINUE, 1 TO EDIT, 2 FOR MAIN MENU:')
      CALL GET_MESSAGE(CURSOR,REPLY)
      IF (REPLY(1:1).EQ.'H') THEN
      CALL COPYSCREEN(FN,BN)
      GO TO 10
      END IF
      READ(REPLY,'(I1)',ERR=11) N
      IF (N.GE.0 .AND. N.LE.2) GO TO 12
11    CALL FU_BEEP
      GO TO 10
12    CALL FU_ERASECAREA
      END



      SUBROUTINE GENERATE_FILE(FNAME,OK,FILE_INCLUDE)
*
*     Based on the information contained in the buffers, this routine
*     generates the target outfile named after FNAME (the 1st argument).
*     It returns a flag OK (2nd argument) which tells if the outfile is
*     created successfully. If an error is found in one of the buffers,
*     OK will be set to false; in that case, this routine will tell the
*     user about the screen/buffer number in which the error is found,
*     and then exit to the main menu.
```

```
*     Calls: DEFINE_FORM, GET_MESSAGE, FU_BEEP, FU_DISPLAYMESSAGE
*     Called by: WRITE_DATA_TO_FILE
*
      CHARACTER*(*) FNAME
      CHARACTER TEMPSTR*50,ERR*34,CH,ECHOSTR*80,CCCC*4
      LOGICAL OK,FILE_INCLUDE
      INTEGER A10X10(10,10)
      INTEGER
      NBASIN,MNPDK,NPRODS,NUMREACHES,CURSOR,FU_DISPLAYMESSAGE
      INTEGER F3_1,F3_2,F3_3,F4(5),F7(1:4),F8(1:5),F9(1:5),F10_1,
     & F11_1,F11(4:9)
      REAL F3_4,F3_5,F3_6,F10(2:7),F11_2,F11_3
      REAL*8 F7R(5:7),F8R(6:8),F9R(6:8)
      REAL DELT,RATE,AMRATE
      CHARACTER BUF02*80,BUF03*36, BUF04(10)*17, BUF05*171, BUF06*171,
     & BUF07(10)*781, BUF08*836, BUF09*836,BUF10(10)*720,
     & BUF11(10)*597
      COMMON/BUF/ BUF02,BUF03,BUF04,BUF05,BUF06,BUF07,BUF08,
     & BUF09,BUF10,BUF11

      OK=.TRUE.
      WRITE(77,'(A)') 'II-E) READ TITLE OF THE PROBLEM'
      WRITE(77,'(A)') BUF02(1:LENGTHOF(BUF02))
      WRITE(77,'(A)') 'II-F) READ NO. OF BASINS, NO. OF PROJ., NO.
     &OF PERIODS, INTEREST RATES'
      READ(BUF03,'(3I6,3F6.0)',ERR=100)NBASIN,MNPDK,NPRODS,DELT,
     & RATE,AMRATE
      IF (NBASIN.GT.0)
     &WRITE(77,'(3I5,3F5.2)') NBASIN,MNPDK,NPRODS,DELT,RATE,AMRATE
      OPEN(70,FILE='ECHOFILE',IOINTENT='INPUT',PAD='YES')
71    READ(70,'(A)',END=72) ECHOSTR
      WRITE(77,'(A)') ECHOSTR(1:LENGTHOF(ECHOSTR))
      GO TO 71
72    CLOSE(UNIT=70)

      IF (NBASIN.NE.0) WRITE(77,'(A)') 'II-G) READ NO. OF PROJECTS
     &OF KIND J:'
      DO I=1,NBASIN
      READ(BUF04(I),'(2X,5I3)',ERR=101) (F4(II),II=1,5)
      WRITE(77,'(6I5)')I, (F4(II),II=1,5)
      END DO

      DO I=1,10
      DO J=1,10
      A10X10(I,J)=0
      END DO
      END DO

      K=1
      READ(BUF05(K:K+8),'(3I3)',ERR=102) I1,I2,I3
      DO WHILE (I1.NE.0)
      IF (I1.LE.10.AND.I2.LE.10) A10X10(I1,I2)=I3
      K=K+9
      READ(BUF05(K:K+8),'(3I3)',ERR=102) I1,I2,I3
```

```
END DO

IF (NBASIN.NE.0)      WRITE(77,'(A)') 'II-H) READ NO. OF UNTREATED
&WATER IMPORT PROJECTS TO EACH BASIN'
DO I=1,NBASIN
DO J=1,NBASIN
WRITE(77,'(3I5)') I,J,A10X10(I,J)
END DO
END DO

DO I=1,10
DO J=1,10
A10X10(I,J)=0
END DO
END DO

K=1
READ(BUF06(K:K+8),'(3I3)',ERR=103) I1,I2,I3
DO WHILE (I1.NE.0)
IF (I1.LE.10.AND.I2.LE.10) A10X10(I1,I2)=I3
K=K+9
READ(BUF06(K:K+8),'(3I3)',ERR=103) I1,I2,I3
END DO

IF (NBASIN.NE.0)      WRITE(77,'(A)') 'II-I) READ NO. OF TREATED WATER
& PROJECTS TO EACH BASIN'
DO I=1,NBASIN
DO J=1,NBASIN
WRITE(77,'(3I5)') I,J,A10X10(I,J)
END DO
END DO

DO I=1,NBASIN
DO J=0,18
K=3+41*J
TEMPSTR=BUF07(I)(K:K+40)
READ(TEMPSTR,'(3I3,I5,F7.0,F13.0,F7.0)',ERR=104)
& (F7(II),II=1,4),(F7R(II),II=5,7)
IF (F7(1).NE.0) THEN
IF (I.EQ.1.AND.J.EQ.0) WRITE(77,'(A)') 'II-J) READ PARAME
&TERS OF WATER PRODUCTION PROJECTS IN EACH BASIN'
WRITE(77,'(5I5,F10.2,F10.0,F10.2)') I,F7(1),F7(2),
&F7(3),F7(4),(F7R(II),II=5,7)
END IF
END DO
END DO

DO J=0,18
K=1+44*J
TEMPSTR=BUF08(K:K+43)
READ(TEMPSTR,'(4I3,I5,F7.0,F13.0,F7.0)',ERR=105)
& (F8(II),II=1,5),(F8R(II),II=6,8)
IF (F8(1).NE.0) THEN
IF (J.EQ.0)      WRITE(77,'(A)') 'II-K) READ UNTREATED WATER
```

```
      &IMPORT PROJECT PARAMETERS IN EACH BASIN'
                WRITE(77,'(5I5,F10.2,F10.0,F10.2)') F8(1),F8(2),F8(3),
      &F8(4),F8(5),(F8R(II),II=6,8)
      END IF
      END DO

      DO J=0,18
      K=1+44*J
      TEMPSTR=BUF09(K:K+43)
      READ(TEMPSTR,'(4I3,I5,F7.0,F13.0,F7.0)',ERR=106)
      & (F9(II),II=1,5),(F9R(II),II=6,8)
      IF (F9(1).NE.0) THEN
      IF (J.EQ.0)       WRITE(77,'(A)') 'II-L) READ TREATED WATER
      & IMPORT PROJECT PARAMETERS IN EACH BASIN'
      WRITE(77,'(5I5,F10.2,F10.0,F10.2)') F9(1),F9(2),F9(3),
      &F9(4),F9(5),(F9R(II),II=6,8)
      END IF
      END DO

      DO I=1,NBASIN
      DO J=0,15
      K=3+45*J
      TEMPSTR=BUF10(I)(K:K+44)
      READ(TEMPSTR,'(I3,6F7.0)',ERR=107) F10_1,(F10(II),II=2,7)
      IF (F10_1.NE.0) THEN
      IF (I.EQ.1.AND.J.EQ.0)          WRITE(77,'(A)') 'II-M) READ DEM
      &AND FOR TREATED & UNTREATED WATER IN EACH BASIN'
      WRITE(77,'(2I5,F5.1,F5.2,F5.1,F5.2,2F5.0))') I,
      &F10_1,(F10(II),II=2,7)
      END IF
      END DO
      END DO

      IF (NBASIN.NE.0) WRITE(77,'(A)') 'II-N) READ SURFACE WATER REQU
      & IREMENTS PARAMETERS FOR EACH BASIN'

      DO I=1,NBASIN
      NUMREACHES=0
      DO J=0,16
      K=3+35*J
      READ(BUF11(I)(K:K+2),'(I3)',ERR=108) F11_1
      IF (F11_1.NE.0) NUMREACHES=NUMREACHES+1
      END DO
      WRITE(77,'(2I5)') I,NUMREACHES
      DO J=0,16
      K=3+35*J
      TEMPSTR=BUF11(I)(K:K+34)
      READ(TEMPSTR,'(I3,2F7.0,6I3)',ERR=108)F11_1,F11_2,F11_3,
      & (F11(II),II=4,9)
C       IF (F11(4).NE.0) F11(4)=1
C     IF (F11(5).NE.0) F11(5)=2
C     IF (F11(6).NE.0) F11(6)=1
C     IF (F11(7).NE.0) F11(7)=2
C     IF (F11(8).NE.0) F11(8)=1
```

```
C   IF (F11(9).NE.0) F11(9)=2
    IF (F11_1.NE.0) THEN
    WRITE(77,'(2I5,2F10.3,8I5)') I,F11_1,F11_2,F11_3,
    & (F11(II),II=4,9)
    END IF
    END DO
    END DO

C   READ STREAM-AQUIFER PARAMETERS FROM A FILE.
    IF (FILE_INCLUDE) THEN
    WRITE(77,'(A)') 'II-O) READ STREAM-AQUIFER INTERACTION COEFF
    &ICIENTS'
    DO I=1,NBASIN
14  READ(19,'(A)',END=66) ECHOSTR
    IF (ECHOSTR(1:1) .NE. ' ') GO TO 66
    READ(BUF04(I),'(2X,5I3)',ERR=101) (F4(II),II=1,5)
    IF(F4(3).NE.0) WRITE(77,'(I5,A)')I,ECHOSTR(1:LENGTHOF(ECHOSTR))
    GO TO 14
    END DO
    END IF
66  CONTINUE
    WRITE(77,'(A)') 'END OF INPUT DATA'
    GO TO 300

100     ERR(29:33)='03'
        GO TO 200
101     ERR(29:31)='04.'
        WRITE(ERR(32:33),'(I2.2)') I
        GO TO 200
102     ERR(29:33)='05'
        GO TO 200
103     ERR(29:33)='06'
        GO TO 200
104     ERR(29:31)='07.'
        WRITE(ERR(32:33),'(I2.2)') I
        GO TO 200
105     ERR(29:33)='08'
        GO TO 200
106     ERR(29:33)='09'
        GO TO 200
107     ERR(29:31)='10.'
        WRITE(ERR(32:33),'(I2.2)') I
        GO TO 200
108     ERR(29:31)='11.'
        WRITE(ERR(32:33),'(I2.2)') I

200     CALL FU_BEEP
        WRITE(ERR(1:28),'(A)') '***** ERROR FOUND IN SCREEN'
        CALL DEFINE_FORM('F0')
        CURSOR=FU_DISPLAYMESSAGE('*****')
        CURSOR=FU_DISPLAYMESSAGE(ERR)
        CURSOR=FU_DISPLAYMESSAGE('*****')
        CURSOR=FU_DISPLAYMESSAGE('PLEASE CORRECT THE DATA!!!')
        CURSOR=FU_DISPLAYMESSAGE(' ')
```

```
      CURSOR=FU_DISPLAYMESSAGE('HIT RETURN TO CONTINUE:')
      CALL GET_MESSAGE(CURSOR,CH)
      OK=.FALSE.

300   CONTINUE
      END


      SUBROUTINE GET_MESSAGE(CURSOR,MSG)
*
*     Called by: MAIN, WRITE_DATA_TO_FILE, PROMPT_EXIT, GENERATE_FILE,
*       LOOP_EDIT, READ_DATA_FROM_FILE
*
      CHARACTER*(*) MSG
      INTEGER CURSOR,JUNK
      CALL FU_MODFORM(JUNK)
      CALL FU_GETMESSAGE(CURSOR,MSG,LEN(MSG))
      CALL SKIPBLANKS(MSG)
      END



      SUBROUTINE LOOP_EDIT(J,EXIT_MAIN)
*
*     Parameters: J = form number.
*     EXIT_MAIN = flag:whether we want to exit to the main menu
*     right after this?
*     A same form may be displayed one for each basin so that users can edit
*     the basin's information corresponding to that form. Therefore we need
*     a loop here to edit such information. It will update BUF04, BUF07,
*     BUF10 OR BUF11.
*     Calls: DEFINE_FORM, PROMPT_EXIT, GET_MESSAGE, FU_PUTFORM,
*       FU_PUTFIELD,
*     FU_POSITION, FU_MODFORM, FU_GETFORM, FU_DISPLAYMESSAGE
*     Called by: EDIT_CURRENT_DATA j   *
      IMPLICIT NONE
      INTEGER I,J,K,L,N,M,II,JUNK,F4(5),KIND,MNPDK,STARTPOS,NPRODS,
      & CURSOR,FU_DISPLAYMESSAGE
      LOGICAL EXIT_MAIN
      CHARACTER CCC*3,TEMPSTR*2,CH3*3,CH
      CHARACTER BUF02*80,BUF03*36, BUF04(10)*17, BUF05*171, BUF06*171,
      & BUF07(10)*781, BUF08*836, BUF09*836, BUF10(10)*720,BUF11(10)*597
      COMMON/BUF/BUF02,BUF03,BUF04,BUF05,BUF06,BUF07,BUF08,BUF09,
      &BUF10,BUF11

      CCC='F  '
      STARTPOS=2
      READ(BUF03,'(2I6)',ERR=150) N,MNPDK
      IF (J.GE.10) THEN
      WRITE(CCC(2:3),'(I2)') J
      ELSE
      WRITE(CCC(2:3),'(I1)') J
      END IF
      CALL DEFINE_FORM(CCC)
```

```
      DO I=1,N
      WRITE(TEMPSTR,'(I2)') I
      IF (J.EQ.4) THEN
      CALL FU_PUTFORM(BUF04(I))
      ELSE IF (J.EQ.7) THEN
      READ(BUF04(I),'(2X,5I3)') (F4(II),II=1,MNPDK)
      CALL FU_PUTFORM(BUF07(I))
      K=0
      DO KIND=1,MNPDK
      DO L=1,F4(KIND)
      K=K+1
      WRITE(CH3,'(I3)') KIND
      CALL FU_PUTFIELD(7*K-5,CH3)
      WRITE(CH3,'(I3)') L
      CALL FU_PUTFIELD(7*K-4,CH3)
      END DO
      END DO
      STARTPOS=4
      ELSE IF (J.EQ.10) THEN
      CALL FU_PUTFORM(BUF10(I))
      READ(BUF03,'(12X,I6)') NPRODS
      DO L=1,NPRODS
      WRITE(CH3,'(I3)') L
      CALL FU_PUTFIELD(7*L-5,CH3)
      END DO
      STARTPOS=3
      ELSE IF (J.EQ.11) THEN
      CALL FU_PUTFORM(BUF11(I))
      END IF

      CALL FU_PUTFIELD(1,TEMPSTR)
3     CURSOR=FU_DISPLAYMESSAGE(
      & 'ENTER: TAB for next field, CTRL-T for previous field,
      & NEW-LINE to exit')
      CALL FU_POSITION(STARTPOS)
      CALL FU_MODFORM(JUNK)
      CALL PROMPT_EXIT(M,J,1*I)
      IF (M.EQ.1) THEN
      GO TO 30
      ELSE
      IF (J.EQ.4) THEN
      CALL FU_GETFORM(BUF04(I))
      ELSE IF (J.EQ.7) THEN
      CALL FU_GETFORM(BUF07(I))
      ELSE IF (J.EQ.10) THEN
      CALL FU_GETFORM(BUF10(I))
      ELSE IF (J.EQ.11) THEN
      CALL FU_GETFORM(BUF11(I))
      END IF
      IF (M.EQ.2) GO TO 77
      END IF
      END DO
```

```
      GO TO 77
150     CALL FU_BEEP
      CALL DEFINE_FORM('F0')
      CURSOR=FU_DISPLAYMESSAGE('***** ERROR FOUND IN SCREEN 03')
      CURSOR=FU_DISPLAYMESSAGE('***** PLEASE CORRECT THE ERROR
     &BEFORE PROCEEDING.')
      CURSOR=FU_DISPLAYMESSAGE(' ')
      CURSOR=FU_DISPLAYMESSAGE('HIT RETURN TO CONTINUE :')
      CALL GET_MESSAGE(CURSOR,CH)
      M=2

77    EXIT_MAIN=(M.EQ.2)
      END




      SUBROUTINE NOLOOP_EDIT(J,EXIT_MAIN)
*
*     To edit a form with form number J.
*     Parameters: see SUBROUTINE LOOP_EDIT
*     It will update BUF03, BUF05, BUF06, BUF08, BUF09
*     Calls: DEFINE_FORM, PROMPT_EXIT, FU_PUTFORM, FU_MODFORM,
         FU_PUTFIELD,
*     FU_POSITION, FU_GETFORM, FU_DISPLAYMESSAGE
*     Called by: EDIT_CURRENT_DATA
*
      IMPLICIT NONE
      INTEGER  I,J,N,M,JUNK,I1,I2,I3,K,L,STARTPOS
      LOGICAL EXIT_MAIN
      CHARACTER*3 CCC,TEMPSTR*2,XXX,YYY,ZZZ
      CHARACTER BUF02*80,BUF03*36, BUF04(10)*17, BUF05*171, BUF06*171,
     &BUF07(10)*781, BUF08*836,BUF09*836,BUF10(10)*720,BUF11(10)*597
      COMMON/BUF/BUF02,BUF03,BUF04,BUF05,BUF06,BUF07,BUF08,BUF09,
     &BUF10,BUF11
      CCC='F  '
      STARTPOS=1
      IF (J.GE.10) THEN
      WRITE(CCC(2:3),'(I2)') J
      ELSE
      WRITE(CCC(2:2),'(I1)') J
      END IF
      CALL DEFINE_FORM(CCC)
      IF (J.EQ.3) THEN
      CALL FU_PUTFORM(BUF03)
      ELSE IF (J.EQ.5) THEN
      CALL FU_PUTFORM(BUF05)
      ELSE IF (J.EQ.6) THEN
      CALL FU_PUTFORM(BUF06)
      ELSE IF (J.EQ.8) THEN
      STARTPOS=4
```

```
      CALL FU_PUTFORM(BUF08)
      I=0
      K=1
      READ(BUF05(K:K+8),'(3I3)') I1,I2,I3
      DO WHILE (I1.NE.0)
      K=K+9
      WRITE(XXX,'(I3)') I1
      WRITE(YYY,'(I3)') I2
      DO L=1,I3
      WRITE(ZZZ,'(I3)') L
      I=I+1
      CALL FU_PUTFIELD(8*I-7, XXX)
      CALL FU_PUTFIELD(8*I-6, YYY)
      CALL FU_PUTFIELD(8*I-5, ZZZ)
      END DO
      READ(BUF05(K:K+8),'(3I3)') I1,I2,I3
      END DO
      ELSE IF (J.EQ.9) THEN
      CALL FU_PUTFORM(BUF09)
      STARTPOS=4
      I=0
      K=1
      READ(BUF06(K:K+8),'(3I3)') I1,I2,I3
      DO WHILE (I1.NE.0)
      K=K+9
      WRITE(XXX,'(I3)') I1
      WRITE(YYY,'(I3)') I2
      DO L=1,I3
      WRITE(ZZZ,'(I3)') L
      I=I+1
      CALL FU_PUTFIELD(8*I-7, XXX)
      CALL FU_PUTFIELD(8*I-6, YYY)
      CALL FU_PUTFIELD(8*I-5, ZZZ)
      END DO
      READ(BUF06(K:K+8),'(3I3)') I1,I2,I3
      END DO
      END IF
20    CALL FU_DISPLAYMESSAGE(
      &'ENTER: TAB for next field,   CTRL-T for previous field,
      & NEW-LINE to exit')
      CALL FU_POSITION(STARTPOS)
      CALL FU_MODFORM(JUNK)
      CALL PROMPT_EXIT(N,J,0)
      IF (N.EQ.1) THEN
      GO TO 20
      ELSE
      IF (J.EQ.3) THEN
      CALL FU_GETFORM(BUF03)
      ELSE IF (J.EQ.5) THEN
      CALL FU_GETFORM(BUF05)
      ELSE IF (J.EQ.6) THEN
      CALL FU_GETFORM(BUF06)
      ELSE IF (J.EQ.8) THEN
      CALL FU_GETFORM(BUF08)
```

```
      ELSE IF (J.EQ.9) THEN
      CALL FU_GETFORM(BUF09)
      END IF
      END IF
      EXIT_MAIN=(N.EQ.2)
      END



      SUBROUTINE READ_DATA_FROM_FILE
      CHARACTER TEMPSTR*50,FNAME*80,CCCC*4,CH
      INTEGER
      NBASIN,MNPDK,NPRODS,NUMREACHES,SUM_UNTREATED,SUM_TREATED
     & ,CURSOR,FU_DISPLAYMESSAGE,SGN,LINE
      INTEGER F3_1,F3_2,F3_3,F4(5),F7(1:4),F8(1:5),F9(1:5),F10_1,
     & F11_1,F11(4:9),NR(10)
      REAL F3_4,F3_5,F3_6,F10(2:7),F11_2,F11_3
      REAL*8 F7R(5:7),F8R(6:8),F9R(6:8)
      REAL DELT,RATE,AMRATE
      LOGICAL EX
      CHARACTER BUF02*80,BUF03*36, BUF04(10)*17, BUF05*171, BUF06*171,
     & BUF07(10)*781, BUF08*836, BUF09*836,BUF10(10)*720,BUF11(10)*597
      COMMON/BUF/BUF02,BUF03,BUF04,BUF05,BUF06,BUF07,BUF08,BUF09,
     &BUF10,BUF11
      CALL INIT_BUFFERS
      CALL DEFINE_FORM('F0')
10    CURSOR=FU_DISPLAYMESSAGE('ENTER THE INPUT FILENAME: ')
      CALL GET_MESSAGE(CURSOR,FNAME)
      OPEN(77,FILE=FNAME,IOINTENT='INPUT',PAD='YES')

12    READ(77,'(A4)',END=333) CCCC
      IF (CCCC.EQ.'II-E') GO TO 13
      IF (CCCC.EQ.'II-F') GO TO 101
      IF (CCCC.EQ.'II-G') GO TO 102
      IF (CCCC.EQ.'II-H') GO TO 103
      IF (CCCC.EQ.'II-I') GO TO 104
      IF (CCCC.EQ.'II-J') GO TO 105
      IF (CCCC.EQ.'II-K') GO TO 106
      IF (CCCC.EQ.'II-L') GO TO 107
      IF (CCCC.EQ.'II-M') GO TO 108
      IF (CCCC.EQ.'II-O') GO TO 109
      IF (CCCC.EQ.'II-N') GO TO 110
      IF (CCCC.EQ.' ') GO TO 12
      BACKSPACE(77)
      READ(77,'(A)') ECHOSTR
      WRITE(*,*) '<207>ERROR: UNKNOWN COMMAND:'
      WRITE(*,*) ' '//ECHOSTR
      WRITE(*,*) '<207>ERROR: UNKNOWN SYMBOL --- ',CCCC
      WRITE(*,*) 'HIT ENTER TO CONTINUE...'
      READ(*,'(A)') ECHOSTR
      GO TO 333

13    READ(77,'(A)',END=333) BUF02
      GO TO 12
```

```
101     READ(77,*,END=333) NBASIN,MNPDK,NPRODS,DELT,RATE,AMRATE
      WRITE(BUF03,'(3I6,F6.1,2F6.2)') NBASIN,MNPDK,NPRODS,DELT,
      & RATE,AMRATE
      GO TO 12

102     DO I=1,NBASIN
      READ(77,'(5X,5I5)',END=333) (F4(II),II=1,5)
      WRITE(BUF04(I),'(2X,5I3)') (F4(II),II=1,5)
      END DO
      GO TO 12

103     SUM_UNTREATED=0
      K=1
      DO I=1,NBASIN
      DO J=1,NBASIN
      READ(77,'(10X,I5)',END=333) I3
      IF (I3.NE.0) THEN
      SUM_UNTREATED=SUM_UNTREATED+I3
      WRITE(BUF05(K:K+8),'(3I3)') I,J,I3
      K=K+9
      END IF
      END DO
      END DO
      GO TO 12

104     SUM_TREATED=0
      K=1
      DO I=1,NBASIN
      DO J=1,NBASIN
      READ(77,'(10X,I5)',END=333) I3
      IF (I3.NE.0) THEN
      SUM_TREATED=SUM_TREATED+I3
      WRITE(BUF06(K:K+8),'(3I3)') I,J,I3
      K=K+9
      END IF
      END DO
      END DO
      GO TO 12

105     I=1
      DO WHILE (I.LE.NBASIN)
      J=I
      K=3
      READ(77,'(5I5,3F10.0)',END=333,ERR=99) IBSN,F7(1),F7(2),F7(3)
      & ,F7(4),(F7R(II),II=5,7)
      DO WHILE (IBSN.EQ.J)
      WRITE(BUF07(I)(K:K+40),'(3I3,I5,F7.2,F13.0,F7.2)')
      & (F7(II),II=1,4),(F7R(II),II=5,7)
      K=K+41
      READ(77,'(5I5,3F10.0)',END=333,ERR=99) IBSN,F7(1),F7(2),
      &F7(3),F7(4),(F7R(II),II=5,7)
      END DO
99    BACKSPACE(77)
```

```
      I=I+1
      END DO
      GO TO 12


106   K=1
      DO I=1,SUM_UNTREATED
      READ(77,'(5I5,3F10.0)',END=333) (F8(II),II=1,3),F8(4),F8(5),
      & (F8R(II),II=6,8)
      WRITE(BUF08(K:K+43),'(4I3,I5,F7.2,F13.0,F7.2)')
      & (F8(II),II=1,5),(F8R(II),II=6,8)
      K=K+44
      END DO
      GO TO 12


107   K=1
      DO I=1,SUM_TREATED
      READ(77,'(5I5,3F10.0)',END=333) (F9(II),II=1,3),F9(4),F9(5),
      & (F9R(II),II=6,8)
      WRITE(BUF09(K:K+43),'(4I3,I5,F7.2,F13.0,F7.2)')
      & (F9(II),II=1,5),(F9R(II),II=6,8)
      K=K+44
      END DO
      GO TO 12


108   I=1
      DO WHILE (I.LE.NBASIN)
      J=I
      K=3
      READ(77,'(2I5,6F5.0)',END=333,ERR=59)
      & IBSN,F10_1,(F10(II),II=2,7)
      DO WHILE (IBSN.EQ.J)
      WRITE(BUF10(I)(K:K+44),'(I3,6F7.2)') F10_1,(F10(II),II=2,7)
      K=K+45
      READ(77,'(2I5,6F5.0)',END=333,ERR=59)
      & IBSN,F10_1,(F10(II),II=2,7)
      END DO
59    BACKSPACE(77)
      I=I+1
      END DO
      GO TO 12


109   READ(77,'(A4)',END=333) CCCC
      IF (CCCC(1:1).NE.' ') GO TO 13
      GO TO 109

110   DO I=1,NBASIN
      READ(77,'(2I5)',END=333) IBSN,NR(IBSN)
      K=3
      DO J=1,NR(i)
```

```
      READ(77,'(5X,I5,2F10.0,6I5)',END=333) F11_1,F11_2,F11_3,
     & (F11(II),II=4,9)
      WRITE(BUF11(I)(K:K+34),'(I3,2F7.2,6I3)') F11_1,F11_2,
     & F11_3,(F11(II),II=4,9)
      K=K+35
      END DO
      END DO

      GO TO 12
333     CLOSE(UNIT=77)
      END



      FUNCTION SGN(N)
      INTEGER SGN,N
      IF (N.GT.0) THEN
      SGN=1
      ELSE
      SGN=0
      END IF
      RETURN
      END



      SUBROUTINE INIT_BUFFERS
*
*     initialize all buffers to blanks
*     Called by: ENTER_NEW_DATA, READ_DATA_FROM_FILE
*
*
*
      CHARACTER BUF02*80,BUF03*36, BUF04(10)*17, BUF05*171, BUF06*171,
     &BUF07(10)*781, BUF08*836, BUF09*836, BUF10(10)*720,BUF11(10)*597
      COMMON/BUF/BUF02,BUF03,BUF04,BUF05,BUF06,BUF07,BUF08,BUF09,
     &BUF10,BUF11
      BUF02=' '
      BUF03=' '
      BUF05=' '
      BUF06=' '
      BUF08=' '
      BUF09=' '
      DO I=1,10
      BUF04(I)=' '
      BUF07(I)=' '
      BUF10(I)=' '
      BUF11(I)=' '
      END DO
      END



      SUBROUTINE COPYSCREEN(FN,BN)
```

```
*
*    Creates a datasensitive file containing the format/information of the
*    current screen.
*    FN: The number corresponding to the form displayed on the screen.
*    F1 thru F11.
*    BN: The basin number. 0 if not applicable. If one screen of the
*    same form is used for each basin (eg: form F7), BN > 0.
*    The output filename of this routine will be:
*    The two digit FN if BN=0          (eg: 03, 05)
*    or
*    The two digit FN followed by .YY where YY is the two digit
*    BN          (eg: 07.02, 10.01)
*
     CHARACTER*5 OUTF
     INTEGER FN,BN
     OUTF=' '
     WRITE(OUTF(1:2),'(I2.2)') FN
     IF (BN.GT.0) THEN
     OUTF(3:3)='.'
     WRITE(OUTF(4:5),'(I2.2)') BN
     END IF
     CALL FU_COPYSCREEN(OUTF)
     END
```

Appendix J

Listing of Program WATER_PLAN_MOD

Version 3.0

```
C     ***************************************************
C     * PROGRAM : WATER_PLAN_MOD
C     *         VERSION 3.0
C     *
C     * THIS PROGRAM IS A LINEAR-INTEGER PROGRAMMING MODEL
C     * WHICH SEEKS TO MINIMIZE CC PLUS OMR OF MULTI-BASIN
C     * & MULTI-PROJECT WATER RESOURCES SYSTEMS SUBJECT TO:
C     *
C     *    1) PROJECT INITIATION
C     *    2) PROJECT LIMITATION
C     *    3) TREATED WATER DEMAND
C     *    4) TOTAL WATER DEMAND
C     *    5) MINIMUM STREAM FLOW REQUIREMENT
C     *
C     * THE PROJECTS WHICH MAY BE CONSIDERED IN EACH BASIN
C     * ARE:
C     *
C     *    1) INSTREAM DIVERSIONS
C     *    2) SURFACE RESERVOIRS
C     *    3) WELL FIELDS
C     *    4) DESALINIZATION PROJECTS
C     *    5) WATER TREATMENT PLANTS
C     *    6) TREATED WATER IMPORTS AND EXPORTS
C     *    7) UNTREATED WATER IMPORTS AND EXPORTS
C     *
C     * NAME OF CONSTRAINTS:
C     *
C     *  1) IDDbtnn: initiation constraint to build diversion
C     *           dam nn in basin b at time period t.
C     *
C     *  2) IRSbtnn: initiation constraint to build reservoir
C     *           nn in basin b at time period t.
C     *
C     *  3) IWFbtnn: initiation constraint to build well field
C     *           nn in basin b at time period t.
C     *
C     *  4) IDPbtnn: initiation constriant to build
C     *           desalinization plant nn in basin b at
C     *           time period t.
C     *
C     *  5) ITPbtnn: initiation constraint to build treament
C     *           plant nn in basin b at time period t.
C     *
C     *  6) IUIabtnn: initiation constraint to import
C     *           untreated water from basin b to basin
C     *           a through project nn during period t.
C     *
C     *  7) ITIabtnn: initiation constraint to import treated
C     *           water from basin b to basin a through
C     *           project nn during period t.
C     *
C     *  8) LDDbtnn: project limitation constraint for
C     *           diversion dam nn in basin b at time
C     *           period t.
```

```
C    *
c    * 9) LRSbtnn: project limitation constraint for
C    *          reservoir nn in basin b at time period
C    *          t.
C    *
C    * 10) LDPbtnn: project limitation constraint for
C    *          decalinization plant nn in basin b at
C    *          time period t.
C    *
C    * 11) LWFbtnn: project limitation constraint for well
C    *          filed nn in basin b at time t.
C    *
C    * 12) LTPbtnn: project limitation constraint for treat-
C    *          ment nn in basin b at time period t.
C    *
C    * 13) LUIabtnn: projrct limitation constraint for un-
C    *          treated water import project nn from
C    *          basin b to basin a at time period t.
C    *
C    * 14) LTIabtnn: project limitation constraint for
C    *          treated water import project nn from
C    *          basin b to basin a at time period t.
C    *
C    * 15) DTWat : demand for treated water in basin a during
C    *          period t.
C    *
C    * 16) DTOat : demand for total water in basin a during
C    *          period t.
C    *
C    * 17) SFLbtr: minimum stream flow requirement of stream
C    *          reach r in basin b during time period t.
C    *
C    * NAME OF VARIABLES:
C    *
C    * 1) DDDbtnn: 0-1 decision variable to build diversion dam
C    *          nn in basin b at time period t.
C    *
C    * 2) DRSbtnn: 0-1 decision variable to build reservoir
C    *          nn in basin b at time period t.
C    *
C    * 3) DWFbtnn: 0-1 decision variable to build well field
C    *          nn in basin b at time period t.
C    *
C    * 4) DDPbtnn: 0-1 decision variable to build desali-
C    *          nization plant nn in basin b at time
C    *          period t.
C    *
C    * 5) DTPbtnn: 0-1 decision variable to build treatment
C    *          plant nn in basin b at time period t.
C    *
C    * 6) DUIabtnn: 0-1 decision variable to import untreated
C    *          water through project nn from basin b to
C    *          basin a at time period t.
C    *
```

```
C     * 7) DTIabtnn: 0-1 decision variable to import treated
C     *          water through project nn from basin b to
C     *          basin a at time period t.
C     *
C     * 8) QDDbtnn : decision variable to divert water from the
C     *          nn-th diversion dam in basin b during time
C     *          period t.
C     *
C     * 9) QRSbtnn : decision variable to divert water from the
C     *          nn-th reservoir in basin b during time
C     *          period t.
C     *
C     *10) QWFbtnn : decision variable to pump water from the
C     *          nn-th well field in basin b during time
C     *          period t.
C     *
C     *11) QDPbtnn : decision variable to process water in the
C     *          nn-th desalinization plant in basin b
C     *          during time period t.
C     *
C     *12) QTPbtnn : decision variable to prcess water in the
C     *          nn-th treatment plant in basin b during
C     *          time period t.
C     *
C     *13) QUIabtnn : decision variable to import untreated
C     *          water from basin b to basin a through
C     *          nnth project during time period t.
C     *
C     *14) QTIabtnn : decision variable to import treated
C     *          water from basin b to basin a through
C     *          nnth project during time period t.
C     *
C     * The authors of this program have checked its integrity
C     * as much as possible. However due to the existing
C     * limitations chances are that some particular loops have
C     * not been accessed by the test data. Therefore, the user
C     * is cautioned to check the results carefully.
C     *
C     * FOR MORE INFORMATION CONTACT:
C     *
C     * 1)  DR. M. HEIDARI                    2)  DR. T. MADDOCK,III
C     *     KANSAS GEOLOGICAL SURVEY              DEPT. OF HYDROLOGY
C     *     UNIVERSITY OF KANSAS                  UNIVERSITY OF ARIZONA
C     *     1930 CONSTANT AV.                     TUCSON , AZ. 85718
C     *     LAWRENCE , KS. 66046                  USA, PH:(602) 621-7115
C     *     USA, PH:(913) 864-3965
C     *
C     * 3)  MR. N. SHAUKAT
C     *     KANSAS GEOLOGICAL SURVEY
C     *     THE UNIVERSITY OF KANSAS
C     *     1930 CONSTANT AV.
C     *     LAWRENCE , KS. 66046
C     *     USA, PH:(913) 864-3965
C     ************************************************
```

```
C
C MAIN PROGRAM
C -----------
C
C       OPEN I/O FILES
C
        CALL IOFILES
C
C       READ IN DATA, AND DO PRELIMINARY PROCESS
C
        CALL IN_DATA
C
C       SET UP MATRIX FOR OPTIMIZER
C
        CALL MATRIX
C
C       SCALE MATRIX FOR OPTIMIZER
C
        CALL SCALER
C
      STOP
      END
      SUBROUTINE  IOFILES
C      ------------------
C     ****************************************************
C     * I ) : THIS SUBROUTINE OPENS ALL I/O FILES USED.
C     *
C     *         I/O FILES:
C     *
C     *         FILE 15: CONTAINS ALL INPUT DATA
C     *         FILE 16: CONTAINS OUTPUT DATA
C     *         FILE 17: CONTAINS UNSCALED MATRIX
C     *         FILE 18: CONTAINS SCRATCH MATRIX DATA
C     *         FILE 19: CONTAINS SCALED MATRIX USED BY
C     *         LINDO
C     ****************************************************
C
        CHARACTER*32 FILE15, FILE16, FILE17, FILE18, FILE19
C
C       INITIAL I/O FILE NAMES
C
        FILE15='PLAN_MOD_IN_DATA'
        FILE16='PLAN_MOD_OUT_DATA'
        FILE17='PLAN_MOD_OUT_MATRIX'
C
C       THESE FILES WILL BE OPEN IN SCALER SUBROUTINE IF NECESSARY
C
C       FILE18='MATRIX_SCRATCH'  :open in scaler
C       FILE19='PLAN_MOD_SCALED_MATRIX' open in scaler
C
C       OPEN FILES
C ----------
C
C THE FOLLOWING OPEN STATEMENTS ARE MACHINE DEPENDENT AND ARE
```

```
C WRITTEN FOR DATA GENERAL MV20000. THE USER MAY HAVE TO CHANGE
C THESE TO MAKE THEM COMPATIBLE WITH THE EXISTING MACHINE.
C
C
      OPEN(15,FILE=FILE15,IOINTENT='INPUT',PAD='YES')
      OPEN(16,FILE=FILE16,IOINTENT='OUTPUT',STATUS='FRESH')
      OPEN(17,FILE=FILE17,IOINTENT='OUTPUT',STATUS='FRESH')
C
    RETURN
    END
    SUBROUTINE IN_DATA
C    -----------------
C    ******************************************************
C    * II ) : THIS SUBROUTINE READS ALL THE INPUT DATA
C    *       FROM FILE 15.
C    ******************************************************
C
     REAL*8 LINE(16)/13*'--------',3*'        '/
    CHARACTER EXIST*3,PROJK(5)*10,ANSWER*1
    COMMON/C01/ROW(17),COL(14),BASIN(10)
    COMMON/C02/TITLE(20)
     COMMON/C03/NPROJ(10,10),NUWIP(10,10),NTWIP(10,10)
     COMMON/C04/LIFE(10,10,10),YIELD(10,10,10),CC(10,10,10)
    COMMON/C05/OMR(10,10,10),IFLAG(10,10,10)
     COMMON/C06/IFUIP(10,10,10),LFUIP(10,10,10),YLDUIP(10,10,10)
     COMMON/C07/CCUIP(10,10,10),OMUIP(10,10,10),CUWIP(10,10,10)
     COMMON/C08/IFTIP(10,10,10),LFTIP(10,10,10),YLDTIP(10,10,10)
     COMMON/C09/CCTIP(10,10,10),OMTIP(10,10,10),CTWIP(10,10,10)
    COMMON/C10/DTWATR(10,10),PLTWTR(10,10),PTTWD(10,10)
    COMMON/C11/DUWATR(10,10),PLUWTR(10,10),PTUWD(10,10)
    COMMON/C12/NRMF(10),FLOWIN(10,10),FLOWOT(10,10),
    &NPLNR(10,10,10)
    COMMON/C13/TCAP(10),CAP(10,10,10)
     COMMON/C14/AFC(10,10,10),AFCUIP(10,10,10),AFCTIP(10,10,10)
    COMMON/C15/BSA(10,10,10),NWFLD(10)
     COMMON/C16/DTWPPP(10,10),DUWPPP(10,10),TWDPPP(10,10)
    COMMON/C17/NBASIN,NPRODS,AMRATE,DELT,MNPDK,RATE,RTPRD
    CHARACTER*4 TLINE(20)
C
C II-A ):INITIALIZE ABREVIATIONS FOR ROW NAMES
C
      DATA ROW/'IDD','IRS','IWF','IDP','ITP','IUI','ITI',
      &'LDD','LRS','LWF','LDP','LTP','LUI','LTI',
     &'DTW','DTO','SFL'/
C
C II-B ):INITIALIZE ABREVIATIONS FOR COLUMN NAMES
C
      DATA COL/'DDD','DRS','DWF','DDP','DTP','DUI','DTI',
      &'QDD','QRS','QWF','QDP','QTP','QUI','QTI'/
C
C II-C ):INITIALIZE ABREVIATIONS FOR BASINS :
C
        DATA BASIN/'A','B','C','D','E','F','G','H','I','J'/
C
```

```
C II-D ):INITIALIZE ABREVIATIONS FOR DIFFERENT KINDS OF PROJECTS
C
      DATA PROJK/'DIVERSION-','RESERVOIR-','AQUIFER-',
     &'DESALINIZ-','TREATMENT-'/
   10 READ(15,1000)TLINE
      IF(TLINE(1).EQ.'II-E')GO TO 15
      IF(TLINE(1).EQ.'II-F')GO TO 20
      IF(TLINE(1).EQ.'II-G')GO TO 25
      IF(TLINE(1).EQ.'II-H')GO TO 45
      IF(TLINE(1).EQ.'II-I')GO TO 55
      IF(TLINE(1).EQ.'II-J')GO TO 65
      IF(TLINE(1).EQ.'II-K')GO TO 110
      IF(TLINE(1).EQ.'II-L')GO TO 145
      IF(TLINE(1).EQ.'II-M')GO TO 180
      IF(TLINE(1).EQ.'II-N')GO TO 205
      IF(TLINE(1).EQ.'II-O')GO TO 230
      IF(TLINE(1).EQ.'END ')RETURN
C
C II-E ):READ TITLE OF THE PROBLEM :
C
   15 READ(15,1000)TITLE
      WRITE(16,2000)TITLE
 2000 FORMAT(////,20X,20A4)
      GO TO 10
C
C II-F ) : READ NBASIN : NO. OF BASINS ,
C           MNPDK : MAXIMUM NO. OF PROJECTS,
C           NPRODS : NO. OF TIME PERIODS IN THIS ANALYSIS,
C           DELT : LENGTH OF EACH TIME PERIOD (YRS),
C           RATE : INTEREST RATE,
C           AMRATE : INTEREST RATE FOR AMORTIZATION.
C
   20 READ(15,1010)NBASIN,MNPDK,NPRODS,DELT,RATE,AMRATE
C
C CONVERT INTEREST RATE PER YEAR TO INTEREST RATE PER TIME
C
      RTPRD=(1.+RATE)**DELT-1.
      GO TO 10
C
C II-G ) : READ
C       NPROJ(IB,J) : NO. OF WATER PRODUCING PROJECTS OF
C       KIND J:
C                 J=1 : NO. OF DIVERSION DAMS
C                 J=2 : NO. OF SURFACE RESERVOIRS
C                 J=3 : NO. OF WELL FIELDS
C                 J=4 : NO. OF DESALINIZATION PLANTS
C                 J=5 : NO. OF WATER TREATMENT PLANTS
C
   25 DO 30 I=1,NBASIN
      READ(15,1015)IB,(NPROJ(IB,J),J=1,MNPDK)
   30 CONTINUE
C
C INITIALIZE THE STREAM_AQUIFER COEFFICIENTS;
C
```

```
      DO 40 IB=1,NBASIN
      IF(NPROJ(IB,3).EQ.0)GO TO 40
       NPIBK3=NPROJ(IB,3)
      DO 35 IP=1,NPIBK3
      DO 35 IT=1,NPRODS
       BSA(IB,IP,IT)=0.
   35 CONTINUE
   40 CONTINUE
     GO TO 10
C
C II-H ): READ NO. OF UNTREATED WATER IMPORTS TO EACH BASIN:
C          IBI : THE IMPORTING BASIN,
C          IBE : THE EXPORTING BASIN,
C          NUWIP : NO OF UNTREATED WATER IMPORT PROJECTS,
C
   45 DO 50 IB=1,NBASIN
      DO 50 JB=1,NBASIN
      READ(15,1015)IBI,IBE,NUWIP(IBI,IBE)
   50 CONTINUE
     GO TO 10
C II-I ) : READ NO. OF TREATED WATER IMPORTS TO EACH BASIN
C          IBI : THE IMPORTING BASIN,
C          IBE : THE EXPORTING BASIN,
C          NTWIP: NO. OF TREATED WATER IMPORT PROJECTS.
C
   55 DO 60 IB=1,NBASIN
      DO 60 JB=1,NBASIN
      READ(15,1015)IBI,IBE,NTWIP(IBI,IBE)
   60 CONTINUE
     GO TO 10
C
C II-J ) : READ PARAMETERS OF WATER PRODUCING PROJECTS IN
C         EACH BASIN
C          IB : BASIN NO.,
C          K : PROJECT KIND,
C          N: NO. ASSOCIATED WITH PROJECT OF KIND K,
C          LIFE : ECONOMIC LIFE OF N-TH PROJECTION (YRS),
C          YIELD : YIELD OF N-TH PROJECT IN (MGD),
C          CC : CAPITAL COST OF N-TH PROJECT ($),
C          OMR :UNIT COST OF OPERATION OF N-TH PROJECT:($/MG)
C          IFLAG :FLAG INDICATING WHETHER N EXISTS (0,1)
C
   65 DO 80 I=1,NBASIN
      DO 75 J=1,MNPDK
       NPKJ=NPROJ(I,J)
      IF(NPKJ.EQ.0)GO TO 75
      DO 70 NP=1,NPKJ
       READ(15,1020)IB,K,N,IFLAG(IB,K,N),LIFE(IB,K,N),
      &YIELD(IB,K,N),CC(IB,K,N),OMR(IB,K,N)
   70 CONTINUE
   75 CONTINUE
   80 CONTINUE
      PAMR=AMRATE*100.
C
```

```
C WRITE TABLE OF WATER PRODUCING PROJECTS PARAMETERS
C
      WRITE(16,2020)DELT,PAMR,DELT
 2020 FORMAT(///,10X,'SUMMARY OF PROJECT PARAMETERS USED IN',
     &' THIS PLANNING ANALYSIS',/22X,'WATER PRODUCTION ',
      &'PARAMETERS'//25X,'PROJECT',14X,'YIELD OVER',4X,
      &'FIXED',4X,'ANNUAL FIXED  OPERATING',14X,'TOTAL YIELD'/,
     &5X,'BASIN',5X,'PROJECT   LIFE    YIELD    TIME',
       &' PERIOD',4X,'COSTS',5X,'COSTS AT',7X,'COSTS',4X,
      &'PROJECT',5X,'OVER TIME PERIOD',/,
        &16X,'KIND',6X,'(YRS)',4X,'(MGD)',6X,'(',F3.1,
       &' YRS)',15X,'(',F3.1,' %)',7X,'(PER MG)',2X,'EXISTING?',
       &6X,'(',F3.1,' YRS)')
C
C CONVERT UNITS AND CALCULATE ANNUAL FIXED COSTS OF PROJECTS
C
      DO 105 IB=1,NBASIN
      TCAP(IB)=0.
      DO 100 K=1,MNPDK
      IF(NPROJ(IB,K).EQ.0)GO TO 100
       NPIBK=NPROJ(IB,K)
      DO 95 N=1,NPIBK
C
C CALCULATE ANNUAL COST OF N-TH PROJECT
C
      NIB=IB
      NK=K
      NN=N
      NIBE=0
      CALL ANCOST(NIB,NK,NN,NIBE)
C
C CONVERT YIELD FROM MGD TO MG PER PLANNING PERIOD
C
      CAP(IB,K,N)=YIELD(IB,K,N)*365.*DELT
      IF(K.EQ.5)GO TO 90
C
C SUM CAPACITIES OF PROJECTS EXCLUDING OF TREATMENT PLANTS
C FOR EACH BASIN.
C
      TCAP(IB)=TCAP(IB)+CAP(IB,K,N)
   90 EXIST=' NO'
       IF(IFLAG(IB,K,N).EQ.1)EXIST='YES'
       WRITE(16,2025)BASIN(IB),PROJK(K),N,LIFE(IB,K,N),
       &YIELD(IB,K,N),CAP(IB,K,N),CC(IB,K,N),AFC(IB,K,N),
      &OMR(IB,K,N),EXIST
 2025  FORMAT(7X,A1,5X,A10,I2,I4,5X,F6.2,2X,F10.0,2X,F11.0,
       &3X,F11.0,1X,F9.2,6X,A3)
  95 CONTINUE
 100 CONTINUE
      WRITE(16,2030)TCAP(IB)
 2030 FORMAT(' ',105X,F12.0//)
 105 CONTINUE
      GO TO 10
C
```

```
C II-K ): READ UNTREATED WATER IMPORT ROJECT PARAMETERS IN
C       EACH BASIN :
C       IBI: NO. OF BASIN IMPORTING UNTREATED WATER
C       IBE: NO. OF BASIN EXPORTING UNTREATED WATER
C       N: NO. ASSOCIATED WITH THE PROJECT
C       IFUIP : FLAG INDICATING N-TH PROJECT EXISTS OR NOT
C             (0,1)
C       LFUIP: ECONOMIC LIFE OF THE N-TH PROJECT
C       YLDUIP: YIELD OF THE N-TH PROJECT IN (MGD)
C       CCUIP : CAPITAL COST OF N-TH PROJECT ($)
C       OMUIP : UNIT COST OF OPERATION OF N-TH PROJECT ($/MG)
C
  110 DO 125 I=1,NBASIN
      DO 120 J=1,NBASIN
      IF(NUWIP(I,J).EQ.0)GO TO 120
       NPK6=NUWIP(I,J)
      DO 115 NP=1,NPK6
       READ(15,1020)IBI,IBE,N,IFUIP(IBI,IBE,N),LFUIP(IBI,IBE,N),
      &YLDUIP(IBI,IBE,N),CCUIP(IBI,IBE,N),OMUIP(IBI,IBE,N)
  115 CONTINUE
  120 CONTINUE
  125 CONTINUE
C
C WRITE TABLE OF UNTREATED WATER IMPORTS
C
      WRITE(16,2035)DELT,PAMR
 2035 FORMAT(///,10X,'SUMMARY OF PROJECT PARAMETERS USED IN',
      &' THIS PLANNING ANALYSIS',/22X,'UNTREATED WATER IMPORT',
       &' PARAMETERS'//22X,'PROJECT',9X,'YIELD OVER',10X,
       &'ANNUAL FIXED','  OPERATING'/,6X,'BASIN',12X,
      &'LIFE    YIELD  TIME PERIOD  FIXED',4X,'COSTS AT'
        &,6X,'COSTS',7X,'PROJECT'/,1X,'IMPORTING EXPORTING',
         &3X,'(YRS)',3X,'(MGD)',3X,'(',F3.1,'  YRS)',3X,'COSTS',
        &7X,'(',F3.1,' %)',4X,'(PER MG)',2X,'EXISTINGING ?'/)
      WRITE(16,2021)LINE
 2021 FORMAT('+',16A8)
C
C CONVERT UNITS AND CALCULATE ANNUAL FIXED COSTS OF PROJECTS
C
      DO 140 IBI=1,NBASIN
      DO 135 IBE=1,NBASIN
       IF(IBI.EQ.IBE.OR.NUWIP(IBI,IBE).EQ.0)GO TO 135
       NPIBK6=NUWIP(IBI,IBE)
      DO 130 N=1,NPIBK6
C
C CALAULATE ANNUAL COSTS OF N-TH PROJECT OF KIND K=6
C
       EXIST=' NO'
        IF(IFUIP(IBI,IBE,N).EQ.1)EXIST='YES'
      NIBI=IBI
      K=6
      NN=N
      NIBE=IBE
      CALL ANCOST(NIBI,K,NN,NIBE)
```

```
      CUWIP(IBI,IBE,N)=365.*DELT*YLDUIP(IBI,IBE,N)
       WRITE(16,2040)BASIN(IBI),BASIN(IBE),N,LFUIP(IBI,IBE,N),
     &YLDUIP(IBI,IBE,N),CUWIP(IBI,IBE,N),
     &CCUIP(IBI,IBE,N),AFCUIP(IBI,IBE,N),OMUIP(IBI,IBE,N)
     &,EXIST
 2040  FORMAT(5X,A1,8X,A1,'-',I1,7X,I3,4X,F6.2,1X,F10.0,
     &F11.0,F11.0,2X,F9.2,5X,A4)
  130 CONTINUE
  135 CONTINUE
  140 CONTINUE
     GO TO 10
C
C II-L ): READ TREATED WATER IMPORT PROJECT PARAMETERS
C       IN EACH BASIN
C       IBI: NO. OF BASIN IMPORTING TREATED WATER
C       IBE: NO. OF BASIN EXPORTING TREATED WATER
C       N : NO. ASSOCIATED WITH THE PROJECT
C       IFTIP : FLAG INDICATING N-TH PROJECT EXISTS (0,1)
C       LFTIP : ECONOMIC LIFE OF THE N-TH PROJECT
C       YLDTIP : YIELD OF THE N-TH PROJECT (MGD)
C       CCTIP : CAPITAL COST OF N-TH PROJECT ($)
C       OMTIMP : UNIT COST OF OPERATION OF N-TH PROJECT($)
C
  145 DO 160 I=1,NBASIN
     DO 155 J=1,NBASIN
     IF(NTWIP(I,J).EQ.0)GO TO 155
      NPK7=NTWIP(I,J)
     DO 150 NP=1,NPK7
      READ(15,1020)IBI,IBE,N,IFTIP(IBI,IBE,N),LFTIP(IBI,IBE,N),
     &YLDTIP(IBI,IBE,N),CCTIP(IBI,IBE,N),OMTIP(IBI,IBE,N)
  150 CONTINUE
  155 CONTINUE
  160 CONTINUE
C
C WRITE TABLE OF TREATED WATER IMPORTS
C
     WRITE(16,2045)DELT,PAMR
 2045 FORMAT(///,10X,'SUMMARY OF PROJECT PARAMETERS USED IN',
     &' THIS PLANNING ANALYSIS',/22X,'TREATED WATER IMPORT',
      &' PARAMETERS'//22X,'PROJECT',9X,'YIELD OVER',10X,
      &'ANNUAL FIXED','  OPERATING'/,6X,'BASIN',12X,
     &'LIFE    YIELD  TIME PERIOD  FIXED',4X,'COSTS AT'
       &,6X,'COSTS',7X,'PROJECT'/,1X,'IMPORTING EXPORTING',
        &3X,'(YRS)',3X,'(MGD)',3X,'(',F3.1,'  YRS)',3X,'COSTS',
       &7X,'(',F3.1,' %)',4X,'(PER MG)',2X,'EXISTINGING ?'/)
     WRITE(16,2021)LINE
C
C CONVERT UNITS AND CALCULATE ANNUAL FIXED COSTS OF PROJECTS
C
     DO 175 IBI=1,NBASIN
     DO 170 IBE=1,NBASIN
     IF(IBI.EQ.IBE.OR.NTWIP(IBI,IBE).EQ.0)GO TO 170
     NPIBK7=NTWIP(IBI,IBE)
     DO 165 N=1,NPIBK7
```

```
C
C CALAULATE ANNUAL COSTS OF N-TH PROJECT OF KIND K=7
C
     EXIST=' NO'
     IF(IFTIP(IBI,IBE,N).EQ.1)EXIST='YES'
     NIBI=IBI
     K=7
     NN=N
     NIBE=IBE
     CALL ANCOST(NIBI,K,NN,NIBE)
     CTWIP(IBI,IBE,N)=365.*DELT*YLDTIP(IBI,IBE,N)
     WRITE(16,2040)BASIN(IBI),BASIN(IBE),N,LFTIP(IBI,IBE,N),
    &YLDTIP(IBI,IBE,N),CTWIP(IBI,IBE,N),
    &CCTIP(IBI,IBE,N),AFCTIP(IBI,IBE,N),OMTIP(IBI,IBE,N)
    &,EXIST
  165 CONTINUE
  170 CONTINUE
  175 CONTINUE
     GO TO 10
C
C II-M ): READ DEMANDS FOR TREATED & UNTREATED WATER IN BASIN:
C       IB: NO. OF BASIN
C       IT: TIME PERIOD
C       DTWATR : DEMAND FOR TREATED WATER IN (MGD)
C       PLTWTR : PERCENT LOSS OF TREATED WATER (DECIMAL)
C       DUWATR : DEMAND FOR UNTREATED WATER IN (MGD)
C       PLUWTR : PERCENT LOSS OF UNTREATED WATER (DECIMAL)
C       PTTWD  : PERCENT TOTAL TREATED WATER DEMAND TO BE
C              SATISFIED (DECIMAL)
C       PTUWD  : PERCENT TOTAL UNTREATED WATER DEMAND TO BE
C              SATISFIED (DECIMAL)
C
  180 DO 185 I=1,NBASIN
     DO 185 IP=1,NPRODS
     READ(15,1025)IB,IT,DTWATR(IB,IT),PLTWTR(IB,IT),
    &DUWATR(IB,IT),PLUWTR(IB,IT),
    &PTTWD(IB,IT),PTUWD(IB,IT)
  185 CONTINUE
C
C WRITE WATER DEMAND TABLE FOR EACH YEAR
C
     WRITE(16,2050)
 2050 FORMAT(///,30X,'SUMMARY OF WATER DEMANDS USED IN THIS ',
      &' ANALYSIS',//36X,'TREATED',5X,'UNTREATED'/4X,'BASIN',
      &5X,'TIME',4X,'TREATED WATER',2X,'WATER',8X,'WATER',7X,
      &'UNTREATED',7X,'TREATED','-WATER',2X,'UNTREATED-WATER'
      &/15X,'PERIOD',5X,'DEMAND',4X,'LOSS FACTOR',4X,'DEMAND'
     &,6X,'LOSS FACTOR',5X,'TARGET DEMAND',3X,'TARGET DEMAND'
       &,/25X,'(MGD)',21X,'(MGD)',28X,'(%)',14X,'(%)'/)
     WRITE(16,2021)LINE
     DO 190 IB=1,NBASIN
     DO 190 IT=1,NPRODS
     WRITE(16,2055)BASIN(IB),IT,DTWATR(IB,IT),PLTWTR(IB,IT),
    &DUWATR(IB,IT),PLUWTR(IB,IT),
```

```
     &PTTWD(IB,IT),PTUWD(IB,IT)
 2055  FORMAT(6X,A1,7X,I4,5X,F8.2,6X,F6.4,6X,F8.2,8X,F6.4,9X,F6.2,
     &9X,F6.2)
  190 CONTINUE
C
C CONVERT WATER DEMANDS FROM MGD TO MG PER PLANNING PERIOD
C
      DO 195 IB=1,NBASIN
      DO 195 IT=1,NPRODS
      DTWPPP(IB,IT)=(365.*DELT*DTWATR(IB,IT)/(1.-PLTWTR(IB,IT)))*
     &(PTTWD(IB,IT)/100.)
      DUWPPP(IB,IT)=(365.*DELT*DUWATR(IB,IT)/(1.-PLUWTR(IB,IT)))*
     &(PTUWD(IB,IT)/100.)
      TWDPPP(IB,IT)=DTWPPP(IB,IT)+DUWPPP(IB,IT)
  195 CONTINUE
C
C WRITE TABLE OF WATER DEMANDS PER PLANNING PERIOD
C
      WRITE(16,2060)DELT,DELT,DELT
 2060  FORMAT(///,49X,'UNTREATED'/4X,'BASIN',5X,'TIME',3X,
     &'TREATED-',
      &'WATER',1X,'TREATED-WATER',3X,'WATER',8X,'UNTREATED',15X,
      &'TOTAL ADJUSTED'/15X,'PERIOD',5X,'DEMAND',5X,
      &'LOSS FACTOR',3X,'DEMAND',8X,'LOSS FACTOR',16X,'DEMAND'/,
       &22X,'(MG/',F3.1,' YRS)',16X,'(MG/',F3.1,' YRS)',25X,
      &'(MG/',F3.1,' YRS)')
      WRITE(16,2021)LINE
      DO 200 IB=1,NBASIN
      DO 200 IT=1,NPRODS
      WRITE(16,2065)BASIN(IB),IT,DTWPPP(IB,IT),PLTWTR(IB,IT),
     &DUWPPP(IB,IT),PLTWTR(IB,IT),TWDPPP(IB,IT)
  200 CONTINUE
  2065  FORMAT(6X,A1,7X,I4,2X,F11.2,6X,F6.4,3X,F11.2,8X,F6.4,
     &18X,F12.2)
      GO TO 10
C
C II-N ): READ SURFACE WATER REQUIREMENTS PARAMETERS FOR BASIN
C  IB: NO. OF BASINS,
C  NRMF(IB): # OF REACHES WITH MINIMUM FLOW REQUIREMENT IN IB,
C  FLOWIN(IB,NR): FLOW INTO REACH NR OF BASIN IB (MGD)
C  FLOWOT(IB,NR): MINIMUM FLOW REQUIREMENT OUT OF REACH NR(MGD)
C  NPLNR(IB,NR,NP): PROJECT NO. WHICH DIVERTS FROM NR
C               NP=1,2 : DIVERSION DAMS
C               NP=3,4 : SURFACE RESERVOIRS
C               NP=5,6 : WELL FIELDS WHICH HAVE INTERACTION
C                     WHITH NR.
C
  205 WRITE(16,2068)
 2068 FORMAT(///,10X,'SUMMARY OF NATURAL INFLOWS AND REQUIRED',
     &' OUTFLOWS',
     &' INTO AND OUT OF DIFFERENT STREAM REACHES',//,4X,'BASIN',
      &2X,'REACH',3X,'NATURAL',10X,'REQUIRED',
     &10X,'PROJECTS WHICH AFFECT STREAM FLOWS',/,
       &13X,'NO.',3X,'FLOWS',12X,'FLOWS',
```

```
     &10X,'DIVERSION  RESERVOIRS  WELL FIELDS',/,
      &19X,'(MGD)',12X,'(MGD)',12X,'DAMS',
     &15X,'(INCLUSIVE)',/)
     WRITE(16,2021)LINE
     DO 215 I=1,NBASIN
     READ(15,1015)IB,NRMF(IB)
     IF(NRMF(IB).EQ.0)GO TO 215
     NRMFIB=NRMF(IB)
     DO 210 N=1,NRMFIB
      READ(15,1030)IB,NR,FLOWIN(IB,NR),FLOWOT(IB,NR),
     &(NPLNR(IB,NR,NP),NP=1,6)
      WRITE(16,2075)IB,NR,FLOWIN(IB,NR),FLOWOT(IB,NR),
     &(NPLNR(IB,NR,NP),NP=1,6)
C
C CONVERT MGD TO MG PER PLANNING PERIOD
C
     FLOWIN(IB,NR)=FLOWIN(IB,NR)*365.*DELT
     FLOWOT(IB,NR)=FLOWOT(IB,NR)*365.*DELT
  210 CONTINUE
  215 CONTINUE
     WRITE(16,2070)DELT,DELT
 2070 FORMAT(///,10X,'SUMMARY OF NATURAL INFLOWS AND REQUIRED',
     &' OUTFLOWS ',
     &'INTO AND OUT OF DIFFERENT STREAM REACHES',//,4X,'BASIN',
      &2X,'REACH',3X,'NATURAL',10X,'REQUIRED',
     &10X,'PROJECTS WHICH AFFECT STREAM FLOWS',/,
       &13X,'NO.',3X,'FLOWS',12X,'FLOWS',
     &10X,'DIVERSION  RESERVOIRS  WELL FIELDS',/,
       &12X,'(MG/',F3.1,' YRS)',6X,'(MG/',F3.1,' YRS)',7X,'DAMS',
     &17X,'(INCLUSIVE)',/)
     WRITE(16,2021)LINE
     DO 225 IB=1,NBASIN
     IF(NRMF(IB).EQ.0)GO TO 225
     NRMFIB=NRMF(IB)
     DO 220 NR=1,NRMFIB
      WRITE(16,2075)IB,NR,FLOWIN(IB,NR),FLOWOT(IB,NR),
     &(NPLNR(IB,NR,NP),NP=1,6)
  2075  FORMAT(3X,I4,4X,I4,3X,F8.1,10X,F8.1,5X,2I5,2X,2I5,2X,2I5)
  220 CONTINUE
  225 CONTINUE
     GO TO 10
C
C II-O ): READ OR GENERATE STREAM-AQUIFER INTERACTION COEFFI-
C        CIENTS FOR EACH WELL FIELD IN EACH BASIN IF STREAM-
C        AQUIFER INTERACTION EXISTS.
C
C     WRITE(*,*)'DO YOU WISH TO GENERATE STREAM_AQUIFER CO-',
C     &'EFFICIENTS?'
C     READ(*,'(A1)')ANSWER
C
C
C GENERATE STREAM_AQUIFER COEFFICIENTS; note:this option is not
C included in this version
C     CALL PROG_BETA
```

```
C    GO TO 256
C
 230 DO 240 IB=1,NBASIN
     IF(NPROJ(IB,3).EQ.0)GO TO 240
     NPIBK3=NPROJ(IB,3)
     DO 235 IP=1,NPIBK3
     READ(15,1035)IIB,NN,(BSA(IIB,NN,IT),IT=1,NPRODS)
 235 CONTINUE
 240 CONTINUE

C
C WRITE STREAM-AQUIFER INTERACTION COEFFICIENTS
C
     WRITE(16,2080)(I,I=1,10)
2080 FORMAT(///,10X,'SUMMARY OF STREAM-AQUIFER INTERACTION',
     &' COEFFICIENTS'//2X,'BASIN',3X,'TIME',30X,'WELL FIELDS',/
      &,4X,'NO.',2X,'PERIOD',1X,10(3X,I2,3X))
     WRITE(16,2021)LINE
     DO 250 IB=1,NBASIN
     IF(NPROJ(IB,3).EQ.0)GO TO 250
     NPIBK3=NPROJ(IB,3)
     IF(NRMF(IB).EQ.0)GO TO 250
     NRMFIB=NRMF(IB)
     DO 245 IT=1,NPRODS
      WRITE(16,2090)BASIN(IB),IT,(BSA(IB,N,IT),N=1,NPIBK3)
 245 CONTINUE
 250 CONTINUE
2085 FORMAT(//1X,I4,10X,14(2X,I4,2X))
2090 FORMAT(5X,A1,5X,I2,3X,10(F8.5))
1000 FORMAT(20A4)
1005 FORMAT(2A4,2A4,2A4)
1010 FORMAT(3I5,3F5.0)
1015 FORMAT(16I5)
1020 FORMAT(5I5,4F10.0)
1025 FORMAT(2I5,14F5.0)
1030 FORMAT(2I5,2F10.0,6I5)
1035 FORMAT(2I5,10F6.0)
     RETURN
     END
     SUBROUTINE MATRIX
C    ----------------
C    ****************************************************
C    * III ):SETS UP THE LINEAR INTEGER PROGRAMMING
C    *       MATRIX AND WRITES IT ON FILE UNIT 17
C    *       IT CALLS THE FOLLOWING SUBROUTINES :
C    *                1) NAMES   (III-A)
C    *                2) COLUMNS (III-B)
C    *                3) RHS     (III-C)
C    *                4) BOUNDS  (III-D)
C    ****************************************************
C
     CALL NAMES
     CALL COLUMNS
     CALL RHS
```

```
      CALL BOUNDS
      WRITE(17,2000)
 2000 FORMAT('ENDATA')
      CLOSE(17)
      RETURN
      END
      SUBROUTINE NAMES
C     ---------------
C     ******************************************************
C     * III-A) : WRITES THE NAMES OF ROWS ON FILE UNIT
C     *        17 IN THE FOLLOWING ORDER:
C     *            1- NAME OF PROBLEM : NAME
C     *            2-BEGININNG OF ROW NAMES : ROWS
C     *            3- OBJECTIVE FUNCTION NAME : MINCOST
C     *            4- CONSTRAINTS ON PROJECT INITIATION
C     *            5- CONSTRAINTS ON PROJECT LIMITATION
C     *           6-CONSTRAINTS ON TREATED WATER DEMANDS
C     *            7- CONSTRAINTS ON TOTAL WATER DEMANDS
C     *            8-CONSTRAINTS ON MINIMUM STREAM FLOWS
C     ******************************************************
C
      COMMON/C01/ROW(17),COL(14),BASIN(10)
      COMMON/C02/TITLE(20)
       COMMON/C03/NPROJ(10,10),NUWIP(10,10),NTWIP(10,10)
       COMMON/C04/LIFE(10,10,10),YIELD(10,10,10),CC(10,10,10)
      COMMON/C05/OMR(10,10,10),IFLAG(10,10,10)
       COMMON/C06/IFUIP(10,10,10),LFUIP(10,10,10),YLDUIP(10,10,10)
       COMMON/C07/CCUIP(10,10,10),OMUIP(10,10,10),CUWIP(10,10,10)
       COMMON/C08/IFTIP(10,10,10),LFTIP(10,10,10),YLDTIP(10,10,10)
       COMMON/C09/CCTIP(10,10,10),OMTIP(10,10,10),CTWIP(10,10,10)
      COMMON/C10/DTWATR(10,10),PLTWTR(10,10),PTTWD(10,10)
      COMMON/C11/DUWATR(10,10),PLUWTR(10,10),PTUWD(10,10)
      COMMON/C12/NRMF(10),FLOWIN(10,10),FLOWOT(10,10),
      &NPLNR(10,10,10)
       COMMON/C16/DTWPPP(10,10),DUWPPP(10,10),TWDPPP(10,10)
      COMMON/C17/NBASIN;NPRODS,AMRATE,DELT,MNPDK,RATE,RTPRD
C
C III-A-1 ): WRITE NAME OF PROBLEM
C
      WRITE(17,2000)(TITLE(I),I=1,2)
 2000 FORMAT('NAME',10X,2A4)
C
C III-A-2 ) : WRITE BEGINNING OF ROWS
C III-A-3 ) : WRITE NAME OF OBJECTIVE FUNCTION
C
      WRITE(17,2001)
 2001  FORMAT('ROWS'/,2X,'N',1X,'MINCOST')
C
C ENTER CONSTRAINTS ROW NAMES FOR EACH BASIN
C
      DO 600 IB=1,NBASIN
C
C III-A-4-a ): NAME OF CONSTRAINTS ON PROJECT INITIATIONS FOR:
C             K=1) DIVESION DAMS
```

```
C              K=2) SURFACE RESERVOIRS
C              K=3) WELL FIELDS
C              K=4) DESALINIZATION PLANTS
C              K=5) TREATMENT PLANTS
C
      DO 105 K=1,MNPDK
      IF(NPROJ(IB,K).EQ.0)GO TO 105
       NPIBK=NPROJ(IB,K)
      DO 100 IT=1,NPRODS
      DO 100 N=1,NPIBK
      IF(IFLAG(IB,K,N).EQ.1)GO TO 100
      N1=N/10
      N2=N-N1*10
       WRITE(17,2002)ROW(K),BASIN(IB),IT,N1,N2
 2002 FORMAT(2X,'G',1X,A3,A1,3I1)
  100 CONTINUE
  105 CONTINUE
C
C III-A-4-b ): NAME OF CONSTRAINTS ON PROJECT INITIATIONS FOR:
C          K=6) UNTREATED WATER IMPORT PROJECTS
C
      DO 120 IBE=1,NBASIN
      IF(IB.EQ.IBE.OR.NUWIP(IB,IBE).EQ.0) GO TO 120
      NPIB6=NUWIP(IB,IBE)
      DO 115 IT=1,NPRODS
      DO 110 N=1,NPIB6
      IF(IFUIP(IB,IBE,N).EQ.1)GO TO 110
      N1=N/10
      N2=N-N1*10
       WRITE(17,2003)ROW(6),BASIN(IB),BASIN(IBE),IT,N1,N2
 2003 FORMAT(2X,'G',1X,A3,2A1,3I1)
  110 CONTINUE
  115 CONTINUE
  120 CONTINUE
C
C III-A-4-c ): NAME OF CONSTRAINTS ON PROJECT INITIATIONS FOR:
C          K=7) TREATED WATER IMPORT PROJECTS
C
      DO 135 IBE=1,NBASIN
      IF(IBE.EQ.IB.OR.NTWIP(IB,IBE).EQ.0)GO TO 135
      NPIB7=NTWIP(IB,IBE)
      DO 130 IT=1,NPRODS
      DO 125 N=1,NPIB7
      IF(IFTIP(IB,IBE,N).EQ.1)GO TO 125
      N1=N/10
      N2=N-N1*10
       WRITE(17,2003)ROW(7),BASIN(IB),BASIN(IBE),IT,N1,N2
  125 CONTINUE
  130 CONTINUE
  135 CONTINUE
C
C III-A-5-a ): NAME OF CONSTRAINTS ON PROJECT LIMITATIONS FOR:
C
C              K=1) DIVERSION DAMS
```

```
C              K=2) SURFACE RESERVOIRS
C              K=3) WELL FIELDS
C              K=4) DESALINIZATION PLANTS
C              K=5) TREATMENT PLANTS
C
      DO 145 K=1,5
      IF(NPROJ(IB,K).EQ.0)GO TO 145
      KK=K+7
      NPIBK=NPROJ(IB,K)
      DO 140 N=1,NPIBK
      IF(IFLAG(IB,K,N).EQ.1)GO TO 140
      N1=N/10
      N2=N-N1*10
      WRITE(17,2004)ROW(KK),BASIN(IB),N1,N2
 2004 FORMAT(2X,'L',1X,A3,A1,2I1)
  140 CONTINUE
  145 CONTINUE
C
C III-A-5-b ): NAME OF CONSTRAINTS ON PROJECT LIMITATIONS FOR:
C          K=6) UNTREATED WATER IMPORT PROJECTS
C
      DO 155 IBE=1,NBASIN
      IF(IB.EQ.IBE.OR.NUWIP(IB,IBE).EQ.0)GO TO 155
      NPIB6=NUWIP(IB,IBE)
      DO 150 N=1,NPIB6
      IF(IFUIP(IB,IBE,N).EQ.1)GO TO 150
      N1=N/10
      N2=N-N1*10
      WRITE(17,2005)ROW(13),BASIN(IB),BASIN(IBE),N1,N2
 2005 FORMAT(2X,'L',1X,A3,2A1,2I1)
  150 CONTINUE
  155 CONTINUE
C
C III-A-5-c ): NAME OF CONSTRAINTS ON PROJECT LIMITATIONS FOR:
C          K=7) TRATED WATER IMPORT PROJECTS
C
      DO 165 IBE=1,NBASIN
      IF(IB.EQ.IBE.OR.NTWIP(IB,IBE).EQ.0)GO TO 165
      NPIB7=NTWIP(IB,IBE)
      DO 160 N=1,NPIB7
      IF(IFTIP(IB,IBE,N).EQ.1)GO TO 160
      N1=N/10
      N2=N-N1*10
      WRITE(17,2005)ROW(14),BASIN(IB),BASIN(IBE),N1,N2
  160 CONTINUE
  165 CONTINUE
C
C III-A-6 ): NAME OF CONSTRAINTS ON TREATED WATER DEMANDS
C
      DO 170 IT=1,NPRODS
      WRITE(17,2006)ROW(15),BASIN(IB),IT
 2006 FORMAT(2X,'G',1X,A3,A1,I1)
  170 CONTINUE
C
```

```
C III-A-7 ) : NAME OF CONSTRAINTS ON TOTAL WATER DEMANDS
C
      DO 175 IT=1,NPRODS
      WRITE(17,2006)ROW(16),BASIN(IB),IT
  175 CONTINUE
C
C III-A-8 ): NAME OF CONSTRAINTS ON MINIMUM STREAM FLOWS
C
      NRMFIB=NRMF(IB)
      DO 180 IT=1,NPRODS
      DO 180 NR=1,NRMFIB
      WRITE(17,2007)ROW(17),BASIN(IB),IT,NR
 2007  FORMAT(2X,'L',1X,A3,A1,2I1)
  180 CONTINUE
  600 CONTINUE
      RETURN
      END

      SUBROUTINE COLUMNS
C     ------------------
C     ****************************************************
C     * III-B ) : SETS UP COEFFICIENTS ASSOCIATED WITH
C     * EACH ELEMENT OF COLUMNS OF MATRIX AND WRITES THEM ON
C     * FILE 17.
C     ****************************************************
C
      COMMON/C01/ROW(17),COL(14),BASIN(10)
      COMMON/C03/NPROJ(10,10),NUWIP(10,10),NTWIP(10,10)
      COMMON/C04/LIFE(10,10,10),YIELD(10,10,10),CC(10,10,10)
      COMMON/C05/OMR(10,10,10),IFLAG(10,10,10)
      COMMON/C06/IFUIP(10,10,10),LFUIP(10,10,10),YLDUIP(10,10,10)
      COMMON/C07/CCUIP(10,10,10),OMUIP(10,10,10),CUWIP(10,10,10)
      COMMON/C08/IFTIP(10,10,10),LFTIP(10,10,10),YLDTIP(10,10,10)
      COMMON/C09/CCTIP(10,10,10),OMTIP(10,10,10),CTWIP(10,10,10)
      COMMON/C10/DTWATR(10,10),PLTWTR(10,10)
      COMMON/C11/DUWATR(10,10),PLUWTR(10,10)
      COMMON/C12/NRMF(10),FLOWIN(10,10),FLOWOT(10,10),
     &NPLNR(10,10,10)
      COMMON/C13/TCAP(10),CAP(10,10,10)
      COMMON/C14/AFC(10,10,10),AFCUIP(10,10,10),AFCTIP(10,10,10)
      COMMON/C15/BSA(10,10,10),NWFLD(10)
      COMMON/C17/NBASIN,NPRODS,AMRATE,DELT,MNPDK,RATE,RTPRD
C
C III-B-1 ): WRITE COLUMN CONTROL CARD :
C
      WRITE(17,2000)
 2000 FORMAT('COLUMNS')
C
C III-B-2 ): WRITE INTEGER MARKER CARD
C
      WRITE(17,2001)
  2001  FORMAT(4X,'BEGIN',5X,"'MARKER'",17X,"'INTORG'")
C
C III-B-3 ): WRITE COEFFICIENTS OF INTEGER VARIABLES
```

```
C
      DO 600 IB=1,NBASIN
C
C III-B-3-a): WRITE COEFFICIENTS OF DECISIONS TO BUILD PROJECTS
C           OF KIND K:
C              K=1) DIVERSION DAMS
C              K=2) SURFACE RESERVOIRS
C              K=3) WELL FIELDS
C              K=4) DESALINIZATION PLANTS
C              K=5) TREATMENT PLANTS
C
      DO 125 K=1,MNPDK
      IF(NPROJ(IB,K).EQ.0)GO TO 125
       NPIBK=NPROJ(IB,K)
      DO 120 IT=1,NPRODS
      DO 115 N=1,NPIBK
      IF(IFLAG(IB,K,N).EQ.1)GO TO 115
      N1=N/10
      N2=N-N1*10
C
C CONVERT ANNUAL FIXED COSTS (AFC) TO PRESENT VALUE ON AN
C ANNUAL BASIS AND SUM UP THESE COSTS OVER THE TIME PERIOD
C LEFT IN PLANNING HORIZON.
C FIND YEARS IN WHICH SUMMATION BEGINS AND ENDS (IYSRT ,IYEND)
C
      IYEND=NPRODS*DELT
      IYSRT=IT*DELT-(DELT-1.0)
      NOYRS=IYEND-IYSRT+1
C
C IS ECONOMIC LIFE OF PROJECT LESS THAN NOYRS ?
C
       IF(LIFE(IB,K,N).LT.NOYRS)IYEND=LIFE(IB,K,N)+IYSRT-1
C
C SUM PRESENT VALUE MULTIPLIER FROM IYSRT TO IYEND
C
      PREV=0.
      DO 100 IY=IYSRT,IYEND
  100 PREV=PREV+(1./((1.+RATE)**IY))
      PVCC=PREV*AFC(IB,K,N)
      IF(PVCC.EQ.0.)GO TO 105
C
C WRITE PRESENT VALUE OF ANNUAL FIXED CAPITAL COST IN
C OBJECTIVE ROW
C
       WRITE(17,2002)COL(K),BASIN(IB),IT,N1,N2,PVCC
 2002 FORMAT(4X,A3,A1,3I1,3X,'MINCOST',3X,F12.2)
  105 CONTINUE
C
C WRITE PROJECT CAPACITIES FOR INITIALIZATION CONSTRAINTS
C
      DO 110 I=IT,NPRODS
       WRITE(17,2003)COL(K),BASIN(IB),IT,N1,N2,ROW(K),BASIN(IB),I,
      &N1,N2,CAP(IB,K,N)
 2003 FORMAT(4X,A3,A1,3I1,3X,A3,A1,3I1,3X,F12.2)
```

```
 110 CONTINUE
C
C WRITE PROJECT LIMITATION CONSTRAINT FOR PROJECT OF KIND K
C
      WRITE(17,2004)COL(K),BASIN(IB),IT,N1,N2,ROW(K+7),
     &BASIN(IB),N1,N2
 2004 FORMAT(4X,A3,A1,3I1,3X,A3,A1,2I1,10X,'1.0000')
 115 CONTINUE
 120 CONTINUE
 125 CONTINUE
C
C III-B-3-b ): WRITE COEFFICIENTS OF DECISIONS TO BUILD UNTREATED
C           WATER TRANSPORT PROJECTS INTO IB :
C
     DO 155 IBE=1,NBASIN
     IF(IB.EQ.IBE.OR.NUWIP(IB,IBE).EQ.0)GO TO 155
     NPIB6=NUWIP(IB,IBE)
     DO 150 IT=1,NPRODS
     DO 145 N=1,NPIB6
     IF(IFUIP(IB,IBE,N).EQ.1)GO TO 145
     N1=N/10
     N2=N-N1*10
C
C CONVERT ANNUAL FIXED COSTS TO PRESENT VALUE ON AN ANNUAL
C BASIS AND SUM BY TIME INCREMENTS EQUAL TO THE LENGTH OF
C PLANNING PERIOD FOR THIS PROJECT.
C
     IYSRT=IT*DELT-(DELT-1.)
     IYEND=NPRODS*DELT
     NOYRS=IYEND-IYSRT+1
      IF(LFUIP(IB,IBE,N).LT.NOYRS)IYEND=LFUIP(IB,IBE,N)+IYSRT-1
C
C SUM PRESENT VALUE MULTIPLIER FROM IYSRT TO IYEND
C
     PREV=0.
     DO 130 IY=IYSRT,IYEND
  130 PREV=PREV+(1./((1.+RATE)**IY))
C
C ADJUST ANNUAL FIXED COST OF PROJECT
C
     PREV=PREV*AFCUIP(IB,IBE,N)
     IF(PREV.EQ.0.)GO TO 135
C
C WRITE FIXED COST COEFFICIENT OF THIS PROJECT IN OBJECTIVE ROW
C
      WRITE(17,2005)COL(6),BASIN(IB),BASIN(IBE),IT,N1,N2,PREV
 2005 FORMAT(4X,A3,2A1,3I1,2X,'MINCOST',3X,F12.2)
C
C WRITE PROJECT CAPACITIES FOR INITIALIZATION CONSTRAINTS:
C
  135 DO 140 IT1=IT,NPRODS
      WRITE(17,2006)COL(6),BASIN(IB),BASIN(IBE),IT,N1,N2,ROW(6),
     &BASIN(IB),BASIN(IBE),IT1,N1,N2,CUWIP(IB,IBE,N)
 2006 FORMAT(4X,A3,2A1,3I1,2X,A3,2A1,3I1,2X,F12.2)
```

```
 140 CONTINUE
C
C WRITE PROJECT LIMITATION CONSTRAINTS
C
     WRITE(17,2007)COL(6),BASIN(IB),BASIN(IBE),IT,N1,N2,ROW(13),
    &BASIN(IB),BASIN(IBE),N1,N2
2007  FORMAT(4X,A3,2A1,3I1,2X,A3,2A1,2I1,9X,'1.0000')
 145 CONTINUE
 150 CONTINUE
 155 CONTINUE
C
C III-B-3-c ): WRITE COEFFICIENTS OF DECISIONS TO BUILD TREATED
C           WATER TRANSPORT PROJECTS INTO IB :
C
     DO 185 IBE=1,NBASIN
     IF(IB.EQ.IBE.OR.NTWIP(IB,IBE).EQ.0)GO TO 185
     NPIB7=NTWIP(IB,IBE)
     DO 180 IT=1,NPRODS
     DO 175 N=1,NPIB7
     IF(IFTIP(IB,IBE,N).EQ.1)GO TO 175
     N1=N/10
     N2=N-N1*10
C
C CONVERT ANNUAL FIXED COSTS TO PRESENT VALUE ON AN ANNUAL
C BASIS AND SUM BY TIME INCREMENTS EQUAL TO THE LENGTH OF
C PLANNING PERIOD FOR THIS PROJECT.
C
     IYSRT=IT*DELT-(DELT-1.)
     IYEND=NPRODS*DELT
     NOYRS=IYEND-IYSRT+1
     IF(LFTIP(IB,IBE,N).LT.NOYRS)IYEND=LFTIP(IB,IBE,N)+IYSRT-1
C
C SUM PRESENT VALUE MULTIPLIER FROM IYSRT TO IYEND
C
     PREV=0.
     DO 160 IY=IYSRT,IYEND
 160 PREV=PREV+(1./((1.+RATE)**IY))
C
C ADJUST ANNUAL FIXED COST OF PROJECT
C
     PREV=PREV*AFCTIP(IB,IBE,N)
     IF(PREV.EQ.0.)GO TO 165
C
C WRITE FIXED COST COEFFICIENT OF THIS PROJECT IN OBJECTIVE ROW
C
     WRITE(17,2005)COL(7),BASIN(IB),BASIN(IBE),IT,N1,N2,PREV
C
C WRITE PROJECT CAPACITIES FOR INITIALIZATION CONSTRAINTS:
C
 165 DO 170 IT1=IT,NPRODS
     WRITE(17,2006)COL(7),BASIN(IB),BASIN(IBE),IT,N1,N2,ROW(7),
    &BASIN(IB),BASIN(IBE),IT1,N1,N2,CTWIP(IB,IBE,N)
 170 CONTINUE
C
```

```
C WRITE PROJECT LIMITATION CONSTRAINTS
C
     WRITE(17,2007)COL(7),BASIN(IB),BASIN(IBE),IT,N1,N2,ROW(14),
    &BASIN(IB),BASIN(IBE),N1,N2
 175 CONTINUE
 180 CONTINUE
 185 CONTINUE
 600 CONTINUE
C
C END OF INTEGER COLUMN SECTION
C
     WRITE(17,2008)
 2008  FORMAT(4X,'FINIS',5X,'"MARKER"',17X,'"INTEND"')
C
C III-B-4 ): WRITE COEFFICIENTS OF THE CONTINUOUS VARIABLES
C          FOR EACH BASIN
C
     DO 700 IB=1,NBASIN
C
C III-B-4-a ):  WRITE COEFFICIENTS OF SUPPLY DECISIONS FOR
C             PROJECTS OF KIND K:
C               K=1) DIVERSION DAMS
C               K=2) SURFACE RESERVOIRS
C               K=3) WELL FIELDS
C               K=4) DESALINIZATION PLANTS
C               K=5) TREATMENT PLANTS
C
     DO 270 K=1,MNPDK
     IF(NPROJ(IB,K).EQ.0)GO TO 270
      NPIBK=NPROJ(IB,K)
     DO 265 IT=1,NPRODS
     DO 260 N=1,NPIBK
     N1=N/10
     N2=N-N1*10
C
C ADJUST OPERATION COSTS TO PRESENT VALUE
C
     PVOMR=OMR(IB,K,N)/((1.+RTPRD)**IT)
C
C IV-A-1) WRITE COEFFICIENT OF OMR IN OBJECTIVE FUNCTION
C
      WRITE(17,2009)COL(K+7),BASIN(IB),IT,N1,N2,PVOMR
 2009  FORMAT(4X,A3,A1,3I1,3X,'MINCOST',3X,F12.4)
C
C SET COEFFICIENT OF THE NON_EXISTING PROJECTS IN INITIATION
C CONSTRAINT ROWS EQUAL TO -1.0. FOR EXISTING PROJECTS, BOUDS
C ON THE CAPACITIES OF THE PROJECTS WILL BE SET.
C
      IF(IFLAG(IB,K,N).EQ.1)GO TO 190
      WRITE(17,2010)COL(K+7),BASIN(IB),IT,N1,N2,ROW(K),
     &BASIN(IB),IT,N1,N2
 2010 FORMAT(4X,A3,A1,3I1,3X,A3,A1,3I1,8X,'-1.0000')
C
C SET COEFFICIENTS OF TREATED WATER (K=3,4,5) DEMAND CONSTRAINTS
```

```
C K EQUAL TO 1.0 .
C
 190 IF(K.LT.3) GO TO 220
C
     WRITE(17,2011)COL(K+7),BASIN(IB),IT,N1,N2,ROW(15),
    &BASIN(IB),IT
 2011  FORMAT(4X,A3,A1,3I1,3X,A3,A1,I1,11X,'1.0000')
C
C IF PROJECT IS NOT A WATER TREATMENT PLANT, SET THE COEFFICIENT
C OF TOTAL WATER DEMAND ROW (DTO) EQUAL TO 1.0
C
     IF(K.EQ.5)GO TO 260
  220 WRITE(17,2011)COL(K+7),BASIN(IB),IT,N1,N2,ROW(16),
    &BASIN(IB),IT
     IF(NRMF(IB).EQ.0)GO TO 260
     NRMFIB=NRMF(IB)
C
C IF PROJECT K IS A DIVERSION DAM OR RESERVOIR (K=1,2), SET THE
C COEFFICIENT OF FLOW REQUIREMENT CONSTRAINT ROW(SFL) EQUAL TO 1
C
     GO TO(225,235,245,260,260) K
C
C SEARCH ARRAY NPLNR(IB,NR,NP) TO FIND WHICH PROJECTS,NP,DIVERT
C WATER FROM REACH NR :
C                    NP=1,2 : DIVERSION DAMS
C                    NP=3,4 : SURFACE RESERVOIRS
C                    NP=5,6 : WELL FIELDS
C
 225 CONTINUE
C
C K=1 ) DIVERSION DAMS
C
     DO 230 NR=1,NRMFIB
       IF(N.EQ.NPLNR(IB,NR,1).OR.N.EQ.NPLNR(IB,NR,2))
    &WRITE(17,2012)
     &COL(K+7),BASIN(IB),IT,N1,N2,ROW(17),BASIN(IB),IT,NR
 2012 FORMAT(4X,A3,A1,3I1,3X,A3,A1,2I1,10X,'1.0000')
  230 CONTINUE
     GO TO 260
  235 CONTINUE
C
C K=2 ) SURFACE RESERVOIRS
C
     DO 240 NR=1,NRMFIB
       IF(N.EQ.NPLNR(IB,NR,3).OR.N.EQ.NPLNR(IB,NR,4))
    &WRITE(17,2012)
     &COL(K+7),BASIN(IB),IT,N1,N2,ROW(17),BASIN(IB),IT,NR
  240 CONTINUE
     GO TO 260
  245 CONTINUE
C
C K=3 ) WELL FIELDS
C WRITE SURFACE_GROUND WATER COEFFICIENTS IN FLOW REQUIREMENT
C CONSTRAINTS FOR PRESENT TIME AND ALL FUTURE TIMES.
```

```
C
      ITT=0
      DO 255 IT1=IT,NPRODS
      ITT=ITT+1
      BETA=BSA(IB,N,ITT)
      DO 250 NR=1,NRMFIB
        IF(N.GE.NPLNR(IB,NR,5).AND.N.LE.NPLNR(IB,NR,6))
     &WRITE(17,2013)
     &COL(K+7),BASIN(IB),IT,N1,N2,ROW(17),BASIN(IB),IT1,NR,BETA
 2013 FORMAT(4X,A3,A1,3I1,3X,A3,A1,2I1,4X,F12.4)
  250 CONTINUE
  255 CONTINUE
  260 CONTINUE
  265 CONTINUE
  270 CONTINUE
C
C III-B-4-b,c ): WRITE COEFFICIENTS OF WATER IMPORT PROJECTS
C III-B-4-b ): UNTREATED WATER IMPORTS
C
      DO 285 IBE=1,NBASIN
      IF(IB.EQ.IBE.OR.NUWIP(IB,IBE).EQ.0)GO TO 285
      NPIBK6=NUWIP(IB,IBE)
      DO 280 IT=1,NPRODS
      DO 275 N=1,NPIBK6
      N1=N/10
      N2=N-N1*10
C
C ADJUST OPERATING COST TO PRESENT VALUE
C
      PVOMR=OMUIP(IB,IBE,N)/((1.0+RTPRD)**IT)
C
C WRITE OPERATING COSTS IN OBJECTIVE FUNCTION
C
      WRITE(17,2014)COL(13),BASIN(IB),BASIN(IBE),IT,N1,N2,PVOMR
 2014 FORMAT(4X,A3,2A1,3I1,2X,'MINCOST',3X,F12.4)
C
C IF PROJECT DOES NOT EXIST , ENTER -1. IN PROJECT INITIATION
C CONSTRAINTS
C
      IF(IFUIP(IB,IBE,N).EQ.0)WRITE(17,2015)COL(13),BASIN(IB),
     &BASIN(IBE),IT,N1,N2,ROW(6),BASIN(IB),BASIN(IBE),IT,N1,N2
 2015 FORMAT(4X,A3,2A1,3I1,2X,A3,2A1,3I1,7X,'-1.0000')
C
C ENTER 1.0 IN TOTAL WATER DEMAND CONSTRAINTS, AND -1.0 IN SAME
C ROW OF BASIN IBE INDICATING EXPORT
C
      WRITE(17,2016)COL(13),BASIN(IB),BASIN(IBE),IT,N1,N2,
     &ROW(16),BASIN(IB),IT,ROW(16),BASIN(IBE),IT
 2016 FORMAT(4X,A3,2A1,3I1,2X,A3,A1,I1,11X,'1.0000',3X,A3,A1,I1,
     &10X,'-1.0000')
  275 CONTINUE
  280 CONTINUE
  285 CONTINUE
C
```

```
C III-B-4-c ): TREATED WATER IMPORTS
C
      DO 300 IBE=1,NBASIN
      IF(IB.EQ.IBE.OR.NTWIP(IB,IBE).EQ.0)GO TO 300
      NPIBK7=NTWIP(IB,IBE)
      DO 295 IT=1,NPRODS
      DO 290 N=1,NPIBK7
      N1=N/10
      N2=N-N1*10
C
C ADJUST OPERATING COST TO PRESENT VALUE
C
      PVOMR=OMTIP(IB,IBE,N)/((1.0+RTPRD)**IT)
C
C WRITE OPERATING COSTS IN OBJECTIVE FUNCTION
C
       WRITE(17,2014)COL(14),BASIN(IB),BASIN(IBE),IT,N1,N2,PVOMR
C
C IF PROJECT DOES NOT EXIST , ENTER -1. IN PROJECT INITIATION
C CONSTRAINTS
C
       IF(IFTIP(IB,IBE,N).EQ.0)WRITE(17,2015)COL(14),BASIN(IB),
      &BASIN(IBE),IT,N1,N2,ROW(7),BASIN(IB),BASIN(IBE),IT,N1,N2
C
C ENTER 1.0 IN TREATED WATER DEMAND CONSTRAINTS, AND -1 IN SAME
C ROW OF BASIN IBE INDICATING EXPORT
C
       WRITE(17,2016)COL(14),BASIN(IB),BASIN(IBE),IT,N1,N2,
      &ROW(15),BASIN(IB),IT,ROW(15),BASIN(IBE),IT
C
C ENTER 1.0 IN TOTAL WATER DEMAND CONSTRAINTS , AND -1 IN SAME
C ROW OF BASIN IBE INDICATING EXPORT
C
       WRITE(17,2016)COL(14),BASIN(IB),BASIN(IBE),IT,N1,N2,
      &ROW(16),BASIN(IB),IT,ROW(16),BASIN(IBE),IT
  290 CONTINUE
  295 CONTINUE
  300 CONTINUE
  700 CONTINUE
C
C THIS IS THE END OF CONTINUOUS VARIABLE SECTION
C
      RETURN
      END
      SUBROUTINE RHS
C     ----------
C     ******************************************************
C     * III-C ):SETS UP THE RIGHT HAND SIDE OF MATRIX
C     ******************************************************
C
      COMMON/C01/ROW(17),COL(14),BASIN(10)
       COMMON/C03/NPROJ(10,10),NUWIP(10,10),NTWIP(10,10)
      COMMON/C05/OMR(10,10,10),IFLAG(10,10,10)
       COMMON/C06/IFUIP(10,10,10),LFUIP(10,10,10),YLDUIP(10,10,10)
```

```
      COMMON/C08/IFTIP(10,10,10),LFTIP(10,10,10),YLDTIP(10,10,10)
      COMMON/C12/NRMF(10),FLOWIN(10,10),FLOWOT(10,10),
     &NPLNR(10,10,10)
      COMMON/C14/AFC(10,10,10),AFCUIP(10,10,10),AFCTIP(10,10,10)
      COMMON/C16/DTWPPP(10,10),DUWPPP(10,10),TWDPPP(10,10)
      COMMON/C17/NBASIN,NPRODS,AMRATE,DELT,MNPDK,RATE,RTPRD
C
C III-C-1 ): WRITE RIGHT HAND SIDE MARKER
C
      WRITE(17,2000)
 2000 FORMAT('RHS')
C
C III-C-2 ): WRITE RHS FOR PROJECT LIMITATION CONSTRAINTS
C
      DO 150 IB=1,NBASIN
C
C III-C-2-a ): WRITE RHS FOR PROJECT LIMITATION CONSTRAINTS
C     FOR PROJECTS OF KIND K :   K=1 : DIVERSION DAMS
C                                K=2 : SURFACE RESERVOIRS
C                                K=3 : WELL FIELDS
C                                K=4 : DESALINIZATION PLANT
C                                K=5 : TREATMENT PLANTS
C
      DO 105 K=1,MNPDK
      IF(NPROJ(IB,K).EQ.0)GO TO 105
      NPIBK=NPROJ(IB,K)
      DO 100 N=1,NPIBK
      IF(IFLAG(IB,K,N).EQ.1)GO TO 100
      N1=N/10
      N2=N-N1*10
      WRITE(17,2001)ROW(K+7),BASIN(IB),N1,N2
 2001  FORMAT(4X,'RHS01',5X,A3,A1,2I1,10X,'1.0000')
  100 CONTINUE
  105 CONTINUE
C
C III-C-2-b ): WRITE RHS FOR PROJECT LIMITATION CONSTRAINTS
C  FOR UNTREATED-WATER PROJECTS:K=6: UNTREATED-WATER PROJECTS
C
      DO 115 IBE=1,NBASIN
      IF(IB.EQ.IBE.OR.NUWIP(IB,IBE).EQ.0)GO TO 115
      NPIBK6=NUWIP(IB,IBE)
      DO 110 N=1,NPIBK6
      IF(IFUIP(IB,IBE,N).EQ.1)GO TO 110
      N1=N/10
      N2=N-N1*10
      WRITE(17,2002)ROW(13),BASIN(IB),BASIN(IBE),N1,N2
 2002  FORMAT(4X,'RHS01',5X,A3,2A1,2I1,9X,'1.0000')
  110 CONTINUE
  115 CONTINUE
C
C III-C-2-c ): WRITE RHS FOR LIMITATION CONSTRAINTS
C FOR TREATED WATER PROJECTS : K=7 : TREATED-WATER PROJECTS
C
      DO 125 IBE=1,NBASIN
```

```
      IF(IB.EQ.IBE.OR.NTWIP(IB,IBE).EQ.0)GO TO 125
      NPIBK7=NTWIP(IB,IBE)
      DO 120 N=1,NPIBK7
      IF(IFTIP(IB,IBE,N).EQ.1)GO TO 120
      N1=N/10
      N2=N-N1*10
      WRITE(17,2002)ROW(14),BASIN(IB),BASIN(IBE),N1,N2
  120 CONTINUE
  125 CONTINUE
C
C III-C-3 ): WRITE RHS FOR TREATED-WATER DEMAND CONSTRAINTS
C
      DO 130 IT=1,NPRODS
      WRITE(17,2003)ROW(15),BASIN(IB),IT,DTWPPP(IB,IT)
 2003  FORMAT(4X,'RHS01',5X,A3,A1,I1,5X,F12.2)
  130 CONTINUE
C
C III-C-4 ): WRITE RHS FOR TOTAL WATER DEMAND CONSTRAINTS
C
      DO 135 IT=1,NPRODS
      WRITE(17,2003)ROW(16),BASIN(IB),IT,TWDPPP(IB,IT)
  135 CONTINUE
C
C III-C-5 ): WRITE RHS FOR MINIMUM STREAM FLOW REQUIREMENT
C           CONSRTAINTS
C
      NRMFIB=NRMF(IB)
      IF(NRMFIB.EQ.0)GO TO 150
      DO 145 NR=1,NRMFIB
      DELFL=FLOWIN(IB,NR)-FLOWOT(IB,NR)
      DO 140 IT=1,NPRODS
      WRITE(17,2004)ROW(17),BASIN(IB),IT,NR,DELFL
 2004  FORMAT(4X,'RHS01',5X,A3,A1,2I1,4X,F12.2)
  140 CONTINUE
  145 CONTINUE
  150 CONTINUE
      RETURN
      END
      SUBROUTINE BOUNDS
C     ----------------
C     ***************************************************
C     * III-D): WRITES BOUNDS ON DIFFERENT VARIABLES
C     ***************************************************
C
      COMMON/C01/ROW(17),COL(14),BASIN(10)
      COMMON/C03/NPROJ(10,10),NUWIP(10,10),NTWIP(10,10)
      COMMON/C05/OMR(10,10,10),IFLAG(10,10,10)
      COMMON/C06/IFUIP(10,10,10),LFUIP(10,10,10),YLDUIP(10,10,10)
      COMMON/C07/CCUIP(10,10,10),OMUIP(10,10,10),CUWIP(10,10,10)
      COMMON/C08/IFTIP(10,10,10),LFTIP(10,10,10),YLDTIP(10,10,10)
      COMMON/C09/CCTIP(10,10,10),OMTIP(10,10,10),CTWIP(10,10,10)
      COMMON/C13/TCAP(10),CAP(10,10,10)
      COMMON/C17/NBASIN,NPRODS,AMRATE,DELT,MNPDK,RATE,RTPRD
C
```

```
C III-D-1): WRITE BOUNDS MARKER CARD
C
      WRITE(17,2000)
 2000 FORMAT('BOUNDS')
C
C III-D-2): WRITE BOUNDS ON WITHDRAWALS FROM PROJECTS OF KIND K:
C                  K=1 : DIVERSION DAMS
C                  K=2 : SURFACE RESERVOIRS
C                  K=3 : WELLFIELDS
C                  K=4 :DESALINIZATION PLANTS
C                  K=5 :TREATMENT PLANTS
C
      DO 115 IB=1,NBASIN
      DO 110 K=1,MNPDK
      IF(NPROJ(IB,K).EQ.0)GO TO 110
       NPIBK=NPROJ(IB,K)
      DO 105 IT=1,NPRODS
      DO 100 N=1,NPIBK
      N1=N/10
      N2=N-N1*10
       WRITE(17,2005)COL(K+7),BASIN(IB),IT,N1,N2,CAP(IB,K,N)
 2005  FORMAT(1X,'UP',1X,'BND01',5X,A3,A1,3I1,3X,F12.2)
  100 CONTINUE
  105 CONTINUE
  110 CONTINUE
  115 CONTINUE
C
C III-D-3 ): WRITE BOUNDS ON UNTREATED-WATER TRANSPORT PROJECTS:
C          K=6 : UNTREAED-WATER IMPORT PROJECTS
C
      DO 135 IBI=1,NBASIN
      DO 130 IBE=1,NBASIN
      IF(IBI.EQ.IBE.OR.NUWIP(IBI,IBE).EQ.0)GO TO 130
      NUIPIB=NUWIP(IBI,IBE)
      DO 125 IT=1,NPRODS
      DO 120 N=1,NUIPIB
      N1=N/10
      N2=N-N1*10
       WRITE(17,2010)COL(13),BASIN(IBI),BASIN(IBE),IT,N1,N2,
     &CUWIP(IBI,IBE,N)
 2010  FORMAT(1X,'UP',1X,'BND01',5X,A3,2A1,3I1,2X,F12.2)
  120 CONTINUE
  125 CONTINUE
  130 CONTINUE
  135 CONTINUE
C
C III-D-4 ): WRITE BOUNDS ON TREATED-WATER TRANSPORT PROJECTS:
C                K=7 : TREATED-WATER IMPORT PROJECTS
C
      DO 155 IBI=1,NBASIN
      DO 150 IBE=1,NBASIN
      IF(IBI.EQ.IBE.OR.NTWIP(IBI,IBE).EQ.0)GO TO 150
      NPIBI7=NTWIP(IBI,IBE)
      DO 145 IT=1,NPRODS
```

```
      DO 140 N=1,NPIBI7
      N1=N/10
      N2=N-N1*10
       WRITE(17,2010)COL(14),BASIN(IBI),BASIN(IBE),IT,N1,N2,
     &CTWIP(IBI,IBE,N)
  140 CONTINUE
  145 CONTINUE
  150 CONTINUE
  155 CONTINUE
C
C III-D-5 ): WRITE BOUNDS ON DECISION VARIABLES FOR PROJECTS OF
C            KIND K=1,2,3,4,5,6,7
C
      DO 210 IB=1,NBASIN
C
C III-D-5-a ): BOUNDS ON PROJECTS OF KIND K=1,5
C
      DO 170 K=1,MNPDK
      IF(NPROJ(IB,K).EQ.0)GO TO 170
       NPIBK=NPROJ(IB,K)
      DO 165 IT=1,NPRODS
      DO 160 N=1,NPIBK
C
C IF PROJECT ALREADY EXIST BY-PASS THIS STEP
C
       IF(IFLAG(IB,K,N).EQ.1)GO TO 160
       N1=N/10
       N2=N-N1*10
       WRITE(17,2015)COL(K),BASIN(IB),IT,N1,N2
  2015  FORMAT(1X,'UP',1X,'BND01',5X,A3,A1,3I1,9X,'1.0000')
  160 CONTINUE
  165 CONTINUE
  170 CONTINUE
C
C III-D-5-b ):BOUNDS ON UNTREATED-WATER TRANSPORT PROJECTS:K=6
C
      DO 190 IBE=1,NBASIN
       IF(IB.EQ.IBE.OR.NUWIP(IB,IBE).EQ.0)GO TO 190
       NPIB6=NUWIP(IB,IBE)
      DO 185 IT=1,NPRODS
      DO 180 N=1,NPIB6
C
C IF PROJECT ALREADY EXIST BY_PASS THIS STEP
C
       IF(IFUIP(IB,IBE,N).EQ.1)GO TO 180
       N1=N/10
       N2=N-N1*10
       WRITE(17,2020)COL(6),BASIN(IB),BASIN(IBE),IT,N1,N2
  2020  FORMAT(1X,'UP',1X,'BND01',5X,A3,2A1,3I1,8X,'1.0000')
  180 CONTINUE
  185 CONTINUE
  190 CONTINUE
C
C III-D-5-c): BOUNDS ON TREATED-WATER TRANSPORT PROJECTS:K=7
```

```
C
      DO 205 IBE=1,NBASIN
      IF(IB.EQ.IBE.OR.NTWIP(IB,IBE).EQ.0)GO TO 205
      NPIB7=NTWIP(IB,IBE)
      DO 200 IT=1,NPRODS
      DO 195 N=1,NPIB7
C
C IF PROJECT EXISTS BY_PASS THIS STEP
C
      IF(IFTIP(IB,IBE,N).EQ.1)GO TO 195
      N1=N/10
      N2=N-N1*10
      WRITE(17,2020)COL(7),BASIN(IB),BASIN(IBE),IT,N1,N2
  195 CONTINUE
  200 CONTINUE
  205 CONTINUE
  210 CONTINUE
      RETURN
      END
      SUBROUTINE ANCOST(IB,K,N,IBE)
C     ----------------------------
C     ******************************************************
C     * CALCULATES ANNUAL FIXED COST OF PROJECT
C     * KINDS 1-7
C     ******************************************************
C
      COMMON/C04/LIFE(10,10,10),YIELD(10,10,10),CC(10,10,10)
      COMMON/C06/IFUIP(10,10,10),LFUIP(10,10,10),YLDUIP(10,10,10)
      COMMON/C07/CCUIP(10,10,10),OMUIP(10,10,10),CUWIP(10,10,10)
      COMMON/C08/IFTIP(10,10,10),LFTIP(10,10,10),YLDTIP(10,10,10)
      COMMON/C09/CCTIP(10,10,10),OMTIP(10,10,10),CTWIP(10,10,10)
      COMMON/C14/AFC(10,10,10),AFCUIP(10,10,10),AFCTIP(10,10,10)
      COMMON/C17/NBASIN,NPRODS,AMRATE,DELT,MNPDK,RATE,RTPRD
C
      IF(K.EQ.6)GO TO 100
      IF(K.EQ.7)GO TO 105
      L=LIFE(IB,K,N)
      GO TO 110
  100 L=LFUIP(IB,IBE,N)
      GO TO 110
  105 L=LFTIP(IB,IBE,N)
  110 CONTINUE
      UCOST=(AMRATE*(1.+AMRATE)**L)/((1.+AMRATE)**L-1)
      IF(K.EQ.6)GO TO 115
      IF(K.EQ.7)GO TO 120
      AFC(IB,K,N)=UCOST*CC(IB,K,N)
      GO TO 125
  115 AFCUIP(IB,IBE,N)=UCOST*CCUIP(IB,IBE,N)
      GO TO 125
  120 AFCTIP(IB,IBE,N)=UCOST*CCTIP(IB,IBE,N)
  125 CONTINUE
      RETURN
      END
      SUBROUTINE SCALER
```

```
C   ----------------
C   ***************************************************
C   * PROGRAM SCALER
C   * PURPOSE :
C   * THIS PROGRAM WILL TAKE AS INPUT THE MATRIX GENERATED BY
C   * PLAN_MOD PROGRAM AND WILL SCALE SUCH MATRIX SO THAT
C   * IT CAN BE USED BY LINDO WITHOUT SCALING ERRORS. THE
C   * ROW,COLUMN, AND OBJECTIVE SCALING FACTORS ARE
C   * TO BE PROVIDED BY THE USER. AN INTERMEDIATE FILE IS
C   * PRODUCED BUT IT CAN BE DISCARDED.
C   * INPUT :
C   *      MATRIX GENERATED BY PLAN_MOD CALLED :
C   *                PLAN_MOD_OUT_MATRIX
C   * OUTPUT :
C   *      SCALED MATRIX CALLED :
C   *                PLAN_MOD_SCALED_MATRIX
C   * PROGRAMMED BY: ERNESTO A. PEREZ
C   *                JAN 28 1985
C   ***************************************************
C
      CHARACTER*32 FILE17, FILE18, FILE19
      CHARACTER*8 FLAGGED(100)
      CHARACTER*1 ANSWER
      LOGICAL    FOUND
      INTEGER WHERE
      CHARACTER REST*26,LINE*62,MARKER*8,TEMP*14
       INTEGER CSF,ORSF,RSF,LOOPER,LAST,I
      REAL*8 NUMBER
      WRITE(*,*)'DO YOU WISH TO SCALE THE MATRIX ?'
      READ(*,'(A1)')ANSWER
       IF(ANSWER.EQ.'N')RETURN
C
C OPEN FILES
C
      FILE17='PLAN_MOD_OUT_MATRIX'
       FILE18='MATRIX_SCRATCH'
      FILE19='PLAN_MOD_SCALED_MATRIX'
        OPEN(17,FILE=FILE17,IOINTENT='INPUT',STATUS='OLD',
     &PAD='YES')
        OPEN(18,FILE=FILE18,IOINTENT='OUTPUT',STATUS='FRESH',
     &PAD='YES',RECFM='DS')
        OPEN(19,FILE=FILE19,IOINTENT='OUTPUT',STATUS='FRESH',
     &PAD='YES',RECFM='DS')
C
      PRINT*,'ENTER COLUMN SCALING FACTOR (CSF). FOR EXAMPLE',
     &' PROBLEM'
       PRINT*,'A CSF=10000 IS APPROPRIATE. CSF=?'
      READ(*,*) CSF
       PRINT*,'ENTER ROW SCALING FACTOR (RSF). FOR EXAMPLE',
     &' PROBLEM'
       PRINT*,'A RSF=10000 IS APPROPRIATE. RSF=?'
      READ(*,*) RSF
       PRINT*,'ENTER OBJECTIVE ROW SCALING FACTOR (ORSF). FOR',
     &' EXAMPLE'
```

```
      PRINT*,'PROBLEM A ORSF=1000000 IS APPROPRIATE. ORSF='
      READ(*,*) ORSF
C
C READ UNTIL COL CARD IS FOUND
C
   10 READ(17,20) LINE
   20 FORMAT(A62)
      IF(LINE(1:3).EQ.'COL')GO TO 30
      WRITE(18,20)LINE
      GO TO 10
   30 WRITE(18,20)LINE
   35 READ(17,40)TEMP,MARKER,NUMBER,REST
   40 FORMAT(A14,A8,F14.2,A26)
      IF(TEMP(1:3).EQ.'RHS')GO TO 60
      IF(TEMP(5:5).NE.'Q')GO TO 50
C
C A COLUMN VARIABLE IS FOUND : SCALE THE COEFFICIENT BY CSF
C
      NUMBER = NUMBER*CSF
   50 WRITE(18,40)TEMP,MARKER,NUMBER,REST
      GO TO 35
   60 WRITE(18,40)TEMP
   65 READ(17,20,END=70) LINE
      WRITE(18,20) LINE
      GO TO 65
   70 CLOSE(18)
      CLOSE (17)
       OPEN(18,FILE=FILE18,IOINTENT='INPUT',STATUS='OLD',
      &PAD='YES')
      LOOPER=0
      LAST=0
      DO 71 I=1,100
      FLAGGED(I)='        '
   71 CONTINUE
   80 READ(18,20)LINE
      IF(LINE(1:3).EQ.'COL')GO TO 90
      WRITE(19,20)LINE
      GO TO 80
   90 WRITE(19,20)LINE
   95 READ(18,100)TEMP,MARKER,NUMBER,REST
      IF(TEMP(1:3).EQ.'RHS')GO TO 130
      IF(MARKER.EQ.'MINCOST ')GO TO 120
  100 FORMAT(A14,A8,F14.5,A26)
      IF((ABS(NUMBER-1.00000)).GT.0.0005)GO TO 105
      LOOPER=LOOPER+1
      FLAGGED(LOOPER)=MARKER
      LAST=LOOPER
  103 WRITE(19,100)TEMP,MARKER,NUMBER,REST
      GO TO 95
  105 NUMBER=NUMBER/(FLOAT(RSF))
      FOUND = .FALSE.
      DO 106 I=1,LAST
      IF(MARKER.EQ.FLAGGED(I))THEN
          FOUND = .TRUE.
```

```
      WHERE = I
    END IF
106 CONTINUE
    IF(FOUND) FLAGGED(WHERE) = '       '
    GO TO 103
120 NUMBER=NUMBER/(FLOAT(ORSF))
    GO TO 103
130 WRITE(19,100)TEMP,MARKER,NUMBER,REST
131 READ(18,100) TEMP,MARKER,NUMBER,REST
    IF(TEMP(1:3).EQ.'BOU')GO TO 150
    FOUND=.FALSE.
    DO 132 I=1,LAST
    IF(MARKER.EQ.FLAGGED(I)) FOUND=.TRUE.
132 CONTINUE
    IF(FOUND)GO TO 133
    NUMBER =NUMBER/(FLOAT(RSF))
133 WRITE(19,100)TEMP,MARKER,NUMBER,REST
    GO TO 131
150 WRITE(19,100)TEMP,MARKER,NUMBER,REST
155 READ(18,100)TEMP,MARKER,NUMBER,REST
    IF(TEMP(1:3).EQ.'END')GO TO 160
    IF(MARKER(1:1).NE.'Q')GO TO 156
    NUMBER=NUMBER/(FLOAT(CSF))
156 WRITE(19,100)TEMP,MARKER,NUMBER,REST
    GO TO 155
160 WRITE(19,100)TEMP
    CLOSE (19)
    CLOSE (18)
    RETURN
    END
```