

TECHNICAL PUBLICATION 86-5

September 1986

**A ROUTING MODEL FOR THE UPPER
KISSIMMEE CHAIN OF LAKES**

**by
Andrew Fan**

**Water Resources Division
Resource Planning Department
South Florida Water Management District**

This public document was promulgated at an annual cost of \$522.13 or \$1.05 per copy to inform the public regarding water resource studies of the District.
RPD 387-5C

ACKNOWLEDGEMENTS

The development of this computer model was completed under the supervision of Jorge Marban , Director of Water Resource Division, Resource Planning Department. The author wishes to thank the following persons for providing ideas and comments on the conceptual framework of the model: Alan Hall, Jorge Marban, Ron Mierau, Kent Loftin, George Hwa, Steve Lin, and Tom Vanlent. Special recognition is given to Kent Loftin and Odile Grosser for their careful review and proof reading of the manuscript, and ideas to improve this documentation.

TABLE OF CONTENTS

LIST OF FIGURES	iii
LIST OF TABLES	iii
I INTRODUCTION	1
II MODEL OVERVIEW	1
III THEORETICAL ASSUMPTIONS AND LIMITATIONS	3
IV DESCRIPTION OF MODEL STUDY AREA	3
V DESCRIPTION OF MODEL COMPONENTS	5
A. Control Structure Characteristics	5
B. Regulation Schedules	6
C. Gate Operation Criteria	6
D. Structure and Canal Flow Equations	7
E. Rainfall and Evaporation	9
F. Watershed Inflows	9
G. Stage-Area Relationships	12
H. Routing Procedures	13
I. Parameter Optimization Procedures	14
VI MODEL USAGE	15
VII MODEL VERIFICATION	17
APPENDICES	
1. Dictionary of Program Symbols (KROUTE)	20
2. Program Listing (KROUTE)	22
3. Program Listing (KPLOT)	39
4. Program Listing (KBUDGET)	49
5. Interactive Computer Session	55
6. Simulation and Calibration Results (1970-1980)	62
7. Typical Forecasting Results	83

LIST OF FIGURES

1.	Study Area Location Map	2
2.	Model Study Area	4
3.	Typical Riprap Control Criteria	7
4.	Comparison of Pan Evaporation Data	10
5.	Pan Evaporation Coefficient	10
6.	Inflow Watersheds	11
7.	Parameter Optimization	15

LIST OF TABLES

1.	Control Structures Parameters	1
2.	Riprap Control Coefficients	6
3.	Structure and Canal Flow Equations	8
4.	Rainfall Stations	9
5.	Stage-Area Relationships	13
6.	Subroutine Functions	16
7.	Input and Output Data Files	16
8.	Comparison of Forecasted and Actual Stages	18

I. INTRODUCTION

A computer model was developed by the SFWMD to simulate the operation of the Upper Kissimmee Chain of Lakes, Florida. The model serves as a management tool to predict the lake conditions so that alternative management schemes, aimed at achieving specific objectives, can be evaluated.

The model area covers a chain of nine lakes (Lakes Alligator, Myrtle, Hart, Gentry, East Tohopekaliga, Tohopekaliga, Cypress, Hatchineha, and Kissimmee) in the upper portion of the Kissimmee River Basin. The lakes are interconnected with canals and outlet control structures that are rigidly regulated. The lake discharges are affected by tailwater conditions and the watershed inflows to the lakes are poorly defined. The hydrologic and hydraulic conditions are unique and existing routing models are inadequate to handle the unusual conditions. For this reason, a computer model was developed to meet the special requirements.

The model was first developed in 1981 to address the problems created by the prolonged drought of 1980-81. Later, the model was used to evaluate the new regulation schedules proposed in 1982, and in 1983 to assess impacts that might be created by the Kissimmee River Demonstration Project. The model has been used routinely to forecast the lake stages at the end of each month using rainfall as the conditional dependent variable. Numerous modifications have been made to improve the model's efficiency and accuracy. Although model improvement and maintenance is a continuing effort, the basic conceptual framework of the model is complete. This report documents the model methodology and illustrates its application.

Section II presents an overview of the model concepts. Section III stipulates the model limitations and assumptions. Section IV describes the study area characteristics. Section V describes the model components in detail. Sections VI and VII illustrate the usage of the model and its verification.

II. MODEL OVERVIEW

The Upper Kissimmee Chain of Lakes Routing Model (KROUTE) is a continuous simulation model designed to simulate the operation of the lake system in the Upper Kissimmee River Basin (Figure 1). The study area covers Lakes Alligator, Myrtle, Hart and Mary Jane, Gentry, East Tohopekaliga and Tohopekaliga, Cypress, Hatchineha, and Kissimmee. The model is capable of simulating the management of the

system according to predetermined regulation schedules, structure operational criteria, and rainfall conditions. The model can be viewed as predominantly physically based, though certain elements, due to the ways they are conceptualized and calibrated, may only be visualized as partly deterministic. This section provides an overview of the model methodology, which will be explained in more detail in the remainder of this publication.

Level pool storage routing is used to route the flow through the system. Storage routing is similar to a water budget computation. Both use the same mass balance equation:

$$Q_{in} + Q_{lake} - Q_{out} + P - E \pm DSTOR + ADJ = 0 \quad \text{---} < 1 >$$

where

- Q_{in} = Structure inflows from upper lake
- Q_{lake} = Watershed inflows
- Q_{out} = Structure outflow to lower lake
- P = Lake precipitation
- E = Lake evaporation
- $DSTOR$ = Change in lake storage
- ADJ = Inflow adjustment (all unestimated flows and errors of water budget).

In the water budget computation, all components except ADJ in Equation 1 are known historical data. Equation 1 is used to quantify the unknown term, ADJ , so as to evaluate the uncertainties of the water budget.

In storage routing, all components are unknown except rainfall, which is predetermined. Evaporation, E , and watershed inflows, Q_{lake} , are estimated as a function of rainfall. In lakes where the inflows are adequately gaged, ADJ is nearly zero and can be neglected; otherwise ADJ must be estimated or included in the watershed inflow term Q_{lake} . The structure inflow term, Q_{in} , is Q_{out} in the next upstream lake from the previous routing step. Equation 1 is used to determine the unknown quantities Q_{out} and $DSTOR$; but since both unknowns are a function of the lake stage, lake stage is the only real unknown.

Routing is performed in daily time steps beginning from the uppermost lake (Lake Alligator) to the lowermost lake (Lake Kissimmee). The lake outflow term, Q_{out} , is affected hydraulically by the lower lake stage (backwater effect), and management constraints described by the lake regulation schedules and gate operation criteria. Due to backwater effect, an iteration technique is used to balance the headwater and tailwater stage relationship because the tailwater stage in the current time step is unknown a priori.

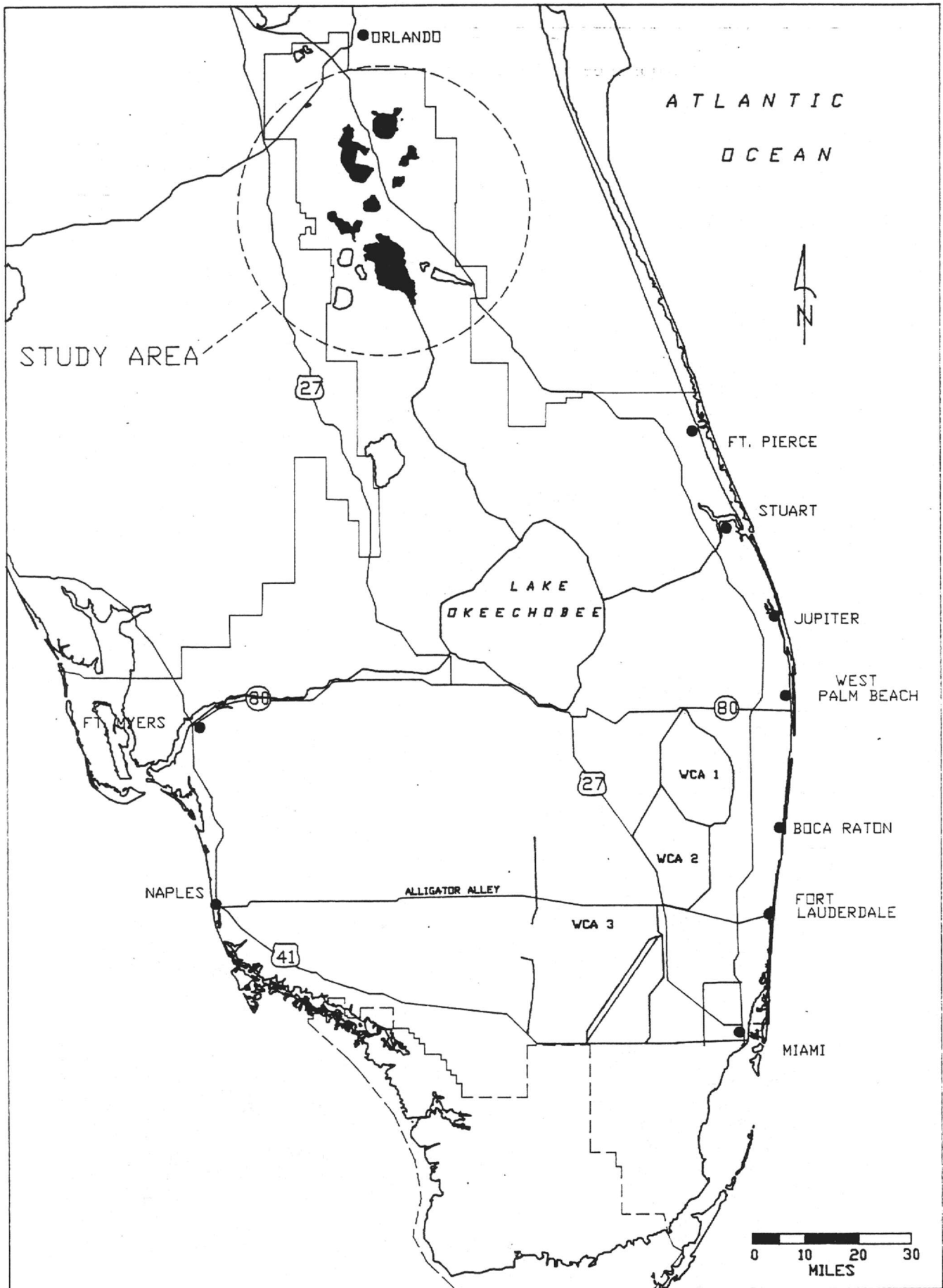


Figure 1. Study Area Location Map

Watershed inflows to the lakes are estimated as direct runoff and base flow. Direct runoff is estimated by a District modified Soil Conservation Service method. Base flow is estimated by an empirical formula which relates potential base flow to actual base flow using water table depth as a dependent variable. Both direct runoff and base flow are dependent on the water table depth. A soil moisture accounting procedure was developed to continuously predict the effective water table elevation.

The model provides three operational modes: "simulation," "calibration," and "forecasting." The three modes differ in the way the lake inflows are obtained. In "simulation," historical rainfall and watershed inflows are input. The model simulates the structure flows and stages and is suitable for evaluating situations where only the management variables change. In "calibration," historical rainfall is input and all other flows are predicted. An optimization option is provided to assist in calibrating the parameters. In "forecasting," rainfall is specified and the model predicts all inflows as a function of rainfall.

Complimentary to the routing program (KROUTE) are a water budget computation program (KBUDGET) and a plotting program (KPLOT). The water budget program is used to verify the historical data and to preprocess the input data files needed to calibrate the routing program. The plotting program presents the routing results in a form suitable for interpretation and verification. A listing of all three programs is included in the Appendix.

Calibration of the model involves the calibration of the watershed parameters. The usage and verification of the model are presented in the last two sections. The results indicate that the logic of the routing scheme is accurate and that the simulation of the structure flow is excellent. In short term forecasting the results are good; in long term forecasting moderately large deviations at times are observed. The major uncertainty of the model lies in the difficulties of accurately forecasting the watershed inflows.

III. THEORETICAL ASSUMPTIONS AND LIMITATIONS

Major assumptions and limitations of the model are presented below. More specific assumptions, such as those that are structure specific, are presented in Section V.

1. A primary assumption of the routing model is that level pool conditions exist. The assumption is

valid as long as the flow through the lake is small relative to the storage. The assumption is reasonable under normal flow conditions but is slightly violated under heavy discharge conditions.

2. The model simulates the management of the system according to a set of management rules. These rules are expressed in regulation schedules, gate operation criteria, and established rules governing the operation of the structures. As long as the operation follows the established rules, the simulation of the management is possible. Under unusual conditions, the operation may differ from the established rules and this explains the inability of the routing model to simulate those events.

3. The model runs in daily time steps and generates daily average flows and stages. The time step resolution is adequate for most applications except for extreme storm events where instantaneous peak stages and flows are important. Nevertheless, an examination of the recorded lake hydrographs suggests that, due to the large size of the lakes, the instantaneous stages are not significantly different from the daily averages. The errors introduced are probably small in comparison to random fluctuation of the lake stages due to wind set up effect and other disturbances.

4. For certain applications where only the management variables change, historical rainfall and inflow data are used. The implicit assumption is that a change in the management will not change the historical hydrologic variables.

5. In forecasting applications, rainfall is specified to be uniformly distributed over time and space for the month of forecast. This is a scenario, rather than a model assumption, and can be modified. The assumption is more acceptable for dry than for wet conditions.

IV. DESCRIPTION OF THE MODEL STUDY AREA

The model covers a nine lake system in the upper portion of the Kissimmee River Basin (Figure 2). The lakes are interconnected with canals and control structures that are rigidly regulated. Alligator Lake is the uppermost lake with no definable surface water inflows. Outflow from Alligator Lake can be made north through a chain of small lakes to East Lake Tohopekaliga, or south through Lake Gentry to Lake Cypress; however, because of the limited capacity of the lakes north of Alligator Lake, discharges have been made primarily south.

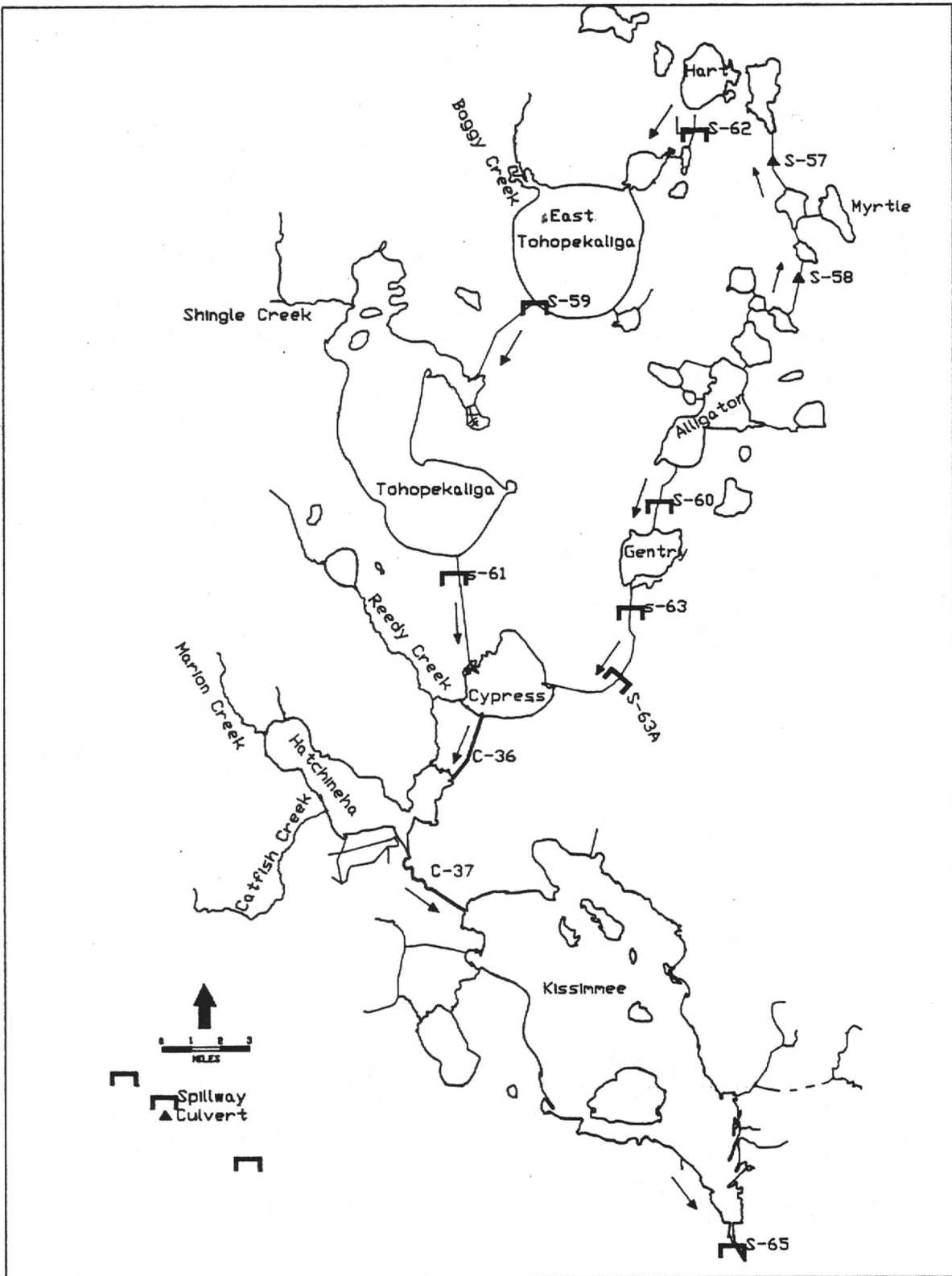


Figure 2. Model Study Area

North of Alligator Lake are the Lake Myrtle-Preston system and the Lake Hart-Mary Jane system, and to the south is the Lake Gentry system. These systems consist of one to five small lakes, and together they make up the headwater portion of the chain. As a group, these small lakes generate only a small portion of the flow in the chain and their influence in the lower lakes is relatively small.

East Lake Tohopekaliga is the first of five major lakes in the chain. The largest inflow to East Lake Tohopekaliga is from Boggy Creek. Second largest is the inflow from Lake Hart through S-62. East Lake Tohopekaliga discharges to Lake Tohopekaliga, which is the second largest lake in the chain. Shingle Creek, which drains to Lake Tohopekaliga, is the largest tributary in the chain.

Lake Cypress receives inflows from both Lake Tohopekaliga and Lake Gentry. There is a stage drop of approximately ten feet between Lakes Gentry and Cypress; and Structures 63 and 63A are used to step down the pools. Lakes Cypress, Hatchineha, and Kissimmee constitute the lower three lake system. They can be considered as a system because their stages tend to be equalized since there are no restricting structures in the connecting canals (C-36 and C-37). At times of high stage, the lower three lakes are connected by swamps, making it difficult to identify the boundaries.

Reedy Creek is the largest tributary inflow to the lower three lake system. Reedy Creek splits into two branches near Lake Cypress. One branch enters Lake Cypress and the other, known as Dead River, enters Lake Hatchineha. The major portion of the flow (approximately 70%) enters Lake Hatchineha. Catfish Creek, which connects Lake Pierce with Lake Hatchineha, is another important tributary to the lower three lake system. Lake Hatchineha also receives inflow from Marion Creek and London Creek from the north, which drains a moderately large area currently under intensive urban development.

Lake Kissimmee is the largest lake in the chain. Four other medium sized lakes drain into Lake Kissimmee, including Lake Rosalie, Lake Tiger, Lake Marian, and Lake Jackson. With the exception of Lake Tiger, however, these lakes contribute little flow to Lake Kissimmee because earth dams were constructed at their outlets. Inflows from the drainage areas of the lower three Lakes, though important, are poorly gaged.

The Kissimmee Chain of Lakes are shallow; average depths range from 13 ft in Lake Alligator to 8 ft in Lake Kissimmee. Geologically, the lakes cut into

the surficial aquifer which has a thickness ranging from 50 to 100 ft in the study area. The surficial aquifer is made up of relatively homogeneous silty fine to medium sand. Permeability is estimated to be low. Although direct seepage to the lake is normally small in comparison to other inflows, at time of drought, seepage may become important.

V. DESCRIPTION OF MODEL COMPONENTS

A. Control Structure Characteristics

The Kissimmee Chain of Lakes is interconnected with canals and discharge control structures that are rigidly regulated. The control structures consist of gated culverts and gated spillways and their physical characteristics are summarized in Table 1 (see also Figure 2 on page 4). Several structures have special operational characteristics and require special treatment:

1. A weir structure was installed in 1979 below the S-59 spillway to raise the tailwater stage so that larger gate openings could be used without creating damaging velocities. A submerged weir flow equation is used in the model to adjust the tailwater stage at S-59 from the lake stage.
2. Lake Alligator has two outlet structures. At the north end is the S-58 gated culvert, and at the south end is the S-60 spillway. The S-58 culvert has seldom been used because of the limited capacity of the small lakes and canals north of Lake Alligator. An assumption is made that the S-58 structure will be operated only when the capacity of S-60 is inadequate to lower the stage at Lake Alligator to regulation schedule within a day.
3. Structure 63A is an automatic spillway located below S-63. Both structures control the outflow from Lake Gentry and together allow a stepped drop of ten feet between Lake Gentry and Lake Cypress. It is assumed that discharges at S-63 and S-63A are the same and therefore only S-63 flow is modeled. Structure 63A maintains, automatically, an optimum headwater pool of 56.5 msl. The model uses this 56.5 msl to calculate the flow through S-63 and is assumed to be unchanged.
4. The S-65 spillway is the outlet of the chain and can discharge up to 11000 cfs. Under actual operation, however, the discharge at S-65 has never exceeded 6000cfs. In day to day operations, the District determines the maximum allowable discharge at S-65 based on the amount of inflow experienced on the

TABLE 1
Control Structures Parameters

Structure	Type	Crest EL (MSL)	Length (Ft)	Diameter (Ft)	Design Q (cfs)	Remarks
S-57	Culvert	52.5(invert)	80	4.5	230	Seldom operated
S-58	Culvert	54.5(invert)	70	4.5	110	
S-59	Spillway	49.1	18	NA	590	Weir below structure
S-60	Spillway	55.0	12	NA	450	
S-61	Spillway	36.9	27	NA	1570	
S-62	Spillway	55.3	14	NA	410	
S-63	Spillway	54.0	15	NA	715	
S-63A	Spillway	49.4	30	NA	2000	Automatic gate
S-65	Spillway	39.3	81	NA	11000	

previous day. Based on statistical analyses of the historical data, two assumptions were made: First, the tailwater stage at S-65 is fixed at 46.3 msl, which is the historical mean stage maintained by S-65A. Second, the maximum discharge at S-65 is limited to 3000 cfs and 5000 cfs, respectively, under dry and wet conditions. Wet conditions are defined as having antecedent monthly rainfall exceeding eight inches. Baffle blocks were installed below S-65 in 1986. The dry condition discharge constraint was raised to 6000 cfs on an interim basis until the model is extended to the lower basin whereby the boundary condition will be shifted from S-65 to S-65E.

B. Regulation Schedules

The lakes are regulated by rigid schedules. The regulation schedules represent the management aspect of the system aimed at optimizing flood control, water conservation, and environmental enhancement. The trend of the regulation schedules generally reaches the minimum and maximum at the beginning and end of the wet season to prepare for flood control and water conservation, respectively. Regulation schedules are, nevertheless, subjective rules which change every few years. In the routing model, the

actual regulation schedules for the years under simulation are entered as breakpoint data in a separate input file.

C. Gate Operation Criteria

The maximum allowable gate opening for each spillway is governed by the "Riprap Control" criteria, so named because the objective is to prevent excessive velocity damage to the riprap around the structures. The criteria were established by the U. S. Army Corps of Engineers and presented in charts such as that shown in Table 2 and Figure 3. The family of curves can be fitted by the following equation which appears to be the same general form of equation used to calculate flow through a weir under submerged conditions:

$$GO = A(HW - TW)^B TWC + D \text{ ----- } <2>$$

where

- GO = Maximum allowable gate opening in ft
- HW = Headwater elevation
- TW = Tailwater elevation
- A, B, C, D = constant coefficients.

TABLE 2
Riprap Control Coefficients
(Equation <2>)

	A	B	C	D
S-59	0.38x10 ⁻⁸	-0.22083	5.41927	-3.673
S-60	2.2588896x10 ⁻¹⁴	-0.51706295	8.2136768	0.
S-61	0.002954	-.15463	2.24833	-10.82
S-62	1.7404157x10 ⁻²²	-.59052612	12.743331	0.
S-63	0.022745	-.21273	1.6471	-8.246
S-65	0.36587777	-.55682769	.95236956	0.

According to Equation 2, the greater the head difference, the smaller the gate opening allowed. The effect on hydraulics is exactly the opposite. Thus, an increase in the head difference may increase the flow rather than reduce it.

Gated culvert structures 57 and 58 do not have gate operation criteria and this presents difficulties in simulating their operation. Their discharge capacities, however, are relatively small and S-58 has seldom been operated. Based on an analysis of the historical data, maximum allowable discharges of 230 and 110 cfs are assigned to S-57 and S-58 as additional operational constraints.

D. Structure and Canal Flow Equations

The flow equations for the control structures and Canals 36 and 37 are summarized in Table 3. Though the flow equations for the structures incorporate all flow conditions, under normal operation submerged controlled flow predominates. Stream flow gaging data were used to calibrate the spillway flow equations. Since most of the measurements were taken under submerged controlled flow conditions, only this type of flow was calibrated. The fitted coefficients are

shown in Table 3. Calibrated values of some coefficients exceeded the theoretical range, which can be attributed to datum errors and other unknown noise in the original data.

Canals 36 and 37 are the open channels connecting Lakes Cypress and Hatchineha, and Lakes Hatchineha and Kissimmee. Because there are no control structures along C-36 and C-37, the stages in the lower three lakes tend to be equalized. The flow rating equations for C-36 and C-37 (Table 3) were established from stream flow gaging data taken in 1983 and 1984. The flow equations shown in Table 3 are in essence a Manning Equation using both the stage and water surface slope as independent variables:

$$Q = A(HW - TW)^B(HW - C)^D \text{-----} <3>$$

where

- Q = Discharge in cfs
- HW = Upper lake stage in ft msl
- TW = Lower lake stage in ft msl
- A, B, C, and D = Calibrated coefficients shown in Table 3

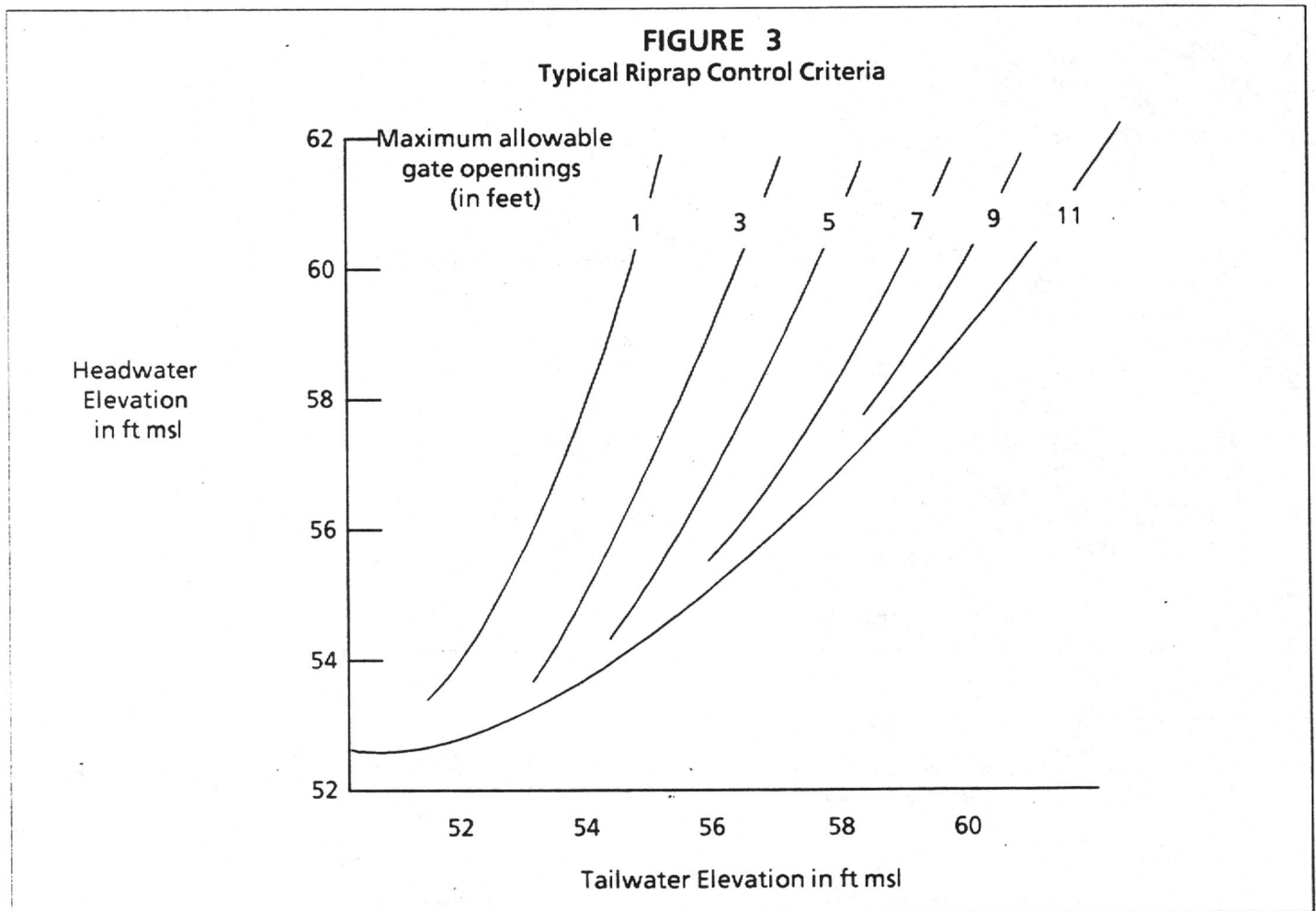


TABLE 3
Structure and Canal Flow Equations

Spillway

<u>Flow Type</u>	<u>Flow Equation</u>	<u>Criteria</u>
Free weir flow	$Q1 = 3.28 AL(HW - CEL)^{1.5}$	$TW < CEL$ and $Q1 < Q2$
Free control flow	$Q2 = 0.75AL \cdot GO [64.4(HW - CEL - 0.5GO)]^{0.5}$	$TW < CEL$ and $HW > 1.1(CEL + GO)$
Submerged weir flow	$Q3 = 0.9AL(TW - CEL) [64.4(HW - TW)]^{0.5}$	$TW > CEL$ and $Q3 < Q4$
Submerged control	$Q4 = (\alpha \cdot GO + \beta)(AL)(GO) [64.4(HW - TW)]^{0.5}$	$TW > CEL$ and $Q4 < Q3$

where $Q1, Q2, Q3, Q4$ = Flow in cfs under four different flow conditions

AL = Spillway length in ft

HW = Headwater elevation in msl

CEL = Spillway crest elevation in msl

GO = Gate opening in ft obtained from Equation <2> and Table 2

α, β = Calibrated flow coefficients shown below:

	S-59	S-60	S-61	S-62	S-63	S-65
α	.1033	0	.0253	0	0	.0375
β	.58	.73	.59	.75	.75	.76

Gated Culvert

<u>Flow Type</u>	<u>Flow Equation</u>	<u>Criteria</u>
Open channel flow	$Q1 = (1.49/n)(AR)(SLOPE)^{0.5}$	$HW < TOP$
Orifice flow	$Q2 = 0.75A [64.4(HW - TW)]^{0.5}$	$HW > TOP$ and $TW < TOP$
Full pipe flow	$Q3 = (1.49/n)(AR)(SLOPE)^{0.5}$	$HW > TOP$ and $TW > TOP$

where

$Q1, Q2, Q3$ = Flow in cfs under three different flow conditions

TOP = Top elevation of pipe in msl

n = Manning n (0.024 for all culverts)

AR = Linearized conveyance coefficient

$$= (1.72y/d - 0.373)(3.142)(A)(d/4)^{0.6667}$$

y = Mean depth of flow in ft

d = Diameter of pipe in ft

A = Cross sectional area of pipe in ft²

$SLOPE$ = Slope of water surface

$$= (HW - TW)/L$$

HW = Headwater elevation in msl

TW = Tailwater elevation in msl

L = Length of pipe in ft

Canal 36 and 37

C-36: $Q = 35.61885873(HW - TW - .19)^{.5511796}(HW - 35.07)^{1.6667}$

C-37: $Q = 87.07430164(HW - TW + .12)^{.4976433}(HW - 42.12)^{1.6667}$

The rating equations compute the instantaneous flow rate, and since the model runs in daily time steps, there may occasionally be a time resolution problem. In order to assure the conservation of mass, the computed total daily discharge is compared with the allowable storage release (which is estimated as the storage above the regulation schedule plus the expected inflows) and the smaller of the two is selected. If the allowable storage release is less than zero, no discharge will be made. Since there are no control structures in C-36 and C-37, the allowable storage release at Lakes Cypress or Hatchineha is calculated as the storage above the downstream lake stage plus the expected inflows. If the allowable storage release is negative due to higher stage in the downstream lake, reverse (negative) flow is permitted.

In summary, therefore, by combining gate operation criteria (Equation 2), flow rating equations (Table 3), and lake regulation schedules, the management of the system can be simulated.

E. Rainfall and Evaporation

In forecasting, rainfall is a predetermined quantity specified by the user. Evaporation is predicted as a function of rainfall. Both rainfall and evaporation are distributed uniformly throughout the lakes unless specified otherwise. Evaporation is predicted by the following frequency correlation equation:

$$E_d = C \{ \overline{E}_m - (R_m - \overline{R}_m) S_E / S_R \} / N_m \quad \text{--- --- < 4 >}$$

where

- E_d = Estimated daily lake evaporation in inches
- C = Pan to lake coefficient (0.8 used)
- \overline{E}_m = Normal pan evaporation for calendar month, m, in inches
- S_E = Standard deviation of pan evaporation (1.7094 used)
- S_R = Standard deviation of rainfall (2.4755 used)
- R_m = Total rainfall for calendar month m in inches
- \overline{R}_m = Normal rainfall for calendar month, m, in inches
- N_m = Number of days in calendar month, m

Assuming both follow the same probability distribution, the above equation equates the frequency occurrence of rainfall and evaporation; that is, a one in ten year rainfall in any month will generate a one in ten year evaporation in the same month. Furthermore, the relationship is inverse; that is, the smaller the amount of rainfall, the greater the evaporation.

Historical rainfall and pan evaporation data were used in model calibration. Rainfall input to the lakes is based on weighted average rainfall from stations around the lakes as listed in Table 4. Lake Alfred Station is the only long term evaporation station in the study area. Evaporation is assumed to be uniformly distributed throughout the lakes in the chain. Figure 4 shows a comparison of the evaporation pan data at Lake Alfred and S-65 for 1982-83, suggesting that the evaporation variation in the study area is small. A 0.8 pan to lake coefficient is used. The 0.8 coefficient was calibrated from a water budget analysis of East Lake Tohopekaliga for the drought period of October 1981 to April 1982. The calibration curve is shown in Figure 5.

TABLE 4
Rainfall Stations

Lake	Station (weighting factor)
Alligator	MRF8(2/6), MRF12(2/6), MRF14(1/6), MRF19(1/6)
Myrtle	MRF8(2/6), MRF12(2/6), MRF14(1/6), MRF19(1/6)
Hart	MRF8(2/5), MRF3(2/5), MRF4(1/5)
Gentry	MRF18(1/2), MRF19(1/2)
East Toho	MRF8(1/3), MRF4(1/3), MRF12(1/3)
Toho	MRF12(1/4), MRF162(1/4), MRF18(1/4), MRF17(1/4)
Cypress	MRF18(1/3), MRF23(1/3), MRF24(1/3)
Hatchineha	MRF24(1/3), MRF18(1/6), MRF140(1/3), MRF23(1/6)
Kissimmee	MRF23(1/4), MRF24(1/4), MRF27(1/4), MRF28(1/4)

F. Watershed Inflows

Thirteen watersheds are simulated in the model. There are four gaged (Boggy Creek, Shingle Creek, Catfish Creek, and Reedy Creek) and nine ungaged watersheds. Each ungaged watershed represents a combination of several local watersheds that drain to a lake. The inflow watersheds and their areas are shown in Figure 6. Watershed inflows to the lakes consist of both direct runoff and base flow. In continuous simulation, base flow can be more important than direct runoff because it occurs continuously. The model simulates direct runoff and base flow separately.

Direct runoff is simulated by a District modified Soil Conservation Service Direct Runoff Formula.

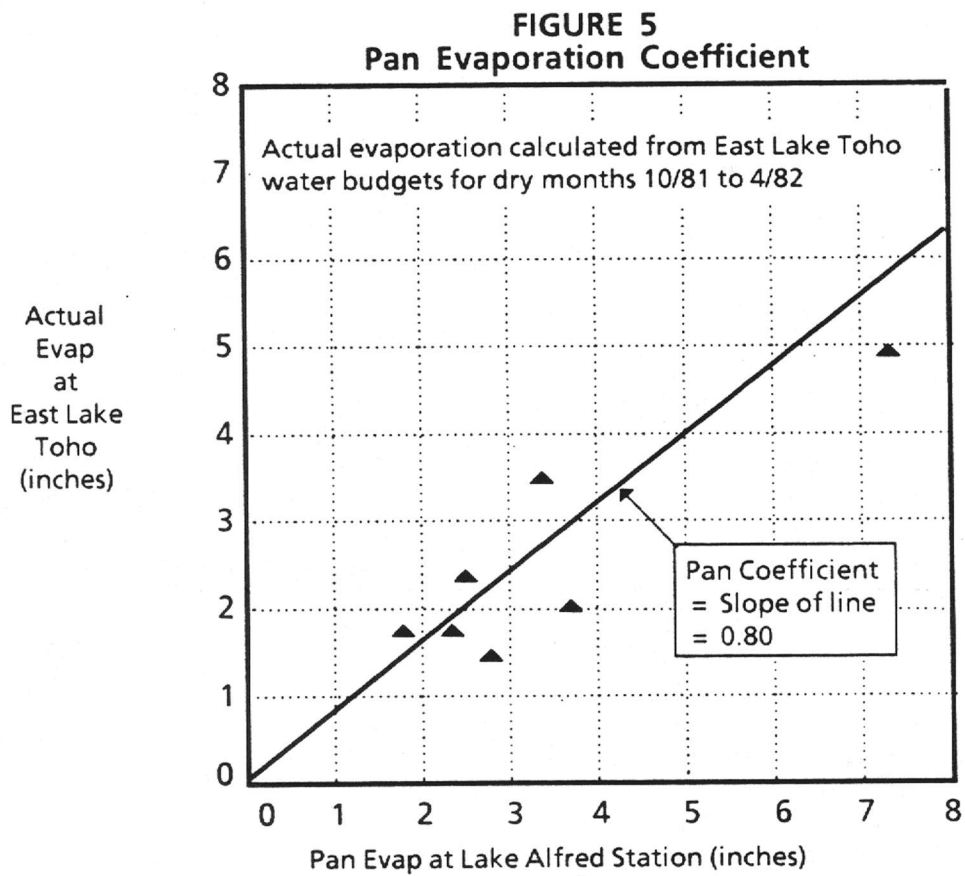
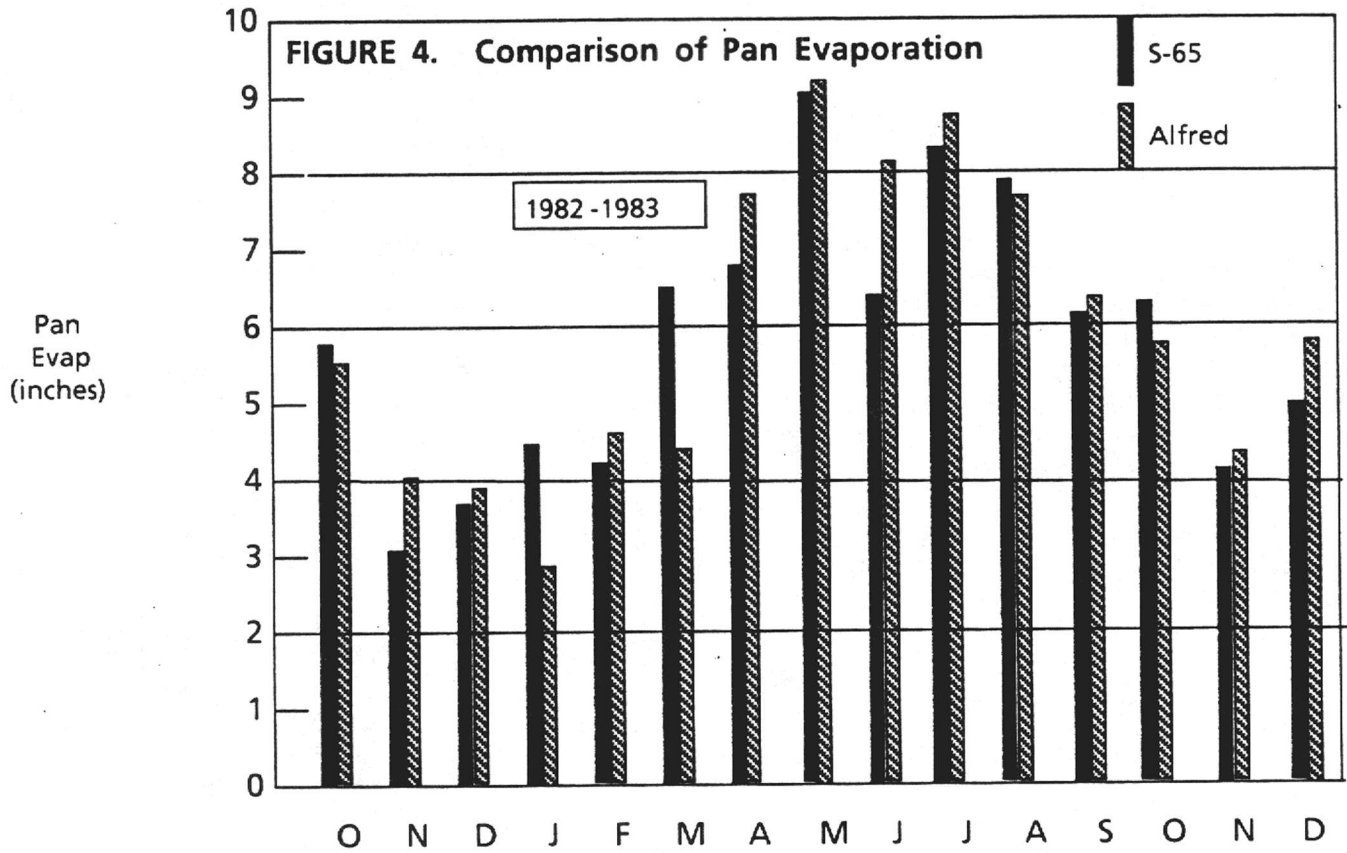
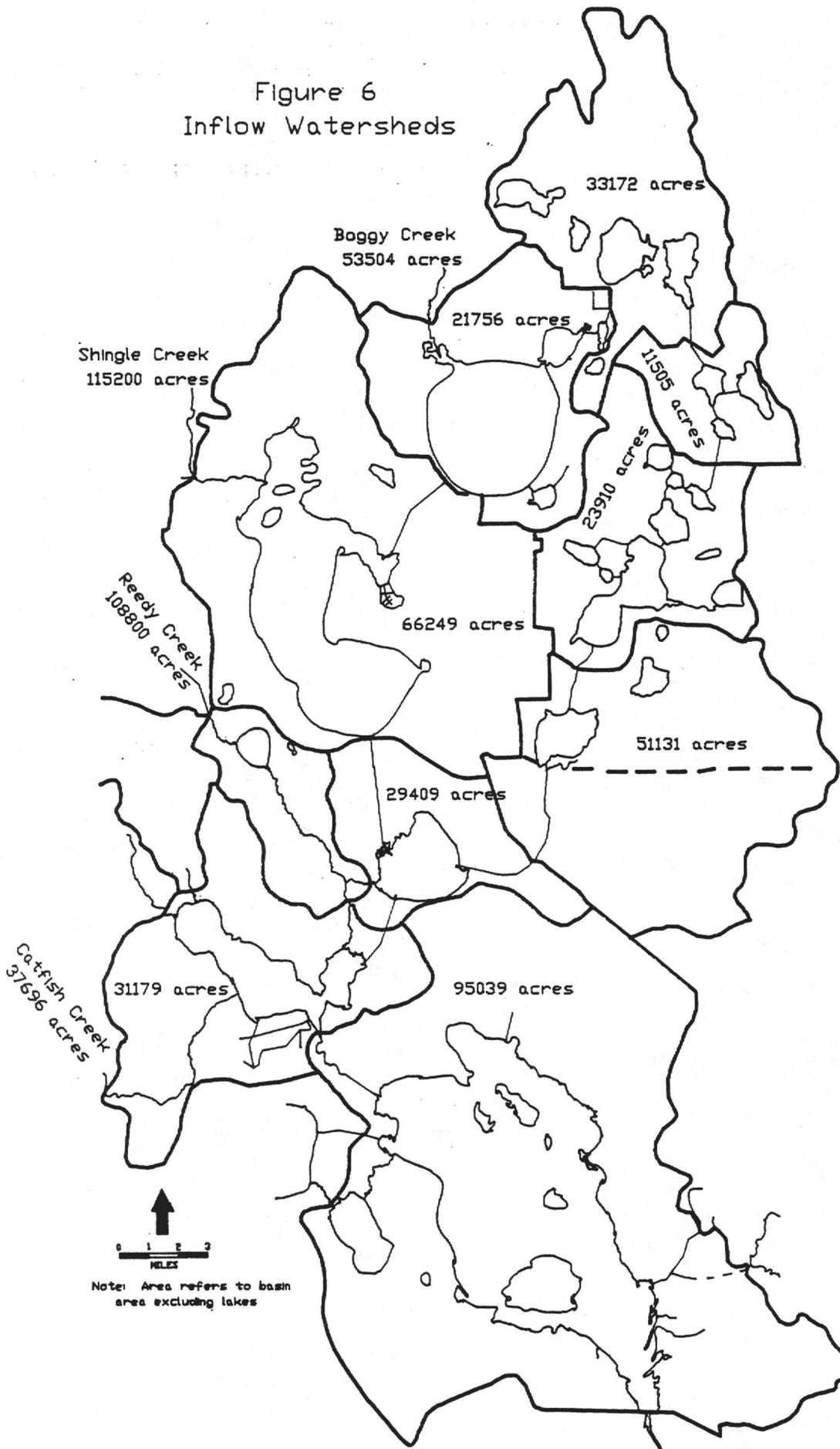


Figure 6
Inflow Watersheds



Routing of the direct runoff (rainfall excess) is not performed because the watershed time of concentrations are smaller than a one day time step.

$$Q = \frac{(P - 0.2S)^2}{(P + 0.8S)} \text{-----} <6>$$

where

- Q = Direct runoff volume in inches
- P = Precipitation in inches
- S = Soil moisture deficit in inches

Base flow simulation is based on the concept of potential and actual base flow. The actual base flow is linearly related to the water table depth:

$$B = \text{SCOEF} \left(\frac{HMAX - WT}{HMAX} \right) \text{-----} <7>$$

where

- B = Base flow in inches
- SCOEF = Potential base flow in inches
- HMAX = Depth at which base flow ceases (5 to 10 ft)
- WT = Water table depth

In the above equation *SCOEF* represents the potential base flow when the water table is near the land surface; *HMAX* represents the effective depth at which the base flow ceases. This corresponds to a situation when the water table has effectively fallen below the stream beds. The formulation of the base flow is analogous to the formulation of the watershed evapotranspiration loss:

$$ET = \text{PET} \left(\frac{ROOT - WT}{ROOT} \right) \text{-----} <8>$$

where

- ET = Watershed evapotranspiration in inches
- PET = Potential ET in Inches = PCOEF x EVAP
- PCOEF = Pan coefficient at PET (0.7 to 0.9)
- EVAP = Pan evaporation in inches
- ROOT = Deep root zone in feet below which ET ceases (5 to 10 ft)
- WT = Water table depth in feet

To evaluate Equations 6, 7, and 8 the knowledge of the water table depth *WT* and soil moisture deficit *S* is needed. Assuming a constant storage coefficient of 0.2 and including a unit conversion factor, the water table depth *WT* (in feet) can be related to *S* (in inches) by a factor of 2.4. Thus only *S* needs to be quantified. A soil moisture accounting procedure is formulated to continuously monitor *S*:

$$S_t = S_{t-1} + ET - P + Q + B \text{-----} <9>$$

where

- S_t* = Soil moisture deficit (in.) at time step *t*
- S_{t-1}* = Soil moisture deficit in inches at previous time step, *t-1*
- ET* = Watershed evapotranspiration loss in inches from Equation 8
- P* = Precipitation in inches
- Q* = Direct runoff in inches from Equation 6
- B* = Base flow in inches from Equation 7

In summary, the watershed discharge is modeled as the summation of direct runoff and base flow from Equations 6 and 7 which, in turn, are functions of the soil moisture deficit *S* determined by Equations 8 and 9. The procedure described requires the calibration of four parameters: *SCOEF*, *HMAX*, *PCOEF*, and *ROOT*. The base flow parameters (*SCOEF* and *HMAX*) are primarily a function of the drainage density and aquifer characteristics; the *ET* parameters (*PCOEF* and *ROOT*) are primarily a function of the land use and soil type. An optimization procedure is provided to assist in calibrating the parameters and is described in Section I.

The current approach is a lumped parameter approach; that is, each watershed is treated as a unit and the parameters are effective parameters for the entire watershed. Thus, although the procedure is physically based and has the ability to reflect changes in the physical conditions, statistical elements are introduced in the process of calibration and the procedure can only be viewed as partly deterministic.

G. Stage-Area Relationships

Polynomial equations were fitted to the stage-area data and the coefficients are shown in Table 5. Polynomial fitting is suitable for interpolation only; extrapolation beyond the range of data used to calibrate the equation can be erroneous. The following linear extrapolation equation is used to project the area above the maximum surveyed stage:

$$AREA = A_{(hmax)} + DA_{(hmax)} \times [h - hmax] \text{---} <10>$$

where

- AREA = Lake area in acres
- A_(h)* = Stage area function
- = $D1(h)^4 + D2(h)^3 + D3(h)^2 + D4(h) + D5$
- h* = Stage in ft msl (> *hmax*)
- D1 to D5* = coefficients for area rating
- DA_(h)* = $dA_{(h)} / dh$
- = $4(D1)(h)^3 + 3(D2)(h)^2 + 2(D3)(h) + D4$
- hmax* = Maximum stage used in polynomial fitting in ft msl

TABLE 5
Stage-Area Relationships

$$AREA = D1(h)^4 + D2(h)^3 + D3(h)^2 + D4(h) + D5$$

Where AREA = lake area in acres
 h = stage in ft msl
 $D1$ to $D5$ = coefficients for stage area rating
 h_{max} = Maximum stage used in polynomial fitting in ft msl
 h_{min} = Minimum stage used in polynomial fitting in ft msl

	D1	D2	D3	D4	D5	h_{max}	h_{min}
Alligator	-22.386	5641.0	-53246×10^6	$.22315 \times 10^8$	$-.35034 \times 10^9$	65.00	59.50
Myrtle	2.0264	-481.93	42983.	$-.17039 \times 10^7$	$.25327 \times 10^8$	65.00	58.00
Hart	.29565	-67.473	5783.8	$-.22044 \times 10^6$	$.31507 \times 10^7$	64.00	56.00
Gentry	1.5463	-366.74	32609.	$-.12881 \times 10^7$	$.19074 \times 10^8$	65.00	57.00
East Toho	-.26352	64.992	-5948.0	$.24036 \times 10^6$	$-.36178 \times 10^7$	65.00	50.00
West Toho	-1.6619	367.90	-30441.	$.11172 \times 10^7$	$-.15346 \times 10^8$	60.00	49.00
Cypress	0.	1.7257	-273.08	14590.	$-.25875 \times 10^6$	58.00	43.00
Hatchineha	0.	-6.0509	854.41	-39014.	$.57813 \times 10^6$	55.00	45.00
Kissimmee	0.	56.094	-8051.3	$.38667 \times 10^6$	$-.61833 \times 10^7$	58.00	42.50

For projection below the minimum surveyed stage, the following equation is used. The formulation is empirical, and the objective is to avoid negative area projection.

$$AREA = A_{(h_{min})} \times (h / h_{min})^{1.5}$$

where

AREA = Lake area in acres
 $A_{(h)}$ = Area function
 h = Stage in ft msl ($< h_{min}$)
 h_{min} = Minimum stage used in polynomial fitting in ft msl

H. Routing Procedures

Routing proceeds from the uppermost lake (Alligator) to the lowermost lake (Kissimmee) by solving the mass balance equation in daily time steps. The mass balance equation is rewritten below with the unknowns placed on the left side of the equation:

$$DSTOR + Q_{out} = Q_{in} + P - E + ADJ + Q_{lake} \quad \text{--- } <11 >$$

where

Q_{in} = Inflows from upper lake structure
 Q_{lake} = Watershed inflows
 P = Lake precipitation
 E = Lake evaporation
 ADJ = Inflow adjustment term -all unestimated flows and errors of water budget
 $DSTOR$ = Change in lake storage
 Q_{out} = Outflow to lower lake

The model can operate in three different modes. Each routing mode differs in the way the terms on the right side of Equation 11 are determined:

●**Simulation Mode:** All terms on the right side of Equation 11 are input from historical records. The model routes the flow through the lakes and simulates only the management aspect of the system. In other words, the model predicts the structure flows and lake stages based on historical hydrologic input. This mode is suitable for evaluating situations where only the management variables change, such as in evaluating a change in the regulation schedules. The implicit assumption is that a change in the management variables will not affect the historical hydrologic variables.

●**Forecasting Mode:** All terms on the right side of Equation 11, except rainfall, are predicted using rainfall as a conditional dependent variable. Rainfall is predetermined. Variation of the management rules, such as the regulation schedules, can also be input as additional conditional variables. This mode is most general, as the ultimate objective of a routing model is to be able to forecast under any conditions.

●**Calibration Mode:** This mode is essentially the same as the previous one except that historical rainfall is used. This mode is used to calibrate and verify the forecasting capability of the model by comparing historical to model results. An optimization option is provided to aid in the calibration of the parameters.

On the left side of Equation 11, the unknowns $DSTOR$ and Q_{out} are both a function of the lake stage

and thus stage is the only unknown. However, Q_{out} is dependent on both the current and lower lake stages due to backwater effect and constraints imposed by management rules. Since the tailwater (lower lake) stage is unknown a priori in the current time step, an iteration technique is used to converge the estimated and computed stages. The process is accomplished by first estimating Q_{out} from the stages in the last time step, computing the storage change $DSTOR$ from Equation 11, and then updating the stages from the following equation:

$$h_t = h_{t-1} + DSTOR / AREA_{(h_t)} \text{-----} <12>$$

where

- h_t = Stage in current time step in ft msl
- h_{t-1} = Stage in last time step in ft msl
- $DSTOR$ = Computed storage change from Equation <11> in acre-ft
- $AREA_{(h_t)}$ = Lake surface area in acres from rating equation as a function of h_t

The updated stages are used to revise Q_{out} and $DSTOR$ by Equation 11 and a small number of iterations per time step are needed to converge the stages.

One particular feature of the model is that it can handle missing records. Missing records are signaled by "?" tags in the input files. When a user specified missing gap is encountered, the model will skip the routing computation, reinitialize the stages to historical at the end of the gap, but continue soil moisture accounting throughout. This provision removes major noise in the input data so that they will not be included in the calibration.

I. Parameter Optimization Procedures

A univariant gradient search procedure is used to aid in the calibration of the watershed parameters. For ungaged watersheds where stream flow data are unavailable, the objective function $f(x)$ is defined as the sum of square deviations of the observed and computed stages of the receiving lakes. For gaged watersheds, $f(x)$ is defined as the sum of square deviations of the logarithmic transformed flows, where x is one of the watershed parameters (SCOEF, HMAX, PCOEF, and ROOT) to be adjusted. The adjustment ∂x for x is obtained from the following formula:

$$\partial x = \alpha [-f(x) / f'(x)] \text{-----} <13>$$

where

- ∂x = Adjustment for x
- x = Parameter value

- = One of 52 parameters: SCOEF, HMAX, PCOEF, or ROOT in thirteen watersheds
- α = Adjustment factor between 0 and 1
- $f(x)$ = Objective function value from model run

$$= \sum [h_{obs} - h_{sim}(x)]^2 \text{ or}$$

$$\sum [\log Q_{obs} - \log Q_{sim}(x)]^2$$

h_{obs}, h_{sim} = Observed and simulated stages of receiving lakes for ungaged watersheds

Q_{obs}, Q_{sim} = Observed and simulated flows for gaged watersheds

$f'(x)$ = Gradient (Partial derivative) of $f(x)$ with respect to parameter x

Since it is not possible to evaluate $f'(x)$ analytically, the gradient (partial derivative) $f'(x)$ is determined numerically from the following equation:

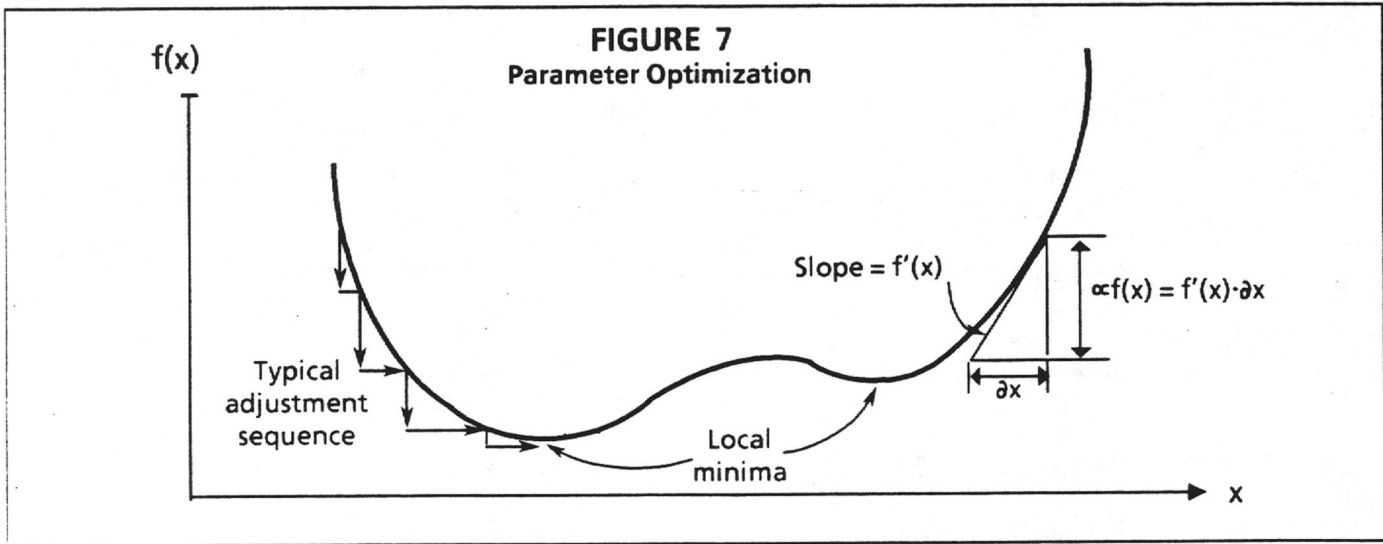
$$f'(x) = [f(x) - f^*(x)] / [x - x^*] \text{-----} <14>$$

where

- $f(x)$ = Objective function value from current model run
- $f^*(x)$ = Objective function value from previous model run
- x = Parameter value of x at current run
- = $x^* + \partial x^*$
- x^* = Parameter value of x at previous run
- ∂x^* = Adjustment for x in previous run

Each adjustment iteration requires running the routing model for the entire simulation period (1970 to 1980) one time to renew the objective function value. The new value $f(x)$ and the old value $f^*(x)$ are used to calculate the gradient $f'(x)$ by Equation 14, which is then used to determine a new adjustment ∂x for x by Equation 13. The procedure effectively moves x down gradient of the objective function (Figure 7). Each parameter is adjusted independently by keeping the other parameters constant. To minimize interactions among the watersheds, optimization proceeds from the most upstream watershed to the downstream ones.

All parameters are bounded at the lower end with a non-negative constraint but there is no constraint in the upper end. If α equals one, Equation 13 will be analogous to the Newton-Raphson Method. An adjustment factor α less than one is used to avoid overstepping the local minimum. Overstepping is indicated by an increase, rather than a decrease, in the objective function. When overstepping occurs, parameter x returns to the previous value and a smaller adjustment of x is made by reducing α until an improvement of the objective function is achieved. In



the program, α is first initialized to 0.75 and then reduced by a factor of 50 whenever no improvement condition is encountered. The optimization of a parameter is considered successful when x does not change by more than one percent from the previous value.

Among the parameters there are strong interactions. The optimized value of a parameter depends strongly on the values of the other parameters. Furthermore, the gradient search technique can only locate the local minimum and there can be multiple local minima. Thus the procedure can result in multiple sets of optimized parameter values.

The optimization procedure therefore will not "automatically calibrate" the parameters. Judgment is needed to fine tune the parameters by assuring that their values fall within reasonable ranges, and that the mathematically optimized results are truly the optimization desired. This was achieved by judiciously selecting the initial values, fixing the values of some parameters, and optimizing the parameters in different sequences. The final optimized parameter values are then entered manually into the routing program (within DATA statements in subroutine INFLOW -- Pages 31 and 32).

VI. MODEL USAGE

The model can operate in three different modes which have been described in detail in Section V. Briefly, a "simulation" mode reads in historical hydrologic data directly and is suitable for evaluating the management components of the system. A "calibration" mode reads in the historical rainfall; all other terms are predicted using rainfall as the

dependent variable. An optimization option is provided to aid in the calibration of the parameters. A "forecasting" mode is essentially the same as the "calibration" mode except that rainfall is predetermined by the user.

All three routing modes are included in one version of the routing program KROUTE. A brief description of the subroutines in KROUTE is summarized in Table 6. Complimentary to KROUTE are a water budget program KBUDGET and a plotting program KPLOT. The input and output file requirements for all three programs are listed in Table 7. Files are attached or created dynamically in the programs so that the user need not explicitly manage the files.

The water budget computation program KBUDGET serves two purposes: One is to verify the historical data which can be achieved by examining the water budget results. Another is to preprocess the input data files for KROUTE. The water budget program is used only initially and is not required in routine application of the routing program.

The plotting program KPLOT presents the results in a form suitable for interpretation and calibration. Several options are available to plot the results and are included in one version of the plotting program. KPLOT normally should run immediately after KROUTE because it uses the output data from KROUTE as input.

The computer time requirement to run KROUTE to simulate eleven years of record is about 7 CPU minutes; however, it takes considerable CPU time in parameter optimization runs. To optimize one parameter for each of the thirteen watersheds requires

Table 6
Subroutine Functions

KROUTE	This main program initializes data, attaches or creates input data files dynamically, and performs routing from lake to lake.
BUILDF	Builds input data file NTAPE5 for "forecasting" run according to user specified rainfall scenario and initial stage conditions.
INFLOW	Computes watershed discharges, evaporation, and performs soil moisture accounting needed for base flow and direct runoff simulation.
OPT	Performs optimization for watershed parameters by a gradient search technique.
STORAGE	Calculates stage area and stage storage relationships from rating equations.
DISCH	Computes flow from spillway, culvert, and canal rating equations.
BLOCK DATA	Contains lake, structure, and miscellaneous parameter data needed for other subroutines.

about 100 iterations and 10 CPU hours. Fortunately optimization runs are just needed initially.

One particular feature of the routing and plotting programs is that they can handle missing records. Missing records are indicated by "?" tags in the input data records (column 3 of TAPE5 and TAPE6). When a missing gap is encountered, the model will skip the routing computation, initialize the stages to historical, but continue soil moisture accounting throughout. Similarly, KPLOT will skip plotting the missing period signaled by "?" in the routing results. This

provision removes major noise from the input data so that they will not be included in the calibration. Missing records or records with errors can be detected by inspecting the water budget results from KBUDGET runs. If a missing gap is small, the gap is filled in by an interpolation or correlation method; otherwise, a "?" tag is inserted thus allowing the routing program to skip computation.

Example applications of KROUTE and KPLOT are illustrated in pages 70 through 77 with partial listing of the output. The results are plotted in pages

Table 7
Input and Output Data Files

<u>Program</u>	<u>Input files</u>	<u>Contents</u>	<u>Output files</u>	<u>Contents</u>
KBUDGET	Tape1	Raw hydrologic data	Tape5	Input data for KROUTE
	Tape2	Raw hydrologic data	Tape6	Input data for KROUTE
KROUTE	Tape4	70-80 Regulation schedules	Tape7	Output stages
	Ntape4	Current regulation schedules	Tape8	Output stages
	Tape5	From KBUDGET run	Tape9	Output flows
	Tape6	From KBUDGET run	Tape11-19	Detailed output by lakes
	Ntape5	Created in "forecasting" run		
KPLOT	Tape2	Raw hydrologic data		
	Tape8	From KROUTE run		
	Tape9	From KROUTE run		

78 through 103 All three programs are listed in the Appendix. As illustrated in the examples, the programs are user friendly with minimum data input requirements.

VII. MODEL VERIFICATION

The reliability of a simulation model should not be judged solely by the sophistication level of its methodology, but also by its performance. Calibration is an exercise to match the historical records by adjusting certain calibration parameters. Being able to match the history is a necessary but insufficient criterion to verify a computer model; for example, any historical record can be fitted by a complex polynomial equation with enough terms but the equation cannot be used for forecasting. Verification is a process to show that a properly calibrated model is dependable in extrapolation. Calibration should be carried out independent of verification; however, the two processes are often mixed in actual practice.

A simulation run of the model for the period 1970-80 is shown in Pages 78 through 98. In this mode, only the management component is simulated. Several events have substantial deviations, which can be attributed to data errors, non-uniform rainfall assumption, variations of the management from the usual rules, and other unknown reasons. There are two gaps in the simulation: one in the year of 1977 and the other in April 1979.

The first gap is due to an unusual drawdown schedule in Lake Kissimmee where a weir structure was placed temporarily in C-37. The second gap is due to missing records in S-59 and S-61. Data errors were identified

in other periods but since their durations were short, they were not excluded in the simulation. In general, however, the simulation is in excellent agreement with the historical records. This shows that the model's routing scheme is correct, and the simulation of the structure flow and management is excellent.

The results of a "calibration" run for the same period is presented in Pages 78 through 98. The plots are shown directly below the "simulation" plots for comparison. In comparison, the deviations between observed and simulated stages are much larger in a "calibration" run than in a "simulation" run. Since the "simulation" run verifies that the simulation of the structure flow and management is excellent, additional deviations in a "calibration" run must have originated from some other sources. Watershed inflow was identified to be the principal source of error. Future improvement of the model should focus on this area.

The model has been used routinely to forecast the lake conditions at the beginning of each month using rainfall as a dependent conditional variable. Typically, five rainfall scenarios ranging from 50% to 150% normal rainfall were used. At the end of the month, the observed stages were compared with the predicted ones with respect to the appropriate rainfall scenario. This provides a true verification process because the future results are unknown a priori. The results are shown in Table 8. Despite the simplistic scenario assumption of uniform rainfall distribution (spatial and temporal), the observed and predicted stages are in close agreement. It verifies that the model is dependable in short term projection, though in long term projection improvement is needed.

APPENDICES

Appendix 1: Dictionary of Program Symbols (KROUTE)

Symbol subscript designation

1=Alligator, 2=Myrtle, 3=Hart, 4=Gentry, 5=East Tohopekaliga,
6=Tohopekaliga, 7=Cypress, 8=Hatchineha, 9=Kissimmee, 10=Boggy Creek,
11=Shingle Creek, 12=Reedy Creek, 13=Catfish Creek

A	Area function.
A1 to A4	Coefficients for allowable gate opening calculation.
ADJ1 to ADJ9	Inflow adjustment (Ungaged watershed inflows).
AL	Length of spillway crest or culvert length.
ALLI to KISS	Lake arrays containing parameters for stage area and stage storage computation.
ALPH	Adjustment coefficient in optimization.
AMC	Array containing antecedent rainfall total for 1 to 6 months.
AR	Conveyance coefficient for part full pipe.
AR1	Conveyance coefficient for full pipe.
AREA1 to AREA9	Lake areas in acres.
B1 to B5	Parameters for spillway flow calculation.
BOGG,CATF	Historical flow data for Boggy and Catfish Creeks.
C1 to C5	Polynomial coefficients in storage function.
CEL	Culvert invert or spillway crest elevation in msl.
COEF,COEF1	Parameter value in optimization runs.
D1 to D5	Polynomial coefficients in area function.
DA	Gradient of storage function.
DIAM	Culvert diameter.
DVOL1 to DVOL9	Excess storage above regulation schedule in acre-feet.
DSTG1 to DSTG9	Change in stage.
DSTOR1 to DSTOR9	Available storage release in acre-feet.
EBOGG,ECATF	Computed flows for Boggy and Catfish Creeks.
EREED,ESHIN	Computed flows for Reedy and Shingle Creeks.
ESTG1 to ESTG9	Computed stages for Lakes Alligator to Kissimmee.
ET	Watershed evapotranspiration function.
EVAP	Daily pan evaporation.
FRICT	Manning coefficient.
FUNC,FUNC1	Objective function in optimization runs.
GO	Calculated gate opening in feet.
HEAD	Hydraulic head difference.
HMAX	Maximum depth at which seepage ceases.
HW	Headwater elevation.
I,J	Sequential day index.
ICASE	Rainfall scenario case index
IDATE,JDATE	Calendar date strings.
IFOR	Routing option indicator
IJDAY	Breakpoint time index for regulation schedules.
IM,JM,KM	Calendar month indexes.
IOPT	Optimization option index.
IOSTAT	IO status used to control reading of input files.
IPAR	Optimization parameter selection index.
IPRINT	Printout option index.
ITYPE	Structure types: 1=Spillway, 2=Culvert.
IY,JY	Calendar year indexes.
JDATE	Calendar date string.

Appendix 1: Dictionary of Program Symbols (KROUTE)

LAKNAME	Lake names
MONTH	Array containing number of days for each calendar month.
NCASE	Number of rainfall scenarios.
NMONTH	Number of months in forecast.
NPIPE	Number of gates or culverts in structure.
PCOEFF	Pan evaporation coefficient.
Q,Q1,Q2,Q3,Q4	Calculated structure flows in cfs.
QMAX	Maximum allowable discharge in structure.
QRAIN	SCS direct runoff function.
RAIN	Normal rainfall in study area.
RAIN1 to RAIN9	Daily rainfall
RAT	Normal rainfall multiplication factors in forecasting runs.
RATIO	Rainfall multiplication ratios.
REED,SHIN	Historical flow data for Reedy and Shingle Creeks.
REG	Regulation schedule array containing breakpoint values for schedules between 1970 to 1985.
RES1 to RES9	Residuals for Lakes Alligator to Kissimmee.
ROOT	Deep root zone depth below which ET ceases.
RSTG1 to RSTG9	Regulation schedules.
S57 to S65	Computed flow for S-57 to S-65.
SCOEF	Seepage coefficient.
SEEP	Seepage or base flow function.
SLOPE	Hydraulic gradient.
SMD	Global soil moisture deficit.
SMD1 to SMD13	Soil moisture deficit for watersheds 1 to 13.
SOIL	Soil moisture accounting function.
STAGMIN	Minimum surveyed stage.
STG1 to STG9	Historical stage data for Lakes Alligator to Kissimmee.
STG1E to STG9E	Interim stages in routing iterations.
STGMAX	Maximum surveyed stage.
STOR	Calculated storage from storage function.
STR57 to STR65	Structure arrays containing parameters for structure flow computation.
TEVAP	Array containing normal monthly pan evaporation in inches.
TRAIN	Array containing normal monthly rainfall in inches.
TW	Tailwater elevation.
V	Storage function.
VOL1 to VOL9	Lake storage in acre-feet.
XAR	Linealized adjustment factor for conveyance coefficient for part full pipe.
Y1	Frequency correlation equation function.

Appendix 2: Program Listing (KROUTE)

```

PROGRAM KROUTE(INPUT)
C *****
C *** ROUTING MODEL FOR KISSIMMEE CHAIN OF LAKES
C *** TAPE4=HISTORICAL REGULATION SCHEDULES(INPUT DATA)
C *** TAPE5,TAPE6=INPUT HYDROLOGIC DATA GENERATED BY "KBUDGET"
C *** NTAPE4=CURRENT REGULATION SCHEDULE (INPUT DATA)
C *** NTAPE5=INPUT INITIAL STAGES AND RAINFALL SCENARIO
C *** (NTAPE4,NTAPE5 WILL REPLACE TAPE4,TAPE5 IN FORCAST OPTION)
C *** TAPE7=SUMMARY OUTPUT OF HISTORICAL STAGE VERSUS COMPUTED STAGE
C *** TAPE8=SUMMARY OUTPUT OF REGULATION STAGE VERSUS COMPUTED STAGE
C *** TAPE9=SUMMARY OUTPUT OF COMPUTED DISCHARGE
C *** TAPE11 THRU TAPE19=DETAILED OUTPUT PER LAKE
C *****
COMMON /A/STR57(15),STR58(15),STR59(15),STR60(15),STR61(15),
$ STR62(15),STR63(15),STR65(15)
COMMON /B/ALLI(13),MYRT(13),HART(13),GENT(13),
$ ETOH(13),TOHO(13),CYPR(13),HATC(13),KISS(13)
COMMON /C/REG(9,15),IJDAY(9,15)
COMMON /D/TEVAP(12),TRAIN(12),AMC(6),MONTH(12)
COMMON /E/ EVAP(365),RAIN1(365),RAIN2(365),
$ RAIN3(365),RAIN4(365),RAIN5(365),RAIN6(365),
$ RAIN7(365),RAIN8(365),RAIN9(365),
$ BOGG(365),CATF(365),REED(365),SHIN(365),
$ EBOGG(365),ECATF(365),EREED(365),ESHIN(365),
$ RES1(365),RES2(365),RES3(365),RES4(365),RES5(365),
$ RES6(365),RES7(365),RES8(365),RES9(365)
COMMON /F/ STG1(0:365),STG2(0:365),STG3(0:365),
$ STG4(0:365),STG5(0:365),STG6(0:365),STG7(0:365),
$ STG8(0:365),STG9(0:365),S65(365),
$ ESTG1(0:365),ESTG2(0:365),ESTG3(0:365),ESTG4(0:365),
$ ESTG5(0:365),ESTG6(0:365),ESTG7(0:365),ESTG8(0:365),
$ ESTG9(0:365)
REAL MYRT,KISS,RAT(6)
CHARACTER LAKNAME*4,IDATE(365)*8,JDATE*8
C ***
PRINT*,'ROUTING OPTION? (1=FORECAST,2=SIMULATION,3=CALIBRATION)'
READ*,IFOR
IF(IFOR.EQ.3)THEN
PRINT*,'OPTIMIZATION OF PARAMETERS NEEDED? 1=NO,2=YES'
READ*,IOPT
IF(IOPT.EQ.2)THEN
PRINT*,'SELECT ONE TO OPTIMIZE:1=SCOEF,2=HMAX,3=PCOEF,4=ROOT'
READ*,IPAR
ENDIF
ELSE
IOPT=1
ENDIF
C *** PRINT HEADING ON OUTPUT FILE ***
IPRINT=0
WRITE(7,('(1H1,4X,4HDATE,2X,4HSTG1,1X,5HESTG1,2X,4HSTG2,1X,
$ 5HESTG2,2X,4HSTG3,1X,5HESTG3,2X,4HSTG4,1X,5HESTG4,2X,
$ 4HSTG5,1X,5HESTG5,2X,4HSTG6,1X,5HESTG6,2X,4HSTG7,1X,
$ 5HESTG7,2X,4HSTG8,1X,5HESTG8,2X,4HSTG9,1X,5HESTG9,
$ 3X,3HS65,2X,4HES65)')

```

Appendix 2: Program Listing (KROUTE)

```

WRITE(8, '(1H1, 4X, 4HDATE, 1X, 5HRSTG1, 1X, 5HESTG1, 1X, 5HRSTG2, 1X,
$ 5HESTG2, 1X, 5HRSTG3, 1X, 5HESTG3, 1X, 5HRSTG4, 1X, 5HESTG4, 1X,
$ 5HRSTG5, 1X, 5HESTG5, 1X, 5HRSTG6, 1X, 5HESTG6, 1X, 5HRSTG7, 1X,
$ 5HESTG7, 1X, 5HRSTG8, 1X, 5HESTG8, 1X, 5HRSTG9, 1X, 5HESTG9,
$ 3X, 3HS65, 2X, 4HES65)')
WRITE(9, '(1H1, 4X, 4HDATE, 8X, 2X, 4HS-58, 2X, 4HS-57, 2X, 4HS-62, 2X,
$ 4HS-60, 2X, 4HS-63, 2X, 4HS-59, 2X, 4HS-61, 2X, 4HS-65, 2X, 4HSHIN,
$ 2X, 4HBOGG, 2X, 4HCATF, 2X, 4HREED, 2X, 4HC-36, 2X, 4HC-37)')
IF(IPRINT.EQ.1)THEN
WRITE(11, '(1H1, 4X, 4HDATE, 3X, 4HRAIN, 2X, 4HEVAP, 2X, 4HRSTG, 3X, 3HSTG,
$ 2X, 4HESTG, 5X, 4HAREA, 6X, 3HVOL, 4X, 5HDSTOR, 4X, 3HS60, 4X, 3HS58, 4X,
$ 3HADJ, T120, 11HTAPE11=ALLI)')
WRITE(12, '(1H1, 4X, 4HDATE, 3X, 4HRAIN, 2X, 4HEVAP, 2X, 4HRSTG, 3X, 3HSTG,
$ 2X, 4HESTG, 5X, 4HAREA, 6X, 3HVOL, 4X, 5HDSTOR, 4X, 3HS57, 4X, 3HS58, 4X,
$ 3HADJ, T120, 11HTAPE12=MYTR)')
WRITE(13, '(1H1, 4X, 4HDATE, 3X, 4HRAIN, 2X, 4HEVAP, 2X, 4HRSTG, 3X, 3HSTG,
$ 2X, 4HESTG, 5X, 4HAREA, 6X, 3HVOL, 4X, 5HDSTOR, 4X, 3HS57, 4X, 3HS62, 4X,
$ 3HADJ, T120, 11HTAPE13=HART)')
WRITE(14, '(1H1, 4X, 4HDATE, 3X, 4HRAIN, 2X, 4HEVAP, 2X, 4HRSTG, 3X, 3HSTG,
$ 2X, 4HESTG, 5X, 4HAREA, 6X, 3HVOL, 4X, 5HDSTOR, 4X, 3HS60, 4X, 3HS63, 4X,
$ 3HADJ, T120, 11HTAPE14=GENT)')
WRITE(15, '(1H1, 4X, 4HDATE, 3X, 4HRAIN, 2X, 4HEVAP, 2X, 4HRSTG, 3X, 3HSTG,
$ 2X, 4HESTG, 5X, 4HAREA, 6X, 3HVOL, 4X, 5HDSTOR, 4X, 3HS62, 4X, 3HS59, 3X,
$ 4HBOGG, 4X, 3HADJ, T120, 11HTAPE15=ETOH)')
WRITE(16, '(1H1, 4X, 4HDATE, 3X, 4HRAIN, 2X, 4HEVAP, 2X, 4HRSTG, 3X, 3HSTG,
$ 2X, 4HESTG, 5X, 4HAREA, 6X, 3HVOL, 4X, 5HDSTOR, 4X, 3HS59, 4X, 3HS61, 3X,
$ 4HSHIN, 4X, 3HADJ, T120, 11HTAPE16=TOHO)')
WRITE(17, '(1H1, 4X, 4HDATE, 3X, 4HRAIN, 2X, 4HEVAP, 2X, 4HRSTG, 3X, 3HSTG,
$ 2X, 4HESTG, 5X, 4HAREA, 6X, 3HVOL, 4X, 5HDSTOR, 4X, 3HC36, 4X, 3HS61, 4X,
$ 3HS63, 3X, 4HREED, 4X, 3HADJ, T120, 11HTAPE17=CYPR)')
WRITE(18, '(1H1, 4X, 4HDATE, 3X, 4HRAIN, 2X, 4HEVAP, 2X, 4HRSTG, 3X, 3HSTG,
$ 2X, 4HESTG, 5X, 4HAREA, 6X, 3HVOL, 4X, 5HDSTOR, 4X, 3HC36, 4X, 3HC37, 3X,
$ 4HCATF, 3X, 4HREED, 4X, 3HADJ, T120, 11HTAPE18=HATC)')
WRITE(19, '(1H1, 4X, 4HDATE, 3X, 4HRAIN, 2X, 4HEVAP, 2X, 4HRSTG, 3X, 3HSTG,
$ 2X, 4HESTG, 5X, 4HAREA, 6X, 3HVOL, 4X, 5HDSTOR, 4X, 3HC37, 4X, 3HS65, 3X,
$ 4HES65, 4X, 3HADJ, T120, 11HTAPE19=KISS)')
ENDIF
C *** OPEN AND BUILD INPUT FILES ***
IF(IFOR.EQ.1)THEN
CALL PF('ATTACH', 'NTAPE4', 'NTAPE4', 'UN', 'AFAN')
OPEN(4, FILE='NTAPE4', STATUS='OLD')
OPEN(5, FILE='NTAPE5')
CALL BUILD
ELSE
CALL PF('ATTACH', 'TAPE4', 'TAPE4', 'UN', 'AFAN')
CALL PF('ATTACH', 'TAPE5', 'TAPE5', 'UN', 'AFAN')
CALL PF('ATTACH', 'TAPE6', 'TAPE6', 'UN', 'AFAN')
ENDIF
C *** READ INITIAL STAGES AND ANTECEDENT RAINFALL ***
ICASE=0
10 REWIND 4
REWIND 5
REWIND 6
IF(IFOR.EQ.1)THEN

```

Appendix 2: Program Listing (KROUTE)

```

        ICASE=ICASE+1
        READ(5,15)AMC(1),AMC(6),NMONTH,NCASE,(RAT(N),N=1,NCASE)
15      FORMAT(2F6.2,2I2,6F6.2)
        READ(5,'(9F6.2)')ESTG1(0),ESTG2(0),ESTG3(0),ESTG4(0),
$      ESTG5(0),ESTG6(0),ESTG7(0),ESTG8(0),ESTG9(0)
        RATIO=RAT(ICASE)
        DO 18 M=2,6
18      AMC(M)=AMC(6)-AMC(6)/6.*(6-M)
        ELSE
        READ(5,20)ESTG1(0),ESTG2(0),ESTG3(0),ESTG4(0)
        READ(6,20)ESTG5(0),ESTG6(0),ESTG7(0),ESTG8(0),ESTG9(0)
20      FORMAT(27X,5(6X,F6.2,7X))
        AMC(1)=TRAIN(12)
        DO 25 M=2,6
25      AMC(M)=AMC(M-1)+TRAIN(13-M)
        ENDIF
C *** GLOBAL INITIALIZATION ***
        DO 30 M=365,334,-1
        RAIN1(M)=RAIN2(M)=RAIN3(M)=RAIN4(M)=AMC(1)/30.
        RAIN5(M)=RAIN6(M)=RAIN7(M)=RAIN8(M)=RAIN9(M)=AMC(1)/30.
30      CONTINUE
        STG1E=ESTG1(0)
        STG2E=ESTG2(0)
        STG3E=ESTG3(0)
        STG4E=ESTG4(0)
        STG5E=ESTG5(0)
        STG6E=ESTG6(0)
        STG7E=ESTG7(0)
        STG8E=ESTG8(0)
        STG9E=ESTG9(0)
        GOTO 60
C *** INITIALIZE LAKE STAGES FOR NEXT YEAR'S SIMULATION ***
50      ESTG1(0)=ESTG1(365)
        ESTG2(0)=ESTG2(365)
        ESTG3(0)=ESTG3(365)
        ESTG4(0)=ESTG4(365)
        ESTG5(0)=ESTG5(365)
        ESTG6(0)=ESTG6(365)
        ESTG7(0)=ESTG7(365)
        ESTG8(0)=ESTG8(365)
        ESTG9(0)=ESTG9(365)
C *** READ LAKE REGULATION SCHEDULE ***
60      DO 90 I=1,9
        READ(4,70)LAKNAME.ID.IYEAR
70      FORMAT(A4,I2,1X,I4)
        READ(4,80)(IJDAY(ID,J),REG(ID,J),J=1,10)
80      FORMAT(10(I6,F6.2))
90      CONTINUE
C ***
C *** READ INPUT HYDROLOGIC DATA ***
C ***
        DO 200 I=1,365
        IF(IFOR.EQ.2 .OR. IFOR.EQ.3)THEN
        READ(5,110,END=230,IOSTAT=IO)

```

Appendix 2: Program Listing (KROUTE)

```

$  IDATE(I),EVAP(I),SHIN(I).BOGG(I),
$  RAIN1(I),STG1(I),RES1(I),RAIN2(I),STG2(I),RES2(I),
$  RAIN3(I),STG3(I),RES3(I),RAIN4(I),STG4(I),RES4(I)
  READ(6,120)JDATE,REED(I),CATF(I),S65(I),
$  RAIN5(I),STG5(I),RES5(I),RAIN6(I),STG6(I),RES6(I),
$  RAIN7(I),STG7(I),RES7(I),RAIN8(I),STG8(I),RES8(I),
$  RAIN9(I),STG9(I),RES9(I)
110  FORMAT(1X,A8,F6.2,2F6.0,4(F6.2,F6.2,F7.0))
120  FORMAT(1X,A8,3F6.0,5(F6.2,F6.2,F7.0))
C --- CHECK IF DATES MATCH ---
  IF(IDATE(I)(1:2).NE.JDATE(1:2) .OR.
  $  IDATE(I)(4:5).NE.JDATE(4:5) .OR.
  $  IDATE(I)(7:8).NE.JDATE(7:8) )THEN
    PRINT*,'DATES NOT MATCH !!!'
    PRINT*,'I.IDATE,JDATE'
    WRITE(*,140)I,IDATE(I),JDATE
140  FORMAT(I3,1X,2A9)
    STOP
  ENDIF
  ELSE
    READ(5,150.END=230)IDATE(I),RAIN
150  FORMAT(1X,A8,F6.3)
    RAIN1(I)=RAIN2(I)=RAIN3(I)=RAIN4(I)=RAIN*RATIO
    RAIN5(I)=RAIN6(I)=RAIN7(I)=RAIN8(I)=RAIN9(I)=RAIN*RATIO
  ENDIF
  IF(IFOR.EQ.1 .OR. IFOR.EQ.3)THEN
    JDATE=IDATE(I)
    READ(JDATE,'(I2.1X,I2.1X,I2)')JM,JD,JY
    CALL INFLOW(I,JM)
  ELSE
    ESHIN(I)=SHIN(I)
    EBOGG(I)=BOGG(I)
    EREED(I)=REED(I)
    ECATF(I)=CATF(I)
  ENDIF
  NDAY=I
200  CONTINUE
C ***
C *** BEGIN ROUTING FROM LAKE ALLIGATOR TO LAKE KISSIMMEE ***
C ***
230  DO 500 J=1,NDAY
    JDATE=IDATE(J)
    READ(JDATE,'(I2.1X,I2.1X,I2)') JM,JD,JY
C *** SKIP COMPUTATION FOR MISSING RECORDS AND REINITIALIZE STAGES ***
    IF(JDATE(3:3).EQ.'?')THEN
      ESTG1(J)=STG1(J)
      ESTG2(J)=STG2(J)
      ESTG3(J)=STG3(J)
      ESTG4(J)=STG4(J)
      ESTG5(J)=STG5(J)
      ESTG6(J)=STG6(J)
      ESTG7(J)=STG7(J)
      ESTG8(J)=STG8(J)
      ESTG9(J)=STG9(J)

```

Appendix 2: Program Listing (KROUTE)

```

GOTO 275
ENDIF
C *** ITERATE LAKE STAGES PER TIME STEP ***
DO 270 ITER=0,3
JADJ=MIN(1,ITER)
STG1E=STG1E + ( ESTG1(J-1+JADJ)-STG1E )*.5
STG2E=STG2E + ( ESTG2(J-1+JADJ)-STG2E )*.5
STG3E=STG3E + ( ESTG3(J-1+JADJ)-STG3E )*.5
STG4E=STG4E + ( ESTG4(J-1+JADJ)-STG4E )*.5
STG5E=STG5E + ( ESTG5(J-1+JADJ)-STG5E )*.5
STG6E=STG6E + ( ESTG6(J-1+JADJ)-STG6E )*.5
STG7E=STG7E + ( ESTG7(J-1+JADJ)-STG7E )*.5
STG8E=STG8E + ( ESTG8(J-1+JADJ)-STG8E )*.5
STG9E=STG9E + ( ESTG9(J-1+JADJ)-STG9E )*.5
C *** LAKE ALLIGATOR ***
CALL STORAGE(ALLI,STG1E,VOL1,AREA1)
CALL DISCH(STR58,STG1E,STG2E,Q58)
CALL DISCH(STR60,STG1E,STG4E,Q60)
Q60A=-3357.+57.78*STG1E
Q60=MIN(Q60A,Q60)
CALL REGSTG(ALLI,JDATE,RSTG1)
DVOL1=(STG1E-RSTG1)*AREA1
IF(DVOL1.LE.0.)THEN
  S58=0.
  S60=0.
ELSE
  IF(DVOL1.LE.Q60*1.9835)THEN
    S60=DVOL1/1.9835
    S58=0.
  ELSE
    S60=Q60
    S58=MIN((DVOL1/1.9835-S60),Q58)
  ENDIF
ENDIF
ADJ1=RES1(J)
DSTOR1=(RAIN1(J)-EVAP(J)*.8)*8258/12.-
$ (S58+S60)*1.9835+ADJ1
DSTG1=DSTOR1/AREA1
ESTG1(J)=ESTG1(J-1)+DSTG1
C *** LAKE MYRTLE AND MARY JANE ***
CALL STORAGE(MYRT,STG2E,VOL2,AREA2)
CALL DISCH(STR57,STG2E,STG3E,Q57)
CALL REGSTG(MYRT,JDATE,RSTG2)
DVOL2=(STG2E-RSTG2)*AREA2+S58*1.9835
IF(DVOL2.LE.0.)THEN
  S57=0.
ELSE
  S57=MIN(Q57,DVOL2/1.9835)
ENDIF
ADJ2=RES2(J)
DSTOR2=(RAIN2(J)-EVAP(J)*.8)*1732/12.+
$ (S58-S57)*1.9835+ADJ2
DSTG2=DSTOR2/AREA2
ESTG2(J)=ESTG2(J-1)+DSTG2

```

Appendix 2: Program Listing (KROUTE)

```

C *** LAKE HART AND MARY JANE ***
  CALL STORAGE(HART,STG3E,VOL3,AREA3)
  CALL DISCH(STR62,STG3E,STG5E,Q62)
  CALL REGSTG(HART,JDATE,RSTG3)
  DVOL3=(STG3E-RSTG3)*AREA3+S57*1.9835
  IF(DVOL3.LE.0.)THEN
    S62=0.
  ELSE
    S62=MIN(Q62,DVOL3/1.9835)
  ENDIF
  ADJ3=RES3(J)
  DSTOR3=(RAIN3(J)-EVAP(J)*.8)*3643/12.+
$   (S57-S62)*1.9835+ADJ3
  DSTG3=DSTOR3/AREA3
  ESTG3(J)=ESTG3(J-1)+DSTG3
C *** LAKE GENTRY ***
  CALL STORAGE(GENT,STG4E,VOL4,AREA4)
  CALL DISCH(STR63,STG4E,56.50,Q63)
  CALL REGSTG(GENT,JDATE,RSTG4)
  DVOL4=(STG4E-RSTG4)*AREA4+S60*1.9835
  IF(DVOL4.LE.0.)THEN
    S63=0.
  ELSE
    S63=MIN(Q63,DVOL4/1.9835)
  ENDIF
  ADJ4=RES4(J)
  DSTOR4=(RAIN4(J)-EVAP(J)*.8)*1776/12.+
$   (S60-S63)*1.9835+ADJ4
  DSTG4=DSTOR4/AREA4
  ESTG4(J)=ESTG4(J-1)+DSTG4
C *** LAKE EAST TOHO ***
C --- CONSIDER WEIR STRUCTURE BELOW S-59 AFTER 1979
  IF(JY.GE.79)THEN
    STG6B=MAX(STG6E,51.1)
    DO 240 ITIME=1,10
      CALL DISCH(STR59,STG5E,STG6B,Q59)
      STG6B=MAX(STG6E,51.1)+(Q59/(.50*30.*(STG6B-51.0)*SQRT(64.4)))*.2
240    CONTINUE
    ELSE
      STG6B=STG6E
    ENDIF
C ---
  CALL STORAGE(ETOH,STG5E,VOL5,AREA5)
  CALL DISCH(STR59,STG5E,STG6B,Q59)
  CALL REGSTG(ETOH,JDATE,RSTG5)
  DVOL5=(STG5E-RSTG5)*AREA5+(EBOGG(J)+S62)*1.9835
  IF(DVOL5.LE.0.)THEN
    S59=0.
  ELSE
    S59=MIN(Q59,DVOL5/1.9835)
  ENDIF
  ADJ5=RES5(J)
  DSTOR5=(RAIN5(J)-EVAP(J)*.8)*12600/12.+
$   (EBOGG(J)+S62-S59)*1.9835+ADJ5

```

Appendix 2: Program Listing (KROUTE)

```

DSTG5=DSTOR5/AREA5
ESTG5(J)=ESTG5(J-1)+DSTG5
C *** LAKE TOHO ***
CALL STORAGE(TOHO,STG6E,VOL6,AREA6)
CALL DISCH(STR61,STG6E,STG7E,Q61)
CALL REGSTG(TOHO,JDATE,RSTG6)
DVOL6=(STG6E-RSTG6)*AREA6+(ESHIN(J)+S59)*1.9835
IF(DVOL6.LE.0.)THEN
    S61=0.
ELSE
    S61=MIN(Q61,DVOL6/1.9835)
ENDIF
ADJ6=RES6(J)
DSTOR6=(RAIN6(J)-EVAP(J)*.8)*21400/12.+
$    (ESHIN(J)+S59-S61)*1.9835+ADJ6
DSTG6=DSTOR6/AREA6
ESTG6(J)=ESTG6(J-1)+DSTG6
C *** LAKE CYPRESS ***
CALL STORAGE(CYPR,STG7E,VOL7,AREA7)
CALL REGSTG(CYPR,JDATE,RSTG7)
C --- COMPUTE FLOW AT C36 AND C37 ---
HEAD78=STG7E-STG8E
HEAD89=STG8E-STG9E
Q36=35.61885873*(STG7E-35.)*1.6667*(ABS(HEAD78)+1E-20)**.5511796
Q37=87.07430164*(STG8E-42.)*1.6667*(ABS(HEAD89)+1E-20)**.4976433
DVOL7=1.13*HEAD78*AREA7+(S61+S63+.3*EREED(J))*1.9835
DVOL7=DVOL7*7480./(7480.+3850.)
IF(DVOL7.LE.0.)THEN
    C36=MAX(-Q36,DVOL7/1.9835)
ELSE
    C36=MIN(Q36,DVOL7/1.9835)
ENDIF
ADJ7=RES7(J)
DSTOR7=(RAIN7(J)-EVAP(J)*.8)*4274/12.+
$    (S61+S63+EREED(J)*.3-C36)*1.9835+ADJ7
DSTG7=DSTOR7/AREA7
ESTG7(J)=ESTG7(J-1)+DSTG7
C *** LAKE HATCHINIHA ***
CALL STORAGE(HATC,STG8E,VOL8,AREA8)
CALL REGSTG(HATC,JDATE,RSTG8)
DVOL8=1.10*HEAD89*AREA8+(ECATF(J)+.7*EREED(J)+C36)*1.9835
DVOL8=DVOL8*34141./(34141.+7480.)
IF(DVOL8.LE.0.)THEN
    C37=MAX(-Q37,DVOL8/1.9835)
ELSE
    C37=MIN(Q37,DVOL8/1.9835)
ENDIF
ADJ8=RES8(J)
DSTOR8=(RAIN8(J)-EVAP(J)*.8)*9733/12.+
$    (ECATF(J)+EREED(J)*.7+C36-C37)*1.9835+ADJ8
DSTG8=DSTOR8/AREA8
ESTG8(J)=ESTG8(J-1)+DSTG8
C *** LAKE KISSIMMEE ***
CALL STORAGE(KISS,STG9E,VOL9,AREA9)

```


Appendix 2: Program Listing (KROUTE)

```

CALL REGSTG(KISS,JDATE,RSTG9)
CALL DISCH(STR65,STG9E,46.30,Q65)
IF(AMC(1).GT.8.)Q65=MIN(3000.,Q65)
IF(AMC(1).LE.8.)THEN
  IF(JY.LT.86)Q65=MIN(5000.,Q65)
  IF(JY.GE.86)Q65=MIN(6000.,Q65)
ENDIF
DVOL9=(STG9E-RSTG9)*AREA9+C37*1.9835
IF(DVOL9.LE.0.)THEN
  ES65=0.
ELSE
  ES65=MIN(Q65,DVOL9/1.9835)
ENDIF
ADJ9=RES9(J)
DSTOR9=(RAIN9(J)-EVAP(J)*.8)*42607/12.+
$ (C37-ES65)*1.9835+ADJ9
DSTG9=DSTOR9/AREA9
ESTG9(J)=ESTG9(J-1)+DSTG9
270 CONTINUE
C *** SUMMARY PRINTOUT FOR ALL LAKES ***
275 IF(IOPT.EQ.1)THEN
  WRITE(7,280)IDATE(J),STG1(J),ESTG1(J),STG2(J),ESTG2(J),
$ STG3(J),ESTG3(J),STG4(J),ESTG4(J),STG5(J),ESTG5(J),
$ STG6(J),ESTG6(J),STG7(J),ESTG7(J),STG8(J),ESTG8(J),
$ STG9(J),ESTG9(J),S65(J),ES65
  WRITE(8,280)IDATE(J),RSTG1,ESTG1(J),RSTG2,ESTG2(J),
$ RSTG3,ESTG3(J),RSTG4,ESTG4(J),RSTG5,ESTG5(J),
$ RSTG6,ESTG6(J),RSTG7,ESTG7(J),RSTG8,ESTG8(J),
$ RSTG9,ESTG9(J),S65(J),ES65
280 FORMAT(' ',A8,18(F6.2),2F6.0)
  WRITE(9,290)IDATE(J),S58,S57,S62,S60,S63,
$ S59,S61,ES65,ESHIN(J),EBOGG(J),ECATF(J),EREED(J),C36,C37
290 FORMAT(A8,8X,14F6.0)
  ENDIF
C *** DETAILED OUTPUT OPTION OF EACH LAKE IF IPRINT=1 ***
  IF(IPRINT.EQ.1)THEN
    WRITE(11,300)IDATE(J),RAIN1(J),EVAP(J),RSTG1,STG1(J),ESTG1(J),
$ AREA1,VOL1,DSTOR1,S60,S58,ADJ1
300 FORMAT(' ',A8,1X,5F6.2,3F9.0,10F7.0)
    WRITE(12,300)IDATE(J),RAIN2(J),EVAP(J),RSTG2,STG2(J),ESTG2(J),
$ AREA2,VOL2,DSTOR2,S57,S58,ADJ2
    WRITE(13,300)IDATE(J),RAIN3(J),EVAP(J),RSTG3,STG3(J),ESTG3(J),
$ AREA3,VOL3,DSTOR3,S57,S62,ADJ3
    WRITE(14,300)IDATE(J),RAIN4(J),EVAP(J),RSTG4,STG4(J),ESTG4(J),
$ AREA4,VOL4,DSTOR4,S60,S63,ADJ4
    WRITE(15,300)IDATE(J),RAIN5(J),EVAP(J),RSTG5,STG5(J),ESTG5(J),
$ AREA5,VOL5,DSTOR5,S62,S59,EBOGG(J),ADJ5
    WRITE(16,300)IDATE(J),RAIN6(J),EVAP(J),RSTG6,STG6(J),ESTG6(J),
$ AREA6,VOL6,DSTOR6,S59,S61,ESHIN(J),ADJ6
    WRITE(17,300)IDATE(J),RAIN7(J),EVAP(J),RSTG7,STG7(J),ESTG7(J),
$ AREA7,VOL7,DSTOR7,C36,S61,S63,EREED(J),ADJ7
    WRITE(18,300)IDATE(J),RAIN8(J),EVAP(J),RSTG8,STG8(J),ESTG8(J),
$ AREA8,VOL8,DSTOR8,C36,C37,ECATF(J),EREED(J),ADJ8

```

Appendix 2: Program Listing (KROUTE)

```

        WRITE(19,300)IDATE(J),RAIN9(J),EVAP(J),RSTG9,STG9(J),ESTG9(J),
$   AREA9,VOL9,DSTOR9,C37,S65(J),ES65,ADJ9
        ENDIF
500   CONTINUE
        IF(IFOR.EQ.1 .AND. ICASE.LT.NCASE)GOTO 10
        IF(IOPT.EQ.2)CALL OPT(IO,IPAR,*10,*50)
        IF(IFOR.NE.1 .AND. IO.EQ.0)GOTO 50
        STOP
        END

```

SUBROUTINE BUILD

```

C *****
C *** THIS SUBROUTINE BUILDS INPUT FILE "NTAPE5" FOR
C *** FORECASTING OPTION USE
C *****
        COMMON /D/TEVAP(12),TRAIN(12),AMC(6),MONTH(12)
        REAL RAT(6)
        DATA NMONTH,NCASE/6,5/
        PRINT*,'ENTER FIRST DATE (MM,DD,YY)'
        READ*.IM,ID,IY
        PRINT*,'ENTER ',NCASE,' MULTIPLICATION RATIOS FOR RAINFALL'
        READ*,(RAT(N),N=1,NCASE)
        PRINT*,'ENTER ANTECEDENT 1 AND 6-MONTH RAINFALL'
        READ*.AMC(1),AMC(6)
        PRINT*,'ENTER INITIAL STAGES FROM LAKE ALLIGATOR TO KISSIMMEE'
        READ*.ESTG1,ESTG2,ESTG3,ESTG4,ESTG5,ESTG6,ESTG7,ESTG8,ESTG9
        WRITE(5,10)AMC(1),AMC(6),NMONTH,NCASE,(RAT(N),N=1,NCASE)
10    FORMAT(2F6.2,2I2,6F6.2)
        WRITE(5,'(9F6.2)')ESTG1,ESTG2,ESTG3,ESTG4,ESTG5,ESTG6,ESTG7,
$      ESTG8,ESTG9
        DO 100 KM=1,NMONTH
            JM=IM+KM-1
            JY=IY
            IF(JM.GT.12)THEN
                JM=JM-12
                JY=IY+1
            ENDIF
            IF(KM.EQ.1)THEN
                JD1=ID
            ELSE
                JD1=1
            ENDIF
            JD2=MONTH(JM)
            DO 100 JD=JD1,JD2
                RAIN=TRAIN(JM)/MONTH(JM)
                WRITE(5,50)JM,JD,JY,RAIN
50    FORMAT(1X,I2,'-',I2,'-',I2,F6.3)
100   CONTINUE
        RETURN
        END

```

Appendix 2: Program Listing (KROUTE)

```

SUBROUTINE INFLOW(I,JM)
C *****
C *** INFLOW PREDICTION BY SCS METHOD
C *** I=DAY INDEX
C *** JM=MONTH
C *** ID=WATERSHED ID
C *** SMD1 TO SMD9=SOIL MOISTURE DEFICIT IN INCHES
C *** AMC(1)=ANTECEDENT 1-MONTH RAINFALL
C *** PCOEF=PAN COEFICIENT FOR POTENTIAL ET (LAND USE FUNCTION)
C *** ROOT=ROOT ZONE DEPTH IN FT BELOW WHICH ET CEASES (LAND USE FUNCTION)
C *** SCOE=SEEPAGE COEFFICIENT IN INCHES (AQUIFER-STREAM FUNCTION)
C *** HMAX=MAXIMUM DEPTH IN FT BELOW WHICH SEEPAGE CEASES (STREAM FUNCTION)
C *** QRAIN=DIRECT RUNOFF FUNCTION IN INCHES FROM SCS METHOD
C *** SEEP=DIRECT SEEPAGE AND BASE FLOW FUNCTION IN INCHES
C *** ET=SOIL EVAPOTRANSPIRATION FUNCTION IN INCHES
C *** SOIL=SOIL MOISTURE ACCOUNTING FUNCTION
C *****
      COMMON /D/TEVAP(12),TRAIN(12),AMC(6),MONTH(12)
      COMMON /E/ EVAP(365),RAIN1(365),RAIN2(365),
      $ RAIN3(365),RAIN4(365),RAIN5(365),RAIN6(365),
      $ RAIN7(365),RAIN8(365),RAIN9(365),
      $ BOGG(365),CATF(365),REED(365),SHIN(365),
      $ EBOGG(365),ECATF(365),EREED(365),ESHIN(365),
      $ RES1(365),RES2(365),RES3(365),RES4(365),RES5(365),
      $ RES6(365),RES7(365),RES8(365),RES9(365)
      COMMON /G/ SCOE(13),HMAX(13),PCOEF(13),ROOT(13)
      SAVE
      DATA SMD1/-99/
C   DATA SCOE/.069,.041,.021,.018,.089,.076,.121,.300,.045,
C   $ .010,.022,.0004,.029/
      DATA SCOE/.069,.037,.016,.010,.089,.072,.112,.293,.040,
      $ .010,.022,.0003,.017/
      DATA HMAX/6.12,5.69,9.62,6.12,6.00,3.95,5.21,4.50,14.2,
      $ 3.63,4.70,2.54,13.22/
      DATA PCOEF/.70,.70,.70,.70,.70,.70,.70,.70,.70,
      $ .70,.70,.70,.70/
      DATA ROOT/5.00,5.00,5.00,5.00,5.00,5.00,5.00,5.00,5.00,
      $ 5.00,5.00,5.00,5.00/
C *** LAKE EVAPORATION FUNCTION BY FREQUENCY CORRELATION
      Y1(YM,YS,XM,XS,CORR,X)=YM+CORR*YS/XS*(X-XM)
C *** SCS DIRECT RUNOFF FUNCTION
      QRAIN(RAIN,SMD)= MAX(0.,RAIN-0.2*SMD)**2/
      $ (RAIN+0.8*SMD+1E-20)
C *** SEEPAGE AND BASE FLOW FUNCTION
      SEEP(SMD.ID)=SCOE(ID)*MAX(0., (HMAX(ID)-SMD/2.4)/HMAX(ID))
C *** SOIL ET FUNCTION
      ET(SMD.ID)=PCOEF(ID)*EVAP(I)*MAX(0.,(ROOT(ID)-SMD/2.4)/ROOT(ID))
C *** SOIL MOISTURE DEFICIT ACCOUNTING FUNCTION
      SOIL(SMD.RAIN.ID)=SMD-RAIN+QRAIN(RAIN,SMD)+SEEP(SMD.ID)+ET(SMD.ID)
C *** INITIALIZE ANTECEDENT MOISTURE DEFICIT
      IF(SMD1.LE.-99)THEN
        SMD=5.
        DO 10 K=0,5
          M=JM-K

```

Appendix 2: Program Listing (KROUTE)

```

      IF(M.LE.0)M=12+M
10    SMD=SMD+0.7*TEVAP(M)-AMC(K+1)
      SMD1=SMD2=SMD3=SMD4=SMD5=SMD6=SMD7=SMD8=SMD9=SMD
      SMD10=SMD11=SMD12=SMD13=SMD
      ENDIF
C *** CALCULATE LAKE EVAPORATION AND SOIL MOISTURE DEFICIT
      EVAP(I)=Y1(TEVAP(JM),-1.7094,TRAIN(JM),2.4755,1.,AMC(1))/MONTH(JM)
      SMD1=MAX(0.,SOIL(SMD1,RAIN1(I),1))
      SMD2=MAX(0.,SOIL(SMD2,RAIN2(I),2))
      SMD3=MAX(0.,SOIL(SMD3,RAIN3(I),3))
      SMD4=MAX(0.,SOIL(SMD4,RAIN4(I),4))
      SMD5=MAX(0.,SOIL(SMD5,RAIN5(I),5))
      SMD6=MAX(0.,SOIL(SMD6,RAIN6(I),6))
      SMD7=MAX(0.,SOIL(SMD7,RAIN7(I),7))
      SMD8=MAX(0.,SOIL(SMD8,RAIN8(I),8))
      SMD9=MAX(0.,SOIL(SMD9,RAIN9(I),9))
      SMD10=MAX(0.,SOIL(SMD10,RAIN5(I),10))
      SMD11=MAX(0.,SOIL(SMD11,RAIN6(I),11))
      SMD12=MAX(0.,SOIL(SMD12,RAIN6(I),12))
      SMD13=MAX(0.,SOIL(SMD13,RAIN9(I),13))
C *** FORECAST LAKE INFLOWS
      RES1(I)=23910*(SEEP(SMD1,1)+QRAIN(RAIN1(I),SMD1))/12
      RES2(I)=11505*(SEEP(SMD2,2)+QRAIN(RAIN2(I),SMD2))/12
      RES3(I)=33172*(SEEP(SMD3,3)+QRAIN(RAIN3(I),SMD3))/12
      RES4(I)=0.5*51131*(SEEP(SMD4,4)+QRAIN(RAIN4(I),SMD4))/12
      RES5(I)=21756*(SEEP(SMD5,5)+QRAIN(RAIN5(I),SMD5))/12
      RES6(I)=66249*(SEEP(SMD6,6)+QRAIN(RAIN6(I),SMD6))/12
      RES7(I)=29409*(SEEP(SMD7,7)+QRAIN(RAIN7(I),SMD7))/12
      RES8(I)=31179*(SEEP(SMD8,8)+QRAIN(RAIN8(I),SMD8))/12
      RES9(I)=95039*(SEEP(SMD9,9)+QRAIN(RAIN9(I),SMD9))/12
      EBOGG(I)=53504*(SEEP(SMD10,10)+QRAIN(RAIN5(I),SMD10))/24.
      ESHIN(I)=115200*(SEEP(SMD11,11)+QRAIN(RAIN6(I),SMD11))/24.
      EREED(I)=108800*(SEEP(SMD12,12)+QRAIN(RAIN6(I),SMD12))/24.
      ECATF(I)=0.75*37696*(SEEP(SMD13,13)+QRAIN(RAIN9(I),SMD13))/24.
C --- 30-DAY MOVING RAINFALL
C    ACC5=ACC6=ACC9=0.
C    DO 100 J=0,29
C      IDAY=I-J
C      IF(IDAY.LE.0)IDAY=365-IDAY
C      ACC5=ACC5+RAIN5(IDAY)
C      ACC6=ACC6+RAIN6(IDAY)
C      ACC9=ACC9+RAIN9(IDAY)
C100 CONTINUE
C    EBOGG(I)=15.005*ACC5 -12.721
C    ESHIN(I)=30.123*ACC6 +5.959
C    EREED(I)=17.154*ACC6 -18.771
C    ECATF(I)= 2.949*ACC9 +24.217
      RETURN
      END

      SUBROUTINE OPT(IO,IPAR,*,*)
C *****
C *** OPTIMIZATION PROCEDURE FOR INFLOW PARAMETERS

```

Appendix 2: Program Listing (KROUTE)

```

C *** IPAR=PARAMETERS TO BE OPTIMIZED: 1=SCOEF, 2=HMAX, 3=PCOEF, 4=ROOT
C *** ID=WATERSHED ID
C *** COEF(ID,IPAR)=EQUIVALENT TO SCOEF(),HMAX(),PCOEF(),ROOT()
C *****
COMMON /E/ EVAP(365),RAIN1(365),RAIN2(365),
$ RAIN3(365),RAIN4(365),RAIN5(365),RAIN6(365),
$ RAIN7(365),RAIN8(365),RAIN9(365),
$ BOGG(365),CATF(365),REED(365),SHIN(365),
$ EBOGG(365),ECATF(365),EREED(365),ESHIN(365),
$ RES1(365),RES2(365),RES3(365),RES4(365),RES5(365),
$ RES6(365),RES7(365),RES8(365),RES9(365)
COMMON /F/ STG1(0:365),STG2(0:365),STG3(0:365),
$ STG4(0:365),STG5(0:365),STG6(0:365),STG7(0:365),
$ STG8(0:365),STG9(0:365),S65(365),
$ ESTG1(0:365),ESTG2(0:365),ESTG3(0:365),ESTG4(0:365),
$ ESTG5(0:365),ESTG6(0:365),ESTG7(0:365),ESTG8(0:365),
$ ESTG9(0:365)
COMMON /G/ COEF(13,4)
SAVE
DATA ID,ITER,FUNC/13,0,0./
IF(ID.LT.0)GOTO 40
C *** CALCULATE OBJECTIVE FUNCTION
DO 30 I=1,365
IF(ID.EQ.1)FUNC=FUNC+ ABS(STG1(I)-ESTG1(I))**2
IF(ID.EQ.2)FUNC=FUNC+ ABS(STG2(I)-ESTG2(I))**2
IF(ID.EQ.3)FUNC=FUNC+ ABS(STG3(I)-ESTG3(I))**2
IF(ID.EQ.4)FUNC=FUNC+ ABS(STG4(I)-ESTG4(I))**2
IF(ID.EQ.5)FUNC=FUNC+ ABS(STG5(I)-ESTG5(I))**2
IF(ID.EQ.6)FUNC=FUNC+ ABS(STG6(I)-ESTG6(I))**2
IF(ID.EQ.7)FUNC=FUNC+ ABS(STG7(I)-ESTG7(I))**2
IF(ID.EQ.8)FUNC=FUNC+ ABS(STG8(I)-ESTG8(I))**2
IF(ID.EQ.9)FUNC=FUNC+ ABS(STG9(I)-ESTG9(I))**2
IF(ID.EQ.10)FUNC=FUNC+ ABS(LOG(.01+BOGG(I))-LOG(.01+EBOGG(I)))**2
IF(ID.EQ.11)FUNC=FUNC+ ABS(LOG(.01+SHIN(I))-LOG(.01+ESHIN(I)))**2
IF(ID.EQ.12)FUNC=FUNC+ ABS(LOG(.01+REED(I))-LOG(.01+EREED(I)))**2
IF(ID.EQ.13)FUNC=FUNC+ ABS(LOG(.01+CATF(I))-LOG(.01+ECATF(I)))**2
30 CONTINUE
RETURN 2
C *** ADJUST PARAMETERS BY GRADIENT SEARCH METHOD
40 ITER=ITER+1
WRITE(20,60)ID,ITER,IPAR,COEF(ID,IPAR),FUNC
WRITE(*,60)ID,ITER,IPAR,COEF(ID,IPAR),FUNC
60 FORMAT('ID=' ,I2,' ITER=' ,I2,' PARAM ' ,I1,'=' ,F6.3,' FUNC=' ,G9.4)
IF(ITER.EQ.1)THEN
COEF1=COEF(ID,IPAR)
FUNC1=FUNC
ALPH=0.75
DCOEF=-0.02*COEF1
COEF(ID,IPAR)=COEF1-DCOEF
FUNC=0.
RETURN 1
ENDIF
IF(ITER.GT.15 .OR. ABS(DCOEF/COEF1).LE.0.01)THEN
ID=ID+1

```

Appendix 2: Program Listing (KROUTE)

```

IF(ID.GT.13)STOP
ITER=0
FUNC=0.
RETURN 1
ENDIF
IF(FUNC.LT.FUNC1 .OR. ITER.EQ.2)THEN
DFUNC=(FUNC-FUNC1)/(COEF(ID,IPAR)-COEF1)
IF(DFUNC.EQ.0.)DFUNC=1E-20
DCOEF=ALPH*FUNC/DFUNC
COEF1=COEF(ID,IPAR)
FUNC1=FUNC
COEF(ID,IPAR)=MAX(1E-20, COEF1-DCOEF)
FUNC=0.
RETURN 1
ELSE
DCOEF=DCOEF/50.
ALPH=ALPH/50.
COEF(ID,IPAR)=MAX(1E-20, COEF1-DCOEF)
FUNC=0.
RETURN 1
ENDIF
END

```

SUBROUTINE STORAGE(LAKE,STAGE,STOR,AREA)

```

C *****
C *** STORAGE AND AREA RATING FUNCTIONS
C *** LAKE(1) TO (5) = PARAMETERS C1 TO C5 FOR STORAGE CALCULATION
C *** LAKE(6) TO (10) = PARAMETERS D1 TO D5 FOR AREA CALCULATION
C *** LAKE(11)=MAXIMUM STAGE LIMIT FOR RATING CURVE
C *** LAKE(12)=MINIMUM STAGE LIMIT FOR RATING CURVE
C *** LAKE(13)=LAKE ID NUMBER
C *****
REAL LAKE(13)
V(S)=C1*S**4+C2*S**3+C3*S**2+C4*S+C5
A(S)=D1*S**4+D2*S**3+D3*S**2+D4*S+D5
DA(S)=4.*D1*S**3+3.*D2*S**2+2*D3*S+D4
C1=LAKE(1)
C2=LAKE(2)
C3=LAKE(3)
C4=LAKE(4)
C5=LAKE(5)
D1=LAKE(6)
D2=LAKE(7)
D3=LAKE(8)
D4=LAKE(9)
D5=LAKE(10)
STGMAX=LAKE(11)
STGMIN=LAKE(12)
STG=STAGE
IF (STG.GT.STGMAX)THEN
AREA=A(STGMAX)+DA(STGMAX)*(STG-STGMAX)
STOR=V(STGMAX)+(AREA+A(STGMAX))/2.*(STG-STGMAX)
RETURN

```

Appendix 2: Program Listing (KROUTE)

```

ENDIF
IF (STG.LT.STGMIN)THEN
  AREA=A(STGMIN)*(STG/STGMIN)**1.5
  STOR=V(STGMIN)*(STG/STGMIN)**1.5
  RETURN
ENDIF
STOR=V(STG)
AREA=A(STG)
RETURN
END

SUBROUTINE DISCH(STRUCT,HW,TW,Q)
C *****
C *** STRUCT(1)=TYPE: 1=GATED SPILLWAY; 2=PIPE CULVERT
C ***      (2)=CULVERT OR SPILLWAY CREST LENGTH
C ***      (3)=CULVERT INVERT OR SPILLWAY CREST ELEVATION
C ***      (4)=NUMBER OF GATES OR CULVERTS
C ***      (5)=CULVERT DIAMETER
C ***      (6)=COMPUTED GATE OPENNING OR ALLOWABLE DISCHARGE FOR CULVERT
C ***      (7)=MANNING N FOR CULVERT STRUCTURE
C ***      (7)-(10)=PARAMETERS A1 TO A4 FOR GATE OPENNING EQUATION
C ***      (11)-(15)=PARAMETERS B1 TO B5 FOR SPILLWAY FLOW EQUATIONS
C *** HW=HEADWATER ELEVATION
C *** TW=TAILWATER ELEVATION
C *** Q=COMPUTED DISCHARGE IN CFS
C *****
  DIMENSION STRUCT(15)
  ITYPE=STRUCT(1)
  AL=STRUCT(2)
  CEL=STRUCT(3)
  NPIPE=STRUCT(4)
  DIAM=STRUCT(5)
  QMAX=STRUCT(6)
  FRICT=STRUCT(7)
  A1=STRUCT(7)
  A2=STRUCT(8)
  A3=STRUCT(9)
  A4=STRUCT(10)
  B1=STRUCT(11)
  B2=STRUCT(12)
  B3=STRUCT(13)
  B4=STRUCT(14)
  B5=STRUCT(15)
  HEAD=HW-CEL
  IF(TW.GE.HW.OR.HW.LE.CEL)THEN
    Q=0.
    GOTO 1000
  ENDIF
  Q1=Q2=Q3=Q4=GO=99999.
  GOTO (100,500)ITYPE
C *** GATED SPILLWAY STRUCTURE ***
100 GO=A1*(HW-TW)**A2*TW**A3+A4
  STRUCT(6)=GO

```

Appendix 2: Program Listing (KROUTE)

```

IF (GO.LE.0.1)GO=0.1
IF(TW.LE.CEL)THEN
C --- FREE WEIR FLOW
  Q1=B1*AL*(HW-CEL)**1.5
C --- FREE GATED FLOW
  IF(HW .GT. 1.1*(CEL+GO))
$ Q2=B2*AL*GO*SQRT(64.4*(HW-CEL-0.5*GO))
  Q=MIN(Q1,Q2)
ELSE
C --- SUBMERGED WEIR FLOW
  Q3=B3*AL*(TW-CEL)*SQRT(64.4*(HW-TW))
C --- SUBMERGED GATED FLOW
  Q4=(B4*GO+B5)*AL*GO*SQRT(64.4*(HW-TW))
  Q=MIN(Q3,Q4)
ENDIF
GOTO 1000
C *** PIPE CULVERT STRUCTURE ***
500 TOP=CEL+DIAM
  IF (HW.LT.TOP)THEN
C --- OPEN CHANNEL FLOW ---
  Y1=HW-CEL
  Y2=TW-CEL
  IF (Y2.LE.(.1*Y1))Y2=.1*Y1
  Y=(Y1+Y2)/2
C STRAIGHT LINE APPROXIMATION FOR HYDRAULIC CONVEYANCE(AR)
  AR1=3.142*DIAM**2/4.*(DIAM/4.)**.6667
  XAR=1.72*Y/DIAM-.373
  IF (XAR.LE..1)XAR=.05
  IF (XAR.GE.1.1)XAR=1.05
  AR=XAR*AR1
  SLOPE=(Y1-Y2)/AL
  Q1=NPIPE*1.49/FRICT*AR*SQRT(SLOPE)
  Q=MIN(Q1,QMAX)
  GOTO 1000
ENDIF
C --- ORIFICE FLOW ---
  IF (HW.GE.TOP.AND.TW.LT.TOP)THEN
  TW1=MAX(CEL+DIAM/2., TW)
  Q2=NPIPE*.75*3.142*DIAM**2/4.*SQRT(2*32.2*(HW-TW1))
  Q=MIN(Q2,QMAX)
  GOTO 1000
ENDIF
C --- FULL PIPE FLOW ---
  IF (HW.GE.TOP.AND.TW.GE.TOP)THEN
  AR=3.142*DIAM**2/4.*(DIAM/4.)**.6667
  SLOPE=(HW-TW)/AL
  Q3=NPIPE*1.49/FRICT*AR*SQRT(SLOPE)
  Q=MIN(Q3,QMAX)
  GOTO 1000
ENDIF
1000 RETURN
END

```


Appendix 2: Program Listing (KROUTE)

```

SUBROUTINE REGSTG(LAKE,JDATE,RSTG)
C *****
C *** LAKE REGULATION STAGE
C *** LAKE(13)=LAKE ID
C *** REG=SCHEDULED REGULATED STAGE ON IJDAY
C *** RSTG=COMPUTED REGULATED STAGE ON IDAY
C *****
COMMON /C/REG(9,15),IJDAY(9,15)
REAL LAKE(13)
INTEGER MONTH(12)
CHARACTER JDATE*8
DATA MONTH/0,31,59,90,120,151,181,212,243,273,304,334/
READ(JDATE,'(I2,1X,I2,1X,I2)')JM,JD,JY
IDAY=MONTH(JM)+JD
ID=LAKE(13)
DO 300 J=1,10
ICHECK=IDAY-IJDAY(ID,J)
IF (ICHECK.LE.0)GOTO 500
300 CONTINUE
500 IF (ICHECK.EQ.0)THEN
RSTG=REG(ID,J)
RETURN
ELSE IF (ICHECK.LT.0)THEN
REG2=REG(ID,J)
REG1=REG(ID,J-1)
IJDAY2=IJDAY(ID,J)
IJDAY1=IJDAY(ID,J-1)
RSTG=REG1+(REG2-REG1)/(IJDAY2-IJDAY1)*(IDAY-IJDAY1)
RETURN
ENDIF
END

BLOCK DATA
COMMON /A/STR57(15),STR58(15),STR59(15),STR60(15),STR61(15),
$ STR62(15),STR63(15),STR65(15)
COMMON /B/ALLI(13),MYRT(13),HART(13),GENT(13),
$ ETOH(13),TOHO(13),CYPR(13),HATC(13),KISS(13)
COMMON /D/TEVAP(12),TRAIN(12),AMC(6),MONTH(12)
REAL MYRT,KISS
DATA
$STR57/2. 80., 52.5, 2, 4.5, 230., .024, 8*0./,
$STR58/2. 70., 54.5, 2, 4.5, 110., .024, 8*0./,
$STR59/1. 18., 49.1, 0, 0, 590., 3.8E-9, -.22083, 5.41927,
$ -3.673, 3.28, .75, .9, .1033, .58/,
$STR60/1. 12., 55., 0, 0, 330., 2.2588896E-14, -.51706295,
$ 8.2136768, 0., 3.28, .75, .9, 0., .73/,
$STR61/1. 27., 36.9, 0, 0, 1570., .002954, -.15463, 2.24833,
$ -10.82, 3.28, .75, .9, .0253, .59/,
$STR62/1. 14., 55.3, 0, 0, 410., 1.7404157E-22, -.59052612,
$ 12.749331, 0., 3.28, .75, .9, 0., .75/,
$STR63/1. 15., 54., 0, 0, 715., .022745, -.21273, 1.6471,
$ -8.246, 3.28, .75, .9, 0., .75/,
$STR65/1. 81., 39.3, 0, 0, 3000., .36588, -.55683, .95237,

```

Appendix 2: Program Listing (KROUTE)

```

$ 0., 3.28, .75, .9, .0375, .76/
DATA ALLI
$/-4.4642004 , 1238.5192 , -126982.37 , 5724291.8 ,
$ -.95950501E+08, -22.386100, , 5641.0255 , -532464.04 ,
$ .22314593E+08, -.35033754E+09, 65., 59.5, 1./
DATA MYRT
$/ 3.0002689 , -707.76736 , 62675.634 , -2468063.4 ,
$ .36452945E+08, 2.0264439 , -481.92500 , 42983.139 ,
$ -1703871.9 , .25327409E+08, 65., 58., 2./
DATA HART
$/ .21157913 , -41.567386 , 3097.0676 , -102266.56 ,
$ 1244906.4 , .29565288 , -67.473332 , 5783.8343 ,
$ -220443.89 , 3150674.3 , 64., 56., 3./
DATA GENT
$/ 2.9258007 , -702.73697 , 63312.710 , -2534177.8 ,
$ .38012042E+08, 1.5462534 , -366.74399 , 32608.560 ,
$ -1288094.7 , .19073801E+08, 65., 57., 4./
DATA ETOH
$/2.829373, -649.2139, 56075.09, -2148076.,
$ 30728530., -.263521, 64.99214, -5948.022, 240362.1,
$ -3617802., 65., 50., 5./
DATA TOHO
$/0, 24.71876, -3310.8, 161410.9, -2828022.,
$ -1.661901, 367.9039, -30441.02, 1117249., -15345700.,
$ 60., 49., 6./
DATA CYPR
$/ .43142302 , -91.027331 , 7294.9886 , -258747.85 ,
$ 3400561.1, 0., 1.7256921 , -273.08199 , 14589.977 ,
$ -258747.85 , 58., 43., 7./
DATA HATC
$/-1.5127289 , 284.80337 , -19506.905 , 578132.96 ,
$ -6250583.0 , 0., -6.0509156 , 854.41011 , -39013.810 ,
$ 578132.96 , 55., 45., 8./
DATA KISS
$/ 14.023537 , -2683.7603 , 193335.42 , -6183331.0 ,
$ .73906134E+08, 0., 56.094148 , -8051.2809 , 386670.84 ,
$ -6183331.0 , 58., 42.5, 9./
DATA TEVAP/3.29,3.92,6.11,7.23,8.06,7.35,
$ 7.47,7.04,6.21,5.36,3.90,3.07/
DATA TRAIN/2.18,2.69,3.19,2.68,4.16,7.38,
$ 7.66,6.97,6.56,3.25,1.65,2.02/
DATA MONTH/31.28,31.30,31.30,31.31,31.30,31.30,31/
END

```

Appendix 3: Program Listing (KPLOT)

```

PROGRAM KPLOT(DATA,INPUT,OUTPUT,TAPE5=DATA,
2TAPE13=OUTPUT,TAPE12=INPUT)
C *****
C *** PLOTTING OF "KROUTE" OUTPUT DATA
C *** XARRAY AND YARRAY MUST BE DIMENSIONED TO AT LEAST 2 PLUS
C *** ACTUAL ARRAY SIZES.
C *** CALL PLOTS(0,0,13) TO INITIATE PLOTTING AND WRITE PLOT DATA TO TAPE13
C *** CALL PLOT(XLEN1+5., 0., 999) TO TERMINATE PLOTTING
C ***          A. FAN 1982
C *****
C ***** KEEP THE STATEMENTS BELOW *****
      CHARACTER*80 TITLE,XLAB,YLAB,LEG,FRMT,TEXT(100),TYPE*2,PTYPE*2
      COMMON /A/ XPLEG,YPLEG,XF,XD,YF,YD,XLEN1,YLEN1,NTH,
2          NPLOT,IBYEAR,IBMONTH
      COMMON /B/ PTYPE
      DATA IBMONTH/1/,IBYEAR/1970/,NTH/0/,NPLOT/1/
C *****
      PRINT*,'PLOTTING OPTION: 1=STAGE,2=FLOW,3=BOTH,4=FORECAST'
      READ*.IOPT
      IF(IOPT.LE.3)THEN
          PRINT*,'ENTER ASSUMPTION USED IN HOLLERINTH'
          READ*,LEG
      ENDIF
      CALL PLOTS(0,0,13)
      IF(IOPT.EQ.1 .OR. IOPT.EQ.3)THEN
          CALL FACTOR(0.4)
          CALL PSTAGE('ALLIGATOR','(1X,A8,T10,2F6.2,4(/))',58.,66.,LEG)
          CALL PSTAGE('MYRTLE','(1X,A8,T22,2F6.2,4(/))',58.,66.,LEG)
          CALL PSTAGE('HART','(1X,A8,T34,2F6.2,4(/))',56.,64.,LEG)
          CALL PSTAGE('GENTRY','(1X,A8,T46,2F6.2,4(/))',56.,64.,LEG)
          CALL PSTAGE('EAST TOHO','(1X,A8,T58,2F6.2,4(/))',52.,60.,LEG)
          CALL PSTAGE('TOHO','(1X,A8,T70,2F6.2,4(/))',48.,56.,LEG)
          CALL PSTAGE('CYPRESS','(1X,A8,T82,2F6.2,4(/))',46.,54.,LEG)
          CALL PSTAGE('HATCHINEHA','(1X,A8,T94,2F6.2,4(/))',46.,54.,LEG)
          CALL PSTAGE('KISSIMMEE','(1X,A8,T106,2F6.2,4(/))',42.,54.,LEG)
      ENDIF
      IF(IOPT.EQ.2 .OR. IOPT.EQ.3)THEN
          CALL PF('ATTACH','TAPE2','TAPE2','UN','AFAN')
          CALL FACTOR(0.4)
          CALL PFLOW('S-58','(A8,T17,F6.0)',0.,60.,LEG)
          CALL PFLOW('S-57','(A8,T23,F6.0)',0.,180.,LEG)
          CALL PFLOW('S-62','(A8,T29,F6.0)',0.,270.,LEG)
          CALL PFLOW('S-60','(A8,T35,F6.0)',0.,210.,LEG)
          CALL PFLOW('S-63','(A8,T41,F6.0)',0.,420.,LEG)
          CALL PFLOW('S-59','(A8,T47,F6.0)',0.,900.,LEG)
          CALL PFLOW('S-61','(A8,T53,F6.0)',0.,1500.,LEG)
          CALL PFLOW('S-65','(A8,T59,F6.0)',0.,3300.,LEG)
          CALL PFLOW('SHINGLE CREEK','(A8,T65,F6.0)',0.,420.,LEG)
          CALL PFLOW('BOGGY CREEK','(A8,T71,F6.0)',0.,420.,LEG)
          CALL PFLOW('CATFISH CREEK','(A8,T77,F6.0)',0.,180.,LEG)
          CALL PFLOW('REEDY CREEK','(A8,T83,F6.0)',0.,450.,LEG)
      ENDIF
      IF(IOPT.EQ.4)THEN
          CALL FACTOR(0.6)

```

Appendix 3: Program Listing (KPLOT)

```

CALL PLOT(0.,3.,-3)
OPEN(5,FILE='NTAPE5',STATUS='OLD')
CALL FSTAGE('ALLIGATOR','(1X,A8,T10,2F6.2)',60.,66.)
CALL FSTAGE('MYRTLE','(1X,A8,T22,2F6.2)',58.,64.)
CALL FSTAGE('HART','(1X,A8,T34,2F6.2)',58.,64.)
CALL FSTAGE('GENTRY','(1X,A8,T46,2F6.2)',58.,64.)
CALL FSTAGE('EAST TOHO','(1X,A8,T58,2F6.2)',54.,60.)
CALL FSTAGE('TOHO','(1X,A8,T70,2F6.2)',51.,57.)
CALL FSTAGE('CYPRESS','(1X,A8,T82,2F6.2)',48.,54.)
CALL FSTAGE('HATCHINEHA','(1X,A8,T94,2F6.2)',48.,54.)
CALL FSTAGE('KISSIMMEE','(1X,A8,T106,2F6.2)',48.,54.)
CALL FSTAGE('S-65 FLOW','(1X,A8,T118,2F6.0)',0.,6000.)
ENDIF
CALL PLOT(XLEN1+5.,-0.5,999)
STOP
END

SUBROUTINE PSTAGE(TITLE,FRMT,YMIN,YMAX,LEG)
REAL DAY(850),RSTG(850),STG(850),ESTG(850)
C ***** KEEP THE STATEMENTS BELOW *****
CHARACTER*(*)TITLE,FRMT,PTYPE*2,LEG,LEG1*30,LEG2*30,LEG3*80,DATE*8
COMMON /A/ XPLEG,YPLEG,XF,XD,YF,YD,XLEN1,YLEN1,NTH,
2          NPLOT,IBYEAR,IBMONTH
COMMON /B/ PTYPE
C *****
REWIND 7
REWIND 8
LEG1='REGULATION'
LEG2='HISTORICAL (DASH LINE)'
LEG3=LEG
CALL NEWPEN(1)
CALL FRAME(TITLE,22.,1.,4015.,8.,YMIN,YMAX,
$          'DAY','STAGE (FT MSL)','YL')
READ(7,'(F2.0)')SKIP
READ(8,'(F2.0)')SKIP
50  SKIP=0
DO 100 I=1,803
80  READ(7,FRMT,END=300,IOSTAT=IO)DATE,STG(I),ESTG(I)
    READ(8,FRMT)DATE,RSTG(I),ESTG(I)
    IF(DATE(3:3).EQ.'?')THEN
        SKIP=999
        GOTO 80
    ENDIF
    IF(SKIP.EQ.999)GOTO 300
    DAY(I)=JULIAN(DATE)-JULIAN('12-31-69')
    NPT=I
100 CONTINUE
300 CALL NEWPEN(4)
    CALL LPLOT(DAY,RSTG,NPT,3,60,LEG1)
    CALL NEWPEN(1)
    CALL LPLOT(DAY,STG,NPT,-4,60,LEG2)
    CALL NEWPEN(2)
    CALL LPLOT(DAY,ESTG,NPT,2,60,LEG3)

```

Appendix 3: Program Listing (KPLOT)

```

LEG1=LEG2=LEG3='
IF(IO.EQ.0)GOTO 50
RETURN
END

```

```

SUBROUTINE PFLOW(TITLE,FRMT,YMIN,YMAX,LEG)
REAL DAY1(50),DAY2(50),FLOW(50),EFLOW(50)
C ***** KEEP THE STATEMENTS BELOW *****
CHARACTER*(*)TITLE,FRMT,PTYPE*2,LEG,LEG1*10,LEG2*80,DATE*8
COMMON /A/ XPLEG,YPLEG,XF,XD,YF,YD,XLEN1,YLEN1,NTH,
2 NPLOT,IBYEAR,IBMONTH
COMMON /B/ PTYPE
C *****
REWIND 2
REWIND 9
LEG1='HISTORICAL'
LEG2=LEG
CALL NEWPEN(1)
CALL FRAME(TITLE,22.,1.,4015.,6.,YMIN,YMAX,
$ 'DAY','QUARTERLY MEAN FLOW (CFS)','YL')
READ(2,'(F2.0)')SKIP
READ(9,'(F2.0)')SKIP
50 SKIP=0
SUM=0
ESUM=0
J=0
NPT=0
DO 100 I=1,803
80 READ(2,FRMT,END=300,IOSTAT=IO)DATE,FLO
READ(9,FRMT)DATE,EFLO
IF (DATE(3:3).EQ.'?')THEN
SKIP=999
GOTO 80
ENDIF
IF(SKIP.EQ.999)GOTO 300
IF (DATE(1:2).EQ.'03'.AND.DATE(4:5).EQ.'28'
$ .OR. DATE(1:2).EQ.'06'.AND.DATE(4:5).EQ.'28'
$ .OR. DATE(1:2).EQ.'09'.AND.DATE(4:5).EQ.'28'
$ .OR. DATE(1:2).EQ.'12'.AND.DATE(4:5).EQ.'28')THEN
NPT=NPT+1
DAY1(NPT)=JULIAN(DATE)-JULIAN('12-31-69')-60
DAY2(NPT)=JULIAN(DATE)-JULIAN('12-31-69')-30
FLOW(NPT)=SUM/J
EFLOW(NPT)=ESUM/J
SUM=0
ESUM=0
J=0
ELSE
J=J+1
SUM=SUM+FLO
ESUM=ESUM+EFLO
ENDIF
100 CONTINUE

```

Appendix 3: Program Listing (KPLOT)

```

300  CALL NEWPEN(2)
      CALL BPLOT(DAY1, FLOW, NPT, 1, .15, LEG1)
      CALL NEWPEN(3)
      CALL BPLOT(DAY2, EFLOW, NPT, 3, .15, LEG2)
      LEG1=LEG2='
      IF(IO.EQ.0)GOTO 50
      RETURN
      END

      SUBROUTINE FSTAGE(TITLE, FRMT, YMIN, YMAX)
C *****
      REAL DAY(20), STG(20), RSTG(20), RAT(6)
      CHARACTER*(*) TITLE, FRMT, PTYPE*2, LEG*100, DATE(20)*8, JDATE*8
      COMMON /A/ XPLEG, YPLEG, XF, XD, YF, YD, XLEN1, YLEN1, NTH,
2          NPLOT, IBYEAR, IBMONTH
      COMMON /B/ PTYPE
C *****

      REWIND 5
      REWIND 8
      READ(8, '(F2.0)') SKIP
      READ(5, '(12X, 2I2, 6F6.2)') NMONTH, NCASE, (RAT(N), N=1, 6)
      READ(5, '(F6.2)') SKIP
      NPT=0
      DO 100 I=1, 300
      READ(5, 150, END=200) JDATE
150  FORMAT(1X, A8)
      IF(MOD(I+9, 10) .EQ. 0) THEN
          NPT=NPT+1
          DATE(NPT)=JDATE
      ENDIF
      IMAX=I
100  CONTINUE
200  CALL NEWPEN(1)
      IF(TITLE(1:4) .EQ. 'S-65') THEN
          CALL FRAME(TITLE, 9..1..120..6..YMIN, YMAX,
$ 'DAY', 'FLOW IN CFS', 'ML')
      ELSE
          CALL FRAME(TITLE, 9..1..120..6..YMIN, YMAX,
$ 'DAY', 'STAGE FT MSL', 'ML')
      ENDIF
      XPLEG=0.2
      YPLEG=-1.2
      WRITE(LEG, 250) DATE(1)
250  FORMAT('SIMULATION BASED ON CONDITIONS OF'.A8)
      NLEG=INDEX(LEG, ' ')
      CALL SYMBOL(XPLEG, YPLEG+0.3, 0.2, LEG, 0..NLEG)
      WRITE(LEG, '(9HRAIN\DATE, 15A6)')(DATE(N)(1:5), N=1, NPT)
      NLEG=INDEX(LEG, ' ')
      CALL SYMBOL(XPLEG+0.04, YPLEG, 0.1, LEG, 0..NLEG)
      DO 800 I=1, NCASE
      K=0
      DO 300 J=1, IMAX
          READ(8, FRMT) JDATE, DATA1, DATA2

```

Appendix 3: Program Listing (KPLOT)

```

        IF(MOD(J+9,10) .EQ. 0)THEN
            K=K+1
            DAY(K)=JULIAN(JDATE)-JULIAN(1)
            RSTG(K)=DATA1
            STG(K)=DATA2
        ENDIF
300  CONTINUE
        CALL NEWPEN(I)
        IRAT=RAT(I)*100
        IF(TITLE(1:4).EQ.'S-65')THEN
            WRITE(LEG,'(I4,4H % .15F6.0)')IRAT,(STG(N),N=1,NPT)
            CALL LPLOT(DAY,STG,NPT,I,1,LEG)
        ELSE
            WRITE(LEG,'(I4,4H % .15F6.2)')IRAT,(STG(N),N=1,NPT)
            CALL LPLOT(DAY,STG,NPT,I,1,LEG)
        ENDIF
800  CONTINUE
        IF(TITLE(1:4).EQ.'S-65')RETURN
        CALL NEWPEN(2)
        WRITE(LEG,'(8H REG STG.15F6.2)') (RSTG(N),N=1,NPT)
        CALL LPLOT(DAY,RSTG,NPT,-6,1,LEG)
        RETURN
        END

        FUNCTION JULIAN(1)
        CHARACTER*8 DATE
C *****
C     CONVERT CHARACTER STRING IN FORM 'MM-DD-YY' TO INTEGER DAYS
C     SINCE 12/31/1799 - YEARS LARGER THAN 1899 ARE ASSUMED
C *****
        INTEGER NDAYS(13),MONTH,DAY
        DATA NDAYS/0,31,59,90,120,151,181,212,243,273,304,334,365/
        READ (DATE,'(I2,1X,I2,1X,I2)') MONTH,DAY,IYR
        JULIAN = 365.25*(IYR+100) + NDAYS(MONTH) + DAY + 1
        IF (IYR.EQ.0 .OR. (MOD(IYR,4).EQ.0 .AND. MONTH.LE.2))
*           JULIAN = JULIAN - 1
        RETURN
        END

        SUBROUTINE FRAME(TITLE,XLEN,XMIN,XMAX,YLEN,YMIN,YMAX,
2           XLAB,YLAB,TYPE)
C *****
C *** SET FRAME FOR PLOTS
C *** TITLE -- TITLE OF PLOT IN HOLLERINTH
C *** XLEN -- LENGTH OF X-AXIS IN INCHES
C *** YLEN -- LENGTH OF Y-AXIS IN INCHES
C *** XMIN,XMAX,YMIN,YMAX -- MINIMUM AND MAXIMUM VALUES FOR X AND Y ARRAY
C *** XLAB -- LABEL OF X-AXIS IN HOLLERINTH
C *** YLAB -- LABEL OF Y-AXIS IN HOLLERINTH
C *** TYPE -- TYPE OF PLOT IN TWO LETTER HOLLERINTH
C ***           'LL'=LINEAR-LINEAR SCALE; 'GG'=LOG-LOG SCALE
C ***           'LG'=SEMILOG WITH LOG Y; 'GL'=SEMILOG WITH LOG X;

```

Appendix 3: Program Listing (KPLOT)

```

C ***      'NL'=NORMAL PROB WITH LINEAR Y; 'NG'=LOG-NORMAL WITH LOG Y
C ***      'EL'=GUMBEL PROB WITH LINEAR Y; 'EG'=LOG-GUMBEL WITH LOG Y
C ***      'ML','MG'=CALENDER MONTHS ON X-AXIS
C ***      'YL','YG'=CALENDER YEARS ON X-AXIS
C ***      NOTE: 1. PROBABILITY SCALE MUST BE ON X-AXIS.  XARRAY MUST BE
C ***              BETWEEN 0.01 AND 0.99.
C ***              2. XARRAY AND/OR YARRAY MUST BE GREATER THAN ZERO FOR LOG SCALE
C ***              3. CALENDER SCALE MUST BE ON X-AXIS.  XARRAY IS IN DAYS
C ***              FROM BEGINING YEAR, IBYEAR; OR FROM BEGINING MONTH, IBMONTH
C ***              MAXIMUM DAYS =14600(40-YEARS) FOR 'YL' AND 'ML'.
C ***              SPECIFY IBYEAR OR IBMONTH (I4 FORMAT) IN MAIN PROGRAM.
C *****
      REAL DUMPX(8),DUMPY(8)
      CHARACTER*(*) TITLE,XLAB,YLAB,TYPE*2,PTYPE*2,PTYPE1*1,PTYPE2*1
      CHARACTER YEAR(41)*4,YEAR1*4,MONTH(12)*3,MONTH1*3
      COMMON /A/ XPLEG,YPLEG,XF,XD,YF,YD,XLEN1,YLEN1,NTH,
2         NPLOT,IBYEAR,IBMONTH
      COMMON /B/ PTYPE
      DATA MONTH /'JAN','FEB','MAR',
2         'APR','MAY','JUN',
3         'JUL','AUG','SEP',
4         'OCT','NOV','DEC'/
      PTYPE=TYPE
      PTYPE1=PTYPE(1:1)
      PTYPE2=PTYPE(2:2)
      NTITLE=INDEX(TITLE,' ')
      NXLAB=INDEX(XLAB,' ')
      NYLAB=INDEX(YLAB,' ')
      IF(NTITLE.EQ.0)NTITLE=LEN(TITLE)
      IF(NXLAB.EQ.0)NXLAB=LEN(XLAB)
      IF(NYLAB.EQ.0)NYLAB=LEN(YLAB)
C *** SET PLOTTING POSITIONS FOR FRAME,TITLE AND LEGEND.
      XLEN1=XLEN
      YLEN1=YLEN
      HTITLE=0.2
      XPTITLE=XLEN1/2.-NTITLE*HTITLE/2.
      YPTITLE=YLEN1+0.1
      XPLEG=0.1
      YPLEG=YLEN1
      IF(NPLOT.LE.1)THEN
          CALL PLOT(0..0.5,-3)
      ELSE
          CALL PLOT(XLEN1+6..0..-3)
      ENDIF
      NPLOT=NPLOT+1
      CALL SYMBOL(XPTITLE,YPTITLE,HTITLE,TITLE,0.,NTITLE)
C *** SET LINEAR SCALE FOR X-AXIS
      IF(PTYPE1.EQ.'L')THEN
          XF=XMIN
          XD=(XMAX-XMIN)/XLEN1
          CALL AXIS(0..0.,XLAB,-NXLAB,XLEN1,0.,XF,XD)
C      ** SET X-GRID **
          CALL PLOT(0..0..3)
          NGRID=IFIX(XLEN1)

```


Appendix 3: Program Listing (KPLOT)

```

DO 10 I=0,NGRID
  X1=I
  CALL PLOT(X1,0.,3)
  CALL PLOT(X1,YLEN1,2)
10  CONTINUE
ENDIF
C *** SET LINEAR SCALE FOR Y-AXIS
IF(PTYPE2.EQ.'L')THEN
  YF=YMIN
  YD=(YMAX-YMIN)/YLEN1
  CALL AXIS(0.,0.,YLAB,NYLAB,YLEN1,90.,YF,YD)
C  ** SET Y-GRID **
  CALL PLOT(0.,0.,3)
  NGRID=IFIX(YLEN1)
  DO 20 I=0,NGRID,2
    Y1=I
    CALL PLOT(0.,Y1,3)
    CALL PLOT(XLEN1,Y1,2)
20  CONTINUE
ENDIF
C *** SET CALENDER YEAR FOR X-AXIS
IF(PTYPE1.EQ.'Y')THEN
  DO 60 I=1,41
    YEAR(I)=' '
60  CONTINUE
  NYEAR=(XMAX-1.)/365.25+0.99999
  DEV=XLEN1/NYEAR
  DO 70 I=1,NYEAR
    IYEAR=IBYEAR+(I-1)
    WRITE(YEAR(I),'(I4)')IYEAR
70  CONTINUE
  CALL PLOT(0.,0.,-3)
  DO 80 I=1,NYEAR
    YEAR1=YEAR(I)
    XPOS=(I-1.)*DEV+(DEV/2.-0.4)
    CALL SYMBOL(XPOS,-0.4,0.2,YEAR1,0.,4)
80  CONTINUE
  XF=1.
  XD=(NYEAR*365.25-1.)/XLEN1
C  ** SET X-GRID FOR CALENDER SCALE **
  NGRID=NYEAR
  DO 90 I=0,NGRID
    X1=I*DEV
    CALL PLOT(X1,0.,3)
    CALL PLOT(X1,YLEN1,2)
90  CONTINUE
ENDIF
C *** SET CALENDER MONTH FOR X-AXIS
IF(PTYPE1.EQ.'M')THEN
  NMONTH=(XMAX-1.)*12./365.25+0.99999
  DEV=XLEN1/NMONTH
  CALL PLOT(0.,0.,3)
  DO 100 I=1,NMONTH
    J=IBMONTH+(I-1)

```

Appendix 3: Program Listing (KPLOT)

```

        IF(J.GT.12)J=J-INT((J-.0001)/12)*12
        MONTH1=MONTH(J)
        XPOS=(I-1.)*DEV+(DEV/2.-0.3)
        CALL SYMBOL(XPOS,-0.4,0.2,MONTH1,0.,3)
100    CONTINUE
        XF=1.
        XD=(NMONTH*365.25/12.-1.)/XLEN1
C     ** SET X-GRID FOR CALENDER SCALE **
        NGRID=NMONTH
        DO 110 I=0,NGRID
            X1=I*DEV
            CALL PLOT(X1,0.,3)
            CALL PLOT(X1,YLEN1,2)
110    CONTINUE
        ENDIF
        RETURN
        END

        SUBROUTINE LPLOT(XARRAY,YARRAY,NPTS,NSYM,LINTYP,LEG)
C *****
C *** PLOTTING OF Y VERSUS X ON FRAME.
C *** IF MULTIPLE PLOTS ON SAME FRAME NEEDED, REPEAT CALL LPLOT .
C *** IF PLOT ON NEW FRAME NEEDED, CALL FRAME THEN CALL LPLOT.
C *** NPTS -- NUMBER OF DATA POINTS.
C *** NSYM -- SYMBOL INDEX NUMBER (0 THROUGH 13 WITH CENTERED SYMBOLS).
C ***      0="[]", 1="O", 2="A", 3="+", 4="X",5="<>", 8="Z",
C ***      9="Y", 11="*", 12="8", 13="1", 15="-", 17=" ",.....ETC.
C ***      IF NSYM IS NEGATIVE, DASHED LINE IS USED.
C *** LINTYP -- 0 = LINE BUT NO SYMBOL
C ***      POSITIVE NUMBER N = LINE AND SYMBOL FOR EVERY N DATA POINT
C ***      NEGATIVE NUMBER N = SYMBOL EVERY N DATA POINT BUT NO LINE
C *** LEG -- LEGEND IN HOLLERINTH
C *** (NTH) -- DEGREE POLYNOMIAL TO BE FITTED. THIS PARAMETER IS IN COMMON
C ***      STATEMENT. SPECIFY NTH IN MAIN PROGRAM IF FITTING IS NEEDED.
C ***      0(DEFAULT)=NO FITTING
C ***      1,2,3=1ST.2ND.3RD ORDER POYNOMIAL
C *****
        REAL XARRAY(*),YARRAY(*).C(10).
        2 XARRAY1(2000).YARRAY1(2000)
        CHARACTER*(*) LEG,PTYPE*2,PTYPE1*1,PTYPE2*1,IBCD*80
        COMMON /A/ XPLEG,YPLEG,XF,XD,YF,YD,XLEN1,YLEN1,NTH.
        2 NPLOT,IBYEAR,IBMONTH
        COMMON /B/ PTYPE
        PTYPE1=PTYPE(1:1)
        PTYPE2=PTYPE(2:2)
        NLEG=INDEX(LEG,' ')
        IF(NLEG.EQ.0)NLEG=LEN(LEG)
        YPLEG=YPLEG-0.18
        IF(LEG(1:3).NE.' ')
        $CALL SYMBOL(XPLEG,YPLEG,0.10,ABS(NSYM),0.,-1)
        CALL SYMBOL(XPLEG+0.14,YPLEG-0.05,0.10,LEG,0,NLEG)
C *** SET SCALING PARAMETERS
        K1=NPTS+1

```

Appendix 3: Program Listing (KPLOT)

```

K2=NPTS+2
XARRAY1(K1)=XF
XARRAY1(K2)=XD
YARRAY1(K1)=YF
YARRAY1(K2)=YD
C *** TRANSFORM XARRAY AND YARRAY TO XARRAY1 AND YARRAY1
DO 10 I=1,NPTS
  XARRAY1(I)=XARRAY(I)
  YARRAY1(I)=YARRAY(I)
  IF(PTYPE1.EQ.'G')XARRAY1(I)=LOG10(XARRAY(I))
  IF(PTYPE2.EQ.'G')YARRAY1(I)=LOG10(YARRAY(I))
  IF(PTYPE1.EQ.'N')XARRAY1(I)=ZZ(XARRAY(I))
  IF(PTYPE1.EQ.'E')XARRAY1(I)=-LOG(-LOG(XARRAY(I)))
10 CONTINUE
  IF(NTH.EQ.0)THEN
    IF(NSYM.GE.0)THEN
      CALL LINE(XARRAY1,YARRAY1,NPTS,1,LINTYP,NSYM)
    ELSE
      CALL DASHL(XARRAY1,YARRAY1,NPTS,1)
      IF(LINTYP.GT.0)
2      CALL LINE(XARRAY1,YARRAY1,NPTS,1,-LINTYP,ABS(NSYM))
    ENDIF
  ENDIF
  IF(NTH.GT.0)THEN
C *** FIT AN NTH ORDER POLYNOMIAL EQUATION IF NTH GREATER THEN ZERO
  C(1)=C(2)=C(3)=C(4)=C(5)=0.
  CALL CRVFT(XARRAY1,YARRAY1,NPTS,1,NTH,C,H)
  NCHR=36+(NTH-1)*22
  WRITE(IBCD,20) (C(J),J=NTH+1,1,-1)
20  FORMAT('Y = ',E13.7,' + ',E13.7,' *X + ',E13.7,' *X**2 + ',
2  E13.7,' *X**3')
  CALL SYMBOL(XLEN1+.2,.2,.12,IBCD,90.,NCHR)
C *** START PLOTTING A NTH ORDER POLYNOMIAL
  IC=3
  X=XF+XD*0.2
  MP=IFIX(XLEN1/0.1)
  DO 40 I=1,MP
    Y=C(1)*X**NTH+C(2)*X**(NTH-1)+C(3)*X**(NTH-2)+C(4)
    Y1=(Y-YF)/YD
    X1=(X-XF)/XD
    IF(X1.GE.(XLEN1-0.2))GOTO 50
    IF(Y1.GE.(YLEN1-0.2).OR.Y1.LE.0.2)GOTO 30
    CALL PLOT(X1,Y1,IC)
    IC=2
30  X1=X1+0.1
    X=XF+X1*XD
40  CONTINUE
50  IF(LINTYP.NE.0)
2  CALL LINE(XARRAY1,YARRAY1,NPTS,1,-ABS(LINTYP),NSYM)
  ENDIF
  RETURN
  END

```

Appendix 3: Program Listing (KPLOT)

```

SUBROUTINE BPLOT(XARRAY,YARRAY,NPTS,IHATCH,WID,LEG)
C *****
C *** PLOTTING OF BAR DIAGRAM OF Y VERSUS X
C *** ONLY LINEAR PLOTS 'LL','ML','YL' ALLOWED.
C *** IF MULTIPLE BARS ON SAME FRAME NEEDED, REPEAT CALL BPLOT.
C *** NPTS -- NUMBER OF DATA POINTS.
C *** IHATCH -- HATCHING INDEX
C ***          1=BAR ONLY, 2=0 DEGREE HATCH, 3=30 DEGREE HATCH,
C ***          4=60 DEGREE HATCH, 5=90 DEGREE HATCH
C *** WID -- WIDTH OF BAR IN INCH
C *** LEG -- LEGEND IN HOLLERINTH
C *****
      REAL XARRAY(*),YARRAY(*),XVA1(4),YVA1(4),XVA2(4),YVA2(4),HATCH(5)
      CHARACTER*(*) LEG.PTYPE*2
      COMMON /A/ XPLEG,YPLEG,XF,XD,YF,YD,XLEN1,YLEN1,NTH,
2          NPLOT,IBYEAR,IBMONTH
      COMMON /B/ PTYPE
      DATA HATCH/-999., 0., 30., 60., 90./
      ANG=HATCH(IHATCH)
      NLEG=INDEX(LEG,' ')
      IF(NLEG.EQ.0)NLEG=LEN(LEG)
      YPLEG=YPLEG-0.25
      IF(LEG(1:3).NE.' ')THEN
      CALL BAR(XPLEG,YPLEG,0.,0.2,WID,0.2,1,0)
      CALL SYMBOL(XPLEG+WID+0.1,YPLEG,0.1,LEG,0.,NLEG)
      XVA1(3)=XVA2(3)=0
      XVA1(4)=XVA2(4)=1
      YVA1(3)=YVA2(3)=0
      YVA1(4)=YVA2(4)=1
      XVA1(1)=XVA1(2)=XPLEG
      XVA2(1)=XVA2(2)=XPLEG+WID
      YVA1(1)=YVA2(1)=YPLEG
      YVA1(2)=YVA2(2)=YPLEG+0.2
      IF(IHATCH.NE.1)
$ CALL SHADE(XVA1,YVA1,XVA2,YVA2,0.0625,ANG,2,1,2,1)
      ENDIF
      DO 10 I=1,NPTS
      XP=(XARRAY(I)-XF)/XD
      HEI=(YARRAY(I)-YF)/YD
      CALL BAR(XP-WID/2.,0.,0.,HEI,WID,HEI,1,0)
      XVA1(1)=XVA1(2)=XP-WID/2.
      XVA2(1)=XVA2(2)=XP+WID/2.
      YVA1(1)=YVA2(1)=0.
      YVA1(2)=YVA2(2)=HEI
      IF(IHATCH.NE.1)
$ CALL SHADE(XVA1,YVA1,XVA2,YVA2,0.0625,ANG,2,1,2,1)
10 CONTINUE
      RETURN
      END

```

Appendix 4: Program Listing (KBUDGET)

```

PROGRAM KBUDGET
C *****
C *** WATER BUDGET MODEL FOR KISSIMMEE CHAIN OF LAKES
C *** TAPE1,TAPE2=HYDROLOGIC INPUT DATA GENERATED BY SIR PROGRAM 'UPKISS'
C *** TAPE5,TAPE6=SUMMARY OUTPUT OF RESIDUALS
C *** TAPE11 THRU TAPE19=DETAILED OUTPUT PER LAKE
C *****
COMMON /B/ALLI(13),MYRT(13),HART(13),GENT(13),
$      ETOH(13),TOHO(13),CYPR(13),HATC(13),KISS(13)
REAL MYRT,KISS
DIMENSION IDAY(365),JDAY(365),RAIN1(365),RAIN2(365),
$ RAIN3(365),RAIN4(365),RAIN5(365),RAIN6(365),
$ RAIN7(365),RAIN8(365),RAIN9(365),EVAP(365),
$ STG1(0:366),STG2(0:366),STG3(0:366),STG4(0:366),
$ STG5(0:366),STG6(0:366),STG7(0:366),STG8(0:366),
$ STG9(0:366),S58(365),S57(365),S62(365),S60(365),
$ S63(365),S59(365),S61(365),S65(365),SHIN(365),
$ BOGG(365),CATF(365),REED(365)
CHARACTER LAKNAME*4,IDATE(365)*8,JDATE(365)*8
C *** PRINT HEADING ON TAPE11 THRU TAPE19 IF DETAILED OUTPUT NEEDED ***
PRINT*,'DETAILED OUTPUT NEEDED? (YES=1,NO=0)'
READ*,IPRINT
IF(IPRINT.EQ.1)THEN
  WRITE(11,'(1H1,20X,11HTAPE11=ALLI)')
  WRITE(12,'(1H1,20X,11HTAPE12=MYRT)')
  WRITE(13,'(1H1,20X,11HTAPE13=HART)')
  WRITE(14,'(1H1,20X,11HTAPE14=GENT)')
  WRITE(15,'(1H1,20X,11HTAPE15=ETOH)')
  WRITE(16,'(1H1,20X,11HTAPE16=TOHO)')
  WRITE(17,'(1H1,20X,11HTAPE17=CYPR)')
  WRITE(18,'(1H1,20X,11HTAPE18=HATC)')
  WRITE(19,'(1H1,20X,11HTAPE19=KISS)')
ENDIF
C *** READ INPUT HYDROLOGIC DATA ***
90 DO 200 I=1,365
  READ(1,110,END=230,Iostat=IO)
  $ IDATE(I),IDAY(I),RAIN1(I),RAIN2(I),RAIN3(I),RAIN4(I),
  $ RAIN5(I),RAIN6(I),RAIN7(I),RAIN8(I),RAIN9(I),EVAP(I),
  $ STG1(I),STG2(I),STG3(I),STG4(I),STG5(I),STG6(I),
  $ STG7(I),STG8(I),STG9(I)
110 FORMAT(A8,1X,I6,1X,F5.2,T17,9F5.2,9F6.2)
  READ(2,120)
  $ JDATE(I),JDAY(I),S58(I),S57(I),S62(I),S60(I),S63(I),
  $ S59(I),S61(I),S65(I),SHIN(I),BOGG(I),CATF(I),REED(I)
120 FORMAT(A8,1X,I6,1X,12F6.0)
C *** CHECK IF DATES MATCH ***
IF(IDAY(I).NE.JDAY(I))THEN
  PRINT*,'DATES NOT MATCH!!!'
  PRINT*,'I,IDATE,JDATE,IDAY,JDAY'
  WRITE(*,140)I,IDATE(I),JDATE(I),IDAY(I),JDAY(I)
140 FORMAT(I3,1X,2A9,2I8)
  STOP
ENDIF
NDAY=I

```

Appendix 4: Program Listing (KBUDGET)

```

200 CONTINUE
C *** INITIALIZE DAY0 AND DAY366 BY FORWARD OR BACKWARD DIFFERENCE APPROXIMATION
230 STG1(0)=STG1(1)-(STG1(2)-STG1(1))
    STG1(NDAY+1)=STG1(NDAY)+(STG1(NDAY)-STG1(NDAY-1))
    STG2(0)=STG2(1)-(STG2(2)-STG2(1))
    STG2(NDAY+1)=STG2(NDAY)+(STG2(NDAY)-STG2(NDAY-1))
    STG3(0)=STG3(1)-(STG3(2)-STG3(1))
    STG3(NDAY+1)=STG3(NDAY)+(STG3(NDAY)-STG3(NDAY-1))
    STG4(0)=STG4(1)-(STG4(2)-STG4(1))
    STG4(NDAY+1)=STG4(NDAY)+(STG4(NDAY)-STG4(NDAY-1))
    STG5(0)=STG5(1)-(STG5(2)-STG5(1))
    STG5(NDAY+1)=STG5(NDAY)+(STG5(NDAY)-STG5(NDAY-1))
    STG6(0)=STG6(1)-(STG6(2)-STG6(1))
    STG6(NDAY+1)=STG6(NDAY)+(STG6(NDAY)-STG6(NDAY-1))
    STG7(0)=STG7(1)-(STG7(2)-STG7(1))
    STG7(NDAY+1)=STG7(NDAY)+(STG7(NDAY)-STG7(NDAY-1))
    STG8(0)=STG8(1)-(STG8(2)-STG8(1))
    STG8(NDAY+1)=STG8(NDAY)+(STG8(NDAY)-STG8(NDAY-1))
    STG9(0)=STG9(1)-(STG9(2)-STG9(1))
    STG9(NDAY+1)=STG9(NDAY)+(STG9(NDAY)-STG9(NDAY-1))
C *** BEGIN WATER BUDGET COMPUTATION
  DO 500 I=1,NDAY
    READ(IDATE(I), '(I2,1X,I2,1X,I2)')IM, ID, IY
C *** LAKE ALLIGATOR ***
    STG1A=STG1(I-1)
    STG1B=STG1(I)
    STG1C=STG1(I+1)
    CALL STORAGE(ALLI,STG1A,VOL1A,AREA1A)
    CALL STORAGE(ALLI,STG1B,VOL1,AREA1)
    CALL STORAGE(ALLI,STG1C,VOL1C,AREA1C)
    DSTOR1=(VOL1C-VOL1A)/2.
    RES1=.8*EVAP(I)/12.*8258-RAIN1(I)/12.*8258+S60(I)*1.984+
$     S58(I)*1.984+DSTOR1
C *** LAKE MYRTLE AND PRESTON ***
    STG2A=STG2(I-1)
    STG2B=STG2(I)
    STG2C=STG2(I+1)
    CALL STORAGE(MYRT,STG2A,VOL2A,AREA2A)
    CALL STORAGE(MYRT,STG2B,VOL2,AREA2)
    CALL STORAGE(MYRT,STG2C,VOL2C,AREA2C)
    DSTOR2=(VOL2C-VOL2A)/2
    RES2=.8*EVAP(I)/12.*1732-RAIN2(I)/12.*1732+S57(I)*1.984-
$     S58(I)*1.984+DSTOR2
C *** LAKE HART AND MARY JANE ***
    STG3A=STG3(I-1)
    STG3B=STG3(I)
    STG3C=STG3(I+1)
    CALL STORAGE(HART,STG3A,VOL3A,AREA3A)
    CALL STORAGE(HART,STG3B,VOL3,AREA3)
    CALL STORAGE(HART,STG3C,VOL3C,AREA3C)
    DSTOR3=(VOL3C-VOL3A)/2.
    RES3=.8*EVAP(I)/12.*3643-RAIN3(I)/12.*3643+S62(I)*1.984-
$     S57(I)*1.984+DSTOR3

```

Appendix 4: Program Listing (KBUDGET)

```

C ***LAKE GENTRY ***
  STG4A=STG4(I-1)
  STG4B=STG4(I)
  STG4C=STG4(I+1)
  CALL STORAGE(GENT,STG4A,VOL4A,AREA4A)
  CALL STORAGE(GENT,STG4B,VOL4,AREA4)
  CALL STORAGE(GENT,STG4C,VOL4C,AREA4C)
  DSTOR4=(VOL4C-VOL4A)/2.
  RES4=.8*EVAP(I)/12.*1776-RAIN4(I)/12.*1776+S63(I)*1.984-
$   S60(I)*1.984+DSTOR4
C *** LAKE EAST TOHO ***
  STG5A=STG5(I-1)
  STG5B=STG5(I)
  STG5C=STG5(I+1)
  CALL STORAGE(ETOH,STG5A,VOL5A,AREA5A)
  CALL STORAGE(ETOH,STG5B,VOL5,AREA5)
  CALL STORAGE(ETOH,STG5C,VOL5C,AREA5C)
  DSTOR5=(VOL5C-VOL5A)/2.
  RES5=.8*EVAP(I)/12.*12600-RAIN5(I)/12.*12600+S59(I)*1.984-
$   S62(I)*1.984-BOGG(I)*1.984+DSTOR5
C *** LAKE TOHO ***
  STG6A=STG6(I-1)
  STG6B=STG6(I)
  STG6C=STG6(I+1)
  CALL STORAGE(TOHO,STG6A,VOL6A,AREA6A)
  CALL STORAGE(TOHO,STG6B,VOL6,AREA6)
  CALL STORAGE(TOHO,STG6C,VOL6C,AREA6C)
  DSTOR6=(VOL6C-VOL6A)/2.
  RES6=.8*EVAP(I)/12.*21400-RAIN6(I)/12.*21400+S61(I)*1.984-
$   S59(I)*1.984-SHIN(I)*1.984+DSTOR6
C *** LAKE CYPRESS ***
  STG7A=STG7(I-1)
  STG7B=STG7(I)
  STG7C=STG7(I+1)
  CALL STORAGE(CYPR,STG7A,VOL7A,AREA7A)
  CALL STORAGE(CYPR,STG7B,VOL7,AREA7)
  CALL STORAGE(CYPR,STG7C,VOL7C,AREA7C)
  DSTOR7=(VOL7C-VOL7A)/2.
C --- COMPUTATION OF FLOW AT C36 AND C37
  HEAD78=STG7(I)-STG8(I)
  HEAD89=STG8(I)-STG9(I)
  Q36=35.61885873*(ABS(HEAD78)+1E-20)**.5511796*(STG7(I)-35.)**1.666
7
  Q37=87.07430164*(ABS(HEAD89)+1E-20)**.4976433*(STG8(I)-42.)**1.666
7
  DVOL7=HEAD78*AREA7+(S63(I)+S61(I)+.3*REED(I))*1.9835
  DVOL7=1.13*DVOL7*7480./(7480.+3850.)
  IF(DVOL7.LE.0.)THEN
    C36=MAX(-Q36,DVOL7/1.9835)
  ELSE
    C36=MIN(Q36,DVOL7/1.9835)
  ENDIF
  RES7=.8*EVAP(I)/12.*4274-RAIN7(I)/12.*4274+C36*1.984-
$   S63(I)*1.984-S61(I)*1.984-REED(I)*1.984*.3+DSTOR7

```

Appendix 4: Program Listing (KBUDGET)

```

C *** LAKE HATCHINEHA ***
  STG8A=STG8(I-1)
  STG8B=STG8(I)
  STG8C=STG8(I+1)
  CALL STORAGE(HATC,STG8A,VOL8A,AREA8A)
  CALL STORAGE(HATC,STG8B,VOL8,AREA8)
  CALL STORAGE(HATC,STG8C,VOL8C,AREA8C)
  DSTOR8=(VOL8C-VOL8A)/2.
  DVOL8=HEAD89*AREA8+(C36+CATF(I)+.7*REED(I))*1.9835
  DVOL8=1.10*DVOL8*34141./(34141.+7480.)
  IF(DVOL8.LE.0.)THEN
    C37=MAX(-Q37,DVOL8/1.9835)
  ELSE
    C37=MIN(Q37,DVOL8/1.9835)
  ENDIF
  RES8=.8*EVAP(I)/12.*9733-RAIN8(I)/12.*9733+C37
  $ *1.984-C36*1.984-CATF(I)*1.984
  $ -REED(I)*1.984*.7+DSTOR8
C *** LAKE KISSIMMEE ***
  STG9A=STG9(I-1)
  STG9B=STG9(I)
  STG9C=STG9(I+1)
  CALL STORAGE(KISS,STG9A,VOL9A,AREA9A)
  CALL STORAGE(KISS,STG9B,VOL9,AREA9)
  CALL STORAGE(KISS,STG9C,VOL9C,AREA9C)
  DSTOR9=(VOL9C-VOL9A)/2.
  RES9=.8*EVAP(I)/12.*42607-RAIN9(I)/12.*42607+
  $ S65(I)*1.984-C37*1.984+DSTOR9
C *** SUMMARY PRINTOUT OF RESIDUALS ***
  WRITE(5,250)IDATE(I),EVAP(I),SHIN(I),BOGG(I),
  $ RAIN1(I),STG1(I),RES1,RAIN2(I),STG2(I),RES2,
  $ RAIN3(I),STG3(I),RES3,RAIN4(I),STG4(I),RES4
  WRITE(6,260)IDATE(I),REED(I),CATF(I),S65(I),
  $ RAIN5(I),STG5(I),RES5,RAIN6(I),STG6(I),RES6,
  $ RAIN7(I),STG7(I),RES7,RAIN8(I),STG8(I),RES8,
  $ RAIN9(I),STG9(I),RES9
250  FORMAT(' ',A8,F6.2,2F6.0,4(F6.2,F6.2,F7.0))
260  FORMAT(' ',A8,3F6.0,5(F6.2,F6.2,F7.0))
C *** DETAILED OUTPUT OPTION OF EACH LAKE IF IPRINT=1 ***
  IF(IPRINT.EQ.1)THEN
    WRITE(11,300)IDATE(I),RAIN1(I),EVAP(I),STG1(I),AREA1,VOL1,
  $ DSTOR1,S60(I),S58(I),RES1
300  FORMAT(' ',A8,1X,3F6.2,3F9.0,10F7.0)
    WRITE(12,300)IDATE(I),RAIN2(I),EVAP(I),STG2(I),AREA2,VOL2,
  $ DSTOR2,S57(I),S58(I),RES2
    WRITE(13,300)IDATE(I),RAIN3(I),EVAP(I),STG3(I),AREA3,VOL3,
  $ DSTOR3,S57(I),S62(I),RES3
    WRITE(14,300)IDATE(I),RAIN4(I),EVAP(I),STG4(I),AREA4,VOL4,
  $ DSTOR4,S60(I),S63(I),RES4
    WRITE(15,300)IDATE(I),RAIN5(I),EVAP(I),STG5(I),AREA5,VOL5,
  $ DSTOR5,S62(I),S59(I),BOGG(I),RES5
    WRITE(16,300)IDATE(I),RAIN6(I),EVAP(I),STG6(I),AREA6,VOL6,
  $ DSTOR6,S59(I),S61(I),SHIN(I),RES6
    WRITE(17,300)IDATE(I),RAIN7(I),EVAP(I),STG7(I),AREA7,VOL7,

```


Appendix 4: Program Listing (KBUDGET)

```

$   DSTOR7,C36,S61(I),S63(I),REED(I),RES7
    WRITE(18,300)IDATE(I),RAIN8(I),EVAP(I),STG8(I),AREA8,VOL8,
$   DSTOR8,C36,C37,CATF(I),REED(I),RES8
    WRITE(19,300)IDATE(I),RAIN9(I),EVAP(I),STG9(I),AREA9,VOL9,
$   DSTOR9,C37,S65(I),RES9
    ENDIF
500  CONTINUE
    IF (IO.EQ.0)GOTO 90
    STOP
    END

```

```

      SUBROUTINE STORAGE(LAKE,STAGE,STOR,AREA)
C *****
C ***      STORAGE AND AREA RATING FUNCTIONS
C *** LAKE(1) TO (5) = PARAMETERS C1 TO C5 FOR STORAGE CALCULATION
C *** LAKE(6) TO (10) = PARAMETERS D1 TO D5 FOR AREA CALCULATION
C *** LAKE(11)=MAXIMUM STAGE LIMIT FOR RATING CURVE
C *** LAKE(12)=MINIMUM STAGE LIMIT FOR RATING CURVE
C *** LAKE(13)=LAKE ID NUMBER
C *****
    REAL LAKE(13)
    V(S)=C1*S**4+C2*S**3+C3*S**2+C4*S+C5
    A(S)=D1*S**4+D2*S**3+D3*S**2+D4*S+D5
    DA(S)=4.*D1*S**3+3.*D2*S**2+2*D3*S+D4
    C1=LAKE(1)
    C2=LAKE(2)
    C3=LAKE(3)
    C4=LAKE(4)
    C5=LAKE(5)
    D1=LAKE(6)
    D2=LAKE(7)
    D3=LAKE(8)
    D4=LAKE(9)
    D5=LAKE(10)
    STGMAX=LAKE(11)
    STGMIN=LAKE(12)
    STG=STAGE
    IF (STG.GT.STGMAX) THEN
      AREA=A(STGMAX)+DA(STGMAX)*(STG-STGMAX)
      STOR=V(STGMAX)+(AREA+A(STGMAX))/2.*(STG-STGMAX)
      RETURN
    ENDIF
    IF (STG.LT.STGMIN) THEN
      AREA=A(STGMIN)*(STG/STGMIN)**1.5
      STOR=V(STGMIN)*(STG/STGMIN)**1.5
      RETURN
    ENDIF
    STOR=V(STG)
    AREA=A(STG)
    RETURN
    END

```

Appendix 4: Program Listing (KBUDGET)

BLOCK DATA

COMMON /B/ALLI(13),MYRT(13),HART(13),GENT(13),

\$ ETOH(13),TOHO(13),CYPR(13),HATC(13),KISS(13)

REAL MYRT,KISS

DATA ALLI

\$/-4.4642004 , 1238.5192 , -126982.37 , 5724291.8 ,

\$ -.95950501E+08, -22.386100 , 5641.0255 , -532464.04 ,

\$.22314593E+08, -.35033754E+09, 65., 59.5, 1./

DATA MYRT

\$/ 3.0002689 , -707.76736 , 62675.634 , -2468063.4 ,

\$.36452945E+08, 2.0264439 , -481.92500 , 42983.139 ,

\$ -1703871.9 , .25327409E+08, 65., 58., 2./

DATA HART

\$/ .21157913 , -41.567386 , 3097.0676 , -102266.56 ,

\$ 1244906.4 , .29565288 , -67.473332 , 5783.8343 ,

\$ -220443.89 , 3150674.3 , 64., 56., 3./

DATA GENT

\$/ 2.9258007 , -702.73697 , 63312.710 , -2534177.8 ,

\$.38012042E+08, 1.5462534 , -366.74399 , 32608.560 ,

\$ -1288094.7 , .19073801E+08, 65., 57., 4./

DATA ETOH

\$ /2.829373, -649.2139, 56075.09, -2148076.,

\$ 30728530., -.263521, 64.99214, -5948.022, 240362.1,

\$ -3617802., .65., 50., 5./

DATA TOHO

\$ /0, 24.71876, -3310.8, 161410.9, -2828022.,

\$ -1.661901, 367.9039, -30441.02, 1117249., -15345700.,

\$ 60., 49., 6./

DATA CYPR

\$/ .43142302 , -91.027331 , 7294.9886 , -258747.85 ,

\$ 3400561.1, 0., 1.7256921 , -273.08199 , 14589.977 ,

\$ -258747.85 , 58., 43., 7./

DATA HATC

\$/-1.5127289 , 284.80337 , -19506.905 , 578132.96 ,

\$ -6250583.0 , 0., -6.0509156 , 854.41011 , -39013.810 ,

\$ 578132.96 , 55., 45., 8./

DATA KISS

\$/ 14.023537 , -2683.7603 , 193335.42 , -6183331.0 ,

\$.73906134E+08, 0., 56.094148 , -8051.2809 , 386670.84 ,

\$ -6183331.0 , 58., 42.5, 9./

END

Appendix 5: Interactive Computer Session

Simulation Run

85/11/21. 14.09.53. T14
(0)SOUTH FLA WATER MGMT DIST S/N 675. NOS 2.3 617/587.
USER NAME: XXXXXXXX
JSN: ABZK, NAMIAF
CHARGE NUMBER:XXXXXXXXXXXX

bat
/get,kroute/un=afan
/ftn5,i=kroute,l=0
7.232 CP SECONDS COMPILATION TIME.
/lgo
ROUTING OPTION? (1=FORECAST,2=SIMULATION,3=CALIBRATION)
? 2
TIME LIMIT
ENTER T TO CONTINUE OR CR KEY TO STOP:
t.*

STOP
384.447 CP SECONDS EXECUTION TIME.

Simulation run completed.
Results written to TAPE7,TAPE8,TAPE9

/attach,calcomp/un=library
/library,calcomp
LIBRARY,CALCOMP.
/get,kplot/un=afan
/ftn5,i=kplot,l=0,b=bplot
3.058 CP SECONDS COMPILATION TIME.
/bplot
PLOTTING OPTION: 1=STAGE,2=FLOW,3=BOTH,4=FORECAST

Set up plotting run

? 3
ENTER ASSUMPTION USED IN HOLLERINTH
? 'simulation 11-20-85'
TIME LIMIT
ENTER T TO CONTINUE OR CR KEY TO STOP:
t.*

Plot both stages and flows

STOP
251.722 CP SECONDS EXECUTION TIME.
get.procfil/un=afan
/begin.runplot
REVERT.COPY TAPE13 TO PLOT12 COMPLETED

Send plots to Calcomp plotter
Plots are shown in Pages 76 through 96

Appendix 5: Interactive Computer Session

Partial Listing of Output (Simulation Run)

TAPE7 contains historical (STG) and estimated (ESTG) stages.

STG1 to STG9 refer to Lakes Alligator to Kissimmee

/xedit,tape7

XEDIT 3.1.00

?? p5

DATE	STG1	ESTG1	STG2	ESTG2	STG3	ESTG3	STG4	ESTG4	STG5	ESTG5	STG6	ESTG6	STG7	ESTG7	STG8
01-02-70	63.82	63.88	63.37	63.24	61.03	61.05	61.81	61.92	57.71	57.88	54.87	54.88	52.38	52.39	52.30
01-03-70	63.87	63.94	63.39	63.14	61.07	61.04	61.83	61.97	57.76	57.96	54.91	54.93	52.41	52.37	52.33
01-04-70	63.87	63.97	63.37	62.99	61.08	61.02	61.74	61.95	57.76	57.99	54.89	54.98	52.43	52.37	52.32
01?05?70	63.86	63.98	63.29	62.83	61.05	61.01	61.64	61.99	57.76	58.00	54.95	55.03	52.43	52.53	52.34
01-06-70	63.88	63.99	63.28	62.73	61.08	61.02	61.60	62.02	57.73	58.03	55.07	55.05	52.63	52.77	52.43

52.35 52.34 52.44 2270. 1051.
52.36 52.47 52.50 2270. 2358.
52.31 52.53 52.45 2270. 1949.
52.38 ***** This day contains missing records ("?" in columns 3 and 6)
52.52 52.29 52.41 2920. 3076.

?? stop

TAPE8 contains regulation (RSTG) and estimated (ESTG) stages, and S-65 flows.

/xedit,tape8

XEDIT 3.1.00

?? p5

DATE	RSTG1	ESTG1	RSTG2	ESTG2	RSTG3	ESTG3	RSTG4	ESTG4	RSTG5	ESTG5	RSTG6	ESTG6	RSTG7	ESTG7	RSTG8
01-02-70	63.99	63.88	61.99	63.24	60.99	61.05	61.99	61.92	58.00	57.88	55.00	54.88	52.39	52.39	52.39
01-03-70	63.97	63.94	61.97	63.14	60.97	61.04	61.97	61.97	58.00	57.96	55.00	54.93	52.37	52.37	52.37
01-04-70	63.96	63.97	61.96	62.99	60.96	61.02	61.96	61.95	58.00	57.99	55.00	54.98	52.36	52.37	52.36
01?05?70	63.95	63.98	61.95	62.83	60.95	61.01	61.95	61.99	58.00	58.00	55.00	55.03	52.35	52.53	52.35
01-06-70	63.93	63.99	61.93	62.73	60.93	61.02	61.93	62.02	58.00	58.03	55.00	55.05	52.34	52.77	52.34

52.35 52.39 52.44 2270. 1051.
52.36 52.37 52.50 2270. 2358.
52.31 52.36 52.45 2270. 1949.
52.38 ***** This day contains missing records ("?" in columns 3 and 6)
52.52 52.34 52.41 2920. 3076.

?? stop

TAPE9 contains estimated flows

/xedit,tape9

XEDIT 3.1.00

?? p5

DATE	S-58	S-57	S-62	S-60	S-63	S-59	S-61	S-65	SHIN	BOGG	CATF	REED	C-36	C-37
01-02-70	0.	230.	339.	0.	0.	0.	0.	1051.	128.	58.	80.	294.	102.	-33.
01-03-70	0.	230.	355.	0.	0.	210.	0.	2358.	207.	79.	95.	310.	82.	-272.
01-04-70	0.	230.	338.	25.	7.	387.	355.	1949.	235.	111.	84.	323.	400.	-33.
01?05?70	0.	230.	339.	129.	173.	424.	1023.	*****	234.	114.	83.	344.	1038.	857.
01-06-70	0.	230.	386.	245.	320.	708.	1566.	3076.	355.	156.	84.	409.	1714.	1484.

?? stop

Appendix 5: Interactive Computer Session

Calibration Run (without optimization)

85/11/21. 14.09.53. T14
(0)SOUTH FLA WATER MGMT DIST S/N 675. NOS 2.3 617/587.
USER NAME: XXXXXXXX
JSN: ABZK, NAMIAF
CHARGE NUMBER:XXXXXXXXXXXXX

bat
/get,kroute/un=afan
/ftn5,i=kroute,l=0
7.232 CP SECONDS COMPILATION TIME.
lgo
ROUTING OPTION? (1=FORECAST,2=SIMULATION,3=CALIBRATION)
? 3
OPTIMIZATION OF PARAMETERS NEEDED? 1=NO,2=YES
? 1
TIME LIMIT
ENTER T TO CONTINUE OR CR KEY TO STOP:
t.*
384.447 CP SECONDS EXECUTION TIME.

Calibration run completed.
Results written to TAPE7,TAPE8 and TAPE9.

/attach,calcomp/un=library
/library,calcomp
LIBRARY,CALCOMP.
/get,kplot/un=afan
/ftn5,i=kplot,l=0.b=bplot
3.058 CP SECONDS COMPILATION TIME.
/bplot
PLOTING OPTION: 1=STAGE,2=FLOW,3=BOTH,4=FORECAST
? 3
ENTER ASSUMPTION USED IN HOLLERINTH
? 'calibration 11-21-85'
TIME LIMIT
ENTER T TO CONTINUE OR CR KEY TO STOP:
t.*
STOP
251.722 CP SECONDS EXECUTION TIME.

Set up plotting run

get,procfil/un=afan
/begin,runplot
REVERT.COPY TAPE13 TO PLOT12 COMPLETED

Send plots to Calcomp plotter
Plots are shown in Pages 76 to 96

Appendix 5: Interactive Computer Session

Calibration Run (with optimization)

```
85/11/21. 14.09.53. T14
(O)SOUTH FLA WATER MGMT DIST S/N 675. NOS 2.3 617/587.
USER NAME: XXXXXXXX
JSN: ABZK, NAMIAF
  CHARGE NUMBER:XXXXXXXXXXXXX
bat
/get,kroute/un=afan
/ftn5,i=kroute,l=0
  7.232 CP SECONDS COMPILATION TIME./lgo
ROUTING OPTION? (1=FORECAST,2=SIMULATION,3=CALIBRATION)
? 3
OPTIMIZATION OF PARAMETERS NEEDED? 1=NO,2=YES
? 2
SELECT ONE TO OPTIMIZE:1=SCOEF,2=HMAX,3=PCOEF.4=ROOT
? 2
*TIME LIMIT*
ENTER T TO CONTINUE OR CR KEY TO STOP:
t.*
  42002.447 CP SECONDS EXECUTION TIME.
```

Optimization run completed. Since optimization takes many CPU hours, it is best to set up a batch job run.

Optimization Results

Optimized parameters are underlined

Xedit tape20

??p1000

PARAMETER HMAX

ID= 1 ITER= 1 PARAM 2= 6.000 FUNC= 1089.	ID= 6 ITER= 1 PARAM 2= 4.930 FUNC= 1363.
ID= 1 ITER= 2 PARAM 2= 6.120 FUNC= 1067.	ID= 6 ITER= 2 PARAM 2= 5.029 FUNC= 1387.
ID= 1 ITER= 3 PARAM 2=10.600 FUNC= 1398.	ID= 6 ITER= 3 PARAM 2= .709 FUNC= 6448.
ID= 1 ITER= 4 PARAM 2= 6.210 FUNC= 1070.	ID= 6 ITER= 4 PARAM 2= 4.942 FUNC= 1366.
<u>ID= 1 ITER= 5 PARAM 2= 6.122 FUNC= 1067.</u>	ID= 6 ITER= 5 PARAM 2= 4.856 FUNC= 1346.
	ID= 6 ITER= 6 PARAM 2= 4.771 FUNC= 1326.
	ID= 6 ITER= 7 PARAM 2= 4.682 FUNC= 1308.
ID= 2 ITER= 1 PARAM 2= 6.000 FUNC= 1187.	
ID= 2 ITER= 2 PARAM 2= 6.120 FUNC= 1192.	ID= 6 ITER= 8 PARAM 2= 4.590 FUNC= 1290.
ID= 2 ITER= 3 PARAM 2= .000 FUNC= .1663E+06	ID= 6 ITER= 9 PARAM 2= 4.491 FUNC= 1271.
<u>ID= 2 ITER= 4 PARAM 2= 5.687 FUNC= 1178.</u>	ID= 6 ITER=10 PARAM 2= 4.387 FUNC= 1255.
ID= 2 ITER= 5 PARAM 2= 5.125 FUNC= 1204.	ID= 6 ITER=11 PARAM 2= 4.271 FUNC= 1239.
ID= 2 ITER= 6 PARAM 2= 5.676 FUNC= 1180.	ID= 6 ITER=12 PARAM 2= 4.133 FUNC= 1225.
	ID= 6 ITER=13 PARAM 2= 3.954 FUNC= 1216.
ID= 3 ITER= 1 PARAM 2= 6.000 FUNC= 1942.	ID= 6 ITER=14 PARAM 2= 3.587 FUNC= 1245.
ID= 3 ITER= 2 PARAM 2= 6.120 FUNC= 1932.	<u>ID= 6 ITER=15 PARAM 2= 3.947 FUNC= 1216.</u>
ID= 3 ITER= 3 PARAM 2=22.894 FUNC= 1959.	
ID= 3 ITER= 4 PARAM 2= 6.455 FUNC= 1908.	ID= 7 ITER= 1 PARAM 2= 4.210 FUNC= 1046.
ID= 3 ITER= 5 PARAM 2= 6.856 FUNC= 1888.	ID= 7 ITER= 2 PARAM 2= 4.294 FUNC= 1045.
ID= 3 ITER= 6 PARAM 2= 7.426 FUNC= 1869.	ID= 7 ITER= 3 PARAM 2=49.893 FUNC= 1297.
ID= 3 ITER= 7 PARAM 2= 8.255 FUNC= 1852.	<u>ID= 7 ITER= 4 PARAM 2= 5.206 FUNC= 1043.</u>
ID= 3 ITER= 8 PARAM 2= 9.621 FUNC= 1847.	ID= 7 ITER= 5 PARAM 2=19.015 FUNC= 1193.
ID= 3 ITER= 9 PARAM 2=17.662 FUNC= 1917.	ID= 7 ITER= 6 PARAM 2= 5.482 FUNC= 1046.
ID= 3 ITER=10 PARAM 2= 9.782 FUNC= 1847.	ID= 7 ITER= 7 PARAM 2= 5.212 FUNC= 1044.
<u>ID= 3 ITER=11 PARAM 2= 9.624 FUNC= 1847.</u>	
	ID= 8 ITER= 1 PARAM 2= 5.440 FUNC= 978.2
ID= 4 ITER= 1 PARAM 2= 6.000 FUNC= 1885.	ID= 8 ITER= 2 PARAM 2= 5.549 FUNC= 981.7
ID= 4 ITER= 2 PARAM 2= 6.120 FUNC= 1884.	ID= 8 ITER= 3 PARAM 2= .000 FUNC= 1381.
ID= 4 ITER= 3 PARAM 2=***** FUNC= 2141.	ID= 8 ITER= 4 PARAM 2= 5.082 FUNC= 970.1
ID= 4 ITER= 4 PARAM 2= 8.976 FUNC= 1922.	<u>ID= 8 ITER= 5 PARAM 2= 4.495 FUNC= 963.6</u>
<u>ID= 4 ITER= 5 PARAM 2= 6.177 FUNC= 1884.</u>	ID= 8 ITER= 6 PARAM 2= 3.197 FUNC= 1031.
	ID= 8 ITER= 7 PARAM 2= 4.469 FUNC= 964.3
ID= 5 ITER= 1 PARAM 2= 5.140 FUNC= 1340.	
ID= 5 ITER= 2 PARAM 2= 5.243 FUNC= 1319.	ID= 9 ITER= 1 PARAM 2= 8.730 FUNC= 873.1
ID= 5 ITER= 3 PARAM 2=10.106 FUNC= 1420.	ID= 9 ITER= 2 PARAM 2= 8.905 FUNC= 872.6
ID= 5 ITER= 4 PARAM 2= 5.340 FUNC= 1299.	ID= 9 ITER= 3 PARAM 2=***** FUNC= 886.9
ID= 5 ITER= 5 PARAM 2= 5.437 FUNC= 1276.	ID= 9 ITER= 4 PARAM 2=13.249 FUNC= 866.1
ID= 5 ITER= 6 PARAM 2= 5.515 FUNC= 1270.	ID= 9 ITER= 5 PARAM 2=21.967 FUNC= 867.8
ID= 5 ITER= 7 PARAM 2= 5.784 FUNC= 1245.	ID= 9 ITER= 6 PARAM 2=13.423 FUNC= 866.1
ID= 5 ITER= 8 PARAM 2= 5.983 FUNC= 1239.	ID= 9 ITER= 7 PARAM 2=14.037 FUNC= 866.0
ID= 5 ITER= 9 PARAM 2= 6.587 FUNC= 1267.	ID= 9 ITER= 8 PARAM 2=20.982 FUNC= 867.4
<u>ID= 5 ITER=10 PARAM 2= 5.995 FUNC= 1238.</u>	<u>ID= 9 ITER= 9 PARAM 2=14.176 FUNC= 866.0</u>

Appendix 5: Interactive Computer Session

ID=10 ITER= 1 PARAM 2= 3.790 FUNC= 5828.
ID=10 ITER= 2 PARAM 2= 3.866 FUNC= 5882.
ID=10 ITER= 3 PARAM 2= .000 FUNC= .2078E+06
ID=10 ITER= 4 PARAM 2= 3.740 FUNC= 5782.
ID=10 ITER= 5 PARAM 2= 3.631 FUNC= 5737.
ID=10 ITER= 6 PARAM 2= 3.422 FUNC= 6205.
ID=10 ITER= 7 PARAM 2= 3.627 FUNC= 5733.

ID=11 ITER= 1 PARAM 2= 6.000 FUNC= 2421.
ID=11 ITER= 2 PARAM 2= 6.120 FUNC= 2435.
ID=11 ITER= 3 PARAM 2= .000 FUNC= .2925E+06
ID=11 ITER= 4 PARAM 2= 5.802 FUNC= 2381.
ID=11 ITER= 5 PARAM 2= 5.592 FUNC= 2358.
ID=11 ITER= 6 PARAM 2= 5.274 FUNC= 2315.
ID=11 ITER= 7 PARAM 2= 5.014 FUNC= 2286.
ID=11 ITER= 8 PARAM 2= 4.708 FUNC= 2265.
ID=11 ITER= 9 PARAM 2= 4.213 FUNC= 2301.

ID=11 ITER=10 PARAM 2= 4.698 FUNC= 2263.
ID=12 ITER= 1 PARAM 2= 6.000 FUNC= .7808E+05
ID=12 ITER= 2 PARAM 2= 6.120 FUNC= .7821E+05
ID=12 ITER= 3 PARAM 2= .000 FUNC= .1366E+06
ID=12 ITER= 4 PARAM 2= 5.060 FUNC= .7674E+05
ID=12 ITER= 5 PARAM 2= 4.232 FUNC= .7482E+05
ID=12 ITER= 6 PARAM 2= 3.746 FUNC= .7300E+05
ID=12 ITER= 7 PARAM 2= 3.455 FUNC= .7136E+05
ID=12 ITER= 8 PARAM 2= 3.265 FUNC= .6984E+05
ID=12 ITER= 9 PARAM 2= 3.134 FUNC= .6860E+05
ID=12 ITER=10 PARAM 2= 3.025 FUNC= .6745E+05
ID=12 ITER=11 PARAM 2= 2.929 FUNC= .6641E+05
ID=12 ITER=12 PARAM 2= 2.837 FUNC= .6548E+05
ID=12 ITER=13 PARAM 2= 2.741 FUNC= .6452E+05
ID=12 ITER=14 PARAM 2= 2.643 FUNC= .6366E+05
ID=12 ITER=15 PARAM 2= 2.535 FUNC= .6298E+05
ID=12 ITER=16 PARAM 2= 2.384 FUNC= .6339E+05

ID=13 ITER= 1 PARAM 2= 5.120 FUNC= 2032.
ID=13 ITER= 2 PARAM 2= 5.222 FUNC= 2013.
ID=13 ITER= 3 PARAM 2=13.222 FUNC= 1818.
ID=13 ITER= 4 PARAM 2=69.075 FUNC= 1966.
ID=13 ITER= 5 PARAM 2=14.339 FUNC= 1827.
ID=13 ITER= 6 PARAM 2=13.245 FUNC= 1819.

END OF FILE

??stop

Forecasting Run

85/11/21. 14.09.53. T14
(0)SOUTH FLA WATER MGMT-DIST S/N 675. -- NOS-2.3 617/587.
USER NAME: XXXXXXXX
JSN: ABZK, NAMIAF
CHARGE NUMBER:XXXXXXXXXXXX

bat
/get,kroute/un=afan
/ftn5,i=kroute,l=0
7.232 CP SECONDS COMPILATION TIME.
lgo
ROUTING OPTION? (1=FORECAST,2=SIMULATION,3=CALIBRATION)
? 1
ENTER FIRST DATE (MM,DD,YY)
? 8,1,85
ENTER 5 MULTIPLICATION RATIOS FOR RAINFALL
? 1.25,1.00,0.75,0.50,0.25
ENTER ANTECEDENT 1 AND 6-MONTH RAINFALL
? 7.11,23.54
ENTER INITIAL STAGES FROM LAKE ALLIGATOR TO KISSIMMEE
? 62.31,60.5,60.09,61.12,56.26,53.44,50.60,50.60,50.60

Fractions of normal rainfall

Enter stages in downstream order
from Lakes Alligator to Kissimmee

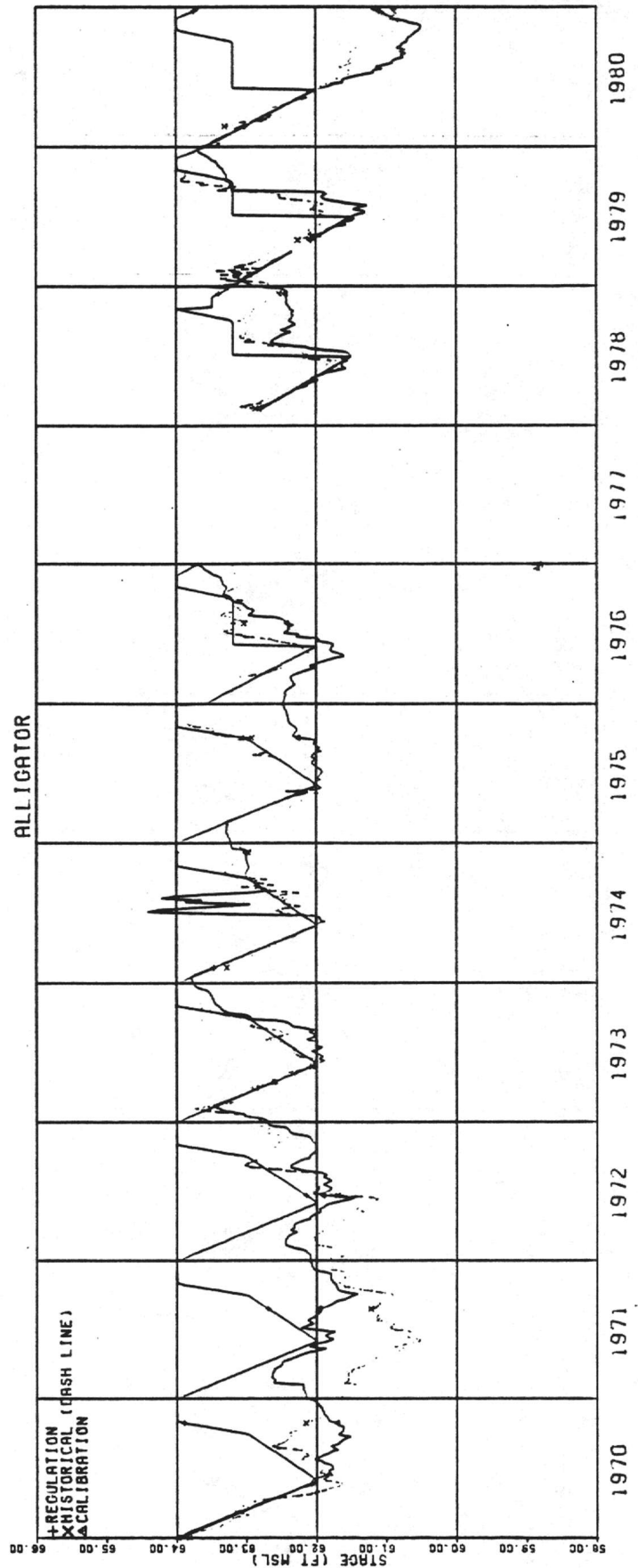
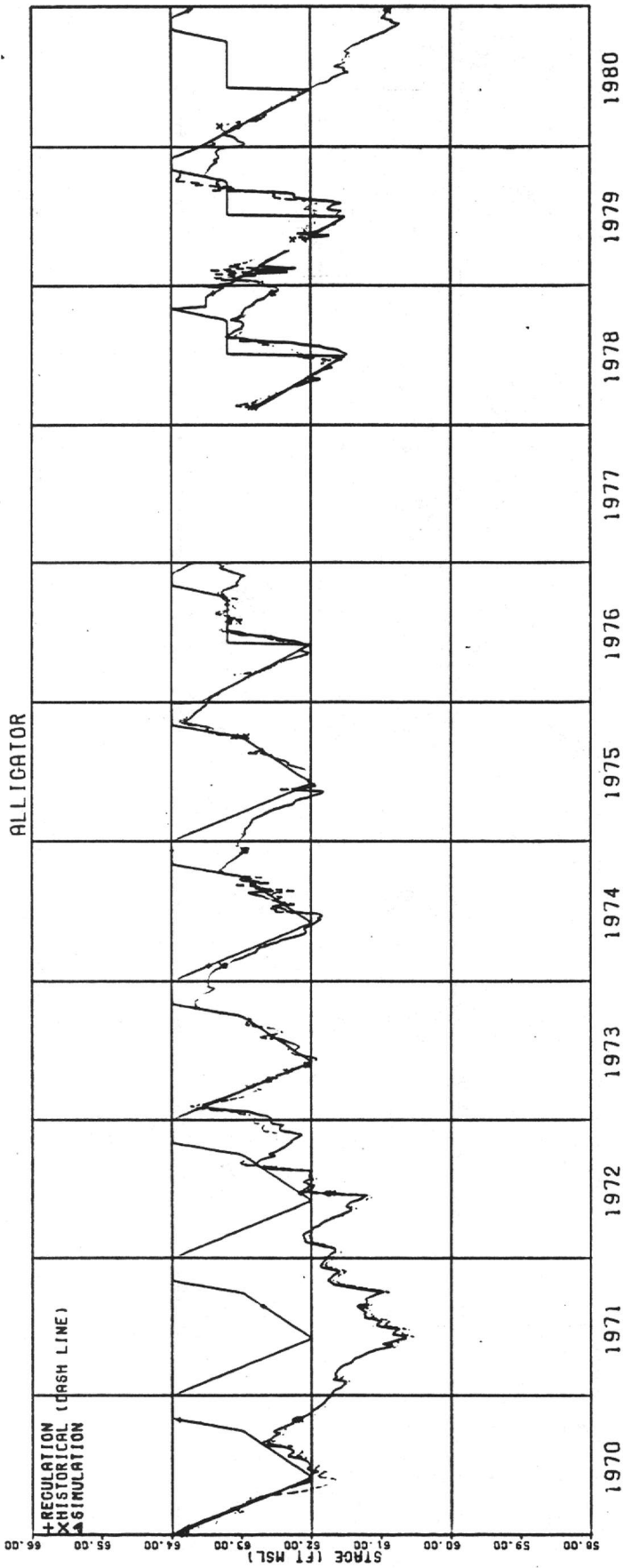
TIME LIMIT
ENTER T TO CONTINUE OR CR KEY TO STOP:
t,*
STOP
80.750 CP SECONDS EXECUTION TIME.

Set up plotting run

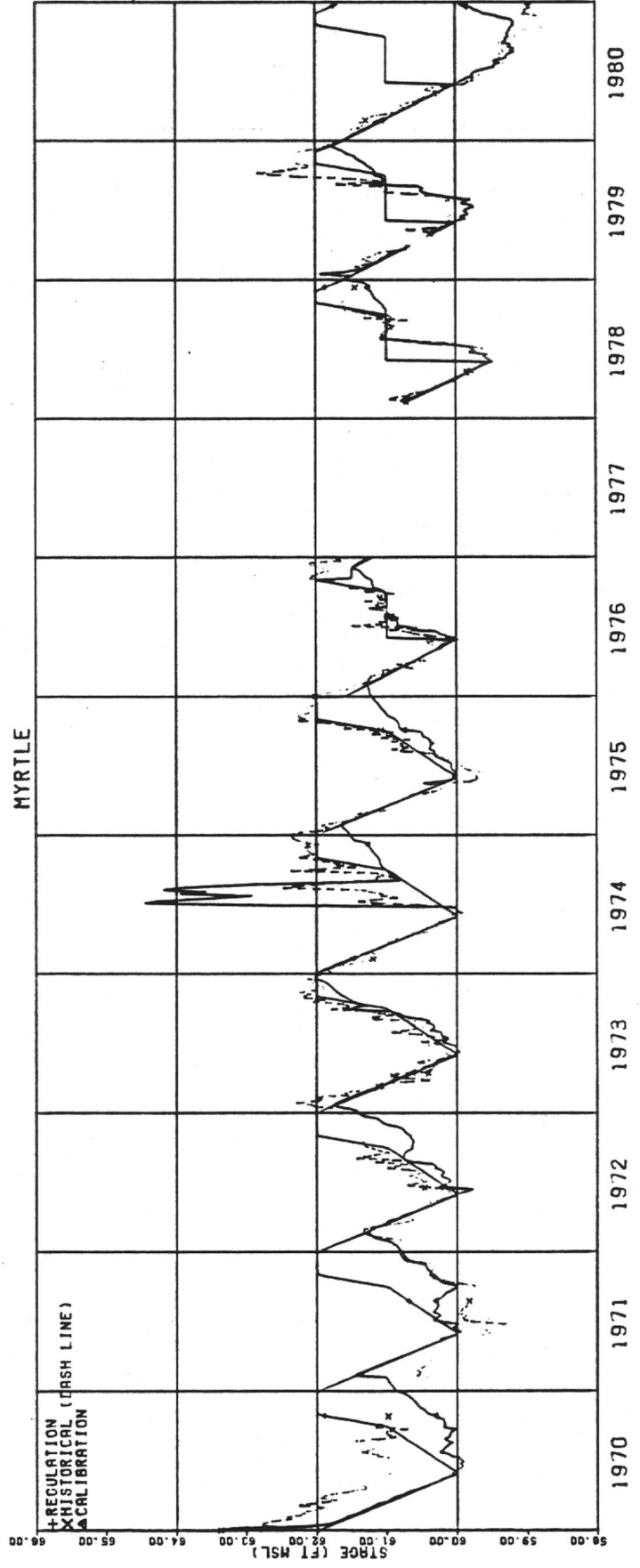
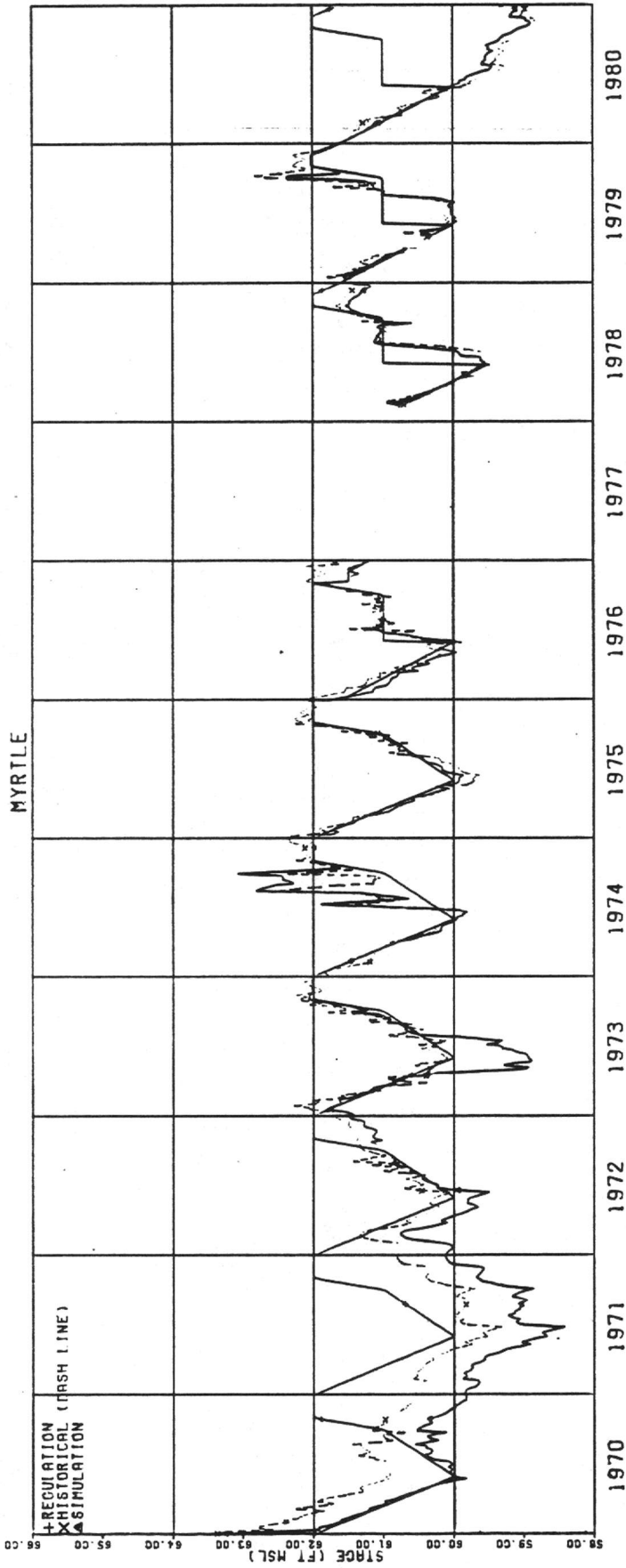
/attach,calcomp/un=library
/library,calcomp
LIBRARY,CALCOMP.
/get,kplot/un=afan
/ftn5,i=kplot,l=0,b=bplot
3.058 CP SECONDS COMPILATION TIME.
/bplot
PLOTING OPTION: 1=STAGE,2=FLOW,3=BOTH,4=FORECAST
? 4
STOP
15.320 CP SECONDS EXECUTION TIME.
get.procfil/un=afan
/begin.runplot
REVERT.COPY TAPE13 TO PLOT12 COMPLETED

Send plots to Calcomp plotter
Plots are shown in Pages 97 to 106

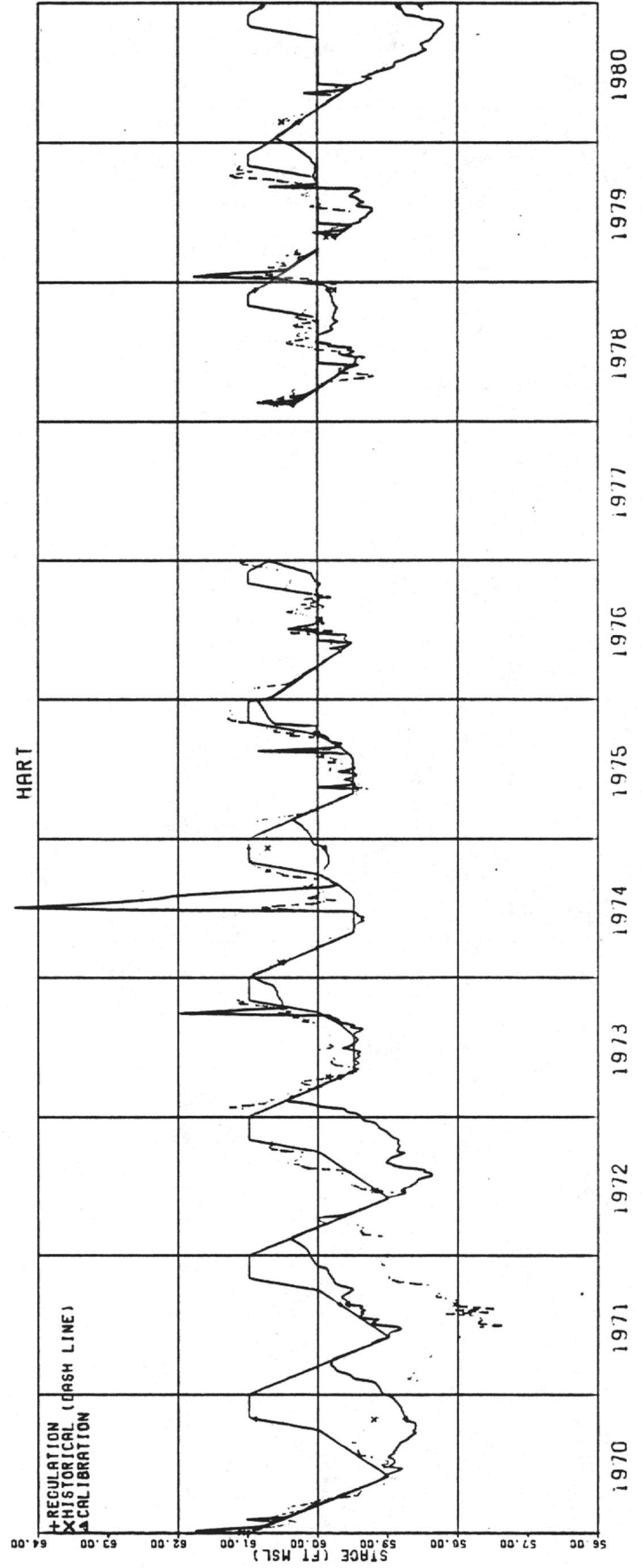
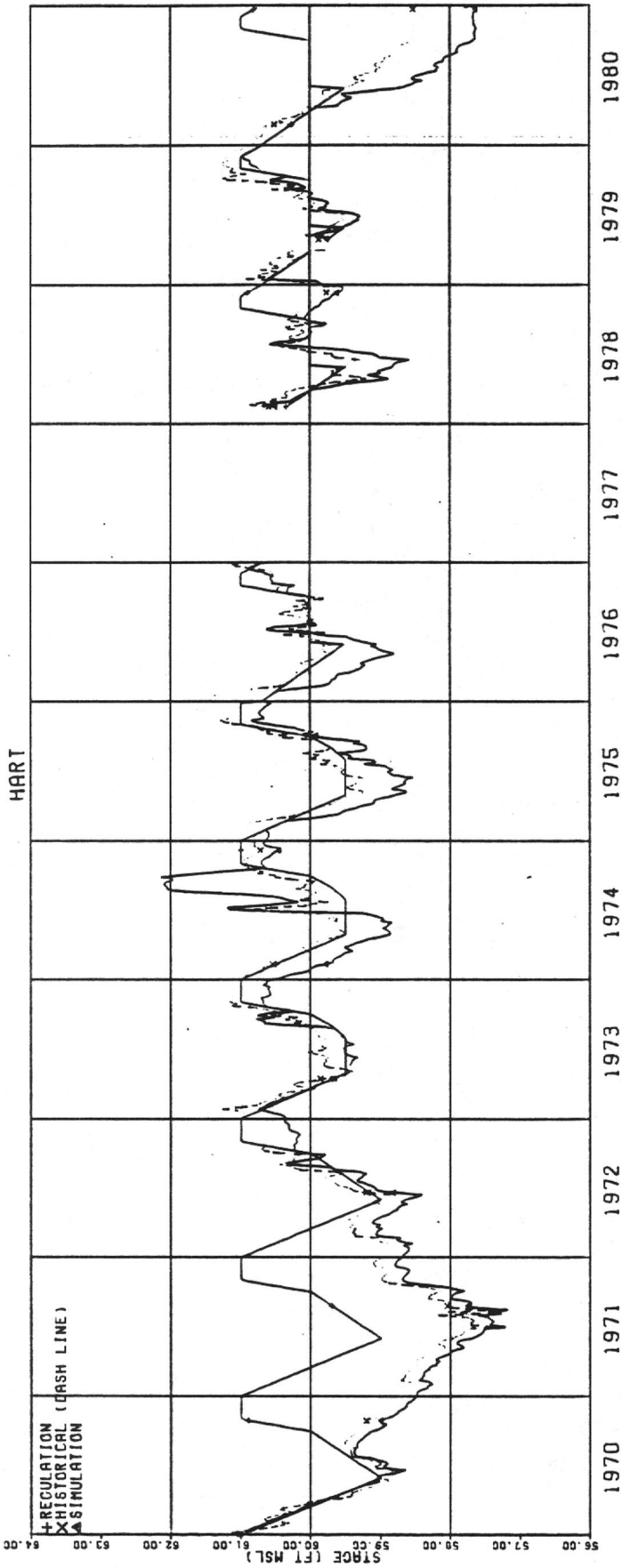
Appendix 6: Simulation and Calibration Results (1970-80)



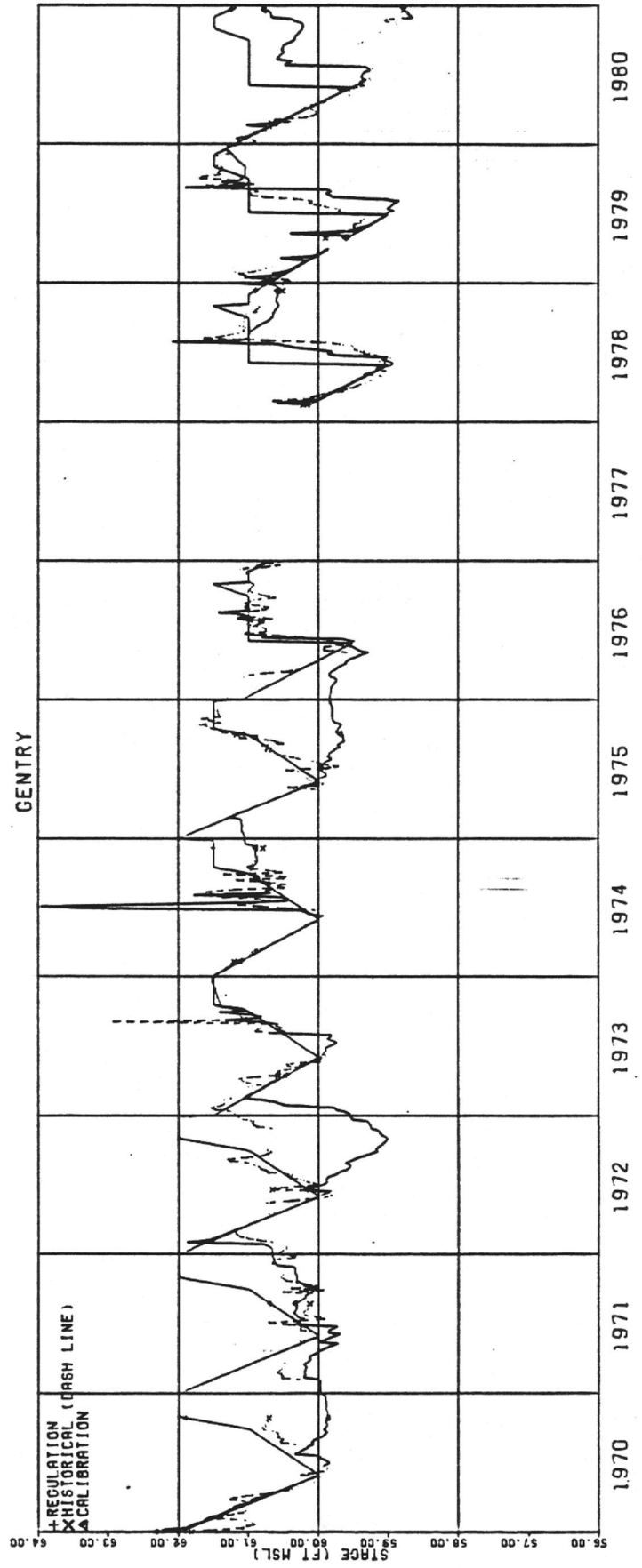
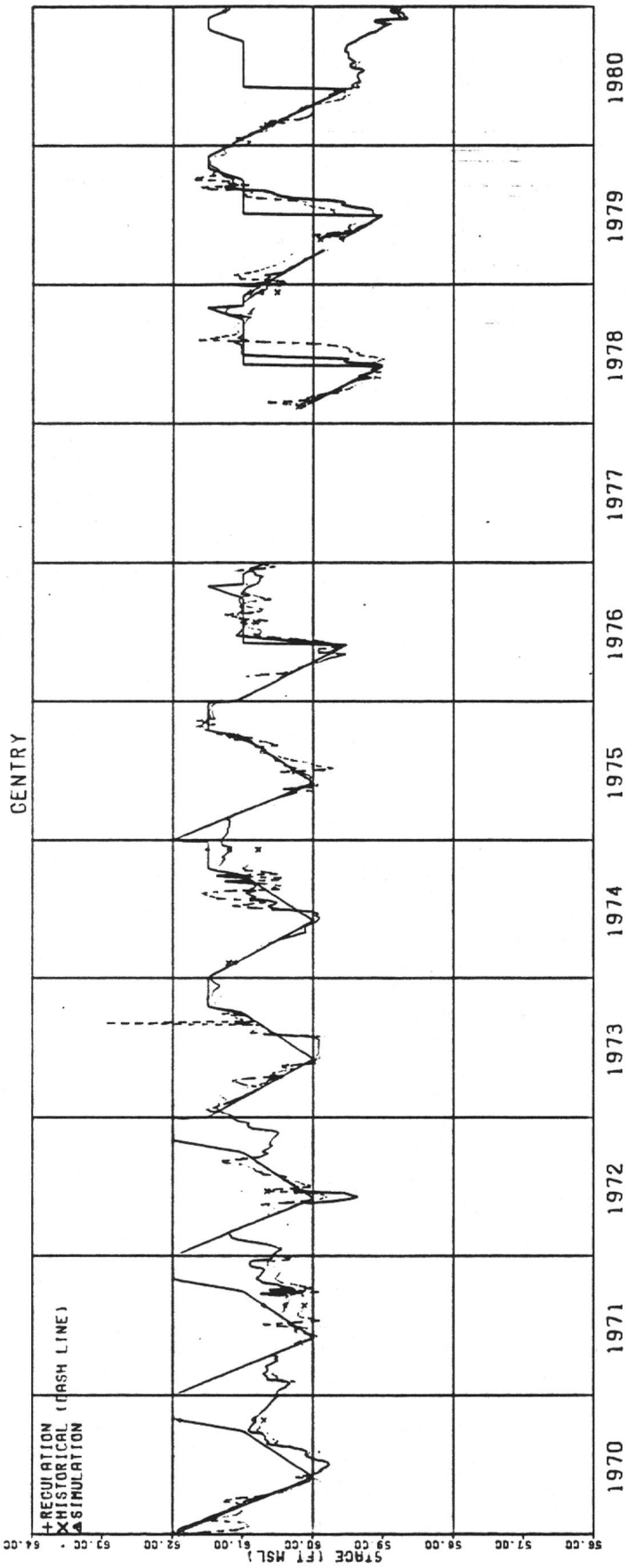
Appendix 6: Simulation and Calibration Results (1970-80)



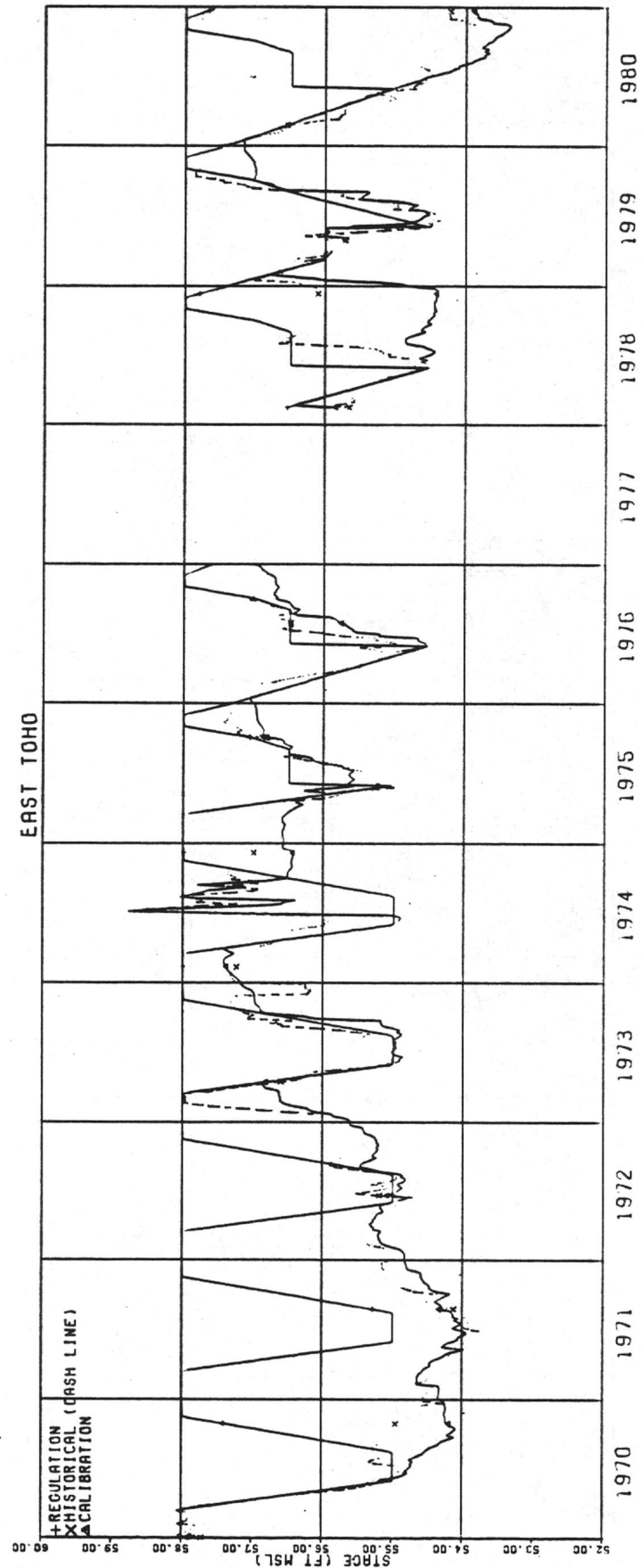
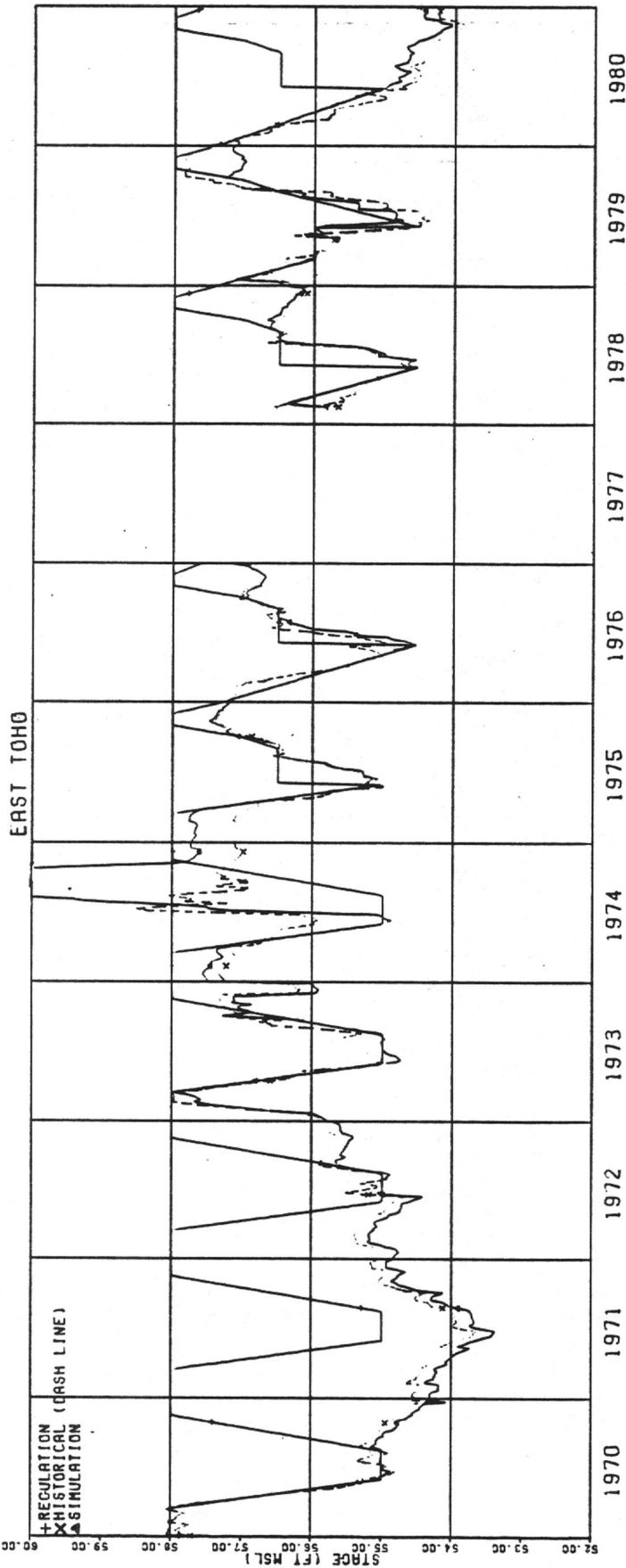
Appendix 6: Simulation and Calibration Results (1970-80)



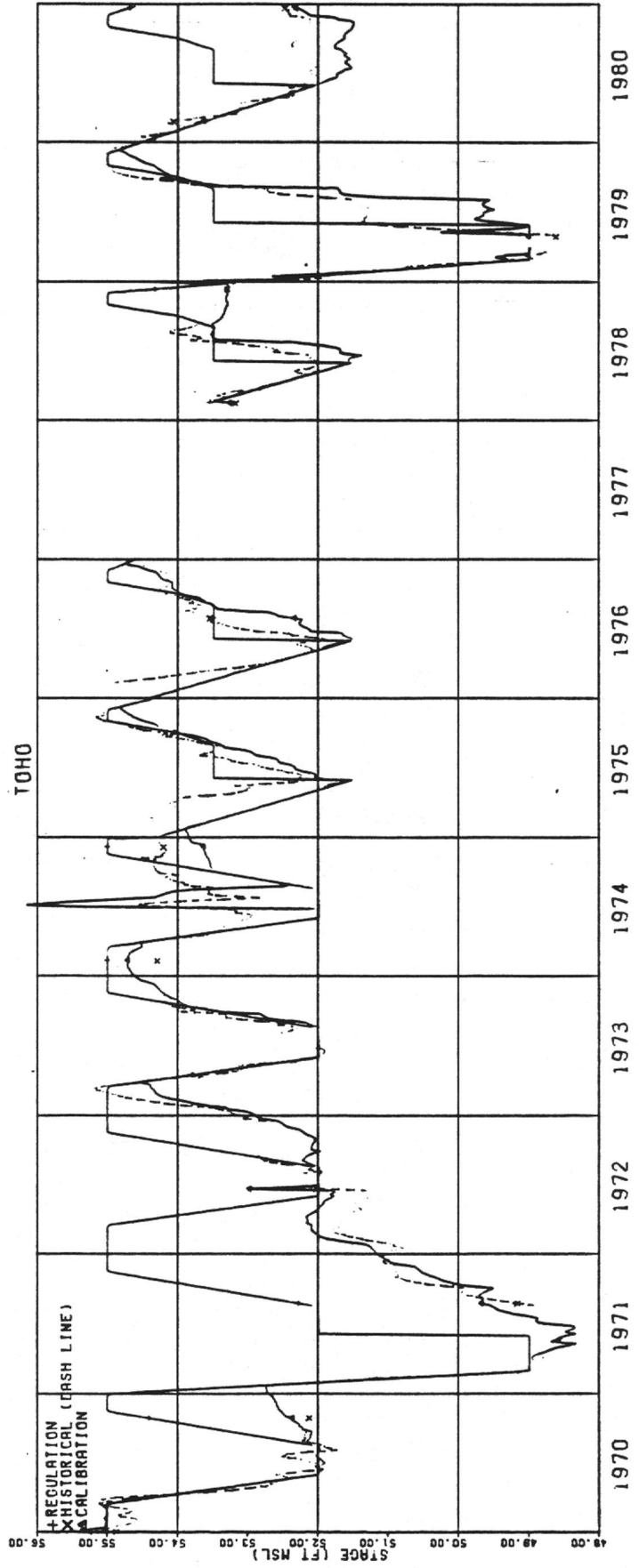
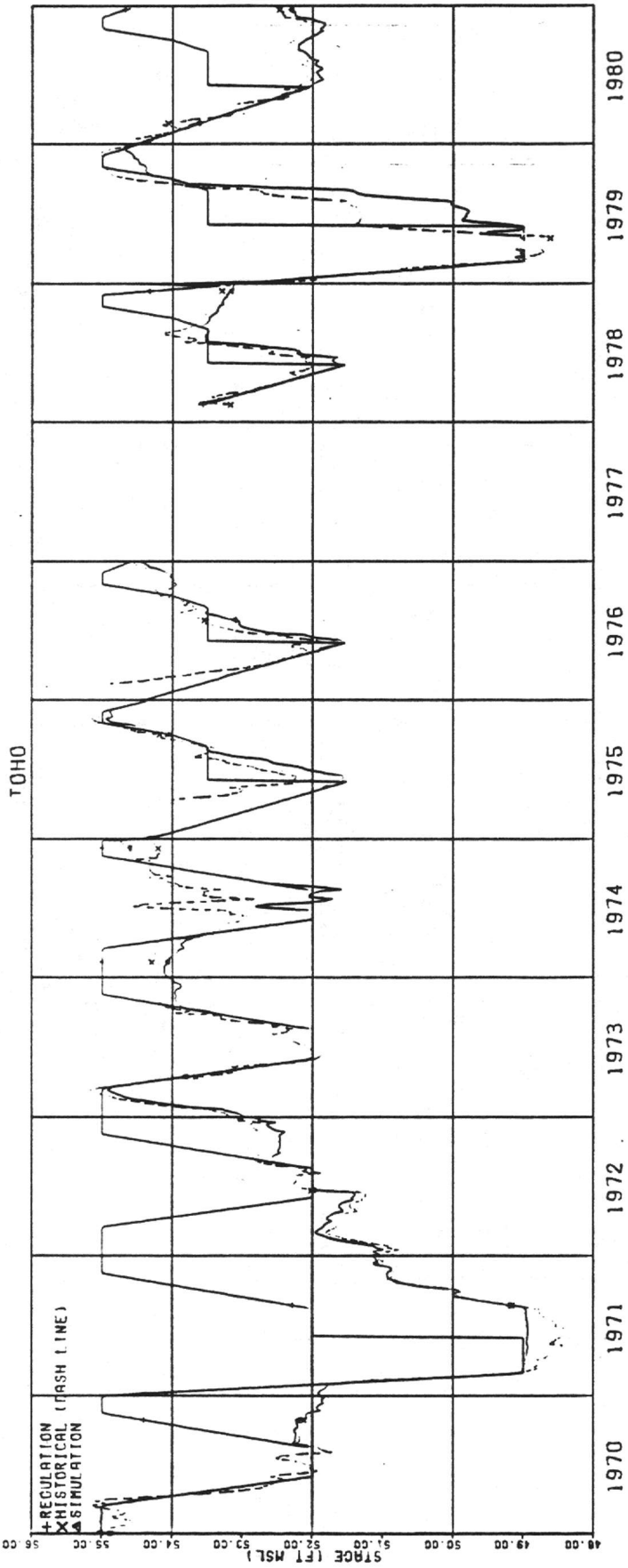
Appendix 6: Simulation and Calibration Results (1970-80)



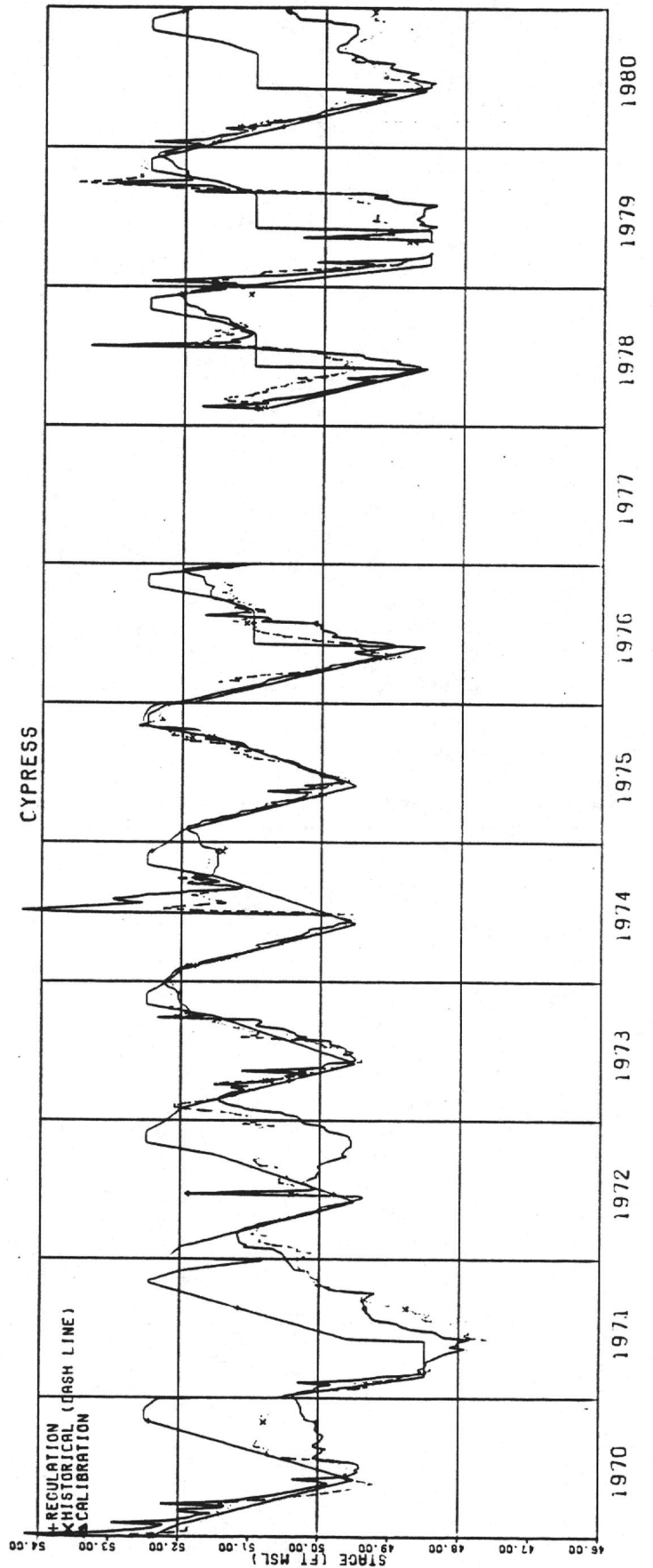
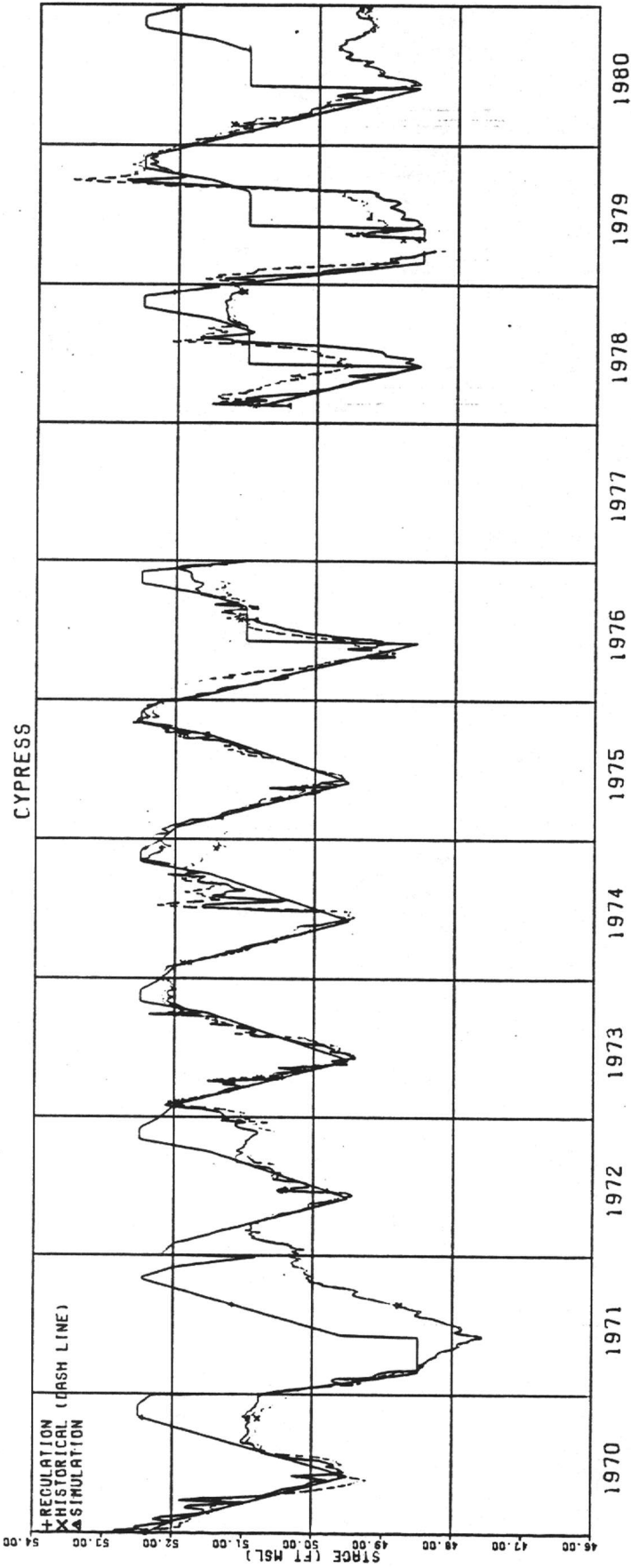
Appendix 6: Simulation and Calibration Results (1970-80)



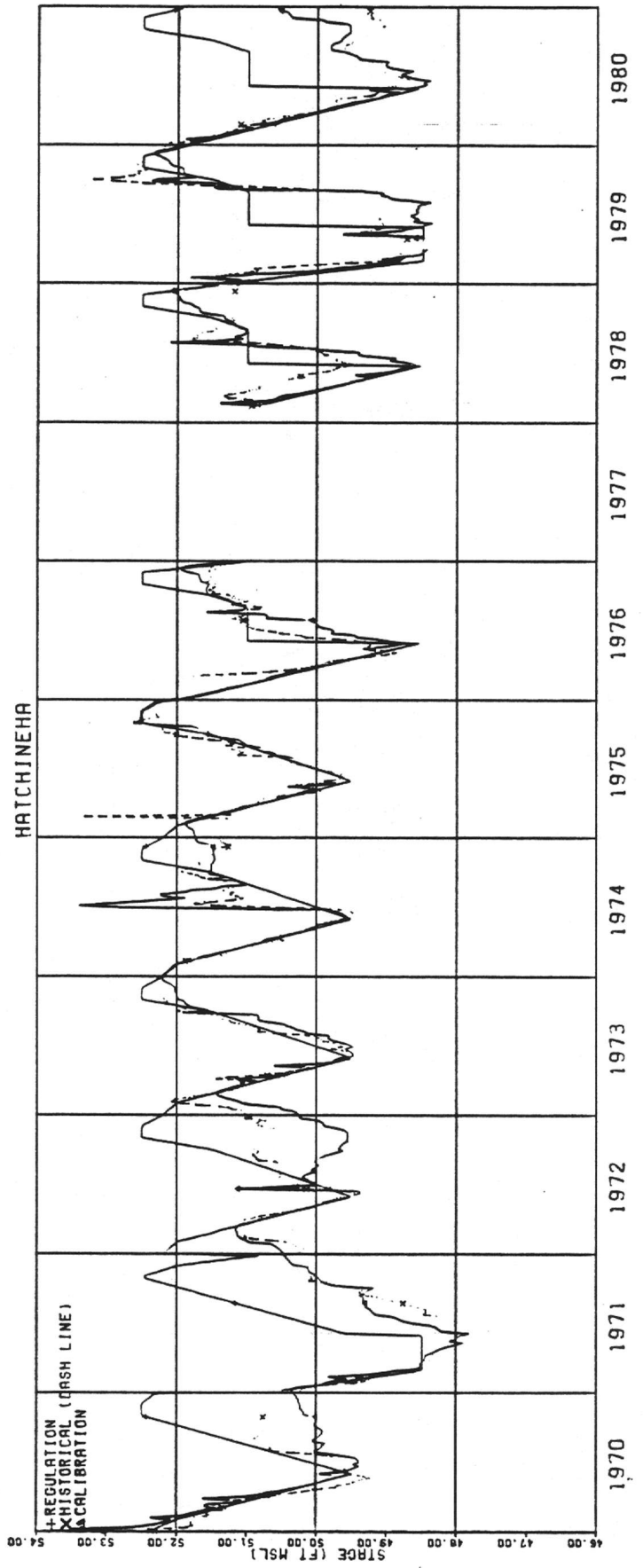
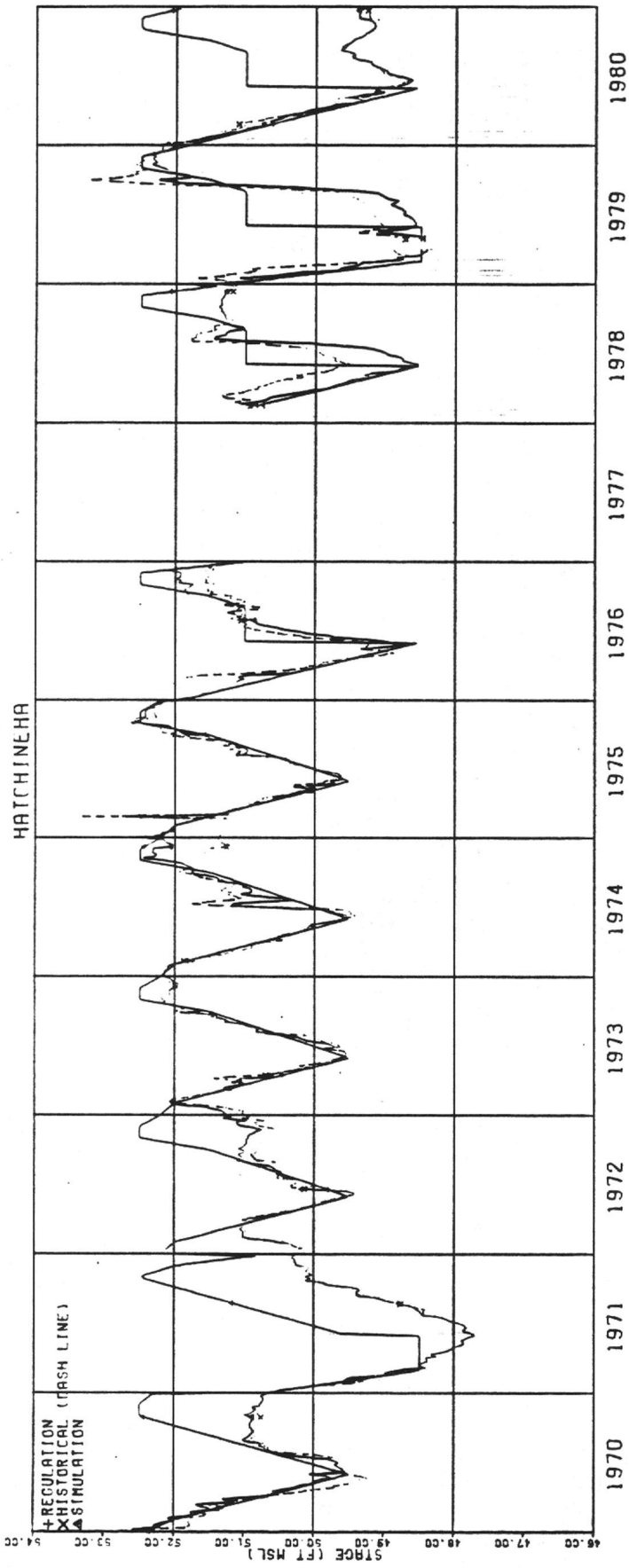
Appendix 6: Simulation and Calibration Results (1970-80)



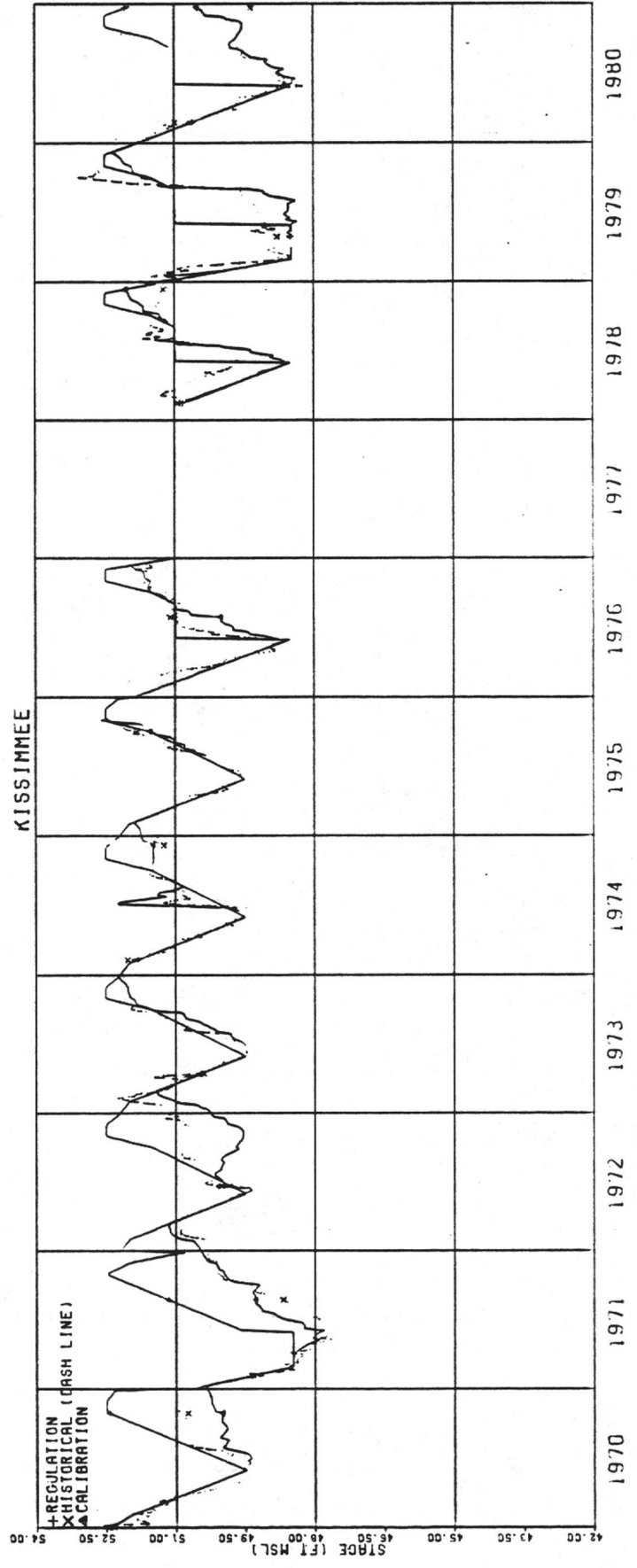
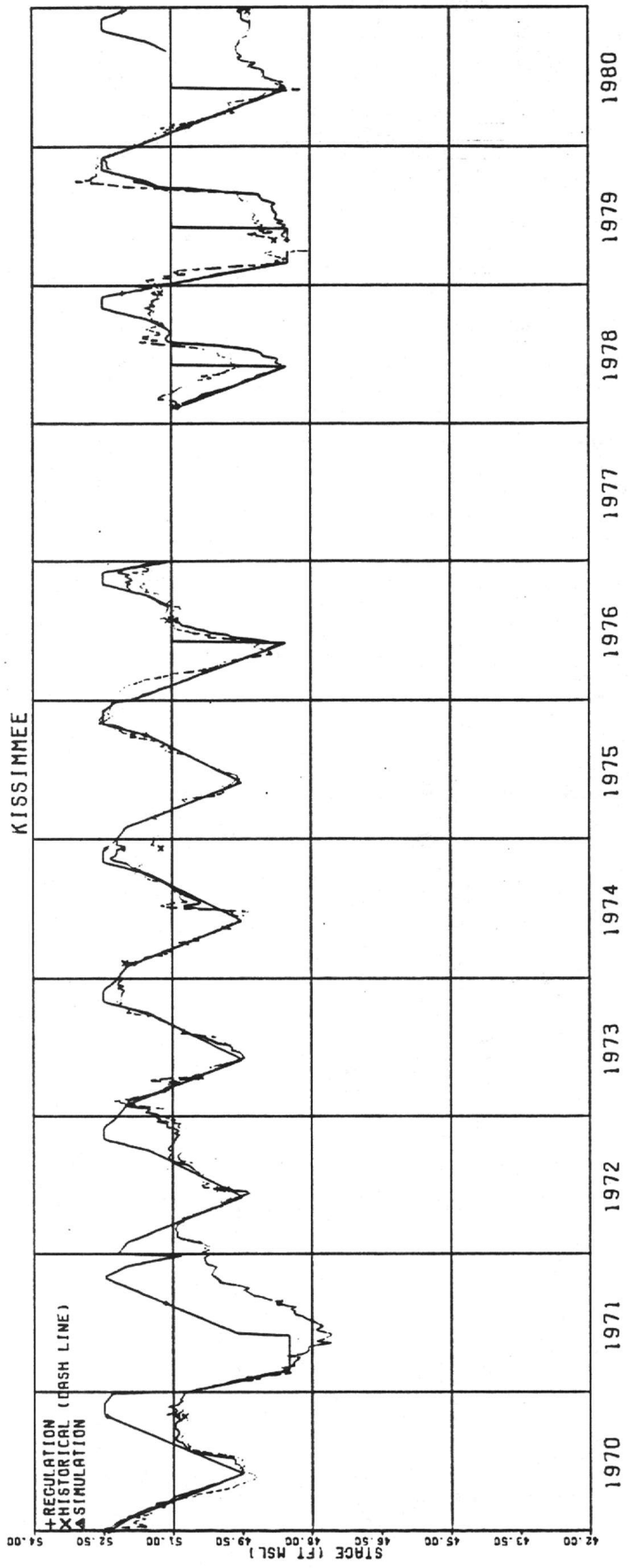
Appendix 6: Simulation and Calibration Results (1970-80)



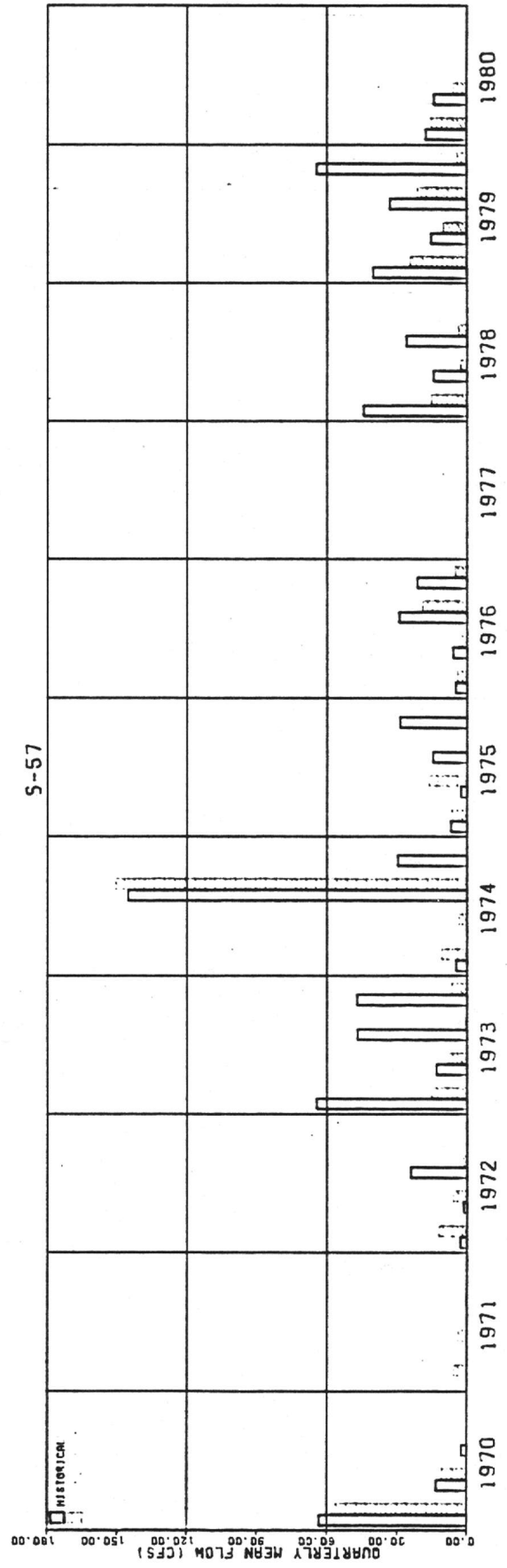
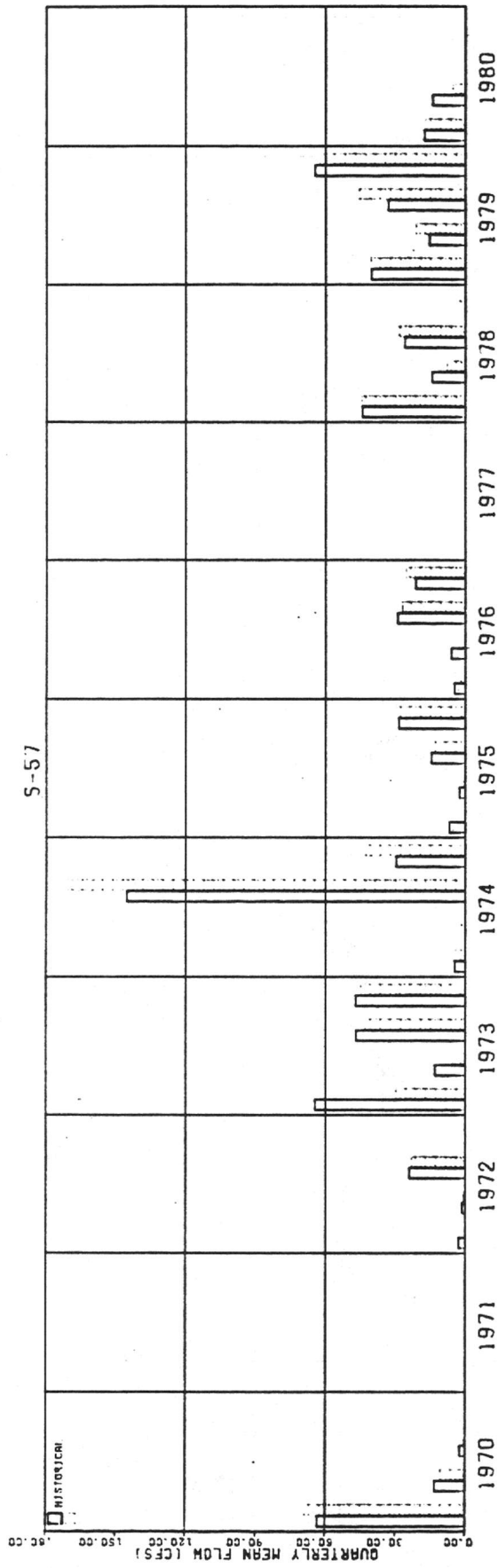
Appendix 6: Simulation and Calibration Results (1970-80)



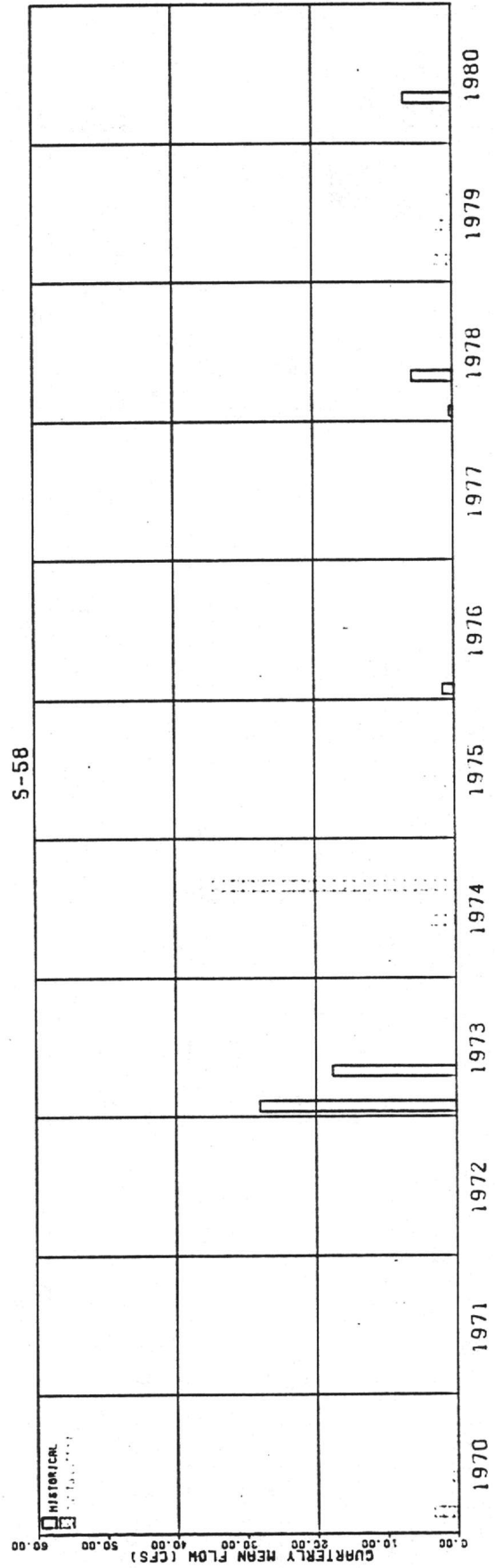
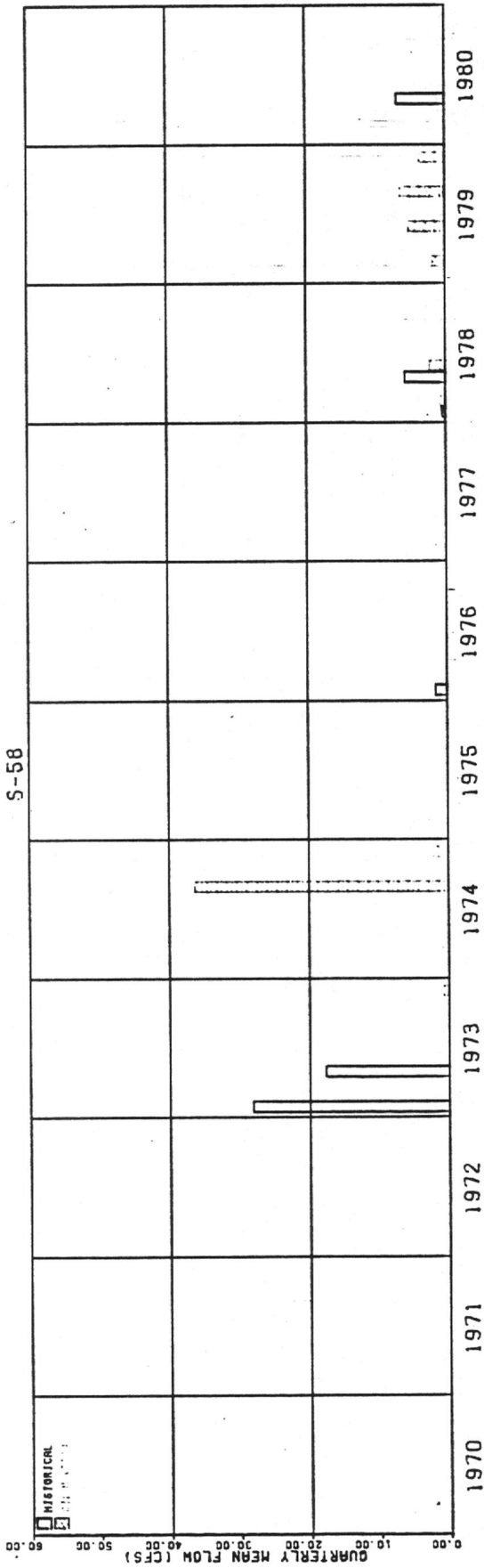
Appendix 6: Simulation and Calibration Results (1970-80)



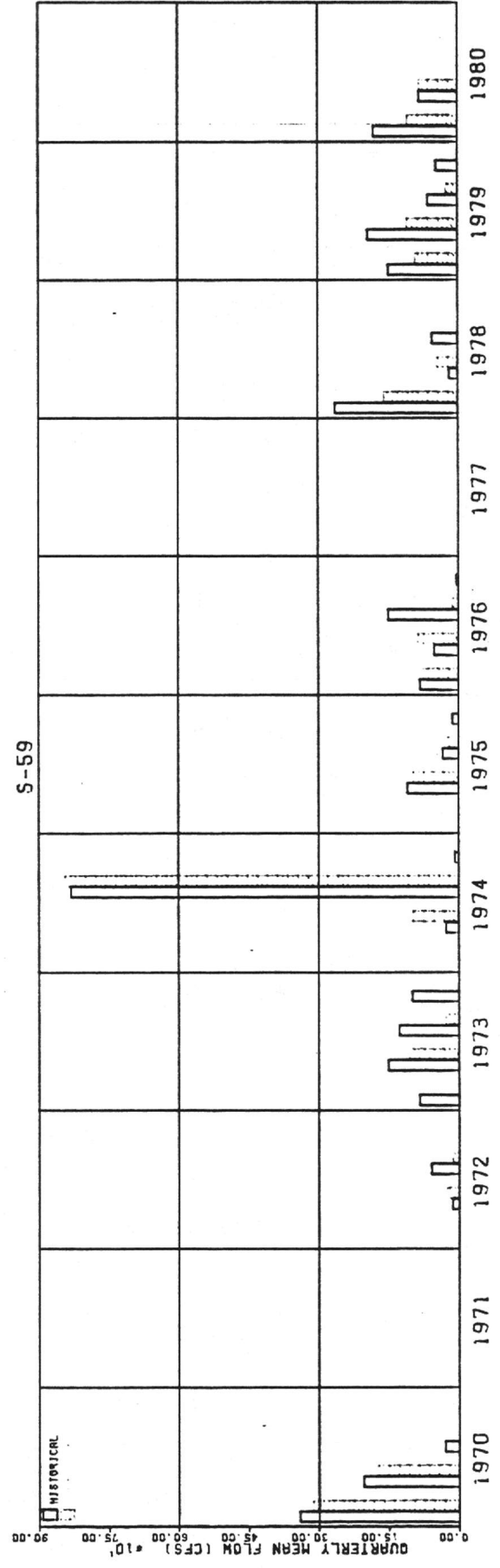
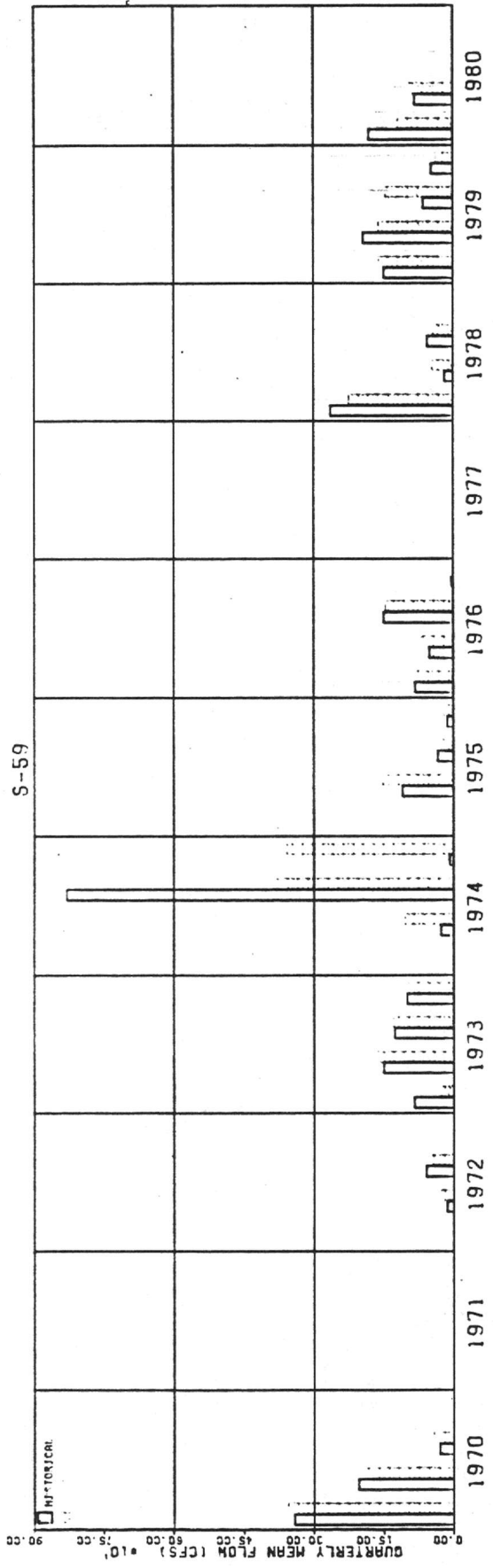
Appendix 6: Simulation and Calibration Results (1970-80)



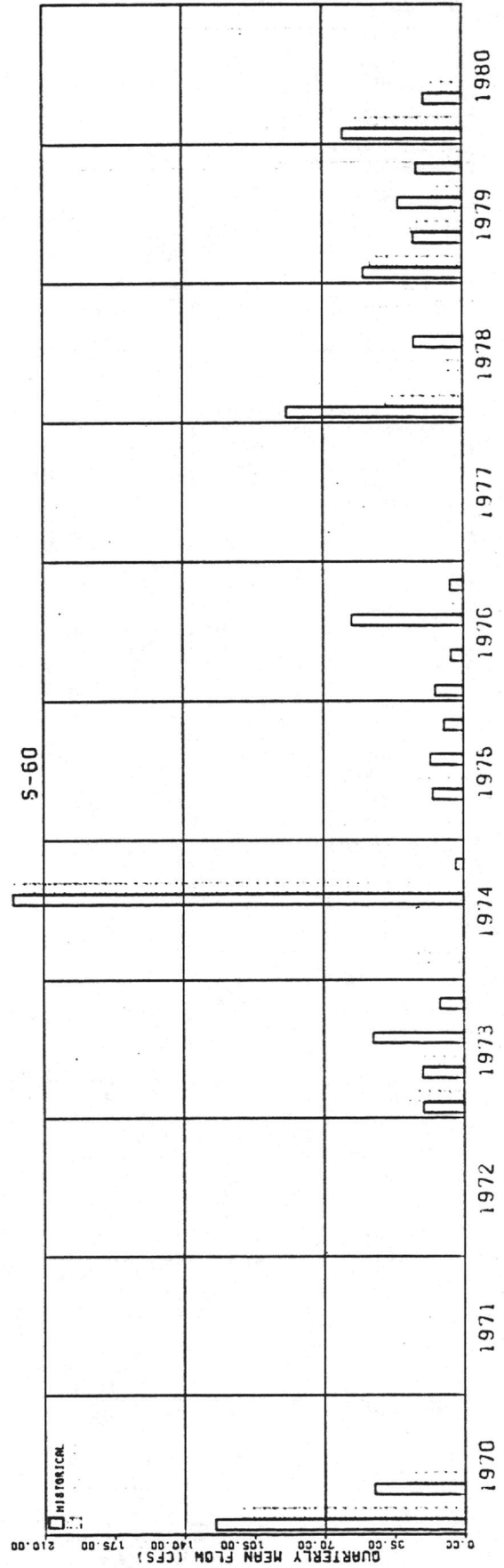
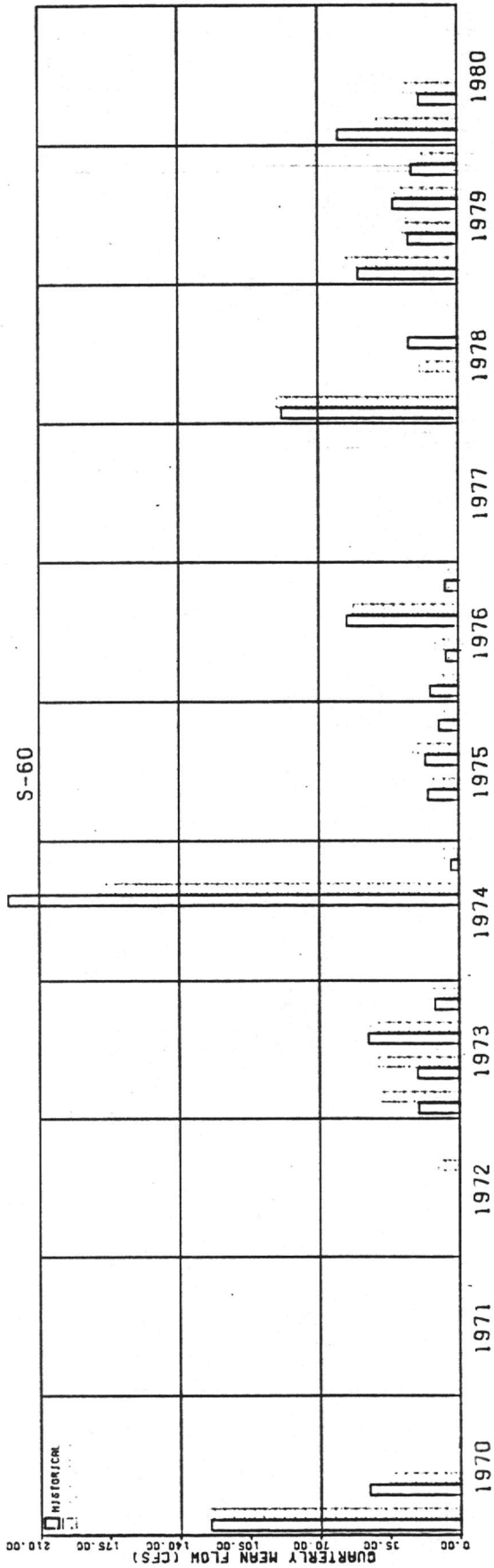
Appendix 6: Simulation and Calibration Results (1970-80)



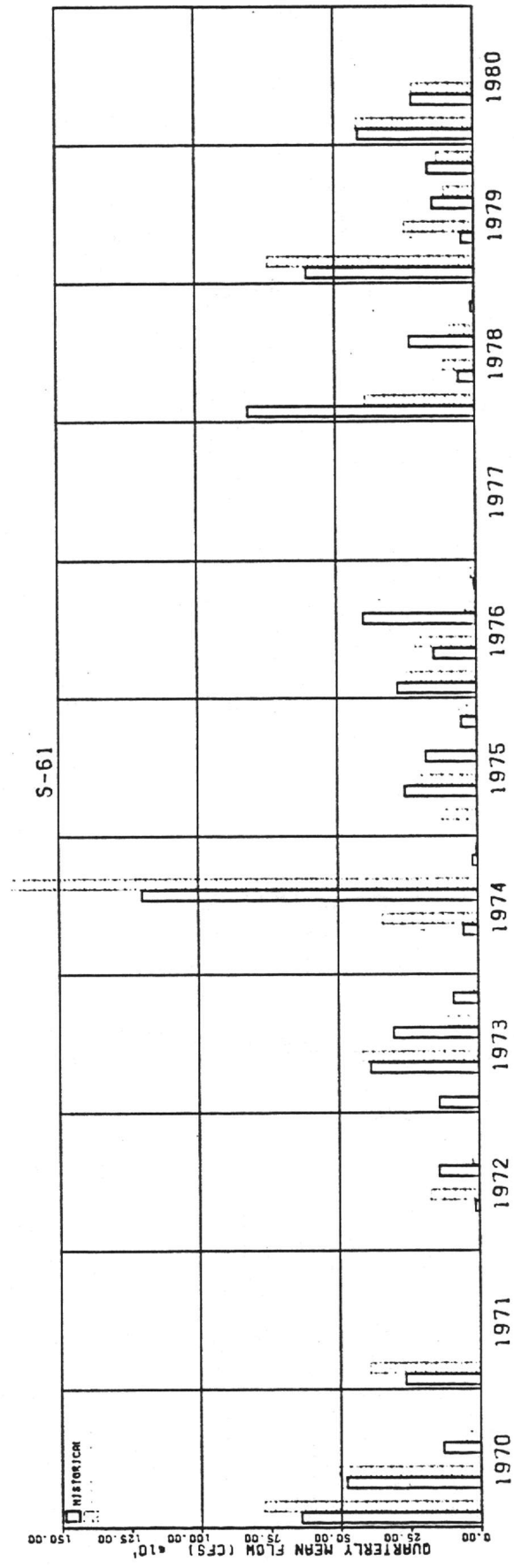
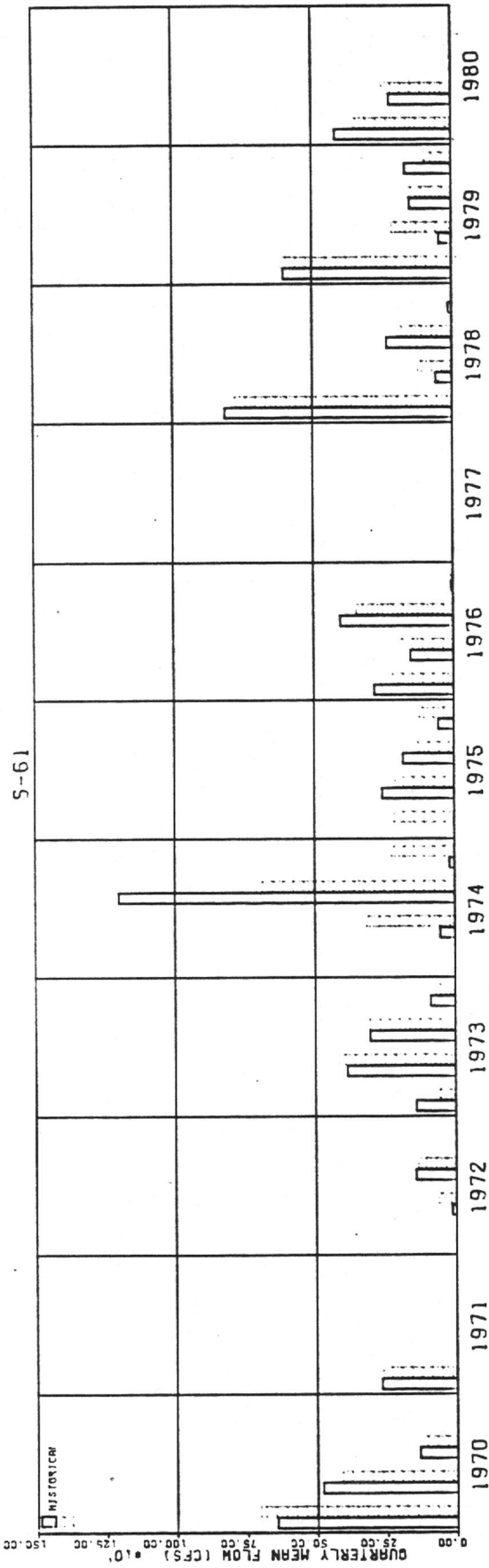
Appendix 6: Simulation and Calibration Results (1970-80)



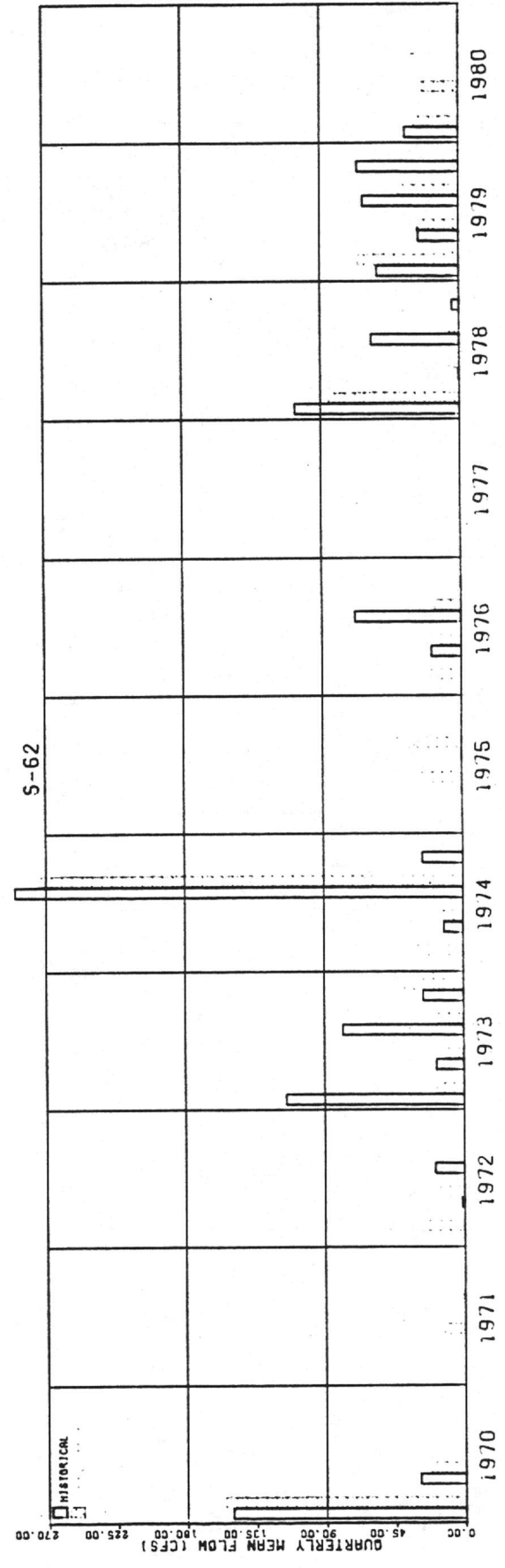
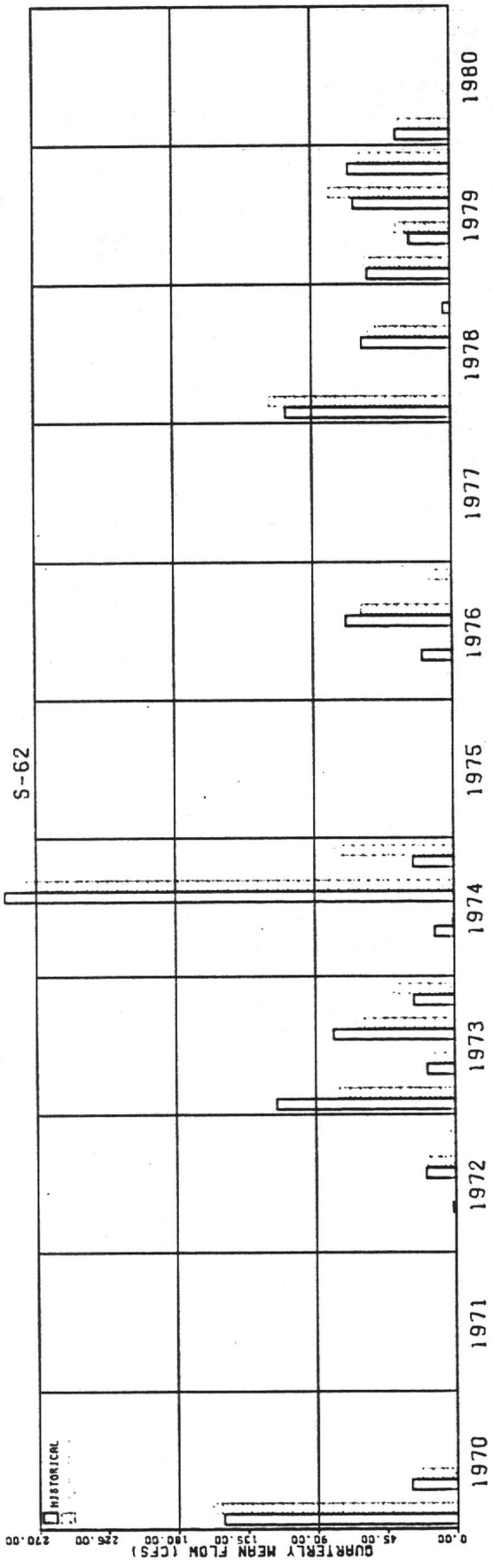
Appendix 6: Simulation and Calibration Results (1970-80)



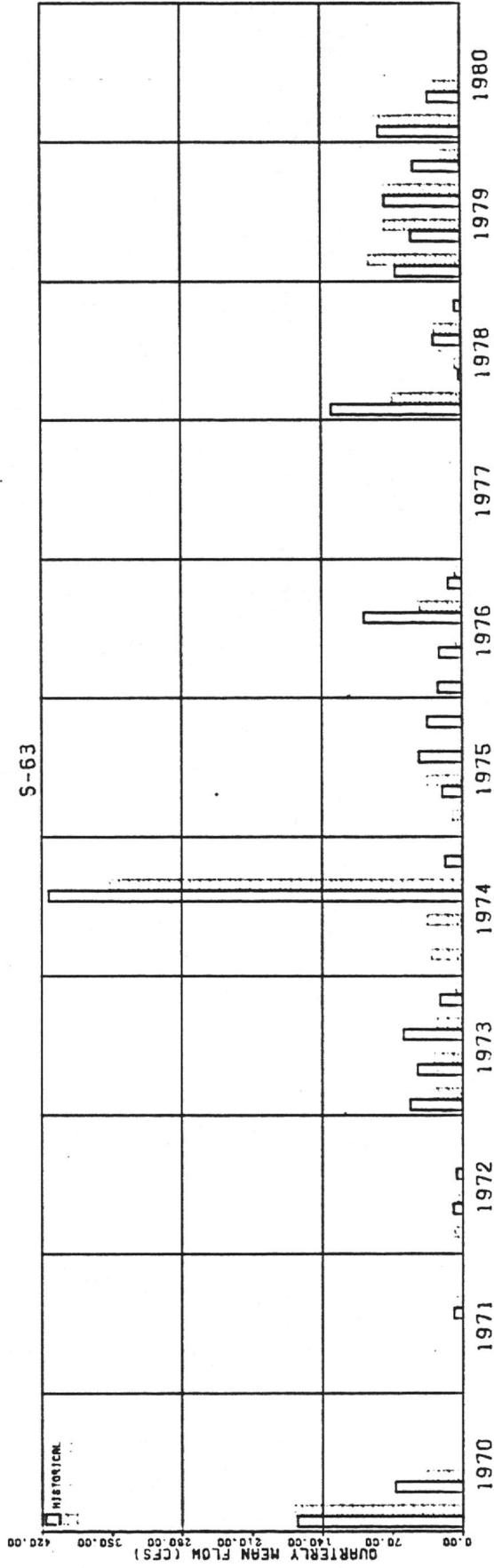
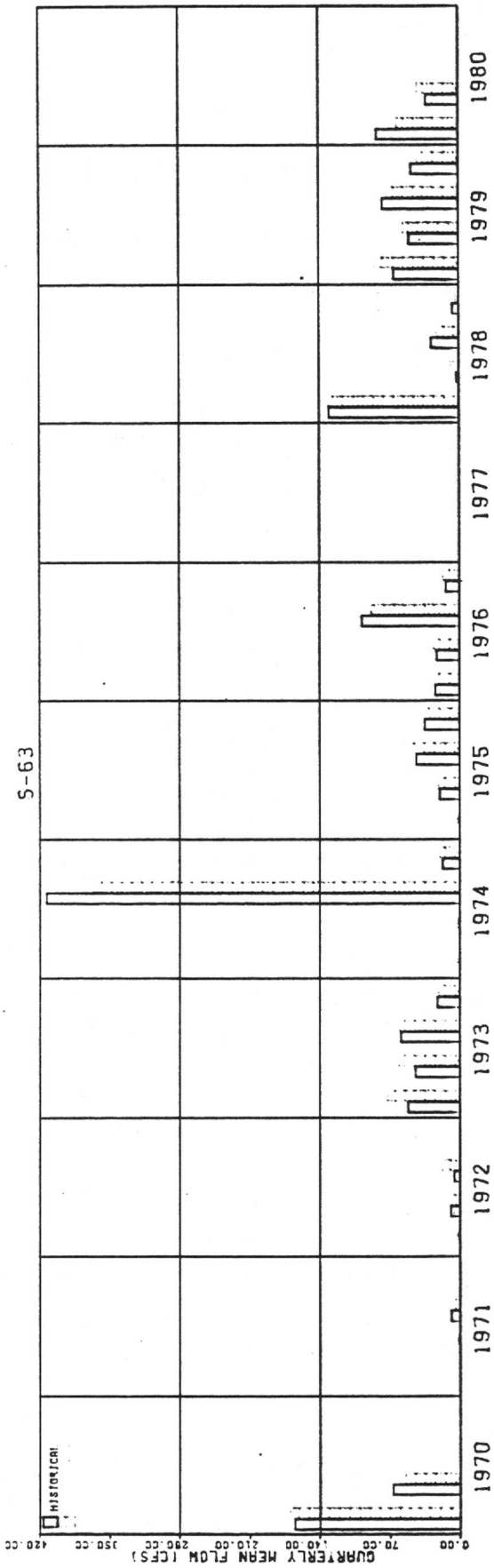
Appendix 6: Simulation and Calibration Results (1970-80)



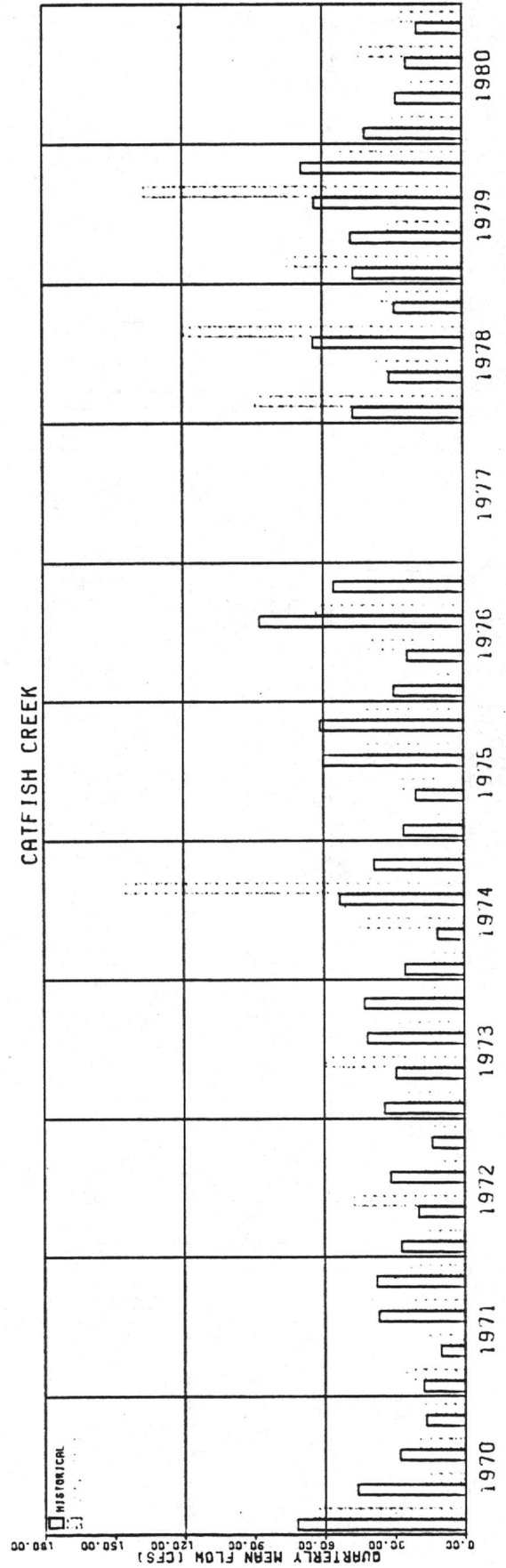
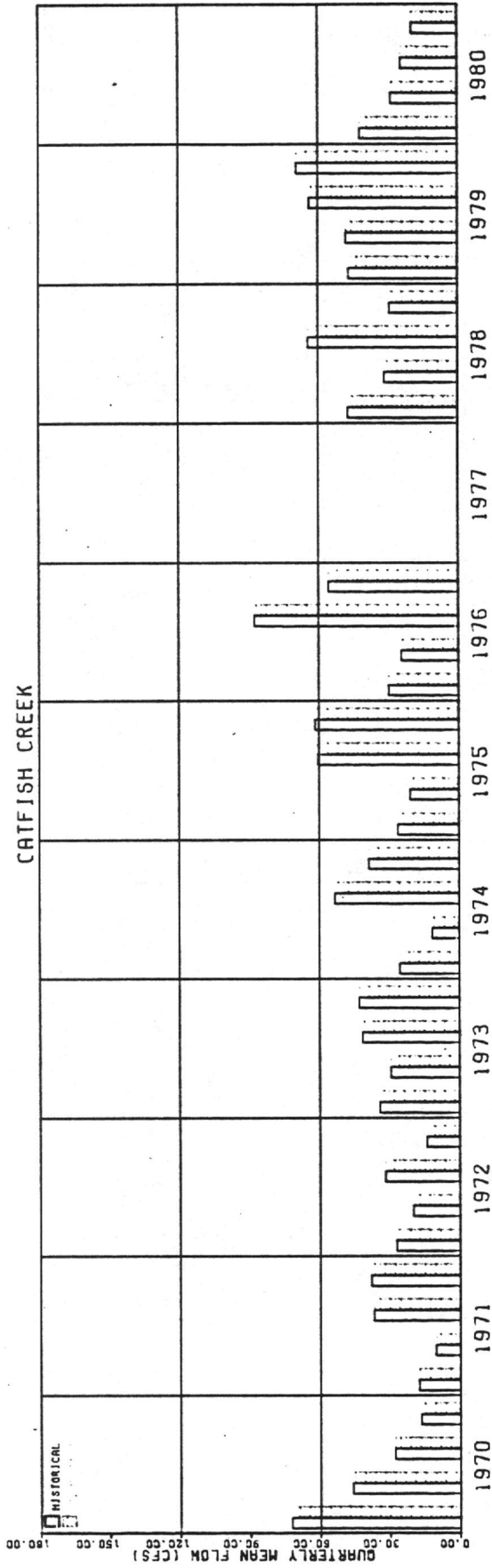
Appendix 6: Simulation and Calibration Results (1970-80)



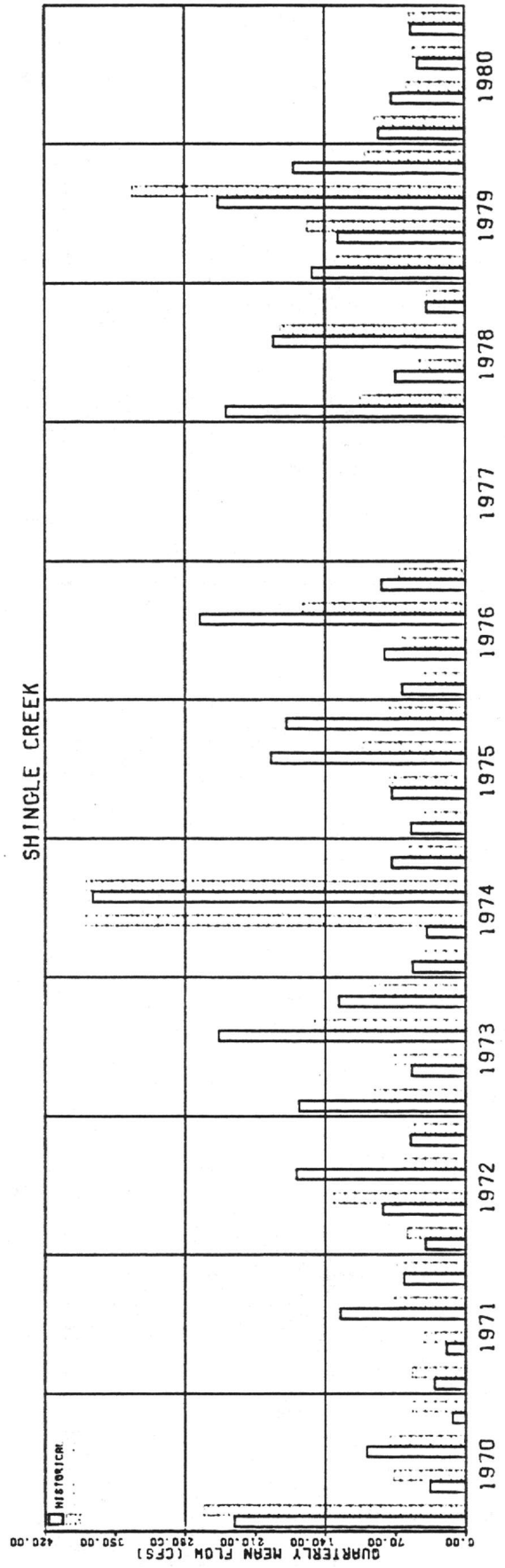
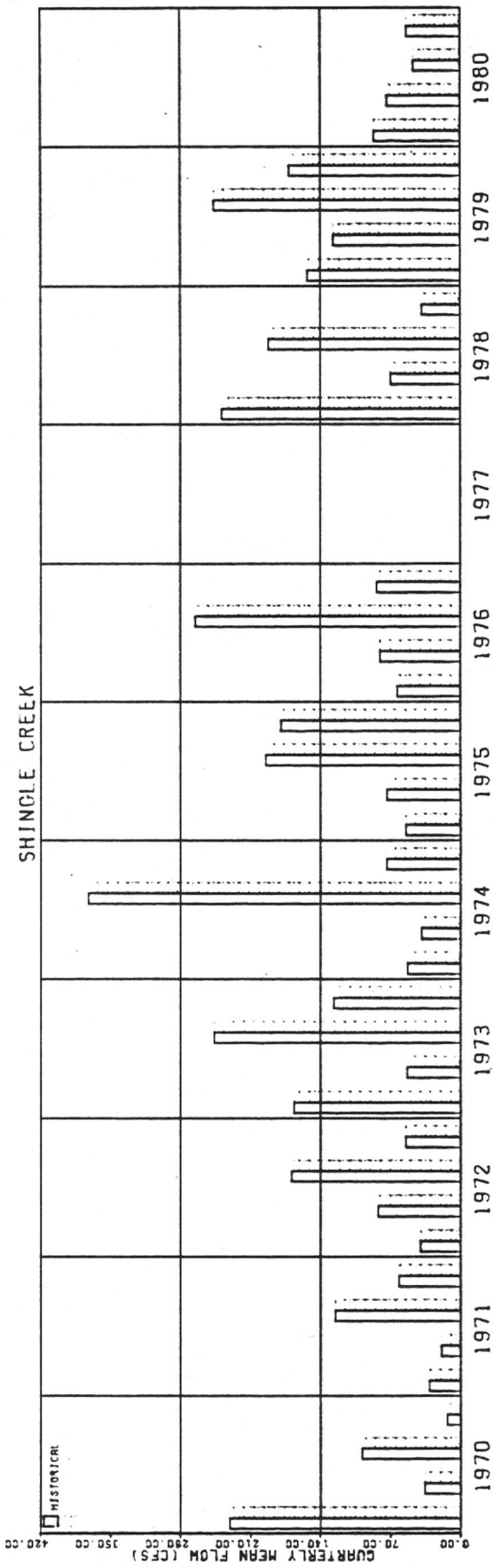
Appendix 6: Simulation and Calibration Results (1970-80)



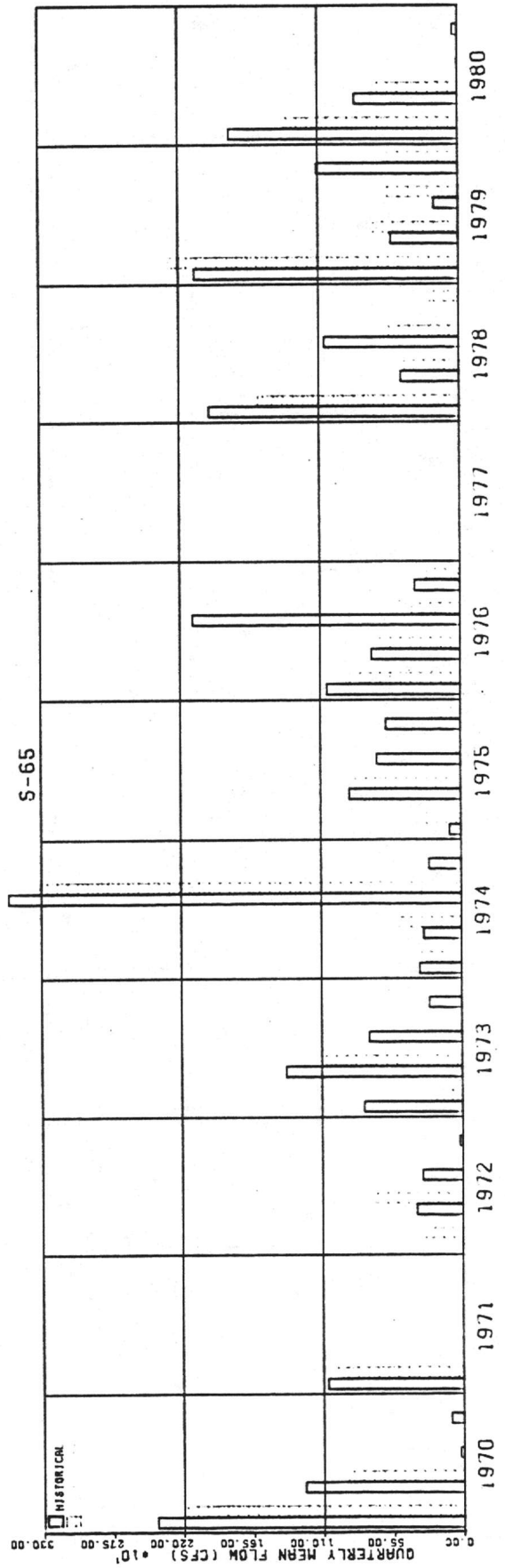
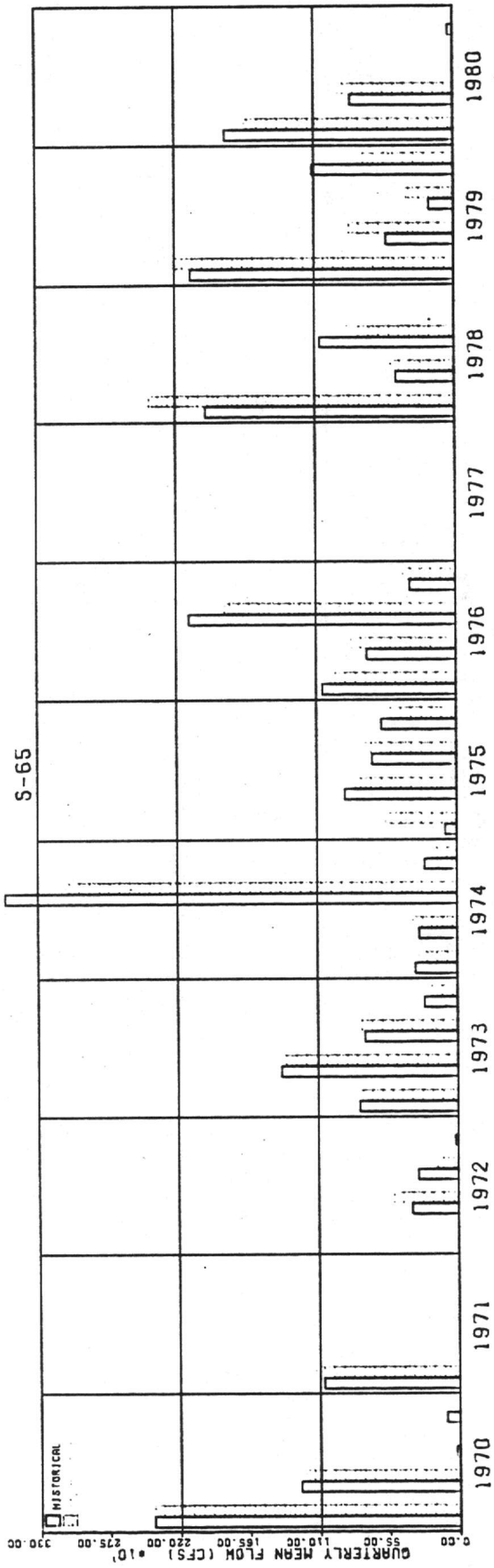
Appendix 6: Simulation and Calibration Results (1970-80)



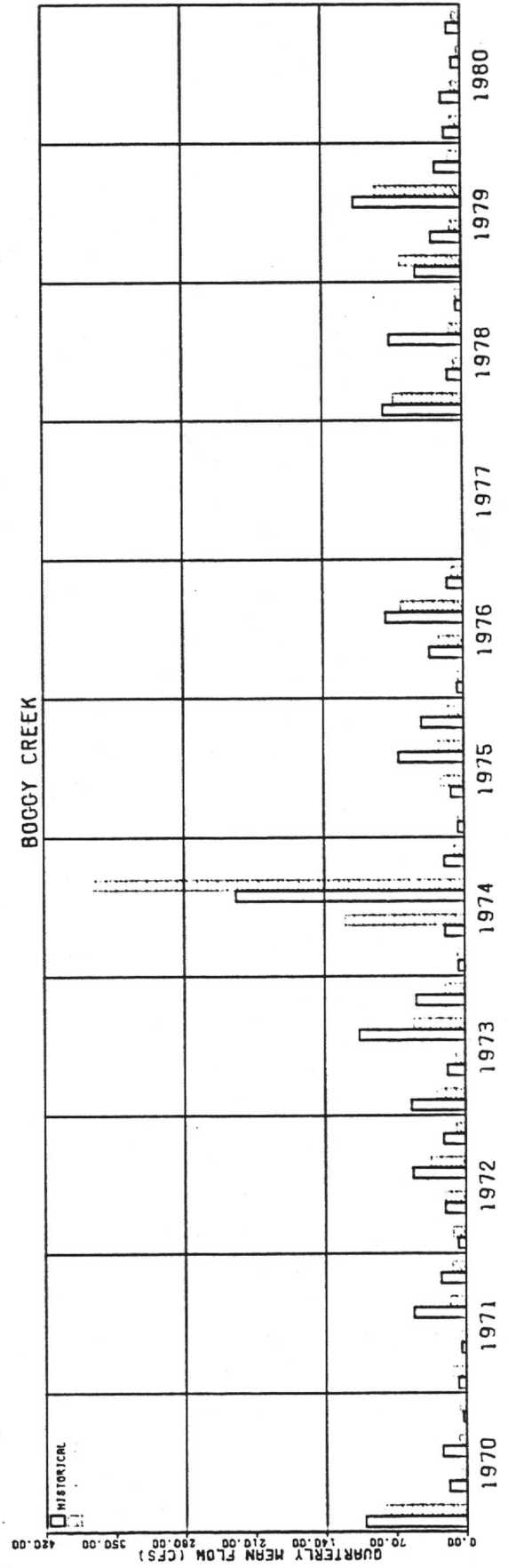
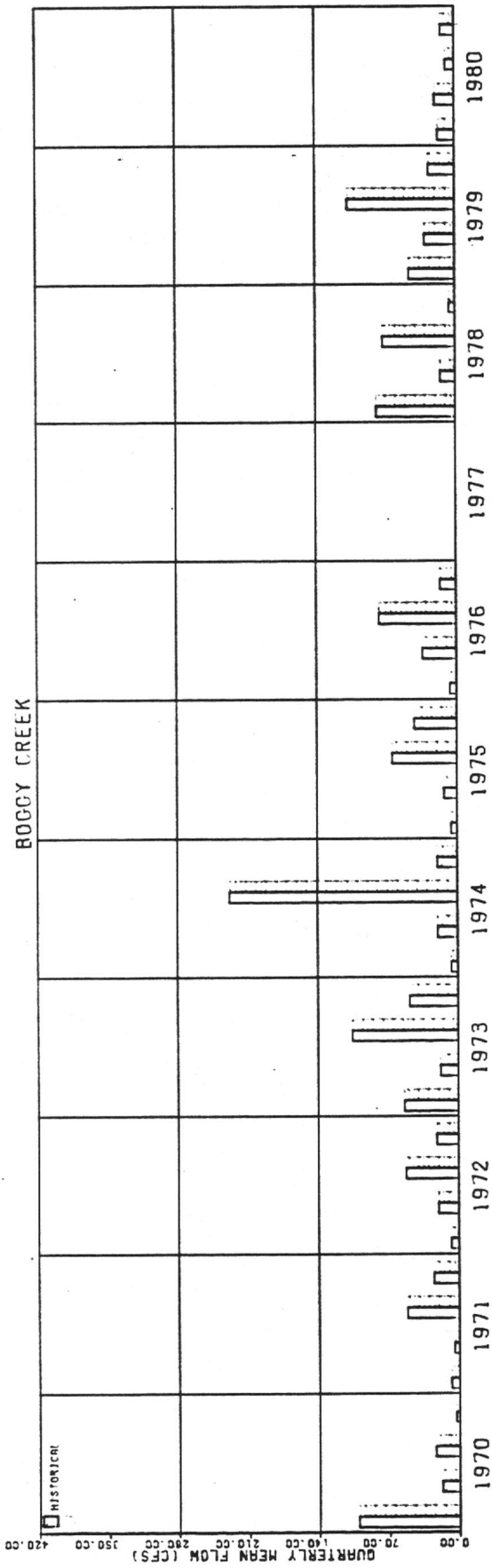
Appendix 6: Simulation and Calibration Results (1970-80)



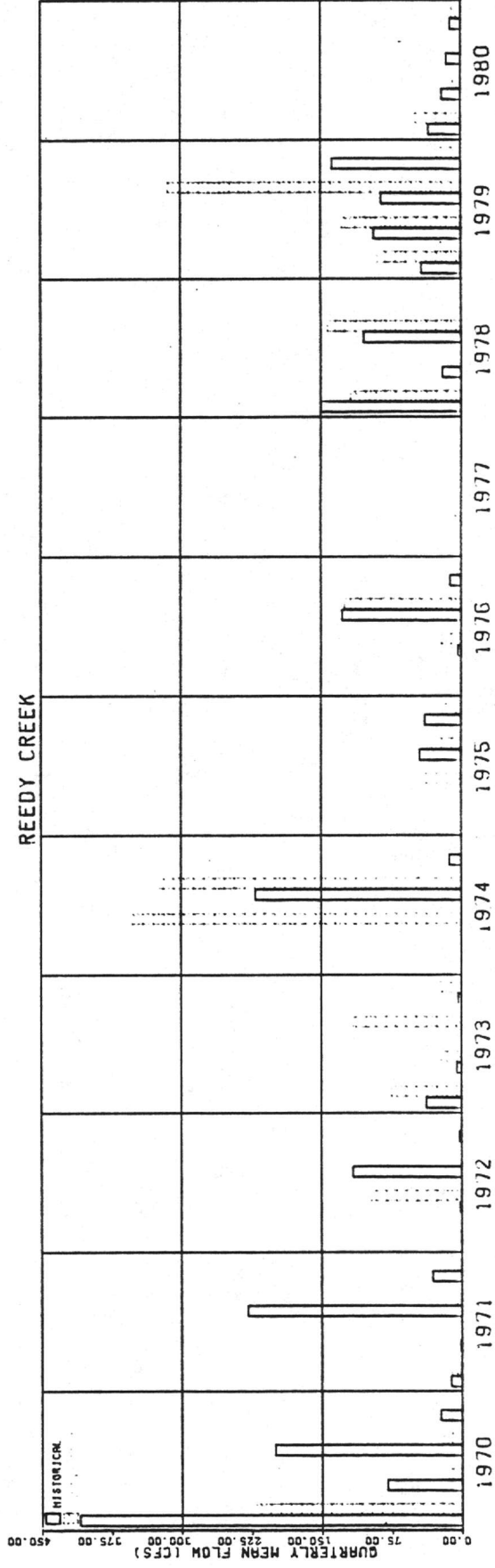
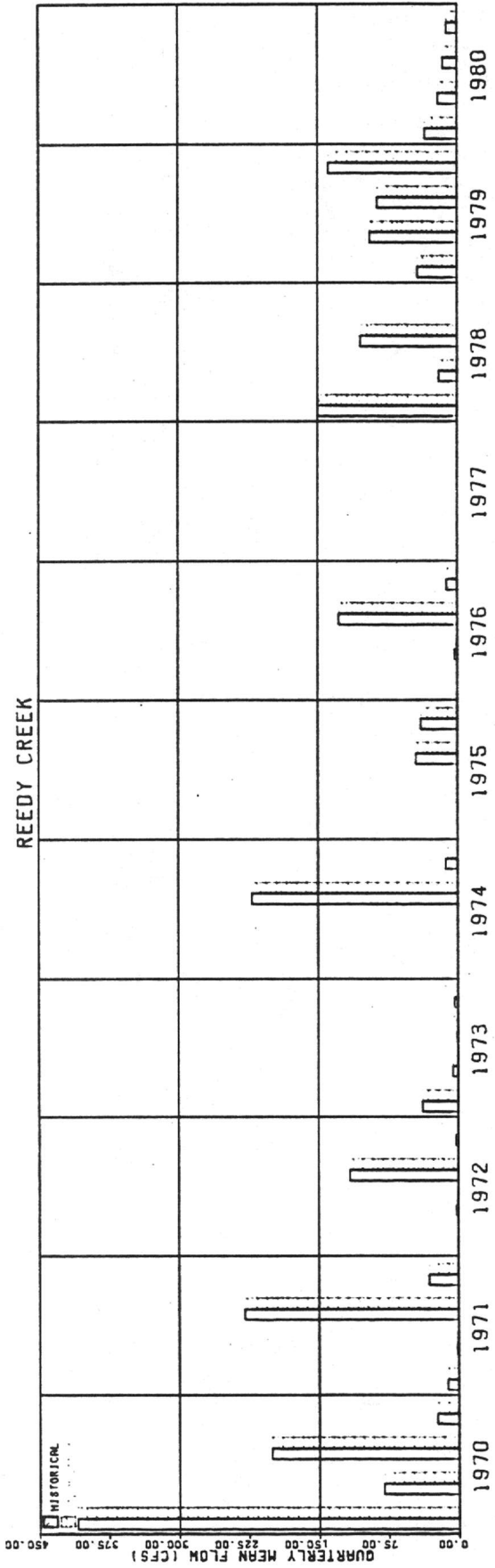
Appendix 6: Simulation and Calibration Results (1970-80)



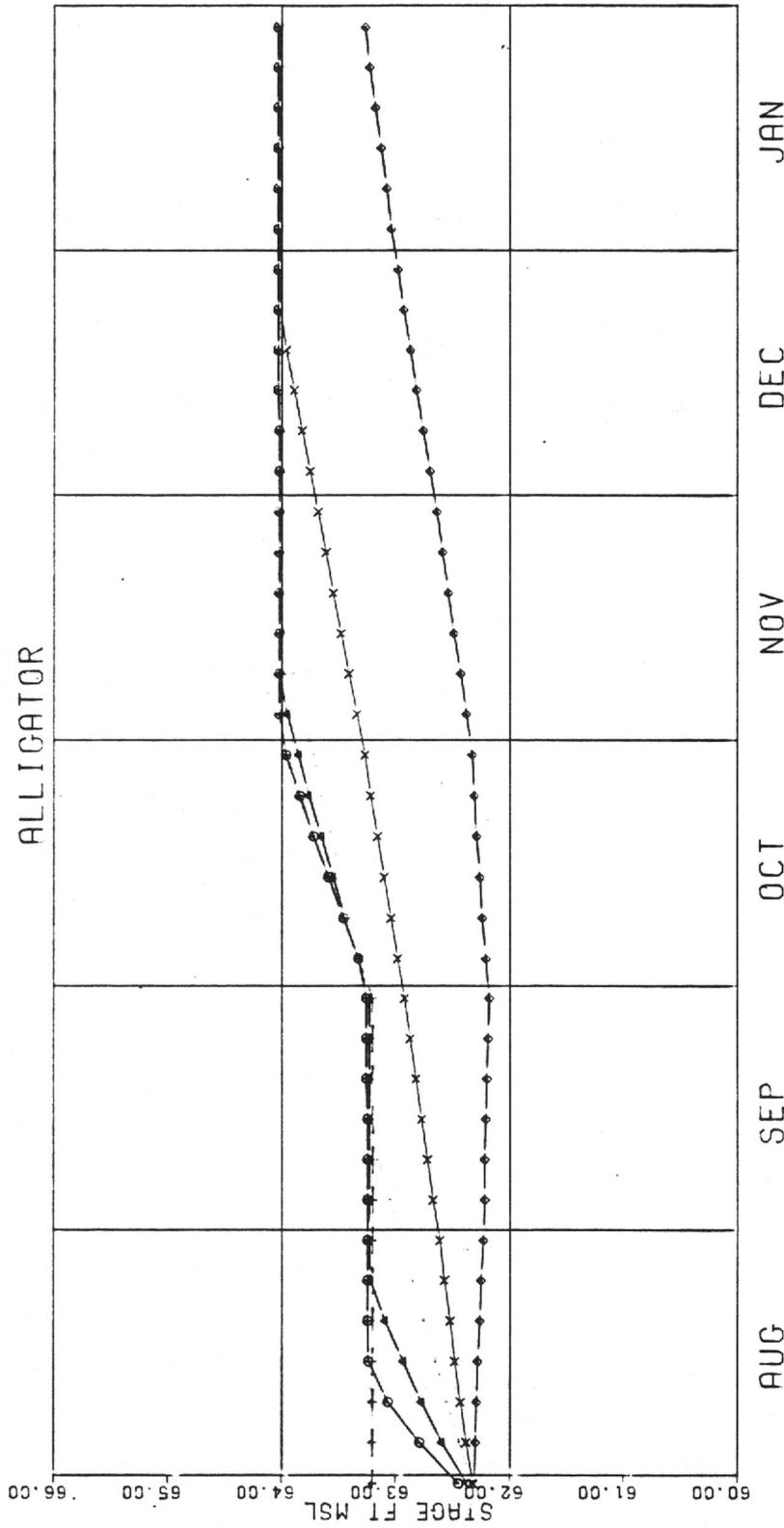
Appendix 6: Simulation and Calibration Results (1970-80)



Appendix 6: Simulation and Calibration Results (1970-80)

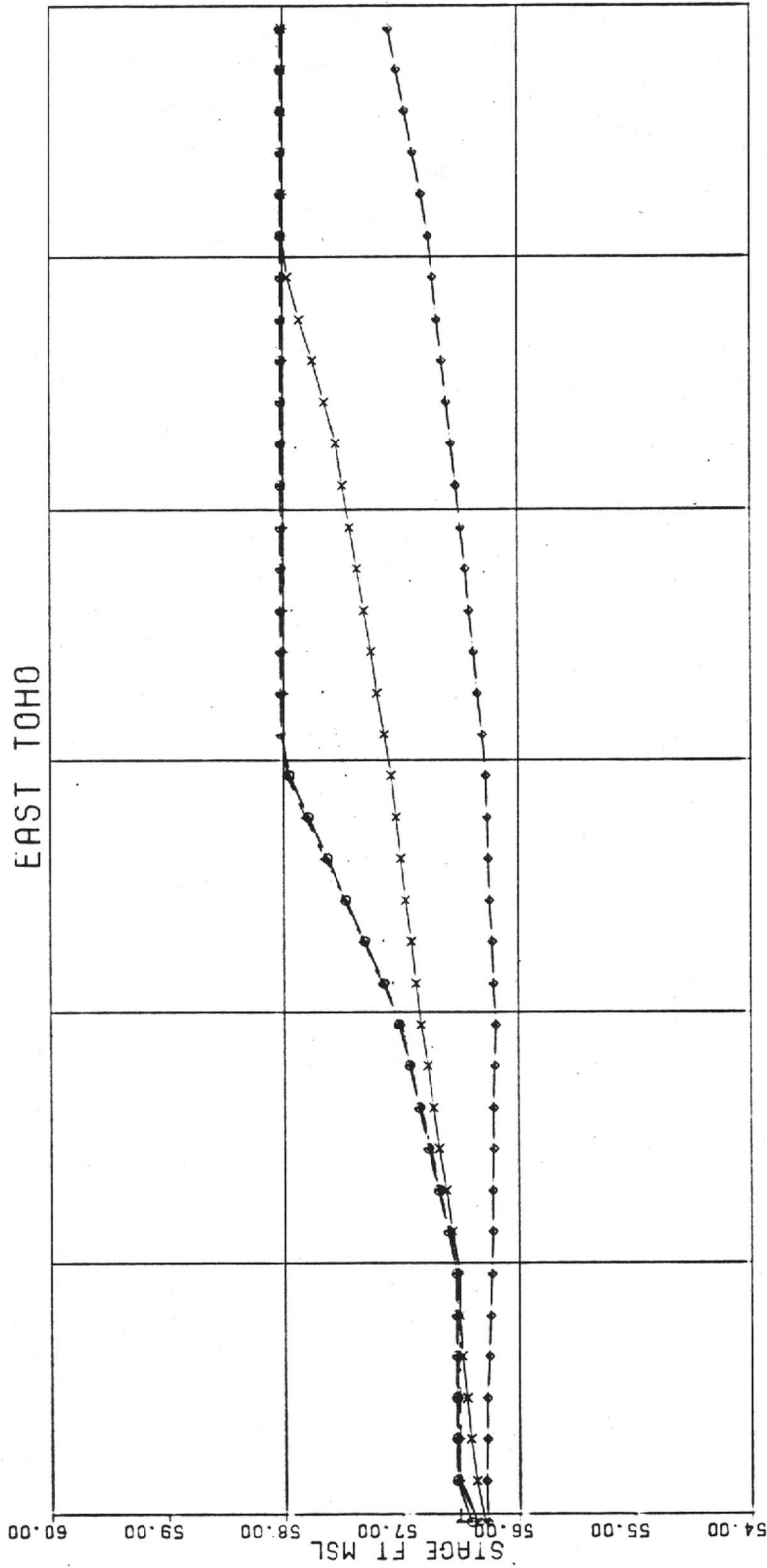


Appendix 7: Typical Forecasting Results



SIMULATION BASED ON INITIAL CONDITIONS OF 8- 1-85

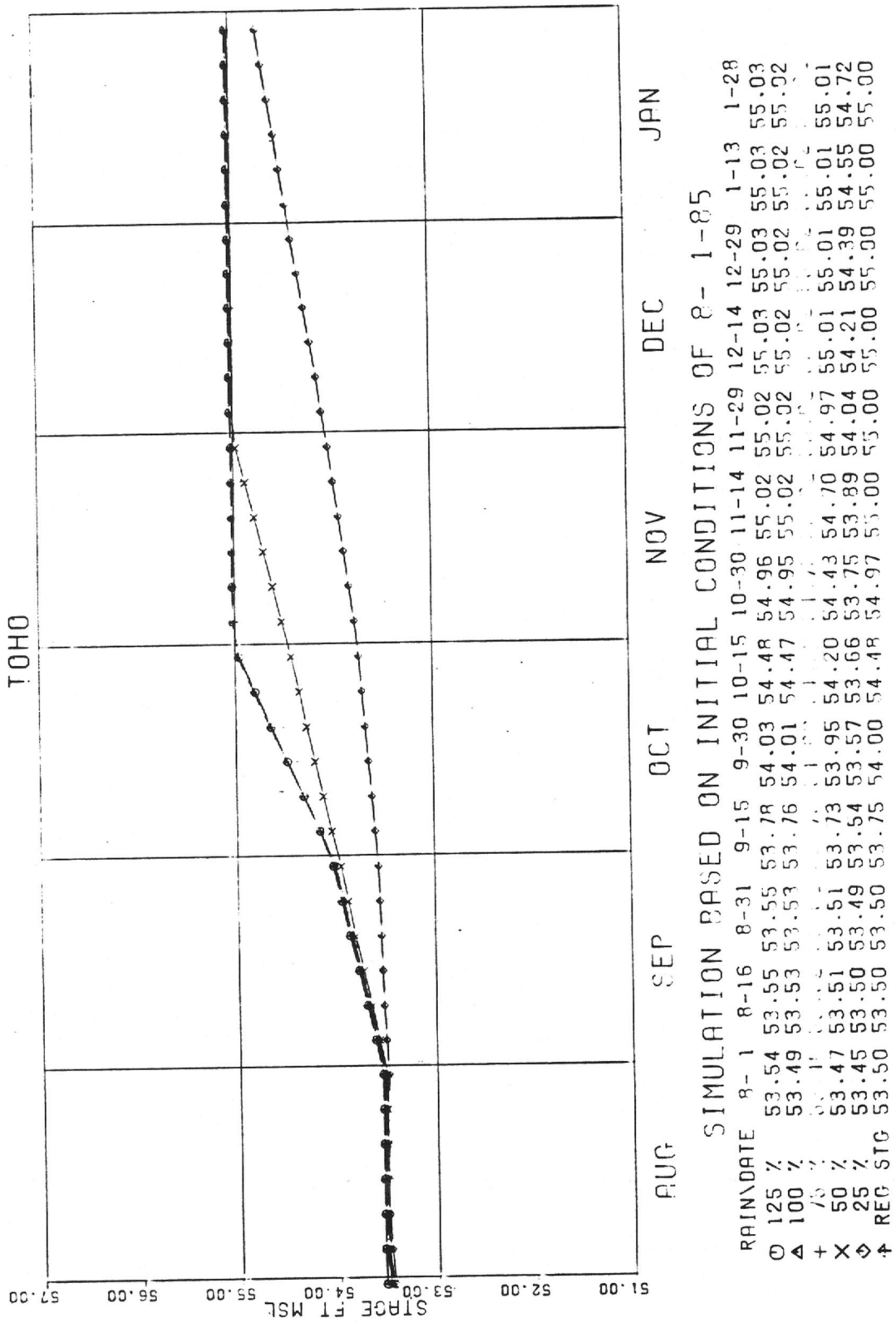
Appendix 7: Typical Forecasting Results



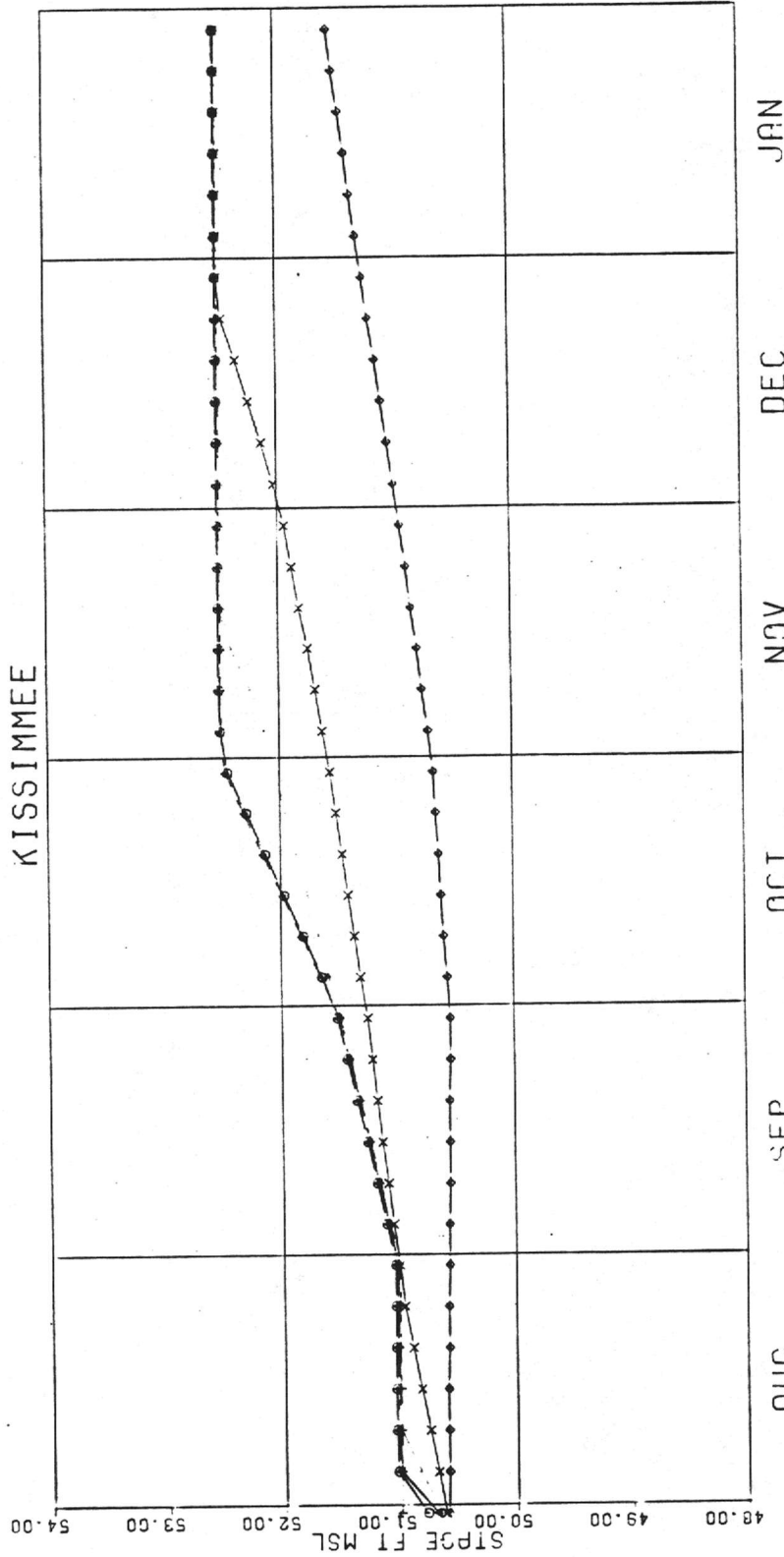
SIMULATION BASED ON INITIAL CONDITIONS OF 2- 1-85

RAIN/DATE	8- 1	8-15	8-31	9-15	9-30	10-15	10-30	11-14	11-29	12-14	12-29	1-13	1-28
○ 125 %	56.41	56.53	56.53	56.77	57.02	57.47	57.95	58.02	58.02	58.02	58.02	58.02	59.02
△ 100 %	56.35	56.52	56.52	56.75	57.00	57.46	57.94	58.02	58.01	58.02	58.02	58.02	58.02
+ 25 %	56.29	56.44	56.50	56.67	56.83	56.96	57.08	57.25	57.43	57.65	57.96	58.01	58.01
◇ 25 %	56.27	56.27	56.23	56.21	56.19	56.24	56.27	56.37	56.48	56.60	56.72	56.89	57.10
† REC STG	56.50	56.50	56.50	56.75	57.00	57.43	57.97	58.00	58.00	58.00	58.00	58.00	58.00

Appendix 7: Typical Forecasting Results



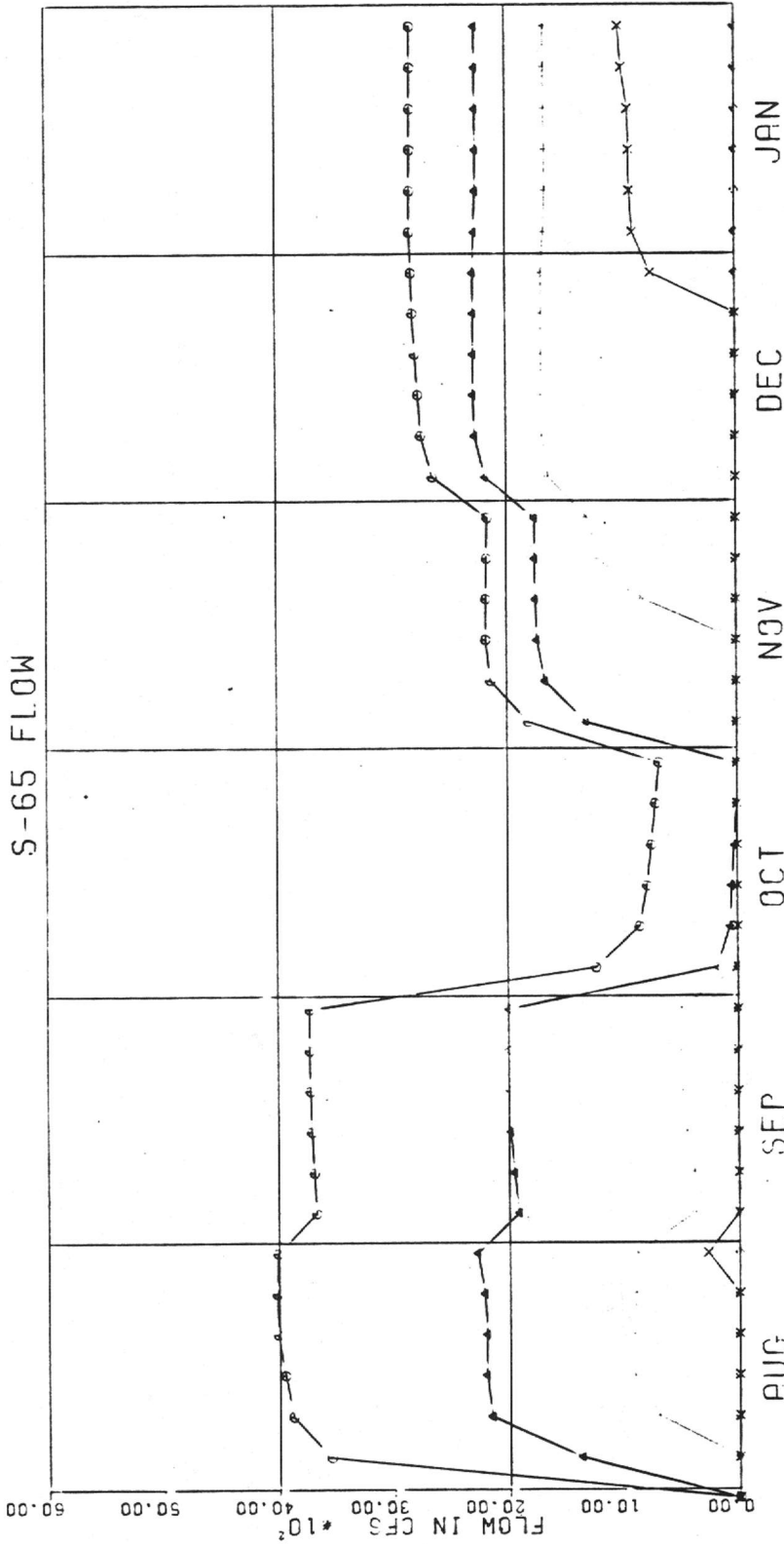
Appendix 7: Typical Forecasting Results



SIMULATION BASED ON INITIAL CONDITIONS OF 8- 1-85

RAIN/DATE	8-1	8-15	8-31	9-15	9-30	10-15	10-30	11-14	11-29	12-14	12-29	1-17	1-28
125 %	50.78	51.04	51.04	51.27	51.52	51.97	52.45	52.52	52.52	52.52	52.52	52.52	52.52
100 %	50.69	51.02	51.02	51.25	51.51	51.97	52.47	52.51	52.51	52.52	52.52	52.52	52.52
75 %	50.62	50.82	51.00	51.14	51.26	51.42	51.57	51.75	51.94	52.24	52.51	52.51	52.51
REC STG	51.00	51.00	51.00	51.25	51.50	51.93	52.47	52.50	52.50	52.50	52.50	52.50	52.50

Appendix 7: Typical Forecasting Results



SIMULATION BASED ON INITIAL CONDITIONS OF 8- 1-85

RAIN/DATE	8- 1	8-16	8-31	9-15	9-30	10-15	10-30	11-14	11-29	12-14	12-29	1-13	1-28
○ 125 %	0.	3948.	4010.	3709.	3726.	785.	680.	2180.	2161.	2760.	2817.	2824.	2822.
△ 100 %	0.	2195.	2271.	1921.	1998.	46.	0.	1732.	1750.	2275.	2277.	2253.	2255.
+ 75 %	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
x 50 %	0.	0.	274.	0.	0.	0.	0.	0.	0.	0.	736.	920.	1011.
◇ 25 %	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.