US Army Corps
of Engineers
Hydrologic Engineering Center

HEC-PRM
Prescriptive Reservoir Model

User's Manual

Version 1.0
November 2003

CPD-95

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
November 2003

2. REPORT TYPE
Computer Program Documentation

3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE
HEC-PRM, Prescriptive Reservoir Model
User's Manual
Version 1.0

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Bob Carl

5d. PROJECT NUMBER

5e. TASK NUMBER

5F. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
US Army Corps of Engineers
Institute for Water Resources
Hydrologic Engineering Center (HEC)
609 Second Street
Davis, CA 95616-4687

8. PERFORMING ORGANIZATION REPORT NUMBER
CPD-95

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/ MONITOR'S ACRONYM(S)

11. SPONSOR/ MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
Also, see HEC-ResPRM Quick Start Guide, CPD-95a

14. ABSTRACT
HEC-PRM ("Prescriptive Reservoir Model") is a generalized computer program that performs deterministic network-flow optimization of reservoir system operations. HEC's Data Storage System (HEC-DSS) is used for storage and retrieval of input and output time-series data. This introductory guide leads the reader through the three modules of the software, using a simplified but informative example to illustrate the modeling process.

15. SUBJECT TERMS
HEC-PRM, Water Resources System Optimization, Reservoir Optimization, Computer Program

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
U	U	U	U	118	19b. TELEPHONE NUMBER

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39-18

HEC-PRM
Prescriptive Reservoir Model

User's Manual

November 2003

US Army Corps of Engineers
Institute for Water Resources
Hydrologic Engineering Center
609 Second Street
Davis, CA 95616

(530) 756-1104
(530) 756-8250 FAX
www.hec.usace.army.mil

CPD-95

Prescriptive Reservoir Model, HEC-PRM, User's Manual

Use of the software described by this document is controlled by certain terms and conditions. The user must acknowledge and agree to be bound by the terms and conditions of usage before the software can be installed or used. For reference, a copy of the terms and conditions of usage are included in Chapter 11 of this document so that they may be examined before obtaining and loading the software. The software described by this document can be downloaded for free from our internet site (www.hec.usace.army.mil).

HEC cannot provide technical support for this software to non-Corps users. Non-Corps individuals should use any internet search engine to locate a vendor that can provide support for the HEC software of interest. However, we will respond to all documented instances of program errors. Documented errors are bugs in the software due to programming mistakes not model problems due to user-entered data.

Table of Contents

List of Figures

List of Tables

CHAPTER 1 - Program Description

Overview of HEC-PRM Procedures and Related Programs

The overall schematic of the Hydrologic Engineering Center's (HEC) Prescriptive Reservoir Model (HEC-PRM) and its relationship to other programs is shown in Figure 1. HEC-PRM requires the use of HEC's Data Storage System (HEC-DSS). HEC-PRM must read the penalty functions, local incremental inflows, elevation-area-capacity relationships, and evaporation rates from HEC-DSS data files - it cannot read them from any other source. As a result, several programs are used to enter and edit input data in preparation for applying HEC-PRM.

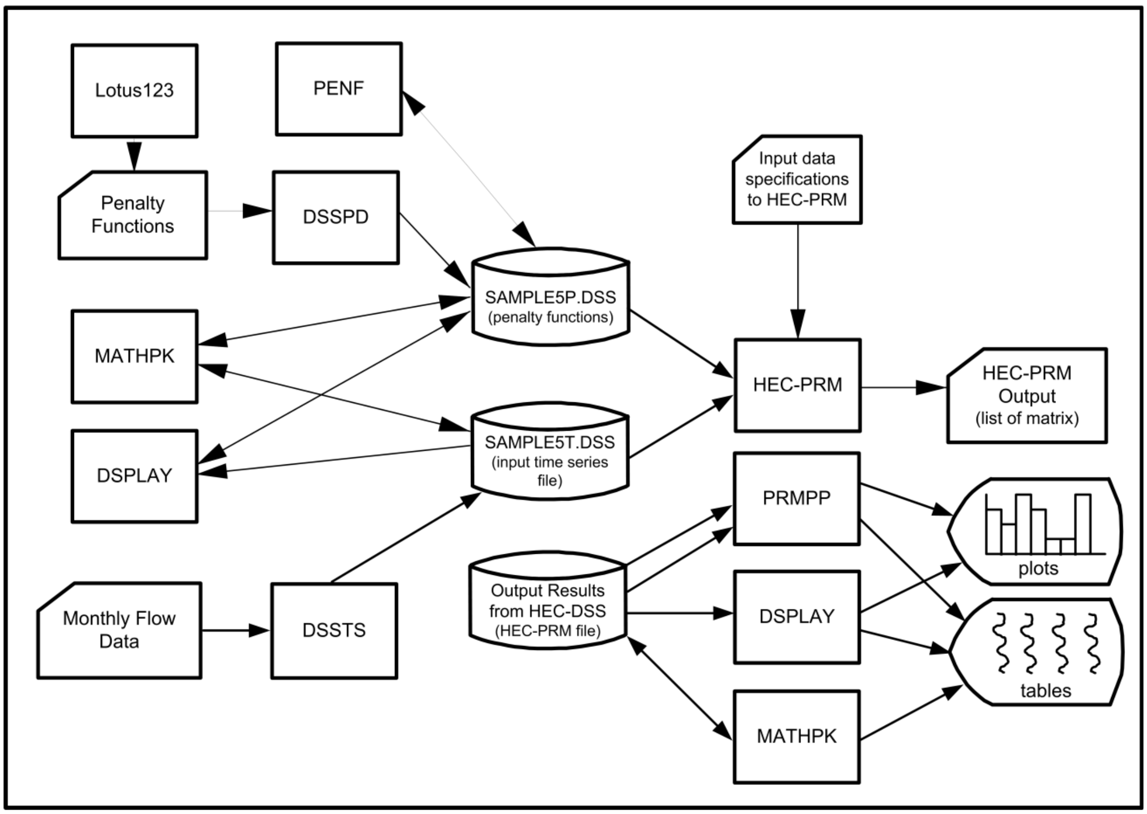

Figure 1. Schematic of HEC-PRM and Related Programs

Typical Procedures for Developing Data Files and Applying HEC-PRM

Although some operations may be performed in a different order, the following procedures are typical for any HEC-PRM study:

1) Enter regular interval time series data into the input time series HEC-DSS data file.

The regular interval time series data consists of incremental local inflows, incremental local depletions (optional), evaporation rates (optional), aigurend natural flows (optional). The standard HEC-DSS utility programs DSSTS or DSSIN may be used to store regular interval monthly time series data in the file. Alternatively, specialized programs, including spreadsheet macros, may be written to read existing formatted data and store it in the HEC-DSS file.

2) **Compute the adjusted local incremental inflow by adding inflows and depletions.**

   The generalized utility programs DSS-MATH or MATHPK allow the user to add, subtract, multiply, divide, etc., hydrographs and store the results in the same or a different HEC-DSS data file. As an alternative, HEC-PRM allows the user to enter multiple inflow hydrographs to one node where one hydrograph may be the inflow and a second hydrograph the fixed depletions. However, each inflow hydrograph requires a separate link, which adds complexity to the HEC-PRM processing.

3) **Enter data and calculate the storage and flow penalty functions for each location and category (water supply, hydropower, etc), and store penalty functions in the "paired data" HEC-DSS data file.**

   These calculations are typically performed using commercial spreadsheet software. This may require the careful selection of format so that the results can be transformed into an ASCII input data file for the DSSPD program. (DSSPD reads an "xxxxxx.txt" file from an Excel spreadsheet and stores the penalty functions using the "Paired Data" convention in HEC-DSS.) Alternatively, the "Store DATAFILE" option of the PIP program may be used. Specialized programs, including spreadsheet macros, may also be written to read existing formatted data and store it in the HEC-DSS file.

4) **Convert the penalty functions into "standard" units.**

   It may be necessary to convert penalty functions into "standard" units before using them as input to HEC-PRM. All penalty functions must have the same penalty units (e.g., $1,000) and storage/flow units (e.g., KAF or thousands of acre-feet per month). For example, storage penalties expressed as penalty in millions of dollars versus pool elevation in feet must be converted to penalty in thousands of dollars versus storage in thousands of acre-feet. Flow penalties expressed as penalty versus flow in thousands of cubic feet per second must be converted to penalty versus flow in thousands of acre-feet per month. (An average value of 30.5 days per month may be used to determine this factor.) The MATHPK function "TABLE" facilitates this.

5) **Compute a composite penalty function for each month at each location.**

   The ultimate goal is to have a separate pathname for each composite penalty function for each location. This step creates the "computed composite" function by adding functions for all categories (except hydropower energy) for a given link. For example, to compute the June composite reservoir storage penalty function, the analyst must add the June navigation storage function, the June pool recreation function, and the June water supply function and store the result in the HEC-DSS data file. The composite computed function is stored in the DSS data file as one record (pathname) for one location and one link type for all months of the year (twelve curves). Note that the hydropower energy functions are stored separately because separate links are defined for hydropower energy only.

6) **Estimate the "edited composite" penalty function from the "computed composite" penalty function. (Optional)**

   Estimating the "edited composite" (or "model") penalty function serves to reduce the number of arcs describing the functions. The "computed composite" function typically contains many ordinates. For computational efficiency, the analyst should determine the simplest "edited" function that still adequately describes the "computed" function. This edited function may be convex or non-convex. In general, convex functions should be used whenever possible, since the solution algorithm guarantees a global optimum solution in this case (unless hydropower is considered). However, non-convex functions may be preferable if these provide a much better representation of the actual (computed) composite penalty function. The DSPLAY program, the PENF program, or commercial spreadsheet software can assist the analyst in this task.

7) Store pertinent paired data functions in the paired function input data file.

Other paired data needs to be entered in the input data DSS data file. This includes hydropower functions relating energy and capacity to flow and reservoir storage, and elevation-area-capacity curves.

8) Prepare HEC-PRM input data specifications.

The analyst invokes a text editor (such as COED) to create and enter information pertinent to the HEC-PRM program. This is an ASCII file which contains some miscellaneous parameters such as the time window associated with the calculations, the HEC-DSS pathname part F under which the computed results are stored, factors which are applied against flow and penalty data, etc. The primary content of this file is a list of nodes and a list of links with associated information such as the connected nodes, the pathname parts for the penalty functions and the regular time series data (inflow and evaporation), upper and lower bounds, etc.

9) Perform the network flow optimization.

To optimize the network, HEC-PRM reads the ASCII data specifications file, retrieves the appropriate time series data from the time series input DSS data file, retrieves the appropriate "edited composite" penalty functions from the paired input data DSS file, generates the network flow solver matrix, calls the solver, and stores the results in the output HEC-DSS data file.

10) Display results graphically or tabular.

The analyst may use either the HEC-PRM post-processor PRMPP, or the DSPLAY program, or commercial spreadsheet software to plot or tabulate the time series results. These include reservoir storage, reservoir releases, channel flows, local inflows, natural flows, energy generated, and capacity. Storage data is expressed in units of thousands of acre-feet. Flow data is expressed in both thousands of acre-feet (KAF) and cubic feet per second (cfs). The MATHPK program may be used to convert these into other units such as pool storage in terms of elevation in feet or flow in terms of thousands of cubic feet per second. Extensive MATHPK macros are written to compute hydropower, duration functions, time series penalties, etc.

11) Review the solver Matrix.

Figure 1 also depicts HEC-PRM writing output data to an ASCII file. It contains an echo of the user input including pathnames, warning messages, and optionally a list of the solver matrix both before and after solution. While some of the information is useful, it is very painful to look at the solver matrix and should be done only for optimization models with a limited number of arcs. Useful output can be created using the PRMPP and DSPLAY programs and possibly MATHPK. In particular, DSPLAY facilitates the output of graphs in the standard HEC-DSS graphics format and includes the capability of creating graphics metafiles which may be imported to other software packages such as word processors and drawing programs.

General Description of HEC-PRM Software

HEC-PRM consists of about 130 program specific subroutines of which forty-six are the generalized solver. It also utilizes many routines from HEC's software library including HEC-DSS routines. HEC-PRM's routines do the following:

- Assign disk files.
- Read user defined input.
- Print user input.
- Read all penalty functions (storage and flow), other paired data functions (elevation-area-capacity) and time series data (evaporation and inflow) from input HEC-DSS data files.
- Generate the solver matrix.
- Print the matrix.
- Call the solver routines.
- Print the computed solver matrix.
- Store the results in an output DSS data file.
- Close all disk files.

Structure of HEC-PRM

The internal flow of information within HEC-PRM is shown in Table 1. HEC-PRM retrieves input data from three sources:

Table 1. HEC-PRM File Assignments

```
HECPRM ?
HECPRM: Prescriptive Reservoir Model - Vers. September 5, 2002  ( 1.019)
    UNIT      KEYWORD          *ABREV    **MAX   DEFAULT
       5      INPUT            I         458     CON
       6      OUTPUT           O         512     CON
     NOP      TS_IN_DSS        T         512     SCRATCH.031
     NOP      PF_IN_DSS        P         512     SCRATCH.032
     NOP      RESULTS_DSS      R         512     SCRATCH.033
     NOP      MIN_COST_SOLN    M          64     PRM_MCST.BIN
      29      TRACE            TR        512     SCRATCH.009
       1      MSG              MS        512     C:\HECEXE\SUP\HECPRM.ERR

    * ABREV - SHORTEST ABBREVIATION ALLOWED FOR KEYWORD
    ** MAX - MAXIMUM # OF CHARACTERS FOR FILENAME (OR STRING)
```

1) An ASCII data specifications file,
2) An HEC-DSS data file containing penalty functions and other paired data, and
3) An HEC-DSS data file containing regular interval time series data.

The input data relationships are shown in Figure 2. HEC-PRM reads the ASCII data specifications file first. It defines many various parameters such as the computational time window, the nodes, and the links. HEC-PRM stores the number of nodes and links as well as associated information such as the nodes which are connected by each link and the pathnames which define the storage location within the HEC-DSS data files for the penalty functions, inflows, and evaporation rates. It then generates the solver matrix which consists of arcs each of which is defined by the following parameters:

1) Source node
2) Target node
3) Lower bound
4) Upper bound
5) Unit cost
6) Amplitude (for the gains solver)
7) Flow (initialized to zero)

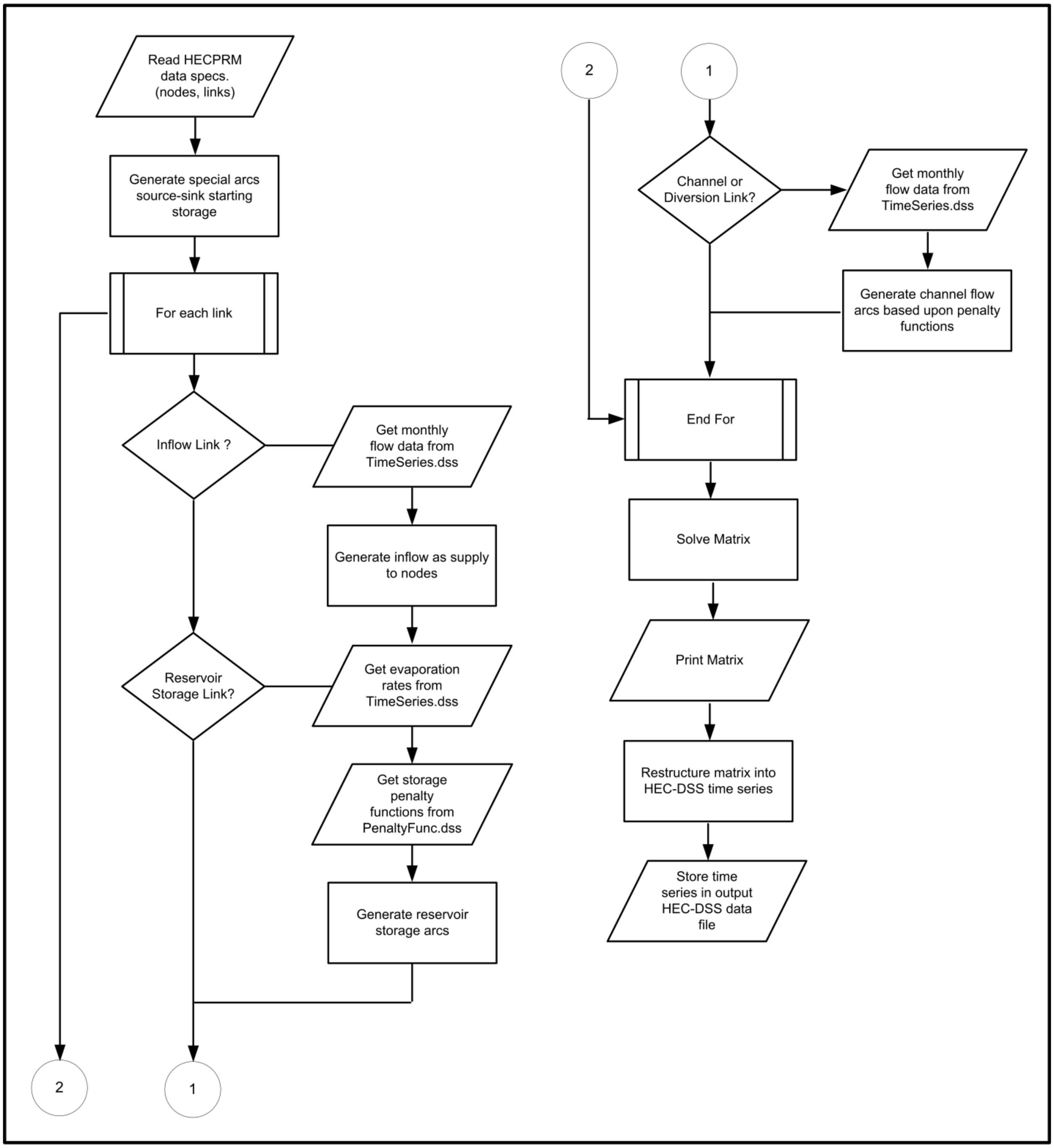

Figure 2. Internal Processes of HEC-PRM

There are several other scalar parameters that must also be set. HEC-PRM builds this matrix by generating one or more "special" arcs and then processing each link in the same sequence in which they were entered by the user. The first arc is always the arc between the super source and the super sink. The second through NRES+1 arcs contain the starting storage for each reservoir where "NRES" is the number of reservoirs. The subsequent arcs are dependent upon the order and type of links entered by the user. When appropriate, data are retrieved from HEC-DSS data files. For example, the inflow links require the retrieval of local incremental adjusted inflow; storage links require the retrieval of evaporation rates and storage penalty functions, etc.

If the user has defined constraints for a given link, the lower and upper bounds on the arcs are set accordingly. The number of arcs associated with a given link for a given month is exactly the same as the number of line segments in the penalty function. All inflow arcs have one arc per time period and have lower and upper bounds which vary for every time period and are set to the local inflow with a unit cost of zero. Once the solver matrix is filled and appropriate parameters are set (such as the number of arcs, the number of nodes, and the total system inflow), HEC-PRM calls the network flow with gains solver. HEC-PRM tracks the progress of solution by displaying on the computer screen the number of iterations and the computed total flow. The solver continues to iterate until the least-cost solution is determined. The typical application contains many unknowns with far fewer equations. Therefore, there can be many feasible solutions to the problem. Minor changes to input data can cause the solver to compute an entirely different least-cost solution.

Once the solution is determined, the solver fills the "FLOW" array with the computed flow in each arc of the matrix. HEC-PRM can then "post-process" this information by adding the flow in all arcs for each time period and storing monthly regular interval time series data in the output HEC-DSS data file. The output monthly time series data includes flow, storage, pool elevations, natural flows, local flows, energy generated, and capacity. Total cost is reported in the ASCII output file. Cost sub-totals (e.g., total hydropower penalty) can be computed by using the program MATHPK and interpolating time series penalties from the appropriate flow and storage penalty functions.

Executing HEC-PRM

HEC-PRM requires the use of the HEC-DSS software. All input penalty functions are read from one DSS data file, all time series input data is read from another DSS data file, and time series and frequency-duration results are written to a third HEC-DSS data file. The files may be defined using "MENUPRM" as described in Chapter 2. Alternatively, these data files may be defined at the DOS prompt which executes HEC-PRM as shown below:

To execute the program, you must enter the program name followed by the appropriate file names. Use the "*ABREV" codes to define the file type. The following files should be defined: INPUT, OUTPUT, TS_IN_DSS, PF_IN_DSS, and RESULTS_DSS. The file "HECPRM.ERR" contains error messages. It is supplied with the program and it should be copied into the \hecexe\sup subdirectory. An example command to execute HEC-PRM is as follows:

```
HECPRM I=ALT1.PRI O=ALT1.PRO T=NPDQ P=NPD_PEN R=ALT1
```

The corresponding files are shown in Table 2.

Data Units

All flow and storage must be entered in consistent units for both the time series data and penalty functions. Default units are thousands of ace-feet (KAF) per month for flow data; thousands of dollars (K$) per thousand ace-feet per month for penalty data; and, feet (FT) per month for evaporation rates. The data may be entered in other units (e.g., penalties in millions of dollars), but the analyst must then perform additional record keeping tasks. HEC-PRM results will be reported (incorrectly) using the default units.

Order of User Input

The records described on the following pages should be entered in the order shown in Table 3.

Table 2. Example Command Line Parameter Description

Key Word	File Name	Description
INPUT	ALT1.PRI	User input data which defines many items including the time window, the nodes, the links, the pathname parts for the penalty functions, the pathname parts for the time series data, etc.
OUTPUT	ALT1.PRO	The tabular output from the program including a listing of the user input (file ALT1.PRI) and a listing of the solver matrix before and after the solution.
TS_IN_DSS	NPDQ.DSS	The HEC-DSS data file which contains all input time series data including the rate of evaporation (EV records) and incremental inflows (IN records).
PF_IN_DSS	NPD_PEN.DSS	The HEC-DSS data file which contains all input penalty functions including flow (PQ records) and reservoir storage (PS records).
RESULTS_DSS	ALT1.DSS	The HEC-DSS data file which contains all computed time series data. The local inflows are also written to this file.

Table 3. Order of Input Records

First	All job control records such as "TIME", "J1", "IDENT", and "ZW".
Second	All node identifier records ("NODE").
Third	All link definition records, including: "LINK", "BL", "BU", "PS", "PS2", "PQ", "PQ2", "QC", "QI", "QL", "QU", "EAC", "EV", and "IN". The "LINK" record is the first record of each link. There is one "LINK" record for each link in the network. All other "LINK" records which are required to define that link follow the "LINK" record.
	Comment records ("..", "**", or " ") may be entered anywhere in the input file.

Note: In the following description, the character "=" indicates a blank character. Numeric data should be right-justified in the fields. The editor COED will automatically justify all records when it is used in the "help program" mode (which is the default for editing HEC-PRM input data files). Be careful, the editor defaults to 80 columns of input data unless it can detect more. If it assumes 80 columns of input, anything entered beyond 80 columns will be lost. All fields which allow the last data value to be entered in column "n" indicate that the field extends through column 240. However, the internal array holding the information may store less than this.

CHAPTER 2 - Overview of HEC-PRM

Introduction

As its name implies, HEC-PRM is a prescriptive model that addresses a reservoir system operation problem as one of optimal long-term allocation of available water. The model identifies the allocation that maximizes benefits (or minimizes costs associated with poor performance) for all defined system purposes. Performance is measured with analyst-provided penalty functions of flow, storage, or both.

To determine the optimal water allocation, the physical system is represented as a network, and the operating problem is formulated as a minimum-cost network flow problem. The objective function of this network problem is the sum of piecewise-linear approximations of the penalty functions. An off-the-shelf solver is used to define the optimal allocation of water within the system. The results of the solver are processed to report and display reservoir releases, storage volumes, channel flows, and other pertinent variables.

The HEC-PRM software is general-purpose. Accordingly, the software includes the following model-building components:

1) Inflow link;
2) Initial-storage link;
3) Diversion link;
4) Final-storage link;
5) Channel-flow link;
6) Simple reservoir-release link;
7) Hydropower reservoir-release link;
8) Reservoir-storage link; and
9) Node.

An analyst can specify the characteristics and configuration of these components to represent any water resources system.

Problem Statement

The problem addressed by HEC-PRM is identification of the optimal long-term operation plan for a reservoir system. This plan will identify the priorities to be assigned to conflicting objectives of operation. For example, the plan will determine whether water should be released from a reservoir if a demand exists for downstream flow for wildlife protection but a conflicting demand exists for continued storage of the water for recreation.

The model quantifies system performance for various purposes in multi-objective terms. In many cases, the economic cost of operation may be the primary consideration, but social and environmental costs can also be considered. These costs are expressed in commensurate terms to permit display of trade-offs in operation for various purposes.

Constraints on the physical system can be included. For example, the outlet capacity of a reservoir can be modeled explicitly. However, inviolable constraints on system operation are used frugally. This avoids the problem described by Hitch and McKean (1960) when they wrote "...casually selected or arbitrary constraints can easily increase system cost or degrade system performance many fold, and lead to solutions that would be unacceptable to the person who set the constraints in the first place." Instead, operation limitations should be imposed through value functions in order to permit clear evaluation of the impacts of limitations. For example, instead of specifying maximum flow requirements for flood control, the system model should represent this objective through high costs of failure to maintain flows or storage levels below flood stage.

Solution Procedure

HEC-PRM considers the reservoir operation planning problem as a problem of optimal allocation of available water. The solution procedure for this water allocation problem is as follows:

1) Represent the physical system as a network;
2) Formulate the allocation problem as a minimum-cost network flow problem;
3) Develop an objective function that represents desirable operation;
4) Solve the network problem with an off-the-shelf solver; and
5) Process the network results to define, in convenient terms, system operation.

Represent System as a Network

For solution of the water allocation problem, the reservoir system is represented as a network. A network is a set of arcs that are connected at nodes. The arcs represent any facilities for transfer of water between two points in space or time. For example, a natural channel transfers water between two points in space, and the reservoir transfers water between two points in time. Both are represented by arcs.

Network arcs intersect at nodes. The nodes may represent actual river or channel junctions, gage sites, monitoring sites, reservoir sites, or water-demand sites. Flow is conserved at each node: the total volume of water in arcs originating at any node equals the total volume in arcs terminating at that node.

Figure 3 illustrates a simple network representation. Node 3 represents a reservoir. Node 4 represents a downstream demand point. Two additional nodes with associated arcs are included to account completely for all water entering and leaving the system. Node 1 is the source node, a hypothetical node that provides all water for the system. Node 2 is the sink node, a hypothetical node to which all water from the system returns. The arc from Node 1 to Node 3 represents the reservoir inflow. The arcs shown as dotted lines represent the beginning-of-period (BOP) and end-of-period (EOP) storage in the reservoir. The BOP storage volume flows into the network from the source node, and the EOP volume flows from the network back to the sink

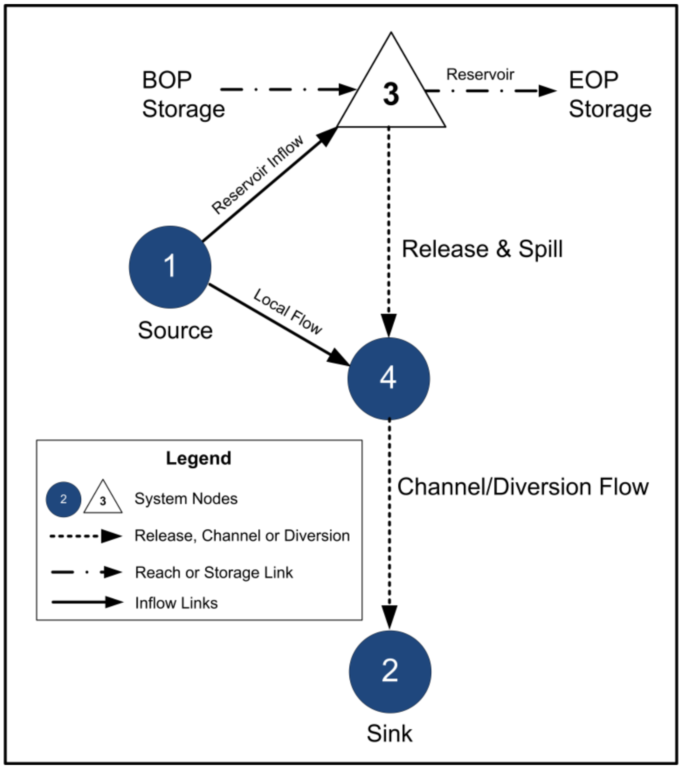

Figure 3. Simplified Single-Period Network

node. The arc from Node 3 to Node 4 represents the total reservoir outflow. The arc from Node 1 to Node 4 represents the local runoff downstream of the reservoir. The arc from Node 4 to Node 2 carries water from the reservoir/demand point network to the sink.

To analyze multiple-period system operation, a layered network is considered. Each layer represents one month. To develop such a layered network, the single-period network representation is duplicated for each time period to be analyzed. Figure 4 illustrates this. A single source node and a single sink node are included. For clarity, these have been omitted from Figure 4. The duplicate networks are connected by arcs that represent reservoir storage. For example, in Figure 4, the arc connecting Node 3 in Period 1 to Node 3 in Period 2 represents the storage. Then flow along this arc is the end-of-period 1 storage, which is equivalent to the beginning-of-period 2 storage. Likewise, the flow along the arc connecting Node 3 in Period 2 to Node 3 in Period 3 represents the end-of-period 2 storage, which also is the beginning-of-period 3 storage.

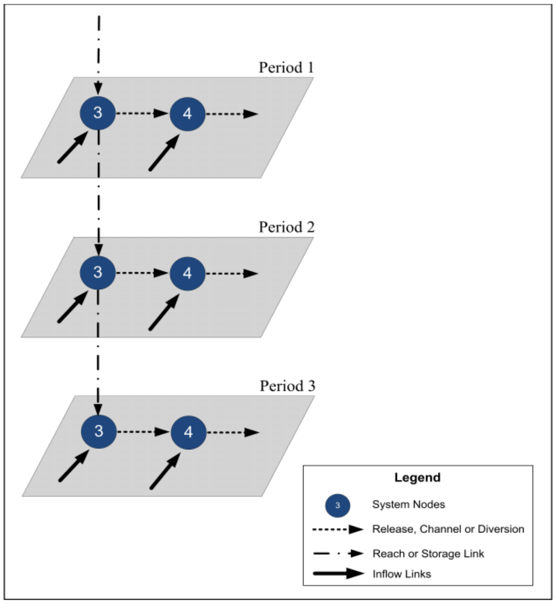

Figure 4. Multiple Period Networks

Formulate the Allocation Problem as a Minimum-cost Network-flow Problem

The goals of and constraints on water allocation within the reservoir system can be represented in terms of flows along the arcs of the network. If a unit cost is assigned for flow along each arc, the objective function for the network is the total cost for flow in all arcs. The ideal operation will be that which minimizes this objective function while satisfying any upper and lower bounds on the flow along each arc. The solution also must maintain continuity at all nodes.

Minimum-cost Objective Function. A network solver finds the optimal flows for the entire network simultaneously, based on the unit cost associated with flow along each arc (e.g., Jensen and Barnes 1987). The functions that specify these costs are defined by the analyst.

The simplest cost function is a linear function, such that shown in Figure 5. This function represents the cost for flow along one arc of a network. The unit cost is the slope of the function. In this case, the unit cost is positive, and the cost increases steadily as the flow increases in the arc.

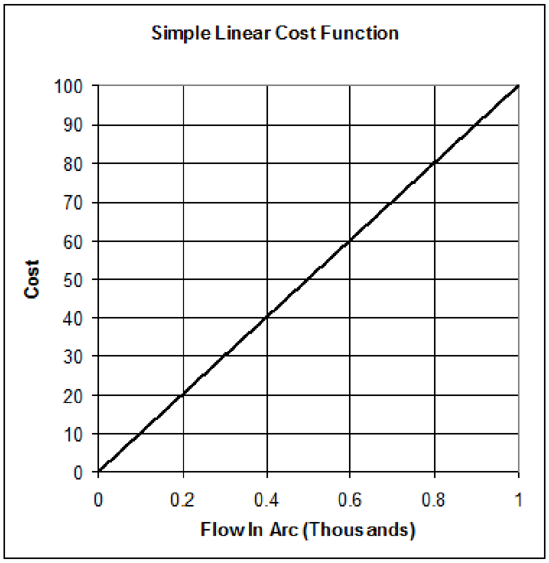

Figure 5. Simplified Linear Cost Function

Unit costs may also be negative, representing benefits for increased flow. The total cost for flow along the arc represented is the product of flow and the unit cost.

The simplest linear function may be too simple to represent adequately many of the goals of reservoir operation. Instead, nonlinear functions, such as those shown in Figure 6, maybe required.

***Piecewise-linear Approximation*.** Nonlinear cost functions can be approximated in a piecewise linear fashion for network model, as shown in Figure 7. Linear segments are selected to represent the pertinent characteristics of the function. The analyst controls the accuracy of the approximation, with more linear segments yielding a more accurate representation. However, the time required for solution of the resulting network-flow programming problem depends on the number of arcs included in the network. Thus, as the approximation improves, the time for solution will likely increase. Jensen and Barnes discuss this approximation in detail (1987, pp. 355-357).

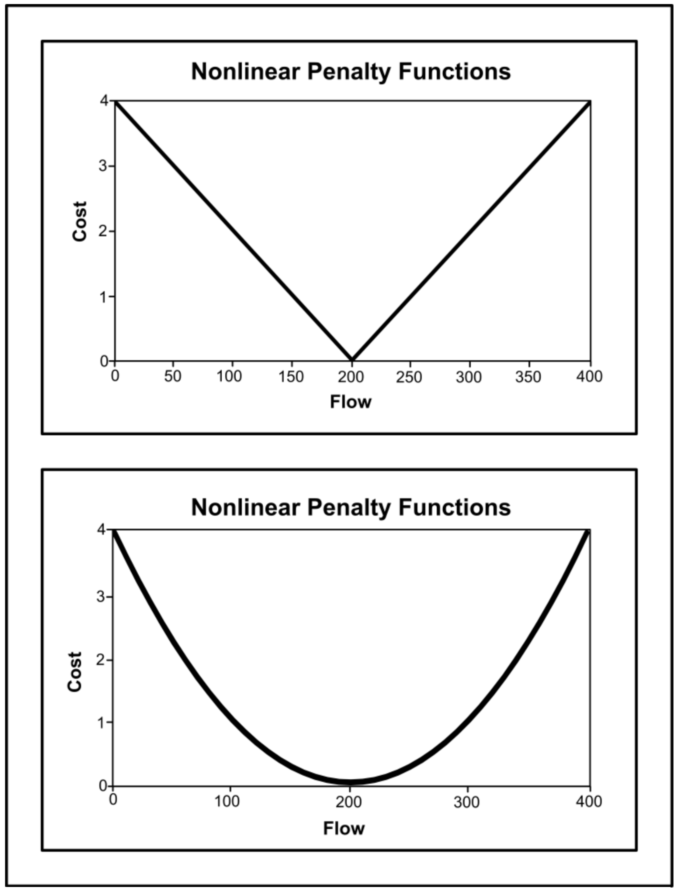

Figure 6. Nonlinear Penalty Functions

The cost functions shown in Figures 6 and 7 are convex – their slope is non-decreasing from left to right. In general, it is best to use convex cost functions, or convex piecewise linear approximations, whenever these represent reality reasonably well. For a model with all convex cost functions, the solution procedure guarantees a globally optimal solution (if a feasible solution exists).

Cases arise, however, in which cost functions are non-convex. For example, the cost of flood damage along a river reach may first increase rapidly as urban or industrial areas near the river are flooded, but then increase at a slower rate as outlying rural areas are flooded. HEC-PRM allows the analyst to specify a non-convex piecewise linear approximation, but a specialized solution procedure called restricted basis entry (e.g., Hadley 1964, pp. 104-111) is required. This procedure will identify a locally optimal solution (if a feasible solution exists), but it cannot guarantee the global optimum. In many practical problems, there are likely to be many local optima. The analyst may find it useful to identify several of these by starting the solution procedure with different initial values.

With a piecewise linear approximation, the physical link for which the function applies is represented in the network by a set of parallel arcs. One arc is included for each linear segment of the piecewise approximation. For example, suppose the cost function in Figure 7 represents the cost of release from the reservoir represented by Node 3 in Figure 3. In the

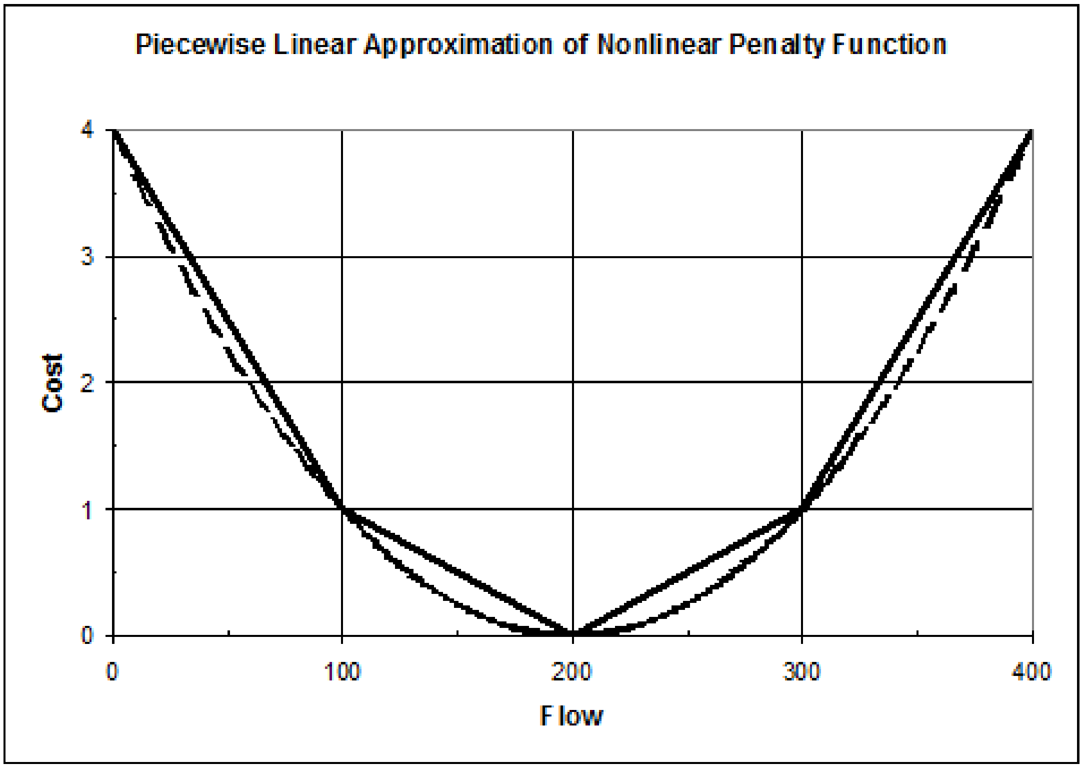

Figure 7. Piecewise Linear Approximation of Non Linear Penalty Function

proposed network model, four parallel arcs will connect Node 3 to Node 4. Characteristics of the arcs are shown on Table 4.

Table 4. Network Model Arc Characteristics for Piecewise Linear Approximation

Arc Number (1)	Lower Bound (2)	Upper Bound (3)	Unit Cost (4)
1	0	= 100	(1-4) / 100 = -0.03
2	0	200 – 100 = 100	(0-1) / 100 = -0.01
3	0	300 – 200 = 100	(1-0) / 100 = 0.01
4	0	400 – 300 = 100	(4-1) / 100 = 0.03

Arc 1 has the least marginal cost. Therefore, as flow is increased from Node 3 to Node 4, flow will pass first through Arc 1. When the capacity of this arc is reached, flow begins to pass through Arc 2. Arc 3 will have non-zero flow if and only if Arc 2 is at its upper bound. Finally, Arc 4 will have non-zero flow only when Arcs 1, 2, and 3 are flowing full. Because the objective is to minimize cost, if two or more arcs are parallel, the one with the lowest unit cost is used first.

Quadratic Approximation. HEC-PRM also provides the analyst the option of specifying quadratic cost functions, or quadratic approximations to higher-order cost functions. For these functions, the analyst must specify two coefficients, c_1 and c_2, to define a cost function as follows:

$$f(x) = c_1 x + c_2 x^2$$

The solution procedure in this case involves implicit linear approximation of the quadratic function, which is done "behind the scenes". The analyst has only to specify desired solution accuracy, if the default value is not appropriate. Although an implicit piecewise linear approximation is used, restricted basis entry is not available for quadratic cost functions. Therefore, these functions must be convex.

Extrapolation of Penalty Functions. Quadratic penalty functions are automatically defined over the entire feasible range of flow and storage levels. If the analyst fails to define a linear or piecewise linear penalty function over the entire feasible range of flows or storage levels, HEC-PRM will extrapolate the function as shown in Figure 8.

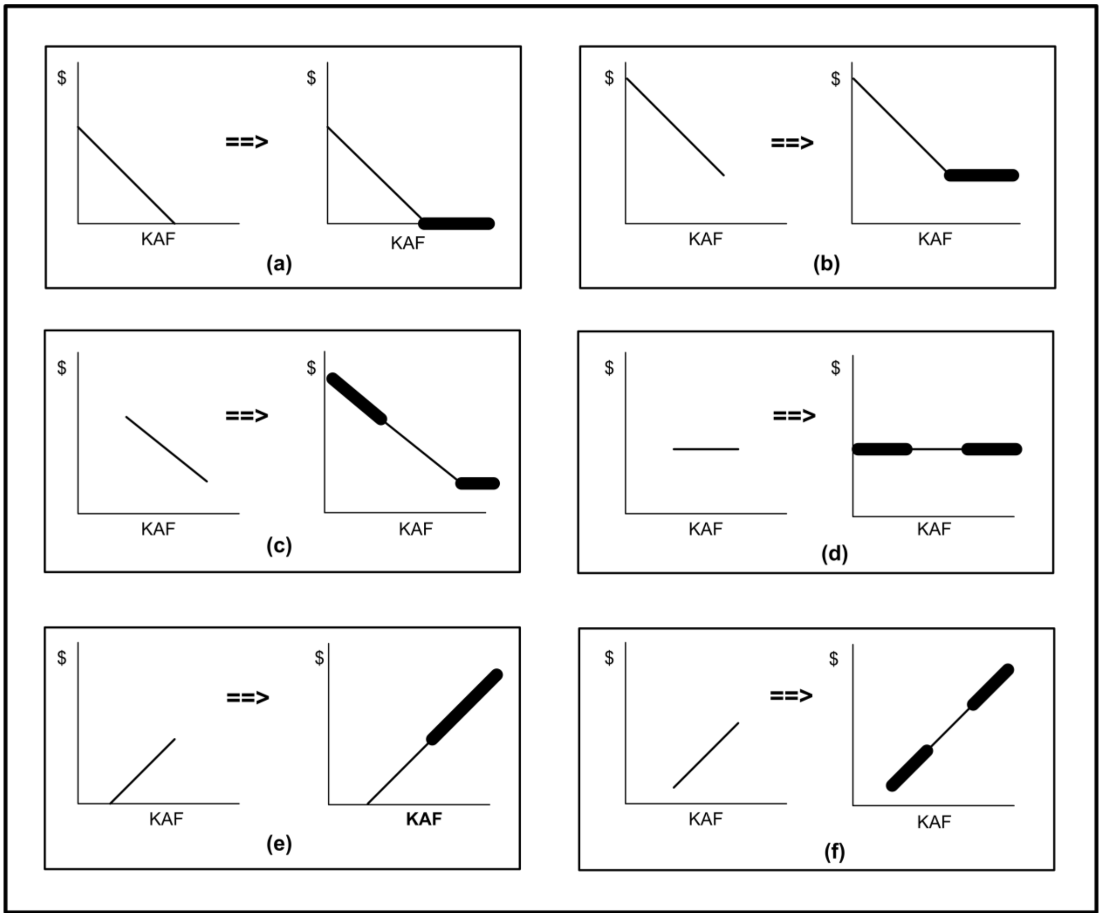

Figure 8. Extrapolation of Linear and Piecewise Linear Penalty Functions

Develop Objective Function Representing Desirable Operation

Penalty Functions. It is unlikely that all goals of system operation can be represented adequately with economic costs. There are often goals that are socially, environmentally, or politically motivated. Consequently, the objective function for the proposed model is formed from penalty functions, rather than cost functions. Although these penalty functions are in commensurate units, the units are not necessarily dollars. Instead, the penalty functions represent the relative economic, social, environmental, and political penalties associated with failure to meet operation goals. For example, even if failure to meet an environmental operation goal has no measurable economic cost, the penalty may be great.

Flow Penalty Functions. All operation goals related to reservoir-release, channel-flow, or diversion flow are expressed with flow penalty functions. These functions may represent operation goals for navigation, water supply, flood control, or environmental protection.

Figure 9 is an example of a flow penalty function. This function represents the relative penalty for diverting flow when the minimum desired diversion is 100 cfs. Less diversion is undesirable. More diversion is acceptable, but that water does not reduce further the penalty. The penalty function of Figure 9 is represented in the network by two parallel arcs. The characteristics of these arcs are shown in Table 5.

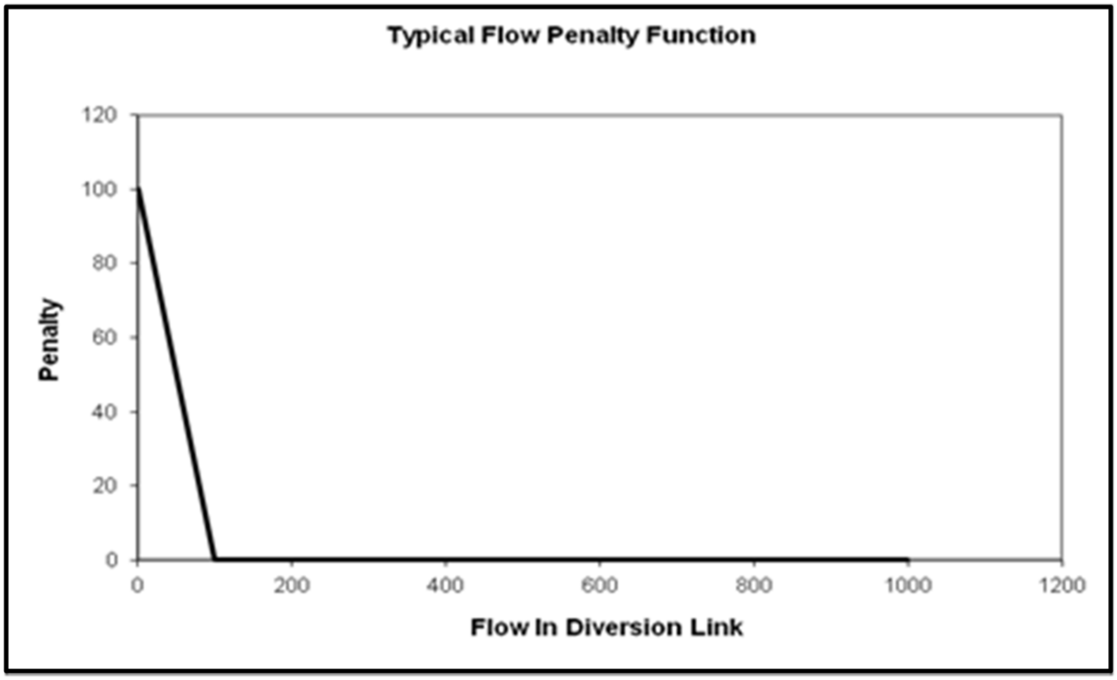

Figure 9. Typical Flow Penalty Function

Table 5. Typical Flow Penalty Function Arc Parameter

Arc Number (1)	Lower Bound (2)	Upper Bound (3)	Unit Cost (4)
1	0	100	(0-100)/100=-1.00
2	0	1000-100=900	0.00

The first arc represents flow up to the desired rate. As the flow increases from 0 cfs to 100 cfs, the total penalty decreases. At 100 cfs, the unit penalty is 0.0. As the flow increases beyond 100 cfs, the unit penalty remains 0.0. Similar flow penalty functions can be developed for reservoir release and channel flow.

Storage Penalty Functions. All reservoir operation goals uniquely related to storage are expressed through penalty functions for arcs that represent reservoir storage. These functions may represent operation goals for reservoir recreation, water supply, or flood control.

Figure 10 is an example of a reservoir storage penalty function. For this example, the top of the permanent pool is 200 kaf, the top of the conservation pool is 800 kaf, and the top of the flood-control pool is 1000 kaf. The function represents the penalty for storage when the reservoir operation goal is to keep the inactive and conservation pools full and the flood control pool empty.

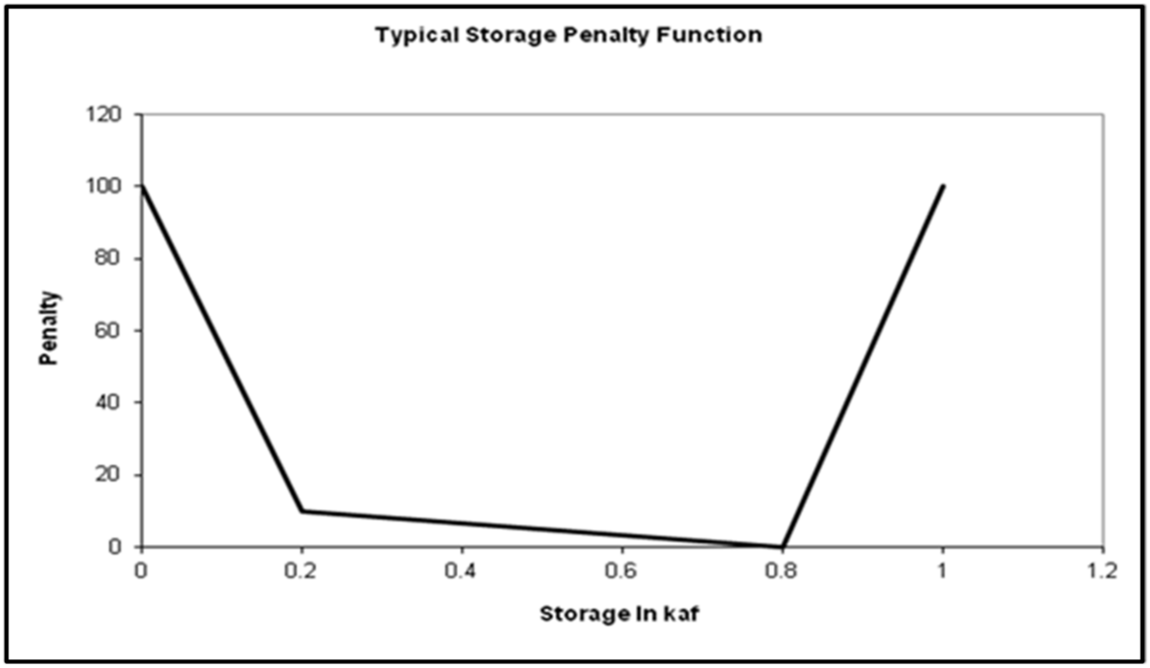

Figure 10. Typical Storage Penalty Function

The function shown in Figure 10 is represented in the network by three parallel arcs. The flow along one arc represents storage in the permanent pool. Increasing the flow along this arc reduces the penalty rapidly. Flow along the second arc represents storage in the conservation pool. Increasing flow along this arc also decreases the penalty, but not as rapidly as does flow along the inactive-pool arc. The third arc represents storage in the flood-control pool. Increasing flow along this arc increases the penalty. The solver will allocate flow to the arcs to minimize the total system penalty: first to the inactive-pool arc, then to the conservation-pool arc, and finally to the flood-control pool arc.

Storage and Flow Penalty Functions. Certain system operation goals depend on both storage and flow. The most significant is hydroelectric energy generated at a reservoir. This is a function of the product of release and head on the turbine. Head is the difference in reservoir-surface elevation and downstream water-surface elevation. Reservoir-surface elevation is a function of reservoir storage, and downstream water-surface elevation is a function of release. Thus, the energy generated is a complex function of storage and flow.

Figure 11 illustrates a typical hydroelectric energy penalty function. Here, penalty is measured in terms of reduction in value of the energy produced, when compared to the firm energy target. Additional energy generated has a value, but that value is less than firm energy. Thus the slope is less.

Other aspects of water resources systems that can be modeled using combined flow-storage penalty functions include navigation and groundwater supply. In the case of a lock and dam system, such as the Panama Canal, the amount of water needed for each lockage is a function of the difference in elevation from upstream to downstream which, in turn, is a function of storage in the canal or reservoir. In the case of groundwater supply, the cost of pumping is a function of the lift required, which is a function of storage in the aquifer. Penalty functions such as those shown in Figure 11 can be developed for these cases.

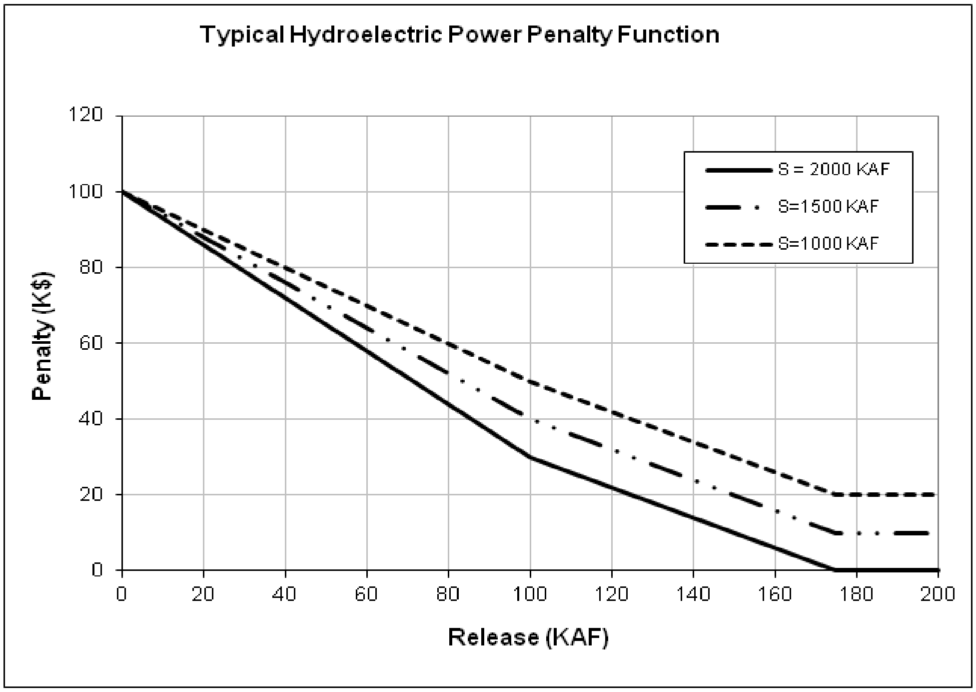

Figure 11. Typical Hydroelectric Power Penalty Function

Solve the Network Problem with an Off-the-shelf

Mathematical Statement of Problem. The optimization problem represented by the network with costs associated with flow can be written as follows (Jensen and Barnes, 1987):

Minimize: $$\sum_K^M C_K F_K \tag{1}$$

Subject to:

$$\sum_{k \in M_0} f_k - \sum_{k \in M_T} a_k f_k = 0 \quad \text{(for all nodes)} \tag{2}$$

$$l_k \leq f_k \leq u_k \quad \text{(for all arcs)} \tag{3}$$

in which:

m	=	total number of network arcs;
c_k	=	unit cost for flow along arc k;
f_k	=	flow along arc k;
M_0	=	the set of all arcs originating at a node;
M_T	=	the set of all arcs terminating at a node;
a_k	=	multiplier (gain) for arc k;
l_k	=	lower bound on flow along arc k;
u_k	=	upper bound on flow along arc k

Equations 1, 2, and 3 represent a special class of linear-programming (LP) problem: the *generalized minimum-cost network-flow problem*. Solution of the problem will yield an optimal allocation of flow within the system.

Network Simplex Method. The generalized minimum-cost network flow problem is solved using the primal network simplex method, as described in Jensen and Barnes (1987). This iterative procedure involves the following main steps:

1) Determine an initial basic feasible solution.
2) Compute the values of the dual variables for each node.
3) Compute the arc marginal costs for all non-basic arcs.
4) If all non-basic arc marginal costs satisfy the optimality conditions, stop; the optimal solution has been found.
5) Determine which arc will leave the basis, compute the new basic feasible solution, and return to Step 2.

Algorithm details can be found in Appendix C or in Jensen and Barnes (1987).

If all cost functions are convex, constant gain factors are used for reservoir evaporation, and the problem is not infeasible, the primal network simplex method will determine a globally optimal solution.

Network Simplex with Restricted Basis Entry. In the case of non-convex penalty functions, and/or non-constant gain factors for reservoir evaporation, a restricted basis entry procedure (Hadley 1964) is used to ensure that parallel arcs fill in the correct sequence. This procedure involves a number of minor modifications to the network simplex method. First, the initial solution must be feasible with respect to the sequenced arcs. A simple initial solution that satisfies this requirement has all 0 flows in parallel arc sets. If a non-zero initial flow is specified, the flows in the parallel arc set are arranged to satisfy the sequence restriction. Next, the method for selecting an arc to enter the basis is modified in two ways. If a parallel set of arcs includes a basic arc, no other arc in that set is allowed to enter the basis. If a parallel set of arcs does not include a basic arc, and one is found to violate the optimality conditions, the arc may enter the basis only if the sequence restriction will remain satisfied. Further details can be found in Appendix C.

If the problem is feasible, the network simplex method with restricted basis entry will find a local optimum that is feasible for the sequenced problem. However, a global optimum cannot be guaranteed. The analyst may find it useful to identify several local optima by starting with different initial values.

Successive Linear Programming. In the case of combined flow and storage penalties, such as those for hydroelectric power, the overall planning problem is being solved via iterative solution of linear approximations. Such successive linear programming techniques are described by Martin (1982), Grygier and Stedinger (1985), and Reznicek and Simonovic (1990). In summary, these techniques convert the combined release-storage penalty functions to release penalty functions by assuming a value of reservoir storage. Given the storage, head can be estimated. Given this head, the unit penalty for release is used, and the flow allocation problem is solved. Then the head assumption is checked, using the storage computed for the optimal allocation. If the assumption is not acceptable, the heads corresponding to the computed storages are used, and the process is repeated. Details are provided in Appendix C.

As with restricted basis entry, the successive linear programming method cannot guarantee a global optimum. Since many local optima may exist for large-scale problems, the analyst might find it useful to identify several local optima.

Post-process Network Results

The network solver finds the flow along each network arc that yields the total minimum-cost allocation for the entire network, subject to the continuity and capacity constraints. The total cost value is computed and reported in two ways: (1) the total cost in the network arcs and (2) the total penalty. The total cost in the network arcs is the objective function value computed by the solver, given by Equation (1). This value is derived from the unit cost of each arc, i.e., the slope of each penalty function. The total cost in the network arcs is usually a large negative number. To compute the total penalty, the "total penalty at zero flow" is first computed. For this calculation, the point at which the slope of a penalty function becomes non-negative is considered to be the zero penalty value. Thus, the total penalty at zero flow is a non-negative number, and it is usually a large positive number. The total penalty equals the total penalty at zero flow (a positive number) plus the cost in the network arcs (a negative number). Figure 12 illustrates these total cost values for a flow of sixty KAF/month.

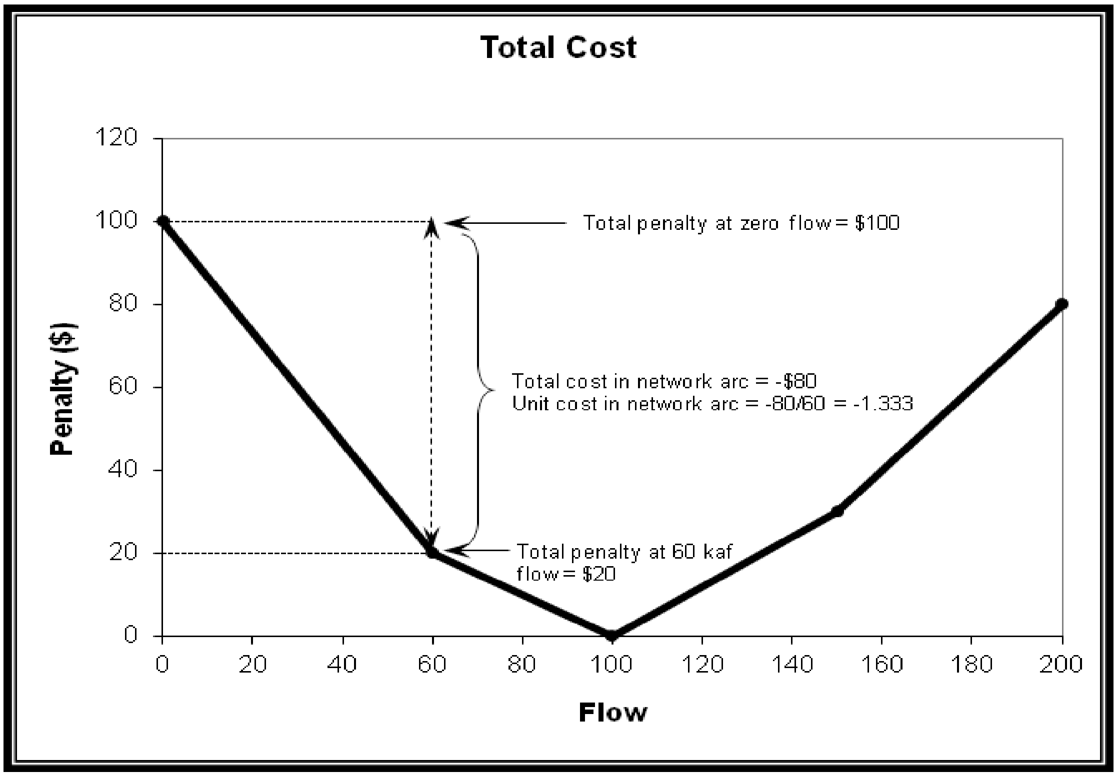

Figure 12. Illustration of Total Cost Calculations

The flows corresponding to the minimum cost solution must be translated into reservoir releases, hydropower generation, storage volumes, diversion rates, and channel flows to be useful to the reservoir system operators. For convenience, HEC-PRM results are translated and stored with the HEC-DSS. Then the results can be displayed or processed further as needed to provide information required for decision making.

Besides to the total cost, HEC-PRM reports other economic information related to the network solution. This information is termed "dual information" because the values are closely related to the optimal values for the dual variables. This dual information can be very useful for sensitivity analysis. For example, if the final solution reports that an arc has flow at its capacity, the *arc marginal cost* is the cost (or benefit) that would result from increasing the arc capacity by one unit. This has obvious ramifications for capacity expansion studies. As another example, the *dual cost* associated with a given node indicates the cost (or benefit) of obtaining an additional unit of water at that node. In the case of a fixed diversion occurring at a node, the dual cost would measure the system-wide economic impact of increasing that diversion by one unit. More details about the dual information, and how it is reported by HEC-PRM, can be found in Appendix C.

Model Building Software

To permit representation of any reservoir system as a network, the HEC-PRM software includes the following model-building components:

1) Inflow link;
2) Initial-storage link;
3) Diversion link;
4) Final-storage link;
5) Channel-flow link;
6) Simple reservoir-release link;
7) Hydropower reservoir-release link;
8) Reservoir-storage link;
9) Node

By selecting the appropriate links and the manner in which they are interconnected, the analyst can describe any system. By describing the characteristics of the links and the penalties associated with flow along the links, the analyst can define operating constraints and goals.

Inflow Link

An inflow link brings flow into the reservoir-system network. It originates at the source node and terminates at any other system node. In Figure 3, the link from Node 1 to Node 3 is an inflow link. It originates at the source node, Node 1, and carries flow into the system at Node 3.

The flow along the arc representing the inflow link is an input to the model. This known inflow may be an observed inflow from the historical record, or it may be an inflow from a sequence generated with a statistical model. To insure that the link carries the specified flow, the arc upper and lower bounds are equal and the unit penalty is zero.

Initial-storage Link

An initial-storage link is a special case of an inflow link. It originates at the source node and terminates at a node that represents a reservoir in the first period of analysis only. It introduces to the network the volume of water initially stored in the reservoir. In Figure 4, the storage link terminating at Node 3 in Period 1 is an initial-storage link; it represents the beginning-of-Period 1 storage.

As an initial-storage link carries a specified flow, no decision is represented by this link. To insure that the link carries the specified flow, the arc upper and lower bounds are equal and the unit penalty is zero.

Diversion Link

A diversion link originates at any system node and terminates at any node. In Figure 3, the arc from Node 4 to Node 2 is a diversion link. It originates in the system at the downstream control point, Node 4. It carries flow out of the system to the sink, Node 2. The flow along a diversion link is a decision variable, selected to minimize total system penalty. The diversion penalty function is specified by the analyst as penalty associated with deviating from the diversion desired. This function may vary by month. The software defines appropriate arc bounds and unit costs to represent the function.

The analyst may also specify inviolable minimum and/or maximum flow for a diversion link. If the analyst specifies both minimum and maximum, and if these values are the same, the diversion link will be represented in the network by a single arc. The upper and lower bounds of the arc are equal. In that case, the only feasible solution is one in which flow equals the specified value, regardless of cost. Any penalty function defined by the analyst for the link is ignored in that case, as it has no impact on the solution.

If the analyst specifies only a lower bound or only an upper bound, the software will impose the bound on the appropriate network arcs. If the penalty function is a simple function, like that of Figure 5, the bound is applied to the single arc representing that function. For example, if the analyst specified a lower bound of 25 cfs and an upper bound of 800 cfs, the network arc will have $l_k = 25$ and $u_k = 800$ (see Equation 3).

For more complex penalty functions, HEC-PRM determines the proper network arcs on which to impose the bound. For example, the penalty function of Figure10 is represented by two parallel arcs, each with a bound and unit cost. If the analyst specifies an inviolable lower bound of 25 cfs and an upper bound of 800 cfs, the network arcs must be adjusted to have parameters shown on Table 6.

Table 6. Diversion Link Arc Characteristics

Arc Number (1)	Lower Bound (2)	Upper Bound (3)	Unit Cost (4)
1	25	100	-1.00
2	0	800-100=700	0.00

For the first arc, the lower bound increases from 0 to 25. The upper bound remains 100. The unit cost does not change. For the second arc, the lower bound remains 0, and the upper bound now is 800 - 100 = 700. The unit cost does not change.

Final-storage Link

A final-storage link is a special case of a diversion link. It carries flow out of the system, but only from a reservoir in the last period of analysis. The final storage link thus originates at any system reservoir and terminates at the sink node. In Figure 4, the storage link originating at Node 3 in Period 3 is a final-storage link. The final-storage link is included in the system model to permit assignment of a future value for water in system reservoirs. Otherwise, the network solver will be indifferent regarding final storage. The solver may choose any storage state, including empty or full, without regard for future use.

Just as with the diversion link, the flow along a final-storage link is a decision variable, selected to minimize total system penalty. A penalty function may be specified by the analyst as the penalty associated with deviating from an ideal final storage, or else an inviolable (fixed) storage value can be specified.

Channel-flow Link

A channel-flow link originates at any non-reservoir node, terminates at any other network node, and represents the flow in a channel reach. The flow along the link is a decision variable, selected to minimize total system penalty.

As with the diversion link, the analyst may specify inviolable minimum and/or maximum flow for a channel-flow link. HEC-PRM will impose these constraints on the appropriate network arcs.

The analyst may specify also a multiplier for flow along a channel-flow link. The multiplier is a_k of Equation 2 for all arcs representing the link. If the multiplier is greater than 1.00, it represents increase of flow in the channel. If the multiplier is less than 1.00, it represents loss of flow.

Simple Reservoir-release Link

The reservoir-release link originates only at a non-hydropower reservoir node, terminates at any other node, and represents the total outflow from a reservoir. This includes release and spill. The flow along a reservoir-outflow link is a decision variable, selected to minimize total system penalty. In Figure 3, the link from Node 3 to Node 4 is a simple reservoir-release link. It originates at a node representing a reservoir and terminates, in this case, at a node representing a demand point.

The analyst may specify inviolable minimum and/or maximum flow constraints. The analyst may specify also a multiplier for flow along a reservoir-release link. HEC-PRM will apply the multiplier and impose the constraints on the appropriate network arcs.

Hydropower Reservoir-release Link

Link Description. A hydropower reservoir-release link (hydro-release link) originates only at a hydropower reservoir node, terminates at any other node, and represents the total outflow from the reservoir. This includes release and spill.

The flow along a hydro-release link is a decision variable, selected to minimize total system penalty. As hydroelectric energy is not a linear function of flow, however, determination of the release that minimizes total penalty requires consideration of storage.

Other Release Penalties. Due to the special nature of the hydro-release link, all other release-related penalties must be defined as a function of flow downstream. This is accomplished by defining a "dummy" node downstream of the hydropower reservoir. The hydro-release link connects the reservoir and this dummy node, and the hydropower penalty function is associated with this link. A channel-flow link connects the dummy node with the next downstream node. All penalty functions normally defined in terms of reservoir release are defined in terms of channel flow instead.

Reservoir-storage Link

Link Description. A reservoir-storage link originates at any reservoir node in a layered, multiple-period network. It represents the volume of water stored in the reservoir at the end of the period. The reservoir-storage link terminates at the node representing the same reservoir in the period following. The flow along a reservoir-storage link is a decision variable, selected to minimize total system penalty.

For example, in Figure 4, the arc from Node 3 in Period 1 to Node 3 in Period 2 is a reservoir-storage link. Flow along the arc leaving the Period 1 layer represents reservoir storage at the end of Period 1. Flow along the arc entering the Period 2 layer represents reservoir storage at the beginning of Period 2.

Evaporation Computation with Link Flow. To approximate reservoir evaporation, a fraction of flow entering the reservoir-storage link may be "lost". For the network model, the relationship of storage and evaporation is given by:

$$S_t = S_{t-1} - EVAP_{t-1} \tag{4}$$

in which:

S_t	=	reservoir storage at beginning of period t;
S_{t-1}	=	reservoir storage at end of period t-1;
$EVAP_{t-1}$	=	volume of reservoir evaporation

The evaporation volume is related to reservoir surface area with the following equation:

$$EVAP_{t-1} = (E_{t-1})\ (A_{t-1}) \tag{5}$$

in which:

E_{t-1}	=	evaporation rate in period t-1
A_{t-1}	=	reservoir surface area in period t-1

The quantity E_{t-1} is input to the model. It may be a historically observed evaporation rate, or it may be generated with a stochastic model. The relationship of surface area and storage can be approximated with a linear function as:

$$A_{t-1} = ß\ S_{t-1} \tag{6}$$

in which:

$ß$	=	a linear coefficient

The value of $ß$ is found from analysis of specified reservoir characteristics. Substituting Equations 5 and 6 into Equation 4 and simplifying yields:

$$S_t = (1 - E_{t-1}\ ß)\ (S_{t-1}) = a_k\ S_{t-1}. \tag{7}$$

The quantity $(1 - E_{t-1}\ ß)$ is the arc multiplier a_k of Eq. (2). The flow out of the reservoir-storage arc, S_t, is the flow into the arc, S_{t-1}, multiplied by $(1 - E_{t-1}\ ß)$. (**Note:** The quantity S_{t-1} is limited by the reservoir-storage arc capacity, but S_t is the value reported by HEC-PRM as the storage at the end of Period t-1. Thus, for positive evaporation rates, solution values for reservoir storage will always be less than the reservoir capacity.)

The value of $ß$ need not be a constant. If the analyst instructs HEC-PRM to read a reservoir storage-area relationship from the HEC-DSS paired data input file, and reservoir storage in each time period is represented by multiple arcs due to a piecewise linear penalty function, the software may compute and assign a different gain factor to each arc (depending on the characteristics of the storage-area relationship). Although this allows a more accurate representation of reservoir evaporation, it may also make the model non-convex, thereby requiring the use of the restricted basis entry solution procedure to ensure that the parallel arcs for reservoir storage fill in sequence.

Nodes

Nodes are included in the model to permit joining the appropriate links. Two or more of the links described may join at a node. The nodes represent system reservoirs, demand points, channel junctions, or diversion points. These may be existing facilities or proposed facilities. Additional nodes may be included in the network for convenience of description.

In addition to the analyst-defined nodes, HEC-PRM adds a source node and a sink node to satisfy the mathematical requirements for defining a network. All water entering the system flows from the source node, and all water leaving the system flows to the sink node.

Typical Penalty Functions

The goals of reservoir system operation are identified by the analyst via penalty functions. The functions define, as a function of flow, storage, or both, the economic, social, and environmental cost for deviating from ideal operation for each of the system operation purposes. These purposes include flood control, lake and stream recreation, water supply, environmental protection, hydroelectric energy generation, and navigation.

Flood-Control Penalty Function

A flood-control penalty function defines the cost of deviating from ideal flood-damage-reduction operation. This function typically will relate penalty to channel-link flow or reservoir release link flow.

Figure 13 is a typical flood-control penalty function. In this example, no penalty is incurred for flows less that 600 cfs, the channel capacity. Between 600 cfs and 1100 cfs, the penalty is slight, increasing to 100 units. The penalty is much greater for flowsexceeding 1100 cfs. This represents significant damage incurred as the flow moves out of the ten to twenty-five year floodplain and into surrounding property.

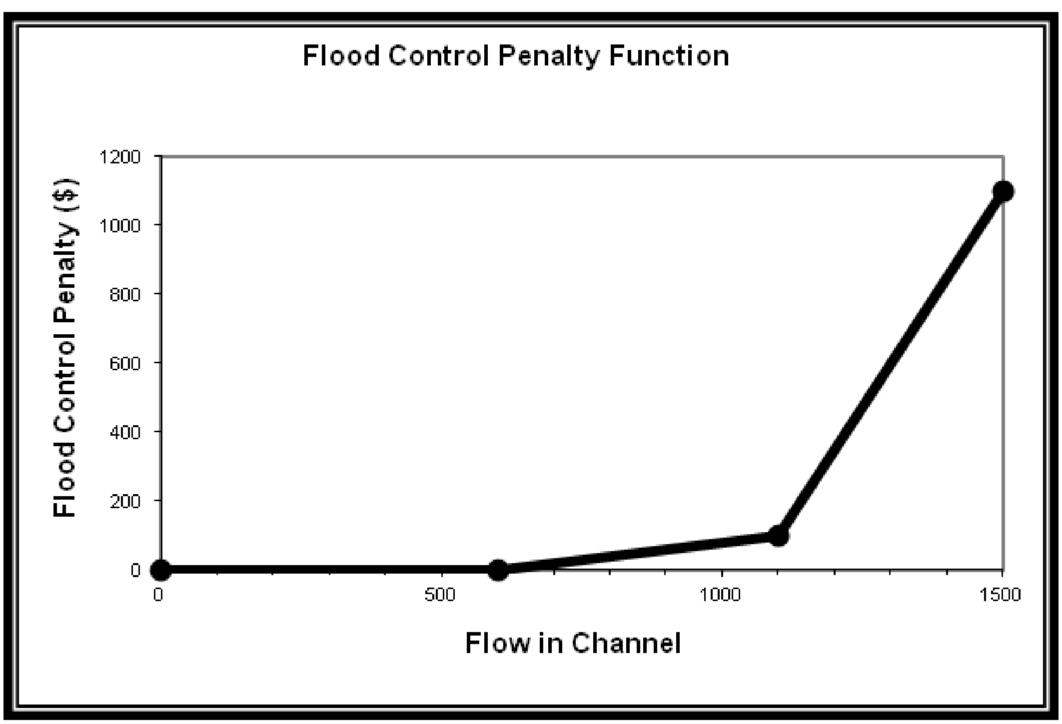

Figure 13. Typical Flood-Control Penalty Function

Recreation Penalty Functions

A recreation penalty functions may represent the relationship of recreation to reservoir storage or channel flow. Figure 14 is an example of a typical lake recreation penalty function. In this example, the desired range of active storage for recreation is 40 to 80 kaf. If the reservoir storage is less than 40 kaf, the boat ramps are inaccessible, and recreation is hazardous. If the reservoir storage is more than 80 kaf, the reservoir is in flood operation, and recreation is hazardous. Consequently, the function is shaped as shown.

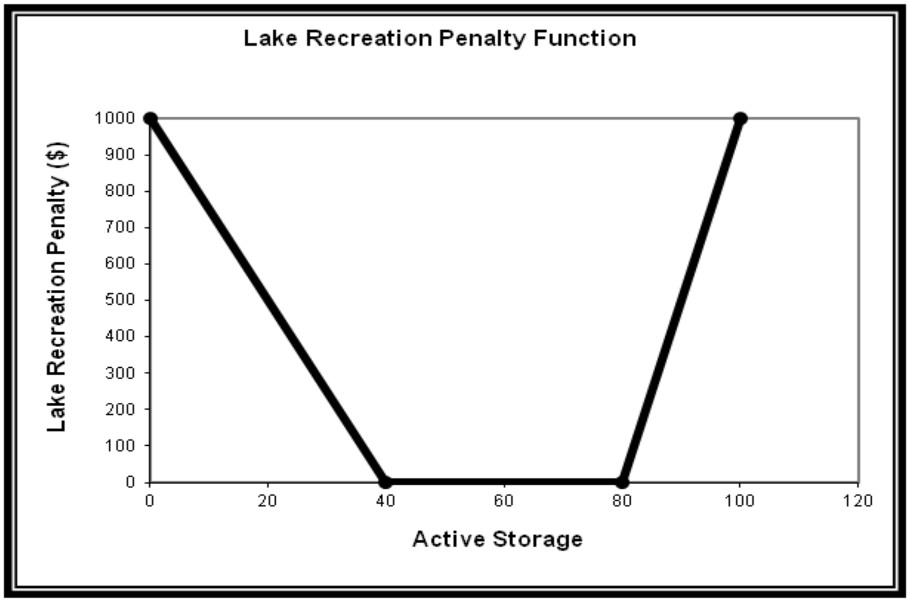

Figure 14. Typical Lake Recreation Penalty Function

Figure 15 is a typical river recreation penalty function. In this example, the desired range of flow for boating, swimming, and fishing is 400 to 500 cfs. If the flow rate is less than 400 cfs, boating and swimming are dangerous due to shallow depths and fishing is poor. If the flow rate exceeds 500 cfs, recreation is hazardous.

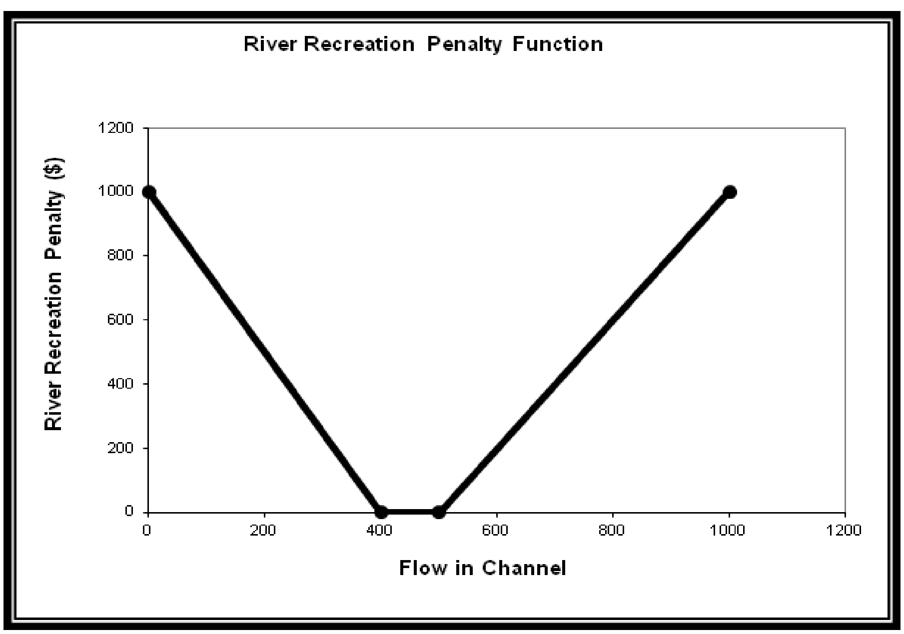

Figure 15. Typical River Recreation Penalty Function

Water Supply Penalty Function

A water supply penalty function describes desired operation for supply of water for municipal and industrial use or for irrigation. A water-supply penalty function may relate to channel-link flow, simple reservoir-release flow, or diversion flow. Figure 16 is a typical water-supply penalty function. In this function, the desired flow for water supply is 100 cfs. If the flow is less, demands are not met, so the penalty is great. If the flow exceeds the desired rate, the water is used, but the benefit is not great, as it is not dependable supply.

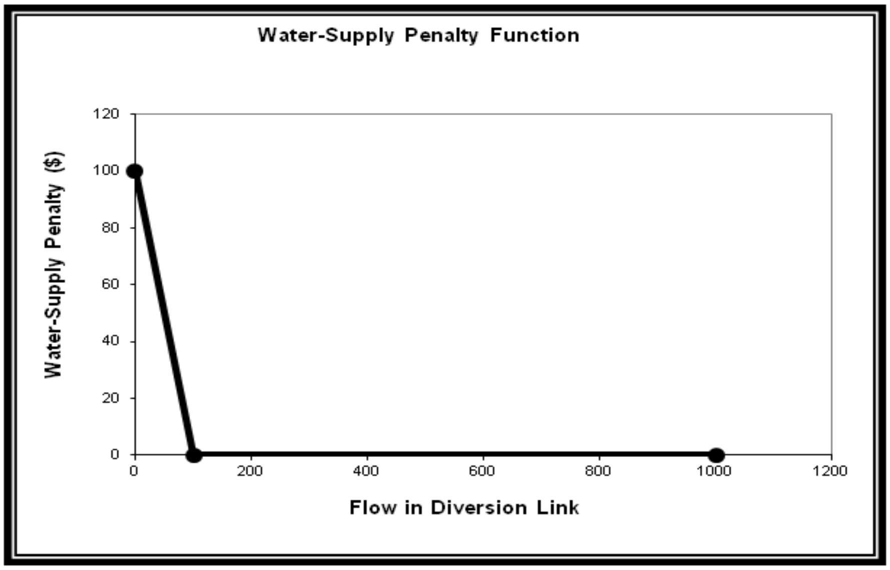

Figure 16. Typical Water Supply Penalty Function

Environmental Penalty Function

An environmental penalty function represents the desired operation for environmental protection. The function may define penalties for flow, storage, or both. A typical case is illustrated by Figure 17. In this example, an average monthly flow between 80 and 120 cfs is required to preserve wildlife habitat. If the flow is less or more, the habitat is severely impaired. Within the 80-120 cfs range, 100 cfs is considered to be ideal. In that case, only the ideal value is assigned zero penalty. For all other flows, the penalty is positive.

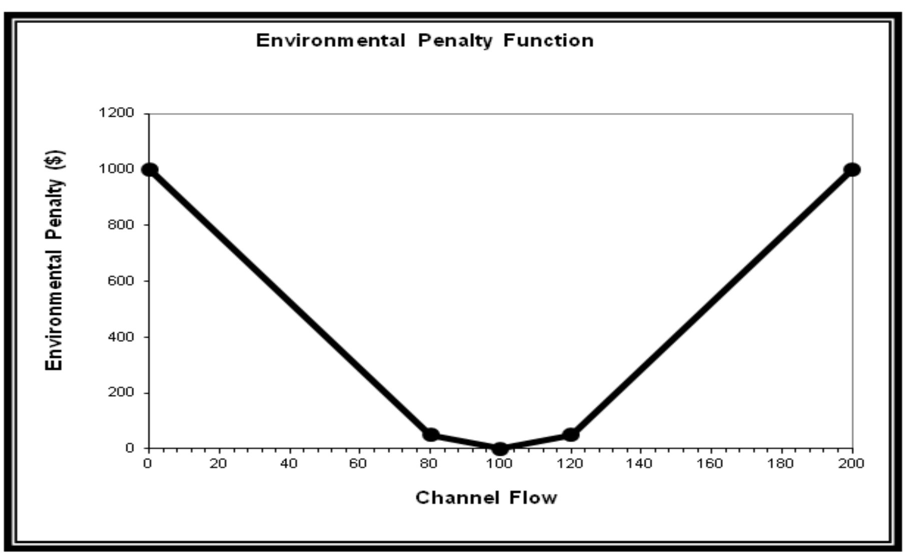

Figure 17. Typical Environmental Penalty Function

Hydroelectric Power Penalty Function

A hydroelectric power penalty function is assigned to a hydro-release link only and defines the cost of deviation from desired system operation for energy production. Figure 18 illustrates an acceptable form of the function. This function defines penalty as a function of release for a specified head (storage). If the head is less than the optimal head for the generator, or if the release is less than optimal for a specified head, the penalty is positive.

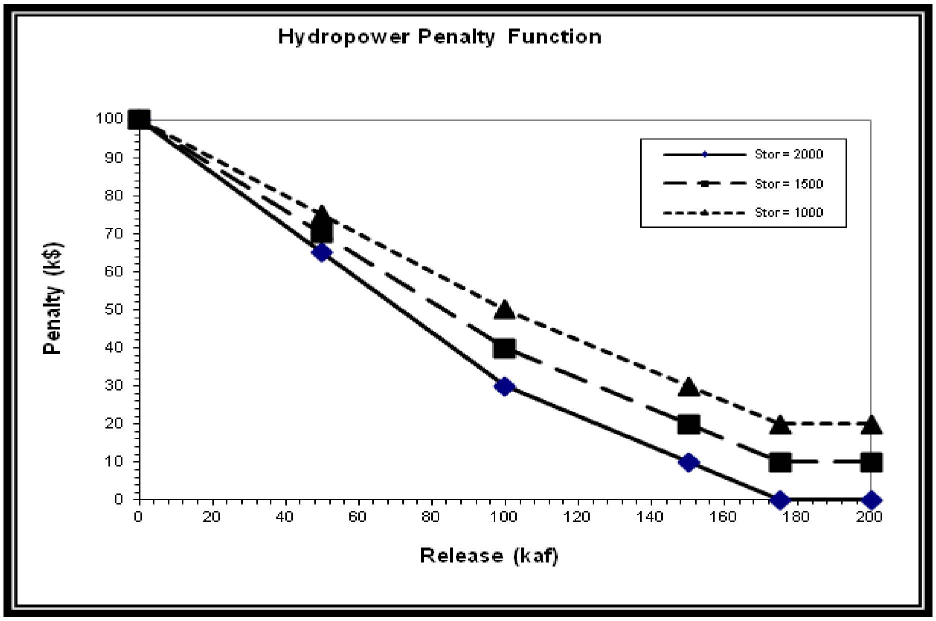

Figure 18. Typical Hydroelectric Power Penalty Function

Navigation Penalty Function

A navigation penalty function defines the cost of deviating from flows desired for vessel traffic in a system channel. Figure 19 is a typical navigation penalty function. In this example, the penalty is great for flows less than 400 cfs; this represents the minimum

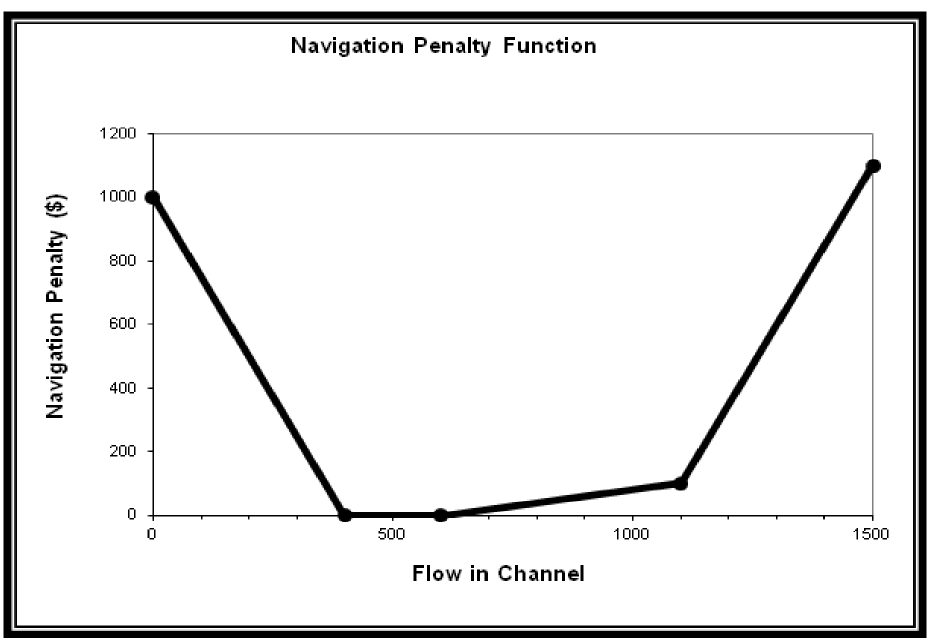

Figure 19. Typical Navigation Penalty Function

desired flow for towing barges in the channel. Between 400 and 600 cfs, the penalty is zero, as this is the desired flow for navigation. Between 600 and 1100 cfs, the penalty increases slightly, representing the increased effort required for navigation. Finally, the penalty increases rapidly if the flow exceeds 1100 cfs. This is the upper limit on desired flow for navigation.

In some cases, such as lock and dam systems, the navigation penalty may be a function of both release and storage. In this case, as with the typical hydroelectric power penalty function, a family of penalty functions would be defined - each representing the release-penalty relationship for a given storage level.

Combined Penalty Functions

If two or more penalty functions apply to a single stream reach or to a single reservoir, the functions are combined to yield a single penalty function. The combined (composite) penalty function then is used in the optimization. For example, a water supply penalty function, an instream recreation penalty function, and a flood damage penalty function may apply for a reservoir release. To combine the functions, the various penalties for a given flow are added. The composite function then is represented in a piecewise linear fashion for the network. Figure 20 illustrates this.

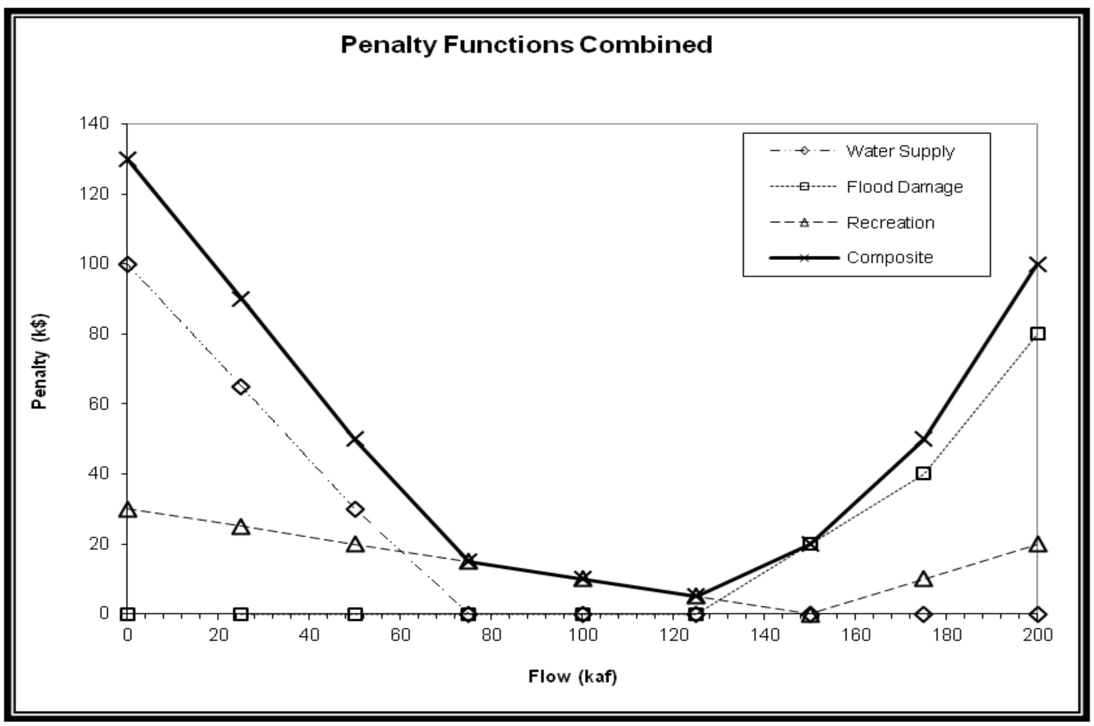

Figure 20. Penalty Functions Combined.

CHAPTER 3 - Use of MENUPRM

Running MENUPRM

To use the HEC-PRM Package on a hard disk system, a menu program may be used to assist the user in applying the different programs. To begin the menu program, enter:

MENUPRM

This should load the program and begin by displaying a banner page as described below. If the banner page does not appear, then the most likely cause of failure is that either one of the following files is missing:

[drive:]\HECEXE\MENUPRM.BAT or

[drive:]\HECEXE\PRMMENUX.EXE

General Menu Program Description

The menu program consists of a banner page, three primary menus and two subordinate menus. The structure of the menus can be illustrated in outline format as shown below:

- Banner page.
- Select Study.
- Enter or edit study name and associated data subdirectory.
- Select Program.
- Define Data Files.
- List files in current directory with mask.

To successfully execute a program, the user must progress through the primary menus. The subordinate menus are used only if the user wishes and are not required for an individual execution.

Function and Cursor Control Keys

Several keys are used to control menu selection, item selection within a menu, and operation selection (edit or list a file and execute a program). The following list summarizes the primary MENUPRM **action keys**.

Key(s)	Description
F2	Reset selection or definition. Example: in the "Define data files" menu, the F2 key resets all file names to those initially selected when this menu was last invoked.
F3	Delete the highlighted files.
F4	Edit the selected file with COED. The F4 key initiates editing the currently selected file with COED. This is operational only from the "Define data files" menu. The MENUPRM program invokes COED in a full screen mode with "Help Program" files, if applicable. You may obtain on-line documentation for COED by pressing the F1 function key while in COED and by following subsequent instructions. A complete COED user's manual is stored in the file "COED.DOC" (also available from HEC in printed form).

Key(s)	**Description**
F5	Define the background screen color. The color is changed by pressing the F5 key in the "Select Study" and "Select Program" menus. You may select from eight colors: black, blue, green, light blue, red, violet, orange, or white.
F6	Define the foreground screen color. The color is changed by repeatedly pressing the F6 key only in the "Select Study" and "Select Program" menus. You may select one of eight colors: black, blue, green, light blue, red, violet, orange, and white. The foreground color may be either normal or intense. To cycle through all possible definitions, you must press the F6 key sixteen times. Portions of the menu are displayed in reverse video. If you select an intense foreground color, sometimes you will get unexpected results. One example is using black as a foreground color. One combination that works well is a blue background and an intense white foreground.
F8	Execute the selected program. Pressing the F8 key initiates execution of the currently selected program (all necessary data files must have been defined). This is operational from the "Select Program" and "Define Data Files" menus.
F9	Return to previous menu. The F9 key allows you to exit the current selected menu and return to the previous menu. For example, if you are in the "Define data files" menu, you can go to the "Select Study" menu by pressing the F9 key twice - the first time you will access the "Select Program" menu and the second time the "Select Study" menu.
F10	Exit to DOS. By pressing the F10 key from any menu, you will immediately terminate the MENUPRM program and return to the DOS environment.
Esc	Reset the current selection or return to the previous menu. Pressing the Esc key resets the current file, study name, data directory, etc. to that previously defined. The Esc key changes only the current selection (e.g. the currently selected file) whereas the F2 key changes all selections (e.g. all defined data files) or resets the current menu selection. The Esc key returns to the previous menu if the user is not editing a file name.
Home	Select the first option, file, study, etc. The Home key controls the item selection on the current menu and page. For example, on the "Select Study" menu, pressing the Home key selects the first study displayed on the current page of studies (the study in the upper left corner of the menu).
End	Select the last option, file, study, etc. The End key controls the item selection on the current menu and page. For example, on the "Select Study" menu, pressing the End key selects the last study displayed on the current page of studies (the study in the lower right corner of the menu).
Home,**7**	Move cursor to first character. Pressing the Home key, releasing it, and then pressing the left arrow key moves the cursor to the first character in a field. This is operational when you are editing study names, filenames, and directory names.
Home,**6**	Move cursor to last character. Pressing the Home key, releasing it, and then pressing the right arrow key moves the cursor to the last character in a field. This is operational when you are editing study names, filenames, and directory names.
7	Move cursor one character to the left. Pressing the left arrow key moves the cursor one character to the left. It is operational when you are editing study names, filenames, and directory names.

Key(s)	Description
6	Move cursor one character to the right. Pressing the right arrow key moves the cursor one character to the right. It is operational when you are editing study names, filenames, and directory names.
8	Select previous study, file, or option. Pressing the up arrow changes the current selection to the one displayed just above the current selection. It is operational in all menus.
9	Select next study, file, or option. Pressing the down arrow changes the current selection to the one displayed just below the current selection. It is operational in all menus.
PgUp	Change to previous page. Pressing the PgUp key moves the display from the current page to a previous page of information. This is operational in the "Select Study" and "Directory Listing" menus and only if there are more studies (or files) than can fit on one page of display.
PgDn	Change to next page. Pressing the PgDn key moves the display from the current page to the next page of information. This is operational in the "Select Study" and "Directory Listing" menus and only if there are more studies (or files) than can fit on one page of display.
Ins	Change insert character mode. Pressing the Ins key toggles the insert character mode between on and off. It is operational when you are editing study names, filenames, and directory names. When the "insert character" mode is on, any character entered will be added to those existing. If the mode is off, any character entered will replace the existing character at the cursor location.
Del	Delete character. Pressing the Del key deletes the character at the current cursor position.
Alt-D	Delete the selected disk file. Pressing the Alt key and holding it down while pressing the "D" key deletes the currently selected file from the disk.
Alt-E	Edit the selected disk file with COED. Pressing the Alt key and holding it down while pressing the "E" key initiates editing the currently selected disk file with COED. This is operational only from the "Define data files" menu. The MENUPRM program invokes COED in a full screen mode and with "Help Program" files, if applicable. You may obtain on-line documentation for COED by pressing the F1 function key while in COED and by following subsequent instructions. A complete COED user's manual is stored in the file "COED.DOC". It may be copied to a printer. Alternatively, a pre-printed manual is available from HEC.
Alt-F1	While in COED, you may get information from the "Help Program" files by pressing Alt-F1. This information should be available for the HEC-PRM input data file which requires a fixed format. You access the help program feature by entering a valid two character record identifier (e.g. "J1") in columns one and two of an input data file and then pressing Alt-F1. You can get descriptions for each input field of the input data by positioning the cursor at the desired field using the tab key and then by pressing the Alt-F1 key. COED then displays the description for that field. It is similar or identical to the description contained in the program user's manual. The COED user's manual contains a more detailed description of the help program.
Alt-L	List the currently selected disk file with "LIST". Pressing the Alt key and holding it down while pressing the "L" key initiates listing the currently selected disk file. The currently selected disk file is indicated by the highlighted box. This is operational only from the "Define data files" menu. Documentation for the LIST program is located on the HEC-PRM Installation diskette labeled "Install PRM" in the file LIST.DOC. If you try to list an

Key(s)	Description
	HECDSS file, the menu will attempt to list its associated catalog file. There is no protection against listing a binary data file.
Alt-P	Print the currently selected disk file using the MS-DOS PRINT command (or with "PROUT" if printing HEC-1, HEC-2, or HEC-5 output). Pressing the Alt key and holding it down while pressing the "P" key initiates printing the currently selected disk file. The currently selected disk file is indicated by the highlighted box. This is operational only from the "Define data files" menu. There is no documentation for the PROUT program. It merely converts the mainframe carriage control character into a code recognized by PC printers to assure proper pagination.
Alt-X	Execute the selected program. Pressing the Alt key and holding it down while pressing the "X" key initiates execution of the currently selected program (all data files must have been defined). This is operational from the "Select Program" and "Define Data Files" menus.

MENUPRM Associated Disk Files

There are several important disk files associated with the menu program. They are described as follows:

File	Description
PRMMENU.DFT	Contains the last selected screen colors, output device, and study. After the first execution of the menu program, this file should always exist in the subdirectory [drive:]\HECEXE\SUP.
PRMMENU.SDY	Contains a cross-reference of study names and associated data subdirectories where all data for each study is stored. This file should be created if you install some of the test data when the Installation Program is run. After the first execution of the menu program, this file should always exist in the subdirectory [drive:]\HECEXE\SUP.
PRMMENU.FIL	Contains a listing of last selected file names, colors, devices, etc. for each study. After the first execution of the menu program for a given study, a file should exist in the subdirectory which contains data for that study. In other words, for each study, there should be a file named "PRMMENU.FIL". If there are ten studies, there should be ten files named "PRMMENU.FIL" located in appropriate subdirectories. For example, if data for the study "Silver Creek" is stored in the subdirectory "D:\DATA\HECPRM\TD21DATA", then the file "D:\DATA\HECPRM\TD21DATA\PRMMENU.FIL" will be created when that study is selected and files are edited.

Description of Menus in MENUPRM

Banner

When executing the menu program, the "Banner" screen is the first information displayed. It gives HEC's address and phone number, the version date of the menu program, and the banner indicating that you are executing the MENUPRM program. This page will disappear after five to ten seconds. You can proceed to the next menu sooner by pressing any ASCII key (such as the <Enter> key or the <space bar>. The banner menu should appear as shown in Figure 21.

Figure 21. MENUPRM Banner Screen

Select Study

The "Select Study" menu allows you to select the study (or set of data) that you wish to analyze. You select a study by maneuvering the highlighted box over the desired study and pressing the <Enter> key. To enter a new study, position the highlighted box over the line "(specify new study)" and press the <Enter> key or begin entering a new study name. To edit an existing study, position the highlighted box over the study and press "Alt-E". The study menu should appear as shown in Figure 22.

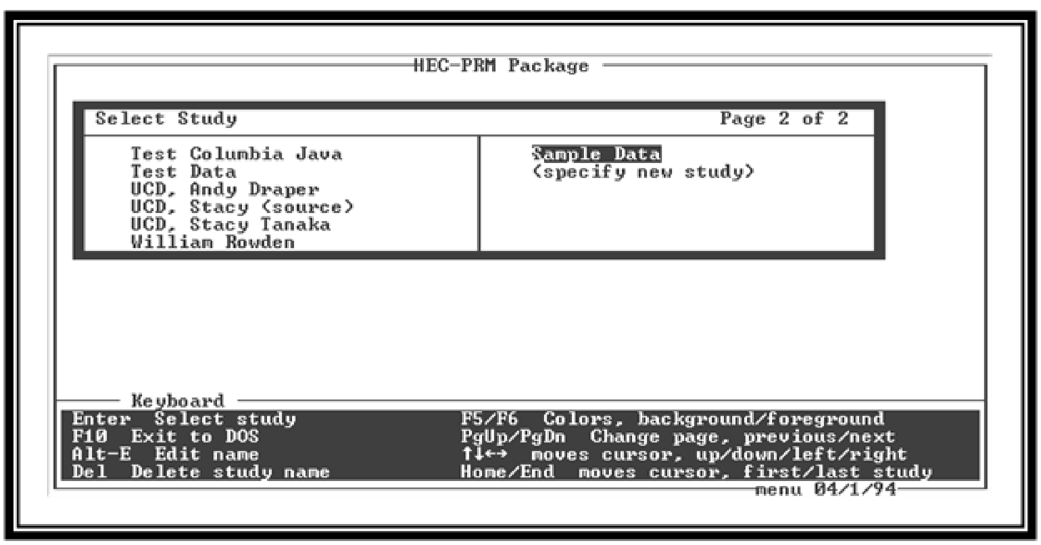

Figure 22. Select Study Menu

Edit Study

For each study, there is an associated subdirectory where the data is stored. Any time you enter a new study (or edit an existing study reference by entering "Alt-E"), you may edit the subdirectory into which the data will be entered. **If data already exists in a subdirectory and you change the subdirectory, the data will not be moved to the new subdirectory**. The subdirectory name is simply the location that the menu program searches to find data files for a given study. If you enter a new study or edit an existing study, the "Edit study name / data directory" menu should appear as shown in Figure 23.

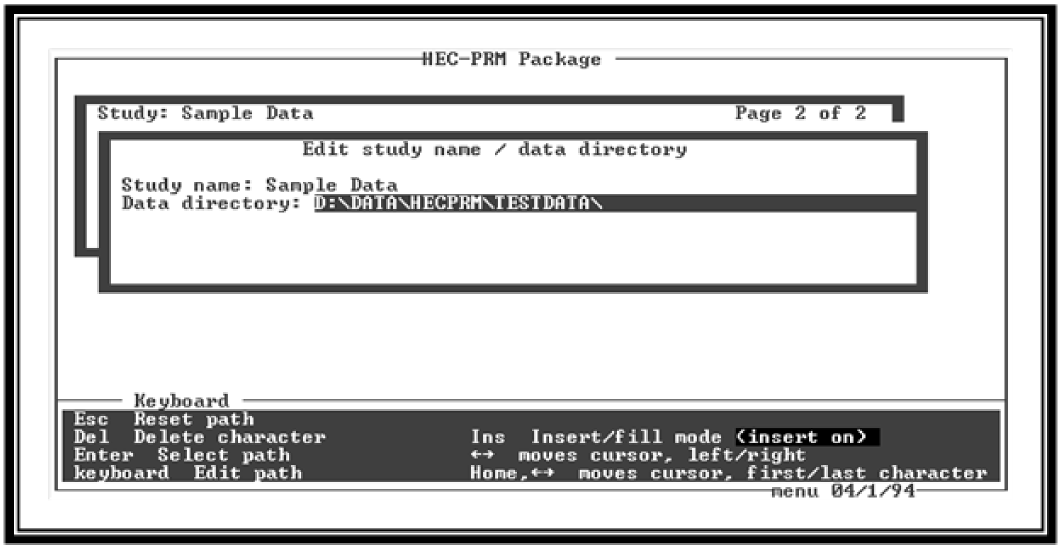

Figure 23. Edit Study Menu

Select Program

The "Select Program" menu allows you to indicate the program that you wish to execute or the program for which you wish to enter or edit data. You may select the program for which you wish to enter or edit data by either entering the appropriate integer (e.g., pressing the "1" key for the HEC-PRM program) or by maneuvering the highlighted box over the desired program and pressing the <Enter> key as shown in Figure 24. If you have already defined the appropriate data files for the desired program, you may execute the program from this menu by pressing and holding the "Alt" key and then pressing the "X" key (or alternatively the F8 function key).

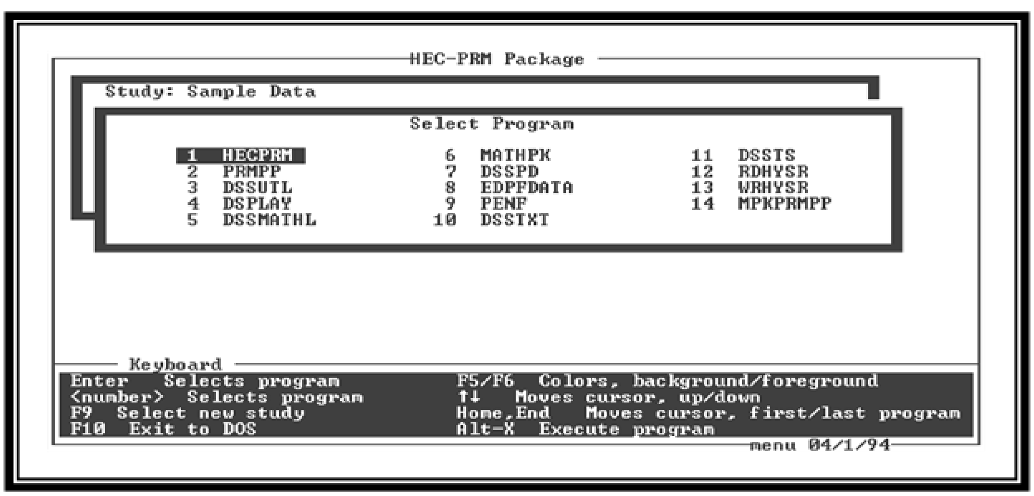

Figure 24. Select Computer Program

Define Data Files

The "Define data file" menu (Figure 25) allows you to enter the file names which contain the input data or output results for each program. To enter or edit a file name, position the highlighted box over the desired filename and then type the filename. If a filename has already been entered but you do not wish to use a file, enter the characters "NONE." (A period follows the characters "NONE") and then press the <Enter> key. The menu program will display the character string "(none)" and will not assign any file when the selected

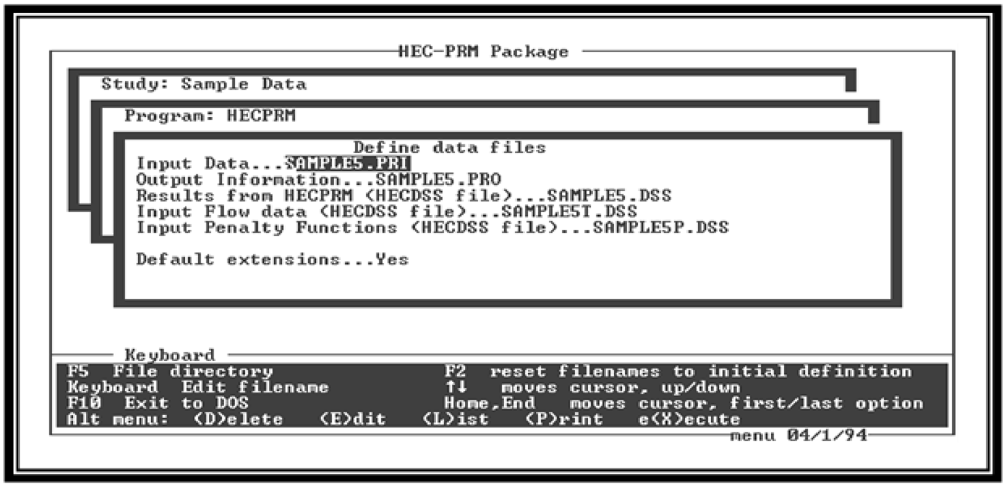

Figure 25. Define Data Files Menu

program is executed. (That FORTRAN unit is actually assigned to the default scratch file internal to each program.) The menu program assumes certain default filename extensions as described later in the "Default Data File Extensions" section. You may override these defaults by entering a period (".") followed by your desired extension. The menu program will use your defined extension for future definitions. This selection is stored in the file "PRMMENU.FIL" which is associated with the selected study. In addition to defining the data file names; you may edit, list, delete, and print files or execute the currently selected program from this menu. The following codes invoke these operations. You must press and hold down the "Alt" key and then press the appropriate key (such as the "L" key) for each operation.

Alt Key	Function Key	Operation
Alt-D	F3	Delete currently highlighted file.
Alt-E	F4	Edit currently highlighted file using COED.
Alt-L	F6	List currently highlighted file using the “LIST” program.
Alt-P		Print currently highlighted file using the “PROUT” program.
Alt-X	F8	Execute the currently selected program using the selected files displayed on the screen.

Data Directory List

You may also define a data file by obtaining a file directory listing and selecting a file from this list. Figure 26 illustrates the Data Directory List.

To get a file directory listing, you may do any one of the following:

File	Description
F5	Pressing the F5 function key lists all files which have the current default extension. For example, the HEC-PRM data input files have the default extension ".PRI". If the "F5" key is pressed, the menu program initiates a "DIR *.PRI" DOS command, then sorts all of the files that meet that criteria, and displays them to the screen. The user may select a file by positioning the highlighted box over a file and pressing the <Enter> key. If the desired file does not exist, press the "Esc" key to return to the previous menu.

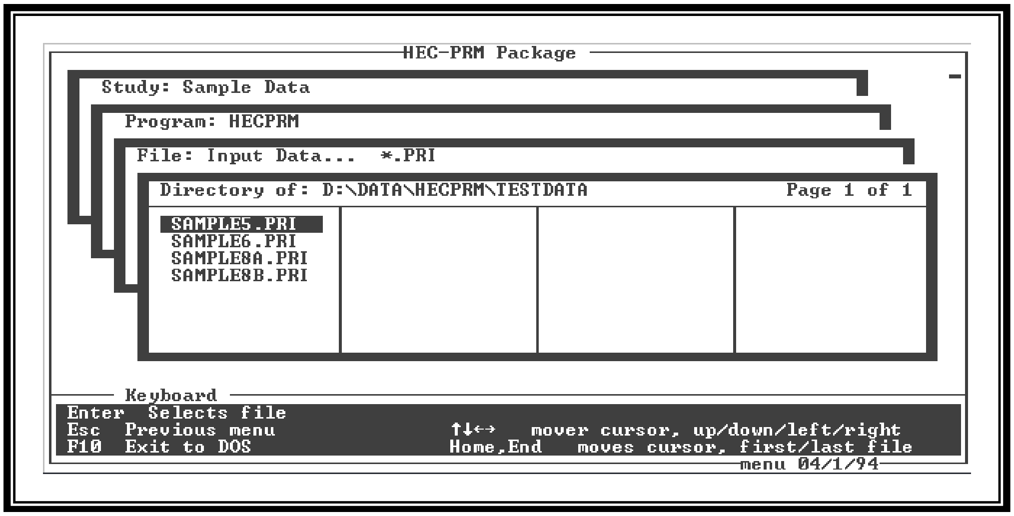

Figure 26. Data Directory List

. Entering the characters "*.*" lists all files (up to a maximum of 300 files) which are in the current default directory. You may select the desired file by positioning the highlighted box over the desired file and pressing the <Enter> key.

SLV01??.E Entering "SLV01??.E" lists any file meeting the user specified file mask "SLV01??.E" (or any other file mask). You may select the desired file by positioning the highlighted box over the desired file and pressing the <Enter> key.

Default Data File Extensions

The menu program assumes certain file extensions for each file (including no extension in some cases). The assumed default extension will be used unless you override it by changing the "Default extensions" option to "NO", entering the filename, followed by a period (".") and up to three characters. If no characters are entered after the period, then no extension is used. For example, the HEC-PRM input data files use the default extension ".PRI", while the HEC-PRM output data files use the default extension ".PRO", and the HECDSS data files use the extension ".DSS". The default extensions are displayed by using the filename "EXAMPLE" or some valid file name and showing the resulting menu for each program.

CHAPTER 4 - Input Data Records

Comment Records (..**)

Columns	Variable	Value	Description
1-2	CINREC	..	Comment Record. Columns 3-n are printed in output. This record may be entered anywhere within the input data stream. There is no limit to the number of comment records.
3-80	CINREC	character	Comments.

Columns	Variable	Value	Description
1-2	CINREC	**	Comment Record. Columns 3-n are printed in output. This record may be entered anywhere within the input data stream. There is no limit to the number of comment records.
3-80	CINREC	character	Comments.

Columns	Variable	Value	Description
1-2	CINREC	*(blank)*	Comment Record. Columns 3-n are printed in output. This record may be entered anywhere within the input data stream. There is no limit to the number of comment records.
3-80	CINREC	character	Comments.

Global Data and General Computation Controls

IDENT – Source and Sink Identifiers

Columns	Variable	Value	Description
1-5	CINREC	IDENT	This record contains the code names for the Super Source and Super Sink nodes. It is an optional record. If you wish to use the default code names of "S_SOURCE" and "S_SINK", you need not enter this record. If entered, it must precede all other records which reference these nodes such as the "NODE " and "LINK" records.
11-20	IDNODE(1)	character	The character identification for the Super Source node. Default is "S_SOURCE". (Maximum 10 characters)
21-30	IDNODE(2)	character	The character identification for the Super Sink node. Default is "S_SINK". (Maximum 10 characters)

CYCLEYEARLY– Yearly Step-wise Operation

This record sets a flag that instructs HEC-PRM break up the time window into multiple runs of one year of operation. This eliminates the infinite foresight associated with the normal HEC-PRM operation. For this option to work, the user can set the starting storage for the first year on the NODE record as is normally done. For subsequent years, HEC-PRM will retrieve the starting storage from the output results DSS file. It will be the computed ending storage from the previous year's simulation. The input time window must be an integral of twelve time periods. For example, 31Oct1961 to 30Sep1970 is valid but 31Oct1961 to 31Oct1970 is invalid.

Penalty function names are modified based upon data entered on the CONDITION record and the condition ID that is stored in the input time series DSS data file and retrieved using the CONDTS record. HEC-PRM will modify part F of the penalty function pathname using the appropriate condition name.

Columns	Variable	Value	Description
1-11	CINREC	CYCLEYEARLY	This record sets a flag that causes HEC-PRM to break up the user's time window into multiple runs of one year each for the user's time window.

TIME – Time Window for Simulation

Columns	Variable	Value	Description
1-4	CINREC	TIME	This record contains the time window for which calculations are performed. It is a required record.
11-80	CINREC	character	Contains the time window in free-format. There must be a starting date and ending date. The starting date must be entered first followed by the ending date. The date consists of the 3 character month and the integer year without any space separating them. The day is not entered as part of the time window since all computations are done on a monthly time interval. The year may be either 2 or 4 digit year (the 2 digit year is assumed to be within the years 1900-1999). An example entry is: TI JAN1930 DEC1940 Internally, the computation time window is stored in variables: YRWIN(2) Year MONWIN(2) Month DAYWIN(2) Day

J1 – Print Level and Factors

Columns	Variable	Value	Description
1-2	CINREC	J1	This record contains specifications for this simulation. The specifications include network solver constants, function factors, and data units.
3-10	EPS	+ real no.	EPS is a small positive number which is used by the Jensen-Barnes "Network with Gains" Solver. HEC-PRM uses EPS to test for iteration convergence and to determine the least cost flow path. (Default is 1.0E-05; Must be a positive number).
11-20	--	--	Not used.
21-30	FACFLO	blank, + real no.	FACFLO scales all incremental inflows for calculations by the network solver. HEC-PRM multiplies all inflows by FACFLO before calling the solver and then divides the computed flows by FACFLO after the solution and before storing results in the output DSS data file. (Default is 1.0; Must be a positive number).
31-40	FACCST	blank, + real no.	FACCST scales all unit costs which are derived from the penalty functions. The scaled costs are used by the network solver. HEC-PRM multiplies all unit costs by FACCST before calling the solver and then divides the unit costs by FACCST after the solution. (Default is 1.0; Must be a positive number).
41-50	IPRINT	blank, + integer no.	IPRINT controls the amount of output from the network flow solver. Normally, you should select IPRINT=1 or leave blank for default. If IPRINT=0, the solver will only do some calculations but it will not solve the problem. If IPRINT is greater than 1, the solver will generate a lot of output. (Default is 1).DSS_PRNT_LVL controls the amount of output written

Columns	Variable	Value	Description
51-60	DSS_PRNT_LVL	blank, + integer	from the HEC-DSS software to the HEC-PRM output file (*.PRO) . A higher number requests more output. Normally, you should select DSS_PRNT_LVL=3. DSS_PRNT_LVL=3 will include output associated with all data written to a DSS data file. DSS_PRNT_LVL=4 will include output associated with all data read from a DSS data file. (Default is 4).
61-70	DSS_DOUBLE	integer	DSS_DOUBLE controls the precision of the computed that are stored in the DSS file. Solver results are double precision numbers and can be stored as hydrographs of double precision numbers. This is helpful when a previous solution is provided as the initial basis for the initial solution of a subsequent simulation. However, older DSS software cannot access double precision data.

DSS_DOUBLE	Description
0	Store computed results in the output DSS data file as single precision numbers.
+	Store computed results in the output DSS data file as double precision numbers.

J2 – Hydropower Convergence Criteria

Columns	Variable	Value	Description
1-2	CINREC	J2	This record contains specifications related to the hydropower operation.
3-10	ALPHA_HPE	+ real no.	The starting range factor for hydropower iterations. This factor will be reduced during the hydropower iterations. It must be less than 1.0. (Default is 0.5).
11-20	ALPHA_HPE_MIN	+ real no.	The minimum value of ALPHA_HPE for convergence criteria. When ALPHA_HPE is less than ALPHA_HPE_MIN, the hydropower iterative scheme has converged. (Default is .01).
21-30	QTOLR_HPE	+ real no.	The flow tolerance value for determining if the flow is close to zero. If the computed flow is less than QTOLR_HPE, HEC-PRM will use the upper bound rather than computed flow to determine the limiting flow range.

J3 – Solver Parameters for Extreme Values

Columns	Variable	Value	Description
1-2	CINREC	J3	This record contains specifications related to the solver parameters.
11-20	EPSC	+ real no.	A small number for comparing costs to zero (defaults to 1e-9 or 1 × 10□9).
21-30	EPSF	+ real no.	A small number for comparing flows to zero (defaults to 1e-9 or 1×10^{-9}).
31-40	BIG	+ real no.	A large number (defaults to 1d9 or 1×10^{9}).

J4 – Quadratic Computation Controls

Columns	Variable	Value	Description
1-2	CINREC	J4	This record contains parameter values for the quadratic cost function option.
3-10	B_STEP	+ real no.	The beginning step size for the quadratic solution procedure. If B_STEP = 0, the default value of 1000 is used.
11-20	E_STEP	+ real no.	The ending step size for the quadratic solution procedure. A smaller value of E_STEP leads to greater precision but longer solution time. Must be smaller than B_STEP. If B_STEP = 0, the default value of 1.0 is used.

JJ – Solver Algorithm Control Parameters

Columns	Variable	Value	Description
1-2	CINREC	JJ	This record contains parameter values related to solution algorithm options.
3-10	WFACT	+ real no. (0-1)	Controls the weight (scaling factor) assigned to the original objective coefficients during Phase 1, and also adjusts the original arc costs in Phase 2. In Phase 1, a value of WFACT greater than zero will encourage the solver to move towards an optimal solution in Phase 1, rather than simply a feasible solution. The Phase 1 scaling operation is not performed if WFACT = 0. In Phase 2, a value of WFACT greater than zero causes the original arc costs to be normalized (to some degree) by the minimum non-zero arc cost. If WFACT = 1, all costs are normalized such that the minimum non-zero arc cost equals 1.0. (Default is 0).
11-20	UFACT	+ real no. (0-1)	Controls the scaling of arc bounds and flows, which can influence the numerical stability of the solution algorithm. If UFACT > 0, all arc bounds and flows are multiplied y the quantity CAPF, where CAPF = UFACT/UMIN. UMIN is the smallest non-zero upper bound (after all lower bounds have been adjusted to 0). The scaling operation is not used if UFACT = 0. (Default is 0).
21-30	PFACT	+ real no. (0-1)	Controls use of a candidate list for selecting an arc to enter the basis. Set to 0 to disable the candidate list and select the first qualifying arc. Set to 1 to use the candidate list instead of the first select procedure. Set to intermediate values to switch between the two procedures. The candidate list is most effective for dense networks, where there are relatively few arcs for the number of nodes. Piecewise linear functions lead to many arcs connecting nodes, which increases the sparsity of the network. Therefore, the candidate list will likely be ineffective for most HEC models. (Default is 0).
31-40	IALTER	+ integer no.	Specifies type of selection procedure used to find an arc to enter the basis. If IALTER = 0, the arcs will be searched in numerical order. If IALTER = 1, the alternative procedure is used to check arcs adjacent to the leaving arc first. IALTER = 1 should lead to faster solution of problems with many piecewise linear cost functions. The alternative selection procedure is disabled when restricted basis entry is used (SEQ_SOLV = 1). (Default is IALTER = 0).
41-50	SEQ_SOLV	+ integer no.	Specifies whether or not restricted basis entry is used to ensure parallel arcs fill in order, regardless of their gains or unit costs. SEQ_SOLV > 0 turns on restricted basis entry. (Default is 0).

PR – Print Parameters

Columns	Variable	Value	Description
1-5	CINREC	PR	This record allows the user to get specific output from HEC-PRM in the ASCII output file. The following code names are entered in free-format. If the name is entered, the output is activated. See the NOPRINT record for suppressing output.
6-80		INP_ECHO	Echo user input records when HEC-PRM reads them. (Default is on).
		USR_NODES	Lists all user nodes. (Default is on).
		USR_LINKS	Lists all user links. (Default is on).
		USR_PATHS	Lists all pathnames generated from user input pathname parts. (Default is off).
		ORIG_ARCS	Lists all solver arcs before the solver is called. (Default is off).
		SOLN_ARCS	Lists all solver arcs after a solution is determined. (Default is off).
		SOLN_NODES	Lists all solver nodes after solution is determined (Default is off).
		FIXED_SPLY	Lists all fixed supply flows to the nodes. (Default is off).

Columns	Variable	Value	Description
		ZERO_FLOW_PNLTY	Lists the calculation results from determining the penalty associated with zero flow and storage in all arcs. (Default is off).
		DUAL_COST	Lists dual costs for each node and arc. The dual cost for a node is the cost of routing an additional unit of flow to that node. The dual cost for an arc, with flow at its capacity, is the amount the objective function would change if the arc capacity were increased by one unit. (Default is off).
		ORIG_NODES	Lists all of the original nodes before calling the solver. (Default is off).
		SOLVER_DATA	Writes solver input to a text file that could be read by a stand-alone version of the network flow solver. It is written to the file "prm_input_solver.txt". (Default is off).

NP – Suppress Print Parameters

Columns	Variable	Value	Description
1-5	NP	CINREC	This record allows the user to suppress specific output from HEC-PRM in the ASCII output file. The following code names are entered in free-format. If the name is entered, the output is suppressed.
		INP_ECHO	Echo user input records when HEC-PRM reads them. (Default in on).
		USR_NODES	Lists all user nodes. (Default is on).
		USR_LINKS	Lists all user links. (Default is on).
		USR_PATHS	Lists all pathnames generated from user input pathname parts. (Default is off).
		ORIG_ARCS	Lists all solver arcs before the solver is called. (Default is off).
		SOLN_ARCS	Lists all solver arcs after a solution is determined. (Default is off).
		SOLN_NODES	Lists all solver nodes after solution is determined (Default is off).
		FIXED_SPLY	Lists all fixed supply flows to the nodes. (Default is off).
		ZERO_FLOW_PNLTY	Lists the calculation results from determining the penalty associated with zero flow and storage in all arcs. (Default is off).
		DUAL_COST	Lists dual costs for each node and arc. The dual cost for a node is the cost of routing an additional unit of flow to that node. The dual cost for an arc, with flow at its capacity, is the amount the objective function would change if the arc capacity were increased by one unit. (Default is off).
		ORIG_NODES	Lists all of the original nodes before calling the solver. (Default is off).
		SOLVER_DATA	Writes solver input to a text file that could be read by a stand-alone version of the network flow solver. (Default is off).

PCAT – Penalty Categories

Columns	Variable	Value	Description
1-4	CINREC	PCAT	These records contain a list of penalty categories. One record is entered for each category. A separate record is entered for storage and flow penalty categories for the same type of use (e.g. recreation storage and recreation flow penalty categories are listed separately). These records are currently optional. HEC-PRM writes a text record which contains this list to the output DSS data file. Information entered here is not currently used for any calculations. It is anticipated that it will be used in the future to identify and calculate individual time series penalties (e.g. the flood damage penalty).
11-20	PNLTY_CODE	character	The penalty category code name (e.g. PS_REC).
21-152	PNLTY_DESC	character	The description for this penalty category.

CONDITION – Conditions With Key to Time Series Data

This record is currently not used. It is planned to be used to help define time varying penalty functions.

Columns	Variable	Value	Description
1-9	CINREC	CONDITION	These records contain a list of conditions and the corresponding integer identifier that is used for time series data to identify the condition on a monthly time interval. One record is entered for each condition. These records are currently optional. Information entered here is not currently used for any calculations. It is anticipated that it will be used in the future to identify time varying penalty functions.
11-20	conditionId	integer	The integer condition ID. This cross-references id's entered in DSS time series data.
21-30	conditionName	character	The name for this condition.
31-132	conditionDescription	character	The description for this condition.

CONDTS – Time Varying Condition Identifier

This record contains the pathname parts for retrieving time varying condition identifiers. The corresponding conditions are used to identify time varying penalty functions. The identifiers are integer numbers that must match an ID_CONDITN entered on the CONDITION records. Currently, HEC-PRM only reads this record and cannot process time varying penalty functions.

Columns	Variable	Value	Description
1-6	CINREC	CONDTS	This record contains the pathname parts to the regular interval time series condition identifier records.
11-132	CTSPRT	character	CTSPRT contains the pathname parts to the record that contains the time varying condition identifiers. It is entered in the format of: A= B=SYSTEM C=CONDITION D= E=1MON F=

Output to HEC-DSS

ZW – Results Pathname Part F

Columns	Variable	Value	Description
1-2	CINREC	ZW	This record contains the character identification for pathname part F of the results for this simulation.
3-80	SIMLID	character	SIMLID is the portion of pathname part F which is used to identify this simulation. It is used to store results under separate DSS records for different simulations (e.g. to differentiate results after changing penalty functions between simulations). HEC-PRM uses SIMLID to form pathname part F. (Maximum of 20 characters; recommend using less). It is entered in the format of F=XXX where XXX is pathname part F.

ZWTS – Control of Time Series Output to HEC-DSS

Columns	Variable	Value	Description
1-4	CINREC	ZWTS	This optional record defines the regular interval time series results written to the Output DSS Data File. By default, HEC-PRM stores the following results in the Output DSS Data File: Reservoir Storage (kaf), Reservoir Evaporation (kaf), Flow (kaf) including reservoir releases, channel flow, and Diversions.

Columns	Variable	Value	Description
5-80	ZW_PARM_TS	character	Parameter names are entered in free-format. If a parameter has already been defined, a new definition will over-ride the previously defined setting. To store results in the Output DSS Data File, enter the parameter name. To prevent the storage of results in the Output DSS Data File, insert a minus ("-") sign before the parameter name (there is no space between the minus sign and the parameter name). To store results for all possible parameters, enter the keyword "ALL". To prevent the storage of any results in the Output DSS Data File, enter the keyword "-ALL". Valid parameters and the default value for controlling the storage of results are shown below:

17 Possible Parameters Which May Be Written to the Output DSS Data File

Parameter	Default
STOR	Yes
STOR_AVG	No
ELEV	No
EVAP(KAF)	Yes
EVAP	No
FLOW_IN	No
FLOW_IN(KAF)	No
ENERGY_GEN	No
POWER_CAP	No
FLOW_LOC	No
FLOW_LOC(KAF)	No
FLOW	No
FLOW(KAF)	Yes
FLOW_DIV	No
FLOW_DIV(KAF)	Yes
FLOW_NAT	No
Parameter	**Default**
FLOW_NAT(KAF)	No
DUAL_COST	No
NAV_LOCK	No

The record sequence below demonstrates suppressing all output and then defining the parameters for output results which will be stored in the Output DSS Data File as: reservoir storage (kaf) and flow (cfs, for reservoir release, channel flow, and diversions):

ZWTS –ALL

ZWTS STOR FLOW FLOW_DIV

The parameters are defined as follows

Parameter	Description
STOR	End-of-Period Reservoir storage (in 1,0(feet).
STOR_AVG	Average for the period reservoir storage acre-feet).
ELEV	End-of-Period Reservoir elevation (inpu function DSS data file must contain elev area-capacity table).
EVAP(KAF)	Reservoir evaporation (in 1,000 acre-fee

Columns	Variable	Value	Description	
			EVAP	Reservoir evaporation (in cfs).
			FLOW_IN	Total inflow (in cfs) to each node.
			FLOW_IN(KAF)	Total inflow (in kaf/month) to each node.
			ENERGY_GEN	Energy generated (input penalty function DSS data file must contain a table of flow-energy generated for various reservoir storages).
			POWER_CAP	Power capacity (input penalty function DSS data file must contain a table of flow-capacity for various reservoir storages).
			FLOW_LOC	Local inflow (in cfs) to each node.
			FLOW_LOC(KAF)	Local inflow (in kaf) to each node.
			FLOW	Total flow (in cfs) leaving this node (reservoir release or channel flow).
			FLOW(KAF)	Total flow (in kaf) leaving this node (reservoir release or channel flow).
			FLOW_DIV	Flow diverted (in cfs) from this node. It is stored with pathname part B containing the "from" node and the "to" node identifiers separated by a "-" character.
			FLOW_DIV(KAF)	Flow diverted (in kaf) from this node. It is stored with pathname part B containing the "from" node and the "to" node identifiers separated by a "-" character.
			FLOW_NAT	Natural flow (in cfs) which would occur at this node (diversions are ignored in calculating natural flows).

Parameter	Description
FLOW_NAT(KAF)	Natural flow (in kaf) which would occur at this node (diversions are ignored in calculating natural flows).
DUAL_COST	The marginal cost associated with an increase in arc capacity or demand at a node.
NAV_LOCK	Navigation lockages (input penalty function DSS data file must contain a table of flow-navigation for various reservoir storages.

ZWFRQ – Control of Frequency-Duration Output to HEC-DSS

Columns	Variable	Value	Description
1-5	CINREC	ZWFRQ	This optional record defines the Frequency-Duration results written to the Output DSS Data File. The following frequency-duration results can be written to the output DSS data file: reservoir storage (kaf), reservoir evaporation (kaf), and flow (kaf), including reservoir releases, channel flow, and diversions. By default, HEC-PRM does not store any frequency-duration data in the output DSS data file.
6-132	ZW_PARM_FRQ	character	Parameter names are entered in free-format. If a parameter has already been defined, a new definition will over-ride the previously defined setting. To store results in the Output DSS Data File, enter the parameter name. To prevent the storage of results in the Output DSS Data File, insert a minus ("-") sign before the parameter name (there is no space between the minus sign and the parameter name). To store results for all possible parameters, enter the keyword "ALL". To prevent the storage of any results in the Output DSS Data File, enter the keyword "-ALL". The parameter names are defined previously for the "ZWTS" record.

Columns	Variable	Value	Description
			The record sequence below demonstrates suppressing all output and then defining the parameters for output results which will be stored in the Output DSS Data File as: reservoir storage (kaf) and flow (cfs, for reservoir release, channel flow, and diversions): ZWFRQ –ALL ZWFRQ STOR FLOW FLOW_DIV Data for one parameter is stored in two pathnames: (1) Monthly frequency-duration curves and (2) Total frequency-duration curves. The record of monthly frequency-duration curves contains 12 curves, one for each month of the year. The first variable is the percent of time a given value of the parameter (e.g. flow) is exceeded, and the second variable is the value for the parameter (e.g. flow). The percent-time-exceeded is computed by the formula R/(N+1), where R is the rank (from highest to lowest) of the parameter value and N is the number of values. The record of total frequency-duration curves contains one curve and represents all values computed during the HEC-PRM analysis. The frequency is computed in a manner similar to that for the monthly curves.

Model Building Data

NODE - Node Identifier

Columns	Variable	Value	Description
1-8	CINREC	NODE	These records contain the identification code for each node and the associated character description. These are required records. All "NODE " records must be entered before entering any "LINK" records. There must be one "NODE " record for each node in the network. HEC-PRM automatically defines the first two nodes to be the Super Source and the Super Sink. You may optionally define the Super Source and Sink on "NODE " records. If you include the "IDENT" record, it must precede the "NODE " records.
11-20	IDNODE	character	The code identifier for this node. This node identifier is used on the "LINK" records to define the network. Each IDNODE must be unique for this input data set.
21-30	RSTO1	blank, + real no.	The initial storage at the beginning of the simulation for this node if it is a reservoir. If this node is not a reservoir, leave this field blank. HEC-PRM defines this node as a reservoir if this field is non-blank. RSTO1 is entered in thousands of acre-feet. (Default is blank; if reservoir, must be a positive real number).
31-40	RSTOK	blank, + real no.	The factor for converting reservoir capacity into area for use in computing evaporation. It is computed by estimating a representative slope of the area-capacity curve. The numerator is the change in area (in thousands of acres), and the denominator is the change in volume (in thousands of acre-feet). (Default is 0.1; Must be zero or a positive real number).
41-50	RSTOT	blank, + real no.	The end-of-period reservoir storage (in thousands of acre-feet) for the last time period of the optimization. This is a constrained value. A target value (which might be unmet, albeit at some cost) could be specified by leaving RSTOT blank and including a PS record with MO = LAST. Enter RSTOT only for reservoirs.

ND – Node Description

Columns	Variable	Value	Description
1-8	CINREC	ND	This record contains the description for the current node defined on the "NODE" record. It is an optional record.
11-80	DENODE	character	The character description for this node. It can contain any textual information.

LINK – Link Identifier

Columns	Variable	Value	Description
1-4	CINREC	LINK	These records contain the information for the links between nodes. They are required records. HEC-PRM defines the overflow arc between the Super Source and the Super Sink to be the first arc in the network. You need not enter a "LINK" record to define this arc but you may. The typical reason why you might enter this link is to specify the unit cost.
11-20	LINKTY	character	The type of link. (No default, the user must enter a value). Must match one of the following codes:

Code	Description
CHAN	Channel link. Originates at a non-reservoir node and term node.
DIVR	Diversion link. Originates at any node and terminates at a Typically, but not necessarily, used to carry flow out of th the link from the Super Source to the Super Sink and the li node to the Super Sink.
INFL	Inflow link. Originates at the Super Source node and term node.
RREL	Reservoir release link. Originates at a reservoir node and other node. Used for releases in which cost is a function o
RSTO	Reservoir storage link. Originates at a reservoir node and same node (in the next time period).
HREL	Hydropower release link. Originates at a reservoir node an other node. Used for releases in which cost is a nonlinear f storage. Note: A reservoir can have either a RREL link or not both. Additional releases can be specified as DIVR lin

Columns	Variable	Value	Description
21-30	LIFROM	character	The node identification from which flow travels through this link. It must match one of the identifiers entered on the "NODE" records or the Super Source or the Super Sink. (No default, user must enter node identifier in this field; Maximum of 10 characters
31-40	LINKTO	character	The node identification to which flow travels through this link. It must match one of the identifiers entered on the "NODE" records or the Super Source or the Super Sink. (No default, user must enter node identifier in this field; Maximum of 10 characters)
41-50	LIAMP	+ real no.	The amplitude to be used for this link. If left blank, it is set to "1.0". (Default is "1.0"; Must be a positive number)
51-60	LICOST	real no.	The constant cost for all time periods for this link. This field is normally left blank. If a number is entered here, do not enter penalty functions. (Default is "0.0"; May be positive, negative, or zero). Note: All links require either a non-zero constant cost or a PQ or PS record. Without either of these, errors can occur in HEC-PRM, but it may still report–erroneously–an optimal solution.
61-70	LILOWR	+ real no.	The lower bound for this link. This is the lowest allowable value of volume (flow / storage) in the penalty function for which the algorithm will solve. This is normally left blank. Care must be exercised in using bounds because the algorithm will abort if it cannot determine a feasible solution. (Default is "0.0"; Must be zero or a positive number)
71-80	LIUPPR	+ real no.	The upper bound for this link. This is the highest allowable value of volume (flow / storage) in the penalty function for which the algorithm will solve. This is normally left blank. Care must be exercised in using bounds because the algorithm will abort if it cannot determine a feasible solution. (Default is to set LIUPPR equal to parameter "BIG" (J3 record). Must be positive number).
81-90	LICONS	blank, 0, + real no.	Constant bounds for this link. The lower and upper bounds are the same for all time periods. Normally, leave blank. Default is a big negative number indicating that it has not been defined. If entered, it must be set to either zero or a positive number.

LD – Link Description

Columns	Variable	Value	Description
1-8	CINREC	LD	This record contains the description for the current link defined on the "LINK" record. It is an optional record.
11-80	LIDESC	character	The character description for this link. It can contain any textual information.

EAC – Elevation-Area-Capacity

Columns	Variable	Value	Description
1-3	CINREC	EAC	This record contains HEC-DSS pathname parts and instructs HEC-PRM to read elevation-area-capacity data for a reservoir. This “triplet” data must be stored in the paired data (penalty function) HEC-DSS file. The order of the triplet data is: elevation (feet), area (thousands of acres), and capacity (thousands of acre-feet). HEC-PRM will use this data, along with net evaporation rates, to determine the volume of evaporation in each month of the analysis. If this record is omitted, and LIAMP is left blank or equal to 1.0, there will be no evaporation computed. Must be entered before the PS record.
4-n	–	character	The HEC-DSS pathname parts, entered in free format. The pathname part is preceded by an equal "=" sign which is preceded by the part identifier (A through F). Part C must be “EL-AR-CAP”. Parts D and E are to be left blank. If a part is blank, enter the part identifier, the equal sign, and at least one blank. HEC-PRM remembers time series parts from the last entry (either "IN", "EV", "QC", "QL" or "QU" records)--you need only update necessary parts. Example entry: EAC B = FOLSOM C = EL-AR-CAP F =

LB –Starting and Ending Time Period Bounds

The optional LB record contains bounds for the first and last time periods of the optimization.

Columns	Variable	Value	Description
1-2	CINREC	LB	This record contains the bounds for the starting and ending time periods of the optimization. It is an optional record. If it is not entered, the bounds are taken from the Link, the monthly varying bounds, or the time varying bounds records. Either the starting or the ending or both bounds may be set. The lower and upper bounds may be set separately or they may be set to the same value.
11-20	LIEXTBOU(1,L)	blank, 0, + real no.	Lower bounds for the first time period.
21-30	LIEXTBOU(2,L)	blank, 0, + real no.	Upper bounds for the first time period.
31-40	LIEXTBOU(3,L)	blank, 0, + real no.	Lower and Upper bounds for the first time period (they are the same).
41-50	LIEXTBOU(4,L)	blank, 0, + real no.	Lower bounds for the last time period.
51-60	LIEXTBOU(5,L)	blank, 0, + real no.	Upper bounds for the last time period.
61-70	LIEXTBOU(6,L)	blank, 0, + real no.	Lower and Upper bounds for the last time period (they are the same).

Monthly Varying Bounds

These records allow the user to specify the lower and upper monthly bounds of flow and are used to define the bounds that vary from one month to the next and do not vary from one year to the next. For example, the bounds for all Januarys are the same and the bounds for all Februarys are the same, but are different than Januarys. The BC record specifies that the upper and lower bounds are the same.

BC – Explicit Monthly Varying Flow

Columns	Variable	Value	Description
1-2	CINREC	BC	This record defines the lower and upper monthly bounds to be the same value. If the bounds remain constant throughout the simulation, they are entered on the "LINK" record. If it follows a monthly pattern (all January bounds are the same, all February bounds are the same but different than January, etc.), then it must be entered on this record.
3-n	LILOWR	0, + real no.	The monthly varying bounds are entered in free-format. Twelve values must be entered. The first value corresponds to January. If you enter adjacent commas, the corresponding value is defined by the value entered on the LINK record.

BL – Monthly Varying Lower Bounds

Columns	Variable	Value	Description
1-2	CINREC	BL	This record contains information about the lower bounds of the penalty function. If the lower bound remains constant throughout the simulation, it is entered on the "LINK" record. If it follows a monthly pattern (all January bounds are the same, all February bounds are the same but different than January, etc.), then it must be entered on this record.
3-n	LILOWR	0, + real no.	The monthly varying lower bound is entered in free-format. Twelve values must be entered. The first value corresponds to January. If you enter adjacent commas, the corresponding value is defined by the value entered on the LINK record.

BU – Monthly Varying Upper Bound

Columns	Variable	Value	Description
1-2	CINREC	BU	This record contains information about the upper bounds of the penalty function. If the upper bound remains constant throughout the simulation, it is entered on the "LINK" record. If it follows a monthly pattern (all January bounds are the same, all February bounds are the same but different than January, etc.), then it must be entered on this record.
3-n	LIUPPR	0, + real no.	The monthly varying upper bound is entered in free-format. Twelve values must be entered. The first value corresponds to January. If you enter adjacent commas, the corresponding value is defined by the value entered on the LINK record.

Time Varying Bounds

Note: The QC, QL, and QU records allow the user to set the lower and upper bounds uniquely for each time period. HEC-PRM reads monthly regular interval time series data from the input time series DSS data file. The QC record sets the upper and lower bounds to the same value (nothing will be optimized for that link). The QL defines the lower bounds and the QU defines the upper bounds. If a QL and / or QU record(s) are entered, then do not enter a QC record.

QC – Explicit Time Varying Flow

Columns	Variable	Value	Description
1-3	CINREC	QC	This record contains HEC-DSS pathname parts and instructs HEC-PRM to read time series data defining the upper and lower bounds for this link. HEC-PRM sets the lower and upper bounds to the same value which can be unique for each time period. If this record is entered, do not enter the records QL and QU.
4-n	CINREC	character	The pathname parts are entered in free-format. The pathname part is preceded by an equal "=" sign which is preceded by the part identifier (A through F). Parts D and E are assumed. Part E is the time interval (1MON), and part D is the standard block time for regular interval time series data. Part D is generated from the time window entered on the "TIME" record.

Columns	Variable	Value	Description
			If a part is blank, enter the part identifier, the equal sign, and at least one blank. The HEC-PRM remembers time series parts from the last entry (either "IN", "EV", "QC", "QL" or "QU" records) --- you need only update necessary parts. Example entry: QC B = GRANITE_P

QL – Time Varying Lower Bound

Columns	Variable	Value	Description
1-3	CINREC	QL	This optional record contains HEC-DSS pathname parts and instructs HEC-PRM to read time series data defining the lower bounds for this link. HEC-PRM sets the lower bounds to the regular time series value which can be unique for each time period. If this record is entered, do not enter the record QC. A QL record may be accompanied by a QU record but it is not required.
4-n	CINREC	character	The pathname parts are entered in free-format. The pathname part is preceded by an equal "=" sign which is preceded by the part identifier (A through F). Parts D and E are assumed. Part E is the time interval (1MON), and part D is the standard block time for regular interval time series data. Part D is generated from the time window entered on the "TIME" record. If a part is blank, enter the part identifier, the equal sign, and at least one blank. HEC-PRM remembers time series parts from the last entry (either "IN", "EV", "QC", "QL" or "QU" records)--you need only update necessary parts. Example entry: QL B=GRANITE_PQU

QU – Time Varying Upper Bound

Columns	Variables	Value	Description
1-3	CINREC	QU	This optional record contains HEC-DSS pathname parts and instructs HEC-PRM to read time series data defining the upper bounds for this link. If this record is entered, do not enter the record QC. A QU record may be accompanied by a QL record but it is not required
4-n	CINREC	character	The pathname parts are entered in free-format in the format described for the "QL" record. Example entry: QU B = Granite_P

Time Varying Initial Flow and Evaporation

The QI records allow the user to specify an initial flow (or storage) time series. HEC-PRM reads these time series from the input time series DSS data file. Providing feasible (or nearly feasible) time series can reduce computation time substantially.
The EV record contains evaporation rates for reservoir storage links. It is an optional record that will be used to compute the arc amplitudes representing evaporation.

QI – Initial Flow

Columns	Variable	Value	Description
1-2	CINREC	QI	This record contains HEC-DSS pathname parts and instructs HEC-PRM to read a time series of initial flows for the link from the time series input DSS data file. These flows might correspond to the solution of a previous run. Due to the single-precision nature of HEC-DSS, results of a previous run will not specify a feasible solution (in double-precision), but experience has shown that Phase I computation time could be reduced significantly. Flows should be specified in thousands of acre-feet per month.
3-n	–	character	The HEC-DSS pathname parts, entered in free format.

EV – Evaporation Rate

Columns	Variable	Value	Description
1-2	CINREC	EV	This record contains HEC-DSS pathname parts and instructs HEC-PRM to read time series data for net evaporation from the input time series DSS data file. The net evaporation should be expressed as a rate (in feet per month). This record is valid only for Reservoir Storage ("RSTO") links. If no EV record is specified for a given reservoir, evaporation from that reservoir is not computed.
3-n	CINREC	character	The pathname parts are entered in free-format. The pathname part is preceded by an equal "=" sign which is preceded by the part identifier (A through F). Parts D and E are assumed. Part E is the time interval (1MON), and part D is the standard block time for regular interval time series data. Part D is generated from the time window entered on the "TIME" record. If part C is not entered, HEC-PRM assumes it will be "EVAP_RATE". If a part is blank, enter the part identifier, the equal sign, and at least one blank. HEC-PRM remembers time series parts from the last entry (either "IN" or "EV" records)--you need only update necessary parts. Example entry: EV B = FTPK

Fixed Cost

Normally, arc costs are computed from the penalty functions. In lieu of entering a penalty function, either the CM or CT records are entered to define a single fixed cost for each time period.

CM – Monthly Fixed Cost

Columns	Variable	Value	Description
1-2	CINREC	CM	This record defines the monthly fixed cost. If the cost remains constant throughout the simulation, they are entered on the "LINK" record. If it follows a monthly pattern (all January costs are the same, all February costs are the same but different than January, etc.), then it must be entered on this record. If a penalty function is entered for a given month, the costs calculated from it will override the value entered on the CM record.
3-n	–	0, + real no.	The monthly varying costs are entered in free-format. Twelve values must be entered. The first value corresponds to January. If you enter adjacent commas, the corresponding value is defined by the value entered on the LINK record.

CT – Time Varying Fixed Cost

Columns	Variable	Value	Description
1-2	CINREC	CT	This record contains HEC-DSS pathname parts and instructs HEC-PRM to read a time series of costs for the link from the time series input DSS data file. If both CM and CT records are entered, values based on the CT record take precedence. If a penalty function is entered for a given month, the costs calculated from it will override the values based on the CT record.
3-n	–	character	The HEC-DSS pathname parts, entered in free format.

Explicit Amplitudes

Normally, amplitudes are computed only for reservoir links that have evaporation based upon the EV record. However, amplitudes may be explicitly defined as monthly values (AM record) or time varying values (AT record). If both AM and AT records are entered, values based upon the AT records take precedence.

AM – Monthly Amplitude

Columns	Variable	Value	Description
1-2	CINREC	AM	This record defines the monthly amplitude. If the amplitude remains constant throughout the simulation, it is entered on the "LINK" record. If it follows a monthly pattern (all January amplitudes are the same, all February amplitudes are the same but different than January, etc.), then they must be entered on this record. If evaporation rates are defined using the EV record, the amplitudes calculated from it will override the values entered on the CM record.
3-n	–	0, + real no.	The monthly varying amplitudes are entered in free-format. Twelve values must be entered. The first value corresponds to January. If you enter adjacent commas, the corresponding value is defined by the value entered on the LINK record.

AT – Time Varying Amplitude

Columns	Variable	Value	Description
1-2	CINREC	AT	This record contains HEC-DSS pathname parts and instructs HEC-PRM to read a time series of amplitudes for the link from the time series input DSS data file. If both AM and AT records are entered, values based on the AT record take precedence.
3-n	–	character	The HEC-DSS pathname parts, entered in free format.

IN – Local Inflow

Columns	Variable	Value	Description
1-2	CINREC	IN	This record contains HEC-DSS pathname parts and instructs the HEC-PRM to read time series data for local inflow. The flow should be expressed as a volume per month in 1,000 Acre-Feet. This record may be entered for any inflow link.
3-n	CINREC	character	The pathname parts are entered in free-format. The pathname part is preceded by an equal "=" sign which is preceded by the part identifier (A through F). Parts D and E are assumed. Part E is the time interval (1MON), and part D is the standard block time for regular interval time series data. Part D is generated from the time window entered on the "TIME" record. If part C is not entered, the HEC-PRM assumes it will be "FLOW_LOC(KAF)". If a part is blank, enter the part identifier, the equal sign, and at least one blank. The HEC-PRM remembers time series parts from the last entry (either "IN" or "EV" records)--you need only update necessary parts. Example entry: IN A = B = FTPK F =

Penalty Function Data

PS – Storage Penalty Function

Columns	Variable	Value	Description
1-2	CINREC	PS	This record contains HEC-DSS pathname parts and instructs HEC-PRM to read a reservoir storage penalty function from the paired function DSS data file. This record is used with "RSTO" links. Penalties should be expressed in thousands of dollars and storage in KAF. This record also identifies the month or range of months that the penalty function is applicable. There can be a separate penalty function for each month of the year, but the penalty function cannot vary from year to year. The first penalty function must be for January or for a range of consecutive months starting in January. The functions must be in chronological order (e.g., you cannot enter the August function before the July function), and you must define a function for each month. Note: A PS or PS2 record is required for all RSTO links, unless a non-zero constant cost (LICOST) is specified.
3-n	CINREC	character	The pathname parts are entered in free-format. The pathname part is preceded by an equal "=" sign which is preceded by the part identifier (A through F). The months are entered in free-format using the 3 character month identifier (e.g. "FEB"). If the penalty function spans 2 or more consecutive months, the starting and ending months must be separated by the "-" (dash) character. Similar to the

Columns	Variable	Value	Description
			pathname parts, the months are preceded by an "=" (equal) sign which is preceded by the characters "MO". If part C is not entered, HEC-PRM assumes it will be "S-P_EDT". Use "MO = LAST" and "E = LAST" to specify the storage penalties for the last time period in the analysis. If a part is blank, enter the part identifier, the equal sign, and at least one blank. The HEC-PRM remembers paired function parts from the last entry (either "PS" or "PQ" records)--you need only update necessary parts. Example entries: PS MO=JAN-MAR B=COULEE_P E=JAN PS MO=JUL B=GRANITE_P E=JUL PS MO=JAN-DEC B=DWORSHAK_P

PS2 – Non-Linear Storage Penalty Function

Columns	Variable	Value	Description
1-3	CINREC	PS2	This record contains HEC-DSS pathname parts and instructs HEC-PRM to read quadratic storage penalty coefficients from the paired function DSS data file. This record is similar to the PS record, except that the penalty function is specified using two paired data values: [(1, C1), (2, C2)], where C1 is the linear coefficient and C2 is the quadratic penalty coefficient.
4-n	–	character	The HEC-DSS pathname parts, entered in free format.

PQ – Flow Penalty Function

Columns	Variable	Value	Description
1-2	CINREC	PQ	This record contains HEC-DSS pathname parts and instructs HEC-PRM to read flow penalty functions from the paired function HEC-DSS data file. This record is used with the following types of LINK records: CHAN, RREL, HREL, and DIVR. The penalty should be expressed in thousands of dollars and the flow in 1,000 Acre-Feet. It also identifies the month or range of months for which this penalty function is applicable. There can be a separate penalty function for each of the months within a year. The penalty function cannot vary from year-to-year. The first penalty function must be for January or for a range of consecutive months starting in January. The functions must be in chronological order (e.g. you cannot enter the August function before the July function). You must define a function for each month.
3-n	CINREC	character	The pathname parts are entered in free-format. The pathname part is preceded by an equal "=" sign which is preceded by the part identifier (A through F). The months are entered in free-format using the 3 character month identifier (e.g. "FEB"). If the penalty function spans 2 or more consecutive months, the starting and ending months must be separated by the "-" (dash) character. Similar to the pathname parts, the months are preceded by an "=" (equal) sign which is preceded by the characters "MO". If part C is not entered, the HEC-PRM assumes it will be "Q(KAF)-P_EDT". If a part is blank, enter the part identifier, the equal sign, and at least one blank. The HEC-PRM remembers paired function parts from the last entry (either "PS" or "PQ" records) --- you need only update necessary parts. Example entries PQ MO=JAN-MAR A= B=COULEE-WELLS_P E=JAN F= PQ MO=JUL B=COULEE-WELLS_P E=JUL PQ MO=JAN-DEC B=COULEE-WELLS_P PQ MO=JAN B=COULEE-WELLS_P C=Q(KAF)-P_HPE

PQ2 – Non-Linear Flow Penalty Function

Columns	Variable	Value	Description
1-3	CINREC	PQ2	This record contains HEC-DSS pathname parts and instructs HEC-PRM to read quadratic flow penalty coefficients from the paired function DSS data file. This record is similar to the PQ record, except that the penalty function is specified using two paired data values: [(1, C1), (2, C2)], where C1 is the linear coefficient and C2 is the quadratic penalty coefficient.

Terminate Input

Any one of the following commands may be entered. If more than one command is entered, only the first is observed. The entry of one of these commands instructs HEC-PRM to terminate reading user input, generate the network matrix, call the solver, and store the results in the output DSS data file.

STOP

Columns	Variable	Value	Description
1-4	CINREC	STOP	Terminate user input and commence generating and solving the network.

FINISH

Columns	Variable	Value	Description
1-4	CINREC	FINISH	Terminate user input and commence generating and solving the network.

QUIT

Columns	Variable	Value	Description
1-4	CINREC	QUIT	Terminate user input and commence generating and solving the network.

CHAPTER 5 - Example User Input and Output

The following lists a typical HEC-PRM output file. The input data records are listed as the first part of the output data. This particular data set is for the 1-year time window from March 1965 through February 1966. By modifying the "TIME" record, it could easily be adapted to the 5-year validation period or the 23-year critical period.

{Banner Page}

```
          Reservoir System Operation Optimization
               Version   1.0;  April 1, 1994
             IBM-PC Compatible  (Lahey 32bit)
             Run date 31MAR94   time 08:48:42


                           HECPRM




              U.S. Army Corps of Engineers
              Hydrologic Engineering Center
               609 Second Street, Suite B
                 Davis, California 95616
                     (916) 756-1104

Start of data processing.
 March     31, 1994   08:48:42
```

{Open DSS files}

```
 -----DSS---ZOPEN:  Existing File Opened,  File: SAMPLE5T.DSS
        Unit:  71;  DSS Version: 6-HB
-----DSS---ZOPEN:  Existing File Opened,  File: SAMPLE5P.DSS
        Unit:  72;  DSS Version: 6-HB
 -----DSS---ZOPEN:  New File Opened,  File: SOLN\SAMPLE5.DSS
        Unit:  73;  DSS Version: 6-IG


SUP_DIRECTORY:    C:\HECEXE\SUP\HECPRM.ERR
NC_SUP_DIRECTORY:       24
```

{Options for writing Auxiliary files}

```
WR_SPLY...........     F
WR_ARC............     F
WR_OLD_ARC........     F
WR_MCS............     T
WR_OARC...........     F
```

{Dimensions allocated for user's system based upon input data file}

```
              Dimensions allocated for this Job

 Number of Nodes.................................      10
 Number of Reservoirs............................       3
 Number of Links.................................      19
 Number of DSS Pathnames.........................      76
 Number of Time Periods..........................     240

 Number of Time Series Buffer....................     260
 Number of Paired Data Buffer (Retrieval)........     700
 Number of Paired Data Buffer (Storage)..........    1400
 Number of Penalty Categories....................      14
 Number of Text Input Lines......................       1
```

```
System Size determined.
March     31, 1994   08:48:43
Elapsed Time is         1 seconds or
                     0.02 minutes or
                    0.000 hours.

User parameters initialized
March     31, 1994   08:48:43
Elapsed Time is         1 seconds or
                     0.02 minutes or
                    0.000 hours.
```

{Echo Input Data}

```
..        SNAKE RIVER:  Dworshak to Granite
ZW        F=SAMPLE5
..
..        20-Year Analysis (1928-1948)
..
..        ALTERNATIVE 1: Current Operation Rules (via Penalty Curves)
..
..        1) FLOW DATA ARE 1980 LEVEL OF DEVELOPMENT
..        2) PENALTY DATA ARE PHASE 1.5 WITH MODIFICATIONS
..
..
IDENT     S_SOURCE    SINK
TIME      JUL1928 JUN1948
..
J11.0E-05 1.0E+06          1.0       1.0         1         3
            EPS            BIGS         FACFLO         FACCST         IPRINT   DSS_PRNT_LVL
      0.0000100       1000000.       1.00000       1.00000              1              3
  0.1000000E-04  0.1000000E+07  0.1000000E+01  0.1000000E+01              1              3

ZWTS  -ALL
ZWTS      STOR FLOW FLOW(KAF) FLOW_NAT FLOW_NAT(KAF) FLOW_IN FLOW_IN(KAF)
ZWFRQ     STOR FLOW FLOW(KAF)
..
PCAT      PS_FDA    Agricultural Flood Damage related to storage
PCAT      PS_FDU    Urban Flood Damage related to storage
PCAT      PS_FIS    Anadroumous Fish related to storage
PCAT      PS_NAV    Navigation related to storage
PCAT      PS_REC    Recreation related to storage
PCAT      PS_WSP    Water Supply related to storage
PCAT      PQ_FDA    Agricultural Flood Damage related to flow
PCAT      PQ_FDU    Urban Flood Damage related to flow
PCAT      PQ_FIS    Anadroumous Fish related to flow
PCAT      PQ_NAV    Navigation related to flow
PCAT      PQ_REC    Recreation related to flow
PCAT      PQ_WSP    Water Supply related to flow
PCAT      PQ_HEA    Hydropower Energy related to flow
PCAT      PQ_HCA    Hydropower Capacity related to flow
..
NODE      DWORSHAK_P    3468.0       0.1    3468.0
ND        Dworshak Reservoir Power Penalties
NODE      DWORSHAK
ND        Dworshak Reservoir Non-Power Penalties
NODE      SPALDING
ND        Spalding
NODE      BROWNLEE_P    1426.7       0.1    1426.7
ND        Brownlee, Oxbow, and Hells Canyon Power Penalties
NODE      BROWNLEE
ND        Brownlee, Oxbow, and Hells Canyon Non-Power Penalties
              NODE      GRANITE_P     1825.0       0.1    1825.0
ND        L.Granite\Little Goose\L.Monumental\Ice Harbor Power Penalties
NODE      GRANITE
ND        L.Granite\Little Goose\L.Monumental\Ice Harbor Non-Power Penalties

LINK      DIVR      S_SOURCE  SINK      1.0       0.0
LD        Continuity Link
```

```
LINK      INFLOW    S_SOURCE  DWORSHAK_P  1.0       0.0
LD        Inflow to Dworshak Reservoir
IN        B=DWORSHAK_P C=FLOW_LOC(KAF) E=1MON F=INC-NAT80

LINK      RSTORAGE  DWORSHAK_PDWORSHAK_P1.0       0.0       1452.2    3468.0
LD        Storage in Dworshak Reservoir
PS        MO=JAN        B=DWORSHAK_P C=S-P_EDT  E=JAN F=
PS        MO=FEB E=FEB
PS        MO=MAR E=MAR
PS        MO=APR E=APR
PS        MO=MAY E=MAY
PS        MO=JUN E=JUN
PS        MO=JUL E=JUL
PS        MO=AUG E=AUG
PS        MO=SEP E=SEP
PS        MO=OCT E=OCT
PS        MO=NOV E=NOV
PS        MO=DEC E=DEC

LINK      HREL      DWORSHAK_PDWORSHAK
LD        Power Release from Dworshak Reservoir
PQ        MO=JAN-FEB B=DWORSHAK_P          C=Q(KAF)-P_HPE E=DEC-FEB
PQ        MO=MAR     B=DWORSHAK_P          C=Q(KAF)-P_HPE E=MAR_NOV
PQ        MO=APR-MAY B=DWORSHAK_P          C=Q(KAF)-P_HPE E=APR-MAY_OCT
PQ        MO=JUN-SEP B=DWORSHAK_P          C=Q(KAF)-P_HPE E=JUN-SEP
PQ        MO=OCT     B=DWORSHAK_P          C=Q(KAF)-P_HPE E=APR-MAY_OCT
PQ        MO=NOV     B=DWORSHAK_P          C=Q(KAF)-P_HPE E=MAR_NOV
PQ        MO=DEC     B=DWORSHAK_P          C=Q(KAF)-P_HPE E=DEC-FEB

LINK      RRELEASE  DWORSHAK   SPALDING                     60.4
LD        Other Releases from Dworshak Reservoir to Spalding
PQ        MO=JAN-DEC    B=DWORSHAK_P        C=Q(KAF)-P_EDT E=JAN F=ZERO

LINK      INFLOW    S_SOURCE  SPALDING  1.0       0.0
LD        Inflow to Spalding
IN        B=SPALDING C=FLOW_LOC(KAF) E=1MON F=INC-NAT80

LINK      CHANNEL   SPALDING  GRANITE_P
LD        Channel from Spalding to Granite/Goose/Monumental/Ice Harbor
PQ        MO=JAN      B=SPALDING           C=Q(KAF)-P_EDT E=JAN F=
PQ        MO=FEB E=FEB
PQ        MO=MAR E=MAR
PQ        MO=APR E=APR
PQ        MO=MAY E=MAY
PQ        MO=JUN E=JUN
PQ        MO=JUL E=JUL
PQ        MO=AUG E=AUG
PQ        MO=SEP E=SEP
PQ        MO=OCT E=OCT
PQ        MO=NOV E=NOV
PQ        MO=DEC E=DEC

LINK      INFLOW    S_SOURCE  BROWNLEE_P1.0       0.0
LD        Inflow to Brownlee, Oxbow & Hells Canyon Reservoirs
IN        B=BROWNLEE_P C=FLOW_LOC(KAF) E=1MON F=INC-NAT80

LINK      RSTORAGE  BROWNLEE_PBROWNLEE_P1.0       0.0       431.7     1426.7
LD        Storage in Brownlee, Oxbow & Hells Canyon Reservoirs
PS        MO=JAN-DEC    B=BROWNLEE_P C=S-P_EDT  E=JAN F=ZERO

LINK      HREL      BROWNLEE_PBROWNLEE
LD        Power Release from Brownlee Reservoir
PQ        MO=JAN-FEB B=BROWNLEE_P          C=Q(KAF)-P_HPE E=DEC-FEB F=
PQ        MO=MAR     B=BROWNLEE_P          C=Q(KAF)-P_HPE E=MAR_NOV
PQ        MO=APR-MAY B=BROWNLEE_P          C=Q(KAF)-P_HPE E=APR-MAY_OCT
PQ        MO=JUN-SEP B=BROWNLEE_P          C=Q(KAF)-P_HPE E=JUN-SEP
PQ        MO=OCT     B=BROWNLEE_P          C=Q(KAF)-P_HPE E=APR-MAY_OCT
PQ        MO=NOV     B=BROWNLEE_P          C=Q(KAF)-P_HPE E=MAR_NOV
PQ        MO=DEC     B=BROWNLEE_P          C=Q(KAF)-P_HPE E=DEC-FEB

LINK      RRELEASE  BROWNLEE  GRANITE_P                     301.9
LD        Other Releases from Brownlee/Oxbow/Hells Canyon to Granite/...
PQ        MO=JAN-DEC    B=BROWNLEE_P         C=Q(KAF)-P_EDT E=JAN F=ZERO
```

```
LINK      INFLOW    S_SOURCE  GRANITE_P 1.0       0.0
LD        Inflow to Lower Granite thru Ice Harbor Reservoirs
IN        B=GRANITE_P  C=FLOW_LOC(KAF) E=1MON F=INC-NAT80

..        Lower limit = 144.0 based on Run-of-River conditions (Four Reservoirs)
LINK      RSTORAGE  GRANITE_P GRANITE_P 1.0                  144.0    1825.0
LD        Storage in Granite Reservoir
PS        MO=JAN        B=GRANITE_P C=S-P_EDT  E=JAN  F=
PS        MO=FEB E=FEB
PS        MO=MAR E=MAR
PS        MO=APR E=APR
PS        MO=MAY E=MAY
PS        MO=JUN E=JUN
PS        MO=JUL E=JUL
PS        MO=AUG E=AUG
PS        MO=SEP E=SEP
PS        MO=OCT E=OCT
PS        MO=NOV E=NOV
PS        MO=DEC E=DEC

LINK      HREL      GRANITE_P GRANITE
LD        Power Release from Granite Reservoir
PQ        MO=JAN-FEB B=GRANITE_P        C=Q(KAF)-P_HPE E=DEC-FEB
PQ        MO=MAR     B=GRANITE_P        C=Q(KAF)-P_HPE E=MAR_NOV
PQ        MO=APR-MAY B=GRANITE_P        C=Q(KAF)-P_HPE E=APR-MAY_OCT
PQ        MO=JUN-SEP B=GRANITE_P        C=Q(KAF)-P_HPE E=JUN-SEP
PQ        MO=OCT     B=GRANITE_P        C=Q(KAF)-P_HPE E=APR-MAY_OCT
PQ        MO=NOV     B=GRANITE_P        C=Q(KAF)-P_HPE E=MAR_NOV
PQ        MO=DEC     B=GRANITE_P        C=Q(KAF)-P_HPE E=DEC-FEB

LINK      RRELEASE  GRANITE   SINK
LD        Release from Granite/Goose/Monumental/Ice Harbor to Sink
PQ        MO=JAN       B=GRANITE_P       C=Q(KAF)-P_EDT E=JAN F=
PQ        MO=FEB E=FEB
PQ        MO=MAR E=MAR
PQ        MO=APR E=APR
PQ        MO=MAY E=MAY
PQ        MO=JUN E=JUN
PQ        MO=JUL E=JUL
PQ        MO=AUG E=AUG
PQ        MO=SEP E=SEP
PQ        MO=OCT E=OCT
PQ        MO=NOV E=NOV
PQ        MO=DEC E=DEC

STOP
```

{Write List of penalty categories (optional) and nodes to output DSS data file}

```
-----DSS---ZWRITE Unit  73; Vers.    1:  //SYSTEM NODES/ID-TYPE-DESC///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SYSTEM PNLTY CAT/ID-DESC///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SYSTEM FILES/ID-FILE_NAME///SAMPLE5/
```

{Write the edited, composite penalty functions to the output DSS data file. The input functions may be modified by HEC-PRM to account for the lower and upper bounds.}

```
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/S-P_EDT//JAN/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/S-P_EDT//FEB/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/S-P_EDT//MAR/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/S-P_EDT//APR/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/S-P_EDT//MAY/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/S-P_EDT//JUN/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/S-P_EDT//LAST/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/S-P_EDT//JUL/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/S-P_EDT//AUG/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/S-P_EDT//SEP/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/S-P_EDT//OCT/SAMPLE5/
```

```
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/S-P_EDT//NOV/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/S-P_EDT//DEC/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_HPE//JAN/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_HPE//FEB/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_HPE//MAR/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_HPE//APR/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_HPE//MAY/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_HPE//JUN/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_HPE//LAST/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_HPE//JUL/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_HPE//AUG/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_HPE//SEP/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_HPE//OCT/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_HPE//NOV/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_HPE//DEC/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_EDT//JAN/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_EDT//FEB/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_EDT//MAR/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_EDT//APR/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_EDT//MAY/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_EDT//JUN/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_EDT//LAST/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_EDT//JUL/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_EDT//AUG/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_EDT//SEP/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_EDT//OCT/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_EDT//NOV/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/Q(KAF)-P_EDT//DEC/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/Q(KAF)-P_EDT//JAN/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/Q(KAF)-P_EDT//FEB/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/Q(KAF)-P_EDT//MAR/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/Q(KAF)-P_EDT//APR/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/Q(KAF)-P_EDT//MAY/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/Q(KAF)-P_EDT//JUN/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/Q(KAF)-P_EDT//LAST/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/Q(KAF)-P_EDT//JUL/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/Q(KAF)-P_EDT//AUG/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/Q(KAF)-P_EDT//SEP/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/Q(KAF)-P_EDT//OCT/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/Q(KAF)-P_EDT//NOV/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/Q(KAF)-P_EDT//DEC/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/S-P_EDT//JAN/SAMPLE5/
```

{Additional output not shown}

```
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/Q(KAF)-P_HPE//DEC/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/Q(KAF)-P_EDT//JAN/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/Q(KAF)-P_EDT//FEB/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/Q(KAF)-P_EDT//MAR/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/Q(KAF)-P_EDT//APR/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/Q(KAF)-P_EDT//MAY/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/Q(KAF)-P_EDT//JUN/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/Q(KAF)-P_EDT//LAST/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/Q(KAF)-P_EDT//JUL/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/Q(KAF)-P_EDT//AUG/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/Q(KAF)-P_EDT//SEP/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/Q(KAF)-P_EDT//OCT/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/Q(KAF)-P_EDT//NOV/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/Q(KAF)-P_EDT//DEC/SAMPLE5/

Matrix generated.
March     31, 1994   08:48:48
Elapsed Time is         6 seconds or
                     0.10 minutes or
                    0.002 hours.


      EPSC      EPSF       BIG
 0.100E-08 0.100E-08 0.100E+10
```

{Print list of Parameters which will be stored in the Output DSS Data File}

```
17 Possible Parameters Which May Be Written to the Output DSS Data File

                         Time        Frequency-
Parameter                Series      Duration
STOR                     Yes          Yes
STOR_AVG                 No           No
ELEV                     No           No
EVAP(KAF)                No           No
EVAP                     No           No
FLOW_IN                  Yes          No
FLOW_IN(KAF)             Yes          No
ENERGY_GEN               No           No
POWER_CAP                No           No
FLOW_LOC                 No           No
FLOW_LOC(KAF)            No           No
FLOW                     Yes          Yes
FLOW(KAF)                Yes          Yes
FLOW_DIV                 No           No
FLOW_DIV(KAF)            No           No
FLOW_NAT                 Yes          No
FLOW_NAT(KAF)            Yes          No
```

{Print List of User's Nodes}

```
 9 Nodes in this system

  # ID          Description
  1 S_SOURCE
  2 SINK
  3 DWORSHAK_P Dworshak Reservoir Power Penalties
  4 DWORSHAK   Dworshak Reservoir Non-Power Penalties
  5 SPALDING   Spalding
  6 BROWNLEE_P Brownlee, Oxbow, and Hells Canyon Power Penalties
  7 BROWNLEE   Brownlee, Oxbow, and Hells Canyon Non-Power Penalties
  8 GRANITE_P  L.Granite\Little Goose\L.Monumental\Ice Harbor Power Penalties
  9 GRANITE    L.Granite\Little Goose\L.Monumental\Ice Harbor Non-Power Penalties
```

{Print List of Reservoirs}

```
 3 Reservoirs in this system

  # ID          Link ID   HPE Link        STO1         STOK         STOT
  1 DWORSHAK_P       6          7      3468.00      0.10000      3468.00
  2 BROWNLEE_P      12         13      1426.70      0.10000      1426.70
  3 GRANITE_P       16         17      1825.00      0.10000      1825.00
```

{Print list of each node and the Incoming and Outgoing Links to that Node}

```
List of Links connecting each node


 Node: S_SOURCE
 -------------------------------------------------------------------------------
                   Incoming Links
                        Link # Type        From
                        ------------------------------------------------------
                   Outgoing Links
                        Link # Type        To
                        ------------------------------------------------------
                             1 DIVR        SINK
                             2 STO1        DWORSHAK_P
```

```
                   3 STO1        BROWNLEE_P
                   4 STO1        GRANITE_P
                   5 INFL        DWORSHAK_P
                   9 INFL        SPALDING
                  11 INFL        BROWNLEE_P
                  15 INFL        GRANITE_P

Node: SINK
-------------------------------------------------------------------------------
                   Incoming Links
                        Link # Type        From
                        ----------------------------------------------------
                             1 DIVR        S_SOURCE
                            18 RREL        GRANITE
                   Outgoing Links
                        Link # Type        To
                        ----------------------------------------------------

Node: DWORSHAK_P   Dworshak Reservoir Power Penalties
-------------------------------------------------------------------------------
                   Incoming Links
                        Link # Type        From
                        ----------------------------------------------------
                             2 STO1        S_SOURCE
                             5 INFL        S_SOURCE
                             6 RSTO        DWORSHAK_P
                   Outgoing Links
                        Link # Type        To
                        ----------------------------------------------------
                             6 RSTO        DWORSHAK_P
                             7 HREL        DWORSHAK

Node: DWORSHAK     Dworshak Reservoir Non-Power Penalties
-------------------------------------------------------------------------------
                   Incoming Links
                        Link # Type        From
                        ----------------------------------------------------
                             7 HREL        DWORSHAK_P
                   Outgoing Links
                        Link # Type        To
                        ----------------------------------------------------
                             8 RREL        SPALDING

Node: SPALDING     Spalding
-------------------------------------------------------------------------------
                   Incoming Links
                        Link # Type        From
                        ----------------------------------------------------
                             8 RREL        DWORSHAK
                             9 INFL        S_SOURCE
                   Outgoing Links
                        Link # Type        To
                        ----------------------------------------------------
                            10 CHAN        GRANITE_P

Node: BROWNLEE_P   Brownlee, Oxbow, and Hells Canyon Power Penalties
-------------------------------------------------------------------------------
                   Incoming Links
                        Link # Type        From
                        ----------------------------------------------------
                             3 STO1        S_SOURCE
                            11 INFL        S_SOURCE
                            12 RSTO        BROWNLEE_P
                   Outgoing Links
                        Link # Type        To
                        ----------------------------------------------------
                            12 RSTO        BROWNLEE_P
                            13 HREL        BROWNLEE
```

```
Node: BROWNLEE     Brownlee, Oxbow, and Hells Canyon Non-Power Penalties
-------------------------------------------------------------------------------
                   Incoming Links
                        Link # Type        From
                        -----------------------------------------------------
                            13 HREL        BROWNLEE_P
                   Outgoing Links
                        Link # Type        To
                        -----------------------------------------------------
                            14 RREL        GRANITE_P

Node: GRANITE_P    L.Granite\Little Goose\L.Monumental\Ice Harbor Power Penalties
-------------------------------------------------------------------------------
                   Incoming Links
                        Link # Type        From
                        -----------------------------------------------------
                             4 STO1        S_SOURCE
                            10 CHAN        SPALDING
                            14 RREL        BROWNLEE
                            15 INFL        S_SOURCE
                            16 RSTO        GRANITE_P
                   Outgoing Links
                        Link # Type        To
                        -----------------------------------------------------
                            16 RSTO        GRANITE_P
                            17 HREL        GRANITE

Node: GRANITE      L.Granite\Little Goose\L.Monumental\Ice Harbor Non-Power Penalties
-------------------------------------------------------------------------------
                   Incoming Links
                        Link # Type        From
                        -----------------------------------------------------
                            17 HREL        GRANITE_P
                   Outgoing Links
                        Link # Type        To
                        -----------------------------------------------------
                            18 RREL        SINK
```

{Print a list of all Links in the System, along with Gains, Constant Costs, and Negative Inflows }

```
    # From       To         Type     Mult    Cost Sum Neg Num Neg Description

    1 S_SOURCE   SINK       DIVR  1.00000    0.00     0.0       0 Continuity Link
    2 S_SOURCE   DWORSHAK_P STO1  1.00000    0.00     0.0       0 Starting Storage for DWORSHAK_P
    3 S_SOURCE   BROWNLEE_P STO1  1.00000    0.00     0.0       0 Starting Storage for BROWNLEE_P
    4 S_SOURCE   GRANITE_P  STO1  1.00000    0.00     0.0       0 Starting Storage for GRANITE_P
    5 S_SOURCE   DWORSHAK_P INFL  1.00000    0.00     0.0       0 Inflow to Dworshak Reservoir
    6 DWORSHAK_P DWORSHAK_P RSTO  1.00000    0.00     0.0       0 Storage in Dworshak Reservoir
    7 DWORSHAK_P DWORSHAK   HREL  1.00000    0.00     0.0       0 Power Release from Dworshak
Reservoir
    8 DWORSHAK   SPALDING   RREL  1.00000    0.00     0.0       0 Other Releases from Dworshak
Reservoir to Spalding
    9 S_SOURCE   SPALDING   INFL  1.00000    0.00     0.0       0 Inflow to Spalding
   10 SPALDING   GRANITE_P  CHAN  1.00000    0.00     0.0       0 Channel from Spalding to
Granite/Goose/Monumental/Ice Harbor
   11 S_SOURCE   BROWNLEE_P INFL  1.00000    0.00     0.0       0 Inflow to Brownlee, Oxbow &
Hells Canyon Reservoirs
   12 BROWNLEE_P BROWNLEE_P RSTO  1.00000    0.00     0.0       0 Storage in Brownlee, Oxbow &
Hells Canyon Reservoirs
   13 BROWNLEE_P BROWNLEE   HREL  1.00000    0.00     0.0       0 Power Release from Brownlee
Reservoir
   14 BROWNLEE   GRANITE_P  RREL  1.00000    0.00     0.0       0 Other Releases from
Brownlee/Oxbow/Hells Canyon to Granite/...
   15 S_SOURCE   GRANITE_P  INFL  1.00000    0.00     0.0       0 Inflow to Lower Granite thru
Ice Harbor Reservoirs
   16 GRANITE_P  GRANITE_P  RSTO  1.00000    0.00     0.0       0 Storage in Granite Reservoir
   17 GRANITE_P  GRANITE    HREL  1.00000    0.00     0.0       0 Power Release from Granite
Reservoir
```

```
   18 GRANITE    SINK       RREL  1.00000    0.00     0.0       0 Release from
Granite/Goose/Monumental/Ice Harbor to Sink

 User input read and printed.

 March     31, 1994   08:48:48

 Elapsed Time is         6 seconds or

                     0.10 minutes or

                    0.002 hours.
```

{Begin to Solve the Problem}

```
Begin first attempt at solution.
 March     31, 1994   08:48:49
 Elapsed Time is         7 seconds or
                     0.12 minutes or
                    0.002 hours.
MIN COST =  0.100E-06 MAX COST =  0.149E+05
MIN CAP. =  0.250E+02 MAX CAP. =  0.100E+07

TREINT : DURATION (SEC.) =     1.00; ELAPSED TIME (SEC.) =     1.00

PHASE_1: DURATION (SEC.) =    49.00; ELAPSED TIME (SEC.) =    50.00

PRINTING FROM SET_STATUS
STATUS: PHASE_1; STEP SIZE= 0.400E+03
CANDIDATE: CALLS=         0; TOTAL=         0; SUCCESSFUL=         0
FIRST SELECT=      8706; ALTER. SELECT=         0
TOTAL ITERATIONS=      8707
COST MULT.=  0.6700344E-04; CAP. MULT.=  0.4000000E-01
P_SWITCH =     1.000; I_SWITCH =         1
     The total infeasibility is  0.000E+00

     Continuing with the optimization.

PHASE_2: DURATION (SEC.) =     1.00; ELAPSED TIME (SEC.) =    51.00

PRINTING FROM SET_STATUS
STATUS: PHASE_2; STEP SIZE= 0.400E+03
CANDIDATE: CALLS=         0; TOTAL=         0; SUCCESSFUL=         0
FIRST SELECT=      8746; ALTER. SELECT=         0
TOTAL ITERATIONS=      8748
COST MULT.=  0.1675086E+05; CAP. MULT.=  0.4000000E-01
P_SWITCH =     1.000; I_SWITCH =         1

OPTIMAL: DURATION (SEC.) =     0.00; ELAPSED TIME (SEC.) =    51.00

 Conclude first solution; Check hydropower.
 March     31, 1994   08:49:40
 Elapsed Time is        58 seconds or
                     0.97 minutes or
                    0.016 hours.



 Total Penalty at zero flow..........     64699774.78             64699774.8
 Fixed Cost..........................    0.0000000000                    0.0
 Cost in the arcs (network cost).....    -56978804.45            -56978804.4
 Fixed Cost Cumulative New...........    -8049900.698             -8049900.7
 Total Cost..........................     15770871.02             15770871.0


 Hydropower checked, cost computed, min. cost solution written to disk.
 March     31, 1994   08:49:42
 Elapsed Time is        60 seconds or
                     1.00 minutes or
                    0.017 hours.
```

```
________________________________________________________________________________
________________________________________________________________________________
 Start solution for hydropower iteration    1.
________________________________________________________________________________
________________________________________________________________________________
________________________________________________________________________________
 March     31, 1994   08:49:42
 Elapsed Time is        60 seconds or
                      1.00 minutes or
                     0.017 hours.
MIN COST =  0.100E-06 MAX COST =  0.149E+05
MIN CAP. =  0.500E+00 MAX CAP. =  0.100E+07

TREINT : DURATION (SEC.) =     1.00; ELAPSED TIME (SEC.) =     1.00

PHASE_1: DURATION (SEC.) =    16.00; ELAPSED TIME (SEC.) =    17.00

PRINTING FROM SET_STATUS
STATUS: PHASE_1; STEP SIZE= 0.200E+05
CANDIDATE: CALLS=         0; TOTAL=         0; SUCCESSFUL=         0
FIRST SELECT=      2669; ALTER. SELECT=         0
TOTAL ITERATIONS=      2670
COST MULT.=  0.6700344E-04; CAP. MULT.=  0.2000000E+01
P_SWITCH =     1.000; I_SWITCH =         1
     The total infeasibility is  0.196E-09

     Continuing with the optimization.

PHASE_2: DURATION (SEC.) =     0.00; ELAPSED TIME (SEC.) =    17.00

PRINTING FROM SET_STATUS
STATUS: PHASE_2; STEP SIZE= 0.200E+05
CANDIDATE: CALLS=         0; TOTAL=         0; SUCCESSFUL=         0
FIRST SELECT=      2678; ALTER. SELECT=         0
TOTAL ITERATIONS=      2680
COST MULT.=  0.1675086E+05; CAP. MULT.=  0.2000000E+01
P_SWITCH =     1.000; I_SWITCH =         1

OPTIMAL: DURATION (SEC.) =     0.00; ELAPSED TIME (SEC.) =    17.00

 Solution completed, Check hydropower.
 March     31, 1994   08:49:59
 Elapsed Time is        77 seconds or
                      1.28 minutes or
                     0.021 hours.

 Hydropower check completed, compute cost.
 March     31, 1994   08:50:00
 Elapsed Time is        78 seconds or
                      1.30 minutes or
                     0.022 hours.



 Total Penalty at zero flow..........     64705572.96              64705573.0
 Fixed Cost..........................    0.0000000000                     0.0
 Cost in the arcs (network cost).....    -52051879.46             -52051879.5
 Fixed Cost Cumulative New...........    -3146762.772              -3146762.8
 Total Cost..........................     15800456.27              15800456.3



 Cost computed, read min. cost soln.
 March     31, 1994   08:50:00
 Elapsed Time is        78 seconds or
                      1.30 minutes or
                     0.022 hours.

 Check hydropower for previous min. cost solution.
 March     31, 1994   08:50:01
 Elapsed Time is        79 seconds or
                      1.32 minutes or
                     0.022 hours.
```

```
 Previous solution read in; re-iterate.
 March     31, 1994   08:50:02
 Elapsed Time is        80 seconds or
                      1.33 minutes or
                     0.022 hours.

 ______________________________________________________________________________
 ______________________________________________________________________________
 ______________________________________________________________________________
 Start solution for hydropower iteration    2.
 ______________________________________________________________________________
 ______________________________________________________________________________
 ______________________________________________________________________________
 March     31, 1994   08:50:02
 Elapsed Time is        80 seconds or
                      1.33 minutes or
                     0.022 hours.
MIN COST =  0.100E-06 MAX COST =  0.149E+05
MIN CAP. =  0.470E+00 MAX CAP. =  0.100E+07

TREINT : DURATION (SEC.) =     0.00; ELAPSED TIME (SEC.) =     0.00

PHASE_1: DURATION (SEC.) =    16.00; ELAPSED TIME (SEC.) =    16.00

PRINTING FROM SET_STATUS
STATUS: PHASE_1; STEP SIZE= 0.213E+05
CANDIDATE: CALLS=         0; TOTAL=         0; SUCCESSFUL=         0
FIRST SELECT=      2684; ALTER. SELECT=        0
TOTAL ITERATIONS=      2685
COST MULT.=  0.6700344E-04; CAP. MULT.=  0.2128898E+01
P_SWITCH =     1.000; I_SWITCH =         1
     The total infeasibility is  0.215E-09

     Continuing with the optimization.

PHASE_2: DURATION (SEC.) =     1.00; ELAPSED TIME (SEC.) =    17.00

PRINTING FROM SET_STATUS
STATUS: PHASE_2; STEP SIZE= 0.213E+05
CANDIDATE: CALLS=         0; TOTAL=         0; SUCCESSFUL=         0
FIRST SELECT=      2689; ALTER. SELECT=        0
TOTAL ITERATIONS=      2691
COST MULT.=  0.1675086E+05; CAP. MULT.=  0.2128898E+01
P_SWITCH =     1.000; I_SWITCH =         1

OPTIMAL: DURATION (SEC.) =     0.00; ELAPSED TIME (SEC.) =    17.00

 Solution completed, Check hydropower.
 March     31, 1994   08:50:19
 Elapsed Time is        97 seconds or
                      1.62 minutes or
                     0.027 hours.

 Hydropower check completed, compute cost.
 March     31, 1994   08:50:20
 Elapsed Time is        98 seconds or
                      1.63 minutes or
                     0.027 hours.


 Total Penalty at zero flow..........     64705572.96              64705573.0
 Fixed Cost.........................    0.0000000000                     0.0
 Cost in the arcs (network cost).....    -52062854.51             -52062854.5
 Fixed Cost Cumulative New...........    -3157432.070              -3157432.1
 Total Cost..........................     15800150.53              15800150.5


 Cost computed, read min. cost soln.
 March     31, 1994   08:50:20
 Elapsed Time is        98 seconds or
                      1.63 minutes or
                     0.027 hours.
```

```
 Check hydropower for previous min. cost solution.
 March     31, 1994   08:50:21
 Elapsed Time is        99 seconds or
                     1.65 minutes or
                    0.027 hours.

 Previous solution read in; re-iterate.
 March     31, 1994   08:50:22
 Elapsed Time is       100 seconds or
                     1.67 minutes or
                    0.028 hours.


 ________________________________________________________________________________
 ________________________________________________________________________________
 ________________________________________________________________________________
 Start solution for hydropower iteration    3.
 ________________________________________________________________________________
 ________________________________________________________________________________
 ________________________________________________________________________________
 March     31, 1994   08:50:22
 Elapsed Time is       100 seconds or
                     1.67 minutes or
                    0.028 hours.
MIN COST =  0.100E-06 MAX COST =  0.149E+05
MIN CAP. =  0.902E+00 MAX CAP. =  0.100E+07

TREINT : DURATION (SEC.) =     0.00; ELAPSED TIME (SEC.) =     0.00

PHASE_1: DURATION (SEC.) =    14.00; ELAPSED TIME (SEC.) =    14.00

PRINTING FROM SET_STATUS
STATUS: PHASE_1; STEP SIZE= 0.111E+05
CANDIDATE: CALLS=         0; TOTAL=         0; SUCCESSFUL=         0
FIRST SELECT=      2694; ALTER. SELECT=         0
TOTAL ITERATIONS=      2695
COST MULT.=  0.6700344E-04; CAP. MULT.=  0.1108825E+01
P_SWITCH =     1.000; I_SWITCH =         1
     The total infeasibility is  0.113E-09

     Continuing with the optimization.

PHASE_2: DURATION (SEC.) =     1.00; ELAPSED TIME (SEC.) =    15.00

PRINTING FROM SET_STATUS
STATUS: PHASE_2; STEP SIZE= 0.111E+05
CANDIDATE: CALLS=         0; TOTAL=         0; SUCCESSFUL=         0
FIRST SELECT=      2697; ALTER. SELECT=         0
TOTAL ITERATIONS=      2699
COST MULT.=  0.1675086E+05; CAP. MULT.=  0.1108825E+01
P_SWITCH =     1.000; I_SWITCH =         1

OPTIMAL: DURATION (SEC.) =     0.00; ELAPSED TIME (SEC.) =    15.00

 Solution completed, Check hydropower.
 March     31, 1994   08:50:37
 Elapsed Time is       115 seconds or
                     1.92 minutes or
                    0.032 hours.

 Hydropower check completed, compute cost.
 March     31, 1994   08:50:38
 Elapsed Time is       116 seconds or
                     1.93 minutes or
                    0.032 hours.


 Total Penalty at zero flow..........     64705572.96              64705573.0
 Fixed Cost..........................    0.0000000000                     0.0
 Cost in the arcs (network cost).....    -52061713.30             -52061713.3
```

```
 Fixed Cost Cumulative New...........    -3155773.160              -3155773.2
 Total Cost..........................     15799632.83              15799632.8



 Cost computed, read min. cost soln.
 March     31, 1994   08:50:38
 Elapsed Time is       116 seconds or
                      1.93 minutes or
                     0.032 hours.
 Check hydropower for previous min. cost solution.
 March     31, 1994   08:50:39
 Elapsed Time is       117 seconds or
                      1.95 minutes or
                     0.032 hours.

 Previous solution read in; re-iterate.
 March     31, 1994   08:50:39
 Elapsed Time is       117 seconds or
                      1.95 minutes or
                     0.032 hours.



 ______________________________________________________________________________
 ______________________________________________________________________________
 ______________________________________________________________________________
 Start solution for hydropower iteration    4.
 ______________________________________________________________________________
 ______________________________________________________________________________
 ______________________________________________________________________________
 March     31, 1994   08:50:39
 Elapsed Time is       117 seconds or
                      1.95 minutes or
                     0.032 hours.
MIN COST =  0.100E-06 MAX COST =  0.149E+05
MIN CAP. =  0.565E+00 MAX CAP. =  0.100E+07

TREINT : DURATION (SEC.) =     1.00; ELAPSED TIME (SEC.) =     1.00

PHASE_1: DURATION (SEC.) =    15.00; ELAPSED TIME (SEC.) =    16.00

PRINTING FROM SET_STATUS
STATUS: PHASE_1; STEP SIZE= 0.177E+05
CANDIDATE: CALLS=         0; TOTAL=         0; SUCCESSFUL=         0
FIRST SELECT=      2767; ALTER. SELECT=        0
TOTAL ITERATIONS=      2768
COST MULT.=  0.6700344E-04; CAP. MULT.=  0.1769330E+01
P_SWITCH =     1.000; I_SWITCH =         1
     The total infeasibility is  0.175E-09

     Continuing with the optimization.

PHASE_2: DURATION (SEC.) =     0.00; ELAPSED TIME (SEC.) =    16.00

PRINTING FROM SET_STATUS
STATUS: PHASE_2; STEP SIZE= 0.177E+05
CANDIDATE: CALLS=         0; TOTAL=         0; SUCCESSFUL=         0
FIRST SELECT=      2768; ALTER. SELECT=        0
TOTAL ITERATIONS=      2770
COST MULT.=  0.1675086E+05; CAP. MULT.=  0.1769330E+01
P_SWITCH =     1.000; I_SWITCH =         1

OPTIMAL: DURATION (SEC.) =     0.00; ELAPSED TIME (SEC.) =    16.00

 Solution completed, Check hydropower.
 March     31, 1994   08:50:55
 Elapsed Time is       133 seconds or
                      2.22 minutes or
                     0.037 hours.

 Hydropower check completed, compute cost.
 March     31, 1994   08:50:56
 Elapsed Time is       134 seconds or
```

```
                      2.23 minutes or
                     0.037 hours.



 Total Penalty at zero flow..........     64705572.96              64705573.0
 Fixed Cost..........................    0.0000000000                     0.0
 Cost in the arcs (network cost).....    -52049048.06             -52049048.1
 Fixed Cost Cumulative New...........    -3146206.944              -3146206.9
 Total Cost..........................     15802731.85              15802731.8



 Cost computed, read min. cost soln.
 March     31, 1994   08:50:56
 Elapsed Time is       134 seconds or
                      2.23 minutes or
                     0.037 hours.
 Check hydropower for previous min. cost solution.
 March     31, 1994   08:50:57
 Elapsed Time is       135 seconds or
                      2.25 minutes or
                     0.038 hours.

 Previous solution read in; re-iterate.
 March     31, 1994   08:50:58
 Elapsed Time is       136 seconds or
                      2.27 minutes or
                     0.038 hours.


 ______________________________________________________________________________
 ______________________________________________________________________________
 ______________________________________________________________________________
 Start solution for hydropower iteration    5.
 ______________________________________________________________________________
 ______________________________________________________________________________
 ______________________________________________________________________________
 March     31, 1994   08:50:58
 Elapsed Time is       136 seconds or
                      2.27 minutes or
                     0.038 hours.
MIN COST =  0.100E-06 MAX COST =  0.149E+05
MIN CAP. =  0.496E+00 MAX CAP. =  0.100E+07

TREINT : DURATION (SEC.) =     0.00; ELAPSED TIME (SEC.) =     0.00

PHASE_1: DURATION (SEC.) =    16.00; ELAPSED TIME (SEC.) =    16.00

PRINTING FROM SET_STATUS
STATUS: PHASE_1; STEP SIZE= 0.202E+05
CANDIDATE: CALLS=         0; TOTAL=         0; SUCCESSFUL=         0
FIRST SELECT=      2846; ALTER. SELECT=         0
TOTAL ITERATIONS=      2847
COST MULT.=  0.6700344E-04; CAP. MULT.=  0.2016741E+01
P_SWITCH =     1.000; I_SWITCH =         1
     The total infeasibility is  0.205E-09

     Continuing with the optimization.

PHASE_2: DURATION (SEC.) =     0.00; ELAPSED TIME (SEC.) =    16.00

PRINTING FROM SET_STATUS
STATUS: PHASE_2; STEP SIZE= 0.202E+05
CANDIDATE: CALLS=         0; TOTAL=         0; SUCCESSFUL=         0
FIRST SELECT=      2847; ALTER. SELECT=         0
TOTAL ITERATIONS=      2849
COST MULT.=  0.1675086E+05; CAP. MULT.=  0.2016741E+01
P_SWITCH =     1.000; I_SWITCH =         1

OPTIMAL: DURATION (SEC.) =     0.00; ELAPSED TIME (SEC.) =    16.00
```

```
 Solution completed, Check hydropower.
 March     31, 1994   08:51:14
 Elapsed Time is       152 seconds or
                      2.53 minutes or
                     0.042 hours.

 Hydropower check completed, compute cost.
 March     31, 1994   08:51:15
 Elapsed Time is       153 seconds or
                      2.55 minutes or
                     0.043 hours.



 Total Penalty at zero flow..........     64703572.92              64703572.9
 Fixed Cost..........................    0.0000000000                     0.0
 Cost in the arcs (network cost).....    -53137182.28             -53137182.3
 Fixed Cost Cumulative New...........    -4236680.399              -4236680.4
 Total Cost..........................     15803071.05              15803071.0



 Cost computed, read min. cost soln.
 March     31, 1994   08:51:15
 Elapsed Time is       153 seconds or
                      2.55 minutes or
                     0.043 hours.
 Check hydropower for previous min. cost solution.
 March     31, 1994   08:51:16
 Elapsed Time is       154 seconds or
                      2.57 minutes or
                     0.043 hours.

 Previous solution read in; re-iterate.
 March     31, 1994   08:51:17
 Elapsed Time is       155 seconds or
                      2.58 minutes or
                     0.043 hours.

 ______________________________________________________________________________
 ______________________________________________________________________________
 ______________________________________________________________________________
 Start solution for hydropower iteration    6.
 ______________________________________________________________________________
 ______________________________________________________________________________
 ______________________________________________________________________________
 March     31, 1994   08:51:17
 Elapsed Time is       155 seconds or
                      2.58 minutes or
                     0.043 hours.
MIN COST =  0.100E-06 MAX COST =  0.149E+05
MIN CAP. =  0.156E+01 MAX CAP. =  0.100E+07

TREINT : DURATION (SEC.) =     0.00; ELAPSED TIME (SEC.) =     0.00

PHASE_1: DURATION (SEC.) =    16.00; ELAPSED TIME (SEC.) =    16.00

PRINTING FROM SET_STATUS
STATUS: PHASE_1; STEP SIZE= 0.640E+04
CANDIDATE: CALLS=         0; TOTAL=         0; SUCCESSFUL=         0
FIRST SELECT=      2825; ALTER. SELECT=         0
TOTAL ITERATIONS=      2826
COST MULT.=  0.6700344E-04; CAP. MULT.=  0.6400000E+00
P_SWITCH =     1.000; I_SWITCH =         1
     The total infeasibility is  0.684E-10

     Continuing with the optimization.

PHASE_2: DURATION (SEC.) =     0.00; ELAPSED TIME (SEC.) =    16.00

PRINTING FROM SET_STATUS
STATUS: PHASE_2; STEP SIZE= 0.640E+04
```

```
CANDIDATE: CALLS=         0; TOTAL=         0; SUCCESSFUL=         0
FIRST SELECT=      2825; ALTER. SELECT=         0
TOTAL ITERATIONS=      2827
COST MULT.=  0.1675086E+05; CAP. MULT.=  0.6400000E+00
P_SWITCH =     1.000; I_SWITCH =         1

OPTIMAL: DURATION (SEC.) =     0.00; ELAPSED TIME (SEC.) =    16.00

 Solution completed, Check hydropower.
 March     31, 1994   08:51:33
 Elapsed Time is       171 seconds or
                      2.85 minutes or
                     0.047 hours.

 Hydropower check completed, compute cost.
 March     31, 1994   08:51:34
 Elapsed Time is       172 seconds or
                      2.87 minutes or
                     0.048 hours.



 Total Penalty at zero flow..........     64700721.21              64700721.2
 Fixed Cost..........................    0.0000000000                     0.0
 Cost in the arcs (network cost).....    -56470066.19             -56470066.2
 Fixed Cost Cumulative New...........    -7557122.251              -7557122.3
 Total Cost..........................     15787777.27              15787777.3



 Cost computed, read min. cost soln.
 March     31, 1994   08:51:34
 Elapsed Time is       172 seconds or
                      2.87 minutes or
                     0.048 hours.
 Check hydropower for previous min. cost solution.
 March     31, 1994   08:51:35
 Elapsed Time is       173 seconds or
                      2.88 minutes or
                     0.048 hours.

 Previous solution read in; re-iterate.
 March     31, 1994   08:51:35
 Elapsed Time is       173 seconds or
                      2.88 minutes or
                     0.048 hours.

Number of Iterations:        7          Minimum Cost:  0.157708710249E+08         15770871.02
  Iteration          Total Cost          Total Cost     Alpha
         1        15770871.02  0.157708710249E+08   0.50000
         2        15800456.27  0.158004562706E+08   0.50000
         3        15800150.53  0.158001505278E+08   0.25000
         4        15799632.83  0.157996328266E+08   0.12500
         5        15802731.85  0.158027318482E+08   0.06250
         6        15803071.05  0.158030710457E+08   0.03125
         7        15787777.27  0.157877772659E+08   0.01563
```

{Write Results to the Output DSS Data File}

```
 Solution complete, Store results in DSS.
 March     31, 1994   08:51:36
 Elapsed Time is       174 seconds or
                      2.90 minutes or
                     0.048 hours.
 -----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/FLOW(KAF)/01JAN1920/1MON/SAMPLE5/
 -----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/FLOW(KAF)/01JAN1930/1MON/SAMPLE5/
 -----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/FLOW(KAF)/01JAN1940/1MON/SAMPLE5/
 -----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/PCT_EXCEED_MON-FLOW(KAF)///SAMPLE5/
 -----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/PCT_EXCEED_TOT-FLOW(KAF)///SAMPLE5/
 -----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/FLOW/01JAN1920/1MON/SAMPLE5/
 -----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/FLOW/01JAN1930/1MON/SAMPLE5/
 -----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/FLOW/01JAN1940/1MON/SAMPLE5/
 -----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/PCT_EXCEED_MON-FLOW///SAMPLE5/
 -----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/PCT_EXCEED_TOT-FLOW///SAMPLE5/
```

```
-----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/FLOW_IN(KAF)/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/FLOW_IN(KAF)/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/FLOW_IN(KAF)/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/FLOW_IN/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/FLOW_IN/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/FLOW_IN/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/FLOW_NAT(KAF)/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/FLOW_NAT(KAF)/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/FLOW_NAT(KAF)/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/FLOW_NAT/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/FLOW_NAT/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //S_SOURCE/FLOW_NAT/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/FLOW(KAF)/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/FLOW(KAF)/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/FLOW(KAF)/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/PCT_EXCEED_MON-FLOW(KAF)///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/PCT_EXCEED_TOT-FLOW(KAF)///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/FLOW/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/FLOW/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/FLOW/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/PCT_EXCEED_MON-FLOW///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/PCT_EXCEED_TOT-FLOW///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/FLOW_IN(KAF)/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/FLOW_IN(KAF)/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/FLOW_IN(KAF)/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/FLOW_IN/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/FLOW_IN/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/FLOW_IN/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/FLOW_NAT(KAF)/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/FLOW_NAT(KAF)/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/FLOW_NAT(KAF)/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/FLOW_NAT/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/FLOW_NAT/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SINK/FLOW_NAT/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/STOR/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/STOR/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/STOR/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/PCT_EXCEED_MON-STOR///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/PCT_EXCEED_TOT-STOR///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/FLOW(KAF)/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/FLOW(KAF)/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/FLOW(KAF)/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/PCT_EXCEED_MON-FLOW(KAF)///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/PCT_EXCEED_TOT-FLOW(KAF)///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/FLOW/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/FLOW/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/FLOW/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/PCT_EXCEED_MON-FLOW///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/PCT_EXCEED_TOT-FLOW///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/FLOW_IN(KAF)/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/FLOW_IN(KAF)/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/FLOW_IN(KAF)/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/FLOW_IN/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/FLOW_IN/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/FLOW_IN/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/FLOW_NAT(KAF)/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/FLOW_NAT(KAF)/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/FLOW_NAT(KAF)/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/FLOW_NAT/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/FLOW_NAT/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //DWORSHAK_P/FLOW_NAT/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/FLOW(KAF)/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/FLOW(KAF)/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/FLOW(KAF)/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/PCT_EXCEED_MON-FLOW(KAF)///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/PCT_EXCEED_TOT-FLOW(KAF)///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/FLOW/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/FLOW/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/FLOW/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/PCT_EXCEED_MON-FLOW///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/PCT_EXCEED_TOT-FLOW///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/FLOW_IN(KAF)/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/FLOW_IN(KAF)/01JAN1930/1MON/SAMPLE5/
```

```
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/FLOW_IN(KAF)/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/FLOW_IN/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/FLOW_IN/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/FLOW_IN/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/FLOW_NAT(KAF)/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/FLOW_NAT(KAF)/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/FLOW_NAT(KAF)/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/FLOW_NAT/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/FLOW_NAT/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //SPALDING/FLOW_NAT/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/STOR/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/STOR/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/STOR/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/PCT_EXCEED_MON-STOR///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/PCT_EXCEED_TOT-STOR///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/FLOW(KAF)/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/FLOW(KAF)/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/FLOW(KAF)/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/PCT_EXCEED_MON-FLOW(KAF)///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/PCT_EXCEED_TOT-FLOW(KAF)///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/FLOW/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/FLOW/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/FLOW/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/PCT_EXCEED_MON-FLOW///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/PCT_EXCEED_TOT-FLOW///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/FLOW_IN(KAF)/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/FLOW_IN(KAF)/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/FLOW_IN(KAF)/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/FLOW_IN/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/FLOW_IN/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/FLOW_IN/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/FLOW_NAT(KAF)/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/FLOW_NAT(KAF)/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/FLOW_NAT(KAF)/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/FLOW_NAT/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/FLOW_NAT/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //BROWNLEE_P/FLOW_NAT/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/STOR/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/STOR/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/STOR/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/PCT_EXCEED_MON-STOR///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/PCT_EXCEED_TOT-STOR///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/FLOW(KAF)/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/FLOW(KAF)/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/FLOW(KAF)/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/PCT_EXCEED_MON-FLOW(KAF)///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/PCT_EXCEED_TOT-FLOW(KAF)///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/FLOW/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/FLOW/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/FLOW/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/PCT_EXCEED_MON-FLOW///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/PCT_EXCEED_TOT-FLOW///SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/FLOW_IN(KAF)/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/FLOW_IN(KAF)/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/FLOW_IN(KAF)/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/FLOW_IN/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/FLOW_IN/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/FLOW_IN/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/FLOW_NAT(KAF)/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/FLOW_NAT(KAF)/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/FLOW_NAT(KAF)/01JAN1940/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/FLOW_NAT/01JAN1920/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/FLOW_NAT/01JAN1930/1MON/SAMPLE5/
-----DSS---ZWRITE Unit  73; Vers.    1:  //GRANITE_P/FLOW_NAT/01JAN1940/1MON/SAMPLE5/

Results stored in DSS, Print network.
March     31, 1994   08:51:39
Elapsed Time is        177 seconds or
                     2.95 minutes or
                    0.049 hours.
```

{Print summary information about this run}

```
 Number of arcs to process:      12225

    Arc # From Node    To Node       Date                          Unit Cost    Amp.         Lower
Bound         Upper Bound              Flow


 Node Information, number of nodes:     1683


 Print any Error Messages for Nodes (Always nodes 1 & 2)
   Ixnode Idnode                e(ixnode)          bt(ixnode)  Period
        1 S_SOURCE          -0.6238538E+06       0.2495415E+05
        2 SINK               0.0000000E+00       0.5988996E+06

 Network printed, compute cost.
 March     31, 1994   08:51:39
 Elapsed Time is       177 seconds or
                     2.95 minutes or
                    0.049 hours.

 Total Penalty at zero flow..........     64699774.78             64699774.8
 Fixed Cost..........................    0.0000000000                    0.0
 Cost in the arcs (network cost).....    -56978804.45            -56978804.4
 Fixed Cost Cumulative New...........    -8049900.698             -8049900.7
 Total Cost..........................     15770871.02             15770871.0


 Status at completion of job.............OPTIMAL
 Number of solver iterations.............      2827
 Number of nodes.........................      1682
 Number of arcs..........................     12221


    -----DSS---ZCLOSE Unit:  71,   File: SAMPLE5T.DSS
               Pointer Utilization:  0.25
               Number of Records:     24
               File Size:     35.8  Kbytes
               Percent Inactive:   0.0
    -----DSS---ZCLOSE Unit:  72,   File: SAMPLE5P.DSS
               Pointer Utilization:  0.32
               Number of Records:    184
               File Size:    211.7  Kbytes
               Percent Inactive:   0.5
    -----DSS---ZCLOSE Unit:  73,   File: SOLN\SAMPLE5.DSS
               Pointer Utilization:  0.35
               Number of Records:    280
               File Size:    282.9  Kbytes
               Percent Inactive:   0.0
 End-of-Run
 March     31, 1994   08:51:39
 Elapsed Time is       177 seconds or
                     2.95 minutes or
                    0.049 hours.
```

CHAPTER 6 - Description of Sample Data for HEC-PRM

Introduction

The sample data sets illustrate eight different operations. Some examples are application-specific. For example, Sample 1 is the application of the program RDHYSR to read data specific to the North Pacific Division, Corps of Engineers. Another sample data set is the execution of the post-processor "PRMPP". This sample requires the user to perform all of the operations since this program is used only in an interactive environment. The Sample Data Sets included with HEC-PRM are:

Sample 1	Read natural flow data formatted in NPD's HYSSR format and store in a DSS file using program RDHYSR.
Sample 2	Read local inflow, depletions, and evaporation data formatted in MRD's format and store in a DSS file using program RDATA0.
Sample 3	Compute the incremental local inflow from the "natural" flows stored in a DSS data file using program MATHPK .
Sample 4	Edit the actual penalty function for June in the DSSPD input data file and store the penalty function in a DSS data file using program DSSPD.
Sample 5	Execute HEC-PRM for a 20 year time window.
Sample 6	Execute MPKPRMPP to read the HEC-PRM input file and modify the MATHPK macro file for the current system.
Sample 7	Execute MATHPK to Post-Process results from HEC-PRM.
Sample 8	Display HEC-PRM results using the post-processer PRMPP.
Sample 9	Read results from MRD's post-processor program "V1.exe" and store in DSS Data File.

Each of these sample data sets is described below and the applicable file assignments are shown.

Sample 1 Description

Read "natural flow data" formatted in NPD's HYSSR format and store in a DSS file using program RDHYSR.

File Assignments for Sample 1

Program ....RDHYSR
Input Data.. SAMPLE1.RHI
Output Messages SAMPLE1.OUT
HYSSR Formatted Data SAMPLE1.DAT
Output HECDSS File SAMPLE1.DSS

Default File Assignments

```
RDHYSR: Reads & Stores HYSSR Formatted Data July 22, 1993  (01.00.01)
      UNIT       KEYWORD          *ABREV      **MAX       DEFAULT
        5        INPUT              I           64        RDHYSR.RHI
        6        OUTPUT             O           64        RDHYSR.OUT
        1        HYSR DATA          H           64        QNAT80A.DAT
       NOP       DSSFILE            D           64        NATURAL
   * ABREV - SHORTEST ABBREVIATION ALLOWED FOR KEYWORD
   ** MAX - MAXIMUM # OF CHARACTERS FOR FILENAME (OR STRING)
```

Example Execution:

```
RDHYSR I=SAMPLE1.RHI O=SAMPLE1.OUT H=SAMPLE1.DAT D=SAMPLE1.DSS
```

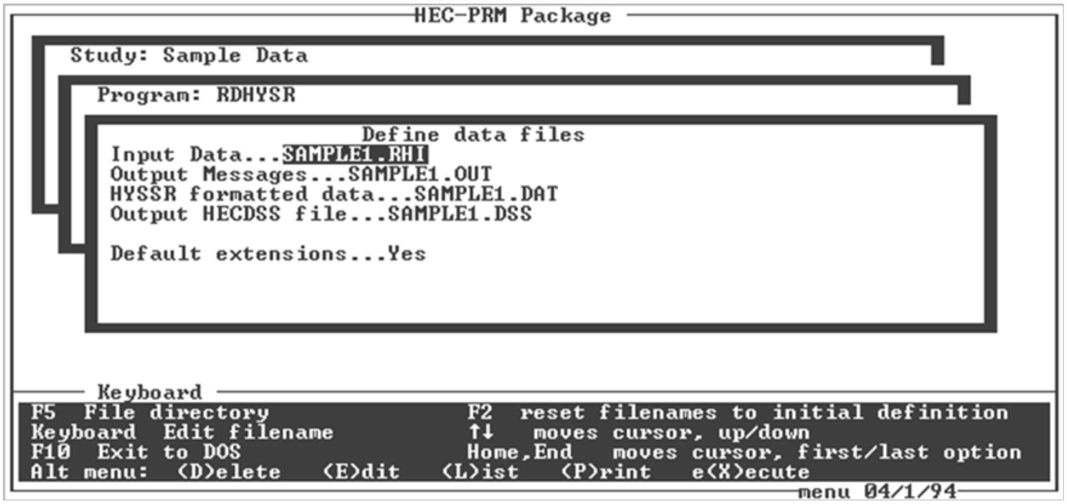

Figure 27. Sample 1 Menu (RDHYSR, Read NPD's Flow Data)

Sample 2 Description

Read local inflow, depletions, and evaporation data formatted in MRD's format and store in a DSS file using program RDATA0.

File Assignments for Sample 2

Program RDATA0
User commands SAMPLE2.D0I
Output Messages SAMPLE2.D0O
Output HECDSS File SAMPLE2.DSS
DODATA file...SAMPLE2.D0D

Default File Assignments

```
   RDATA0 ?
     UNIT       KEYWORD         *ABREV       **        DEFAULT
       5        INPUT             I          64        CON
       6        OUTPUT            O          64        CON
       1        D0DATA            D          64        D0DATA
      NOP       DSSFILE          DS          64        MRD.DSS
* ABREV - SHORTEST ABBREVIATION ALLOWED FOR KEYWORD
** MAX - MAXIMUM # OF CHARACTERS FOR FILENAME (OR STRING)
```

Example Execution:

```
RDATA0 I=SAMPLE2.D0I O=SAMPLE2.D0O D=SAMPLE2.D0D DSS=SAMPLE2.DSS
```

Sample 3 Description

Compute the incremental local inflows from the "natural" flows stored in a DSS data file. This sample uses the equivalent of the flows stored in Sample 1.

File Assignments for Sample 3

Program ....MATHPK
User commands SAMPLE3.MPI
Output Messages SAMPLE3.MPO
Tabulation Output File SAMPLE3.MPT
Macro File SAMPLE3.MAC

Default File Assignments

```
MATHPK ?
MATHPK - Version 2.0.58;  January 18, 1994
      UNIT       KEYWORD          *ABREV      **MAX       DEFAULT
        5        INPUT              I          64         CON
        6        OUTPUT             O          64         CON
        1        1SCRATCH           1          64         SCRATCH.001
        2        2SCRATCH           2          64         SCRATCH.002
       NOP       TAB_FILE           T          64         MPK.MPT
       29        TRACE             TR          64         SCRATCH.009
       30        SCRATCH            S          64         SCRATCH.004
       NOP       FUNFILE            F          64         MPK.FUN
       NOP       MACFILE            M          64         MPK.MAC
       NOP       LOGFILE            L          64         MPK.LOG
       NOP       Buffer Size        B           8         240000
   * ABREV - SHORTEST ABBREVIATION ALLOWED FOR KEYWORD
   ** MAX - MAXIMUM # OF CHARACTERS FOR FILENAME (OR STRING)
```

Example Execution:

```
MATHPKI=SAMPLE3.MPE o=SAMPLE3.MPO T=SAMPLE3.MPT M=SAMPLE3.MAC
```

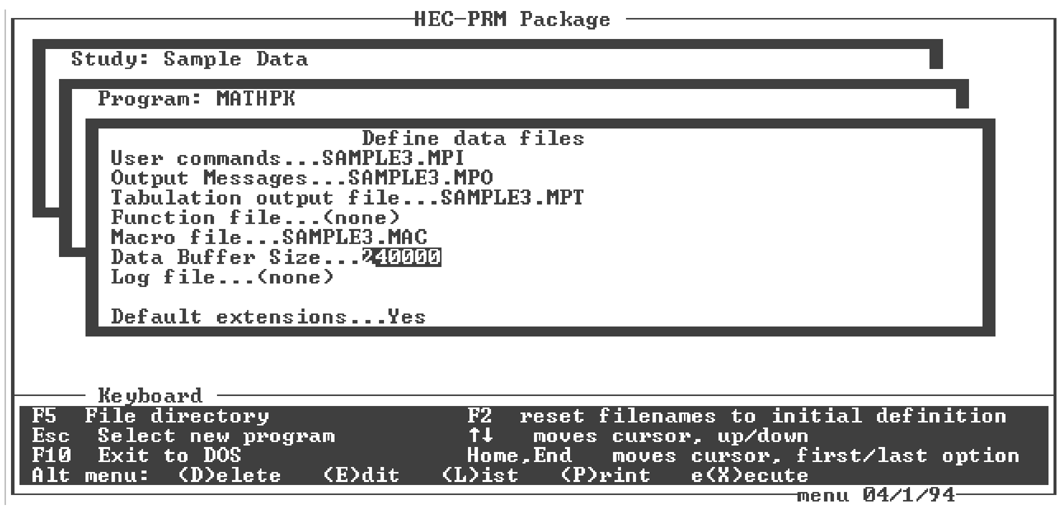

Figure 28. Sample 3 Menu (MATHPK, Compute Incremental Local Flow)

Sample 4 Description

Edit the actual penalty function for June in the DSSPD input data file and store the penalty function in a DSS data file.

File Assignments for Sample 4

Program ....DSSPD
User commands SAMPLE4.PDI
Output Messages SAMPLE4.PDO
HECDSS file SAMPLE4.DSS

Default File Assignments

```
dsspd ?
DSSPD:  Version 3.2.0 ,  August, 1990
      UNIT       KEYWORD          *ABREV      **MAX       DEFAULT
        5        INPUT              I           30        CON
        6        OUTPUT             O           30        CON
       NOP       DSSFILE            D           30
        8        LOG                L           30        SCRATCH.002
   * ABREV - SHORTEST ABBREVIATION ALLOWED FOR KEYWORD
   ** MAX - MAXIMUM # OF CHARACTERS FOR FILENAME (OR STRING)
Stop - Program terminated.
```

Example Execution:

```
DSSPD I=SAMPLE4.PDI O=SAMPLE4.PDO DS=SAMPLE4.DSS
```

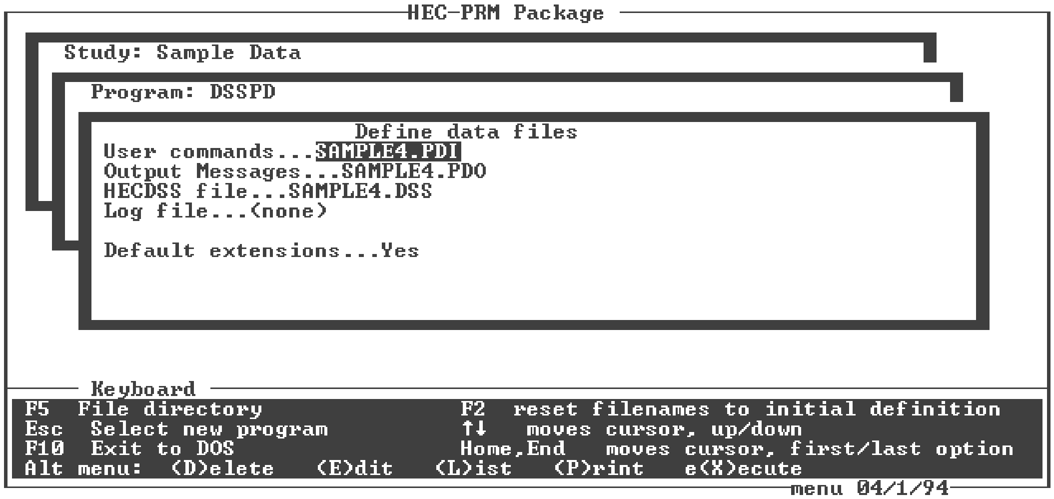

Figure 29. Sample 4 Menu (DSSPD, Stored Edited Penalty Function in DSS Data file)

Sample 5 Description

Execute HEC-PRM for a 20 year time window.

File Assignments for Sample 5

Program ....HECPRM
Input Data SAMPLE5.PRI
Output Information...SAMPLE5.PRO
Results from HEC-PRM (HECDSS file) SAMPLE5.DSS
Input Flow data (HECDSS file) SAMPLE5T.DSS
Input Penalty Functions (HECDSS file)...................SAMPLE5P.DSS

Default File Assignments

```
HECPRM ?
HECPRM: Prescriptive Reservoir Model - Vers. April 1, 1994  ( 1.0)
UNIT      KEYWORD               *ABREV        **MAX   DEFAULT
5               INPUT                       I                 36        CON
6               OUTPUT                      O                 64        CON
NOP          TS_IN_DSS         T            64      SCRATCH.031
NOP          PF_IN_DSS         P            64      SCRATCH.032
NOP          RESULTS_DSS       R            64      SCRATCH.033
NOP          MIN_COST_SOLN     M            64      PRM_MCST.BIN
29           TRACE                 TR       64      SCRATCH.009
1                MSG                                MS                64
C:\HECEXE\SUP\HECPRM.ERR
* ABREV - SHORTEST ABBREVIATION ALLOWED FOR KEYWORD
** MAX - MAXIMUM # OF CHARACTERS FOR FILENAME (OR STRING)
```

Example Execution:

```
HECPRM I=SAMPLE5.PRI O=SAMPLE5.PRO R=SAMPLE5.DSS TS=SAMPLE5T.DSS PF=SAMPLE5P.DSS
```

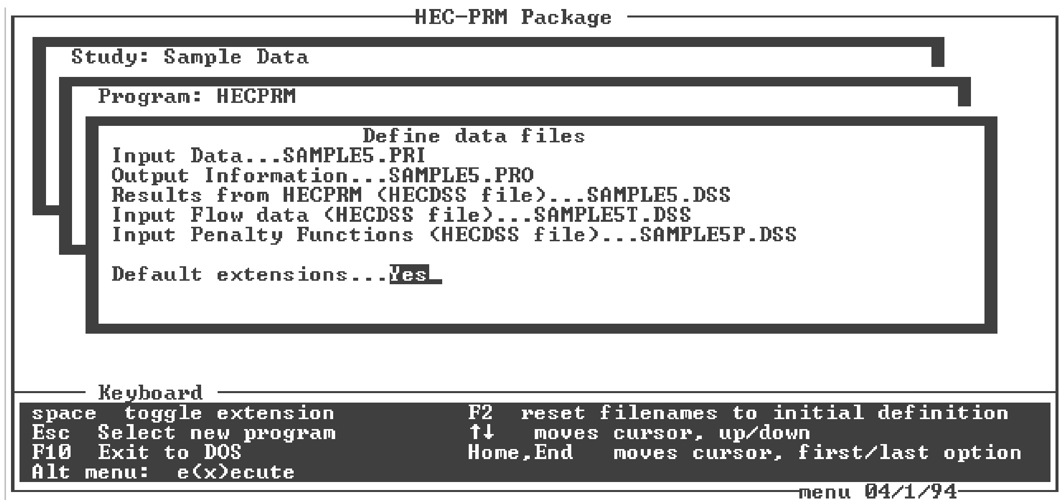

Figure 30. Sample 5 Menu (HEC-PRM, Twenty Year Optimization)

Sample 6 Description

Execute MPKPRMPP to read the HEC-PRM input file and modify the MATHPK macro file for the current system.

File Assignments for Sample 6

Program .... MPKPRMPP
HEC-PRM Input Data File SAMPLE5.PRI
Output Information... CON
MATHPK Macro file SAMPLE6.MAC

Default File Assignments

```
MPKPRMPP ?
MPKPRMPP - Update MATHPK macro file for PRM - ver 1.0.00 (July 20, 1993)
      UNIT       KEYWORD          *ABREV      **MAX       DEFAULT
       5         INPUT              I           64        ALT1.PRI
       6         OUTPUT             O           64        CON
      21         MACRO              M           64        MPKT1.MAC
   * ABREV - SHORTEST ABBREVIATION ALLOWED FOR KEYWORD
   ** MAX - MAXIMUM # OF CHARACTERS FOR FILENAME (OR STRING)
```

Example Execution:

```
MPKPRMPP I=SAMPLE5.PRI M=SAMPLE6.MAC
```

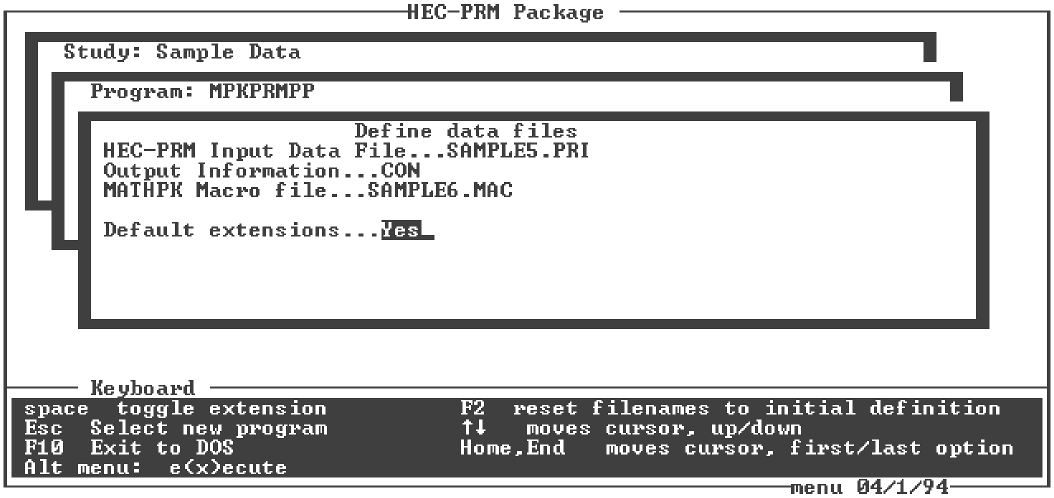

Figure 31. Sample 6 Menu (MPKPRMPP, Modify MATHPK Macro for Post-Processing)

Sample 7 Description

Execute MATHPK to Post-process results from HEC-PRM. (The results from Sample 6 which were written to file SAMPLE6.MAC are copied into "SAMPLE7.MAC". Normally, the same file is used but they are separated here so that input and output are stored separately).

File Assignments for Sample 7

Program .... MATHPK
User commands SAMPLE7.MPI
Output Messages SAMPLE7.MPO
abulation Output File.. SAMPLE7.MPT
MATHPK Macro file SAMPLE7.MAC

Default File Assignments

```
MATHPK ?
MATHPK - Version 2.0.58;  January 18, 1994
UNIT        KEYWORD           *ABREV      **MAX       DEFAULT
5           INPUT               I           64        CON
6           OUTPUT              O           64        CON
1           1SCRATCH            1           64        SCRATCH.001
2           2SCRATCH            2           64        SCRATCH.002
NOP         TAB_FILE            T           64        MPK.MPT
29          TRACE               TR          64        SCRATCH.009
30          SCRATCH             S           64        SCRATCH.004
```

NOP	FUNFILE	F	64	MPK.FUN
NOP	MACFILE	M	64	MPK.MAC
NOP	LOGFILE	L	64	MPK.LOG
NOP	Buffer Size	B	8	240000

* ABREV - SHORTEST ABBREVIATION ALLOWED FOR KEYWORD
** MAX - MAXIMUM # OF CHARACTERS FOR FILENAME (OR STRING)

Example Execution:

```
MATHPK I=SAMPLE7.MPI O=SAMPLE7.MPO T=SAMPLE7.MPT M=SAMPLE7.MAC
```

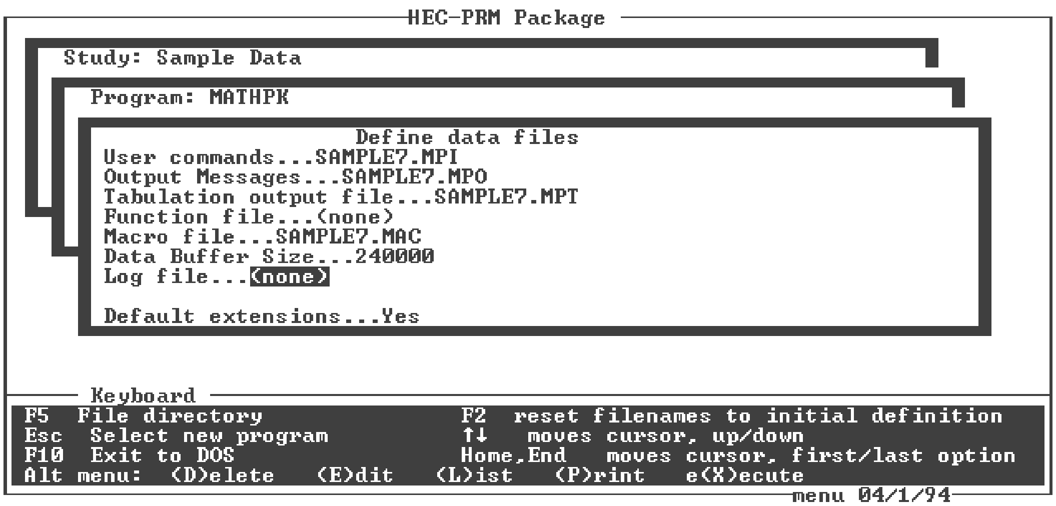

Figure 32. Sample 7 Menu (MATHPK, Post-Process Results)

Sample 8 Description

Display HEC-PRM Results using the post-processer PRMPP.

File Assignments for Sample 8

Program .... PRMPP
HEC-PRM Input Data File SAMPLE8A.PRI
Output Reports SAMPL8A.PPO
Results from HEC-PRM (HECDSS file) SAMPLE8.DSS
Parameter Definition File .... PRMPPDEF.NLS

Default File Assignments

There are no default file assignments for PRMPP.

Example Execution:

```
PRMPP SAMPLE5.PRI SAMPLE5.DSS PRMPPDEF.NLS
```

Sample 9 Description

Read results computed by MRD's simulation program ABS01 and post-processor "V1.EXE" and store the results in a DSS data file. Due to the nature of V1.EXE, it must be executed four times. Each time, a separate set of results are written to separate intermediate binary output files. RDMATF is executed four times to store the results in a DSS Data File.

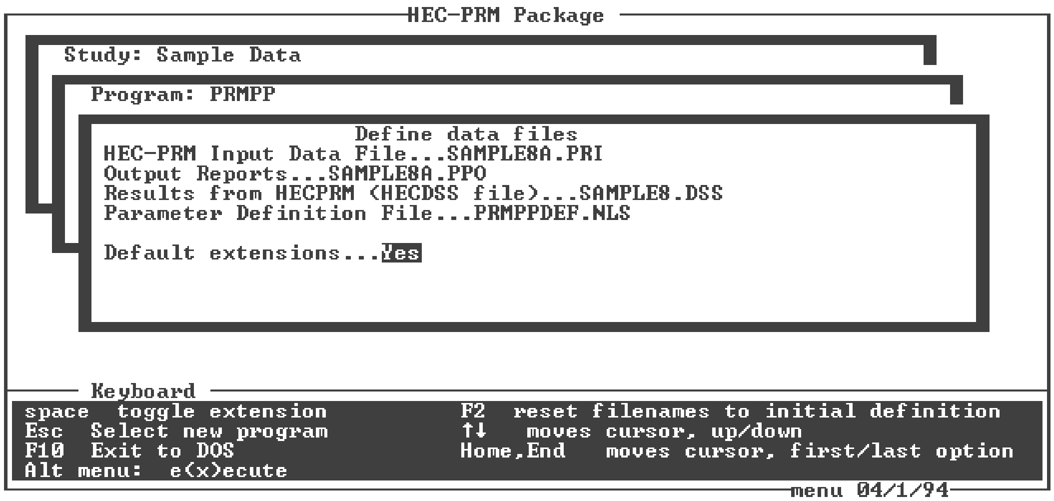

Figure 33. Sample 8 Menu (PRMPP, Display Results)

Typical File Assignments for Sample 9

Program .... RDMATF
User commands ....CON
Output Messages SAMPLE9A.RDO
HECDSS file SAMPLE9.DSS
DMATFILE file SAMPLE9A.MAT

Default File Assignments

```
RDMATF ?
      UNIT       KEYWORD          *ABREV      **MAX      DEFAULT
      5INPUT     I                   64         CON
      6OUTPUT    O                   64         CON
      21         DMATFILE           D             64       DMATFILE
      NOP        DSS_FILE           DS            64       SCRATCH.031
         * ABREV - SHORTEST ABBREVIATION ALLOWED FOR KEYWORD
      ** MAX - MAXIMUM # OF CHARACTERS FOR FILENAME (OR STRING)
```

Example Execution:

```
RDMATF O=SAMPLE9A.RDO DSS=SAMPLE9.DSS DM=SAMPLE9A.MAT
```

CHAPTER 7 - Recommended Pathname Part C

General Recommendations

Currently, the pathname part C is the most critical part for passing data between programs and for use in macros. HEC-PRM naming conventions and recommendations for part C follow. Recommendations/conventions for the other parts are as follows:

Part	Description
A	Leave blank.
B	The maximum 10 character location identifier. Must be unique for each node. For diversions only, part B consists of the "from" node identifier, followed by the dash ("-") character, followed by the "to" node identifier (e.g. S_SOURCE-S_SINK).
C	Recommendations are below.
D	For paired data, generally not used. For time series data, follows standard regular interval time series convention, which is the starting date of the data block (e.g., 01JAN1930).
E	For penalty functions, contains the month or months for which edited functions apply. For time series data, follows standard regular interval time series convention (1MON).
F	An alternative identifier (e.g., ALT1 for time series results) or a qualifier of penalty functions (e.g., CRITERIA A).

HEC-PRM Pathname Part C For Regular Interval Time Series Input and Output Data

Part C	Data Units	Description for Regular Interval Time Series Data
FLOW_IN FLOW_IN(KAF)	cfs kaf	Total inflow to a location. HEC-PRM includes all upstream releases, diversions, and local inflows.
FLOW_LOC FLOW_LOC(KAF)	cfs kaf	Incremental local inflow to location (all inflow above location and below all upstream nodes).
FLOW_NAT FLOW_NAT(KAF)	cfs kaf	Natural Flow at a location. Includes all runoff from drainage area and may be adjusted for a level of development and/or evaporation.
FLOW_DIV FLOW_DIV(KAF)	cfs kaf	Flow diverted from (to) this location. Pathname Part B contains the "from" node and "to" node identifiers (e.g. S_SOURCE-S_SINK).
FLOW FLOW(KAF)	cfs kaf	Reservoir outflow or flow leaving control point.
FLOW_LEAK *FLOW_LEAK(KAF)*	cfs kaf	Flow "leaking" from a link with amplitude (gain factor) not equal to 1.0. Applies to all non-reservoir storage links (e.g. channel, reservoir release, hydropower release links).
EVAP EVAP(KAF)	cfs kaf	Reservoir Pool evaporation. Applies to Reservoir Storage Links only.

Part C	**Data Units**	**Description for Regular Interval Time Series Data**
EVAP_RATE EVAP_RATE(FT)	inches/mo feet/mo	Reservoir Pool evaporation rate. Applies to reservoir storage links only.
STOR	kaf	Reservoir end-of-period (EOP) storage (instantaneous).
STOR_AVG	kaf	Average (AVG) reservoir storage.
ELEV	feet	Reservoir pool elevation (EOP) or channel water surface elevation (AVG).
ENERGY_GEN	mw mwh kmwh	Energy generated.
ENERGY_REQ	mw mwh kmwh	Energy required.
NAV_LOCK	units	Lockages produced.
MARG_COST	K$	Marginal (reduced) cost for non-reservoir storage links. This is the change in the objective function that would result from a unit increase (1 KAF) in link capacity. Applies to non-reservoir storage links.
MARG_COST_S	K$	Marginal (reduced) cost for reservoir storage links.
DUAL_TERM	K$	Dual value (shadow price) for non-reservoir storage arcs. This is the change in the objective function that would result from obtaining an additional unit of flow (1 KAF) at the terminal node of the link.
DUAL_TERM_S	K$	Dual value (shadow price) for reservoir storage arcs.
POWER_CAP	mw	Power capability.
PS_FDU *PS_EDT* *PS_ACT*	K$	Storage penalties (flood damage urban, edited, and actual).
PQ_FDU *PQ_EDT* *PQ_ACT*	K$	Flow penalties (flood damage urban, edited, and actual).
PQ_HPE *PQ_HPA*	K$	Hydropower Energy & Capacity combined penalties (edited and actual).
PS_HCE *PS_HCA*	K$	Hydropower Capacity penalties as a function of storage (edited and actual).
PQ_HCE *PQ_HCA*	K$	Hydropower Capacity penalties as a function of flow (edited and actual).
PQ_HEE *PQ_HEA*	K$	Hydropower energy penalty only (edited and actual; doesn't include capacity).
PT_EDT *PT_ACT*	K$	Total penalty for location. Includes storage, flow, and hydropower penalties.
MON	-	Integer number identifying month (1-12; 1 for January).
CFS_KAF	KAF/CFS	Factor to convert flow in CFS into KAF per month by multiplication. (Conversely, divide to convert KAF into CFS).

Note: *Italics* denote items that can be computed and stored in the HEC-DSS data file by available MATHPK macros.

HEC-PRM Pathname Part C For Paired Function Input and Output Data

Part C	Data Units	Description for Paired Function Data
EL-AR-CAP	feet-kac-kaf	Elevation-area-capacity curve for a reservoir pool.
Q(KAF)-NRG Q-NRG	kaf-mw cfs-mw	Hydropower energy generation family of curves. Each curve represents flow-energy generation for a given storage. Storage is entered as curve label.
Q(KAF)-NAV_LOCK Q-NAV_LOCL	kaf-units cfs-units	Navigation production family of curves. For applications such as the Panama Canal system, each curve represents flow-lockage production for a given storage. Storage is entered as a curve label.
Q(KAF)-CAP Q-CAP	kaf-mw cfs-mw	Hydropower capacity family of curves. Each curve represents flow-capacity for a given storage. Storage is entered as a curve label.
S-P_FDU S-P_WSP S-P_EDT	kaf-k$	Storage penalty functions for urban flood damage, water supply, and composite edited.
Q-P_FDU Q-P_WSP Q-P_EDT Q(KAF)-P_FDU Q(KAF)-P_WSP Q(KAF)-P_EDT	cfs-k$ kaf-k$	Flow penalty functions for urban flood damage, water supply, and composite edited.
Q-P_HPA Q-P_HPE Q(KAF)-P_HPA Q(KAF)-P_HPE	cfs-k$ kaf-k$	Flow-Hydropower penalty functions (actual and edited for combined energy and capacity). Each record contains a family of curves corresponding to different assumed pool storage.
Q-P_NAV Q(KAF)-P_NAV	cfs-k$ kaf-k$	Flow-Navigation penalty functions. For applications such as the Panama Canal system, each record contains a family of curves corresponding to different assumed pool storage.
S-P_HCE *S-P_HCA*	kaf-k$	Storage-Hydropower Capacity penalty functions (edited and actual).
Q-P_HCE *Q-P_HCA* *Q(KAF)-P_HCE* *Q(KAF)-P_HCA*	cfs-k$ kaf-k$	Flow-Hydropower Capacity penalty functions (edited and actual).
Q-P_HEE *Q-P_HEA* *Q(KAF)-P_HEE* *Q(KAF)-P_HEA*	cfs-k$ kaf-k$	Flow-Hydropower Energy penalty functions (edited and actual).

Note: *Italics* denote items that can be computed and stored in the HEC-DSS data file by available MATHPK macros.

APPENDIX A - References

Grygier, J.C., and Stedinger, J.R. (1985). "Algorithms For Optimizing Hydropower System Operation," *Water Resources Research* 21(1), 1-10.

Hadley, G. (1964). *Nonlinear and Dynamic Programming*. Addison-Wesley Publishing, Inc., Reading, MA.

Hitch, C.J., and McKean, R. (1960). *The Economics of Defense in the Nuclear Age*. Harvard University Press, Cambridge, MA.

Jensen, P.A., and Barnes, J.W. (1980). *Network Flow Programming*. John Wiley & Sons, New York, NY.

Jensen, P.A., Bhaumik, G., and Driscoll, W. (1974). "Network Flow Modeling of Multireservoir Distribution Systems," *CRWR-107*, Center for Research in Water Resources, University of Texas, Austin, TX.

Martin, Q.W. (1982). "Multireservoir Simulation and Optimization Model SIM-V," *UM-38*, Texas Department of Water Resources, Austin, TX.

Reznicek, K.K., and Simonovic, S.P. (1990). "An Improved Algorithm For Hydropower Optimization," *Water Resources Research* 26(2), 189-198.

Appendix B - Glossary

ARC Connects two nodes of a network. In network-flow programming, each arc has three parameters: a lower bound, which is the minimal amount that can flow along the arc; an upper bound, which is the maximum amount that can flow along the arc; and a cost for each unit that flows along the arc. Arcs of a generalized network also have an arc multiplier.

CHANNEL-FLOW LINK Represents the flow in a channel reach. A channel-flow link originates at any non-reservoir node and terminates at any network node.

CONSTRAINT Limits the decision variables to their feasible or permissible values.

CONVEX FUNCTION A function $f(X)$ for which the following is true for any two distinct points X_1 and X_2 and for $0 < \lambda < 1$: $f(\lambda X_1 + (1 - \lambda)X_2) < \lambda f(X_1) + (1 - \lambda)f(X_2)$

DECISION VARIABLE The unknowns which are to be determined from the solution of the model.

DIVERSION LINK Carries flow out of the system. A diversion link originates at any system node and terminates at the sink node.

FINAL-STORAGE LINK Carries flow out of the system, from a reservoir in the last period of analysis. It originates at a reservoir node and terminates at the sink node.

GAIN A factor (multiplier) that transforms the amount of flow in an arc. The flow leaving the arc equals the flow entering the arc times the gain. (The convention in HEC-PRM is for arc capacity to apply to the flow entering the arc).

GENERALIZED NETWORK FLOW PROGRAM A generalized network flow program is a network flow program in which arc flows have gains (multipliers) that are not equal to 1.0. For arcs with non-unity gains, the flow leaving the arc equals the flow entering the arc times the gain. If all gains are equal to 1.0, the problem is called a *pure network flow program.*

HYDROPOWER RESERVOIR-RELEASE LINK Represents the release from a hydropower reservoir. The penalty function for a hydropower reservoir-release link depends on both the release from the reservoir and the storage in the reservoir.

INFLOW LINK Brings flow into the reservoir-system network. An inflow link originates at the source node and terminates at any system node.

INITIAL-STORAGE LINK Introduces to the network the volume of water initially stored in a system reservoir. The initial-storage link originates at the source node and terminates at a reservoir node in the first period of analysis only.

NETWORK A collection of arcs and nodes.

NETWORK-FLOW PROGRAMMING An optimization procedure for allocating flow along the arcs of a network. Network-flow programming is a special class of linear programming.

NODE The junction of two or more network arcs. The node may represent a system reservoir, demand point, channel junction, diversion point. The sum of flow in arcs originating at a node equals the sum of flow in all arcs terminating at the node.

OBJECTIVE FUNCTION Defines the overall effectiveness of a system as a mathematical function of its decision variables. The optimal solution to the model yields the best value of the objective function, while satisfying all constraints.

PENALTY FUNCTION Defines the penalty for less-than-perfect operation as a function of flow, storage, or both.

PIECEWISE LINEAR APPROXIMATION An approximation in which a non-linear function is represented by linear segments, arranged sequentially.

RESERVOIR-STORAGE LINK Represents the volume of water stored in a reservoir at the end of a period. The link originates at any reservoir in a layered, multiple-period network and terminates at the node representing the same reservoir in the period following.

SIMPLE RESERVOIR-RELEASE LINK Represents the total outflow from a non-hydropower reservoir. Flow in the link includes release and spill.

SINK NODE Is the hypothetical absorber of all flow in the network. All diversion links and final-storage links terminate at the sink node.

SOLVER Finds the minimum-cost allocation of flow to the network arcs, subject to the upper and lower bounds on arc flows and to continuity at the network nodes.

SOURCE NODE Is the hypothetical provider of all flow in the network. All inflow links and initial-storage links originate at the source node. No user-defined links terminate at the source node.

APPENDIX C - Optimization Algorithms

HEC-PRM uses a number of specialized optimization algorithms to determine an optimal set of flows in a network representing a water resources system. The primary algorithm used is a primal simplex method for generalized network flow programming problems (Jensen and Barnes, 1987). If non-convex penalty functions, or nonlinear storage-area relationships, are specified, then the primal simplex method is modified for restricted basis entry (e.g., Hadley, 1964). Finally, if combined flow and storage penalty functions are specified, e.g. for hydroelectric power generation, then a successive linear programming algorithm is used in conjunction with the primal simplex method (with or without restricted basis entry). A summary of these algorithms is presented here for convenience. Further details can be found elsewhere (Hadley, 1964; Martin, 1982; Grygier and Stedinger, 1985; Jensen and Barnes, 1987).

Mathematical Statement of Problem

The optimization problem represented by the network with costs associated with flow can be written as follows (Jensen and Barnes, 1987):

Minimize:

$$\sum_{k}^{m} c_k f_k \qquad \text{(C-1)}$$

subject to:

$$\sum_{k \in M_O} f_k - \sum_{k \in M_T} a_k f_k \qquad \text{(for all nodes)} \qquad \text{(C-2)}$$

$$l_k \leq f_k \leq u_k \qquad \text{(for all arcs)} \qquad \text{(C-3)}$$

in which:

- m = total number of network arcs;
- c_k = unit cost for flow along arc k;
- f_k = flow along arc k;
- M_O = the set of all arcs originating at a node;
- M_T = the set of all arcs terminating at a node;
- a_k = multiplier for arc k;
- l_k = lower bound on flow along arc k; and
- u_k = upper bound on flow along arc k.

Equations C-1, C-2, and C-3 represent a special class of linear-programming (LP) problem: the *generalized minimum-cost network-flow problem*. Solution of the problem will yield an optimal allocation of flow within the system.

Network Simplex Method

The generalized minimum-cost network flow problem is solved using the primal network simplex method, as described in Jensen and Barnes (1987). This iterative procedure involves the following main steps:

1. Determine an initial basic feasible solution.
2. Compute the values of the dual variables for each node. These can be interpreted as the cost of obtaining an additional unit of flow at each node in the system.

3. Compute the arc marginal costs for all non-basic arcs. In any given iteration of the solution algorithm, the non-basic arcs are generally those which have zero flow or flow at the arc capacity. The marginal cost is the cost of increasing flow in the arc by one unit.
4. If all non-basic arc marginal costs satisfy the optimality conditions, stop; the optimal solution has been found. Otherwise, select an arc to enter the basis.
5. Determine which arc will leave the basis, compute the new basic feasible solution, and return to Step 2.

Network Simplex with Restricted Basis Entry

In the case of non-convex penalty functions, a restricted basis entry procedure (Hadley 1964) are used to ensure physically realistic solutions, i.e., that parallel arcs fill in the correct sequence. This procedure involves a number of minor modifications to the network simplex method. First, the initial solution must be feasible with respect to the sequenced arcs. A simple initial solution that satisfies this requirement has all 0 flows in parallel arc sets. If a non-zero initial flow is specified, the flows in the parallel arc set are arranged to satisfy the sequence restriction. Next, the method for selecting an arc to enter the basis is modified in two ways. If a parallel set of arcs includes a basic arc, no other arc in that set is allowed to enter the basis. If a parallel set of arcs does not include a basic arc, and one is found to violate the optimality conditions, the arc may enter the basis only if the sequence restriction will remain satisfied.

Successive Linear Programming

In the case of combined flow and storage penalties, such as those for hydroelectric power, the overall planning problem is be solved via iterative solution of linear approximations. Such successive linear programming techniques are described by Martin (1982), Grygier and Stedinger (1985), and Reznicek and Simonovic (1990). In summary, these techniques convert the combined release-storage penalty functions to release penalty functions by assuming a value of reservoir storage. Given the storage, head can be estimated. Given this head, the unit penalty for release is used, and the flow allocation problem is solved. Then the head assumption is checked, using the storage computed for the optimal allocation. If the assumption is not acceptable, the heads corresponding to the computed storages are used, and the process is repeated.

The algorithm proposed by Grygier and Stedinger (1985) is employed in HEC-PRM. This algorithm solves the nonlinear problem as follows:

1. **Initialize:** Set ITER (iteration counter) = 0. Set ITMAX = the maximum number of iterations allowed (must be > 1). Set CANDPEN (candidate optimal objective function value) = a very large number. Set)R_{max} = 0.50. Set $R_{j,upper}$ = release corresponding to maximum power generation at maximum head for reservoir *j*. ()R_{max} and $R_{j,upper}$ are used in constraining release in step 3, and are subject to change as we collect information on performance with alternative values.) For each reservoir *j*, for each period *t*, estimate $S_{j,t}$, the end-of-period storage. Go to step 2.

2. **Set Up the Network:** Set ITER = ITER + 1. If ITER > ITMAX, declare the candidate solution the optimal solution and stop. Otherwise, use the elevation-capacity function for reservoir *j* to determine the end-of-period head. Average the beginning-of-period and end-of-period heads. Select the "closest" user-provided linear approximation of the combined release-storage penalty function for each period. Set up the system network with arc bounds and costs to represent these combined penalty functions, along with flow and storage penalty functions for other purposes. Go to step 3.

3. **Limited Variation:** If ITER = 1, go to step 4. Otherwise, constrain flow on the reservoir release links so the total release does not vary from the candidate solution by

more than)R_{max}. The link lower bound would be $R_{j,t}$(1 +)R_{max}). If the candidate release is zero, set the upper bound equal $R_{j,upper}$. Go to step 4.

4. **Solve the Network:** Solve the resulting flow-allocation problem to find CURRPEN, the penalty associated with the current approximation. Use the best available network solver at this step. If a previous network solution is available, and if the solver can use it as a starting point, let it. Go to step 5.

5. **Check for Solution to Nonlinear Problem:** For each reservoir *j*, for each period *t*, determine $S_{j,t-1}$ and $S_{j,t}$ from the current solution of the network. Do these values differ from the values used in step 2 to select the approximation? If all are close enough, declare the current solution optimal and stop. Otherwise, go to step 6.

6. **Update Candidate Solution:** If CURRPEN < CANDPEN, it is an improvement, so save the current solution (storages, releases, etc.) as the candidate optimal solution, set CANDPEN = CURRPEN, and go to step 2. Otherwise, go to step 7.

7. **Decrease the Allowable Variation:** Set)R_{max} =)R_{max}/2. If)R_{max} < minimum value, declare the candidate solution optimal and stop. Otherwise, go to step 2.

APPENDIX D – Dual Cost Information

HEC-PRM writes the following dual information to DSS:

- Marginal (reduced) cost for each arc = the cost of a unit increase in arc capacity
- Dual value (shadow price) for each node = the cost of obtaining one additional unit of flow at the node

The marginal (reduced) costs are labeled "MARG_COST" or "MARG_COST_S". In the absence of degeneracy, all non-basic arcs (those with flows at their upper or lower bounds) should have non-zero marginal costs, indicating the cost (benefit) of increasing the lower (upper) bound by one unit. For a reservoir, two time series of marginal costs are output: "MARG_COST" represents the marginal cost for the release link; and "MARG_COST_S" represents the marginal cost for the storage link. Here, "link" is emphasized because of the minor complication arising when links comprise parallel arcs (due to a piecewise linear cost function). In this case, HEC-PRM outputs only the total flow on the link and, similarly, only the marginal cost for the link is reported.

The dual values (shadow prices) at the nodes are labeled "DUAL_TERM" and "DUAL_TERM_S" in the PRM output. These indicate the value (cost) of increasing (decreasing) the node's fixed external flow by one unit. Since all PRM output is labeled by arc in DSS, the (arbitrary) choice was made to write the values for the terminal node of each arc. However, there is also an option (hard-wired in the code) to print the values for the origin node of each arc. These are appropriately labeled as "DUAL_ORIGIN" and "DUAL_ORIGIN_S". Again, for reservoirs, the distinction is made between the terminal node of the release link ("DUAL_TERM") and the terminal node of the storage link ("DUAL_TERM_S"), which is actually the origin node of the storage link in the next time period. Of course, in any given time period, the origin node of the release link is the same as that of the storage link.

Considering an arc *k* from node *i* to node *j*, as shown in Figure 34, the arc marginal costs d_k are actually computed from the dual values (π_i, π_j), the arc unit cost c_k, and the gain factor a_k,

according to the following formula:

$$d_k = \pi_i + c_k - a_k \pi_j$$

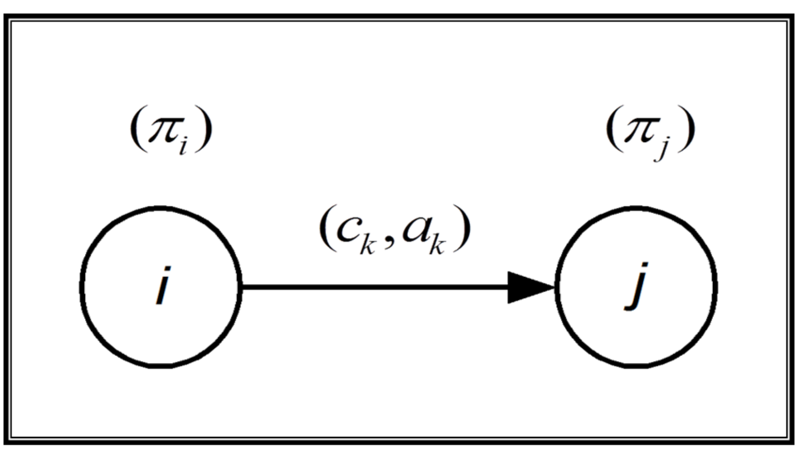

Figure 34. Definition of Dual Variables for Nodes

As an example, consider the system shown in Figure 35with 4 nodes and 3 arcs. A consistent set of dual information for this system is given in **Error! Reference source not found.**. In this case, the "DUAL_TERM" values for the reservoir in the first and second time periods are 2.5 and –1.5, respectively. "DUAL_TERM_S" for the reservoir in the first period equals –0.555.

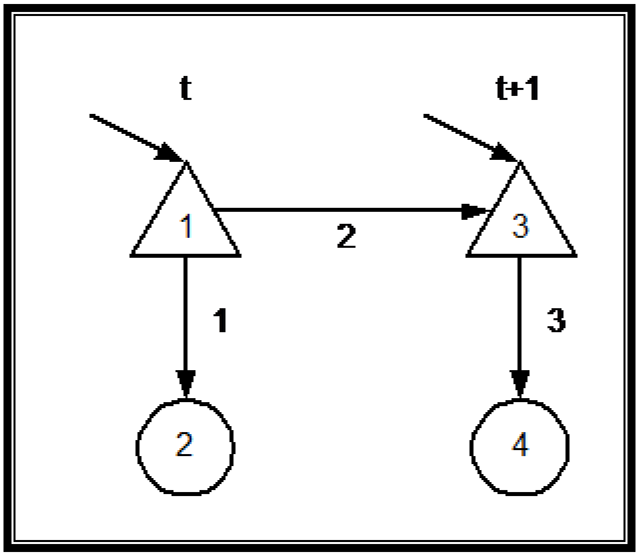

Figure 35. Example System

Table 7. Example Dual Cost Information

Arc, *k*	Orig., *i*	Term., *j*	π_i	π_j	c_k	a_k	d_k
1	1	2	2.000	2.500	1.500	1.000	1.000
2	1	3	2.000	-0.555	-2.500	0.900	0.000
3	3	4	-0.555	-1.500	0.500	1.000	-1.555

APPENDIX E - MATHPK Macros

MACRO DO_PRM

```
MACRO DO_PRM prm_dssfn pf_dssfn mpk_dssfn ts_dssfn alternative start_date end_date
```

```
!RUN OPEN prm_dssfn pf_dssfn mpk_dssfn ts_dssfn start_date end_date
SET.F PRLOUT=2
..      SET.F PRLOUT=5
SET.F PRLDSS=3
SET.F PL=84
SET.F P=(12X,2I5,15I10)
TAB.F-A FI=mpk_dssfn.mpt v=
TAB.FA

!RUN DO_STOR 'alternative'

SET.F PL=84
!RUN DO_CHAN 'alternative'

SET.F PL=60
..      !RUN NPD_SYS  'alternative'
ENDMACRO
```

MACRO OPEN & 50YR

```
MACRO OPEN prm_dssfn pf_dssfn mpk_dssfn ts_dssfn start_date end_date
```

```
OP prm_dssfn   1
OP pf_dssfn    2
OP ts_dssfn    4

TI.R start_date 2400
TI start_date 2400 end_date 2400

..        Do not extrapolate when interpolating
SET.O FU=TABLE IEXTR=1

SET.F PL=46
CLEAR
TA.F-A V=
TA.FA
COM IXMON=$DATE(1MON,MONTH)
COM NUM_DAY_MON=$DATE(1MON,DAY)
..      compute FACTOR; converts cfs to kaf by multiplication
COM FACTOR=.00198347*NUM_DAY_MON

DEF.T E=1MON V=PENTMP
SET.V V=IXMON FO=(I6)
SET.V V=NUM_DAY_MON FO=(I6)
ENDMACRO
```

```
MACRO 50YR
```

```
TI 31JUL1928 2400 30JUN1978 2400
ENDMACRO
```

MACRO MONVARY

```
MACRO MONVARY  partB parameter category indep_var depend_var pfalter file_index
```

```
..        12 months of data in one pathname
PURGE V=depend_var
DEF.T E=1MON V=depend_var
COM depend_var=0
```

```
PURGE V=PNLTY_FUNC
GET PNLTY_FUNC A= B=partB C=parameter-P_category D= E= F=pfalter file_index

PURGE V=PNLTY_FUNC_MAX, PNLTY_VALID
COM PNLTY_FUNC_MAX=MAX(PNLTY_FUNC)
DEF.C V=PNLTY_VALID
COM PNLTY_VALID=0
COM IF(PNLTY_FUNC_MAX(, 2) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 3) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 4) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 5) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 6) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 7) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 8) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 9) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(,10) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(,11) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(,12) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(,13) GT 0) PNLTY_VALID=1

COM IF(IXMON EQ  1) depend_var=TABLE(indep_var, PNLTY_FUNC(,1), PNLTY_FUNC(, 2))
COM IF(IXMON EQ  2) depend_var=TABLE(indep_var, PNLTY_FUNC(,1), PNLTY_FUNC(, 3))
COM IF(IXMON EQ  3) depend_var=TABLE(indep_var, PNLTY_FUNC(,1), PNLTY_FUNC(, 4))
COM IF(IXMON EQ  4) depend_var=TABLE(indep_var, PNLTY_FUNC(,1), PNLTY_FUNC(, 5))
COM IF(IXMON EQ  5) depend_var=TABLE(indep_var, PNLTY_FUNC(,1), PNLTY_FUNC(, 6))
COM IF(IXMON EQ  6) depend_var=TABLE(indep_var, PNLTY_FUNC(,1), PNLTY_FUNC(, 7))
COM IF(IXMON EQ  7) depend_var=TABLE(indep_var, PNLTY_FUNC(,1), PNLTY_FUNC(, 8))
COM IF(IXMON EQ  8) depend_var=TABLE(indep_var, PNLTY_FUNC(,1), PNLTY_FUNC(, 9))
COM IF(IXMON EQ  9) depend_var=TABLE(indep_var, PNLTY_FUNC(,1), PNLTY_FUNC(,10))
COM IF(IXMON EQ 10) depend_var=TABLE(indep_var, PNLTY_FUNC(,1), PNLTY_FUNC(,11))
COM IF(IXMON EQ 11) depend_var=TABLE(indep_var, PNLTY_FUNC(,1), PNLTY_FUNC(,12))
COM IF(IXMON EQ 12) depend_var=TABLE(indep_var, PNLTY_FUNC(,1), PNLTY_FUNC(,13))

COM IF(PNLTY_VALID LT 0.5) depend_var=-903E30

ENDMACRO
```

MACRO MONVARY1

```
MACRO MONVARY1 partB parameter category indep_var depend_var pfalter file_index
..Penalty function is stored in a separate path for each month
..- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
PURGE V=depend_var
DEF.T E=1MON V=depend_var
COM depend_var=0

PURGE V=PJAN,PFEB,PMAR,PAPR,PMAY,PJUN,PJUL,PAUG,PSEP,POCT,PNOV,PDEC
GET PJAN A= B=partB C=parameter-P_category D= E=JAN F=pfalter file_index
GET PFEB A= B=partB C=parameter-P_category D= E=FEB F=pfalter file_index
GET PMAR A= B=partB C=parameter-P_category D= E=MAR F=pfalter file_index
GET PAPR A= B=partB C=parameter-P_category D= E=APR F=pfalter file_index
GET PMAY A= B=partB C=parameter-P_category D= E=MAY F=pfalter file_index
GET PJUN A= B=partB C=parameter-P_category D= E=JUN F=pfalter file_index
GET PJUL A= B=partB C=parameter-P_category D= E=JUL F=pfalter file_index
GET PAUG A= B=partB C=parameter-P_category D= E=AUG F=pfalter file_index
GET PSEP A= B=partB C=parameter-P_category D= E=SEP F=pfalter file_index
GET POCT A= B=partB C=parameter-P_category D= E=OCT F=pfalter file_index
GET PNOV A= B=partB C=parameter-P_category D= E=NOV F=pfalter file_index
GET PDEC A= B=partB C=parameter-P_category D= E=DEC F=pfalter file_index

PURGE V=PNLTY_FUNC_MAX, PNLTY_VALID
DEF.P V=PNLTY_FUNC_MAX(1,13)
COM PNLTY_FUNC_MAX(1, 2)=MAX(PJAN(,2))
COM PNLTY_FUNC_MAX(1, 3)=MAX(PFEB(,2))
COM PNLTY_FUNC_MAX(1, 4)=MAX(PMAR(,2))
COM PNLTY_FUNC_MAX(1, 5)=MAX(PAPR(,2))
COM PNLTY_FUNC_MAX(1, 6)=MAX(PMAY(,2))
COM PNLTY_FUNC_MAX(1, 7)=MAX(PJUN(,2))
COM PNLTY_FUNC_MAX(1, 8)=MAX(PJUL(,2))
```

```
COM PNLTY_FUNC_MAX(1, 9)=MAX(PAUG(,2))
COM PNLTY_FUNC_MAX(1,10)=MAX(PSEP(,2))
COM PNLTY_FUNC_MAX(1,11)=MAX(POCT(,2))
COM PNLTY_FUNC_MAX(1,12)=MAX(PNOV(,2))
COM PNLTY_FUNC_MAX(1,13)=MAX(PDEC(,2))
DEF.C V=PNLTY_VALID
COM PNLTY_VALID=0
COM IF(PNLTY_FUNC_MAX(, 2) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 3) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 4) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 5) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 6) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 7) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 8) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 9) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(,10) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(,11) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(,12) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(,13) GT 0) PNLTY_VALID=1
COM IF(IXMON EQ  1) depend_var=TABLE(indep_var, PJAN(,1), PJAN(,2))
COM IF(IXMON EQ  2) depend_var=TABLE(indep_var, PFEB(,1), PFEB(,2))
COM IF(IXMON EQ  3) depend_var=TABLE(indep_var, PMAR(,1), PMAR(,2))
COM IF(IXMON EQ  4) depend_var=TABLE(indep_var, PAPR(,1), PAPR(,2))
COM IF(IXMON EQ  5) depend_var=TABLE(indep_var, PMAY(,1), PMAY(,2))
COM IF(IXMON EQ  6) depend_var=TABLE(indep_var, PJUN(,1), PJUN(,2))
COM IF(IXMON EQ  7) depend_var=TABLE(indep_var, PJUL(,1), PJUL(,2))
COM IF(IXMON EQ  8) depend_var=TABLE(indep_var, PAUG(,1), PAUG(,2))
COM IF(IXMON EQ  9) depend_var=TABLE(indep_var, PSEP(,1), PSEP(,2))
COM IF(IXMON EQ 10) depend_var=TABLE(indep_var, POCT(,1), POCT(,2))
COM IF(IXMON EQ 11) depend_var=TABLE(indep_var, PNOV(,1), PNOV(,2))
COM IF(IXMON EQ 12) depend_var=TABLE(indep_var, PDEC(,1), PDEC(,2))
COM IF(PNLTY_VALID LT 0.5) depend_var=-903E30
ENDMACRO
```

MACRO MONVARY3

```
MACRO MONVARY3 partB parameter category indep_var1 indep_var2 depend_var pfalter file_index
```

```
..Penalty function is stored in a separate path for each month with family of curves
..    indep_var1 is flow
..    indep_var2 is storage
..- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

PURGE V=depend_varDEF.T E=1MON V=depend_var
COM depend_var=0

PURGE V=PJAN,PFEB,PMAR,PAPR,PMAY,PJUN,PJUL,PAUG,PSEP,POCT,PNOV,PDEC
GET PJAN A= B=partB C=parameter-P_category D= E=JAN F=pfalter file_index
GET PFEB A= B=partB C=parameter-P_category D= E=FEB F=pfalter file_index
GET PMAR A= B=partB C=parameter-P_category D= E=MAR F=pfalter file_index
GET PAPR A= B=partB C=parameter-P_category D= E=APR F=pfalter file_index
GET PMAY A= B=partB C=parameter-P_category D= E=MAY F=pfalter file_index
GET PJUN A= B=partB C=parameter-P_category D= E=JUN F=pfalter file_index
GET PJUL A= B=partB C=parameter-P_category D= E=JUL F=pfalter file_index
GET PAUG A= B=partB C=parameter-P_category D= E=AUG F=pfalter file_index
GET PSEP A= B=partB C=parameter-P_category D= E=SEP F=pfalter file_index
GET POCT A= B=partB C=parameter-P_category D= E=OCT F=pfalter file_index
GET PNOV A= B=partB C=parameter-P_category D= E=NOV F=pfalter file_index
GET PDEC A= B=partB C=parameter-P_category D= E=DEC F=pfalter file_index

..          ST.V4S
..          ST.T

COM IF(IXMON EQ  1) depend_var=TABLE3P(indep_var1, indep_var2, PJAN)
COM IF(IXMON EQ  2) depend_var=TABLE3P(indep_var1, indep_var2, PFEB)
COM IF(IXMON EQ  3) depend_var=TABLE3P(indep_var1, indep_var2, PMAR)
COM IF(IXMON EQ  4) depend_var=TABLE3P(indep_var1, indep_var2, PAPR)
COM IF(IXMON EQ  5) depend_var=TABLE3P(indep_var1, indep_var2, PMAY)
COM IF(IXMON EQ  6) depend_var=TABLE3P(indep_var1, indep_var2, PJUN)
COM IF(IXMON EQ  7) depend_var=TABLE3P(indep_var1, indep_var2, PJUL)
```

```
COM IF(IXMON EQ  8) depend_var=TABLE3P(indep_var1, indep_var2, PAUG)
COM IF(IXMON EQ  9) depend_var=TABLE3P(indep_var1, indep_var2, PSEP)
COM IF(IXMON EQ 10) depend_var=TABLE3P(indep_var1, indep_var2, POCT)
COM IF(IXMON EQ 11) depend_var=TABLE3P(indep_var1, indep_var2, PNOV)
COM IF(IXMON EQ 12) depend_var=TABLE3P(indep_var1, indep_var2, PDEC)

..          !IF('partB'.EQ.'BROWNLEE_P') THEN
..          SET.F TILN=1 TI=Computed data for partB
..          SET.F T=(F10.1)
..          TAB V=indep_var1 indep_var2 depend_var
..          SET.F P=(1X,I3,I8,15I10)
..          TAB V=PJAN PFEB PMAR PAPR PMAY PJUN PJUL PAUG PSEP POCT PNOV PDEC
..          !ENDIF
 URGE V=PNLTY_FUNC_MAX, PNLTY_VALID
DEF.P V=PNLTY_FUNC_MAX(1,13)
COM PNLTY_FUNC_MAX(1, 2)=MAX(PJAN(,2))
COM PNLTY_FUNC_MAX(1, 3)=MAX(PFEB(,2))
COM PNLTY_FUNC_MAX(1, 4)=MAX(PMAR(,2))
COM PNLTY_FUNC_MAX(1, 5)=MAX(PAPR(,2))
COM PNLTY_FUNC_MAX(1, 6)=MAX(PMAY(,2))
COM PNLTY_FUNC_MAX(1, 7)=MAX(PJUN(,2))
COM PNLTY_FUNC_MAX(1, 8)=MAX(PJUL(,2))
COM PNLTY_FUNC_MAX(1, 9)=MAX(PAUG(,2))
COM PNLTY_FUNC_MAX(1,10)=MAX(PSEP(,2))
COM PNLTY_FUNC_MAX(1,11)=MAX(POCT(,2))
COM PNLTY_FUNC_MAX(1,12)=MAX(PNOV(,2))
COM PNLTY_FUNC_MAX(1,13)=MAX(PDEC(,2))
DEF.C V=PNLTY_VALID
COM PNLTY_VALID=0
COM IF(PNLTY_FUNC_MAX(, 2) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 3) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 4) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 5) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 6) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 7) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 8) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 9) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(,10) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(,11) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(,12) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(,13) GT 0) PNLTY_VALID=1
 ..          ST.V4 V=PNLTY_VALID
..          TAB V=PNLTY_VALID
COM IF(PNLTY_VALID LT 0.5) depend_var=-903E30
ENDMACRO
```

MACRO HPA_NPD

```
MACRO HPA_NPD partB parameter indep_var1 indep_var2 depend_var pfalter file_index
..Penalty function is stored in a separate path for each month with family of curves
..    indep_var1 is flow
..    indep_var2 is storage
..- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

PURGE V=depend_var
DEF.T E=1MON V=depend_var
COM depend_var=0

PURGE V=PJAN,PFEB,PMAR,PAPR,PMAY,PJUN,PJUL,PAUG,PSEP,POCT,PNOV,PDEC
GET PMAR A= B=partB C=parameter D= E=MAR_NOV       F=pfalter file_index
GET PAPR A= B=partB C=parameter D= E=APR-MAY_OCT   F=pfalter file_index
GET PJUN A= B=partB C=parameter D= E=JUN-SEP       F=pfalter file_index
GET PDEC A= B=partB C=parameter D= E=DEC-FEB       F=pfalter file_index

COM IF(IXMON EQ  1) depend_var=TABLE3P(indep_var1, indep_var2, PDEC)
COM IF(IXMON EQ  2) depend_var=TABLE3P(indep_var1, indep_var2, PDEC)
COM IF(IXMON EQ  3) depend_var=TABLE3P(indep_var1, indep_var2, PMAR)
COM IF(IXMON EQ  4) depend_var=TABLE3P(indep_var1, indep_var2, PAPR)
COM IF(IXMON EQ  5) depend_var=TABLE3P(indep_var1, indep_var2, PAPR)
```

```
COM IF(IXMON EQ  6) depend_var=TABLE3P(indep_var1, indep_var2, PJUN)
COM IF(IXMON EQ  7) depend_var=TABLE3P(indep_var1, indep_var2, PJUN)
COM IF(IXMON EQ  8) depend_var=TABLE3P(indep_var1, indep_var2, PJUN)
COM IF(IXMON EQ  9) depend_var=TABLE3P(indep_var1, indep_var2, PJUN)
COM IF(IXMON EQ 10) depend_var=TABLE3P(indep_var1, indep_var2, PAPR)
COM IF(IXMON EQ 11) depend_var=TABLE3P(indep_var1, indep_var2, PMAR)
COM IF(IXMON EQ 12) depend_var=TABLE3P(indep_var1, indep_var2, PDEC)

PURGE V=PNLTY_FUNC_MAX, PNLTY_VALID
DEF.P V=PNLTY_FUNC_MAX(1,13)
COM PNLTY_FUNC_MAX(1, 2)=MAX(PDEC(,2))
COM PNLTY_FUNC_MAX(1, 3)=MAX(PDEC(,2))
COM PNLTY_FUNC_MAX(1, 4)=MAX(PMAR(,2))
COM PNLTY_FUNC_MAX(1, 5)=MAX(PAPR(,2))
COM PNLTY_FUNC_MAX(1, 6)=MAX(PAPR(,2))
COM PNLTY_FUNC_MAX(1, 7)=MAX(PJUN(,2))
COM PNLTY_FUNC_MAX(1, 8)=MAX(PJUN(,2))
COM PNLTY_FUNC_MAX(1, 9)=MAX(PJUN(,2))
COM PNLTY_FUNC_MAX(1,10)=MAX(PJUN(,2))
COM PNLTY_FUNC_MAX(1,11)=MAX(PAPR(,2))
COM PNLTY_FUNC_MAX(1,12)=MAX(PMAR(,2))
COM PNLTY_FUNC_MAX(1,13)=MAX(PDEC(,2))
DEF.C V=PNLTY_VALID
COM PNLTY_VALID=0
COM IF(PNLTY_FUNC_MAX(, 2) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 3) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 4) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 5) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 6) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 7) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 8) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(, 9) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(,10) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(,11) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(,12) GT 0) PNLTY_VALID=1
COM IF(PNLTY_FUNC_MAX(,13) GT 0) PNLTY_VALID=1

..          ST.V4 V=PNLTY_VALID
..          TAB V=PNLTY_VALID
COM IF(PNLTY_VALID LT 0.5) depend_var=-903E30
ENDMACRO
```

MACRO MONSAME

```
MACRO MONSAME partB parameter category indep_var depend_var pfalter file_index
```

```
PURGE V=depend_var
DEF.T E=1MON V=depend_var
COM depend_var=0

PURGE V=PNLTY_FUNC
GET PNLTY_FUNC A= B=partB C=parameter-P_category D= E= F=pfalter file_index

PURGE V=PNLTY_FUNC_MAX, PNLTY_VALID
COM PNLTY_FUNC_MAX=MAX(PNLTY_FUNC)
DEF.C V=PNLTY_VALID
COM PNLTY_VALID=0
COM IF(PNLTY_FUNC_MAX(, 2) GT 0) PNLTY_VALID=1

COM depend_var=TABLE(indep_var,PNLTY_FUNC(,1),PNLTY_FUNC(,2))
COM IF(PNLTY_VALID LT 0.5) depend_var=-903E30
ENDMACRO
```

MACRO CLASSDEF

```
MACRO CLASSDEF start end interval

DEF.C V=NU_CLASS
COM NU_CLASS = end - start / interval + 1
PURGE V=CLASS
DEF.P V=CLASS(NU_CLASS,2)
COM CLASS(,1)=1
COM CLASS(,2)=interval
COM CLASS(1,2)=start
COM CLASS=ACCUM(CLASS)
ENDMACRO
```

MACRO HYSSROUT & HYSSR1

```
MACRO HYSSROUT alternative

!RUN HYSSR1 'MICA_P'    'alternative'
!RUN HYSSR1 'ARROW'     'alternative'
!RUN HYSSR1 'DUNCAN'    'alternative'
ENDMACRO

MACRO HYSSR1  location alternative

DEF.T E=1MON V=TSVAR
COM TSVAR=0
SET.V V=TSVAR U=K$ T=UNT A= B=location E=1MON F=alternative
PUT TSVAR C=PS_FDA      3
PUT TSVAR C=PS_FDU      3
PUT TSVAR C=PS_FIS      3
PUT TSVAR C=PS_NAV      3
PUT TSVAR C=PS_REC      3
PUT TSVAR C=PS_WSP      3
PUT TSVAR C=PS_ACT      3
PUT TSVAR C=PS_EDT      3
PUT TSVAR C=PQ_HPE      3
PUT TSVAR C=PQ_HPA      3
PUT TSVAR C=PQ_FDA      3
PUT TSVAR C=PQ_FDU      3
PUT TSVAR C=PQ_FIS      3
PUT TSVAR C=PQ_NAV      3
PUT TSVAR C=PQ_REC      3
PUT TSVAR C=PQ_WSP      3
PUT TSVAR C=PQ_ACT      3
PUT TSVAR C=PQ_EDT      3

SET.V V=TSVAR U=MW
PUT TSVAR C=ENERGY_GEN 1
PUT TSVAR C=POWER_CAP


GET STOR //location/STOR//1MON/HYSSR/ 4
GET FLOW //location/FLOW//1MON/HYSSR/ 4
PUT STOR 3
PUT STOR 1
PUT FLOW 3
PUT FLOW 1
ENDMACRO
```

MACRO DO_RES & DO_RES1

```
MACRO DO_RES res_node hpe_loc alternative filename
!RUN DO_RES1 'res_node' 'hpe_loc' 'alternative'
!RUN DO_RES3 'res_node' 'hpe_loc' 'alternative'
ENDMACRO
```

```
MACRO DO_RES1  res_node hpe_loc alternative
..                                             Get release & storage
TI -1M
GET STORAGE A= B=res_node C=STOR E=1MON F=alternative 1
TI -0M
GET STOR_AVG A= B=res_node C=STOR_AVG E=1MON F=alternative 1
GET RELEASE A= B=res_node C=FLOW(KAF) E=1MON F=alternative 1
..
!RUN MONVARY 'res_node' S FDU STORAGE FDU_POOL ' ' 2
!RUN MONVARY 'res_node' S FDA STORAGE FDA_POOL ' ' 2
!RUN MONVARY 'res_node' S WSP STORAGE WSP_POOL ' ' 2
!RUN MONVARY 'res_node' S NAV STORAGE NAV_POOL ' ' 2
!RUN MONVARY 'res_node' S REC STORAGE REC_POOL ' ' 2
!RUN MONVARY 'res_node' S FIS STORAGE FIS_POOL ' ' 2
!RUN MONVARY 'res_node' S ACT STORAGE ACT_POOL ' ' 2
!RUN MONVARY1 'res_node' S EDT STORAGE EDT_POOL 'alternative' 1

DEF.T E=1MON  V=HPE_ACT_REL,HPE_EDT_REL,HPE_NRG,HPE_CAP
COM HPE_ACT_REL=m
COM HPE_EDT_REL=m
NRG_ACT_REL=m
CAP_ACT_REL=m
COM HPE_NRG=-903E30
COM HPE_CAP=-903E30

!IF ( 'hpe_loc' .EQ. 'HPE' ) THEN
!RUN HPA_NPD  'res_node' 'Q(KAF)-P_HPA' 'RELEASE' 'STORAGE' 'HPE_ACT_REL'  ' '  2
!RUN HPA_NPD  'res_node' 'Q(KAF)-P_HEA' 'RELEASE' 'STORAGE' 'NRG_ACT_REL'  ' '  2
!RUN HPA_NPD  'res_node' 'Q(KAF)-P_HCA' 'RELEASE' 'STORAGE' 'CAP_ACT_REL'  ' '  2
!RUN MONVARY3 'res_node' 'Q(KAF)' 'HPE' 'RELEASE' 'STORAGE' 'HPE_EDT_REL'  'alternative' 1
PURGE V=T_NRG T_CAP HPE_NRG HPE_CAP
GET T_NRG A= B=res_node C=Q(KAF)-NRG D= E= F=       2
GET T_CAP A= B=res_node C=Q(KAF)-CAP D= E= F=       2
COM HPE_NRG=TABLE3P(RELEASE,STOR_AVG,T_NRG)
COM HPE_CAP=TABLE3P(RELEASE,STOR_AVG,T_CAP)
!ENDIF

SET.V V=HPE_NRG A= B=res_node C=ENERGY_GEN E=1MON F=alternative U=MW T=PER-AVER
SET.V V=HPE_CAP A= B=res_node C=POWER_CAP  E=1MON F=alternative U=MW T=PER-AVER
..      PUT HPE_NRG   1
..      PUT HPE_CAP   1

!RUN MONVARY  'res_node' 'Q(KAF)' FDU RELEASE FDU_REL  ' ' 2
!RUN MONVARY  'res_node' 'Q(KAF)' FDA RELEASE FDA_REL  ' ' 2
!RUN MONVARY  'res_node' 'Q(KAF)' WSP RELEASE WSP_REL  ' ' 2
!RUN MONVARY  'res_node' 'Q(KAF)' NAV RELEASE NAV_REL  ' ' 2
!RUN MONVARY  'res_node' 'Q(KAF)' REC RELEASE REC_REL  ' ' 2
!RUN MONVARY  'res_node' 'Q(KAF)' FIS RELEASE FIS_REL  ' ' 2
!RUN MONVARY  'res_node' 'Q(KAF)' ACT RELEASE ACT_REL  ' ' 2
!RUN MONVARY1 'res_node' 'Q(KAF)' EDT RELEASE EDT_REL 'alternative' 1

SET.V V=STORAGE     FO=(F10.1)
SET.V V=RELEASE     FO=(F10.2)
ENDMACRO
```

MACRO DO_RES3

```
MACRO DO_RES3 res_node hpe_loc alternative
SET.V V=STORAGE     U=KAF T=INST-VAL A=  B=res_node C=STOR             F=alternative
SET.V V=RELEASE     U=KAF T=PER-AVER A=  B=res_node C=FLOW(KAF)        F=alternative
SET.V V=FDA_POOL    U=K$  T=PER-AVER A=  B=res_node C=PS_FDA           F=alternative
SET.V V=FDU_POOL    U=K$  T=PER-AVER A=  B=res_node C=PS_FDU           F=alternative
SET.V V=FIS_POOL    U=K$  T=PER-AVER A=  B=res_node C=PS_FIS           F=alternative
SET.V V=NAV_POOL    U=K$  T=PER-AVER A=  B=res_node C=PS_NAV           F=alternative
SET.V V=REC_POOL    U=K$  T=PER-AVER A=  B=res_node C=PS_REC           F=alternative
SET.V V=WSP_POOL    U=K$  T=PER-AVER A=  B=res_node C=PS_WSP           F=alternative
SET.V V=ACT_POOL    U=K$  T=PER-AVER A=  B=res_node C=PS_ACT           F=alternative
SET.V V=EDT_POOL    U=K$  T=PER-AVER A=  B=res_node C=PS_EDT           F=alternative
SET.V V=FDA_REL     U=K$  T=PER-AVER A=  B=res_node C=PQ_FDA           F=alternative
SET.V V=FDU_REL     U=K$  T=PER-AVER A=  B=res_node C=PQ_FDU           F=alternative
SET.V V=FIS_REL     U=K$  T=PER-AVER A=  B=res_node C=PQ_FIS           F=alternative
SET.V V=NAV_REL     U=K$  T=PER-AVER A=  B=res_node C=PQ_NAV           F=alternative
SET.V V=REC_REL     U=K$  T=PER-AVER A=  B=res_node C=PQ_REC           F=alternative
SET.V V=WSP_REL     U=K$  T=PER-AVER A=  B=res_node C=PQ_WSP           F=alternative
SET.V V=ACT_REL     U=K$  T=PER-AVER A=  B=res_node C=PQ_ACT           F=alternative
SET.V V=EDT_REL     U=K$  T=PER-AVER A=  B=res_node C=PQ_EDT           F=alternative

PUT FDA_POOL    1
PUT FDU_POOL    1
PUT FIS_POOL    1
PUT NAV_POOL    1
PUT REC_POOL    1
PUT WSP_POOL    1
PUT ACT_POOL    1
PUT EDT_POOL    1

PUT FDA_REL     1
PUT FDU_REL     1
PUT FIS_REL     1
PUT NAV_REL     1
PUT REC_REL     1
PUT WSP_REL     1
PUT ACT_REL     1
PUT EDT_REL     1

SET.V V=HPE_ACT_REL U=K$  T=PER-AVER A=  B=res_node C=PQ_HPA F=alternative
SET.V V=NRG_ACT_REL U=K$  T=PER-AVER A=  B=res_node C=PQ_HEA F=alternative
SET.V V=CAP_ACT_REL U=K$  T=PER-AVER A=  B=res_node C=PQ_HCA F=alternative
SET.V V=HPE_EDT_REL U=K$  T=PER-AVER A=  B=res_node C=PQ_HPE F=alternative
PUT HPE_ACT_REL 1
PUT NRG_ACT_REL 1
PUT CAP_ACT_REL 1
PUT HPE_EDT_REL 1
ENDMACRO
```

MACRO DO_CP & DO_CP1 & DO_CP3

```
MACRO DO_CP loc_norm alternative filename
!RUN DO_CP1 'loc_norm' 'alternative'
!RUN DO_CP3 'loc_norm' 'alternative'
ENDMACRO
```

```
MACRO DO_CP1  loc_norm alternative
GET FLOW A= B=loc_norm C=FLOW(KAF) D= E=1MON F=alternative 1

!RUN MONVARY  'loc_norm' 'Q(KAF)' FDA FLOW FDA_FLOW '       ' 2
!RUN MONVARY  'loc_norm' 'Q(KAF)' FDU FLOW FDU_FLOW '       ' 2
!RUN MONVARY  'loc_norm' 'Q(KAF)' FIS FLOW FIS_FLOW '       ' 2
!RUN MONVARY  'loc_norm' 'Q(KAF)' NAV FLOW NAV_FLOW '       ' 2
!RUN MONVARY  'loc_norm' 'Q(KAF)' REC FLOW REC_FLOW '       ' 2
!RUN MONVARY  'loc_norm' 'Q(KAF)' WSP FLOW WSP_FLOW '       ' 2
!RUN MONVARY  'loc_norm' 'Q(KAF)' ACT FLOW ACT_FLOW '       ' 2
```

```
!RUN MONVARY1 'loc_norm' 'Q(KAF)' EDT FLOW EDT_FLOW 'alternative' 1

SET.V V=FDA_FLOW    U=K$ T=' '
SET.V V=FDU_FLOW    U=K$ T=' '
SET.V V=FIS_FLOW    U=K$ T=' '
SET.V V=NAV_FLOW    U=K$ T=' '
SET.V V=REC_FLOW    U=K$ T=' '
SET.V V=WSP_FLOW    U=K$ T=' '
SET.V V=ACT_FLOW    U=K$ T=' '
SET.V V=EDT_FLOW    U=K$ T=' '
ENDMACRO
```

MACRO DO_CP3 loc_norm alternative

```
SET.V V=FLOW        U=KAF T=PER-AVER A=  B=loc_norm C=FLOW(KAF)       F=alternative
SET.V V=FDA_FLOW    U=CFS T=PER-AVER A=  B=loc_norm C=PQ_FDA          F=alternative
SET.V V=FDU_FLOW    U=K$  T=PER-AVER A=  B=loc_norm C=PQ_FDU          F=alternative
SET.V V=FIS_FLOW    U=K$  T=PER-AVER A=  B=loc_norm C=PQ_FIS          F=alternative
SET.V V=NAV_FLOW    U=K$  T=PER-AVER A=  B=loc_norm C=PQ_NAV          F=alternative
SET.V V=REC_FLOW    U=K$  T=PER-AVER A=  B=loc_norm C=PQ_REC          F=alternative
SET.V V=WSP_FLOW    U=K$  T=PER-AVER A=  B=loc_norm C=PQ_WSP          F=alternative
SET.V V=ACT_FLOW    U=K$  T=PER-AVER A=  B=loc_norm C=PQ_ACT          F=alternative
SET.V V=EDT_FLOW    U=K$  T=PER-AVER A=  B=loc_norm C=PQ_EDT          F=alternative

PUT FDA_FLOW  1
PUT FDU_FLOW  1
PUT FIS_FLOW  1
PUT NAV_FLOW  1
PUT REC_FLOW  1
PUT WSP_FLOW  1
PUT ACT_FLOW  1
PUT EDT_FLOW  1

ENDMACRO
```

MACRO DO_STOR & DO_CHAN

MACRO DO_STOR alternative

```
!RUN DO_RES DWORSHAK_P HPE 'alternative' DWORSHAK
!RUN DO_RES BROWNLEE_P HPE 'alternative' BROWNLEE
!RUN DO_RES GRANITE_P  HPE 'alternative' GRANITE
ENDMACRO
```

MACRO DO_CHAN alternative

```
!RUN DO_CP SPALDING   'alternative' SPALDING
ENDMACRO
```