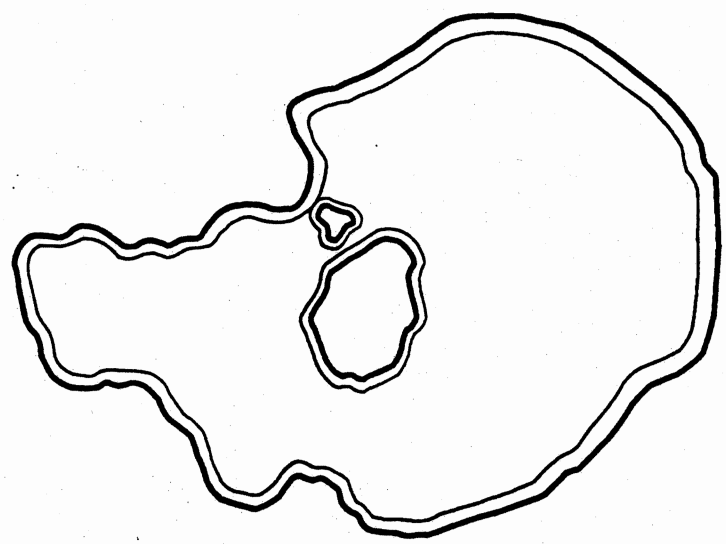
An Auxiliary Report
Prepared for the

Mono Basin Water Rights EIR

LAAMP (Los Angeles Aqueduct Monthly Program)
Documentation, Version 2



Prepared under the Direction of:

California State Water
Resources Control Board
Division of Water Rights
P.O. Box 2000
Sacramento, CA 95810

Prepared With Funding from:

Los Angeles Department of
Water and Power
Aqueduct Division
P.O. Box 111
Los Angeles, CA 90051

Mono Basin EIR Auxiliary Report No. 5

An Auxiliary Report
Prepared for the
Mono Basin Water Rights EIR Project

This auxiliary report was prepared to support the environmental impact report (EIR) on the amendment of appropriative water rights for water diversions by the City of Los Angeles Department of Water and Power (LADWP) in the Mono Lake Basin. Jones & Stokes Associates is preparing the EIR under the technical direction of the California State Water Resources Control Board (SWRCB). EIR preparation is funded by LADWP.

SWRCB is considering revisions to LADWP's appropriative water rights on four streams tributary to Mono Lake, Lee Vining Creek, Rush Creek, Parker Creek, and Walker Creek. LADWP has diverted water from these creeks since 1941 for power generation and municipal water supply. Since the diversions began, the water level in Mono Lake has fallen by 40 feet.

The Mono Basin water rights EIR examines the environmental effects of maintaining Mono Lake at various elevations and the effects of possible reduced diversions of water from Mono Basin to Owens Valley and the City of Los Angeles. Flows in the four tributary creeks to Mono Lake and water levels in Mono Lake are interrelated. SWRCB's decision on amendments to LADWP's water rights will consider both minimum streamflows to maintain fish populations in good condition and minimum lake levels to protect public trust values.

This report is one of a series of auxiliary reports for the EIR prepared by subcontractors to Jones & Stokes Associates, the EIR consultant, and contractors to LADWP. Information and data presented in these auxiliary reports are used by Jones & Stokes Associates and SWRCB, the EIR lead agency, in describing environmental conditions and conducting the impact analyses for the EIR. Information from these reports used in the EIR is subject to interpretation and integration with other information by Jones & Stokes Associates and SWRCB in preparing the EIR.

The information and conclusions presented in this auxiliary report are solely the responsibility of the author.

Copies of this auxiliary report may be obtained at the cost of reproduction by writing to Jim Canaday, Environmental Specialist, State Water Resources Control Board, Division of Water Rights, P.O. Box 2000, Sacramento, CA 95810.

LAAMP (Los Angeles Aqueduct Monthly Program) Documentation Version 2

Prepared for:

Jones and Stokes Associates

Prepared by:

Luhdorff & Scalmanini
Consulting Engineers
Woodland, CA

January, 1992

LSCE 90-1-082

Contents

Contents (Cont.)

Contents (Cont.)

List of Tables

List of Figures

I. Introduction

Background

At the request of Jones and Stokes Associates (JSA) and the State Water Resources Control Board (SWRCB), Luhdorff and Scalmanini, Consulting Engineers (LSCE) have modified and enhanced a FORTRAN program (LAAMP, or Los Angeles Aqueduct Monthly Program) that simulates the monthly operation of the Los Angeles Aqueduct from Lee Vining Creek in Mono County to below Bouquet Reservoir in Los Angeles County. The original program was developed by staff of the Aqueduct Division of the Los Angeles Department of Water and Power (LADWP) as part of an overall effort to prepare an Environmental Impact Report on Mono Basin water rights amendments.

The documentation the initial version of LAAMP was distributed in July, 1991. The comments received from various agencies and interest groups were used as the basis for improving and modifying LAAMP. This documentation describes the updated version of LAAMP that was used to simulate LADWP aqueduct operations for each alternative water rights amendment analyzed in the Mono Basin EIR.

LAAMP simulates the basic operation of the aqueduct system in terms of managing available water resources by diverting surface water and pumping groundwater for the purposes of local use (primarily irrigation, recreation, wildlife and environmental enhancement), storing runoff in either surface reservoirs or groundwater basins, or exporting water to Los Angeles. In addition, the program tracks and "spills" excess surface water that cannot be transported or stored in the aqueduct system.

Objectives

The work undertaken by JSA and LSCE to develop and test LAAMP had four major objectives:

1) Develop a clear and concise conceptualization of the aqueduct system that can be applied under various alternative management scenarios.

2) Develop a user interface that allows for input and output in a "spreadsheet" environment. This interface will facilitate the development of alternatives, and the evaluation of impacts and mitigations.

3) Develop a single FORTRAN program that can be used to simulate the entire system, and develop documentation of the program.

4) Test the program with various inputs to demonstrate its validity under the range of conditions that are to be considered in the EIR alternatives.

Work associated with meeting the four objectives was divided between JSA and LSCE. JSA took the lead on items 2 and 4, while LSCE took the lead on items 1 and 3. The user interface and the testing of the program are the subject of separate reports prepared by JSA. This report documents the work associated with the development of a conceptualization of the aqueduct system, and the FORTRAN program itself. As part of the fourth objective, it was necessary to modify LAAMP in order to constrain groundwater pumping in the Owens Valley. This was accomplished through development of the program ALAAMP, which is also documented in this report.

This model documentation does not contain guidelines regarding the most appropriate input data for simulations, nor does it support any management philosophy. It simply documents the simulation program, its features, flexibility, assumptions, and possible limitations. While some of the features of the program may be viewed as unnecessary for particular applications, users have almost complete flexibility to simulate any set of inputs and corresponding aqueduct management philosophy. With appropriate specification of input parameters, LAAMP can simulate a wide range of management possibilities from maximum export to Los Angeles every year with little or no regard to in-basin uses or environmental protection, to maximum environmental protection with little or no regard to the amount of water exported to Los Angeles. In reality, these two extreme approaches need to be balanced through the appropriate and realistic selection of operations rules and input parameters. The LAAMP simulation results, can be used to help determine a balanced management approach. The selection of input parameters is more fully discussed in JSA documentation of the user interface or "spreadsheets".

II. Description of The Aqueduct System

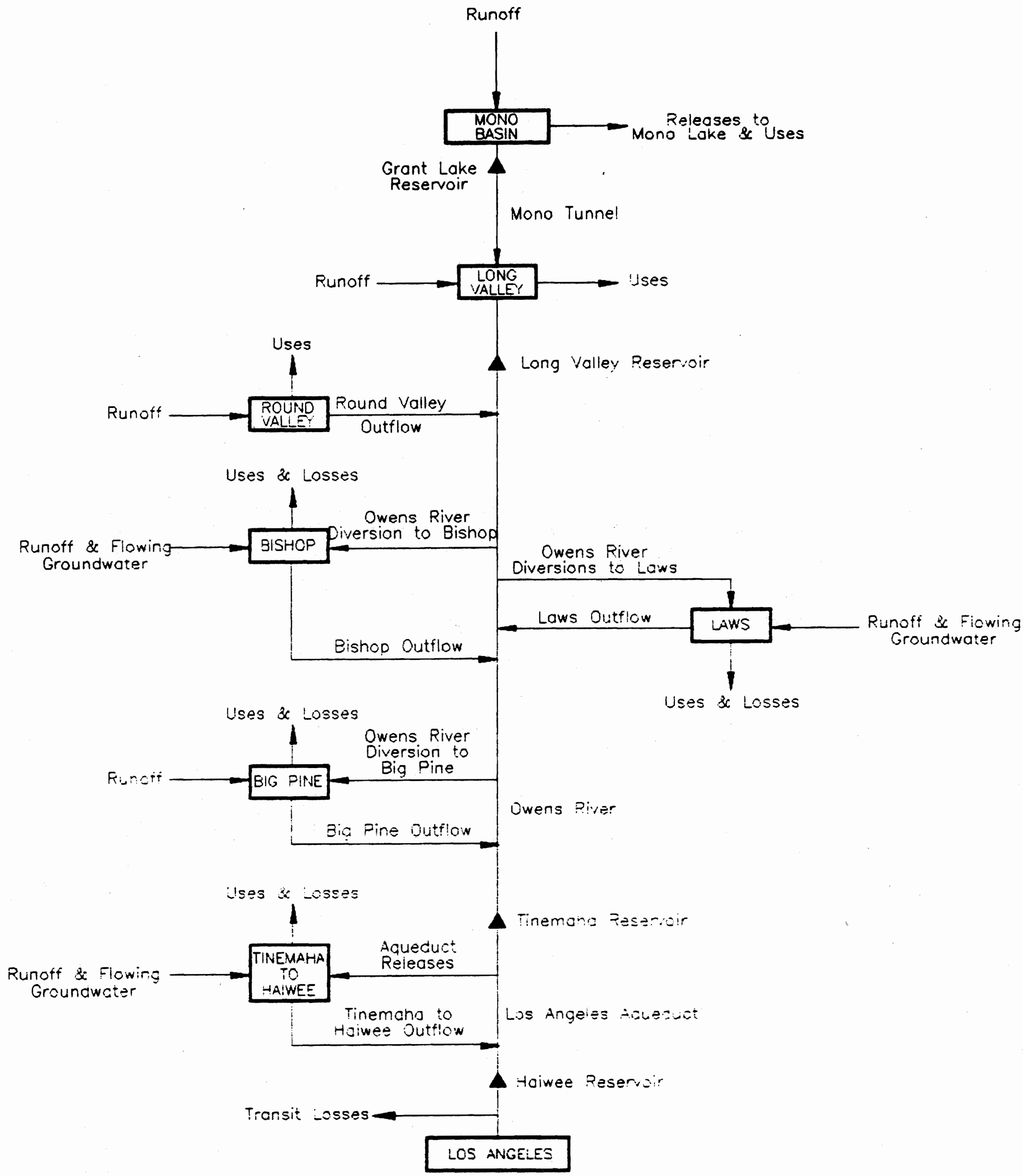
Background

The Los Angeles Aqueduct system consists of surface and ground-water facilities that were built to capture and deliver high quality water and electrical energy to the City of Los Angeles. The facilities of the aqueduct system consist of diversion structures, open canals, stream courses, closed conduits, tunnels, wells, reservoirs, spreading basins, and hydroelectric generating facilities. All water supplying the aqueduct system is exported from Mono and Inyo Counties. This description of the aqueduct system is limited to the Mono County and Inyo County portions of the aqueduct. A schematic diagram of the system is presented in Figure 1.

Mono Basin

The major component of water supply to the aqueduct is surface runoff. The northernmost streams that can be diverted into the aqueduct system are located in the Mono Basin. These streams are Lee Vining, Walker, Parker, and Rush Creeks, all of which are tributaries to Mono Lake. Lee Vining, Walker, and Parker Creeks can be diverted into a closed conduit that empties into Grant Lake Reservoir in the June Lake Loop area. Some water from these creeks has been used for irrigation of Los Angeles-owned land in the Mono Basin. Water can be released into the natural stream beds of these three creeks, either as an intentional release (e.g. minimum flow releases) or in response to high runoff conditions when flow exceeds the capacity of the conduit. Rush Creek flows directly into Grant Lake Reservoir.

The water from Grant Lake can either be released into Rush Creek to flow into Mono Lake



Aqueduct System Conceptualization

FIGURE 1

Luhdorff & Scalmanini Consulting Engineers

Jones & Stokes Associates
Mono Lake EIR

90-1-082
JAN. 1992

hydroelectric facilities on its way to Los Angeles. Boquet Reservoir provides emergency storage and was not included in LAAMP. A transit loss occurs between Haiwee and Los Angeles.

III. Conceptualization of Modeled System

The brief description of the conceptualized aqueduct system in this section is intended to introduce the underlying theme of the simulation program: the primary objective of the Los Angeles Aqueduct is to export water from the eastern Sierra to Los Angeles, and all available water is exported up to the user specified export demand. In other words, the user, through specification of local "use" and release parameters, can achieve nearly any feasible result in terms of export to Los Angeles. Viewed from a different perspective, the user can achieve any desired result in terms of environmental protection through the specification of "use" parameters. Excess runoff is stored in reservoirs as ground-water recharge. Storage releases and ground-water pumping are used to augment low runoff. The details of the approach and the assumptions needed to implement this conceptualization are described in the following documentation of the program. Prior to that detailed discussion, the overall conceptualization is discussed.

LAAMP operates on a fairly simple water management concept: after all local demands and regulatory releases are met (the individual demands and releases are user specified outside of the program code to facilitate flexibility), all water available to the aqueduct system on a monthly basis is either exported, stored, spread (ground-water recharge), or spilled. If the amount of available water is less than the desired export target for the month, the deficit is reduced or eliminated by exporting water from the Mono Basin, releasing surface storage, pumping Owens Valley groundwater, and reducing uses. If, on the other hand, the amount of available water is greater than the desired export target for the month, the excess water can be exported (at aqueduct capacity) stored in surface reservoirs, diverted and spread to enhance groundwater recharge, and spilled from the aqueduct system to Mono Lake or Owens Lake.

The specification of the desired export target, the comparison of available water to desired export, and the process of making up an export deficit or allocating excess water is made monthly. Regulatory releases and other uses are generally specified on a monthly basis, except for the specification of an annual target release for Mono Lake which is made at the beginning of the runoff year (April 1) and is based on user specified lake level triggers.

An important feature of the program is the inclusion of certain system constraints, either physical (i.e. conduit, canal, and well field capacities), or regulatory (e.g. Upper Owens River flooding constraint, Rock Creek minimum flows, Pleasant Valley outflow minimum, groundwater pumping limits). These monthly constraints govern the possible management of water in the eastern Sierra, and are factors limiting the feasible LAAMP simulations.

IV. General Description of LAAMP

LAAMP consists of a main program and 17 subroutines (listed in Table 1). The main program acts as the control for the annual and monthly looping of the simulation, and all calls to subroutines. Each subroutine is described in detail following a description of the main program which contains brief descriptions of each subroutine. The descriptions provided are designed to be read with a copy of the source code ready for reference, a copy of which is presented in Appendix A.

The main program and all subroutines feature an INCLUDE command for the file "LAAMP.CMB". This file contains common blocks for array variables that are used throughout the simulation program. A copy of this file is also provided in Appendix A.

Array Dimensions and Time Series Management

All monthly "time series" arrays are dimensioned for runoff years 1940-41 to 1989-90 and for runoff months 1 (April) to 12 (March), as appropriate. An additional "month" (13) is included to represent the annual sum of all twelve months. These time period of the arrays correspond to the period of hydrologic data supplied by LADWP. The arrays are thus dimesioned (40:89,13).

Since it is assumed that these hydrologic input data would "drive" all future simulations, this scheme of array dimensions allows rapid and easy comparison of all calculated parameters to historic hydrology.

The program operates on a runoff year basis (month 1 = April, month 12 = March). The conversion for runoff year months and calendar year months is provided by the array

TABLE 1
Subroutine Listing

SUBROUTINE	DESCRIPTION
READBH	Reads bathymetry and hydrologic input data.
READPRN	Reads PRN file created by spreadsheet.
HYDCALC	Indexes years for fish releases, Mono Lake releases, storage targets, and export targets.
MLTAR	Calculates annual Mono Lake release target.
MBFLUSH	Calculates monthly minimum flow requirements in four Mono Basin creeks.
MBFISH	Calculates "actual" creek releases.
MLR	Calculates monthly lake releases.
MBIRRG	Calculates monthly Mono Basin irrgation releases.
CONDUIT	Calculates monthly Lee Vining Conduit flow.
WP	Calculates minimum West Portal flow, and amount available to West Portal.
OWENS	Establishes monthly Rock Creek Diversion, Long Valley inflow, outflow from Round Valley area; uses, losses, diversion pumping and outflow from Laws, Bishop, Big Pine, and Tinemaha to Haiwee areas.
TOOMUCH	Monthly allocation of water that is in excess of target export: store in Tinemha and Haiwee, pumping reduction, Long Valley storage increase, increase export, spreading in Laws, Big Pine, and Tinemaha to Haiwee, aqueduct releases.
NOTENUF	Adds water to system to meet export target (monthly): Tinemaha and Haiwee storage reduction, Mono Basin export, storage decrease, increase pumping, Owens Valley use reduction, export reduction.
GLVSTOR	Calculates Grant and Long Valley storage (end-of-month); "transfers" storage from Grant to Long Valley under certain conditions.

Table 1 (cont.)
Subroutine Listing

GRNPUMP	Checks year-to-date pumping with Green Book limits and modifies monthly pumping limits accordingly.
OUT1	Writes all output files
BATHY	Bathymetry routine for Grant and Long Valley reservoirs and Mono Lake.

variable IMORL. This conversion is made only for output to more conveniently identify calendar months and years.

V. Description of Main Program

The organization of the main program is presented in flowchart form on Figure 2. The INCLUDE statements for the files LAAMP.INT and LAAMP.DST are used to make the program more readable. These files include the initialization (to zero) of all arrays (LAAMP.INT) and data statements for certain arrays (LAAMP.DST). Copies of LAAMP.INT AND LAAMP.DST are presented in Appendix A.

Execution Prior to Entering Main Simulation Loop

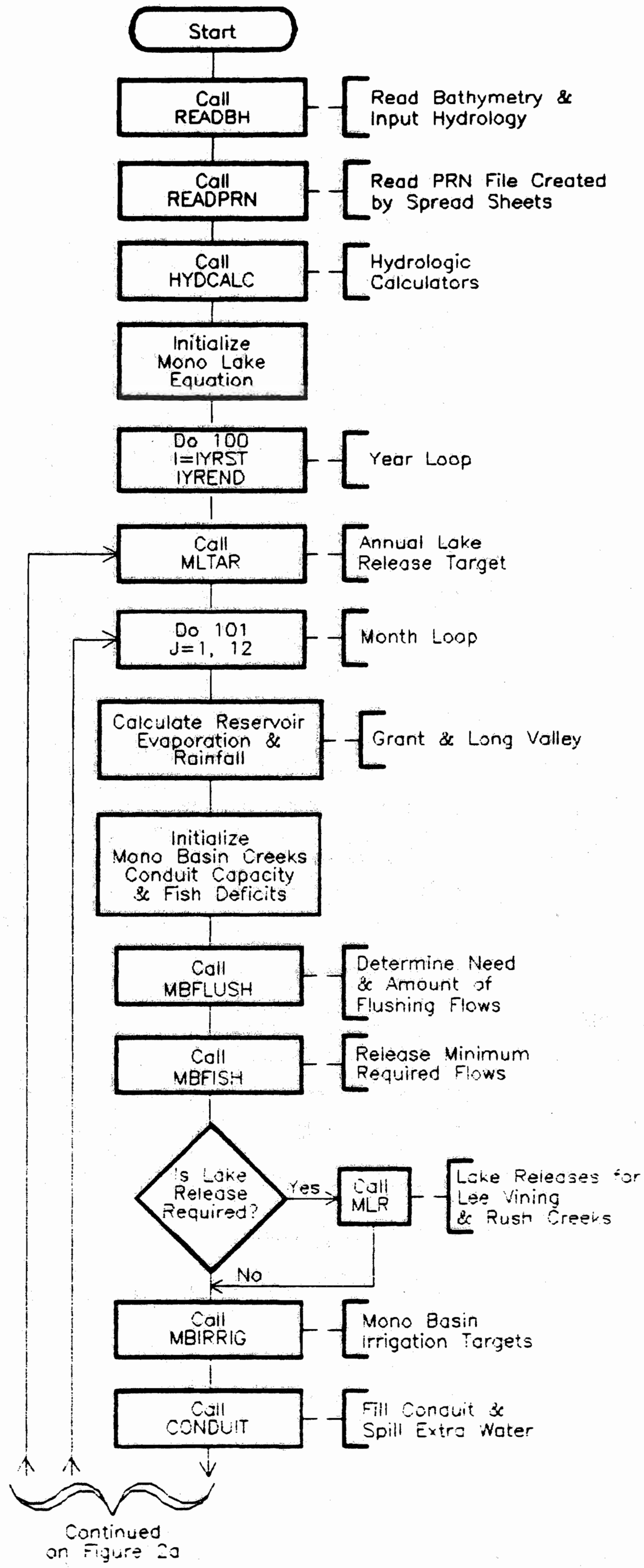
The first significant executable statements in the program call subroutines (READBH and READPRN) that reads all data needed for the simulation. The details of these subroutines are described below.

After all data are read, the program calls the subroutine that performs all hydrologic calculations (HYDCALC). This subroutine is described in detail below.

The regression equation that estimates unmeasured inflows to calculate Mono Lake elevation is then initialized based on the assumed annual evaporation rate of Mono Lake. The details of the calculation of Mono Lake elevations are described below.

Annual Loop Initialization and Runoff Year Calculations

The annual loop is initialized next. Note that the beginning and end of the years in the simulation are user specified with the available 50 year sequence of runoff years 1940-41 to 1989-90. Any number of years within the sequence can be simulated by specifying a beginning year and ending year. Note that a limitation of the program as currently written



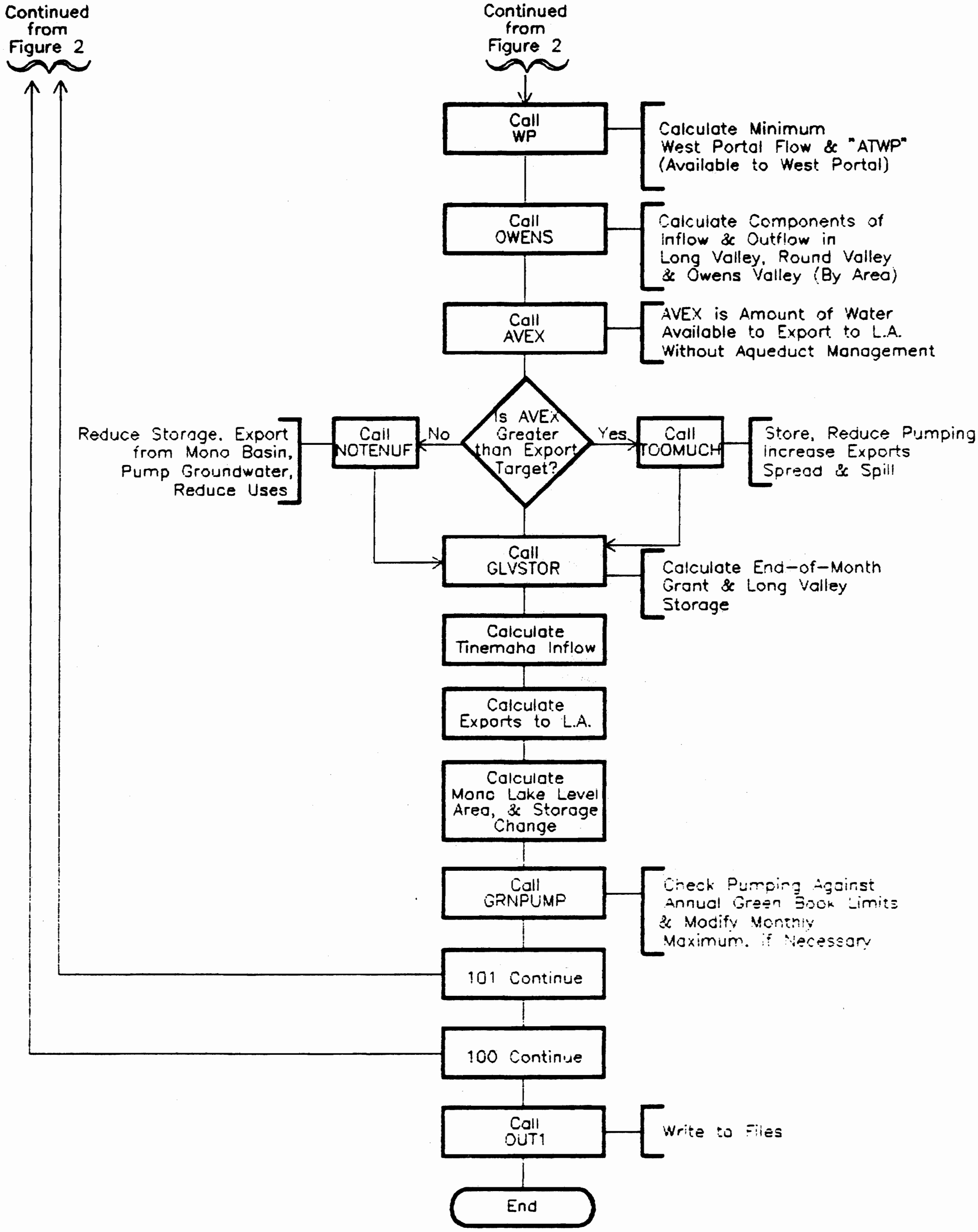
Flowchart of LAAMP Program

FIGURE 2

Luhdorff & Scalmanini Consulting Engineers

Jones & Stokes Associates
Mono Lake EIR

90-1-082
JAN. 1992



Flowchart of LAAMP Program

FIGURE 2a

Luhdorff & Scalmanini Consulting Engineers

Jones & Stokes Associates
Mono Lake EIR

90-1-082
JAN. 1992

is that only consecutive years in the hydrologic record from runoff years 1940-41 to 1989-90 can be simulated. Minor alterations to the code in the form of an additional subroutine that would "rearrange" the hydrologic record could easily be added, if necessary, to simulate a different sequence of hydrology.

The MLTAR subroutine, which calculates the annual Mono Lake release target is then called. The annual Mono Lake release target is calculated based on criteria that consider April 1 lake level and runoff year type. Three trigger elevations are specified. This lake level trigger matrix is user specified in JSA'a MONO spreadsheet, and is more fully described in the description of the subroutine READPRN. The target lake release is then corrected for anticipated minimum flow releases. Implementation of this annual target is more fully described in the description of the subroutine MLTAR. Note that if the lake level is higher than the highest level in the matrix, LRT is set equal to -1 which indicates that no release is needed.

Execution Within the Monthly Loop

The monthly loop is initialized in the next step. Since the program operates on a runoff year basis, month number 1 is April and month number 12 is March. The first calculations for a month involve estimating the storage charge from Grant Lake and Long Valley Reservoir due to precipitation and evaporation that is assumed to occur at the beginning of the month.

The simulation for a month proceeds in the following fashion: 1) make all user specified releases and meet all other "uses" for the Mono Basin and Owens Valley from runoff and pumping and Owens River diversions, 2) compare the amount of water available for export after meeting these "uses" to the user specified monthly export target, 3) either make up (or reduce) a deficit (CALL NOTENUF) or allocate excess water (CALL TOOMUCH) and calculate actual monthly export, 4) calculate end-of-month storages for Grant and Long Valley Reservoirs, Tinemaha inflow, and flow to the Los Angeles, 5) calculate flow to Los

Angeles, 6) calculate end-of-month Mono Lake level, and 7) check current year pumping with "Green Book" limits.

Of note in the storage calculation for Grant Lake is the ability to "move" storage from Grant to Long Valley if Grant storage is above a user defined maximum, if upper Owens River maximum flow is not exceeded, and if the Mono Lake elevation is not below the minimum target level.

Calculation of Mono Lake Elevation

The method used to calculate Mono Lake elevation is based on a monthly water balance model. The underlying concept of the water balance model involves the identification of "explained" and "unexplained" water balance from month to month. An initial version of the water balance model assumed that "explained" storage changes included 100% of the LADWP reported releases from the diverted streams (which do not include Mill and Dechambeau Creeks), and 100% of Cain Ranch precipitation over the surface area of the lake. The residual "unexplained" storage change was conceptualized to include evaporation as a function of surface area and other inflow that was assumed to be a function of the measured runoff from the diverted streams. To calculate "unexplained" storage change, the following monthly regression equations were solved:

USC = (RI*RO) + (EVAP*AREA)

where:

USC = "UNEXPLAINED" STORAGE CHANGE (AF)
RI = MONTHLY RUNOFF FRACTION
RO = MEASURED RUNOFF (AF)
EVAP = EVAPORATION (FT)
AREA = MONO LAKE SURFACE AREA (ACRES)

In this form, each month had unique coefficients for RI and EVAP. In addition, the evaporation terms summed to 36.22 inches per year, and the monthly distribution of evaporation (Table 2) matched well with pan data in the region. These regression equations could not differentiate a lower constant inflow from a higher monthly evaporation rate.

The predictions using this set of monthly equations were considered good, but additional analysis was necessary to further improve the model both conceptually and in terms of confidence in predictions.

A second version of the monthly water balance model, the one used in the current version of LAAMP, included the monthly evaporation as one of the "explained" storage change components. By including several alternate annual evaporation rates, and applying the monthly distribution of evaporation generated by the original model, new estimates of "explained" storage change were generated, one for each annual evaporation rate. The residual "unexplained" monthly storage change, were then regressed with a single equation as a function of measured monthly runoff.

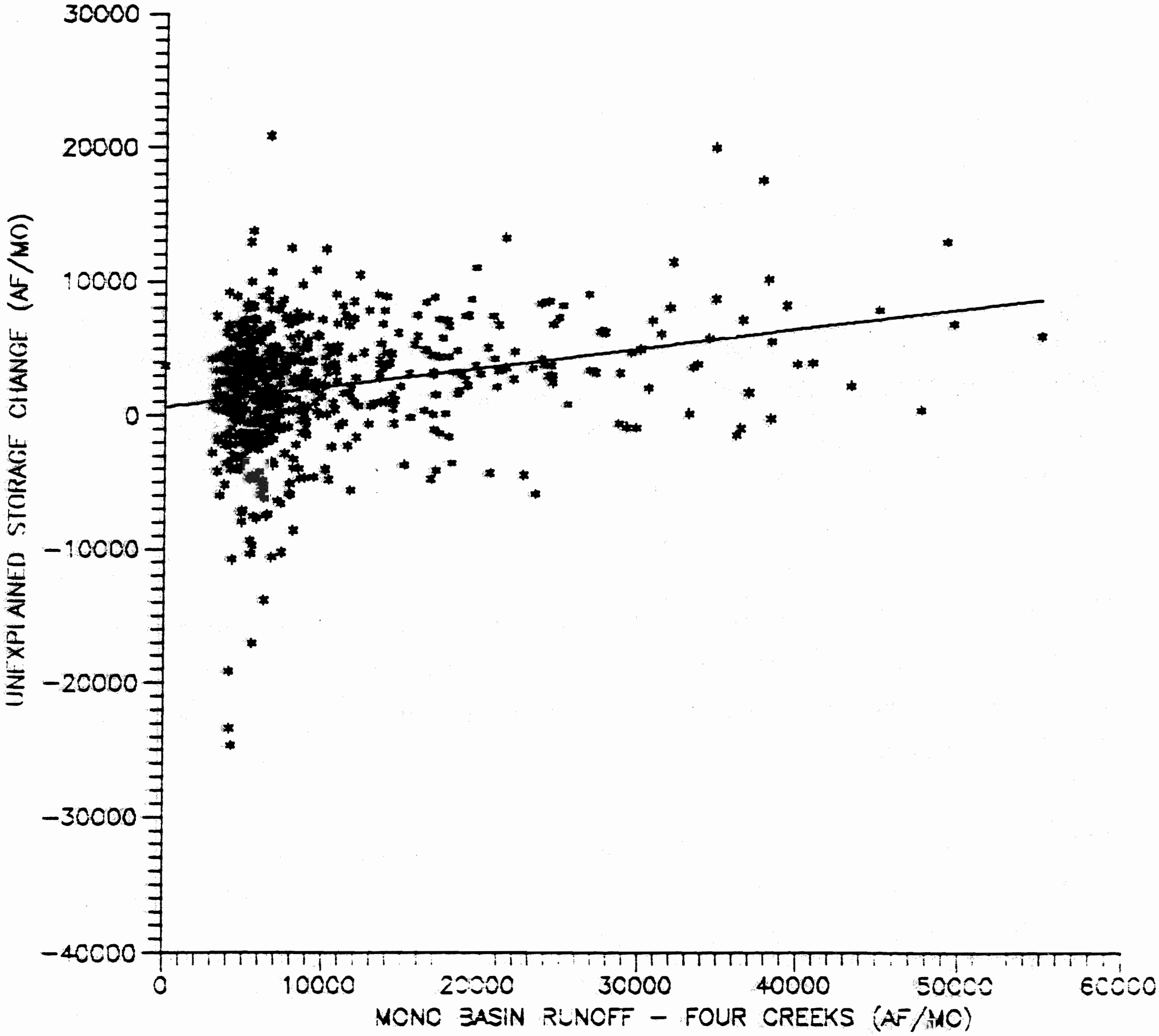
A scattergram with the line of best fit of runoff versus "unexplained" storage change for 38.83 inches of evaporation is presented in Figure 3. The period of analysis was from April 1941 to March 1990.

Note that in this version of the model all months are "lumped" together in order to generate a single storage change equation as opposed to the original version which used a different equation for each month. It was assumed that evaporation was the most significant and only quantifiable seasonal influence on the system. Attempts to generate monthly regression models resulted in greater prediction errors than the single model that represents all months.

The resultant model equation is in the following form:

TABLE 2
Calculated Monthly Mono Lake Evaporation Rates

Month	Evaporation (inches/month)	% of Total Annual
January	1.36	3.75
February	0.97	2.67
March	0.79	2.19
April	1.56	4.30
May	3.17	8.75
June	4.29	11.86
July	4.79	13.24
August	5.94	16.41
September	4.76	13.15
October	4.11	11.36
November	2.88	7.95
December	1.59	4.38
Total	36.22	100.00



Runoff vs. Unexplained Storage Change
38.83 Inches Annual Evaporation

FIGURE 3

Luhdorff & Scalmanini
Consulting Engineers

Jones & Stokes Associates
Mono Lake EIR

90-1-082
Jan. 1992

USC = YINT + (RI*RO)

where:

USC = "UNEXPLAINED" STORAGE CHANGE (AF)
YINT = CONSTANT
RI = MONTHLY RUNOFF INDEX
RO = MEASURED RUNOFF (AF)

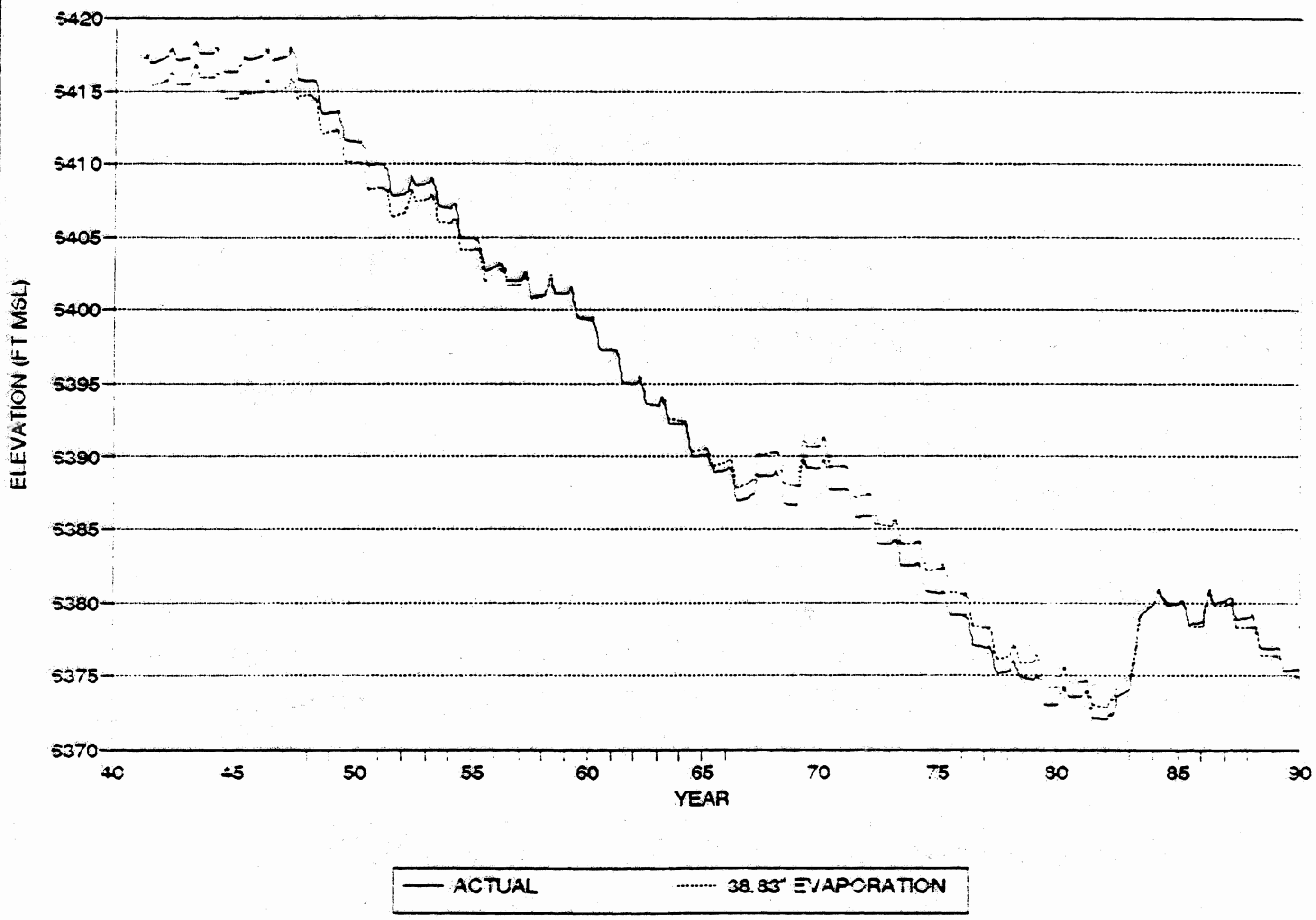
Values of YINT and RI are dependent upon the value of annual evaporation. Through regression analysis, the following equations were developed to generate appropriate values of YINT and RI for any given value of annual evaporation (EVAP) expressed in inches:

YINT = -8651.95 + (239.897*EVAP)
RI = -0.206935 + (0.00905776*EVAP)

The above equations to estimate YINT and RI were developed for evaporation values between 36 and 44 inches. A value of 38.83 inches yielded the smallest residual error, although any assumed evaporation rate between 36 and 44 inches provides a similar match with the historical record.

The model equations were then used with a April 1, 1941 Mono Lake elevation and historical precipitation and runoff data to simulate lake level fluctuations from 1941 to 1990. Results for 38.83 inches of evaporation are presented as Figure 4 in the form of a hydrograph that depicts actual lake levels and modeled lake levels. The error in the monthly estimates for several evaporation values are presented as Figure 5.

The average error for the entire time period for each evaporation rate was then evaluated (Figure 6) as annual evaporation versus average error. Note that an evaporation value of 38.83 inches yields an average error of zero.



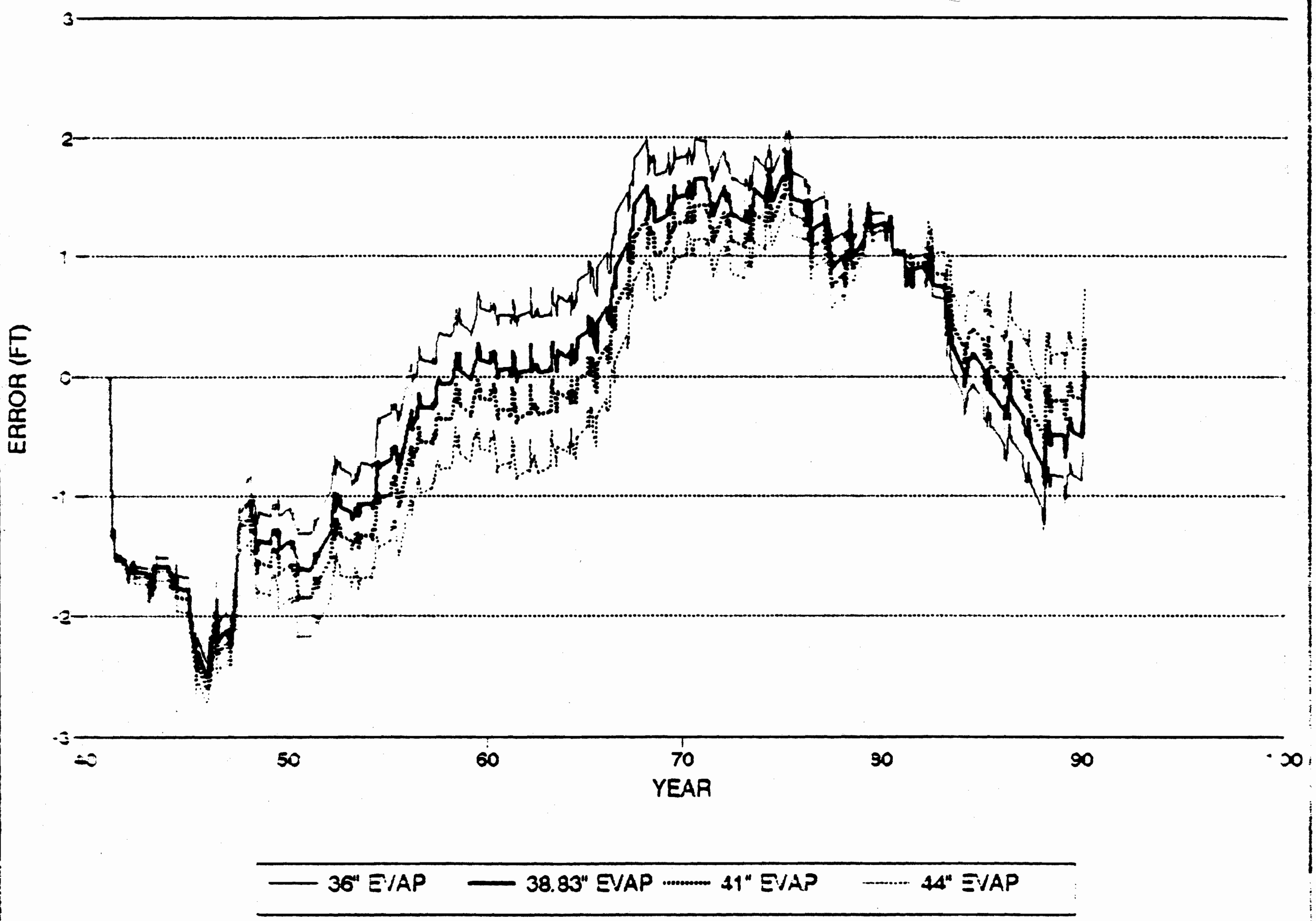
Mono Lake Elevation Comparison
38.83 Inches Annual Evaporation

FIGURE 4

Lundorff & Scalmanini Consulting Engineers

Jones & Stokes Associates
Mono Lake EIR

90-1-082
Jan. 1992



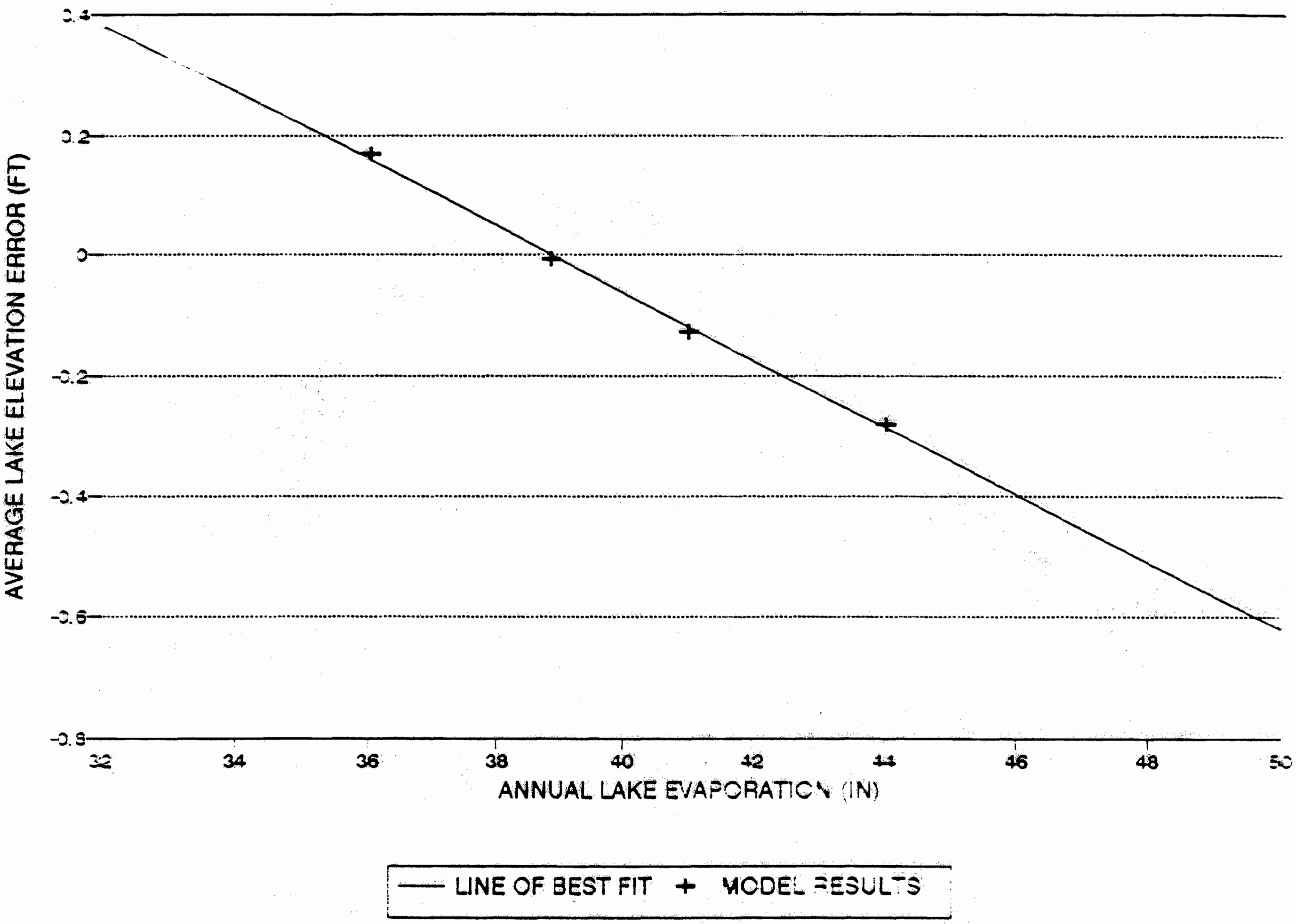
Mono Lake Elevation Estimation Errors

FIGURE 5

Luhdorff & Scalmanini Consulting Engineers

Jones & Stokes Associates
Mono Lake EIR

90-1-082
Jan. 1992



Mono Lake Evaporation Analysis

FIGURE 6

Lundorff & Scalmanini
Consulting Engineers

Jones & Stokes Associates
Mono Lake EIR

90-1-082
Jan. 1992

Execution at the End of a Simulation

The program then loops back to simulate another year, or, if the simulation is finished, the subroutine OUT1 are called. The output variables were developed jointly by LSCE and JSA such that the output from LAAMP can be imported into output spreadsheets developed by JSA for statistical and graphical analyses.

VI. Description of Subroutines READBH and READPRN

The three files that represent the input to any LAAMP simulation are read in these subroutines: 1) a bathymetry file for Grant and Long Valley Reservoirs, and Mono Lake, 2) a hydrology input file, and 3) a "spreadsheet" file that contains user specifications for all releases, "uses", parameters that fluctuate monthly in each year and constraints.

File BATHY.DAT

The file BATHY.DAT contains the bathymetry data (elevation, area and volume) for Grant and Long Valley Reservoirs, and Mono Lake.

File INPHYD.DAT

The file INPHYD.DAT contains all historical hydrologic data that "drive" a LAAMP simulation. The individual parameters contained in this file are listed in both FORTRAN variable name and descriptive name in Table 3.

File LAAMP.PRN

The file LAAMP.PRN represents the "spreadsheet" data that was developed by JSA. Since JSA work is documented elsewhere, the discussion presented in this report is limited to defining what is contained in the file, and does not present any detailed discussion as to how values should or could be selected.

The initial four lines of the file contain data necessary to start and end the simulation (IYRST and IYREND), establish the initial values of Mono Lake elevation (MLL), Grant

TABLE 3
Hydrologic Input Variable Names

Variable	Description
LVACT	Lee Vining Actual Flow
WCRO	Walker Creek Runoff
PCRO	Parker Creek Runoff
RSACT	Rush Creek Actual Flow
CPCP	Cain Ranch Precipitation
MTMAKE	Mono Tunnel Make
ORAEP	Owens River above East Portal
LVRO	Long Valley Runoff
LVPCP	Long Valley Precipitation
HCSPR	Hot Creek Springs
RCLRV	Rock Creek at Little Round Valley
RVRO	Round Valley Runoff
BCSPR	Birchim Canyon Springs
LWRO	Laws Runoff
FSLU	Fish Slough
BISACT	Bishop Area Actual
XKEO	Keough Hot Springs
BPRO	Big Pine Runoff
THRO	Tinemaha to Haiwee Runoff

storage (GSTOR), and Long Valley storage (LVSTOR), establish the "wet-normal-dry" criteria (DRY, WET, AVOMRO), and set a user specified options for the use of Grant Lake storage in meeting minimum creek flows and lake releases (IDRFF).

The data that establish year type include the average combined runoff for Mono Basin and Owens Valley (AVOMRO), and the wet and dry percentages of this average (WET and DRY) that are used to define "wet", "normal", or "dry" years for Grant and Long Valley storage "targets", desired Haiwee export amounts, the distribution of annual pumping limits, and the source of water (pumping or diversions) that is used to meets demands in the Owens Valley . The implementation of these variables is more fully described in the documentation of specific subroutines.

The IDRFF toggle allows the user to either permit only the monthly flow of Rush Creek to be made available for minimum Rush Creek flow release (IDRFF=1), or allow a certain portion of the first-of-month storage in Grant Lake be used as supplement to meet minimum flow requirements (IDRFF=2). Details of the IDRFF=2 option are provided in the description of the subroutine MONO.

The next 12 lines of the file contain certain identifying text information that is passed through the program. This identifying information is obtained from the actual spreadsheets that are used to develop this input file.

The next group of data include Mono Basin fractions to determine wet, normal, and dry years for minimum creek flow purposes (FFIDW, FFIDD, FFMBRO). Note that these parameters may be used to specify year type on criteria that are different from other parameters that require year type designation (e.g. Mono Lake level, storage and export targets). The details of the use of these parameters are further identified in the description of the subroutine HYDCALC.

The next group of data specify the cycle period for flushing flows for the Mono Basin creeks, and a toggle to conveniently "turn-off" minimum flow requirements (ILVCYC, IWCYC, IPCYC, IRSCYC, IDFISH). Note that each creek is set independently so the user is free, for example to flush Lee Vining Creek on a two year cycle, and others at a three year cycle. Details of flushing criteria are presented in the description of subroutines (HYDCALC and MBFLUSH).

The next four lines of data represent the monthly minimum flow requirements for each of the Mono Basin creeks in cubic feet per second (LVCFS, WCFS, PCFS, RSCFS). These values are converted into acre-feet per month in DO loop 302 (the next 6 lines of code).

The flushing flows for each creek under wet, normal, and dry year types, expressed in acre-feet per month, are read in the next 12 lines of the input file. These values are considered to be in addition to the minimum flow requirements.

The matrix of criteria for annual Mono Lake release and the user specified annual evaporation rate are read next. The first line of the matrix identifies the criteria to classify a year as "wet", "normal", or "dry" for lake release purposes (MLIDW, MLIDD, MLMBRO), and the evaporation rate in inches (MLEVAP). As in the minimum creek flow, the user is free to specify the criteria for year type for the purposes of lake release independent of other year type driven parameters. The next three lines of the matrix specify the release rule based on April 1 lake level (LLS), and the release criteria for each of the year types (LLRF). Note that the values can either be expressed as an actual release target in acre-feet (LLRF greater than 1), or as a calculated release target as a percentage of total Mono Basin Runoff for current year (LLRF between 0 and 1). If LLRF is negative, it connotes that a specified acre-foot amount is available to export; the runoff above this amount is the release target.

The next 12 lines represent the user defined target maximum and minimum storage levels for wet, normal and dry year types in Grant (GSTMAX and GSTMIN) and Long Valley (LVSMAX and LVSMIN).The details of the use of these parameters are described in the discussion of subroutines OWENS, NOTENUF, and TOOMUCH.

The next several lines of the file contains monthly values for various parameters. These are detailed in Table 4, and represent uses, losses, system constraints, and other parameters that vary monthly in each year. An important variable in this list is HCAP, which contain the monthly Haiwee export targets for wet, normal and dry years. These values can represent the physical capacity of the aqueduct, or some desired target export amount that is based on MWD programs or voluntary cutbacks in severe droughts.

TABLE 4
Uses Variable Names

Variable	Description
BIGGW	Big Pine Pumping Limits (max and min)
BPPMP	Big Pine Monthly % of Annual Pumping
BIGIRR	Big Pine Irrigation
BIGLOS	Big Pine Losses
BIGREC	Big Pine Recreation Uses
BIGSPD	Big Pine Spreading Capacity
BISCAP	Bishop Canal Capacity
BISGW	Bishop Pumping Limits (max and min)
BPMP	Bishop Monthly % of Annual Pumping
BISIRR	Bishop Irrigation
BISLOS	Bishop Losses
BISREC	Bishop Recreation Uses
BISSPR	Bishop Flowing Groundwater
BOQEV	Boquet Reservoir Evaporation
BOQFSH	Boquet Reservoir Fish Release
GRCAP	Grant Outflow Capacity
UORMIN	Upper Owens River Minimum Flow
UORMAX	Upper Owens River Maximum Flow
GIBDIV	Gibbs Creek Diversion
GIBFAR	Gibbs Farrington Siphon
GRAEV	Grant Evaporation Rate
HAEV	Haiwee Evaporation Rate
HLALOS	Haiwee to Los Angeles Losses
HCAP	Haiwee Export Target
LAWGW	Laws Pumping Limits (max and min)
LPMP	Laws Monthly % of Annual Pumping
LAWIRR	Laws Irrigation
LAWLOS	Laws Area Losses
LAWREC	Laws Recreation Uses
LAWSPD	Laws Spreading Capacity
LONGEV	Long Valley Reservoir Evaporation Rate
LVCAP	Long Valley Outflow Capacity
LVGAIN	Long Valley Gains
LVIRR	Long Valley Irrigation
MBGAIN	Mono Basin Gains
ODITCH	O-Ditch Diversion
PARDIV	Parker Creek Diversions
PARIRR	Parker Creek Irrigation
PVMIN	Pleasant Valley Reservoir Minimum Outflow
PVMAX	Pleasant Valley Reservoir Maximum Outflow
PVTRAN	Pleasant Valley to Tinemha Transit Losses
RCKFSH	Rock Creek Fish Flow
RVIRR	Round Valley Irrigation

TABLE 4 (cont)
Uses Variable Names

Variable	Description
RVLOS	Round Valley Losses
RVTRAN	Owens Gorge Transit Gains
SAND3	Sand Trap 3 Release
SAND4	Sand Trap 4 Release
THGW	Tinemaha to Haiwee Pumping Limits (max and min)
THPMP	Tinemaha to Haiwee Monthly % of Annual Pumping
THIRR	Tinemaha to Haiwee Irrigation
THLOS	Tinemaha to Haiwee Losses
THREC	Tinemaha to Haiwee Recreation Uses
THSPD	Tinemaha to Haiwee Spreading Capacity
THSPIL	Tinemaha to Hawiee Maintenance Spill
THSPR	Tinemaha to Haiwee Flowing Groundwater
THTRAN	Tinemaha to Haiwee Transit Losses
TINEV	Tinemaha Reservoir Evaporation Rate
USEDRY	Owens Valley Use Reduction Factor
LUSEP	Laws Pumping for Uses Factor
BUSEP	Bishop Pumping for Uses Factor
BPUSEP	Big Pine Pumping for Uses Factor
WALIRR	Walker Irrigation

VII. Description of Subroutine HYDCALC

Calculations that are independent of operations, but specific to a simulation are made in this subroutine. These include: 1) indexing the year type for minimum flows and the timing of flushing flows in the four Mono Basin Creeks, and 2) indexing year type for Mono Lake release, Grant and Long Valley storage and Haiwee export, and for Owens Valley use reduction.

Mono Basin Minimum Flow Requirement Indexing

The initial pass at indexing minimum flow requirements is to consider all years (1940 to 1989) as having base release requirements (i.e. no flushing flows). This is then modified by the individual creek cycle data (ILVCYC, IWCYC, IPCYC, IRSCYC) by a routine that cycles through the years of the simulation and indexes the years as wet year flush, normal year flush, or dry year flush. Note that the flushing indexing is simulation specific. As an example, assume that the cycle period for Lee Vining Creek is three years, and the simulation begins in 1940. The first year that a flushing flow would be required in this example would be 1942. If the simulation begins in 1942, the first year of a flushing flow would be 1944.

This indexing is done through the array variable IFFYT that can take either a 1, 2 or 3 (base requirement for wet, normal or dry year), or a 4, 5, or 6 (flushing requirement for wet, normal, or dry year) as its value, and has as identifiers I and J in this subroutine. The I represents the runoff year that is to be indexed, and the J indexes the four creeks as follows:

1 = Lee Vining
2 = Walker

3 = Parker

4 = Rush

Other Indexing of Year Type

The indexing of year type (wet normal, and, dry) for Mono Lake release, reservoir storage, and Haiwee export, are done through the variables IMLYT, IGSYT, ILVYT, AND IHCYT. Note that the criteria for lake releases is independent of the other three.

The indexing of year type for Owens Valley use reduction and pumping reductions due to drought conditions are done through the variables IRFC, USEDRY, REDFAC, and PCAP. The Inyo/Los Angeles Agreement provides for the possibility of reduced in-valley uses in the event of consecutive dry years. This provision is implemented by specifying the percentage reduction (USEDRY) in the first through fifth consecutive dry year, and tracking consecutive dry years with IRFC. The year is "indexed" by setting the values of REDFAC equal to the possible reduction (a value of 1 means no reduction). This concept is more fully described in the discussion of subroutine NOTENUF.

PCAP is designed to simulate decreases in pumping capacity that are caused by reduced water levels. It is assumed that a 10% decrease in annual capacity will occurin each succeeding dry year.

VIII. Description of Subroutine MLTAR

This subroutine establishes the annual Mono Lake release target. Five scenarios are possible: 1) the April 1 lake elevation is above the highest target (no release required), 2) the user specified criteria calls for no release, 3) a specific acre-foot release is required, 4) a release based on a percentage of runoff is required, and 5) a release of all runoff above a user defined value of available export is required.

The subroutine determines the gross amount of the required lake release, and then credits this annual release with minimum flow targets from the four creeks. As the simulation proceeds, the target is credited when releases are made, and debited when actual minimum flows are below targets. Lake releases are made from Lee Vining Creek and Grant Lake until the annual lake release is satisfied. Lake releases are not made directly from Walker or Parker Creeks.

IX. Description of Subroutine MBFLUSH

This subroutine establishes the target minimum flow requirement for the month by checking the year type (IFFYT) for each creek and the total flows (minimum flows, lake release, and spill) that have occurred in the months since the last flush. If the flushing flow that is required was exceeded in any month since the last flushing year, then only the base flow is required. If not, the appropriate flushing flow is chosen based on the wet year, normal year, dry year index previously discussed. The final required flows for the month, including flushing flows, are stored in variables WLVFF, WWFF, WPFF, and WMG1FF.

X. Description of Subroutine MBFISH

This subroutine calculates the actual minimum flow releases to each of the four Mono Basin Creeks based on targets established in MBFLUSH and available runoff. An overall theme of this subroutine and subroutines MLR, MBIRR, and CONDUIT is the use of the variables LVAVAIL, WAVAIL, PAVAIL, AND RSAVAIL. These represent the amount of water that is available for use, diversion, or lake release in each creek at any point in the simulation. Each time water is needed, the target "demand" is compared to the amount available to see if the target can be met. Once the use, diversion, or release is made, that amount is subtracted from the what was available prior to the use, diversion or release. This method provides a convenient method to track the water and develop an understanding of the consequences of user specified targets on allocating available runoff.

The initial values for the "available" variables are the runoff values for each creek for the specific month. Lee Vining and Rush use impaired values, that include the effect of SCE storage, LVACT and RSACT, while Walker and Parker use unimpaired runoff values. The LVACT is first reduced by the O-Ditch diversion prior to setting it equal to LVAVAIL as the diversion of Lee Vining Creek into O-Ditch occurs under a separate water right upstream of the diversion location.

The actual minimum flows are then calculated creek by creek by first comparing available to the target. If the available exceeds the target, the actual minimum flow (ALVFF, AWFF, APFF, AMG1FF) is set to the target, and the available is decremented by the amount of the minimum flow. If the available is below the target, the actual minimum flow is set to available, the available is set to zero, and the deficit is calculated (LVFD, WFD, PFD, MG1FD). Since the annual lake release target assumes that all minimum flows can be met, tracking the deficits allows for the increase of the lake release target, as appropriate.

In the case where Rush Creek available is below the target, the user has an option to make some of the storage in Grant (above minimum storage target) available to meet minimum flow requirements. This option is toggled with the variable IDRFF.

If IDRFF equals 2, and the available in Rush Creek is not enough to meet the Mono Gate 1 target, the following procedure is implemented to provide additional water for minimum flow requirements: 1) compare first of month storage in Grant with end-of-month "minimum" that is year type sensitive, 2) if the first-of-month storage is greater than the end-of-month minimum, this difference is added to the available flow in Rush Creek and the total is made available for minimum release (TAV), 3) if TAV is greater than the minimum flow requirement, the actual minimum flow is set to the requirement, available is set to zero, and the amount needed from storage is stored in GFFREL, and 4) if TAV is less than the minimum flow requirement, actual minimum flow is set to TAV, and the drop in storage is stored in GFFREL. If after number 1 above, the first-of-month storage is less than the end-of-month minimum, the actual minimum release is set to available, and available is set to zero, and no releases are made from Grant storage.

XI. Description of Subroutine MLR

This subroutine makes releases to Mono Lake in accordance with the following procedure:

1) The annual lake release target is incremented by the amount of monthly minimumflow deficit. This occurs only if a lake level release is needed.

2) The accumulated total releases for the runoff year (minimum releases, lake releases and spills) are compared to the annual lake release target calculated at the beginning of the runoff year.

3) If the target has not yet been reached, first-of-month Grant Lake storage is released until the target is reached or until Grant storage reaches end-of-month minimum. If first-of-month storage is less than end-of-month minimum, no release is made. The amount released is stored in array GLREL, and the annual lake release target is decremented.

4) Step 2 is repeated (note that the target may have been reduced by GLREL), and if the target has not been met, Rush Creek flow is released until the target is reached, or no more water is available. The amount released is stored in the variable RSLR, and the annual lake release target is decremented.

5) Step 2 is repeated (note that the target may have been reduced by GLREL and RSLR), and if the target has not been met, Lee Vining Creek flow is released until the target is reached, or no more water is available. The amount released is stored in the variable LVLR, and the annual lake release target is decremented.

XII. Description of Subroutine MBIRRG

The irrigation diversions and releases are made in this subroutine. A list of each diversion and release that the program simulates, and the source of the water is listed in Table 5. In each case the water available from the specified source is compared to the "target" release or diversion. If the available exceeds the target, the actual diversion or release is set equal to the target, and the available is decremented. If the available is less than the target, the actual diversion is set equal to the available and the available is set to zero.

Of note in this list are the Gibbs Farrington Siphon, and Sand Traps 3 and 4 which are releases from the conduit. In order to meet these "demands", water must enter the conduit upstream of the release point. In the case of the Gibbs Farrington Siphon, water from Lee Vining Creek "enters" the conduit, and then, assuming no losses, it is released from the conduit. The program accounts for the flow in the conduit by reducing the remaining capacity of the conduit by the amount of the release. Sand Trap 3 release is assumed to come only from Walker Creek, and Sand Trap 4 release is assumed to come only from Parker Creek. The remaining conduit capacity is reduced appropriately for each these releases.

It is recognized that some question exists as to the legality of some or all of these releases for irrigation under Los Angeles' current water rights. If it is decided that some or all of these releases cannot be legally made, the appropriate "targets" simply need to be set to zero without affecting any other portion of the program.

In addition, it is recognized that irrigation of Los Angeles owned land in the Mono Basin may not continue as it has in the past due to future release requirements (minimum stream flows and Mono Lake releases). The inherent flexibility in this program allows the user to

TABLE 5
Mono Basin Irrigation Variables

Target Variable Name	Actual Variable Name	Description	Creek
ODITCH	ODITCH	O-Ditch	Lee Vining
GIBDIV	AGIBDIV	Gibbs Creek Diversion	Lee Vining
GIBFAR	AGIBFAR	Gibbs Farrington Siphon	Lee Vining
WALIRR	AWALIRR	Walker Creek Release	Walker
SAND3	ASAND3	Sand Trap 3	Walker
PARDIV	APARDIV	Parker Creek Diversion	Parker
PARIRR	APARIRR	Parker Creek Releases	Parker
SAND4	ASAND4	Sand Trap 4	Parker

investigate the "impact" to irrigation by either specifying current irrigation target "demands" and evaluating the system's ability to meet these demands, or by reducing target "demands" under an alternative concept of land management.

XIII. Description of Subroutine CONDUIT

This subroutine compares the available water in Lee Vining, Walker and Parker Creeks to remaining conduit capacity as appropriate. If the available water exceeds the remaining capacity, the excess is spilled, and the conduit is filled to capacity. If the available is less than capacity, all water is diverted into the conduit. Based on a review of historic data, the spill at Lee Vining Creek is assumed to begin at a runoff value of 14,000 AF/month, although the capacity is approximately 18,000 AF/month. The remaining capacity of the conduit below Walker and Parker are adjusted to reflect water that enters the conduit upstream.

XIV. Description of Subroutine WP

This subroutine calculates a "minimum" West Portal flow, and the amount that is "available" for export from the Mono Basin if needed to meet the month's target Haiwee export to Los Angeles.

A concept that has been discussed in the past, but never formally accepted, is a minimum flow in the Upper Owens River to provide for fisheries requirements. The program implements this concept by calculating a "minimum" West Portal flow in the event that the sum of Owens River Above East Portal (ORAEP) and Mono Tunnel Make (MTMAKE) is less than the minimum Upper Owens River flow (UORMIN). This export is drawn from storage in Grant, but not below end-of-month minimum. A higher minimum Upper Owens River flow requirement has the effect of pulling more available water from the Mono Basin to Long Valley.

The final calculation of the subroutine sets the variable ATWP, which represent the amount of water available in Grant Reservoir for export if required. The calculation proceeds according to the following procedure:

1) The Grant storage at the end of the month with only "minimum" West Portal flow is calculated as the first-of-month storage plus the inflow (GIN) minus all outflows (GFFREL, GLREL, and MINWPF), and stored as variable TGS. Required Rush Creek flows and lake releases are passed through Grant without changing Grant storage.

2) The East Portal flow, which at this point consists of tunnel make (MTMAKE) and "minimum" West Portal flow (MINWPF) is calculated, and stored as variable TEPF.

3) The maximum West Portal flow allowed for the month is calculated by considering the flood flow maximum at Inaja Ranch (UORMAX), the flow in the Owens River above the East Portal, the tunnel make, the "minimum" West Portal flow, and the Grant outflow capacity. If these requirements do not allow any export, ATWP is set to zero for the month.

4) The actual West Portal flow for the year plus ATWP is checked against the annual water rights limit of export (167,000 acre-feet), and ATWP is modified to ensure that the limit is not exceeded.

The water that is available for export is simply "held" in Grant storage until and if subroutine NOTENUF is called. If some additional Grant storage is exported, actual end-of-month Grant storage is adjusted in the main program. The details of the conversion of ATWP to actual export is further discussed in the description of subroutine NOTENUF.

XV. Description of Subroutine OWENS

This subroutine performs the same type of water accounting procedures as the subroutines for the Mono Basin, and represents the initial pass at simulating operations in the Long Valley to Haiwee portion of the aqueduct system. It proceeds along the following: 1) calculates the Rock Creek diversion, 2) calculates Long Valley Reservoir inflow, 3) calculates the inflow, uses, and outflow from Round Valley, 4) sets the release in Long Valley such that the minimum Pleasant Valley Reservoir outflow and target "maximum" end-of-month Long Valley Reservoir storage are met, 5) calculates the Pleasant Valley Reservoir outflow, and 6) calculates the inflow, uses and outflow of the Laws, Bishop, Big Pine, and Tinemaha to Haiwee areas separately.

The Rock Creek diversion is made by diverting 80 percent of the flow at Rock Creek at Little Round Valley (RCLRV) that is in excess of the Rock Creek minimum monthly flow requirement (RCKFSH).

Long Valley inflow is the sum of measured runoff into Long Valley, Hot Creek Springs, the Rock Creek diversion, the minimum West Portal flow, Mono Tunnel make, and the unmeasured gains in Long Valley. Irrigation demands are subtracted from this result.

Round Valley inflow is the sum of Rock Creek flow that is not diverted, other measured runoff and Birchim Canyon Springs. Uses and losses within Round Valley include irrigation, recreation, and losses associated with runoff and uses. The outflow from Round Valley is the difference between inflow and uses and losses. Horton Creek outflow is assumed to be 20 percent of Round Valley outflow without Birchim Canyon Springs.

The minimum Long Valley outflow is calculated next. The minimum Pleasant Valley outflow requirement can be met with the transit gain in the gorge and Round Valley outflow. If this is not enough, water is released from Long Valley to meet the minimum. The outflow from Long Valley reservoir is then increased if necessary to meet the target maximum end-of-month storage in Long Valley.

Pleasant Valley outflow is then calculated and compared to the maximum outflow specified by the user. If Pleasant Valley outflow is greater than the maximum value, Long Valley outflow is reduced accordingly.

The next group of calculations involve the inflow, uses, losses and outflow in each of the four sub-areas of the Owens Valley: Laws, Bishop, Big Pine, and Tinemaha to Haiwee. Since the calculations follow a similar pattern in each area, the procedure is only described once.

The uses and losses are calculated for the area, summed, and stored in local variable (LUL, BUL, BPUL, THUL). In all areas except Tinemaha to Haiwee, the amount of pumping that is needed to meet these uses is calculated by multiplying the monthly use pumping factor (LUSEP, BUSE, BPUSEP) by uses and losses. The pumping use factor is year type dependent, and it is assumed that 1-LUSEP is factor for Owens River diversion that is multiplied by uses and losses to estimate the amount of diversion that will be allocated for uses.

The calculated pumping value is then compared to the minimum pumping specified by the user, corrected if necessary, and any losses associated with the pumping is then added. The pumping is stored in the use pumping arrays (ULAWP, UBISP, UBPP, UTHP). Later any pumping for export will be stored in export pumping arrays (ELAWP, EBISP, EBPP, ETHP). The sum of these two represents the total pumping for each area.

The amount of Owens River diversion is estimated by multiplying uses and losses (LUL, BUL, BPUL) by the appropriate factor (1-LUSEP, 1-BUSEP, 1-BPUSEP), and adding diversion related losses. This diversion is checked with the maximum diversion rate specified by the user. If the calculated diversion is greater than the specified maximum, the diversion is set equal to the maximum, and pumping is increased. Pumping is then compared to specified or calculated maximum pumping (depending on the value of IDOVP) and adjusted accordingly.

Area inflow is then calculated, uses and losses are calculated, and outflow from the area is calculated. Outflow can never be less than zero. If output shows zero, it generally means that use deficits occur in a particular area, and pumping constraints may be restricting the ability to meet use demands.

XVI. Description of Subroutine TOOMUCH

This subroutine is called if the available water for export (the sum of Pleasant Valley Reservoir outflow, and the outflows from the four identified areas of the Owens Valley less transit losses and less diversions) is greater than the user specified export for the month. The subroutine starts by calculating the amount of excess water.

The procedure to allocate the available water that is in excess of the target export (DIFF) proceeds with the following priorities: 1) store in Tinemaha and Haiwee, 2) reduce pumping, to minimum if necessary, 3) store the water in Long Valley Reservoir, 4) increase exports above target levels to aqueduct capacity, 5) spread water in the Laws area, 4) spread water in the Big Pine area, 6) spread water on the alluvial fans in the Tinemaha to Haiwee area, and 7) spill water east of the aqueduct in the Tinemaha to Haiwee area.

Store in Tinemaha and Haiwee

Simulated storage changes in these two reservoirs is limited to 2000 acre-feet per month in each reservoir. Total storage in either reservoir cannot exceed 10,000 acre-feet, and 10,000 acre-feet is the initialized storage at the beginning of a simulation.

Reduce Pumping

The amount of available reduction in each area is calculated as the difference in the pumping calculated in OWENS and the user specified minimum pumping. If the amount of available reduction is greater than the excess available water, pumping needs to be reduced, but not to minimum. A reduction factor (PRFAC) is calculated and is applied to each area.

If the amount of available reduction is less than the excess, all pumping is set to minimum and DIFF is decremented.

Store Excess in Long Valley Reservoir

Allocating some or all of the excess water into Long Valley storage is done by reducing the Long Valley outflow, previously calculated, thereby allowing water into storage above the target end-of-month "maximum" storage, which has been specified by the user. In any case, the "physical" maximum storage (180,000 acre-feet) is not exceeded and Long Valley outflow is not reduced to the point of violating minimum Pleasant Valley Reservoir outflow.

The rules to store in Long Valley are as follows: 1) calculate the end-of-month storage and store as variable TLVS, 2) calculate the minimum Pleasant Valley Reservoir outflow and store as variable TPVM, 3) calculate the "maximum" storage increase that could take place, which is defined as the difference between "physical" maximum storage and TLVS and store as variable MAXSI, 4) calculate the maximum that the target end-of-month "maximum" storage can be violated, which is assumed to be 20% and store as variable MAXTV, 5) calculate the allowable storage increase (STINC), as the smaller of MAXSI and MAXTV, 6) calculate the reduction in Long Valley outflow as the smaller of the DIFF and STINC, adjust Long Valley outflow, and readjust to ensure compliance of the minimum Pleasant Valley Reservoir minimum outflow, if necessary, and 7) decrement DIFF by the Long Valley Outflow reduction.

Increase Exports

Prior to spreading and/or spilling, the program allows for the export of water above the user specified target, up to a maximum of 1585 acre-feet per day (800 cfs).

Spread Excess in Laws Area

If DIFF is still positive, excess water still exists and water is diverted into the McNally Canals for spreading in the Laws area until DIFF is eliminated or user defined spreading capacity is reached. DIFF is decremented by the amount of the diversion. The actual spreading is stored as variable ALSPRD, and the diversion LORDIV is incremented to that which has already taken place.

Spread Excess in Big Pine Area

If DIFF is still positive, water is diverted for spreading in the Big Pine area until DIFF is eliminated or user defined spreading capacity is reached. DIFF is decremented by the amount of the diversion. The actual spreading is stored as variable ABPPRD, and the diversion BPORDIV is incremented to that which has already taken place.

Spread Excess on Alluvial Fans

If DIFF is still positive, water is diverted onto the alluvial fans in the Tinemaha to Haiwee area until DIFF is eliminated or user defined spreading capacity is reached. DIFF is decremented by the amount of the diversion. The actual spreading is stored as variable ATHFSP.

Spill Excess Water

The first seven "places" to allocate excess water ostensibly provide for the potential recapture by LADWP at a later time, either through a subsequent reduction in surface storage, a subsequent increase in baseflow to the Owens River or Los Angeles Aqueduct (i.e. increase in transit gain or decrease in transit loss), or subsequent groundwater pumping. Although a limitation of the simulation program is the lack of tracking of this water (except for the surface reservoir storage), it is generally acknowledged that benefits accrue from these excess water management practices.

If DIFF is still positive after storing and spreading water as described above, the excess water is released from the aqueduct system. This release represents the operational spill that are made east of the aqueduct in the Tinemaha to Haiwee area, and is stored as variable THOS.

XVII. Description of Subroutine NOTENUF

This subroutine is called if the available water for export (the sum of Pleasant Valley Reservoir outflow, and the outflows from the four identified areas of the Owens Valley, less transit losses and less diversions) is less than the user specified Haiwee export for the month.

The basic objective of this subroutine is to obtain additional water for export, without violating any user specified constraints. The priority to make-up or reduce the deficit is as follows: 1) reduce Tinemaha and Haiwee storage, 2) export available water from the Mono Basin, 3) increase Long Valley Reservoir outflow to allow for no storage change, if a storage increase is required by end-of-month storage target specification, 4) pump Owens Valley groundwater, 5) increase Long Valley outflow to allow end-of-month storage to go to the user specified minimum, 6) reduce Owens Valley uses under certain conditions (consecutive dry years), and make that reduction available for export, and 7) accept an export deficit as unavoidable.

Reduce Tinemaha and Haiwee Storage

Reduction in storage in either reservoir is limited to 2000 acre-feet per month.

Export from the Mono Basin

If a deficit still exits, the water that is available for export from the Mono Basin (ATWP) is converted into actual export (WPF), up to the amount of the deficit or the allowable Upper Owens River maximum (UORMAX). If the additional water is in excess of Long Valley outflow capacity, the export is reduced such that the reservoir outflow matches capacity.

DIFF is decremented by the value of WPF.

First Decrease in Long Valley Storage

If DIFF is still greater than zero, the end-of-month storage in Long Valley is calculated and stored in the variable TLVS. If TLVS is greater than the first-of-month storage, and is greater than the user specified minimum, then reservoir outflow is increased, up to the capacity of the penstock and up to the amount of the deficit, in order to prevent a storage increase during the month. DIFF is decremented by the amount of the increased outflow.

Owens Valley Groundwater Pumping

If DIFF is still positive, Owens Valley pumping is increased up to the amount of the deficit or the available pumping capacity. The amount of available pumping in each area is first calculated by subtracting the use associated pumping from the maximum specified pumping, and summing the results.

If the deficit is greater than the total available pumping, pumping is increased to maximum in each area, and DIFF is decremented.

If the deficit is less than the total available pumping, pumping needs to be increased, but not to maximum. In this case, an export pumping factor (EPFAC) is calculated and applied to each area, and DIFF is set to zero.

Second Decrease in Long Valley Storage

If DIFF is still positive, Long Valley outflow is increased until user specified "minimum" storage is reached or until penstock capacity is reached. DIFF is decremented by this increase in outflow.

Reduction in Owens Valley Uses

If DIFF is still positive, Owens Valley uses are reduced by the factor REDFAC (for consecutive dry years), the specification of which was discussed in the description of subroutine HYDCALC. REDFAC is multiplied by TOTUSE and the result, a total use reduction in acre-feet is stored in the variable REDUC. DIFF is decremented REDUC, or is set to zero, depending on whether the entire reduction is needed.

Export Reduction

If, after proceeding through these rules to increase the amount of water available to export, DIFF is still positive, the program simply sets EXPRED equal to DIFF, or export reduction. When this occurs, the interpretation is that the user cannot meet the desired export "demand" given the parameters that have been specified.

XVIII. Description of Subroutine GLVSTOR

This subroutine calculates the end-of-month storages for Grant Lake and Long Valley Reservoir. The storage in Grant is calculated first, and this value is compared to the target maximum storage specified by the user. If the Mono Lake elevation is greater than the minimum target level specified by the user, and Grant storage is higher than the user specified maximum, an attempt is made to "move" some of the storage from Grant to Long Valley if space is available.

If end of month Grant storage is above 50,000 acre-feet after the attempt to move storage, spill from Grant is calculated. If spill from Grant is necessary, an attempt to balance the spill between Grant and Lee Vining Creek is made by "moving" some of the spill to Lee Vining Creek. This procedure is done by reducing the amount of water that was previously diverted into the Lee Vining Conduit.

Long Valley storage is then calculated. If storage in Long Valley exceeds 120 percent of the target maximum (or the physical capacity of 180,000 acre-feet), Long Valley outflow is increased, subject to penstock capacity, and the water is spilled in the Owens Valley. If storage still exceeds the maximum, West Portal flow is reduced, and the water is spilled into Mono Lake. If storage still exceeds the user defined maximum, the excess storage will remain in Long Valley. Values above 180,000 AF indicate a spill would occur.

XIX. Description of Subroutine GRNPUMP

Based on discussions with LADWP, Inyo County, the State Water Resources Control Board, JSA and LSCE, the use of annual limits on Owens Valley groundwater pumping that appear in the Green Book together with the monthly minimum and maximum values was agreed to be an acceptable alternative to other methods of handling pumping.

The implementation of this approach is based on the comparison of the annual limit to the sum of year to date pumping and minimum pumping for the rest of the year. If the sum is higher than the annual limit, only minimum pumping is allowed during the rest of the year.

XX. Description of Output SUBROUTINE

The subroutine OUT1 writes the results of a simulation into various output files. The output is primarily designed to address water balance questions for each basin or area and these files can be imported into spreadsheets developed by JSA for statistical and graphical analysis. Each file is summarized in Appendix B in a simple table format that lists the variables in each output file.

XXI. Description of Subroutine BATHY

This bathymetry subroutine simply returns a reservoir surface area (needed to make evaporation calculations) given a storage volume, or Mono Lake elevation and surface area given a storage volume.

XXII. Description of ALAAMP Program

In order to accurately reflect Owens Valley pumping in simulations that involve the reduction of export from the Mono Basin, an alternative version of LAAMP, called ALAAMP, has been developed. The basic theme of ALAAMP is that pumping calculated by LAAMP for a "base case" is used as an input to LAAMP, and all pumping "decisions" normally found in LAAMP are supplanted by pumping input.

Due to this change, the main program (AMAIN.FOR replaces MAIN.FOR) and the following subroutines have been modified: AOWENS.FOR replaces OWENS.FOR, ATOOMUCH.FOR replaces TOOMUCH.FOR, and ANOTENUF.FOR replaces NOTENUF.FOR. The source code of these replacement portions are presented in Appendix C. A description of these subroutines, and how they have been modified relative to the original subroutines is presented below.

AMAIN.FOR and READPUMP.FOR

The only change in the main program is an additional call to the subroutine READPUMP. This subroutine is also presented in Appendix C, and simply reads a PUMPING.OUT file that has renamed PUMPING.IN. By reading in a pumping file created by a run of LAAMP, the user is specifying that groundwater pumping in the Owens Valley is a given, and ALAAMP does nothing to cause additional pumping or lower pumping in the event of changed operations in other parts of the system.

AOWENS.FOR

Since pumping is read as an input in ALAAMP, simulated operations in the Owens Valley are modified to incorporate the given pumping rather than calculate it. The amount of outflow from each area also include the "export" pumping component read in PUMPING.IN.

ATOOMUCH.FOR

Since pumping is specified, ATOOMUCH.FOR has been modifed to remove the routine that reduced pumping in the Owens Valley. All other routines are unmodified.

ANOTENUF.FOR

Since pumping is specified, ANOTENUF.FOR has been modified to remove the routine that increased pumping for export.

APPENDIX A

LAAMP Source Code Listing

```
      PROGRAM LAAMP
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
      INCLUDE 'LAAMP.INT'
      INCLUDE 'LAAMP.DST'
C
C  PROGRAM TO SIMULATE THE OPERATION OF THE LOS ANGELES AQUEDUCT
C  MODIFIED AND ENHANCED FROM A LADWP SIMULATION MODEL
C
C     WILLIAM R. HUTCHISON, LUHDORFF & SCALMANINI CONSULTING ENGINEERS
C     RUSSEL T. BROWN, JONES & STOKES ASSOCIATES
C     PAUL L. WISHEROPP, JONES & STOKES ASSOCIATES
C
C      JULY 15, 1991
C         MODIFIED AUGUST 8, 1991
C         MODIFIED AUGUST 12, 1991
C         MODIFIED SEPTEMBER 17, 1991
C         MODIFIED SEPTEMBER 30, 1991
C         MODIFIED OCTOBER 14, 1991
C         MODIFIED DECEMBER 20, 1991
C
C********* OPEN FILES AND READ BATHYMETRY DATA, INPUT HYDROLOGY AND
C           "SPREADSHEET" DATA
C
      WRITE (*,10)
   10 FORMAT (' READING FILES AND PREPARING DATA')
C
C******** READ DATA
C
      CALL READBH (X)
      CALL READPRN (IYRST,IYREND,ILVCYC,IWCYC,IPCYC,IRSCYC,FFMBRO,
     +  FFIDW,FFIDD,MLMBRO,MLIDW,MLIDD,AVOMRO,WET,DRY,IDOVP,IDRFF)
C
C******** HYDROLOGIC CALCULATIONS
C
      CALL HYDCALC (IYRST,IYREND,ILVCYC,IWCYC,IPCYC,IRSCYC,FFMBRO,
     +  FFIDW,FFIDD,MLMBRO,MLIDW,MLIDD,AVOMRO,WET,DRY,IDOVP)
C
C******** INITITALIZE MONO LAKE EQUATION
C
      CALL BATHY (1,MLL(IYRST,1),MLVOL(IYRST,1),MLAREA(IYRST,1))
      RI=-0.206935+(0.00905776*MLEVAP(13))
      YINT=-8651.95+(238.897*MLEVAP(13))
C
C********* START SIMULATION
C
      WRITE (*,11) IYRST+1900,IYREND+1901
   11 FORMAT (' SIMULATION FROM APRIL',I5,1X,'TO MARCH',I5/)
C
C*** ANNUAL LOOP
C
      DO 100 I=IYRST,IYREND

      WRITE (*,102) I,I+1
```

```
  102 FORMAT (1H+,' SIMULATING RUNOFF YEAR ',I2,'-',I2)
C
C******* LAKE RELEASE TARGET
C
      CALL MLTAR (I)
C
C*** MONTHLY LOOP
C
      DO 101 J=1,12
C
C******* CALCULATE RESERVOIR EVAPORATION AND RAINFALL
C
C*** GRANT
C
      CALL BATHY (2,GSTOR(I,J),GELEV,GSURF)
      GRAIN=(CPCP(I,J)/12)*GSURF
      GEVAP(I,J)=(GRAEV(J)*(GSURF/1000))-GRAIN
C
C*** LONG VALLEY
C
      CALL BATHY (3,LVSTOR(I,J),LVELEV,LVSURF)
      LVRAIN=(LVPCP(I,J)/12)*LVSURF
      LVEV(I,J)=(LONGEV(J)*(LVSURF/1000))-LVRAIN
C
C*************** CALCULATE AVAILABLE FROM EACH AREA
C
C
C********* MONO BASIN
C
C************** SET AVAILABLE TO RUNOFF/ACTUAL AND CONDUIT
C               MAX FLOWS AND INITIALIZE FISH DEFICITS
C
      LVAVAIL=LVACT(I,J)-ODITCH(J)
      IF (LVAVAIL.GT.14000) LVSPILL(I,J)=-7000+(0.5*LVAVAIL)
      IF (LVAVAIL.GT.22000) LVSPILL(I,J)=LVAVAIL-18000
      LVAVAIL=LVAVAIL-LVSPILL(I,J)
      IF (LRT(I).NE.-1) LRT(I)=LRT(I)-LVSPILL(I,J)
      LEEMAX(I,J)=LEEMAX(I,J)-LVSPILL(I,J)
      IF (LEEMAX(I,J).LT.0) LEEMAX(I,J)=0
C
      WAVAIL=WCRO(I,J)
      PAVAIL=PCRO(I,J)
      RSAVAIL=RSACT(I,J)+MBGAIN(1,J)
      LVCMAX=300*1.9835*DAYS(J)
      WMAX=325*1.9835*DAYS(J)
      CAGMAX=350*1.9835*DAYS(J)
      LVFD=0
      WFD=0
      PFD=0
      MG1FD=0
C
C
C************** ESTABLISH FISH FLOWS AFTER CREDITING SPILLS AND
C               CHECKING IF SPILLS AND LAKE RELEASES OF PAST MONTHS
```

```
C               MEET FLUSHING REQUIREMENTS
C
      CALL MBFLUSH (I,J,WLVFF,WWFF,WPFF,WRSFF,ILVCYC,IWCYC,IPCYC,
     +      IRSCYC,IYRST)
C
C
C****************  CALCULATE ACTUAL FISH RELEASES
C
C
      CALL MBFISH (I,J,WLVFF,WWFF,WPFF,WRSFF,LVFD,WFD,PFD,MG1FD,
     +             LVAVAIL,WAVAIL,PAVAIL,RSAVAIL,IDRFF)
C
C
C****** LAKE RELEASES
C
      IF (LRT(I).GT.-1) THEN
         CALL MLR (I,J,LVFD,WFD,PFD,MG1FD,LVAVAIL,RSAVAIL)
      END IF
C
C************************ IRRIGATION
C
      CALL MBIRRIG (I,J,LVAVAIL,WAVAIL,PAVAIL,LVCMAX,WMAX,CAGMAX)
C
C********** SPILL EXTRA AND SEND AVAILABLE TO CONDUIT
C
      CALL CONDUIT (I,J,LVAVAIL,WAVAIL,PAVAIL,RSAVAIL,
     +   LVCMAX,WMAX,CAGMAX)
C
C******* CALCULATE MINIMUM WEST PORTAL FLOW
C
      CALL WP (I,J)
C
C
C********* OWENS RIVER BASIN
C
      CALL OWENS (I,J)
C
C****************  CHECK AVEX (AVAILABLE TO EXPORT)
C
C   "AVEX" IS WATER THAT CAN BE EXPORTED AFTER ALL USES ARE
C   SATISFIED (NOT AMOUNT OF USE IS SPECIFIED AND COMPARED TO
C   RUNOFF AND USER SPECIFIED MINIMUM PUMPING. THIS VALUE
C   IS THEN COMPARED TO TARGET HAIWEE EXPORT "CAPACITY" (HCAP).
C
C         IF OVAVAIL>HCAP INC STORAGE, SPREAD AND SPILL
C         IF OVAVAIL<HCAP DEC STORAGE, PUMP GROUNDWATER,
C            EXPORT FROM MONO BASIN, AND REDUCE USES
C
C
C*******  COMPARE
C
      AVAIL=PVOUT(I,J)+LAWSOUT(I,J)+BISOUT(I,J)+BPOUT(I,J)+THOUT(I,J)
     +                 -THAR(I,J)-LORDIV(I,J)-BISORD(I,J)-BPORDIV(I,J)
      OPVOUT(I,J)=PVOUT(I,J)
```

```
      BBTL=PVTRAN(1,J)+(PVTRAN(2,J)*LBBPRO(I,J))
     +                (PVTRAN(3,J)*PVOUT(I,J))
     +                +(PVTRAN(4,J)*UBPP(I,J))
      TINOUT=PVOUT(I,J)+LAWSOUT(I,J)+BISOUT(I,J)+BPOUT(I,J)
     +                 -BBTL-TINEV(J)
      OLBTL=THTRAN(1,J)+(THTRAN(2,J)*THRO(I,J))
     +                 +(THTRAN(3,J)*TINOUT)
     +                 +(THTRAN(4,J)*UTHP(I,J))
      AVEX(I,J)=MAX(0,AVAIL-BBTL-OLBTL)
      IF (AVEX(I,J).GE.HCAP(IHCYT(I),J)) THEN
         IDTMNE(I,J)=1
         CALL TOOMUCH (I,J)
      ELSE
         IDTMNE(I,J)=2
         CALL NOTENUF (I,J)
      END IF
C
C***** CALCULATE GRANT AND LONG VALLEY STORAGES
C
      CALL GLVSTOR (I,J)
C
C***** CALCULATE TINEMAHA INFLOW
C
      PVTTL(I,J)=PVTRAN(1,J)+(PVTRAN(2,J)*LBBPRO(I,J))
     +                      +(PVTRAN(3,J)*PVOUT(I,J))
     +                      +(PVTRAN(4,J)*(UBPP(I,J)+EBPP(I,J)))
      TININF(I,J)=PVOUT(I,J)+LAWSOUT(I,J)+BISOUT(I,J)+BPOUT(I,J)
     +  -PVTTL(I,J)
C
C********* CALCULATE FLOW TO CITY
C
      FLOWLA(I,J)=HAEX(I,J)-HLALOS(1,J)+(HLALOS(2,J)*HAEX(I,J))
C
C******* CALCULATE MONO LAKE LEVEL, AREA AND STORAGE CHANGE
C
      TREL(I,J)=ALR(I,J)+TFR(I,J)+TSPILL(I,J)
      XSC=TREL(I,J)+((CPCP(I,J)/12)*MLAREA(I,J))
     1             -(MLEVAP(J)*MLAREA(I,J))
      USC=YINT+(RI*MBRO(I,J))
      TSC=XSC+USC
      MLVOL(I,J+1)=MLVOL(I,J)+TSC
      MLSC(I,J)=MLVOL(I,J+1)-MLVOL(I,J)
      CALL BATHY (4,MLVOL(I,J+1),MLL(I,J+1),MLAREA(I,J+1))
      IF (J.EQ.12.AND.I.LT.89) THEN
         MLL(I+1,1)=MLL(I,13)
         MLVOL(I+1,1)=MLVOL(I,13)
         MLAREA(I+1,1)=MLAREA(I,13)
      END IF
C
C******* CHECK PUMPING AGAINST "GREEN BOOK" LIMITS
C
      IF (IDOVP.EQ.3) CALL GRNPUMP (I,J)
C
C**** END MONTHLY LOOP
```

```
 101 CONTINUE

***** END OF ANNUAL LOOP

 100 CONTINUE

********* OUTPUT

     WRITE (*,200)
 200 FORMAT (' WRITING OUTPUT')
     CALL OUT1 (IYRST,IYREND)

     STOP ' SIMULATION COMPLETED'
     END
```

```
      SUBROUTINE READBH (X)
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
      X=0
C
C*** GRANT LAKE AND LONG VALLEY
C
      WRITE (*,91)
   91 FORMAT (' READING BATHY.DAT')
      OPEN (2,FILE='BATHY.DAT')
C
      READ (2,*) ((GLBATHY(I,J),I=1,73),J=1,3),
     1           ((LVBATHY(I,J),I=1,97),J=1,3)
      DO 100 I=1,170
      READ (2,101) (MNBATHY(I,J),J=1,3)
  101 FORMAT (3F9.0)
  100 CONTINUE
      CLOSE (2)
C
C****** READ HYDROLOGY FILE
C
      WRITE (*,92)
   92 FORMAT (' READING INPHYD.DAT')
      OPEN (3,FILE='INPHYD.DAT')
C
      DO 200 I=40,89
      DO 201 J=1,13
C
      READ (3,202) LVACT(I,J),WCRO(I,J),PCRO(I,J),
     1 SEPCRO,RSACT(I,J),CPCP(I,J),MTMAKE(I,J),
     2 ORAEP(I,J),LVRO(I,J),HCHWY(I,J),HCSPR(I,J),
     3 RCLRV(I,J),LVPCP(I,J),RVRO(I,J),BCSPR(I,J),LWRO(I,J),
     4 FSLU(I,J),BISACT(I,J),XKEO(I,J),BPRO(I,J),
     5 TINPCP,THRO(I,J),HAIPCP
C
      MBRO(I,J)=LVACT(I,J)+WCRO(I,J)+PCRO(I,J)+RSACT(I,J)+MBGAIN(1,J)
      LBBPRO(I,J)=LWRO(I,J)+BISACT(I,J)+BPRO(I,J)
      OVRO(I,J)=LBBPRO(I,J)+THRO(I,J)+RVRO(I,J)
      MBOVRO(I,J)=MBRO(I,J)+OVRO(I,J)+LVRO(I,J)
  202 FORMAT (10X,5F8.0,F8.2,6F8.0,F8.2,7F8.0,F8.2,F8.0,F8.2)
C
  201 CONTINUE
  200 CONTINUE
      CLOSE (3)
C
      RETURN
      END
```

```
      SUBROUTINE READPRN (IYRST,IYREND,ILVCYC,IWCYC,IPCYC,IRSCYC,
     +  FFMBRO,FFIDW,FFIDD,MLMBRO,MLIDW,MLIDD,AVOMRO,WET,
     +  DRY,IDOVP,IDRFF)
C
      IMPLICIT REAL (A-H,L-Z)
      CHARACTER*60 TEXT(13)
      INCLUDE 'LAAMP.CMB'
C
C******* READ "SPREADSHEET" FILE
C
      WRITE (*,94)
   94 FORMAT (' READING LAAMP.PRN')
      OPEN (5,FILE='LAAMP.PRN')
C
C***** SIMULATION CONTROL DATA
C
      READ (5,407) I1,I2
      IYRST=I1-1900
      IYREND=I2-1900
      READ (5,401) MLL(IYRST,1),GSTOR(IYRST,1),LVSTOR(IYRST,1)
      READ (5,401) DRY,WET,AVOMRO
      READ (5,407) IDRFF,IDOVP
C
C***** TEXT HEADER
C
      DO 299 I=1,12
      READ (5,400) TEXT(I)
  299 CONTINUE
C
C***** MONO BASIN FISH FLOWS
C
      DO 300 I=1,6
      BLVFF(I,13)=0
      BWFF(I,13)=0
      BPFF(I,13)=0
      BRSFF(I,13)=0
      FLVFF(I,13)=0
      FWFF(I,13)=0
      FPFF(I,13)=0
      FRSFF(I,13)=0
  300 CONTINUE
C
      READ (5,401) FFIDW,FFIDD,FFMBRO
      READ (5,402) ILVCYC,IWCYC,IPCYC,IRSCYC,IDFSH
      DO 301 I=1,3
      READ (5,403) (LVCFS(J),J=1,12)
      READ (5,403) (WCFS(J),J=1,12)
      READ (5,403) (PCFS(J),J=1,12)
      READ (5,403) (RSCFS(J),J=1,12)
      DO 302 J=1,12
      BLVFF(I,J)=1.9835*LVCFS(J)*DAYS(J)
      BWFF(I,J)=1.9835*WCFS(J)*DAYS(J)
      BPFF(I,J)=1.9835*PCFS(J)*DAYS(J)
      BRSFF(I,J)=1.9835*RSCFS(J)*DAYS(J)
```

```
  302 CONTINUE
  301 CONTINUE
      DO 303 I=4,6
      READ (5,403) (FLVFF(I,J),J=1,12)
      READ (5,403) (FWFF(I,J),J=1,12)
      READ (5,403) (FPFF(I,J),J=1,12)
      READ (5,403) (FRSFF(I,J),J=1,12)
  303 CONTINUE
C
      DO 304 I=1,6
      DO 305 J=1,12
      BLVFF(I,13)=BLVFF(I,13)+BLVFF(I,J)
      BWFF(I,13)=BWFF(I,13)+BWFF(I,J)
      BPFF(I,13)=BPFF(I,13)+BPFF(I,J)
      BRSFF(I,13)=BRSFF(I,13)+BRSFF(I,J)
      FLVFF(I,13)=FLVFF(I,13)+FLVFF(I,J)
      FWFF(I,13)=FWFF(I,13)+FWFF(I,J)
      FPFF(I,13)=FPFF(I,13)+FPFF(I,J)
      FRSFF(I,13)=FRSFF(I,13)+FRSFF(I,J)
  305 CONTINUE
  304 CONTINUE
      IF (IDFSH.EQ.0) THEN
         DO 502 J=1,13
         DO 503 I=1,6
         BLVFF(I,J)=0
         BWFF(I,J)=0
         BPFF(I,J)=0
         BRSFF(I,J)=0
         FLVFF(I,J)=0
         FWFF(I,J)=0
         FPFF(I,J)=0
         FRSFF(I,J)=0
  503    CONTINUE
  502    CONTINUE
      END IF
C
C***** MONO LAKE RELEASE RULES
C
      READ (5,601) MLIDW,MLIDD,MLMBRO,MLEVAP(13)
      READ (5,404) LLS(1),(LLRF(1,J),J=1,3)
      READ (5,404) LLS(2),(LLRF(2,J),J=1,3)
      READ (5,404) LLS(3),(LLRF(3,J),J=1,3)
      MLEVAP(1)=(MLEVAP(13)/12)*0.0430
      MLEVAP(2)=(MLEVAP(13)/12)*0.0875
      MLEVAP(3)=(MLEVAP(13)/12)*0.1186
      MLEVAP(4)=(MLEVAP(13)/12)*0.1324
      MLEVAP(5)=(MLEVAP(13)/12)*0.1641
      MLEVAP(6)=(MLEVAP(13)/12)*0.1315
      MLEVAP(7)=(MLEVAP(13)/12)*0.1136
      MLEVAP(8)=(MLEVAP(13)/12)*0.0795
      MLEVAP(9)=(MLEVAP(13)/12)*0.0438
      MLEVAP(10)=(MLEVAP(13)/12)*0.0375
      MLEVAP(11)=(MLEVAP(13)/12)*0.0267
      MLEVAP(12)=(MLEVAP(13)/12)*0.0219
```

```
C
C***** TARGET STORAGES AND TARGET EXPORT
C
      READ (5,403) (GSTMAX(1,J),J=1,12)
      READ (5,403) (GSTMIN(1,J),J=1,12)
      READ (5,403) (GSTMAX(2,J),J=1,12)
      READ (5,403) (GSTMIN(2,J),J=1,12)
      READ (5,403) (GSTMAX(3,J),J=1,12)
      READ (5,403) (GSTMIN(3,J),J=1,12)
      READ (5,403) (LVSMAX(1,J),J=1,12)
      READ (5,403) (LVSMIN(1,J),J=1,12)
      READ (5,403) (LVSMAX(2,J),J=1,12)
      READ (5,403) (LVSMIN(2,J),J=1,12)
      READ (5,403) (LVSMAX(3,J),J=1,12)
      READ (5,403) (LVSMIN(3,J),J=1,12)
C
C***** USES
C
      READ (5,405) (BIGGW (2,J),J=1,13)
      READ (5,405) (BIGGW (1,J),J=1,13)
      READ (5,403) (BPPMP(1,J),J=1,12)
      READ (5,403) (BPPMP(2,J),J=1,12)
      READ (5,403) (BPPMP(3,J),J=1,12)
      READ (5,405) (BIGIRR(J),J=1,13)
      READ (5,405) (BIGLOS(1,J),J=1,13)
      READ (5,403) (BIGLOS(2,J),J=1,12)
      READ (5,403) (BIGLOS(3,J),J=1,12)
      READ (5,403) (BIGLOS(4,J),J=1,12)
      READ (5,403) (BIGLOS(5,J),J=1,12)
      READ (5,403) (BIGLOS(6,J),J=1,12)
      READ (5,405) (BIGREC(J),J=1,13)
      READ (5,405) (BIGSPD(J),J=1,13)
      READ (5,405) (BISCAP(J),J=1,13)
      READ (5,405) (BISGW (2,J),J=1,13)
      READ (5,405) (BISGW (1,J),J=1,13)
      READ (5,403) (BPMP(1,J),J=1,12)
      READ (5,403) (BPMP(2,J),J=1,12)
      READ (5,403) (BPMP(3,J),J=1,12)
      READ (5,405) (BISIRR(J),J=1,13)
      READ (5,405) (BISLOS(1,J),J=1,13)
      READ (5,403) (BISLOS(2,J),J=1,12)
      READ (5,403) (BISLOS(3,J),J=1,12)
      READ (5,403) (BISLOS(4,J),J=1,12)
      READ (5,403) (BISLOS(5,J),J=1,12)
      READ (5,405) (BISREC(J),J=1,13)
      READ (5,405) (BISSPR(J),J=1,13)
      READ (5,405) (GCAP(J),J=1,13)
      READ (5,405) (MCAP(J),J=1,13)
      READ (5,405) (LCAP(J),J=1,13)
      READ (5,405) (WCAP(J),J=1,13)
      DO 500 I=40,89
      DO 501 J=1,13
      GOUTMAX(I,J)=GCAP(J)
      MG1MAX(I,J)=MCAP(J)
```

```
      LEEMAX(I,J)=LCAP(J)
      WPFMAX(I,J)=WCAP(J)
  501 CONTINUE
  500 CONTINUE
      READ (5,405) (UORMIN(J),J=1,13)
      READ (5,405) (UORMAX(J),J=1,13)
      READ (5,405) (GIBDIV(J),J=1,13)
      READ (5,405) (GIBFAR(J),J=1,13)
      READ (5,405) (GRAEV (J),J=1,13)
      READ (5,405) (HAEV  (J),J=1,13)
      READ (5,405) (HLALOS(1,J),J=1,13)
      READ (5,403) (HLALOS(2,J),J=1,12)
      READ (5,403) (HLALOS(3,J),J=1,12)
      READ (5,405) (HCAP(1,J),J=1,13)
      READ (5,405) (HCAP(2,J),J=1,13)
      READ (5,405) (HCAP(3,J),J=1,13)
      READ (5,405) (LAWGW (2,J),J=1,13)
      READ (5,405) (LAWGW (1,J),J=1,13)
      READ (5,403) (LPMP(1,J),J=1,12)
      READ (5,403) (LPMP(2,J),J=1,12)
      READ (5,403) (LPMP(3,J),J=1,12)
      READ (5,405) (LAWIRR(J),J=1,13)
      READ (5,405) (LAWLOS(1,J),J=1,13)
      READ (5,403) (LAWLOS(2,J),J=1,12)
      READ (5,403) (LAWLOS(3,J),J=1,12)
      READ (5,403) (LAWLOS(4,J),J=1,12)
      READ (5,403) (LAWLOS(5,J),J=1,12)
      READ (5,403) (LAWLOS(6,J),J=1,12)
      READ (5,405) (LAWREC(J),J=1,13)
      READ (5,405) (LAWSPD(J),J=1,13)
      READ (5,405) (LONGEV(J),J=1,13)
      READ (5,405) (LVCAP (J),J=1,13)
      READ (5,405) (LVGAIN(1,J),J=1,13)
      READ (5,403) (LVGAIN(2,J),J=1,12)
      READ (5,403) (LVGAIN(3,J),J=1,12)
      READ (5,403) (LVGAIN(4,J),J=1,12)
      READ (5,405) (LVIRR (J),J=1,13)
      READ (5,405) (MBGAIN(1,J),J=1,13)
      READ (5,403) (MBGAIN(2,J),J=1,12)
      READ (5,403) (MBGAIN(3,J),J=1,12)
      READ (5,403) (MBGAIN(4,J),J=1,12)
      READ (5,405) (ODITCH(J),J=1,13)
      READ (5,405) (PARDIV(J),J=1,13)
      READ (5,405) (PARIRR(J),J=1,13)
      READ (5,405) (PVMIN(J),J=1,13)
      READ (5,405) (PVMAX(J),J=1,13)
      READ (5,405) (PVTRAN(1,J),J=1,13)
      READ (5,403) (PVTRAN(2,J),J=1,12)
      READ (5,403) (PVTRAN(3,J),J=1,12)
      READ (5,403) (PVTRAN(4,J),J=1,12)
      READ (5,403) (PVTRAN(5,J),J=1,12)
      READ (5,405) (RCKFSH(J),J=1,13)
      READ (5,405) (RVIRR (J),J=1,13)
      READ (5,405) (RVLOS (1,J),J=1,13)
```

```
      READ (5,403) (RVLOS (2,J),J=1,12)
      READ (5,403) (RVLOS (3,J),J=1,12)
      READ (5,403) (RVLOS (4,J),J=1,12)
      READ (5,405) (RVTRAN(1,J),J=1,13)
      READ (5,403) (RVTRAN(2,J),J=1,12)
      READ (5,403) (RVTRAN(3,J),J=1,12)
      READ (5,405) (SAND3 (J),J=1,13)
      READ (5,405) (SAND4 (J),J=1,13)
      READ (5,405) (THGW  (2,J),J=1,13)
      READ (5,405) (THGW  (1,J),J=1,13)
      READ (5,403) (THPMP(1,J),J=1,12)
      READ (5,403) (THPMP(2,J),J=1,12)
      READ (5,403) (THPMP(3,J),J=1,12)
      READ (5,405) (THWIRR (J),J=1,13)
      READ (5,405) (THEIRR (J),J=1,13)
      READ (5,405) (THLOS (1,J),J=1,13)
      READ (5,403) (THLOS (2,J),J=1,12)
      READ (5,403) (THLOS (3,J),J=1,12)
      READ (5,403) (THLOS (4,J),J=1,12)
      READ (5,403) (THLOS (5,J),J=1,12)
      READ (5,403) (THLOS (6,J),J=1,12)
      READ (5,405) (THWREC (J),J=1,13)
      READ (5,405) (THEREC(J),J=1,13)
      READ (5,405) (THSPD (J),J=1,13)
      READ (5,405) (THSPIL(J),J=1,13)
      READ (5,405) (THSPR (J),J=1,13)
      READ (5,405) (THTRAN(1,J),J=1,13)
      READ (5,403) (THTRAN(2,J),J=1,12)
      READ (5,403) (THTRAN(3,J),J=1,12)
      READ (5,403) (THTRAN(4,J),J=1,12)
      READ (5,403) (THTRAN(5,J),J=1,12)
      READ (5,405) (TINEV (J),J=1,13)
      READ (5,406) (USEDRY(J),J=1,5)
      READ (5,403) (LUSEP(3,J),J=1,12)
      READ (5,403) (LUSEP(2,J),J=1,12)
      READ (5,403) (LUSEP(1,J),J=1,12)
      READ (5,403) (BUSEP(3,J),J=1,12)
      READ (5,403) (BUSEP(2,J),J=1,12)
      READ (5,403) (BUSEP(1,J),J=1,12)
      READ (5,403) (BPUSEP(3,J),J=1,12)
      READ (5,403) (BPUSEP(2,J),J=1,12)
      READ (5,403) (BPUSEP(1,J),J=1,12)
      READ (5,405) (WALIRR(J),J=1,13)

****** INITIAL AND MINIMUM GW ELEV

       DO 310 I=1,10
       READ (5,408) WSEL(IYRST,I),MINWSE(I)
  310 CONTINUE

****** DONE READING USES

      CLOSE (5)
       WRITE (*,999) MINWSE(10)
```

```
C  999 FORMAT (F10.2)
C
C*** FORMAT STATEMENTS
C
  400 FORMAT (A60)
  401 FORMAT (F8.0,2F9.0)
  601 FORMAT (F8.0,3F9.0)
  402 FORMAT (I8,4I9)
  403 FORMAT (F8.0,11F9.0)
  404 FORMAT (F8.0,3F9.0)
  405 FORMAT (F8.0,12F9.0)
  406 FORMAT (F8.0,4F9.0)
  407 FORMAT (I8,I9)
  408 FORMAT (F8.0,F9.0)
C
      RETURN
      END
```

```
      SUBROUTINE HYDCALC (IYRST,IYREND,ILVCYC,IWCYC,IPCYC,IRSCYC,
     +  FFMBRO,FFIDW,FFIDD,MLMBRO,MLIDW,MLIDD,AVOMRO,WET,
     +  DRY,IDOVP)
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
C
C***** INITIALIZE TINEMAHA AND HAIWEE STORAGE
C
      TINSTOR(IYRST,1)=10000
      HAISTOR(IYRST,1)=10000
C
C***** INDEX FISH FLOW BY YEAR AND CREEK CYCLE -- SUBJECT
C         TO CHANGE LATER BASED ON COMPARISON OF TOTAL RELEASES
C         WITHIN CYCLE
C
C** INITIALLY SET ALL CREEKS IN ALL YEARS TO BASE FISH FLOWS
C
      DO 400 I=40,89
      DO 401 J=1,4
      IF (MBRO(I,13).GT.(FFIDW*FFMBRO)) IFFYT(I,J)=1
      IF (MBRO(I,13).LE.(FFIDW*FFMBRO).AND.
     +    MBRO(I,13).GE.(FFIDD*FFMBRO)) IFFYT(I,J)=2
      IF (MBRO(I,13).LT.(FFIDD*FFMBRO)) IFFYT(I,J)=3
  401 CONTINUE
  400 CONTINUE
C
C** CHECK RUNOFF WITH INDICES AND SET ACCORDINGLY
C
      DO 402 I=(IYRST+ILVCYC-1),IYREND,ILVCYC
      IF (MBRO(I,13).GT.(FFIDW*FFMBRO)) IFFYT(I,1)=4
      IF (MBRO(I,13).LE.(FFIDW*FFMBRO).AND.
     +    MBRO(I,13).GE.(FFIDD*FFMBRO)) IFFYT(I,1)=5
      IF (MBRO(I,13).LT.(FFIDD*FFMBRO)) IFFYT(I,1)=6
  402 CONTINUE
C
      DO 403 I=(IYRST+IWCYC-1),IYREND,IWCYC
      IF (MBRO(I,13).GT.(FFIDW*FFMBRO)) IFFYT(I,2)=4
      IF (MBRO(I,13).LE.(FFIDW*FFMBRO).AND.
     +    MBRO(I,13).GE.(FFIDD*FFMBRO)) IFFYT(I,2)=5
      IF (MBRO(I,13).LT.(FFIDD*FFMBRO)) IFFYT(I,2)=6
  403 CONTINUE
C
      DO 404 I=(IYRST+IPCYC-1),IYREND,IPCYC
      IF (MBRO(I,13).GT.(FFIDW*FFMBRO)) IFFYT(I,3)=4
      IF (MBRO(I,13).LE.(FFIDW*FFMBRO).AND.
     +    MBRO(I,13).GE.(FFIDD*FFMBRO)) IFFYT(I,3)=5
      IF (MBRO(I,13).LT.(FFIDD*FFMBRO)) IFFYT(I,3)=6
  404 CONTINUE
C
      DO 405 I=(IYRST+IRSCYC-1),IYREND,IRSCYC
      IF (MBRO(I,13).GT.(FFIDW*FFMBRO)) IFFYT(I,4)=4
      IF (MBRO(I,13).LE.(FFIDW*FFMBRO).AND.
     +    MBRO(I,13).GE.(FFIDD*FFMBRO)) IFFYT(I,4)=5
      IF (MBRO(I,13).LT.(FFIDD*FFMBRO)) IFFYT(I,4)=6
```

```
  405 CONTINUE
C
C******* SET MONO LAKE YEAR TYPE WITH MLBRO
C
      DO 406 I=40,89
      IF (MBRO(I,13).GT.(MLIDW*MLMBRO)) IMLYT(I)=1
      IF (MBRO(I,13).LE.(MLIDW*MLMBRO).AND.
     +    MBRO(I,13).GE.(MLIDD*MLMBRO)) IMLYT(I)=2
      IF (MBRO(I,13).LT.(MLIDD*MLMBRO)) IMLYT(I)=3
  406 CONTINUE
C
C******* SET GRANT STORAGE, LONG VALLEY STORAGE, HAIWEE
C         OUTFLOW AND USE REDUCTION FACTOR INDICES WITH
C         TOTAL OV/MB RUNOFF
C
      IRFC=0
      DO 410 I=40,89
      IF (MBOVRO(I,13).GT.(WET*AVOMRO)) THEN
         IGSYT(I)=1
         ILVYT(I)=1
         IHCYT(I)=1
         IRFC=0
      END IF
C
      IF (MBOVRO(I,13).LE.(WET*AVOMRO).AND.
     +    MBOVRO(I,13).GE.(DRY*AVOMRO)) THEN
         IGSYT(I)=2
         ILVYT(I)=2
         IHCYT(I)=2
         IRFC=0
      END IF
C
      IF (MBOVRO(I,13).LT.(DRY*AVOMRO)) THEN
         IGSYT(I)=3
         ILVYT(I)=3
         IHCYT(I)=3
         IRFC=IRFC+1
         IF (IRFC.GT.5) IRFC=5
      END IF
C
      IF (IRFC.GT.0) THEN
         REDFAC(I)=1.0-USEDRY(IRFC)
         IF (IRFC.EQ.1) PCAP(I)=0.9000
         IF (IRFC.EQ.2) PCAP(I)=0.8100
         IF (IRFC.EQ.3) PCAP(I)=0.7290
         IF (IRFC.EQ.4) PCAP(I)=0.6561
         IF (IRFC.GE.5) PCAP(I)=0.5905
      ELSE
         REDFAC(I)=0
         PCAP(I)=1.00
      END IF
C
  410 CONTINUE
C
```

```
C****** ANNUAL PUMPING LIMIT (LA'S REGRESSIONS)
C
      IF (IDOVP.EQ.1) THEN
         DO 420 I=40,89
         TOVPUMP=(1/((2.64311E-11*OVRO(I,13))-2.9124E-6))
         BBPUMP=6259.9+(0.421044*TOVPUMP)
         TMLAW=(0.524566*BBPUMP)-10786.67
         IF (OVRO(I,13).LE.657550) THEN
            TMBIS=37500-(0.05703*OVRO(I,13))
         ELSE
            TMBIS=0
         END IF
         TMBP=-7473.05+(-1.08661*TMLAW)
     +                 +(0.9429*BBPUMP)
     +                 +(0.012942*220000)
     +                 +(0.00468*OVRO(I,13))
         TMTH=(0.57728*TOVPUMP)-8034.57
C
         TMLAW2=MAX(LAWGW(2,13),TMLAW)
         MAXLAW(I,13)=MIN(LAWGW(1,13),TMLAW2)
C
         TMBIS2=MAX(BISGW(2,13),TMBIS)
         MAXBIS(I,13)=MIN(BISGW(1,13),TMBIS2)
C
         TMBP2=MAX(BIGGW(2,13),TMBP)
         MAXBP(I,13)=MIN(BIGGW(1,13),TMBP2)
C
         TMTH2=MAX(THGW(2,13),TMTH)
         MAXTH(I,13)=MIN(THGW(1,13),TMBP2)
C
C******** MONTHLY PUMPING LIMIT
C
         DO 421 J=1,12
         IF (OVRO(I,13).LT.493150) IDP=1
         IF (OVRO(I,13).GE.493150.AND.OVRO(I,13).LE.657550) IDP=2
         IF (OVRO(I,13).GT.657550) IDP=3
C
         TMLAW=MAX(MAXLAW(I,13)*LPMP(IDP,J)/100,LAWGW(2,J))
         MAXLAW(I,J)=MIN(TMLAW,LAWGW(1,J))
C
         TMBIS=MAX(MAXBIS(I,13)*BPMP(IDP,J)/100,BISGW(2,J))
         MAXBIS(I,J)=MIN(TMBIS,BISGW(1,J))
C
         TMBP=MAX(MAXBP(I,13)*BPPMP(IDP,J)/100,BIGGW(2,J))
         MAXBP(I,J)=MIN(TMBP,BIGGW(1,J))
C
         TMTH=MAX(MAXTH(I,13)*THPMP(IDP,J)/100,THGW(2,J))
         MAXTH(I,J)=MIN(TMTH,THGW(1,J))
C
  421    CONTINUE
  420    CONTINUE
C
      END IF
C
```

```
C****** USER DEFINED MAXIMUM AND MINIMUM PUMPING
C
      IF (IDOVP.EQ.3) THEN
         DO 500 I=40,89
         DO 501 J=1,13
C
         MAXLAW(I,J)=LAWGW(1,J)*PCAP(I)
         MAXBIS(I,J)=BISGW(1,J)*PCAP(I)
         MAXBP(I,J)=BIGGW(1,J)*PCAP(I)
         MAXTH(I,J)=THGW(1,J)*PCAP(I)
C
  501    CONTINUE
  500    CONTINUE
C
      END IF
C
      RETURN
      END
```

```
      SUBROUTINE BATHY (ID,PARAM,X1,X2)
      IMPLICIT REAL (A-H,L-Z)
      DIMENSION BATH(170,3)
      COMMON /BATHYM/ GLBATHY(73,3), LVBATHY(97,3), MNBATHY(170,3)
C
C   Variable IA is a flag which controls which lake
C   bathymetry parameter is calculated
C         IA = 1  refers to elevation
C         IA = 2  refers to area
C         IA = 3  refers to volume
C
      IA=3
C
      IF (ID.EQ.1) THEN
         IME=170
         IKEY=2
      END IF
C
      IF (ID.EQ.2) THEN
         IME=73
         IKEY=0
      END IF
C
      IF (ID.EQ.3) THEN
         IME=97
         IKEY=0
      END IF
C
      IF (ID.EQ.4) THEN
         IME=170
         IKEY=0
      END IF
C
      DO 10 IM=2,IME
C
C           Determine which reservoir or lake is utilizing this
C            subroutine and initialize appropriate interpolation
C            variables
C
      IF (ID.EQ.1.OR.ID.EQ.4) THEN
         BATH(IM,IA)=MNBATHY(IM,IA)
         BATH(IM-1,IA)=MNBATHY(IM-1,IA)
         BATH(IM,IA-2)=MNBATHY(IM,IA-2)
         BATH(IM-1,IA-2)=MNBATHY(IM-1,IA-2)
         BATH(IM,IA-1)=MNBATHY(IM,IA-1)
         BATH(IM-1,IA-1)=MNBATHY(IM-1,IA-1)
      ENDIF
C
      IF (ID.EQ.2) THEN
         BATH(IM,IA)=GLBATHY(IM,IA)
         BATH(IM-1,IA)=GLBATHY(IM-1,IA)
         BATH(IM,IA-2)=GLBATHY(IM,IA-2)
         BATH(IM-1,IA-2)=GLBATHY(IM-1,IA-2)
         BATH(IM,IA-1)=GLBATHY(IM,IA-1)
```

```
          BATH(IM-1,IA-1)=GLBATHY(IM-1,IA-1)
        ENDIF
C
        IF (ID.EQ.3) THEN
          BATH(IM,IA)=LVBATHY(IM,IA)
          BATH(IM-1,IA)=LVBATHY(IM-1,IA)
          BATH(IM,IA-2)=LVBATHY(IM,IA-2)
          BATH(IM-1,IA-2)=LVBATHY(IM-1,IA-2)
          BATH(IM,IA-1)=LVBATHY(IM,IA-1)
          BATH(IM-1,IA-1)=LVBATHY(IM-1,IA-1)
        ENDIF
C
        IF(BATH(IM,IA-IKEY).LT.PARAM) THEN
           GO TO 10
        ELSE
C
C                 Interpolating scheme
C
           IF (ID.GT.1) THEN
              CVOL = (BATH(IM,IA) - PARAM)/(BATH(IM,IA) -
     +                BATH(IM-1,IA))
              ELEV = BATH(IM,IA-2) - CVOL*(BATH(IM,IA-2) -
     +                BATH(IM-1,IA-2))
              AREA = BATH(IM,IA-1) - CVOL*(BATH(IM,IA-1) -
     +                BATH(IM-1,IA-1))
              X1=ELEV
              X2=AREA
              GO TO 15
           ELSE
              CELEV= (BATH(IM,IA-2) - PARAM)/(BATH(IM,IA-2) -
     +                BATH(IM-1,IA-2))
              VOL  = BATH(IM,IA) - CELEV*(BATH(IM,IA) -
     +                BATH(IM-1,IA))
              AREA = BATH(IM,IA-1) - CELEV*(BATH(IM,IA-1) -
     +                BATH(IM-1,IA-1))
              X1=VOL
              X2=AREA
              GO TO 15
           END IF
C
        END IF
   10   CONTINUE
   15   RETURN
        END
```

```
      SUBROUTINE MLTAR (I)
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
C
C***** MONO LAKE RELEASE TARGETS
C
      IF (MLL(I,1).GE.LLS(3)) THEN
         LRT(I)=-1
         GOTO 100
      END IF
C
      IF (MLL(I,1).LT.LLS(3)) IDLL(I)=3
      IF (MLL(I,1).LT.LLS(2)) IDLL(I)=2
      IF (MLL(I,1).LT.LLS(1)) IDLL(I)=1
C
      IF (LLRF(IDLL(I),IMLYT(I)).LT.0.01.AND.
     1    LLRF(IDLL(I),IMLYT(I)).GT.-0.01) THEN
         LRT(I)=-1
         GOTO 100
      END IF
C
      IF (LLRF(IDLL(I),IMLYT(I)).GT.1.00) THEN
         LRT(I)=LLRF(IDLL(I),IMLYT(I))
      END IF
C
      IF (LLRF(IDLL(I),IMLYT(I)).GT.0.AND.
     1    LLRF(IDLL(I),IMLYT(I)).LE.1) THEN
         LRT(I)=LLRF(IDLL(I),IMLYT(I))*MBRO(I,13)
      END IF
C
      IF (LLRF(IDLL(I),IMLYT(I)).LT.0) THEN
         LRT(I)=MBRO(I,13)+LLRF(IDLL(I),IMLYT(I))
      END IF

      IF (IFFYT(I,1).GT.3) THEN
         LRT(I)=LRT(I)-BLVFF(IFFYT(I,1)-3,13)-FLVFF(IFFYT(I,1),13)
      ELSE
         LRT(I)=LRT(I)-BLVFF(IFFYT(I,1),13)
      END IF

      IF (IFFYT(I,2).GT.3) THEN
         LRT(I)=LRT(I)-BWFF(IFFYT(I,2)-3,13)-FWFF(IFFYT(I,2),13)
      ELSE
         LRT(I)=LRT(I)-BWFF(IFFYT(I,2),13)
      END IF

      IF (IFFYT(I,3).GT.3) THEN
         LRT(I)=LRT(I)-BPFF(IFFYT(I,3)-3,13)-FPFF(IFFYT(I,3),13)
      ELSE
         LRT(I)=LRT(I)-BPFF(IFFYT(I,3),13)
      END IF

      IF (IFFYT(I,4).GT.3) THEN
         LRT(I)=LRT(I)-BRSFF(IFFYT(I,4)-3,13)-FRSFF(IFFYT(I,4),13)
```

```
      ELSE
         LRT(I)=LRT(I)-BRSFF(IFFYT(I,4),13)
      END IF
C
  100 OLRT(I)=LRT(I)
C
      RETURN
      END
```

```
              T1=AWFF(II,JJ)
              T2=WSPILL(II,JJ)
              T4=T1+T2
              HIFLO=MAX(HIFLO,T4)
              JJ=JJ+1
  201        CONTINUE
             IF (HIFLO.GT.FWFF(IFFYT(I,2),J)) THEN
                WWFF=BWFF(IFFYT(I,2)-3,J)
                IF (LRT(I).GT.-1) LRT(I)=LRT(I)+FWFF(IFFYT(I,2),J)
             END IF
          END IF
       END IF
C
C*** PARKER
C
       IF (IFFYT(I,3).LE.3) THEN
          WPFF=BPFF(IFFYT(I,3),J)
       ELSE
          WPFF=BPFF(IFFYT(I,3)-3,J)+FPFF(IFFYT(I,3),J)
          IF (I.GT.IYRST+IPCYC.AND.FPFF(IFFYT(I,3),J).GT.0) THEN
             JJ=J+1
             II=I-IPCYC
             HIFLO=0
             DO 202 K=1,(IPCYC*12)-1
              IF (JJ.EQ.13) THEN
                 II=II+1
                 JJ=1
              END IF
              T1=APFF(II,JJ)
              T2=PSPILL(II,JJ)
              T4=T1+T2
              HIFLO=MAX(HIFLO,T4)
              JJ=JJ+1
  202         CONTINUE
              IF (HIFLO.GT.FPFF(IFFYT(I,3),J)) THEN
                 WPFF=BPFF(IFFYT(I,3)-3,J)
                 IF (LRT(I).GT.-1) LRT(I)=LRT(I)+FPFF(IFFYT(I,3),J)
              END IF
          END IF
       END IF
C
C*** RUSH
C
       IF (IFFYT(I,4).LE.3) THEN
          WRSFF=BRSFF(IFFYT(I,4),J)
       ELSE
          WRSFF=BRSFF(IFFYT(I,4)-3,J)+FRSFF(IFFYT(I,4),J)
          IF (I.GT.IYRST+IRSCYC.AND.FRSFF(IFFYT(I,4),J).GT.0) THEN
             JJ=J+1
             II=I-IRSCYC
             HIFLO=0
             DO 203 K=1,(IRSCYC*12)-1
              IF (JJ.EQ.13) THEN
                 II=II+1
```

```
              JJ=1
            END IF
            T1=AMG1FF(II,JJ)
            T2=RSLR(II,JJ)
            T3=GSPILL(II,JJ)
            T4=GFFREL(II,JJ)
            T5=GLREL(II,JJ)
            T6=T1+T2+T3+T4+T5
            HIFLO=MAX(HIFLO,T6)
            JJ=JJ+1
203       CONTINUE
          IF (HIFLO.GT.FRSFF(IFFYT(I,4),J)) THEN
            WRSFF=BRSFF(IFFYT(I,4)-3,J)
            IF (LRT(I).GT.-1) LRT(I)=LRT(I)+FRSFF(IFFYT(I,4),J)
          END IF
        END IF
      END IF
C
      RETURN
      END
```

```
      SUBROUTINE MBFISH (I,J,WLVFF,WWFF,WPFF,WMG1FF,LVFD,WFD,PFD,MG1FD,
     +              LVAVAIL,WAVAIL,PAVAIL,RSAVAIL,IDRFF)
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
C
C** LEE VINING
C
      IF (LVAVAIL*0.9.GE.WLVFF) THEN
         ALVFF(I,J)=WLVFF
         LVAVAIL=LVAVAIL-ALVFF(I,J)
      ELSE
         ALVFF(I,J)=LVAVAIL
         LVAVAIL=0
         LVFD=WLVFF-ALVFF(I,J)
      END IF
      ALVFF(I,13)=ALVFF(I,13)+ALVFF(I,J)
      IF (LEEMAX(I,J).GT.ALVFF(I,J)) THEN
         LEEMAX(I,J)=LEEMAX(I,J)-ALVFF(I,J)
      ELSE
         EA=ALVFF(I,J)-LEEMAX(I,J)
         LVAVAIL=LVAVAIL+EA
         ALVFF(I,J)=LEEMAX(I,J)
         LEEMAX(I,J)=0
      END IF
C
C** WALKER
C
      IF (WAVAIL*0.9.GE.WWFF) THEN
         AWFF(I,J)=WWFF
         WAVAIL=WAVAIL-AWFF(I,J)
      ELSE
         AWFF(I,J)=WAVAIL
         WAVAIL=0
         WFD=WWFF-AWFF(I,J)
      END IF
      AWFF(I,13)=AWFF(I,13)+AWFF(I,J)
C
C** PARKER
C
      IF (PAVAIL*0.9.GE.WPFF) THEN
         APFF(I,J)=WPFF
         PAVAIL=PAVAIL-APFF(I,J)
      ELSE
         APFF(I,J)=PAVAIL
         PAVAIL=0
         PFD=WPFF-APFF(I,J)
      END IF
      APFF(I,13)=APFF(I,13)+APFF(I,J)
C
C** RUSH
C
C***** CHECK IDRFF TOGGLE FOR USE OF STORAGE IN MEETING REQTS.
C
C  IF IDRFF=1  USE ONLY RUSH CK ABOVE GRANT
```

```
C  IF IDRFF=2  MAY USE GRANT STORAGE IF CREEK FLOW IS NOT ENOUGH
C
      IF (RSAVAIL.GE.WMG1FF) THEN
         AMG1FF(I,J)=WMG1FF
         RSAVAIL=RSAVAIL-AMG1FF(I,J)
      ELSE
         IF (IDRFF.EQ.1) THEN
            AMG1FF(I,J)=RSAVAIL
            RSAVAIL=0
            MG1FD=WMG1FF-AMG1FF(I,J)
         ELSE
            IF (GSTOR(I,J)-GEVAP(I,J).GT.GSTMIN(IGSYT(I),J)) THEN
               TAV=RSAVAIL+GSTOR(I,J)-GEVAP(I,J)-GSTMIN(IGSYT(I),J)
               IF (TAV.GE.WMG1FF) THEN
                  AMG1FF(I,J)=WMG1FF
                  GFFREL(I,J)=AMG1FF(I,J)-RSAVAIL
                  RSAVAIL=0
               ELSE
                  AMG1FF(I,J)=TAV
                  GFFREL(I,J)=AMG1FF(I,J)-RSAVAIL
                  RSAVAIL=0
                  MG1FD=WMG1FF-TAV
               END IF
            ELSE
               AMG1FF(I,J)=RSAVAIL
               RSAVAIL=0
               MG1FD=WMG1FF-RSAVAIL
            END IF
         END IF
      END IF
C
      IF (GOUTMAX(I,J).GT.AMG1FF(I,J)) THEN
         GOUTMAX(I,J)=GOUTMAX(I,J)-AMG1FF(I,J)
      ELSE
         AMG1FF(I,J)=GOUTMAX(I,J)
         GOUTMAX(I,J)=0
      END IF
      IF (MG1MAX(I,J).GT.AMG1FF(I,J)) THEN
         MG1MAX(I,J)=MG1MAX(I,J)-AMG1FF(I,J)
      ELSE
         AMG1FF(I,J)=MG1MAX(I,J)
         MG1MAX(I,J)=0
      END IF
      AMG1FF(I,13)=AMG1FF(I,13)+AMG1FF(I,J)
      TFR(I,J)=ALVFF(I,J)+AWFF(I,J)+APFF(I,J)+AMG1FF(I,J)
      TFR(I,13)=TFR(I,13)+TFR(I,J)
C
      RETURN
      END
```

```
      SUBROUTINE MBFLUSH (I,J,WLVFF,WWFF,WPFF,WRSFF,ILVCYC,IWCYC,
     +   IPCYC,IRSCYC,IYRST)
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
C
C** LEE VINING
C
      IF (IFFYT(I,1).LE.3) THEN
         WLVFF=BLVFF(IFFYT(I,1),J)
      ELSE
         WLVFF=BLVFF(IFFYT(I,1)-3,J)+FLVFF(IFFYT(I,1),J)
         IF (I.GT.IYRST+ILVCYC.AND.FLVFF(IFFYT(I,1),J).GT.0) THEN
            JJ=J+1
            II=I-ILVCYC
            HIFLO=0
            DO 200 K=1,(ILVCYC*12)-1
             IF (JJ.EQ.13) THEN
                II=II+1
                JJ=1
             END IF
             T1=ALVFF(II,JJ)
             T2=LVSPILL(II,JJ)
             T3=LVLR(II,JJ)
             T4=T1+T2+T3
             HIFLO=MAX(HIFLO,T4)
             JJ=JJ+1
  200       CONTINUE
            IF (HIFLO.GT.FLVFF(IFFYT(I,1),J)) THEN
               WLVFF=BLVFF(IFFYT(I,1)-3,J)
               IF (LRT(I).GT.-1) LRT(I)=LRT(I)+FLVFF(IFFYT(I,1),J)
            END IF
         END IF
      END IF
C
C*** CREDIT LEE VINING WITH SPILL
C
      WLVFF=WLVFF-LVSPILL(I,J)
      IF (WLVFF.LT.0) WLVFF=0
C
C*** WALKER
C
      IF (IFFYT(I,2).LE.3) THEN
         WWFF=BWFF(IFFYT(I,2),J)
      ELSE
         WWFF=BWFF(IFFYT(I,2)-3,J)+FWFF(IFFYT(I,2),J)
         IF (I.GT.IYRST+IWCYC.AND.FWFF(IFFYT(I,2),J).GT.0) THEN
            JJ=J+1
            II=I-IWCYC
            HIFLO=0
            DO 201 K=1,(IWCYC*12)-1
             IF (JJ.EQ.13) THEN
                II=II+1
                JJ=1
             END IF
```

```
      SUBROUTINE MLR (I,J,LVFD,WFD,PFD,MG1FD,LVAVAIL,RSAVAIL)
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
C
C   PRIORITY OF RELEASES
C
C       1.  GRANT STORAGE ABOVE MINIMUM
C       2.  RUSH CREEK
C       3.  LEE VINING CREEK
C
C*** ADD FISH FLOW DEFICITS TO TARGET
C
      IF (LRT(I).GE.0) LRT(I)=LRT(I)+LVFD+WFD+PFD+MG1FD
C
C*** GRANT STORAGE RELEASE
C
      IF (LRT(I).GT.0) THEN
         TAV=GSTOR(I,J)-GEVAP(I,J)-GSTMIN(IGSYT(I),J)
         IF (TAV.GT.0) THEN
            GLREL(I,J)=MIN(LRT(I),TAV)
            IF (GOUTMAX(I,J).GT.GLREL(I,J)) THEN
               GOUTMAX(I,J)=GOUTMAX(I,J)-GLREL(I,J)
            ELSE
               GLREL(I,J)=GOUTMAX(I,J)
               GOUTMAX(I,J)=0
            END IF
            IF (MG1MAX(I,J).GT.GLREL(I,J)) THEN
               MG1MAX(I,J)=MG1MAX(I,J)-GLREL(I,J)
            ELSE
               GLREL(I,J)=MG1MAX(I,J)
               MG1MAX(I,J)=0
            END IF
            GLREL(I,13)=GLREL(I,13)+GLREL(I,J)
            LRT(I)=LRT(I)-GLREL(I,J)
         END IF
      END IF
C
      IF (LRT(I).GT.0) THEN
         RSLR(I,J)=MIN(LRT(I),RSAVAIL)
         IF (GOUTMAX(I,J).GT.GLREL(I,J)) THEN
            GOUTMAX(I,J)=GOUTMAX(I,J)-RSLR(I,J)
         ELSE
            RSLR(I,J)=GOUTMAX(I,J)
            GOUTMAX(I,J)=0
         END IF
         IF (MG1MAX(I,J).GT.RSLR(I,J)) THEN
            MG1MAX(I,J)=MG1MAX(I,J)-RSLR(I,J)
         ELSE
            RSLR(I,J)=MG1MAX(I,J)
            MG1MAX(I,J)=0
         END IF
         LRT(I)=LRT(I)-RSLR(I,J)
         RSAVAIL=RSAVAIL-RSLR(I,J)
         RSLR(I,13)=RSLR(I,13)+RSLR(I,J)
```

```
      END IF
C
      IF (LRT(I).GT.0) THEN
         LVLR(I,J)=MIN(LRT(I),LVAVAIL)
         IF (LEEMAX(I,J).GT.LVLR(I,J)) THEN
            LEEMAX(I,J)=LEEMAX(I,J)-LVLR(I,J)
         ELSE
            LVLR(I,J)=LEEMAX(I,J)
            LEEMAX(I,J)=0
         END IF
         LRT(I)=LRT(I)-LVLR(I,J)
         LVAVAIL=LVAVAIL-LVLR(I,J)
         LVLR(I,13)=LVLR(I,13)+LVLR(I,J)
      END IF
      ALR(I,J)=ALR(I,J)+GLREL(I,J)+RSLR(I,J)+LVLR(I,J)
      ALR(I,13)=ALR(I,13)+ALR(I,J)
C
      RETURN
      END
```

```
      SUBROUTINE MBIRRIG (I,J,LVAVAIL,WAVAIL,PAVAIL,LVCMAX,WMAX,CAGMAX)
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
C
C****************** GIBBS CREEK DIVERSION
C
      IF (LVAVAIL.GE.(GIBDIV(J))) THEN
         AGIBDIV(I,J)=GIBDIV(J)
         LVAVAIL=LVAVAIL-AGIBDIV(I,J)
      ELSE
         AGIBDIV(I,J)=LVAVAIL
         LVAVAIL=0
      END IF
      AGIBDIV(I,13)=AGIBDIV(I,13)+AGIBDIV(I,J)
C
C******************* GIBBS FARRINGTON SIPHON
C
      IF (LVAVAIL.GE.(GIBFAR(J))) THEN
         AGIBFAR(I,J)=GIBFAR(J)
         LVAVAIL=LVAVAIL-AGIBFAR(I,J)
      ELSE
         AGIBFAR(I,J)=LVAVAIL
         LVAVAIL=0
      END IF
      LVCMAX=LVCMAX-AGIBFAR(I,J)
      AGIBFAR(I,13)=AGIBFAR(I,13)+AGIBFAR(I,J)
C
C******************* WALKER CREEK IRRIGATION RELEASE
C
      IF (WAVAIL.GE.(WALIRR(J))) THEN
         AWALIRR(I,J)=WALIRR(J)
         WAVAIL=WAVAIL-AWALIRR(I,J)
      ELSE
         AWALIRR(I,J)=WAVAIL
         WAVAIL=0
      END IF
      AWALIRR(I,13)=AWALIRR(I,13)+AWALIRR(I,J)
C
C****************** SAND TRAP 3
C
      IF (WAVAIL.GE.(SAND3(J))) THEN
         ASAND3(I,J)=SAND3(J)
         WAVAIL=WAVAIL-ASAND3(I,J)
      ELSE
         ASAND3(I,J)=WAVAIL
         WAVAIL=0
      END IF
      WMAX=WMAX-ASAND3(I,J)
      ASAND3(I,13)=ASAND3(I,13)+ASAND3(I,J)
C
C****************** PARKER CREEK DIVERSIONS
C
      IF (PAVAIL.GE.(PARDIV(J))) THEN
         APARDIV(I,J)=PARDIV(J)
```

```
         PAVAIL=PAVAIL-APARDIV(I,J)
      ELSE
         APARDIV(I,J)=PAVAIL
         PAVAIL=0
      END IF
      APARDIV(I,13)=APARDIV(I,13)+APARDIV(I,J)
C
C****************** PARKER CREEK IRRIGATION RELEASES
C
      IF (PAVAIL.GE.(PARIRR(J))) THEN
         APARIRR(I,J)=PARIRR(J)
         PAVAIL=PAVAIL-APARIRR(I,J)
      ELSE
         APARIRR(I,J)=PAVAIL
         PAVAIL=0
      END IF
      APARIRR(I,13)=APARIRR(I,13)+APARIRR(I,J)
C
C****************** SAND TRAP 4
C
      IF (PAVAIL.GE.(SAND4(J))) THEN
         ASAND4(I,J)=SAND4(J)
         PAVAIL=PAVAIL-ASAND4(I,J)
      ELSE
         ASAND4(I,J)=PAVAIL
         PAVAIL=0
      END IF
      CAGMAX=CAGMAX-ASAND4(I,J)
      ASAND4(I,13)=ASAND4(I,13)+ASAND4(I,J)
C
C******* TOTAL IRRIGATION
C
      MBIRR(I,J)=ODITCH(J)+AGIBDIV(I,J)+AGIBFAR(I,J)+AWALIRR(I,J)
     +            +ASAND3(I,J)+APARDIV(I,J)+APARIRR(I,J)+ASAND4(I,J)
C
      RETURN
      END
```

```
      SUBROUTINE CONDUIT (I,J,LVAVAIL,WAVAIL,PAVAIL,RSAVAIL,
     +   LVCMAX,WMAX,CAGMAX)
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
C
C***** LEE VINING
C
      LVTCON(I,J)=LVAVAIL
      LVAVAIL=0
      WMAX=WMAX-LVTCON(I,J)
C
C****** WALKER
C
      IF (WAVAIL.GT.WMAX) THEN
         WSPILL(I,J)=WAVAIL-WMAX
         WTCON(I,J)=WAVAIL-WSPILL(I,J)
      ELSE
         WTCON(I,J)=WAVAIL
      END IF
      WAVAIL=0
      CAGMAX=CAGMAX-(LVTCON(I,J)+WTCON(I,J))
C
C****** PARKER
C
      IF (PAVAIL.GT.CAGMAX) THEN
         PSPILL(I,J)=PAVAIL-CAGMAX
         PTCON(I,J)=PAVAIL-PSPILL(I,J)
      ELSE
         PTCON(I,J)=PAVAIL
      END IF
      PAVAIL=0
      CONAG(I,J)=LVTCON(I,J)+WTCON(I,J)+PTCON(I,J)
      TSPILL(I,J)=LVSPILL(I,J)+WSPILL(I,J)+PSPILL(I,J)
C
C**** ACCUMULATE SPILLS AND CONAG
C
      LVSPILL(I,13)=LVSPILL(I,13)+LVSPILL(I,J)
      WSPILL(I,13)=WSPILL(I,13)+WSPILL(I,J)
      PSPILL(I,13)=PSPILL(I,13)+PSPILL(I,J)
      TSPILL(I,13)=TSPILL(I,13)+TSPILL(I,J)
      CONAG(I,13)=CONAG(I,13)+CONAG(I,J)
C
C**** CALCULATE GRANT INFLOW
C
      RUSHTG(I,J)=RSAVAIL
      GIN(I,J)=RUSHTG(I,J)+CONAG(I,J)
      GIN(I,13)=GIN(I,13)+GIN(I,J)
C
      RETURN
      END
```

```
      SUBROUTINE WP (I,J)
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
C
C**** MINIMUM WEST PORTAL FLOW
C
      TGS=0
      IF (GOUTMAX(I,J).GT.0.AND.WPFMAX(I,J).GT.0) THEN
         MINWPF(I,J)=MAX(0,UORMIN(J)-ORAEP(I,J)-MTMAKE(I,J))
         TGS=GSTOR(I,J)-GEVAP(I,J)+GIN(I,J)-GFFREL(I,J)-GLREL(I,J)
     +                   -MINWPF(I,J)
         IF (TGS.LT.GSTMIN(IGSYT(I),J)) THEN
            DIFF=GSTMIN(IGSYT(I),J)-TGS
            IF (MINWPF(I,J).GT.DIFF) THEN
               MINWPF(I,J)=MINWPF(I,J)-DIFF
               TGS=GSTOR(I,J)-GEVAP(I,J)+GIN(I,J)-GFFREL(I,J)
     +                      -GLREL(I,J)-MINWPF(I,J)
            ELSE
               MINWPF(I,J)=0
               TGS=GSTOR(I,J)-GEVAP(I,J)+GIN(I,J)-GFFREL(I,J)
     +                      -GLREL(I,J)
            END IF
         END IF
         IF (GOUTMAX(I,J).GT.MINWPF(I,J)) THEN
            GOUTMAX(I,J)=GOUTMAX(I,J)-MINWPF(I,J)
         ELSE
            MINWPF(I,J)=GOUTMAX(I,J)
            GOUTMAX(I,J)=0
         END IF
         IF (WPFMAX(I,J).GT.MINWPF(I,J)) THEN
            WPFMAX(I,J)=WPFMAX(I,J)-MINWPF(I,J)
         ELSE
            MINWPF(I,J)=WPFMAX(I,J)
            WPFMAX(I,J)=0
         END IF
      END IF
C
C******* CALCULATE AMOUNT AVAILABLE TO WEST PORTAL FLOW
C
C
C** PHYSICAL, FLOODING, AND LAKE RELEASE CONSTRAINTS
C
      STAV=MAX(0,TGS-GSTMIN(IGSYT(I),J))
      FLLIM=MAX(0,UORMAX(J)-ORAEP(I,J)-(MINWPF(I,J)+MTMAKE(I,J)))
      ATWP(I,J)=MIN(STAV,FLLIM)
      IF (ATWP(I,J).GT.GOUTMAX(I,J)) ATWP(I,J)=GOUTMAX(I,J)
      IF (ATWP(I,J).GT.WPFMAX(I,J)) ATWP(I,J)=WPFMAX(I,J)
      IF (LRT(I).GT.0) ATWP(I,J)=0
C
C** ANNUAL WATER RIGHTS CONSTRAINT
C
      IF (WPF(I,13)+ATWP(I,J).GT.167000) THEN
         IF (WPF(I,13).GE.167000) THEN
            ATWP(I,J)=0
```

```
         ELSE
            ATWP(I,J)=167000-WPF(I,J)
         END IF
      END IF
C
      RETURN
      END
```

```
      SUBROUTINE OWENS (I,J)
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
C
C************************   LONG VALLEY   ****************************
C
C**** ROCK CREEK DIVERSION
C
      IF (RCLRV(I,J).GT.RCKFSH(J)) THEN
         RCD(I,J)=0.8*(RCLRV(I,J)-RCKFSH(J))
      ELSE
         RCD(I,J)=0
      END IF
C
C**** LONG VALLEY INFLOW
C
      LVIN(I,J)=LVRO(I,J)+HCSPR(I,J)-LVIRR(J)+RCD(I,J)+MINWPF(I,J)
     +       +MTMAKE(I,J)
     +       +LVGAIN(1,J)
     +       +(LVGAIN(2,J)*LVRO(I,J))
     +       +(LVGAIN(3,J)*LVIRR(J))
C
C***********************    ROUND VALLEY    ***************************
C
      RCTRV(I,J)=RCLRV(I,J)-RCD(I,J)
      RVIN(I,J)=RCTRV(I,J)+(RVRO(I,J)-RCLRV(I,J))+BCSPR(I,J)
      RVUL=RVIRR(J)+RVREC(J)+RVLOS(1,J)+(RVLOS(2,J)*RVRO(I,J))
     +       +(RVLOS(3,J)*RVIRR(J))
      RVOUT(I,J)=MAX(0,RVIN(I,J)-RVUL)
C
C*** RELEASE ENOUGH WATER TO MEET PV MINIMUM OUTFLOW
C
      LVPVT=RVTRAN(1,J)+(RVTRAN(2,J)*LVSTOR(I,J))+RVOUT(I,J)
      LVRMIN=MAX(0,PVMIN(J)-LVPVT)
      LVOUT(I,J)=LVRMIN
      IF (LVOUT(I,J).GT.LVCAP(J)) LVOUT(I,J)=LVCAP(J)
      PVOUT(I,J)=LVOUT(I,J)+LVPVT
C
C*** RELEASE WATER IF NEEDED TO MATCH TARGET EOM STORAGE
C
      TLVST=LVSTOR(I,J)-LVEV(I,J)+LVIN(I,J)-LVOUT(I,J)
      IF (TLVST.GT.LVSMAX(ILVYT(I),J)) THEN
         LVDUMP=TLVST-LVSMAX(ILVYT(I),J)
         LVRTMT=MIN(LVDUMP,LVCAP(J)-LVOUT(I,J))
         LVOUT(I,J)=LVOUT(I,J)+LVRTMT
         PVOUT(I,J)=PVOUT(I,J)+LVRTMT
      END IF
C
C***************** BISHOP
C
C
C*** USES AND LOSSES
C
      BUSE=BISIRR(J)+BISREC(J)
```

```
      BLOSS=BISLOS(3,J)*BUSE
      BUL=BUSE+BLOSS
C
C*** PUMPING
C
      UBISP(I,J)=BUSEP(IHCYT(I),J)*BUL
      IF (UBISP(I,J).LT.BISGW(2,J)) UBISP(I,J)=BISGW(2,J)
      UBISP(I,J)=UBISP(I,J)+(BISLOS(4,J)*UBISP(I,J))
C
C*** OWENS RIVER DIVERSION
C
      BISORD(I,J)=(1-BUSEP(IHCYT(I),J))*BUL
C
C*** CHECK DIVERSION WITH PVOUT
C
      MAXDIV=MAX(0,PVOUT(I,J)-PVMIN(J))
      IF (BISORD(I,J).GT.MAXDIV) THEN
         DIFF=BISORD(I,J)-MAXDIV
         UBISP(I,J)=UBISP(I,J)+DIFF
         BISORD(I,J)=MAXDIV
      END IF
C
C*** CHECK PUMPING WITH MAX PUMPING
C
      MAXP=MAXBIS(I,J)-(BISLOS(4,J)*UBISP(I,J))
      IF (UBISP(I,J).GT.MAXP) UBISP(I,J)=MAXP
      IF (UBISP(I,J).LT.BISGW(2,J)) UBISP(I,J)=BISGW(2,J)
C
C*** BISHOP AREA INFLOW AND OUTFLOW
C
      BISIN(I,J)=BISACT(I,J)+BISSPR(J)+BISORD(I,J)+UBISP(I,J)
      ABUSE(I,J)=UBISP(I,J)+BISORD(I,J)
      IF (ABUSE(I,J).GT.BUSE) ABUSE(I,J)=BUSE
      BUL=ABUSE(I,J)+BLOSS+BISLOS(1,J)
     +              +(BISLOS(2,J)*BISACT(I,J))
     +              +(BISLOS(4,J)*UBISP(I,J))
      BISOUT(I,J)=BISIN(I,J)-BUL
      IF (BISOUT(I,J).LT.0) THEN
         DIFF=BUL-BISIN(I,J)
         UBISP(I,J)=UBISP(I,J)+DIFF
         BISIN(I,J)=BISIN(I,J)+DIFF
         BISOUT(I,J)=0
      END IF
C
C*** CALCULATE ACTUAL USES
C
      ABUSE(I,J)=UBISP(I,J)+BISORD(I,J)
      IF (ABUSE(I,J).GT.BUSE) ABUSE(I,J)=BUSE
C
C*************** LAWS
C
C*** USES AND LOSSES
C
      LUSE=LAWIRR(J)+LAWREC(J)
```

```
      LLOSS=LAWLOS(3,J)*LUSE
      LUL=LUSE+LLOSS
C
C*** PUMPING
C
      ULAWP(I,J)=LUSEP(IHCYT(I),J)*LUL
      IF (ULAWP(I,J).LT.LAWGW(2,J)) ULAWP(I,J)=LAWGW(2,J)
      ULAWP(I,J)=ULAWP(I,J)+(LAWLOS(5,J)*ULAWP(I,J))
C
C*** OWENS RIVER DIVERSION INTO MCNALLY CANALS
C
      LORDIV(I,J)=(1-LUSEP(IHCYT(I),J))*LUL
      LORDIV(I,J)=LORDIV(I,J)+(LAWLOS(4,J)*LORDIV(I,J))
C
C*** CHECK DIVERSION WITH PVOUT, IF TOO HIGH INCREASE PUMPING
C
      MAXDIV=MAX(0,PVOUT(I,J)-PVMIN(J)-BISORD(I,J))
      MAXDIV=MAX(0,MAXDIV-(LAWLOS(4,J)*MAXDIV))
      IF (LORDIV(I,J).GT.MAXDIV) THEN
         DIFF=LORDIV(I,J)-MAXDIV
         ULAWP(I,J)=ULAWP(I,J)+DIFF
         ULAWP(I,J)=ULAWP(I,J)+(LAWLOS(5,J)*ULAWP(I,J))
         LORDIV(I,J)=MAXDIV
      END IF
C
C*** CHECK PUMPING WITH MAXIMUM PUMPING
C
      MAXP=MAXLAW(I,J)-(LAWLOS(5,J)*MAXLAW(I,J))
      IF (ULAWP(I,J).GT.MAXP) ULAWP(I,J)=MAXP
      IF (ULAWP(I,J).LT.LAWGW(2,J)) ULAWP(I,J)=LAWGW(2,J)
C
C*** LAWS AREA INFLOW AND OUTFLOW
C
      LAWSIN(I,J)=LWRO(I,J)+ULAWP(I,J)+LORDIV(I,J)+FSLU(I,J)
      ALUSE(I,J)=ULAWP(I,J)+LORDIV(I,J)
      IF (ALUSE(I,J).GT.LUSE) ALUSE(I,J)=LUSE
      LUL=ALUSE(I,J)+LLOSS+LAWLOS(1,J)
     +              +(LAWLOS(2,J)*LWRO(I,J))
     +              +(LAWLOS(4,J)*LORDIV(I,J))
     +              +(LAWLOS(5,J)*ULAWP(I,J))
      LAWSOUT(I,J)=LAWSIN(I,J)-LUL
      IF (LAWSOUT(I,J).LT.0) THEN
         DIFF=LUL-LAWSIN(I,J)
         ULAWP(I,J)=ULAWP(I,J)+DIFF
         LAWSIN(I,J)=LAWSIN(I,J)+DIFF
         LAWSOUT(I,J)=0
      END IF

 ** CALCULATE ACTUAL USES

      ALUSE(I,J)=ULAWP(I,J)+LORDIV(I,J)
      IF (ALUSE(I,J).GT.LUSE) ALUSE(I,J)=LUSE

 ************* BIG PINE
```

```
C
      BPUSE=BIGIRR(J)+BIGREC(J)
      BPLOSS=BIGLOS(3,J)*BPUSE
      BPUL=BPUSE+BPLOSS
C
C*** PUMPING
C
      UBPP(I,J)=BPUSEP(IHCYT(I),J)*BPUL
      IF (UBPP(I,J).LT.BIGGW(2,J)) UBPP(I,J)=BIGGW(2,J)
      UBPP(I,J)=UBPP(I,J)+(BIGLOS(5,J)*UBPP(I,J))
C
C*** OWENS RIVER DIVERSION
C
      BPORDIV(I,J)=(1-BPUSEP(IHCYT(I),J))*BPUL
      BPORDIV(I,J)=BPORDIV(I,J)+(BIGLOS(4,J)*BPORDIV(I,J))
C
C*** CHECK DIVERSION
C
      MAXDIV=MAX(0,PVOUT(I,J)-PVMIN(J)-LORDIV(I,J)-BISORD(I,J)
     +      +LAWSOUT(I,J)+BISOUT(I,J))
      IF (BPORDIV(I,J).GT.MAXDIV) THEN
         DIFF=BPORDIV(I,J)-MAXDIV
         UBPP(I,J)=UBPP(I,J)+DIFF
         UBPP(I,J)=UBPP(I,J)+(BIGLOS(5,J)*UBPP(I,J))
         BPORDIV(I,J)=MAXDIV
      END IF
C
C*** CHECK PUMPING
C
      MAXP=MAXBP(I,J)-(BIGLOS(5,J)*MAXBP(I,J))
      IF (UBPP(I,J).GT.MAXP) UBPP(I,J)=MAXP
      IF (UBPP(I,J).LT.BIGGW(2,J)) UBPP(I,J)=BIGGW(2,J)
C
C*** BIG PINE AREA INFLOW AND OUTFLOW
C
      BPIN(I,J)=BPRO(I,J)+UBPP(I,J)+BPORDIV(I,J)+XKEO(I,J)
      ABPUSE(I,J)=UBPP(I,J)+BPORDIV(I,J)
      IF (ABPUSE(I,J).GT.BPUSE) ABPUSE(I,J)=BPUSE
      BPUL=ABPUSE(I,J)+BPLOSS
     +                 +BIGLOS(1,J)
     +                 +(BIGLOS(2,J)*BPRO(I,J))
     +                 +(BIGLOS(4,J)*BPORDIV(I,J))
     +                 +(BIGLOS(5,J)*UBPP(I,J))
      BPOUT(I,J)=BPIN(I,J)-BPUL
      IF (BPOUT(I,J).LT.0) THEN
         DIFF=BPUL-BPIN(I,J)
         UBPP(I,J)=UBPP(I,J)+DIFF
         BPIN(I,J)=BPIN(I,J)+DIFF
         BPOUT(I,J)=0
      END IF
C
C*** CALCULATE ACTUAL USES
C
      ABPUSE(I,J)=UBPP(I,J)+BPORDIV(I,J)
```

```
      IF (ABPUSE(I,J).GT.BPUSE) ABPUSE(I,J)=BPUSE
C
C************ TINEMAHA TO HAIWEE
C
      THWUSE=THWIRR(J)+THWREC(J)
      THEUSE=THEIRR(J)+THEREC(J)
      THWLOSS=THLOS(3,J)*THWUSE
      THELOSS=THLOS(3,J)*THEUSE
      THWUL=THWUSE+THWLOSS
      THEUL=THEUSE+THELOSS
C
C*** PUMPING FOR WEST USES
C
      UTHP(I,J)=THWUL
      UTHP(I,J)=UTHP(I,J)+(THLOS(5,J)*UTHP(I,J))
      MAXP=MAXTH(I,J)-(THLOS(5,J)*UTHP(I,J))
      IF (UTHP(I,J).GT.MAXP) UTHP(I,J)=MAXP
      IF (UTHP(I,J).LT.THGW(2,J)) UTHP(I,J)=THGW(2,J)
C
C*** AQUEDUCT RELEASES FOR EAST USES
C
      AVAIL=MAX(0,PVOUT(I,J)+LAWSOUT(I,J)+BISOUT(I,J)+BPOUT(I,J)
     +                 -LORDIV(I,J)-BISORD(I,J)-BPORDIV(I,J))
      IF (AVAIL.GT.THEUL) THEN
         THAR(I,J)=THEUL
      ELSE
         THAR(I,J)=AVAIL
      END IF
:
:*** TINEMAHA TO HAIWEE INFLOW AND OUTFLOW
:
      THIN(I,J)=THRO(I,J)+THSPR(J)+UTHP(I,J)+THAR(I,J)
      IF (UTHP(I,J).GT.THWUSE) THEN
         ATHUSE(I,J)=THWUSE+THAR(I,J)
      ELSE
         ATHUSE(I,J)=UTHP(I,J)+THAR(I,J)
      END IF
      THUL=ATHUSE(I,J)+THWLOSS+THLOS(1,J)
     +               +(THLOS(2,J)*THRO(I,J))
     +               +(THLOS(5,J)*UTHP(I,J))
      THOUT(I,J)=THIN(I,J)-THUL
      IF (THOUT(I,J).LT.0) THEN
         DIFF=THUL-THIN(I,J)
         UTHP(I,J)=UTHP(I,J)+DIFF
         THIN(I,J)=THIN(I,J)+DIFF
         THOUT(I,J)=0
      END IF

  **  ACTUAL USES

      IF (UTHP(I,J).GT.THWUSE) THEN
         ATHUSE(I,J)=THWUSE+THAR(I,J)
      ELSE
         ATHUSE(I,J)=UTHP(I,J)+THAR(I,J)
```

```
      END IF
C
C**** TOTAL USES
C
      TOTUSE(I,J)=ALUSE(I,J)+ABUSE(I,J)+ABPUSE(I,J)+ATHUSE(I,J)
      OLAWSOUT(I,J)=LAWSOUT(I,J)
      OBISOUT(I,J)=BISOUT(I,J)
      OBPOUT(I,J)=BPOUT(I,J)
      OTHOUT(I,J)=THOUT(I,J)
      OTHAR(I,J)=THAR(I,J)
C
      RETURN
      END
```

```
      SUBROUTINE TOOMUCH (I,J)
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
C
C******* NEED TO STORE, SPREAD, AND SPILL
C
C          PRIORITY RANKING
C
C        1. STORE IN TINEMAHA AND HAIWEE
C        2. REDUCE PUMPING TO MINIMUM
C        3. STORE IN LONG VALLEY
C        4. INCREASE EXPORTS
C        5. SPREAD IN LAWS
C        6. SPREAD IN BIG PINE
C        7. SPREAD ON TINEMAHA TO HAIWEE FANS
C        8. AQUEDUCT RELEASES
C
C*** CALCULATE AMOUNT OF WATER THAT IS GOING TO STAY IN VALLEY
C
      DIFF=AVEX(I,J)-HCAP(IHCYT(I),J)
C
C*** STORE IN TINEMAHA AND HAIWEE
C
      IF (DIFF.GT.0) THEN
         TSC=MIN(2000,DIFF)
         TINSTOR(I,J+1)=MIN(TINSTOR(I,J)+TSC,10000)
         TSC=TINSTOR(I,J+1)-TINSTOR(I,J)
         IF (J.EQ.12.AND.I.LT.89) TINSTOR(I+1,1)=TINSTOR(I,J+1)
         DIFF=DIFF-TSC
      END IF
      IF (DIFF.GT.0) THEN
         HSC=MIN(2000,DIFF)
         HAISTOR(I,J+1)=MIN(HAISTOR(I,J)+HSC,10000)
         HSC=HAISTOR(I,J+1)-HAISTOR(I,J)
         IF (J.EQ.12.AND.I.LT.89) HAISTOR(I+1,1)=HAISTOR(I,J+1)
         DIFF=DIFF-HSC
      END IF

 *** REDUCE PUMPING TO MINIMUM, OR REDUCE DIFF TO ZERO

      IF (DIFF.GT.0) THEN
         LDIF=ULAWP(I,J)-(LAWGW(2,J)+(LAWLOS(5,J)*LAWGW(2,J)))
         BDIF=UBISP(I,J)-(BISGW(2,J)+(BISLOS(4,J)*BISGW(2,J)))
         BPDIF=UBPP(I,J)-(BIGGW(2,J)+(BIGLOS(5,J)*BIGGW(2,J)))
         THDIF=UTHP(I,J)-(THGW(2,J)+(THLOS(5,J)*THGW(2,J)))
         MAXPR=LDIF+BDIF+BPDIF+THDIF
         IF (MAXPR.GT.DIFF) THEN
            PRFAC=1-(DIFF/MAXPR)
            ULAWP(I,J)=MAX(ULAWP(I,J)*PRFAC,LAWGW(2,J))
            ULAWP(I,J)=ULAWP(I,J)+(LAWLOS(5,J)*ULAWP(I,J))
            UBISP(I,J)=MAX(UBISP(I,J)*PRFAC,BISGW(2,J))
            UBISP(I,J)=UBISP(I,J)+(BISLOS(4,J)*UBISP(I,J))
            UBPP(I,J)=MAX(UBPP(I,J)*PRFAC,BIGGW(2,J))
            UBPP(I,J)=UBPP(I,J)+(BIGLOS(5,J)*UBPP(I,J))
```

```
            UTHP(I,J)=MAX(UTHP(I,J)*PRFAC,THGW(2,J))
            UTHP(I,J)=UTHP(I,J)+(THLOS(5,J)*UTHP(I,J))
            DIFF=0
         ELSE
            DIFF=DIFF-MAXPR
            ULAWP(I,J)=LAWGW(2,J)+(LAWLOS(5,J)*LAWGW(2,J))
            UBISP(I,J)=BISGW(2,J)+(BISLOS(4,J)*BISGW(2,J))
            UBPP(I,J)=BIGGW(2,J)+(BIGLOS(5,J)*BIGGW(2,J))
            UTHP(I,J)=THGW(2,J)+(THLOS(5,J)*THGW(2,J))
         END IF
C
C*** RECALCULATE INFLOW AND OUTFLOW
C
C*** BISHOP AREA INFLOW AND OUTFLOW
C
         BISIN(I,J)=BISACT(I,J)+BISSPR(J)+BISORD(I,J)+UBISP(I,J)
         BUL=ABUSE(I,J)+BISLOS(1,J)
         BISOUT(I,J)=BISIN(I,J)-BUL
         IF (BISOUT(I,J).LT.0) THEN
            BD=BUL-BISIN(I,J)
            UBISP(I,J)=UBISP(I,J)+BD
            BISOUT(I,J)=0
            DIFF=DIFF+BD
         END IF
C
C*** LAWS AREA INFLOW AND OUTFLOW
C
         LAWSIN(I,J)=LWRO(I,J)+ULAWP(I,J)+LORDIV(I,J)+FSLU(I,J)
         LUL=ALUSE(I,J)+LAWLOS(1,J)
         LAWSOUT(I,J)=LAWSIN(I,J)-LUL
         IF (LAWSOUT(I,J).LT.0) THEN
            LD=LUL-LAWSIN(I,J)
            ULAWP(I,J)=ULAWP(I,J)+LD
            LAWSOUT(I,J)=0
            DIFF=DIFF+LD
        END IF
C
C*** BIG PINE AREA INFLOW AND OUTFLOW
C
         BPIN(I,J)=BPRO(I,J)+UBPP(I,J)+BPORDIV(I,J)+XKEO(I,J)
         BPUL=ABPUSE(I,J)+BIGLOS(1,J)
         BPOUT(I,J)=BPIN(I,J)-BPUL
         IF (BPOUT(I,J).LT.0) THEN
            BPD=BPUL-BPIN(I,J)
            UBPP(I,J)=UBPP(I,J)+BPD
            BPOUT(I,J)=0
            DIFF=DIFF+BPD
         END IF
C
C*** TINEMAHA TO HAIWEE INFLOW AND OUTFLOW
C
         THIN(I,J)=THRO(I,J)+THSPR(J)+UTHP(I,J)+THAR(I,J)
         THUL=ATHUSE(I,J)+THLOS(1,J)
         THOUT(I,J)=THIN(I,J)-THUL
```

```
         IF (THOUT(I,J).LT.0) THEN
            THD=THUL-THIN(I,J)
            UTHP(I,J)=UTHP(I,J)+THD
            THOUT(I,J)=0
            DIFF=DIFF+THD
         END IF
C
      END IF
C
C*** STORE IN LONG VALLEY
C
      IF (DIFF.GT.0) THEN
         LVORED=0
         TLVS=LVSTOR(I,J)-LVEV(I,J)+LVIN(I,J)-LVOUT(I,J)
         TPVM=MAX(0,PVMIN(J)-(RVOUT(I,J)+RVTRAN(1,J)
     +        +(RVTRAN(2,J)*TLVS)))
         IF (LVOUT(I,J).GT.TPVM) THEN
            LVORED=MIN(DIFF,LVOUT(I,J)-TPVM)
            TLVS=TLVS+LVORED
            MAXLVS=MIN(180000,1.2*LVSMAX(ILVYT(I),J))
            IF (TLVS.GT.MAXLVS) THEN
               RR=TLVS-MAXLVS
               LVORED=LVORED-RR
            END IF
         END IF
         DIFF=DIFF-LVORED
         LVOUT(I,J)=LVOUT(I,J)-LVORED
         PVOUT(I,J)=PVOUT(I,J)-LVORED
      END IF
C
C*** INCREASE EXPORTS
C
      XX=0
      IF (DIFF.GT.0) THEN
         XX=MIN(DIFF,(1585*DAYS(J))-HCAP(IHCYT(I),J))
         IF (XX.LT.0) XX=0
         HAEX(I,J)=HAEX(I,J)+XX
         DIFF=DIFF-XX
      END IF
C
C*** SPREAD WATER IN LAWS
C
      IF (DIFF.GT.0) THEN
         AVAIL=MAX(0,PVOUT(I,J)-PVMIN(J)-LORDIV(I,J)-BISORD(I,J))
         ALSPRD(I,J)=MIN(LAWSPD(J),DIFF)
         IF (ALSPRD(I,J).GT.AVAIL) ALSPRD(I,J)=AVAIL
         DIFF=DIFF-ALSPRD(I,J)
         LORDIV(I,J)=LORDIV(I,J)+ALSPRD(I,J)
      END IF
      ALSPRD(I,13)=ALSPRD(I,13)+ALSPRD(I,J)
C
C*** SPREAD WATER IN BIG PINE
C
      IF (DIFF.GT.0) THEN
```

not constrained by LV cap?

```
         AVAIL=MAX(0,PVOUT(I,J)-PVMIN(J)-LORDIV(I,J)-BISORD(I,J)
     +           +LAWSOUT(I,J)+BISOUT(I,J))
         ABPSPRD(I,J)=MIN(BIGSPD(J),DIFF)
         IF (ABPSPRD(I,J).GT.AVAIL) ALSPRD(I,J)=AVAIL
         DIFF=DIFF-ABPSPRD(I,J)
         BPORDIV(I,J)=BPORDIV(I,J)+ABPSPRD(I,J)
      END IF
      ABPSPRD(I,13)=ABPSPRD(I,13)+ABPSPRD(I,J)
C
C*** SPREAD ON FANS BETWEEN TINEMAHA AND HAIWEE
C
      IF (DIFF.GT.0) THEN
         ATHFSP(I,J)=MIN(THSPD(J),DIFF)
         AVAIL=MAX(0,THRO(I,J)-THLOS(1,J)-(THLOS(2,J)*THRO(I,J)))
         IF (ATHFSP(I,J).GT.AVAIL) ATHFSP(I,J)=AVAIL
         THOUT(I,J)=THOUT(I,J)-ATHFSP(I,J)
         DIFF=DIFF-ATHFSP(I,J)
      END IF
      ATHFSP(I,13)=ATHFSP(I,13)+ATHFSP(I,J)
C
C*** SPILL AS A LAST RESORT
C
      IF (DIFF.GT.0) THOS(I,J)=DIFF
      THOS(I,13)=THOS(I,13)+THOS(I,J)
C
C*** CALCULATE EXPORT
C
      HAEX(I,J)=PVOUT(I,J)+BISOUT(I,J)+LAWSOUT(I,J)+BPOUT(I,J)+
     1          THOUT(I,J)-TSC-HSC+XX-THOS(I,J)-THAR(I,J)-LORDIV(I,J)-
     2          BISORD(I,J)-BPORDIV(I,J)-PVTRAN(1,J)-THTRAN(1,J)
C
      RETURN
      END
```

Something missing?
Still exceeding HCAR.

```
      SUBROUTINE NOTENUF (I,J)
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
C
C**********  DECREASE STORAGE, PUMP, REDUCE USES
C
C     PRIORITY RANKING
C
C    1. DRAW DOWN TINEMAHA AND HAIWEE STORAGE
C    2. EXPORT FROM THE MONO BASIN (FISH AND LAKE RELEASE HAVE
C          ALREADY BEEN SATISDIED)
C    3. DO NOT INCREASE STORAGE IN LONG VALLEY (RELEASE ALL INFLOW)
C    4. PUMP GROUNDWATER FROM OV
C    5. DECREASE LV STORAGE TO MINIMUM
C    6. REDUCE OV USES IF THIRD CONS. DRY YEAR
C    7. REDUCE EXPORT
C
C**** CALCULATE AMOUNT OF WATER NEEDED TO MEET EXPORT TARGET
C
      DIFF=HCAP(IHCYT(I),J)-AVEX(I,J)
C
C*** MAKE UP DEFICIENCIES
C
C*** TINEMAHA AND HAIWEE
C
      IF (DIFF.GT.0) THEN
         TSC=MIN(DIFF,2000)
         TINSTOR(I,J+1)=MAX(0,TINSTOR(I,J)-TSC)
         TSC=TINSTOR(I,J)-TINSTOR(I,J+1)
         IF (J.EQ.12.AND.I.LT.89) TINSTOR(I+1,1)=TINSTOR(I,J+1)
         DIFF=DIFF-TSC
      END IF
      IF (DIFF.GT.0) THEN
         HSC=MIN(DIFF,2000)
         HAISTOR(I,J+1)=MAX(0,HAISTOR(I,J)-HSC)
         HSC=HAISTOR(I,J)-HAISTOR(I,J+1)
         IF (J.EQ.12.AND.I.LT.89) HAISTOR(I+1,1)=HAISTOR(I,J+1)
         DIFF=DIFF-HSC
      END IF
C
C*** MONO BASIN EXPORT
C
      IF (DIFF.GT.0) THEN
         WPF(I,J)=MIN(DIFF,ATWP(I,J))
         IF (LVOUT(I,J)+WPF(I,J).GT.LVCAP(J)) THEN
            RWP=LVOUT(I,J)+WPF(I,J)-LVCAP(J)
            WPF(I,J)=WPF(I,J)-RWP
         END IF
         LVIN(I,J)=LVIN(I,J)+WPF(I,J)
         LVOUT(I,J)=LVOUT(I,J)+WPF(I,J)
         PVOUT(I,J)=PVOUT(I,J)+WPF(I,J)
         DIFF=DIFF-WPF(I,J)
      END IF
C
```

```
C******* LONG VALLEY
C
C** RELEASE TO MATCH LAST MONTH'S STORAGE IF POSSIBLE
C
      IF (DIFF.GT.0) THEN
         TLVS=LVSTOR(I,J)-LVEV(I,J)+LVIN(I,J)-LVOUT(I,J)
         IF (TLVS.GT.LVSTOR(I,J).AND.
     +        LVSTOR(I,J).GT.LVSMIN(ILVYT(I),J)) THEN
            ORED=MIN(TLVS-LVSTOR(I,J),LVCAP(J)-LVOUT(I,J))
            IF (ORED.LT.0) ORED=0
            LVNER1=MIN(DIFF,ORED)
            LVOUT(I,J)=LVOUT(I,J)+LVNER1
            PVOUT(I,J)=PVOUT(I,J)+LVNER1
         END IF
      END IF
      DIFF=DIFF-LVNER1
C
C*** PUMP GROUNDWATER, IF NEEDED
C
      IF (DIFF.GT.0) THEN
         AFL=MAX(0,MAXLAW(I,J)-ULAWP(I,J))
         AFL=MAX(0,AFL-(LAWLOS(5,J)*AFL))
         AFB=MAX(0,MAXBIS(I,J)-UBISP(I,J))
         AFB=MAX(0,AFB-(BISLOS(4,J)*AFB))
         AFBP=MAX(0,MAXBP(I,J)-UBPP(I,J))
         AFBP=MAX(0,AFBP-(BIGLOS(5,J)*AFBP)-(PVTRAN(4,J)*AFBP))
         AFTH=MAX(0,MAXTH(I,J)-UTHP(I,J))
         AFTH=MAX(0,AFTH-(THLOS(5,J)*AFTH)-(THTRAN(4,J)*AFTH))
         TOTAV=AFL+AFB+AFBP+AFTH
         IF (DIFF.GT.TOTAV) THEN
            ELAWP(I,J)=AFL
            EBISP(I,J)=AFB
            EBPP(I,J)=AFBP
            ETHP(I,J)=AFTH
         ELSE
            TELAW=0
            TEBIS=0
            TEBP=0
            TETH=0
  900       TELAW=TELAW+10
            IF (AFL.EQ.0) TELAW=0
            TEBIS=TEBIS+10
            IF (AFB.EQ.0) TEBIS=0
            TEBP=TEBP+10
            IF (AFBP.EQ.0) TEBP=0
            TETH=TETH+10
            IF (AFTH.EQ.0) TETH=0
            TE=TELAW+TEBIS+TEBP+TETH
            IF (TE.LE.DIFF) GOTO 900
            ELAWP(I,J)=TELAW
            EBISP(I,J)=TEBIS
            EBPP(I,J)=TEBP
            ETHP(I,J)=TETH
         END IF
```

```
      REDUC(I,13)=REDUC(I,13)+REDUC(I,J)
      TOTUSE(I,J)=ALUSE(I,J)+ABUSE(I,J)+ABPUSE(I,J)+ATHUSE(I,J)
C
C**************REDUCE EXPORT, IF NEEDED
C
      IF (DIFF.GT.0) EXPRED(I,J)=DIFF
      EXPRED(I,13)=EXPRED(I,13)+EXPRED(I,J)
C
C************ HAIWEE EXPORT
C
      HAEX(I,J)=PVOUT(I,J)+TSC+HSC+LAWSOUT(I,J)+BISOUT(I,J)+
     1          BPOUT(I,J)+THOUT(I,J)-PVTRAN(1,J)-THTRAN(1,J)-
     2          THAR(I,J)-LORDIV(I,J)-BISORD(I,J)-BPORDIV(I,J)
C
      RETURN
      END
```

```
      SUBROUTINE GLVSTOR (I,J)
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
C
C***** CALCULATE GRANT STORAGE, SENDING EXTRA WATER TO LV IF MLL
C        IS GREATER THAN LOWEST TARGET
C
      WPF(I,J)=WPF(I,J)+MINWPF(I,J)
      EPF(I,J)=WPF(I,J)+MTMAKE(I,J)
      GSTOR(I,J+1)=GSTOR(I,J)+GIN(I,J)-GEVAP(I,J)-GFFREL(I,J)
     +  -GLREL(I,J)-WPF(I,J)
      MAXLVS=MIN(180000,1.2*LVSMAX(ILVYT(I),J))
      TLVS=LVSTOR(I,J)+LVIN(I,J)-LVOUT(I,J)-LVEV(I,J)
      IF (MLL(I,J).GT.LLS(1)) THEN
         IF (MAXLVS.GT.TLVS.AND.GSTOR(I,J+1).GT.GSTMAX(IGSYT(I),J)) THEN
            DIFF=MIN(MAXLVS-TLVS,GSTOR(I,J+1)-GSTMAX(IGSYT(I),J))
            TGOUT=WPF(I,J)+AMG1FF(I,J)+GLREL(I,J)+RSLR(I,J)
            UORF=EPF(I,J)+ORAEP(I,J)
            DIFF=MIN(DIFF,GOUTMAX(I,J)-TGOUT)
            DIFF=MAX(DIFF,0)
            DIFF=MIN(DIFF,WPFMAX(I,J)-WPF(I,J))
            DIFF=MAX(DIFF,0)
            DIFF=MIN(DIFF,UORMAX(J)-UORF)
            DIFF=MAX(DIFF,0)
            GSTOR(I,J+1)=GSTOR(I,J+1)-DIFF
            LVIN(I,J)=LVIN(I,J)+DIFF
            WPF(I,J)=WPF(I,J)+DIFF
            EPF(I,J)=EPF(I,J)+DIFF
         END IF
      END IF
      IF (GSTOR(I,J+1).GT.50000) THEN
         GSPILL(I,J)=GSTOR(I,J+1)-50000
         GSTOR(I,J+1)=50000
      END IF
      TSPILL(I,J)=TSPILL(I,J)+GSPILL(I,J)
      TSPILL(I,13)=TSPILL(I,13)+GSPILL(I,J)
      IF (J.EQ.12.AND.I.LT.89) GSTOR(I+1,1)=GSTOR(I,13)
C
C*** BALANCE GRANT SPILL
C
      IF (GSPILL(I,J).GT.0) THEN
         X1=MIN(0.5*GSPILL(I,J),LEEMAX(I,J))
         IF (LVTCON(I,J).GT.X1) THEN
            LVTCON(I,J)=LVTCON(I,J)-X1
         ELSE
            X1=LVTCON(I,J)
            LVTCON(I,J)=0
         END IF
         LVSPILL(I,J)=LVSPILL(I,J)+X1
         CONAG(I,J)=CONAG(I,J)-X1
         GIN(I,J)=GIN(I,J)-X1
         GSPILL(I,J)=GSPILL(I,J)-X1
      END IF
C
```

```
C***** CALCULATE LONG VALLEY STORAGE
C
      LVSTOR(I,J+1)=LVSTOR(I,J)+LVIN(I,J)-LVOUT(I,J)-LVEV(I,J)
      IF (LVSTOR(I,J+1).GT.MAXLVS) THEN
         ETHOS=MIN(LVSTOR(I,J+1)-MAXLVS,LVCAP(J)-LVOUT(I,J))
         IF (ETHOS.LT.0) ETHOS=0
         LVOUT(I,J)=LVOUT(I,J)+ETHOS
         PVOUT(I,J)=PVOUT(I,J)+ETHOS
         THOS(I,J)=THOS(I,J)+ETHOS
         LVSTOR(I,J+1)=LVSTOR(I,J)+LVIN(I,J)-LVOUT(I,J)-LVEV(I,J)
      END IF
      IF (LVSTOR(I,J+1).GT.MAXLVS) THEN
         WPR=LVSTOR(I,J+1)-MAXLVS
         WPR=MIN(WPR,WPF(I,J)-MINWPF(I,J))
         WPF(I,J)=WPF(I,J)-WPR
         TREL(I,J)=TREL(I,J)+WPR
         LVIN(I,J)=LVIN(I,J)-WPR
         LVSTOR(I,J+1)=LVSTOR(I,J)+LVIN(I,J)-LVOUT(I,J)-LVEV(I,J)
      END IF
C
      IF (J.EQ.12.AND.I.LT.89) LVSTOR(I+1,1)=LVSTOR(I,13)
C
      RETURN
      END
```

```
      SUBROUTINE GRNPUMP (I,J)
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
C
C****** INITIALIZE MINIMUM PUMPING FOR REST OF YEAR
C
      MINLAW=0
      MINBIS=0
      MINBP=0
      MINTH=0
C
C******* SUM MINIMUM PUMPING FOR REST OF YEAR
C
      IF (J.LT.12) THEN
         DO 900 JJ=J+1,12
         MINLAW=MINLAW+LAWGW(2,JJ)
         MINBIS=MINBIS+BISGW(2,JJ)
         MINBP=MINBP+BIGGW(2,JJ)
         MINTH=MINTH+THGW(2,JJ)
  900    CONTINUE
      ELSE
         MINLAW=0
         MINBIS=0
         MINBP=0
         MINTH=0
      END IF
C
C******** SUM ANNUAL PUMPING THROUGH THIS MONTH
C
      ULAWP(I,13)=ULAWP(I,13)+ULAWP(I,J)
      ELAWP(I,13)=ELAWP(I,13)+ELAWP(I,J)
      LAWPUMP=ULAWP(I,13)+ELAWP(I,13)+MINLAW
      UBISP(I,13)=UBISP(I,13)+UBISP(I,J)
      EBISP(I,13)=EBISP(I,13)+EBISP(I,J)
      BISPUMP=UBISP(I,13)+EBISP(I,13)+MINBIS
      UBPP(I,13)=UBPP(I,13)+UBPP(I,J)
      EBPP(I,13)=EBPP(I,13)+EBPP(I,J)
      BPPUMP=UBPP(I,13)+EBPP(I,13)+MINBP
      UTHP(I,13)=UTHP(I,13)+UTHP(I,J)
      ETHP(I,13)=ETHP(I,13)+ETHP(I,J)
      THPUMP=UTHP(I,13)+ETHP(I,13)+MINTH
C
C****** IF ANNUAL PUMPING + REST-OF-YEAR MINIMUM IS HIGHER THAN
C        "GREEN BOOK" LIMITS, THEN REDUCE MONTHLY MAXIMUM TO ZERO
C        FOR REST OF YEAR
C
      IF (J.LT.12) THEN
         IF (LAWPUMP.GT.MAXLAW(I,13)) THEN
            DO 901 JJ=J+1,12
            MAXLAW(I,JJ)=0
  901       CONTINUE
         END IF
         IF (BISPUMP.GT.MAXBIS(I,13)) THEN
            DO 902 JJ=J+1,12
```

```
            MAXBIS(I,JJ)=0
  902       CONTINUE
         END IF
         IF (BPPUMP.GT.MAXBP(I,13)) THEN
            DO 903 JJ=J+1,12
            MAXBP(I,JJ)=0
  903       CONTINUE
         END IF
         IF (THPUMP.GT.MAXTH(I,13)) THEN
            DO 904 JJ=J+1,12
            MAXTH(I,JJ)=0
  904       CONTINUE
         END IF
      END IF
C
      RETURN
      END
```

```
      SUBROUTINE OUT1 (IYRST,IYREND)
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
C
      OPEN (1,FILE='SUMMARY.OUT')
      OPEN (2,FILE='AVEX.OUT')
      OPEN (3,FILE='MONOTRIB.OUT')
      OPEN (4,FILE='MONOBAL.OUT')
C
      DO 100 I=IYRST,IYREND
      DO 101 J=1,12
C
      IMO=IMORL(J)
      IF (IMO.LT.4) IYR=I+1
      IF (IMO.GE.4) IYR=I
C
      LAWPUMP=ULAWP(I,J)+ELAWP(I,J)
      BISPUMP=UBISP(I,J)+EBISP(I,J)
      BPPUMP=UBPP(I,J)+EBPP(I,J)
      THPUMP=UTHP(I,J)+ETHP(I,J)
      BBPUMP=LAWPUMP+BISPUMP+BPPUMP
      OVPUMP=BBPUMP+THPUMP
      TUSES=TOTUSE(I,J)+RVIRR(J)+LVIRR(J)
      SPREAD=ALSPRD(I,J)+ABPSPRD(I,J)+ATHFSP(I,J)
      LOSS=AVEX(I,J)-(OPVOUT(I,J)+OLAWSOUT(I,J)+OBISOUT(I,J)
     +              +OBPOUT(I,J)+OTHOUT(I,J)-OTHAR(I,J))
      LVNIRR=ODITCH(J)+AGIBDIV(I,J)+AGIBFAR(I,J)
      WCIRR=AWALIRR(I,J)+ASAND3(I,J)
      PCIRR=APARDIV(I,J)+APARIRR(I,J)+ASAND4(I,J)
      TOTIRR=LVNIRR+WCIRR+PCIRR
      TOTGAIN=MBGAIN(1,J)+(MBGAIN(2,J)*MBRO(I,J))
     +                   +(MBGAIN(3,J)*MBIRR(I,J))
C
      WRITE (1,102) IYR,IMO,AVEX(I,J),TUSES,HAEX(I,J),HCAP(IHCYT(I),J),
     + MLL(I,J+1),GSTOR(I,J+1),TREL(I,J),WPF(I,J),LVSTOR(I,J+1),
     + LVOUT(I,J),PVOUT(I,J),TININF(I,J),OVPUMP,SPREAD,THOS(I,J),
     + TINSTOR(I,J+1),HAISTOR(I,J+1)
  102 FORMAT (2I5,4F8.0,F8.2,12F8.0)
C
      WRITE (2,103) IYR,IMO,IDTMNE(I,J),AVEX(I,J),OPVOUT(I,J),
     + OLAWSOUT(I,J),OBISOUT(I,J),OBPOUT(I,J),OTHOUT(I,J),LOSS,
     + OTHAR(I,J)
  103 FORMAT (3I5,8F10.0)
C
      WRITE (3,104) IYR,IMO,LVACT(I,J),ALVFF(I,J),LVLR(I,J),LVNIRR,
     +  LVTCON(I,J),LVSPILL(I,J),WCRO(I,J),AWFF(I,J),WCIRR,
     +  WTCON(I,J),WSPILL(I,J),PCRO(I,J),APFF(I,J),PCIRR,PTCON(I,J),
     +  PSPILL(I,J),RSACT(I,J),CONAG(I,J),TOTGAIN,AMG1FF(I,J),
     +  RSLR(I,J),GLREL(I,J),GSTOR(I,J+1),GEVAP(I,J),WPF(I,J),
     +  GSPILL(I,J)
  104 FORMAT (2I5,26F8.0)
C
      WRITE (4,105) IYR,IMO,MBRO(I,J),TREL(I,J),TFR(I,J),ALR(I,J),
     + TSPILL(I,J),TOTGAIN,TOTIRR,WPF(I,J),MLL(I,J+1),MLAREA(I,J+1),
```

```
     + CPCP(I,J),MLSC(I,J)
  105 FORMAT (2I5,8F10.0,F10.2,F10.0,F10.2,F10.0)
  101 CONTINUE
  100 CONTINUE
      CLOSE (1)
      CLOSE (2)
      CLOSE (3)
      CLOSE (4)
C
C
      OPEN (1,FILE='LONG.OUT')
      OPEN (2,FILE='ROUND.OUT')
      OPEN (3,FILE='LAWS.OUT')
      OPEN (4,FILE='BISHOP.OUT')
      OPEN (5,FILE='POWER.OUT')
C
      DO 200 I=IYRST,IYREND
      DO 201 J=1,12
      IMO=IMORL(J)
      IF (IMO.LT.4) IYR=I+1
      IF (IMO.GE.4) IYR=I
C
      LVG=LVGAIN(1,J)+(LVGAIN(2,J)*LVRO(I,J))+(LVGAIN(3,J)*LVIRR(J))
      RVL=RVLOS(1,J)+(RVLOS(2,J)*RVRO(I,J))
     +              +(RVLOS(3,J)*(RVIRR(J)+RVREC(J)))
      GTG=RVTRAN(1,J)+(RVTRAN(2,J)*LVSTOR(I,J))
      LU=ALUSE(I,J)
      LP=ULAWP(I,J)+ELAWP(I,J)
      LL=LAWLOS(1,J)+(LAWLOS(2,J)*LWRO(I,J))+(LAWLOS(3,J)*LU)
     +   +(LAWLOS(4,J)*LORDIV(I,J))+(LAWLOS(5,J)*LP)
      BU=ABUSE(I,J)
      BP=UBISP(I,J)+EBISP(I,J)
      BL=BISLOS(1,J)+(BISLOS(2,J)*BISACT(I,J))+(BISLOS(3,J)*BU)
     +   +(BISLOS(4,J)*BP)
      HCTOR=(RVOUT(I,J)-BCSPR(I,J))*0.15
      PPVOUT=PVOUT(I,J)-HCTOR
C
      WRITE (1,202) IYR,IMO,WPF(I,J),MTMAKE(I,J),LVRO(I,J),
     + RCD(I,J),HCSPR(I,J),LVG,LVIRR(J),LVOUT(I,J),LVEV(I,J),
     + LVSTOR(I,J+1)
  202 FORMAT (2I5,10F8.0)
C
      WRITE (2,203) IYR,IMO,LVOUT(I,J),RVRO(I,J),BCSPR(I,J),
     + RCD(I,J),RVIRR(J),RVREC(J),RVL,RVOUT(I,J),GTG,
     + HCTOR,PPVOUT,PVOUT(I,J)
  203 FORMAT (2I5,12F10.0)
C
      WRITE (3,204) IYR,IMO,LWRO(I,J),FSLU(I,J),LAWIRR(J),LAWREC(J),
     + LU,LL,ULAWP(I,J),ELAWP(I,J),LP,LORDIV(I,J),ALSPRD(I,J),
     + LAWSIN(I,J),LAWSOUT(I,J)
  204 FORMAT (2I5,13F8.0)
C
      WRITE (4,205) IYR,IMO,BISACT(I,J),BISSPR(J),BISIRR(J),
     + BISREC(J),BU,BL,UBISP(I,J),EBISP(I,J),BP,BISORD(I,J),
```

```
     +  BISIN(I,J),BISOUT(I,J)
  205 FORMAT (2I5,12F10.0)
C
      WRITE (5,206) IYR,IMO,LVOUT(I,J),GTG,PPVOUT,HAEX(I,J),HLALOS(1,J)
  206 FORMAT (2I5,5F10.0)
C
  201 CONTINUE
  200 CONTINUE
      CLOSE (1)
      CLOSE (2)
      CLOSE (3)
      CLOSE (4)
      CLOSE (5)
C
      OPEN (1,FILE='BIGPINE.OUT')
      OPEN (2,FILE='TINEMAHA.OUT')
      OPEN (3,FILE='PUMPING.OUT')
      OPEN (4,FILE='OWENSFSH.OUT')
      OPEN (5,FILE='ALT.SUM')
C
      DO 300 I=IYRST,IYREND
      DO 301 J=1,12
C
      IMO=IMORL(J)
      IF (IMO.LT.4) IYR=I+1
      IF (IMO.GE.4) IYR=I
C
      BPU=ABPUSE(I,J)
      BPP=UBPP(I,J)+EBPP(I,J)
      BPL=BIGLOS(1,J)+(BIGLOS(2,J)*BPRO(I,J))+(BIGLOS(3,J)*BPU)
     +   +(BIGLOS(4,J)*BPORDIV(I,J))+(BIGLOS(5,J)*BPP)
      THU=ATHUSE(I,J)
      THP=UTHP(I,J)+ETHP(I,J)
      THL=THLOS(1,J)+(THLOS(2,J)*THRO(I,J))+(THLOS(3,J)*THU)
     +   +(THLOS(4,J)*ATHFSP(I,J))+(THLOS(5,J)*THP)
      LAWPUMP=ULAWP(I,J)+ELAWP(I,J)
      BISPUMP=UBISP(I,J)+EBISP(I,J)
      BPPUMP=UBPP(I,J)+EBPP(I,J)
      THPUMP=UTHP(I,J)+ETHP(I,J)
      ORAEPC=((ORAEP(I,J)/DAYS(J))*365)/724
      ORBEP=ORAEP(I,J)+MTMAKE(I,J)+WPF(I,J)
      ORBEPC=((ORBEP/DAYS(J))*365)/724
      PVOUTC=((PVOUT(I,J)/DAYS(J))*365)/724
      ORBLD=PVOUT(I,J)-LORDIV(I,J)
      ORBLDC=((ORBLD/DAYS(J))*365)/724
      ORBBD=ORBLD-BISORD(I,J)
      ORBBDC=((ORBBD/DAYS(J))*365)/724
      ORBLR=ORBBD+LAWSOUT(I,J)
      ORBLRC=((ORBLR/DAYS(J))*365)/724
      ORBBR=ORBLR+BISOUT(I,J)-BPORDIV(I,J)
      ORBBRC=((ORBBR/DAYS(J))*365)/724
      TINOUT=((TININF(I,J)/DAYS(J))*365)/724
      PVTTLC=((PVTTL(I,J)/DAYS(J))*365)/724
      HCTOR=(RVOUT(I,J)-BCSPR(I,J))*0.15
```

```
      PPVOUTC=(((PVOUT(I,J)-HCTOR)/DAYS(J))*365)/724
      LAWPUMP=ULAWP(I,J)+ELAWP(I,J)
      BISPUMP=UBISP(I,J)+EBISP(I,J)
      BPPUMP=UBPP(I,J)+EBPP(I,J)
      THPUMP=UTHP(I,J)+ETHP(I,J)
      BBPUMP=LAWPUMP+BISPUMP+BPPUMP
      OVPUMP=BBPUMP+THPUMP
      LVF=ALVFF(I,J)+LVLR(I,J)+LVSPILL(I,J)
      WF=AWFF(I,J)+WSPILL(I,J)
      PF=APFF(I,J)+PSPILL(I,J)
      RF=AMG1FF(I,J)+RSLR(I,J)+GLREL(I,J)+GSPILL(I,J)
C
      WRITE (1,302) IYR,IMO,BPRO(I,J),XKEO(I,J),BIGIRR(J),BIGREC(J),
     +  BPU,BPL,UBPP(I,J),EBPP(I,J),BPP,BPORDIV(I,J),ABPSPRD(I,J),
     +  BPIN(I,J),BPOUT(I,J)
  302 FORMAT (2I5,13F8.0)
C
      WRITE (2,303) IYR,IMO,THRO(I,J),THSPR(J),THWIRR(J),THWREC(J),
     +  THEIRR(J),THEREC(J),THU,
     +  THL,UTHP(I,J),ETHP(I,J),THP,ATHFSP(I,J),THAR(I,J),THOS(I,J),
     +  TINSTOR(I,J+1),HAISTOR(I,J+1),THIN(I,J),THOUT(I,J)
  303 FORMAT (2I5,18F8.0)
C
      WRITE (3,304) IYR,IMO,
     +  LAWGW(2,J),MAXLAW(I,J),ULAWP(I,J),ELAWP(I,J),LAWPUMP,
     +  BISGW(2,J),MAXBIS(I,J),UBISP(I,J),EBISP(I,J),BISPUMP,
     +  BIGGW(2,J),MAXBP(I,J),UBPP(I,J),EBPP(I,J),BPPUMP,
     +  THGW(2,J),MAXTH(I,J),UTHP(I,J),ETHP(I,J),THPUMP
  304 FORMAT (2I5,20F8.0)
C
      WRITE (4,305) IYR,IMO,ORAEPC,ORBEPC,PPVOUTC,PVOUTC,ORBBDC,
     +  ORBLDC,ORBLRC,ORBBRC,PVTTLC,TINOUT
  305 FORMAT (2I5,10F10.0)
C
      WRITE (5,306) IYR,IMO,LVF,WF,PF,RF,TFR(I,J),MLL(I,J+1),
     1    WPF(I,J),OVPUMP,HAEX(I,J)
  306 FORMAT (2I5,5F10.0,F10.2,3F10.0)
C
  301 CONTINUE
  300 CONTINUE
      CLOSE (1)
      CLOSE (2)
      CLOSE (3)
      CLOSE (4)
      CLOSE (5)
C
      RETURN
      END
```

```
C
C*********** COMMON BLOCKS FOR PROGRAM LAMP.FOR***********
C
      COMMON /MBHYD/ WCRO(40:89,13),PCRO(40:89,13),
     + MBRO(40:89,13),CPCP(40:89,13)
C
      COMMON /LVHYD/ ORAEP(40:89,13),LVRO(40:89,13),
     + HCHWY(40:89,13),RCLRV(40:89,13),MTMAKE(40:89,13),
     + HCSPR(40:89,13),LVPCP(40:89,13)
C
       COMMON /OVHYD/ RVRO(40:89,13),LWRO(40:89,13),
     + BSRO(40:89,13),BCRO(40:89,13),BPRO(40:89,13),
     + LBBPRO(40:89,13),THRO(40:89,13),OVRO(40:89,13),
     + MBOVRO(40:89,13),BCSPR(40:89,13),FSLU(40:89,13),
     + XKEO(40:89,13)
C
      COMMON /MBVA/ AGIBDIV(40:89,13),AGIBFAR(40:89,13),
     + ALR(40:89,13),ALVFF(40:89,13),AMG1FF(40:89,13),
     + APARDIV(40:89,13),APARIRR(40:89,13),APFF(40:89,13),
     + ASAND3(40:89,13),ASAND4(40:89,13),AWALIRR(40:89,13),
     + AWFF(40:89,13)
C
      COMMON /MBVGI/ CONAG(40:89,13),EPCAP(13),
     + GEVAP(40:89,13),GIBDIV(13),GIBFAR(13),GIN(40:89,13),
     + GRAEV(13),GRCAP(13),GSPILL(40:89,13),GSTMAX(9,13),GSTMIN(9,13),
     + GSTOR(40:89,13),GOUTMAX(40:89,13),GCAP(13)

      COMMON /MBVLM/ LLRF(3,3),LLS(3),LRT(40:89),OLRT(40:89),
     + LVACT(40:89,13),BLVFF(6,13),FLVFF(6,13),
     + LVLR(40:89,13),LVSPILL(40:89,13),LVSTOR(40:89,13),
     + LVTCON(40:89,13),MBGAIN(10,13),MBIRR(40:89,13),MLL(40:89,13),
     + LEEMAX(40:89,13),MG1MAX(40:89,13),MLAREA(40:89,13),
     + MLVOL(40:89,13),MLSC(40:89,13),MCAP(13),LCAP(13),MLEVAP(13)
C
      COMMON /MBVOR/ ODITCH(13),PARDIV(13),PARIRR(13),
     + BPFF(6,13),FPFF(6,13),PSPILL(40:89,13),PTCON(40:89,13),
     + RSACT(40:89,13),BRSFF(6,13),FRSFF(6,13),RSLR(40:89,13),
     + RUSHTG(40:89,13)
C
       COMMON /MBVTY/ SAND3(13),SAND4(13),TFR(40:89,13),TREL(40:89,13),
     + WALIRR(13),BWFF(6,13),FWFF(6,13),WPF(40:89,13),WSPILL(40:89,13),
     + WTCON(40:89,13),WPFMAX(40:89,13),WCAP(13)
C
      COMMON /OVVA/ ABPSPRD(40:89,13),ALSPRD(40:89,13),
     + ATHFSP(40:89,13)
C
      COMMON /OVVB/ BIGGW(2,13),BIGIRR(13),
     + BIGLOS(10,13),BIGREC(13),BIGSPD(13),BISACT(40:89,13),
     + BISCAP(13),BISGW(2,13),BISIN(40:89,13),BISIRR(13),BISLOS(10,13),
     + BISORD(40:89,13),BISOUT(40:89,13),OBISOUT(40:89,13),
     + BISREC(13),BISSPR(13),BPIN(40:89,13),
     + BPORDIV(40:89,13),BPOUT(40:89,13),OBPOUT(40:89,13)
C
      COMMON /OVVEH/ EPF(40:89,13),EXPRED(40:89,13),FLOWLA(40:89,13),
```

```
     + HAEV(13),HAEX(40:89,13),HCAP(9,13),HLALOS(10,13),
     + HAISTOR(40:89,13)
C
      COMMON /OVVL/ LAWGW(2,13),LAWIRR(13),LAWLOS(10,13),
     + LAWREC(13),LAWSIN(40:89,13),OLAWSOUT(40:89,13),
     + LAWSOUT(40:89,13),LAWSPD(13),LONGEV(13),
     + LORDIV(40:89,13),LVEV(40:89,13),
     + LVGAIN(10,13),LVIN(40:89,13),LVIRR(13),LVCAP(13),
     + LVOUT(40:89,13),LVSMAX(9,13),LVSMIN(9,13)
C
      COMMON /OVVMP/ MAXBIS(40:89,13),MAXBP(40:89,13),
     + MAXLAW(40:89,13),MAXTH(40:89,13),OPVOUT(40:89,13),PVMIN(13),
     + PVOUT(40:89,13),PVTRAN(10,13),PVMAX(13),PVTTL(40:89,13),
     + PCAP(40:89)
C
      COMMON /OVVR/ RCD(40:89,13),RCKFSH(13),
     + RCTRV(40:89,13),REDFAC(40:89),REDUC(40:89,13),
     + RVLOS(10,13),RVIN(40:89,13),RVIRR(13),RVOUT(40:89,13),
     + RVREC(13),RVTRAN(10,13)
C
      COMMON /OVVTW/ THGW(2,13),THIN(40:89,13),THWIRR(13),
     + THEIRR(13),THLOS(10,13),THTRAN(10,13),THOS(40:89,13),
     + THOUT(40:89,13),THWREC(13),THEREC(13),OTHOUT(40:89,13),
     + THSPIL(13),THSPR(13),THSPD(13),TINEV(13),TINSTOR(40:89,13),
     + TININF(40:89,13),TOTUSE(40:89,13),UORMIN(13),UORMAX(13),
     + LUSEP(3,13),BUSEP(3,13),BPUSEP(3,13),USEDRY(5)
C
      COMMON /TIME/ DAYS(12),IMORL(12)
C
      COMMON /BATHYM/ GLBATHY(73,3),LVBATHY(97,3),MNBATHY(170,3)
C
      COMMON /FISH/ LVCFS(12),WCFS(12),PCFS(12),RSCFS(12),MG1TR(10),
     +  PTR(10),WTR(10),LVTR(10)
C
      COMMON /YRINDEX/ IFFYT(40:89,4),IGSYT(40:89),ILVYT(40:89),
     +  IHCYT(40:89),IMLYT(40:89),IDLL(40:89)
C
      COMMON /HC/ LPMP(3,12),BPMP(3,12),BPPMP(3,12),THPMP(3,12)
C
C      COMMON /OVDTWCB/ CMAXP(10),TWSEL(25,10),MINWSE(10),TP(25),
C     +  WSEL(40:89,10)
C
      COMMON /PUMP1/ ULAWP(40:89,13),UBISP(40:89,13),UBPP(40:89,13),
     +  UTHP(40:89,13),ELAWP(40:89,13),EBISP(40:89,13),EBPP(40:89,13),
     +  ETHP(40:89,13)
C
      COMMON /OUTPUT/ IDTMNE(40:89,13),GFFREL(40:89,13),
     +  GLREL(40:89,13),AVEX(40:89,13),MINWPP(40:89,13),
     +  ATWP(40:89,13),TSPILL(40:89,13)
C
      COMMON /USES/ ABUSE(40:89,13),ALUSE(40:89,13),ABPUSE(40:89,13),
     +  ATHUSE(40:89,13),THAR(40:89,13),OTHAR(40:89,13)
```

```
C**** DATA STATEMENTS
C
      DATA (DAYS(J),J=1,12)/30,31,30,31,31,30,31,30,31,31,28.25,31/
      DATA (IMORL(J),J=1,12)/4,5,6,7,8,9,10,11,12,1,2,3/
C
C      DATA (YINT(J),J=1,12)/ -4.227610,13.653998,7.005902,3.737981,
C     +    18.563331,10.074578,10.407162,-0.325178,3.406666,2.788617,
C     +    14.860419,6.123419 /
C
C      DATA (CMLS(J),J=1,12)/ 1.000640, 0.997832, 0.998855, 0.999359,
C     +    0.997024, 0.998362, 0.998330, 1.000022, 0.999455, 0.999558,
C     +    0.997677, 0.999037 /
C
C      DATA (CCPCP(J),J=1,12)/ 0.060561,0.054893,0.089478,0.059092,
C     +    0.106352,0.076871,0.023117,0.032547,0.039225,0.060955,
C     +    0.051866,0.050414 /
C
C      DATA (CTREL(J),J=1,12)/ 0.0000256, 0.0000215, 0.0000224,
C     +    0.0000227, 0.0000225, 0.0000235, 0.0000298, 0.0000340,
C     +    0.0000315, 0.0000207, 0.0000256, 0.0000235 /
```

```
C
C**** INITIALIZE EVERYTHING
C
      DO 690 I=40,89
      DO 691 J=1,13
      AGIBDIV(I,J)=0
      AGIBFAR(I,J)=0
      ALR(I,J)=0
      ALVFF(I,J)=0
      AMG1FF(I,J)=0
      APARDIV(I,J)=0
      APARIRR(I,J)=0
      APFF(I,J)=0
      ASAND3(I,J)=0
      ASAND4(I,J)=0
      AWALIRR(I,J)=0
      AWFF(I,J)=0
      CONAG(I,J)=0
      GEVAP(I,J)=0
      GIN(I,J)=0
      GSPILL(I,J)=0
      GSTOR(I,J)=0
      LRT(I)=0
      LVACT(I,J)=0
      LVLR(I,J)=0
      LVSPILL(I,J)=0
      LVSTOR(I,J)=0
      LVTCON(I,J)=0
      MLL(I,J)=0
      PSPILL(I,J)=0
      PTCON(I,J)=0
      RSACT(I,J)=0
      RSLR(I,J)=0
      TREL(I,J)=0
      WSPILL(I,J)=0
      WTCON(I,J)=0
      ABPSPRD(I,J)=0
      ALSPRD(I,J)=0
      ATHFSP(I,J)=0
      BISACT(I,J)=0
      BISIN(I,J)=0
      BISORD(I,J)=0
      BISOUT(I,J)=0
      BPIN(I,J)=0
      BPORDIV(I,J)=0
      BPOUT(I,J)=0
      EPF(I,J)=0
      EXPRED(I,J)=0
      FLOWLA(I,J)=0
      HAEX(I,J)=0
      HCSPR(I,J)=0
      LAWSIN(I,J)=0
      LAWSOUT(I,J)=0
      LORDIV(I,J)=0
```

```
      LVEV(I,J)=0
      LVIN(I,J)=0
      LVOUT(I,J)=0
      MAXBIS(I,J)=0
      MAXBP(I,J)=0
      MAXLAW(I,J)=0
      MAXTH(I,J)=0
      PVOUT(I,J)=0
      RCD(I,J)=0
      RCTRV(I,J)=0
      REDFAC(I)=0
      REDUC(I,J)=0
      RVIN(I,J)=0
      RVOUT(I,J)=0
      THIN(I,J)=0
      THOS(I,J)=0
      THOUT(I,J)=0
      TININF(I,J)=0
      WPF(I,J)=0
      IGSYT(I)=0
      ILVYT(I)=0
      IHCYT(I)=0
      IMLYT(I)=0
      ULAWP(I,J)=0
      UBISP(I,J)=0
      UBPP(I,J)=0
      UTHP(I,J)=0
      ELAWP(I,J)=0
      EBISP(I,J)=0
      EBPP(I,J)=0
      ETHP(I,J)=0
      TFR(I,J)=0
      TSPILL(I,J)=0
      IDTMNE(I,J)=0
      GFFREL(I,J)=0
      GLREL(I,J)=0
      AVEX(I,J)=0
      MINWPF(I,J)=0
      ATWP(I,J)=0
      RUSHTG(I,J)=0
      OLAWSOUT(I,J)=0
      OBISOUT(I,J)=0
      OBPOUT(I,J)=0
      OTHOUT(I,J)=0
      OTHAR(I,J)=0
  691 CONTINUE
  690 CONTINUE
```

APPENDIX B

Summary of Output Files

Appendix B
Summary of Output Files
All Values in Acre-Feet Unless Noted otherwise

SUMMARY.OUT

IYR	Year
IMO	Month
AVEX	Available to Export (before TOOMUCH or NOTENUF)
TUSES	Total Uses
HAEX	Calculated Haiwee Export
HCAP	Haiwee Export Target
MLL	End-of-Month Mono Lake Elevation (ft MSL)
GSTOR	End-of-Month Grant Storage
TREL	Total Releases to Mono Lake
WPF	West Portal Flow (Mono Basin Export)
LVSTOR	End-of-Month Long Valley Storage
LVOUT	Long Valley Outflow
PVOUT	Pleasant Valley Outflow
TININF	Tinemaha Inflow
OVPUMP	Total Owens Valley Pumping
SPREAD	Total Owens Valley Spreading
THOS	Tinemaha to Haiwee Operational Spreading
TINSTOR	End-of-Month Tinemaha Storage
HAISTOR	End-of Month Haiwee Storage

AVEX.OUT

IYR	Year
IMO	Month
IDTMNE	TOOMUCH/NOTENUF Index
AVEX	Available to Export (before TOOMUCH or NOTENUF)
OPVOUT	Pleasant Valley Outflow (before TOOMUCH or NOTENUF)
OLAWSOUT	Laws Area Outflow (before TOOMUCH or NOTENUF)
OBISOUT	Bishop Area Outflow (before TOOMUCH or NOTENUF)
OBPOUT	Big Pine Area Outflow (before TOOMUCH or NOTENUF)
OTHOUT	Tinemaha to Haiwee Outflow (before TOOMUCH or NOTENUF)
LOSS	Pleasant Valley to Haiwee Losses (before TOOMUCH or NOTENUF)
OTHAR	Tinemaha to Haiwee Aqueduct Release to meet uses

MONOTRIB.OUT

IYR	Year
IMO	Month
LVACT	Lee Vining Creek Impaired Runoff
ALVFF	Lee Vining Creek Minimum Release
LVLR	Lee Vining Creek Lake Release
LVNIRR	Lee Vining Creek Irrigation Release
LVTCON	Lee Vining Creek Contribution to Conduit
LVSPILL	Lee Vining Creek Spill
WCRO	Walker Creek Runoff
AWFF	Walker Creek Minimum Release
WCIRR	Walker Creek Irrigation Release
WTCON	Walker Creek Contribution to Conduit
WSPILL	Walker Creek Spill
PCRO	Parker Creek Runoff
APFF	Parker Creek Minimum Release
PCIRR	Parker Creek Irrigation Release
PTCON	Parker Creek Contribution to Conduit
PSPILL	Parker Creek Spill
RSACT	Rush Creek Impaired Runoff
CONAG	Conduit Flow above Grant
TOTGAIN	Mono Basin Gains
AMG1FF	Mono Gate 1 Minimum Release
RSLR	Rush Creek Lake Release
GLREL	Grant Lake Release to Mono Lake
GSTOR	End-of-Month Grant Storage
GEVAP	Grant Lake Evaporation
WPF	West Portal Flow (Mono Basin Export)
GSPILL	Grant Lake Spill

MONOBAL.FOR

IYR	Year
IMO	Month
MBRO	Mono Basin Runoff (four creeks)
TREL	Total Releases to Mono Lake
TFR	Total Minimum Release Flows
ALR	Total LAke Release Flows
TSPILL	Total Spills
TOTGAIN	Mono Basin Gains
TOTIRR	Total Irrigation Releases
WPF	West Portal Flow (Mono Basin Export)
MLL	End-of-Month Mono Lake Elevation (ft MSL)
MLAREA	End-of-Month Mono Lake Surface Area (acres)
CPCP	Cain Ranch Precipitation (inches)
MLSC	Mono Lake Volume Change

LONG.OUT

IYR	Year
IMO	Month
WPF	West Portal Flow (Mono Basin Export)
MTMAKE	Mono Tunnel Make
LVRO	Long Valley Runoff
RCD	Rock Creek Diversion
HCSPR	Hot Creek Springs
LVG	Long Valley Gains
LVIRR	Long Valley Irrigation
LVOUT	Long Valley Outflow
LVEV	Long Valley Reservoir Net Evaporation
LVSTOR	End-of-Month Long Valley Storage

ROUND.OUT

IYR	Year
IMO	Month
LVOUT	Long Valley Outflow
RVRO	Round Valley Runoff
BCSPR	Birchim Canyon Springs
RCD	Rock Creek Diversion
RVIRR	Round Valley Irrigation
RVREC	Round Valley Recreation Uses
RVL	Round Valley Losses
RVOUT	Round Valley Area Outflow
GTG	Long Valley to Pleasant Valley Transit Gain
HCTOR	Assumed Horton Creek Flow to Owens River
PPVOUT	True Pleasant Valley Outflow
PVOUT	Owens River Below Horton Creek

LAWS.OUT

IYR	Year
IMO	Month
LWRO	Laws Area Runoff
FSLU	Fish Slough Flow
LAWIRR	Laws Area Irrigation Target
LAWREC	Laws Area Recreation Uses Target
LU	Total Laws Area Uses
LL	Total Laws Area Losses
ULAWP	Laws Area Uses Pumping
ELAWP	Laws Area Export Pumping
LP	Total Laws Pumping
LORDIV	Owens River Diversion to Laws Area
ALSPRD	Laws Area Spreading
LAWSIN	Laws Area Inflow
LAWSOUT	Laws Area Outflow

BISHOP.OUT

IYR	Year
IMO	Month
BISACT	Bishop Area Impaired Runoff
BISSPR	Bishop Area Flowing Groundwater
BISIRR	Bishop Area Irrigation Target
BISREC	Bishop Area Recreation Use Target
BU	Total Bishop Area Uses
BL	Total Bishop Area Losses
UBISP	Bishop Area Use Pumping
EBISP	Bishop Area Export Pumping
BP	Total Bishop Area Pumping
BISORD	Owens River Diversion to Bishop Area
BISIN	Bishop Area Inflow
BISOUT	Bishop Area Outflow

POWER.OUT

IYR	Year
IMO	Month
LVOUT	Long Valley Outflow
GTG	Long Valley to Pleasant Valley Gain
PPVOUT	Pleasant Valley Outflow
HAEX	Haiwee Outflow
HLALOS	Haiwee to Los Angeles Losses

BIGPINE.OUT

IYR	Year
IMO	Month
BPRO	Big Pine Area Runoff
XKEO	Keough Hot Springs Flow
BIGIRR	Big Pine Area Irrigation Target
BIGREC	Big Pine Area Recreation Uses Target
BPU	Total Big Pine Area Uses
BPL	Total Big Pine Area Losses
UBPP	Big Pine Area Uses Pumping
EBPP	Big Pine Area Export Pumping
BPP	Total Big Pine Area Pumping
BPORDIV	Owens River Diversion to Big Pine Area
ABPSPRD	Big Pine Area Spreading
BPIN	Big Pine Area Inflow
BPOUT	Big Pine Area Outflow

TINEMAHA.OUT

IYR	Year
IMO	Month
THRO	Tinemaha to Haiwee Runoff
THSPR	Tinemaha to Haiwee Flowing Groundwater
THWIRR	Tinemaha to Haiwee Irrigation Target (West)
THWREC	Tinemaha to Haiwee Recreation Use Target (West)
THEIRR	Tinemaha to Haiwee Irrigation Target (East)
THEREC	Tinemaha to Haiwee Recreation Use Target (East)
THU	Total Tinemaha to Haiwee Uses
THL	Total Tinemaha to Haiwee Losses
UTHP	Tinemaha to Haiwee Uses Pumping
ETHP	Tinemaha to Haiwee Export Pumping
THP	Total Tinemaha to Haiwee Pumping
ATHFSP	Tinemaha to Haiwee Fan Spreading
THAR	Tinemaha to Haiwee Aqueduct Release to Meet East Uses
THOS	Tinemaha to Haiwee Operational Release
TINSTOR	End-of-Month Tinemaha Storage
HAISTOR	End-of-Month Haiwee Storage
THIN	Tinemaha to Haiwee Area Inflow
THOUT	Tinemaha to Haiwee Area Outflow

PUMPING.OUT

IYR	Year
IMO	Month
LAWGW	Laws Area Minimum Pumping
MAXLAW	Laws Area Maximum Pumping
ULAWP	Laws Area Uses Pumping
ELAWP	Laws Area Export Pumping
LAWPUMP	Total Laws Area Pumping
BISGW	Bishop Area Minimum Pumping
MAXBIS	Bishop Area Maximum Pumping
UBISP	Bishop Area Uses Pumping
EBISP	Bishop Area Export Pumping
BISPUMP	Total Bishop Area Pumping
BIGGW	Big Pine Area Minimum Pumping
MAXBP	Big Pine Area Maximum Pumping
UBPP	Big Pine Area Uses Pumping
EBPP	Big Pine Area Export Pumping
BPPUMP	Total Big Pine Area Pumping
THGW	Tinemaha to Haiwee Area Minimum Pumping
MAXTH	Tinemaha to Haiwee Area Maximum Pumping
UTHP	Tinemaha to Haiwee Area Uses Pumping
ETHP	Tinemaha to Haiwee Area Export Pumping
THPUMP	Total Tinemaha to Haiwee Area Pumping

OWENSFSH.OUT (note all values in cfs)

IYR	Year
IMO	Month
ORAEPC	Owens River Above East Portal
ORBEPC	Owens River Below East Portal
PPVOUTC	Pleasant Valley Outflow
PVOUTC	Owens River Below Horton Creek
ORBBDC	Owens River Below Bishop Diversion
ORBLDC	Owens River Below Laws Diversion
ORBLRC	Owens River Below Laws Return
ORBBRC	Owens River Below Bishop Return/Big Pine Diversion
PVTTLC	Pleasant Valley to Tinemaha Transit Loss
TINOUT	Tinemaha Outflow

ALT.SUM

IYR	Year
IMO	Month
LVF	Total Lee Vining Creek Flow Below Conduit
WF	Total Walker Creek Flow Below Conduit
PF	Total Parker Creek Flow Below Conduit
RF	Total Rush Creek Flow Below Mono Gate 1
TFR	Total Minimum flow Releases
MLL	End-of-Month Mono Lake Elevation (ft MSL)
WPF	West Portal Flow (Mono Basin Export)
OVPUMP	Total Owens Valley Pumping
HAEX	Haiwee Export

APPENDIX C

ALAAMP Source Listing

```
      PROGRAM ALAAMP
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
      INCLUDE 'LAAMP.INT'
      INCLUDE 'LAAMP.DST'
C
C  PROGRAM TO SIMULATE THE OPERATION OF THE LOS ANGELES AQUEDUCT
C  MODIFIED AND ENHANCED FROM A LADWP SIMULATION MODEL
C
C  ALAAMP READS PUMPING FILE GENERATED BY BASE CASE OF LAAMP
C    DEVELOPED DECEMBER 30, 1991
C
C     WILLIAM R. HUTCHISON, LUHDORFF & SCALMANINI CONSULTING ENGINEERS
C     RUSSEL T. BROWN, JONES & STOKES ASSOCIATES
C     PAUL L. WISHEROPP, JONES & STOKES ASSOCIATES
C
C      JULY 15, 1991
C         MODIFIED AUGUST 8, 1991
C         MODIFIED AUGUST 12, 1991
C         MODIFIED SEPTEMBER 17, 1991
C         MODIFIED SEPTEMBER 30, 1991
C         MODIFIED OCTOBER 14, 1991
C         MODIFIED DECEMBER 20, 1991
C
C********* OPEN FILES AND READ BATHYMETRY DATA, INPUT HYDROLOGY AND
C             "SPREADSHEET" DATA
C
      WRITE (*,10)
   10 FORMAT (' READING FILES AND PREPARING DATA')
C
C******** READ DATA
C
      CALL READBH (X)
      CALL READPRN (IYRST,IYREND,ILVCYC,IWCYC,IPCYC,IRSCYC,FFMBRO,
     +  FFIDW,FFIDD,MLMBRO,MLIDW,MLIDD,AVOMRO,WET,DRY,IDOVP,IDRFF)
C
C******** HYDROLOGIC CALCULATIONS
C
      CALL HYDCALC (IYRST,IYREND,ILVCYC,IWCYC,IPCYC,IRSCYC,FFMBRO,
     +  FFIDW,FFIDD,MLMBRO,MLIDW,MLIDD,AVOMRO,WET,DRY,IDOVP)
C
C******** READ PUMPING FILE
C
      CALL READPUMP (IYRST,IYREND)
C
C******** INITITALIZE MONO LAKE EQUATION
C
      CALL BATHY (1,MLL(IYRST,1),MLVOL(IYRST,1),MLAREA(IYRST,1))
      RI=-0.206935+(0.00905776*MLEVAP(13))
      YINT=-8651.95+(238.897*MLEVAP(13))
C
C********* START SIMULATION
C
      WRITE (*,11) IYRST+1900,IYREND+1901
```

```
   11 FORMAT (' SIMULATION FROM APRIL',I5,1X,'TO MARCH',I5/)
C
C*** ANNUAL LOOP
C
      DO 100 I=IYRST,IYREND

      WRITE (*,102) I,I+1
  102 FORMAT (1H+,' SIMULATING RUNOFF YEAR ',I2,'-',I2)
C
C******* LAKE RELEASE TARGET
C
      CALL MLTAR (I)
C
C*** MONTHLY LOOP
C
      DO 101 J=1,12
C
C******* CALCULATE RESERVOIR EVAPORATION AND RAINFALL
C
C*** GRANT
C
      CALL BATHY (2,GSTOR(I,J),GELEV,GSURF)
      GRAIN=(CPCP(I,J)/12)*GSURF
      GEVAP(I,J)=(GRAEV(J)*(GSURF/1000))-GRAIN
C
C*** LONG VALLEY
C
      CALL BATHY (3,LVSTOR(I,J),LVELEV,LVSURF)
      LVRAIN=(LVPCP(I,J)/12)*LVSURF
      LVEV(I,J)=(LONGEV(J)*(LVSURF/1000))-LVRAIN
C
C*************** CALCULATE AVAILABLE FROM EACH AREA
C
C
C********* MONO BASIN
C
C************** SET AVAILABLE TO RUNOFF/ACTUAL AND CONDUIT
C                MAX FLOWS AND INITIALIZE FISH DEFICITS
C
      LVAVAIL=LVACT(I,J)-ODITCH(J)
      IF (LVAVAIL.GT.14000) LVSPILL(I,J)=-7000+(0.5*LVAVAIL)
      IF (LVAVAIL.GT.22000) LVSPILL(I,J)=LVAVAIL-18000
      LVAVAIL=LVAVAIL-LVSPILL(I,J)
      IF (LRT(I).NE.-1) LRT(I)=LRT(I)-LVSPILL(I,J)
      LEEMAX(I,J)=LEEMAX(I,J)-LVSPILL(I,J)
      IF (LEEMAX(I,J).LT.0) LEEMAX(I,J)=0
C
      WAVAIL=WCRO(I,J)
      PAVAIL=PCRO(I,J)
      RSAVAIL=RSACT(I,J)+MBGAIN(1,J)
      LVCMAX=300*1.9835*DAYS(J)
      WMAX=325*1.9835*DAYS(J)
      CAGMAX=350*1.9835*DAYS(J)
      LVFD=0
```

```
      WFD=0
      PFD=0
      MG1FD=0
C
C
C************** ESTABLISH FISH FLOWS AFTER CREDITING SPILLS AND
C               CHECKING IF SPILLS AND LAKE RELEASES OF PAST MONTHS
C               MEET FLUSHING REQUIREMENTS
C
      CALL MBFLUSH (I,J,WLVFF,WWFF,WPFF,WRSFF,ILVCYC,IWCYC,IPCYC,
     +      IRSCYC,IYRST)
C
C
C****************  CALCULATE ACTUAL FISH RELEASES
C
C
      CALL MBFISH (I,J,WLVFF,WWFF,WPFF,WRSFF,LVFD,WFD,PFD,MG1FD,
     +             LVAVAIL,WAVAIL,PAVAIL,RSAVAIL,IDRFF)
C
C
C****** LAKE RELEASES
C
      IF (LRT(I).GT.-1) THEN
         CALL MLR (I,J,LVFD,WFD,PFD,MG1FD,LVAVAIL,RSAVAIL)
      END IF
C
C************************ IRRIGATION
C
      CALL MBIRRIG (I,J,LVAVAIL,WAVAIL,PAVAIL,LVCMAX,WMAX,CAGMAX)
C
C********* SPILL EXTRA AND SEND AVAILABLE TO CONDUIT
C
      CALL CONDUIT (I,J,LVAVAIL,WAVAIL,PAVAIL,RSAVAIL,
     +   LVCMAX,WMAX,CAGMAX)
C
C******* CALCULATE MINIMUM WEST PORTAL FLOW
C
      CALL WP (I,J)
C
C
C********* OWENS RIVER BASIN
C
      CALL OWENS (I,J)
C
C****************  CHECK AVEX (AVAILABLE TO EXPORT)
C
C   "AVEX" IS WATER THAT CAN BE EXPORTED AFTER ALL USES ARE
C   SATISFIED (NOT AMOUNT OF USE IS SPECIFIED AND COMPARED TO
C   RUNOFF AND USER SPECIFIED MINIMUM PUMPING. THIS VALUE
C   IS THEN COMPARED TO TARGET HAIWEE EXPORT "CAPACITY" (HCAP).
C
C           IF OVAVAIL>HCAP INC STORAGE, SPREAD AND SPILL
C           IF OVAVAIL<HCAP DEC STORAGE, PUMP GROUNDWATER,
C              EXPORT FROM MONO BASIN, AND REDUCE USES
```

```
C
C
C*******  COMPARE
C
      AVAIL=PVOUT(I,J)+LAWSOUT(I,J)+BISOUT(I,J)+BPOUT(I,J)+THOUT(I,J)
     +                -THAR(I,J)-LORDIV(I,J)-BISORD(I,J)-BPORDIV(I,J)
      OPVOUT(I,J)=PVOUT(I,J)
      BBTL=PVTRAN(1,J)+(PVTRAN(2,J)*LBBPRO(I,J))
     +                +(PVTRAN(3,J)*PVOUT(I,J))
     +                +(PVTRAN(4,J)*UBPP(I,J))
      TINOUT=PVOUT(I,J)+LAWSOUT(I,J)+BISOUT(I,J)+BPOUT(I,J)
     +                 -BBTL-TINEV(J)
      OLBTL=THTRAN(1,J)+(THTRAN(2,J)*THRO(I,J))
     +                 +(THTRAN(3,J)*TINOUT)
     +                 +(THTRAN(4,J)*UTHP(I,J))
      AVEX(I,J)=MAX(0,AVAIL-BBTL-OLBTL)
      IF (AVEX(I,J).GE.HCAP(IHCYT(I),J)) THEN
         IDTMNE(I,J)=1
         CALL TOOMUCH (I,J)
      ELSE
         IDTMNE(I,J)=2
         CALL NOTENUF (I,J)
      END IF
C
C***** CALCULATE GRANT AND LONG VALLEY STORAGES
C
      CALL GLVSTOR (I,J)
C
C***** CALCULATE TINEMAHA INFLOW
C
      PVTTL(I,J)=PVTRAN(1,J)+(PVTRAN(2,J)*LBBPRO(I,J))
     +                      +(PVTRAN(3,J)*PVOUT(I,J))
     +                      +(PVTRAN(4,J)*(UBPP(I,J)+EBPP(I,J)))
      TININF(I,J)=PVOUT(I,J)+LAWSOUT(I,J)+BISOUT(I,J)+BPOUT(I,J)
     +  -PVTTL(I,J)
C
C********* CALCULATE FLOW TO CITY
C
      FLOWLA(I,J)=HAEX(I,J)-HLALOS(1,J)+(HLALOS(2,J)*HAEX(I,J))
C
C******* CALCULATE MONO LAKE LEVEL, AREA AND STORAGE CHANGE
C
      TREL(I,J)=ALR(I,J)+TFR(I,J)+TSPILL(I,J)
      XSC=TREL(I,J)+((CPCP(I,J)/12)*MLAREA(I,J))
     1            -(MLEVAP(J)*MLAREA(I,J))
      USC=YINT+(RI*MBRO(I,J))
      TSC=XSC+USC
      MLVOL(I,J+1)=MLVOL(I,J)+TSC
      MLSC(I,J)=MLVOL(I,J+1)-MLVOL(I,J)
      CALL BATHY (4,MLVOL(I,J+1),MLL(I,J+1),MLAREA(I,J+1))
      IF (J.EQ.12.AND.I.LT.89) THEN
         MLL(I+1,1)=MLL(I,13)
         MLVOL(I+1,1)=MLVOL(I,13)
         MLAREA(I+1,1)=MLAREA(I,13)
```

```
      SUBROUTINE OWENS (I,J)
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
C
C***********************   LONG VALLEY   ***************************
C
C**** ROCK CREEK DIVERSION
C
      IF (RCLRV(I,J).GT.RCKFSH(J)) THEN
         RCD(I,J)=0.8*(RCLRV(I,J)-RCKFSH(J))
      ELSE
         RCD(I,J)=0
      END IF
C
C**** LONG VALLEY INFLOW
C
      LVIN(I,J)=LVRO(I,J)+HCSPR(I,J)-LVIRR(J)+RCD(I,J)+MINWPF(I,J)
     +      +MTMAKE(I,J)
     +      +LVGAIN(1,J)
     +      +(LVGAIN(2,J)*LVRO(I,J))
     +      +(LVGAIN(3,J)*LVIRR(J))
C
C**********************    ROUND VALLEY   ***************************
C
      RCTRV(I,J)=RCLRV(I,J)-RCD(I,J)
      RVIN(I,J)=RCTRV(I,J)+(RVRO(I,J)-RCLRV(I,J))+BCSPR(I,J)
      RVUL=RVIRR(J)+RVREC(J)+RVLOS(1,J)+(RVLOS(2,J)*RVRO(I,J))
     +      +(RVLOS(3,J)*RVIRR(J))
      RVOUT(I,J)=MAX(0,RVIN(I,J)-RVUL)
C
C*** RELEASE ENOUGH WATER TO MEET PV MINIMUM OUTFLOW
C
      LVPVT=RVTRAN(1,J)+(RVTRAN(2,J)*LVSTOR(I,J))+RVOUT(I,J)
      LVRMIN=MAX(0,PVMIN(J)-LVPVT)
      LVOUT(I,J)=LVRMIN
      IF (LVOUT(I,J).GT.LVCAP(J)) LVOUT(I,J)=LVCAP(J)
      PVOUT(I,J)=LVOUT(I,J)+LVPVT
C
C*** RELEASE WATER IF NEEDED TO MATCH TARGET EOM STORAGE
C
      TLVST=LVSTOR(I,J)-LVEV(I,J)+LVIN(I,J)-LVOUT(I,J)
      IF (TLVST.GT.LVSMAX(ILVYT(I),J)) THEN
         LVDUMP=TLVST-LVSMAX(ILVYT(I),J)
         LVRTMT=MIN(LVDUMP,LVCAP(J)-LVOUT(I,J))
         LVOUT(I,J)=LVOUT(I,J)+LVRTMT
         PVOUT(I,J)=PVOUT(I,J)+LVRTMT
      END IF
C
C**************** BISHOP
C
C
C*** USES AND LOSSES
C
      BUSE=BISIRR(J)+BISREC(J)
```

```
      BLOSS=BISLOS(3,J)*BUSE
      BUL=BUSE+BLOSS
C
C*** OWENS RIVER DIVERSION
C
      BISORD(I,J)=(1-BUSEP(IHCYT(I),J))*BUL
C
C*** CHECK DIVERSION WITH PVOUT
C
      MAXDIV=MAX(0,PVOUT(I,J)-PVMIN(J))
      IF (BISORD(I,J).GT.MAXDIV) BISORD(I,J)=MAXDIV
C
C*** BISHOP AREA INFLOW AND OUTFLOW
C
      BISIN(I,J)=BISACT(I,J)+BISSPR(J)+BISORD(I,J)+UBISP(I,J)
      ABUSE(I,J)=UBISP(I,J)+BISORD(I,J)
      IF (ABUSE(I,J).GT.BUSE) ABUSE(I,J)=BUSE
      BUL=ABUSE(I,J)+BLOSS+BISLOS(1,J)
      BISOUT(I,J)=BISIN(I,J)-BUL
      IF (BISOUT(I,J).LT.0) THEN
         DIFF=BUL-BISIN(I,J)
         ABUSE(I,J)=ABUSE(I,J)-DIFF
         BISOUT(I,J)=0
      END IF
C
C*** CALCULATE ACTUAL USES AND OUTFLOW
C
      IF (ABUSE(I,J).GT.BUSE) ABUSE(I,J)=BUSE
      BISOUT(I,J)=BISOUT(I,J)+EBISP(I,J)
C
C*************** LAWS
C
C*** USES AND LOSSES
C
      LUSE=LAWIRR(J)+LAWREC(J)
      LLOSS=LAWLOS(3,J)*LUSE
      LUL=LUSE+LLOSS
C
C*** OWENS RIVER DIVERSION INTO MCNALLY CANALS
C
      LORDIV(I,J)=(1-LUSEP(IHCYT(I),J))*LUL
      LORDIV(I,J)=LORDIV(I,J)+(LAWLOS(4,J)*LORDIV(I,J))
C
C*** CHECK DIVERSION WITH PVOUT, IF TOO HIGH INCREASE PUMPING
C
      MAXDIV=MAX(0,PVOUT(I,J)-PVMIN(J)-BISORD(I,J))
      MAXDIV=MAX(0,MAXDIV-(LAWLOS(4,J)*MAXDIV))
      IF (LORDIV(I,J).GT.MAXDIV) LORDIV(I,J)=MAXDIV
C
C*** LAWS AREA INFLOW AND OUTFLOW
C
      LAWSIN(I,J)=LWRO(I,J)+ULAWP(I,J)+LORDIV(I,J)+FSLU(I,J)
      ALUSE(I,J)=ULAWP(I,J)+LORDIV(I,J)
      IF (ALUSE(I,J).GT.LUSE) ALUSE(I,J)=LUSE
```

```
      LUL=ALUSE(I,J)+LLOSS+LAWLOS(1,J)
      LAWSOUT(I,J)=LAWSIN(I,J)-LUL
      IF (LAWSOUT(I,J).LT.0) THEN
         DIFF=LUL-LAWSIN(I,J)
         ALUSE(I,J)=ALUSE(I,J)-DIFF
         LAWSOUT(I,J)=0
      END IF
C
C*** CALCULATE ACTUAL USES AND OUTFLOW
C
      IF (ALUSE(I,J).GT.LUSE) ALUSE(I,J)=LUSE
      LAWSOUT(I,J)=LAWSOUT(I,J)+ELAWP(I,J)
C
C************** BIG PINE
C
      BPUSE=BIGIRR(J)+BIGREC(J)
      BPLOSS=BIGLOS(3,J)*BPUSE
      BPUL=BPUSE+BPLOSS
C
C*** OWENS RIVER DIVERSION
C
      BPORDIV(I,J)=(1-BPUSEP(IHCYT(I),J))*BPUL
      BPORDIV(I,J)=BPORDIV(I,J)+(BIGLOS(4,J)*BPORDIV(I,J))
C
C*** CHECK DIVERSION
C
      MAXDIV=MAX(0,PVOUT(I,J)-PVMIN(J)-LORDIV(I,J)-BISORD(I,J)
     +      +LAWSOUT(I,J)+BISOUT(I,J))
      IF (BPORDIV(I,J).GT.MAXDIV) BPORDIV(I,J)=MAXDIV
C
C*** BIG PINE AREA INFLOW AND OUTFLOW
C
      BPIN(I,J)=BPRO(I,J)+UBPP(I,J)+BPORDIV(I,J)+XKEO(I,J)
      ABPUSE(I,J)=UBPP(I,J)+BPORDIV(I,J)
      IF (ABPUSE(I,J).GT.BPUSE) ABPUSE(I,J)=BPUSE
      BPUL=ABPUSE(I,J)+BPLOSS+BIGLOS(1,J)
      BPOUT(I,J)=BPIN(I,J)-BPUL
      IF (BPOUT(I,J).LT.0) THEN
         DIFF=BPUL-BPIN(I,J)
         ABPUSE(I,J)=ABPUSE(I,J)-DIFF
         BPOUT(I,J)=0
      END IF
C
C*** CALCULATE ACTUAL USES AND OUTFLOW
C
      IF (ABPUSE(I,J).GT.BPUSE) ABPUSE(I,J)=BPUSE
      BPOUT(I,J)=BPOUT(I,J)+EBPP(I,J)
C
C************ TINEMAHA TO HAIWEE
C
      THWUSE=THWIRR(J)+THWREC(J)
      THEUSE=THEIRR(J)+THEREC(J)
      THWLOSS=THLOS(3,J)*THWUSE
      THELOSS=THLOS(3,J)*THEUSE
```

```
      THWUL=THWUSE+THWLOSS
      THEUL=THEUSE+THELOSS
C
C*** AQUEDUCT RELEASES FOR EAST USES
C
      AVAIL=MAX(0,PVOUT(I,J)+LAWSOUT(I,J)+BISOUT(I,J)+BPOUT(I,J)
     +                 -LORDIV(I,J)-BISORD(I,J)-BPORDIV(I,J))
      IF (AVAIL.GT.THEUL) THEN
         THAR(I,J)=THEUL
      ELSE
         THAR(I,J)=AVAIL
      END IF
C
C*** TINEMAHA TO HAIWEE INFLOW AND OUTFLOW
C
      THIN(I,J)=THRO(I,J)+THSPR(J)+UTHP(I,J)+THAR(I,J)
      IF (UTHP(I,J).GT.THWUSE) THEN
         ATHUSE(I,J)=THWUSE+THAR(I,J)
      ELSE
         ATHUSE(I,J)=UTHP(I,J)+THAR(I,J)
      END IF
      THUL=ATHUSE(I,J)+THWLOSS+THLOS(1,J)
      THOUT(I,J)=THIN(I,J)-THUL
      IF (THOUT(I,J).LT.0) THEN
         DIFF=THUL-THIN(I,J)
         ATHUSE(I,J)=ATHUSE(I,J)-DIFF
         THOUT(I,J)=0
      END IF
C
C***  ACTUAL OUTFLOW
C
      THOUT(I,J)=THOUT(I,J)+ETHP(I,J)
C
C**** TOTAL USES
C
      TOTUSE(I,J)=ALUSE(I,J)+ABUSE(I,J)+ABPUSE(I,J)+ATHUSE(I,J)
      OLAWSOUT(I,J)=LAWSOUT(I,J)
      OBISOUT(I,J)=BISOUT(I,J)
      OBPOUT(I,J)=BPOUT(I,J)
      OTHOUT(I,J)=THOUT(I,J)
      OTHAR(I,J)=THAR(I,J)
C
      RETURN
      END
```

```
      END IF
C
C**** END MONTHLY LOOP
C
  101 CONTINUE
C
C***** END OF ANNUAL LOOP
C
  100 CONTINUE
C
C********* OUTPUT
C
      WRITE (*,200)
  200 FORMAT (' WRITING OUTPUT')
      CALL OUT1 (IYRST,IYREND)
C
      STOP ' SIMULATION COMPLETED'
      END
```

```
      SUBROUTINE TOOMUCH (I,J)
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
C
C******* NEED TO STORE, SPREAD, AND SPILL
C
C            PRIORITY RANKING
C
C         1. STORE IN TINEMAHA AND HAIWEE
C         2. STORE IN LONG VALLEY
C         3. INCREASE EXPORTS
C         4. SPREAD IN LAWS
C         5. SPREAD IN BIG PINE
C         6. SPREAD ON TINEMAHA TO HAIWEE FANS
C         7. AQUEDUCT RELEASES
C
C*** CALCULATE AMOUNT OF WATER THAT IS GOING TO STAY IN VALLEY
C
      DIFF=AVEX(I,J)-HCAP(IHCYT(I),J)
C
C*** STORE IN TINEMAHA AND HAIWEE
C
      IF (DIFF.GT.0) THEN
         TSC=MIN(2000,DIFF)
         TINSTOR(I,J+1)=MIN(TINSTOR(I,J)+TSC,10000)
         TSC=TINSTOR(I,J+1)-TINSTOR(I,J)
         IF (J.EQ.12.AND.I.LT.89) TINSTOR(I+1,1)=TINSTOR(I,J+1)
         DIFF=DIFF-TSC
      END IF
      IF (DIFF.GT.0) THEN
         HSC=MIN(2000,DIFF)
         HAISTOR(I,J+1)=MIN(HAISTOR(I,J)+HSC,10000)
         HSC=HAISTOR(I,J+1)-HAISTOR(I,J)
         IF (J.EQ.12.AND.I.LT.89) HAISTOR(I+1,1)=HAISTOR(I,J+1)
         DIFF=DIFF-HSC
      END IF
C
C*** STORE IN LONG VALLEY
C
      IF (DIFF.GT.0) THEN
         LVORED=0
         TLVS=LVSTOR(I,J)-LVEV(I,J)+LVIN(I,J)-LVOUT(I,J)
         TPVM=MAX(0,PVMIN(J)-(RVOUT(I,J)+RVTRAN(1,J)
     +        +(RVTRAN(2,J)*TLVS)))
         IF (LVOUT(I,J).GT.TPVM) THEN
            LVORED=MIN(DIFF,LVOUT(I,J)-TPVM)
            TLVS=TLVS+LVORED
            MAXLVS=MIN(180000,1.2*LVSMAX(ILVYT(I),J))
            IF (TLVS.GT.MAXLVS) THEN
               RR=TLVS-MAXLVS
               LVORED=LVORED-RR
            END IF
         END IF
         DIFF=DIFF-LVORED
```

```
         LVOUT(I,J)=LVOUT(I,J)-LVORED
         PVOUT(I,J)=PVOUT(I,J)-LVORED
      END IF
C
C*** INCREASE EXPORTS
C
      XX=0
      IF (DIFF.GT.0) THEN
         XX=MIN(DIFF,(1585*DAYS(J))-HCAP(IHCYT(I),J))
         IF (XX.LT.0) XX=0
         HAEX(I,J)=HAEX(I,J)+XX
         DIFF=DIFF-XX
      END IF
C
C*** SPREAD WATER IN LAWS
C
      IF (DIFF.GT.0) THEN
         AVAIL=MAX(0,PVOUT(I,J)-PVMIN(J)-LORDIV(I,J)-BISORD(I,J))
         ALSPRD(I,J)=MIN(LAWSPD(J),DIFF)
         IF (ALSPRD(I,J).GT.AVAIL) ALSPRD(I,J)=AVAIL
         DIFF=DIFF-ALSPRD(I,J)
         LORDIV(I,J)=LORDIV(I,J)+ALSPRD(I,J)
      END IF
      ALSPRD(I,13)=ALSPRD(I,13)+ALSPRD(I,J)
C
C*** SPREAD WATER IN BIG PINE
C
      IF (DIFF.GT.0) THEN
         AVAIL=MAX(0,PVOUT(I,J)-PVMIN(J)-LORDIV(I,J)-BISORD(I,J)
     +           +LAWSOUT(I,J)+BISOUT(I,J))
         ABPSPRD(I,J)=MIN(BIGSPD(J),DIFF)
         IF (ABPSPRD(I,J).GT.AVAIL) ALSPRD(I,J)=AVAIL
         DIFF=DIFF-ABPSPRD(I,J)
         BPORDIV(I,J)=BPORDIV(I,J)+ABPSPRD(I,J)
      END IF
      ABPSPRD(I,13)=ABPSPRD(I,13)+ABPSPRD(I,J)
C
C*** SPREAD ON FANS BETWEEN TINEMAHA AND HAIWEE
C
      IF (DIFF.GT.0) THEN
         ATHFSP(I,J)=MIN(THSPD(J),DIFF)
         AVAIL=MAX(0,THRO(I,J)-THLOS(1,J)-(THLOS(2,J)*THRO(I,J)))
         IF (ATHFSP(I,J).GT.AVAIL) ATHFSP(I,J)=AVAIL
         THOUT(I,J)=THOUT(I,J)-ATHFSP(I,J)
         DIFF=DIFF-ATHFSP(I,J)
      END IF
      ATHFSP(I,13)=ATHFSP(I,13)+ATHFSP(I,J)
C
C*** SPILL AS A LAST RESORT
C
      IF (DIFF.GT.0) THOS(I,J)=DIFF
      THOS(I,13)=THOS(I,13)+THOS(I,J)
C
C*** CALCULATE EXPORT
```

```
C
      HAEX(I,J)=PVOUT(I,J)+BISOUT(I,J)+LAWSOUT(I,J)+BPOUT(I,J)+
     1           THOUT(I,J)-TSC-HSC+XX-THOS(I,J)-THAR(I,J)-LORDIV(I,J)-
     2           BISORD(I,J)-BPORDIV(I,J)-PVTRAN(1,J)-THTRAN(1,J)
C
      RETURN
      END
```

```
      SUBROUTINE NOTENUF (I,J)
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
C
C**********  DECREASE STORAGE, PUMP, REDUCE USES
C
C      PRIORITY RANKING
C
C     1. DRAW DOWN TINEMAHA AND HAIWEE STORAGE
C     2. EXPORT FROM THE MONO BASIN (FISH AND LAKE RELEASE HAVE
C           ALREADY BEEN SATISDIED)
C     3. DO NOT INCREASE STORAGE IN LONG VALLEY (RELEASE ALL INFLOW)
C     4. DECREASE LV STORAGE TO MINIMUM
C     5. REDUCE OV USES IF THIRD CONS. DRY YEAR
C     6. REDUCE EXPORT
C
C**** CALCULATE AMOUNT OF WATER NEEDED TO MEET EXPORT TARGET
C
      DIFF=HCAP(IHCYT(I),J)-AVEX(I,J)
C
C*** MAKE UP DEFICIENCIES
C
C*** TINEMAHA AND HAIWEE
C
      IF (DIFF.GT.0) THEN
         TSC=MIN(DIFF,2000)
         TINSTOR(I,J+1)=MAX(0,TINSTOR(I,J)-TSC)
         TSC=TINSTOR(I,J)-TINSTOR(I,J+1)
         IF (J.EQ.12.AND.I.LT.89) TINSTOR(I+1,1)=TINSTOR(I,J+1)
         DIFF=DIFF-TSC
      END IF
      IF (DIFF.GT.0) THEN
         HSC=MIN(DIFF,2000)
         HAISTOR(I,J+1)=MAX(0,HAISTOR(I,J)-HSC)
         HSC=HAISTOR(I,J)-HAISTOR(I,J+1)
         IF (J.EQ.12.AND.I.LT.89) HAISTOR(I+1,1)=HAISTOR(I,J+1)
         DIFF=DIFF-HSC
      END IF
C
C*** MONO BASIN EXPORT
C
      IF (DIFF.GT.0) THEN
         WPF(I,J)=MIN(DIFF,ATWP(I,J))
         IF (LVOUT(I,J)+WPF(I,J).GT.LVCAP(J)) THEN
            RWP=LVOUT(I,J)+WPF(I,J)-LVCAP(J)
            WPF(I,J)=WPF(I,J)-RWP
         END IF
         LVIN(I,J)=LVIN(I,J)+WPF(I,J)
         LVOUT(I,J)=LVOUT(I,J)+WPF(I,J)
         PVOUT(I,J)=PVOUT(I,J)+WPF(I,J)
         DIFF=DIFF-WPF(I,J)
      END IF
C
C******* LONG VALLEY
```

```
C
C** RELEASE TO MATCH LAST MONTH'S STORAGE IF POSSIBLE
C
      IF (DIFF.GT.0) THEN
         TLVS=LVSTOR(I,J)-LVEV(I,J)+LVIN(I,J)-LVOUT(I,J)
         IF (TLVS.GT.LVSTOR(I,J).AND.
     +       LVSTOR(I,J).GT.LVSMIN(ILVYT(I),J)) THEN
            ORED=MIN(TLVS-LVSTOR(I,J),LVCAP(J)-LVOUT(I,J))
            IF (ORED.LT.0) ORED=0
            LVNER1=MIN(DIFF,ORED)
            LVOUT(I,J)=LVOUT(I,J)+LVNER1
            PVOUT(I,J)=PVOUT(I,J)+LVNER1
         END IF
      END IF
      DIFF=DIFF-LVNER1
C
C*** RELEASE MORE FROM STORAGE IF NEEDED (TO MINIMUM)
C
      IF (DIFF.GT.0) THEN
         TLVS=LVSTOR(I,J)-LVEV(I,J)+LVIN(I,J)-LVOUT(I,J)
         SAV=MAX(0,TLVS-LVSMIN(ILVYT(I),J))
         IF (SAV.GT.0) THEN
            MAXDROP=MIN(DIFF,SAV)
            LVNER2=MIN(LVCAP(J)-LVOUT(I,J),MAXDROP)
            LVOUT(I,J)=LVOUT(I,J)+LVNER2
            PVOUT(I,J)=PVOUT(I,J)+LVNER2
          END IF
      END IF
      DIFF=DIFF-LVNER2
C
C****** REDUCE USES IF NEEDED
C
      IF (DIFF.GT.0) THEN
         LR=ALUSE(I,J)*REDFAC(I)
         BR=ABUSE(I,J)*REDFAC(I)
         BPR=ABPUSE(I,J)*REDFAC(I)
         THR=ATHUSE(I,J)*REDFAC(I)
         REDUC(I,J)=LR+BR+BPR+THR
         IF (DIFF.GT.REDUC(I,J)) THEN
            DIFF=DIFF-REDUC(I,J)
         ELSE
            REDUC(I,J)=DIFF
            AHRF=0.00
  901       AHRF=AHRF+0.01
            LR=ALUSE(I,J)*AHRF
            BR=ABUSE(I,J)*AHRF
            BPR=ABPUSE(I,J)*AHRF
            THR=ATHUSE(I,J)*AHRF
            TR=LR+BR+BPR+THR
            IF (AHRF.GT.REDFAC(I)) GOTO 902
            IF (TR.LT.REDUC(I,J)) GOTO 901
  902       DIFF=DIFF-(LR+BR+BPR+THR)
         END IF
         ALUSE(I,J)=ALUSE(I,J)-LR
```

```
           ABUSE(I,J)=ABUSE(I,J)-BR
           ABPUSE(I,J)=ABPUSE(I,J)-BPR
           ATHUSE(I,J)=ATHUSE(I,J)-(THR*0.5)
           THAR(I,J)=THAR(I,J)-(THR*0.5)
           LAWSOUT(I,J)=LAWSOUT(I,J)+LR
           BISOUT(I,J)=BISOUT(I,J)+BR
           BPOUT(I,J)=BPOUT(I,J)+BPR
           THOUT(I,J)=THOUT(I,J)+THR
        END IF
        REDUC(I,13)=REDUC(I,13)+REDUC(I,J)
        TOTUSE(I,J)=ALUSE(I,J)+ABUSE(I,J)+ABPUSE(I,J)+ATHUSE(I,J)
C
C**************REDUCE EXPORT, IF NEEDED
C
        IF (DIFF.GT.0) EXPRED(I,J)=DIFF
        EXPRED(I,13)=EXPRED(I,13)+EXPRED(I,J)
C
C************ HAIWEE EXPORT
C
        HAEX(I,J)=PVOUT(I,J)+TSC+HSC+LAWSOUT(I,J)+BISOUT(I,J)+
     1           BPOUT(I,J)+THOUT(I,J)-PVTRAN(1,J)-THTRAN(1,J)-
     2           THAR(I,J)-LORDIV(I,J)-BISORD(I,J)-BPORDIV(I,J)
C
        RETURN
        END
```

```
      SUBROUTINE READPUMP (IYRST,IYREND)
      IMPLICIT REAL (A-H,L-Z)
      INCLUDE 'LAAMP.CMB'
C
      OPEN (1,FILE='PUMPING.IN')
C
      DO 100 I=IYRST,IYREND
      DO 101 J=1,12
      READ (1,102) I1,I2,X1,X2,ULAWP(I,J),ELAWP(I,J),X3,
     1                   X1,X2,UBISP(I,J),EBISP(I,J),X3,
     2                   X1,X2,UBPP(I,J),EBPP(I,J),X3,
     3                   X1,X2,UTHP(I,J),ETHP(I,J),X3
  102 FORMAT (2I5,20F8.0)
C
  101 CONTINUE
  100 CONTINUE
C
      RETURN
      END
```