Titre: Title:	Computer model for optimal reservoir operation with specific constraints
Auteur: Author:	Gail Faveri
Date:	1992
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Faveri, G. (1992). Computer model for optimal reservoir operation with specific constraints [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/57019/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/57019/
Directeurs de recherche: Advisors:	Guy Leclerc
Programme: Program:	Génie civil

UNIVERSITÉ DE MONTRÉAL

COMPUTER MODEL FOR OPTIMAL RESERVOIR OPERATION WITH SPECIFIC CONSTRAINTS

par

Gail FAVERI

DÉPARTEMENT DE GÉNIE CIVIL

ÉCOLE POLYTECHNIQUE

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION

DU GRADE DE MAÎTRISE ÈS SCIENCES APPLIQUÉES (M.Sc.A.)

(GÉNIE CIVIL)

Avril 1992

ISBN 0-315-77657-9

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE

Ce mémoire intitulé:

COMPUTER MODEL FOR OPTIMAL RESERVOIR OPERATION

WITH SPECIFIC CONSTRAINTS

presenté par: GAIL FAVERI

en vue de l'obtention du grade de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. MARCHE, Claude, DScA, président

M. LECLERC, Guy, PhD, membre et directeur de recherche

M. LONG, N. Duc, DScA, membre

SOMMAIRE

L'agence de Mahawéli de Sri Lanka, un pays au sud-est des Indes, gère les réservoirs et les centrales hydro-électriques du Projet d'irrigation du Mahawéli. La fin première du système est l'irrigation des rizières, et la seconde est la production hydro-électrique. Dans ce projet de recherche, un logiciel en deux composantes est mis en point pour aider à trouver une politique de prélèvements qui maximise la production d'électricité et satisfasse les exigences d'irrigation.

La première composante du modèle choisit les superficies à cultiver afin de maximiser les revenus nets; elle utilise une formulation déterministe de programmation linéaire. L'optimisation porte sur trois saisons de mousson, avec une révision au début de chaque saison pour prendre en compte les apports prévus les plus récents. Cette composante calcule aussi la répartition de l'eau dans le système pour les trois saisons et détermine ainsi les niveaux de stockage requis dans chaque réservoir à la fin de chaque saison.

La deuxième composante, basée également sur une formulation déterministe de programmation linéaire, recherche la politique de prélèvements pour la prochaine période d'exploitation afin de maximiser la production d'électricité, si possible sans permettre de déficits d'irrigation, pour le

reste de la présente saison de culture. Elle peut également offrir les niveaux cibles à atteindre dans les réservoirs à la fin de la saison. Une nouvelle solution est recherchée à intervalles réguliers d'un quart mois.

Pour cette deuxième composante l'horizon de l'optimisation, la fin de la saison de mousson, est proche, et les prévisions sont plus fiables que pour la première composante du logiciel, qui a un horizon lointain de trois saisons. Cependant, puisque les politiques identifiées dans la deuxième composante sont mises en oeuvre chaque semaine sans connaître les apports réels futurs, les conséquences des erreurs dans les prévisions sont plus importantes que pour la première composante. Des ajustements continus à intervalles réguliers minimiseront, le plus possible, les conséquences des erreurs de prévisions.

La programmation linéaire exige que les relations entre les variables de décision soient linéaires; la fonction d'énergie, le produit du débit par la hauteur de chute, est non linéaire. La formulation adoptée estime l'énergie produite par une fonction linéaire qui tient compte de l'importance du stockage. Sur l'ensemble des centrales, l'estimation a une précision de 5%; avec une meilleure sélection des paramètres connexes, l'erreur individuelle diminue. Cette formulation utilisant une qui linéarise l'estimation d'énergie est jugée satisfaisante pour comparer les politiques; les effets des niveaux des réservoirs sur la production hydro-électrique sont inclus.

Les deux formulations sont mises en oeuvre à l'aide de la procédure de programmation linéaire SAS.OR, qui est très flexible et d'utilisation facile. SAS.OR est un module de SAS "Statistical Analysis System". Deux programmes en FORTRAN préparent les fichiers SAS.OR sur un micro-ordinateur compatible IBM, façon interactive. L'usager peut entrer des nouvelles données, mettre les fichiers à jour pour le prochain intervalle d'exploitation et corriger ou enregistrer sur les fichiers du micro, l'information sur les récoltes et les paramètres d'exploitation avant de créer un nouveau fichier SAS.OR.

Le logiciel a été verifié pour deux cas: une année d'hydraulicité moyenne et une année sèche historique, l'année 1955-56. Pour l'année d'hydraulicité moyenne, la politique de prélèvements trouvée produit 3684 GW-hr d'électricité, soit environ un tiers de plus que l'électricité ferme de 2711 GW-hr, sans produise de déficit d'irrigation. Pour l'année sèche, la politique de prélèvements trouvée produit 2519 GW-hr avec moins de 20 Mm^3 de déficits. Pour cette denière, il est recommande de ne pas exploiter tous les champs cultivables.

Le logiciel trouve effectivement des politiques de prélèvements, qui peuvent être vérifiées et améliorées par un modèle de simulation. L'agence de Mahawéli pourra ainsi définir les prélèvements hebdomadaires des réservoirs, des centrales et des canaux du système pour optimiser la production d'électricité tout en permettant une florissante culture de riz.

ABSTRACT

The Mahaweli Authority of Sri Lanka has the responsibility of regulating the reservoirs and power plants of the Mahaweli Irrigation Project. The primary purpose is to supply water for the cultivation of rice; however, electrical power may be produced from five power plants. A two-tier computer model has been developed to assist the Mahaweli Authority in determining the release policy which will produce the most electrical power yet ensure that sufficient water is retained for future irrigation needs. The model may be used for any configuration of reservoirs and canals operated under similar goals.

The first tier uses a deterministic linear programming formulation to plan the cropping pattern which provides the highest net revenue for the user-selected objectives and the allocation of water, particularly the reservoir storage volumes necessary at the end of the growing seasons. The planning horizon is nominally three monsoon seasons, a year and a half; however, because of the unreliability of long-range forecasts, the solution is revised at least once a season to reflect current conditions and expected streamflows.

The second tier also uses a deterministic linear programming formulation to select the short-term release policy which produces the most electricity, will incur the least irrigation

deficits, if any, and will meet target storage volumes. While the forecast horizon is only to the end of the current monsoon season, streamflow discrepancies are more critical than in the first tier; again the solution is continually revised every time step to update forecasts, crop water requirements and operating characteristics.

Linear programming formulations require that all functions be linear, however the generation of electricity is a function of the release and of the head, a function of the storage. Thus, the accurate calculation of energy is a non-linear function. Therefore a straight line approximation accounts for the effects of the storage level on the energy produced. The overall estimation error is less than five percent and adjustments to the constants involved may increase the accuracy of individual poor estimates. The formulation thus may select a policy which will retain water to increase the storage and generation capabilities of later time steps.

Both formulations are solved with a SAS.OR linear programming solution package which facilitates the preparation of sparse arrays. Two interactive FORTRAN programs were written to prepare the SAS.OR datafiles on an IBM-compatible personal computer.

The model has been tested on an average year, using mean monthly inflows and crop water requirements, and also on the water year 1955-56, a historically dry year. The inflows forecasted were assumed to be perfectly correct. During the average year, the policy selected by the model produced 3 684 GW-hr, over one third more than the projected firm energy of 2 711 GW-hr and no irrigation deficits. In the case of the dry year, the model selected a policy which failed by less than 20 Mm^3 to deliver the committed irrigation demand. Not all the land was under cultivation and 2 519 GW-hr of electricity was produced.

The two tier model does provide policies which a simulation model may more accurately assess. The correct application of the linear programming formulations ensure that the final policy will be close to the optimal for the predicted inflows and current and expected operating conditions. It is hoped that the Mahaweli Authority will find the model useful in defining weekly release policies which produce the most electricity yet support flourishing rice cultivation.

REMERCIEMENTS

L'auteur tient à remercier monsieur le professeur R. Kahawita, le co-directeur du mémoire qui a posé le problème du premier, également, monsieur le professeur G. Leclerc, le co-directeur du mémoire qui a donné plusieurs conseils techniques, et finalement le monsieur Machanayake, le directeur de l'Authorité Mahaweli qui a clarifié le but de la formulation. Sans leurs avis et leurs directions, ce projet ne serait jamais fait.

TABLE OF CONTENTS

CHAPTER 6 - THE IMPLEMENTATION OF THE MODEL BY COMPUTER

CHAPTER 7 - TEST APPLICATIONS OF THE MODEL

LIST OF FIGURES

FIGURES IN THE APPENDICES

LIST OF TABLES

TABLES IN THE APPENDICES

CHAPTER 1

INTRODUCTION

All over the world, multi-reservoir projects store water from rainy or spring snowmelt seasons in a number of inter-connected reservoirs for use during a dryer season or for transfer to comparatively dryer regions. Due to the spatial distribution of the reservoirs, the differences in reservoir capacities and release structures, and the inter-connection between the reservoirs, significant benefits may be obtained by operating the system to allow fuller reservoirs, benefitting from wetter catchment areas, to compensate for reservoirs which may be temporarily disadvantaged by hydrological conditions.

The operation of any reservoir has a direct bearing on the operation of others linked by the same river or by canals. Its releases may eventually flow to another reservoir, augmenting its inflows. A quantity of water released from one reservoir for a specific use frees the other reservoirs from supplying that demand. To fully realize the benefits of multi-reservoir projects, the operation of a reservoir should be coordinated with the operation of the other reservoirs in the entire project. When several reservoirs are involved, such global coordination becomes difficult without the use of a computer, and, in some cases, even with considerable computing facilities.

1.1 OPTIMAL RESERVOIR REGULATION

An infinite number of possible release policies might define the operation of a multi-reservoir project. Some may be clearly inappropriate, while many may be equally appealing at first sight and others may not even be immediately evident. Mathematical optimization techniques may be used to quantify the benefits from release policies and to select the best policy possible for a given system. These techniques use the computer to investigate the large number of possible releases which combine into a single release policy. The optimization of reservoir operation depends on the hydrology, the uses of the water released and the physical capacities of the reservoir network, including the distribution canals.

1.1.1 Hydrological Considerations

The hydrology of the catchment area for the reservoir network has a large impact on how the reservoirs are operated. When reservoirs are expected to fill after heavy precipitation, considerable amounts of water may be released in anticipation of the runoff. When a dry spell is anticipated, water is rationed from the reservoirs, so that a reserve for future needs will be available. Clearly, how well a particular combination of reservoir releases will satisfy the demands of the water users, will depend on the time distribution of the water supply, and to some extent, on the reliability of the predictions of that water supply.

1.1.2 Policy Objectives

The objectives of the reservoir network form the basis for selecting the best policy and rejecting those which are undesirable. Hydro-electric reservoirs release flows to meet the energy demands of electrical consumers through the year. Irrigation reservoirs release according to the water demands of the growing crops. The release policy for reservoirs which serve a mix of users must balance any conflicting demands for a limited supply by establishing which users have priority over others. All the mathematical tools for the optimization of reservoir operation require unequivocally defined objectives for the allocation of the water. Thus the benefits from each use can be easily quantified and comparisons made between various combinations of operating policies for the reservoirs in the network.

1.1.3 Physical Considerations

The number and size of the reservoirs connected by a natural river system and canals affects the choices of possible beneficial policies. Even when surplus water arrives in a monsoon season, a reservoir may not store more water than it may hold, nor may it deliver more water by canal than the channel capacity. An optimal policy chosen for a network with large upstream reservoirs and smaller downstream reservoirs closer to the user is unlikely to be as beneficial to a network which has large downstream reservoirs and little upland storage. The optimal operating policy depends on the physical sizes of the reservoir and canals, the configuration of the reservoir network, the uses and priorities of the water stored, and the hydrology of the predicted pattern of water supply.

1.2 DEVELOPMENT OF AN OPTIMAL OPERATING POLICY

The development of the present computer model to optimize the operation of a multi-reservoir network had a specific reservoir project in mind: the Mahaweli Irrigation Project in Sri Lanka. The project incorporates recently built hydro-electric dams with centuries-old irrigation tanks, which are village reservoirs, linked by ancient and modern canals. There are three major inter-basin transfers of water supplementing natural inflows. A full description of the project is presented in the following chapter. In the Mahaweli Project, water for irrigation, particularly rice cultivation, has priority over hydro-electric power generation. Due to the particular geography of the project, water may be used for both purposes most of the time. It is necessary to choose a mathematical technique to determine the optimal operating policy for this multi-reservoir dual-purpose project in Sri Lanka. Four techniques are generally used in the optimization of water reservoir regulation, a brief review follows. The third chapter reports on some applications of the various techniques in more detail.

1.2.1 Simulation

Simulation techniques attempt to model the responses of a system for a given set of operating conditions under a given set of operating rules. An unrestricted number of mathematical relations define and describe the flow of water in the system and calculate the benefits from the operation. Any number of reservoirs and canals may be included. However, the user may only evaluate a predetermined operating policy which the model does not ameliorate; only the consequences of the given policy are simulated.

Some simulation models include optimizing algorithms which try policies close to the current policy to investigate possible improvements. The possible benefits of small perturbations in the operating releases are evaluated such that the best policy of those tried will be found. However, the global optimal policy may be very dissimilar from the current policy and its greater benefits would not be discovered by the optimizing algorithms. Detailed simulation models, however, are very useful, and sometimes necessary, to verify the optimal policies determined by other techniques which have had to simplify and overlook complicated details in the real life problem to select a policy.

1.2.2 Dynamic Programming

Dynamic programming (DP) also investigates complicated functions as it calculates the benefits due to a number of possible combinations of control variables in a recursive equation. All possible values for a set of control variables are used to determine, one stage at a time, the best combination of values for the control variables. In the optimization of reservoir problems the control variables are generally the releases from each reservoir during a time-step, a stage. However, when the number of reservoirs in a system exceeds three the number of state variables become excessive and the calculations and memory required burgeon into a problem too large for even the largest of today's computers. Similar to most multi-reservoir systems, the Mahaweli Project currently has over three reservoirs, and more are planned.

1.2.3 Non-linear Programming

Non-linear programming techniques are state-of-the-art formulations of optimal programming problems. They may use recursive equations to find the optimal policy or even linear programming structures which are constrained with Lagrangian penalty terms to incorporate the non-linear function. Theoretically, they are not restricted to small systems of a few reservoirs nor to simple, linear functions. However they are new, they have yet to be verified, and they have not been widely used for real-life operating problems.

1.2.4 Linear Programming

Linear programming (LP) is a method well-suited for optimizing a problem with a large number of decision variables. Standard solution codes are available and post-optimal analyses of the cost and price variations are easily conducted. The final solution to a linear programming formulation may be proven mathematically to be optimal. However, because the procedure requires that the variables be multiplied by independent constant coefficients and not by other variables nor quantities which depend on other variables, the linear programming formulation may not accurately reflect the real life situation, especially in hydro-electric applications.

Reservoir systems may have terms where the coefficients change with the magnitude of another variable, such as evaporation loss coefficients, a function of the storage volume. The equation calculating the hydro-electric power generation is non-linear as it implies the product of the turbine release (the unknown variable desired from the solution of the reservoir regulation problem) and the head. In some cases, the head is constant, but usually the head varies with the storage in the reservoir, (another variable), whose magnitude depends on previous releases and inflows. There are thus significant non-linearities in these systems.

1.3 METHODOLOGY OF THE MODEL PROPOSED FOR SRI LANKA

A two-tier linear programming optimization model was developed to cope with the dual purpose, dual priority and dual timing of the release decision-making process found in the specific reservoir project in Sri Lanka. The model may be applied to other reservoir networks in a monsoon climate where irrigation has a higher priority than electrical power production. A simplification of the estimation of the energy being produced by a release policy maintained the LP requirement of solely linear functions. The resulting inaccuracies were within acceptable limits and any improvements made by a simulation model will be assured of being in the vicinity of the global optimum. Deterministic forecasts of the hydrological inputs are used in the model, simplifying it, and allowing the user to rapidly and explicitly assess the consequences of any given hydrological sequence.

Two micro-computer programs, which prepare the data files for the linear programming solution of the two-tier problem, although written for a specific project in terms of the objectives, allow any configuration of the reservoir network. This flexibility in the physical components not only allows for the inclusion of further details of the Sri Lankan system but also for the application to another system or for the future expansion of irrigation districts, power plants, new reservoirs, or dewatering and repairs of existing facilities. The parameters of the physical components are capable of weekly variations at all sites, to reflect the possible need for periodic flood storage, seasonally varying conveyance losses, and crop growth differences. The hydrological, crop consumption, power and system configuration data used to test the formulation of the model was supplied by the Mahaweli Authority of Sri Lanka. It is hoped that the model will assist in the weekly operation of that system.

1.3.1 The Operating Problem

The following pages describe the development of an optimization model which may be used to select an operating policy for the reservoirs in the Mahaweli Irrigation Project.

Chapter two describes the specific system that was modeled and outlines the operating objectives. The location and physical capacities of the reservoirs, canals, irrigation and electrical components of the Mahaweli system are given in detail in this chapter. The implications of the dual purposes and dual timing on the selection of optimal operating policies are explored. The last section of chapter two lists the performance objectives of the model.

1.3.2 Alternative Approaches Proposed in the Literature

Chapter three describes how other authors have approached similar problems as reported in the literature. Various mathematical techniques are presented for comparison. A justification of the approach selected for the current model is then presented with an outline of its formulation and use.

1.3.3 The Two-tier Model

The details of the mathematical formulation of each tier of the model are presented in chapters four and five. Each chapter describes the scope of each tier and the information required before explaining the linear programming formulation of each. Chapter four discusses the seasonal planning step and chapter five the short-term operating step of the model.

1.3.4 Computer Implementation

The computer implementation of the two tiers of the model is subsequently described in chapter six. An interactive program was written for an IBM compatible personal computer to prepare the linear programming data files. These data files are transferred to a mainframe for solution using a commercially available linear programming package. The PC files which are necessary for each tier of the model are explained. Also sample LP data files and sample results are given and the format discussed.

1.3.5 Model Application

Chapter seven presents some policies found by the applications of the two-tier model. A year of monthly average inflows was predicted to evaluate the performance of the model. An extremely dry year was also simulated to compare the policies chosen in each hydrological case. Programming problems encountered are mentioned as well as how they were corrected.

1.3.6 Discussions and Conclusions

The last two chapters discuss the use of the model and present the conclusions. Chapter eight lists some failings of the model, recommends the use of a simulation model to verify the policies found, and lists methods of modifying the existing model. Areas of further research are recommended before full use of the model occurs. A final section of chapter eight summarizes the current model. Chapter nine concludes the main body, evaluating how well the model meets the initial objectives.

1.3.7 Appendices

Appendix A contains a manual for the use of the data file preparation programs. Information on how to run the program, the data fields for each file, and possible run time error messages are also included. The final section of the manual gives more information on the SAS.OR linear programming package: how to manipulate the data files to add or delete constraints or variables, and how to combine them. Full listings of each data file preparation program and the definition of computer variables are given in Appendix B. Appendix C has the list of the decision variables and the constants or application parameters presented in chapter 4 and 5.

CHAPTER 2
THE DEFINITION OF THE PROBLEM

The objective of the present work is to formulate a model which may determine an optimal operating policy for a multi-reservoir dual purpose project such as the Mahaweli System in Sri Lanka. An elaboration of the physical structure, the goals and the timeframe of the decision-making process, found in the Sri Lankan project are necessary to define the parameters so that feasible policies will be found, one preferred over the others. For the solution to be valid, the parameters must be accurate.

2.1 A DESCRIPTION OF THE PROJECT

The Mahaweli Authority has the overall responsibility for the management of the power plants, the reservoirs, the village storage ponds called tanks, as well as for the development of the irrigation districts of the Mahaweli River Project. The map of the region of Sri Lanka in Figure 2.1 indicates the location of the four rivers, the seven reservoirs, the five power plants and the irrigation tanks and districts that currently comprise the project. The majority of the irrigation tanks and reservoirs lie in the relatively flat lowlands of the north and east. The upper reservoirs of

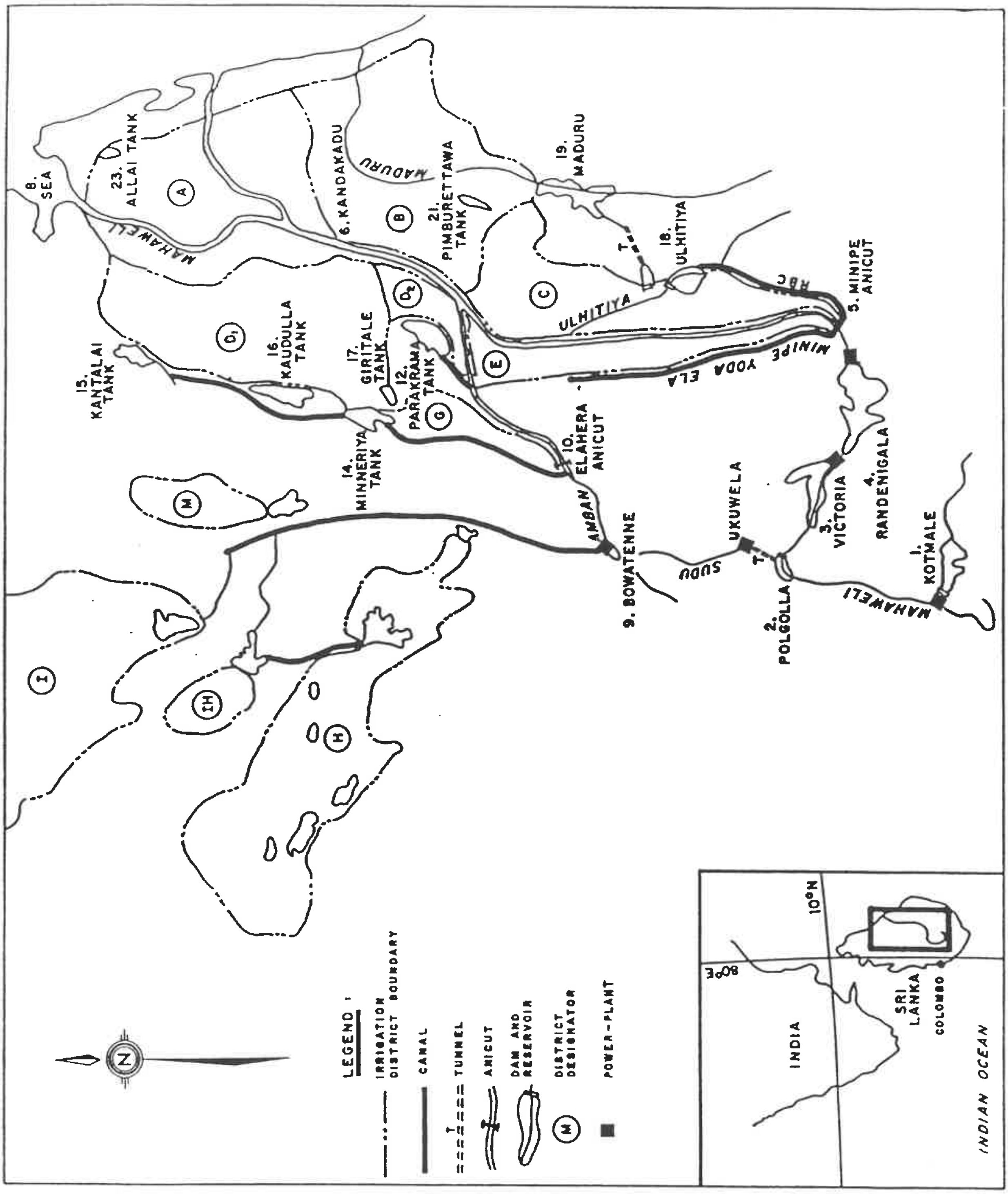

Figure 2.1: Location map of the Mahaweli Irrigation Project

the Mahaweli and Amban rivers are in the mountainous terrain of the central highlands of Sri Lanka. The high heads available in this region have spurred the construction of hydro-electric developments in the upper reaches of the Mahaweli River.

Since the power plant reservoirs are on the same river as the diversion canals for most of the ancient irrigation districts, their operation has significant consequences for the downstream irrigated fields. From the upper Mahaweli at Polgolla, through a power tunnel to the turbines at Ukuwela and into the Sudu river to the Bowatenne reservoir, flows may be diverted to augment the water supplies of the H, I, and M, districts to the north. This diversion is upstream of the largest power plants on the Mahaweli.

The diagram of the engineering works in Figure 2.2 illustrates the flow pattern between the various reservoirs and irrigation districts.

While the irrigation project has over twenty reservoirs and tanks and an optimizing model which could handle all the reservoirs and tanks was desired, detailed information on only 17 reservoirs and tanks was available. Information was available for another hydro-electric complex to the south-west of the Mahaweli headwaters. When the optimizing model was tested, data for a combined system was used.

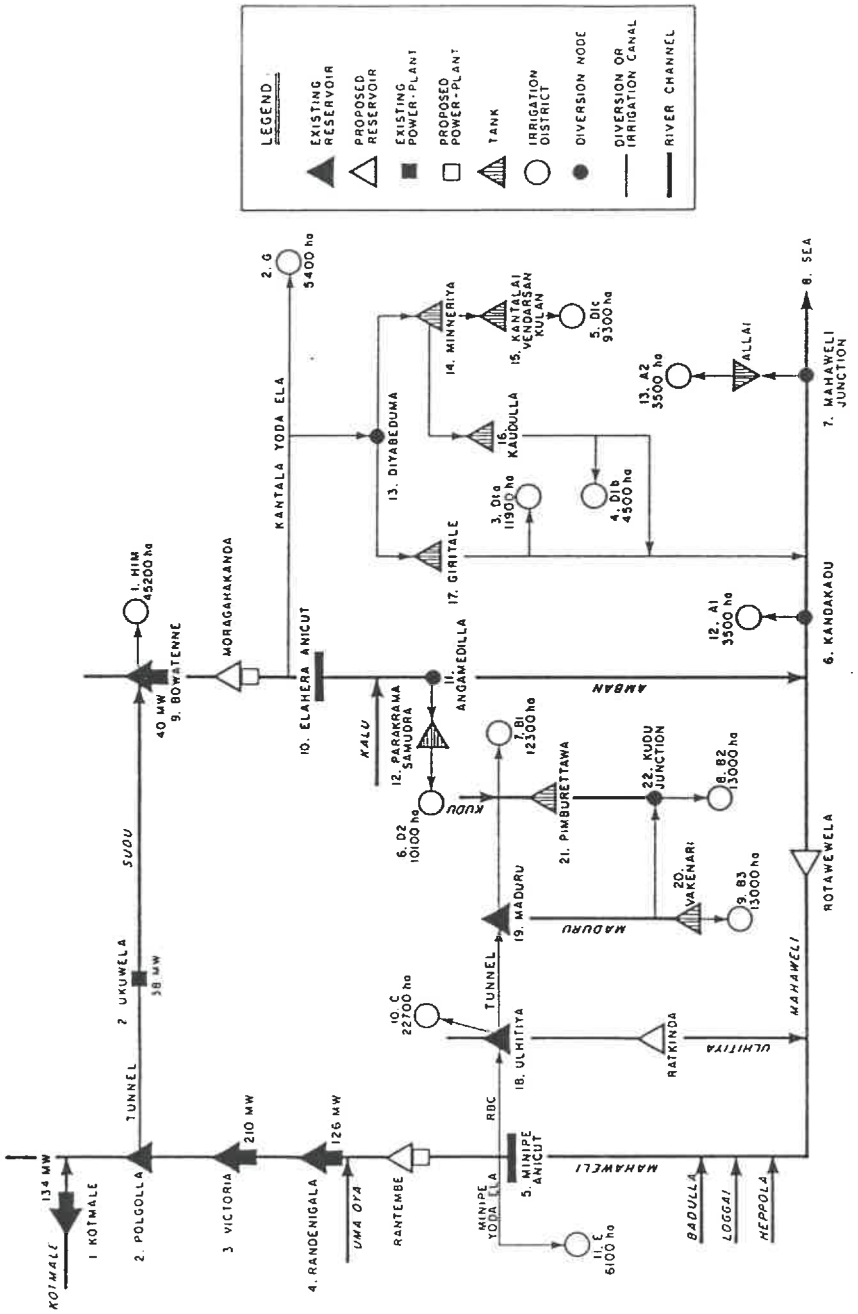

Figure 2.2 Diagram of the engineering works

2.1.1 The Configuration

Only the locations which were included in a Mahaweli Authority - Acres International Report [1986] on a 32-year simulation planning model were tested in the final application. The report tables inflows, crop water demands, reservoir and power plant data. The irrigation districts used in the model were revised to reflect the available crop water demand information. Figure 2.3 shows the schematic of the revised system.

When substantial quantities of water are diverted to meet the irrigation needs of the north, the electrical production of the Mahaweli suffers. More of the island's electrical demand must be met by costly thermal generation or by the hydro-electric power plants on the Maskeliya and Kehelgamu rivers. These rivers, to the south-east of the Mahaweli headlands, receive high year-round rainfalls on land too rugged for irrigation, but where the generation of hydro-electric power contributes significantly to the national energy production. Their energy production is limited by streamflow conditions, however, they may sometimes produce more energy allowing the Mahaweli system to divert or retain more water for irrigation purposes. The operators of the

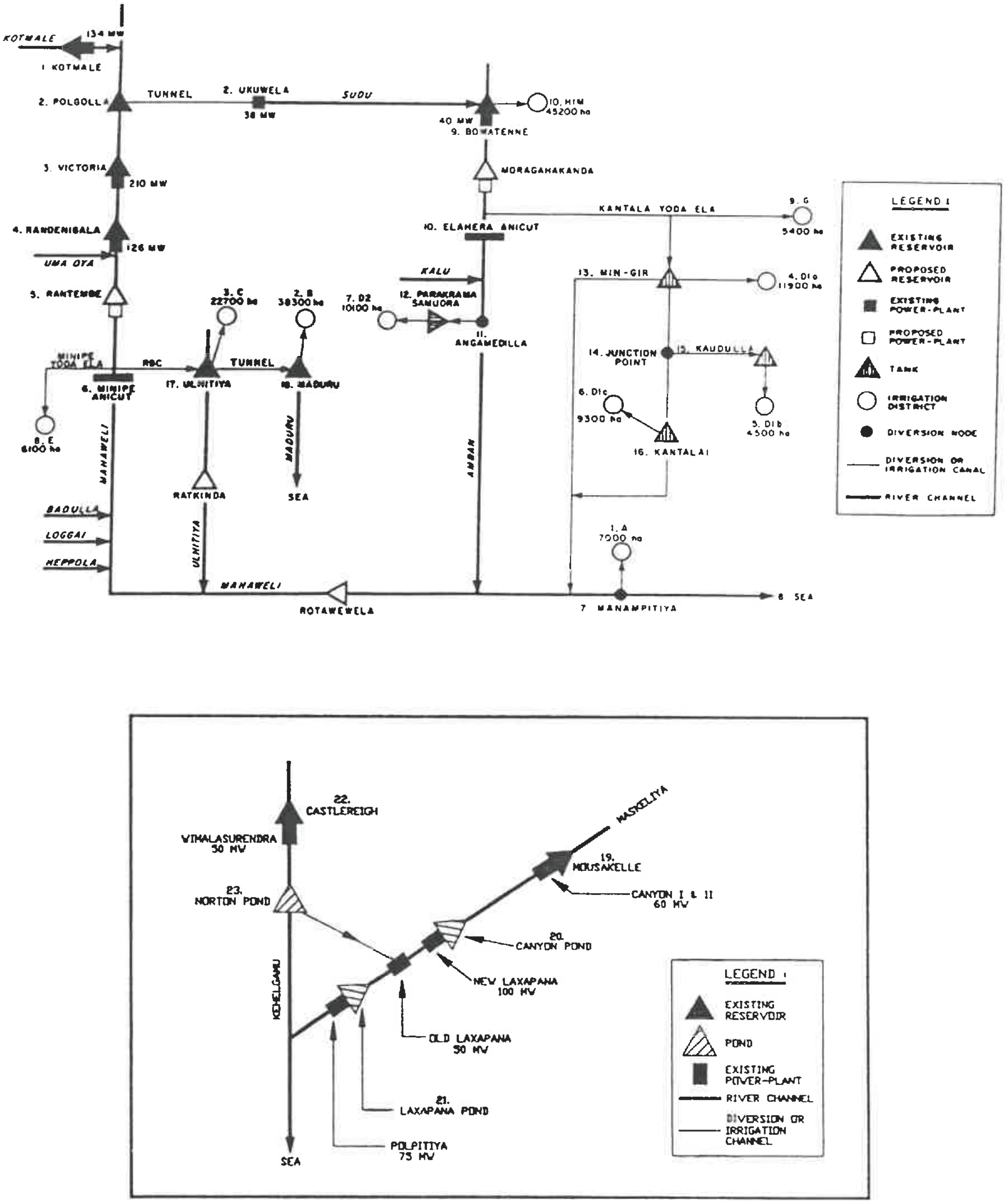

Figure 2.3: The schematic diagram of the configuration used in the applications of the model

reservoirs in the Mahaweli system must consider both demands for water releases and the interplay between the benefits due to releases for irrigation or for power throughout the year and throughout the project.

Therefore, the power plants of the Kehelgamu-Maskeliya complex were also included in the model. These power plants and reservoirs serve no irrigation demands and do not divert to the Mahaweli basin. However, their electrical production feeds the same grid as the Mahaweli power plants. At times when energy production in the Mahaweli Basin may be restricted by the irrigation demands, the Kehelgamu and Maskeliya power plants may produce more to compensate, and an overall better weekly operating policy for the entire island could be determined by including all the hydro-electric power plants in the model. The map of Figure 2.4 shows the mountainous location of these power plants.

Approximately, 160 000 hectares of land are presently irrigated from the storage tanks with a combined volume of over 1 600 million cubic metres (Mm^3). The power plant reservoirs have an additional combined volume of 1 000 Mm^3, while the power plants have a total installed generation capacity of 584 MW (Mahaweli Authority - Acres International [1986].

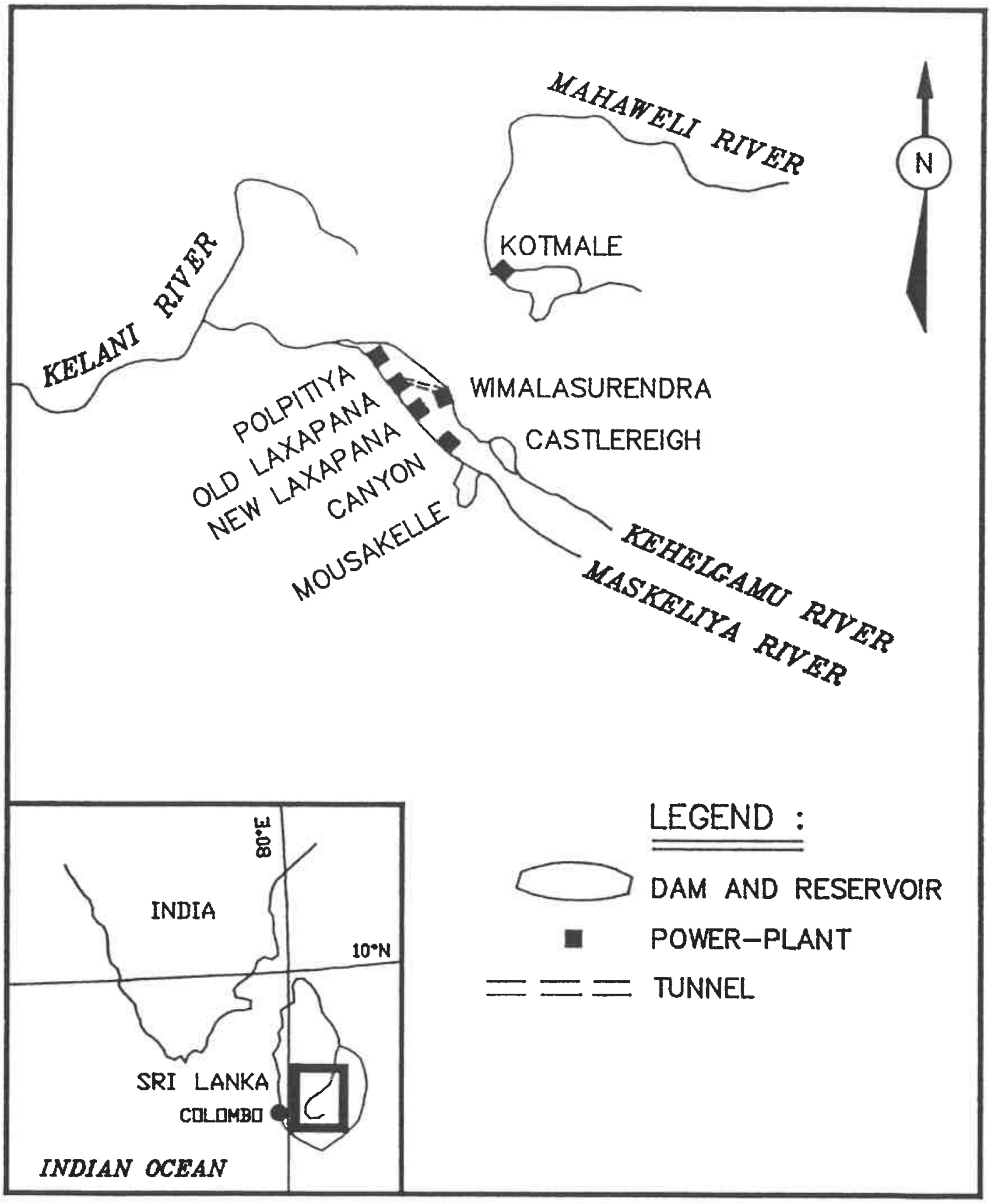

Figure 2.4 Location of the Kehelgamu and Maskeliya Complex

The hydro-electric data used for the power plants is given in Table 2.1. The reservoirs and their sizes are given in Table 2.2. The irrigation districts are shown in Table 2.3.

TABLE 2.1

Power plants in the Mahaweli Project

NODE NUMBER		NAME	MAXIMUM CAPACITY (MW)	RATED HEAD (m)	FLOW MINIMUM (m3/s)	CAPACITY MAXIMUM (m3/s)
1	1	KOTMALE	135	201	4.5*	80.5
2	2	UKEWELA	38	78	4.5*	58.6
3	3	VICTORIA	210	190	4.5*	137.5
4	4	RANDENIGALA	126	78	4.5*	221.9
5	9	BOWATENNE	40	55	4.5*	93.3
6		RANTEMBE	17	30	4.5*	675.0
7	19	CANYON	60	204	4.5*	133.6
8	20	NEW LAXAPANA	100	578	4.5*	75.0
9	21	POLPITIYA	210	190	4.5*	125.0
10	22	WIMALASURENDRA	50	227	4.5*	99.0
11	23	OLD LAXAPANA	50	449	4.5*	48.2

* arbitrary value, reduced when necessary

Values from Mahaweli Authority - Acres International 1986 Report, page 8 and 10

Maximum Flow Capacities calculated from installed capacity, power coefficients.

TABLE 2.2

Reservoirs in the Mahaweli Project

	NODE NUMBER	NAME	STORAGE MINIMUM (Mm)	CAPACITY MAXIMUM (Mm)
1	1	KOTMALE	22.2	500.0
2	2	POLGOLLA	1.0*	2.0*
3	3	VICTORIA	34.0	768.0
4	4	RANDENIGALA	270.0	1000.0
5		RANTEMBE	0.0	16.6
6	8	SEA	0.0	99999.0*
7	9	BOWATENNE	15.0	60.0
8	12	PARAKRAME	20.0	130.0
9	13	MINNERIYA-GIRITALE	0.0	136.0
10	15	KAUDULLA	5.5	136.0
11	16	KANTALAI-VENDARASAN	0.0	160.0
12	17	ULHITIYA	40.0	230.0
13	18	MADURU	0.0	478.0
14	19	MOUSAKELLE	3.9*	140.0
15	22	CASTLEREIGH	3.0*	44.8

* arbitrary value

Values from Mahaweli Authority - Acres International 1986 Report, appendix III

TABLE 2.3

Irrigation districts used in the applications of the model

	NODE NUMBER	DISTRICT NUMBER	NAME	LAND FOR IRRIGATION MAHA SEASON (hectares)
1	7	1	A	7 000
2	18	2	B	39 300
3	17	3	C	22 700
4	13	4	D1a	11 900
5	15	5	D1b	4 500
6	16	6	D1c	9 300
7	12	7	D2	10 100
8	6	8	E	6 100
9	10	9	G	5 400
10	9	10	HIM	30 800

Values from Mahaweli Authority - Acres International 1986 Report, page 12.

Plans are underway to develop more irrigation in the lowland areas southeast of the Mahaweli River, and to build more reservoirs and power plants. The power plant at Victoria is to receive more turbines. Even at the present time, the project is a large and complex system, with inter-basin transfers, tributaries, and sub-systems of canal-fed storage tanks supplying the irrigation districts. The model chosen for the Mahaweli Basin must be flexible to include any desired variations and actual operating conditions.

2.1.2 The Hydrology in the Project

The Mahaweli River is the largest river in Sri Lanka, starting in the central mountains and descending to the north-east plains. Its drainage basin is over one million hectares with an average annual flow of over eight billion cubic metres ($8\ 10^9\ m^3$) at the mouth of the river, (Sri Lankan Irrigation Department [1974]). The other rivers in the project have smaller drainage basins and lower flows, hardly a quarter of the magnitude of the Mahaweli. The larger flow volume of the Mahaweli is due to runoff from both monsoon rainfall seasons, the summer-time Yala and the winter-time Maha, in its upper reaches. The Kotmale basin, the uppermost watershed of the Mahaweli in the central highlands, yields the highest annual runoff per hectare in the country. In contrast, the other rivers used to supplement the irrigation supply start further north and east where they receive only the winter (Maha) rainy monsoon season due to the rain shadow of the central mountains.

The monsoon climate of Sri Lanka, dividing the year into two rainy seasons, which on the plains to the north and east of the central mountains are a wet and a dry season, is erratic. Unlike northern climates, winter snowfall may not be measured and used to predict later streamflow volumes. The timing and quantity of the monsoon rainfall is a function of continental pressure systems far offshore. The water available to the Mahaweli Authority varies every year and, in some years, has failed to arrive in the expected amounts.

The majority of the flatter land traditionally used for agriculture and in the irrigation districts of the Mahaweli Project lies in the dry zone. To irrigate the rice paddies in the dry zone, engineers have diverted water from the Mahaweli for centuries, storing the water in the ancient tanks, such as at Parakrame Samudra, or irrigating directly as from the Minipe Anicut to the fields of district E.

The time of concentration of the basin is very short. Due to the steep, rocky slopes of the watershed, flood peaks reach the sea in a day or so. The water released from an upstream reservoir also reaches downstream reservoirs within a day, simplifying streamflow calculations for periods longer than two days.

2.2 THE IMPORTANCE OF THE WATER USES

The use of the water for irrigation has long-standing political and sociological importance for the Mahaweli Authority. It is a goal of the government to become as self-sufficient in rice production as in centuries past. The development of new irrigation areas is often accompanied by new settlements, the supply of water for the fields may not be arbitrarily denied to generate hydro-electric power elsewhere, nor be seen to be favouring other districts. The importance of the benefits from irrigation cannot always be reduced to a simple economic factor such as the price in Sri Lankan rupees. The decision-maker at the Mahaweli Authority has to weigh the relative importance of power production and irrigation demands, in the cases where the two goals conflict. The model was developed considering that the project was an irrigation project first and foremost, and that any electrical benefits were secondary and supplemental.

As practised in Sri Lanka, the irrigation of rice entails flooding the fields to prepare the soil before the seed is sown. Once the rice shoots are established, water is again applied to sit on the fields and impede the growth of weeds. A relatively high and constant demand for water throughout the three to four month growing period is thus necessary. Traditionally, upland crops, such as beans, chillies, and vegetables, which require less intensive irrigation, have been grown on higher land or in the dry Yala season. The goal of self-sufficiency in rice production requires more fields in year-round rice cultivation, and a better management of the available streamflows.

2.3 THE DECISION TIMEFRAME

The timeframe for which the decision-maker operates the reservoirs is also pertinent. Since the monsoon rains may be early or late, the date that the reservoirs will again be full is uncertain. To allocate water in the present entails weighing the likelihood of future water supplies and needs. Presently twice a year, at the beginning of each monsoon season, the level of irrigation for the next season is determined from the amount of land seeded in rice, in other crops, or idle. The crop pattern is based on actual precipitation and current reservoir and tank storage levels. While considerable amounts of electricity may be generated in the current season, the reservoirs may be so depleted that with low rainfalls in the next season, little electrical generation or irrigation can occur in the following seasons. The decision of how much water to allocate for the irrigation of the crops and for electrical generation must consider the demands and expected water supplies further into the future than the current season.

Prediction techniques confidently forecast for, at most, a month into the future. Midway through the season is too late for more or less fields to be planted and proportionally more or less water released for irrigation. However, releases for hydro-electric generation may respond more spontaneously, diminishing or increasing as the actual water supply allows. They are only a function of current reservoir levels, crop water requirements and the streamflow and rainfall predicted over the short term. The required optimizing model must reflect this dual decision framework and the uncertainty involved in the streamflow predictions.

2.4 REQUIREMENTS FOR AN OPTIMIZING MODEL

The model that the Mahaweli Authority may use to assist in the allocation of water under its jurisdiction must be a simple to use, comprehensive decision-making tool. The optimizing model, while undertaking to maximize the electrical production, must recognize the higher priority of the irrigation demand for water. It must allow for future expansion of the project, more irrigation districts, new power plants, additions and repairs to those presently in operation, and additional reservoirs. The flexibility desired means the model may be applied to any irrigation and electrical project, with any configuration of reservoirs and canals, with similar priorities. The model must allow the decision-maker to consider the system on a weekly basis, while accounting for future water requirements as far ahead as a year and a half, in an uncertain monsoon environment. It must be able to respond to shortfalls in expected precipitation, or conversely, more rain than forecast, allowing the decision-maker to view the future consequences of present policies. The optimizing model will determine, based on the above considerations, how much water to release each week to crops already growing, how much water may be released to generate electricity each week, and how much water needs to be stored in reserve for the future.

CHAPTER 3

OPTIMIZATION IN RESERVOIR OPERATION

Since Thomas and Revelle [1966] analysed the trade-off between irrigation and power uses for the water retained by the Aswan Dam, numerous studies have been published discussing the optimization of water resources management. The techniques used have ranged from classical linear programming used by Thomas and Revelle [1966], to chance constraints (Eisel [1972], and rule curve variations (Revelle, Joeres and Kirby [1969]), modified dynamic programming (Fults, Hancocks and Logan [1974]), combinations of linear and dynamic programming (Becker, and Yeh [1974]), as well as nonlinear programming (Laufer and Morel-Seytoux [1979]) to mention but a few authors. They attempted to analyse, not only the operation of existing facilities but also, in some cases, the planning decisions of when, where and to what capacity to build reservoirs in a multi-purpose system. The problem of optimal management of several reservoirs, with the nonlinear economic function of power generation as well as irrigation uses, and the uncertainty of water demands and streamflows, is not easily solved as discussed by Yeh [1985]. Though many models have been reported, few have been actually used in practice. In the following, the techniques which have been proposed to solve the problem will be presented and their shortcomings discussed.

3.1 SIMULATION

Sigvaldason [1976] used a simulation model of the Trent River system in Ontario, to analyse the forty-eight reservoir basin in detail. The primary purpose of the regulation was to maintain lake levels for recreation, for municipal water supply, and for fish and wildlife, as well as for some low priority hydro-electric power regulation. Various zones in each reservoir were identified and penalties assigned for differing from a perceived optimal operating level. Flows in the connecting canals and natural streams were also assigned zones of varying penalties. The out-of kilter algorithm mathematically determined the operating policy which minimized the penalties due to violations form the desired operating levels, finding a feasible policy very efficiently. The algorithm works on an analogue of an electrical network, with nodes and arcs. Initially, the user must give preferred reference levels, which are not expected to be met at all times due to the variable nature of streamflows. In the years when the preferred levels are not optimal due to the expected level of streamflows, a truly optimal policy may be over-looked.

3.2 MODIFIED DP

Various modifications to the dynamic programming algorithm as expounded by Bellman [1957] have been proposed retaining its capacity to exactly model nonlinear benefits, but reducing the number of calculations required, by restricting the range of choices. Unfortunately, the solutions may not be definite global optimal solutions.

3.2.1 DDDP

Discrete Differential Dynamic Programming (DDDP) starts from a trial release policy and calculates the benefits from policies slightly different from the first policy in an attempt to find a better policy. More than one attempt may be made as the trial policy is shifted in the direction of higher benefits. Samaratunge and Samaratunge [1980] used a DDDP model to optimize the generation of power on the Kehelgamu and Maskeliya rivers in Sri Lanka, the basin to the south-west of the Mahaweli with five power plants and two monthly storage reservoirs.

Meredith [1975] suggested using DDDP with the reservoirs as the stages and the monthly time step as the states. Because forecasts in any case, are not reliable for longer than about five months, the limit of four control variables in dynamic programming would not be too restrictive and one would be permitted to have many reservoir stages. However the regulation of the Mahaweli Project must consider the irrigation requirements of an entire growing season in shorter time steps than one month. The Yala season is five and a half months long, the Maha is one month longer.

An inordinate amount of computer time is still required for large multi-reservoir systems when using either form of DDDP. The solution may depend upon the starting trial policy as only local optima are found, and the shifts in trajectory may be insufficient to locate the direction of the true global optimum.

3.2.2 Other Modifications of DP

Turgeon [1980] has optimized a single-purpose multi-reservoir operation on the Ottawa River for waterpower production in a two state DP problem solved sequentially. In turn, reservoirs are isolated from the rest and the remaining are aggregated into one effective reservoir. However, if the reservoirs are distinctly different, the aggregation is unsuitable. Turgeon [1981] has also developed an iterative algorithm similar to DP, termed progressive optimality. The procedure reduces the calculation and, especially, the memory requirements of the standard procedure but a global optimum cannot be assured.

3.3 LP-DP

Many recent studies (Grygier and Stedinger [1985]; Mohammedi and Marino [1983, 1984]; Gilbert and Shane [1982]; Becker, Yeh, et al. [1976]) have nested LP formulations in a DP formulation of the benefits to the system as a whole. These statements of the problem treat the releases from all the reservoirs as one state variable, whose value is calculated by the LP problems at each time step stage. The generation of power is calculated within the DP step, allowing the linear programming step to deal with the releases in a large number of reservoirs. The authors of these studies have analysed the inter-basin networks managed by the California Valley Project and the Tennessee Valley Authority, which operate for other purposes than hydro-electric generation, though power is still of the highest priority.

3.4 NONLINEAR PROGRAMMING

A hierarchical approach was used for a multi-purpose multi-reservoir system in California. Adiguzel and Coskunoglu [1984] nested models of successively smaller timeframes within one analysis. All the models calculating the electrical production used a nonlinear programming technique of Lagrangian and penalty parameters to compensate for the nonlinear nature of the power production function. Soliman and Christensen [1986] used a functional analysis with a minimum norm formulation to optimize the operation of a system of reservoirs with variable heads. Lagrangian and Kuhn-Tucker multipliers were used to treat the nonlinear terms.

3.5 APPLICATIONS OF LP

Despite its restriction to linear relations, linear programming has been used in a number of situation where the number of reservoirs and power plants have been large. The nonlinear terms have either been ignored, considered linear using constant average coefficients, or piece-wise linearized using mixed integer programming. LP problems may be easily coded for solution by a computer and the results analysed for their sensitivity to inflow magnitudes or variations, the influence of various parameters gauged, and the results checked.

3.5.1 Linear Decision Rule

Leclerc and Marks [1973] have used linear decision rules to analyse a basin in Quebec for streamflow regulation and recreational purposes. The method uses the recurrent pattern of annual flows to develop an optimal policy for short-term releases - expressed by the linear decision rules. Conflicting purposes are handled in the economic function or as constraints. The resulting operating policy is thus a standard policy to be followed year after year according to the month. It depends on a regular cyclical streamflow pattern for maximum benefits to be realized.

3.5.2 Benders Decomposition

Pereira and Pinto [1985] report on an iterative solution to LP problems each solving the decomposed benefits of the economic function. The economic function was partitioned into linear terms using a mathematical principle proposed by J.F. Benders [1962]. This procedure was able to solve the stochastic operation of a 37 reservoir system for hydro-electric purposes in Brazil on a micro-computer.

3.5.3 Successive Estimation of the Power Benefits

For an irrigation and power project in India, Chaturvedi and Srivastava [1981] proposed an LP formulation to determine the capacities of proposed new reservoirs as well as the monthly operation of each reservoir. The power production term utilized an "effective" head in preliminary runs. Subsequent runs used the head associated with the release policy of the previous run, until no significant difference was observed. The authors reported that only two runs were necessary to match the effective and actual heads over the monthly periods and for the planning goals of their study.

3.6 UNCERTAINTY

The uncertainty of the inflows, and to a lesser extent the unknown future demands for power and irrigation, add a further complexity to the problem of multi-purpose, multi-reservoir operation. Stochastic analyses, explicitly considering the probability of an inflow quantity, result in nonlinear terms and more choices to consider. Chance-constrained linear programming includes additional constraints to account for the unknown quantity of future flows. Using historical streamflow sequences might be valid for long-term planning purposes, but for short-term operation they are likely to yield dangerous release policies. Predicted deterministic flow sequences are only as good as the prediction and as previously stated, are difficult to forecast further ahead than five months. Monsoon precipitation is very unpredictable.

Stochastic analyses and chance-constrained techniques require confidence in the probability analysis of the inflows to each reservoir. They result in probable courses of action, but the decision-maker is still left with the dilemma of what immediate action to take. When a stochastic model determines one policy based on a series of equally probable hydrological sequences, the model ignores the inflows which are the most probable. A weighting procedure might be employed to ensure that a policy beneficial for the hydrological sequence most likely to occur, is the policy selected.

A repetitive technique, which is updated every time step, has successfully been used to determine short-time policies while considering the future water requirements of a system. The predictions for the first few time steps, which are likely the most accurate, are combined with estimates of the later time steps to enable the model to determine a policy which looks ahead into the future. Because the model updates the operating policy every time step, the actual short-term operation reflects the better predictions of the short-term period. The longer-term operating policies are not enacted though the continually updated model may recommend them on the basis of the short-term forecasts once their time has come.

3.6.1 Uses of the Repetitive Technique

Adiguzel and Coskunoglu [1984] describe the process of repetitive solution as repetitive optimization via rolling horizons. They used it to update their nonlinear programming model of the California Valley Project, including the actual streamflows of the last period, before determining the releases of the next. Dagli and Moles [1980] termed the process adaptive planning (AP) in conjunction with their solution of the operation of a multi-purpose reservoir system in Turkey. They used a linear programming solution, linearizing nonlinear terms.

3.7 OPTIMIZATION IN IRRIGATION APPLICATIONS

Because of the priority of irrigation demands in the Mahaweli Project, reports of the optimization of irrigated agriculture were also studied. The same optimizing techniques are found but with a new application. For example, Hall and Butcher [1968,1969] presented a dynamic programming formulation to determine how much water to deliver to crops under scarce water conditions. Rhenals and Bras [1981] reported on a stochastic dynamic programming technique treating a similar situation. The procedures suggested involve detailed investigations of the soil-moisture content, plant yield and weekly potential evapo-transpiration.

3.7.1 Crop Allocation

Two reports describe irrigation projects and the allocation of land and water to crops. Maji and Heady [1980] used deterministic and chance-constrained formulations to determine the best use for water, labour and land resources in a one reservoir, six irrigation district project in northern India. Linear programming solved the planning problem. In southern India, Mohanty, Sahu, and Rao [1976] used a hierarchial model to first plan the agricultural allocation, then secondly decide the optimal release schedule, in a one-reservoir two-power plant four-irrigation zone project. Linear programming was used to both allocate crops in the irrigation districts and to determine how much water to release for power production, irrigation, and flood.

3.8 IN SUMMATION

Many techniques not discussed here have been proposed for the solution of the optimal operation of reservoir systems with multiple goals. Similar to the above techniques, they each have their advantages and their shortcomings. They may not guarantee that a radically different policy might not have higher benefits. They may be difficult to implement, requiring detailed cost functions, many parameters, or custom designed computer codes. The Mahaweli Authority requires a model that is easy to use, reasonably simple, yet still accurate, and which gives a release policy of the most benefit to planted crops. Some aspects of the various models mentioned above have been combined in the model presented below.

No single model above appears to account for all the aspects of the regulation of the reservoirs in the Mahaweli

Project. Some of the optimization techniques are incapable of analysing the large number of reservoirs, power plants and irrigation districts which comprise the project. Others do not recognize the dual decision timeframe implicit in the operation of the reservoirs for both irrigation and hydro-electric purposes, with irrigation commanding a higher priority. Others are too complicated, requiring a great familiarity, not only with the operation of the Sri Lankan reservoirs, but also with nonlinear optimizing techniques.

A dual decision timeframe occurs because the irrigation allocation is decided for the growing season before the crops are planted while the waterpower release schedule may be revised at any time. If later in the season, more or less water is available than expected, it is not possible to change the crops planted, although revisions may be made to the cropping plans for the next season. The production of energy may, however, increase or decrease according to actual streamflows whether or not they were anticipated well ahead of time. How much water to release for electrical energy production is a decision independent of past decisions.

3.8.1 The Hierarchial Approach

Therefore, the approach has been to divide the formulation into two models, one used before the other. The hierarchial structure reflects the priority of irrigation over electrical production, and the difference in decision timing. The seasonal planning model, PROJPLAN, permits the decision-maker to quantify the revenues to be earned from a water policy determined for the cropping pattern planned for the next three seasons. At the same time, PROJPLAN has determined the storage levels necessary in each reservoir at the end of the growing seasons so that enough water will be available for the subsequent season's planned cropping pattern.

The short-term operating model PROJOP uses these reservoir storage targets to ensure that the operating release policy which maximizes the electrical production leaves sufficient water in the reservoirs for the next season. PROJOP calculates, for the next time step, the best release policy for every reservoir, diversion and turbine, considering the irrigation and electrical production demands for the entire season and the reservoir storage targets of the final time step. Figure 3.1 illustrates the structure of the hierarchial model.

Both models use adaptive, repetitive updating of current reservoir levels and streamflow forecasts to allow the decision-maker to respond to unforeseen hydrological events as they become apparent. The release policy is revised each week with a new PROJOP run as the growing season progresses; the short-term model may thus revise the release policy according to the changes in anticipated future conditions. PROJPLAN runs during the season may also update future cropping plans and the concomitant target reservoir storages. The repetitive solutions, using updated streamflow forecasts and reservoir levels, allow the model to respond to uncertain streamflows.

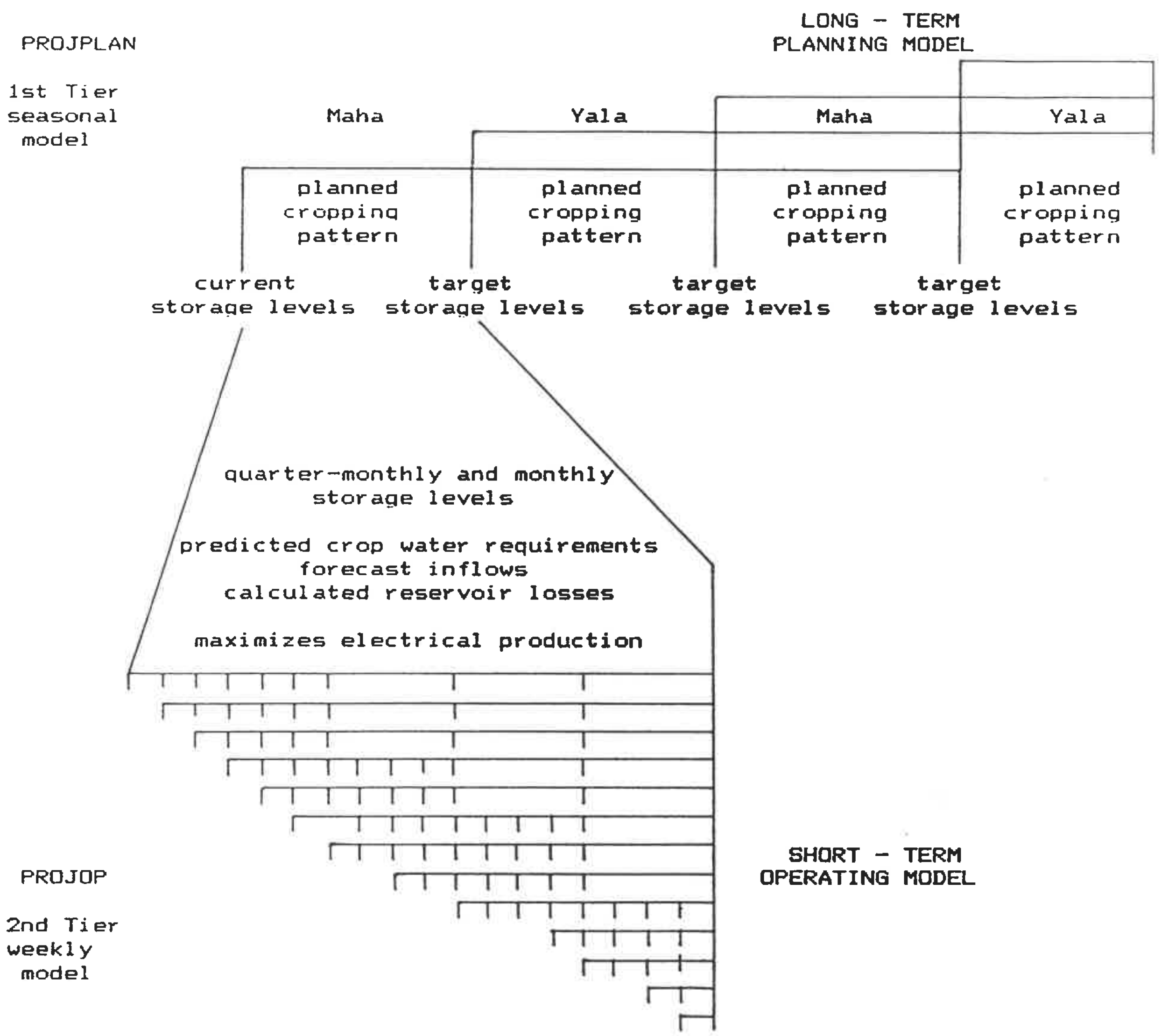

Figure 3.1: Schematic structure of the proposed model

3.8.2 THE DETERMINISTIC LP FORMULATION

The model comprises two linear programming formulations which simplify the reality of the streamflows from five rivers, sixteen reservoirs, five power plants, five inter-basin diversions, and the ten irrigation districts of the Mahaweli System into a set of linear equations. The planning model includes the yields, benefits and production costs of the crops planted in the districts.

The operating model calculates the benefits of electrical production with linearized equations. The optimal release policy for the predicted inflows is then solved by the linear programming algorithm which permits the consideration of a number of time steps, all the reservoirs, diversions, power plants, and irrigation districts, using a standard computerized code. The sensitivity of solutions to the input data is easily analysed. The possibility of future expansion poses no problems for later use of the same model.

The operating policy, which may be verified and improved by a simulation model with optimizing algorithms and detailed nonlinear junctions, is guaranteed to be the optimal solution for the set of linear equations solved. The LP solution policy might not accurately reflect reality, due to the necessary simplifications, and the uncertainty of future inflows, however the solution policy should be as close as possible to the real optimal policy. A sound base policy for further investigations with simulation trials is thus found.

The adaptive deterministic linear programming formulations in a two-tier hierarchial model were developed for the Mahaweli Project in Sri Lanka, but may be easily applied to other multi-reservoir dual purpose systems which place a higher priority on irrigation than on power production. The following chapters discuss each model, preliminary results, and the use of the models in further detail.

CHAPTER 4

THE SEASONAL PLANNING COMPONENT - PROJPLAN

The formulation of PROJPLAN determines how the streamflows expected over the next eighteen months may maximize crop revenues in the irrigation project. PROJPLAN quantifies irrigation benefits, stipulating the reservoir storage necessary to realize them. The seasonal diversions of water and the amount of reservoir storage necessary and feasible for this level of production are determined with a linear programming model because of the large number of crops, districts and time steps. The solution proposes a cropping pattern which maximizes crop revenue for the three seasons under consideration, and determines the reservoir storage and canal diversions required in the irrigation system. A few optional features are incorporated to allow the user to consider various restrictions on the cropping patterns.

While the formulation was developed to determine an optimal cropping pattern, it may also be given a fixed pattern, or a high number of pre-determined crop-hectares. It may be used to determine how much additional land in what crop and in what district may be cultivated this season, or whether it is more worthwhile, in terms of Rupees or some social objectives, to hold the water for irrigation during the next season. The following section presents the mathematical formulation of the model. The names used to identify the constraints and the objective function for the SAS.OR linear programming procedure are also included. The presentation of the formulation follows.

4.1 CONSTANTS AND VARIABLES

The symbols used to present the formulation of PROJPLAN may be subdivided into those whose values are defined by the solution (shown in uppercase) and those whose values are defined by the user as constants in the formulation (shown in lowercase).

4.1.1 Decision Variables

Decision variables are solved by the linear programming formulation. The important decision variables in PROJPLAN are those representing the number of hectares to be planted in a particular crop in each district and those representing the final season reservoir storages. The interseasonal releases and diversions, the corresponding reservoir storages and optional variables for the differences in cultivated land may also be obtained from the solution. The complete list of symbols for these variables and their definitions is given in Appendix C.

4.1.2 User-defined Parameters

The user-defined parameters are constants for a given PROJPLAN formulation. They are the crop values, configuration size, local inflows, and other parameters which define the system. Their values are supplied for the linear programming solution of PROJPLAN. An alphabetical list is included in the appendices.

4.2 OBJECTIVE FUNCTION

The linear programming formulation of PROJPLAN maximizes the revenue from the crops selected by the user or by the formulation. The fundamental function, named OBJ for the SAS.OR solution procedure, decides which feasible solution is the best:

$$\max z = \sum^{I} \sum^{J} \sum^{K} (c_{ijk} H_{ijk}) \qquad 4.1$$

$$c_{ijk} = p_{ijk} - cl_{ijk} - f_{ijk} - w_{ik} \, u_{ijk}$$

where:

I = 3 consecutive growing seasons starting October 1 or April 15

J = number of prospective and possible crops, up to 50 dimensioned in PROJPLAN

K = number of irrigation districts, up to 20 dimensioned in PROJPLAN, currently 14 districts operate in the Mahaweli Irrigation Project

H_{ijk} = decision variable of the number of hectares to be sown in crop j in district k in season i

c_{ijk} = net benefit (Rs/ha) of the crop j in district k in season i

cl_{ijk} = cost of labour (Rs/ha) associated with growing crop j in district k over the entire season i

f_{ijk} = fertilizer costs (Rs/ha) required to grow crop j in district k over the entire season i

p_{ijk} = price (Rs/ha) of crop j grown in district k in season i

u_{ijk} = water requirement (Mm^3/ha) of the crop j in district k over the entire season i, including delivery losses

w_{ik} = cost of water delivered (Rs/1000 m3) to district k in season i. This term may be used to minimize irrigation deliveries to certain districts by increasing their costs relative to the other districts

At the same time, the formulation selects a feasible seasonal or sub-seasonal water allocation. The costs of labour, fertilizer, and water are subtracted from the gross revenue for each crop in each district in each season. Thus, when the formulation is used to find an optimal cropping pattern, the crops which yield the highest net return are chosen as long as the selected constraints are met.

4.3 PROJPLAN CONSTRAINTS

There are two sets of basic constraints in the current formulation: the constraints on the consumption of water, and those on the amount of suitable land.

4.3.1 The Water Balance Restriction

The irrigation water required throughout the season by each crop hectare sown in a district must not exceed the total amount of water forecast as available to that irrigation district in that season, including water remaining in the irrigation tank from the last season. An irrigation district may be supplied by a number of upstream reservoirs, by diversions from other streams, and from return flow from other irrigation districts. The water balance restriction was modeled using continuity equations for every node, whether reservoir, tank or diversion, during each season, because of the complex interrelationships between the components in the project. Subseasons were used to account for varying water demands during the growing season.

Many terms are available in the water-balance equation to model all the modes of water transfer. Not every continuity equation has every term listed here, depending on the physical features of the node. The water balance equations, labeled CNTnSiτ, have the general form:

(REGULATED + NET) INFLOW - OUTFLOW = CHANGE IN STORAGE 4.2

REGULATED INFLOW includes the terms:

$$Q_{i\tau(n-1)} + \sum^{N^1} Q_{i\tau(n+q)} + \sum^{N^2} \alpha_{i\tau(n-p)} QD_{i\tau(n-p)} + \sum^{N^3} \beta_{i\tau(n-m)} QR_{i\tau(n-m)}$$

where:

N^1 = number of tributary nodes whose regulated releases reach the node n

N^2 = number of nodes whose diverted releases flow to the node n

N^3 = number of nodes whose irrigation return flows eventually reach the node n

$Q_{i\tau n}$ = regulated releases (Mm^3) delivered to the districts served by the node n during the sub-season τ of the season i

$QD_{i\tau(n-p)}$ = diversion releases (Mm^3) from a node, n-p, to the node n during the sub-season τ of season i

$QR_{i\tau(n-m)}$ = irrigation releases (Mm^3) from a node, n-m, of which a quantifiable percentage reaches node n during the same sub-season τ of season i

$\alpha_{i\tau(n-p)}$ = diversion canal transport loss as a percent of the seasonal flow, particular to the diversion from the node n-p to the node n, during the sub-season τ of season i

$\beta_{i\tau(n-m)}$ = irrigation return flow percentage of the irrigation turnout at the node n-m which flows to the node n during the sub-season τ of season i

NET INFLOW includes:

$$INF_{i\tau n} - LOSSn_{i\tau n}$$

$inf_{i\tau n}$ = predicted sub-seasonal inflows (Mm^3) from the drainage basin of the node n during the sub-season τ of the season i. The storage at the node n at the beginning of the first sub-season is included in $inf_{1,1,n}$

$loss_{i\tau n}$ = estimated seasonal evaporation and seepage losses from the node n during the sub-season τ of the season i

OUTFLOW includes the terms:

$$Q_{i\tau n} + QD_{i\tau n} + QR_{i\tau n}$$

where:

$Q_{i\tau n}$ = regulated releases (Mm^3) to the downstream node, n+1, or n-q (when n is the last node on a tributary flowing to the node n-q), during the sub-season τ of the season i

CHANGE IN STORAGE includes the terms:

$$S_{i\tau n} - S_{i(\tau-1)n}$$

where:

$S_{i\tau n}$ = storage (Mm^3) at the node n at the end of the sub-season τ of the season i. The storage volume at the beginning of the first season should be included with the predicted inflows at the node n

$S_{i(\tau-1)n}$ = storage (Mm^3) at the node n at the beginning of sub-season τ of the season i. When $\tau = 1$, the storage at the end of the preceding season enters into the calculation:

$$S_{i0n} = S_{(i-1)Ln}$$

where L = number of last sub-season in season i-1

RESTRICTIONS:

The following water-balance variables may be bounded:

$$qmin_{i\tau n} \leq Q_{i\tau n} \leq qmax_{i\tau n}$$

$$QD_{i\tau n} \leq qdmax_{i\tau n}$$

$$SMIN_{i\tau n} \leq S_{i\tau n} \leq SMAX_{i\tau n}$$

$qmin_{i\tau n}$ = minimum downstream sub-seasonal regulated release (Mm^3) allowed from the node n during the sub-season τ of the season i

$qmax_{i\tau n}$ = maximum downstream sub-seasonal regulated release (Mm^3) allowed from the node n during the sub-season τ of the season i

$qdmax_{i\tau n}$ = maximum regulated diversion canal releases (Mm^3) from node n during the sub-season τ of the season i, limited by the canal capacity

$smin_{i\tau n}$ = minimum allowed storage (Mm^3) at node n at the end of the sub-season τ of season i

$smax_{i\tau n}$ = maximum allowed storage (Mm^3) at the node n at the end of the sub-season τ of the season i

Irrigation releases are calculated by the equations labeled WATnSiτ in the SAS.OR solutions:

$$\sum^{J} \sum^{K^1} ul_{ijk\tau}H_{ijk} = QR_{i\tau n} \qquad 4.3$$

where:

K^1 = the list of irrigation districts k which obtain water from the node n, up to 5 irrigation districts form one node

$ul_{ijk\tau}$ = water requirement (Mm^3/ha) of the crop j in district k during the sub-season τ in the season i, including the delivery losses

4.3.2 Land Restriction

No more hectares may be planted in crops than there is suitable land in each district. This physical constraint is expressed in two ways. The number of hectares of a crop grown in a district in a season is restricted to the maximum number of hectares of land considered suitable for that crop by a SAS upper bound constraint:

$$H_{ijk} \leq t_{ijk} \quad \text{for every } i,j,k$$

t_{ijk} = maximum land (ha) suitable for growing crop j in district k in season i.

As well, the sum of hectares cropped in a district must not exceed the total hectares of arable land in that district for that season, the series of constraints named LANDkSi:

$$\sum^{J} H_{ijk} \leq hh_{ik}, \quad i = 1,2,3 \quad k = 1,2,3,\ldots K \qquad 4.4$$

where:

hh_{ik} = total arable land (ha) in the district k in the season i

4.4 OPTIONAL CONSTRAINTS

PROJPLAN also permits the user to include optional constraints to calculate the possible yield from a selected pattern and to ensure that it meets production targets. Also the user may wish to avoid disparate land cultivation between seasons. In this case, additional constraints calculate the difference in the total amount of land being cropped in two seasons so that it may be restricted or minimized. These optional constraints are appended to the fundamental objective function and the two preceding purely physical constraints. An alternative objective function, explained later at the end of section 4.4.2, may be used, requiring constraints which calculate the difference in land unused between seasons.

4.4.1 Production Targets

There are a series of constraints allowing the user to specify levels of production. The user may request that a crop be grown in a district in a season on at least a certain amount of hectares. This is done by simply restricting the crop to a lower bound equal to that amount of hectares:

$$H_{ijk} \geq b_{ijk} \text{ for every } i,j,k$$

where:

b_{ijk} = land (ha) required to grow a given level of crop j in district k in season i, may be zero

In the case of perennial or user-specified crops, the user would set this minimum equal to the maximum suitable land, thus defining the cropping level to be used in the solution.

Crops that have a longer growing season, such as sugar cane, have a separate set of constraints, named LNGjSi for the SAS.OR procedure, ensuring that the hectares planted in one season will be supplied with water for the next three seasons:

$$H_{ijk} - H_{(i+1)jk} \leq 0, \quad i=1,2, \quad j \epsilon J^1, \quad k=1,2,3,...K \qquad 4.5$$

where:

J^1 = list of crops with growing seasons longer than 6 months

This constraint is not really optional. It is superfluous when all the prospective crops are seeded and harvested in one season. Whenever the user wishes PROJPLAN to determine a level of cultivation for sugar cane or other longer growing crops, it becomes obligatory.

A certain level of production of one prospective crop may be desired, but the user may not wish to stipulate how much by district but to allow the linear programming formulation to determine where the crop might best be grown. Then the user specifies that total production from all the districts of a certain crop must exceed a minimum production target in the series of constraints labeled MNCRPjSi:

$$\sum^{K} (y_{ijk}H_{ijk}) \geq ymin_{ij}, \; i = 1,2,3, \; j \in J^2 \qquad 4.6$$

More than one crop may have production targets.

where:

$ymin_{ij}$ = minimum production target (T) for the crop j in the season i, used to guarantee production of certain crops over all the districts

J^2 = list of crops with minimum production quotas in a given season

Alternatively, the user may specify that some combination of crops must meet a pre-defined level. This would apply when varieties of rice are of equal value to the target setters, or when a crop was entered as two because yields or water requirements were not constant at different locations in the same district. These constraints, CCROPjpSi in the SAS.OR procedure, combine the yield from the various crops:

$$\sum^{Jp} \sum^{K} (y_{ijk} H_{ijk}) \geq pcrop_{i,jp} \quad , \ i = 1,2,3 \qquad 4.7$$

where:

Jp = number of prospective crops, j, combined in a group

y_{ijk} = yield (T/ha) of the crop j in the district k in the season i

$pcrop_{i,jp}$ = minimum production level (T) for that combination of crops jp in the season i

More than one combination per season may be included and the same crop may appear in more than one combination.

These optional constraints are not exclusive, the formulation of the planning component may include all the modes of guaranteeing minimum production simultaneously or in any permutation. Other units than tonnes may be used to measure productivity, the yield/hectare and the target yield must, of course, be in consistent units. So must the productivity units of prospective crops whose yields are to be summed to meet a combined crop target. The optional constraints are mutually compatible and append to either objective function and the physical land and water constraints.

4.4.2 Even Land Use

Finally, there are the constraints which calculate the difference in land cultivated in a solution from the total available that season and compare the difference between seasons. This fourth set of constraints is only included when the formulation is deriving an optimal cropping pattern and the user wishes to minimize the occurrence of large numbers of hectares uncultivated in a dry season while a wet season is fully cropped. The constraints are appended to the formulation which may include any of the previously described optional constraints.

Three constraints, labeled DIFi, one for each season, calculate the difference between seasons of the land left idle by the chosen cropping pattern:

$$\left\{ \sum^{J} \sum^{K} H_{ijk} - \sum^{K} hh_{ik} \right\} - \left\{ \sum^{J} \sum^{K} H_{(i+1)jk} - \sum^{K} hh_{(i+1)k} \right\}$$

$$= DD_{(2i-1)} - DD_{2i} \qquad 4.8$$

where:

DD_m = difference in the number of uncultivated hectares in the solution. If the difference is positive, the even index m, (2i), is non-zero and the corresponding odd term is zero. Conversely, if the difference is negative, the odd index m, (2i-1), is non-zero and the corresponding even term is zero.

DEFi equations define the absolute value of the difference:

$$D_i = DD_{(2i-1)} + DD_{2i} \qquad 4.9$$

where:

D_i = absolute difference between seasons of the quantity of land left idle in all the districts

The equation of the last season compares the third to the first. The difference is expressed as the difference of two non-zero terms to obtain the absolute difference. The absolute difference of land which will be left idle from one season to the next may then be restricted or minimized, but not both.

When the seasonal difference in idle land is to be limited to a given quantity of hectares, the absolute difference is upper bounded and this restriction applies:

$$D_i \leq dmax_i, \quad i = 1,2,3$$

where:

D_1 = difference (ha) of the available land not cultivated in season 1 and season 2

D_2 = difference (ha) of the available land not cultivated in season 2 and season 3

D_3 = difference (ha) of the available land not cultivated in season 1 and season 3

$dmax_i$ = limit in hectares of the allowable difference between the season i and the season i+1 of idle land

When the cost of idle land may be quantified in Rs, an alternative objective function may be used, which is still identified as OBJ in the SAS.OR datafile:

$$\max \quad Z = \sum^{I} \sum^{J} \sum^{K} (c_{ijk} H_{ijk}) - wd \qquad 4.1a$$

c_{ijk} is the same as in 4.2, the net benefit/hectare

and

$$wd = \sum^{I} cc_i D_i \qquad 4.11$$

where:

wd = cost (Rs) of disparate hectares left idle

cc_1 = cost (Rs/ha) of land left uncultivated in season 1 v/s the amount left idle in season 2

cc_2 = cost (Rs/ha) of land left uncultivated in season 3 v/s the amount left idle in season 2

cc_3 = cost (Rs/ha) of land left uncultivated in season 3 v/s the amount left idle in season 1

With this additional term in the objective function, the linear programming solution tries to minimize disparate land use between one season and another as well as maximizing the net revenue. The constraints of available water and land, and any desired production constraints, apply in this formulation. Furthermore, the two constraints defining the absolute difference of the land left uncultivated between seasons by a solution appear in the formulation. The datafile preparation program is capable of twenty-four different formulations, combining the basic constraints and fundamental objective function with the optional constraints according to the planning needs and specifications of the user. Some of these formulations are given in Figures 4.1 to 4.3. Table 4.1 shows the possible formulations.

Table 4.1

Table of Possible PROJPLAN Formulations

	Basic	Restricting Seasonal Land Differences	Minimizing Seasonal Land Differences
Basic formulation	1	2	3
Production Targets	4	5	6
Long-growing Crops	7	8	9
Long-growing Crops & Production Targets	10	11	12
Combined Crops	13	14	15
Combined Crops & Long-growing Crops	16	17	18
Combined Crops & Production Targets	19	20	21
Combined Crops & Production Targets & Long-growing Crops	22	23	24

BASIC FORMULATION OF PROJPLAN:

$$\max z = \sum^{I} \sum^{J} \sum^{K} (c_{ijk} H_{ijk}) \qquad 4.1$$

Subject to:

(REGULATED + NET) INFLOW - OUTFLOW = CHANGE OF STORAGE 4.2

for every i, τ, n

$$\sum^{J} \sum^{K^1} ul_{ijk\tau} H_{ijk} = QR_{i\tau n} \qquad 4.3$$

for every i, τ, n
where n supplies an
irrigation node

$$\sum^{J} H_{ijk} \leq hh_{ik} \qquad 4.4$$

for every i, k

$$H_{ijk} \leq t_{ijk}$$

for every i, j, k

Figure 4.1: The basic formulation of PROJPLAN

FORMULATION WITH SOME MINIMUM CROP PRODUCTION:

$$\max z = \sum^{I} \sum^{J} \sum^{K} (c_{ijk} H_{ijk}) \qquad 4.1$$

Subject to:

(REGULATED + NET) INFLOW - OUTFLOW = CHANGE OF STORAGE 4.2

for every i, τ, n

$$\sum^{J} \sum^{K^1} ul_{ijk\tau} H_{ijk} = QR_{i\tau n} \qquad 4.3$$

for every i, τ, n
where n supplies an
irrigation node

$$\sum^{J} H_{ijk} \leq hh_{ik} \qquad 4.4$$

for every i, k

$$H_{ijk} \leq t_{ijk}$$

for every i, j, k

$$H_{ijk} - H_{(i+1)jk} \leq 0 \qquad 4.5$$

for every I, J, K
where crop j
takes 3 seasons
to grow

$$\sum^{K} (y_{ijk} H_{ijk}) \geq ymin_{ij} \qquad 4.6$$

for every i, j, k
where a minimum
amount of crop j
must be grown in
season i

Figure 4.2: A PROJPLAN formulation including crop targets

FORMULATION MINIMIZING LAND DIFFERENCE

$$\max Z = \sum^{I}\sum^{J}\sum^{K} (c_{ijk}H_{ijk}) - WD \qquad 4.1a$$

subject to:

(REGULATED + NET) INFLOW - OUTFLOW = CHANGE OF STORAGE 4.2
for every i, τ, n

$$\sum^{J}\sum^{K^1} ul_{ijk\tau}H_{ijk} = QR_{i\tau n} \qquad 4.3$$

for every i, τ, n
where n supplies an
irrigation node

$$\sum^{J} H_{ijk} \leq hh_{ik} \qquad 4.4$$

for every i, k

$$H_{ijk} \leq t_{ijk}$$

for every i, j, k

$$H_{ijk} - H_{(i+1)jk} \leq 0 \qquad 4.5$$

for every i, j, k
where crop j
takes 3 seasons
to grow

$$(\sum^{J}\sum^{K} H_{ijk} - \sum^{K} hh_{ik}) - (\sum^{J}\sum^{K} H_{(i+1)jk} - \sum^{K} hh_{(i+1)k}) = DD_{(2i+1)} - DD_{2i} \qquad 4.8$$

$$D_i = DD_{(2i+1)} + DD_{2i} \qquad 4.9$$

for every i

$$WD = \sum^{I} cc_i D_i \qquad 4.11$$

Figure 4.3: A PROJPLAN formulation which minimizes the difference in idle land between seasons

4.5 PROJPLAN FEATURES

The PROJPLAN formulation, with its comprehensive allocation of irrigation water throughout the project and its flexible options of alternative constraints provides a valuable planning model for the selection of crops.

4.5.1 Sub-seasonal Time Step

At times, the irrigation demands of one crop selected by the formulation, is not distributed evenly throughout the growing season but a large proportion of water is required at the beginning. This results in a critical period when the tank or reservoir must start the season with enough water to supply the irrigation demand of the first few months, even though there will be surplus water later in the season. To ensure that the weekly operation of the reservoir or tank will be able to supply the initial high demand period, the continuity equations may be calculated more frequently, up to five times in one season, to reflect these critical periods.

The end date of each sub-season continuity equation must be the same throughout the Mahaweli Project, as all the equations are interlinked through the various stream and diversion flows. The user must therefore identify the few dates which are critical for any prospective crop in all the districts. When the critical date is mid-June for some districts and the end of June for others, there will be two sub-seasons, one ending mid-June, one ending at the end of June. Thus the high demand period will be assured of water and the reservoir levels for the end of the season will be realistic given the forecast of water supply and demand.

4.5.2 Equal Land Cultivation Between Seasons

In some years, under some hydrologic conditions and crop prices, water demands and yields, the most revenue may result in a cropping pattern which crops intensely in one season while the subsequent season has considerably less land under cultivation. The social or economic consequences of farmers without irrigation for those undercultivated lands for one season may be unacceptable politically. In that case, the user may include either modification to the model that disallows large differences in the land cultivated between one season and the next. In truly dry years, both seasons may be undercropped but the formulation distributes the idle land more evenly between the seasons.

4.5.3 Irrigation Return Flows

Irrigation return flows may be included in the regulated inflows. The percentage of an irrigation diversion to fields above a given node which arrive in the same growing season from a different node may be calculated in the sum of water available to the node. While this source of water is included in the formulation, its predictability may be questioned. The user may decide to include these terms only for the nodes where long established evidence may be used to quantify the constants of the term.

4.5.4 Variations on Losses and Restrictions

The user may vary the diversion canal losses as a percent of the flow. If a canal loses more water to evaporation or seepage during one hotter or dryer period, the losses may be calculated at that higher rate for only that time. Similarly, the return flow percentage may vary during the growing season or between a Yala and Maha season.

The restrictions on the variables in the continuity equations may differ according to the sub-season. In the case of the downstream or of the diverted releases, whose limits are likely defined as a flow in m^3/s, this is necessary because the time steps are of differing duration. Note that the two monsoon seasons are of different length. The restrictions may not apply to every node with a diversion or a spilling capacity. The user may define these bounds as needed. The storage variables are restricted by the physical capacity of the reservoir and the elevation of release structures, which are unlikely to vary from one season or sub-season to another. The user may wish to vary the limits, however, to ensure flood storage capacity during a critical sub-season, or when a low-level outlet is closed temporarily for repairs.

4.6 SUMMARY

PROJPLAN is a planning model which allows the user to allocate water and crops based on predicted inflows for the next three growing seasons. A number of optional constraints permit detailed crop analysis and combinations to meet production targets. As well the irrigation demands of previously selected crops may be included in the calculations of the water demand. The sub-seasonal time step is included to ensure that the seasonal balance of water accounts for early peaks in demand. The reservoir storages at the end of the first monsoon season are used as storage targets by the short-term operating model PROJOP, described in the next chapter.

CHAPTER 5

THE SHORT-TERM OPERATING COMPONENT - PROJOP

The second component of the model, PROJOP, is also a deterministic linear programming model. The goal of the optimization is to maximize the production of electricity from all the power sites in operation for the rest of the growing season, subject to the higher priority of irrigation uses. Therefore the objective function includes a penalty cost for any occurrence of an untimely irrigation deficit. As well, the reservoir storages required to irrigate the planned cropping pattern of the next growing season, determined by the higher-tier PROJPLAN model, are set as storage targets for the final time step.

5.1 PROJOP FORMULATION

To obtain accurate energy estimates, the constants of the energy production equation may require adjusting. The slope of the energy curve, given by the energy constants as a single straight line, varies with the size of the release as well as with the storage. When the expected releases are grossly over or under estimated, the model will significantly miscalculate the energy produced at a single power plant. When this happens, the constants should be adjusted to more accurately reflect the energy production curve for the releases determined by a previous run of PROJOP with the same inflows and initial reservoir levels. The subsequent run may suggest a completely different release policy, requiring an iterative process before the final solution has more accurately estimated the power production.

To compensate for the uncertainty of streamflows five or six months into the future, the model is repeatedly run with newly predicted inflows entered on a continuous basis. The weekly updating of the projected streamflows determines the best system response to current and anticipated conditions until the end of the current monsoon season.

The linear programming formulation allows for a large number of reservoirs and hydroelectric plants in a complicated system with many time steps considered at one time. The upper and lower bounds of the turbine, diversion, and irrigation flows, and reservoir storage volumes may vary with the time step, reflecting seasonal changes in the physical parameters or operating objectives.

The following pages present the deterministic linear programming formulation: an explanation of the objective function and each of the constraints with the definition of the variable and constant terms and the SAS.OR solution procedure names.

The symbols used to present the formulation may be categorized as those which represent the decision variables (in uppercase) and those which define the current formulation (in lowercase). The complete list of symbols for these variables and their definitions is given in Appendix C.

5.1.1 Decision Variables

The important decision variables are those for the turbine releases, the corresponding energy produced for each power plant, and the releases at the other reservoir and diversion sites for the current step. The solution to the linear programming formulation also yields values for the rest of the season based on the current streamflow predictions, as well as the corresponding reservoir storages, any irrigation deficits and variables necessary for the linearization of more complex functions and the calculation of absolute differences.

5.1.2 User-defined Parameters

The user-defined parameters are constants for a particular formulation at a given time for a given configuration of reservoirs, irrigation districts and canals, and hydroelectric plants. These include the size of the formulation, the irrigation water demands, predicted streamflows and final reservoir storages, the linearization parameters, and the penalty costs of the objective function. Their values must be defined for the linear programming procedure before solution.

5.2 THE OBJECTIVE FUNCTION

The objective function of PROJOP, called OBJ, has at least two sets of terms.

1) The first set of terms defines the power production from all power sites included in the optimization
2) The second set of terms defines the irrigation deficits. The terms assign a penalty for agricultural losses due to untimely irrigation deficits, permitting a complex function using mixed integer programming where desired.

Thus the objective function has the mathematical form:

$$\max z = \mu_1 \sum^{II} \sum^{NP} ENG_{in}$$

$$- \mu_2 \sum^{II} \sum^{Ni} \Omega_{i1n} \; QDF_{in}$$

$$- \mu_2 \sum^{II} \sum^{N^1} \sum^{K^1} \{\Omega_{ikn} D_{ikn} + \Phi_{i(k-1)n} X_{ikn}\} \qquad 5.1$$

where:

II = number of time steps left in the current season

Np = number of hydro-electric power plants modeled in the project

μ_1 = unit price (Rs/GW-hr) of electrical production

ENG_{in} = estimated energy produced (GW-hr) at the time step i at the node n

μ_2 = penalty price (Rs/Mm^3) of the reduced yields due to a unit volume of irrigation deficit

Ni = number of nodes which deliver water to irrigation districts where the losses due to irrigation deficits are considered linear

Ω_{ikn} = the slope of the losses v/s water deficit curve for the crops of the irrigation district at the node n at the time step i due to a deficit of the magnitude of the linearization interval k

QDF_{in} = irrigation deficit (Mm^3) occurring at time step i at the node n

Ni^1 = number of nodes which deliver water to irrigation districts where the losses due to irrigation deficits are linearized by mixed integer variables

K^1 = number of intervals used to linearize the irrigation losses curve.

D_{ikn} = linearization variable, the incremental irrigation deficit (Mm^3) occurring at the time step i at the irrigation node n in the linearization interval k

Φ_{ikn} = the losses (Rs) due to a reduction of yield from a deficit of the magnitude of the minimum level for the linearization interval k at the node n at the time step i

X_{ikn} = integer variable permitting only one interval k to have non-zero values at the time step i at irrigation node n

5.3 THE BASIC CONSTRAINTS

There are two sets of obligatory constraints in the formulation of PROJOP. There are constraints due to the physical properties of the system. As well, the linearization of non-linear functions to increase the model's accuracy requires more constraints.

5.3.1 The Water Balance Restriction

The water - balance equation (continuity), called CNTiNn for the SAS.OR procedure, is written for every node, even those without significant local inflow nor with any storage, and for every time step. The series of equations and terms are very similar to those in PROJPLAN, only there is one index for the time step.

(REGULATED + LOCAL) INFLOW - OUTFLOW = CHANGE IN STORAGE

. . . 5.2

The general form presented above includes the following specific terms which may not all be present at any one node, depending on the physical features of the individual modes:

REGULATED INFLOW includes the terms:

$$QS_{i(n-1)} + \sum^{N^1} \pi_{in'} \; QD_{i(n-p)} + QTB_{i(n-z)}$$

$$+ \sum^{N^2} QS_{i(n-q)} + \sum^{N^3} \pi'_{in'} \; QR_{(i-x)(n-m)}$$

where:

$QS_{i(n-1)}$ = volume of downstream flows from the node n-1 to node n during the time step i

N^1 = list of nodes which divert to the node n

$\pi_{in'}$ = percentage of volume which remains after transporting water in the canal diversion n from the node n-p to the node n at the time step i

$QD_{i(n-p)}$ = regulated diversion releases (Mm^3) from node n-p during the time step i

$QTB_{i(n-z)}$ = turbine volume (Mm^3) from node n-z to node n during the time step i

N^2 = list of nodes which are the last nodes on tributaries to the node n

$QS_{i(n-q)}$ = volume of downstream flows from the node n-q, a tributary to node n, during time step i

N^3 = list of nodes where a percentage of their irrigation deliveries return to the node n at the time step

$\pi'_{in'}$ = percentage of irrigation flow, diverted at n-m (the n'), which will contribute to the regulated inflow of the node n during the time step i

and

$QR_{(i-x)(n-m)}$ = the volume of flow diverted for irrigation at node n-m during the time step i-x, a significant portion of which flows to the node n during the time step i

LOCAL INFLOWS are the forecast flows for each node:

$$inf_{in}$$

where:

inf_{in} = volume of local inflow (Mm^3) in the natural basin of the node n, not including upstream regulated nor diverted flows during time step i. The inflows of the first time step include any reservoir or tank storage, therefore the current storage is added to the local inflow by the data preparation program

OUTFLOW includes the terms:

$$QD_{ikn} + QS_{in} + QTB_{in} + QR_{in}$$

where:

QD_{ikn} = regulated diversion releases (Mm^3) from node n during time step i for the linearization interval k. When the function for the penalty of irrigation losses is linear, k=1

QS_{in} = regulated downstream releases (Mm^3) at time step i at node n that was not used to generate electricity or if there is no power plant at n, the volume of flow to the downstream node

QTB_{in} = turbine volume (Mm^3) during the time step i at the node n

and

QR_{in} = irrigation releases (Mm^3) delivered to the irrigation district from the node n during the time step i

CHANGE IN STORAGE includes the terms:

$$(1 + \sigma_{in}\delta_{kn})ST_{in} \quad - \quad ST_{(i-1)n}$$

where:

σ_{in} = reservoir loss coefficient (mm) used to calculate the evaporation and seepage losses in the tank or reservoir at node n at time step i

δ_{kn} = slope of the surface area - storage curve for the reservoir or tank at the node n in the linearization interval k. When the function of reservoir losses are linear, k=1

and

ST_{in} = storage volume (Mm^3) of the reservoir or tank at the node n at the end of time step i

RESTRICTIONS: The following terms are restricted:

$$qdmin_{in} \leq QD_{in} \leq qdmax_{in}$$
$$qrmin_{in} \leq QR_{in} \leq qrmax_{in}$$
$$qsmin_{in} \leq QS_{in} \leq qsmax_{in}$$
$$qtbmin_{in} \leq QTB_{in} \leq qtbmax_{in}$$
$$smin_{in} \leq ST_{in} \leq smax_{in}$$
$$sf_n \leq ST_{IIn} \quad . \quad . \quad . \quad 5.3$$

where:

$qdmin_{in}$ = minimum regulated diversion canal releases (Mm^3) from node n at the time step i

$qdmax_{in}$ = maximum regulated diversion canal releases (Mm^3) from node n at the time step i limited by the canal capacity

$qrmin_{in}$ = minimum irrigation releases (Mm^3) from node n at the time step i

$qrmax_{in}$ = maximum irrigation releases (Mm^3) from node n at the time step i, limited by canal capacities

$qsmin_{in}$ = minimum downstream releases (Mm^3) from node n at the time step i

$qsmax_{in}$ = maximum downstream releases (Mm^3) from node n at the time step i

$qtbmin_{in}$ = minimum turbine releases (Mm^3) from the power plant at the node n at time step i

$qtbmax_{in}$ = maximum turbine releases (Mm^3) from the power plant at the node n at time step i

$smin_{in}$ = minimum allowed storage (Mm^3) at the node n at the end of time step i

$smax_{in}$ = maximum allowed storage (Mm^3) at node n at the end of time step i

sf_n = target storage (Mm^3) for the end of this monsoon season at node n

ST_{IIn} = storage volume (Mm^3) of the reservoir or tank at node n at the end of the last time step II

5.4 THE DEFINITION OF THE IRRIGATION DEFICIT

The second set of constraints defines an irrigation deficit for every time step at every node where there are irrigation deliveries, in an equation labeled IRRiNn in the SAS.OR procedure:

$$QDF_{in} = cwr_{in} - QR_{in} \quad . \quad . \quad . \quad 5.4$$

where:

cwr_{in} = volume of water (Mm^3) demanded by crops supplied by the node n for the time step i, including transportation losses, a given quantity based on the known cropping pattern and expected evapo-transportation rates for the remainder of the season

5.5 THE LINEARIZATION CONSTRAINTS

The other constraints in the formulation linearize the power function, the reservoir water loss function or the irrigation revenue loss function due to water shortages. Only the power function linearization is obligatory.

Originally, the power linearization used a mixed integer procedure which takes an inordinate amount of computer time when there are many integer variables. Fifty or so integers require one hour of CPU time. Another approach therefore, was developed and successfully implemented.

5.5.1 Linearization of the Energy Produced

The function for energy production is a product of the head and the flow. However the head varies with the flow released in the previous time step of the solution, making the problem non-linear.

Where the head does not vary greatly, a power coefficient based on the average head is sufficiently accurate. At the power sites with highly variable heads, and high heads at that, the energy produced is not accurately estimated by a single power coefficient based on an average head.

Not only is the estimation inaccurate but the release policy chosen as optimal will not reflect the advantages of releasing water from a fuller reservoir. More energy will be generated when more water is released from a reservoir at a given level. The subsequent time step would then have lower reservoirs levels, thus the same release in the later time step will generate less electricity. If a single power coefficient is used the model will still estimate the same amount of production. Using a second or third run with power coefficients adjusted to the reservoir levels of the previous solution will increase the accuracy of the estimates. An alternative policy, which increases reservoir levels by initially releasing less, yet thereby produces more energy over the duration of the monsoon season, would be overlooked. Wherever the reservoir levels above a power plant fluctuate significantly, a single power coefficient based on an average head is not sufficient to find an operating policy optimal for the entire season.

Mixed integer linearization procedures using differing power coefficients to multiply the turbine releases depending on the reservoir levels were developed for PROJOP. However, the branch and bound solution of mixed integers increased computer time geometrically. Complete studies of the operation of reservoirs with large variations in the head, were only possible one site at a time and for a third of a season. Only four possible power coefficients at one site were feasible for a run at the beginning of a season.

The branch and bound procedure did not allow an adequate investigation of the optimal operating policy. Therefore another linearization approach to the estimation of power generation (Sigvaldason [1987]) was incorporated into the model. The energy function was approximated by splitting the function into three terms for those power sites where an average head calculation was insufficient.

This approach uses only one constraint, named EPiNn, for each time step i and for each node n with a power plant estimated by a variable head, to define the amount of energy produced in a time stem:

$$ENG_{in} = h_{in}\ QTB_{in} + c_{in}\ (0.5ST_{in} + 0.5ST_{(i-1)n} - sto_{in}) \qquad . \quad . \quad . \quad 5.5$$

where

h_{in} = power coefficient (GW-Hr/Mm^3) for a representative head in the operating range of the reservoir n during time step i.

c_{in} = correcting factor (GW-hr/Mm^3) used to adjust the estimate of energy production according to the average volume of water stored at the node n at time step i multiplied by an energy factor.

and

sto_{in} = representative storage volume (Mm^3) of the reservoir at the node n for the time step i

Both the power coefficient, h_{in}, and the correcting factor, c_{in}, differ with the duration of the time step. They incorporate the duration of the time step, the density of water and the efficiency of the turbines in their derivation.

Figure 5.1 shows the curvilinear function of power generation as a function of head and at various release amounts, superimposed with two dashed lines representing the straight-line approximation of the three-term linearization used in the formulation. Varying c_{in} adjusts the slope of the power line to the curve representing the actual power produced by the release.

The first term of the power approximation multiplies a head representative of the operating range by the flow variable. Where the head at a power plant does not vary greatly, where the reservoir fluctuations are small, the first term alone is sufficient to estimate the power production: c_{in} equals zero.

Otherwise, the second and third terms correct the estimation of the first term according to whether the average storage variable in the time step was above or below a constant reference volume. The greater the difference, the greater the proportional correction. The correcting terms would reduce the value calculated in the first term when the average storage level in a time step was below the constant reference volume.

The magnitude of the turbine flows is necessary to quantify the constants of the second and third terms as is seen by the poor estimation of the energy produced by a release of 200 Mm^3 with a correcting factor, C_{in}, estimated from a flow at 25 Mm^3. A detailed account of the estimation and refinement of the power estimation constants is presented in subsequent chapters.

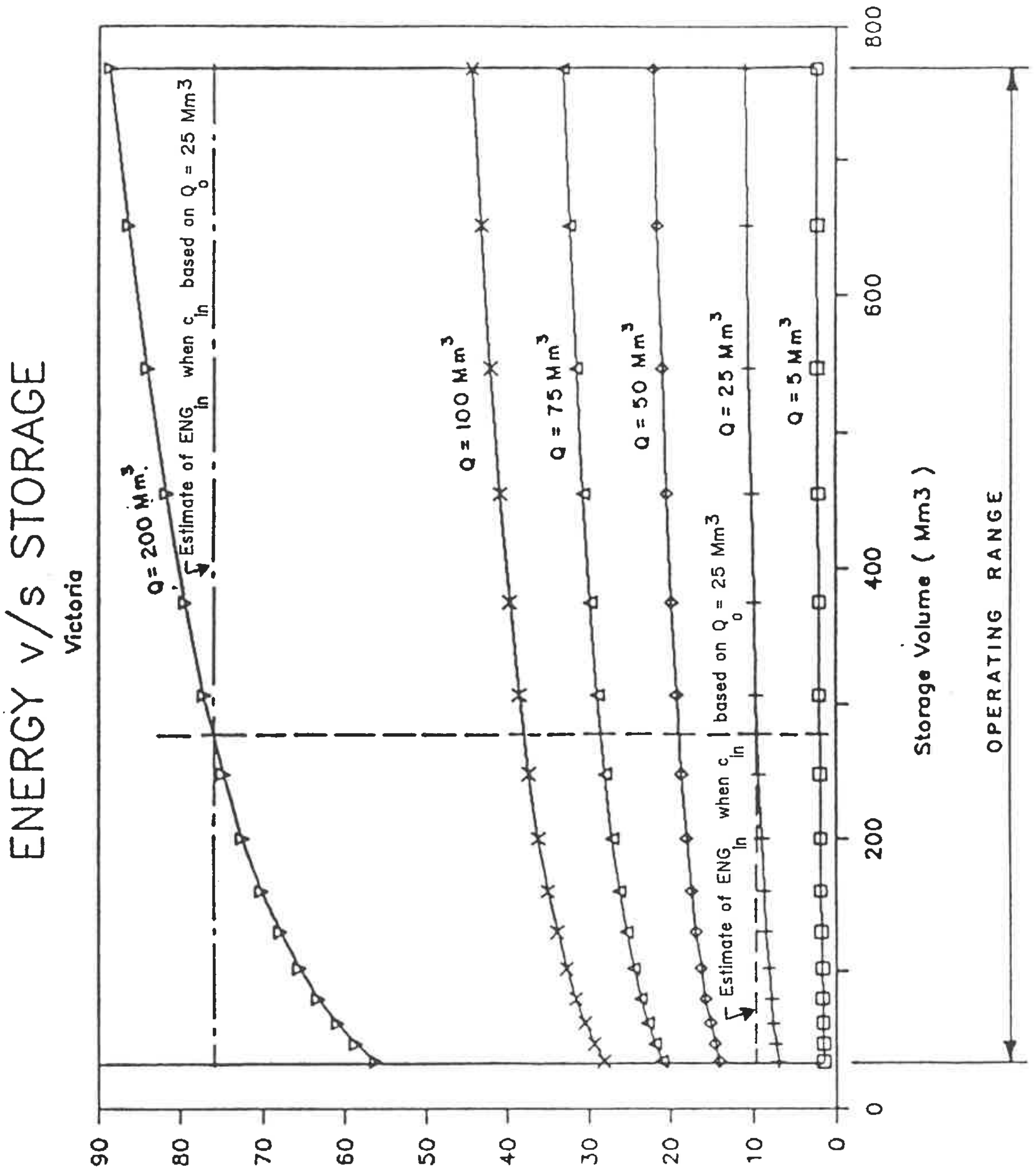

Figure 5.1 Straight line estimates of the energy function

5.5.2 Mixed Integer Linearization

The original linearization method, which is not included in the current version of PROJOP, is presented here for comparison. Five constraints are required to linearize the electrical production as a function of the water released (QTB) and the head (a function, in turn, of the average reservoir storage, $0.5ST_{in} + 0.5ST_{(i-1)n}$, during the time step). Figure 5.3 shows the approximation graphically. The linearized function appears as:

$$\sum^{K} (\beta_{ikn} \, Q_{ikn}) \qquad \text{a.1}$$

where:

K = number of intervals used to linearize the power curve

β_{ikn} = average power coefficient (GW-hr/Mm^3) which calculates the amount of power generated at time step i at power plant n for the average storage in the interval k

and

Q_{ikn} = linearization variable, the turbine release at time step i at the power plant n in a linearization interval k

During every linearization interval k at the node n in the time step i there is the constraint, FiNnEk:

$$99999\ Z_{ikn} - Q_{ikn} \geq 0 \qquad \text{a.2}$$

where:

Z_{ikn} = integer variable used to select one interval k at the time step i at the node n for the linearization of the electrical power curve

The constraint, called ZiNnEk in the SAS.OR data file, also exists for every time step i and the node n for every linearization interval k:

$$(\alpha_{i(k+1)n} - \alpha_{ikn})\ Z_{ikn} - S_{ikn} \geq 0 \qquad \text{a.3}$$

where:

α_{kn} = maximum storage volume (Mm^3) in interval k at the power plant node n

and

S_{ikn} = linearization variable, the incremental storage (average ST for the time step i- minimum storage of the interval k) at power plant node n

For every time step i at the node n there are three more constraints. Their names for the SAS.OR procedure are listed to the left:

QiNn : $\sum^{K} Q_{ikn} - QTB_{in} = 0$ a.4

ZiNn : $\sum^{K} Z_{ikn} \leq 1$ a.5

SiNn : $\sum^{K} (\alpha_{i(k-1)n} Z_{ikn} + S_{ikn}) - 05(ST_{in} - ST_{(i-1)n}) = 0$ a.6

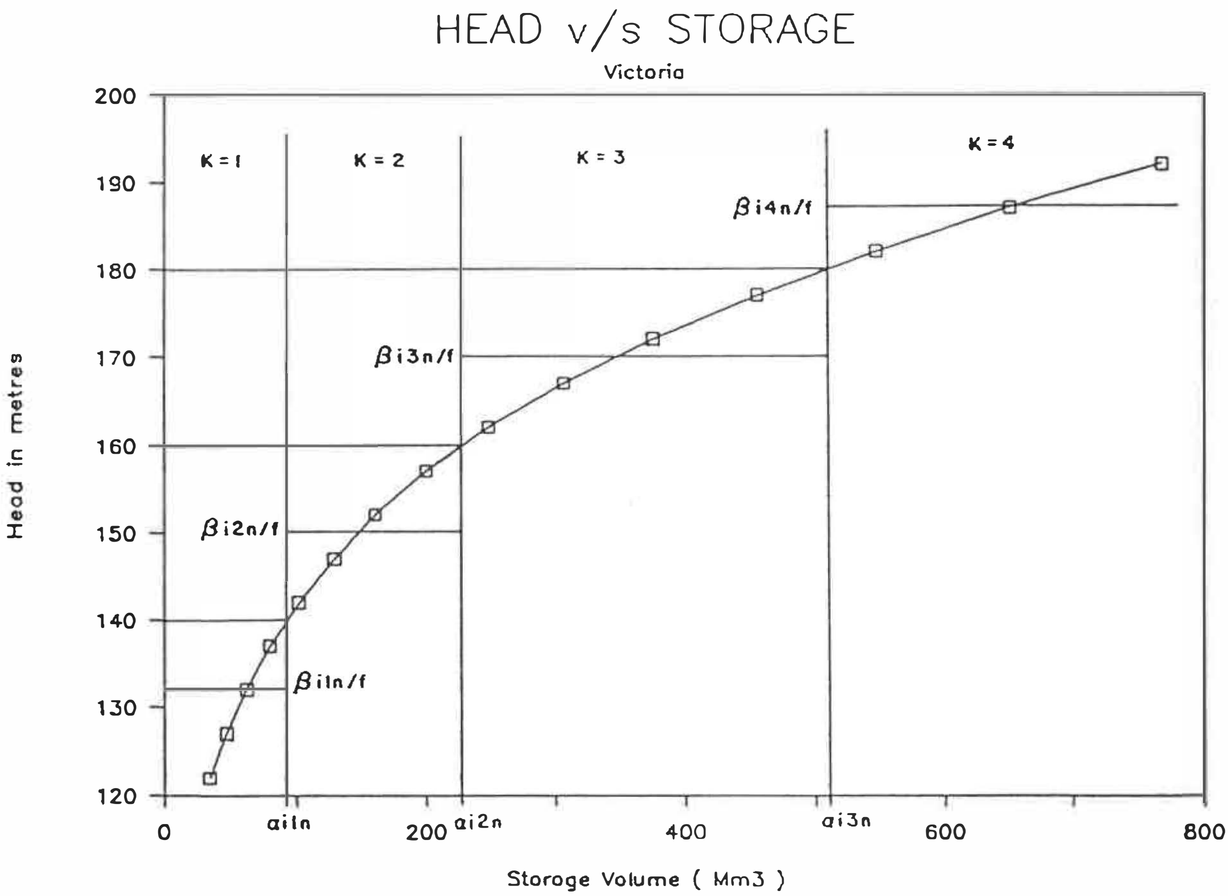

Figure 5.2 The mixed integer estimate of energy production using the head versus storage curve

5.5.3 Linearization of Reservoir Losses

Three constraints linearize the evaporation and infiltration losses in reservoirs as a function of the surface area (a function of the final storage volume of the time step) and the independent monthly loss coefficient. The linearized function replaces the CHANGE OF STORAGE terms in the continuity equation with:

$$ST_{in} - ST_{(i-1)n} + \sum^{K^3} \sigma_{in} \, (\epsilon_{kn} \, Y_{ikn} + \delta_{kn} \, T_{ikn}) \qquad . \quad . \quad .5.6$$

where

K^3 = number of intervals used to linearize the surface area - storage curve of the reservoir or tank

ϵ_{kn} = minimum surface area (km^2) of the reservoir or tank at the node n in linearization interval k

Y_{ikn} = integer variable used to select one interval k at the time step i at the node n for the linearization of the surface area - slope curve

T_{ikn} = linearization variable, the incremental storage (ST_{in} - minimum storage of the interval k) at the end of the time step i at the reservoir or tank at node n

During every linearization interval k at the node n in the time step i there is the constraint, labeled YiNnEk:

$$(\Gamma_{(k+1)n} - \Gamma_{kn})\ Y_{ikn} - T_{ikn} \geq 0 \qquad . \quad . \quad . \quad 5.7$$

where:

Γ_{kn} = maximum storage volume (Mm^3) of the interval k at the reservoir or tank n

For every time step i at the node n there are also two more constraints which are named RiNn and YiNn for the SAS.OR procedure:

$$\text{RiNn :} \quad \sum^{K3} (\Gamma_{(k-1)n} Y_{ikn} + T_{ikn}) - ST_{in} = 0 \qquad . \quad . \quad 5.8$$

$$\text{YiNn :} \quad \sum^{K3} Y_{ikn} \leq 1 \qquad . \quad . \quad . \quad 5.9$$

Figure 5.3 shows the approximation graphically:

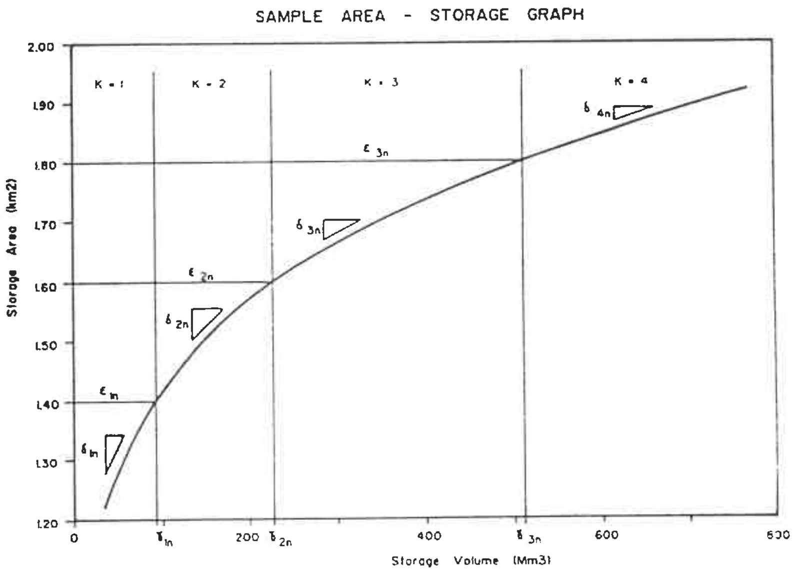

Figure 5.3 Mixed integer estimation of the surface area

5.5.4 Linearization of Irrigation Deficits

There are also three constraints in the current version of PROJOP which use mixed integer linearizations to more accurately calculate the losses in yield due to irrigation deficits when the user wishes to do so. The linearized function in the current version of PROJOP appears as:

$$\sum^{K^1} \Omega_{ikn} \; D_{ikn} + \Phi_{i(k-1)n} \; X_{ikn} \qquad 5.10$$

During every linearization interval k at the node n in the time step i there is the constraint, labeled DiNiEk:

$$(\Theta_{i(k+1)n} - \Theta_{ikn}) \; X_{ikn} - D_{ikn} \geq 0 \qquad 5.11$$

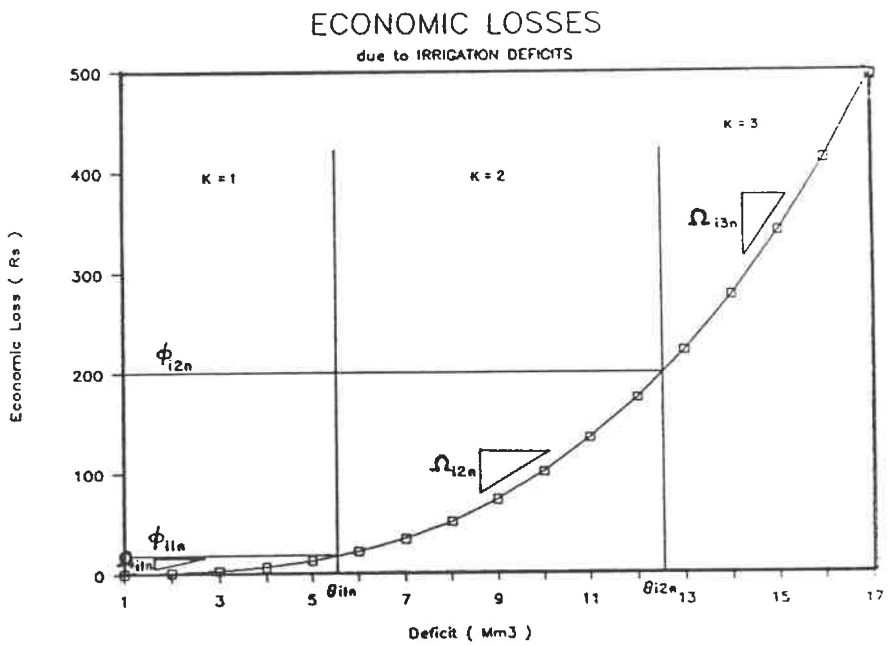

Figure 5.4 Mixed integer estimation of the costs of water deficits

For every time step i at the node n there are two more constraints, shown with their SAS.OR names to the left:

$$\text{WiNn}: \quad \sum^{K^1} (\theta_{i(k-1)n} \; X_{ikn} + D_{ikn}) - QDF_{in} = 0 \qquad 5.12$$

$$\text{XiNn}: \quad \sum^{K_1} X_{ikn} \leq 1 \qquad 5.13$$

5.6 OPTIONAL FORMULATION

In wet forecast years, when inflows in the current season are greater than first forecast, and the benefits to the next season of reservoir levels higher than the targets from PROJPLAN are quantifiable, the user may wish to assign a premium to higher final levels. An optional set of terms for the objective function allows for solutions which increase the amount of water stored in reservoirs for the next growing season.

A solution which generated a large amount of electrical power in the final month of the growing season, and scarcely any at the beginning, may generate a lot of energy in total, but would not be as satisfactory a solution as one where the monthly totals of energy produced were more consistent. Another set of terms in the objective function and some additional constraints, may be incorporated in the formulation to smooth the fluctuations in total energy produced from one time step to another.

5.6.1 Optional Terms in the Objective Function

The two additional sets of optional terms which may be included in the objective function are:

$$+ \sum^{Nr} \mu_{3n} \; ST_{IIn} \qquad \text{(optional)}$$

$$- \mu_4 \sum^{(II-1)} DIFi \qquad \text{(optional)} \qquad . \; . \; . \; 5.1a$$

where:

Nr = number of nodes which have significant storage

μ_{3n} = unit price (Rs /Mm^3) which can be assigned to the agricultural or other benefits of a reservoir n storing more water at the end of the growing season than stipulated by PROJPLAN. When equal to zero, future benefits of additional water are ignored

II = number of time steps left in the season

μ_4 = an arbitrarily assigned unit price (Rs/GW-hr) for a difference in power production between one time step and another, chosen not to unduly influence the power production.

DIF_i = absolute difference (GW-hr) between the power produced in one time step and the next

The first optional set of terms maximizes benefits due to higher reservoir storage volumes at the end of the season. The benefits may be set to zero and this term disregarded. The second optional set of terms, which may appear in the objective function of PROJOP, minimizes the fluctuations in power generated from one week to the next.

The terms are independent: one may appear in the formulation without the other, or both may appear. The original first set of terms must appear. The inclusion in the objective function of the set of terms to minimize power fluctuations involves the addition of the constraints which define the quarter monthly and monthly variation variables.

There are thus four possible objective functions:

1. The basic objective which maximizes the energy production and weights the penalties of irrigation deficits alone;
2. The basic objective with an additional set of terms to increase storage at the end of the season;
3. The basic objective with an additional set of terms to even energy production;
4. The basic objective function with both optional sets of terms.

5.6.2 Optional Constraints

Also, the user may include optional constraints to incorporate particular energy policy objectives. Two optional series of constraints restrict the energy production in a time step to obtain an even production rate throughout the season, each in an alternative fashion. There are the constraints, named POWi, which guarantee a minimum production of electricity in each time step:

$$\sum^{Np} (ENG_{in}) \geq pmin_i \quad . \quad . \quad . \quad 5.14$$

where:

$pmin_i$ = minimum total energy production (GW-hr) required at the time step i

Alternatively, or in conjunction with the energy limits, the difference between the total amount of energy produced in one time step and the next may be calculated by the equations labeled PDFi and PPPi for the SAS.OR procedure:

$$\text{PDFi}: \quad \sum^{Np} ENG_{in} - ¥(\sum^{Np} ENG_{(i+1)n}) = DIFP_i - DIFM_i \quad . . 5.15$$

where:

¥ = 4, when i is the last quarter month time step before the final monthly time steps when optimizing at the start of the season, otherwise, it is equal to 1

$DIFP_i$ = positive difference between the total power produced at the time step 1 and that produced at the time step i+1

and

$DIFM_i$ = negative difference between the total power produced at the time step i and that produced at the time step 1+1

$$\text{PPPi}: \quad DIFP_i + DIFM_i = DIF_i \quad . . . 5.16$$

The variable DIFi may be minimized in the objective function or restricted:

$$DIF_i \leq DIFMAX_i \quad . . . 5.17$$

where:

$DIFMAX_i$	=	maximum allowed difference (GW-hr) in energy generated between the time step i and i+1

The two terms, DIFP and DIFM, are necessary to define the absolute value.

5.7 PROJOP FEATURES

The formulation of PROJOP, with its linearized functions, optional electrical production restrictions, and quantification of irrigation deficits, provides a versatile operating model.

5.7.1 Rationing under Drought Conditions

The second set of terms in the objective function, assigning a penalty to any irrigation deficits, enables the linear programming formulation to find an optimal release policy even in expected dry conditions. Under anticipated flow conditions, the high penalty cost of the deficits will prohibit the model from selecting release policies which allow irrigation deficits. Without including the possibility of deficits, when the forecast water supplies were unable to meet the irrigation demands of already planted crops, the user would obtain infeasible solutions which do not suggest any policies.

By weighting the penalty costs for each irrigation district and for the critical weeks in the growing season, the user may obtain release policies which will minimize the overall irrigation damage under drought conditions.

The penalty for an irrigation deficit may be estimated by a varying rate using the mixed integer linearization technique included in the data-file preparation program. Its use is restricted to the few districts or critical periods where the penalty per Mm^3 of water deficit varies with the size of the deficit. When this is the case too often, the number of integer variables required will be too many for quick solution.

Another method, (Norrie [1987]), which does not use mixed integer programming, exploits the concave nature of a cost function and the simplex method choice of variables with lower costs before those with higher costs. When the cost associated with high deficits is higher than those associated with a lower deficit, the simplex method will choose the deficit variable for an irrigation district associated with a lower cost before the deficit variable with a higher cost unless constrained by the upper limit of the lower cost deficit variable. It will not select more than one deficit variable a district as non-zero, under these conditions. To use this method, the objective function has these terms for the irrigation deficit:

$$\sum^{II} \sum^{Ni^2} \sum^{K^2} \Omega_{ikn} \quad D_{ikn} \qquad \text{b.1}$$

where:

Ni^2 = number of nodes which deliver water to irrigation districts where the losses due to irrigation deficits are linearized by the second method

K^2 = number of intervals used to linearize the irrigation losses curve using the second method

Constraints of the following form are needed to define the deficit variables for each time step and each irrigation node:

$$\sum^{K^2} D_{ikn} - QDF_{in} = 0 \qquad \text{b.2}$$

Each D_{ikn} has an upper limit equal to the magnitude of the deficit at the end of the linearization interval minus that at the beginning. In the linearization interval the costs of the deficit are a constant rate. This second method is not included in the current version of PROJOP.

5.7.2 Irrigation Return Flows

Note that the irrigation return flows are included in the regulated inflows. The percentage of an irrigation diversion in the current or previous time step from a different node which returns to fields above a given node may be calculated in the sum of water available to the node.

While this source of water is included in the formulation, its predictability may be questioned. The user may decide to include these terms only for the nodes where long established evidence may quantify the constants of the term. The time lag and the percentage may be varied with the time step in the growing season.

5.7.3 Variation of the Parameters

The linear programming formulation allows the operating bounds from one time step to the next to be independent. Reservoir storage bounds may be reduced or increased for flood storage provision, repairs or temporary operating conditions. The other bounds are variable with the time step because of the varying duration. When a turbine is scheduled for repairs, the amount of flow which may generate power would be reduced. As well, the user may vary the diversion canal losses as a percent of the flow. Whenever predictably more water in a canal evaporates or seeps during a hotter or dryer period, the losses may be calculated at the higher rate for only that time. By varying these limits the user ensures the formulation is modeling the real life operating conditions of the system.

5.7.4 Definition of the Energy Produced

The energy produced at one power plant during one time step is a decision variable in the formulation. This is useful because it allows quick and ready identification of electrical production. Also the decision variable may be subjected to upper or lower bounds and included in the objective function and the constraints, either individually or summed by time step or location to meet differing policy objectives.

5.7.5 Increasing Final Storage Levels

When the hydrology of the current season evolves as wetter than first expected, the user may search for a release policy which is not only better for this season, in terms of irrigation and electrical benefits, but also for the next season. In tandem with PROJPLAN tests for the next seasons, the optional storage terms may be included in the objective function of PROJOP. The user may alternatively vary the target storage volumes, the SF_n bounds, to quantify the effect on this season's electrical and irrigation benefits, and, by using those volumes for PROJPLAN starting levels, to quantify the effect on the projected agricultural revenue from the Mahaweli Project during the next season.

5.7.6 Continual Updating of Adaptive Planning

The operating policy determined by the model is continually updated at each time step, allowing the model to respond quickly to both equipment maintenance scheduling and to less predictable hydrological occurrences. The short-term policy is determined by the model each week and is not a predetermined rule curve nor a weekly release rule. The actual current reservoir levels, revised streamflow predictions, and any revisions to the weekly crop water requirements, are likely to shift the policy from the anticipated policy of the quarter month before. The reservoir operator releases only the amount recommended by the model for the next time step. Subsequent runs are performed before determining the actual releases of future time steps. This allows the model to respond to updated current reservoir levels and streamflow forecasts.

While applicable for a short term of one time step, a quarter month, the reservoir release policies were derived from consideration of initial water levels, of deterministic forecasts of the streamflows available, and of water demands for an entire growing season, and even a year and a half ahead, because of the included storage targets from the planning model. As expectations of the hydrological conditions change throughout the season, the series of policies are based on a continually mutating set of conditions. They do not result in the actual optimal policy. The operator is unable to change an earlier decision based on current knowledge of events in the immediate past, but is forced to make the best decision for the short-term based on current conditions, predictions for the future, and the storage goals of the irrigation planning step. The optimal policy determined with full hindsight or perfect foreknowledge will be better, but that knowledge does not occur in real life. The continually updated policy of the short-term operating model ensures that the operating policies are the best, based on the current state and the best estimates of future conditions for the objectives defined.

5.7.7 Use as a Regulatory Guideline

PROJOP finds an optimal release policy for the next quarter month for the regulation of the reservoir system as defined by the linear programming formulation. The objectives and the flow restrictions will have been chosen to reflect the goals of the regulating authority for that growing season. The model reduces the complex problem of reservoir regulation to simple linear equations; this entails simplifications and inaccurate estimates. The energy production the model has calculated from a release policy may be infeasible due to over-estimation or because of hourly operating constraints overlooked by the model. The quarter monthly delivery of water to a district may be impossible because of the hydraulic constraints of reservoir levels. The Ceylon Electrical Board models would have to determine the daily and hourly release schedules based on the releases recommended by PROJOP.

While the policies of the continually updated solutions may be the best, given the known and expected hydrological conditions for the series of linear constraints of a linear programming formulation, they will not be optimal for the real life situation of the Mahaweli Project of Sri Lanka when too many simplifications and approximations of real parameters were necessary in the modelling. PROJOP locates the optimal peak of the operating surface defined by its parameters. If the model has realistic constants and accurate flow predictions, a simulation model, that includes all non-linear relationships and evaluates the cost and returns of water use in the irrigation and hydro-electric components of the project, may verify the feasibility and optimality of the release policy determined by PROJOP.

When the uncertainty of flow prediction is high, PROJOP may be executed with various streamflows to identify the penalties and benefits of other streamflow sequences. Trials with various streamflow, crop water requirements, and electrical demands may also be performed to evaluate the penalties and benefits associated with different probable scenarios and thus somewhat quantify the risks.

The optimal solution to the linear programming problem of PROJOP may serve as a base policy. A simulation model with optimizing features may then find a better solution to a more detailed formulation of the reservoir regulation, having been given a base policy close to the global optimum of the real life situation. The confidence of the operator in the release policy recommended by the short-term model increases when the results are verified by simulation.

5.8 SUMMARY

The PROJOP formulation maximizes the production of hydro-electricity subject to the irrigation demands of already planted crops and the target storages of the next growing season. Restrictions on the production of electricity may be necessary to balance the production throughout the growing season, while wetter conditions than forecast may make higher final reservoir storages desirable.

The electrical production function is linearized in such a way that the loss in accuracy is minimized and, more importantly, that possible beneficial policies are not overlooked. Reservoir storages losses and the penalties of irrigation losses may be also more accurately modelled with mixed integer linearization, although with a severe penalty in increased computing time.

The model requires updating every time step and its policies should be tested with a simulation model before their implementation. The formulation of PROJOP may be solved with the linear programming package SAS.OR. With the large number of reservoirs, irrigation districts and hydro-electric plants involved the datafile is large and unwieldy.

A program was written for an IBM PC to prepare the data necessary for a SAS.OR solution program. The next chapter details the data preparation programs for both PROJOP and PROJPLAN. Chapter 7 discusses applications of the models.

CHAPTER 6

THE IMPLEMENTATION OF THE MODEL BY COMPUTER

The solution of the two formulations in the model, each with hundreds of constraints and hundreds of variables, requires the use of a computer. A number of linear programming solution procedures are available commercially, most for use on a mainframe computer. The SAS.OR package [1986] from the SAS Institute was used to solve the linear programming formulations at the Ecole Polytechnique. The SAS.OR linear programming package has a special feature for the entry of sparse constraint sets, requiring only the entry of non-zero values. The other commonly known SAS features, such as combining, sorting, and statistical analysis of data files are supported.

A version for micro-computers is being developed, which may or may not be large enough to incorporate the large number of variables and constraints. The number of variables and constraints differ in PROJPLAN because the number of prospective crops and the number of critical sub-seasons may differ. PROJOP problems have continually less variables and less constraints as the number of time steps left in the growing season diminish. At the present time, the actual solution of the linear programming problem is done on an IBM mainframe computer.

The data file required by the SAS.OR package, while simpler than some, is lengthy and involved. To save mainframe computer costs, two interactive FORTRAN programs were written for use on an IBM compatible personal computer to prepare the PROJPLAN and PROJOP problems for solution. The data files, written to a floppy diskette, are transferred to the mainframe where the SAS.OR Linear Programming package solves the problem. Before submission, the user may wish to modify the data files with user-supplied constraints using the combining features available with SAS. The results of the SAS run are the final solution. The computer implementation entails firstly the preparation of a SAS.OR LP data file, performed interactively on a PC, and secondly the batch submission of the data file to the SAS.OR LP procedure on a mainframe computer.

6.1 THE PERSONAL COMPUTER FILES

All the micro-computer files are in ASCII format, sequentially accessed, and are referred to as OLD within the FORTRAN programs. To be opened, the file name of the file must already exist on the diskette. When reading files, this is certainly the case, if not the user has supplied the wrong diskette or the wrong name, the file may not be read in either case. When overwriting a previous file, such as after correcting or adding to the information stored, again the file would be OLD. However, when creating a file, or saving a separate version, the user must request the data preparation program to write to a file that already exists on the diskette, if in name only. Before a run, the user should open a file with the desired name, save a blank file, verify that the name appears in the diskette directory, and then commence the preparation program.

Tables 6.1 and 6.2 list the files required by each formulation and approximate sizes (to the nearest 500 bytes) based on the files used in the test runs conducted. Where the name is given the file must appear in the drive specified and with that name. All the files may be corrected by the user with the interactive program before the data is used to prepare the SAS.OR data file. The data preparation programs are user-friendly, not only quick and easy to use, but also facilitating data corrections and corrective actions for false moves.

TABLE 6.1

Files required by PROJPLAN

	FILE	NAME	ADVANCES	PREPARED INTER-ACTIVELY	OBLIGATORY *	LIKELY SIZE (BYTES)
1	Crop and District	b:Crops	y	n	y	10 000 to 20 000
2	Con-figura-tion	u-s	n	y	y	1 000
3	Inflows	u-s	#	y	y	2 000
4	Crop Water Demands	u-s	y	y	n	3 000
5	Vari-ations	u-s	y	y	n	2 500
6	SAS Data-file	b:Plan.dat		y	y	20 000

* The files are obligatory as their data is necessary. The user may enter data interactively, without saving it on a file, therefore the inflow and configuration files may be considered not strictly obligatory.

\# To be updated with new estimates each run

u-s = user-specified drive and filename y = yes n = no

TABLE 6.2

Files required by PROJOP

FILE		NAME	ADVANCES	PREPARED INTER-ACTIVELY	OBLIGATORY *	LIKELY SIZE (BYTES)
1	Con-figur-tion	u-s	n	y	y	1 500
2	Power plants	u-s	n	n	y	1 500
3	Reser-voirs	u-s	n	n	y	2 500
4	Irri-gation Losses	u-s	n	n	y	1 500
5	Irri-gation Demands	u-s	#	n	y	2 500
6	Canal Charac-teristics	u-s	n	y	y	1 500
7	Vari-ations	u-s	y	y	n	10 000
8	Inflows	u-s	#	n	y	25 000
9	Return Flows	u-s	n	y	n	1 000
10	SAS Data-file	b:Release.dat		y	y	20 000

* The files are obligatory as their information is necessary.

To be updated with new estimates each run

u-s = user-specified drive and file name y = yes n = no

6.2 THE PREPARATION OF A PROJPLAN RUN

PROJPLAN looks at the water available in the next three growing seasons and determines a cropping pattern and an allocation of water on a seasonal or sub-seasonal basis to the irrigation fields, the reservoirs, and the canals. The formulation has variables for the regulated stream and canal releases, the reservoir storage volumes, and the crops, both definite and prospective. PROJPLAN needs to also define the constraints which will apply to the run, and include the proper variables in each. The quick entry of codes selecting the optional constraints is done interactively, as may be the entry of inflow and configuration data; however, files are used to enter more involved and permanent data.

6.2.1 District and Crop Data

The file B:Crops, which must be read from the B drive at the beginning of the data preparation program, is a lengthy file. It contains the crop and district details and any production minimum limits. First the number of crops and districts appearing in the file is given, and the constants for any desired restriction on interseasonal land utilization limits. Any minimum production limit and the growing season code are entered at the beginning of the crop information with the crop name. The information of the definite and prospective crops is entered by crop, by season and by district. The price for a crop may change from one season to another; labour costs may vary from one district to another; the water use may differ for crops in the same district. For every planted and prospective crop, the per hectare price, yield, labour and fertilizer costs, and water use plus the limits on the land are entered on a line for every season and district. The seasonal water use includes the transportation losses incurred by transferring the water from the delivery node to the planted crop. The maximum suitable and any stipulated minimum amount of land for a crop or crop variety in a district is also given by crop, by season, and by district. Varieties of a crop within a district may be entered as separate crops when there are significant differences in the yield, costs, water uses, etc.

Note that if the maximum and minimum limits are identical, the crop is definitely planted, whether a perennial or user-selected crop or a crop in progress. For a crop, such as sugar cane, which grows for more than one or two monsoon seasons, a long-growing season code is assigned which applies in all districts. PROJPLAN is then restricted to a cropping pattern which allocates the same number of hectares to the crop in all subsequent seasons. The Crops file also lists the district information, the size of water delivery by district. The array used to combine certain crops to meet minimum production constraints appears at the end of the Crops file.

The seasonal information in the file may be advanced one season forward by the data preparation file. Corrections or changes may also be performed and the new file saved, if desired, under an existing alternate file name. New districts or prospective crops may be added and others deleted with the interactive program, but for complete entry of the crop and district data, a text editor would be more efficient. The arrays for the combination of crop varieties, however, would be difficult to code by hand while quickly entered with the program once the crop and district data are correct.

6.2.2 Reservoir Network

The model requires minimum and maximum reservoir levels, and a listing of the reservoirs and tanks and how they are connected to the streams, diversions, and irrigation districts. The seasonal capacities and losses of the diversion canals are also entered. If a significant amount of return flows from irrigation in one district is later available to the system, the user may store the percentage of flow and the list of nodes whose irrigation diversion is expected to contribute later in the season. The file will also store minimum or maximum flow restrictions, where required. The information on the reservoir and irrigation network is easily entered directly to the PROJPLAN program by answering a series of questions as prompted. The information may then be saved for the next time under a previously existing diskette file name.

6.2.3 Crop Water Demands and Inflow Forecasts

The seasonal or sub-seasonal crop water requirements and inflow forecasts may be entered interactively or by reading two prepared files. Both files must agree on the number of sub-seasons in each season with the numbers entered at the beginning of the data preparation program. When there are no sub-seasons identified, the seasonal water use per hectare given in the Crops file will be used and the sub-seasonal data is unnecessary. The crop water requirement file lists by crop, by district, and then by sub-season the amount of irrigation water required per hectare by the crop. This amount should include the seepage and evaporation losses incurred in delivering the water to the fields where the crop would be planted. The crop water requirements file may be advanced one season by the program at the request of the user.

The forecast inflows may be entered from a file which has the same format for either seasonal or sub-seasonal entry. The node name is listed as a check, subsequently listed are the predicted inflows for that node. The inflows may be negative if expected losses are greater than the expected local inflows for the duration of the time step. The inflows may not be advanced by the program because one monsoon later, new predictions are likely to be better and to differ from those previous.

6.2.4 The Variation File

The last PROJPLAN file is optional. The data preparation program does calculate the canal capacities and downstream flow restrictions from the number of days in a sub-season to find the maximum or minimum volume permitted. However, the user may wish to vary some limits. To avoid correcting the limits interactively a second time, the user may store any seasonal variations to the canal or reservoir capacities, diversion canal losses, the percent of irrigation flows returning in the same time step, and the restrictions on downstream flow in a pre-existing user-named file. The file may be advanced one monsoon season. The number of sub-seasons must agree with the number entered at the beginning of the data preparation program.

6.3 THE PREPARATION OF A PROJOP RUN

PROJOP maximizes the amount of electricity which may be generated in the rest of the monsoon season considering the irrigation requirement of already planted crops, the water supply forecast for the current season and the level of immediate irrigation necessary for the planned cropping pattern of the next season. Each updating run has less time steps as the end of the growing season approaches.

The formulation uses the same flow and storage variables as PROJPLAN, replacing the downstream release variable of PROJPLAN with both a variable for water that is released through the turbines to generate electricity and one for the water that is released downstream, surplus to electrical demands at the power plant nodes. The power plants which divert turbine flows to another stream, such as Polgolla-Ukuwela, are identified interactively so that their turbine flow variable is included in the proper water balance equation. The formulation does not have a variable for the number of hectares cropped; the crop water demand for each time step is as constant for the run. There is another set of variables for the estimations of the energy produced at a power plant in one time step. As well, the mixed linearization procedures induce integer and other variables into the formulation. PROJOP must also define the constraints which will apply to the run, and include the proper variables in each.

The quick entry of codes selecting the optional constraints, of the price for electricity, of the current reservoir levels and of the target storage volumes for the end of the season, is done interactively, after the entry and updating of the file data. Data files are used to enter the more involved and permanent data. After the entry, review and any necessary updating of the more permanent data, the user may vary some of the flow or reservoir capacities, loss or estimation constants according to the time step using the interactive program. These variations would reflect flood constraints or repairs to the various facilities which do not last the duration of a season. To minimize the entry time of these variations, the user should enter the constants which are in effect for the longest number of time steps in the permanent files correcting the others with the interactive program. The user-friendly interactive program for the operating step of the model, calling on previously prepared files when possible, easily prepares the SAS.OR date file for weekly trials of probable scenarios.

6.3.1 Calculation Interval

At the beginning of a growing season, the local inflow for each time step for every location has to be forecast for the entire season. The time steps later in the season, when future conditions are less certain, need not account for the transfer of water in as much detail and are a month long. The first few weeks of every PROJOP run has time steps of a quarter month until there are an integer number of months left in the growing season. Tables 6.3 and 6.4 list the total number of time steps, the number of monthly and the number of quarter monthly time steps PROJOP requires for the first date of every time step in the year.

TABLE 6.3

Tabulation of the calculation interval of PROJOP;

MAHA SEASON

DATE	QUARTER MONTH IN SEASON	QUARTER MONTHS LEFT IN SEASON	NO. OF TIME STEPS	PROJOP QUARTER MONTHLY	QUANTITIES MONTHLY	HALF MONTH
1/10	1	26	13	8	4	1
9/10	2	25	12	7	4	1
17/10	3	24	11	6	4	1
25/10	4	23	10	5	4	1
1/11	5	22	12	8	3	1
9/11	6	21	11	7	3	1
17/11	7	20	10	6	3	1
25/11	8	19	9	5	3	1
1/12	9	18	11	8	2	1
9/12	10	17	10	7	2	1
17/12	11	16	9	6	2	1
25/12	12	15	8	5	2	1
1/1	13	14	10	8	1	1
9/1	14	13	9	7	1	1
17/1	15	12	8	6	1	1
25/1	16	11	7	5	1	1
1/2	17	10	9	8	0	1
8/2	18	9	8	7	0	1
15/2	19	8	7	6	0	1
22/2	20	7	6	5	0	1
1/3	21	6	6	6	0	0
9/3	22	5	5	5	0	0
17/3	23	4	4	4	0	0
25/3	24	3	3	3	0	0
1/4	25	2	2	2	0	0
8/4	26	1	1	1	0	0

TABLE 6.4

Tabulation of the calculation interval of PROJOP;

YALA SEASON

DATE	QUARTER MONTH IN SEASON	QUARTER MONTHS LEFT IN SEASON	NO. OF TIME STEPS	PROJOP QUANTITIES QUARTER MONTHLY	MONTHLY	HALF MONTH
15/4	1	22	10	6	4	-
22/4	2	21	9	5	4	-
1/5	3	20	11	8	3	-
9/5	4	19	10	7	3	-
17/5	5	18	9	6	3	-
25/5	6	17	8	5	3	-
1/6	7	16	10	8	2	-
9/6	8	15	9	7	2	-
17/6	9	14	8	6	2	-
25/6	10	13	7	5	2	-
1/7	11	12	9	8	1	-
9/7	12	11	8	7	1	-
17/7	13	10	7	6	1	-
25/7	14	9	6	5	1	-
1/8	15	8	8	8	0	-
9/8	16	7	7	7	0	-
17/8	17	6	6	6	0	-
25/8	18	5	5	5	0	-
1/9	19	4	4	4	0	-
9/9	20	3	3	3	0	-
17/9	21	2	2	2	0	-
25/9	22	1	1	1	0	-

Notice that as the season progresses, one time step covering a month, becomes four quarter month time steps, once the forecast range for that date comes within two months. This causes the total number of time steps in the run, listed in the fourth column, to momentarily increase. The Maha season has a half month time step to cover the first two weeks of April until that period may be forecast confidently as quarter month time steps.

Multivariate time-lagged regression models, developed from historic monthly and quarter monthly streamflows, may be used to assist in the prediction of inflows. The number of time steps for which flows need to be predicted are at most the next eight quarter months and the four and a half subsequent months to the end of the Maha season. Any date falls in the same quarter month or month time step every year.

6.3.2 The Configuration File

This file is similar to the PROJPLAN file for the reservoir network, with significant differences. There is a vector for the nodes with power plants and the reservoir, canal, and irrigation return flow data are found on separate files. A different file must be used for each program. The information on this file is relatively permanent, while nodes may be added or deleted, or flow patterns changed, they will remain the same for the rest of the season modeled in the run. The file lists four character codes which match the power plant, reservoir, irrigation and inflow data to the correct node when the respective files are read. This checks that data for one node will not be mis-assigned to another after insertion or deletion of nodes or re-arrangement of the files. The manual explains the file line by line and item by item. The user may create, amend or augment the network interactively, storing the data on the same file or under a different OLD file name.

6.3.3 The Power, Reservoir and Canal Files

Three files contain the information of a permanent nature that is too lengthy to include in the configuration file. The information in these files must follow the order in which the nodes are numbered. The power plant file contains the minimum and maximum releases, in Mm^3, that are allowed through the turbines of each plant. The three constants necessary for estimating the energy produced are given in this file. The program may be used to calculate the average head constant, used for the energy estimation at every power plant from a head and turbine efficiency supplied interactively. These constants may then be stored in the file for subsequent use. Minus one (-1.0) is used to indicate that a power plant does not have significant variations in the head to warrant using the correcting factor, nor the reference storage, in the formulation for that node. The reservoir file contains the minimum and maximum storage volumes in Mm^3, the monthly reservoir loss coefficients and the constants for estimating the surface area from the reservoir storage volumes. The canal file lists any minimum diversion flows and inter-basin diversion canal losses as a percent of flow. These files may be created interactively with the PROJOP preparation program; however, when many numbers are to be entered, the user would likely find using a text editor quicker, once the format of the file was known. A few numbers are quickly updated with the program.

6.3.4 The Irrigation Losses and Demand Files

The penalty cost functions associated with irrigation deficits and the times for which the functions apply throughout the growing season are likely to be permanent data for any given growing season. The irrigation demands of the crops planted in the current season, per district and per quarter month, are continually updated as future evaporation, precipitation and percolation estimates warrant. The two sets of data may be combined in one file. Because the estimates of the water demand, which include transportation and field losses, may be calculated by a separate program, the user may alternatively decide to keep a file with only the crop economic losses and another with only the current estimated crop water requirements for the entire growing season. The interactive program will even combine or split these files according to the directives of the user. While the program only uses the data for the time steps which are left, it will read data for an entire growing season. This allows for the use of same data file for two or three consecutive quarter months, since blanks or numbers are included for the beginning time steps of a season even when the season is nearly over. The user may update or amend the information using the PROJOP data preparation program, especially the economic loss data, but complete full season data entry using the interactive program is not advised.

6.3.5 The Variations File

After the user has varied any of the seasonal constants to reflect weekly operating conditions or the difference in the correcting factor of the energy estimation function due to the longer duration of monthly time steps, the variations may be saved in a user-named file. In a subsequent run, this file may be read and the variations used as is, if the run is for the same quarter month, or they may be advanced one time step before continuing. When it is the first time step of a month, the program does not advance the data, the variations must be re-entered interactively. When a consistent formula is derived for decomposing all the monthly coefficients into quarter monthly, the user may re-write this subroutine to enable continuous use of this option. This file is most easily generated and corrected using the interactive preparation program.

6.3.6 The Inflow File

The inflow file contains the forecast local inflows for the remaining time steps in the monsoon season for the specified inflow nodes in the order that they are numbered in the configuration. The predicted inflows are updated every time step as the season progresses, enabling the operating tier of the model to respond to anticipated hydrological conditions as best as possible. The number of time steps left in the growing season, the number of quarter months, and the mnemonic code name assigned in the configuration, are used to ensure that the proper inflows are read for that evaluation. The inflow forecasts may be verified and corrected by the data file preparation program before the SAS.OR data file is written; however, the program does not save the inflow file, it is meant to be used only once.

6.3.7 The Irrigation Return Flow File

This file is not necessary when return flows are insignificant or to be ignored. The return flow file consists of a list of nodes which receive a known percentage of the irrigation diversion of other nodes with a lag period measured in quarter months. When the user includes the return flow option, the data preparation program will ask for the volume of irrigation water diverted in previous time steps according to the lag time duration. From these quantities the program calculates the amount of flow to add to the inflows of the first few time steps. The percentages which return in a given quarter month may vary with the time of year, however the lag time is constant throughout the season. The file may be created using the data file preparation program, reviewed or amended, and saved for a later time.

6.4 THE DATAFILES FOR THE SAS.OR LP PROCEDURE

The linear programming formulations of the current model contain many sparse constraints. Many of the constraints are specific to one node and one time step, therefore, only the variables of that node or time step have non-zero coefficients. Other variables for far away nodes or distant time steps do not enter in the constraint. Instead of entering blanks for every zero coefficient, which might take many lines, the sparsedata option of the SAS.OR LP procedure allows for the entry by variables after the identification of all the constraints, whether equals, greater than, less than, etc. The variables are listed, giving the constraints in which they are non-zero and the coefficient that occurs. When there are constant known terms in a constraint, these are summed and located to the right hand side of the constraint. These non-zero values are listed as though they were coefficients to a variable named _RHS_ in the constraint.

Upper and lower bounds and the identification of integer variables are treated as though they were constraints with the bound being the coefficient. The LP procedure sparsedata option does not require the data file to be in any order. The manual in the appendices describes the SAS.OR files in more detail and describes how they may be combined. Figure 6.1 shows an abbreviated data file which has the many repetitive lines for each node removed, to show the typical form of PLAN.DAT. These files are created by the data preparation program according to the options selected for the formulation and the figures given for the coefficients. Figure 6.2 shows the typical form of RELEASE.DAT.

These files are created by the data preparation programs according to the options selected for the formulations and the values given for the constant parameters. The user is advised to check that no output fields are too small, resulting in asterisks, nor that any line continues past column 72. When the data file lists the problem to the user's satisfaction the file may be transferred to the mainframe for solution.

```
 INCLUDE SOUMET.A.OS
 /U400S  JOB  (XXXXXX,xxxxxx),'FAVERI',MSGLEVEL=(1,1),
/        MSGCLASS=A,CLASS=2,TIME=2,REGION=2000K
*JOBPARM L=5
*ROUTE  PRINT MUSIC
/STEP EXEC PROC=SAS,OPTIONS='LS=80'
/SAS.SYSPRINT DD SYSOUT=A
/SYSIN    DD  *
 DATA PLANDATA;
 INPUT  TYPE  $  COL  $
        _ROW1_ $ _COEF1_ _ROW2_ $ _COEF2_ _ROW3_ $ _COEF3_;
 CARDS;
 .  FS1S1       CNT1S13  1       CNT1S21  -1       UPPER    180.000
 .  S3S11       LOWER   34.000    .     .            .        .
 .  FS5S3       CNT5S33  1       UPPER       16.600 LOWER    0.000
 .  QR6S31      CNT6S31  1       WAT6S31  -1       UPPER    49.283
 .  QR12S32     CNT12S32  1      WAT12S32  -1      UPPER    36.806
 .   Q1S12      CNT1S12  1       CNT2S12  -1         .        .
 .   Q6S21      CNT6S21  1       CNT7S21  -1         .        .
 .   Q17S13     CNT17S13  1      CNT7S13  -1         .        .
 .  QD10S13     CNT10S13  1      CNT13S13  -0.9500 UPPER    499.392
 .  QD13S31     CNT13S31  1      CNT14S31  -0.9750 UPPER     91.066
 LE        .     LAND10S1     .   LAND10S2   .   LAND10S3     .
 GE        .     MNCRP5S3     .     .        .      .         .
 GE        .     CCROP1S1     .     .        .      .         .
 LOWERBD   .     LOWER        .     .        .      .         .
 EQ        .     CNT7S11      .    CNT7S12    .    CNT7S13      .
 EQ        .     WAT18S32     .    WAT18S33    .      .         .
 UPPERBD   .     UPPER        .     .        .      .         .
 LE        .     LG15K10     .   LG25K10    .      .         .
 MAX       .     OBJ          .     .        .      .         .
 .     I2J1K1     OBJ        88.66   LAND1S2     1    UPPER  7000.00
 .     I2J1K1     WAT7S21   0.005670    .        .      .       .
 .     I2J1K1     WAT7S22   0.005670    .        .      .       .
 .     I2J1K1     CCROP1S2     2.06     .        .      .       .
 .     I3J5K4     OBJ       176.20   LAND4S3     1    UPPER  9900.00
 .     I3J5K4     LG25K4      -1        .        .      .       .
 .     I3J5K4     WAT13S31   0.002440   .        .      .       .
 .     I3J5K4     WAT13S33   0.010980   .        .      .       .
 .     I3J5K4     MNCRP5S3    53.90     .        .      .       .
 .    _RHS_ CNT1S11   109.700   CNT1S12    81.800   CNT1S13    83.850
 .    _RHS_ CNT17S33   123.450   CNT18S11     6.000   CNT18S12     6.000
 .    _RHS_  LAND4S1   11900.  LAND4S2   11900.  LAND4S3   11900.
 .    _RHS_  LAND10S1  30800.  LAND10S2  30800.  LAND10S3  30800.
 .    _RHS_  MNCRP5S3   1000.      .        .        .       .
 .    _RHS_  CCROP 1S2 45000.      .        .        .       .
 .    _RHS_  CCROP 1S3 50000.      .        .        .       .
 ;
PROC LP SPARSEDATA MAXIT1=500 MAXIT2=500;
/*
```

Figure 6.1: A typical PLAN.DAT data file

```
 INCLUDE SOUMET.A.OS
 /U400S  JOB  (XXXXXX,xxxxxx),'FAVERI',MSGLEVEL=(1,1),
/        MSGCLASS=A,CLASS=2,TIME=2,REGION=2000K
*JOBPARM L=5
*ROUTE  PRINT MUSIC
/STEP EXEC PROC=SAS,OPTIONS='LS=80'
/SAS.SYSPRINT DD SYSOUT=A
/SYSIN    DD  *
 DATA RELEASE;
 INPUT _TYPE_ $ _COL_ $
        _ROW1_ $ _COEF1_ _ROW2_ $ _COEF2_ _ROW3_ $ _COEF3_
        _ROW4_ $ _COEF4_ _ROW5_ $ _COEF5_ _ROW6_ $ _COEF6_;
 CARDS;
 MAX  .  OBJ  .  .  .  .  .  .  .  .  .  .  .  .
 INTEGER  .  INT  .  .  .  .  .  .  .  .  .  .  .  .  .
 LOWERBD  .  LOW  .  .  .  .  .  .  .  .  .  .  .  .  .
 UPPERBD  .  UPB  .  .  .  .  .  .  .  .  .  .  .  .  .
 .  ST13N1 CNT13N1 1.001 UPB   180.0  LOW  22.2 OBJ  0.00   .  .  .  .
 .  ST1N1  CNT1N1   1.000 CNT2N1 -1 UPB   180.0  LOW  22.2   .  .  .  .
 .  ST1N8  CNT1N8  1  CNT2N8 -1  .  .  .  .  .  .  .  .  .
 .  _RHS_  EP7N4    0.354    EP8N4    0.354   EP9N4    0.354
           EP10N4    0.354   EP11N4    0.354   EP12N4    0.354
 EQ .  EP8N5 .  EP9N5 .  EP10N5 .  EP11N5 .  EP12N5 .  EP13N5 .
 . ENG3N9 OBJ  10.00         EP3N9 -1  .  .  .  .  .  .  .  .  .  .
 . ST4N9   EP4N9   0.001282 EP5N9  0.001282   .  .  .  .  .  .  .  .  .  .
            UPB   352.1  LOW  11.9   EP12N19 0.4490  .  .
 . QTB13N19 CNT13N19 1  CNT13N20 -1
            UPB   176.0  LOW   5.9   EP13N19 0.4490  .  .
 . QTB1N20 CNT1N20 1  CNT1N21 -1
            UPB    49.4  LOW   3.0   EP1N20 1.3343  .  .
 . QDF7N6    IRR7N6  1  OBJ   -250.000   .  .  .  .  .  .  .  .  .  .
 .  QR8N6    IRR8N6 1 CNT8N6 1   .  .  .  .  .  .  .  .  .  .
 . QDF8N6    IRR8N6  1  OBJ   -222.222   .  .  .  .  .  .  .  .  .  .
 EQ . IRR1N6 .  .  .  .  .  .  .  .  .  .  .  .  .
          IRR10N6    1.753  IRR11N6    1.252 IRR12N6    0.626
 .  _RHS_ IRR13N6    0.626   .  .  .  .  .  .  .  .  .  .  .  .
 .  QS4N1 CNT4N1 1  CNT4N2 -1  .  .  .  .  .  .  .  .  .
 .  QS5N1 CNT5N1 1  CNT5N2 -1  .  .  .  .  .  .  .  .  .
 .  QS5N2 LOW   2.8   .  .  .  .  .  .  .  .  .  .
 .  QD5N6 CNT5N6 1  CNT5N17 -1.000   UPB    42.2   .  .  .  .  .  .  .
 . Q2D8N14 CNT8N14 1  CNT8N15 -1.000   UPB    22.4   .  .  .  .  .  .  .
 . Q2D9N14 CNT9N14 1  CNT9N15 -1.000   UPB    89.6   .  .  .  .  .  .  .
 EQ . CNT8N17 . CNT9N17 . CNT10N17 . CNT11N17 . CNT12N17 . CNT13N17 .
 EQ . CNT1N20 .  .  .  .  .  .  .  .  .  .  .
 .  _RHS_ CNT7N23    2.150  CNT8N23    2.150 CNT9N23    1.600
          CNT10N23    2.600 CNT11N23    2.40 CNT12N23    2.600
 ;
PROC LP SPARSEDATA TIME=1920 IMAXIT=5000 MAXIT1=500 MAXIT2=500;
/*
```

Figure 6.2: A typical RELEASE.DAT data file

6.5 SOLUTIONS FROM SAS.OR

The solution takes less than one minute of CPU time, under 1000 iterations to find the first feasible solution, less under wet conditions, and under 500 iterations to the optimal feasible solution, producing output similar to that of Figures 6.3 when no mixed integer linearization is involved. That procedure, because of the branch and bound algorithm used to check the non-integer solutions, increases computer time immensely and produces a log of the integer search procedure in the output.

Following the integer solution log, the print-out, as shown in Figure 6.4, continues listing all the variables in alphabetical order. The actual solution values of the variables appear in the sixth column under the heading ACTIVITY. The last column indicates the sensitivity of the solution for the variable to the coefficients and prices given in the problem. Following the list of variables is a list of the constraints in the program, showing how well the solution was able to meet or surpass the right hand side constants for that constraint. More information on the print-out formats may be found in the appendices and in a SAS.OR manual.

Occasionally the problem as given will be infeasible, no combination of non-negative values could be applied to the variables to satisfy the constraints. The last pages of the met and the values used for its variables. From there, the user may decide why the problem is infeasible: incorrect information in the data file, too stringent bounds, not enough water.

The following chapter describes the application of the two-part model to the Mahaweli Project in Sri Lanka and how any infeasible problems encountered were remedied.

L I N E A R   P R O G R A M M I N G   P R O C E D U R E

PROBLEM SUMMARY

MAX OBJ	OBJECTIVE FUNCTION
RHS	RHS VARIABLE
TYPE	TYPE VARIABLE
PROBLEM DENSITY	0.009

VARIABLE TYPE	NUMBER
STRUCTURAL	
NONNEGATIVE	128
UPPER BOUNDED	340
UPPER AND LOWER BOUNDED	72
LOGICAL	
SLACK	50
SURPLUS	4
TOTAL	594

CONSTRAINT TYPE	NUMBER
LE	50
EQ	224
GE	4
FREE	1
TOTAL	279

L I N E A R   P R O G R A M M I N G   P R O C E D U R E

SOLUTION SUMMARY

TERMINATED SUCCESSFULLY

OBJECTIVE VALUE	41957381
PHASE 1 ITERATIONS	273
PHASE 2 ITERATIONS	400
INITIAL B.F. VARIABLES	57
TIME USED (SECS)	26.55
NUMBER OF INVERSIONS	7
MACHINE EPSILON	1.00000E-08
MACHINE INFINITY	7.23701E+75
INVERT FREQUENCY	150
MAX PHASE 1 ITERATIONS	500
MAX PHASE 2 ITERATIONS	500
TIME LIMIT (SECS)	120.00

Figure 6.3: The SAS.OR solution summary

LINEAR PROGRAMMING PROCEDURE

VARIABLE SUMMARY

COL	VARIABLE NAME	STATUS	TYPE	PRICE	ACTIVITY	REDUCED COST
1	FS1S1	ALTER	UPLOWBD	0	22.200000	0
2	FS1S2	ALTER	UPLOWBD	0	22.200000	0
4	FS12S1	BASIC	UPLOWBD	0	70.904869	0
19	FS18S1	BASIC	UPPERBD	0	120.563	0
20	FS18S2	BASIC	UPPERBD	0	237.430	0
21	FS18S3	ALTER	UPPERBD	0	478.000	0
46	I1J1K6		UPPERBD	79.66	0	-3.263833
84	I1J5K4		UPPERBD	176.2	9900.000	77.727703
151	I2J6K10		UPPERBD	77.53	15800.000	24.988000
153	I2J6K3	BASIC	UPPERBD	79.65	5452.000	0
154	I2J6K4		UPPERBD	76.9	0	-12.751196
171	I3J2K10	BASIC	UPPERBD	84.29	5625.000	0
172	I3J2K2		UPPERBD	93.01	0	-16.36667
229	QD11S12	BASIC	UPPERBD	0	21.669173	0
235	QD11S33	ALTER	UPPERBD	0	166.000	0
236	QD13S11	BASIC	UPPERBD	0	73.950000	0
260	QD2S11		UPPERBD	0	152.700	4409.091
261	QD2S12	BASIC	UPPERBD	0	134.180	0
270	QD6S13	BASIC	UPPERBD	0	357.860	0
271	QD6S21	BASIC	UPPERBD	0	63.750000	0
281	QR10S31	BASIC	UPPERBD	0	11.550000	0
282	QR10S32	BASIC	UPPERBD	0	26.120000	0
328	QR18S22	BASIC	UPPERBD	0	142.636	0
329	QR18S31	BASIC	UPPERBD	0	38.900000	0
379	Q11S33	BASIC	NON-NEG	0	590.280	0
380	Q13S11		NON-NEG	0	0	-464.148
491	S15S11	BASIC	UPLOWBD	0	45.646344	0
492	S15S12	BASIC	UPLOWBD	0	38.706344	0
547	LAND10S1		SLACK	0	0	-8.894545
548	LAND10S2	BASIC	SLACK	0	3213.333	0

CONSTRAINT SUMMARY

ROW	CONSTRAINT ID	TYPE	S/S COL	RHS	ACTIVITY	DUAL ACTIVITY
1	CCROP1S1	GE	541	50000.000	73317.821	0
2	CCROP1S2	GE	542	45000.000	95505.000	0
3	CCROP1S3	GE	543	50000.000	99395.148	0
13	CNT10S12	EQ		28.440000	28.440000	4409.1
22	CNT11S13	EQ		142.155	142.155	0
96	CNT3S22	EQ		235.200	235.200	0
97	CNT3S31	EQ		51.800000	51.800000	0
152	LAND10S2	LE	548	30800.000	27586.667	0
153	LAND10S3	LE	549	30800.000	30800.000	72.259
197	LG25K9	LE	593	0	0	179.72
198	MNCRP5S3	GE	594	1000.000	2334000	0
199	WAT10S11	EQ		0	0	4409.1
279	OBJ	OBJECTIVE		41957381	41957381	0

Figure 6.4: An abbreviated SAS.OR solution print out

CHAPTER 7

TEST APPLICATIONS OF THE MODEL

Several test applications of the computer programs were conducted. The trials described in this chapter used the set of reservoirs, irrigation districts and power plants as described in Chapter 2. The tests may be grouped into three categories, a year of average streamflows, a dry year, and a year of average streamflows with calibration of the correcting factor for the energy estimations.

7.1 PURPOSE

To verify that the programs and models worked as envisioned, a number of trial applications were conducted, even as the models were undergoing development. The description of the test trials in this chapter may serve to illustrate the approach of the models and their solutions. As well, any problems or malfunctions which were discovered have solutions discussed.

Although the models are to be run consecutively with updated flow forecasts and water levels at regular intervals, the test applications were run only once. The planning model, PROJPLAN, was evaluated once at the beginning of a Maha season, to provide target storage levels for the operating model. The operating model, PROJOP, was tested at the beginning of the Maha season and at the beginning of a Yala.

The flow forecasts were based on thirty-two years of reported streamflow given in the Mahaweli Authority - Acres International 1986 Report. In the year and a half of average streamflows, the historical mean monthly flow was used. In the dry year the worst streamflows in a consecutive year and a half, 1955-56, were used.

The anticipated use of the models requires continuous updating to correct for the uncertainty of future flows. The resilience of the cropping patterns and release policies selected by the models to varying flow forecasts is an interesting and important characteristic which should be determined before the models are implemented.

No operating conclusions should be made on the basis of the applications reported here until the user has supplied more realistic, up-to-date economic and technical data for trials of both steps of the model. The applications to the system in Sri Lanka are described here to aid the user in applying the model to actual conditions.

7.1.1 The Configuration of the Test Applications

In the process of development of the formulations and of the data preparation programs, numerous tests were conducted using mainly fabricated data and simple five node configurations. By the time the final versions of the two models were ready, the Mahaweli Authority-Acres International Report [1986], which tables the inflow, crop water demand, reservoir and power plant data used in a 32-year simulation planning model, was available.

Recent tests, described in this chapter, used the data contained in that report, although the linear programming formulations of PROJPLAN and PROJOP are capable of modelling more reservoirs and canals than did the simulation model. Early full-scale tests, using streamflow estimated from yearly volumes at nearby gauge sites distributed weekly, included more nodes than shown on the diagram of Figure 2.2. As the more complete report data was only available for the nodes used in the simulation model, the recent application tests of the models were conducted on a less complex configuration.

By using the same configuration of the Mahaweli System the results may be compared between PROJOP and PROJPLAN solutions and the various simulation models used previously. Therefore, the test applications apply to the 22 node system shown in Figure 2.2 which has 10 irrigation districts, 11 power plants, and 15 reservoirs. Two of the reservoirs and five of the power plants are not actually in the Mahaweli Basin but are included so that the electrical production will be maximized jointly. Total energy production figures may be compared directly with reported values.

7.1.2 Choice of the Average Water Year

The irrigation and electrical production which resulted from modelling a year and a half of average streamflows may be compared with traditional operations, to evaluate the PROJPLAN and PROJOP models. The mean annual monthly flows and the crop irrigation demands of the 30+ year period simulated and reported in the Mahaweli Authority - Acres International Report [1986] were used to validate the functioning of the models. The results may be compared to the values reported from the simulation model and formed the basis of comparison for other tests of PROJOP and PROJPLAN with different hydrological inputs.

To accurately compare the electrical production during the average water year, PROJPLAN had a fixed cropping pattern utilizing all the land and consuming the historic mean water demand. It defined the end of season reservoir targets for PROJOP, which then distributed the mean monthly inflows to satisfy the irrigation demands and maximize the electrical production. The electrical production and canal diversions estimated by PROJOP were compared to reported mean and historic values to verif the models would determine realistic policies. The operating policies determined by PROJOP did not significantly differ from those implicit in the 1986 report.

7.1.3 Choice of the Dry Year

Lack of water causes planning dilemmas. The critical choices occur when a release policy does not satisfy everyone, the planner must offer justification to the unfortunate of the final choice of allocation. How the PROJPLAN and PROJOP models function under the conditions of scarce water is an important and interesting question. Thus the driest year and a half of record, 1955-56, was also modelled by the two programs. The results were compared to the base year, the average water year.

PROJPLAN was allowed to pick a cropping pattern from a list of possible crops, with fictitious prices and water demands based on available information. The irrigation schedule of crop water demands for PROJOP was based on the cropping pattern selected by PROJPLAN. The end of season reservoir levels of the PROJPLAN model were included as the target storage levels for the one season of the PROJOP model.

7.1.4 Calibration of the Correcting Factor

As mentioned in Chapter 5, the function for the production of hydroelectricity is not linear. The PROJOP model uses an approximation with three constants whose values are predetermined. The best value for the correcting factor, in particular, is determined according to the turbine release. A set of PROJOP trials were conducted which varied the value of the correcting factor according to the turbine release of a previous solution to investigate the influence of poor estimates on the release policy.

These trials were conducted with the average water year hydrological data. Crop water demands followed the mean annual schedule published in the Mahaweli Authority - Acres International 1986 Report. Firm energy limits were included in the PROJOP model.

7.2 DESCRIPTION OF SYSTEM DATA

The Mahaweli Authority - Acres International 1986 Report listed the monthly inflows and crop water demands, canal and reservoir capacities. Other information required by the models was obtained from other sources. The system capacities, the inflows and the energy parameters and restrictions are three broad categories where the details of the values used in these test applications are explained.

7.2.1 System Capacities and Losses

The ten irrigation districts, fifteen reservoirs, and eleven power plants listed in Tables 2.1 to 2.3 were modelled in the test applications. The reservoir losses, elevation - storage curves, channel and reservoir capacities and the channel losses remained the same in all the test applications discussed here and were applied exactly as tabled by the Mahaweli Authority and Acres International Report.

The reservoir losses are applied as a percent of the reservoir storage at the end of the time step. The percent is determined from the loss reported in millimetres and the calculation of the surface area for that reservoir storage volume. The coefficient for determining the surface area from the storage volume was estimated from the elevation - storage table of the report. These coefficients are stored on the reservoir file. Note this linear approximation of the evaporation loss entails minor inaccuracies when the surface area is not a linear function of the storage volume.

The reservoir capacities included in all test applications discussed here are listed in Table 2.2 and follow those published in the 1986 report. The minimum reservoir levels were chosen as those associated with the minimum sill elevation, as in the case of the Ulhitiya reservoir which diverts water to Maduru when the elevation of the reservoir is above 100 m. When no minimum operating levels were mentioned, the reservoirs were allowed to empty completely. Reservoirs in the Mahaweli Project were started in the Maha season at their lowest permissible levels, those in the K-M basin, where no irrigation is practised, arbitrarily started and ended full.

The channel capacities used in the applications were based on the flow rates listed in the report. Because the length of the sub-seasons in PROJPLAN vary and operation of the canals may not be conducted at maximum levels twenty-four hours a day, seven days a week, the actual volume permitted in a time step was actually less. The channel losses listed in the report were used as given, with no seasonal variations made.

7.2.2 Inflow and Crop Adjustments

The models do not always use time steps of a month, which was how the data was available. The monthly data on the streamflows and irrigation requirements was divided by four to obtain the quarter monthly values and divided by two for the half month time step of April required by PROJOP. This is unrealistically regular, however no more realistic mechanism was available.

The crop water demands and the streamflows for the seasonal model were also adjusted for the irregular time steps of PROJPLAN. The values for the month of April were divided in two, including the first half in the Maha season and the second half in the Yala.

The time steps for the water allocation in PROJPLAN needed more refinement. According to the data available, the following sub-seasons were identified:

1. the month of October,
2. the month of November,
3. the rest of the Maha season,
4. the last half of April and the month of May,

5. and the last four months of the Yala season.

These five sub-seasons isolated the times when the crops required proportionally more water and were the least likely to have the water available.

The Maha season in any one district has only two sub-seasons. However, the critical time was later in some districts than in others. The different timing resulted in three sub-seasons for calculations.

No irrigation return flows were included in the test applications of the model to the Mahaweli - K-M test data due to lack of informed sources.

7.2.3 Energy Parameters and Restrictions

The parameters used to evaluate the energy being produced by the releases at a power plant in a time step are given in Table 7.1.

A benefit of 10 Rs was assigned to every GW-hr of energy produced. The relative value of this benefit to the penalty assigned to irrigation deficits is important; the penalty due to irrigation deficits differed throughout the growing season and for different nodes. The value was typically 200 Rs, ranging from 190 to 250. These magnitudes do not permit irrigation deficits without true shortages of water. The option of assigning a benefit to surplus storage at the end of a PROJOP horizon was not included in these runs.

The annual firm power projected for both the K-M and Mahaweli power plants was summed and the total distributed evenly throughout the year, which caused no problems in the average year. No other restriction was placed on the energy production.

TABLE 7.1

Energy Estimating Constants

NODE	POWER PLANT	MIDWAY ELEVATION m	HEAD m	h_{in} GW-hr/Mm^3	sto_{in} Mm^3	c_{in}/Qo*
1	KOTMALE	183	182.4	0.4222	85.0	0.0005634
2	UKUWELA	-	78.0	0.1800	-	-
3	VICTORIA	157	157.0	0.3625	275.0	0.0002202
4	RANDENIGALA	218	64.0	0.1477	600.0	0.0011800
5	RANTEMBE	-	31.5	0.0727	-	-
9	BOWATENNE	52	51.2	0.1182	33.2	0.0005130
19	CANYON	195	218.2	0.4490	66.0	0.0004228
20	NEW LAXAPANA	-	578.0	1.3343	-	-
21	POLPITIYA	-	259.0	0.5979	-	-
22	WIMALASURENDRA	219	194.5	0.5036	24.0	0.0011248
23	OLD LAXAPANA	-	449.0	1.0365	-	-

* Qo = 5 Mm^3 for a quarter monthly time step
Qo = 100 Mm^3 for a monthly time step
Qo = 50 Mm^3 for a half monthly time step

c_{in} used in the dry and average year applications was the Qo for the time step times the value shown in the table.

7.3 MEAN MONTHLY FLOW YEAR

The model was tested with the case of an average water year when all the Mahaweli basin reservoirs were empty. The streamflows were derived from the mean monthly flows of the 32-year reconstituted data in the simulation report apportioned to sub-seasonal or quarter monthly values as described above. The mean monthly crop water demands of the same 32-year period for the "present irrigation practice" case were used. This meant that the PROJPLAN formulation had a completely fixed cropping pattern, one crop was entered which had a price of 100 Rs/hectare, the seasonal water use/hectare as reported, and minimum and maximum land limits equal to the size of each district.

7.3.1 Discussion of the Test Application and Solution

The results of the application for average inflow and average crop water requirements show that the operating model was able to find a policy which met the irrigation demands and generated 3684 GW-hr, over one quarter more than the projected annual firm energy of 2711 GW-hr.

A cursory post-optimal analysis of the PROJOP printout was conducted. The reduced cost column across from the irrigation demand deficit variables, QDFin, indicate for which districts and time steps the forecasts of water shortages are the most crucial. The reduced cost column is a reflection of the penalty costs the user has given to any irrigation demand deficits as well as the capability of the reservoirs and diversions to deliver timely water supplies.

The values in the dual variable column across from the irrigation constraints suggest that the irrigation of the H, I, and M district, in particular, reduces the overall generation of electricity. This is a fact of the real-life water allocation problem which was not explicitly entered with the input data. That the model attests this conclusion promotes more confidence in the model's ability to mirror reality.

Nevertheless, operating decisions based on post-optimal analyses of the test application will not be valid because the parameters and weights used in the model have been arbitrarily chosen from limited knowledge of the Mahaweli system. The test applications show that the linear programming models will provide further insights on the functioning of the system.

The reservoir levels selected for the end of the PROJOP time steps by the model have been plotted to compare with rule curves or typical reservoir levels observed at the reservoirs. Note that the first eight time periods are quarter months, the next four are months, the thirteenth is a half month, the next six are again quarter months and the final four periods are months. Figures 7.1 to 7.5 show the storage levels for each time step plotted for Victoria Reservoir, Randenigala Reservoir, the combined Minneriya-Giritale Reservoir, Ulhitiya Reservoir and Mousakelle Reservoirs. Reservoir levels fluctuate between the minimum and full supply levels in an expected and acceptable fashion.

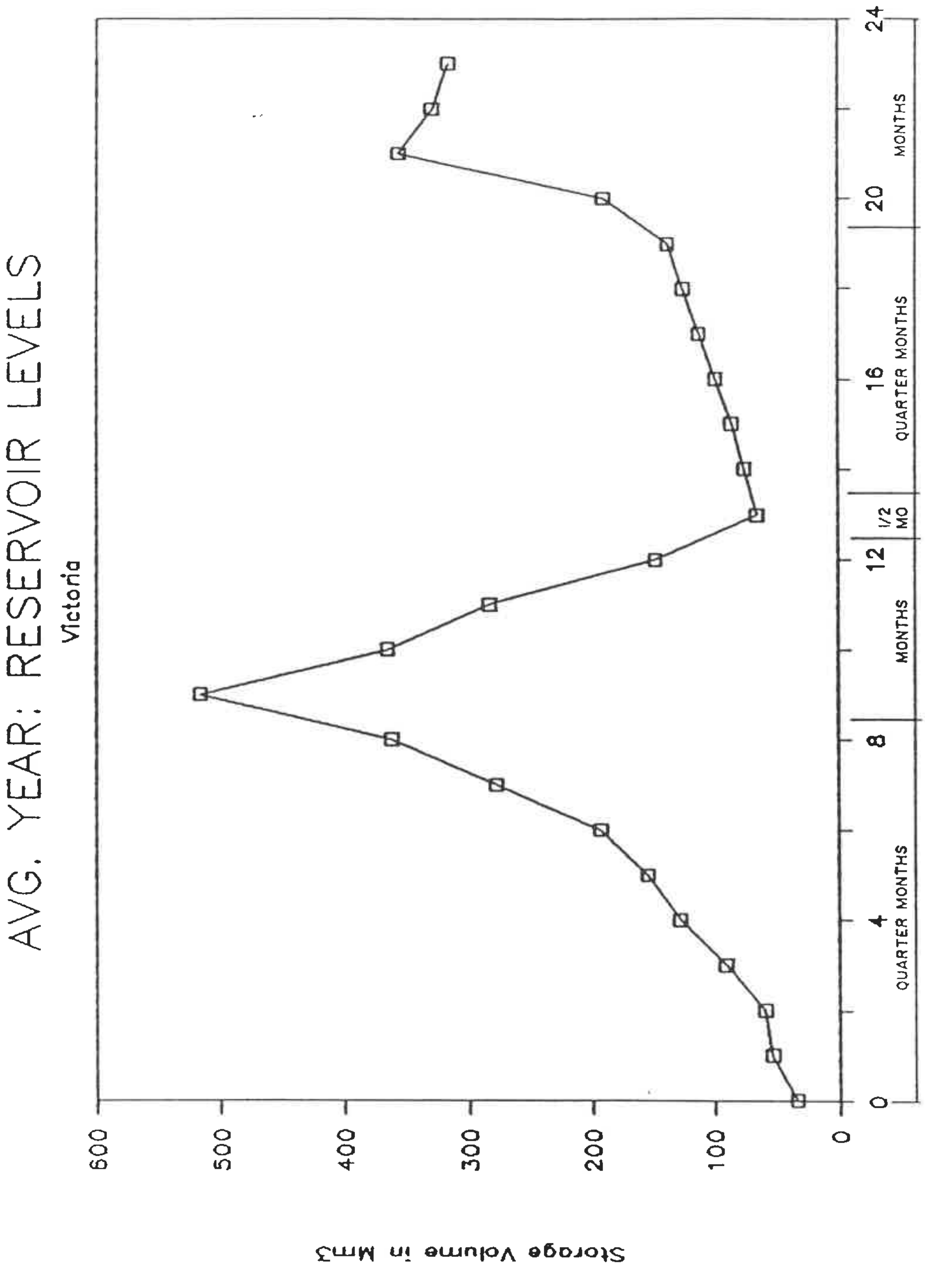

Figure 7.1 Average water year: reservoir storage Victoria

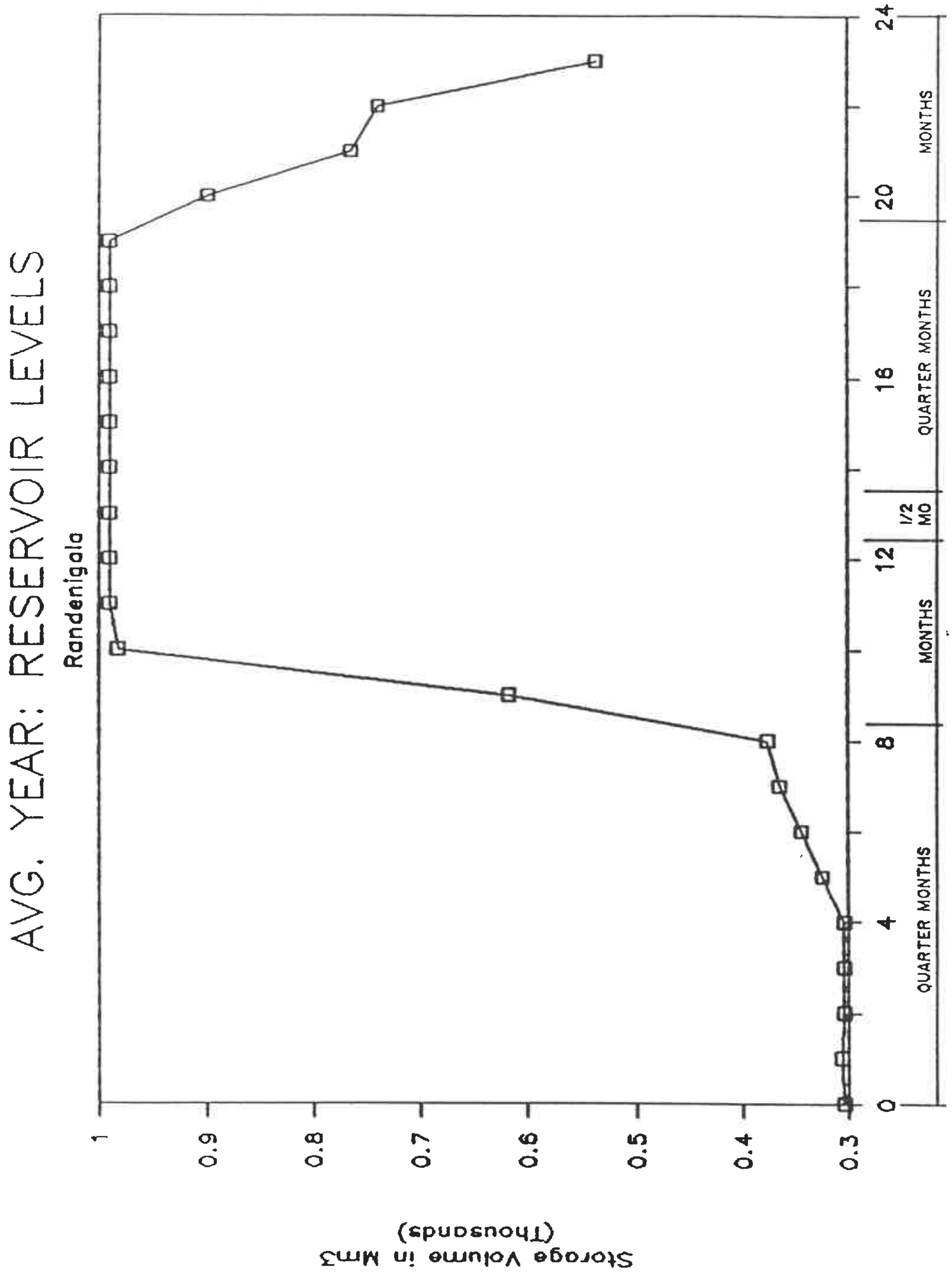

Figure 7.2 Average water year: reservoir storage Randenigala

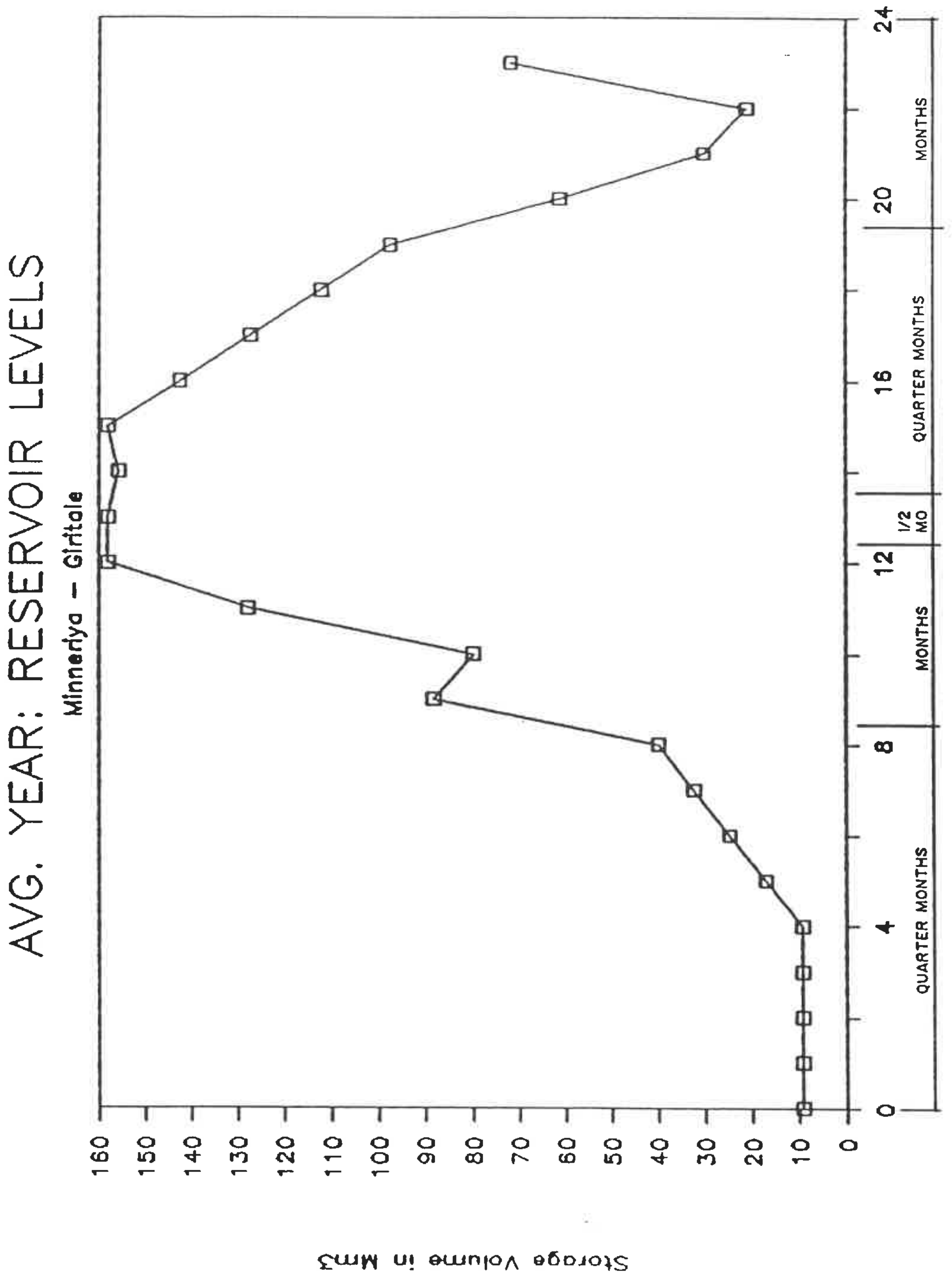

Figure 7.3 Average water year: reservoir storage Minneriya and Giritale

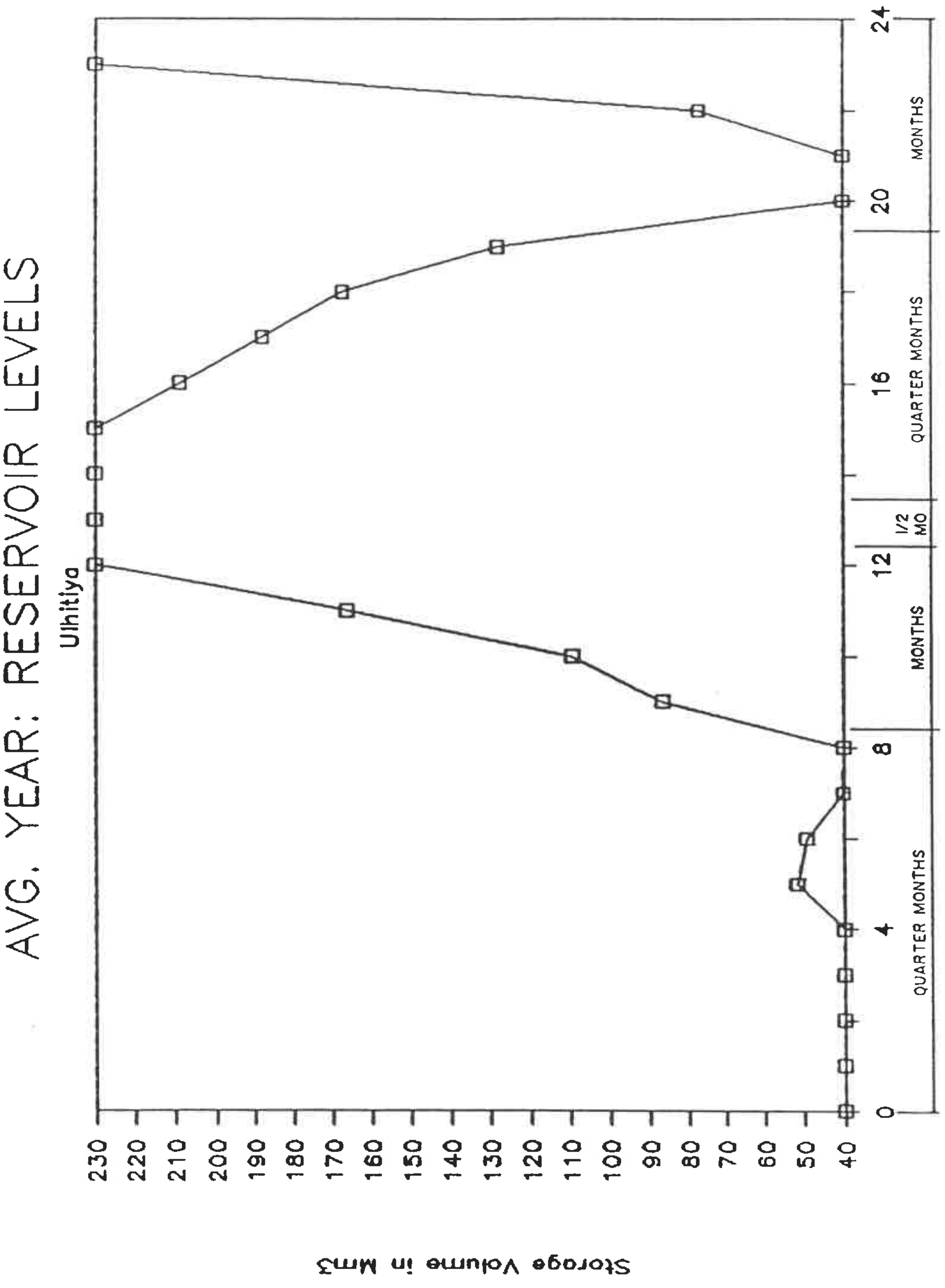

Figure 7.4 Average water year: reservoir storage Ulhitiya

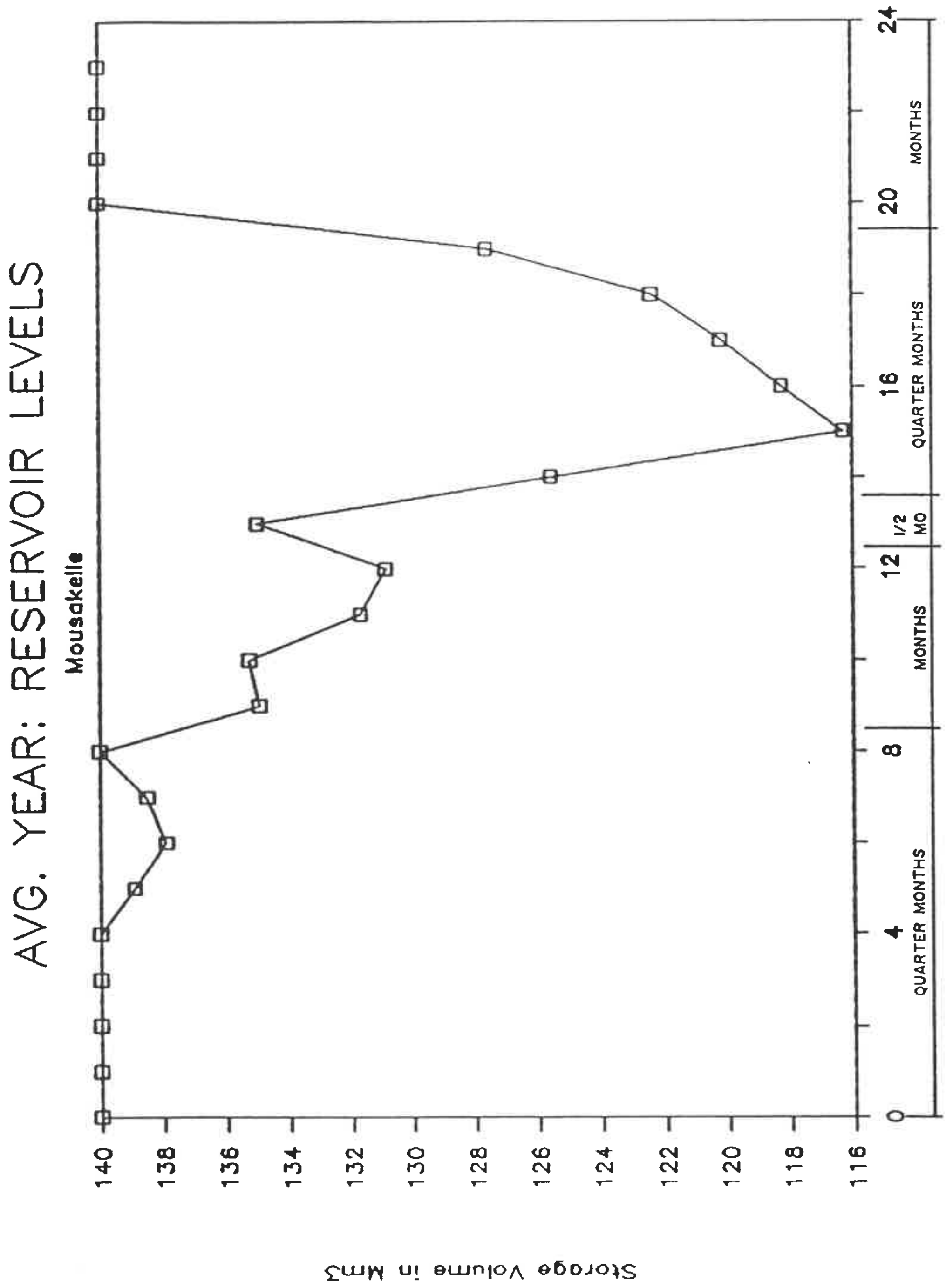

Figure 7.5 Average water year: reservoir storage Mousakelle

Figures 7.6 to 7.8 show the diverted volumes from the Polgolla to Bowatenne Reservoirs, the Minipe Anicut to the Ulhitiya Reservoir, and from the Ulhitiya to Maduru Reservoirs. Neither the reservoir levels nor the diversion releases show any absurd or preposterous release policy. Figure 7.9 shows the total energy produced each time step.

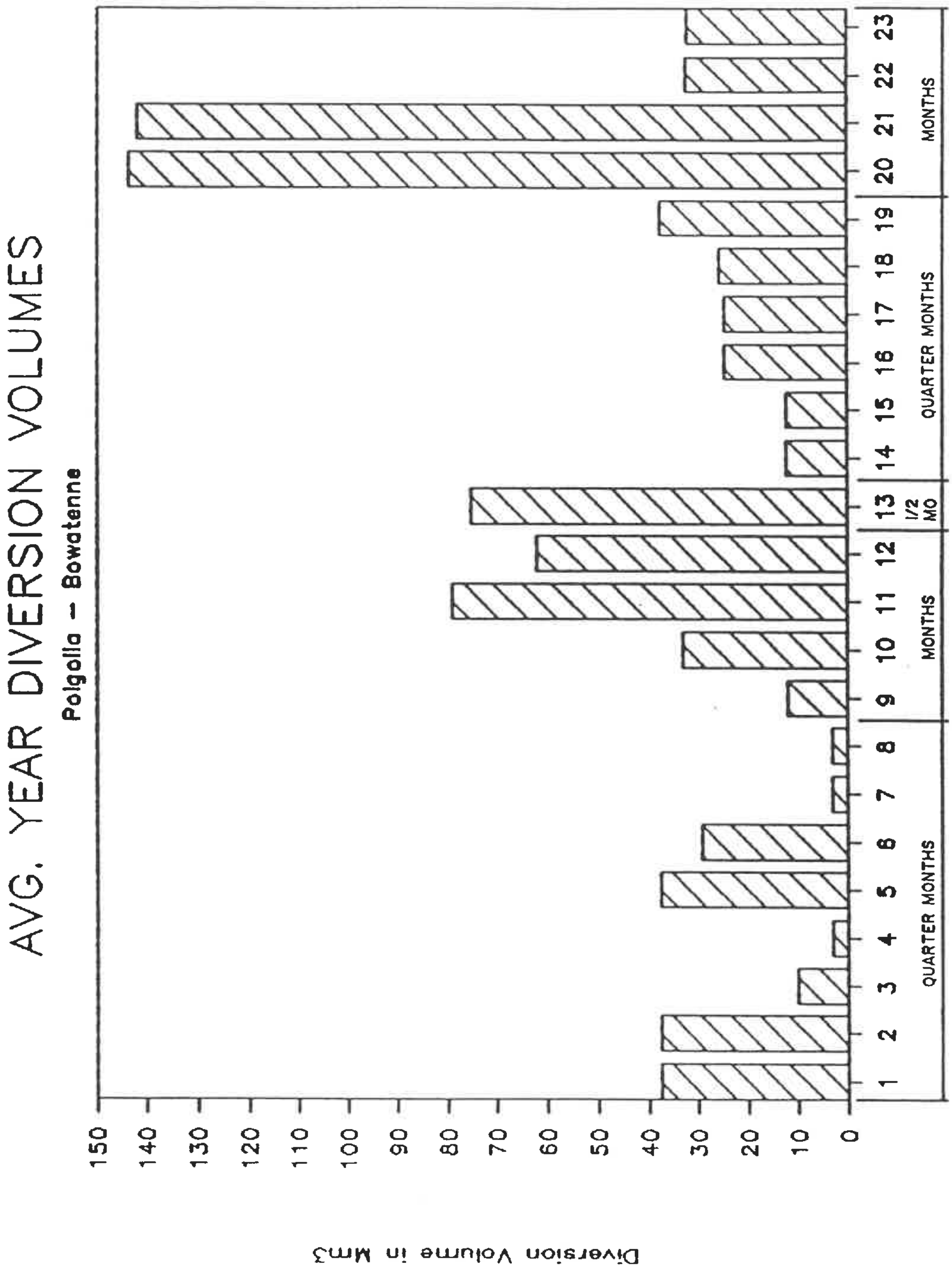

Figure 7.6 Avg. water year: div. vol. - Polgolla

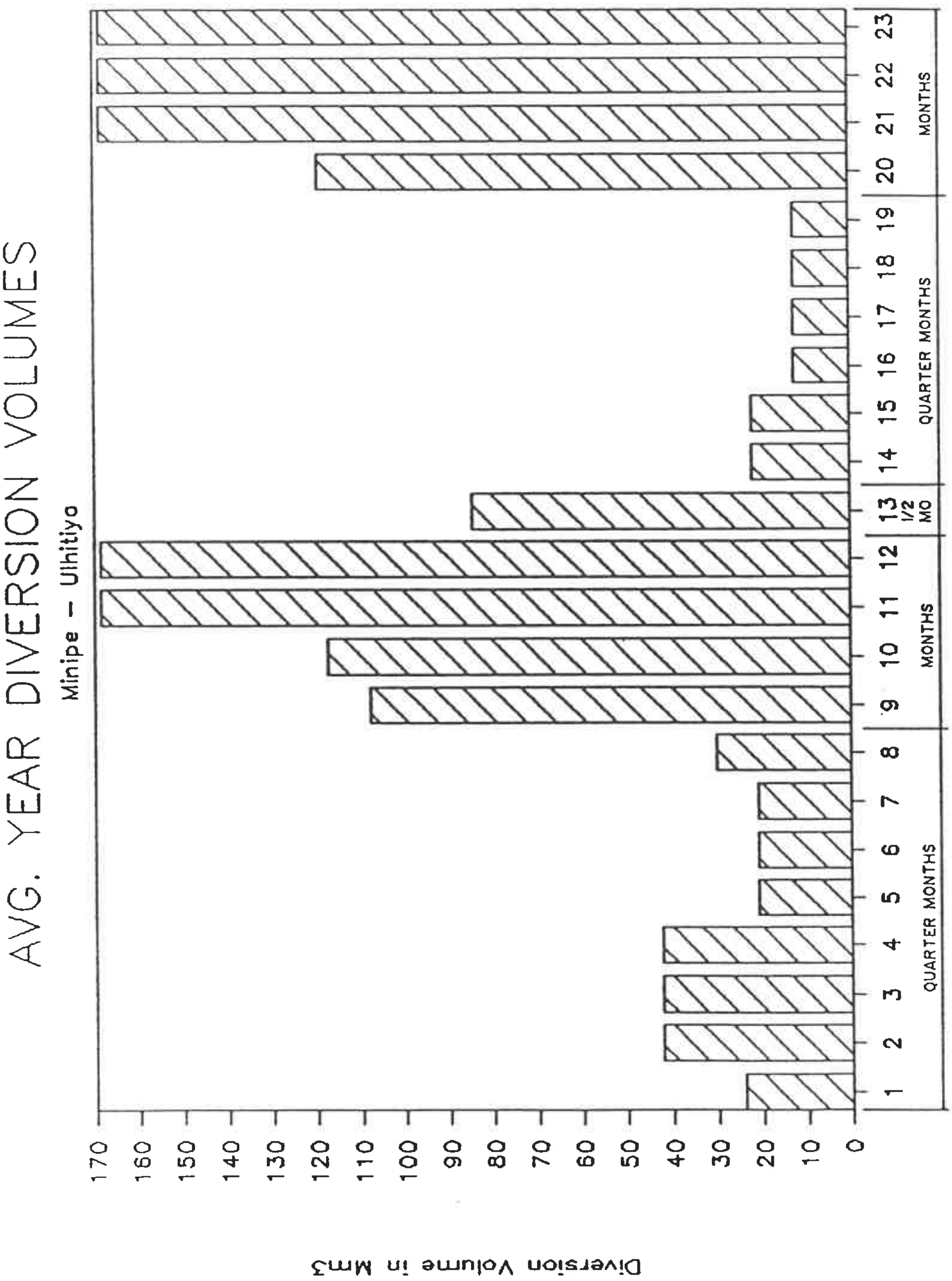

Figure 7.7 Average water year: diversion volumes - Minipe

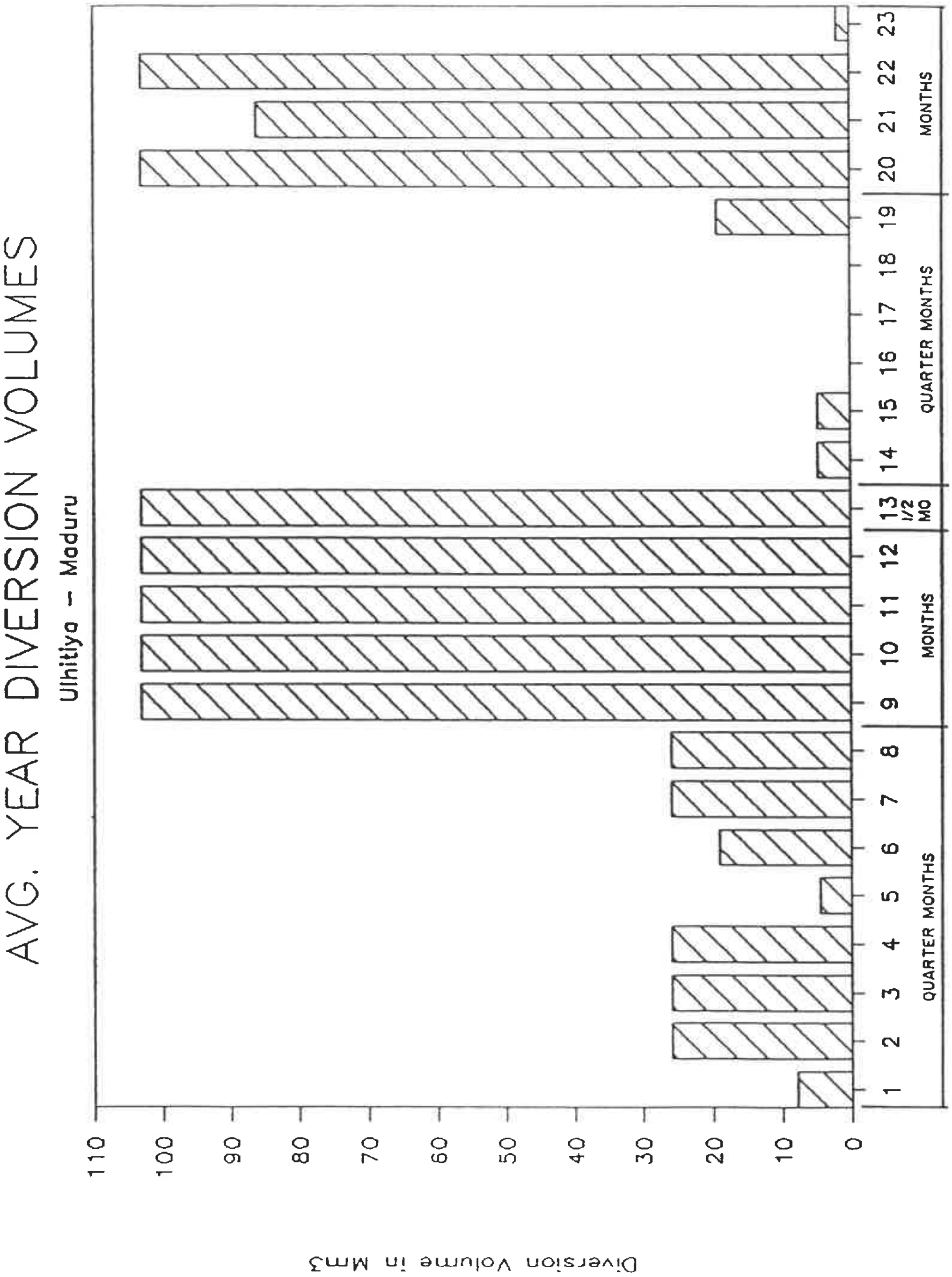

Figure 7.8 Average water year: diversion volumes - Ulhitiya

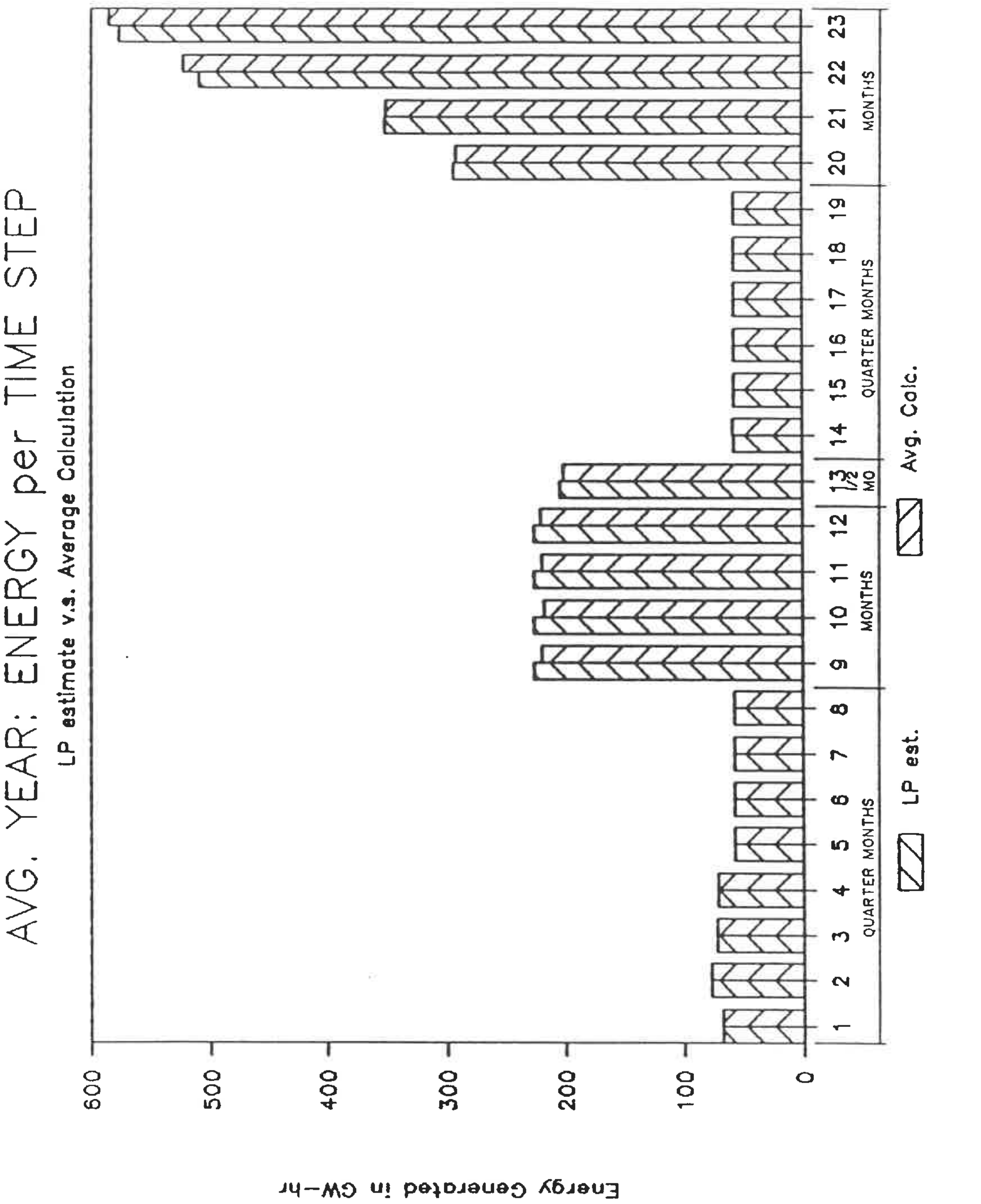

Figure 7.9 Energy production in the average water year

7.3.2 Problems Discovered during the Test Application

During the test applications of an average water year, a few problems arose. Because the cropping pattern is completely fixed, the SAS.OR solution for PROJPLAN produced warnings for each LAND constraint, which said that there was only the one variable. As well adjustments were required to ensure adequate water budgeting, both during the seasons and at the start.

Since the test application of the average water year used the average crop water demands for a completely fixed cropping pattern in PROJPLAN, the SAS.OR Linear Programming procedure warned that every season and every district had a superfluous LAND constraint because only one crop appeared in each. The removal of the LAND constraints from the SAS.OR data file by a text editor suppressed those warnings; although results would have been obtained with the LAND constraints and warnings intact.

More disquieting was that without any crops to select, the PROJPLAN formulation has no way of judging which feasible solution of the remaining water balance equations is better than another. With the surplus water of the average flow year, PROJPLAN suggested maintaining Victoria reservoir at higher than necessary levels at the end of the Yala season. However, the PROJOP run which had this target storage level was unable to fill the reservoir without incurring irrigation deficits as water for power production was continually released due to the firm energy constraints.

To obtain a reservoir level that was equally feasible but more desirable operationally, a restriction on the minimum releases allowed from Victoria reservoir during each sub-season was included in the PROJPLAN run. This solution recommended reservoir levels that were still adequate for the next season, yet were not operationally difficult to achieve.

Whenever a completely fixed cropping pattern is entered, or whenever the predicted inflows allow for a cropping pattern which generates the most revenue possible (i.e. the available water is not limiting the solution) the user may find that PROJPLAN selects undesirable reservoir levels which are feasible within the sub-season time steps of that formulation but non-achievable with the additional goals and shorter time steps of the operating tier. Some adjustments to PROJPLAN, similar to the minimum reservoir release mentioned above, may thus be required.

Secondly, the first tests of this application were infeasible because the level of cropping in district B was too high to be immediately serviced by the Maduru reservoir when it was empty. To maintain the cultivation at the average level for this district, the Maduru reservoir started the Maha season with 100 Mm^3 of storage.

Furthermore, early solutions were designated infeasible, without the usual page for the guilty constraint. Instead, only a storage value for the Mousakelle reservoir, slightly above full supply level, was starred and marked as infeasible. The solution kept this high reservoir level in the solution to attain the full supply level in the reservoir at the end of the season. Thus, to keep this reservoir from filling higher than capacity, it was allowed to finish one season at 5 Mm^3 less than full.

7.4 DRY YEAR APPLICATION

The model was also tested with the case of a very dry inflow year: the water year 1955-56 with all the reservoirs empty at the start of the season. The streamflows used were derived from the reconstituted data in the simulation report as described above. To fully test its formulation, PROJPLAN selected the cropping pattern.

The final storages of the PROJPLAN seasonal allocation were supplied as target values to the PROJOP runs. The crop water demands were distributed through the year according to percentages mentioned below and the seasonal volumes used for PROJPLAN for the selected crops. The turbine data was the same as in the average year application.

7.4.1 Definition of PROJPLAN Input Data

Since no better crop statistical data was available, fictitious data was prepared. Table 7.2 shows the typical seasonal data used for each of the six fictitious prospective crops.

The data was permuted around these values for each district and each season in a random manner. The sub-seasonal crop water demands/hectare were proportioned according to the recorded present case monthly water use for the first three crops. The seasonal water demands of Crop 4 and Crop 6 were split 30% - 30% - 40% in the Maha seasons and 60% - 40% in the Yala.

TABLE 7.2

Typical seasonal data used for crop selection

DISTRICT: 10 HIM SEASON: 2

CROP	PRICE Rs/ha	YIELD	COSTS LABOUR Rs/ha	FERT. Rs/ha	IRR. DEMAND 1000 m3/h	SUITABLE LAND ha
CROP 1	100.00	2.25	0.00	0.00	14.98	30800
CROP 2	110.00	2.47	0.00	0.00	15.98	30800
CROP 3	90.00	2.24	0.00	0.00	13.98	9800
CROP 4	45.00	7.50	0.00	0.00	4.98	30800
CROP 5	200.00	0.00	0.00	0.00	14.98	9800
CROP 6	88.00	5.90	0.00	0.00	6.98	15800

The water use of Crop 5 was constant all year round and it was designated as a long growing season crop. The production of the first three crops combined had to be above a certain level, and the fifth crop had a production target applicable in the third season when it would be harvested. No restriction was placed on the amount of land uncultivated in a solution.

7.4.2 Results of the Dry Year Application

The results of this application to a dry year, specifically the water year 1955-56, show that the operating model was able to find a policy which delayed the irrigation deficits to the end of the season where they would be less disastrous and that 2 519 GW-hr of hydroelectric energy was generated by the hydroelectric plants that year. This production is nominally less than the projected annual firm energy of 2 711 GW-hr.

Post optimal analyses of both the PROJPLAN and PROJOP printouts revealed districts which might benefit from more available cultivable land, which crops were more price sensitive, and less critical reservoirs. However, because the models were not using accurate prices, costs, nor other parameters, the particular conclusions are worthless. When the user supplies up-to-date, accurate information for these parameters, the parametric analyses of the models will provide valuable planning information.

The overall irrigation deficit was less than 20 Mm^3 in that dry year, distributed over 5 districts and 8 time steps. The largest deficit in any time step for any district was 12 Mm^3. To obtain these low deficits in the operating step, cultivation in district B was severely restricted and partially restricted in districts C and H, I, and M by the seasonal planning step.

PROJPLAN selected fictitious crops according to their prices and water demands. Tables 7.3 to 7.5 show the cropping solution for the next three seasons and compare the estimated crop water requirements (CWR) for the cropping pattern selected with those listed per district for 1955-57 tabled in the simulation report for both the "present irrigation practice" and the more efficient irrigation practice, "improved", for which water demands were prepared.

It is significant to note that while the fabricated water consumption values compare favourably with actual practice, a large difference occurs in the three districts where the selected cropping pattern suggested partial cultivation of the available land. In district B, less than half the land is planted in irrigated crops in the first Maha season and slightly over a third in the Yala. Cultivation is also reduced in district C. Districts H, I and M have slightly

TABLE 7.3

The first Maha Cropping Solution

Season 1				CROP							
DISTRICT	Hectares	1	2	3	4	5	6	Ha/dist	CWR/dist	55-56	Improved
1	7000		3078.9			2000.0	1921.1	7000.0	93.7	74.1	56.8
2	39300					11300.0	5916.7	17216.7	216.2	614.6	475.4
3	22700					7700.0	9981.8	17681.8	179.6	290.1	186.5
4	11900		2000.0			9900.0		11900.0	184.7	114.5	115.4
5	4500		4000.0			500.0		4500.0	63.4	43.2	31.3
6	9300		9000.0			300.0		9300.0	135.3	213.4	63.1
7	10100		6004.8			500.0	3595.2	10100.0	119.6	97.3	100.6
8	6100		5300.0			800.0		6100.0	9.0	75.8	48.8
9	5400		4700.0			700.0		5400.0	88.7	51.5	36.1
10	30800		5625.0			9800.0	15375.0	30800.0	349.3	473.4	420.8
Total		0.0	39708.7	0.0	0.0	43500.0	36789.8	119998.5	1439.5	2047.9	1534.8
Yeild		0.0	99395.1	0.0	0.0	2334000.0	217060				
Net Revenue			3511010			7845054	2934420	Crop Revenue Total (Rs):			14290484

TABLE 7.4

The Yala Season Cropping Solution

Season 2				CROP							
DISTRICT	Hectares	1	2	3	4	5	6	Ha/dist	CWR/dist	55-56	Improved
1	7000		3078.9			2000.0		7000.0	92.4	116.3	99.7
2	39300					11300.0	2187.5	13487.5	202.2	616.0	522.2
3	22700					7700.0	5452.0	13152.0	163.9	411.4	334.5
4	11900		2000.0			9900.0		11900.0	165.4	160.4	130.4
5	4500		4000.0			500.0		4500.0	53.3	60.7	53.1
6	9300		9000.0			300.0		9300.0	107.2	137.0	112.2
7	10100		9600.0			500.0		10100.0	113.1	132.9	116.8
8	6100		5300.0			800.0		6100.0	75.5	129.2	107.5
9	5400		4700.0			700.0		5400.0	63.9	87.9	75.4
10	30800				1986.7	9800.0	15800.0	27586.7	266.9	543.0	465.8
Total		0.0	37678.9	0.0	1986.7	43500.0	23439.5	108526.2	1303.8	2394.8	2017.6
Yeild		0.0	95505.0	0.0	14900.0	0.0	138293				
Net Revenue			3728357		74560	7839483	1837244	Crop Revenue Total (Rs):			13479644

TABLE 7.5

The final Maha Cropping Solution

Season 3 DISTRICT	Hectares	HECTARES IN CROP 1	2	3	4	5	6	Total Ha. Cultiv. in Dist.	PROJPLAN CWR of dist.	55-56 CWR of dist.	Improved CWR of dist.
1	7000		3078.9			2000.0	1921.1	7000.0	93.7	74.1	56.8
2	39300					11300.0	5916.7	17216.7	216.2	614.6	475.4
3	22700					7700.0	9981.8	17681.8	179.6	290.1	186.5
4	11900		2000.0			9900.0		11900.0	184.7	114.5	115.4
5	4500		4000.0			500.0		4500.0	63.4	43.2	31.3
6	9300		9000.0			300.0		9300.0	135.3	213.4	63.1
7	10100		6004.8			500.0	3595.2	10100.0	119.6	97.3	100.6
8	6100		5300.0			800.0		6100.0	9.0	75.8	48.8
9	5400		4700.0			700.0		5400.0	88.7	51.5	36.1
10	30800		5625.0			9800.0	15375.0	30800.0	349.3	473.4	420.8
Total		0.0	39708.7	0.0	0.0	43500.0	36789.8	119998.5	1439.5	2047.9	1534.8
Yeild		0.0	99395.1	0.0	0.0	2334000.0	217060				
Net Revenue			3511010			7845054	2934420	Crop Revenue Total (Rs):			14290484

over 10% of their land uncultivated in the Yala season.

When real prices and realistic costs are used, the value of the objective function of the PROJPLAN formulation is the net revenue expected from the cropping pattern and the water allocation of the solution for the next three seasons. The revenue may be easily calculated for the second season to evaluate the irrigation benefits of meeting the current storage targets.

Figures 7.10 to 7.14 show the variations in reservoir level determined by the PROJOP model for the dry year hydrology and the crop water demands determined by the

PROJPLAN cropping pattern. They are compared to the levels determined by the model for the average year application. As expected, reservoir levels are considerably lower in the dryer year. At Victoria, the storage volumes for the final time steps in the dry year are larger than for the mean flow year, likely because PROJPLAN set a higher storage target in the dry year than was arbitrarily set for the mean flow year.

Figures 7.15 to 7.17 show the canal diversions in the dry year compared to the average water year. Diversions in the three canals shown were severely less during the Maha season. Figure 7.18 presents the electrical production calculated from the average head compared to the linear programming estimates of the energy produced.

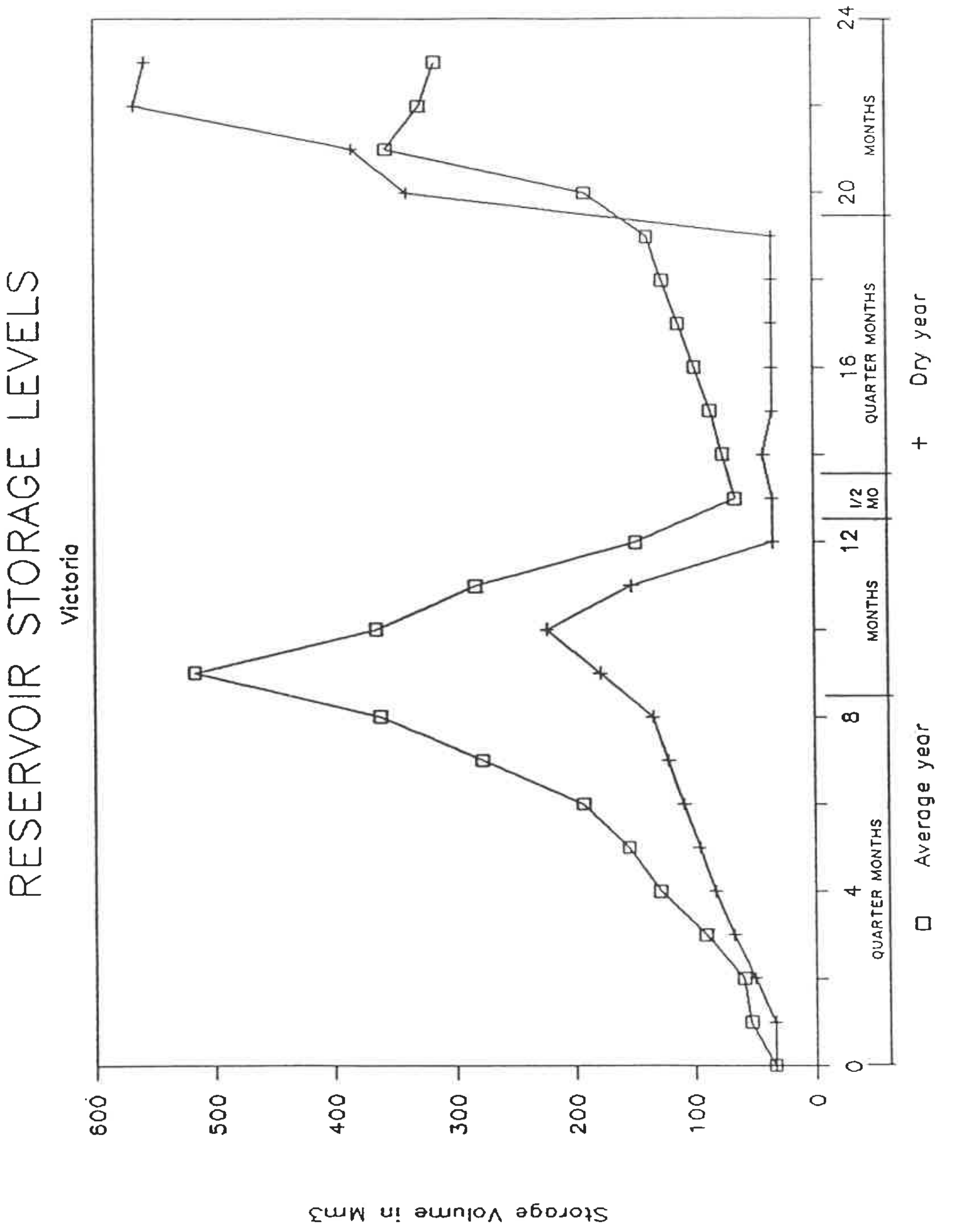

Figure 7.10 Dry year: reservoir storage - Victoria

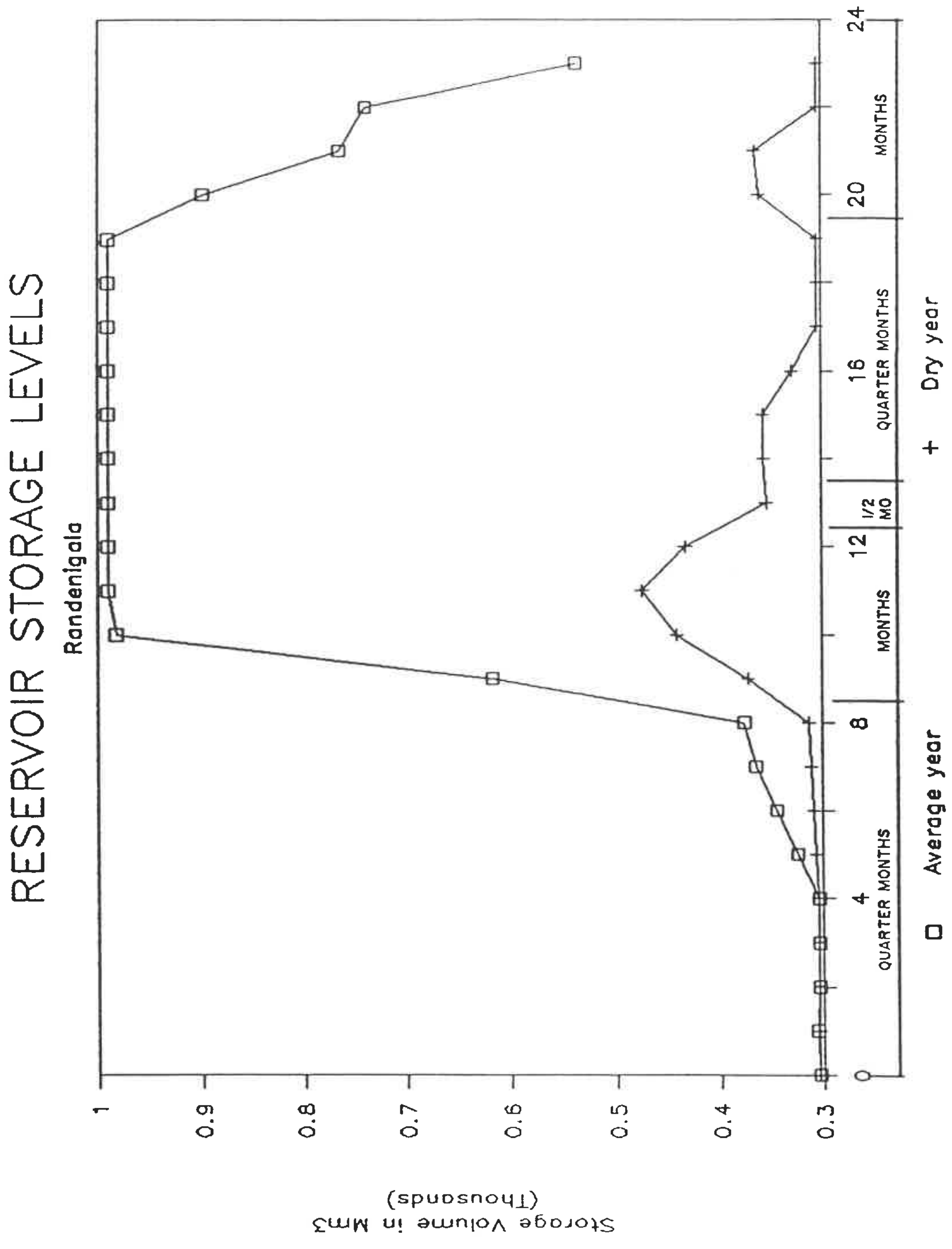

Figure 7.11 Dry year: reservoir storage - Randenigala

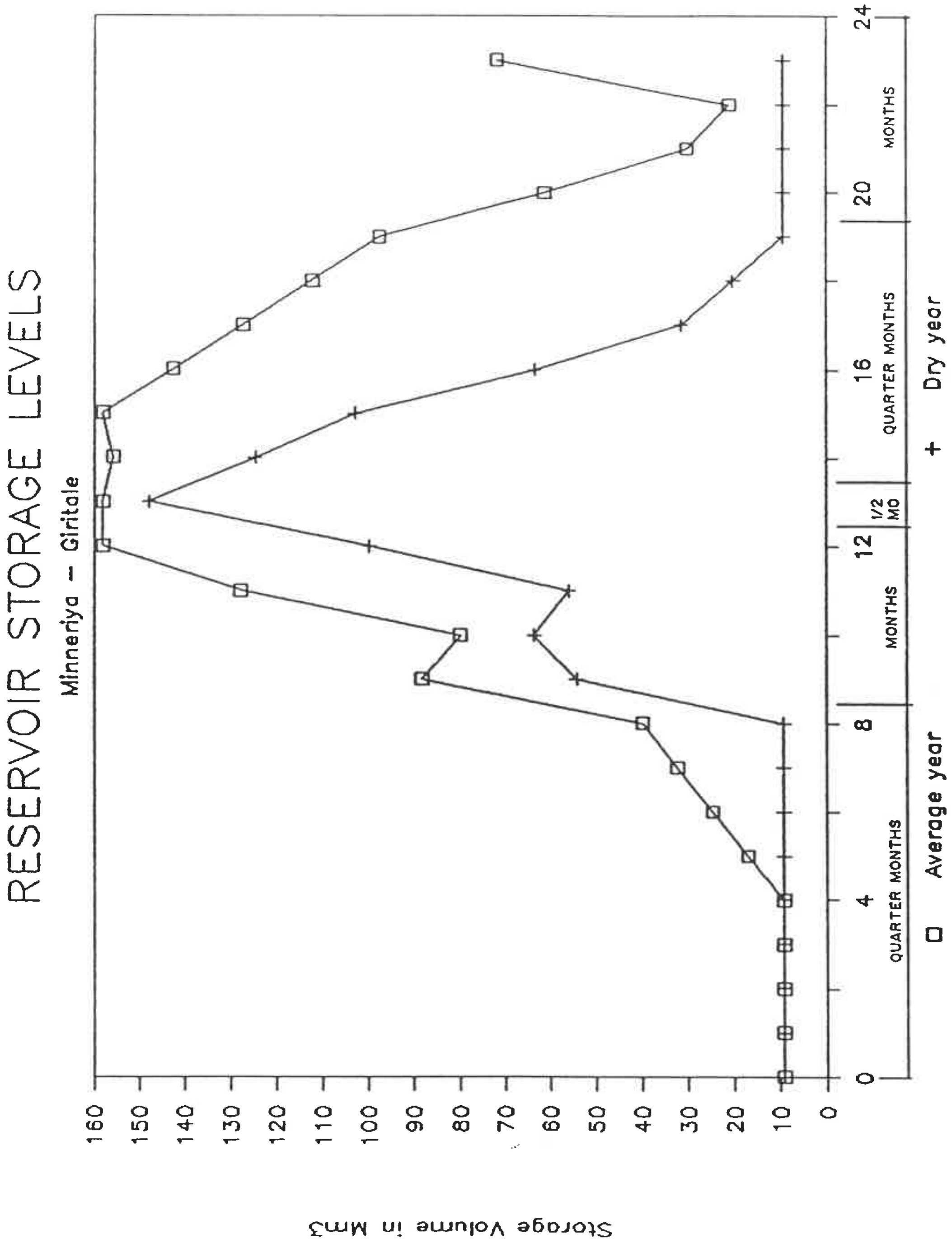

Figure 7.12 Dry year: reservoir storage - Minneriya

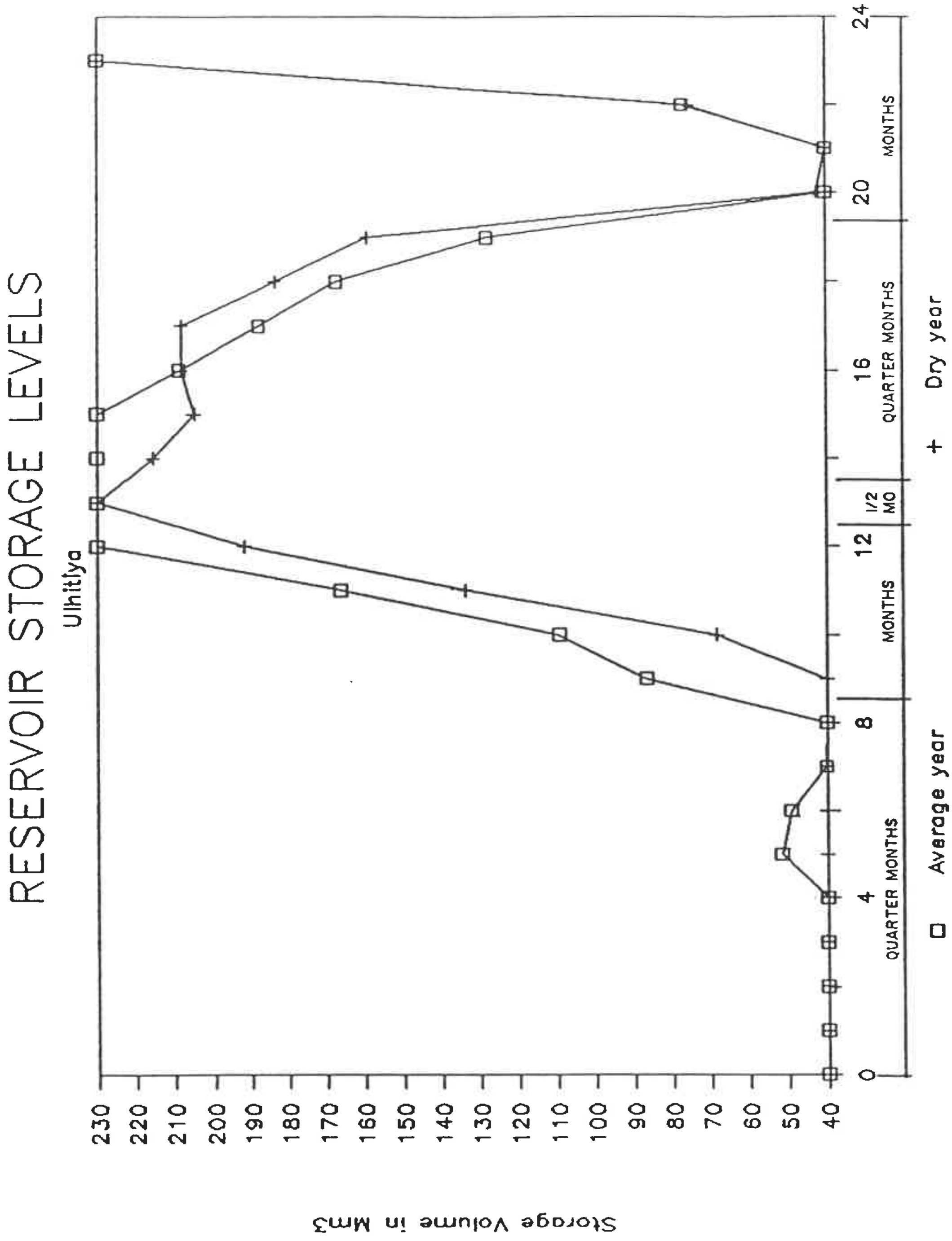

Figure 7.13 Dry year: **reservoir storage - Ulhitiya**

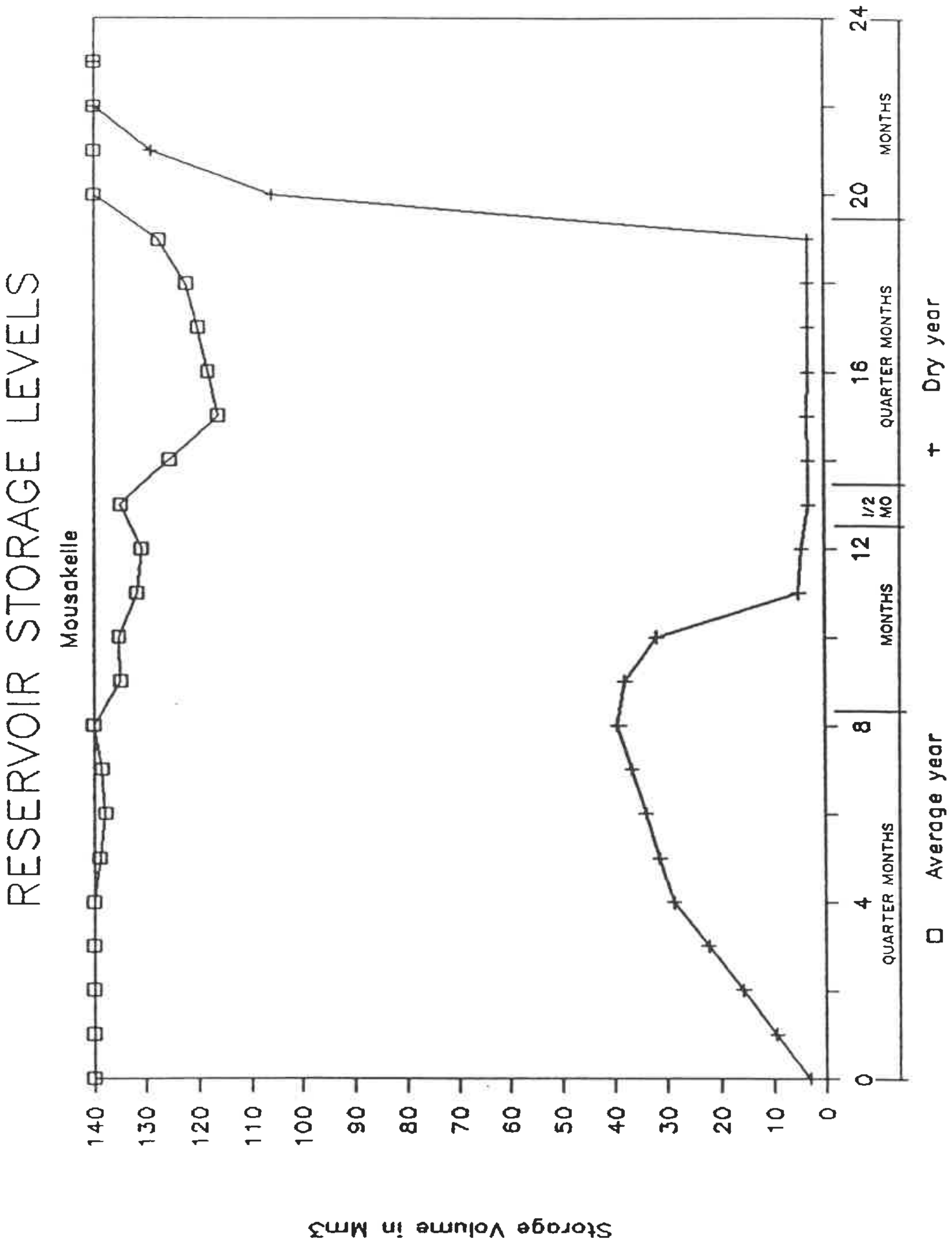

Figure 7.14 Dry year: reservoir storage - Mousakelle

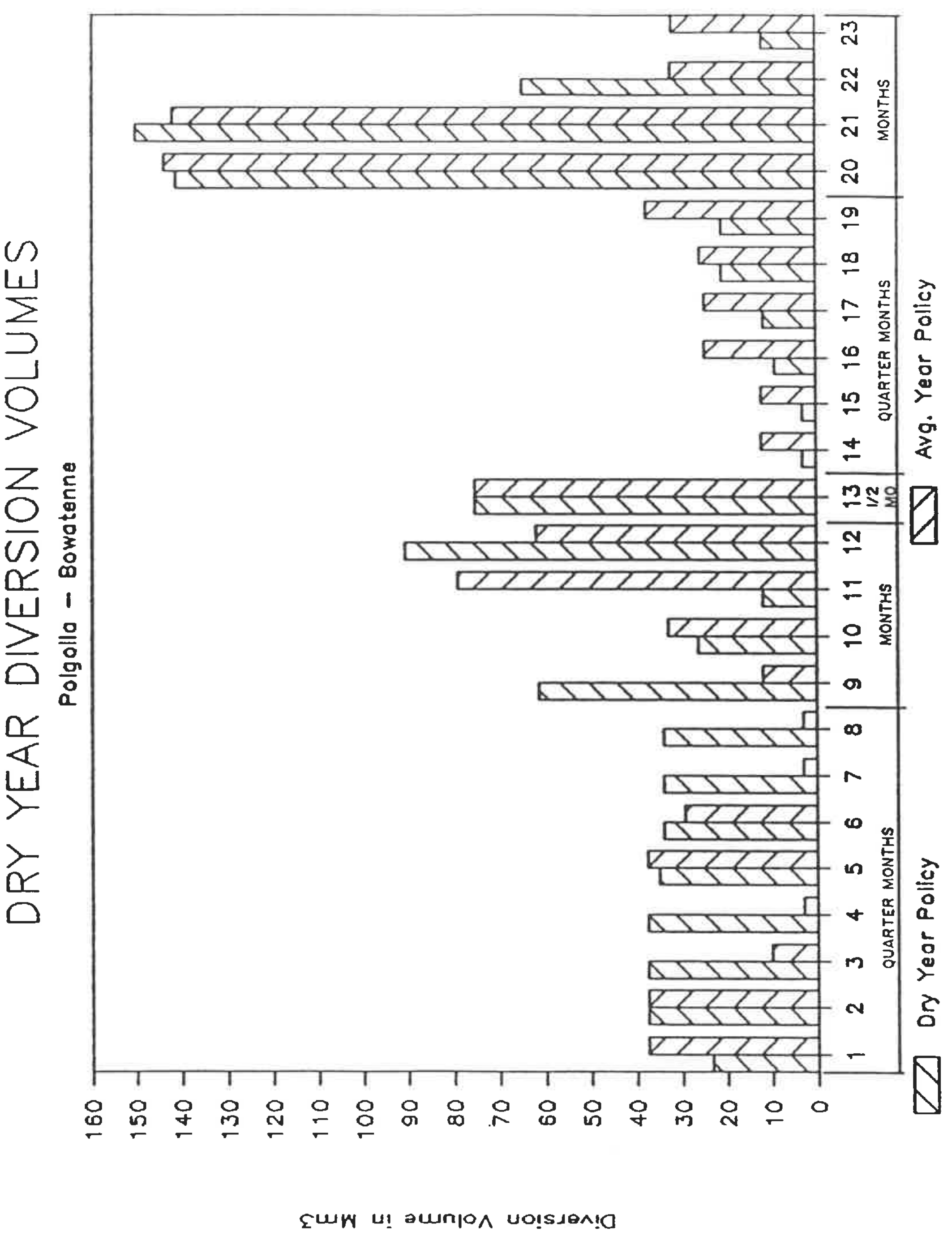

Figure 7.15 Dry year: diversion volumes - Polgolla

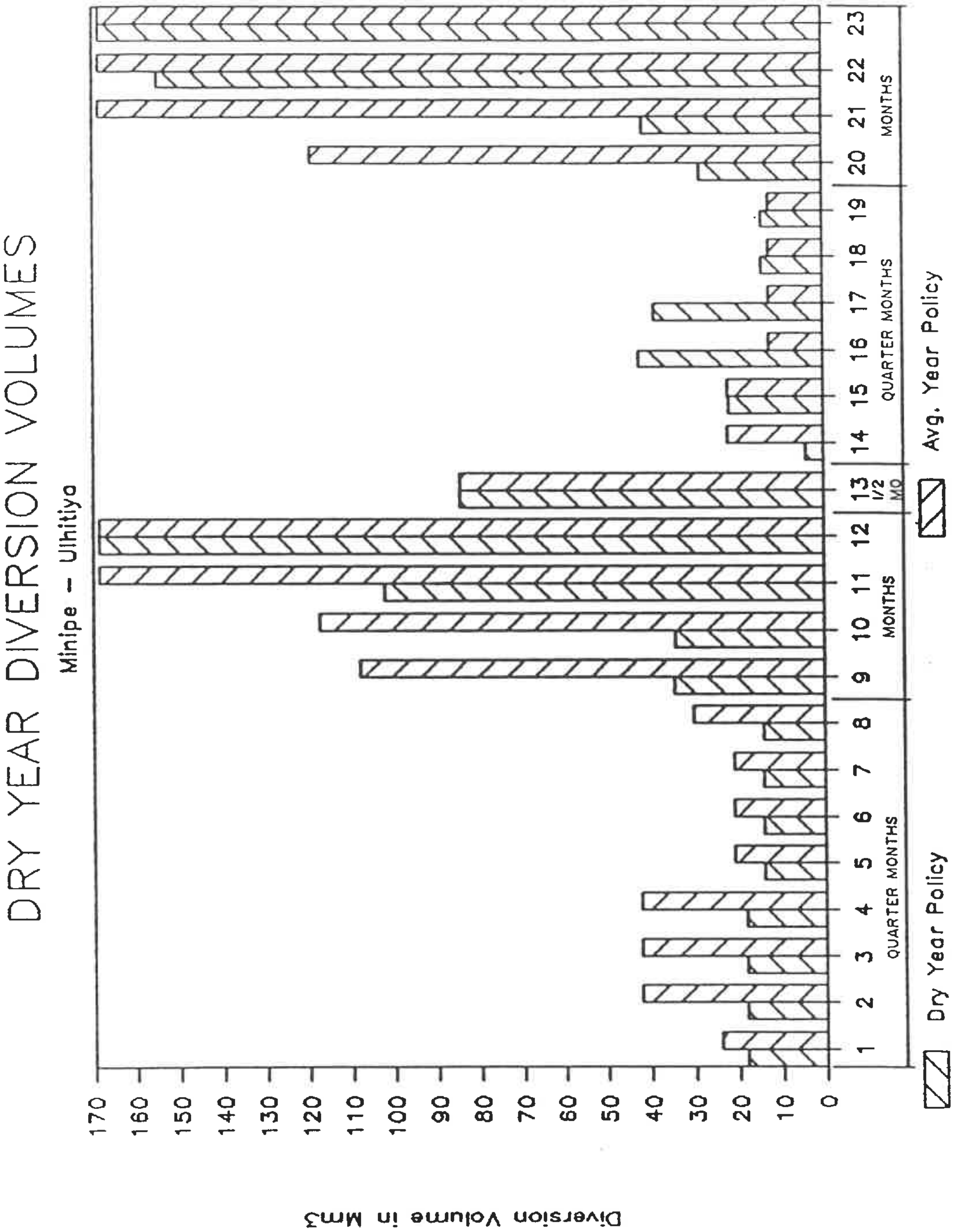

Figure 7.16 Dry year: diversion volumes - Minipe

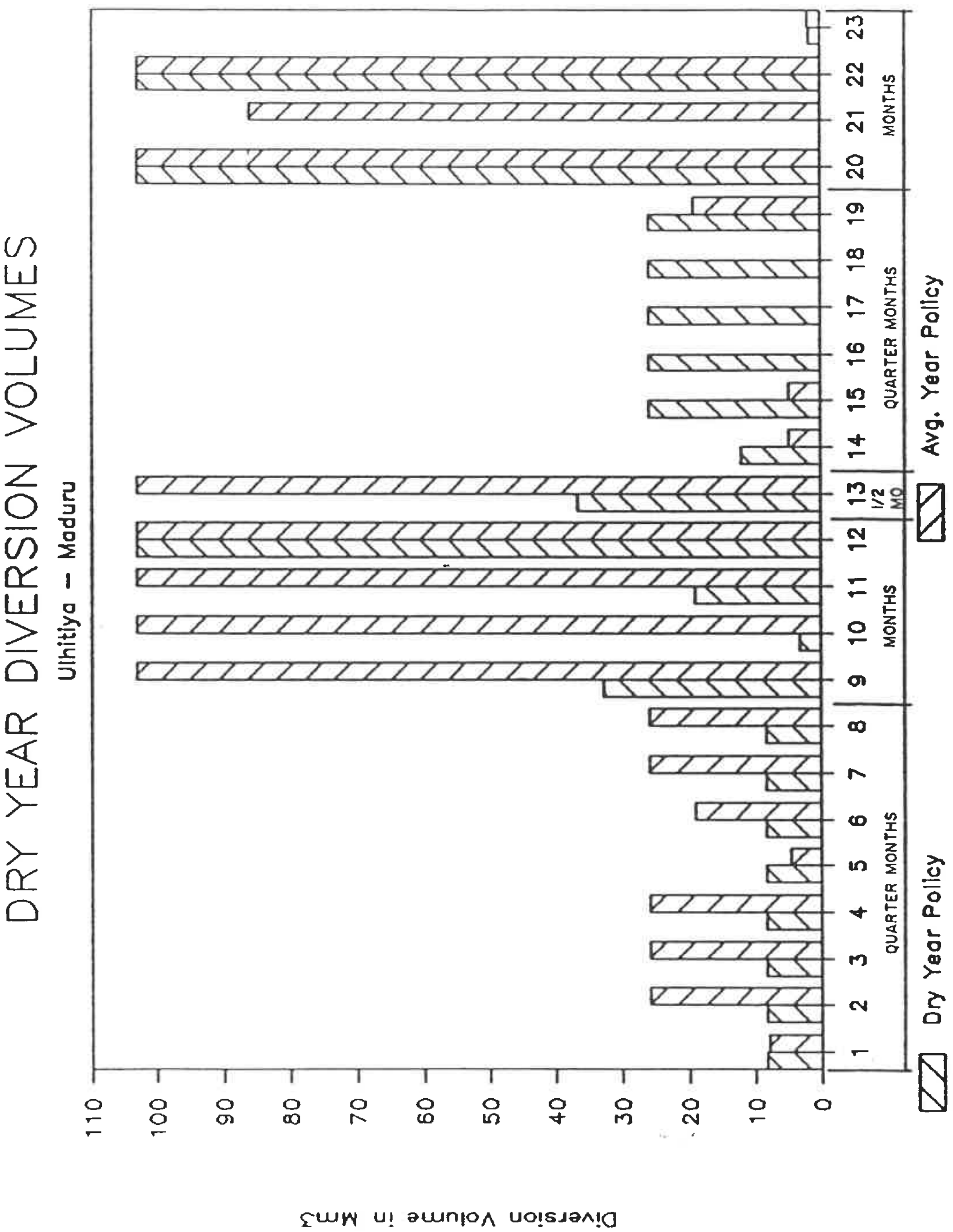

Figure 7.17 Dry year: diversion volumes - Ulhitiya

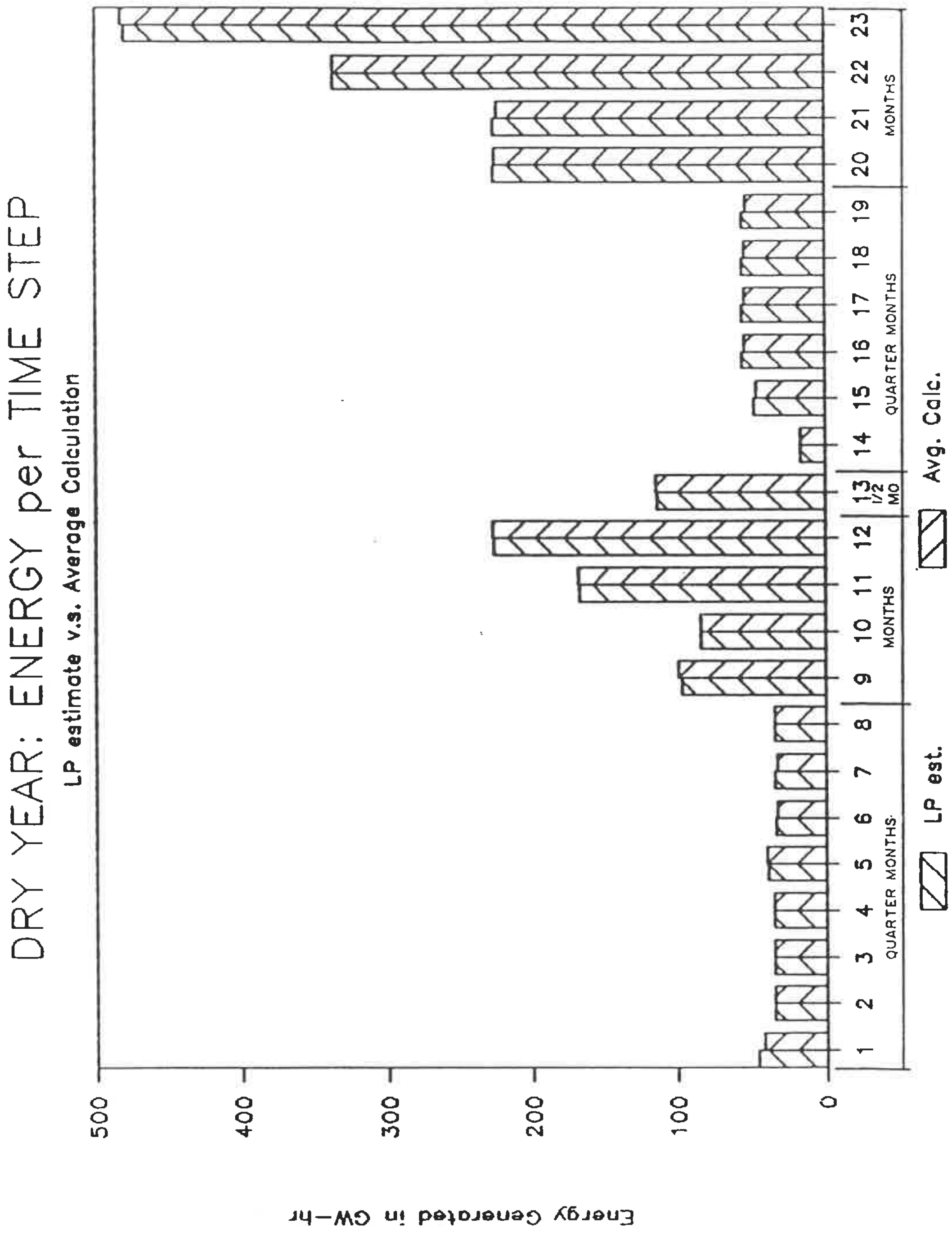

Figure 7.18 Energy production in the dry year

7.4.3 Problems Discovered in the Dry Year Applications

The greatest difficulty was meeting the projected annual firm energy commitment. The trial with the streamflows for an average year had shown that the production of electricity was not evenly distributed through the year, but that more was produced in the Yala season, per month, than in the Maha.

Also, the dry conditions in the Mahaweli basin extended southwest across the divide and Kehelgamu-Maskeliya power plants there had considerable difficulties releasing the 4.5 M^3/quarter month arbitrarily set as a lower turbine limit.

The overall minimum energy limit by time step was relaxed by including new variables to the SAS.OR data file on the mainframe. One variable with a coefficient of plus one for every time step, SHORTi, was included in the POWi constraints.

$$POW_i: \quad \sum^{Np}(ENG_{in}) + SHORT_i \geq pmin_i \ldots\ldots.7.a$$

where:

$SHORT_i$ = decision variable for the shortfall in target energy (GW-hr) from total energy supplied by all power plants in the time step i

$pmin_i$ = Target total energy (GW-hr) desired from all power plants in the time step i

These variables represented the shortfall in energy production per time step from the constant annual firm energy target. In the objective function, OBJ, the SHORTi variables appeared with coefficients of minus one hundred. Because of the high penalty they were given in the objective function, the optimal solution reduced the values of the shortfall variables to as little as possible. The penalty was not too high that irrigation deficits were incurred to produce more electricity. This provided a faster way of obtaining a feasible solution than to lower the production targets by trial and error until a feasible solution was found.

The minimization of the production difference between time steps, an option of this formulation, was used as an alternative way to determine a feasible energy target. However, the annual energy produced by the policy which minimized the difference in the overall energy being produced from one time step to another was considerably less than the annual energy produced by unrestricted production during favourable conditions and a high cost for production shortfalls during critical times. With the option of minimizing the production difference, the amount of energy produced during the critical period lowers the production during the more favourable times.

The minimum energy target for each time step was not the only impediment to a feasible solution for the operating formulation. Although not the driest year on record in the Kehelgamu and Maskeliya River basin, 1955-56 was a dry year. This caused problems with the arbitrarily set operating targets. The power plants on the Maskeliya River were unable to release the minimum turbine release of 4.5 m3/s early in the season, therefore, that minimum was reduced to 2.0 m3/s for those particular time steps. The Castlereigh and Mousakelle reservoirs were unable to end the Maha season full, although they started full. Their target storages were reduced.

7.5 MODIFICATION OF THE ENERGY COEFFICIENTS

All the values for the annual energy produced by the release policies mentioned above are not the values which were estimated by the linear programming formulation. Instead, they have been calculated by multiplying the release in each time step by the head associated with the average storage volume of the time step, multiplied by a turbine efficiency of 85%, multiplied by a specific weight for water of 9.777 KN/m3, divided by 3600 s/hr. As seen in section 5.5.1, the PROJOP model has to linearize the curvilinear function of power generation. Some of its calculations will be on the straight line of Figure 5.1 far from the actual power generation, even when the slope of the straight line approximation is based on an accurate volume of water released in the time step.

The energy from the power plants with constant head was evaluated in a similar calculation, only the head never varied. The average head of the other power plants in a time step was found by interpolating the reported elevation - storage curve and by considering that the rated head occurred at full supply level. These estimates of the energy produced at each power plant in each time step are still not what may actually be produced but are closer estimates than the straight line approximations of the linear programming formulation.

Over an entire monsoon season and for all the power plants, the calculations of the total production by the linear programming formulation are no more than 5% out. However, for a particular power plant and at a particular time step the difference between the two estimates may be considerable, since the large under-estimations are cancelled out by the large over-estimations in the totals. Also some of the large discrepancies in terms of percentages are due to the inaccurate modelling of the curved lower end of the energy production function. The linear programming calculation of the energy generated at a power plant for a time step may be 0.75 GW-hr while the more accurate average head estimate is 0.25 GW-hr. The estimates are only 0.5 GW-hr out in absolute terms, while the percent difference is 300 times. The majority of the energy values per power plant per time step were only 3% out.

The accuracy of the energy estimated by the linear programming formulation, which was selecting the release policy of the solution based on these estimates, could be improved by better estimates of only three constants:

the average head factor, h_{in},

the reference storage volume, sto_{in},

or the correcting factor, c_{in}.

These three values were derived for the test applications from a graph of the head v/s storage for the reservoir of each power plant. Figure 7.19 shows how h_{in}, sto_{in}, and c_{in}, were derived.

The average head factor, h_{in}, was chosen at the midpoint of the operating range of water elevations. The reference storage volume, sto_{in} was chosen at the midpoint elevation. The derivation of the trial values for these constants shows that the magnitude of the correcting factor increases with the turbine release. Prior knowledge of the relative magnitudes of the turbine releases or a few investigative iterative trial runs has to be used to quantify the correcting factor.

The accuracy of the linear programming solution to the value of the correcting factor is more significant in relative

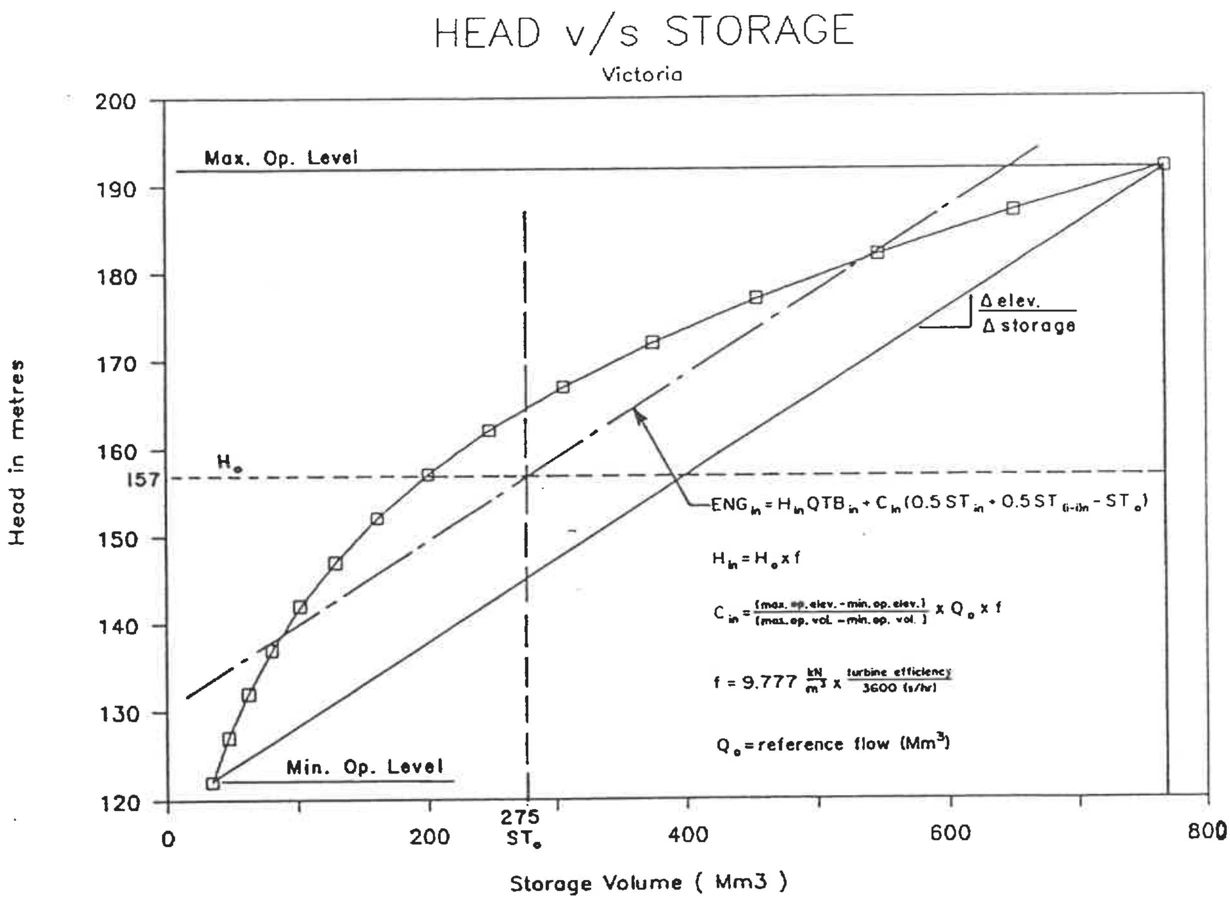

Figure 7.19: Derivation of the Energy Constants

terms than in absolute. A reference volume of 100 Mm^3 per time step gives an adequate estimate for releases from 45 - 175 Mm^3, while a reference volume of 5 Mm^3 will give very inaccurate estimates when the release is just 20 Mm^3/time step, as may be seen on Figure 5.1. Once the calculated release becomes about two or three times greater or lesser than the reference flow, the inaccuracies become high, depending on how far the average storage during the time step is from the reference storage level.

Another major inaccuracy results when the reservoir is low and the value of the correcting factor is high. The straight line estimate shown on Figure 5.1 becomes steeper than the line from a more accurate low correcting factor. When a release is small, the steep line crosses the reference storage at low energy production levels. Therefore, below certain reservoir levels, the estimate of energy produced is negative. As the estimate is a linear programming variable, the solution will not allow negative variables nor a small release when the reservoir is low. Not only does the mismatched value for the correcting factor result in inaccurate estimates of the energy produced, but it also disallows possible favourable release policies. Clearly, too low values for the correcting factor are preferable to too high.

While an inaccurate power coefficient, h_{in}, or representative storage volume, sto_{in}, will decrease the accuracy of the linear programming solution, only the effects of changing the reference flow were studied in the application of the model to the Mahaweli and K-M systems. Better estimates and better policies may be possible from changing the other constants or from using a different formula for deriving the value of the correcting factor.

7.5.1 Changes to the Correcting Factor

The differences in the energy calculated by PROJOP with the correcting factor at 5 Mm^3/quarter month and 100 Mm^3/month and the head-flow estimation factor were as great as 50% in a few cases for a few time steps. The energy estimates of the average water year were inaccurate when compared to the amount of energy calculated from the average storage volume and release during a time step. Therefore, the c_{in} of the power plants and time steps listed in Table 7.6 were revised in the RELEASE.DAT file to determine how the changes would improve the energy estimates.

TABLE 7.6

Adjustments to the correcting factors

POWER PLANT NODE	TIME STEP	LAST REF. Qo Mm^3	TURBINE RELEASE Mm^3	ENERGY EST. GW-HR	ENERGY CALC. GW-HR	NEW REF Qo Mm^3
4	9 M	100	11.9	0.538	1.674	40
19	10 M	100	11.9	8.265	5.640	20
9	10 M	100	19.7	3.702	2.545	20
19	11 M	100	11.9	8.197	5.635	20
19	12 M	100	11.9	8.104	5.628	20
19	13 M	50	5.9	4.064	2.793	20
1	7 Y	100	12.72	7.864	5.684	20
3	7 Y	100	11.9	1.864	4.185	20
9	7 Y	100	11.9	0.799	1.335	20
9	8 Y	100	11.9	1.954	1.473	20
9	9 Y	100	11.9	2.782	1.538	20

M = Maha season Y = Yala season

The values for $c_{in}/2$ multiplying the ST_{in} and $ST_{(i+1)n}$ were changed in the file. In addition the values for the _RHS_ of the EPiNn, where c_{in} multiplies sto_{in}, were revised. The data file for the average year was used, and no other values were modified.

7.5.2 Results of Adjusted Correcting Factors

These adjustments to the correcting factors according to the turbine releases of a previous run did result in better overall accuracy, a slightly better policy in terms of annual energy production, and curious adjustments to the diversion release policy. Table 7.7 shows the improvement in the energy estimates with the adjusted correcting factors. Figure 7.20 shows the improvements to the estimates by comparing overall energy estimates of the adjusted c_{in} to the overall energy calculated using an average head for the time step.

Comparisons of the release policy from the PROJOP run with adjusted c_{in} to that of the test application of the average water year from which the adjustments were made show very little differences in the reservoir levels. A curious difference in the diversion release policy seems to be due to the variation of the correcting factor by time step rather than to the actual magnitude of this constant.

TABLE 7.7

The adjusted correcting factors

POWER PLANT NODE	TIME STEP	REFERENCE FLOW,Qo Mm^3	TURBINE RELEASE Mm^3	ENERGY EST. GW-HR	ENERGY CALC. GW-HR	OLD ESTIMATE GW-HR
4	9 M	40	11.9	1.265	1.672	0.538
19	10 M	20	11.9	5.927	5.634	8.265
9	10 M	20	19.7	2.602	2.545	3.702
19	11 M	20	11.9	5.914	5.635	8.197
19	12 M	20	11.9	5.895	5.628	8.104
19	13 M	20	5.9	3.215	2.793	4.064
1	7 Y	20	11.9	5.518	5.315	7.864
3	7 Y	20	11.9	3.852	4.208	1.864
9	7 Y	20	11.9	1.224	1.272	0.799
9	8 Y	20	11.9	1.220	1.267	1.954
9	9 Y	20	11.9	1.451	1.438	2.782

M = Maha season Y = Yala season

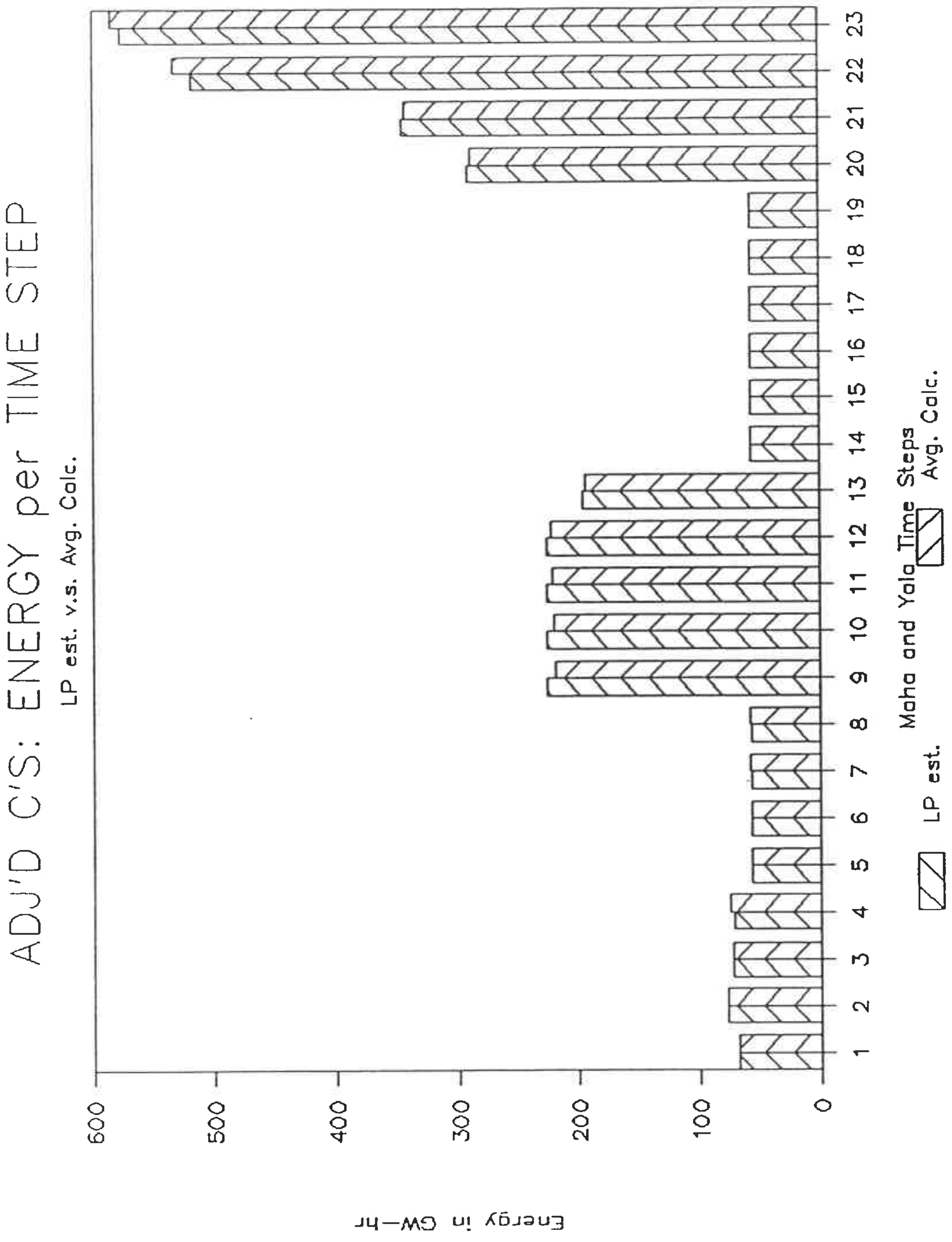

Figure 7.20 Energy production in the average water year with adjusted correcting factors

7.5.3 Problems in Calibrating the Correcting Factors

There is still some large discrepancies between the estimates of the energy produced and the energy calculated for a time step at a power plant. Some were of time steps and nodes whose correcting factors had remained unchanged, others were of time steps and nodes where the release policy had now changed to a value further from the reference flow.

In fact, some earlier tests had found a circular adjustment would occur. Subsequent runs would suggest releases near reference flows which had already been used. The iterations would not converge to a single policy with a highly accurate estimate of the energy produced. Instead, nearly identical global policies were being selected. The subsequent solution may have a better annual production and better total accuracy, but still with highly inaccurate estimates at some sites.

Revising the correcting factor at these locations and time steps would result in marginally better policies. Again in the subsequent runs, some sites would have releases differing from the last solution and their correcting factor, resulting in highly inaccurate energy estimates; the value of the release would be the same as a previous trial.

Obviously, the inaccurate assessment of the benefits from the varying release policies by the linear programming formulation is inadequate for PROJOP to find a better optimal operating policy. The linear programming formulation has its limitations. A simulation model which would accurately estimate the energy production, the irrigation benefits, and evaluate other release policies somewhat the same as the final linear programming solution may be used to adjust the operating policy suggested by PROJOP.

It is up to the user to decide whether the increase in the accuracy of the total energy estimates of the formulation and the changes in the diversion release policy, warrant adjustments to the correcting factor or the other energy constants before investigating the policy with a simulation model.

Overall the PROJPLAN and PROJOP models function as intended. Storage targets are set by the irrigation component, PROJPLAN, and PROJOP is able to realistically model the power production such that the energy produced is a maximum value within the limits set by irrigation demands. The release policies are at a peak of the solution surface as broadly and globally defined by the linear programming data. A simulation model may then more accurately define the quartermonthly release policy.

CHAPTER 8.

DISCUSSIONS

A two-tier model has been developed to assist in the choice of a short-term operating policy for the hydro-electric plants of the Mahaweli Project. The policy allocates water to irrigation as its first priority. To adequately consider the irrigation possibilities, a seasonal planning step determines the cropping pattern which maximizes the net revenue within the storage, delivery and expected hydrological constraints.

The second step identifies the operating policy of the next quarter month which will likely fulfill the irrigation commitments of the rest of the season, allow enough water to be ready for the start of irrigation next season, and will produce electricity as required, not only for the immediate time step, but for the rest of the season. The operating policy of subsequent time steps is revised before actual implementation by updating reservoir levels, inflow forecasts, and when necessary, revising the crop water requirements of the planted crops, or even adjusting the cropping levels of the next season with the concomitant effect of raising or lowering the reservoir storage targets for the end of the season.

Both steps of the model use linear programming formulations. All of the physical components of the project may be included and the full range of possible policies are considered for selection. Only linear relationships may be used, which in the case of the estimation of the energy produced by a policy, introduces some inaccuracy.

8.1 INTELLIGENT APPRAISAL OF THE SOLUTION POLICIES

Sound, experienced engineering judgment must assess every linear programming solution, whether the policy is reasonable and why particular choices were made the way they were. Correctly applied, with constraints and objective functions which reflect real operating restrictions and policy goals, the model will produce reasonable operating policies.

8.1.1 PROJPLAN Final Seasonal Reservoir Levels

One case where the objective function of the seasonal planning step of the model is inadequate was mentioned already. When the cropping pattern is fixed, as was used in the case of the average year, the value of the objective function is also fixed. The allocation of water to the reservoirs, diversion canals, and irrigation districts does not enter the function to be maximized. Therefore, the first feasible solution found, out of the many possible, is the solution reported in the computer print-out.

This is not the case when there are a large number of hectares of prospective crops. In this case, as many high revenue earning crops will be planted as the land and production restrictions allow, to the limit of the water delivery system and the expected volumes of water. The SAS.OR solution procedure may try many water allocation policies before the one which generates the most revenue is found.

Because more revenue comes from more revenue producing crops, the least surplus of water will be used to irrigate in the Amban valley districts or in the lower Maduru basin: the water will not be stored unproductive in a reservoir. The applications reported in the last chapter have used a higher cost for the water delivered to the Amban valley districts to discourage the planting of crops high in water use there. More energy may be generated by water used to irrigate the lower Mahaweli basin.

As long as unused water from one feasible solution may be used productively than in another, the formulation should provide suitable seasonal target storages for the operating model. Again, in years when plenty of water is available and it is the land and production constraints which are driving the solution for the cultivation pattern, the feasible water allocation is not unique. Under these saturated conditions, the seasonal target storages are not expected to cause problems to the operating step, due to the abundance of water.

If there are problems, sound engineering judgment should suggest the corrective actions to take.

8.1.2 Simulation Tests of PROJOP Policy

A detailed, accurate simulation model should be used to confirm the release policy of PROJOP. The solution to the linear programming formulation for the operating step may suggest policies which are infeasible in real life.

A diversion at Minipe or a reservoir may not be possible hydraulically for the level of the river or reservoir. The turbine releases may not be conducted at the average head and the energy produced in the time step may be less than what the model estimated. The minimum electrical production for that time step may not be met.

Only by testing the solution policy by calculating the actual reservoir and river levels and the actual generation schedule in a simulation model, will the user discover beforehand that some adjustment to the coefficients in the model are necessary for it to advise a feasible policy.

A detailed, accurate simulation model is useful, as well, in selecting the optimal turbine release policy when the linear programming formulation has reached the limit of its accuracy in the estimation of the benefits due to the electrical production of a policy. As mentioned in the previous chapter, adjusting the correcting factor for the magnitude of the release in the previous trial may result in a series of turbine policies which fluctuate, with little difference in the total energy production estimated.

A simulation model with a better ability to calculate the energy produced by a policy will assist in selecting which policy is actually producing the most electricity while still meeting the irrigation demands. The use of the two models is symbiotic, one finds probable good policies, the other accurately evaluates which is the best.

A detailed, accurate simulation model may also be used to investigate the consequences of executing a policy when the streamflows predicted for the reservoir system differ, either in magnitude or timing or both. The risks of the operation may be assessed and compared with traditional or rule curve operation. This type of test may be done weekly, comparing the consequence of releasing the suggested immediate volumes, or less often, using the anticipated releases as well as the immediate releases.

A simulation model may be expensive to run; the user may try many hydrological sequences with PROJOP, itself. Those sequences where there were differences in the solution policies are the most likely to have high risks if the original policy was executed under those conditions. The risks may then be assessed by the simulation model for those hydrological sequences.

8.1.3 Modifying the Formulations of the Model

The model may be improved without any changes to the computer coding of the data preparation programs, simply by inclusion or exclusion of the optional terms in the objective function and constraints, and by adjustment of the coefficients. The programs and the formulation were designed to reflect not only the changing nature of the reservoir configuration but also changing priorities.

The SAS.OR sparsedata format permits quick modifications to the model. Variables or constraints may be entered in any order. New variables or constraints may be listed at the end of the datafile, a previously defined variable may be repeated in a new constraint listed at the end. The user should avoid constraints with only one variable.

Modifications to the PROJPLAN-PROJOP model may add restrictions so that the solution policy will be feasible in the real life conditions of that particular period. Adjustments may be necessary so that the energy estimates are more accurate for a particular critical period. Other modifications may be used so that the feasible reservoir levels in PROJPLAN reflect the operational goals of electrical production on the Mahaweli River or so that the estimated energy production under good conditions is not greater than the transmission capacity. None of these changes should be undertaken lightly.

8.1.4 Dangers of Misrepresenting the Situation

Whenever the coefficients are entered for a variable in a constraint, or a particular option is selected over another, or any modifications are included to obtain a quick remedy to an initial infeasible solution, the user should beware. Thoughtful consideration should ensure that the constraints in the model truthfully represent reality and that their calculation will be as accurate as possible, given their linear nature.

The actual physical and operating conditions should define the limits of a diversion in a time step or the capacity of a reservoir. When a quick remedy is sought, one must determine what one variable is irregular and why, and apply a constraint which deals with that particular need. The objectives should reflect the desired goals of each step of the model.

Special care must be taken with the weighted objective function that the values chosen not leave only one feasible policy, pre-determined by the user. Models may be most valuable when they reveal a policy, which would otherwise be completely disregarded and discounted, as being optimal, or when the user discovers new relationships between the system's components which were never evident before.

The linear programming formulations, to be of any value, must be allowed to select, from a full range of possible policies, the release policy which best satisfies the stated goals of the operation. They must not be unduly restricted to certain policies, nor unduly favour any. The objective function must mirror the stated goals of the operation, and be able to judge each policy on its merits alone.

8.2 FURTHER TESTING AND RESEARCH

To date the model has been applied to the simplified configuration of the Mahaweli Authority - Acres International 32-year simulation planning model using the information available in its report. Only static inflow sequences have been tried. Clearly, its performance and versatility have not yet been tested. As the model is used, gains acceptance, and is, perhaps, mis-applied, faults and omissions will come to light.

It is hoped that the engineers of the Mahaweli Authority will take such an opportunity to improve the conception of the formulations of the two steps of the model and not just discard the linear programming model. Before the model may be used in normal operation, further testing is required with accurate information.

8.2.1 Energy Constraints

The estimate of the energy produced selects one release policy as being better than another. Obviously, the more accurate the energy estimate by the linear constraint, for the conditions of the particular time step and power plant, the better the policy selection. Only the effects of the correcting factor on the accuracy of the estimate and the variations in the policy have been investigated.

It may well be that different reference storage volumes, shifting the constant for the head for different time steps, or that deriving the correcting factor directly from a plot of observed energy versus storage may produce better, more easily quantifiable constants with more accurate energy estimates. Further tests should confirm or improve the derivation of the values for these constants.

8.2.2 Sensitivity to Inflows

The model utilizes continuously revised, deterministic predicted inflows. Predictions, especially of the weather, often prove erroneous. A model which selects the same or nearly the same course of action for a wide variation of predictions provides very safe, sound policies unless the model is overlooking, to some degree, the considerable gains or risks associated with the hydrological conditions of some of those predictions.

None of the recent trials attempted to investigate the response of the model to streamflows differing from those predicted. How conservative are its policies? How easily is the release policy influenced by possible gains from estimated flows higher than eventual streamflows? These questions need to be answered before giving the last word on the model.

The Mahaweli Authority in Sri Lanka may use a simulation model to translate the policies of the model into reservoir levels, irrigation deliveries, electrical production, etc. for different predicted streamflow sequences.

They know the accuracy rate of streamflow predictions, and how extremely the conditions might vary. They have the experience and data to assess whether a multi-variate lag regression model may predict future streamflows based on past already observed flows. They may evaluate how well long-range weather forecasts may revise flows estimated by regression and the reliability of those predictions for generating quarter monthly local inflows for a full season ahead, or for the sub-seasons of three PROJPLAN seasons.

This type of knowledge is required for an adequate investigation of the effect of misjudging the anticipated hydrology and of the tolerance of the linear programming solution to deviations in the predicted inflows. Due to the familiarity of local conditions required, this research has been left to the user.

8.3 CONCLUDING REMARKS

PROJPLAN may be a powerful planning tool for the Mahaweli Authority who may use it to consider three seasons into the future, to determine a beneficial cropping pattern for the current season, and to set reservoir storage targets for the beginning of the next season. As forecasts of available water become more definite, and at the beginning of each new monsoon season, PROJPLAN will be re-run, to revise the current cropping and storage target plans.

While the estimates of water available in the future or crop prices may be inaccurate, it is not advisable to regulate the reservoirs with a myopic vision of only one season. An optimizing model with no consideration for the future is likely to advise immediate total consumption, a policy which may leave the next season dry and totally unproductive. Used judiciously, PROJPLAN may quantify the irrigation returns of water policies and assist in finding the policy which maximizes irrigation returns.

PROJOP may prove a quick and efficient method for determining a weekly base release policy which may be tested and improved by a more accurate simulation model. The linear programming formulation ensures that the solution is obtained from the entire range of possible policies. The linearization of the energy function retains the influence of variable reservoir levels, minimizes the inaccuracies of the estimation (which may result in a non-optimal solution for the real conditions) and may be quickly and efficiently solved by the computer solution procedure.

Due to its computer speed and ease of application, the formulation and data preparation program may be used to find a series of policies for any number of inflow, operating, and demand conditions. Repetitive updating with current reservoir levels and the latest streamflow and crop water demand forecasts may adapt the operating policy to changing future conditions.

By choosing the appropriate penalty and benefit weights, including the applicable terms and constraints available in PROJOP, and by confirming the solutions with a simulation model, the Mahaweli Authority would be assured that the quarter monthly solutions are the best given the current reservoir levels, the irrigation and electrical demands, the projected inflows and the projected water needs of the next season.

Both steps have easy to follow data preparation programs which write the datafiles of the formulations for solution by the LP procedure of SAS.OR. The expansion of irrigation districts, the construction of new reservoirs and the variations with time of any operating characteristics pose no problems.

Not only are the data preparation programs flexible and easy to use, allowing correction of data, additions and deletions, variations over time and the selection of optional constraints and terms in the objective function, but also the sparsedata format of the solution package allows for the inclusion of new variables in existing constraints and the addition of completely new constraints.

The result is a flexible and computer-efficient model, applicable to a number of situations and goals. After the actual restrictions and goals of an optimal release policy for the year ahead have been identified, quantified, and incorporated into both steps of the model, the user may be satisfied that the results are not only optimal for the formulation used, but also applicable to the operation of the Mahaweli Project for irrigation and hydro-electric purposes in the near future.

A practical, simple-to-use hierarchial computer model has been developed to assist in the selection of a short-term operating policy for irrigation and power reservoirs according to selected objectives, current conditions and forecast hydrology.

CHAPTER 9.

CONCLUSIONS

The performance objectives of a regulation model for the specific application to the irrigation and hydro-electric reservoir system in Sri Lanka are explained at the end of chapter two. The model was to:

1) recognize the importance of irrigation use over hydro-electric generation,
2) permit the allocation of water to occur at different times of the year,
3) include new power plants, reservoirs, and irrigation districts once they become operational,
4) allow for seasonal and periodic variations in capacities due to flood control restrictions, maintenance schedules or seasonal conditions,
5) be simple to use yet still comprehensive in scope
6) reflect the uncertainty associated with the streamflows in the monsoon climate.

The model has been examined for its attainment of these objectives during its application to the two test years and the previous tests with alternate configurations.

9.1 THE DUAL PURPOSES

The reservoir regulation model developed was required to optimize the weekly turbine releases while satisfying crop water requirements for irrigation water. For rice cultivation in the monsoon climate of Sri Lanka, the seasonal irrigation allotment effectively occurs when the crops are planted. This creates a dual decision timeframe in addition to the dual purpose of the reservoir operations.

For social and political reasons, the benefits from irrigation are paramount. The desired model needed to place more priority on the agricultural benefits, and needed to consider the allocation of water for irrigation before the short term benefits of generating more electricity. The two-tier structure of PROJPLAN and PROJOP, where the weighted objective function of the operating step is restricted by the target storages and irrigation demands from crops selected by the planning step, ensures that this occurs.

9.2 SIMPLE AND FLEXIBLE SOLUTION TECHNIQUES

Many reservoirs, tanks, irrigation districts and power plants comprise the Mahaweli Project. Not all of these may need to be considered as some are not yet constructed nor fully planned and some have fallen into disuse. The varying and large quantity of structural components require optimizing techniques which are flexible and effective for large systems. The linear programming algorithm regards the entire feasible operating range of a large number of components.

The solution package chosen, the sparsedata option of SAS.OR, permits quick and efficient consideration of numerous varying coefficients. The data preparation programs further assist in the easy application of the model. Due to the high degree of flexibility included, the model is not limited to use by the Mahaweli Authority, but may be used for any reservoir system which has irrigation uses of greater importance than the generation of electrical power.

The linear programming algorithm requires sacrificing some accuracy when estimating the amount of energy produced at a site where the head varies considerably with the storage. Non-optimal solutions may occur when the constants used to linearize the energy function are selected too high and possible policies are disregarded as infeasible due to the estimation error. Iterations may be used in PROJOP to improve the energy estimates, however a detailed simulation model with nonlinearized functions will eventually be necessary to confidently assess the optimal weekly policy.

9.3 UNCERTAINTY

The monsoon climate of Sri Lanka is erratic and difficult to predict. The uncertain environment requires continual update and revisions of forecasts, crop water requirements, crop prices, and reservoir levels. The model has been designed to provide an operating policy for the immediate time step while looking ahead into the future. Before the next time step, the model is updated, and another solution found. This adaptive, repetitive technique allows the model to respond to previously unexpected hydrological or operational conditions as rapidly as they become known.

The response of the model to varying hydrological conditions as the season progresses has not been tested. The user may gain more confidence in a final operating policy after testing various streamflow predictions. PROJPLAN and PROJOP may be submitted differing predicted streamflow sequences; a comparison of the resulting solutions may suggest a least risk policy. Alternatively, policies found by the model may be tested with a simulation model under various flow conditions to evaluate the release policy with the most benefits and least risks.

9.4 RESULTS OF THE APPLICATIONS TO TEST YEARS

The tests of the model done using perfectly predicted forecasts of a year of monthly average inflows and the year 1955-56, show that the model performs satisfactorily. One third more of the annual firm energy, 2 711 GW-hr, was produced by the policy selected and no irrigation deficits occurred during the average year.

During the expected dry year, the worst on record, less land was cropped, little irrigation deficits occurred, and the annual energy produced was 2 519 GW-hr. During these and previous tests on different configurations, the model provided the desired planning and operating policies once the proper constants and constraints were included for the modelling conditions.

9.5 CONCLUDING APPRAISAL

The two step planning and operating model has fulfilled the objective of comprehensive, flexible, and easy use. It maintains the relative ranking of its dual purposes. With practice in the selection of the energy constants, the user may become adept at identifying those which produce truly optimal policies for the actual and predicted conditions.

The model should be tested under various predicted streamflows and crop requirements with the release policies evaluated by an accurate simulation of the benefits and losses from agricultural and electrical returns. Research into the performance of the model under various hydrological conditions and predictions, accurately quantified by a simulation model, is required. Further research may decide how reliable the adaptive, repetitive technique proves to be in choosing the optimal policy in uncertain climates.

PROJPLAN and PROJOP combined provide the user with a computer tool which determines the most hydro-electricity which may be generated in the week, while satisfying the irrigation demands of currently planted crops and those of future seasons, according to the latest forecasts of water supply and demand.

REFERENCES

(1) ADIGUZEL, R. I., and COSKUNOGLU, O., 1984 "A Descriptive Decision Process Model for Hierarchical Management of Interconnected Reservoir Systems", Water Resources Research, Vol. 20, No. 7, pp. 803-811.

(2) BECKER, L., and YEH, W. W-G., 1974 "Optimization of Real Time Operation of Multiple Reservoir Systems", Water Resources Research, Vol. 10, No. 6, pp. 1107-1112.

(3) BECKER, L., YEH, W. W-G., FULTS, D. and SPARKS, D., 1976 "Operations Models for the Central Valley Project", Journal of the Water Resources Planning and Management Division, ASCE, Vol. 102, No. WR1, pp. 101-115.

(4) BELLMAN, R. E., 1957 "Dynamic Programming", Princeton University Press.

(5) BENDERS, J. F., 1962 "Partitioning Procedures for Solving Mixed Variables Programming Problems", Numerische Mathematik, Vol. 4, pp.238-252.

(6) CHATURVEDI, M. C., and D. K. SRIVASTAVA, 1981 "Study of a Complex Water Resources System with Screening and Simu-

lation Models", Water Resources Research, Vol. 17, No. 4, pp. 783-794.

(7) DAGLI, C. H., and MILES, J.F., 1980 "Determining Operating Policies for a Water Resources System", Journal of Hydrology, Vol. 47, No. 34, pp. 297-306.

(8) EISEL, L. M., 1981 "Chance Constrained Reservoir Model", Water Resources Research, Vol. 17, No. 4, pp. 783-794.

(9) FULTS, D. M., HANCOCK, L. F., and LOGAN, G. R., 1976 "A Practical Monthly Optimal Operations Model", Journal of the Water Resources Planning and Management Division, ASCE, Vol. 102, No. WR1, pp. 63-76.

(10) GILBERT, K. C., and SHANE, R. M., 1982 "TVA Hydro-scheduling Model:Practical Aspects", Journal of the Water Resources Planning and Management Division, ASCE, Vol. 108, No. WR1, pp. 1-19.

(11) GILBERT, K. C., and SHANE, R. M., 1982 "TVA Hydro-scheduling Model: Theoretical Aspects", Journal of the Water Resources Planning and Management Division, ASCE, Vol. 108, No. WR1, pp. 21-36.

(12) GRYGIER, J. C., and STEDINGER, J. R., 1985 "Algorithms for Optimizing Hydropower System Operation", Water Resources Research, Vol. 21, No. 1, pp. 1-10.

(13) HALL, W. A. and BUTCHER W. S., 1968 "Optimal Timing of Irrigation", Journal of Irrigation and Drainage Division, ASCE, Vol. 94, No. IR2, pp. 267-275.

(14) IRRIGATION DEPARTMENT - SRI LANKA GOVERNMENT, 1974 "Annual Rainfall and Surface Runoff", Colombo.

(15) LAUFER, F., and MOREL-SEYTOUX, H. J., 1979 "Optimal Weekly Releases from a Seasonal Reservoir", Water Resources Research, Vol. 15, No. 2, pp. 383-398.

(16) LECLERC, G., and MARKS, D. H., 1973 "Determination of the Discharge Policy for Existing Reservoir Networks under Differing Objectives", Water Resources Research, Vol. 9, No. 5, pp. 1155-1165.

(17) MAHAWELI AUTHORITY and ACRES INTERNATIONAL, 1986 "Studies of Operating Policy Options", Internal report.

(18) MAJI, C., and HEADY E. O., 1980 "Optimal Reservoir Management and Crop Planning under Deterministic and

Stochastic Inflow", Water Resources Research, Vol. 16, No. 3, pp. 438-443.

(19) MARINO, M. A., and MOHAMEDI, B., 1983 "Reservoir Operation by Linear and Dynamic Programming ", Journal of the Water Resources Planning and Management Division, ASCE, Vol. 109, No. WR4, pp. 303-330.

(20) MEREDITH, D. D., 1975 "Optimal Operation of Multiple Reservoir Systems", Journal of the Hydraulics Division, ASCE, Vol. 10, No. 2, pp. 299-312.

(21) MOHAMEDI, B., and MARINO, M. A., 1984 "Reservoir Operation: Choice of Objective Function", Journal of the Water Resources Planning and Management Division, ASCE, Vol. 110, No. WR1, pp. 15-29.

(22) MOHANTY, R. P., SAHU, K.C., and RAO, B. S. R., 1976 "Optimization Studies on Some Operational Aspects of a Multiple Purpose Reservoir System: a Case", Irrigation and Power, October.

(23) NORRIE, D., 1987, personal communication.

(24) PEREIRA, M. V. F., and PINTO, L. M. V. G., 1985 "Stochastic Optimization of a Multireservoir Hydroelectric System: a Decomposition Approach", Water Resources Research, Vol. 21, No. 6, pp. 779-792.

(25) REVELLE, C., JORES, E., and KIRBY, W., 1969 "The Linear Decision Rule in Reservoir Management and Design", Water Resources Research, Vol. 5, No. 4, pp.767-777.

(26) RHENALS, A. E., and BRAS, R. L., 1981 "The Irrigation Scheduling Problem and Evapo-transpiration Uncertainty", Water Resources Research, Vol. 17, No. 5, pp. 1328 - 1338.

(27) SAMARATUNGE, P. J., and SAMARATUNGE, T., 1980 "Optimal Operation of the Kehelgamu Oya - Maskeliya Oya River Basin Reservoir System, Sri Lanka", in International Conference on Water Resources Development, Taipei, Taiwan, pp. 165-173.

(28) SAS INSTITUTE INC., 1986 "SAS.OR Program", Carey, N.C., zip 27511, U.S.A.

(29) SIGVALDASON, O. T., 1976 " A Simulation Model for Operating a Multipurpose Multireservoir System", Water Resources Research, Vol. 12, No. 2, pp. 263-278.

(30) SIGVALDASON, O. T., 1987 "Acres seminar",

(31) SOLIMAN, S. A., and CHRISTENSEN, G. S., 1986 "Application of Functional Analysis to Optimization of a Variable Head Multireservoir Power System for Long-term Regulation", Water Resources Research, Vol. 22, No. 6, pp. 852-858.

(32) THOMAS, H. A. Jr., and REVELLE, R., 1966, "On the Efficient Use of High Aswan Dam for Hydropower and Irrigation", Management Science, Vol. 12, No. 8, pp. B-296 - B-311.

(33) TURGEON, A., 1980 "Optimal Operation of Multi-reservoir Power Systems with Stochastic Inflows", Water Resources Research, Vol. 16, No. 2, pp. 275-283.

(34) TURGEON, A., 1981 "Optimal Short-term Hydro Scheduling from the Principle of Progressive Optimality", Water Resources Research, Vol. 17, No. 3, pp. 481-486.

(35) YEH, W. W-G., 1985 "Reservoir Management and Operations Models: a State-of-the-art Review", Water Resources Research, Vol. 21, No. 12, pp. 1797-1818.

APPENDIX A

A.1 THE DATA PREPARATION PROGRAMS PROJPLAN AND PROJOP

The two FORTRAN programs, which have been written to prepare the data files required by SAS.OR for the formulation of PROJPLAN or PROJOP, are explained here. A separate section will explain in detail the format requirements of the files used by the interactive programs to prepare the SAS.OR data files, another section will list possible error messages and what they mean, and the final section explains the format of the SAS.OR data files and their use. A complete listing of the two FORTRAN programs and explanations of every variable term used are included in a second appendix. It is hoped that the user will feel confident to amend or revise the formulations and the FORTRAN programs as necessary.

A.1.1 How to Start

Both personal computer programs are interactive. The user selects actions from menus and responds to prompts to review, update, and write the data as desired. Both have a similar structure and appearance. Every attempt has been made to have the programs user-friendly and self-explanatory, however, a few words of caution should be heeded.

To run the programs, which are compiled into executable files ending with .EXE and are in illegible binary code, the user simply types PROJPLAN or PROJOP at the DOS prompt and strikes the RETURN to enter in the message. A brief description of the program's function will then appear on the screen, and the user may then follow the programs through, answering every question as it appears on the screen, selecting data files on diskettes, the next desired action, an option in the formulation or directly entering numbers and names.

There is a main menu in each program to which control returns, until the user signifies that all the data has been corrected. After that point, there is no turning back until after the SAS.OR data file has been written on the diskette. At any time the user may depress CTRL-BREAK to end the program and return to DOS, losing any data corrections that were not saved.

The programs also terminate abruptly, with no recourse, if there is no file on the diskette in the specified drive of the correct name, including the extension. PROJPLAN always looks for CROPS in the B: drive at the beginning of a run, PLAN.DAT or RELEASE.DAT are always written to the drive B: at the end of a successful run. At the start of a PROJPLAN or PROJOP run, these files may be empty, but their names, at least, must be on the specified diskette in the B: drive or the run will abruptly terminate.

The other files have user specified names and may be elsewhere than the B: drive. After starting either program, the program diskette may be removed and the drive used for a file diskette. Files may also be placed on the program diskette if there is sufficient space.

When specifying a file name the user is limited to 10 characters, including the extension and the period, and to save a file the name must also already appear on the diskette. The new file may be empty, but the name must be OLD.

Another abrupt termination will occur if the user has typed in an alphabetical character for the response to a question requiring a number. Even answering a number with a decimal point when an integer number was expected will cause a sudden end to the session.

Other abrupt terminations occur when a file entered has numbers or letters in the wrong place, or there is a problem with a write format not matching the data type of the output. The user should not encounter the latter difficulties until revising the FORTRAN programs.

A1.2 Default Actions

To speed the session, the user may take advantage of the built-in default responses to the questions. Some questions do indeed require a user-specified reply, but many more were felt to have a likely response which was set as the default value. For these questions, whenever the user does not respond with the letter requested in the question line, the opposite action occurs. For example, if the question is:

Do you wish to correct the values? Answer Y to do so

The program will allow for corrections if the user answered

Y or y, but will continue if the user answered anything else or nothing before depressing the RETURN.

Questions requiring number answers, for example the possible responses listed in a menu, also have default actions which occur if the user answers 0 (zero) or nothing before the RETURN. The response listed in a menu for zero is the default action. Often answering a number greater than any listed in a menu will cause the default action to occur.

Other questions requiring number answers may be replied to with nothing entered before the RETURN. A value of zero will be assigned to the quantity requested. This response is useful when the user is asked for the column or line number of an incorrect number when no corrections are desired. The zero column or zero line will be revised and not any in use. Whenever the default value for a null response is not to assign zero, the default value is mentioned in the question.

The entry of an entirely new configuration is the default response to a prompt in each program. The user may therefore arrive in this subroutine unintentionally. Not to panic, the beginning question allows the user to backtrack by replying N. It is advisable that a new user try all the default reactions of each program to discover the consequences and to gain familiarity and ease with the program.

A1.3 Correcting Data

The primary purpose of the personal computer programs are to review the data from the last time, updating and correcting where required. There are a number of tables to display the data found by the user selecting the appropriate item on a menu. Most tables are corrected item by item, whereas some, such as the crop data, may be corrected row by row, a crop at a time, or some by column.

A special table, used more or less the same way in both programs, is the review table of the nodes as listed in the current configuration. Please refer to Fig. A1.1 for an example. This table is displayed one node at a time, with the possibility for correction of only the last displayed node. After six nodes are displayed on the screen, the user may request the program to display the nodes from the beginning again.

To correct any of the information shown for that node, the user answers the number at the top of the column with the wrong value. Column one lists the node's name; answering 1

RANDENIYAGAL	RANT	----	0.000	----	0	----	0.	990.	303. ?
RANTEMBE	MINI	----	0.000	----	0	----	0.	17.	0. ?
MINIPE	MANN	ULHI	64.000	----	8	E	18.	0.	0. ?

Do you wish to see the nodes again? Answer Y to review the nodes already displayed. N or blank for default action

If the current line in the table has any incorrect entries, enter the INTEGER above one column which is not correct. Enter 0 or nothing before the RETURN to continue to the next node.
Enter 99 if you wish to store what you have updated so far.

1. THE NODE	2. DOWN STREAM NODE	3. TO:	4. DIVERSION CAPACITY m3/s	5. FROM	6. IRRIGATION DISTRICT No.	7. IRRIGATION DISTRICT NAME	8. CAP. m3/s	9. RESERVOIR MAX MCM	10. RESERVOIR MIN MCM
MANNAMPITIYA	SEA	----	0.000	----	1	A	10.	0.	0. ? 0 or blank
SEA	----	----	0.000	----	0	----	0.	99999.	0. ? 0 or blank
BOWATENNE	ELAH	----	0.000	POLG	10	HIM	28.	60.	15. ?

Figure A1.1: Sample REVIEW TABLE

allows for the correction of spelling errors or the removal of the node from the configuration. Column two lists the downstream node, either one on the same stream or one on another stream. Four dashes imply that the node has no spills. When the user wishes to delete or insert nodes on the same stream, the user replies 2. One may also revise the downstream node.

Columns three, four and five list the inter-basin diversions, the node to which the flow goes, the capacity in m^3/s, and for the nodes which receive inter-basin transfers, the node whence the water came. Columns six, seven and eight list the details of irrigation diversions. Columns nine and ten list reservoir capacities in PROJPLAN and columns eight, nine and ten list the code names for the reservoir, power plant and inflow files in PROJOP.

Two or more corrections may be made on one line. Changing data in one column sometimes results in changes to the others. Before attempting to correct these apparent mistakes, the user should review the nodes from the beginning to verify there really is an error. Answering 99 to the column error prompt allows the user to save the configuration to the diskette. Answering any other number or nothing results in the default, another line in the table is displayed.

To enter new nodes which are not to be inserted between existing nodes, the user would answer Y to the question at the bottom of the review table, asking if more nodes need to be entered. To check or revise what one has entered, one uses the correcting procedure of the REVIEW TABLE. Once the configuration of canals, reservoirs, power plants and irrigation districts have been entered and saved in a data file on a diskette, few future changes are expected.

A1.4 Specifying the Diversion of Turbined Flows

The procedure for entering the reservoirs and canals was not made specific to the Mahaweli Project although all features of the Sri Lankan river system were incorporated. One of those features, is that the water which has generated electricity at Polgolla does not flow downstream, but is diverted to the Sudu River. Another power plant in the K-M complex also diverts turbine flows, and possibly other power plants to be built will as well.

PROJOP considers this possibility after the entry of the current and target reservoir levels and the prices and options for the generation of electrical power. The power plants which also occur at inter-basin diversions are listed before the data file is written. The user is asked to count how many of the power plants divert the turbine flow. A vector will then be created to list the power plants with those diverting their turbine flow first in the list. PROJOP uses this list to write the SAS.OR data file so that the spills from these nodes go downstream and the turbine flows go to the diversion node.

When the list of power plants appears on the screen, the user first finds the power plant or plants which divert turbine flow and asks PROJOP to place these first in the list, by replying with the numbers to the left of their node names. The adjusted list of power plants will again appear, if all the diverting power plants are at the beginning, the user may then reply 0 and PROJOP will write the data file.

Before running PROJPLAN, the user should prepare a Crops file on the diskette for the B: drive. Inflows and Crop Water Demands files may also be prepared. The following section describes in detail the three files which are easier to prepare with a text editor program and not with PROJPLAN. A configuration file may be prepared using PROJPLAN and saved for future use, a previously prepared configuration file may be updated. The Variations file saves the current crop data to be used one monsoon season later, it is not an obligatory file.

After the section on PROJPLAN files, the PROJOP files which are most easily prepared by a text editor are described in detail. The Power Plant, Reservoir and Irrigation Loss files are similar. The Inflows and Irrigation Demands files are the last files described.

A2.1 THE INPUT FILES OF PROJPLAN

There are five input files of PROJPLAN; two are best prepared by the program. Although Crops is a lengthy file when there are many crops and many districts, the user is advised to prepare it without the inserting routines of PROJPLAN. The following section details the information needed in the file and in which column and in which line it should be placed.

The optional file for storing the net inflows is much shorter, a second section describes the file. The final section explains the file listing the crop water requirements by sub-season in detail.

A2.2.1 The Crops File

The crop information required by PROJPLAN is stored in a file called CROPS. The first line of the file has a title card, 80 characters of the user's choice. The first four characters may be used to identify the beginning monsoon season. When written in uppercase characters, the program will modify the title after it has advanced the values by one season.

The second line starts with two I5 fields for the number of districts and the number of crop varieties. The three F12.0 fields which end the line contain the penalty cost or the values for the minimization of land differences. The first number is the penalty cost or the maximum limit of differences in idle land between the first two seasons, the second between the last two seasons, and the third between the first and last season.

The next block of lines are organized by crop. The first line of each block has an A8 field for the crop name starting in column 2. The growing season code (1 for more than one season) is an integer value in column 12. Three F10.0 fields store any minimum production targets required from all the districts, one for each season.

The following data line is repeated for each season three times before being repeated for the next district. The order of the districts follows the numbering given to the configuration and is used consistently throughout this file. F8.0 fields are used to store the price, the yield, the cost of labour, fertilizer costs and seasonal crop irrigation water requirements per hectare of crop planted in that district and in that season. The seasonal crop water requirement is entered in 1000 m^3. Also in the line, in F8.0 fields, are the limits of suitable land for the crop and any minimum production level expressed as a minimum number of hectares for the district.

The values in these repeated lines may vary according to the season and district, however the format is identical. Blocks of these lines follow for each crop variety to be planted or to be possibly planted.

After the crop specific data has been entered, the district delivery costs are listed, again in F8.0 fields. The water delivery costs of all the districts in the first season appear before the others. Continuing on the same line (unless there are exactly 10 districts, in which case, on the next line), the delivery costs in the second season are listed, then the third.

An identical format is used for the available land in each district in each season: the values for all the districts for the first season, followed by those of the second and third with no blanks until the last line. The last set of lines store the combinations of crops which are expected to meet a target production level.

Even when no crops are combined, blank lines must appear, two more than three times the number of crop varieties. The user may supply blank lines and follow the interactive routines in PROJPLAN to store the combinations correctly for subsequent runs, or may successfully complete the lines following these requirements:

Three I3 fields in one line list the number of groups of crops which may be combined to meet a production target, one for each season. The next line has F8.0 fields for the target values, in tonnes or whatever unit is consistent with the yield given in the crop blocks. The target values are listed by season, ten per line with no blanks until the end of the third season. Finally, there are lines for every crop variety, comprised of 10 I3 fields.

They form three tables, one for each season, one after the other. The tables have zero values except for crops which appear in a combination to meet a production target. The integer value is the group number for that season, and will show up in the first non-zero column from the left. When a crop variety is included in more than one group. more than one column will be non-zero, however the integer values will always be different, and increasing from left to right.

The following example file shows the data arranged for two districts and three crops. Column counting lines have been inserted, they should not appear on files for submission to PROJPLAN. The underlines show the blank spaces and would not appear in an actual data file. The second season has nominally less land suitable than the other Maha seasons. Two crops are combined in one group in the first season, two others in the second, and all three in the third. As well, the third season has a second group of two crops which must meet a combined production target. The example file is set up to quickly illustrate the information which belongs in a given field; it may not provide a feasible linear programming problem.

```
1234567890_ _ +_ _ 2_ _ +_ _ 3__ _+_ _ 4_ _ +_ _ 5_ _ +
MAHA  3 CROP  example file 11/87
_ _ 2_ _ 3_ _ _5000.00_ _ _5000.00_ _ _ 500.00
_RICE_ _ _ 0_100000.00_ 60000.00_ 100000.00
_ 285.00_ _ 2.27_ _ 0.67_ _ 1.00_ _13.0045000.00_ 335.00
_ 285.00_ _ 2.07_ _ 0.67_ _ 1.00_ _13.00_ 200.00_ _ 0.00
_ 285.00_ _ 2.27_ _ 0.67_ _ 1.00_ _13.0045200.00_ 135.00
_ 285.00_ _ 2.27_ _ 0.67_ _ 1.00_ _13.00_5000.00_ 335.00
_ 285.00_ _ 2.07_ _ 0.67_ _ 1.00_ _13.00_5000.00_ 235.00
1234567890_ _ +_ _ 2_ _ +_ _ 3_ _ +_ _ 4_ _ +_ _ 5_ _ +
_MAIZE_ _ _0_ _ _ 0.00_ _ _ 0.00_ _ _ 0.00
_ 421.00_ _ 0.87_ _ 0.36_ _ 0.85_ _ 6.6740000.00_ 350.00
```

```
_ 421.00_ _ 0.87_ _ 0.36_ _ 0.85_ _ 6.6738000.00_ 350.00
_ 421.00_ _ 0.87_ _ 0.36_ _ 0.85_ _ 6.6740000.00_ 350.00
_ 421.00_ _ 0.87_ _ 0.36_ _ 0.85_ _ 6.67_4000.00_ 350.00
_ 421.00_ _ 0.87_ _ 0.36_ _ 0.85_ _ 6.67_4000.00_ 350.00
_ 421.00_ _ 0.87_ _ 0.36_ _ 0.85_ _ 6.67_4000.00_ 350.00
_MILLET_ _ 0
_ 221.00_ _ 0.61_ _ 0.67_ _ 1.00_ _ 2.3334000.00_ _ 0.00
1234567890_ _ +_ _ 2_ _ +_ _ 3_ _ +_ _ 4_ _ +_ _ 5_ _ +
_ 221.00_ _ 0.61_ _ 0.67_ _ 1.00_ _ 2.3334000.00_ _ 0.00
_ 221.00_ _ 0.61_ _ 0.67_ _ 1.00_ _ 2.3334000.00_ _ 0.00
_ 221.00_ _ 0.61_ _ 0.67_ _ 1.00_ _ 2.33_5000.00_ _ 0.00
_ 221.00_ _ 0.61_ _ 0.67_ _ 1.00_ _ 2.33_5000.00_ _ 0.00
_ 221.00_ _ 0.61_ _ 0.67_ _ 1.00_ _ 2.33_5000.00_ _ 0.00
_ _ 1.50_ _ 1.00_ _ 1.50_ _ 1.00_ _ 1.50_ _ 1.00
45200.00_5400.0045000.00_5300.0045200.00_5400.00
_ 1_ 1_ 2
_ 500.00_ 200.00_ 450.00_ 200.00
1234567890_ _ +_ _ 2_ _ +_ _ 3_ _ +_ _ 4_ _ +_ _ 5_ _ +
_ 1_ 0_ 0_ 0_ 0_ 0_ 0_ 0_ 0_ 0
_ 1_ 0_ 0_ 0_ 0_ 0_ 0_ 0_ 0_ 0
_ 0_ 0_ 0_ 0_ 0_ 0_ 0_ 0_ 0_ 0
_ 0_ 0_ 0_ 0_ 0_ 0_ 0_ 0_ 0_ 0
_ 1_ 0_ 0_ 0_ 0_ 0_ 0_ 0_ 0_ 0
_ 1_ 0_ 0_ 0_ 0_ 0_ 0_ 0_ 0_ 0
_ 1_ 2_ 0_ 0_ 0_ 0_ 0_ 0_ 0_ 0
_ 1_ 0_ 0_ 0_ 0_ 0_ 0_ 0_ 0_ 0
_ 1_ 2_ 0_ 0_ 0_ 0_ 0_ 0_ 0_ 0
```

A2.2.2 The Inflow File

The net inflows per sub-season for each node are entered in a user-named file. The first line has three I2 fields with the number of sub-seasons which are in each season. The next four columns are character fields where the user may store the name of the first season. The remaining columns are character fields for an identifying title.

The next set of lines have an A20 field for the node name in the first twenty columns. These are for identification purposes only. F10.0 fields follow, as many as required for the number of sub-seasons in each of the three seasons, spilling into the immediately proceeding lines, if necessary.

These lines are repeated for every node in the order the nodes are given in the configuration file. The net inflows may equal zero, or be negative, but enough lines should appear for every node.

The following example is for a three node configuration, with one sub-season in the Maha season and two in the Yala; the solution will be starting in the Maha season. The last field is shown one column short to fit the page. The files on the supplied diskettes show eight sub-seasons, the fields for the final two sub-seasons starting in the first column of the next line. The column counting lines do not occur in the file. The underlines show the blank spaces and would not appear in an actual data file.

```
1234567890_ _ +_ _ 2_ _ +_ _ 3_ _ +_ _ 4_ _ +_ _ 5_ _ +
_2_1_1MAHA Sample Net Inflow File 11/87
_sample_FIRST_NODE_ _ _ 89.550_ _ 95.220_ _712.485_ _ _84.55

_sample SEA_ _ _ _ _ _ _ _ 0.000_ _ _0.000_ _ _0.000_ _ _ 0.00
_sample_PARAKRAMA_SA_ _ 19.300_ _ -0.400_ _-20.050_ _ -10.35
1234567890_ _ +_ _ 2_ _ +_ _ 3_ _ +_ _ 4_ _ +_ _ 5_ _ +
```

A2.2.3 The Crop Water Demand File

The sub-seasonal crop water demands are entered in a user-named file by crop. The first line is similar to that of the net inflow file: three I2 fields for the number of sub-seasons in each season, an A4 character field for the name of the starting monsoon season in upper case letters, and the remainder of the line character fields for a user supplied title.

The block of lines which follow give the expected water demands per sub-season from every district for the first crop. The first line of the block is the name of the crop, two spaces followed by an A8 field. The next lines have one blank space, an A4 character field for a district identifier, then fifteen F5.0 real number fields for the sub-seasonal crop water consumption in thousands of cubic metre per hectare, listed in the order they occur. A line for every district is included in the block in the order the districts are listed in the Crops file. The blocks for each crop are in the order that the crops appear in the Crops file.

The sample file shown is for a three crop, two district problem with one sub-season in the Maha season and two in the Yala. The run starts in a Maha season. Every hectare of rice in district A is expected to require 4550 m^3 in the second sub-season of the Yala season. The column counting line does not appear in the file. The underlines show the blank spaces and would not appear in an actual data file.

```
1234567890_ _ +_ _ 2_ _ +_ _ 3_ _ +_ _ 4_ _ +_ _ 5_ _ +
_1_2_1MAHA  Sample Crop Water Requirement File 11/87
_ _ _ RICE
_ A_ _ 13.0_8.45_4.55_13.0
_ B_ _ 13.0_7.79_5.21_13.0
_ C_ _ 13.0_6.82_6.18_13.0
_ _ _MAIZE
_ A_6.67_3.33_3.34_6.67
_ B_6.67_3.33_3.34_6.67
_ C_6.67_3.33_3.34_6.67
1234567890_ _ +_ _ 2_ _ +_ _ 3_ _ +_ _ 4_ _ +_ _ 5_ _ +
_ _ MILLET
_ A_2.33_1.33_1.00_2.33
_ B_2.33_1.57_0.76_2.33
```

```
_ C_2.33_1.46_0.87_2.33
1234567890_ _ +_ _ 2_ _ +_ _ 3_ _ +_ _ 4_ _ +_ _ 5_ _ +
```

A2.3 INPUT FIELDS OF PROJOP FILES

The input files for PROJOP are smaller and simpler, they are to be used at least four times a month. Many will not require frequent revisions. The user is advised to enter the data for the configuration and canal characteristics files by the interactive subroutines of PROJOP. The file storing the time-dependent variations is also most easily revised with PROJOP. The five remaining files are more easily prepared with a text editor, or in the case of the expected crop water requirements and the expected streamflow files, by another computer program.

The format of the power coefficient file is explained first, the reservoir data file second. The quarter monthly crop water requirements may be included at the end of the economic loss file or be contained in a file all of their own. The third section explains the single format that is used for both cases. The fourth section describes the inflow file and the fifth the optional irrigation return flow file.

A2.3.1 The Power Plant File

For each power plant, this file stores the minimum and maximum turbine flow, and the three constants used to linearize the energy function when the head is not constant.

The first line starts with an I4 field showing the number of power plants in the configuration. This amount must agree with the number in the configuration file. The remaining 76 columns are character values; the user may supply an identifying title.

The rest of the file is in sets of four lines, one for each power plant. The first of the set has a blank space, then an A4 field for the character code assigned to the power plant in the configuration file. The codes for each power plant must agree with those in the configuration file and they must appear in the same order. This ensures that the coefficients are being applied to the proper node.

On the same line as the character code for the power plant, there are two F10.0 real number fields for the minimum and maximum releases allowed through the turbines, expressed in m^3/s. Any releases above the maximum from the node are spills; they do not generate electricity.

The next three lines store the values for the linearization of the energy function in F9.0 fields preceded by one blank. The first line contains the reference storage level in Mm^3. The second has the correcting factor, the c_{in} which will be used for most of the time steps in the run. The variation file will store the correcting factors by time step, allowing for adjustments. The final line of the set has the power coefficients, h_{in}.

When reservoir storage has little effect on the turbine production and the energy is a linear function of the release, the correcting factor is entered as -1.0, the reference storage is ignored, and the energy is estimated with only the first term of equation 5.5 using the power coefficient in the final line. The four lines are repeated for every power plant in the configuration.

The following example file is for two power plants, one without any reservoir storage. The column counting lines do not occur in the file. The underlines show the blank spaces and would not appear in an actual data file.

```
1234567890_ _ +_ _ 2_ _ +_ _ 3_ _ +_ _ 4_ _ +_ _ 5_ _ +
_ _2  Example power plant file, 11/87
_ RES_ _ _4.50_ _ _154.87
_ _ _56.75
_0.0025632
_0.6799000
_NONE_ _ _7.8_ _ _ 345.5_
_-1.000000
_ 0.763200
1234567890_ _ +_ _ 2_ _ +_ _ 3_ _ +_ _ 4_ _ +_ _ 5_ _ +
```

A2.3.2 The Reservoir File

The reservoir file contains information on the maximum and minimum permitted storage levels, the expected net evaporation and seepage losses and the values which estimate the surface area from the storage volume. The first line starts with an I4 field for the number of storage reservoirs and tanks in the configuration. This amount must agree with the number in the configuration file. The remaining 76 columns may be used to write a descriptive title.

There are four lines for each storage node. The first line has a blank space, then an A4 field for the character code of the reservoir. The code must be the same as used in the configuration file and follow the order established there. Continuing that line are two F10.0 real number fields for the minimum and maximum storage levels of the reservoir in Mm^3.

The variation subroutine of PROJOP will allow the user to vary these values for a given time step; any adjusted values may be stored on a variation file. The final item on the line, in an I3 field, is the number of linearization intervals for the surface area - storage level curve. This number will usually be 1; because when mixed integers are used to linearize the curve, the more accurate estimation requires so much more computer time for so little difference, that in the case of most surface area estimates, a single linearization interval suffices.

The second line is a listing of the monthly loss coefficients for the reservoir in mm. They include both expected seepage and evaporation losses and are applied to the surface area of the reservoir or tank. There is a blank space in the first column of the line, then twelve F6.0 fields.

The final two lines list the linearization constants. There is one blank, then nine F6.0 fields in each line. the third line of the set for each reservoir lists the storage levels, in Mm^3, which fall at the end of each linearization interval. The last line lists the surface area, in km^2, at the end of each interval.

When there is only one interval, any two storage and surface areas which have the ratio desired for approximating the whole range of storage and surface area values may be listed in the first F6.0 field. The underlines show the blank spaces and would not appear in an actual data file.

```
1234567890_ _ +_ _ 2_ _ +_ _ 3_ _ +_ _ 4_ _ +_ _ 5_ _ +
_ _2  Example reservoir file, 11/87
_FAT_ _ _ 15.0_ _ _1300.0_ 3
_ _17.0_ 35.0_ 71.0_110.0_154.0_145.0_ 78.0_ 61.0_ 14.0_ 2
_ _50.0_349.71300.0
_ _ 2.0_ 20.0_178.5
_ BOX_ _ _ 15.0_ _ _890.0_ 1
_ _14.0_ 38.0_ 41.0_110.0_184.0_148.0_ 48.0_ 61.0_ 14.0_ 2
_ 300.0
_ _ 1.0
1234567890_ _ +_ _ 2_ _ +_ _ 3_ _ +_ _ 4_ _ +_ _ 5_ _ +
```

The example file is for two reservoirs, one with three linearization intervals and one with a constant ratio. The column counting lines are not included in an actual file. Note that two, and most of a third, seepage and evaporation monthly loss fields are not shown due to the width of the page.

A2.3 The Irrigation File

The irrigation economic loss data and the estimated crop water requirements may follow each other in the same file or be split into two files. The format is exactly the same, instead of following the last line of economic loss data, the first line of the crop water demand file appears at the beginning of its own file.

The economic loss data quantifies the penalty assigned to any irrigation deficit occurring in that time step, at a constant rate for any magnitude of loss, or, using the mixed linearization procedures, varying the loss with the magnitude of the deficit. The penalty may be more severe during critical periods of the growing season than during other times. Time is measured in units of quarter months of the entire growing season, from start to finish, for both the economic loss data and the estimated irrigation demands.

The first line of the economic loss file has an I4 field listing the number of irrigation districts in the configuration. A descriptive title may be written in the 76 character columns which follow. The next set of three lines are repeated for each irrigation district in the order the irrigation districts appear in the configuration of nodes, i.e. by node number of the delivery node.

The character code of the irrigation district is entered in an A4 field after an initial blank. An I4 field follows with the integer number of penalty curves which occur that season. Twelve I4 fields may be used to enter the last quarter month that a penalty rate applies and how many linearization intervals are used. The irrigation loss penalty rates must be entered in chronological order. The rate or curve which applies until the fifth quarter month would come before the rate or curve which applies from the fifth to fifteenth quarter month.

The final two lines of each three line set express the economic loss for each rate or curve in as many F7.0 fields as necessary. The first character of each line is a blank. When the sum of the linearization intervals for the number of curves to apply in a growing season is more than 11, each line will continue into the immediately following line.

The second line is the irrigation deficit upper bound of a linearization interval for each curve which applies during the season, while the third line is the penalty cost for a deficit of that magnitude, for each interval of each curve. Normally, the mixed linearization intervals would not be used as too much computer time is required for a solution.

Therefore, a constant rate, independent of the size of the irrigation deficit, but which varies according to the critical periods of the crops in each district, is entered in these two lines for each period. Any deficit, in Mm^3, and any penalty cost which has the required ratio is entered in the F7.0 fields.

The estimated crop water requirements may follow the above information, or may be stored in an independent file, when their revisions occur more frequently than the updating of the economic losses. The first line of the expected crop irrigation demands gives the number of irrigation districts in an I4 field. A title may be written in the 76 column character field which remains. The next set of four lines are used for every district in the configuration in the order in which the districts appear there.

PROJOP initially reads only the first five columns in an A5 field, matching the characters with those from the configuration file, to check that the proper district is being entered. The three lines which follow list the quarter monthly anticipated district irrigation demands, in Mm^3, in F7.0 fields, 11 per line. Twenty six values must be entered even in the Yala season; when less quarter months remain in the growing season, blanks or values may be used to fill the spaces of the quarter months already past.

The following example file is for a two district configuration, with ten quarter months left of a Yala reason. The most critical period is past for the first district, where mixed integer linearization was used to describe the losses as a function of the magnitude of the deficit during the first ten quarter months. Note that the final rate or curve must apply to the end of the growing season, quarter month 22 in the Yala season. In district A, the penalty which would occur from any irrigation deficit would be 41.667 (250/6) for any time step from quarter month 10 to quarter month 15.

Three fields have been dropped from the irrigation demand lines to fit the page. The actual file would not have the column counting lines and would have 80 columns. The underlines show the blank spaces and would not appear in an actual data file.

```
1234567890_ _ +_ _ 2_ _ +_ _ 3_ _ +_ _ 4_ _ +_ _ 5_ _ +
_ _2  Example Irrigation File Yala 11/87 Economic Losses
_ _ A_ _3_ 10_ _3_ 15_ _1_ 22_ _1
_ _ 1.00_ _2.00_ 20.00_ _6.00_ _9.00
_ 1500.0_2000.0_9000.0_250.00_570.00
_ _ B_ _3_ _7_ _1_ 15_ _1_ 22_ _1
_ _ 6.00_ _9.00_ 10.00
_ 1500.0_2000.0_1900.0
_ _2 Example Irrigation File Yala 11/87 Crop Water Req
_ _ A   Any value used for unnecessary quarter months
1234567890_ _ +_ _ 2_ _ +_ _ 3_ _ +_ _ 4_ _ +_ _ 5_ _ +
_ _0.00_ _0.00_ _0.07_ _9.99_ _9.99_ _9.99_ _9.99_ _9.99
_0.0000_0.4382_0.4382_0.4382_0.4382_ 0.313_ 0.313_ 0.313
_ 0.0_ _ 0.0_ _ 0.0_ _ 0.0_
_ _ B   Blank lines used for unnecessary quarter months
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ 3.83_ _3.83_ _3.83_ _3.83_ 2.736_ 2.736_ 2.736

1234567890_ _ +_ _ 2_ _ +_ _ 3_ _ +_ _ 4_ _ +_ _ 5_ _ +
```

A2.3.3 The Inflow File

The most frequently revised file stores the predicted streamflows for the inflow nodes of the configuration. The values are in Mm^3 a time step, for whatever duration of time step. Tables 5.1 and 5.2 may be used to identify how many time steps and their duration are required for any data. The file starts with a title line. This first line has an I4 field listing the number of nodes for which inflows are supplied; this number must be that given in the configuration file. The remaining 76 columns are character fields for an identifying title.

The next two or three lines are repeated for every inflow node in the order the nodes appear in the configuration. The first line of the set has an A5 field for the character code of the configuration file. Two I4 fields follow listing the first the number of quarter months and then the total number of time steps of the run. PROJOP may verify that the predicted inflows are indeed for the current run. The next line lists the expected inflows for each time step in chronological order, using twelve F6.0 fields a line as many lines as necessary.

The sample inflow file is for three nodes, with 10 quarter months left of a Yala season. The column counting line does not appear in the actual file. The underlines show the blank spaces and would not appear in an actual data file.

```
1234567890_ _ +_ _ 2_ _ +_ _ 3_ _ +_ _ 4_ _ +_ _ 5_ _ +
_ _3  Example Inflow File, Yala - 10 quarter months to go
_NOD1_ _6_ _7   Anything may be written here
_12.57_12.57_12.57_ 13.0_ 13.0_ 13.0_ 73.0_ 92.5_ 87.9_ 77.2
_NOD2_ _6_ _7   Anything may be written here
_42.97_52.97_48.97_42.67_ 32.8_ 43.6_133.8_142.5_135.7_132.4
_NOD3_ _6_ _7   Anything may be written here
_21.33_27.67_22.47_16.87_17.45_21.45_90.45_20.45_ 38.8_ 13.0
1234567890_ _ +_ _ 2_ _ +_ _ 3_ _ +_ _ 4_ _ +_ _ 5_ _ +
```

A2.3.4 The Optional Irrigation Return Flow File

When there are significant irrigation return flows which are quantifiable and reliable, the information required to include the flows in the continuity equations may be entered and stored on a diskette file. The node numbers of the locations which receive the return flow, whence the irrigation deliveries are made, the time in quarter months the flows take to return, and the estimated percentages returning are stored in a single optional file.

The first line of the file uses an I5 field for the number of locations where irrigation return flows occur. The remaining 75 columns are character fields for an identifying title. The next line has I4 fields in sets of three for each occurrence of return flow. The integer in the first field of the set is the number of the node whose partial irrigation deliveries return to the node numbered in the second field. The third integer in the set is the number of quarter months required for the percentage of irrigation delivery to appear in the continuity equations of the second node. The sets are repeated sequentially for every location.

When seven or more locations are listed, the I4 fields will continue another line. The final lines list the percentage of irrigation which returns for each quarter month of the season, from the first to the last quarter month in F10 fields. The values for one location follow in the same line as the preceding if there are empty fields. The percentages are given in the order the locations are given in the integer lines.

The following example file has two instances of return flow, from the node numbered 3 to the node numbered 7, and from the node numbered 8 to the node numbered 7. The percentage which returns did not vary in either case, though the lag times differed. A tenth of the irrigation turn out from node 3 arrives at the node 7 in 2 quarter months, the return flows from 8 arrive in the same quarter month that they are delivered. Only 6 real number fields of the eight per line are shown as there are only 59 columns printed on the page. The column counting lines are not included in the file entered to PROJOP. The underlines show the blank spaces and would not appear in an actual data file.

```
1234567890_ _ +_ _ 2_ _ +_ _ 3_ _ +_ _ 4_ _ +_ _ 5_ _ +
_ _ 2          Sample Return Flow File MAHA season 11/87
_ _3_ _7_ _2_ _8_ _7_ _0
_ _ _0.100_ _ _0.100_ _ _0.100_ _ _0.100_ _ _0.100_ _ _0.10
_ _ _0.100_ _ _0.100_ _ _0.100_ _ _0.100_ _ _0.100_ _ _0.10
_ _ _0.100_ _ _0.100_ _ _0.100_ _ _0.100_ _ _0.100_ _ _0.10
_ _ _0.100_ _ _0.100_ _ _0.126_ _ _0.126_ _ _0.126_ _ _0.12
_ _ _0.126_ _ _0.126_ _ _0.126_ _ _0.126_ _ _0.126_ _ _0.12
_ _ _0.126_ _ _0.126_ _ _0.126_ _ _0.126_ _ _0.126_ _ _0.12
_ _ _0.126_ _ _0.126_ _ _0.126_ _ _0.126
1234567890_ _ +_ _ 2_ _ +_ _ 3_ _ +_ _ 4_ _ +_ _ 5_ _ +
```

A3.1 The Meaning of Run Time Error Messages

During the execution of the data preparation programs an abrupt termination may occur giving an error message number. When the current drive has the FORTRAN compiler error message file, an explanation of the error also follows. A few error message numbers which may possibly occur and the likely corrective action are listed below for users who may not have the error message file.

-1 Add blank lines to the length of the file.

1236 Reduce the size of an array, perhaps the number of crops, or of districts, or perhaps of nodes.

2015 Check that the filename is spelled correctly, that the correct diskette is in the specified drive, that the filename is listed in the diskette directory.

2514 Verify that the integers in the file being read when the error occurred are correctly placed in the fields.

2519 Verify that the real numbers in the file being read when the error occurred are correctly placed in the fields.

3000 Increase the working space on DOS memory, or reduce the size of the problem.

4001 Connect a math coprocessor to the personal computer.

4002 Upgrade your DOS version to 2.1 or later.

Any other error message requires correction to the FORTRAN code and recompilation.

A4.1 Running the SAS.OR Sparsedata LP Package

The sparsedata LP package available with the SAS.OR computer code for mainframes was used to solve the datasets generated by the interactive micro-computer programs for the PROJPLAN and PROJOP formulations. A certain format for the entry of the data must be followed as defined, however, there is considerable flexibility within those limits.

Data may be entered in any order, separated into incomplete datasets which may then be combined into various full sets, and the combined data set printed if desired. All of the SAS facilities for datasets are available to the user.

While the interactive programs developed for the personal computer may be used to generate a completely new data file, for quick changes of certain constants, some familiarity with the SAS data files would be useful. When the time has come to add new constraints or revise existing ones, the programmer will need to know what are the items in the SAS.OR data files.

The following section describes the data files in detail. If multiple runs are to be done with mostly the same constants, but with inflows or prices varied, the data set may be divided into two, one part that doesn't change and the other with one set of variable data. Additional datasets need only contain the variable data. At run time the user may ask SAS to combine the desired data and to perform the LP solutions to the combined datasets sequentially. This procedure is described in the last section.

A4.2 Description of Data Files

The sample datasets shown in Figures 6.1 and 6.2 are from actual data files generated by PROJPLAN and PROJOP. The length of the data files has been shortened, as it was unnecessary to show the same format repeated many times for each time step, reservoir or district.

Before the dataset may be solved by SAS.OR it needs a header in job control language which will execute the SAS job on a mainframe. Figures 6.1 and 6.2 show these five lines as they were used at the Ecole Polytechnique. It is up to the user to determine how to run the SAS job on the computer available and to type in the necessary job control lines.

Every variable has been given a name of eight characters or less, identifying to which time step, district or node the variable belongs. The constraints, objective function, upper and lower bound constraints have also been identified with names of eight characters or less.

Longer names are permitted, however SAS.OR does not distinguish between two names which differ in the ninth or more characters. ANYNAME123 would be considered the same as ANYNAME199. The names are not allowed to start with a number, but must start with a letter from A to Z.

The output file generated by SAS.OR lists the variables in alphabetical order with any numbers in the name appearing after Z. The constraints are also listed in the output file in the same alphabetical order. This results in a numbered name such as FS12S12 coming before FS9S12.

The SAS.OR data file begins with information identifying the data and telling how to interpret the datalines. The first line of a data file gives the dataset name, which may be eight or less significant characters long, followed by a semi-colon. The semi-colon is used to terminate a SAS command and must not be neglected when required nor inserted when not.

The second line defines how the data will be read: the identity, the order, and the number of data units which will appear on a line. The third line announces the actual data lines, it is the last to terminate with a semi-colon until the semi-colon at the end of all the data lines in the data set.

The data lines differ between PROJPLAN and PROJOP because only three pairs of data units were used per line in PROJPLAN and six per line are used in PROJOP. This was to reduce the number of lines in PROJOP, as more variables do appear in more than three constraints.

Lines which start with a period in column two give the name of the variable which appears in the subsequent constraints listed on that line, and the coefficient assigned to that variable in that constraint. The identification of constraints is done in the lines which start with keywords in column two. Note that the objective function, upper and lower bounds and the identification of a variable as an integer are treated as though they were constraints. Table A4.1 lists the acceptable keywords and their significance.

TABLE A4.1

Acceptable keywords for SAS.OR PROC LP

MAX	-	the constraint is the objective function to be maximized.
MIN	-	the constraint is the objective function to be minimized.
EQ	-	the sum of the variables times their coefficients must be equal to zero or the value of the _RHS_ variable for the constraint.
GE	-	the sum of the variables times their coefficients must be greater than or equal to zero or the value of the _RHS_ variable for the constraint.
LE	-	the sum of the variables times their coefficients must be less than or equal to zero or the value of the _RHS_ variable for the constraint.
GT	-	the sum of the variables times their coefficients must be greater than zero, or the value of the _RHS_ variable for the constraint.
LT	-	the sum of the variables times their coefficients must be less than zero or the value of the _RHS_ variable for the constraint.
UPPERBD	-	upper bound designation
LOWERBD	-	lower bound designation
INTEGER	-	integer designation

The known terms of a constraint, manipulated mathematically to appear on the right hand side of the inequality or equation, are listed in the lines which have a period in column two and the special keyword _RHS_ in the column which would otherwise be used for the variable name. The names of each non-zero constraint with the calculated value follows on these lines.

Each line follows the format specified in the INPUT line at the beginning of the dataset, using periods to fill the line with either eight or fourteen datapoints. A space must appear between names and between the names and the coefficients which may take as much space as required.

No specific order for the lines within a dataset is necessary. The file writing subroutines of the data file preparation programs naturally batch the lines for defining variables, constraints, and the non-zero _RHS_ constants together as they go through the lists for each node and time step. To modify a constraint or add new ones, the SAS.OR data lines may be added anywhere after the line "CARDS;" and before the next ";", anywhere in the dataset.

SAS.OR reads only seventy-two characters per card line. If the eight or fourteen datapoints it expects to read per line have not yet appeared, it will continue to the next card line, reading the data there as though they were appended to the card above. Thus two card lines may be used to enter one line of data, and the periods marking empty datapoints must not be forgotten.

At the end of all the data is a semi-colon signifying that all that preceded is a part of the dataset mentioned in the first line. Following is a command line asking for the linear programming procedure, PROC LP SPARSEDATA, terminating with a semi-colon.

For problems which will take more time or more iterations than the default values, there are optional parameters which may also be included to request for more time and iterations. The error messages of the SAS.OR package explain when and how to use them when they are required. Some other commands which may be used are explained in the section on combining datasets.

A4.3 The SAS.OR Solution Format

Figure 6.3 shows an example of the SAS.OR output for a PROJPLAN run. There is a small table listing the size of the problem, the number of constraints, the number of variables, another table lists the length of time it took to find a solution, how many iterations, etc. Figure 6.4 shows a typical solution, with columns identifying the variables by number, name, type, and what role they played in the linear programming solution.

Under the heading, "ACTIVITY", is listed the optimal value for that variable found in that solution. The final column lists the dual values of the variables, their reduced cost, a measure of how much their price in the objective function would have to change before the basis of the linear programming solution would exchange them.

An attempt has been made to have the variables of most interest printed out first, by judicious choice of their initial letter and the placement of the numbers designating the time step and the node within the name. The user may change the order by renaming the variables.

The final pages of the print out list the constraints and how well the solution was able to meet equations or exceed non-equalities. If a non-feasible solution occurs, when no set of values were able to satisfy all the constraints, the last pages will list the constraint or constraints it was unable to meet and the values assigned for the variables within that constraint. This allows the user to decide why the problem given had no solution.

A4.4 Combination of Data files

The combination of data files is most useful when the user wishes to test the response of the formulation to various streamflow forecasts, differing crop water requirements, differing power estimation constants, or any other set of constants. Instead of repeating the whole data file for each trial in the series, only the part of the data file which will change is repeated for each trial. The whole series of trials are submitted at one time in one run.

Combination entails isolating the major part of the data file which will remain unchanged during the series of trials; this subset is placed in its own dataset, with its own name, input, cards and semi-colon line at the end. When naming datasets, remember that only the first eight characters are significant, the characters after eight will not distinguish new datasets.

The lines of the data file which contain the information that will be changed a number of times come before or after the large invariate subset. These lines must also be written in their very own dataset, with their own unique name, input, and cards lines at the start and semi-colon line at the end.

The second smaller subset is repeated for as many times as there are variations in the series of trials, with each new subset given a new name and the new coefficients. All the subsets appear in the same data file to be submitted to SAS.OR.

```
/SYSIN     DD   *
 DATA BASEDATA;
 INPUT _TYPE_ $ _COL_ $
   _ROW1_ $ _COEF1_ _ROW2_ $ _COEF2_ _ROW3_ $ _COEF3_;
 CARDS;
 .  VAR1    CNT1  1   CNT2   -1      UPPER  180.000
 .  VAR2    CNT1 -1   UPPER 16.600  LOWER    0.000
 LOWERBD   .   LOWER      .     .     .     .      .
 UPPERBD   .   UPPER      .     .     .     .      .
 EQ     .    CNT1    .  CNT2     .     .     .
 MAX     .   OBJ     .     .     .     .     .
;
 DATA FLO1DATA;
 INPUT _TYPE_ $ _COL_ $
   _ROW1_ $ _COEF1_ _ROW2_ $ _COEF2_ _ROW3_ $ _COEF3_;
 CARDS;
 .     _RHS_   CNT1 109.700  CNT2  81.800     .        .
 .   VAR3      CNT2    -1     CNT1   4.55   LOWER     3.33
 ;
 DATA FLO2DATA;
 INPUT _TYPE_ $ _COL_ $
   _ROW1_ $ _COEF1_ _ROW2_ $ _COEF2_ _ROW3_ $ _COEF3_;
 CARDS;
 .     _RHS_   CNT1  99.300  CNT2  91.200     .        .
 .   VAR3      CNT2    -1     CNT1   4.55   LOWER     2.50
 ;
 DATA COMB1DAT;
  SET BASEDATA FLO1DATA;
 PROC LP SPARSEDATA MAXIT1=500 MAXIT2=500
 DATA COMB2DAT;
  SET BASEDATA FLO2DATA;
 PROC LP SPARSEDATA MAXIT1=500 MAXIT2=500
/*
```

Figure A4.1: Sample dataset combination

The example in Fig.A4.1 shows two small datasets to be combined into one before the solution by the linear programming procedure. The final three lines above "/*" would be repeated for each trial in the actual series. The first line of these final three, names the combined data set. This name may be the same for each trial. The second line lists the datasets to be combined. They may be in any order, and will be the names given to the major subset and to each trial dataset. The last line of each triplet is the same last line of the data file seen in Figures 6.1 or 6.2, asking for a linear programming solution to the above defined combined dataset. At the end of all the triplets for each trial in the series would come the "/*", end of file mark.

When numerous trials are to be conducted in one series, the user may place the triplet of the combining commands immediately after each trial dataset. The user may then use the same name for each trial dataset because the linear programming solution is performed before the creation of the next trial dataset. This would save runtime file space.

To verify the combination of a dataset, the user may request SAS.OR to print out the dataset. This command must occur before any re-definition of the dataset name, either before or after the linear programming solution. The PROC PRINT command, written in the same fashion as PROC LP, also specifies the dataset, using DATA=datasetname with a semi-colon at the end of the line. Other SAS commands may be used, the interested reader may refer to SAS manuals to learn the commands for sorting datasets, isolating lines with particular variables, and creating specific datasets as desired.

APPENDIX B

This appendix lists the FORTRAN data preparation programs, line by line. The appendix has two portions, one for each data preparation program. The SAS.OR data file written by each program is similar, however PLAN.DATA is written with three pairs of a constraint and its coefficient per line and RELEASE.DATA with six. This means that every line has three or six pairs, using dots to represent blank quantities. Different format lines are required depending on the number of actual values which will appear on a data file line and the number of dots.

The two data preparation programs use two different methods of writing the necessary format lines. PROJPLAN stores the format lines in a character array; the write subroutines select the correct line to use before writing the variable, constraints and coefficients onto the diskette. PROJOP builds the format line in a concatenation of character strings, using considerably less computer memory and FORTRAN code. Both methods are retained in case another compiler does not support character string concatenation in the same fashion.

APPENDIX B1

The data preparation program for PROJPLAN reads the Crops file, the configuration file, if one has been prepared, the inflow and crop water requirements files, and a variation file if the user so wishes. The information from these files are displayed so that the user may correct or update any value.

The user selects the information to review from an initial menu. Once the updating and saving of revised files has been done, PROJPLAN writes the SAS.OR datafile, PLAN.DATA. There is a subroutine for each type of line in the file. Table B1.1 shows the structure of PROJPLAN.

TABLE B1.1

The programming structure of PROJPLAN

PROJPLAN

calls from the menu:

```
 CONFIG   CORRIG     CHANGE     CHANJ3     COMCRP     VARDATA

                        which may call

          DELETE     CHANG2                GCHANJ
                       ADD          COSTS
                     LOOPMK       CROPTB
                                     which calls CHANJ1
```

once all the input data is updated, PROJPLAN calls:

CNT
DATALN
CHOICE
RHS

The following table describes in which FORTRAN file each program unit is stored, and its function.

TABLE B1.2

Location and function of FORTRAN program units

UNIT NAME	IN FILE	FUNCTION
PROJPLAN	Projplan.for	Main program, calls others, writes updated files
CONFIG	Hscheme.for	Enters new configuration nodes
CORRIG	Hscheme.for	Corrects configuration nodes
DELETE	Hscheme.for	Deletes nodes from configuration
ADD	Hscheme.for	Inserts nodes between existing nodes
LOOPMK	Hscheme.for	Prepares configuration vectors for changes
CHANGE	Hchange.for	Reviews the data on b:CROPS
COSTS	Hccost.for	Reviews and corrects idle land costs
CHANG2	Hchange.for	Corrects district data
CROPTB	Hchange.for	Displays crop data in district tables
CHANGJ1	Hchange.for	Corrects crop data in district tables
CHANGJ3	Projplan.for	Corrects minimum crop production targets
COMCRP	Hcomcrp.for	Groups crops under common production target
GCHANJ	Hcomcrp.for	Changes the crop groupings
VARDATA	Vardata.for	Adjusts data for sub-seasonal durations
CNT	Hcnt.for	Writes the continuity variables to the datafile, b:PLAN.DATA
DATALN	Hdatn.for	Writes the constraint names to b:PLAN.DATA
CHOICE	Hcoen.for	Writes the crop variables to b:PLAN.DATA
RHS	Hrhn.for	Writes the RHS constants of the constraints to b:PLAN.DATA

Before showing a complete FORTRAN listing, the common block arrays used in the data preparation program will be explained. Definitions of other variables, arrays, or subroutine arguments appear in the program listing. There are 12 common blocks used by one or another of the subroutines.

The SHR common block contains the crop data arrays for every season, every crop and every district. The array names are listed below with a short description.

P(3,50,20)	is the array of prices per hectare.
Y(3,50,20)	is the array of yields per hectare.
CL(3,50,20)	is the array of labour costs per hectare.
F(3,50,20)	is the array of fertilizer costs per hectare.
U(3,50,20)	is the array of expected seasonal water consumption, thousands of m^3 per hectare.
T(3,50,20)	is the array of maximum suitable land.
B(3,50,20)	is the array of minimum cultivated land.
Q(3,50,20)	is an array of the net return per hectare, calculated by the data preparation program.

The NEW1 common block contains seasonal arrays, some crop group arrays and the long growing season code vector:

NG(3)	is the number of groups combining each season to meet a production target.
WD(3)	is the penalty cost or limit on idle land each season.
JCROP(3,50,10)	is the array listing to which group, if any, a crop belongs.

PCROP(3,25) is the array of common production targets.

LNGTH(50) is the code for the length of the growing season, 0 means one season.

The PROD common block contains the arrays for minimum production targets by crop variety.

YMIN(3,50) is an array of minimum production targets.

NMINJ(50) is an array of crop numbers of the crops with production targets.

NMINI(50) is an array of season numbers associated with production targets.

The CROP common block is a character block.

NAME(50) is a vector of crop variety names.

The DELTN common block contains the array for changes in the number of crops.

IX(50) is a vector of crops which were deleted.

The SHC common block stores a few of the configuration vectors.

JIRR(21) is the vector of nodes which irrigate.

JRES(25) is the vector of nodes with reservoirs or tanks.

JNOSPL(20) is the vector of nodes which do not send water to the node numbered one higher.

JDIV(2,15) is the array of nodes which divert water. The first number of the pair diverts, the second receives.

The SHP common block has more configuration vectors.

JTRB(2,10) is the array of nodes which are tributaries. The first number of the pair releases its water to the second node.

SLO(25) is the vector of minimum storage levels in Mm^3.

SUP(25) is the vector of maximum storage levels in Mm^3.

J1(2,10) is an array of nodes involved with irrigation return flow. A percentage of the turnout from the first node is received by the second.

C1(10) is a vector of the percentage of irrigation turnouts which return.

The CAP common block lists the capacities of the various components of the configuration.

KIRR(21) is a vector of the districts irrigated by the irrigation nodes.

CCAPI(21) is a vector of the irrigation diversion capacities, m^3/s.

CCAPD(15) is a vector of the diversion capacities, m^3/s.

CRHS(3,35,5) is an array of net sub-seasonal inflows, Mm^3.

CLOSS(15) is a vector of channel losses in percent.

The NOM common block is a character block for the names in the configuration file.

NNAME(40) is a vector of node names.

IDNAME(21) is a vector of irrigation district names.

The FLO common block has the arrays necessary to define downstream flow restrictions.

NLIM(21) is the vector of nodes with downstream flow restrictions.

NSS(3) is a vector of the number of sub-seasons in each season.

FLOMIN(20) is a vector of the minimum downstream releases in m^3/s.

The common block, VAR, has the arrays of values which change by sub-season.

FLMIN(3,5,20) is an array of minimum flow restrictions, Mm3.

FLMAX(3,5,20) is an array of maximum flow restrictions, Mm3.

CHCPI(3,5,21) is an array of irrigation canal capacities, Mm3.

P1(3,5,10) is an array of percent return flow.

CHCPD(3,5,15) is an array of diversion canal capacities, Mm^3.

CHLSS(3,5,15) is an array of canal losses, percent.

SMAX(3,5,25) is an array of maximum storage levels, Mm^3.

SMIN(3,5,25) is an array of minimum storage levels, Mm^3.

The SUB common block stores the sub-seasonal crop water demands.

WCOEF(3,50,20,5) is an array of expected sub-seasonal crop water requirements, by crop and by district.

```
c************************************************************************
c
c This FORTRAN programme asks for the data necessary to decide how many
c hectares to plant in each crop in each district in each of three grow-
c ing seasons for the Mahaweli Irrigation Project. It is dimensioned
c for 50 different crops and 20 irrigation districts, though less may be
c analysed. The data are arranged for the file b:PLAN.DATA to be sub-
c mitted to a SAS.OR linear programming procedure for solution. The op-
c timization maximizes the returns after fertilizer, labour and water
c costs are deducted from the price per hectare expected for each
c crop and may minimize the seasonal variation in hectares planted. The
c optimization is subject to the continuity equations of the reservoirs,
c diversions and irrigation nodes in the Mahaweli Project, as calculated
c on a seasonal or sub-seasonal basis.
c The decision variables, the number of hectares planted in each crop j
c in each district k in each season i are constrained by the amount of
c land available in the district that season HH(i,k), and the amount of
c water forecast available for irrigation X(i,n) in each of the next
c three seasons for the continuity equation of the node n. The water
c surplus to the irrigation needs of one season, or which may be better
c used the next season, is stored in the reservoir or tank at the node
c n. This level of end of season storage is represented by the decision
c variable, FSnSi, associated with the node n of the continuity
c equations.
c
c Only so many hectares may be planted in one crop in a district in a
c season, these upperbounds are the T(i,j,k). The lower bounds,
c B(i,j,k), stipulate that at least so much of a crop in the
c season in that distict must be grown. Alternatively, constraints for
c each crop ensure that so much must be produced from all districts, or
c that certain crops are to be grouped together to supply a minimum
c amount.
c
c    Also the file b:CROPS keeps the crop data for next time, when a
c subsequent run requires only review and updating when necessary. The
c information required by the continuity equations may be stored on
c files named by the user, as may a file of predicted inflows.
c
c ***********************************************************************
c    W(i,k)   = delivery cost of water in season i to district k
c    HH(i,k)  = total available land in season i in district k
c    DURATN(i,is)  = number of days in sub-season is in season i
      DIMENSION W(3,20), HH(3,20), DURATN(3,5)
      COMMON /SHR/ P(3,50,20), Y(3,50,20), CL(3,50,20), F(3,50,20), U(3,
     150,20), T(3,50,20), B(3,50,20), Q(3,50,20)
      COMMON /NEW1/ NG(3), WD(3), JCROP(3,50,10), PCROP(3,25), LNGTH(50)
      COMMON /PROD/ YMIN(3,50), NMINJ(50), NMINI(50)
      COMMON /CROP/ NAME(50)
      COMMON /DELTN/ IX(50)
      COMMON/SHC/ JIRR(21), JRES(25), JNOSPL(20), JDIV(2,15)
      COMMON/SHP/ JTRB(2,10), SLO(25), SUP(25), J1(2,10), C1(10)
      COMMON/CAP/ KIRR(21),CCAPI(21), CCAPD(15),CRHS(3,35,5), CLOSS(15)
```

```
      COMMON/NOM/     NNAME(40),     IDNAME(21)
      COMMON/FLO/     NLIM(21),    NSS(3),    FLOMIN(20),     FLOMAX(20)
      COMMON/VAR/FLMIN(3,5,20),        FLMAX(3,5,20),     CHCPI(3,5,21),      P1(3,5,10)
     1        ,CHCPD(3,5,15),      CHLSS(3,5,15),      SMAX(3,5,25),      SMIN(3,5,25)
      COMMON/SUB/     WCOEF(3,50,20,5)
C
      CHARACTER    * 12   NNAME,    FNAME,    NAME1,    KNAME(40)
      CHARACTER    * 8   NAME
      CHARACTER    * 4   NAME2,    NAME3,    NAME4,    NAME5,    IDNAME,    TITLE1(18),
     1                        TITLE2(18),    TITLE3(18),     DUMMYT(18)
      CHARACTER    * 2   ANS
C   START   OF   PROGRAM,      READ   IN   LAST   VALUES    USED
          IND   =   5
           NX   =   0
3     WRITE     (*,1000)
      READ     (*,9000)     DUMMY
      OPEN    (9,FILE='B:CROPS',          STATUS='OLD')
      READ     (9,1104)     ( TITLE1(M),      M=1,18      )
      WRITE    (*,1001)     ( TITLE1(M),      M=1,18      )
      READ     (*,9001)     ANS
        IND   =   1
        IF   (  ANS    .EQ.    'y'  .OR.    ANS    .EQ.    'Y'  )  GO   TO   301
      READ     (9,1100)     KK,   JJ,   WD(1),    WD(2),    WD(3)
      DO   5  J=1,    JJ
      READ    (9,1101)     NAME(J),     LNGTH(J),      YMIN(1,J),      YMIN(2,J),      YMIN(3,J)
      READ    (9,1102)     ( ( P(I,J,K),     Y(I,J,K),     CL(I,J,K),      F(I,J,K),
     1                      U(I,J,K),     T(I,J,K),     B(I,J,K),     I=1,3    ),  K=1,KK     )
5     CONTINUE
      READ    (9,1103)     ( ( W(I,K),     K=1,KK     ),  I=1,3    )
      READ    (9,1103)     ( ( HH(I,K),     K=1,KK     ),  I=1,3    )
      READ     (9,1105)     ( NG(I),    I  =   1,3   )
      READ     (9,1106)     ( (PCROP(I,N),       N=1,NG(I)       ),  I  =   1,3   )
      READ     (9,1107)     ( ( ( JCROP(I,J,N),        N=1,10      ),  J=1,JJ      ),  I  =   1,3   )
      CLOSE     ( 9  )
C   Read    in   the   configuration         file
7     WRITE     (  *,1003      )
      READ     (  *,9000      )  FNAME
      IF   (  FNAME     .EQ.    '                  '  )  GO    TO    10
      OPEN     (  9,FILE=FNAME,           STATUS='OLD')
      READ     (9,1104)     ( TITLE2(M),      M=1,18      )
c  Check     configuration          filename
      WRITE    (*,1001)     ( TITLE2(M),      M=1,18      )
      READ     (*,9001)     ANS
        IND   =   2
        IF   (  ANS    .EQ.    'y'  .OR.    ANS    .EQ.    'Y'  )  GO   TO   301
      READ     (  9,9009     )  JI,   JR,   JS,   JD,   JT,   IRT,   JN,   NLIM(1)
      READ     (  9,9010     )  (  JIRR(K),     CCAPI(K),     KIRR(K),     IDNAME(K),       K=1,JI     )
      READ     (  9,9015     )  (  JRES(K),     SLO(K),     SUP(K),     K=1,JR      )
      READ     (  9,9011     )  (  JNOSPL(K),       K=1,JS      )
      READ     (  9,9011     )  (  JDIV(1,K),     JDIV(2,K),      K=1,JD      )
      READ     (  9,9013     )  (  CCAPD(K),       K=1,JD      )
      READ     (  9,9013     )  (  CLOSS(K),       K=1,JD      )
```

```
      READ   ( 9,9011   ) ( JTRB(1,K),    JTRB(2,K),    K=1,JD    )
      READ   ( 9,9014   ) ( NNAME(K),     K=1,JN    )
      READ   ( 9,9015   ) ( NLIM(K+1),FLOMIN(K),FLOMAX(K),      K=1,NLIM(1)-1     )
      READ   ( 9,9011   ) ( J1(1,K),    J1(2,K),    K=1,IRT    )
      READ   ( 9,9013   ) ( C1(K),    K=1,IRT    )
      CLOSE  ( 9 )
      GO TO 20
c Enter new configuration
10    CALL CONFIG ( 1, JI, JR, JS, JD, JT, JN )
c  1 = starting afresh
c JI = number of irrigation nodes plus 2
c JR = number of reservoir nodes plus 2
c JS = number of nodes not spilling to node numbered one higher plus 2
c JD = number of diversions plus 2
c JT = number of tributaries plus 2
c JN = total number of nodes
C Definition of the problem, constraints, objective function, number
C of sub-seasons
c MINCRP is an integer code for the method of assuring production
c        of certain crops, FORMAT 1004 explains the code
c LANDIF is an integer code for the method of decreasing idle land
c        FORMAT 1005 explains the code
20    WRITE ( *, 1004 )
      READ  ( *, 9002 ) MINCRP
      WRITE ( *,1005  )
      READ  ( *,9002  ) LANDIF
21    DO 22 I=1,3
        WRITE ( *,1006  ) I
        READ  ( *,9002  ) NSS(I)
        IF ( NSS(I) .EQ. 0 ) NSS(I) = 1
22    CONTINUE
C Permission to change the definition of the problem
      WRITE ( *,1007 ) MINCRP,   LANDIF,   (NSS(I),   I=1,3)
      READ  ( *,9001)  ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 20
C Choose next course of action
25    WRITE ( *,7002  )
      READ  ( *,9002  ) ICHOIX
      GO TO ( 80, 100, 7, 300, 600, 400, 500, 20, 700 ) ICHOIX
C Review irrigation return flows
30    WRITE ( *,7003  )
      WRITE ( *,7004  ) (K,C1(K),NNAME(J1(1,K)),NNAME(J1(2,K)),     K=1,IRT)
      WRITE ( *,7005  )
      READ  ( *,9001  ) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 25
C Additions, deletions or corrections are required to the irrigation
C district flows
31    WRITE ( *,7006  )
      READ  ( *,9002  ) ICH
       GO TO ( 40, 50, 60 ) ICH
      GO TO 31
C Add more Turnouts which have return flows
```

```
40    WRITE  ( *,7007  )
      READ  ( *,9002  ) NX
       IF ( NX .LE. 0 ) GO TO 30
      WRITE  ( *,9007  ) ( N, NAME(N),   N=1,JN  )
        DO 42 NN=1,NX
          WRITE  ( *,7008  ) NN
          READ  ( *,9002  ) J1(1,IRT+NN)
          WRITE  ( *,7009  )
          READ  ( *,9002  ) J1(2,IRT+NN)
          WRITE  ( *,7010  )
42         READ  ( *,9003  ) C1(IRT+NN)
        IRT = IRT + NX
        GO TO 30
C  Deletions
50    WRITE  ( *,7011  )
      READ  ( *,9002  ) NX
       IF ( NX .LE. 0 ) GO TO 30
        DO 52 NN=1,NX
          WRITE  ( *,7012  )
           READ  ( *,9002  ) KX
            DO 52 N1=KX,IRT
              IP1 = 1 + N1
        J1(2,N1)   = J1(2,IP1)
        J1(1,N1)   = J1(1,IP1)
52         C1(N1)   = C1(IP1)
        IRT = IRT - 1
         GO TO 30
C  Corrections
60    WRITE  ( *,7020  )
      READ  ( *,9002  ) NX
       IF ( NX .LE. 0 ) GO TO 30
        DO 69 NN=1,NX
         WRITE  ( *,7021  )
          READ  ( *,9002  ) IC
          WRITE  ( *,7022  )
          READ  ( *,9002  ) KX
          GO TO ( 66, 64, 65 ) IC
            GO TO 30
C  Node of turnout number wrong
64    WRITE  ( *,9007  ) ( N, NNAME(N),   N=1,JN  )
      WRITE  ( *,7023  ) KX, NNAME(J1(2,KX)),    C1(KX)
       READ  ( *,9002  ) J1(1,KX)
        GO TO 69
C Node of return wrong
65    WRITE  ( *,9007  ) ( N, NNAME(N),   N=1,JN  )
      WRITE  ( *,7024  ) KX, C1(KX),   NNAME(J1(1,KX))
       READ  ( *,9002  ) J1(2,KX)
        GO TO 69
C Percentage  of irrigation  wrong
66    WRITE  ( *,7025  ) NNAME(J1(1,KX)),    NNAME(J1(2,KX)),    KX
       READ  ( *,9003  ) C1(KX)
69    CONTINUE
```

```
      GO TO 30
C The CROPS, CONFIGURATION, INFLOW or CWR file called is incorrect
C The INFLOW file called is in correct, IND = 3
C The CWR file called is incorrect, IND = 4
C The CONFIGURATION file called is incorrect, IND=2
301   CLOSE (9)
      GO TO ( 3, 7, 405, 662 ) IND
C The datafile does not have the same number of sub-seasons
      WRITE (*,4010) ( NSS(I),I=1,3 ), IS1,IS2,IS3, (TITLE3(M), M=1,18)
      GO TO 21
C Display the nodes and characteristics of the current configuration
c 6 lines at a time
100             IS = 2
                IG = 2
                IR = 2
                IT = 2
      NBEG = 1
      JTIMES = JN/6 + 1
      DO 205 L=1,JTIMES
        WRITE ( *,8000 )
        NEND = 5 + NBEG
        IF ( NEND .GT. JN ) NEND = JN
      DO 200 I=NBEG, NEND
c Define values for each column in the REVIEW table
105       NAME1 = NNAME(I)
        DO 111 IM = 2,(JDIV(1,1)+1)
          IF ( JDIV(2,IM) .EQ. I ) GO TO 12
111       CONTINUE
          NAME4='----'
          GO TO 14
12        NAME4 = NNAME( JDIV(1,IM) )
14        DO 15 II = 2,(JDIV(1,1)+1)
          IF ( JDIV(1,II) .EQ. I ) GO TO 18
15        CONTINUE
          NAME3='----'
          CAP1= 0.0
          GO TO 121
18        NAME3 = NNAME(JDIV(2,II))
          CAP1 = CCAPD(II)
121         IF ( I .EQ. JNOSPL(IS) ) GO TO 125
23          NAME2 = NNAME(I+1)
            GO TO 131
125         IS = IS + 1
          IF ( JTRB(1,IT) .EQ. I ) GO TO 27
          NAME2 = '----'
            GO TO 131
27          NAME2 = NNAME( JTRB(2,IT) )
          IT = IT + 1
131         IRRD = 0
          NAME5 = '----'
          CAP2 = 0.0
        IF ( JIRR(IG) .EQ. I ) GO TO 35
```

```
      GO TO 141
35    IRRD = CKIRR(IG)
      NAME5 = IDNAME(IG)
      CAP2 = CCAPI(IG)
      IG = IG + 1
      DO 38 L1=2,5
      IF ( JIRR(IG) .LT. I .OR. JIRR(IG) .GT. I ) GO TO 141
      WRITE ( *,8001 ) NNAME(I), KIRR(IG), IDNAME(IG)
38    IG = IG + 1
141   CAP3 = 0.00
      CAP4 = 0.00
      IF ( JRES(IR) .EQ. I ) GO TO 45
      GO TO 150
45    CAP3 = SUP(IR)
      CAP4 = SLO(IR)
      IR = IR + 1
150   WRITE (*,3000) NAME1, NAME2, NAME3, CAP1, NAME4, IRRD, NAME5,
     1 CAP2, CAP3, CAP4
      READ (*,9002) INT
      IF ( INT .LE. 0 ) GO TO 200
      IF ( INT .GE. 99 ) GO TO 210
c Amend a value in the current line
      CALL CORRIG (INT,I,JN,IC,IM, II, IS, IG, IR, IT, NAME3, NAME4)
c INT = integer of incorrect column
c I = node number of current line
c JN = total number of nodes
c IC = code to skip replacement
c IM = diversion TO number of current line
c II = diversion FROM number of current line
c IS = nospill number of current line
c IG = irrigation number of current line
c IR = reservoir number of current line
c IT = tributary number of current line
c NAME3, NAME4 = column names which may be redefined by CORRIG directly
      WRITE ( *,8000 )
c Rewrite REVIEW table, changing the column to new value, if possible
      GO TO ( 110, 120, 130, 140, 150, 160, 165, 170, 180, 190) INT
      GO TO 150
110   NAME1 = NNAME(I)
      GO TO 150
120   NAME2 = 'PASS'
      GO TO 150
130   CAP1 = CCAPD(JDIV(1,1+1))
      GO TO 150
140   CAP1 = CCAPD(II)
      GO TO 150
160   IF ( IC .EQ. 2 ) GO TO 200
      IRRD = KIRR(IG-1)
      NAME5 = IDNAME(IG-1)
      CAP2 = CCAPI(IG-1)
      GO TO 150
165   NAME5 = IDNAME(IG-1)
```

```
      GO   TO   150
170     CAP2   =  CCAPI(IG-1)
      GO   TO   150
180     IF  ( IC  .EQ.   2  )  GO   TO   200
        CAP3   =  SUP(IR-1)
        CAP4   =  SLO(IR-1)
      GO   TO   150
190     CAP4   =  SLO(IR-1)
      GO   TO   150
200     CONTINUE
c  Start   REVIEW   table   again,   if  user   so   desires
      WRITE   (  *,3001    )
      READ    (  *,9001    )  ANS
      IF  (  ANS   .EQ.   'Y'  .OR.   ANS   .EQ.   'y'  )  GO   TO   100
      NBEG    =  NEND    +  1
205     CONTINUE
c  Amend   total   values   after   corrections
210               JI   =   1  +  JIRR(1)
                  JR   =   1  +  JRES(1)
                  JS   =   1  +  JNOSPL(1)
                  JD   =   1  +  JDIV(1,1)
                  JT   =   1  +  JTRB(1,1)
c   Does   the   user   wish   to   store   the   new   configuration?
330     WRITE   (*,3003)
      READ    (*,9001)     ANS
      IF  (  ANS   .EQ.   'Y'  .OR.   ANS   .EQ.   'y'  )  GO   TO   215
        GO   TO   220
215     WRITE   (  *,3004    )  FNAME
      READ    (  *,9000    )  NAME1
      IF  (  NAME1    .EQ.   '                 '  )  GO   TO   216
        FNAME    =   NAME1
216     OPEN   (  10,FILE=FNAME,          STATUS='OLD')
      WRITE   (*,1002)    (  TITLE2(M),     M=1,18     )
      READ    (*,1104)    (  DUMMYT(M),       M=1,18     )
        IF  (  DUMMYT(1)       .EQ.   '                 '  )  GO   TO   218
          DO   217   M=1,18
217               TITLE2(M)       =   DUMMYT(M)
218     WRITE   (  10,1104     )  (  TITLE2(M),      M=1,18      )
      WRITE   (  10,9009     )  JI,  JR,  JS,  JD,  JT,  IRT,  JN,  NLIM(1)
      WRITE   (10,9020)      (  JIRR(K),    CCAPI(K),    KIRR(K),    IDNAME(K),     K=1,JI    )
      WRITE   (  10,9015     )  (  JRES(K),    SLO(K),    SUP(K),    K=1,JR      )
      WRITE   (  10,9011     )  (  JNOSPL(K),       K=1,JS      )
      WRITE   (  10,9011     )  (  JDIV(1,K),      JDIV(2,K),      K=1,JD      )
      WRITE   (  10,9023     )  (  CCAPD(K),      K=1,JD      )
      WRITE   (  10,9023     )  (  CLOSS(K),      K=1,JD      )
      WRITE   (  10,9011     )  (  JTRB(1,K),      JTRB(2,K),      K=1,JT      )
      WRITE   (  10,9014     )  (  NNAME(K),       K=1,JN      )
      WRITE   (  10,9015     )  (  NLIM(K+1),FLOMIN(K),FLOMAX(K),              K=1,NLIM(1)-1       )
      WRITE   (  10,9011     )  (  J1(1,K),     J1(2,K),     K=1,IRT      )
      WRITE   (  10,9022     )  (  C1(K),    K=1,IRT      )
          CLOSE     (  10   )
c   Does   the   user   wish   to   append   nodes   to   the   configuration
```

```
220   WRITE (*,3002)
      READ (*,9001) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 225
      GO TO 25
225   CALL CONFIG ( 2, JI, JR, JS, JD, JT, JN )
c 2 = starting afresh
c JI = number of irrigation nodes plus 2
c JR = number of reservoir nodes plus 2
c JS = number of nodes not spilling to node numbered one higher plus 2
c JD = number of diversions plus 2
c JT = number of tributaries plus 2
c JN = total number of nodes
      GO TO 25
C DISPLAY THE CROP AND DISTRICT INFORMATION
C
300   CALL CHANGE ( JJ, KK, W, HH, MINCRP, LANDIF, NX, IFS, NOGO )
c JJ = number of crop varieties
c KK = number of irrigation districts
c W = array of delivery costs
c HH = array of total land per district
c NX = number of crops deleted
c IFS = code used for season advanced
c NOGO = code used if configuration data not matching Crops data
      IF ( NOGO .GE. 99 ) GO TO 25
C Change first season on title card if necessary
      NAME2 = TITLE1(1)
      IF ( TITLE1(1) .EQ. 'MAHA' .AND. IFS .GE. 1 ) NAME2='YALA'
      IF ( TITLE1(1) .EQ. 'YALA' .AND. IFS .GE. 1 ) NAME2='MAHA'
      TITLE1(1) = NAME2
C minimum production of certain crops guaranteed.
      IF ( MINCRP .GE. 1 ) CALL CHANJ3 ( JJ, NMAX, NREM, NX )
c JJ = total number of crops
c NMAX = maximum number of crops and seasons subjected to targets
c NREM = NMAX modulus 3, the extra crops and seasons for the last line
c NX = number of crops deleted
c Grouping some crops under common targets
      IF ( MINCRP .GT. 1 ) CALL COMCRP ( JJ )
c Does the user wish to store revisions
      WRITE ( *,1008 )
      READ ( *,9001 ) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 350
      GO TO 25
C All data updated, Write the revised values to the file CROPS.
350   OPEN (9,FILE='B:CROPS', STATUS='OLD')
      WRITE (*,1002) ( TITLE1(M), M=1,18 )
      READ (*,1104) ( DUMMYT(M), M=1,18 )
      IF ( DUMMYT(1) .EQ. '          ' ) GO TO 355
      DO 352 M=1,18
352   TITLE1(M) = DUMMYT(M)
355   WRITE (9,1104) ( TITLE1(M), M=1,18 )
      WRITE (9,1100) KK, JJ, WD(1), WD(2), WD(3)
      DO 360 J=1, JJ
```

```
      WRITE (9,1101) NAME(J), LNGTH(J), ( YMIN(I,J), I= )
      WRITE (9,1102) ( ( P(I,J,K), Y(I,J,K), CL(I,J,K), F(I,J,K),
     1 U(I,J,K), T(I,J,K), B(I,J,K), I=1,3 ), K=,,KK )
360   CONTINUE
      WRITE (9,1103) ( ( W(I,K), K=1,KK ), I=1,3 )
      WRITE (9,1103) ( ( HH(I,K), K=1,KK ), I=1,3 )
      WRITE (9,1105) ( NG(I), I = 1,3 )
      WRITE (9,1106) ( (PCROP(I,N), N=1,NG(I) ), I = 1,3 )
      WRITE (9,1107) ( ( ( JCROP(I,J,N), N=1,10 ), J=1,JJ ), I = 1,3 )
      CLOSE ( 9 )
        GO TO 25
C Ask if the RHS constants are on a datafile or will need to be entered
C  interactivly
400   WRITE ( *,4000 )
      READ ( *,9001 ) ANS
        IF ( ANS .EQ. 'y' .OR. ANS .EQ. 'Y' ) GO TO 465
C  Read and verify RHS constants from datafile
405   WRITE ( *,4001 )
      READ ( *,9000 ) FNAME
      IF ( FNAME .EQ. '                ' ) GO TO 462
      OPEN ( 9,FILE=FNAME, STATUS='OLD')
      READ (9,9104) IS1, IS2, IS3, ( TITLE3(M), M=1,18 )
          IND = 5
        IF ( IS1 .NE. NSS(1) .OR. IS2 .NE. NSS(2) .OR. IS3 .NE. NSS(3) )
     1 GO TO 301
      WRITE (*,1001) ( TITLE3(M), M=1,18 )
      READ (*,9001) ANS
        IND = 3
        IF ( ANS .EQ. 'y' .OR. ANS .EQ. 'Y' ) GO TO 301
      DO 461 N=1,JN
461   READ (9,9030) KNAME(N), ( (CRHS(I,N,IS), IS=1,NSS(I)), I=1,3 )
      CLOSE (9)
c Display net inflows for 15 nodes at a time
462        NTIMES = JN/15 + 1
            NBEG = 1
      DO 447 L=1,NTIMES
         WRITE ( *,4009 )
         WRITE ( *,4002 )
         NEND = 15 + NBEG
         IF ( NEND .GT. JN ) NEND = JN
         DO 445 I=1,3
           WRITE ( *,4021 ) I, ( IS, IS=1,NSS(I) )
443         DO 446 N=NBEG, NEND
c Verify everything correct
446           WRITE ( *,4022 ) N, NNAME(N), KNAME(N), ( CRHS(I,N,IS),
     1                                      IS=1,NSS(I) )
            WRITE ( *,4003 )
            READ ( *,9002 ) NX
            IF ( NX .LE. 0 ) GO TO 445
C Corrections too many, read different file, or correct in stages
         IF ( NX .GT. 5 ) GO TO 405
C Corrections one at a time allowed
```

```
      DO 444 NN=1,NX
        WRITE ( *,4004 )
        READ ( *,9002 ) NWRONG
        WRITE ( *,4005 ) NNAME(NWRONG)
        READ ( *,9002 ) IWRONG
        WRITE ( *,4006 ) NNAME(NWRONG), IWRONG
444     READ ( *,9003 ) CRHS(I,NWRONG,IWRONG)
      GO TO 443
445   CONTINUE
447   NBEG = NEND + 1
C If desired, the corrected file saved on diskette
455   WRITE ( *,4007 )
      READ ( *,9001 ) ANS
      NALLOW = 1
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 456
        GO TO 25
456   WRITE ( *,4008 ) FNAME
      READ ( *,9000 ) NAME1
      IF ( NAME1 .EQ. '            ' ) GO TO 459
        FNAME = NAME1
459   OPEN ( 9,FILE=FNAME, STATUS='OLD')
      WRITE (*,1002) ( TITLE3(M), M=1,18 )
      READ (*,1104) ( DUMMYT(M), M=1,18 )
      IF ( DUMMYT(1) .EQ. '            ' ) GO TO 463
        DO 460 M=1,18
460       TITLE3(M) = DUMMYT(M)
463   WRITE ( 9,9104 ) NSS(1), NSS(2), NSS(3), ( TITLE3(M), M=1,18 )
      DO 464 N=1,JN
464   WRITE ( 9,9031 ) NNAME(N),( (CRHS(I,N,IS),IS=1,NSS(I)),I=1,3 )
      CLOSE (9)
        GO TO 25
C Calculate and enter the RHS constants interactively
465   DO 479 N=1,JN
      WRITE ( *,4009 )
      DO 479 I=1,3
      DO 479 IS=1,NSS(I)
470   WRITE ( *,4020 ) I, NNAME(N), IS
      READ ( *,9003 ) DINF
475   WRITE ( *,4025 ) I, NNAME(N), IS
      READ ( *,9003 ) EVAP
      WRITE ( *,4026 ) NNAME(N), IS, I, DINF, EVAP
      READ ( *,9002 ) INT
        GO TO ( 470, 475 ) INT
        IF ( INT .GT. 2 ) GO TO 470
479   CRHS(I,N,IS) = DINF - EVAP
      GO TO 462
C Review or amend restrictions to the regulated downstream releases
80    WRITE ( *,1803 )
      WRITE ( *,1804 ) ( K, NNAME(NLIM(K+1)), FLOMIN(K), FLOMAX(K),
     1                K=1,NLIM(1)-1 )
      WRITE ( *,1805 )
      READ ( *,9002 ) ICH
```

```
      GO TO ( 810, 820, 830, 215 ) ICH
      GO TO 25
C Additions, deletions, or changes are required to NLIM, FLOMAX, and FLOMIN
810  WRITE ( *,1806 )
      READ ( *,9002 ) NX
       IF ( NX .LE. 0 ) GO TO 80
      WRITE ( *,1807 ) ( N, NNAME(N), N=1,JN )
        DO 812 N=1,NX
          WRITE ( *,1808 ) N
          READ ( *,9002 ) NNEW
          WRITE ( *,1809 ) NNAME(NNEW)
          READ ( *,9003 ) FLOMIN(NLIM(1))
          WRITE ( *,1810 ) NNAME(NNEW)
          READ ( *,9003 ) FLOMAX(NLIM(1))
           IF ( FLOMAX(NLIM(1)) .LE. 0.05 ) FLOMAX(NLIM(1)) = 99999
          NLIM(1) = NLIM(1) + 1
812     NLIM(NLIM(1)) = NNEW
        GO TO 80
C Deletions
820  WRITE ( *,1811 )
      READ ( *,9002 ) NX
       IF ( NX .LE. 0 ) GO TO 80
        DO 826 N=1,NX
          WRITE ( *,1812 ) N
          READ ( *,9002 ) KX
          DO 824 M=KX,NLIM(1)-2
           FLOMAX(M) = FLOMAX(M+1)
           FLOMIN(M) = FLOMIN(M+1)
824        NLIM(M+1) = NLIM(M+2)
826     NLIM(1) = NLIM(1) - 1
        GO TO 80
C Corrections
830  WRITE ( *,1813 )
      READ ( *,9002 ) NX
       IF ( NX .LE. 0 ) GO TO 80
        DO 835 N=1,NX
          WRITE ( *,1815 )
          READ ( *,9002 ) KX
832       WRITE ( *,1816 ) KX
          READ ( *,9000 ) NAME1
           IF ( NAME1 .EQ. ' ' ) GO TO 833
C The correct min/max flow values were assigned to the wrong node
C Find the node number of correct node
      DO 855 K=1,JN
       IF ( NNAME(K) .EQ. NAME1 ) GO TO 858
855  CONTINUE
      WRITE (*,1814) NAME1
      GO TO 832
C Replace the node number, NLIM(KX), with the correct number, K
858    NLIM(KX+1) = K
833       WRITE ( *,1817 ) NNAME(NLIM(KX+1))
          READ ( *,9003 ) DUMMY
```

```
      IF ( DUMMY .GT. 0.005 ) FLOMIN(KX) = DUMMY
      WRITE ( *,1818 ) NNAME(NLIM(KX+1))
      READ ( *,9003 ) DUMMY
      IF ( DUMMY .GT. 0.005 ) FLOMAX(KX) = DUMMY
835   CONTINUE
      GO TO 80
C Entry of sub-seasonal crop water requirements
600   WRITE ( *,6000 )
      READ ( *,9001 ) ANS
      IF ( ANS .EQ. 'y' .OR. ANS .EQ. 'Y' ) GO TO 605
      GO TO 662
C Calculate the constant water demand U(i,j,k)/NSS(i)
605   DO 610 J=1,JJ
      DO 610 K=1,KK
      DO 610 I=1,3
      WCONST = U(I,J,K)/NSS(I)
      DO 610 IS=1,NSS(I)
610   WCOEF(I,J,K,IS) = WCONST
C Display the constant water demand of each crop each season, paged
C by district
615   WRITE (*,6010)
      READ (*,9002) ICROP
      IF ( ICROP .LE. 0 ) GO TO 655
      DO 625 L=1,JJ
      J = L
C Allow only specific crops to be reviewed
      IF ( ICROP .LT. 99 ) J = ICROP
      NTIMES = KK/15 + 1
      NBEG = 1
      DO 645 LL=1,NTIMES
      DO 645 I=1,3
      WRITE ( *,6002 ) NAME(J), I
      NEND = 15 + NBEG
      IF ( NEND .GT. KK ) NEND = KK
      WRITE ( *,6021 ) ( IS, IS=1,NSS(I) )
643   DO 646 K=NBEG, NEND
C Find the district name to match the district
      DO 6465 KLOOK = 1,JIRR(1)+1
      IF ( KIRR(KLOOK) .EQ. K ) GO TO 646
6465  CONTINUE
646   WRITE ( *,6022 ) K, IDNAME(KLOOK), KNAME(K),
     1 ( WCOEF(I,J,K,IS), IS=1,NSS(I) )
      WRITE ( *,6003 )
      READ ( *,9002 ) NX
      IF ( NX .LE. 0 ) GO TO 645
C Check if the wrong file read
      WRITE (*,6005)
      READ (*,9001) ANS
      IF ( ANS .EQ. 'y' .OR. ANS .EQ. 'Y' ) GO TO 662
C Determine if a whole row or whole column will be changed
      WRITE ( *,6020 )
      READ ( *,9002 ) ICHOIX
```

```
      GO TO ( 630, 640 ) ICHOIX
C Corrections performed one at a time
      DO 644 NN=1,NX
        WRITE ( *,6004 )
        READ ( *,9002 ) KX
        WRITE ( *,6006 ) IIX, KX
        READ ( *,9002 ) ISX
        WRITE ( *,6007 ) KX, ISX, IIX
        READ ( *,9003 ) DUMMY
        IF ( DUMMY .LT. 0 ) GO TO 644
          WCOEF( IIX, J, KX, ISX ) = DUMMY
644     CONTINUE
      GO TO 643
C Corrections performed for a whole row, an individual season
630   WRITE ( *,6004 )
      READ ( *,9002 ) KX
        IIX = I
      DO 632 IS=1,NSS(IIX)
        WRITE ( *,6007 ) KX, IS, IIX
        READ ( *,9003 ) DUMMY
        IF ( DUMMY .LT. 0 ) GO TO 632
          WCOEF( IIX, J, KX, IS ) = DUMMY
632   CONTINUE
      GO TO 643
C Corrections performed for a whole column
640   WRITE ( *,6041 ) IIX
      READ ( *,9002 ) ISX
      DO 642 K=1,KK
        WRITE ( *,6007 ) K, ISX, IIX
        READ ( *,9003 ) DUMMY
        IF ( DUMMY .LT. 0 ) GO TO 642
          WCOEF( IIX, J, K, ISX ) = DUMMY
642   CONTINUE
      GO TO 643
C All corrections done on this page of crop values, next screen
645   CONTINUE
      IF ( ICROP .LT. 99 ) GO TO 615
647   NBEG = NEND + 1
625   CONTINUE
C If desired, the corrected file saved on diskette
655     WRITE ( *,6008 )
        READ ( *,9001 ) ANS
          NALLO1 = 1
        IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 656
          GO TO 25
656     WRITE ( *,6009 ) FNAME
        READ ( *,9000 ) NAME1
        IF ( NAME1 .EQ. '          ' ) GO TO 659
          FNAME = NAME1
659   OPEN ( 9,FILE=FNAME, STATUS='OLD')
      WRITE (*,1002) ( TITLE3(M), M=1,18 )
      READ (*,1104) ( DUMMYT(M), M=1,18 )
```

```
      IF ( DUMMYT(1) .EQ. '                    ' ) GO TO 663
        DO 660 M=1,18
660       TITLE3(M) = DUMMYT(M)
663   WRITE (9,9104) NSS(1), NSS(2), NSS(3), ( TITLE3(M), M=1,18 )
      DO 664 J=1,JJ
        WRITE (9,9000) NAME(J)
        DO 664 K=1,KK
664   WRITE (9,9061) KNAME(K),( (WCOEF(I,J,K,IS),IS=1,NSS(I)),I=1,3 )
      CLOSE (9)
        GO TO 25
C Read and verify sub-seasonal CWR from datafile
662   WRITE ( *,6001 )
      READ ( *,9000 ) FNAME
      IF ( FNAME .EQ. '                    ' ) GO TO 615
      OPEN ( 9,FILE=FNAME, STATUS='OLD')
      READ (9,9104) IS1, IS2, IS3, ( TITLE3(M), M=1,18 )
        IND = 5
        IF ( IS1 .NE. NSS(1) ) GO TO 301
        IF ( IS2 .NE. NSS(2) ) GO TO 301
        IF ( IS3 .NE. NSS(3) ) GO TO 301
      WRITE (*,1001) ( TITLE3(M), M=1,18 )
      READ (*,9001) ANS
        IND = 4
        IF ( ANS .EQ. 'y' .OR. ANS .EQ. 'Y' ) GO TO 301
      DO 661 J=1,JJ
        READ (9,9000) NAME(J)
        DO 661 K=1,KK
661   READ (9,9060) KNAME(K), ( (WCOEF(I,J,K,IS), IS=1,NSS(I)), I=1,3 )
      CLOSE (9)
C Ask if the crop requirements need to be advanced one season
      WRITE ( *,1009 ) TITLE3(1)
      READ ( *,9001 ) ANS
      IF ( ANS .EQ. 'N' .OR. ANS .EQ. 'n' ) GO TO 650
        GO TO 615
C First season now MAHA, second YALA or vice versa
650   DO 653 I=1,2
        DO 653 J=1,JJ
          DO 653 K=1,KK
            DO 653 IS=1,NSS(I)
653   WCOEF(I,J,K,IS) = WCOEF(I+1,J,K,IS)
C Set last season's CWR equal to first
        DO 670 J=1,JJ
          DO 670 K=1,KK
            DO 670 IS=1,NSS(3)
670   WCOEF(3,J,K,IS) = WCOEF(1,J,K,IS)
C Change season on file's titlecard
          NAME2 = TITLE3(1)
        IF ( TITLE3(1) .EQ. 'MAHA' ) NAME2 = 'YALA'
        IF ( TITLE3(1) .EQ. 'YALA' ) NAME2 = 'MAHA'
          TITLE3(1) = NAME2
          GO TO 615
C Determine how long, in days, each sub-season is
```

```
500     DO 550 I=1,3
        DO 550 IS=1,NSS(I)
501       WRITE ( *,5000 ) IS, I
          READ ( *,9003 ) DURATN(I,IS)
C   Check that the duration is correct before continuing
          WRITE ( *,5001 ) DURATN(I,IS), I, IS
          READ ( *,9001 ) ANS
           IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 501
C   Calculate flow restrictions from the DURATN in days of the sub-season
          DO 505 N=1,NLIM(1)
             FLMIN(I,IS,N) = FLOMIN(N) * DURATN(I,IS) * 0.0864
              FLMAX(I,IS,N) = 99999
          IF ( FLOMAX(N) .LT. 99999 )
     1          FLMAX(I,IS,N) = FLOMAX(N) * DURATN(I,IS) * 0.0864
505       CONTINUE
C   Calculate irrigation channel capacities from the DURATN in days
          DO 510 N=1,JIRR(1)
             CHCPI(I,IS,N) = 99999
          IF ( CCAPI(N+1) .LT. 99999 )
     1          CHCPI(I,IS,N) = CCAPI(N+1) * DURATN(I,IS) * 0.0864
510       CONTINUE
C   Calculate the diversion channel capacities from the DURATN in days
C   and set channel losses constant
          DO 515 N=1,JDIV(1,1)
           CHCPD(I,IS,N) = 99999
          IF ( CCAPD(N+1) .LT. 99999 )
     1          CHCPD(I,IS,N) = CCAPD(N+1) * DURATN(I,IS) * 0.0864
515          CHLSS(I,IS,N) = CLOSS(N+1)
C   Set reservoir limits constant
          DO 520 N=1,JRES(1)
              SMIN(I,IS,N) = SLO(N+1)
520            SMAX(I,IS,N) = SUP(N+1)
C   If there are irrigation return flows, set the percentage constant
        IF ( IRT .LE. 0 ) GO TO 550
          DO 525 N=1,IRT
525          P1(I,IS,N) = C1(N)
550     CONTINUE
C   For the review, entry or saving of variable constants, use the
C                      VARDATA subroutine
        CALL VARDATA ( JN, IRT )
c   JN = total number of nodes
c   IRT = number of instances of irrigation return flows
        NALLO2 = 1
        GO TO 25
C   All the values have now been updated. The file b:PLAN.DATA will be
C   prepared for inclusion as a sparsedata SAS.OR datafile.
C
C  Check that INFLOWS have been entered
700     IF ( NALLOW .GT. 0 ) GO TO 701
          WRITE ( *,7100 )
          GO TO 400
C   Check that CWR have been entered
```

```
701     IF ( NALLO1 .GT. O ) GO TO 702
        WRITE ( *,7101 )
        GO TO 600
C   Check that the duration of the subseasons has been entered
702     NSSUM = NSS(1) + NSS(2) + NSS(3)
        IF ( NSSUM .LE. 3 .OR. NALLO2 .GT. O ) GO TO 705
        WRITE ( *,7102 )
        GO TO 500
C   Convert CWR to Mm3
705     DO 715 I=1,3
        DO 715 J=1,JJ
        DO 715 K=1,KK
        DO 715 IS=1,NSS(I)
715     WCOEF(I,J,K,IS) = WCOEF(I,J,K,IS) / 1000
C   Convert percent values for inclusion in LP calculations
        DO 750 I=1,3
        DO 750 IS=1,NSS(I)
        DO 740 N=1,JDIV(1,1)
740     CHLSS(I,IS,N) = CHLSS(I,IS,N) / 100
        DO 745 N=1,IRT
745     P1(I,IS,N) = P1(I,IS,N) / 100
750     CONTINUE
C   Ask that the PLAN.DATA diskette be inserted
        WRITE (*,1010)
        READ (*,9001 ) ANS
        OPEN (8, FILE='B:PLAN.DATA', STATUS='OLD')
        WRITE ( 8,2000 )
C
C   All the values for the sparsedata file have been defined for the run.
C   The variables of the continuity constraint will be written
        CALL CNT ( JN, IRT )
c   JN = total number of nodes
c   IRT = number of instances of irrigation return flows
        CALL DATALN ( JJ, KK, MINCRP, LANDIF, NMAX, NREM, JN )
c   JJ = number of crop varieties
c   KK = number of irrigation districts
c   NMAX = total number of production targets by season and crop
c   NREM = number of targets to be written on last line
c   JN = total number of nodes
c
C   OBJECTIVE FUNCTION
        DO 755 I=1,3
        DO 755 J=1,JJ
        DO 755 K=1,KK
755     Q(I,J,K) = P(I,J,K) - F(I,J,K) - CL(I,J,K) - U(I,J,K)*W(I,K)
C   Now the IiJjKk variables will be written.
        DO 760, K=1,KK
        DO 760, J=1,JJ
        DO 760, I=1,3
          IF ( T(I,J,K) .LE. 0.0 ) GO TO 760
            CALL CHOICE ( I, J, K, MINCRP, LANDIF )
c   I = season index
```

```
c   J  =  crop  index
c   K  =  irrigation  district  index
c  MINCRP  =  production  code
c  LANDIF  =  idle  land  code
760     CONTINUE
C  Write  the  constants  on  the  RHS  of  the  constraints
      CALL  RHS  ( JJ,  KK,  HH,  MINCRP,  LANDIF,  NMAX,  NREM,  JN  )
c   JJ  =  number  of  crop  varieties
c   KK  =  number  of  irrigation  districts
c   HH  =  array  of  total  land  per  district
c  NMAX  =  maximum  number  of  crops  and  seasons  subjected  to  targets
c  NREM  =  NMAX  modulus  3,  the  extra  crops  and  seasons  for  the  last  line
c   JN  =  total  number  of  nodes  in  the  configuration
      CLOSE  ( 8 )
C
1000    FORMAT  ( ' This  programme  prepares  the  crop.data  file  for  SAS.OR
     1solution  of  the  optimal  cropping  pattern  for  the  next  three  seaso
     2ns.  Answer  the  questions  as  they  ap-pear  on  the  screen.'/  Plea
     3se  hit  the  RETURN  when  ready  to  read  the  file  CROPS  on  drive  b:'/)
1001    FORMAT  ( ' The  file  to  be  read  has  the  following  title  card:'/1H  ,
     118A4/'  Do  you  wish  to  enter  another  file?  Answer  Y  to  change  the
     2diskette  in  the  drive  or  the  filename  ' )
1002    FORMAT  ( ' The  current  title  card  for  the  file  is:'/1H  ,18A4/'  Do
     1you  wish  to  enter  another  title?  If  you  enter  nothing  before  the
     2return,  the  above  title  card  will  be  used.  Enter  a  title  in  80  or
     3  less  characters,  the'/'  first  4  spaces  are  reserved  for  the  name
     4of  the  first  season  in  capital'/'  letters'/  )
1003    FORMAT  ( ' Under  what  name  is  the  configuration  of  the  Mahaweli  Pr
     5oject  saved?'/'  Enter  b:FNAME,  where  b:  is  the  drive  specification
     6  and  FNAME  is  the  name  in  10  or  less  letters.'/'  If  you  enter  not
     7hing  before  the  RETURN  you  may  enter  an  entirely  new'/'  configurat
     8ion  ' )
1004    FORMAT  ( ' Do  you  wish  to  specify  minimum  production  limits  for  ce
     3rtain  crops?'/'  Enter  3  if  you  wish  to  group  some  crops  as  well  as
     4  specifying  minimum  productionby  district  and  by  crop'/6X,'2  if  yo
     5u  wish  to  group  some  crops  under  a  minimum  production  limit'/6X,
     6'1  if  you  wish  to  specify  minimum  crop  production  throughout  the
     7project'/  6X,'0  or  nothing  before  the  RETURN  to  require  minimum  he
     8ctares  planted  in  each  crop,  in  each  district.  ')
1005    FORMAT  ( ' Do  you  wish  that  as  equal  amounts  of  land  as  possible
     1are  cultivated  between'/'  seasons,  even  under  severe  water  shortag
     2es,  ANSWER:'  /  2  if  the  difference  between  seasons  may  b
     3e  assessed  monetarily'/'  1  if  you  wish  to  only  set  limits  to  how
     4  much  land  may  be  left  idle'/'  0  or  nothing  if  you  prefer  not  to
     5have  any  such  constraint.  ' )
1006    FORMAT  ( ' How  many  sub-seasons  will  be  required  to  balance  the  c
     1ontinuity  equations  of  theseason  ',I2,'?  The  crop  water  demand  sh
     2ould  be  constant  during  a  sub-season.'/'  Every  sub-season  occurs  a
     3t  the  same  time  in  every  district.'/'  Enter  an  INTEGER  number,  fiv
     4e  or  less  ')
1007    FORMAT  ( ' The  following  has  been  entered:'//3X,1H[,I2,1H],'  the
     1code  for  crop  production  minimum  limits'/20X,'(  1  -  production  lim
```

```
     2ited by crop'/22X,'2 - crops grouped to meet a limit'/22X,'3 - cro
     3ps grouped and surpassing individual district and'/26X,'crop limit
     4s)'//3X,1H[,I2,1H],' the code for equal land use between seasons'/
     520X,'( 1 - idle land between seasons restricted to a maximum'/ 22X
     6,'2 - the difference may be compared with the economic'/26X,'retur
     7ns of the crops'//3X,1H[,I2,1H],' the number of sub-seasons in sea
     8son 1'//3X,1H[,I2,1H],' the number of sub-seasons in season 2'//3X
     9,1H[,I2,1H],' the number of sub-seasons in season 3'//' Do you wis
     1h to re-enter any of these values? Enter Y to re-do ' )
1008  FORMAT ( ' Do you wish to save the updated data in the file b:CROP
     1S? Answer Y to save. ' )
1009  FORMAT ( ' Do you wish to start with the values as read, with the
     1',A4,' season first?'/' Answer N to advance the values one season
     2. The new values for the third season will be the same as the new
     3 values for the first season ' )
1010  FORMAT ( ' Hit RETURN when you are ready to open the file b:PLAN.D
     1ATA on drive b:. ')
1803  FORMAT ( ' The following nodes have these restrictions on their'/
     1 regulated downstream releases:'/' 99999.0 is for unlimited flow'
     2/15X,4HNAME,9X,9HMIN. FLOW,3X,9HMAX. FLOW/30X,4Hm3/s,8X,4Hm3/s/ )
1804  FORMAT ( I3,9X,A12,2X,F8.2,F13.1 )
1805  FORMAT ( /' Is this table correct? Enter the code for the error:'
     1/' 0 or nothing to continue with these values'/
     2' 1 to add more nodes to the list'/
     3' 2 to remove some nodes from the list'/
     4' 3 to correct a value listed'/' 4 to save any corrections ' )
1806  FORMAT ( ' How many more nodes do you wish to add? Enter an INTEG
     1ER value. ' )
1807  FORMAT ( I3,2X,A12, I3,2X,A12, I3,2X,A12, I3,2X,A12 )
1808  FORMAT ( ' What is the number of the ',I2,'st node which now has r
     1estrictions to the regulated releases? Enter an INTEGER value ' )
1809  FORMAT ( ' What is the minimum flow in m3/s above which the regula
     1ted downstream releases of ',A15,' must be? ' )
1810  FORMAT ( ' What is the maximum flow in m3/s below which the regula
     1ted downstream releases of ',A15,' must be? ' )
1811  FORMAT ( ' How many nodes do you wish to remove? Enter an INTEGER
     1 number. ' )
1812  FORMAT ( ' What is the number to the left of the node to be remove
     1d the ',I2,'st time? Enter the latter numbers BEFORE the first n
     2umbers when deleting more than one node.'/' Enter an INTEGER numbe
     3r. ' )
1813  FORMAT ( ' How many nodes do you wish to correct? ' )
1814  FORMAT ( ' No node of the name ',A15, ' was found. Is it correct
     2ly spelt in full? ')
1815  FORMAT ( ' What is the number to the left of the line in which an
     1error occurs? Enter an INTEGER number. ' )
1816  FORMAT ( ' Enter the correct name for the node in line ',I3,'. If
     1 the name is correct, enter nothing before RETURN. ' )
1817  FORMAT ( ' What is the correct minimum flow in m3/s above which th
     1e regulated downstream'/' releases of ',A15,' must be?'/' If the v
     2alue listed was correct, enter nothing before the RETURN. ' )
1818  FORMAT ( ' What is the correct maximum flow in m3/s below which th
```

```
     1e  regulated  downstream'/'  releases  of  ',A15,'  must  be?'/'  If  the  v
     2alue  listed  was  correct,  enter  nothing  before  the  RETURN.  ' )
2000  FORMAT  ( 15H  DATA  PLANDATA;/23H  INPUT  _TYPE_  $  _COL_
$/9X,51H_ROW1
     1_  $  _COEF1_  _ROW2_  $  _COEF2_  _ROW3_  $  _COEF3_;/7H  CARDS;  )
3000  FORMAT  ( 1X,A12,2X,A4,3X,A4,F10.3,3X,A4,I4,3X,A4,1X,3F7.0,2H  ? )
3001  FORMAT  ( ' Do  you  wish  to  see  the  nodes  again?  Answer  Y  to  revie
     1w  the  nodes  already  dis-played.  ' )
3002  FORMAT  ( ' Do  you  wish  to  add  nodes  to  the  configuration?  Answe
     1r  Y  if  you  do.  ' )
3003  FORMAT  ( ' Do  you  wish  to  store  the  updated  version?  Answer  Y  if
     1you  do:  ' )
3004  FORMAT  ( ' Do  you  wish  to  save  the  configuration  under  ',A12,
     1  '?  You  may  change  the  diskette  in  drive  b:  before  entering.'
     2/  ' Enter  b:FNAME,  where  FNAME  is  the  name  of  an  existing  file  in
     310  or  less  letters  to  change  the  name.'/  ' If  you  enter  nothing  be
     4fore  RETURN,  the  name  will  stay  the  same.  '/ )
4000  FORMAT  ( ' Do  you  wish  to  enter  the  seasonal  estimated  volume  of  w
     1ater  available  for'/'  irrigation  and  the  estimated  volume  of  seaso
     2nal  water  losses  for  every  water  -  balance  node  interactively?
     3Answer  Y  if  you  have  no  file  prepared.  ' )
4001  FORMAT  ( ' What  is  the  name  of  the  file  which  contains  the  net  wat
     1er  available  for'/'  irrigation  by  sub-season  by  water-balance  node
     2  in  the  order  of  the  water-balance  nodes  in  the  current  configurat
     3ion?'/  ' Enter  b:FNAME,  where  FNAME  is  the  name  of  the  file  in  10
     4or  less  letters.'/'  If  you  enter  nothing  before  the  return  you  wil
     5l  be  able  to  review  what  has'/'  already  been  entered  ' )
4002  FORMAT  ( ' These  are  the  net  inflows  to  the  water  balance  nodes  i
     1n  MCM:'/'  No.  Current  Name  Name  on  File')
4021  FORMAT  ( ' In  the  season  ',I1,12X,10Hsub-season,6X,I1,3I13  )
4022  FORMAT  ( I3,2X,A12,3X,A12,3X,4F13.3  )
4003  FORMAT  ( ' Are  these  correct?  Answer  the  INTEGER  number  wrong.'/'
     1  99  or  more  allows  you  to  change  the  file'/'  0  or  nothing  allows  y
     2ou  to  continue  with  these  values.  ' )
4004  FORMAT  ( ' What  is  the  INTEGER  to  the  left  of  the  row  which  has  a
     1  value  you  would  like  to  correct?  ' )
4005  FORMAT  ( ' What  is  the  INTEGER  for  the  sub-season  of  the  value  at
     1the  water  -  balance  node'/  A15,'  which  you  would  like  to  correct
     2?  ')
4006  FORMAT  ( ' What  is  the  net  water  inflow  at  the  node  ',A15,'  in  th
     1e  sub-season  ',I3,'  after  seasonal  evaporation  and  seepage  losses
     2have  been  deducted,  in  MCM?  ' )
4007  FORMAT  ( ' Do  you  wish  to  save  the  updated  version?  Answer  Y  if  y
     1ou  do.  ' )
4008  FORMAT  ( ' Do  you  wish  to  save  the  updated  version  under  the  same
     1filename,  ',A15/'  Enter  b:FNAME  where  b:  is  the  drive  and  FNAME  is
     2the  new  name  or  nothing  to  keep  the  same  name.  ' )
4009  FORMAT  ( 80(1H*)/'  * Please  include  the  current  volume  of  water  st
     1ored  in  any  reservoir  with  the  *'/'  * inflow  to  that  node  in  the  f
     2irst  sub-season  of  the  first  season.  ',T80,1H*/80(1H*)  )
4010  FORMAT  ( ' The  number  of  sub-seasons  on  the  datafile  and  given  in
     1  this  run,  do  not  agree.  Currently  there  are  ',I2,'  sub-seasons  i
```

```
     2n the first season,',I3,' sub-seasons in the second and ',I2,' in
     3the last. The first line of the file is:'/1X,3I2,18A4 )
4020 FORMAT ( ' In the season ',I3,' what is the expected amount of run
     1off that will be available for storage or immediate diversion or
     2irrigated consumption at the water-balance node ',A15,' during th
     3e sub-season ',I2,'? Answer in MCM/sub-season. ' )
4025 FORMAT ( ' In the season ',I3,' what is the expected loss of water
     2 due to evaporation and seepage at the water-balance node ',A
     315,' during the sub-season',I2,'?'/ Answer in MCM/sub-season. ' )
4026 FORMAT ( ' At the water-balance node ',A15, ' during the sub-seas
     1on ',I3/ of the season ',I2,' the local inflow is:',F12.2 / t
     2he sub-seasonal losses are:',F12.2/ Please check that the figures
     3 have been entered correctly.'/ ' Enter 1 to redo completely'/5X,'
     42 to enter a different figure for the losses'/ 5X,'0 or nothing to
     5 continue with these values. ' )
5000 FORMAT ( ' How long is the sub-season ',I3,' in the season ',I3,'?
     1 Answer in the number of days ' )
5001 FORMAT ( ' Do you wish to CORRECT the value? '/1X,F5.1,' days in t
     1he sub-season ',I3,' of the season',I3,' have been entered'/ Answ
     2er Y to re-enter the value ' )
6000 FORMAT ( ' Do you wish to enter the sub-seasonal crop water demand
     1s from a prepared file orinteractively? Sub-seasonal values may b
     2e calculated from the seasonal value'/ listed in CROPS.'/ Answer
     3 Y if you have no file prepared or if each season has ONE sub-seas
     4on ' )
6001 FORMAT ( ' Enter b:FNAME where b: designates the drive and FNAME t
     1he name of a file in 10 or less characters which contains the cro
     2p water requirements prepared for this run. If you enter nothing
     3before the RETURN you will be able to review what has already been
     4 entered ' )
6002 FORMAT ( ' The crop ',A8,' has the following sub-seasonal water de
     1mands in'/ 1000 m3/hectare for the following districts in the sea
     2son',I2 )
6021 FORMAT ( 2X,21H No. Cur.NAME On File,2X,11HSUB-SEASON:,5I7 )
6022 FORMAT ( I5,5X,A4,4X,A4,14X,5F7.3 )
6003 FORMAT ( ' How many do you wish to correct? Enter the INTEGER num
     1ber of values which you would like to change ' )
6004 FORMAT ( ' What is the district number at the beginning of the row
     1 which contains the'/ incorrect value? Enter an INTEGER value ')
6005 FORMAT ( ' Please verify that the names for the districts agree if
     1 the values have been'/ entered from a diskette. If you wish to
     2enter another file, answer Y'/ Depress CNTL-BREAK to return to DO
     3S ' )
6006 FORMAT ( ' What is the INTEGER for the sub-season in season ',I2,'
     1 which is incorrect in'/ district ',I2,' Enter an INTEGER value '
     2 )
6007 FORMAT ( ' What should the value be for the crop-water requirement
     1 of this crop'/ in 1000 m3/hectare for the district ',I2,', the s
     2ub-season',I2,' in the season,',I2,'? If you enter a negative nu
     3mber the previous value will remain ' )
6008 FORMAT ( ' Do you wish to save the corrected sub-seasonal crop wat
     1er requirements? Answer Y to save ' )
```

```
6009     FORMAT    ( ' The corrected sub-seasonal crop water requirements will
     1 be saved in the file '/1X,A12/' Enter b:FNAME where b: designates
     2 the drive and FNAME the name of an existing'/' file in 10 or less
     3 letters if you wish to save the data in a different file '/ )
6010     FORMAT    ( ' Do you wish to review the crop water requirements of a
     1particular crop? Enter the INTEGER of the particular crop, or 99
     2 to review the crop water requirements of all the crops ' )
6020     FORMAT    ( ' Would you like to change the values by district, by sea
     1son or one by one?'/' Enter 2 to change the values in a number of
     2districts, by the sub-season'/7X,'1 to change the values in one di
     3strict over the season'/7X,'0 to correct values one by one ' )
6041     FORMAT    ( ' Enter the INTEGER for the sub-season in season ',I2,' w
     2hich contains the values to be corrected ' )
7002     FORMAT    ( ' Do you wish to review any of the data from the files?'/
     1 ' Enter 0 or nothing to review and correct irrigation ret
     2urn flows'/7X,'1 for the review of downstream flow restrictions'/
     3 7X,'2 for the REVIEW procedures and saving of the configuration'
     4/7X,'3 for the entry of another configuration file'
     5/ 7X,'4 for the review of the CROPS data, including the minimum ta
     6rget levels'/9X,'of crops or groups of crops'/7X,'5 for the entry
     7and review of the sub-season crop water requirements'/7X,'6 for th
     8e entry and review of the forecast available water'/ 7X,'7 for the
     9 review of any data that may vary from one sub-season to another'/
     17X,'8 for the revision of the formulation and number of sub-season
     1s'/7x,'9 when all data is correct ' )
7003     FORMAT    ( ' This table shows the percentage of irrigation return fl
     1ows, how much, whence and whither:'/ 4H No.,5X,13HPERCENTAGE OF ,2
     24X,5HFLOWS/ 10X,10HIRRIGATION,8X,14HFROM THE NODE,3X,12HTO THE
N
     3ODE )
7004     FORMAT    ( I5,5X,F8.3,12X,A12,5X,A12 )
7005     FORMAT    ( ' Are these all correct? Answer Y to continue with these
     1 values.'/' Corrected values may be saved on the configuration fil
     2e using the REVIEW'/' procedure save. ' )
7006     FORMAT    ( ' Enter the code for the error:'/ 10X,' 3 - correction of
     1 existing value'/ 10X,' 2 - deletion of an entire line'/ 10X,' 1 -
     2 addition of a node which has return flows to another node'/
     3' Enter an INTEGER. ' )
7007     FORMAT    ( ' How many do you wish to add? Enter an INTEGER value ')
7008     FORMAT    ( ' What is the INTEGER to the left of the additional node
     1, number ',I3,', whose irrigation turnouts will return to ano
     2ther node in the system? ' )
7009     FORMAT    ( ' What is the INTEGER to the left of the node which rece
     1ives the return flow from this node ' )
7010     FORMAT    ( ' What is the percentage of irrigation flows that return?
     1 Enter the percent value. ' )
7011     FORMAT    ( ' How many do you wish to delete? Enter an INTEGER value
     1 ' )
7012     FORMAT    ( ' What is the INTEGER of the line you wish to delete? ' )
7020     FORMAT    ( ' How many do you wish to correct? Enter an INTEGER valu
     1e. ' )
7021     FORMAT    ( ' What do you wish to change?' /' Enter 3 - change the n
```

```
     1ode to which flows return'/ 7X,'2 - change the node of turnout'/
     27X,'1 - change the percentage of flow ')
7022 FORMAT ( ' What is the number of the line you wish to change? Ent
     1er the INTEGER to the left of the line ' )
7023 FORMAT ( ' What is the INTEGER of the node in line ',I3,' of the
     1table whose irrigation turnouts will return to ',A12,1H., F5.
     23,' of the flows are expected to return. ' )
7024 FORMAT ( ' What is the INTERGER of the node in line ',I3,' to whi
     1ch ',F5.3,' of the irrigation turnout of the node ',A12,' does f
     2low? ' )
7025 FORMAT ( ' What is the percentage of the irrigation turnout of the
     1 node ',A12,' whichflows into the node ',A12,' of the ',I3,' lin
     2e of the table? ' )
7100 FORMAT ( ' Please enter the predicted inflows' )
7101 FORMAT ( ' Please enter the crop water requirements' )
7102 FORMAT ( ' Please enter the length of the sub-seasons' )
8000 FORMAT ( ' If the current line in the table has any incorrect e
     1ntries, enter the INTEGERabove one column which is not correct. E
     2nter 0 or nothing before the RETURN to continue to the next node.'
     3/' Enter 99 if you wish to store what you have updated so far.'//
     45X,2H1.,9X,2H2.,5X,2H3.,7X,2H4.,6X,2H5.,3X,2H6.,3X,2H7.,7X,2H8.,5X
     5,2H9.,5X,3H10./4X,3HTHE,8X,4HDOWN,9X,9HDIVERSION,12X,10HIRRIGATION
     6, 7X,9HRESERVOIR/3X,5H NODE,5X,23HSTREAM TO: CAPACITY,3X,
     714HFROM DISTRICT,6X,17HCAP. MAX MIN /14X,5H NODE,13X,
     8 4Hm3/s,10X,9HNo. NAME,4X,4Hm3/s,6X,3HMCM,4X,3HMCM/ )
8001 FORMAT ( ' The node ',A15,' also services the irrigation district
     1 ',I3,',',A4 )
9000 FORMAT ( A12 )
9001 FORMAT ( A2 )
9002 FORMAT ( I5 )
9003 FORMAT ( F10.0 )
9007 FORMAT ( I3,2X,A12,I3,2X,A12,I3,2X,A12,I3,2X,A12,I3,2X,A12 )
9009 FORMAT ( 8I3 )
9010 FORMAT ( I3,F12.0,I3,A4 )
9011 FORMAT ( 26I3 )
9013 FORMAT ( 10F8.0 )
9014 FORMAT ( 6A12 )
9015 FORMAT ( 4(I4,2F8.2) )
9020 FORMAT ( I3,F12.2,I3,A4 )
9022 FORMAT ( 10F8.3 )
9023 FORMAT ( 10F8.2 )
9030 FORMAT ( A20,6F10.0/8F10.0/F10.0 )
9031 FORMAT ( A20,6F10.3/8F10.3/F10.3 )
9060 FORMAT ( A4,15F5.0 )
9061 FORMAT ( A4,15F5.2 )
9104 FORMAT ( 3I2,18A4 )
1100  FORMAT ( 2I3,3F12.2,I3 )
1101  FORMAT ( A9, I3, 3F10.2 )
1102  FORMAT ( 7F8.2 )
1103  FORMAT ( 10F8.2 )
1104  FORMAT ( 18A4 )
1105  FORMAT ( 3I3 )
```

```
1106    FORMAT   ( 10F8.2 )
1107    FORMAT   ( 10I3 )
1108    FORMAT   ( 10A8 )
C
      STOP
      END
C
      SUBROUTINE   CHANJ3   ( JJ, NMAX, NREM, NX )
c  This subroutine assigns or reviews the minimum production targets
      COMMON   /PROD/   YMIN(3,50),   NMINJ(50),   NMINI(50)
      COMMON   /CROP/   NAME(50)
      COMMON   /DELTN/  IX(50)
      CHARACTER   * 8  NAME
      CHARACTER   * 2  ANS
C  If NX is non-zero, ie, deletions were made from the number of crops saved
C  on the diskette in the last run, the YMIN values will also need to be
C  deleted.  This will follow the procedure for deleting in CHANJ1.
        IF ( NX .LE. 0 ) GO TO 10
           JSUB  =  JJ
              L = 1
      DO  108  J=1,JJ
         IF ( J .LT. IX(L) ) GO TO 108
C Determine which crop to substitute in
103        JSUB  =  JSUB + 1
        DO  104  M=1,NX
        IF ( (JSUB) .EQ. IX(M) ) GO TO 103
104      CONTINUE
          DO  105  I=1,3
105          YMIN(I,J)  =  YMIN(I,JSUB)
                  L = L + 1
           IF ( L .GT. NX ) GO TO 10
108     CONTINUE
C  Write and ask for corrections:
10         LTIMES  =  JJ/15 + 1
           NBEG  =  1
        DO  40  L  =  1,LTIMES
           NEND  =  NBEG + 14
           IF ( NEND .GT. JJ ) NEND  =  JJ
20      WRITE  (*,1001)
       WRITE  (*,1002)  ( J, NAME(J),  YMIN(1,J),  YMIN(2,J),  YMIN(3,J),
    1                      J=NBEG,  NEND  )
       WRITE  (*,1003)
       READ  (*,2001)   NX
       IF ( NX .LE. 0 ) GO TO 40
       DO  30  J=1,NX
         WRITE  (*,1004)
         READ  (*,2001)   JWRONG,   I
         WRITE  (*,1000)   NAME(JWRONG),   I
30       READ  (*,2000)   YMIN(I,JWRONG)
      GO  TO  20
40     NBEG  =  NEND + 1
C
```

```
C Find the non-zero elements of YMIN and fill NMINJ(j1) and NMINI(j1),
C records of the crop and season numbers with non-zero min.prod.quotas.
                J1 = 0
        DO 60 I = 1,3
         DO 50 J=1,JJ
            IF ( YMIN(I,J) .LE. 0.0 ) GO TO 50
                J1 = J1 + 1
              NMINJ(J1) = J
              NMINI(J1) = I
50        CONTINUE
60       CONTINUE
C Assign the values of number of cards with three full values, NMAX, and
C the number of non-zero min.prod.quotas left over, NREM, = 1 or 2.
                NMAX = J1/3 * 3
                NREM = J1 - NMAX
C
C Debug of mincrp = 1 functions
C       write (*,900) JJ, j1, nmax, nrem
C       write (*,950) ( nminj(j), nmini(j), nminj(j+1), nmini(j+1), nminj
C     1(j+2),nmini(j+2), j=1,J1,3 )
C900     format ( 4i5)
C950     format ( 6i5 )
c
1000    FORMAT ( ' For the crop', A10,' in season',I3,' from now, what is
     1 the minimum required production? ' )
1001    FORMAT ( ' CROP SEASON 1 SEASON 2 SEASON 3')
1002    FORMAT ( I5,A10,3F12.2 )
1003    FORMAT ( ' These are the minimum required production entered. If
     1you wish to combine some crops under one production quota, you can
     2 do so later (providing you typed 2 or 3 to answer the very first
     3question). Note the crop number, the integer to the left of the c
     4rop name, of the crops you will want to combine.'/' Do you wish to
     5 make corrections to this table? Answer 0 or nothing for none
     6required, or the number of wrong entries: ')
1004    FORMAT ( ' What is the integer to the left of an incorrect entry?
     1 What is the season?'/' Enter a number for the integer, a space, a
     2nd a number for the season: ')
2000    FORMAT ( F9.0 )
2001    FORMAT ( 2I2 )
C
      RETURN
      END
^Z
      SUBROUTINE CONFIG ( NADD, JI, JR, JS, JD, JT, J )
c NADD = initialization code, 1 = starting afresh
c JI = number of irrigation nodes plus 2
c JR = number of reservoir nodes plus 2
c JS = number of nodes not spilling to node numbered one higher plus 2
c JD = number of diversions plus 2
c JT = number of tributaries plus 2
c J = total number of nodes
c
```

```
C    This    subroutine    defines    the    configuration    of    the    system,    the
C    reservoirs,    irrigation    districts,    hydro-electric    plants    etc.
      COMMON/SHC/    JIRR(21),    JRES(25),    JNOSPL(20),    JDIV(2,15)
      COMMON/SHP/    JTRB(2,10),    SLO(25),    SUP(25),    J1(2,10),    C1(10)
      COMMON/CAP/    KIRR(21),CCAPI(21),    CCAPD(15),CRHS(3,35,5),    CLOSS(15)
      COMMON/FLO/    NLIM(21),    NSS(3),    FLOMIN(20),    FLOMAX(20)
      COMMON/NOM/    JNAME(40),    IDNAME(21)
      DIMENSION    KIR(5)
c  KIR   is  a  temporary   vector   of  all  the  irrigation   districts   which   are
c  served   by  one  node
      CHARACTER   * 12  JNAME,   FNAME
      CHARACTER   * 4  IDNAME
      CHARACTER   * 2  ANS
C
      WRITE  ( *,1000  )
      READ   ( *,9001  )  ANS
      IF  ( ANS  .EQ.  'N'  .OR.  ANS  .EQ.  'n'  )  GO  TO  110
C  To  add  (continue  from  80)  or  start  from  scratch:
      IF  ( NADD  .GT.  1 )  GO  TO  80
             JI  =  2
             JR  =  2
             JS  =  2
             JD  =  2
             JT  =  2
              J  =  1
        CCAPD(1)   =  0
        CCAPD(1)   =  0
        CLOSS(1)   =  0
          SLO(1)   =  0
          SUP(1)   =  0
         KIRR(1)   =  0
C  First  of  new  configuration:
      WRITE  ( *,1001  )
      READ   ( *,9000  )  JNAME(J)
C  Common  set  of  entry  questions:
10     WRITE  ( *,2002  )  JNAME(J)
      READ   ( *,9001  )  ANS
      IF  ( ANS  .EQ.  'N'  .OR.  ANS  .EQ.  'n'  )  GO  TO  305
                  JRES(JR)   =  J
         WRITE  ( *,8001  )  JNAME(J)
         READ   ( *,9003  )  SUP(JR)
         WRITE  ( *,8003  )  JNAME(J)
         READ   ( *,9003  )  SLO(JR)
                    JR  =  JR  +  1
305      WRITE  ( *,3015  )  JNAME(J)
        READ   ( *,9001  )  ANS
          IF  ( ANS  .EQ.  'Y'  .OR.  ANS  .EQ.  'y'  )  GO  TO  306
            GO  TO  30
C  Any  limits  to  minimum  or  maximum  downstream  releases
306      WRITE  ( *,3016  )  JNAME(J)
        READ   ( *,9003  )  FLOMIN(NLIM(1))
        WRITE  ( *,3017  )  JNAME(J)
```

```
      READ   ( *,9003   ) FLOMAX(NLIM(1))
      IF  ( FLOMAX(NLIM(1))    .LE.  0  ) FLOMAX(NLIM(1))     =  99999
      NLIM(1)    .=  NLIM(1)    +  1
      NLIM(NLIM(1))     =  J
30    WRITE    ( *,2003    ) JNAME(J)
     READ    ( *,9005    ) KIR(1),   KIR(2),    KIR(3),    KIR(4),    KIR(5)
        IF  ( KIR(1)   .LE.  0  ) GO   TO   40
          KIRR(JI)     =  KIR(1)
          WRITE    ( *,2033    )
          READ    ( *,9003    ) CCAPI(JI)
          WRITE    ( *,2034    ) KIRR(JI)
          READ    ( *,9006    ) IDNAME(JI)
           JIRR(JI)    =  J
        JI  =  JI  +  1
      DO   35   L=2,5
        IF  ( KIR(L)   .LE.  0  ) GO   TO   40
          KIRR(JI)     =  KIR(L)
          WRITE    ( *,2034    ) KIRR(JI)
          READ    ( *,9006    ) IDNAME(JI)
           JIRR(JI)    =  J
35         JI  =  JI  +  1
C  Ask   if  there   is  another    node   downstream    on  the   same   stream.    If
C  there   is  find   out   if  it  has   offstream    sources    and   then   go   through   the
C  loop   again.   If  not   find   out   if  it  flows   into   a  previosly    entered    node
C  or   is  the   end   of  the   configuration.
40    WRITE    ( *,1002    ) JNAME(J)
     READ    ( *,9001    ) ANS
     IF  ( ANS   .EQ.   'N'  .OR.   ANS   .EQ.   'n'  ) GO   TO   55
     WRITE    ( *,2000    ) JNAME(J)
     GO   TO   75
55    WRITE    ( *,1003    ) JNAME(J)
     READ    ( *,9001    ) ANS
          JNOSPL(JS)     =  J
                JS  =  JS  +  1
     IF  ( ANS   .EQ.   'Y'  .OR.   ANS   .EQ.   'y'  ) GO   TO   65
60    WRITE    ( *,1004    )
     READ    ( *,9001    ) ANS
     IF  ( ANS   .EQ.   'N'  .OR.   ANS   .EQ.   'n'  ) GO   TO   100
     GO   TO   70
65    WRITE    ( *,4001    )
     WRITE    ( *,9004    ) ( K,  JNAME(K),    K=1,   (J-1)   )
     WRITE    ( *,3001    ) JNAME(J)
     READ    ( *,9002    ) JTRB(2,JT)
     IF  ( JTRB(2,JT)    .LE.   0  ) GO   TO   60
          JTRB(1,JT)     =  J
                JT  =  JT  +  1
              GO   TO   60
C  The   next   downstream    station   or  the   new   node   on  another    stream:
70    WRITE    ( *,1005    )
75          J  =  J  +  1
     READ    ( *,9000    ) JNAME(J)
     WRITE    ( *,4000    ) JNAME(J)
```

```
      READ ( *,9001 ) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 77
        GO TO 10
77      WRITE ( *,4001 )
        WRITE ( *,9004 ) ( K, JNAME(K), K=1, (J-1) )
        WRITE ( *,4002 ) JNAME(J)
        READ ( *,9002 ) JDIV(1,JD)
        IF ( JDIV(1,JD) .LE. 0 ) GO TO 10
        WRITE ( *,2033 )
        READ ( *,9003 ) CCAPD(JD)
        WRITE ( *,2035 )
        READ ( *,9003 ) CLOSS(JD)
             JDIV(2,JD) = J
                  JD = JD + 1
                GO TO 10
C The nodes are to be added on to an existing configuration
80         JI = JI + 1
           JR = JR + 1
           JS = JS + 1
           JD = JD + 1
           JT = JT + 1
             GO TO 70
C Ready to return to HECTARE for verification
100            JIRR(1) = JI - 2
               JRES(1) = JR - 2
               JNOSPL(1) = JS - 2
             JDIV(1,1) = JD - 2
             JTRB(1,1) = JT - 2
                 JI = JI - 1
                 JR = JR - 1
                 JS = JS - 1
                 JD = JD - 1
                 JT = JT - 1
C
1000  FORMAT ( ' This subroutine enters the configuration of reservoirs,
     1 irrigation schemes, and canals and diversions, or it adds onto
     2an existing configuration previously stored on the diskette.',
     3 ' If you wish to only insert or delete nodes between node
     4s already entered on the same stream or canal or you wish to add o
     5r delete components of previously entered nodes then you can use
     6 the correcting proce-dures of the REVIEW.'/' Do you want to use
     7this subroutine? Answer N to return to the REVIEW '/' procedures
     8.' )
1001  FORMAT ( ' What is the name of the most upstream node on the main
     1 stream? Enter a unique name of 12 or less letters. ' )
1002  FORMAT ( ' Is there a new node downstream of' ,A14,' which you ha
     1ve not yet entered? Answer N if this is the last node on this st
     2ream or canal. ' )
1003  FORMAT ( ' Do the spills, turbine discharges, or return flows of '
     1, A15,' flow to a previously entered node? Answer Y to enter whi
     2ch node. ' )
1004  FORMAT ( ' Do you have more nodes to enter? Answer N to return t
```

```
     1o the REVIEW procedures where you can save what you have entered
     2so far, and return to add more. ' )
1005 FORMAT ( ' What is the name of the most upstream node on another
     1stream? Enter a unique name of 12 or less letters. ' )
2000 FORMAT ( ' What is the name of the next node downstream of ',A15,
     1' Enter a unique name of 12 or less letters. ' )
2002 FORMAT ( ' Is there a reservoir or tank at the node ',A15, '? An
     1swer N if there is none. ' )
2003 FORMAT ( ' What is the INTEGER associated with the irrigation dist
     1rict served by ',A15, '? Enter 0 if this is no turn out for an ir
     2rigation district at this node. If there is more than one crop d
     3istrict served by this node, enter as many as five different INTE
     4GERS associated with these districts, using three spaces for each.
     5 ' )
2033 FORMAT ( ' What is the capacity of the diversion canal in m3/s?
     1 If there is no limit, enter 99999: ' )
2034 FORMAT ( ' What is the name of the irrigation district which is ',
     1I3,'th in the CROPS sequence of districts? Enter a name in 4 or l
     2ess spaces. ' )
2035 FORMAT ( ' What is the amount of flow lost by the canal as a perce
     1nt of flow released '/' upstream? ' )
3001 FORMAT ( ' Enter the INTEGER for the node to which the spills,
     1 turbine discharges, or return flows from ',A15,' naturally flow?
     2 ' )
3015 FORMAT ( ' Are there restrictions on the regulated downstream rele
     1ases at the node '/1H ,A12,', either minimum or maximum levels? A
     2nswer Y to specify the'/' restrictions. ' )
3016 FORMAT ( ' What is the minimum flow level in m3/s above which the
     1 regulated downstream'/' releases of ',A12,' must be? ' )
3017 FORMAT ( ' What is the maximum flow level in m3/s below which the
     1 regulated downstream'/' releases of ',A12,' must be? ' )
4000 FORMAT ( ' Does the node ',A15, ' receive water from OTHER than i
     1ts natural drainage basin - offstream sources, diversions o
     2r PREVIOUSLY entered tributaries? Answer Y if it does. ')
4001 FORMAT ( ' The following lists the nodes previously entered. ' )
4002 FORMAT ( ' Enter the INTEGER to the left of the node from which w
     1ater is diverted, returnsor naturally flows to the node . ',A15, '
     2. If the source node has not yet been entered, enter 0. ' )
8001 FORMAT ( ' What is the maximum capacity of the ',A15,' reservoir?
     1 Enter the'/' storage volume in MCM ' )
8003 FORMAT ( ' What is the minimum operating level of the ',A15,' rese
     1rvoir? Enter the storage volume in MCM ' )
9000 FORMAT ( A12 )
9001 FORMAT ( A2 )
9002 FORMAT ( I5 )
9003 FORMAT ( F10.0 )
9004 FORMAT ( I5, A15, I5, A15, I5, A15, I5, A15 )
9005 FORMAT ( 5I3 )
9006 FORMAT ( A4 )
C
110  CONTINUE
     RETURN
```

```
      END
c
      SUBROUTINE CORRIG ( NX, I, JN, IX, IM, II, IS, IG, IR, IT,
     1                    NAME3, NAME4 )
c  NX = integer of incorrect column
c  I = node number of current line
c  JN = total number of nodes
c  IX = code to skip replacement
c  IM = diversion TO number of current line
c  II = diversion FROM number of current line
c  IS = nospill number of current line
c  IG = irrigation number of current line
c  IR = reservoir number of current line
c  IT = tributary number of current line
c NAME3, NAME4 = column names which may be redefined by CORRIG directly
c
C  This subroutine corrects items in the REVIEW table one item at a time
      COMMON/SHC/ JIRR(21), JRES(25), JNOSPL(20), JDIV(2,15)
      COMMON/SHP/ JTRB(2,10), SLO(25), SUP(25), J1(2,10), C1(10)
      COMMON/CAP/ KIRR(21),CCAPI(21), CCAPD(15),CRHS(3,35,5), CLOSS(15)
      COMMON/FLO/ NLIM(21), NSS(3), FLOMIN(20), FLOMAX(20)
      COMMON/NOM/ JNAME(40), IDNAME(21)
      DIMENSION KIR(5)
      CHARACTER * 12 JNAME, FNAME
      CHARACTER * 4 NAME3, NAME4, IDNAME
      CHARACTER * 2 ANS
C
c  Correct the column identified as incorrect
      GO TO ( 100, 200, 300, 400, 500, 600, 610, 700, 800, 900 ) NX
      GO TO 1001
C  There was an error in the name of the node
100   WRITE (*,1000)
      READ (*,9000) JNAME(I)
          GO TO 1001
C  There was an error in the downstream node
200   WRITE (*,2000) JNAME(I+1)
      READ (*,9002) IX
             GO TO ( 201, 230, 250, 290, 65 ) IX
             GO TO 1001
C  Delete the downstream node
201       CALL DELETE ( I, JN )
c   I = node number of current line
c  JN = total number of nodes
             GO TO 1001
C  Insert downstream nodes
230   CALL ADD (I,JN,IM,II,IS,IG,IR,IT )
c   I = node number of current line
c  JN = total number of nodes
c  IM = diversion TO number of current line
c  II = diversion FROM number of current line
c  IS = nospill number of current line
c  IG = irrigation number of current line
```

```
c   IR  =  reservoir   number   of  current   line
c   IT  =  tributary   number   of  current   line
         GO  TO  1001
C  Remove  this  node  from  the  JNOSPL  vector,  first  checking  that  it  does  not
C  already  flow  to  a  node  on  another  stream.
250    DO  260  K=2,JTRB(1,1)  + 1
         IF  ( JTRB(1,K)  .EQ.  I ) GO  TO  265
260    CONTINUE
       GO  TO  275
265    WRITE  ( *,2003  ) JNAME(I),  JNAME(JTRB(2,K)),  JNAME(I),  JNAME(I+1)
       READ  ( *,9001  ) ANS
       IF ( ANS  .EQ.  'Y' .OR.  ANS  .EQ.  'y' )  GO  TO  200
C   A  tributary  did  exist  at  this  node  and  is  now  deleted
         DO  270  KK=K,JTRB(1,1)
            KPLUS  =  KK  +  1
        JTRB(1,KK)  =  JTRB(1,KPLUS)
270      JTRB(2,KK)  =  JTRB(2,KPLUS)
         JTRB(1,1)  =  JTRB(1,1)  - 1
C   The  node  now  flows  to  the  node  numbered  one  higher
275    DO  280  K=(IS-1),JNOSPL(1)
280       JNOSPL(K)  =  JNOSPL(K+1)
         JNOSPL(1)  =  JNOSPL(1)  - 1
          GO  TO  1001
C  The  node  belongs  in  the  NOSPL  vector  and  is  not,  or  is,  but  has  no
C      trib  node  listed  which  it  should
290    IF  ( JNOSPL(IS-1)  .EQ.  I ) GO  TO  297
         JNOSPL(1)  =  JNOSPL(1)  +  1
       DO  293  KK=2,JNOSPL(1)
         IF  ( JNOSPL(KK)  .GT.  I ) GO  TO  294
293     CONTINUE
294     CALL  LOOPMK(  JNOSPL(1),  1,  KK,  KTOP,  KOLD,  KSTOP  )
c  JNOSPL(1)  =  number  of  nodes
c           1  =  number  of  nodes  to  be  added
c          KK  =  place  in  vector  after  which  a  node  will  be  added
c        KTOP  =  new  final  number  of  nodes  in  the  vector,  a  number  to  be
c                 calculated
c        KOLD  =  old  final  number  of  nodes  in  the  vector,  a  number  to  be
c                 calculated
c       KSTOP  =  the  number  of  times  nodes  in  the  vector  need  to  be  shifted  to
c                 make  space  for  the  new  nodes,  a  number  to  be  calculated
      DO  296  K=1,KSTOP
296         JNOSPL(KTOP-K)  =  JNOSPL(KOLD-K)
             JNOSPL(KK)  =  I
                       IS  =  IS  +  1
C   Check  to  see  if  the  node,  which  does  not  flow  to  a  node  numbered
C   one  higher,  may  still  flow  to  another  node
C   If  it  does  not  flow  to  another  node,  remove  that  tributary
297    WRITE  ( *,1003  ) JNAME(I)
       READ  ( *,9001  ) ANS
       IF  ( ANS  .EQ.  'Y'  .OR.  ANS  .EQ.  'y'  )  GO  TO  64
        DO  298  K1  =  2,JTRB(1,1)+1
          IF  ( JTRB(1,K1)  .EQ.  I ) GO  TO  299
```

```
298     CONTINUE
        GO TO 1001
299     DO 202 KK=K1,JTRB(1,1)
           KPLUS = KK + 1
        JTRB(1,KK) = JTRB(1,KPLUS)
202     JTRB(2,KK) = JTRB(2,KPLUS)
        JTRB(1,1) = JTRB(1,1) - 1
           IT = IT - 1
        GO TO 1001
C The node is a new tributary
64      WRITE ( *,4001 )
        WRITE ( *,9004 ) ( K, JNAME(K), K=1, (JN) )
        WRITE ( *,3001 ) JNAME(I)
        READ ( *,9002 ) JDUMMY
        IF ( JDUMMY .LE. 0 ) GO TO 200
        JTRB(1,1) = JTRB(1,1) + 1
        DO 165 JT = 2,JTRB(1,1)
          IF ( JTRB(1,JT) .GT. I ) GO TO 166
165     CONTINUE
166     CALL LOOPMK ( JTRB(1,1), 1, JT, KTOP, KOLD,KSTOP )
c JTRB(1,1) = number of tributary nodes
c       1 = number of nodes to be added
c       JT = place in vector after which a node will be added
c     KTOP = new final number of nodes in the vector, a number to be
c            calculated
c     KOLD = old final number of nodes in the vector, a number to be
c            calculated
c     KSTOP = the number of times nodes in the vector need to be shifted to
c            make space for the new nodes, a number to be calculated
        DO 167 K=1,KSTOP
          JTRB(1,KTOP-K) = JTRB(1,KOLD-K)
167       JTRB(2,KTOP-K) = JTRB(2,KOLD-K)
            JTRB(1,JT) = I
            JTRB(2,JT) = JDUMMY
               IT = IT + 1
        GO TO 1001
C The node was already defined as a tributary flow but to the wrong
C node
65      WRITE ( *,4001 )
        WRITE ( *,9004 ) ( K, JNAME(K), K=1, (JN) )
        WRITE ( *,3001 ) JNAME(I)
        READ ( *,9002 ) JDUMMY
        IF ( JDUMMY .LE. 0 ) GO TO 200
            JTRB(1,IT-1) = I
            JTRB(2,IT-1) = JDUMMY
        GO TO 1001
C There is a mistake in saying JNAME(I) diverts
C          - it does divert but is not listed
C          - it doesnt divert at all
C          - it diverts to a different node from the one listed
300     WRITE (*,3000) JNAME(I)
        READ (*,9000) FNAME
```

```
      IF ( FNAME .EQ. 'none' .OR. FNAME .EQ. 'NONE' ) GO TO 360
      DO 355 K=1,JN
        IF ( JNAME(K) .EQ. FNAME ) GO TO 358
355   CONTINUE
      WRITE (*,3002) FNAME
      GO TO 300
358   WRITE (*,3003)
      READ (*,9001) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 359
C Diverts in a new diversion
      JDIV(1,JDIV(1,1)+2) = I
      JDIV(2,JDIV(1,1)+2) = K
      WRITE ( *,4000 )
      READ ( *,9003 ) CCAPD(JDIV(1,1)+2)
      JDIV(1,1)=JDIV(1,1) + 1
      NAME3 = JNAME(K)
      GO TO 1001
C Only diverts to a different node
359   JDIV(2,II) = K
          NAME3 = JNAME(K)
       GO TO 1001
C The node does not even divert water to any other node.
360   WRITE ( *,3004 ) JNAME(I)
      READ ( *,9001 ) ANS
          NAME3 = ' '
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 1001
         JDIV(1,1) = JDIV(1,1) - 1
         DO 365 K=II,JDIV(1,1)
             KPLUS = K + 1
          CCAPD(K) = CCAPD(KPLUS)
         JDIV(1,K) = JDIV(1,KPLUS)
365       JDIV(2,K) = JDIV(2,KPLUS)
          NAME3 = '----'
            GO TO 1001
C There was an error in the diversion capacity.
400   WRITE (*,4000)
      READ (*,9003) CCAPD(II)
            GO TO 1001
C There is a mistake in saying JNAME(I) receives offstream water
C       - it does receive but is not listed
C       - it doesnt receive at all
C       - it receives from a different node than the one listed
500   WRITE (*,5000) JNAME(I)
      READ (*,9000) FNAME
      IF ( FNAME .EQ. 'none' .OR. FNAME .EQ. 'NONE' ) GO TO 560
      DO 553 K=1,JN
        IF ( JNAME(K) .EQ. FNAME ) GO TO 555
553   CONTINUE
      WRITE (*,3002) FNAME
      GO TO 500
555   WRITE (*,3003)
      READ (*,9001) ANS
```

```
      IF ( ANS .EQ. 'N' .OR. ANS .EQ. 'n' ) GO TO 558
C A new diversion has been entered
      JDIV(2,JDIV(1,1)+2) = I
      JDIV(1,JDIV(1,1)+2) = K
      WRITE ( *,4000 )
      READ ( *,9003 ) CCAPD(JDIV(1,1)+2)
      JDIV(1,1)=JDIV(1,1) + 1
      NAME4 = JNAME(K)
      GO TO 1001
C Only the node from which water is diverted has changed
558    JDIV(1,IM) = K
      NAME4 = JNAME(K)
       GO TO 1001
C The node does not even divert water to any other node.
560    WRITE ( *,3004 ) JNAME(I)
      READ ( *,9001 ) ANS
      NAME4 = '  '
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 1001
          JDIV(1,1) = JDIV(1,1) - 1
          DO 565 K=IM,JDIV(1,1)
              KPLUS = K + 1
           CCAPD(K) = CCAPD(KPLUS)
          JDIV(1,K) = JDIV(1,KPLUS)
565        JDIV(2,K) = JDIV(2,KPLUS)
      NAME4 = '----'
             GO TO 1001
C There was an error in the irrigation node or file
600    WRITE (*,6000) JNAME(I)
      READ (*,9002) IX
      GO TO ( 625, 650, 675 ) IX
      GO TO 1001
C Ther was an error in the name of the irrigation district
610    WRITE (*,6003) KIRR(IG-1), JNAME(I)
      READ (*,9000) FNAME
        IF ( FNAME .EQ. '      ' ) GO TO 1001
          IDNAME(IG-1) = FNAME
      GO TO 1001
C The INTEGER of the district needs changing
625    WRITE (*,6001) JNAME(I)
      READ (*,9005) ( KIR(N), N=1,5 )
      IF ( KIR(1) .LE. 0 ) GO TO 650
        KIRR(IG-1) = KIR(1)
        WRITE (*,6002) KIRR(IG-1)
        READ (*,9006) IDNAME(IG-1)
        DO 630 L=2,5
          IF ( KIR(L) .LE. 0 ) GO TO 1001
C Open up KIRR, JIRR, IDNAME and CCAPI vectors
            CALL LOOPMK ( JIRR(1), 1, IG, KTOP, KOLD, KSTOP )
c    JIRR(1) = number of irrigation nodes
c          1 = number of nodes to be added
c         IG = place in vector after which a node will be added
c       KTOP = new final number of nodes in the vector, a number to be
```

```
c                  calculated
c         KOLD    =  old  final  number   of  nodes   in  the  vector,   a  number   to  be
c                  calculated
c        KSTOP    =  the  number  of  times  nodes  in  the  vector  need  to  be  shifted  to
c                  make  space  for  the  new  nodes,  a  number  to  be  calculated
               DO   627   K=1,KSTOP
                 JIRR(KTOP-K)        =  JIRR(KOLD-K)
                 KIRR(KTOP-K)        =  KIRR(KOLD-K)
               IDNAME(KTOP-K)        =  IDNAME(KOLD-K)
627              CCAPI(KTOP-K)       =  CCAPI(KOLD-K)
               KIRR(IG)     =  KIR(L)
               JIRR(IG)     =  I
               WRITE   (*,6002)    KIRR(IG)
               READ    (*,9006)    IDNAME(IG)
630                  IG  =  IG  +  1
           GO   TO   1001
C  There  really  is  no  irrigation  at  this  node,  delete  from  JIRR  and  KIRR
C      and  IDNAME   and  CCAPI
650                 IG  =  IG  -  1
           DO   665   K=(IG),JIRR(1)+1
              ... KPLUS    =  K  +  1
              CCAPI(K)     =  CCAPI(KPLUS)
             IDNAME(K)     =  IDNAME(KPLUS)
               KIRR(K)     =  KIRR(KPLUS)
665             JIRR(K)    =  JIRR(KPLUS)
               JIRR(1)     =  JIRR(1)    -  1
           IF  (  JIRR(IG-1)     .EQ.   I  )  GO   TO   650
                 GO   TO   1001
C  There  really  IS  irrigation  at  this  node
675          JIRR(1)     =  JIRR(1)    +  1
         CALL    LOOPMK(    JIRR(1),   1,  (IG-1),   KTOP,   KOLD,   KSTOP    )
c     JIRR(1)    =  number   of  irrigation   nodes
c            1  =  number   of  nodes   to  be  added
c          IG-1   =  place   in  vector   after   which   a  node   will  be  added
c         KTOP    =  new   final  number   of  nodes   in  the  vector,   a  number   to  be
c                  calculated
c         KOLD    =  old  final  number   of  nodes   in  the  vector,   a  number   to  be
c                  calculated
c        KSTOP    =  the  number  of  times  nodes  in  the  vector  need  to  be  shifted  to
c                  make  space  for  the  added  node,  a  number  to  be  calculated
         DO   676   K=1,KSTOP
            KIRR(KTOP-K)        =  KIRR(KOLD-K)
           CCAPI(KTOP-K)        =  CCAPI(KOLD-K)
          IDNAME(KTOP-K)        =  IDNAME(KOLD-K)
676          JIRR(KTOP-K)       =  JIRR(KOLD-K)
                JIRR(IG)     =  I
         WRITE    (  *,6001    )  JNAME(I)
         READ     (  *,9005    )  (  KIR(N),    N=1,5    )
              IF   (  KIR(1)    .LE.   0  )  GO   TO   1001
                KIRR(IG)     =  KIR(1)
                WRITE    (  *,4000    )
                READ     (  *,9003    )  CCAPI(IG)
```

```
      WRITE (*,6002) KIRR(IG)
      READ (*,9006) IDNAME(IG)
      IG = IG + 1
      DO 39 L=2,5
      IF ( KIR(L) .LE. 0 ) GO TO 1001
C Open up KIRR, JIRR, and CCAPI vectors
      CALL LOOPMK ( JIRR(1), 1, IG, KTOP, KOLD, KSTOP )
c JIRR(1) = number of irrigation nodes
c   1 = number of nodes to be added
c   IG = place in vector after which a node will be added
c   KTOP = new final number of nodes in the vector, a number to be
c          calculated
c   KOLD = old final number of nodes in the vector, a number to be
c          calculated
c   KSTOP = the number of times nodes in the vector need to be shifted to
c          make space for the added node, a number to be calculated
      DO 38 K=1,KSTOP
      JIRR(KTOP-K) = JIRR(KOLD-K)
      KIRR(KTOP-K) = KIRR(KOLD-K)
      IDNAME(KTOP-K) = IDNAME(KOLD-K)
38    CCAPI(KTOP-K) = CCAPI(KOLD-K)
      KIRR(IG) = KIR(L)
      JIRR(IG) = I
      CCAPI(IG) = 0
      WRITE (*,6002) KIRR(IG)
      READ (*,9006) IDNAME(IG)
39    IG = IG + 1
      GO TO 1001
C There was an error in the irrigation capacity
700   WRITE (*,4000)
      READ (*,9003) CCAPI(IG-1)
      GO TO 1001
C There was an error in the reservoir capacities at this node
800   WRITE (*,8000) JNAME(I)
      READ (*,9002) IX
      GO TO ( 825, 850, 875 ) IX
      GO TO 1001
C There was an error in the capacity
825   WRITE ( *,8001 ) JNAME(I)
      READ ( *,9003 ) SUP(IR-1)
      GO TO 1001
C There really is no reservoir at this node, delete from JRES and EVAP
C SLO and SUP
850   DO 865 K=(IR-1),JRES(1)+1
      KPLUS = K + 1
      SUP(K) = SUP(KPLUS)
      SLO(K) = SLO(KPLUS)
865   JRES(K) = JRES(KPLUS)
      JRES(1) = JRES(1) - 1
      IR = IR - 1
      GO TO 1001
C There really IS a reservoir at this node
```

```
875     JRES(1)     =  JRES(1)     +  1
        CALL   LOOPMK(    JRES(1),    1,  (IR-1),   KTOP,   KOLD,   KSTOP    )
c     JRES(1)     =  number    of  storage    nodes
c          1  =  number     of  nodes    to   be   added
c        IR-1    =   place    in   vector    after    which   a  node    will   be   added
c        KTOP     =  new    final   number     of  nodes    in   the    vector,   a  number     to   be
c                calculated
c        KOLD     =  old   final    number     of  nodes     in   the    vector,    a  number     to   be
c                calculated
c       KSTOP     =  the    number     of  times     nodes     in   the    vector     need    to   be   shifted    to
c                make     space    for   the    added     node,    a  number     to   be   calculated
       DO    876    K=1,KSTOP
876          JRES(KTOP-K)          =  JRES(KOLD-K)
               JRES(IR)       =   I
       WRITE    (  *,8001     )  JNAME(I)
       READ     (  *,9003     )  SUP(IR)
       WRITE    (  *,8003     )  JNAME(I)
       READ     (  *,9003     )  SLO(IR)
          IR   =   IR   +   1
        GO    TO    1001
C   The    minimum      storage      volume      of   the    reservoir      is   wrong
900     WRITE     (*,8003)      JNAME(I)
       READ      (*,9003)      SLO(IR-1)
           GO    TO    1001
C
1000     FORMAT       ( '  What     is   the    correct      name      for    the    last     node?      Enter      a  n
        1ame     of   12    or    less     letters      ' )
1003     FORMAT       ( '  Do     the    spills,     turbine      discharges,       or   return     flows     of   '
        1,  A15,'     flow    to   a   previously       entered       node?      Answer      Y   if  the     node
        2  is   the    last    node     on   a'/    tributary.       ' )
2000     FORMAT       ( '  Enter     the    code     for   the    correction       desired:      '/  '  1  -  del
        1ete    the    downstream        node'/     '  2  -  insert    one    or   more     nodes      between
        2  this     node     and     the    next     previously       entered      downstream         node'/     '  3
        3  -  the    flows     from    this     node    DO    flow    to   the      node',A15        /  '  4  -  t
        4his     node     is   the    last    on   a   stream.       You    may    also    indicate      any    node
        5to    which     its   flows     naturally       spill'/    '  5  -  the    flows     from    this     no
        6de    flow    not    to   the     node    listed     but    to   another        node'/     0   or   nothi
        7ng     will    return     to   the    REVIEW      table.     ')
2003     FORMAT       ( '  There     is   a  tributary      flowing      from    ',A15,'     to   ',A15,'     .
        1  Answer      Y   to   keep     the    tributary      flow    and     return     to   the    last    menu
        2  Otherwise        ',A15,'     will    spill    to   ',A15     )
3000     FORMAT       ( '  What     is   the    name     of   the    node     to   which      ',A15,'     does     div
        1ert    water?      If  it  does     not    actually      divert     water,     nor    enter     anothe
        2r   stream,      or   the    node     to   which     it  diverts     water     has    not    yet    been
        3entered,        Answer       NONE.      ' )
3001     FORMAT       ( '  Enter     the    INTEGER       for   the    node     to    which     the    spills,
        1  turbine        discharges,        or   return     flows     from     ',A15,'     naturally       flow.
        2'/    If  you    enter     0,   you    will    return     to   the    last    menu      where     you    may
        3  enter     3  to   delete     this     node     from     the    list   of   last     nodes      on   a  tr
        4ibutary       or   0  to   continue.       ' )
3002     FORMAT       ( '  No     node     of   the    name     ',A15,     '   was     found.       Is  it  correct
        2ly    spelt     in   full?    ')
```

```
3003   FORMAT  ( ' Was a diversion listed in the table? Answer Y to redir
      3ect the node of diversion: ' )
3004   FORMAT  ( ' Answer Y if there is a diversion at ',A15,' otherwise
      1the diversion at this node will be deleted! ' )
4000   FORMAT  ( ' Enter the correct value for the channel capacity in m3/
      1s'/ ( 99999 is for unlimited capacity ) ' )
4001   FORMAT  ( ' The following lists the nodes previously entered. ' )
5000   FORMAT  ( ' What is the name of the node which diverts water to
      1',A15, ' ? If the node which diverts to it has not yet be entere
      3d, Answer NONE. ' )
6000   FORMAT  ( ' Enter the code for the correction desired: '/ ' 1 - cha
      1nge the INTEGER associated with the irrigation district'/ ' 2 - de
      2lete ',A15, ' from the list of irrigation nodes '/ ' 3 - add this
      3 node to the list of irrigation nodes'/ 0 or nothing will return
      4to the next line of the REVIEW table. ' )
6001   FORMAT  ( ' What is the INTEGER associated with the irrigation dist
      1rict served by ',A15, '? Enter 0 if this is no turn out for an ir
      2rigation district at this node. If there is more than one crop d
      3istrict served by this node, enter as many as five different INTE
      4GERS associated with these districts, using three spaces for each.
      5 ' )
6002   FORMAT  ( ' What is the name of the irrigation district which is ',
      1I3,'th in the CROPS sequence of districts? Enter a name in 4 or l
      2ess spaces. ' )
6003   FORMAT  ( ' Enter the correct name in 4 or less spaces for the irri
      1gation district which is ',I3,'th in the Crops sequence of distric
      2ts and is serviced by the node ',A12,' Enter 0 or nothing to retur
      3n to the last line where you may add, or delete,'/ irrigation dis
      4tricts from the configuration by correcting column 6 ' )
8000   FORMAT  ( ' Enter the code for the correction desired: '/ ' 1 - cha
      1nge the maximum capacity of the reservoir'/ 2 - delete ',A15,' fr
      2om the list of nodes with reservoirs'/ 3 - add this node to the l
      3ist of nodes with reservoirs'/ 0 or nothing will return to the n
      4ext line of the REVIEW table. ' )
8001   FORMAT  ( ' What is the maximum capacity of the ',A15,' reservoir?
      1 Enter the'/ storage volume in MCM ' )
8003   FORMAT  ( ' What is the minimum operating level of the ',A15,' rese
      1rvoir? Enter the storage volume in MCM ' )
9000   FORMAT  ( A12 )
9001   FORMAT  ( A2 )
9002   FORMAT  ( I5 )
9003   FORMAT  ( F10.0 )
9004   FORMAT  ( I5,A15,I5,A15,I5,A15,I5,A15 )
9005   FORMAT  ( 5I3 )
9006   FORMAT  ( A4 )
C
1001    CONTINUE
       RETURN
       END
C
      SUBROUTINE LOOPMK ( J, JP, K, KT, KO, KS )
c   J = number of nodes in the vector where some nodes are to be added
```

```
c JP = number of nodes to be added
c K = place in vector after which a node will be added
c KT = new final number of nodes in the vector, a number to be
c calculated
  KO = old final number of nodes in the vector, a number to be
c calculated
c KS = the number of times nodes in the vector need to be shifted to
c make space for the added nodes, a number to be calculated
c
c This function subroutine defines three variables used to open up
c vectors so that new nodes will be entered in ascending order in the
c vector
        KT = J + 2
        KO = KT - JP
        KS = KO - K
    RETURN
    END
c
    SUBROUTINE DELETE (I, JN)
c I = node number of the current line
c JN = total number of nodes
c
c This subroutines deletes the node(s) downstream of the node listed in
c the current line of the REVIEW table
    COMMON/SHC/ JIRR(21), JRES(25), JNOSPL(20), JDIV(2,15)
    COMMON/SHP/ JTRB(2,10), SLO(25), SUP(25), J1(2,10), C1(10)
    COMMON/CAP/ KIRR(21),CCAPI(21), CCAPD(15),CRHS(3,35,5), CLOSS(15)
    COMMON/FLO/ NLIM(21), NSS(3), FLOMIN(20), FLOMAX(20)
    COMMON/NOM/ JNAME(40), IDNAME(21)
    CHARACTER * 12 JNAME, FNAME
    CHARACTER * 4 IDNAME
    CHARACTER * 2 ANS
C
        IP1 = I + 1
c Determine if the downstream node has flow restrictions
    DO 203 K=2,NLIM(1)
      IF ( NLIM(K) .EQ. IP1 ) GO TO 205
203 CONTINUE
      GO TO 207
c It does, remove from those vectors
205 DO 206 KK=K,NLIM(1)
      FLOMIN(KK) = FLOMIN(KK+1)
      FLOMAX(KK) = FLOMAX(KK+1)
206     NLIM(KK) = NLIM(KK+1)
        NLIM(1) = NLIM(1) - 1
c Determine if the downstream node has irrigation deliveries
207 DO 208 K=2,JIRR(1)
      IF ( JIRR(K) .EQ. (IP1) ) GO TO 310
208 CONTINUE
      GO TO 313
c It does, remove from those vectors
310 DO 312 KK=K,JIRR(1)
```

```
          KIRR(KK)    = KIRR(KK+1)
          CCAPI(KK)   = CCAPI(KK+1)
          IDNAME(KK)  = IDNAME(KK+1)
312       JIRR(KK)    = JIRR(KK+1)
          JIRR(1)     = JIRR(1) - 1
c Determine if the downstream node has storage facilities
313   DO 314 K=2,JRES(1)
        IF ( JRES(K) .EQ. (IP1) ) GO TO 316
314   CONTINUE
        GO TO 319
c It does, remove from those vectors
316     DO 318 KK=K,JRES(1)
          SLO(KK)   = SLO(KK+1)
          SUP(KK)   = SUP(KK+1)
318       JRES(KK)  = JRES(KK+1)
          JRES(1)   = JRES(1) - 1
c Determine if the downstream node does not flow to the node numbered
c  one higher
319   DO 320 K=2,JNOSPL(1)
        IF ( JNOSPL(K) .EQ. IP1 ) GO TO 322
320   CONTINUE
        GO TO 325
c It does, remove from that vector
322     DO 324 KK=K,JNOSPL(1)
324       JNOSPL(KK)  = JNOSPL(KK+1)
          JNOSPL(1)   = JNOSPL(1) - 1
c Determine if the downstream flow is a tributary to another node
325   DO 326 K=2,JTRB(1,1)
        IF ( JTRB(1,K) .EQ. IP1 ) GO TO 328
        IF ( JTRB(2,K) .EQ. IP1 ) GO TO 338
326   CONTINUE
        GO TO 331
C There is natural flows from the node to be deleted to another stream.
C What is to happen to the flows of the tributary.
328   WRITE ( *,1030 ) JNAME(IP1), JNAME(JTRB(2,K))
      READ ( *,9001 ) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 329
      WRITE ( *,1031 ) JNAME(JTRB(2,K))
      READ ( *,9000 ) FNAME
      DO 106 KK=1,JN
        IF ( JNAME(KK) .EQ. FNAME ) GO TO 108
106   CONTINUE
      WRITE (*,1002) FNAME
      GO TO 328
108   JTRB(1,K) = KK
      GO TO 331
C The TRIBUTARY is to be entirely deleted
329   DO 105 KK = K,JTRB(1,1)
          KPLUS = KK + 1
        JTRB(1,KK)  = JTRB(1,KPLUS)
105     JTRB(2,KK)  = JTRB(2,KPLUS)
        JTRB(1,1)   = JTRB(1,1) - 1
```

```
      GO TO 331
C There is a TRIBUTARY flowing to the node to be deleted
338   WRITE ( *,2030 ) JNAME(IP1), JNAME(JTRB(1,K))
      READ ( *,9001 ) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 329
      WRITE ( *,2031 ) JNAME(JTRB(1,K))
      READ ( *,9000 ) FNAME
      DO 336 KK=1,JN
        IF ( JNAME(KK) .EQ. FNAME ) GO TO 308
336   CONTINUE
      WRITE (*,1002) FNAME
      GO TO 338
308   JTRB(2,K) = KK
c Determine if there is any diversions to or from the downstream node
331   DO 335 K=2,JDIV(1,1)
        IF ( JDIV(1,K) .EQ. IP1 ) GO TO 340
        IF ( JDIV(2,K) .EQ. IP1 ) GO TO 350
335   CONTINUE
      GO TO 400
C There is a diversion from the node to be deleted. Is the
C diversion node served from another node or is the diversion
C also to be deleted.
340   WRITE ( *,1000 ) JNAME(IP1), JNAME(JDIV(2,K))
      READ ( *,9001 ) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 345
      WRITE ( *,1001 ) JNAME(JDIV(2,K))
      READ ( *,9000 ) FNAME
      DO 146 KK=1,JN
        IF ( JNAME(KK) .EQ. FNAME ) GO TO 118
146   CONTINUE
      WRITE (*,1002) FNAME
      GO TO 340
118   JDIV(1,K) = KK
      WRITE ( *,1003 ) JNAME(JDIV(1,K)), JNAME(JDIV(2,K))
      READ ( *,9003 ) CCAPD(K)
      WRITE ( *,3035 )
      READ ( *,9003 ) CLOSS(K)
      GO TO 400
C The diversion is to be entirely deleted
345   DO 165 KK = K,JDIV(1,1)
              KPLUS = KK + 1
          CCAPD(KK) = CCAPD(KPLUS)
          CLOSS(KK) = CLOSS(KPLUS)
        JDIV(1,KK) = JDIV(1,KPLUS)
165       JDIV(2,KK) = JDIV(2,KPLUS)
          JDIV(1,1) = JDIV(1,1) - 1
          GO TO 400
C There is a diversion to the node to be deleted
350   WRITE ( *,2000 ) JNAME(IP1), JNAME(JDIV(1,K))
      READ ( *,9001 ) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 345
      WRITE ( *,2001 ) JNAME(JDIV(1,K))
```

```
      READ   ( *,9000   ) FNAME
      DO  246   KK=1,JN
        IF  ( JNAME(KK)    .EQ.   FNAME    ) GO  TO  218
246   CONTINUE
      WRITE   (*,1002)    FNAME
      GO  TO  350
218   JDIV(2,K)    =  KK
      WRITE   ( *,1003   ) JNAME(JDIV(1,K)),        JNAME(JDIV(2,K))
      READ    ( *,9003   ) CCAPD(K)
      WRITE   ( *,3035   )
      READ    ( *,9003   ) CLOSS(K)
c   Decrease    the  total   number   of  nodes
400   JN  =  JN  - 1
      DO  405   KK  =  IP1,   JN
405   JNAME(KK)     =  JNAME(KK+1)
C   Now  all  the  vectors   have  the  I+1   node   removed.     Next  all  nodes
C   greater   than  I+1  will  be  renumbered    one  less.
      DO  410   K=2,   JIRR(1)
        IF  ( JIRR(K)    .GT.   I ) JIRR(K)    =  JIRR(K)    - 1
410   CONTINUE
      DO  415   K=2,   NLIM(1)
        IF  ( NLIM(K)    .GT.   I ) NLIM(K)    =  NLIM(K)    - 1
415   CONTINUE
      DO  430   K=2,(JRES(1)      +  1)
        IF  ( JRES(K)    .GT.   I ) JRES(K)    =  JRES(K)    - 1
430   CONTINUE
      DO  440   K=2,(JNOSPL(1)      +  1)
        IF  ( JNOSPL(K)    .GT.   I ) JNOSPL(K)    =  JNOSPL(K)    - 1
440   CONTINUE
      DO  450   K=2,(JDIV(1,1)      +  1)
        IF  ( JDIV(1,K)    .GT.   I ) JDIV(1,K)    =  JDIV(1,K)    - 1
        IF  ( JDIV(2,K)    .GT.   I ) JDIV(2,K)    =  JDIV(2,K)    - 1
450   CONTINUE
      DO  460   K=2,(JTRB(1,1)      +  1)
        IF  ( JTRB(1,K)    .GT.   I ) JTRB(1,K)    =  JTRB(1,K)    - 1
        IF  ( JTRB(2,K)    .GT.   I ) JTRB(2,K)    =  JTRB(2,K)    - 1
460   CONTINUE
C
1000  FORMAT   ( ' There  is  a  diversion   from  the  node  to  be  deleted   ',A15
     1,'  to  the  node  ',A15,'.  Is  the  second   node  served  from  another
     2node  previously   enteredor   is  the  diversion   to  be  deleted   completel
     3y?  Answer   Y  to  destroy   the  diversionto    the  second   node.  ')
1001  FORMAT   ( ' What  is  the  name  of  the  previously   entered   node  which
     1now  diverts  to  ',A15,  '  Enter  a  name  of  12  or  less  letters.  ' )
1002  FORMAT   ( ' No  node  of  the  name  ',A15,  ' was  found.   Is  it  correct
     2ly  spelt?  ')
1003  FORMAT   ( ' What  is  the  capacity   of  the  diversion   canal  in  m3/s  fro
     1m  the  node  ',A15,'  to  the  node  ',A15,1H   )
2000  FORMAT   ( ' There  is  a  diversion   to  the  node  to  be  deleted   ',A15,'
     2 from  the  node  ',A15,'.  Does  the  second   node  divert  water  to  an
     3other  previously   entered   node  or  is  the  diversion   to  be  deleted   c
     4ompletely?    Answer   Y  to  destroy   the  diversion   from  the  second   nod
```

```
      5e.  ' )
2001  FORMAT  ( ' What is the name of the previously entered node which
     1now receives water from ',A15, ' Enter a name of 12 or less letter
     2s.  ' )
1030  FORMAT  ( ' The node to be deleted ',A15,' naturally flows to the
     1 node ',A15,'. Does the second node receive flows from another
     2node previously enteredor is the tributary to be deleted completel
     3y? Answer Y to destroy the tributaryto the second node. ')
1031  FORMAT  ( ' What is the name of the previously entered node which
     1now flows directly to ',A15, ' Enter a name of 12 or less letters
     2.  ' )
2030  FORMAT  ( ' There is a tributary to the node to be deleted ',A15,'
     2 from the node ',A15,'. Does water from the second node flow to
     3 another previously entered node or is the tributary to be del
     4eted completely? Answer Y to destroy the tributary from the secon
     5d node.  ' )
2031  FORMAT  ( ' What is the name of the previously entered node which
     1now receives water from ',A15, ' Enter a name of 12 or less letter
     2s.  ' )
3035  FORMAT  ( ' What is the amount of flow lost by the canal as a perce
     1nt of flow released '/' upstream?  ' )
9000  FORMAT  ( A12  )
9001  FORMAT  ( A2  )
9003  FORMAT  ( F10.0  )
      RETURN
      END
c
      SUBROUTINE ADD (I,JN,IM,II,IS,IG,IR,IT  )
c  I = node number of current line
c  JN = total number of nodes
c  IM = diversion TO number of current line
c  II = diversion FROM number of current line
c  IS = nospill number of current line
c  IG = irrigation number of current line
c  IR = reservoir number of current line
c  IT = tributary number of current line
c
C  This subroutine inserts downstream nodes after the current node and
c  before previously downstream nodes
      COMMON/SHC/  JIRR(21), JRES(25), JNOSPL(20), JDIV(2,15)
      COMMON/SHP/  JTRB(2,10), SLO(25), SUP(25), J1(2,10), C1(10)
      COMMON/CAP/  KIRR(21),CCAPI(21), CCAPD(15),CRHS(3,35,5), CLOSS(15)
      COMMON/FLO/  NLIM(21),. NSS(3), FLOMIN(20), FLOMAX(20)
      COMMON/NOM/  JNAME(40), IDNAME(21)
      DIMENSION  KIR(5)
      CHARACTER  * 12 JNAME, FNAME
      CHARACTER  * 4 IDNAME
      CHARACTER  * 2 ANS
C
330   WRITE (*,3001) JNAME(I), JNAME(I+1)
      READ (*,9002) JPLUS
          JN = JN + JPLUS
```

```
C First all nodes will be renumbered in all the vectors
      DO 331 K=2, (JIRR(1) + 1)
       IF ( JIRR(K) .GT. I ) JIRR(K) = JIRR(K) + JPLUS
331    CONTINUE
      DO 332 K=2, (JTRB(1,1) + 1)
       IF ( JTRB(1,K) .GE. I ) JTRB(1,K) = JTRB(1,K) + JPLUS
       IF ( JTRB(2,K) .GE. I ) JTRB(2,K) = JTRB(2,K) + JPLUS
332    CONTINUE
      DO 333 K=2,(JRES(1) + 1)
       IF ( JRES(K) .GT. I ) JRES(K) = JRES(K) + JPLUS
333    CONTINUE
      DO 334 K=2,(JNOSPL(1) + 1)
       IF ( JNOSPL(K) .GT. I ) JNOSPL(K) = JNOSPL(K) + JPLUS
334    CONTINUE
      DO 335 K=2,(JDIV(1,1) + 1)
       IF ( JDIV(1,K) .GT. I ) JDIV(1,K) = JDIV(1,K) + JPLUS
       IF ( JDIV(2,K) .GT. I ) JDIV(2,K) = JDIV(2,K) + JPLUS
335    CONTINUE
      DO 336 K=2,(NLIM(1) + 1)
       IF ( NLIM(K) .GT. I ) NLIM(K) = NLIM(K) + NPLUS
336    CONTINUE
        CALL LOOPMK( (JN-1), JPLUS, (I+1), KTOP, KOLD, KSTOP )
c       JN-1 = number of nodes
c      JPLUS = number of nodes to be added
c        I+1 = place in vector after which a node will be added
c       KTOP = new final number of nodes in the vector, a number to be
c              calculated
c       KOLD = old final number of nodes in the vector, a number to be
c              calculated
c      KSTOP = the number of times nodes in the vector need to be shifted to
c              make space for the new nodes, a number to be calculated
        DO 337 K=1,KSTOP
337          JNAME(KTOP-K) = JNAME(KOLD-K)
C If the upstream node was at the end of the stream, the JNOSPL vector
C will contain the latest new downstream node
                 ICOMPR = I + JPLUS
                   ISMN = IS - 1
      IF ( JNOSPL(ISMN) .EQ. ICOMPR ) JNOSPL(ISMN) = ICOMPR + JPLUS
C
C Second the various components for each new node will be entered
      DO 500 N=1,JPLUS
C The names of the nodes
340              I = I + 1
        WRITE ( *,3010 ) JNAME(I-1)
        READ ( *,9000 ) JNAME(I)
        WRITE ( *,3015 ) JNAME(I)
        READ ( *,9001 ) ANS
         IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 3405
           GO TO 3406
C Any limits to minimum or maximum downstream releases
3405     WRITE ( *,3016 ) JNAME(I)
        READ ( *,9003 ) FLOMIN(NLIM(1))
```

```
      WRITE ( *,3017 ) JNAME(I)
      READ ( *,9003 ) FLOMAX(NLIM(1))
      IF ( FLOMAX(NLIM(1)) .LE. 0 ) FLOMAX(NLIM(1)) = 99999
       NLIM(1) = NLIM(1) + 1
       NLIM(NLIM(1)) = I
C Any diversions
3406  WRITE ( *,4000 ) JNAME(I)
      READ ( *,9001 ) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 341
        GO TO 20
341   WRITE ( *,4001 )
      WRITE ( *,9004 ) ( K, JNAME(K), K=1, (JN-1) )
      WRITE ( *,4002 ) JNAME(I)
      READ ( *,9002 ) JDUMMY
      IF ( JDUMMY .LE. 0 ) GO TO 20
          JDIV(1,1) = JDIV(1,1) + 1
          JDIV(1,(JDIV(1,1)+1)) = JDUMMY
          JDIV(2,(JDIV(1,1)+1)) = I
          WRITE ( *,3032 )
          READ ( *,9003 ) CCAPD((JDIV(1,1)+1))
          WRITE ( *,3035 )
          READ ( *,9003 ) CLOSS((JDIV(1,1)+1))
C The reservoir file if any
20      WRITE ( *,2002 ) JNAME(I)
        READ ( *,9001 ) ANS
        IF ( ANS .EQ. 'N' .OR. ANS .EQ. 'n' ) GO TO 30
       JRES(1) = JRES(1) + 1
       DO 353 KK=2,JRES(1)
        IF ( JRES(KK) .GT. I ) GO TO 355
353     CONTINUE
355     CALL LOOPMK( JRES(1), 1, KK, KTOP, KOLD, KSTOP )
c   JRES(1) = number of nodes
c        1 = number of nodes to be added
c       KK = place in vector after which a node will be added
c     KTOP = new final number of nodes in the vector, a number to be
c            calculated
c     KOLD = old final number of nodes in the vector, a number to be
c            calculated
c    KSTOP = the number of times nodes in the vector need to be shifted to
c            make space for the new nodes, a number to be calculated
       DO 356 K=1,KSTOP
           SLO(KTOP-K) = SLO(KOLD-K)
           SUP(KTOP-K) = SUP(KOLD-K)
356         JRES(KTOP-K) = JRES(KOLD-K)
     WRITE ( *,8001 ) JNAME(I)
     READ ( *,9003 ) SUP(KK)
     WRITE ( *,8003 ) JNAME(I)
     READ ( *,9003 ) SLO(KK)
             JRES(KK) = I
                 IR = IR + 1
C Any irrigation district
30    WRITE ( *,2003 ) JNAME(I)
```

```
      READ    ( *,9005    ) KIR(1),    KIR(2),    KIR(3),    KIR(4),    KIR(5)
        IF  ( KIR(1)    .LE.   0 ) GO   TO   500
          JIRR(1)     =  JIRR(1)     +  1
          DO   363    KK=2,JIRR(1)
            IF  ( JIRR(KK)      .GT.   I ) GO   TO   364
363         CONTINUE
364         CALL    LOOPMK(      JIRR(1),      1,  KK,   KTOP,     KOLD,     KSTOP      )
c     JIRR(1)      =   number      of   nodes
c           1  =  number     of   nodes    to   be    added
c          KK    =  place     in   vector     after     which     a   node     will   be    added
c        KTOP     =  new     final    number      of   nodes     in   the    vector,     a   number      to   be
c                 calculated
c        KOLD     =  old    final    number      of   nodes     in   the    vector,     a   number      to   be
c                 calculated
c       KSTOP     =  the    number     of   times    nodes     in   the    vector     need    to   be   shifted     to
c                 make    space     for   the    new    nodes,     a   number      to   be   calculated
      DO   366   K=1,KSTOP
        KIRR(KTOP-K)           =  KIRR(KOLD-K)
       CCAPI(KTOP-K)           =  CCAPI(KOLD-K)
      IDNAME(KTOP-K)           =  IDNAME(KOLD-K)
366      JIRR(KTOP-K)          =  JIRR(KOLD-K)
           JIRR(KK)       =  I
           KIRR(KK)       =  KIR(1)
         WRITE     ( *,3032     )
         READ      ( *,9003     ) CCAPI(KK)
         WRITE     ( *,2034     ) KIRR(KK)
         READ      ( *,9006     ) IDNAME(KK)
                   IG  =  IG  +  1
      DO   39   L=2,5
        IF  ( KIR(L)     .LE.   0 ) GO   TO   500
          JIRR(1)     =  JIRR(1)     +  1
          DO   393    KK=2,JIRR(1)
            IF  ( JIRR(KK)      .GT.   I ) GO   TO   394
393         CONTINUE
394         CALL    LOOPMK(      JIRR(1),      1,  KK,   KTOP,     KOLD,     KSTOP      )
c     JIRR(1)      =   number      of   nodes
c           1  =  number     of   nodes    to   be    added
c          KK    =  place     in   vector     after     which     a   node     will   be    added
c        KTOP     =  new     final    number      of   nodes     in   the    vector,     a   number      to   be
c                 calculated
c        KOLD     =  old    final    number      of   nodes     in   the    vector,     a   number      to   be
c                 calculated
c       KSTOP     =  the    number     of   times    nodes     in   the    vector     need    to   be   shifted     to
c                 make    space     for   the    new    nodes,     a   number      to   be   calculated
      DO   396   K=1,KSTOP
        KIRR(KTOP-K)           =  KIRR(KOLD-K)
       CCAPI(KTOP-K)           =  CCAPI(KOLD-K)
      IDNAME(KTOP-K)           =  IDNAME(KOLD-K)
396       JIRR(KTOP-K)         =  JIRR(KOLD-K)
           JIRR(KK)       =  I
           KIRR(KK)       =  KIR(L)
           WRITE     ( *,2034     ) KIRR(KK)
```

```
               READ    ( *,9000   ) IDNAME(KK)
              CCAPI(KK)      =  0.0
39                  IG  =  IG  +  1
C
500    CONTINUE
C
2002   FORMAT    ( ' Is there  a reservoir  at  the  node  ',A15,  '?  Answer   N  i
       1f  there  is  none.   ' )
2003   FORMAT    ( ' What  is  the  INTEGER   associated   with  the  irrigation   dist
       1rict  served  by  ',A15,  '?  Enter  0  if  this  is  no  turn  out  for  an  ir
       2rigation  district  at  this  node.  If  there  is  more  than  one  crop  d
       3istrict  served  by  this  node,  enter  as  many  as  five  different  INTE
       4GERS  associated  with  these  districts,  using  three  spaces  for  each
       5  ' )
2034   FORMAT    ( ' What  is  the  name  of  the  irrigation  district  which  is  ',
       1I3,'th  in  the  CROPS  sequence  of  districts?  Enter  a  name  in  4  or  l
       2ess  spaces.   ' )
2035   FORMAT    ( ' What  is  the  amount  of  flow  lost  by  the  canal  as  a  perce
       1nt  of  flow  released  '/'  upstream?   ' )
3001   FORMAT    ( ' How  many  nodes  do  you  wish  to  insert  between  ',A15,'  a
       1nd  ',A15,'?  All  the  nodes  entered  now  would  be  on  the  same  stre
       2am  as  the  first  node.  You  can  add  new  offstream  nodes  later.  E
       3nter  an  INTEGER  value:  ' )
3010   FORMAT    ( ' What  is  the  name  of  the  next  node  downstream  of  ',A15,
       1'  Enter  a  unique  name  of  12  or  less  letters.   ' )
3015   FORMAT    ( ' Are  there  restrictions  on  the  regulated  downstream  rele
       1ases  at  the  node  '/1H  ,A12,',  either  minimum  or  maximum  levels?  ')
3016   FORMAT    ( ' What  is  the  minimum  flow  level  in  m3/s  above  which  the
       1  regulated  downstream'/'  releases  of  ',A12,'  must  be?  ' )
3017   FORMAT    ( ' What  is  the  maximum  flow  level  in  m3/s  below  which  the
       1  regulated  downstream'/'  releases  of  ',A12,'  must  be?  ' )
3032   FORMAT    ( ' What  is  the  capacity  of  the  diversion  canal  in  m3/s?
       1  If  there  is  no  limit,  enter  99999:  ' )
3035   FORMAT    ( ' What  is  the  amount  of  flow  lost  by  the  canal  as  a  perce
       1nt  of  flow  released  '/'  upstream?   ' )
4000   FORMAT    ( ' Does  the  node  ',A15,  '  receive  water  from  OTHER  than  i
       1ts  natural  drainage  basin  -  offstream  sources,  diversions  o
       2r  previously  entered  tributaries  Answer  Y  if  it  does.  ')
4001   FORMAT    ( ' The  following  lists  the  nodes  previously  entered.  ' )
4002   FORMAT    ( ' Enter  the  INTEGER  to  the  left  of  the  node  from  which  w
       1ater  is  diverted,  returnsor  naturally  flows  to  the  node  ',A15,  '
       2.  If  the  source  node  has  not  yet  been  entered,  enter  0.  ' )
8001   FORMAT    ( ' What  is  the  maximum  capacity  of  the  ',A15,'  reservoir?
       1  Enter  the'/'  storage  volume  in  MCM  ' )
8003   FORMAT    ( ' What  is  the  minimum  operating  level  of  the  ',A15,'  rese
       1rvoir?  Enter  the  storage  volume  in  MCM  ' )
9000   FORMAT    ( A12  )
9001   FORMAT    ( A2  )
9002   FORMAT    ( I5  )
9003   FORMAT    ( F10.0   )
9004   FORMAT    ( I5,A15,I5,A15,I5,A15,I5,A15          )
9005   FORMAT    ( 5I3  )
```

```
9006    FORMAT  ( A4 )
C
1001     CONTINUE
        RETURN
        END
^Z
       SUBROUTINE    CHANGE   ( JJ, KK, W, HH, MINCRP, LANDIF, NX, IFS, NOGO
)
c  This subroutine reviews the information in b:CROPS, amending where
c    required
       DIMENSION   W(3,20),  HH(3,20)
       COMMON   /SHR/  P(3,50,20),  Y(3,50,20),  CL(3,50,20),  F(3,50,20),  U(3,
      150,20),  T(3,50,20),  B(3,50,20),  Q(3,50,20)
       COMMON   /NEW1/  WD(3),  NG(3),  JCROP(3,50,10),  PCROP(3,25),  LNGTH(50)
       COMMON   /PROD/  YMIN(3,50),  NMINJ(50),  NMINI(3)
       COMMON   /CROP/  NAME(50)
       COMMON   /DELTN/  IX(50)
       COMMON   /NOM/  JNAME(40),  IDNAME(21)
       COMMON   /CAP/  KIRR(21)
C
       CHARACTER   * 12  JNAME
       CHARACTER   * 8  NAME
       CHARACTER   * 4  IDNAME
       CHARACTER   * 2  ANS
           NOGO  =  0
C
C  Determine what needs reviewing or changes.  After looking at one set
C  of data, the program returns for another choice.
100     IF ( LANDIF .GT. 0 ) GO TO 110
            WRITE  ( *,1001 )
            GO TO 130
110      WRITE  (*,1000)
130      READ   (*,2000)   ICHOIX
        GO TO ( 40, 10, 30, 50, 60, 70, 20 ) ICHOIX
            GO TO 80
10       CONTINUE
C  All the season district - crops tables will need revision.
C
C   TABLES  BY  DISTRICT  OF  VALUES  FOR  UPDATE
       DO  25  I  =  1,3
       DO  25  K  =  1,KK
          CALL   CROPTB   ( I, K, JJ, MINCRP,  KK, NOGO  )
           IF ( NOGO  .EQ.  99 ) GO TO 80
25       CONTINUE
       GO TO 100
C  Review costs or limits of disparate amounts of idle land in the
C   objective function.
20       CALL   COSTS   ( JJ, KK, LANDIF   )
       GO TO 100
C
C   TABLE  OF  WATER  PRICES  AND  TOTAL  ARABLE  LAND  BY  DISTRICT
C  Water costs and land available table needs revision.
```

```
30      WRITE   (*,4002)
        DO  35  K=1,KK
C  Find  district  name
        DO  33  K1=2,KK+1
          IF  ( KIRR(K1)  .EQ.  K )  GO  TO  35
33      CONTINUE
C  No  district  K  in  configuration,  return  to  main  change  menu
        WRITE   (*,4004)   K
          NOGO  =  99
          GO  TO  80
35      WRITE   (*,4003)   K,  IDNAME(K1),   W(1,K),   W(2,K),   W(3,K),
     1                                       HH(1,K),  HH(2,K),  HH(3,K)
        WRITE   (*,3004)
        READ    (*,4000)   NCHANJ
        IF  ( NCHANJ  .LE.  0 )  GO  TO  100
            CALL  CHANJ2  (NCHANJ,  JJ,  KK,  W,  HH  )
c  NCHANJ  =  number  of  items  to  change  in  the  table
            GO  TO  30
C  Many  (or  one)  season  and  district  -  crops  table  needs  revision.
40        WRITE   (*,1002)
          READ    (*,2000)   NTIMES
3          DO  15  L=1,  NTIMES
             WRITE   (*,1003)
             READ    (*,2001)   I,  K
             CALL  CROPTB  ( I,  K,  JJ,  MINCRP,  KK,  NOGO  )
             IF  ( NOGO  .EQ.  99 )  GO  TO  80
15      CONTINUE
             GO  TO  100
C  Review  the  length  of  the  growing  season  entered  for  all  crops.
50      WRITE   (*,5000)
        WRITE   (*,5001)   ( J,  NAME(J),   LNGTH(J),   J=1,JJ  )
        WRITE   (*,5002)
        READ    (*,2000)   NX
        IF  ( NX  .LE.  0 )  GO  TO  100
          DO  55  N=1,NX
            WRITE   (*,5003)
            READ    (*,2000)   JX
            WRITE   (*,5004)   NAME(JX)
55          READ    (*,2000)   LNGTH(JX)
        GO  TO  50
C
60      WRITE   (*,6000)
C  If  allowed  will  assign  the  data  one  season  ahead,
C     ie.  season  1  =  season  2  of  last  run
C          season  2  =  season  3  of  last  run
C          season  3  =  newly  defined  season  1
        READ    (*,2002)   ANS
        IF  ( ANS  .EQ.  'Y'  .OR.  ANS  .EQ.  'y' )  GO  TO  61
          GO  TO  100
61        DO  65  K=1,KK
             W(1,K)  =  W(2,K)
             HH(1,K)  =  HH(2,K)
```

```
      WD(1)    =  WD(2)
      W(2,K)   =  W(3,K)
      HH(2,K)  =  HH(3,K)
      WD(2)    =  WD(3)
      W(3,K)   =  W(1,K)
      HH(3,K)  =  HH(1,K)
      WD(3)    =  WD(1)
      DO 65 J=1,JJ
C First to be assigned are the data which appear in all formulations:
C Reassign season 1 data equal to last times season 2
      P(1,J,K)  =  P(2,J,K)
      Y(1,J,K)  =  Y(2,J,K)
      F(1,J,K)  =  F(2,J,K)
      CL(1,J,K) =  CL(2,J,K)
      U(1,J,K)  =  U(2,J,K)
      T(1,J,K)  =  T(2,J,K)
C Reassign season 2 data equal to last times season 3
      P(2,J,K)  =  P(3,J,K)
      Y(2,J,K)  =  Y(3,J,K)
      F(2,J,K)  =  F(3,J,K)
      CL(2,J,K) =  CL(3,J,K)
      U(2,J,K)  =  U(3,J,K)
      T(2,J,K)  =  T(3,J,K)
C Reassign season 3 data equal to new season 1 data
      P(3,J,K)  =  P(1,J,K)
      Y(3,J,K)  =  Y(1,J,K)
      F(3,J,K)  =  F(1,J,K)
      CL(3,J,K) =  CL(1,J,K)
      U(3,J,K)  =  U(1,J,K)
      T(3,J,K)  =  T(1,J,K)
C Now assign the values which are specific to this run.
      IF ( MINCRP .GE. 1 .AND. MINCRP .LT. 3 ) GO TO 62
      B(1,J,K)  =  B(2,J,K)
      B(2,J,K)  =  B(3,J,K)
      B(3,J,K)  =  B(1,J,K)
      IF ( MINCRP .LT. 1 ) GO TO 65
62    YMIN(1,J)  =  YMIN(2,J)
      YMIN(2,J)  =  YMIN(3,J)
      YMIN(3,J)  =  YMIN(1,J)
      IF ( MINCRP .LT. 2 ) GO TO 65
      NG(1)  =  NG(2)
      NG(2)  =  NG(3)
      NG(3)  =  NG(1)
      PCROP(1,J)  =  PCROP(2,J)
      PCROP(2,J)  =  PCROP(3,J)
      PCROP(3,J)  =  PCROP(1,J)
      DO 64 N=1,10
        JCROP(1,J,N)  =  JCROP(2,J,N)
        JCROP(2,J,N)  =  JCROP(3,J,N)
64      JCROP(3,J,N)  =  JCROP(1,J,N)
65    CONTINUE
      IFS  =  IFS + 1
```

```
      IF ( IFS .GE. 2 ) IFS = 0
          GO TO 100
C   An entire district is to be deleted or added
70    WRITE ( *,1070 ) KK
      READ ( *,2000 ) NTIMES
        KK = KK + NTIMES
        IF ( NTIMES .EQ. 0 ) GO TO 100
        IF ( NTIMES .GT. 0 ) GO TO 77
C   A district is to be deleted
          NTIMES = 0 - NTIMES
      DO 75 NX=1,NTIMES
71          WRITE ( *,1071 ) NX
            READ ( *,2000 ) K
C   Find district name
      DO 170 K1=2,KK+1+NTIMES
        IF ( KIRR(K1) .EQ. K ) GO TO 72
170     CONTINUE
C   No district K in configuration, return to 70
      WRITE (*,4004) K
        GO TO 70
72          WRITE ( *,1072 ) K, IDNAME(K1)
            READ ( *,2002 ) ANS
              IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 73
                GO TO 71
73          DO 75 KD=K,KK + NTIMES - 1
                KP = KD + 1
        DO 75 I=1,3
            W(I,KD) = W(I,KP)
           HH(I,KD) = HH(I,KP)
        DO 75 J=1,JJ
          P(I,J,KD) = P(I,J,KP)
          Y(I,J,KD) = Y(I,J,KP)
          F(I,J,KD) = F(I,J,KP)
         CL(I,J,KD) = CL(I,J,KP)
          U(I,J,KD) = U(I,J,KP)
          T(I,J,KD) = T(I,J,KP)
75        B(I,J,KD) = B(I,J,KP)
      GO TO 100
77     DO 79 I=1,3
      DO 79 K=KK-NTIMES+1,KK
          CALL CROPTB ( I, K, JJ, MINCRP, KK, NOGO )
         IF ( NOGO .EQ. 99 ) GO TO 80
79     CONTINUE
      GO TO 100
C
1000   FORMAT ( ' How much would you like to review or correct?'/' Enter
     17 - Only the costs of disparate amounts of idle land in one season
     2.'/ 7X,'6 - Only add or delete one or more districts'/7X, '5 - Onl
     3y have the current season 1 and season 3 data replaced with season
     4'/12X,'2 and season 2 data with the previous season 3'/7X,'4 - Onl
     5y the length of the growing season.'/7X,'3 - Only the district dat
     6a table showing land available and water costs.'/7X,'2 - All the c
```

```
     7rop   data   tables   for   every   district   and   every   season.'/7X,'1    - Only
     8 specific   crop   data   tables   identified   by   the   season   and   district.'
     9/7X,'0   or   nothing   - Nothing   more,   ready   to   continue   with   the   curre
     1nt   values.   ')
1001   FORMAT   ( ' How   much   would   you   like   to   review   or   correct?'/'   Enter
     16  - Only   add   or   delete   one   or   more   districts'/   7X,
     15  - Only   have   the   current   season   1   and   season   3   data   replaced   with
     2  season   2,   and   season   2   data   with   the   previous   season   3.'/
     2   '   4  -  Only   the   length   of   the   growing   season.'/'   3
     3  Only   the   district   data   table   showing   land   available   and   water   cos
     4ts.'/'   2  -  All   the   crop   data   tables   for   every   district   and   e
     5very   season.'/'   1  -  Only   specific   crop   data   tables   identifie
     6d   by   the   season   and   district.'/'   0   or   nothing   -  Nothing   more
     7,   ready   to   continue   with   the   current   values.   ')
1002   FORMAT   ( ' How   many   individual   crop   data   tables   do   you   wish   to   see
     1  and/or   correct?'/'   If   you   wish   to   see   all   those   from   one   district
     2,   that   would   be   three,   one   for'/'   every   season.   ' )
1003   FORMAT   ( ' For   which   season,   1,   2,   or   3   seasons   from   now,   and   for
     1  which   district,   do   you   wish   to   review   the   crop   data   table?   Ente
     2r   the   season   first,   a   space   or   comma,   then   the   district   number.   '
     3  )
1070   FORMAT   ( ' There   are   presently   ',I3,'   districts.   How   many   distric
     1ts   do   you   wish   to   add   or'/'   delete?   Enter   a   negative   number   if   yo
     2u   wish   to   delete   a   district.'/'   Do   not   forget   to   change   the   distri
     3ct   data   tables   as   well!   ' )
1071   FORMAT   ( ' What   is   the   INTEGER   of   the   ',I3,'st   district   you   wish   t
     1o   delete?'/'   This   is   according   to   the   order   it   appears   in   the   CROP
     2S   file.'/'   Enter   the   LARGEST   integers   first.   ' )
1072   FORMAT   ( ' Are   you   sure   you   wish   to   delete   district   ',I2,1X,A4,'?
     1  Answer   Y   to   remove   that   dis  -trict   from   the   crop   files.   ' )
2000   FORMAT   ( I2  )
2001   FORMAT   ( 2I2  )
2002   FORMAT   ( A2  )
3004   FORMAT   ( ' How   many   items   do   you   wish   to   change   in   this   table?   ')
4000   FORMAT   ( I5  )
4002   FORMAT(   23X,   '1.',7X,'2.',7X,'3.',12X,'4.',7X,'5.',7X,'6.'/
     1  5x,'   DISTRICT',4X,'IRR.   DEL.   COSTS   ( Rs/1000   m3   )',3X,'LAND   AVAI
     2LABLE   ( Hectares   ) '/  18X,'   season   1   season   2   season   3',2X,'   se
     3ason   1   season   2   season   3'/)
4003   FORMAT   ( 1X,I3,3X,A4,5X,3F10.2,3X,3F10.2   )
4004   FORMAT   ( ' No   district   ',I3,'   has   been   found   in   the   configuration
     1to   match   that   district   in'/'   CROPS.   ' )
5000   FORMAT   ( ' Following   is   a   list   of   the   crop   names   with   the   crop   num
     1ber   to   the   left   and   the   code   for   the   growing   season   to   the   right.
     2'/'   0  -  is   for   1   growing   season,   1   is   for   two   or   more   growing   sea
     3sons.')
5001   FORMAT   ( I3,A10,I2,3x,I3,A10,I2,3X,I3,A10,I2,3X,I3,A10,I2   )
5002   FORMAT   ( ' If   you   wish   to   change   the   codes   of   any   of   the   crops,   fi
     1rst   note   the   crop   numbersthen   enter   the   number   of   crops   you   wish   t
     2o   change.'/'   Zero   or   nothing   before   RETURN   will   return   you   to   the
     3preceeding   menu.   ')
5003   FORMAT   ( ' What   is   the   crop   number   of   a   crop   whose   code   you   would
```

```
     1like to change? ')
5004  FORMAT ( ' What is the correct code for the crop ',A10,'? ')
6000  FORMAT ( ' If allowed the data will be assigned one season ahead,'
     1/' ie. season 1 = season 2 of last run'
     2/' season 2 = season 3 of last run'
     3/' season 3 = newly defined season 1.'/' Answer Y if
     4 this is what you want to do. ' )
C
80    CONTINUE
      RETURN
      END
c
      SUBROUTINE CHANJ1 ( NCHANJ, JJ, K, I, MINCRP, NEND, NX, KK )
c This subroutine corrects items in the CROPS table
      COMMON /SHR/ P(3,50,20), Y(3,50,20), CL(3,50,20), F(3,50,20), U(3,
     150,20), T(3,50,20), B(3,50,20), Q(3,50,20)
      COMMON /CROP/ NAME(50)
      COMMON /DELTN/ IX(50)
      COMMON /NOM/ JNAME(40), IDNAME(21)
      COMMON /CAP/ KIRR(21)
       CHARACTER * 4 IDNAME
       CHARACTER * 12 JNAME
       CHARACTER * 8 NAME
       CHARACTER * 2 ANS
C
       WRITE (*, 1000)
       READ (*, 1001) ICHOIX
       GO TO ( 100, 200, 300, 400 ) ICHOIX
            DO 20 M=1, NCHANJ
               WRITE (*,4001)
               READ (*,1001) J
               WRITE (*, 4004)
               READ (*, 1001) NCOL
               IF ( NCOL .GT. 6 ) GO TO 29
               IF ( NCOL .GE. 4 ) GO TO 25
                 IF ( NCOL - 2 ) 21, 22, 23
21                 WRITE (*,1002)
                  READ (*,2001) NAME(J)
                  GO TO 20
22                 WRITE(*,1003) NAME(J), K
                  READ (*,2002) P(I,J,K)
                  GO TO 20
23                 WRITE (*,1004) NAME(J), K
                  READ (*,2002) Y(I,J,K)
                  GO TO 20
25                IF ( NCOL - 5 ) 26, 27, 28
26                 WRITE (*,1005) NAME(J), K
                  READ (*,2002) CL(I,J,K)
                  GO TO 20
27                 WRITE (*,1006) NAME(J), K
                  READ (*,2002) F(I,J,K)
                  GO TO 20
```

```
28                 WRITE   (*,1007)    NAME(J),     K
                  READ    (*,2002)    U(I,J,K)
                  GO   TO   20
29                 IF  (  NCOL    .GT.   7  )  GOTO     30
                  WRITE   (*,1008)    NAME(J),     K
                  READ    (*,2002)    T(I,J,K)
                  GO   TO   20
30                 WRITE   (*,1009)    NAME(J),     K
                  READ    (*,2002)    B(I,J,K)
20             CONTINUE
      GO   TO   999
C   ADD   A  CROP,    IE  JJ  =  II  +  JJ
100    WRITE    (*,4002)
      READ    (*,1001)     II
        IPLUS1     =  JJ  +  1
        IPLUS     =  JJ  +  II
           DO   101   J=IPLUS1,       IPLUS
                  WRITE   (*,1002)
                  READ    (*,2001)    NAME(J)
                  WRITE(*,1003)       NAME(J),     K
                  READ    (*,2002)    P(I,J,K)
                  WRITE   (*,1004)    NAME(J),     K
                  READ    (*,2002)    Y(I,J,K)
                  WRITE   (*,1005)    NAME(J),     K
                  READ    (*,2002)    CL(I,J,K)
                  WRITE   (*,1006)    NAME(J),     K
                  READ    (*,2002)    F(I,J,K)
                  WRITE   (*,1007)    NAME(J),     K
                  READ    (*,2002)    U(I,J,K)
                  WRITE   (*,1008)    NAME(J),     K
                  READ    (*,2002)    T(I,J,K)
                IF  (  MINCRP     .GE.   1  .AND.    MINCRP     .LT.    3  )  GO   TO   101
                  WRITE   (*,1009)    NAME(J),     K
                  READ    (*,2002)    B(I,J,K)
101    CONTINUE
           JJ  =  IPLUS
         NEND     =  NEND     +  II
      GO   TO   999
C   DELETE    A  CROP,    IE  JJ=JJ-1,     AND   THE   CROP    NOT   WANTED      DELETED
200    WRITE    (*,4003)
      READ    (*,1001)     NX
      DO   202   L  =  1,NX
        WRITE    (*,   4005    )
202       READ    (*,   1001)    IX(L)
        WRITE    (*,2003)     (  IX(L),    L=1,NX      )
        WRITE    (*,4007)
        READ    (*,2004)     ANS
         IF  (  ANS    .EQ.   'Y'  .OR.   ANS    .EQ.   'y'  )  GO   TO   200
           NEND     =  NEND     -  NX
             JJ  =  JJ  -  NX
           JSUB     =  JJ
              L  =  1
```

```
      DO 208 J=1,JJ
         IF ( J .LT. IX(L) ) GO TO 208
C Determine which crop to substitute in
203          JSUB = JSUB + 1
        DO 204 M=1,NX
        IF ( (JSUB) .EQ. IX(M) ) GO TO 203
204      CONTINUE
             NAME(J) = NAME(JSUB)
          DO 205 IHERE=1,3
          DO 205 KHERE=1,KK
              P(IHERE,J,KHERE) = P(IHERE,JSUB,KHERE)
              Y(IHERE,J,KHERE) = Y(IHERE,JSUB,KHERE)
              F(IHERE,J,KHERE) = F(IHERE,JSUB,KHERE)
             CL(IHERE,J,KHERE) = CL(IHERE,JSUB,KHERE)
              U(IHERE,J,KHERE) = U(IHERE,JSUB,KHERE)
205            T(IHERE,J,KHERE) = T(IHERE,JSUB,KHERE)
         IF ( MINCRP .GE. 1 .AND. MINCRP .LT. 3 ) GO TO 207
          DO 206 I=1,3
          DO 206 K=1,KK
206            B(IHERE,J,KHERE) = B(IHERE,JSUB,KHERE)
207                     L = L + 1
           IF ( L .GT. NX ) GO TO 999
208     CONTINUE
        GO TO 999
C  CHANGE AN ENTIRE ROW
300      WRITE (*,4006)
        READ (*, 1001) J
                  WRITE (*,1002)
                  READ (*,2001) NAME(J)
                  WRITE(*,1003) NAME(J), K
                  READ (*,2002) P(I,J,K)
                  WRITE (*,1004) NAME(J), K
                  READ (*,2002) Y(I,J,K)
                  WRITE (*,1005) NAME(J), K
                  READ (*,2002) CL(I,J,K)
                  WRITE (*,1006) NAME(J), K
                  READ (*,2002) F(I,J,K)
                  WRITE (*,1007) NAME (J), K
                  READ (*,2002) U(I,J,K)
                  WRITE (*,1008) NAME (J), K
                  READ (*,2002) T(I,J,K)
                IF ( MINCRP .GE. 1 .AND. MINCRP .LT. 3 ) GO TO 999
                  WRITE (*,1009) NAME (J), K
                  READ (*,2002) B(I,J,K)
        GO TO 999
C  CHANGE AN ENTIRE COLUMN
400      WRITE (*,4004)
          READ (*,1001) NCOL
              IF ( NCOL .GT. 6 ) GO TO 49
              IF ( NCOL .GE. 4 ) GO TO 45
                IF ( NCOL - 2 ) 41, 42, 43
41          DO 409 J=1,JJ
```

```
      WRITE (*,1002)
409   READ (*,2001) NAME(J)
      GO TO 999
42    DO 401 J=1,JJ
      WRITE(*,1003) NAME(J), K
401   READ (*,2002) P(I,J,K)
      GO TO 999
43    DO 402 J=1,JJ
      WRITE (*,1004) NAME(J), K
402   READ (*,2002) Y(I,J,K)
      GO TO 999
45    IF ( NCOL - 5 ) 46, 47, 48
46    DO 403 J=1,JJ
      WRITE (*,1005) NAME(J), K
403   READ (*,2002) CL(I,J,K)
      GO TO 999
47    DO 404 J=1,JJ
      WRITE (*,1006) NAME(J), K
404   READ (*,2002) F(I,J,K)
      GO TO 999
48    DO 405 J=1,JJ
      WRITE (*,1007) NAME(J), K
405   READ (*,2002) U(I,J,K)
      GO TO 999
49    IF ( NCOL .GT. 7 ) GO TO 50
      DO 415 J=1,JJ
      WRITE (*,1008) NAME(J), K
415   READ (*,2002) T(I,J,K)
      GO TO 999
50    DO 425 J=1,JJ
      WRITE (*,1009) NAME(J), K
425   READ (*,2002) B(I,J,K)
C
1000  FORMAT ( ' Do you wish to add a crop, delete one, or change the v
     1alues in a whole row?'/' Answer 1 to add a row'/' Answer 2 to del
     2ete a crop completely'/' Answer 3 to change the entire row'/'
     3Answer 4 to change an entire column'/' If you answer 0 or nothing
     4, you can make changes item by item. ' )
1001  FORMAT (I5)
1002  Format( ' List the crop you are thinking of growing in the next',
     1' few months in a word of 8 or less letters. ')
1003  FORMAT ( ' What is the expected price for the amount of', A10,
     1 ' which can be grown on a hectare in district',I3, '? ')
1004  FORMAT ( ' What is the expected yield of',A10,' which can be grow
     1n on a hectare in dis-trict', I3, '? ' )
1005  FORMAT ( ' What is the cost per hectare of the labour required to
     1grow this much', A10,' per hectare in district',I5,'? ')
1006  FORMAT ( ' What is the expected cost per hectare of the fertilizer
     1 required to grow this much', A10,' per hectare in district',I5
     2, '? ')
1007  FORMAT ( ' How much water in cubic metres is required per hectare
     1 by ', A10,' in district',I5,' ? ')
```

```
1008   FORMAT   ( ' What  is  the  total  amount  of  hectares  suitable  for', A10
      1,'  in  district',I5,  '?  ')
1009   FORMAT   ( ' How  many  hectares  are  required  to  grow  the  minimum  acce
      1ptable  amount  of', A9,'  crop  in  district  ',I2,'?  ' )
2001    FORMAT   ( A8  )
2002    FORMAT   ( F8.0  )
2003    FORMAT   ( 10I7  )
2004    FORMAT   ( A2  )
4001   FORMAT   ( ' Find  a  row  you  want  to  change.  Enter  its  number  ')
4002   FORMAT   ( ' How  many  crops  do  you  wish  to  add  to  those  already  ther
      1e?  ')
4003   FORMAT   ( ' How  many  crops  do  you  wish  to  delete  from  those  already
      1  there?  They  will  disap-pear  from  all  districts  and  seasons.  ')
4004   FORMAT   ( ' What  column  do  you  wish  to  change?  Enter  the  number  '
      1,'beside  the  title.  ')
4005   FORMAT   ( ' What  is  the  integer  beside  the  crop  you  wish  to  delete?
      1  Give  the  lowest  integers  first,  one  at  a  time.  ')
4006   FORMAT   ( ' What  is  the  integer  beside  the  crop  you  want  to  replace
      1?  ')
4007   FORMAT   ( ' These  are  the  integer(s)  of  the  crops  that  will  be  dele
      1ted.  If  they  are  not  in  ascending  order  or  you  wish  to  delete  oth
      2ers,  Enter  Y  to  re-enter  correctly.  ')
999     CONTINUE
       RETURN
       END
c
       SUBROUTINE  CHANJ2  (NCHANJ,  JJ,  KK,  W,HH  )
c  This  subroutine  corrects  items  in  the  district  table
       DIMENSION  W(3,20),  HH(3,20)
C
        WRITE  (*,  1000)
        READ  (*,  1001)  ICHOIX
        IF  (ICHOIX  .GE.  2)  GO  TO  200
        IF  (ICHOIX  .GE.  1)  GO  TO  100
            DO  20  M=1,  NCHANJ
              WRITE  (*,4006)
              READ  (*,1001)  K
              WRITE  (*,  4004)
              READ  (*,  1001)  NCOL
                IF  ( NCOL  .GT.  3  )  GO  TO  24
                IF  ( NCOL  -  2  )  21,  22,  23
21                 WRITE  (*,1010)  K
                  READ  (*,2002)  W  (1,K)
                  GO  TO  20
22                 WRITE(*,1011)  K
                  READ  (*,2002)  W  (2,K)
                  GO  TO  20
23                 WRITE  (*,1012)  K
                  READ  (*,2002)  W  (3,K)
                  GO  TO  20
24               IF  ( NCOL  -  5  )  25,  26,  27
25                 WRITE  (*,1013)  K
```

```
                READ    (*,2002)    HH  (1,K)
                GO  TO  20
26               WRITE   (*,1014)    K
                READ    (*,2002)    HH  (2,K)
                GO  TO  20
27               WRITE   (*,1015)    K
                READ    (*,2002)    HH  (3,K)
                GO  TO  20
20            CONTINUE
      GO  TO  999
C  CHANGE   AN  ENTIRE   ROW
100     WRITE   (*,4006)
      READ   (*,  1001)   K
                WRITE   (*,1010)    K
                READ    (*,2002)    W(1,K)
                WRITE(*,1011)       K
                READ    (*,2002)    W(2,K)
                WRITE   (*,1012)    K
                READ    (*,2002)    W(3,K)
                WRITE   (*,1013)    K
                READ    (*,2002)    HH(1,K)
                WRITE   (*,1014)    K
                READ    (*,2002)    HH(2,K)
                WRITE   (*,1015)    K
                READ    (*,2002)    HH(3,K)
      GO  TO  999
C  CHANGE   AN  ENTIRE   COLUMN
200      WRITE   (*,4004)
      READ   (*,1001)    NCOL
              IF  (  NCOL   .GT.   3  )  GO   TO   49
              IF  (  NCOL   -  2  )  46,   47,   48
46           DO   403   K=1,KK
                WRITE   (*,1010)    K
403              READ    (*,2002)    W(1,K)
                GO  TO  999
47           DO   404   K=1,KK
                WRITE   (*,1011)    K
404              READ    (*,2002)    W(2,K)
                GO  TO  999
48           DO   405   K=1,KK
                WRITE   (*,1012)    K
405              READ    (*,2002)     W(3,K)
                GO  TO  999
49          IF  (  NCOL    -  5  )  51,   52,   53
51           DO   406   K=1,KK
                WRITE   (*,1013)    K
406              READ    (*,2002)    HH(1,K)
                GO  TO  999
52           DO   416   K=1,KK
                WRITE   (*,1014)    K
416              READ    (*,2002)    HH(2,K)
                GO  TO  999
```

```
53          DO 426 K=1,KK
                WRITE (*,1015) K
426             READ (*,2002) HH(3,K)
                GO TO 999
C
1000    FORMAT ( ' What kind of changes do you wish to make?
     1/' Answer 1 to change an entire row'/
     2' Answer 2 to change the whole column'/' If you answer 0 or noth
     4ing, you can make changes item by item. ' )
1001    FORMAT (I5)
1010    FORMAT ( ' What is the cost of irrigation water delivered to distr
     1ict', I4, ' in season 1? ')
1011    FORMAT ( ' What is the cost of irrigation water delivered to distr
     1ict', I4, ' in season 2? ')
1012    FORMAT ( ' What is the cost of irrigation water delivered to distr
     1ict', I4, ' in season 3? ')
1013    FORMAT ( ' What is the total number of hectares available for cult
     1ivation in district',I5,' during season 1? ' )
1014    FORMAT ( ' What is the total number of hectares available for cult
     1ivation in district', I5, ' during season 2? ')
1015    FORMAT ( ' What is the total number of hectares available for cult
     1ivation in district', I5, ' during season 3? ')
2002    FORMAT (F8.0)
4004    FORMAT ( ' What column do you wish to change? Enter the number '
     1,'above the title. ')
4006    FORMAT ( ' What is the integer for the district whose values need
     1replacing? ' )
999     CONTINUE
        RETURN
        END
C
        SUBROUTINE CROPTB ( I, K, JJ, MINCRP, KK, NOGO )
c This subroutine displays the crop data in a table for each district
c and each season
        COMMON /SHR/ P(3,50,20), Y(3,50,20), CL(3,50,20), F(3,50,20), U(3,
     150,20), T(3,50,20), B(3,50,20), Q(3,50,20)
        COMMON /CROP/ NAME(50)
        COMMON /DELTN/ IX(50)
        COMMON /NOM/ JNAME(40), IDNAME(21)
        COMMON /CAP/ KIRR(21)
C
        CHARACTER * 12 JNAME
        CHARACTER * 8 NAME
        CHARACTER * 4 IDNAME
        CHARACTER * 2 ANS
C Find district name
        DO 33 K1=2,KK+1
          IF ( KIRR(K1) .EQ. K ) GO TO 35
33      CONTINUE
C No district K in configuration, return to main change menu
        WRITE (*,4004) K
          NOGO = 99
```

```
      GO TO 100
c Display 15 crops at a time in a table by district and by season
35      NBEG = 1
        JTIMES = JJ/15 + 1
        DO 9 LL = 1, JTIMES
          NEND = 14 + NBEG
          IF ( NEND .GT. JJ ) NEND = JJ
5         WRITE (*,3000) K, IDNAME(K1), I
          WRITE (*,3001)
          IF ( MINCRP .GE. 1 .AND. MINCRP .LT. 3 ) GO TO 6
          WRITE (*,3002) ( J, NAME(J), P(I,J,K), Y(I,J,K), CL(I,J,K),
     1    F(I,J,K), U(I,J,K), T(I,J,K), B(I,J,K), J=NBEG, NEND )
          GO TO 8
6         WRITE (*,3022) ( J, NAME(J), P(I,J,K), Y(I,J,K), CL(I,J,K),
     1    F(I,J,K), U(I,J,K), T(I,J,K), J=NBEG, NEND )
8         WRITE (*,3003)
          READ (*,4000) NCHANJ
          IF ( NCHANJ .LE. 0 ) GO TO 9
            CALL CHANJ1 ( NCHANJ, JJ, K, I, MINCRP, NEND, NX, KK )
c NCHANJ = number of items to change in the table
            GO TO 5
9         NBEG = NEND + 1
C
100   CONTINUE
3000  FORMAT ( ' For district', I3,1X,A4,' and season',I4,' the values
     1 are as shown: ')
3001  FORMAT ( 7H 1.CROP,5X,7H2.PRICE,3X,8H3. YIELD,9X,5HCOSTS,4X,7H6.
I
     1RR.,2X,22H7. SUITABLE 8.MINIMUM/31X,19H4.labour 5. fert.,3X,6HD
     2EMAND,4X,4HLAND,7X,5HPROD. /13X,5HRs/ha,21X,5HRs/ha,7X,10H1000 m3/
     3ha,4X,2Hha,8X,2Hha )
3002  FORMAT ( I3,1X,A8,F7.2,6F10.2 )
3022  FORMAT ( I3,1X,A8,F7.2,5F10.2,6X,'N / A ')
3003  FORMAT ( ' How many items do you wish to change in this table?'/'
     1 Answer with an integer number or you may hit the RETURN to contin
     2ue on to the next.')
4000  FORMAT ( I5 )
4004  FORMAT ( ' No district ',I3,' has been found in the configuration
     1to match that district in CROPS. ' )
      RETURN
      END
^Z
      SUBROUTINE COSTS ( JJ, KK, LANDIF )
C This subroutine assigns the penalty value of idle hectares or the
c restricted limit. The seasonal
C values are saved on the diskette file b:CROPS3 for the next run.
      COMMON /NEW1/ NG(3), WD(3), JCROP(3,50,10), PCROP(3,25), LNGTH(50)
      COMMON /CROP/ NAME(50)
      CHARACTER * 8 NAME
      CHARACTER * 2 ANS
C
c Determine if idle land penalized or restricted
```

```
900    IF ( LANDIF .LT. 2 ) GO TO 901
         WRITE (*,1090) WD(1), WD(2), WD(3)
         GO TO 902
901    WRITE (*,1091) WD(1), WD(2), WD(3)
902    READ (*,2001) NX
       IF ( NX .LE. 0 ) GO TO 5
        DO 910 I=1,NX
          WRITE (*,1092)
          READ (*,2001) IX
          WRITE (*,1093) IX
910        READ (*,2002) WD(IX)
       GO TO 900
C
1090   FORMAT ( ' These are the costs in Rs last used for hectares left u
      1ncultivated:'/' 1',F9.2,' between season 1 and 2'/' 2',F9.2,' be
      2tween season 2 and 3'/' 3',F9.2,' between season 1 and 3'///' How
      3 many would you like to change? Answer 0 or nothing before the
      4RETURN to return to the last menu. ')
1091   FORMAT ( ' These are the maximum allowable differences in uncultiv
      1ated land between seasons in hectares:'/' 1',F9.2,' between seaso
      2n 1 and 2'/' 2',F9.2,' between season 2 and 3'/' 3',F9.2,' betwe
      3en season 1 and 3'///' How many would you like to change? Answer
      40 or nothing before the RETURN to return to the last menu. ')
1092   FORMAT ( ' What is the integer to the left of the value to be chan
      1ged? ')
1093   FORMAT ( ' What is the correct value for no.',I3,'? ')
2001   FORMAT ( I2 )
2002   FORMAT ( F8.0 )
5      CONTINUE
         RETURN
         END
^Z
      SUBROUTINE COMCRP ( JJ )
c This subroutine combines the yields of certain crop varieties under a
common target as a constraint in the LP formulation
      COMMON /NEW1/ NG(3), WD(3), JCROP(3,50,10), PCROP(3,25), LNGTH(50)
      COMMON /CROP/ NAME(50)
      DIMENSION J1(50), L1(50), I1X(10)
      CHARACTER * 8 NAME, NAMES(50)
      CHARACTER * 2 ANS
C
c Dispaly the combination of crops entered from b:CROPS or just revised
      DO 600, I=1,3
10        WRITE ( *,1000 ) I, NG(I)
          WRITE ( *,1001 ) ( K, PCROP(I,K), K=1,NG(I) )
          ISTART = 0
          WRITE ( *,1002 )
          READ ( *,2000 ) ICHOIX
          GO TO ( 100,200,300,400,50 ) ICHOIX
C Continue to next season
          GO TO 600
C Redo completely
```

```
50      WRITE ( *,1003 )
        READ ( *,2002 ) ANS
        IF ( ANS .EQ. 'N' .OR. ANS .EQ. 'n' ) GO TO 10
        WRITE ( *,1004) I
        READ ( *,2000 ) NG(I)
        DO 12, J=1,JJ
           L1(J) = 0
        DO 12, NJ=1,10
12         JCROP(I,J,NJ) = 0
        DO 61, N=1,NG(I)
           ISTART = 100
        CALL GCHANJ ( I, JJ, N, ISTART, L1 )
c     I = season index
c     JJ = total number of crops
c     N = group number in question
c ISTART = code for complete revision, ISTART > 0. ISTART = 0 create
c        additional groups
c     L1 = vector of the last group number for each crop
61      CONTINUE
           GO TO 10
C
100     WRITE ( *,1100 )
C Delete groups of crops
      READ (*,2000) NX
      IF ( NX .EQ. 0 ) GO TO 10
      DO 110 L = 1,NX
        WRITE (*,1101)
110     READ (*, 2000) I1X(L)
        WRITE (*,1102) ( I1X(L), L=1,NX )
        WRITE (*,1103)
        READ (*,2002) ANS
         IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 115
          GO TO 100
115        NG(I) = NG(I) - NX
           NSUB = NG(I)
              L = 1
      DO 190 N=1,NG(I)
         IF ( N .LT. I1X(L) ) GO TO 190
C Determine which crop to substitute in
120        NSUB = NSUB + 1
        DO 130 M=1,NX
        IF ( (NSUB) .EQ. I1X(M) ) GO TO 120
130     CONTINUE
          PCROP(I,N) = PCROP(I,NSUB)
          DO 170 J=1,JJ
              NJ = 1
140         IF ( JCROP(I,J,NJ) .NE. N ) GO TO 160
150           JCROP(I,J,NJ) = JCROP(I,J,NJ+1)
                 NJ = NJ + 1
              IF ( JCROP(I,J,NJ) .LE. 0 ) GO TO 170
                 GO TO 150
160         NJ = NJ + 1
```

```
            IF ( JCROP(I,J,NJ) .LE. 0 ) GO TO 170
            GO TO 140
170     CONTINUE
          DO 180 J=1,JJ
          DO 180 NJ=1,10
            IF ( JCROP(I,J,NJ) .EQ. NSUB ) JCROP(I,J,NJ) = N
180     CONTINUE
            L = L + 1
          IF ( L .GT. NX ) GO TO 10
190     CONTINUE
C Delete groups that are greater than NG(I):
        DO 195 J=1,JJ
        DO 195 NJ=1,10
        IF ( JCROP(I,J,NJ) .GT. NG(I) ) JCROP(I,J,NJ) = 0
195     CONTINUE
          GO TO 10
200     WRITE ( *,1200 )
C Add groups of crops
        READ ( *,2000 ) IPLUS
        ISTART = 10
        DO 250, K=1,IPLUS
          CALL GCHANJ ( I, JJ, NG(I)+K, ISTART, L1 )
c     I = season index
c     JJ = total number of crops
c NG(I)+K= group number in question
c ISTART = code for complete revision, ISTART > 0. ISTART = 0 create
c          additional groups
c     L1 = vector of the last group number for each crop
250     CONTINUE
          NG(I) = NG(I) + IPLUS
          GO TO 10
C Change the crop combinations
300     DO 350, N=1,NG(I)
          CALL GCHANJ ( I, JJ, N, ISTART, L1 )
c     I = season index
c     JJ = total number of crops
c     N = group number in question
c ISTART = code for complete revision, ISTART > 0. ISTART = 0 create
c          additional groups
c     L1 = vector of the last group number for each crop
350     CONTINUE
          GO TO 10
400     WRITE ( *,1400 )
C Change the minimum crop production figures
        READ ( *,2000 ) NX
        DO 450, L=1,NX
          WRITE ( *,1401 ) L
           READ ( *,2000 ) IX
          WRITE ( *,1402 ) IX
450            READ ( *,2001 ) PCROP(I,IX)
          GO TO 10
600     CONTINUE
```

```
C
1000    FORMAT    ( ' In season',I3,' there are already',I3,' group(s) of cro
     1ps which combined must have a minimum production of' )
1001    FORMAT    ( I5,F9.2,I5,F9.2,I5,F9.2,I5,F9.2,I5,F9.2 )
1002    FORMAT    ( ' Answer 5 to redo the groups without retaining any from
     1the last time' / 8X,'4 if you wish to change on
     2ly the production target'/ 8X,'3 if you wish to regroup or review
     3the crops combined in the group(s)'/10X,'in this season'/ 8X,'2 if
     4 you wish to add new groups'/ 8X,'1 if you wish to delete groups'/
     58X,'0 or nothing to continue with these values ')
1003    FORMAT    ( ' Do you really wish to recombine the crops from scratch?
     1 Answer N to return to the last menu ')
1004    FORMAT    ( ' How many groups of combined crops are there in season',
     1 I3, '? ')
1100    FORMAT    ( ' How many groups do you wish to delete? Answer 0 to dir
     1ectly return to the last menu. ')
1101    FORMAT    ( ' Which is the integer beside the minimum production of t
     1he group you would like to delete? Answer the lowest integers fi
     2rst. ')
1102    FORMAT    ( ' These are the group numbers to be deleted:'/ 10I5 )
1103    FORMAT    ( ' These groups will immediately disappear. If this is wh
     1at you wish, answer Y ' )
1200    FORMAT    ( ' How many groups do wish to add? ')
1400    FORMAT    ( ' How many groups have different production quantities?
     1 ')
1401    FORMAT    ( ' What is the integer to the left of the ',I3,'st incorre
     2ct quantity? ')
1402    FORMAT    ( ' What is the new quantity for the group',I3,'? ')
2000    FORMAT    ( I3 )
2001    FORMAT    ( F9.0 )
2002    FORMAT    ( A2 )
        RETURN
        END
c
        SUBROUTINE GCHANJ ( I, JJ, N, ISTART, L1 )
c  This subroutine amends the crop groupings
        COMMON /NEW1/ NG(3), WD(3), JCROP(3,50,10), PCROP(3,25)
        COMMON /CROP/ NAME(50)
        DIMENSION J1(50), L1(50), N1(50)
        CHARACTER * 8 NAME, NAMES(50)
        CHARACTER * 2 ANS
C
C  Start with previous values or immediately enter new:
        IF ( ISTART .GT. 20 ) GO TO 15
C  Find the last non-zero value in JCROP in the season for each crop.
        DO 6 J=1,JJ
           L1(J) = 0
          DO 5 NJ=1,10
           IF ( JCROP(I,J,NJ) .LE. 0 ) GO TO 6
              L1(J) = L1(J) + 1
5         CONTINUE
6       CONTINUE
```

```
      IF ( ISTART .GT. 2 ) GO TO 15
C Write out PCROP(I,N) and the list of crops in the group and ask for
C corrections
25    WRITE ( *,1010 ) N, I, PCROP(I,N)
        L = 1
      DO 30, J=1,JJ
      DO 30 NJ=1,L1(J)
        IF ( JCROP(I,J,NJ) .NE. N ) GO TO 30
          NAMES (L) = NAME(J)
            J1 (L) = J
            N1 (L) = NJ
              L = L + 1
30    CONTINUE
            NC = L - 1
32    WRITE ( *,1011 ) ( J1(J), NAMES(J), J=1,NC )
      WRITE ( *,1012 )
      READ ( *,2002 ) ANS
      IF ( ANS .EQ. 'N' .OR. ANS .EQ. 'n' ) GO TO 35
        GO TO 61
c Determine if corrections required
35      WRITE ( *, 1013 )
        READ ( *,2002 ) ANS
        IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 50
          WRITE ( *,1014 )
          READ ( *,2000 ) NX
          IF ( NX .LE. 0 ) GO TO 55
            DO 40, M=1,NX
              WRITE ( *,1015 ) M
              READ ( *,2000 ) JX
              DO 42, J=1,NC
                IF ( J1(J) .EQ. JX ) GO TO 45
42            CONTINUE
              WRITE ( *,1016 ) JX
                GO TO 32
45            JCROP(I,JX,N1(J)) = JCROP(I,JX,L1(JX))
              JCROP(I,JX,L1(JX)) = 0
              L1(JX) = L1(JX) - 1
              WRITE ( *,1017 ) N
              READ ( *,2000 ) JC
              IF ( JC .LE. 0 ) NC = NC - 1
              L1(JC)=L1(JC) + 1
40          JCROP(I,JC,L1(JC)) = N
            GO TO 25
50      WRITE ( *,1004 ) N, I
        READ ( *,2001 ) PCROP(I,N)
        GO TO 25
55    DO 58, J=1,NC
          JCROP(I,J1(J),N1(J)) = JCROP(I,J1(J),L1(J1(J)))
        JCROP(I,J1(J),L1(J1(J))) = 0
                L1(J1(J)) = L1(J1(J)) - 1
58    CONTINUE
15    WRITE ( *,1002 ) N, I
```

```
          READ     ( *,2000    ) NC
          DO  20,  NN=1,NC
                WRITE    ( *,1003    ) N,  I
                READ     ( *,2000    ) J
                   L1(J)    =  L1(J)    +  1
20              JCROP(I,J,L1(J))         =  N
          WRITE    ( *,1004    ) N,  I
          READ     ( *,2001    ) PCROP(I,N)
          GO  TO  25
61         CONTINUE
C
1002     FORMAT     ( ' How    many    crops    are   in  group    number:',I3,'      in  season',     I
     13,  '?  ')
1003     FORMAT     ( ' What    is  the   integer    to  the   left   of  a  crop    in  group',I3,
     1'  of  season',I3,'?        ')
1004     FORMAT     ( ' What    is  the   minimum     production      quota    of  the   combined      cr
     1ops   of  group',I3,'     of  season',I3,'?        ' )
1010     FORMAT     ( ' The    minimum     quota    for   group',I3,'     in  season',I3,'      is',F
     19.2,'.'/'     The   crops   in  the   group    are:'   )
1011     FORMAT     ( I3,2X,A8,I3,2X,A8,I3,2X,A8,I3,2X,A8,I3,2X,A8,I3,2X,A8                    )
1012     FORMAT     ( ' Is  this   correct?     Answer     N  to  make    any   desired     correcti
     1ons.    ')
1013     FORMAT     ( ' Do   you   wish   to  amend    the   minimum     crop    target?     Answer     Y
     1  to  do   so')
1014     FORMAT     ( ' How    many    crop    numbers     are   wrong?     If  you   answer     zero    yo
     1u   can   re-enter    this   group    all  over   again.    ')
1015     FORMAT     ( ' What    is  the   integer    beside     the   ',I3,'st    crop   you   want   t
     1o   replace?       ')
1016     FORMAT     ( ' You    have    given    the   integer,',I3,',      which    is  not   beside
     1a   crop   in  the   list.   Redo    your    corrections.       ')
1017     FORMAT     ( ' What    is  the   integer    of  the   crop    that   does    belong     in  gro
     1up',I3,'?         Answer    zero    to  delete.     ')
2000     FORMAT     ( I3  )
2001     FORMAT     ( F9.0    )
2002     FORMAT     ( A2   )
        RETURN
        END
^Z
        SUBROUTINE          VARDATA       ( JN,   IRT    )
c    This    subroutine       displays     and    corrects     the   time    dependent        data
        COMMON/SHC/          JIRR(21),     JRES(25),      JNOSPL(20),      JDIV(2,15)
        COMMON/FLO/          NLIM(21),     NSS(3),     FLOMIN(20),       FLOMAX(20)
        COMMON/SHP/          JTRB(2,10),       SLO(25),      SUP(25),     JRTF(2,10),        C1(10)
        COMMON/VAR/FLMIN(3,5,20),              FLMAX(3,5,20),        CHCPI(3,5,21),        P1(3,5,10)
       1          ,CHCPD(3,5,15),        CHLSS(3,5,15),         SMAX(3,5,25),        SMIN(3,5,25)
        COMMON/NOM/          NNAME(40)
C
        CHARACTER         *  12   NNAME,      FNAME,     NAME1
        CHARACTER         *  13   CHAR(6)
        CHARACTER         *  4   TITLE(18),       DUMMYT(18)
        CHARACTER         *  2   ANS
        DATA      CHAR/       ' return     flow   ',  ' restriction        ',  ' irrigation       ',
```

```
     1          ' diversion     ', ' diversion     ', ' reservoir    '/
C
C  Ask if user wishes to enter saved variations,
C                   review any catagory of data
C                or to save corrected data on a file
10    WRITE ( *,1000 )
      READ ( *,9001 ) IND
      IF ( IND .LE. 1 ) GO TO 100
      IF ( IND .GT. 7 ) GO TO 800
      WRITE ( *,1001 ) ( J, NNAME(J), J=1,JN )
20    WRITE ( *,1002 ) CHAR(IND-1)
      READ ( *,9001 ) INODE
      IF ( INODE .LE. 0 ) GO TO 10
      GO TO ( 100, 200, 300, 400, 500, 600, 700, 800 ) IND
      GO TO 999
C  Save the file
100   IF ( IND .LE. 0 ) GO TO 999
        WRITE ( *,1008 ) FNAME
        READ ( *,9005 ) NAME1
          IF ( NAME1 .EQ. '            ' ) GO TO 159
            FNAME = NAME1
159   OPEN ( 9,FILE=FNAME, STATUS='OLD')
      WRITE (*,1006) ( TITLE(M), M=1,18 )
      READ (*,9004) ( DUMMYT(M), M=1,18 )
        IF ( DUMMYT(1) .EQ. '        ' ) GO TO 163
          DO 160 M=1,18
160         TITLE(M) = DUMMYT(M)
163   WRITE ( 9,9100 ) NSS(1), NSS(2), NSS(3), JN, ( TITLE(M), M=1,18 )
      DO 164 N=1,IRT
164   WRITE ( 9,9101 ) ( (P1(I,IS,N), IS=1,NSS(I)), I=1,3 )
      DO 165 N=1,NLIM(1)-1
165   WRITE ( 9,9101 ) ((FLMIN(I,IS,N),FLMAX(I,IS,N),IS=1,NSS(I)),I=1,3)
      DO 166 N=1,JIRR(1)
166   WRITE ( 9,9101 ) ( (CHCPI(I,IS,N), IS=1,NSS(I)), I=1,3 )
      DO 167 N=1,JDIV(1,1)
167   WRITE ( 9,9101 ) ((CHCPD(I,IS,N),CHLSS(I,IS,N),IS=1,NSS(I)),I=1,3)
      DO 168 N=1,JRES(1)
168   WRITE ( 9,9101 ) ((SMIN(I,IS,N),SMAX(I,IS,N), IS=1,NSS(I)), I=1,3)
      CLOSE (9)
        GO TO 10
C  Review percent of irrigation turnout which returns
200   DO 299 J1=1,IRT
          J = J1
      IF ( INODE .GE. 99 ) GO TO 230
      DO 225 J=1,IRT
        IF ( JRTF(1,J) .EQ. INODE ) GO TO 230
225   CONTINUE
C  No RETURN FLOW was found at the node requested, try again
      WRITE ( *,1003 ) CHAR(IND-1), NNAME(INODE), INODE
        GO TO 20
230   WRITE ( *,2000 ) NNAME(JRTF(1,J)), NNAME(JRTF(2,J))
      DO 231 I=1,3
```

```
      WRITE ( *, 9006 ) I
231      WRITE ( *, 9008 ) (IS,P1(I,IS,J), IS=1,NSS(I) )
      WRITE ( *,2002 )
      READ ( *,9001 ) ITIMES
      IF ( ITIMES .LE. 0 ) GO TO 250
C Determine if a whole row or whole column will be changed
         WRITE ( *,1004 )
         READ ( *,9002 ) ANS
         IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 245
C Corrections performed one at a time
       DO 244 NN=1,ITIMES
         WRITE ( *,2005 ) NNAME(JRTF(1,J))
         READ ( *,9001 ) IX
         WRITE ( *,2006 ) IX
         READ ( *,9001 ) ISX
         WRITE ( *,2007 ) NNAME(JRTF(1,J)), ISX, IX
         READ ( *,9003 ) DUMMY
         IF ( DUMMY .LE. 0 ) GO TO 244
           P1( IX, ISX, J ) = DUMMY
244        CONTINUE
      GO TO 230
C Corrections performed for an individual season
245   WRITE ( *,2005 ) NNAME(JRTF(1,J))
      READ ( *,9001 ) IX
      DO 232 IS=1,NSS(IX)
        WRITE ( *,2007 ) NNAME(JRTF(1,J)), IS, IX
        READ ( *,9003 ) DUMMY
        IF ( DUMMY .LE. 0 ) GO TO 232
          P1( IX, IS, J ) = DUMMY
232     CONTINUE
      GO TO 230
250   IF ( INODE .LT. 99 ) GO TO 20
299   CONTINUE
      GO TO 10
C Review sub-seasonal downstream flow restrictions
300   DO 399 J1=1,NLIM(1)-1
          J = J1
      IF ( INODE .GE. 99 ) GO TO 330
      DO 325 J=1,NLIM(1)-1
        IF ( NLIM(J+1) .EQ. INODE ) GO TO 330
325   CONTINUE
C No RESTRICTION was found at the node requested, try again
      WRITE ( *,1003 ) CHAR(IND-1), NNAME(INODE), INODE
      GO TO 20
330   WRITE ( *,3000 ) NNAME(NLIM(J+1))
      DO 331 I=1,3
        WRITE ( *, 9007 ) I
331     WRITE ( *, 9009 ) (IS,FLMAX(I,IS,J),FLMIN(I,IS,J), IS=1,NSS(I) )
      WRITE ( *,3002 )
      READ ( *,9001 ) ITIMES
      IF ( ITIMES .LE. 0 ) GO TO 350
C Determine if a whole row or whole column will be changed
```

```
                WRITE   ( *,1004    )
                READ    ( *,9002    ) ANS
                IF ( ANS  .EQ.  'Y' .OR.  ANS  .EQ.  'y' ) GO  TO  345
C  Corrections    performed    one  at  a  time
              DO  344   NN=1,ITIMES
                WRITE   ( *,3005    ) NNAME(NLIM(J+1))
                READ    ( *,9001    ) IX
                WRITE   ( *,3006    ) IX,  NNAME(NLIM(J+1))
                READ    ( *,9001    ) ISX
                WRITE   ( *,3007    ) NNAME(NLIM(J+1)),         ISX,  IX
                READ    ( *,9003    ) DUMMY
                IF ( DUMMY    .LE.  0 ) GO  TO  3445
                   FLMAX(   IX,  ISX,  J ) =  DUMMY
3445             WRITE   ( *,3008    ) NNAME(NLIM(J+1)),         ISX,  IX
                READ    ( *,9003    ) DUMMY
                IF ( DUMMY    .LE.  0 ) GO  TO  344
                   FLMIN(   IX,  ISX,  J ) =  DUMMY
344              CONTINUE
           GO  TO  330
C  Corrections    performed    for  an  individual    season
345     WRITE   ( *,3005    ) NNAME(NLIM(J+1))
        READ    ( *,9001    ) IX
           DO  332   IS=1,NSS(IX)
              WRITE   ( *,3007    ) NNAME(NLIM(J+1)),         IS,  IX
              READ    ( *,9003    ) DUMMY
              IF ( DUMMY    .LE.  0 ) GO  TO  333
                 FLMAX(   IX,  IS,  J ) =  DUMMY
333            WRITE   ( *,3008    ) NNAME(NLIM(J+1)),         IS,  IX
              READ    ( *,9003    ) DUMMY
              IF ( DUMMY    .LE.  0 ) GO  TO  332
                 FLMIN(   IX,  IS,  J ) =  DUMMY
332          CONTINUE
        GO  TO  330
350      IF ( INODE    .LT.  99  ) GO  TO  20
399      CONTINUE
        GO  TO  10
C  Review   irrigation   canal   sub-seasonal      capacities
400      DO  499  J1=1,JIRR(1)
                J  =  J1
        IF ( INODE    .GE.  99  ) GO  TO  430
        DO  425   J=1,JIRR(1)
          IF ( JIRR(J+1)     .EQ.  INODE    ) GO  TO  430
425       CONTINUE
C  No  IRRIGATION     was  found  at  the  node  requested,     try  again
        WRITE   ( *,1003    ) CHAR(IND-1),      NNAME(INODE),       INODE
         GO  TO  20
430      WRITE   ( *,4000    ) NNAME(JIRR(J+1))
        DO  431   I=1,3
           WRITE   ( *,  9006   ) I
431         WRITE   ( *,  9008   ) (IS,CHCPI(I,IS,J),          IS=1,NSS(I)        )
        WRITE   ( *,4002    )
        READ    ( *,9001    ) ITIMES
```

```
      IF ( ITIMES .LE. 0 ) GO TO 450
C Determine if a whole row or whole column will be changed
            WRITE ( *,1004 )
            READ ( *,9002 ) ANS
            IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 445
C Corrections performed one at a time
          DO 444 NN=1,ITIMES
            WRITE ( *,4005 ) NNAME(JIRR(J+1))
            READ ( *,9001 ) IX
            WRITE ( *,4006 ) IX
            READ ( *,9001 ) ISX
            WRITE ( *,4007 ) NNAME(JIRR(J+1)), ISX, IX
            READ ( *,9003 ) DUMMY
            IF ( DUMMY .LE. 0 ) GO TO 444
               CHCPI( IX, ISX, J ) = DUMMY
444          CONTINUE
        GO TO 430
C Corrections performed for an individual season
445   WRITE ( *,4005 ) NNAME(JIRR(J+1))
      READ ( *,9001 ) IX
        DO 432 IS=1,NSS(IX)
          WRITE ( *,4007 ) J, IS, IX
          READ ( *,9003 ) DUMMY
          IF ( DUMMY .LE. 0 ) GO TO 432
             CHCPI( IX, IS, J ) = DUMMY
432     CONTINUE
      GO TO 430
450   IF ( INODE .LT. 99 ) GO TO 20
499   CONTINUE
      GO TO 10
C Review diversion canal sub-seasonal capacities
500   DO 599 J1=1,JDIV(1,1)
          J = J1
      IF ( INODE .GE. 99 ) GO TO 530
      DO 525 J=1,JDIV(1,1)
        IF ( JDIV(1,(J+1)) .EQ. INODE ) GO TO 530
525     CONTINUE
C No DIVERSION was found at the node requested, try again
      WRITE ( *,1003 ) CHAR(IND-1), NNAME(INODE), INODE
       GO TO 20
530   WRITE ( *,5000 ) NNAME(JDIV(1,(J+1))), NNAME(JDIV(2,J+1))
      DO 531 I=1,3
        WRITE ( *, 9006 ) I
531     WRITE ( *, 9008 ) (IS,CHCPD(I,IS,J), IS=1,NSS(I) )
      WRITE ( *,5002 )
      READ ( *,9001 ) ITIMES
      IF ( ITIMES .LE. 0 ) GO TO 550
C Determine if a whole row or whole column will be changed
            WRITE ( *,1004 )
            READ ( *,9002 ) ANS
            IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 545
C Corrections performed one at a time
```

```
      DO 544 NN=1,ITIMES
        WRITE ( *,5005 ) NNAME(JDIV(1,(J+1)))
        READ ( *,9001 ) IX
        WRITE ( *,5006 ) IX, NNAME(JDIV(1,(J+1)))
        READ ( *,9001 ) ISX
        WRITE ( *,5007 ) NNAME(JDIV(1,(J+1))), ISX, IX
        READ ( *,9003 ) DUMMY
        IF ( DUMMY .LE. 0 ) GO TO 544
          CHCPD( IX, ISX, J ) = DUMMY
544     CONTINUE
      GO TO 530
C Corrections performed for an individual season
545   WRITE ( *,5005 ) NNAME(JDIV(1,(J+1)))
      READ ( *,9001 ) IX
      DO 532 IS=1,NSS(IX)
        WRITE ( *,5007 ) NNAME(JDIV(1,(J+1))), IS, IX
        READ ( *,9003 ) DUMMY
        IF ( DUMMY .LE. 0 ) GO TO 532
          CHCPD( IX, IS, J ) = DUMMY
532   CONTINUE
      GO TO 530
550   IF ( INODE .LT. 99 ) GO TO 20
599   CONTINUE
      GO TO 10
C Review diversion canal sub-seasonal transportation losses as a
C percent of flow
600   DO 699 J1=1,JDIV(1,1)
        J = J1
      IF ( INODE .GE. 99 ) GO TO 630
      DO 625 J=1,JDIV(1,1)
        IF ( JDIV(1,J+1) .EQ. INODE ) GO TO 630
625   CONTINUE
C No DIVERSION was found at the node requested, try again
      WRITE ( *,1003 ) CHAR(IND-1), NNAME(INODE), INODE
      GO TO 20
630   WRITE ( *,6000 ) NNAME(JDIV(1,J+1)), NNAME(JDIV(2,J+1))
      DO 631 I=1,3
        WRITE ( *, 9006 )
631     WRITE ( *, 9008 ) (IS,CHLSS(I,IS,J), IS=1,NSS(I) )
      WRITE ( *,6002 )
      READ ( *,9001 ) ITIMES
      IF ( ITIMES .LE. 0 ) GO TO 650
C Determine if a whole row or whole column will be changed
        WRITE ( *,1004 )
        READ ( *,9002 ) ANS
        IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 645
C Corrections performed one at a time
      DO 644 NN=1,ITIMES
        WRITE ( *,6005 ) NNAME(JDIV(1,(J+1)))
        READ ( *,9001 ) IX
        WRITE ( *,6006 ) IX, NNAME(JDIV(1,(J+1)))
        READ ( *,9001 ) ISX
```

```
            WRITE   ( *,6007   ) NNAME(JDIV(1,(J+1))),          ISX,  IX
            READ    ( *,9003   ) DUMMY
            IF  ( DUMMY   .LE.  0 ) GO  TO  644
               CHLSS(   IX,  ISX,  J ) = DUMMY
644            CONTINUE
      GO  TO  630
C Corrections   performed   for  an  individual   season
645   WRITE   ( *,6005   ) NNAME(JDIV(1,(J+1)))
      READ    ( *,9001   ) IX
      DO  632  IS=1,NSS(IX)
         WRITE   ( *,6007   ) NNAME(JDIV(1,(J+1))),          IS,  IX
         READ    ( *,9003   ) DUMMY
         IF  ( DUMMY   .LE.  0 ) GO  TO  632
            CHLSS(   IX,  IS,  J ) = DUMMY
632      CONTINUE
      GO  TO  630
650   IF  ( INODE   .LT.  99 ) GO  TO  20
699   CONTINUE
      GO  TO  10
C Review   reservoir   storage   limits   for  each   sub-season
700   DO  799  J1=1,JRES(1)
            J  =  J1
      IF. ( INODE   .GE.  99 ) GO  TO  730
      DO  725  J=1,JRES(1)
        IF  ( JRES(J+1)   .EQ.  INODE   ) GO  TO  730
725   CONTINUE
C No  RESERVOIR   was  found  at  the  node  requested,   try  again
      WRITE   ( *,1003   ) CHAR(IND-1),    NNAME(INODE),    INODE
       GO  TO  20
730   WRITE   ( *,7000   ) NNAME(JRES(J+1))
      DO  731  I=1,3
          WRITE   ( *,  9007   ) I
731         WRITE   ( *,  9009   ) (IS,SMAX(I,IS,J),SMIN(I,IS,J),         IS=1,NSS(I)      )
      WRITE   ( *,7002   )
      READ    ( *,9001   ) ITIMES
      IF  ( ITIMES   .LE.  0 ) GO  TO  750
C Determine   if  a  whole   row  or  whole   column   will  be  changed
            WRITE   ( *,1004   )
            READ    ( *,9002   ) ANS
            IF  ( ANS   .EQ.  'Y'  .OR.  ANS   .EQ.  'y'  ) GO  TO  745
C Corrections   performed   one  at  a  time
          DO  744  NN=1,ITIMES
            WRITE   ( *,7005   ) NNAME(JRES(J+1))
            READ    ( *,9001   ) IX
            WRITE   ( *,7006   ) IX,  NNAME(JRES(J+1))
            READ    ( *,9001   ) ISX
            WRITE   ( *,7007   ) NNAME(JRES(J+1)),        ISX,  IX
            READ    ( *,9003   ) DUMMY
            IF  ( DUMMY   .LE.  0 ) GO  TO  743
               SMAX(   IX,  ISX,  J ) = DUMMY
743            WRITE   ( *,7008   ) NNAME(JRES(J+1)),        ISX,  IX
            READ    ( *,9003   ) DUMMY
```

```
            IF ( DUMMY .LE. 0 ) GO TO 744
               SMIN( IX, ISX, J ) = DUMMY
744         CONTINUE
        GO TO 730
C Corrections performed for an individual season
745     WRITE ( *,7005 ) NNAME(JRES(J+1))
        READ ( *,9001 ) IX
        DO 748 IS=1,NSS(IX)
          WRITE ( *,7007 ) NNAME(JRES(J+1)), IS, IX
          READ ( *,9003 ) DUMMY
          IF ( DUMMY .LE. 0 ) GO TO 747
             SMAX( IX, IS, J ) = DUMMY
747       WRITE ( *,7008 ) NNAME(JRES(J+1)), IS, IX
          READ ( *,9003 ) DUMMY
          IF ( DUMMY .LE. 0 ) GO TO 748
             SMIN( IX, IS, J ) = DUMMY
748     CONTINUE
      GO TO 730
750   IF ( INODE .LT. 99 ) GO TO 20
799   CONTINUE
      GO TO 10
C Read in previous file of constants
800   WRITE ( *,1007 )
      READ ( *,9005 ) FNAME
      IF ( FNAME .EQ. '          ' ) GO TO 10
805   OPEN ( 10,FILE=FNAME, STATUS='OLD')
      READ ( 10,9100 ) N1, N2, N3, NN, ( TITLE(M), M=1,18 )
      IF ( N1 .NE. NSS(1) .OR. N2 .NE. NSS(2) .OR. N3 .NE.NSS(3) )
     1                GO TO 888
      IF ( NN .NE. JN ) GO TO 889
      WRITE (*,1005) ( TITLE(M), M=1,18 )
      READ (*,9002) ANS
        IF ( ANS .EQ. 'y' .OR. ANS .EQ. 'Y' ) GO TO 10
      DO 844 N=1,IRT
844   READ ( 10,9101 ) ( (P1(I,IS,N), IS=1,NSS(I)), I=1,3 )
      DO 845 N=1,NLIM(1)-1
845   READ (10,9101) ((FLMIN(I,IS,N),FLMAX(I,IS,N),IS=1,NSS(I)), I=1,3)
      DO 846 N=1,JIRR(1)
846   READ ( 10,9101 ) ( (CHCPI(I,IS,N), IS=1,NSS(I)), I=1,3 )
      DO 847 N=1,JDIV(1,1)
847   READ (10,9101) ((CHCPD(I,IS,N),CHLSS(I,IS,N),IS=1,NSS(I)),I=1,3)
      DO 848 N=1,JRES(1)
848   READ (10,9101) ((SMIN(I,IS,N),SMAX(I,IS,N), IS=1,NSS(I)), I=1,3)
      CLOSE (10)
C Ask if the user wishes to use the information as read or to advance the
C data one season forward
      WRITE ( *,1009 ) TITLE(1)
      READ ( *,9002 ) ANS
      IF ( ANS .EQ. 'N' .OR. ANS .EQ. 'n' ) GO TO 850
        GO TO 10
C First season now MAHA, second YALA or vice versa
850   DO 855 I=1,2
```

```
      DO 855 IS=1,NSS(I)
        DO 851 N=1,IRT
851       P1(I,IS,N) = P1(I+1,IS,N)
        DO 852 N=1,NLIM(1)-1
          FLMIN(I,IS,N) = FLMIN(I+1,IS,N)
852       FLMAX(I,IS,N) = FLMAX(I+1,IS,N)
        DO 853 N=1,JIRR(1)
853       CHCPI(I,IS,N) = CHCPI(I+1,IS,N)
        DO 854 N=1,JDIV(1,1)
          CHCPD(I,IS,N) = CHCPD(I+1,IS,N)
854       CHLSS(I,IS,N) = CHLSS(I+1,IS,N)
        DO 855 N=1,JRES(1)
          SMIN(I,IS,N) = SMIN(I+1,IS,N)
          SMAX(I,IS,N) = SMAX(I+1,IS,N)
855       CONTINUE
C Set last season's values equal to first
      DO 865 IS=1,NSS(3)
        DO 861 N=1,IRT
861       P1(3,IS,N) = P1(1,IS,N)
        DO 862 N=1,NLIM(1)-1
          FLMIN(3,IS,N) = FLMIN(1,IS,N)
862       FLMAX(3,IS,N) = FLMAX(1,IS,N)
        DO 863 N=1,JIRR(1)
863       CHCPI(3,IS,N) = CHCPI(1,IS,N)
        DO 864 N=1,JDIV(1,1)
          CHCPD(3,IS,N) = CHCPD(1,IS,N)
864       CHLSS(3,IS,N) = CHLSS(1,IS,N)
        DO 865 N=1,JRES(1)
          SMIN(3,IS,N) = SMIN(1,IS,N)
          SMAX(3,IS,N) = SMAX(1,IS,N)
865       CONTINUE
C Change season on file's titlecard
      IF ( TITLE(1) .EQ. 'MAHA' ) NAME1 = 'YALA'
      IF ( TITLE(1) .EQ. 'YALA' ) NAME1 = 'MAHA'
        TITLE(1) = NAME1
      GO TO 10
C The number of sub-seasons on file do not agree with the current values
888   WRITE ( *,1010 ) N1, NSS(1), N2, NSS(2), N3, NSS(3)
    GO TO 999
C The number of nodes on file is not the same as the current number
889   WRITE ( *,1011 ) NN, JN
    GO TO 999
999   CONTINUE
C
1000  FORMAT ( ' Do the system constants CHANGE?'/' Enter 8 to enter the
     1 file containing any changes'/7X,'7 to review reservoir storage li
     2mits'/ 7X,'6 to review diversion canal losses as a percent of flow
     3'/ 7X,'5 to review diversion canal sub-seasonal capacities'/ 7X,'4
     4 to review irrigation canal sub-seasonal capacities'/ 7X,'3 to rev
     5iew the sub-seasonal downstream flow restrictions'/ 7X,'2 to revie
     6w the percent of irrigation deliveries which returns elsewhere'/
     77X,'1 to save any changes on an existing file'/ 7X,'0 or nothing t
```

```
     8o   return   to   the   main   menu'/'   Enter   an   INTEGER   ' )
1001   FORMAT   ( I3,5X,A12,I3,5X,A12,I3,5X,A12,I3,5X,A12   )
1002   FORMAT   ( ' What   is   the   number   for   the   specific   ',A13,'   you   wish   to
     1   review?'/'   Enter   99   if   you   wish   to   see   the   values   for   every   node
     2or   one   INTEGER'/'   If   you   enter   0   or   nothing   you   will   return   to   th
     3e   CHANGES   menu   ' )
1003   FORMAT   ( ' There   was   no   ',A13,'   found   at   the   node   ',A12,'   the   numb
     1er   ',I3   )
1004   FORMAT   ( ' Do   you   wish   to   correct   all   the   values   for   one   season?
     1Answer   Y   if   you   do   ' )
1005   FORMAT   ( ' The   file   to   be   read   has   the   following   title   card:'/1H
     118A4/'   Do   you   wish   to   enter   another   file?   Answer   Y   to   change   the
     2diskette   in   the   drive   or   the   filename   ' )
1006   FORMAT   ( ' The   current   title   card   for   the   file   is:'/1H   ,18A4/'   Do
     1you   wish   to   enter   another   title?   If   you   enter   nothing   before   the
     2return,   the   above   title   card   will   be   used.   Enter   a   title   in   80   or
     3   less   characters,   the'/'   first   4   spaces   are   reserved   for   the   name
     4of   the   first   season   in   capital   letters   ' )
1007   FORMAT   ( ' What   is   the   drive   and   name   of   the   file   which   stores   the
     1   sub-seasonal   variations?   Enter   b:FNAME   where   b:   is   the   drive   and
     2FNAME   is   the   name   of   the   file   in   10   or   less   characters   ' )
1008   FORMAT   ( ' Do   you   wish   to   save   the   updated   version   under   the   same
     1filename,   ',A15,'Enter   b:FNAME   where   b:   specifies   the   drive   and   FN
     2AME   the   new   name   in   10   or   less   letters   or   nothing   to   keep   the   same
     3   name.   ' )
1009   FORMAT   ( ' Do   you   wish   to   start   with   the   values   as   read,   with   the
     1',A4,'   season   first?'/'   Answer   N   to   advance   the   values   one   season.
     2   The   new   values   for   the   third   season   will   be   the   same   as   the   new
     3values   for   the   first   season   ' )
1010   FORMAT   ( ' The   file   has   ',I3,'   sub-seasons   in   the   first   season,   wh
     1ile   the   current   run   has   ',2I4,'   in   the   second,   v/s   the   current   ',I
     23,'   and   ',I3,'   sub-seasons   in   the   last   and   not   ',I3,'.   Please   chec
     3k   you   have   the   right   file   and   the   right   numbers   of   sub-seasons.'   )
1011   FORMAT   ( ' The   file   has   ',I3,'   nodes   listed   while   the   current   conf
     1iguration   has   ',I3/'   Please   check   you   have   the   right   file   and   the
     2right   number   of   nodes.   ' )
2000   FORMAT   ( ' These   are   the   percentage   of   irrigation   flows   from   ',A12
     1,'   which   arrive'/'   at   ',A12,'   in   the   same   sub-season:'   )
2002   FORMAT   ( ' How   many   of   the   percentages   would   you   like   to   revise?   '
     1   )
2005   FORMAT   ( ' What   is   the   INTEGER   of   the   season   which   has   an   incorrec
     1t   percentage   of   the'/'   return   flow   at   ',A12   )
2006   FORMAT   ( ' What   is   the   INTEGER   of   the   sub-season   in   the   season   ',I
     13,'   which   has   an   incorrect   percentage?   ' )
2007   FORMAT   ( ' What   is   the   correct   percentage   of   the   irrigation   flows
     1from   the   reservoir   ',A12,'   which   arrive   during   the   sub-season',I3,
     2'   of   season   ',I3/'   If   you   enter   0   or   nothing,   nothing   will   change'
     3/'   Enter   0.00001   to   enter   a   zero   value   ' )
3000   FORMAT   ( ' These   are   the   minimum   and   maximum   allowed   downstream   fl
     1ows   from   the   node   '/   1X,A12,'   during   the   sub-seasons:'   )
3002   FORMAT   ( ' How   many   of   the   restricted   downstream   flows   would   you   l
     1ike   to   revise?   ')
```

```
3005    FORMAT    ( ' What is the INTEGER of the season which has an incorrec
     1t restriction at the node',A12 )
3006    FORMAT    ( ' What is the INTEGER of the sub-season in the season ',I
     13,' which has an incorrect restriction at the node ',A12 )
3007    FORMAT    ( ' What is the correct value for the MAXIMUM allowed relea
     1se from the node ',A12,' during the sub-season',I3,' of season',I3
     2,' in Mm3? '/' If you enter 0 or nothing, the value will not chang
     3e'/' Enter 0.00001 to enter a zero value ' )
3008    FORMAT    ( ' What is the correct value for the MINIMUM allowed relea
     1se from the node ',A12,' during the sub-season',I3,' of season',I3
     2,' in Mm3? '/' If you enter 0 or nothing, the value will not chang
     3e'/' Enter 0.00001 to enter a zero value ' )
4000    FORMAT    ( ' These are the maximum volumes in Mm3 which may be deliv
     1ered to the irrigation at '/ 1X,A12,' during each sub-season:' )
4002    FORMAT    ( ' How many of the volume limits would you like to revise?
     1 ')
4005    FORMAT    ( ' What is the INTEGER of the season which has an incorrec
     1t capacity for the'/' irrigation district ',A12 )
4006    FORMAT    ( ' What is the INTEGER of the sub-season in the season ',I
     13,' which has an incorrect'/' capacity? ' )
4007    FORMAT    ( ' What is the correct sub-seasonal capacity for the irrig
     1ation deliveries of ',A12,' during the sub-season',I3,' of season
     2',I3,' in Mm3? '/' If you enter 0 or nothing, the value will not c
     3hange'/' Enter 0.00001 to enter a zero value ' )
5000    FORMAT    ( ' These are the maximum volumes in Mm3 which may be deliv
     1ered by the canal from '/1X,A12,' to ',A12,' in the sub-season:' )
5002    FORMAT    ( ' How many of the values would you like to revise? ' )
5005    FORMAT    ( ' What is the INTEGER of the season which has an incorrec
     1t capacity for the'/' diversion from ',A12 )
5006    FORMAT    ( ' What is the INTEGER of the sub-season in the season ',I
     13,' which has an incorrect'/' capacity for the diversion from ',
     2 A12 )
5007    FORMAT    ( ' What is the correct sub-seasonal capacity of the divers
     1sion from ',A12/' during the sub-season',I3,' of season ',I3,' in
     2Mm3?'/' If you enter 0 or nothing, nothing will change'/' Enter 0
     3.00001 to enter a zero value ' )
6000    FORMAT    ( ' These are the channel losses as a percent of flow in th
     1e canal from '/1X,A12,' to ',A12,' in each sub-season:' )
6002    FORMAT    ( ' How many of the losses would you like to revise? ' )
6005    FORMAT    ( ' What is the INTEGER of the season which has an incorrec
     1t loss for the'/' diversion from ',A12 )
6006    FORMAT    ( ' What is the INTEGER of the sub-season in the season ',I
     13,' which has an incorrect loss for the diversion from ',A12 )
6007    FORMAT    ( ' What is the correct sub-seasonal loss as a percent of f
     1low in the canal from ',A12,' during the sub-season',I3,' of seaso
     2n ',I3/' If you enter 0 or nothing, nothing will change'/' Enter 0
     3.00001 to enter a zero value ' )
7000    FORMAT    ( ' These are the minimum and maximum storage volumes allow
     1ed in the reservoir '/ 1X,A12,' during the sub-seasons: ' )
7002    FORMAT    ( ' How many of the storage volume bounds would you like to
     1 revise? ')
7003    FORMAT    ( ' Do you wish to correct all the values for one season?
```

```
     1Answer Y if you do ' )
7005 FORMAT ( ' What is the INTEGER of the season which has an incorrec
     1t bound for the reservoirat '.A12 )
7006 FORMAT ( ' What is the INTEGER of the sub-season in the season ',I
     13,' which has an incorrect bound for the reservoir at '.A12 )
7007 FORMAT ( ' What is the correct bound for the MAXIMUM allowed stora
     1ge in the reservoir '.A12,' during the sub-season',I3,' of season
     2',I3,' in Mm3?' '/' If you enter 0 or nothing, the bound will not c
     3hange'/' Enter 0.00001 to enter a zero value ' )
7008 FORMAT ( ' What is the correct bound for the MINIMUM allowed stora
     1ge in the reservoir '.A12,' during the sub-season',I3,' of season
     2',I3,' in Mm3? '/' If you enter 0 or nothing, the bound will not c
     3hange'/' Enter 0.00001 to enter a zero value ' )
9001 FORMAT ( I2 )
9002 FORMAT ( A2 )
9003 FORMAT ( F10.0 )
9004 FORMAT ( 18A4 )
9005 FORMAT ( A12 )
9006 FORMAT ( /5X,6HSEASON,2X,I3/ )
9007 FORMAT ( /5X,6HSEASON,2X,I3,9X,7HMAXIMUM,8X,7HMINIMUM/ )
9008 FORMAT ( 3X,10Hsub-season,I3,6X,F9.2 )
9009 FORMAT ( 3X,10Hsub-season,I3,6X,F9.2,6X,F9.2 )
9100 FORMAT ( 4I2,18A4 )
9101 FORMAT ( 8F10.3 )
      RETURN
      END
^Z
      SUBROUTINE CNT ( JN,IRT )
C This subroutine defines the FLOW() vectors which designate nodes
C which flow to nodes numbered one greater. Then the variables which
C appear in the continuity equations are written to the diskette.
      COMMON/SHC/ JIRR(21), JRES(25), JNOSPL(20), JDIV(2,15)
      COMMON/SHP/ JTRB(2,10), SLO(25), SUP(25), J1(2,10), C1(10)
      COMMON/NOM/ JNAME(40)
      COMMON/FLO/ NLIM(21), NSS(3)
      COMMON/VAR/FLMIN(3,5,20), FLMAX(3,5,20), CHCPI(3,5,21), P1(3,5,10)
     1 ,CHCPD(3,5,15), CHLSS(3,5,15), SMAX(3,5,25), SMIN(3,5,25)
      INTEGER FLOW(30)
      CHARACTER * 150 LFD(16), LFF(8), LF
      CHARACTER * 12 JNAME
      CHARACTER * 2 ANS
c LFD = array of format lines for writing flow variables, QDnSiT
c                                         diversion variables, QDnSiT
c                                     and irrigation variables, QRnSiT
c       according to the magnitude of the node number
      DATA LFD/'( 2H .4H Q,I1,1HS,2I1,T18,3HCNT,I1,1HS,2I1,3H 1,T37,
     13HCNT,I1,1HS,2I1,4H -1,T58,1H.,5X,1H. )',
     2 '( 2H .4H Q,I1,1HS,2I1,T18,3HCNT,I1,1HS,2I1,3H 1,T37,
     23HCNT,I2,1HS,2I1,4H -1,T58,1H.,5X,1H. )',
     3 '( 2H .4H Q,I2,1HS,2I1,T18,3HCNT,I2,1HS,2I1,3H 1,T37,
     33HCNT,I1,1HS,2I1,4H -1,T58,1H.,5X,1H. )',
     4 '( 2H .4H Q,I2,1HS,2I1,T18,3HCNT,I2,1HS,2I1,3H 1,T37,
```

```
     43HCNT,I2,1HS,2I1,4H                  -1,T58,1H.,5X,1H.          )',
     5        '(  2H   .,4H   QD,I1,1HS,2I1,T18,3HCNT,I1,1HS,2I1,3H                1,T37,
     53HCNT,I1,1HS,2I1,F9.4,T56,6HUPPER                  ,F10.3    )',
     6        '(  2H   .,4H   QD,I1,1HS,2I1,T18,3HCNT,I1,1HS,2I1,3H                1,T37,
     63HCNT,I2,1HS,2I1,F9.4,T56,6HUPPER                  ,F10.3    )',
     7        '(  2H   .,4H   QD,I2,1HS,2I1,T18,3HCNT,I2,1HS,2I1,3H                1,T37,
     73HCNT,I1,1HS,2I1,F9.4,T56,6HUPPER                  ,F10.3    )',
     8        '(  2H   .,4H   QD,I2,1HS,2I1,T18,3HCNT,I2,1HS,2I1,3H                1,T37,
     83HCNT,I2,1HS,2I1,F9.4,T56,6HUPPER                  ,F10.3    )',
     9        '(  2H   .,4H   QD,I1,1HS,2I1,T18,3HCNT,I1,1HS,2I1,3H                1,T37,
     93HCNT,I1,1HS,2I1,F9.4,T58,1H.,6X,1H.                 )',
     1        '(  2H   .,4H   QD,I1,1HS,2I1,T18,3HCNT,I1,1HS,2I1,3H                1,T37,
     13HCNT,I2,1HS,2I1,F9.4,T58,1H.,6X,1H.                 )',
     1        '(  2H   .,4H   QD,I2,1HS,2I1,T18,3HCNT,I2,1HS,2I1,3H                1,T37,
     13HCNT,I1,1HS,2I1,F9.4,T58,1H.,6X,1H.                 )',
     2        '(  2H   .,4H   QD,I2,1HS,2I1,T18,3HCNT,I2,1HS,2I1,3H                1,T37,
     23HCNT,I2,1HS,2I1,F9.4,T58,1H.,6X,1H.                 )',
     3        '(  2H   .,4H   QR,I1,1HS,2I1,T18,3HCNT,I1,1HS,2I1,2X,F8.3,T4
     30,1H.,7X,1H.,8X,1H.,5X,1H.            )',
     4        '(  2H   .,4H   QR,I1,1HS,2I1,T18,3HCNT,I2,1HS,2I1,2X,F8.3,T3
     49,1H.,7X,1H.,8X,1H.,5X,1H.            )',
     5        '(  2H   .,4H   QR,I2,1HS,2I1,T18,3HCNT,I1,1HS,2I1,2X,F8.3,T3
     59,1H.,7X,1H.,8X,1H.,5X,1H.            )',
     6        '(  2H   .,4H   QR,I2,1HS,2I1,T18,3HCNT,I2,1HS,2I1,2X,F8.3,T3
     69,1H.,7X,1H.,8X,1H.,5X,1H.            )'/
c    LFF   =  array   of  format   lines   for  writing   flow   variables,    QnSiT,
c          with   downstream    flow   restrictions    according    to  the   magnitude
c          of  the   node    numbers
      DATA    LFF/'(   2H   .,4H    Q,I1,1HS,2I1,T18,3HCNT,I1,1HS,2I1,3H              1,T37,
     13HCNT,I1,1HS,2I1,4H                 -1,T55,6HLOWER          ,F10.3    )',
     2        '(  2H   .,4H   Q,I1,1HS,2I1,T18,3HCNT,I1,1HS,2I1,3H                 1,T37,
     23HCNT,I1,1HS,2I1,4H                 -1,T55,6HUPPER          ,F10.3    )',
     3        '(  2H   .,4H   Q,I1,1HS,2I1,T18,3HCNT,I1,1HS,2I1,3H                 1,T37,
     33HCNT,I2,1HS,2I1,4H                 -1,T55,6HLOWER          ,F10.3    )',
     4        '(  2H   .,4H   Q,I1,1HS,2I1,T18,3HCNT,I1,1HS,2I1,3H                 1,T37,
     43HCNT,I2,1HS,2I1,4H                 -1,T55,6HUPPER          ,F10.3    )',
     5        '(  2H   .,4H   Q,I2,1HS,2I1,T18,3HCNT,I2,1HS,2I1,3H                 1,T37,
     53HCNT,I1,1HS,2I1,4H                 -1,T55,6HLOWER          ,F10.3    )',
     6        '(  2H   .,4H   Q,I2,1HS,2I1,T18,3HCNT,I2,1HS,2I1,3H                 1,T37,
     63HCNT,I1,1HS,2I1,4H                 -1,T55,6HUPPER          ,F10.3    )',
     7        '(  2H   .,4H   Q,I2,1HS,2I1,T18,3HCNT,I2,1HS,2I1,3H                 1,T37,
     73HCNT,I2,1HS,2I1,4H                 -1,T55,6HLOWER          ,F10.3    )',
     8        '(  2H   .,4H   Q,I2,1HS,2I1,T18,3HCNT,I2,1HS,2I1,3H                 1,T37,
     83HCNT,I2,1HS,2I1,4H                 -1,T55,6HUPPER          ,F10.3    )'/
C
C    Definition    of  FLOW    vector    containing    the   nodes    whose    spills   flow
C    naturally     to  nodes    numbered      one   greater.
             L  =  2
            IN  =  1
       DO   4  N=1,JN
         IF  ( JNOSPL(L)     .EQ.   N  )  GO   TO   3
         FLOW(IN)       =  N
```

```
            IN = IN + 1
          GO TO 4
3           L = 1 + L
4       CONTINUE
C   Write variables and equations for nodes with reservoirs
        DO 20 L=1,JRES(1)
            N = JRES(L+1)
C   Write FSnSi, storage volume variable, in the UPB, LOW, CNTnSi,nss(i) and
C                                       CNTnSi+1,1 constraints
          IF ( SUP(L+1) .GE. 99999 ) GO TO 15
          IF ( N .GT. 9 ) GO TO 13
C   The node number takes one digit to write
          WRITE ( 8,1001 ) ( N,I,N,I,NSS(I), N,(I+1),SMAX(I,NSS(I),L),
     1                    N,I, SMIN(I,NSS(I),L), I=1,2 )
          WRITE ( 8,1003 ) N, N,NSS(3), SMAX(I,NSS(3),L), SMIN(I,NSS(3),L)
C   Write SnSi,is, the sub-seasonal storage variable in the same constraints
            DO 12 I=1,3
            DO 12 IS=1,( NSS(I)-1 )
               ISP = IS + 1
12          WRITE ( 8,1002 ) N,I,IS, N,I,IS, N,I,ISP, SMAX(I,IS,L),
     1                    N,I,IS, SMIN(I,IS,L)
          GO TO 20
C   The node number takes two digits to write
13        WRITE ( 8,1005 ) ( N,I,N,I,NSS(I), N,(I+1),SMAX(I,NSS(I),L),
     1                    N,I, SMIN(I,NSS(I),L), I=1,2 )
          WRITE ( 8,1007 ) N, N,NSS(3), SMAX(I,NSS(3),L), SMIN(I,NSS(3),L)
C   Write SnSi,is, the sub-seasonal storage variable in the same constraints
            DO 132 I=1,3
            DO 132 IS=1,( NSS(I)-1 )
               ISP = IS + 1
132         WRITE ( 8,1006 ) N,I,IS, N,I,IS, N,I,ISP, SMAX(I,IS,L),
     1                    N,I,IS, SMIN(I,IS,L)
          GO TO 20
C   The storage volume is unbounded ABOVE and the node number may be
C   written in one digit
15          IF ( N .GT. 9 ) GO TO 18
          WRITE (8,1004) ( N,I,N,I,NSS(I),N,(I+1),SMIN(I,NSS(I),L),I=1,2 )
          WRITE (8,1008) N, N,NSS(3), SMIN(I,NSS(3),L)
C   Write SnSi,is, the sub-seasonal storage variable in the same constraints
            DO 152 I=1,3
            DO 152 IS=1,NSS(I)-1
               ISP = IS + 1
152         WRITE ( 8,1009 ) N,I,IS, N,I,IS, N,I,ISP, SMIN(I,IS,L)
          GO TO 20
C   The node number takes two digits to write
18        WRITE (8,1011) ( N,I,N,I,NSS(I),N,(I+1),SMIN(I,NSS(I),L),I=1,2 )
          WRITE (8,1012) N, N,NSS(3), SMIN(I,NSS(3),L)
C   Write SnSi,is, the sub-seasonal storage variable in the same constraints
            DO 182 I=1,3
            DO 182 IS=1,NSS(I)-1
               ISP = IS + 1
182         WRITE ( 8,1013 ) N,I,IS, N,I,IS, N,I,ISP, SMIN(I,IS,L)
```

```
20      CONTINUE
C   Write variables and equations of the nodes with irrigation diversion
C   Here also the equations defining the seasonal irrigation demand are
c   written to the diskette, and any return flows.
        DO 31 L = 1, JIRR(1)
            N = JIRR(L+1)
          IF ( N .EQ. JIRR(L-1) ) GO TO 31
          DO 30 I = 1,3
          DO 30 IS=1,NSS(I)
          IF ( CHCPI(I,IS,L) .GE. 99999 ) GO TO 25
            IF ( N .GT. 9 ) GO TO 23
        WRITE (8,2001) N,I,IS, N,I,IS, N,I,IS, CHCPI(I,IS,L)
          GO TO 30
23      WRITE (8,2005) N,I,IS, N,I,IS, N,I,IS, CHCPI(I,IS,L)
          GO TO 30
C   There is unlimited irrigation canal capacity
25        IF ( N .GT. 9 ) GO TO 28
          WRITE ( 8,2002 ) N,I,IS, N,I,IS, N,I,IS
          GO TO 30
28        WRITE ( 8,2006 ) N,I,IS, N,I,IS, N,I,IS
30        CONTINUE
31        CONTINUE
c   Write the flow varialbes when there are irrigation return flows in
c   the water balance equations
        IF ( IRT .LE. 0 ) GO TO 33
        DO 32 L=1,IRT
        DO 32 I=1,3
        DO 32 IS=1,NSS(I)
            IND = 13
          IF ( J1(1,L) .GT. 9 ) IND = 15
          IF ( J1(2,L) .GT. 9 ) IND = IND + 1
            LF = LFD(IND)
32        WRITE ( 8,LF ) J1(1,L),I,IS, J1(2,L),I,IS, -P1(I,IS,L)
C   Write the variables representing natural flows between nodes
33      DO 40 L = 1, IN-1
        DO 40 I=1,3
        DO 40 IS=1,NSS(I)
            N = FLOW(L)
        IF ( NLIM(1) .LE. 0 ) GO TO 37
C   Check if there are flow restrictions at this node
        DO 35 NN = 2,NLIM(1)
          IF ( NLIM(NN) .EQ. N ) GO TO 39
35      CONTINUE
C   There are no flow restrictions at the node N
37          IND = 1
          IF ( N .GT. 9 ) IND = 3
          IF ( N+1 .GT. 9 ) IND = IND + 1
            LF = LFD(IND)
          WRITE ( 8,LF ) N,I,IS, N,I,IS, N+1,I,IS
            GO TO 40
C   There are flow restrictions at this node
39              IND = 1
```

```
      IF ( N+1 .GT. 9 ) IND = IND + 2
      IF ( N .GT. 9 ) IND = IND + 4
      IF ( FLMAX(I,IS,NN-1) .GE. 99999.0 ) GO TO 391
      IF ( FLMIN(I,IS,NN-1) .LE. 0.0 ) GO TO 392
C There are restrictions on both the maximum and minimum sub-seasonal
C downstream releases
         LF = LFF(IND)
      WRITE ( 8,LF ) N,I,IS, N,I,IS, N+1,I,IS, FLMIN(I,IS,NN-1)
         IF ( N .GT. 9 ) GO TO 395
      WRITE ( 8,4000 ) N,I,IS, FLMAX(I,IS,NN-1)
         GO TO 40
395   WRITE ( 8,4001 ) N,I,IS, FLMAX(I,IS,NN-1)
         GO TO 40
C There is NO upperbound on the downstream releases, there is only a
c restriction on the minimum sub-seasonal flow
391      LF = LFF(IND)
      WRITE ( 8,LF ) N,I,IS, N,I,IS, N+1,I,IS, FLMIN(I,IS,NN-1)
         GO TO 40
C There is only a sub-seasonal upper bound on the downstream releases
392      IND = IND + 1
         LF = LFF(IND)
      WRITE ( 8,LF ) N,I,IS, N,I,IS, N+1,I,IS, FLMAX(I,IS,NN-1)
40    CONTINUE
c Write the variables when there are tributary flow in the continuity
c   equations
      DO 50 L = 2, JTRB(1,1)+1
      DO 50 I=1,3
      DO 50 IS=1,NSS(I)
         N = JTRB(1,L)
      IF ( NLIM(1) .LE. 0 ) GO TO 42
C Check if there are flow restrictions at this node
      DO 44 NN = 2,NLIM(1)
      IF ( NLIM(NN) .EQ. N ) GO TO 45
44    CONTINUE
C There are no flow restrictions at the node N
42       IND = 1
      IF ( N .GT. 9 ) IND = 3
      IF ( JTRB(2,L) .GT. 9 ) IND = IND + 1
         LF = LFD (IND)
      WRITE ( 8, LF ) N,I,IS, N,I,IS, JTRB(2,L),I,IS
         GO TO 50
C There are flow restrictions at this node
45       IND = 1
      IF ( JTRB(2,L) .GT. 9 ) IND = IND + 2
      IF ( N .GT. 9 ) IND = 4 + IND
      IF ( FLMAX(I,IS,NN-1) .GE. 99999.0 ) GO TO 47
      IF ( FLMIN(I,IS,NN-1) .LE. 0.0 ) GO TO 48
C There are restrictions on both the maximum and minimum sub-seasonal
C downstream releases
         LF = LFF(IND)
      WRITE ( 8,LF ) N,I,IS, N,I,IS, JTRB(2,L),I,IS, FLMIN(I,IS,NN-1)
      IF ( N .GT. 9 ) GO TO 46
```

```
      WRITE ( 8,4000 ) N,I,IS, FLMAX(I,IS,NN-1)
         GO TO 50
46     WRITE ( 8,4001 ) N,I,IS, FLMAX(I,IS,NN-1)
         GO TO 50
C There is NO upperbound on the downstream releases, there is only a
c restriction on the minimum sub-seasonal flow
47       LF = LFF(IND)
      WRITE ( 8,LF ) N,I,IS, N,I,IS, JTRB(2,L),I,IS, FLMIN(I,IS,NN-1)
         GO TO 50
C There is only a sub-seasonal upper bound on the downstream releases
48       IND = IND + 1
         LF = LFF(IND)
      WRITE ( 8,LF ) N,I,IS, N,I,IS, JTRB(2,L),I,IS, FLMAX(I,IS,NN-1)
50     CONTINUE
C Write the variables representing diverted flows
      DO 60 L = 1,JDIV(1,1)
         N = JDIV(1,L+1)
      DO 60 I=1,3
      DO 60 IS=1,NSS(I)
         CHLSS(I,IS,L) = CHLSS(I,IS,L) - 1.00
       IF ( CHCPD(I,IS,L) .GE. 99999 ) GO TO 55
C        CCAPD(L) = CCAPD(L) * 1000
        IND = 5
       IF ( N .GT. 9 ) IND = 7
       IF ( JDIV(2,L+1) .GT. 9 ) IND = IND + 1
         LF = LFD (IND)
      WRITE ( 8, LF ) N,I,IS, N,I,IS, JDIV(2,L+1),I,IS, CHLSS(I,IS,L),
     1                 CHCPD(I,IS,L)
       GO TO 60
C There is unbounded diversion capacity
55       IND = 9
       IF ( N .GT. 9 ) IND = 11
       IF ( JDIV(2,L+1) .GT. 9 ) IND = IND + 1
         LF = LFD (IND)
      WRITE ( 8, LF ) N,I,IS, N,I,IS, JDIV(2,L+1),I,IS, CHLSS(I,IS,L)
60     CONTINUE
C All the interactive data-entry and verification is over, tell the
C user to go for coffee.
      WRITE ( *,1010 )
C
1001   FORMAT ( 2H .,4H FS,I1,1HS,I1,T18,3HCNT,I1,1HS,2I1,3H 1,T37,
     13HCNT,I1,1HS,I1,1H1,4H -1,T56,6HUPPER ,F11.3/2H .,4H FS,I1,1HS,
     2I1,T18,6HLOWER ,F8.3,T39,1H.,5X,1H.,T58,1H.,5X,1H. )
1002   FORMAT ( 2H .,3H S,I1,1HS,2I1,T18,3HCNT,I1,1HS,2I1,3H 1,T37,
     13HCNT,I1,1HS,2I1,4H -1,T56,6HUPPER ,F11.3/2H .,3H S,I1,1HS,2I1,
     2T18,6HLOWER ,F8.3,T39,1H.,5X,1H.,T58,1H.,5X,1H. )
1003   FORMAT ( 2H .,4H FS,I1,2HS3,T18,3HCNT,I1,2HS3,I1,3H 1,T37,
     16HUPPER ,F12.3,T56,6HLOWER ,F8.3 )
1004   FORMAT ( 2H .,4H FS,I1,1HS,I1,T18,3HCNT,I1,1HS,2I1,3H 1,T37,
     13HCNT,I1,1HS,I1,1H1,4H -1,T56,6HLOWER ,F11.3 )
1005   FORMAT ( 2H .,4H FS,I2,1HS,I1,T18,3HCNT,I2,1HS,2I1,3H 1,T37,
     13HCNT,I2,1HS,I1,1H1,4H -1,T56,6HUPPER ,F11.3/2H .,4H FS,I2,1HS,
```

```
     2I1,T18,6HLOWER          ,F8.3,T39,1H.,5X,1H.,T58,1H.,5X,1H.              )
1006    FORMAT    ( 2H   .,3H    S,I2,1HS,2I1,T18,3HCNT,I2,1HS,2I1,3H                1,T37,
     13HCNT,I2,1HS,2I1,4H             -1,T56,6HUPPER          ,F11.3/2H      .,3H    S,I2,1HS,2I1,
     2T18,6HLOWER          ,F8.3,T39,1H.,5X,1H.,T58,1H.,5X,1H.              )
1007    FORMAT    ( 2H   .,4H    FS,I2,2HS3,T18,3HCNT,I2,2HS3,I1,3H               1,T37,
     16HUPPER        ,F12.3,T56,6HLOWER          ,F8.3    )
1008    FORMAT    ( 2H   .,4H    FS,I1,2HS3,T18,3HCNT,I1,2HS3,I1,3H               1,T37,
     16HLOWER        ,F12.3,T58,1H.,7X,1H.              )
1009    FORMAT    ( 2H   .,3H    S,I1,1HS,2I1,T18,3HCNT,I1,1HS,2I1,3H                1,T37,
     13HCNT,I1,1HS,2I1,4H             -1,T56,6HLOWER          ,F11.3     )
1010    FORMAT    ( ' Now    printing    to   the    file    b:PLAN.DATA.        Use    this    fi
     1le   for   the    SAS.OR     PROC    LPsparsedata        datafile     to   select     the    number
     2  of   hectares     to    plant    in   each    crop    in   each    district.    ' )
1011    FORMAT    ( 2H   .,4H    FS,I2,1HS,I1,T18,3HCNT,I2,1HS,2I1,3H                1,T37,
     13HCNT,I2,1HS,I1,1H1,4H             -1,T56,6HLOWER          ,F11.3     )
1012    FORMAT    ( 2H   .,4H    FS,I2,2HS3,T18,3HCNT,I2,2HS3,I1,3H               1,T37,
     16HLOWER        ,F12.3,T58,1H.,7X,1H.              )
1013    FORMAT    ( 2H   .,3H    S,I2,1HS,2I1,T18,3HCNT,I2,1HS,2I1,3H                1,T37,
     13HCNT,I2,1HS,2I1,4H             -1,T56,6HLOWER          ,F11.3     )
2001    FORMAT    ( 2H   .,4H    QR,I1,1HS,2I1,T18,3HCNT,I1,1HS,2I1,3H                1,T37,
     13HWAT,I1,1HS,2I1,4H             -1,T56,6HUPPER          ,F9.3     )
2005    FORMAT    ( 2H   .,4H    QR,I2,1HS,2I1,T18,3HCNT,I2,1HS,2I1,3H                1,T37,
     13HWAT,I2,1HS,2I1,4H             -1,T56,6HUPPER          ,F9.3     )
2002    FORMAT    ( 2H   .,4H    QR,I1,1HS,2I1,T18,3HCNT,I1,1HS,2I1,3H                1,T37,
     13HWAT,I1,1HS,2I1,4H             -1,T58,1H.,5X,1H.            )
2006    FORMAT    ( 2H   .,4H    QR,I2,1HS,2I1,T18,3HCNT,I2,1HS,2I1,3H                1,T37,
     13HWAT,I2,1HS,2I1,4H             -1,T58,1H.,5X,1H.            )
4000    FORMAT    ( 2H   .,4H     Q,I1,1HS,2I1,T15,6HUPPER              ,F10.3,T39,1H.,7X,1H.,
     18X,1H.,5X,1H.          )
4001    FORMAT    ( 2H   .,4H     Q,I2,1HS,2I1,T15,6HUPPER              ,F10.3,T39,1H.,7X,1H.,
     18X,1H.,5X,1H.          )
      RETURN
      END
^Z
      SUBROUTINE         DATALN     ( JJJ,   KK,   MINCRP,      LANDIF,     NMAX,     NREM,     JN   )
C  This    subroutine      writes     the    input    cards     and    constraint      identifying
C  lines    to   the    datafile     b:PLAN.DATA
      COMMON      /PROD/     YMIN(3,50),      NMINJ(50),      NMINI(50)
      COMMON      /NEW1/     NG(3),     WD(3),     JCROP(3,50,10),          PCROP(3,25),        LNGTH(50)
      COMMON/SHC/           JIRR(21),      JRES(25),      JNOSPL(20),       JDIV(2,15)
      COMMON/FLO/           NLIM(21),      NSS(3)
      DIMENSION        JJ(3),    ISS(3),    II(3)
      CHARACTER         *  100    LFB(14),      LFR(14),      LFD(14),      LFL(4),     LFW(14),      LF
c   LFB   =  array    of   format     lines    for   writing     the    continuity      equations,
c           CNTnSiT     according       to   the    magnitude       of   the    node     numbers
      DATA     LFB/'(    3H   EQ,7X,1H.,7X,3HCNT,I1,1HS,2I1,5X,1H.,7X,1H.,7X,1H.,
     18X,1H.,6X,1H.          )',
     2            '(  3H   EQ,7X,1H.,7X,3HCNT,I2,1HS,2I1,5X,1H.,7X,1H.,7X,1H.,
     28X,1H.,6X,1H.          )',
     3            '(  3H   EQ,7X,1H.,7X,3HCNT,I1,1HS,2I1,5X,1H.,4X,4H                CNT,I1,1
     3HS,2I1,4X,1H.,8X,1H.,6X,1H.                 )',
     4            '(  3H   EQ,7X,1H.,7X,3HCNT,I1,1HS,2I1,5X,1H.,4X,4H                CNT,I2,1
```

```
     4HS,2I1,4X,1H.,8X,1H.,6X,1H.                    )',
     5         '(  3H    EQ,7X,1H.,7X,3HCNT,I2,1HS,2I1,5X,1H.,4X,4H              CNT,I1,1
     5HS,2I1,4X,1H.,8X,1H.,6X,1H.                    )',
     6         '(  3H    EQ,7X,1H.,7X,3HCNT,I2,1HS,2I1,5X,1H.,4X,4H              CNT,I2,1
     6HS,2I1,4X,1H.,8X,1H.,6X,1H.                    )',
     7         '(  3H    EQ,7X,1H.,7X,3HCNT,I1,1HS,2I1,6X,1H.,4X,4H              CNT,I1,1
     7HS,2I1,4X,1H.,3X,4H         CNT,I1,1HS,2I1,5X,1H.          )',
     8         '(  3H    EQ,7X,1H.,7X,3HCNT,I2,1HS,2I1,6X,1H.,4X,4H              CNT,I1,1
     8HS,2I1,4X,1H.,3X,4H         CNT,I1,1HS,2I1,5X,1H.          )',
     9         '(  3H    EQ,7X,1H.,7X,3HCNT,I1,1HS,2I1,6X,1H.,4X,4H              CNT,I2,1
     9HS,2I1,4X,1H.,3X,4H         CNT,I1,1HS,2I1,5X,1H.          )',
     1         '(  3H    EQ,7X,1H.,7X,3HCNT,I2,1HS,2I1,6X,1H.,4X,4H              CNT,I2,1
     1HS,2I1,4X,1H.,3X,4H         CNT,I1,1HS,2I1,5X,1H.          )',
     1         '(  3H    EQ,7X,1H.,7X,3HCNT,I2,1HS,2I1,6X,1H.,4X,4H              CNT,I1,1
     1HS,2I1,4X,1H.,3X,4H         CNT,I2,1HS,2I1,5X,1H.          )',
     2         '(  3H    EQ,7X,1H.,7X,3HCNT,I1,1HS,2I1,6X,1H.,4X,4H              CNT,I1,1
     2HS,2I1,4X,1H.,3X,4H         CNT,I2,1HS,2I1,5X,1H.          )',
     3         '(  3H    EQ,7X,1H.,7X,3HCNT,I1,1HS,2I1,6X,1H.,4X,4H              CNT,I2,1
     3HS,2I1,4X,1H.,3X,4H         CNT,I2,1HS,2I1,5X,1H.          )',
     4         '(  3H    EQ,7X,1H.,7X,3HCNT,I2,1HS,2I1,6X,1H.,4X,4H              CNT,I2,1
     4HS,2I1,4X,1H.,3X,4H         CNT,I2,1HS,2I1,5X,1H.          )'/
c  LFR  =  array  of  format  lines  for  writing  the  constraints  limiting  the
c         production  of  certain  crops  in  certain  seasons  to  at  least  the
c         production  targets  according  to  the  magnitude  of  the  crop
c         numbers
      DATA  LFR/'(  3H  GE,7X,1H.,7X,5HMNCRP,I1,1HS,I1,5X,1H.,4X,5HMNCRP,I1
     1,1HS,I1,3X,1H.,4X,5HMNCRP,I1,1HS,I1,3X,1H.                 )',
     2         '(  3H    GE,7X,1H.,7X,5HMNCRP,I2,1HS,I1,4X,1H.,4X,5HMNCRP,I1
     2,1HS,I1,3X,1H.,4X,5HMNCRP,I1,1HS,I1,3X,1H.                 )',
     3         '(  3H    GE,7X,1H.,7X,5HMNCRP,I2,1HS,I1,4X,1H.,4X,5HMNCRP,I2
     3,1HS,I1,2X,1H.,4X,5HMNCRP,I1,1HS,I1,3X,1H.                 )',
     4         '(  3H    GE,7X,1H.,7X,5HMNCRP,I2,1HS,I1,4X,1H.,4X,5HMNCRP,I2
     4,1HS,I1,2X,1H.,4X,5HMNCRP,I2,1HS,I1,2X,1H.                 )',
     5         '(  3H    GE,7X,1H.,7X,5HMNCRP,I1,1HS,I1,5X,1H.,4X,5HMNCRP,I2
     5,1HS,I1,2X,1H.,4X,5HMNCRP,I1,1HS,I1,3X,1H.                 )',
     6         '(  3H    GE,7X,1H.,7X,5HMNCRP,I1,1HS,I1,5X,1H.,4X,5HMNCRP,I2
     6,1HS,I1,2X,1H.,4X,5HMNCRP,I2,1HS,I1,2X,1H.                 )',
     7         '(  3H    GE,7X,1H.,7X,5HMNCRP,I1,1HS,I1,5X,1H.,4X,5HMNCRP,I1
     7,1HS,I1,3X,1H.,4X,5HMNCRP,I2,1HS,I1,2X,1H.                 )',
     8         '(  3H    GE,7X,1H.,7X,5HMNCRP,I2,1HS,I1,4X,1H.,4X,5HMNCRP,I1
     8,1HS,I1,3X,1H.,4X,5HMNCRP,I2,1HS,I1,2X,1H.                 )',
     9         '(  3H    GE,7X,1H.,7X,5HMNCRP,I1,1HS,I1,5X,1H.,4X,5HMNCRP,I1
     9,1HS,I1,3X,1H.,6X,1H.,8X,1H.                 )',
     1         '(  3H    GE,7X,1H.,7X,5HMNCRP,I1,1HS,I1,5X,1H.,4X,5HMNCRP,I2
     1,1HS,I1,2X,1H.,8X,1H.,6X,1H.                 )',
     1         '(  3H    GE,7X,1H.,7X,5HMNCRP,I2,1HS,I1,4X,1H.,4X,5HMNCRP,I1
     1,1HS,I1,3X,1H.,8X,1H.,6X,1H.                 )',
     2         '(  3H    GE,7X,1H.,7X,5HMNCRP,I2,1HS,I1,4X,1H.,4X,5HMNCRP,I2
     2,1HS,I1,2X,1H.,8X,1H.,6X,1H.                 )',
     3         '(  3H    GE,7X,1H.,7X,5HMNCRP,I1,1HS,I1,5X,1H.,7X,1H.,7X,1H.
     3,8X,1H.,6X,1H.          )',
     4         '(  3H    GE,7X,1H.,7X,5HMNCRP,I2,1HS,I1,4X,1H.,7X,1H.,7X,1H.
```

```
     4,8X,1H.,6X,1H.                )'/
c  LFD  = array of format lines for writing the constraints grouping some
c         crops under a common production target, CCROPnjSi,
c         according to the magnitude of the group number
      DATA LFD/'(  3H  GE,7X,1H.,7X,5HCCROP,I1,1HS,I1,5X,1H.,4X,5HCCROP,I1
     1,1HS,I1,3X,1H.,4X,5HCCROP,I1,1HS,I1,3X,1H.                )',
     2          '(  3H  GE,7X,1H.,7X,5HCCROP,I2,1HS,I1,4X,1H.,4X,5HCCROP,I1
     2,1HS,I1,3X,1H.,4X,5HCCROP,I1,1HS,I1,3X,1H.                )',
     3          '(  3H  GE,7X,1H.,7X,5HCCROP,I1,1HS,I1,5X,1H.,4X,5HCCROP,I2
     3,1HS,I1,2X,1H.,4X,5HCCROP,I1,1HS,I1,3X,1H.                )',
     4          '(  3H  GE,7X,1H.,7X,5HCCROP,I2,1HS,I1,4X,1H.,4X,5HCCROP,I2
     4,1HS,I1,2X,1H.,4X,5HCCROP,I1,1HS,I1,3X,1H.                )',
     5          '(  3H  GE,7X,1H.,7X,5HCCROP,I1,1HS,I1,5X,1H.,4X,5HCCROP,I1
     5,1HS,I1,3X,1H.,4X,5HCCROP,I2,1HS,I1,2X,1H.                )',
     6          '(  3H  GE,7X,1H.,7X,5HCCROP,I2,1HS,I1,4X,1H.,4X,5HCCROP,I1
     6,1HS,I1,3X,1H.,4X,5HCCROP,I2,1HS,I1,2X,1H.                )',
     7          '(  3H  GE,7X,1H.,7X,5HCCROP,I1,1HS,I1,5X,1H.,4X,5HCCROP,I2
     7,1HS,I1,2X,1H.,4X,5HCCROP,I2,1HS,I1,2X,1H.                )',
     8          '(  3H  GE,7X,1H.,7X,5HCCROP,I2,1HS,I1,4X,1H.,4X,5HCCROP,I2
     8,1HS,I1,2X,1H.,4X,5HCCROP,I2,1HS,I1,2X,1H.                )',
     9          '(  3H  GE,7X,1H.,7X,5HCCROP,I1,1HS,I1,5X,1H.,7X,1H.,7X,1H.
     9,6X,1H.,6X,1H.          )',
     1          '(  3H  GE,7X,1H.,7X,5HCCROP,I2,1HS,I1,4X,1H.,7X,1H.,7X,1H.
     1,6X,1H.,6X,1H.          )',
     1          '(  3H  GE,7X,1H.,7X,5HCCROP,I1,1HS,I1,5X,1H.,4X,5HCCROP,I1
     1,1HS,I1,3X,1H.,6X,1H.,8X,1H.            )',
     2          '(  3H  GE,7X,1H.,7X,5HCCROP,I2,1HS,I1,4X,1H.,4X,5HCCROP,I1
     2,1HS,I1,3X,1H.,6X,1H.,8X,1H.            )',
     3          '(  3H  GE,7X,1H.,7X,5HCCROP,I1,1HS,I1,5X,1H.,4X,5HCCROP,I2
     3,1HS,I1,2X,1H.,6X,1H.,8X,1H.            )',
     4          '(  3H  GE,7X,1H.,7X,5HCCROP,I2,1HS,I1,4X,1H.,4X,5HCCROP,I2
     4,1HS,I1,2X,1H.,6X,1H.,8X,1H.            )'/
c  LFL  = array of format lines for writing the long growing season
c         constraints, LG1jKk and LG2jKk, according to the magnitude of
c         the crop or district number
      DATA LFL/'(  3H  LE,7X,1H.,7X,     3HLG1,I1,1HK,I1,6X,1H.,4X,      3HLG2,I1,1
     1HK,I1,3X,1H.,6X,1H.,8X,1H.            )',
     2          '(  3H  LE,7X,1H.,7X,     3HLG1,I1,1HK,I2,5X,1H.,4X,      3HLG2,I1,1
     2HK,I2,3X,1H.,6X,1H.,8X,1H.            )',
     3          '(  3H  LE,7X,1H.,7X,     3HLG1,I2,1HK,I1,6X,1H.,4X,      3HLG2,I2,1
     3HK,I1,3X,1H.,6X,1H.,8X,1H.            )',
     4          '(  3H  LE,7X,1H.,7X,     3HLG1,I2,1HK,I2,5X,1H.,4X,      3HLG2,I2,1
     4HK,I2,3X,1H.,6X,1H.,8X,1H.            )'/
c  LFW  = array of format lines for writing the constraints defining the
c         water demand at each irrigation node for each sub-season, WATnSiT,
c         according to the magnitude of the node numbers
      DATA LFW/'(  3H  EQ,7X,1H.,7X,3HWAT,I1,1HS,2I1,5X,1H.,7X,1H.,7X,1H.,
     18X,1H.,6X,1H.          )',
     2          '(  3H  EQ,7X,1H.,7X,3HWAT,I2,1HS,2I1,5X,1H.,7X,1H.,7X,1H.,
     28X,1H.,6X,1H.          )',
     3          '(  3H  EQ,7X,1H.,7X,3HWAT,I1,1HS,2I1,5X,1H.,4X,4H      WAT,I1,1
     3HS,2I1,4X,1H.,8X,1H.,6X,1H.            )',
```

```
     4       '( 3H  EQ,7X,1H.,7X,3HWAT,I1,1HS,2I1,5X,1H.,4X,4H              WAT,I2,1
     4HS,2I1,4X,1H.,8X,1H.,6X,1H.                 )',
     5       '( 3H  EQ,7X,1H.,7X,3HWAT,I2,1HS,2I1,5X,1H.,4X,4H              WAT,I1,1
     5HS,2I1,4X,1H.,8X,1H.,6X,1H.                 )',
     6       '( 3H  EQ,7X,1H.,7X,3HWAT,I2,1HS,2I1,5X,1H.,4X,4H              WAT,I2,1
     6HS,2I1,4X,1H.,8X,1H.,6X,1H.                 )',
     7       '( 3H  EQ,7X,1H.,7X,3HWAT,I1,1HS,2I1,6X,1H.,4X,4H              WAT,I1,1
     7HS,2I1,4X,1H.,3X,4H           WAT,I1,1HS,2I1,5X,1H.          )',
     8       '( 3H  EQ,7X,1H.,7X,3HWAT,I2,1HS,2I1,6X,1H.,4X,4H              WAT,I1,1
     8HS,2I1,4X,1H.,3X,4H           WAT,I1,1HS,2I1,5X,1H.          )',
     9       '( 3H  EQ,7X,1H.,7X,3HWAT,I1,1HS,2I1,6X,1H.,4X,4H              WAT,I2,1
     9HS,2I1,4X,1H.,3X,4H           WAT,I1,1HS,2I1,5X,1H.          )',
     1       '( 3H  EQ,7X,1H.,7X,3HWAT,I2,1HS,2I1,6X,1H.,4X,4H              WAT,I2,1
     1HS,2I1,4X,1H.,3X,4H           WAT,I1,1HS,2I1,5X,1H.          )',
     1       '( 3H  EQ,7X,1H.;7X,3HWAT,I2,1HS,2I1,6X,1H.,4X,4H              WAT,I1,1
     1HS,2I1,4X,1H.,3X,4H           WAT,I2,1HS,2I1,5X,1H.          )',
     2       '( 3H  EQ,7X,1H.,7X,3HWAT,I1,1HS,2I1,6X,1H.,4X,4H              WAT,I1,1
     2HS,2I1,4X,1H.,3X,4H           WAT,I2,1HS,2I1,5X,1H.          )',
     3       '( 3H  EQ,7X,1H.,7X,3HWAT,I1,1HS,2I1,6X,1H.,4X,4H              WAT,I2,1
     3HS,2I1,4X,1H.,3X,4H           WAT,I2,1HS,2I1,5X,1H.          )',
     4       '( 3H  EQ,7X,1H.,7X,3HWAT,I2,1HS,2I1,6X,1H.,4X,4H              WAT,I2,1
     4HS,2I1,4X,1H.,3X,4H           WAT,I2,1HS,2I1,5X,1H.          )'/
C
c   Determine if idle land enters the formulation
      IF ( LANDIF .GT. 0 ) WRITE ( 8,1022 )
c   Define the suitable land constraints for each district and season
        DO 10, K=1,KK
          IF ( K .GT. 9 ) GO TO 8
          WRITE ( 8,1001 ) K, K, K
          GO TO 10
8         WRITE ( 8,1011 ) K, K, K
10        CONTINUE
C
c   Determine if the crop production has targets, if so, how many
          IF ( MINCRP .LE. 0 ) GO TO 20
            IF ( NMAX .LT.3 ) GO TO 58
           DO 25, JC = 1,NMAX,3
              IND = 1
                J1 = NMINJ(JC)
               IS1 = NMINI(JC)
                J2 = NMINJ(JC+1)
               IS2 = NMINI(JC+1)
                J3 = NMINJ(JC+2)
               IS3 = NMINI(JC+2)
C All the values which will be printed on one line have been defined,
C next the format will be assigned according to the size, one space or
C two, of the crop number:
            IF ( J1 .GT. 9 ) GO TO 40
            IF ( J2 .GT. 9 ) GO TO 50
            IF ( J3 .GT. 9 ) IND = IND + 6
               GO TO 55
40          IF ( J2 .GT. 9 ) GO TO 45
```

```
      IF ( J3 .GT. 9 ) IND = IND + 6
           IND = IND + 1
           GO TO 55
45    IF ( J3 .GT. 9 ) IND = IND + 1
           IND = IND + 2
           GO TO 55
50    IF ( J3 .GT. 9 ) IND = IND + 1
           IND = IND + 4
55         LF = LFR (IND)
      WRITE ( 8,LF ) J1, IS1, J2, IS2, J3, IS3
25    CONTINUE
C The last line of MNCRP values will be printed according to whether
C there is one or two more than a multiple of three non-zero values:
C If there is a multiple of three non-zero values no more
C lines will be printed.
58    GO TO ( 65, 60 ) NREM
      GO TO 70
60    J1 = NMINJ( NMAX+1 )
      IS1 = NMINI( NMAX+1 )
      J2 = NMINJ( NMAX+2 )
      IS2 = NMINI( NMAX+2 )
      IND = 9
      IF ( J1 .GT. 9 ) IND = IND + 2
      IF ( J2 .GT. 9 ) IND = IND + 1
      LF = LFR (IND)
      WRITE ( 8,LF ) J1, IS1, J2, IS2
      GO TO 70
65    IND = 13
      IF ( NMINJ(NMAX+1) .GT. 9 ) IND = IND + 1
      LF = LFR (IND)
      WRITE ( 8,LF ) NMINJ (NMAX+1), NMINI (NMAX+1)
70    IF ( MINCRP .LT. 2 ) GO TO 20
C If MINCRP = 1 or 2, all the names of the constraints guaranteeing
C minimum production of a crop have been written to the disk. Now the
C names of the constraints guaranteeing the minimum production of groups
C of crops will be written to the diskfile b:PLAN.DATA
      DO 100, I = 1,3
      K = 1
      DO 110, J = 1,NG(I)
      IND = 1
      JJ(K) = J
      ISS(K) = I
      K = K + 1
      IF (K .GT. 3 ) GO TO 105
      GO TO 110
105   K = 1
      IF ( JJ(1) .GT. 9 ) IND = IND + 1
      IF ( JJ(2) .GT. 9 ) IND = IND + 2
      IF ( JJ(3) .GT. 9 ) IND = IND + 4
      LF = LFD (IND)
      WRITE ( 8, LF ) ( JJ(L), ISS(L), L=1,3 )
110   CONTINUE
```

```
            IND = 9
            GO TO ( 100, 120, 150 ) K
120         IF ( JJ(I) .GT. 9 ) IND = IND + 1
            LF = LFD (IND)
      WRITE ( 8, LF ) JJ(1), ISS(1)
            GO TO 100
150         IF ( JJ(1) .GT. 9 ) IND = IND + 1
            IF ( JJ(2) .GT. 9 ) IND = IND + 2
            IND = IND + 2
            LF = LFD (IND)
      WRITE ( 8, LF ) JJ(1), ISS(1), JJ(2), ISS(2)
100   CONTINUE
20          WRITE ( 8,1003 )
C Next the constraints of continuity
            N = 1
            I = 1
            IS = 1
C Find one line of three sub-seasonal continuity equations at a time
9           NRMDER = 0
11          NRMDER = 1 + NRMDER
            IF ( NRMDER .GT. 3 ) GO TO 15
            JJ(NRMDER) = N
            II(NRMDER) = I
            ISS(NRMDER) = IS
            IS = 1 + IS
            IF ( IS .LE. NSS(I) ) GO TO 11
C Next season
            IS = 1
            I = 1 + I
            IF ( I .LE. 3 ) GO TO 11
C Next node
            I = 1
            N = 1 + N
            IF ( N .LE. JN ) GO TO 11
C The last line of CNT values will be printed according to whether
C there is one, two, or three three sub-seasonal values more to go
      GO TO ( 17, 16, 15 ) NRMDER
            GO TO 180
C All the values which will be printed on one line have been defined,
C next the format according to the size, one space or two, of the NODE
C number will be assigned:
15          IND = 7
            IF ( JJ(1) .GT. 9 ) IND = IND + 1
            IF ( JJ(2) .GT. 9 ) IND = IND + 2
            IF ( JJ(3) .GT. 9 ) IND = IND + 4
            LF = LFB (IND)
      WRITE ( 8,LF ) ( JJ(M),II(M),ISS(M), M=1,3 )
            IF ( N .LE. JN ) GO TO 9
            GO TO 180
16          IND = 3
            IF ( JJ(1) .GT. 9 ) IND = IND + 2
            IF ( JJ(2) .GT. 9 ) IND = IND + 1
```

```
          LF = LFB (IND)
        WRITE ( 8,LF ) ( JJ(M),II(M),ISS(M), M=1,2 )
          GO TO 180
17        IND = 1
        IF ( JJ(1) .GT. 9 ) IND = IND + 1
          LF = LFB (IND)
        WRITE ( 8,LF ) JJ(1), II(1), ISS(1)
C Now the WAT equations for each node with irrigation:
180       N = 2
          I = 1
          IS = 1
C Find one line of three sub-seasonal WAT equations at a time
189       NRMDER = 0
181       NRMDER = 1 + NRMDER
        IF ( NRMDER .GT. 3 ) GO TO 185
        JJ(NRMDER) = JIRR(N)
        II(NRMDER) = I
        ISS(NRMDER) = IS
          IS = 1 + IS
        IF ( IS .LE. NSS(I) ) GO TO 181
C Next season
          IS = 1
          I = 1 + I
        IF ( I .LE. 3 ) GO TO 181
C Next node
          I = 1
          N = 1 + N
C There is one WAT equation for each node delivering irrigation water,
C even where more than one district is serviced from one node.
C Since up to five districts may be serviced from one node, check at
C most four times, that there are no more districts serviced from the
C same node.
        DO 183 NLOOP = 1,4
        IF ( JIRR(N) .EQ. JIRR(N-1) ) GO TO 182
        GO TO 184
182       N = N + 1
183     CONTINUE
184       IF ( N .LE. JIRR(1)+1 ) GO TO 181
C The last line of WAT values will be printed according to whether
C there is one, two, or three three sub-seasonal values more to go
        GO TO ( 187, 186, 185 ) NRMDER
          GO TO 125
C All the values which will be printed on one line have been defined,
C next the format according to the size, one space or two, of the NODE
C number will be assigned:
185       IND = 7
        IF ( JJ(1) .GT. 9 ) IND = IND + 1
        IF ( JJ(2) .GT. 9 ) IND = IND + 2
        IF ( JJ(3) .GT. 9 ) IND = IND + 4
          LF = LFW (IND)
        WRITE ( 8,LF ) ( JJ(M),II(M),ISS(M), M=1,3 )
          IF ( N .LE. JIRR(1)+1 ) GO TO 189
```

```
                GO  TO  125
186          IND  =  3
          IF ( JJ(1)  .GT.  9 ) IND = IND + 1
          IF ( JJ(2)  .GT.  9 ) IND = IND + 1
             LF  =  LFW   (IND)
          WRITE  ( 8,LF  ) ( JJ(M),II(M),ISS(M),        M=1,2   )
             GO  TO  125
187            IND  =  1
           IF ( JJ(1)  .GT.  9 ) IND = IND + 1
              LF  =  LFW   (IND)
           WRITE   ( 8,LF  )  JJ(1),   II(1),   ISS(1)
C  Now that the constraints  for the formulation  selected  have  been
C  written  to the diskette,  the upperbound   constraint  will be listed.
125     WRITE  ( 8,1005   )
C  The  LNG1jDk   and LNG2jDk   constraints   for crops  of long  growing   season
C  will  be  written:
             DO  130  J=1,JJJ
                IF ( LNGTH(J)   .LT.  1 ) GO  TO  130
             DO  129  K=1,KK
                  IND  =  1
                  IF ( K .GT.  9 ) IND  =  IND  + 1
                  IF ( J .GT.  9 ) IND  =  IND  + 2
                     LF  =  LFL(IND)
                  WRITE   ( 8,  LF  ) J,  K,  J,  K
129           CONTINUE
130           CONTINUE
          WRITE   ( 8,  1009   )
C  The  land  utilization   variables   Dn  will  be  written   if  necessary
        IF ( LANDIF   .LT.  1 ) GO  TO  140
             WRITE   ( 8,  1006   )
        IF ( LANDIF   .LT.  2 ) GO  TO  135
             WRITE   ( 8,  1007   ) -WD(1),   -WD(2),   -WD(3)
                GO  TO  140
135           WRITE   ( 8,  1008   ) WD(1),   WD(2),   WD(3)
140     CONTINUE
C
C3H   LE,7X,1H.,7X,6HWATER1,7X,1H.,5X,6HWATER2,4X,1H.,5X,6HWATER3,4X,1H.
)
1022    FORMAT   ( 3H  EQ,7X,1H.,7X,5HDIFF1,8X,1H.,5X,5HDIFF2,5X,1H.,6X,5HDIF
     1F3,4X,1H./3H       EQ,7X,1H.,7X,5HDEFD1,8X,1H.,5X,5HDEFD2,5X,1H.,6X,5HDE
     1FD3,4X,1H.      )
1001    FORMAT   ( 3H  LE,7X,1H.,7X,4HLAND,I1,2HS1,6X,1H.,4X,4HLAND,I1,2HS2,4
     1X,1H.,3X,4HLAND,I1,2HS3,5X,1H.                  )
1011    FORMAT   ( 3H  LE,7X,1H.,7X,4HLAND,I2,2HS1,5X,1H.,4X,4HLAND,I2,2HS2,3
     1X,1H.,3X,4HLAND,I2,2HS3,4X,1H.                  )
1003   FORMAT   ( 8H  LOWERBD,3X,1H.,6X,5HLOWER,T32,1H.,T40,1H.,T48,1H.,T57,
     11H.,6X,1H.      )
1004    FORMAT   ( 3H  EQ,7X,1H.,7X,3HWAT,I1,2HS1,6X,1H.,4X,4H             WAT,I1,2HS2,4X
     1,1H.,3X,4H       WAT,I1,2HS3,5X,1H.             )
1044    FORMAT   ( 3H  EQ,7X,1H.,7X,3HWAT,I2,2HS1,5X,1H.,4X,4H             WAT,I2,2HS2,3X
     1,1H.,3X,4H       WAT,I2,2HS3,4X,1H.             )
1005    FORMAT   ( 8H  UPPERBD,3X,1H.,6X,5HUPPER,T32,1H.,T40,1H.,T48,1H.,T57,
```

```
     11H.,6X,1H.        )
1009    FORMAT    ( 4H    MAX,7X,1H.,6X,3HOBJ,T32,1H.,T40,1H.,T48,1H.,T57,1H.,6X
     1,1H.     )
1006    FORMAT    ( 2H    .,4X,3HMD1,8X,5HDEFD1,8X,1H1,5X,5HDIFF1,7X,1H1,6X,1H.,
     16X,1H./2H       .,4X,3HMD2,8X,5HDEFD1,8X,1H1,5X,5HDIFF1,6X,2H-1,6X,1H.,
     26X,1H./2H       .,4X,3HMD3,8X,5HDEFD2,8X,1H1,5X,5HDIFF2,7X,1H1,6X,1H.,6X
     3,1H./2H      .,4X,3HMD4,8X,5HDEFD2,8X,1H1,5X,5HDIFF2,6X,2H-1,6X,1H.,6X,
     41H./2H      .,4X,3HMD5,8X,5HDEFD3,8X,1H1,5X,5HDIFF3,7X,1H1,6X,1H.,6X,1H
     5./2H     .,4X,3HMD6,8X,5HDEFD3,8X,1H1,5X,5HDIFF3,6X,2H-1,6X,1H.,6X,1H.
     6   )
1007    FORMAT    ( 2H    .,5X,2HM1,8X,3HOBJ,F11.2,5X,5HDEFD1,6X,2H-1,7X,1H.,6X,
     71H./2H      .,5X,2HM2,8X,3HOBJ,F11.2,5X,5HDEFD2,6X,2H-1,7X,1H.,6X,1H.
     6/2H    .,5X,2HM3,8X,3HOBJ,F11.2,5X,5HDEFD3,6X,2H-1,7X,1H.,6X,1H.                    )
1008    FORMAT    ( 2H    .,5X,2HM1,8X,5HUPPER,F9.0,5X,5HDEFD1,6X,2H-1,7X,1H.,6X
     1,1H./2H      .,5X,2HM2,8X,5HUPPER,F9.0,5X,5HDEFD2,6X,2H-1,7X,1H.,6X,1H.
     2/2H    .,5X,2HM3,8X,5HUPPER,F9.0,5X,5HDEFD3,6X,2H-1,7X,1H.,6X,1H.                    )
      RETURN
      END
^Z
      SUBROUTINE    CHOICE    ( I,  J,  K,  MINCRP,    LANDIF    )
c  Write   the   crop   variables,    IiJjKk,   the   number   of  hectares    to  be
c  planted    in  a  crop   variety    in  a  season    in  a  district,   to  the   SAS.OR
c  datafile.
      COMMON     /SHR/    P(3,50,20),     Y(3,50,20),     CL(3,50,20),      F(3,50,20),      U(3,
     150,20),    T(3,50,20),     B(3,50,20),     Q(3,50,20)
      COMMON     /PROD/    YMIN(3,50),     NMINJ(50)
      COMMON     /NEW1/    NG(3),    WD(3),    JCROP(3,50,10),      PCROP(3,25),      LNGTH(50)
      COMMON/SHC/        JIRR(21),     JRES(25),     JNOSPL(20),      JDIV(2,15)
      COMMON/CAP/        KIRR(21)
      COMMON/SUB/        WCOEF(3,50,20,5)
      COMMON/FLO/        NUM(21),     NSS(3)
      DIMENSION      JC(3),    IS(3),    KC(3)
      CHARACTER      * 100    LFO(8),    LFM(8),    LFW(8),    LFD(12),     LFC(8),     LFL(16)
      CHARACTER      * 100    LFL3(16),    LF
      CHARACTER      * 150    LFL2(16),    LFC1(16),     LFC2(16),     LFC3(16),     LFS(12),
     1                LF3
c   LFO   =  array    of  format    lines    for  writing    the  crop    variable     in  the
c         objective     function,    OBJ,    the   LANDkli,    total    district    land
c         constraint,     and   the   upperbound     constraint,     UPPER,     of  suitable
c         land    for  cultivation     of  that   crop    variety,    according     to  the
c         magnitudes      of  the   crop   and   district     numbers.^V^VThese             apply    to
c         every    variable.
      DATA    LFO/'(    5H   .   ,2H    I,I1,1HJ,I1,1HK,        I1,T18,3HOBJ,4X,F9.2,T37,4
     1HLAND,I1,1HS,I1,T49,1H1,T54,5HUPPER,F9.2)',
     2           '(  4H   .   ,2H    I,I1,1HJ,I2,1HK,         I1,T18,3HOBJ,4X,F9.2,T37,4H
     2LAND,I1,1HS,I1,T49,1H1,T54,5HUPPER,F9.2)',
     3           '(  4H   .   ,2H    I,I1,1HJ,I1,1HK,         I2,T18,3HOBJ,4X,F9.2,T37,4H
     3LAND,I2,1HS,I1,T49,1H1,T54,5HUPPER,F9.2)',
     4           '(  3H   .  ,2H   I,I1,1HJ,I2,1HK,         I2,T18,3HOBJ,4X,F9.2,T37,4HL
     4AND,I2,1HS,I1,T49,1H1,T54,5HUPPER,F9.2)',
     5           '(  5H   .    ,2H    I,I1,1HJ,I1,1HK,        I1,T18,5HLOWER,F9.2,T40,1H
     5.,T49,1H.,T56,1H.,T64,1H.               )
```

```
     6          '(  4H  .  ,2H   I,I1,1HJ,I2,1HK,        I1,T18,5HLOWER,F9.2,T40,1H.
     6,T49,1H.,T56,1H.,T64,1H.                 ) ',
     7          '(  4H  .  ,2H   I,I1,1HJ,I1,1HK,        I2,T18,5HLOWER,F9.2,T40,1H.
     7,T49,1H.,T56,1H.,T64,1H.                 ) ',
     8          '(  3H  .  ,2H   I,I1,1HJ,I2,1HK,        I2,T18,5HLOWER,F9.2,T40,1H.,
     8T49,1H.,T56,1H.,T64,1H.                  ) '/
c    LFM   =  array  of  format  lines  for  writing  crop  variables  in  the
c           MNCRP  constraints  on  production  targets  for  individual  crop
c           varieties  according  to  the  magnitudes  of  the  crop  and  district
c           numbers
      DATA  LFM/'(  5H  .  ,2H   I,I1,1HJ,I1,1HK,        I1,T18,5HMNCRP,I1,1HS,I1,F
     19.2,T40,1H.,T49,1H.,T56,1H.,T64,1H.          ) ',
     2          '(  4H  .  ,2H   I,I1,1HJ,I2,1HK,        I1,T18,5HMNCRP,I1,1HS,I1,F9
     2.2,T40,1H.,T49,1H.,T56,1H.,T64,1H.          ) ',
     3          '(  4H  .  ,2H   I,I1,1HJ,I1,1HK,        I2,T18,5HMNCRP,I1,1HS,I1,F9
     3.2,T40,1H.,T49,1H.,T56,1H.,T64,1H.          ) ',
     4          '(  3H  .  ,2H   I,I1,1HJ,I2,1HK,        I2,T18,5HMNCRP,I1,1HS,I1,F9.
     42,T40,1H.,T49,1H.,T56,1H.,T64,1H.           ) ',
     5          '(  5H  .  ,2H   I,I1,1HJ,I1,1HK,        I1,T18,5HMNCRP,I2,1HS,I1,F
     59.2,T40,1H.,T49,1H.,T56,1H.,T64,1H.          ) ',
     6          '(  4H  .  ,2H   I,I1,1HJ,I2,1HK,        I1,T18,5HMNCRP,I2,1HS,I1,F9
     6.2,T40,1H.,T49,1H.,T56,1H.,T64,1H.          ) ',
     7          '(  4H  .  ,2H   I,I2,1HJ,I1,1HK,        I2,T18,5HMNCRP,I2,1HS,I1,F9
     7.2,T40,1H.,T49,1H.,T56,1H.,T64,1H.          ) ',
     8          '(  3H  .  ,2H   I,I1,1HJ,I2,1HK,        I2,T18,5HMNCRP,I2,1HS,I1,F9.
     82,T40,1H.,T49,1H.,T56,1H.,T64,1H.           ) '/
c    LFW   =  array  of  format  lines  for  writing  crop  variables  in  the  WATnSiT
c           constraints  defining  the  sub-seasonal  water  demand
c           according  to  the  magnitudes  of  the  crop,  district  and  node  numbers
      DATA  LFW/'(  5H  .  ,2H   I,I1,1HJ,I1,1HK,        I1,T18,3HWAT,I1,1HS,2I1,F1
     11.6,T40,1H.,T49,1H.,T56,1H.,T64,1H.)           ',
     2          '(  4H  .  ,2H   I,I1,1HJ,I2,1HK,        I1,T18,3HWAT,I1,1HS,2I1,F11
     2.6,T40,1H.,T49,1H.,T56,1H.,T64,1H.)            ',
     3          '(  4H  .  ,2H   I,I1,1HJ,I1,1HK,        I2,T18,3HWAT,I1,1HS,2I1,F11
     3.6,T40,1H.,T49,1H.,T56,1H.,T64,1H.)            ',
     4          '(  3H  .  ,2H   I,I1,1HJ,I2,1HK,        I2,T18,3HWAT,I1,1HS,2I1,F11.
     46,T40,1H.,T49,1H.,T56,1H.,T64,1H.)             ',
     5          '(  5H  .  ,2H   I,I1,1HJ,I1,1HK,        I1,T18,3HWAT,I2,1HS,2I1,F1
     51.6,T40,1H.,T49,1H.,T56,1H.,T64,1H.)           ',
     6          '(  4H  .  ,2H   I,I1,1HJ,I2,1HK,        I1,T18,3HWAT,I2,1HS,2I1,F11
     6.6,T40,1H.,T49,1H.,T56,1H.,T64,1H.)            ',
     7          '(  4H  .  ,2H   I,I1,1HJ,I1,1HK,        I2,T18,3HWAT,I2,1HS,2I1,F11
     7.6,T40,1H.,T49,1H.,T56,1H.,T64,1H.)            ',
     8          '(  3H  .  ,2H   I,I1,1HJ,I2,1HK,        I2,T18,3HWAT,I2,1HS,2I1,F11.
     86,T40,1H.,T49,1H.,T56,1H.,T64,1H.)             '/
c    LFD   =  array  of  format  lines  for  writing  the  crop  variables,  in  the
c           DIFFi  constraints,  defining  the  difference  between  seasons  in
c           the  amounts  of  idle  land  according  to  the  magnitudes  of  the
c           crop  and  district  numbers
      DATA  LFD/'(  5H  .  ,2H   I,I1,1HJ,I1,1HK,        I1,T18,5HDIFF1,      F9.2,T37,
     15HDIFF3,T48,2H-1,T56,1H.,T64,1H.)',
     2          '(  4H  .  ,2H   I,I1,1HJ,I2,1HK,        I1,T18,5HDIFF1,      F9.2,T37,5
```

```
     2HDIFF3,T48,2H-1,T56,1H.,T64,1H.)',
     3          '(  4H   .  ,2H   I,I1,1HJ,I1,1HK,        I2,T18,5HDIFF1,          F9.2,T37,5
     3HDIFF3,T48,2H-1,T56,1H.,T64,1H.)',
     4          '(  3H   . ,2H   I,I1,1HJ,I2,1HK,        I2,T18,5HDIFF1,          F9.2,T37,5H
     4DIFF3,T48,2H-1,T56,1H.,T64,1H.)',
     5          '(  5H   .   ,2H   I,I1,1HJ,I1,1HK,        I1,T18,5HDIFF1,7X,2H-1,T37
     5,5HDIFF2,F9.2,T56,1H.,T64,1H.)',
     6          '(  4H   .  ,2H   I,I1,1HJ,I2,1HK,        I1,T18,5HDIFF1,7X,2H-1,T37,
     65HDIFF2,F9.2,T56,1H.,T64,1H.)',
     7          '(  4H   .  ,2H   I,I1,1HJ,I1,1HK,        I2,T18,5HDIFF1,7X,2H-1,T37,
     75HDIFF2,F9.2,T56,1H.,T64,1H.)',
     8          '(  3H   . ,2H   I,I1,1HJ,I2,1HK,        I2,T18,5HDIFF1,7X,2H-1,T37,5
     8HDIFF2,F9.2,T56,1H.,T64,1H.)',
     9          '(  5H   .   ,2H   I,I1,1HJ,I1,1HK,        I1,T18,5HDIFF2,7X,2H-1,T37
     9,5HDIFF3,F9.2,T56,1H.,T64,1H.)',
     1          '(  4H   .  ,2H   I,I1,1HJ,I2,1HK,        I1,T18,5HDIFF2,7X,2H-1,T37,
     15HDIFF3,F9.2,T56,1H.,T64,1H.)',
     1          '(  4H   .  ,2H   I,I1,1HJ,I1,1HK,        I2,T18,5HDIFF2,7X,2H-1,T37,
     15HDIFF3,F9.2,T56,1H.,T64,1H.)',
     2          '(  3H   . ,2H   I,I1,1HJ,I2,1HK,        I2,T18,5HDIFF2,7X,2H-1,T37,5
     2HDIFF3,F9.2,T56,1H.,T64,1H.)'/
c   LFL  =  array  of  format  lines  for  writing  the  crop  variables  in  the
c           long  growing  season  constraints  according  to  the  magnitudes  of
c           the  crop  and  district  numbers.
      DATA  LFL/'(  5H   .   ,2H   I,I1,1HJ,I1,1HK,        I1,T18,    3HLG1,I1,1HK,I1,6X
     1,1H1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     2          '(  4H   .  ,2H   I,I1,1HJ,I2,1HK,        I1,T18,    3HLG1,I1,1HK,I1,6X,
     21H1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     3          '(  4H   .  ,2H   I,I1,1HJ,I1,1HK,        I2,T18,    3HLG1,I1,1HK,I1,6X,
     31H1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     4          '(  3H   . ,2H   I,I1,1HJ,I2,1HK,        I2,T18,    3HLG1,I1,1HK,I1,6X,1
     4H1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     5          '(  5H   .   ,2H   I,I1,1HJ,I1,1HK,        I1,T18,    3HLG1,I2,1HK,I1,6X
     5,1H1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     6          '(  4H   .  ,2H   I,I1,1HJ,I2,1HK,        I1,T18,    3HLG1,I2,1HK,I1,6X,
     61H1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     7          '(  4H   .  ,2H   I,I1,1HJ,I1,1HK,        I2,T18,    3HLG1,I2,1HK,I1,6X,
     71H1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     8          '(  3H   . ,2H   I,I1,1HJ,I2,1HK,        I2,T18,    3HLG1,I2,1HK,I1,6X,1
     8H1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     9          '(  5H   .   ,2H   I,I1,1HJ,I1,1HK,        I1,T18,    3HLG1,I1,1HK,I2,6X
     9,1H1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     1          '(  4H   .  ,2H   I,I1,1HJ,I2,1HK,        I1,T18,    3HLG1,I1,1HK,I2,6X,
     11H1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     1          '(  4H   .  ,2H   I,I1,1HJ,I1,1HK,        I2,T18,    3HLG1,I1,1HK,I2,6X,
     11H1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     2          '(  3H   . ,2H   I,I1,1HJ,I2,1HK,        I2,T18,    3HLG1,I1,1HK,I2,6X,1
     2H1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     3          '(  5H   .   ,2H   I,I1,1HJ,I1,1HK,        I1,T18,    3HLG1,I2,1HK,I2,6X
     3,1H1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     4          '(  4H   .  ,2H   I,I1,1HJ,I2,1HK,        I1,T18,    3HLG1,I2,1HK,I2,6X,
     41H1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
```

```
      5          '( 4H  .  ,2H  I,I1,1HJ,I1,1HK,          I2,T18,    3HLG1,I2,1HK,I2,6X,
      51H1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
      6          '( 3H  . ,2H  I,I1,1HJ,I2,1HK,          I2,T18,    3HLG1,I2,1HK,I2,6X,1
      6H1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)'/
c   LFL2   =  array   of  format   lines   for  writing   the  crop   variables   in  the
c          long   growing    season    constraints     according    to  the  magnitudes    of
c          the   crop   and   district    numbers.
       DATA    LFL2/'(    5H  .   ,2H  I,I1,1HJ,I1,1HK,          I1,T18,    3HLG1,I1,1HK,I1,6
      1X,2H-1,T37,        3HLG2,I1,1HK,I1,6X,1H1,T56,1H.,T64,1H.)',
      2          '( 4H  .  ,2H  I,I1,1HJ,I2,1HK,          I1,T18,    3HLG1,I1,1HK,I1,6X,
      2,2H-1,T37,       3HLG2,I1,1HK,I1,6X,1H1,T56,1H.,T64,1H.)',
      3          '( 4H  .  ,2H  I,I1,1HJ,I1,1HK,          I2,T18,    3HLG1,I1,1HK,I1,6X,
      3,2H-1,T37,       3HLG2,I1,1HK,I1,6X,1H1,T56,1H.,T64,1H.)',
      4          '( 3H  . ,2H  I,I1,1HJ,I2,1HK,          I2,T18,    3HLG1,I1,1HK,I1,6X,2
      4H-1,T37,      3HLG2,I1,1HK,I1,6X,1H1,T56,1H.,T64,1H.)',
      5          '( 5H  .   ,2H  I,I1,1HJ,I1,1HK,          I1,T18,    3HLG1,I2,1HK,I1,6X
      5,2H-1,T37,       3HLG2,I2,1HK,I1,6X,1H1,T56,1H.,T64,1H.)',
      6          '( 4H  .  ,2H  I,I1,1HJ,I2,1HK,          I1,T18,    3HLG1,I2,1HK,I1,6X,
      62H-1,T37,      3HLG2,I2,1HK,I1,6X,1H1,T56,1H.,T64,1H.)',
      7          '( 4H  .  ,2H  I,I1,1HJ,I1,1HK,          I2,T18,    3HLG1,I2,1HK,I1,6X,
      72H-1,T37,      3HLG2,I2,1HK,I1,6X,1H1,T56,1H.,T64,1H.)',
      8          '( 3H  . ,2H  I,I1,1HJ,I2,1HK,          I2,T18,    3HLG1,I2,1HK,I1,6X,2
      8H-1,T37,      3HLG2,I2,1HK,I1,6X,1H1,T56,1H.,T64,1H.)',
      9          '( 5H  .   ,2H  I,I1,1HJ,I1,1HK,          I1,T18,    3HLG1,I1,1HK,I2,6X
      9,2H-1,T37,       3HLG2,I1,1HK,I2,6X,1H1,T56,1H.,T64,1H.)',
      1          '( 4H  .  ,2H  I,I1,1HJ,I2,1HK,          I1,T18,    3HLG1,I1,1HK,I2,6X,
      12H-1,T37,      3HLG2,I1,1HK,I2,6X,1H1,T56,1H.,T64,1H.)',
      1          '( 4H  .  ,2H  I,I1,1HJ,I1,1HK,          I2,T18,    3HLG1,I1,1HK,I2,6X,
      12H-1,T37,      3HLG2,I1,1HK,I2,6X,1H1,T56,1H.,T64,1H.)',
      2          '( 3H  . ,2H  I,I1,1HJ,I2,1HK,          I2,T18,    3HLG1,I1,1HK,I2,6X,2
      2H-1,T37,      3HLG2,I1,1HK,I2,6X,1H1,T56,1H.,T64,1H.)',
      3          '( 5H  .   ,2H  I,I1,1HJ,I1,1HK,          I1,T18,    3HLG1,I2,1HK,I2,6X
      3,2H-1,T37,       3HLG2,I2,1HK,I2,6X,1H1,T56,1H.,T64,1H.)',
      4          '( 4H  .  ,2H  I,I1,1HJ,I2,1HK,          I1,T18,    3HLG1,I2,1HK,I2,6X,
      42H-1,T37,      3HLG2,I2,1HK,I2,6X,1H1,T56,1H.,T64,1H.)',
      5          '( 4H  .  ,2H  I,I1,1HJ,I1,1HK,          I2,T18,    3HLG1,I2,1HK,I2,6X,
      52H-1,T37,      3HLG2,I2,1HK,I2,6X,1H1,T56,1H.,T64,1H.)',
      6          '( 3H  . ,2H  I,I1,1HJ,I2,1HK,          I2,T18,    3HLG1,I2,1HK,I2,6X,2
      6H-1,T37,      3HLG2,I2,1HK,I2,6X,1H1,T56,1H.,T64,1H.)'/
c   LFL3   =  array   of  format   lines   for  writing   the  crop   variables   in  the
c          long   growing    season    constraints     according    to  the  magnitudes    of
c          the   crop   and   district    numbers.
       DATA    LFL3/'(    5H  .   ,2H  I,I1,1HJ,I1,1HK,          I1,T18,    3HLG2,I1,1HK,I1,6
      1X,2H-1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
      2          '( 4H  .  ,2H  I,I1,1HJ,I2,1HK,          I1,T18,    3HLG2,I1,1HK,I1,6X,
      22H-1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
      3          '( 4H  .  ,2H  I,I1,1HJ,I1,1HK,          I2,T18,    3HLG2,I1,1HK,I1,6X,
      32H-1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
      4          '( 3H  . ,2H  I,I1,1HJ,I2,1HK,          I2,T18,    3HLG2,I1,1HK,I1,6X,2
      4H-1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
      5          '( 5H  .   ,2H  I,I1,1HJ,I1,1HK,          I1,T18,    3HLG2,I2,1HK,I1,6X
      5,2H-1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
```

```
     6            '(  4H   .   ,2H    I,I1,1HJ,I2,1HK,           I1,T18,     3HLG2,I2,1HK,I1,6X,
     62H-1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     7            '(  4H   .   ,2H    I,I1,1HJ,I1,1HK,           I2,T18,     3HLG2,I2,1HK,I1,6X,
     72H-1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     8            '(  3H   . ,2H    I,I1,1HJ,I2,1HK,           I2,T18,     3HLG2,I2,1HK,I1,6X,2
     8H-1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     9            '(  5H   .    ,2H    I,I1,1HJ,I1,1HK,          I1,T18,     3HLG2,I1,1HK,I2,6X
     9,2H-1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     1            '(  4H   .   ,2H    I,I1,1HJ,I2,1HK,           I1,T18,     3HLG2,I1,1HK,I2,6X,
     12H-1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     1            '(  4H   .   ,2H    I,I1,1HJ,I1,1HK,           I2,T18,     3HLG2,I1,1HK,I2,6X,
     12H-1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     2            '(  3H   . ,2H    I,I1,1HJ,I2,1HK,           I2,T18,     3HLG2,I1,1HK,I2,6X,2
     2H-1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     3            '(  5H   .    ,2H    I,I1,1HJ,I1,1HK,          I1,T18,     3HLG2,I2,1HK,I2,6X
     3,2H-1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     4            '(  4H   .   ,2H    I,I1,1HJ,I2,1HK,           I1,T18,     3HLG2,I2,1HK,I2,6X,
     42H-1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     5            '(  4H   .   ,2H    I,I1,1HJ,I1,1HK,           I2,T18,     3HLG2,I2,1HK,I2,6X,
     52H-1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)',
     6            '(  3H   . ,2H    I,I1,1HJ,I2,1HK,           I2,T18,     3HLG2,I2,1HK,I2,6X,2
     6H-1,T40,1H.,T49,1H.,T56,1H.,T64,1H.)'/
c   LFC    =  array   of  format    lines  for  writing    the  crop    variables    in  the
c           constraints        grouping       crop     varieties      under      common        production
c           targets       according       to   the    magnitudes         of
c           the   crop    and    district      numbers.
      DATA    LFC/'(     5H    .     ,2H    I,I1,1HJ,I1,1HK,            I1,T18,5HCCROP,I1,1HS,I1,F
     19.2,T40,1H.,T49,1H.,T56,1H.,T64,1H.                          )  ',
     2            '(  4H   .   ,2H    I,I1,1HJ,I2,1HK,           I1,T18,5HCCROP,I1,1HS,I1,F9
     2.2,T40,1H.,T49,1H.,T56,1H.,T64,1H.                           )  ',
     3            '(  4H   .   ,2H    I,I1,1HJ,I1,1HK,           I2,T18,5HCCROP,I1,1HS,I1,F9
     3.2,T40,1H.,T49,1H.,T56,1H.,T64,1H.                           )  ',
     4            '(  3H   . ,2H    I,I1,1HJ,I2,1HK,           I2,T18,5HCCROP,I1,1HS,I1,F9.
     42,T40,1H.,T49,1H.,T56,1H.,T64,1H.                            )  ',
     5            '(  5H   .    ,2H    I,I1,1HJ,I2,1HK,          I1,T18,5HCCROP,I2,1HS,I1,F
     59.2,T40,1H.,T49,1H.,T56,1H.,T64,1H.                          )  ',
     6            '(  4H   .   ,2H    I,I1,1HJ,I2,1HK,           I1,T18,5HCCROP,I2,1HS,I1,F9
     6.2,T40,1H.,T49,1H.,T56,1H.,T64,1H.                           )  ',
     7            '(  4H   .   ,2H    I,I1,1HJ,I2,1HK,           I1,T18,5HCCROP,I2,1HS,I1,F9
     7.2,T40,1H.,T49,1H.,T56,1H.,T64,1H.                           )  ',
     8            '(  3H   . ,2H    I,I1,1HJ,I2,1HK,           I2,T18,5HCCROP,I2,1HS,I1,F9.
     82,T40,1H.,T49,1H.,T56,1H.,T64,1H.                            )  '/
c   LFC1    =  array   of  format    lines  for  writing    the  crop    variables    in  the
c           constraints        grouping       crop     varieties      under      common        production
c           targets       according       to   the    magnitudes         of
c           the   crop    and    district      numbers.
      DATA    LFC1/'(     5H    .     ,2H    I,I1,1HJ,I1,1HK,            I1,T18,5HCCROP,I1,1HS,I1,
     1F9.2,T37,5HCCROP,I1,1HS,I1,F8.2,T54,5HCCROP,I1,1HS,I1,F9.2                         )',
     2            '(  4H   .   ,2H    I,I1,1HJ,I2,1HK,           I1,T18,5HCCROP,I1,1HS,I1,F9
     2.2,T37,5HCCROP,I1,1HS,I1,F8.2,T54,5HCCROP,I1,1HS,I1,F9.2                        )',
     3            '(  4H   .   ,2H    I,I1,1HJ,I1,1HK,           I2,T18,5HCCROP,I1,1HS,I1,F9
     3.2,T37,5HCCROP,I1,1HS,I1,F8.2,T54,5HCCROP,I1,1HS,I1,F9.2                        )',
```

```
4          '(  3H   .  ,2H    I,I1,1HJ,I2,1HK,          I2,T18,5HCCROP,I1,1HS,I1,F9.
42,T37,5HCCROP,I1,1HS,I1,F8.2,T54,5HCCROP,I1,1HS,I1,F9.2                       )',
5          '(  5H   .    ,2H    I,I1,1HJ,I1,1HK,        I1,T18,5HCCROP,I2,1HS,I1,F
59.2,T37,5HCCROP,I1,1HS,I1,F8.2,T54,5HCCROP,I1,1HS,I1,F9.2                       )',
6          '(  4H   .   ,2H    I,I1,1HJ,I2,1HK,         I1,T18,5HCCROP,I2,1HS,I1,F9
6.2,T37,5HCCROP,I1,1HS,I1,F8.2,T54,5HCCROP,I1,1HS,I1,F9.2                        )',
7          '(  4H   .   ,2H    I,I1,1HJ,I1,1HK,         I2,T18,5HCCROP,I2,1HS,I1,F9
7.2,T37,5HCCROP,I1,1HS,I1,F8.2,T54,5HCCROP,I1,1HS,I1,F9.2                        )',
8          '(  3H   .  ,2H    I,I1,1HJ,I2,1HK,          I2,T18,5HCCROP,I2,1HS,I1,F9.
82,T37,5HCCROP,I1,1HS,I1,F8.2,T54,5HCCROP,I1,1HS,I1,F9.2                       )',
9          '(  5H   .    ,2H    I,I1,1HJ,I1,1HK,        I1,T18,5HCCROP,I1,1HS,I1,F
99.2,T37,5HCCROP,I2,1HS,I1,F8.2,T54,5HCCROP,I1,1HS,I1,F9.2                       )',
1          '(  4H   .   ,2H    I,I1,1HJ,I2,1HK,         I1,T18,5HCCROP,I1,1HS,I1,F9
1.2,T37,5HCCROP,I2,1HS,I1,F8.2,T54,5HCCROP,I1,1HS,I1,F9.2                        )',
1          '(  4H   .   ,2H    I,I1,1HJ,I1,1HK,         I2,T18,5HCCROP,I1,1HS,I1,F9
1.2,T37,5HCCROP,I2,1HS,I1,F8.2,T54,5HCCROP,I1,1HS,I1,F9.2                        )',
2          '(  3H   .  ,2H    I,I1,1HJ,I2,1HK,          I2,T18,5HCCROP,I1,1HS,I1,F9.
22,T37,5HCCROP,I2,1HS,I1,F8.2,T54,5HCCROP,I1,1HS,I1,F9.2                       )',
3          '(  5H   .    ,2H    I,I1,1HJ,I1,1HK,        I1,T18,5HCCROP,I2,1HS,I1,F
39.2,T37,5HCCROP,I2,1HS,I1,F8.2,T54,5HCCROP,I1,1HS,I1,F9.2                       )',
4          '(  4H   .   ,2H    I,I1,1HJ,I2,1HK,         I1,T18,5HCCROP,I2,1HS,I1,F9
4.2,T37,5HCCROP,I2,1HS,I1,F8.2,T54,5HCCROP,I1,1HS,I1,F9.2                        )',
5          '(  4H   .   ,2H    I,I1,1HJ,I1,1HK,         I2,T18,5HCCROP,I2,1HS,I1,F9
5.2,T37,5HCCROP,I2,1HS,I1,F8.2,T54,5HCCROP,I1,1HS,I1,F9.2                        )',
6          '(  3H   .  ,2H    I,I1,1HJ,I2,1HK,          I2,T18,5HCCROP,I2,1HS,I1,F9.
62,T37,5HCCROP,I2,1HS,I1,F8.2,T54,5HCCROP,I1,1HS,I1,F9.2                       )'/
c    LFC2   =  array   of  format   lines  for  writing   the  crop   variables   in  the
c           constraints     grouping     crop   varieties    under    common     production
c           targets    according     to  the  magnitudes     of
c           the   crop   and   district    numbers.
      DATA    LFC2/'(    5H   .    ,2H   I,I1,1HJ,I1,1HK,         I1,T18,5HCCROP,I1,1HS,I1,
7F9.2,T37,5HCCROP,I1,1HS,I1,F8.2,T54,5HCCROP,I2,1HS,I1,F9.2                        )',
8          '(  4H   .   ,2H    I,I1,1HJ,I2,1HK,         I1,T18,5HCCROP,I1,1HS,I1,F9
8.2,T37,5HCCROP,I1,1HS,I1,F8.2,T54,5HCCROP,I2,1HS,I1,F9.2                        )',
9          '(  4H   .   ,2H    I,I1,1HJ,I1,1HK,         I2,T18,5HCCROP,I1,1HS,I1,F9
9.2,T37,5HCCROP,I1,1HS,I1,F8.2,T54,5HCCROP,I2,1HS,I1,F9.2                        )',
2          '(  3H   .  ,2H    I,I1,1HJ,I2,1HK,          I2,T18,5HCCROP,I1,1HS,I1,F9.
22,T37,5HCCROP,I1,1HS,I1,F8.2,T54,5HCCROP,I2,1HS,I1,F9.2                       )',
1          '(  5H   .    ,2H    I,I1,1HJ,I1,1HK,        I1,T18,5HCCROP,I2,1HS,I1,F
19.2,T37,5HCCROP,I1,1HS,I1,F8.2,T54,5HCCROP,I2,1HS,I1,F9.2                       )',
2          '(  4H   .   ,2H    I,I1,1HJ,I2,1HK,         I1,T18,5HCCROP,I2,1HS,I1,F9
2.2,T37,5HCCROP,I1,1HS,I1,F8.2,T54,5HCCROP,I2,1HS,I1,F9.2                        )',
3          '(  4H   .   ,2H    I,I1,1HJ,I1,1HK,         I2,T18,5HCCROP,I2,1HS,I1,F9
3.2,T37,5HCCROP,I1,1HS,I1,F8.2,T54,5HCCROP,I2,1HS,I1,F9.2                        )',
4          '(  3H   .  ,2H    I,I1,1HJ,I2,1HK,          I2,T18,5HCCROP,I2,1HS,I1,F9.
42,T37,5HCCROP,I1,1HS,I1,F8.2,T54,5HCCROP,I2,1HS,I1,F9.2                       )',
5          '(  5H   .    ,2H    I,I1,1HJ,I1,1HK,        I1,T18,5HCCROP,I1,1HS,I1,F
59.2,T37,5HCCROP,I2,1HS,I1,F8.2,T54,5HCCROP,I2,1HS,I1,F9.2                       )',
6          '(  4H   .   ,2H    I,I1,1HJ,I2,1HK,         I1,T18,5HCCROP,I1,1HS,I1,F9
6.2,T37,5HCCROP,I2,1HS,I1,F8.2,T54,5HCCROP,I2,1HS,I1,F9.2                        )',
7          '(  4H   .   ,2H    I,I1,1HJ,I1,1HK,         I2,T18,5HCCROP,I1,1HS,I1,F9
7.2,T37,5HCCROP,I2,1HS,I1,F8.2,T54,5HCCROP,I2,1HS,I1,F9.2                        )',
```

```
8        '( 3H  . ,2H   I,I1,1HJ,I2,1HK,        I2,T18,5HCCROP,I1,1HS,I1,F9
82,T37,5HCCROP,I2,1HS,I1,F8.2,T54,5HCCROP,I2,1HS,I1,F9.2          )',
9        '( 5H   .   ,2H   I,I1,1HJ,I1,1HK,     I1,T18,5HCCROP,I2,1HS,I1,F
99.2,T37,5HCCROP,I2,1HS,I1,F8.2,T54,5HCCROP,I2,1HS,I1,F9.2        )',
3        '( 4H   .   ,2H   I,I1,1HJ,I2,1HK,     I1,T18,5HCCROP,I2,1HS,I1,F9
3.2,T37,5HCCROP,I2,1HS,I1,F8.2,T54,5HCCROP,I2,1HS,I1,F9.2         )',
1        '( 4H   .   ,2H   I,I1,1HJ,I1,1HK,     I2,T18,5HCCROP,I2,1HS,I1,F9
1.2,T37,5HCCROP,I2,1HS,I1,F8.2,T54,5HCCROP,I2,1HS,I1,F9.2         )',
2        '( 3H   . ,2H   I,I1,1HJ,I2,1HK,       I2,T18,5HCCROP,I2,1HS,I1,F9.
22,T37,5HCCROP,I2,1HS,I1,F8.2,T54,5HCCROP,I2,1HS,I1,F9.2          )'/
c   LFC3  = array  of  format  lines  for  writing  the  crop  variables  in  the
c        constraints   grouping   crop   varieties   under   common   production
c        targets   according   to   the   magnitudes   of
c        the   crop   and   district   numbers.
   DATA   LFC3/'(   5H   .   ,2H   I,I1,1HJ,I1,1HK,   I1,T18,5HCCROP,I1,1HS,I1,
1F9.2,T37,5HCCROP,I1,1HS,I1,F9.2,T56,1H.,T64,1H.            ) ',
2        '( 4H   .   ,2H   I,I1,1HJ,I2,1HK,     I1,T18,5HCCROP,I1,1HS,I1,F9
2.2,T37,5HCCROP,I1,1HS,I1,F9.2,T56,1H.,T64,1H.              ) ',
3        '( 4H   .   ,2H   I,I1,1HJ,I1,1HK,     I2,T18,5HCCROP,I1,1HS,I1,F9
3.2,T37,5HCCROP,I1,1HS,I1,F9.2,T56,1H.,T64,1H.              ) ',
4        '( 3H   . ,2H   I,I1,1HJ,I2,1HK,       I2,T18,5HCCROP,I1,1HS,I1,F9.
42,T37,5HCCROP,I1,1HS,I1,F9.2,T56,1H.,T64,1H.               ) ',
5        '( 5H   .   ,2H   I,I1,1HJ,I1,1HK,     I1,T18,5HCCROP,I2,1HS,I1,F
59.2,T37,5HCCROP,I1,1HS,I1,F9.2,T56,1H.,T64,1H.             ) ',
6        '( 4H   .   ,2H   I,I1,1HJ,I2,1HK,     I1,T18,5HCCROP,I2,1HS,I1,F9
6.2,T37,5HCCROP,I1,1HS,I1,F9.2,T56,1H.,T64,1H.              ) ',
7        '( 4H   .   ,2H   I,I1,1HJ,I1,1HK,     I2,T18,5HCCROP,I2,1HS,I1,F9
7.2,T37,5HCCROP,I1,1HS,I1,F9.2,T56,1H.,T64,1H.              ) ',
8        '( 3H   . ,2H   I,I1,1HJ,I2,1HK,       I2,T18,5HCCROP,I2,1HS,I1,F9.
82,T37,5HCCROP,I1,1HS,I1,F9.2,T56,1H.,T64,1H.               ) ',
9        '( 5H   .   ,2H   I,I1,1HJ,I1,1HK,     I1,T18,5HCCROP,I1,1HS,I1,F
99.2,T37,5HCCROP,I2,1HS,I1,F9.2,T56,1H.,T64,1H.             ) ',
1        '( 4H   .   ,2H   I,I1,1HJ,I2,1HK,     I1,T18,5HCCROP,I1,1HS,I1,F9
1.2,T37,5HCCROP,I2,1HS,I1,F9.2,T56,1H.,T64,1H.              ) ',
1        '( 4H   .   ,2H   I,I1,1HJ,I1,1HK,     I2,T18,5HCCROP,I1,1HS,I1,F9
1.2,T37,5HCCROP,I2,1HS,I1,F9.2,T56,1H.,T64,1H.              ) ',
2        '( 3H   . ,2H   I,I1,1HJ,I2,1HK,       I2,T18,5HCCROP,I1,1HS,I1,F9.
22,T37,5HCCROP,I2,1HS,I1,F9.2,T56,1H.,T64,1H.               ) ',
3        '( 5H   .   ,2H   I,I1,1HJ,I1,1HK,     I1,T18,5HCCROP,I2,1HS,I1,F
39.2,T37,5HCCROP,I2,1HS,I1,F9.2,T56,1H.,T64,1H.             ) ',
4        '( 4H   .   ,2H   I,I1,1HJ,I2,1HK,     I1,T18,5HCCROP,I2,1HS,I1,F9
4.2,T37,5HCCROP,I2,1HS,I1,F9.2,T56,1H.,T64,1H.              ) ',
5        '( 4H   .   ,2H   I,I1,1HJ,I1,1HK,     I2,T18,5HCCROP,I2,1HS,I1,F9
5.2,T37,5HCCROP,I2,1HS,I1,F9.2,T56,1H.,T64,1H.              ) ',
6        '( 3H   . ,2H   I,I1,1HJ,I2,1HK,       I2,T18,5HCCROP,I2,1HS,I1,F9.
62,T37,5HCCROP,I2,1HS,I1,F9.2,T56,1H.,T64,1H.               ) '/
C   Assign  the  value  of  MIND  according  to  the  magnitude  of  J , K
C        and  the  coefficients  of  the  constraint  equations.
              MIND  =  1
              IIND  =  I - 1
   IF  ( J .GT.  9 ) MIND  =  MIND  +  1
   IF  ( K .GT.  9 ) MIND  =  MIND  +  2
```

```
            IND = MIND
          LF = LFO(IND)
        WRITE ( 8, LF ) I, J, K, Q(I,J,K), K, I, T(I,J,K)
      IF ( LANDIF .LT. 1 ) GO TO 1
C Next the DIFFn coefficients (-1, WD), selon the season I, will be
C written to the disk.
            IND = MIND
            COEF = 1.0
          IF ( I .GT. 2 ) IND = IND + 4
          IF ( I .GT. 1 ) IND = IND + 4
            LF = LFD(IND)
        WRITE ( 8, LF ) I, J, K, COEF
1         IF ( LNGTH(J) .LE. 0 ) GO TO 4
C If the crop takes more than one season to grow it will appear in the
C LNGijDk constraints ensuring that it utilizes the same amount of land
C in the two seasons it takes to grow.
           IND = MIND
          IF ( J .GT. 9 ) IND = IND + 4
          IF ( K .GT. 9 ) IND = IND + 8
            GO TO ( 2, 3 ) IIND
            LF = LFL(IND)
        WRITE ( 8, LF ) I, J, K, J, K
            GO TO 4
2           LF3 = LFL2(IND)
        WRITE ( 8,LF3 ) I, J, K, J, K, J, K
            GO TO 4
3           LF = LFL3(IND)
        WRITE ( 8, LF ) I, J, K, J, K
4     IF ( U(I,J,K) .LE. 0 ) GO TO 10
C If the crop requires irrigation, its seasonal water requirement is
C tallied in the water balance equations.
C Find what N corresponds to K
      DO 205 N=2,JIRR(1)+1
        IF ( KIRR(N) .EQ. K ) GO TO 210
205    CONTINUE
      WRITE ( *,1000 ) K
      STOP
210         IND = MIND
      IF ( JIRR(N) .GT. 9 ) IND = IND + 4
            LF = LFW(IND)
        DO 211 ISS=1,NSS(I)
211     WRITE ( 8, LF ) I, J, K, JIRR(N), I, ISS, WCOEF(I,J,K,ISS)
C
10    IF ( MINCRP .GE. 1 .AND. MINCRP .LT. 3 ) GO TO 20
         IF ( B(I,J,K) .LE. 0 ) GO TO 20
C If lower limits are placed on the number of hectares to be planted in
C a district, write them:
            IND = MIND + 4
            LF = LFO (IND)
         WRITE ( 8, LF ) I, J, K, B(I,J,K)
20    IF ( YMIN(I,J) .LE. 0 .AND. MINCRP .GT. 1 ) GO TO 50
      IF ( YMIN(I,J) .LE. 0 .OR. MINCRP .LE. 0 ) GO TO 100
```

```
C If there is minimum production required of a crop, it will be written:
      IND = MIND
      IF ( J .GT. 9 ) IND = IND + 4
      LF = LFM(IND)
9     WRITE ( 8, LF ) I, J, K, J, I, Y(I,J,K)
      IF ( MINCRP .LE. 1 ) GO TO 100
50    IF ( JCROP(I,J,1) .LE. 0 ) GO TO 100
C Crops are grouped, write the coefficients for the CCROP constraint:
      L = 1
      N = 1
105   IF ( JCROP(I,J,L) .LE. 0 ) GO TO 200
      IND = MIND
      JC(N) = JCROP(I,J,L)
      IS(N) = I
      N = N + 1
      L = L + 1
      IF (N .GT. 3 ) GO TO 110
      GO TO 105
200   GO TO ( 100, 120, 150 ) N
110   N = 1
      IF ( JC(1) .GT. 9 ) IND = IND + 4
      IF ( JC(2) .GT. 9 ) IND = IND + 8
      IF ( JC(3) .GT. 9 ) GO TO 115
      LF3 = LFC1 (IND)
      GO TO 118
115   LF3 = LFC2 (IND)
118   WRITE ( 8, LF3 ) I, J, K, ( JC(M), IS(M), Y(I,J,K), M=1,3 )
      GO TO 105
120   IF ( JC(1) .GT. 9 ) IND = IND + 4
      LF = LFC (IND)
      WRITE ( 8, LF ) I, J, K, JC(1), IS(1), Y(I,J,K)
      GO TO 100
150   IF ( JC(1) .GT. 9 ) IND = IND + 4
      IF ( JC(2) .GT. 9 ) IND = IND + 8
      LF3 = LFC3 (IND)
      WRITE (8,LF3) I,J,K, JC(1),IS(1),Y(I,J,K), JC(2),IS(2),Y(I,J,K)
C
100   CONTINUE
1000  FORMAT ( ' There is no point listed in the configuration as servin
     1g the ',I3,'th district. ' )
      RETURN
      END
^Z
      SUBROUTINE RHS ( JJ, KK, HH, MINCRP, LANDIF, NMAX, NREM, JN )
C This subroutine writes the _RHS_ values in the SAS sparsedata format
C to the file b:PLAN.DATA
      COMMON /PROD/ YMIN(3,50), NMINJ(50), NMINI(50)
      COMMON /NEW1/ NG(3), WD(3), JCROP(3,50,10), PCROP(3,25), LNGTH(50)
      COMMON/CAP/ KIRR(21), CCAPI(21), CCAPD(15), CRHS(3,35,5)
      COMMON/FLO/ NLIM(21), NSS(3)
      DIMENSION SUM(3), JK(3), ISS(3), R(3), HH(3,20), II(3)
      CHARACTER * 100 LFR(10), LFB(14), LFW(6), LFC(14), LF
```

```
c LFR   = array of format lines to write the RHS constants of the LANDkSi
c         constraints which define the total land available in a district
c         for a season, and the production targets of the individual crop
c         varieties of a MNCRPjSi constraint, according to the magnitudes
c         of the crop and district numbers
      DATA LFR/'( 2H .,T6,11H_RHS_ LAND,I1,1HS,I1,F9.0,6H LAND,I1,1HS,
     1I1,F9.0,6H LAND,I1,1HS,I1,F9.0 )',
     2 '( 2H .,T6,11H_RHS_ LAND,I2,1HS,I1,F8.0,6H LAND,I2,1HS,
     2I1,F8.0,6H LAND,I2,1HS,I1,F8.0 )',
     3 '( 2H .,T6,12H_RHS_ MNCRP,I1,1HS,I1,F8.0,7H MNCRP,I1,1H
     3S,I1,F8.0,7H MNCRP,I1,1HS,I1,F8.0 )',
     4 '( 2H .,T6,12H_RHS_ MNCRP,I2,1HS,I1,F7.0,7H MNCRP,I1,1H
     4S,I1,F8.0,7H MNCRP,I1,1HS,I1,F8.0 )',
     5 '( 2H .,T6,12H_RHS_ MNCRP,I1,1HS,I1,F8.0,7H MNCRP,I2,1H
     5S,I1,F7.0,7H MNCRP,I1,1HS,I1,F8.0 )',
     6 '( 2H .,T6,12H_RHS_ MNCRP,I2,1HS,I1,F7.0,7H MNCRP,I2,1H
     6S,I1,F7.0,7H MNCRP,I1,1HS,I1,F8.0 )',
     7 '( 2H .,T6,12H_RHS_ MNCRP,I1,1HS,I1,F8.0,7H MNCRP,I1,1H
     7S,I1,F8.0,7H MNCRP,I2,1HS,I1,F7.0 )',
     8 '( 2H .,T6,12H_RHS_ MNCRP,I2,1HS,I1,F7.0,7H MNCRP,I1,1H
     8S,I1,F8.0,7H MNCRP,I2,1HS,I1,F7.0 )',
     9 '( 2H .,T6,12H_RHS_ MNCRP,I1,1HS,I1,F8.0,7H MNCRP,I2,1H
     9S,I1,F7.0,7H MNCRP,I2,1HS,I1,F7.0 )',
     1 '( 2H .,T6,12H_RHS_ MNCRP,I2,1HS,I1,F7.0,7H MNCRP,I2,1H
     1S,I1,F7.0,7H MNCRP,I2,1HS,I1,F7.0 )'/
c LFB   = array of format lines to write the RHS constants of the CNTnSiT
c         constraints which allocate the water to each node every
c         sub-season, according to the magnitude of the node number
      DATA LFB/'( 2H .,12H _RHS_ CNT,I1,1HS,2I1,F10.3,6X,1H.,7X,1H.,6X
     1,1H.,7X,1H. )',
     2 '( 2H .,12H _RHS_ CNT,I2,1HS,2I1,F10.3,6X,1H.,7X,1H.,6X
     2,1H.,7X,1H. )',
     3 '( 2H .,12H _RHS_ CNT,I1,1HS,2I1,F10.3,6H CNT,I1,1HS,
     32I1,F10.3,7X,1H.,7X,1H. )',
     4 '( 2H .,12H _RHS_ CNT,I1,1HS,2I1,F10.3,6H CNT,I2,1HS,
     42I1,F10.3,7X,1H.,7X,1H. )',
     5 '( 2H .,12H _RHS_ CNT,I2,1HS,2I1,F10.3,6H CNT,I1,1HS,
     52I1,F10.3,7X,1H.,7X,1H. )',
     6 '( 2H .,12H _RHS_ CNT,I2,1HS,2I1,F10.3,6H CNT,I2,1HS,
     62I1,F10.3,7X,1H.,7X,1H. )',
     7 '( 2H .,12H _RHS_ CNT,I1,1HS,2I1,F10.3,6H CNT,I1,1HS,
     72I1,F10.3,6H CNT,I1,1HS,2I1,F10.3 )',
     8 '( 2H .,12H _RHS_ CNT,I2,1HS,2I1,F10.3,6H CNT,I1,1HS,
     82I1,F10.3,6H CNT,I1,1HS,2I1,F10.3 )',
     9 '( 2H .,12H _RHS_ CNT,I1,1HS,2I1,F10.3,6H CNT,I2,1HS,
     92I1,F10.3,6H CNT,I1,1HS,2I1,F10.3 )',
     1 '( 2H .,12H _RHS_ CNT,I2,1HS,2I1,F10.3,6H CNT,I2,1HS,
     12I1,F10.3,6H CNT,I1,1HS,2I1,F10.3 )',
     1 '( 2H .,12H _RHS_ CNT,I1,1HS,2I1,F10.3,6H CNT,I1,1HS,
     12I1,F10.3,6H CNT,I2,1HS,2I1,F10.3 )',
     2 '( 2H .,12H _RHS_ CNT,I2,1HS,2I1,F10.3,6H CNT,I1,1HS,
     22I1,F10.3,6H CNT,I2,1HS,2I1,F10.3 )',
```

```
     3          '(  2H   .,12H      _RHS_     CNT,I1,1HS,2I1,F10.3,6H                CNT,I2,1HS,
     3 2I1,F10.3,6H           CNT,I2,1HS,2I1,F10.3             )',
     4          '(  2H   .,12H      _RHS_     CNT,I2,1HS,2I1,F10.3,6H                CNT,I2,1HS,
     4 2I1,F10.3,6H           CNT,I2,1HS,2I1,F10.3             )'/
c LFW   = array  of  format   lines   to  write   the   RHS   constants    of  the  MNCRPjSi
c         constraints     which    limit   the   production     of  individual     crop
c         varieties    to  more    than    the   production      targets,     according     to  the
c         magnitude      of  the   crop    number
       DATA    LFW/'(    2H   .,T6,12H_RHS_          MNCRP,I1,1HS,I1,F8.0,7H               MNCRP,I1,1H
     1 S,I1,F8.0,6X,1H.,7X,1H.             )',
     2          '(  2H   .,T6,12H_RHS_          MNCRP,I1,1HS,I1,F8.0,7H               MNCRP,I2,1H
     2 S,I1,F7.0,6X,1H.,7X,1H.             )',
     3          '(  2H   .,T6,12H_RHS_          MNCRP,I2,1HS,I1,F7.0,7H               MNCRP,I1,1H
     3 S,I1,F8.0,6X,1H.,7X,1H.             )',
     4          '(  2H   .,T6,12H_RHS_          MNCRP,I2,1HS,I1,F7.0,7H               MNCRP,I2,1H
     4 S,I1,F7.0,6X,1H.,7X,1H.             )',
     5          '(  2H   .,T6,12H_RHS_          MNCRP,I1,1HS,I1,F8.0,6X,1H.,7X,1H.,6
     5 X,1H.,7X,1H.       )',
     6          '(  2H   .,T6,12H_RHS_          MNCRP,I2,1HS,I1,F7.0,6X,1H.,7X,1H.,6
     6 X,1H.,7X,1H.       )'/
c LFC   = array  of  format   lines   to  write   the   RHS   constants    of  the  CCROPnjSi
c         constraints     which    combine    the   production     of  certain    crop
c         varieties    under    a  common     target,    according     to  the   magnitude
c         of  the   crop   group    number
       DATA    LFC/'(    2H   .,T6,12H_RHS_          CCROP,I1,1HS,I1,F8.0,7H               CCROP,I1,1H
     1 S,I1,F8.0,7H          CCROP,I1,1HS,I1,F8.0          )',
     2          '(  2H   .,T6,12H_RHS_          CCROP,I2,1HS,I1,F7.0,7H               CCROP,I1,1H
     2 S,I1,F8.0,7H          CCROP,I1,1HS,I1,F8.0          )',
     3          '(  2H   .,T6,12H_RHS_          CCROP,I1,1HS,I1,F7.0,7H               CCROP,I2,1H
     3 S,I1,F7.0,7H          CCROP,I1,1HS,I1,F8.0          )',
     4          '(  2H   .,T6,12H_RHS_          CCROP,I2,1HS,I1,F7.0,7H               CCROP,I2,1H
     4 S,I1,F7.0,7H          CCROP,I1,1HS,I1,F8.0          )',
     5          '(  2H   .,T6,12H_RHS_          CCROP,I1,1HS,I1,F8.0,7H               CCROP,I1,1H
     5 S,I1,F8.0,7H          CCROP,I2,1HS,I1,F7.0          )',
     6          '(  2H   .,T6,12H_RHS_          CCROP,I2,1HS,I1,F7.0,7H               CCROP,I1,1H
     6 S,I1,F8.0,7H          CCROP,I2,1HS,I1,F7.0          )',
     7          '(  2H   .,T6,12H_RHS_          CCROP,I1,1HS,I1,F8.0,7H               CCROP,I2,1H
     7 S,I1,F7.0,7H          CCROP,I2,1HS,I1,F7.0          )',
     8          '(  2H   .,T6,12H_RHS_          CCROP,I2,1HS,I1,F7.0,7H               CCROP,I2,1H
     8 S,I1,F7.0,7H          CCROP,I2,1HS,I1,F7.0          )',
     9          '(  2H   .,T6,12H_RHS_          CCROP,I1,1HS,I1,F8.0,6X,1H.,7X,1H.,9
     9 X,1H.,7X,1H.       )',
     1          '(  2H   .,T6,12H_RHS_          CCROP,I2,1HS,I1,F7.0,6X,1H.,7X,1H.,9
     1 X,1H.,7X,1H.       )',
     1          '(  2H   .,T6,12H_RHS_          CCROP,I1,1HS,I1,F8.0,7H               CCROP,I1,1H
     1 S,I1,F8.0,6X,1H.,7X,1H.             )',
     2          '(  2H   .,T6,12H_RHS_          CCROP,I2,1HS,I1,F7.0,7H               CCROP,I1,1H
     2 S,I1,F8.0,6X,1H.,7X,1H.             )',
     3          '(  2H   .,T6,12H_RHS_          CCROP,I1,1HS,I1,F8.0,7H               CCROP,I2,1H
     3 S,I1,F7.0,6X,1H.,7X,1H.             )',
     4          '(  2H   .,T6,12H_RHS_          CCROP,I2,1HS,I1,F7.0,7H               CCROP,I2,1H
     4 S,I1,F7.0,6X,1H.,7X,1H.             )'/
```

```
C
c     First write the seasonal water constraints:
      N = 1
      I = 1
      IS = 1
      R(1) = 5.5
C     Find one line of three non-zero sub-seasonal inflows at a time
9     NRMDER = 1
C     Skip the zero value
11    IF ( R(NRMDER) .NE. 0.0 ) NRMDER = NRMDER + 1
      IF ( NRMDER .GT. 4 ) GO TO 15
      JK(NRMDER) = N
      II(NRMDER) = I
      ISS(NRMDER) = IS
      R(NRMDER) = CRHS(I,N,IS)
      IS = 1 + IS
      IF ( IS .LE. NSS(I) ) GO TO 11
C     Next season
      IS = 1
      I = 1 + I
      IF ( I .LE. 3 ) GO TO 11
C     Next node
      I = 1
      N = 1 + N
      IF ( N .LE. JN ) GO TO 11
C     The last line of CNT values will be printed according to whether
C     there is one, two or three non-zero values to go
      GO TO ( 17, 16, 15 ) NRMDER-1
      GO TO 18
C     All the values which will be printed on one line have been defined,
C     next the format according to the size, one space or two, of the NODE
C     number will be assigned:
15    IND = 7
      IF ( JK(2) .GT. 9 ) IND = IND + 1
      IF ( JK(3) .GT. 9 ) IND = IND + 2
      IF ( JK(4) .GT. 9 ) IND = IND + 4
      LF = LFB (IND)
      WRITE ( 8,LF ) ( JK(M),II(M),ISS(M),R(M), M=2,4 )
      IF ( N .LE. JN ) GO TO 9
      GO TO 18
16    IND = 3
      IF ( JK(2) .GT. 9 ) IND = IND + 2
      IF ( JK(3) .GT. 9 ) IND = IND + 1
      LF = LFB (IND)
      WRITE ( 8,LF ) ( JK(M),II(M),ISS(M),R(M), M=2,3 )
      GO TO 18
17    IND = 1
      IF ( JK(2) .GT. 9 ) IND = IND + 1
      LF = LFB (IND)
      WRITE ( 8,LF ) JK(2), II(2), ISS(2), R(2)
18    IF ( LANDIF .LT. 1 ) GO TO 8
C     The seasonal differences between the amounts of idle land are calculated
```

```
C and entered as the RHS of the DIFF constraints when LANDIF = 2. For
C LANDIF=1, where the difference in idle land is limited to a given
C value, the same difference in land available between seasons is
C calculated.
      DO 5, I=1,3
        SUM(I) = 0
          DO 5, K=1,KK
5     SUM(I) = SUM(I) + HH(I,K)
          DFS1 = SUM(1) - SUM(2)
          DFS2 = SUM(2) - SUM(3)
          DFS3 = SUM(3) - SUM(1)
      WRITE (8,1002) DFS1, DFS2, DFS3
C Next to be written are the maximum available land constraints by dis-
C trict with a check to see if the district number takes one or two
C spaces.
c*********************************************************************
c 8     DO 10 K=1,KK          this suddenly not compiling LF with the (
c                   IN = 1
c              IF ( K .GT. 9 ) IN = IN + 1
c                   LF = LFR (IN)
c       write ( *,99999 ) If                   skipped
c99999  FORMAT ( 1X,A100 )
c10     WRITE ( 8,LF) ( K, I, HH(I,K), I=1,3 )
c *******************************************************************
8      DO 10 K=1,KK
        IF ( K .GT. 9 ) GO TO 85
          WRITE ( 8,1000 ) ( K, I, HH(I,K), I=1,3 )
          GO TO 10
85        WRITE ( 8,1001 ) ( K, I, HH(I,K), I=1,3 )
10     CONTINUE
       IF ( MINCRP .LE. 0 ) GO TO 70
C If the minimum production of certain crops is guaranteed by a minimum
C quota, the RHS lower limits will be written to the diskette:
          IF ( NMAX .LT. 3 ) GO TO 158
        DO 65 JC=1,NMAX,3
                IND = 3
              J1 = NMINJ(JC)
             IS1 = NMINI(JC)
              Y1 = YMIN(IS1,J1)
              J2 = NMINJ(JC+1)
             IS2 = NMINI(JC+1)
              Y2 = YMIN(IS2,J2)
              J3 = NMINJ(JC+2)
             IS3 = NMINI(JC+2)
              Y3 = YMIN(IS3,J3)
C All the values which will be printed on one line have been defined,
C next the format according to the size, one space or two, of the crop
C number will be assigned:
        IF ( J1 .GT. 9 ) IND = IND + 1
        IF ( J2 .GT. 9 ) IND = IND + 2
        IF ( J3 .GT. 9 ) IND = IND + 4
              LF = LFR (IND)
```

```
      WRITE ( 8,LF ) J1, IS1, Y1, J2, IS2, Y2, J3, IS3, Y3
65    CONTINUE
C The last line of MNCRP values will be printed according to whether
C there is one or two more than a multiple of three non-zero values:
C If there is a multiple of three non-zero MNCRP values no more
C lines will be printed.
158   GO TO ( 165, 160 ) NREM
      GO TO 170
160   J1 = NMINJ( NMAX+1 )
      IS1 = NMINI( NMAX+1 )
      Y1 = YMIN( IS1,J1 )
      J2 = NMINJ( NMAX+2 )
      IS2 = NMINI( NMAX+2 )
      Y2 = YMIN( IS2,J2 )
      IND = 1
      IF ( J1 .GT. 9 ) IND = IND + 2
      IF ( J2 .GT. 9 ) IND = IND + 1
      LF = LFW (IND)
      WRITE ( 8,LF ) J1, IS1, Y1, J2, IS2, Y2
      GO TO 170
165   IND = 5
      IF ( NMINJ(NMAX+1) .GT. 9 ) IND = IND + 1
      LF = LFW (IND)
      WRITE ( 8,LF ) NMINJ (NMAX+1), NMINI (NMAX+1),
     1 YMIN(NMINI(NMAX+1), NMINJ(NMAX+1))
170   IF ( MINCRP .LE. 1 ) GO TO 70
C Now the RHS constants of the constraints guaranteeing the minimum
C production of groups of crops will be written to the diskfile
C B:Crop3.data
      DO 100, I = 1,3
      K = 1
      DO 110, J = 1,NG(I)
      IND = 1
      JK(K) = J
      ISS(K) = I
      R(K) = PCROP(I,J)
      K = K + 1
      IF (K .GT. 3 ) GO TO 105
      GO TO 110
105   K = 1
      IF ( JK(1) .GT. 9 ) IND = IND + 1
      IF ( JK(2) .GT. 9 ) IND = IND + 2
      IF ( JK(3) .GT. 9 ) IND = IND + 4
      LF = LFC (IND)
      WRITE ( 8, LF ) ( JK(L), ISS(L), R(L), L=1,3 )
110   CONTINUE
      IND = 9
      GO TO ( 100, 120, 150 ) K
120   IF ( JK(I) .GT. 9 ) IND = IND + 1
      LF = LFC (IND)
      WRITE ( 8, LF ) JK(1), ISS(1), R(1)
      GO TO 100
```

```
150      IF  ( JK(1)   .GT.   9 ) IND   =  IND   +  1
        IF  ( JK(2)   .GT.   9 ) IND   =  IND   +  2
                 IND   =  IND   +  2
                LF   =  LFC   (IND)
       WRITE    ( 8,  LF  ) JK(1),   ISS(1),    R(1),   JK(2),    ISS(2),    R(2)
100     CONTINUE
C  Finally,    the   last   line   of  the   dataset,    the   proc   lp  sparsedata      line   is
C  written.
70           WRITE     (8,1004)
C
1000     FORMAT      ( 2H   .,T6,11H_RHS_          LAND,I1,1HS,I1,F9.0,6H            LAND,I1,1HS,
      1I1,F9.0,6H        LAND,I1,1HS,I1,F9.0          )
1001     FORMAT      ( 2H   .,T6,11H_RHS_          LAND,I2,1HS,I1,F8.0,6H            LAND,I2,1HS,
      2I1,F8.0,6H        LAND,I2,1HS,I1,F8.0          )
1002     FORMAT      ( 2H   .,15H      _RHS_     DIFF1,F11.2,7H          DIFF2,F11.2,7H          DIFF3,F
      111.2    )
1004    FORMAT     ( 2H   :/41HPROC       LP   SPARSEDATA        MAXIT1=500        MAXIT2=500;/2H/*
)
        RETURN
        END
^Z
```

APPENDIX B2

The data preparation program for PROJOP reads any configuration file prepared, or enters a new configuration. Subsequently, it reads the power, reservoir, irrigation and canal files as desired by the user. Once the time independent data has been entered, the date of the current run may be given interactively. If any of the coefficients are to vary over time, the user may adjust the values interactively, or they may be retrieved from a file stored on diskette from a previous run. The file of predicted inflows for the current run is read.

The user will find it useful to save any revisions or adjustments as they are made in case the run must be restarted. The user may re-enter data from the main menu or choose a different order than that presented by the menu. The program verifies that it has the necessary information before continuing. The user may include irrigation return flows in the continuity equations, that option is entered after the predicted inflows.

After the current reservoir levels and target storages are entered interactively, the program is ready to write RELEASE.DATA to the diskette. The data preparation file for PROJOP writes the datalines by reservoirs, power plants, irrigation delivery nodes, and downstream and diverted flows. Table B2.1 shows the structure of PROJOP.

TABLE B2.1

The programming structure of PROJOP

PROJOP

calls from the menu:

CONFIG CORRIG DPOWER DRES DIRR DCANAL

which may call

DELETE CALC LOOPMK LOOPMK
ADD
LOOPMK

once all the data have been entered from the files,

PROJOP

is allowed to call:

PERIOD DINFLO RETQR (optional)

once the current reservoir levels have been entered,

PROJOP

calls

DWRITE

which, in turn, calls

WRES LFTYPE WPOWER LFRHS WIRR

which call

LFII	LFII	LFII
LFTYPE	LFTYPE	LFTYPE
LFRHS	LFRHS	LFRHS

The following table describes in which FORTRAN file each program unit is stored, and the function of the subroutine.

TABLE B2.2

Location and function of FORTRAN program units

UNIT NAME	IN FILE	FUNCTION
PROJOP	Projop.for	Main program, calls others
CONFIG	Dscheme.for	Enters new configuration nodes
CORRIG	Dscheme.for	Corrects configuration nodes
DELETE	Dscheme.for	Deletes nodes from configuration
ADD	Dscheme.for	Inserts nodes between existing nodes
LOOPMK	Dscheme.for	Prepares configuration vectors for changes
DPOWER	Readin.for	Enters power plant constants
CALC	Readin.for	Calculates power coefficient
DRES	Readin.for	Enters reservoir constants
DIRR	Readin.for	Enters irrigation constants
DCANAL	Readin.for	Enters canal constants
PERIOD	Dperiod.for	Enters starting date, varies the constants by time step
DINFLO	Dinflo.for	Enters predicted inflows
RETQR	Retqr.for	Enters irrigation return flows
DWRITE	Write.for	Writes datalines to RELEASE.DATA, calls other subroutines
WPOWER	Wlin.for	Writes power plant variables and constraints
WRES	Wlin.for	Writes reservoir variables and constraints
LFII	Wlf.for	Writes specific format lines
LFTYPE	Wlf.for	Writes format lines for constraint type lines
LFRHS	Wlf.for	Writes format lines for RHS constant term lines

Before showing a complete FORTRAN listing, the common block arrays used in the data preparation program will be explained. Definitions of other variables, arrays, or subroutine arguments appear in the program listing. There are

13 common blocks used by one or another of the subroutines.

The SHR common block contains configuration vectors of the nodes with common characteristics.

NPOWER(15) is the vector of nodes which have power plants.

NIRR(21) is the vector of nodes which have irrigation canals.

NRES(25) is the vector of nodes which have storage.

NNOSPL(20) is the vector of nodes which do not spill to the node numbered one higher.

NDIV(2,15) is the array of nodes where there are diversions. The first number of the pair diverts water to the second node.

The SHP common block stores a few more of the configuration vectors.

NINFLO(30) is the vector of nodes for which predicted inflows are prepared on file.

NLIM(30) is the vector of nodes which have restrictions on their downstream releases.

NTRB(2,10) is the array of nodes which are tributaries. The first number of the pair releases its water to the second node.

NRF(3,10) is the array of node numbers and lag times for irrigation return flows. The first number of the triplet is the node where the irrigation flows are diverted. The second number is the node where some of the flows are included in the continuity equation. The final number is the time in quarter months the flows take to return.

C1(26,10) is the array of the percentage of irrigation delivery which returns every quarter month for each instance of return flow. The same array is used to store the values per time step.

The NEW common block has the arrays and vectors restricting the downstream releases to minimum or maximum limits.

FLOMIN(30) is the vector of minimum downstream releases in m^3/s at every restricted node.

FLOMAX(30) is the vector of maximum downstream releases in m^3/s at every restricted node.

FMX(14,30) is the array of maximum flow releases per time step in Mm^3 at every restricted node.

FMN(14,30) is the array of minimum flow releases per time step in Mm^3 at every restricted node.

The common block CAN stores the vectors of canal characteristics.

CCAPI(21) is the vector of the maximum capacities of the irrigation canals in m^3/s.

CCAPD(15) is the vector of the maximum capacities of the diversion canals in m^3/s.

CMINI(21) is the vector of the minimum capacities of the irrigation canals in m^3/s.

CMIND(15) is the vector of the minimum capacities of the diversion canals in m^3/s.

CHLSS(15) is the vector of the diversion canal channel losses as a percent of flow.

The CAP common block lists the vectors of constant turbine and reservoir storage restrictions.

TMAX(15) is the vector of maximum turbine releases in m^3/s.

TMIN(15) is the vector of minimum turbine releases in m^3/s.

SUP(25) is the vector of maximum storage levels in Mm^3.

SLO(25) is the vector of minimum storage levels in Mm^3.

The VAR common block has the arrays of coefficients varied by time step.

VLS(14,25) is the array of minimum desired reservoir storage levels in Mm^3 per time step.

VMS(14,25) is the array of maximum desired reservoir storage levels in Mm^3 per time step.

CIX(14,21) is the array of maximum desired irrigation canal flow in Mm^3 per time step.

CIN(14,21) is the array of minimum desired irrigation canal flow in Mm^3 per time step.

TMX(14,15) is the array of maximum desired turbine release in Mm^3 per time step.

TMN(14,15) is the array of minimum desired turbine release in Mm^3 per time step.

CDX(14,15) is the array of maximum desired diversion canal flow in Mm^3 per time step.

CDN(14,15) is the array of minimum desired diversion canal flow in Mm^3 per time step.

VL(14,15) is the array of channel loss as a percent of flow for each time step.

The STG common block has the vectors whose values are entered interactively after everything else has been entered.

STO(40) is the vector of current reservoir storage levels in Mm^3.

ST12(40) is the vector of target reservoir storage levels in Mm^3.

APOW(14) is a vector of minimum electrical production per time step in GW-hr or a vector of maximum energy difference allowed per time step.

ERHS(14) is the vector of the RHS constant terms of the linearized energy production equations for each time step. The same vector is used for the linearized equations of every power plant, one at a time.

The common block POW has the arrays of power plant coefficients.

H(14,15) is the array of power coefficients based on the effective head of each power plant for each time step.

C(14,15) is the array of corrector-multipliers for each power plant and for each time step. When equal to -1, the energy is calculated as the release multiplied by the head, there is no other term adjusting for the reservoir storage level.

STO(14,15) is the array of reference storage levels used to adjust the energy calculated for the reservoir storage level for each reservoir and each time step.

The common block RES has the arrays of reservoir coefficients.

GAMMA(25,9) is the array of storage volumes at the upper limit of each linearization interval for each reservoir.

DELTA(25,9) is the array of ratios of area to volume for each linearization interval at each reservoir.

EPSI(25,9) is the array of surface areas at the upper limit of each linearization interval for each reservoir.

KIN(25) is the vector of the number of linearization intervals used at each reservoir.

DGAMA(25,9) is the array of incremental storage volumes between the limits of each linearization interval for each reservoir.

CLOSS(14,25) is the array of losses due to seepage and evaporation for every time step and every reservoir.

The common block IRR stores the arrays of irrigation coefficients.

THETA(6,21,9) is an array of irrigation deficits values at the upper limit of each linearization interval for each irrigation node and for every curve of the season.

OMEGA(6,21,9) is an array of the ratios of penalty costs to deficits for each linearization interval for each irrigation node and for every curve of the season.

PHI(6,21,9) is an array of the penalty costs values at the upper limit of each linearization interval for each irrigation node and for every curve of the season.

ID(6,21) is the array of last quarter months that a penalty cost ratio or curve applies for an irrigation node.

KINI(6,21) is the array of the number of linearization intervals used for each penalty cost curve for every irrigation node.

DTHET(6,21,9) is the array of the incremental irrigation deficits between the limits of each linearization interval for each irrigation node and for every curve of the season.

IC(21) is the vector of the number of curves or single ratios used at an irrigation node for each season.

CWR(26,21) is the array of estimated crop water demands for every quarter month of the season from every irrigation node. The same array is used for the irrigation requirements by time step.

IID(14,21) is the array of which single penalty ratio or curve to use for every time step at every irrigation node.

The common block INF has a sole array.

FLOW(26,40) is an array of the predicted inflows at every node in the configuration and for every time step.

The RSC common block contains the costs used in the objective function of the linear programming formulation.

R01 is the relative price assigned to every GW-hr of energy produced.

R02 may be used for the relative weight of irrigation penalties. It is not presently used in PROJOP.

R03(40) is a vector of the relative benefit to assign to the storage volume of reservoirs fuller than the end of season target. The default value is 1 or the value of the preceding reservoir.

R0 is the penalty cost of a difference in energy production from one time step to another.

The final common block, called NOM, has character vectors.

NNAME(40) is the vector of the names of all the nodes in the configuration.

WREQ(21) is the vector of the mnenomic character codes which verify the data entered is assigned to the correct irrigation node.

EVAP(25) is the vector of the mnenomic character codes which verify the data entered is assigned to the correct reservoir node.

INFLOW(30) is the vector of the mnenomic character codes which verify the inflow data entered is for the correct node.

POWER(21) is the vector of the mnenomic character codes which verify the data entered is assigned to the correct power plant node.

```
c************************************************************************
C
C  This FORTRAN programme asks for the necessary data to write a SAS.OR
c linear programming data file, RELEASE.DATA in drive B:, which may be
c solved to find the optimal release schedule for hydro-electric power
c generation, yet still satisfy the irrigation requirements of the
c current season's crops and anticipated needs of later seasons.  It is
c dimensioned for 15 power plants and 20 irrigation districts in a
c system with 40 or less nodes, 25 which may have storage facilities
c and 15 which may divert water from one basin to another.
c  A weighted objective function is used to evaluate the returns from
c energy production without unduly penalizing irrigation in the current
c and future seasons. An optional term may be used in the objective
c function to reduce fluctuations in the amount of energy generated from
c one time step to the next.
c  The formulation is subject to the continuity equations of the water
c allocation among the nodes for every time step in the season.  Time
c steps of a quarter month are used for the immediate two months or until
c there are an integer number of months left in the season.  Month or half
c month time steps are used for the remaining time.  The solution is
c updated every quarter month to reflect actual reservoir levels, revised
c flow forecasts, and operational schedules.
c  PC files may be used to store the various information required each
c run.  The program may be used interactively to update and vary the
c information for each time.  Checking procedures verify that the correct
c file is being entered and that all the information necessary has been
c entered.
c************************************************************************
      COMMON/SHR/ NPOWER(15), NIRR(21), NRES(25), NNOSPL(20), NDIV(2,15)
      COMMON/SHP/ NINFLO(30), NLIM(31), NTRB(2,10), NRF(3,10), C1(26,10)
      COMMON/NEW/ FLOMIN(30), FLOMAX(30), FMX(14,30), FMN(14,30)
      COMMON/CAN/ CCAPI(21), CCAPD(15), CMINI(21), CMIND(15), CHLSS(15)
      COMMON/CAP/ TMAX(15), TMIN(15), SUP(25), SLO(25)
      COMMON/VAR/ VLS(14,25),VMS(14,25),CIX(14,21),CIN(14,21),
     1          TMX(14,15), TMN(14,15), CDX(14,15), CDN(14,15), VL(14,15)
      COMMON/STG/ ST0(40), ST12(40), APOW(14), ERHS(14)
      COMMON/POW/ H(14,15), C(14,15), STO(14,15)
      COMMON/RES/ GAMMA(25,9), DELTA(25,9), EPSI(25,9), KIN(25),
     1             DGAMA(25,9), CLOSS(14,25)
      COMMON/IRR/ THETA(6,21,9), OMEGA(6,21,9), PHI(6,21,9), ID(6,21),
     1          KINI(6,21), DTHET(6,21,9), IC(21), CWR(26,21), IID(14,21)
      COMMON/INF/ FLOW(26,40)
      COMMON/RSC/ RO1, RO2, RO3(40), RO4
      COMMON/NOM/ NNAME(40), WREQ(21), EVAP(25), INFLOW(30), POWER(15)
c
      INTEGER CT1, CT2, CT3, CT4, CT5, CT6, CT7
      CHARACTER * 47 CHAR(2)
      CHARACTER * 12 NAME1, NNAME, WREQ, EVAP, FNAME, INFLOW, POWER,

     1               FIRST(6), FNAME1, FNAME2, FNAME3, FNAME4
      CHARACTER * 4 NAME2, NAME3, NAME4
      CHARACTER * 4 NAME5, NAME6, NAME7, NAME8
```

```
      CHARACTER * 2 ANS
c
      DATA CT1/0/,CT2/0/,CT3/0/,CT4/0/,CT5/0/,CT6/0/,CT7/0/,CT8/0/
      DATA CHAR/' minimum power production per period ( GW-hr ) ',
     1         ' maximum power difference per period ( GW-hr ) '/
C
1     WRITE ( *,1000 )
      READ  ( *,9000 ) FNAME1
      IF ( FNAME1 .EQ. '            ' ) GO TO 2
C  Read in the configuration file
      OPEN ( 9,FILE=FNAME1, STATUS='OLD')
      READ ( 9,9100 ) ( FIRST(M), M=1,6 )
      READ ( 9,9009 ) NI, NP, NR, NS, ND, NF, NT, NLIM(1), NN
      READ ( 9,9010 ) ( NIRR(K), CCAPI(K), WREQ(K), K=1,NI )
      READ ( 9,9012 ) ( NPOWER(K), POWER(K), K=1,NP )
      READ ( 9,9012 ) ( NRES(K), EVAP(K), K=1,NR )
      READ ( 9,9011 ) ( NNOSPL(K), K=1,NS )
      READ ( 9,9011 ) ( NDIV(1,K), NDIV(2,K), K=1,ND )
      READ ( 9,9013 ) ( CCAPD(K), K=1,ND )
      READ ( 9,9012 ) ( NINFLO(K), INFLOW(K), K=1,NF )
      READ ( 9,9011 ) ( NTRB(1,K), NTRB(2,K), K=1,NT )
      READ ( 9,9012 ) ( K, NNAME(K), K=1,NN )
      READ ( 9,9014 ) ( NLIM(K+1),FLOMIN(K),FLOMAX(K), K=1,NLIM(1)-1 )
      WRITE ( *,3006 ) FNAME1, ( FIRST(K), K=1,6 )
      CLOSE ( 9 )
      CT1 = 1
      GO TO 3
C  Enter new configuration
2     CALL CONFIG(1,NI,NP,NR,NS,ND,NF,NT,NN)
c  1 = code for initial entry of a configuration
c NI = two more than the number of irrigation nodes
c NP = two more than the number of nodes with power plants
c NR = two more than the number of nodes with reservoirs
c NS = two more than the number of nodes which do not release to the
c      node numbered one higher
c ND = two more than the number of nodes with diversions
c NF = two more than the number of nodes with downstream flow
c      restrictions
c NT = two more than the number of nodes which are tributaries to
c      other nodes in the configuration
c NN = the number of nodes in the configuration
      CT1 = 1
C Display main menu, progressing as the files are read
3     MINCT = CT1 + CT2 + CT3 + CT4 + CT5 + CT6 + CT7
      WRITE ( *,1002 ) (  I+1, I = 1,MINCT+1 )
      READ  ( *,9002 ) ICHOIS
      GO TO ( 809, 101, 202, 303, 404, 505, 606, 707, 808, 909 ) ICHOIS
      IF ( ICHOIS .GT. 10 ) GO TO 3
      GO TO 1
c Display configuration data and verify its accuracy:
101            IS = 2
               IG = 2
```

```
              IR = 2
              IP = 2
              IL = 2
              IT = 2
c  Display REVIEW table, 6 lines at a time
     NBEG = 1
     NTIMES = NN/6 + 1
     DO 205 L=1,NTIMES
       WRITE ( *,2000 )
       NEND = 5 + NBEG
       IF ( NEND .GT. NN ) NEND = NN
     DO 200 I=NBEG, NEND
c  Assign values for each column
5        NAME1 = NNAME(I)
         DO 10 IM = 2,(NDIV(1,1)+1)
           IF ( NDIV(2,IM) .EQ. I ) GO TO 12
10       CONTINUE
           NAME4='—'
           GO TO 14
12       NAME4 = NNAME( NDIV(1,IM) )
14       DO 15 II = 2,(NDIV(1,1)+1)
           IF ( NDIV(1,II) .EQ. I ) GO TO 18
15       CONTINUE
           NAME3='—'
           CAP1= 0.0
          GO TO 20
18       NAME3 = NNAME(NDIV(2,II))
         CAP1 = CCAPD(II)
20        IF ( I .EQ. NNOSPL(IS) ) GO TO 25
23        NAME2 = NNAME(I+1)
           GO TO 30
25        IS = IS + 1
         IF ( NTRB(1,IT) .EQ. I ) GO TO 27
         NAME2 = '—'
           GO TO 30
27        NAME2 = NNAME( NTRB(2,IT) )
         IT = IT + 1
30        NAME5 = '—'
          CAP2 = 0.0
        IF ( NIRR(IG) .EQ. I ) GO TO 35
          GO TO 40
35        NAME5 = WREQ(IG)
          CAP2 = CCAPI(IG)
            IG = IG + 1
40      NAME6 = '—'
        IF ( NRES(IR) .EQ. I ) GO TO 45
          GO TO 50
45        NAME6 = EVAP(IR)
          IR = IR + 1
50      NAME7 = '—'
        IF ( NPOWER(IP) .EQ. I ) GO TO 55
          GO TO 60
```

```
55        NAME7 = POWER(IP)
          IP = IP + 1
60       NAME8 = '---'
         IF ( NINFLO(IL) .EQ. I ) GO TO 65
          GO TO 100
65        NAME8 = INFLOW(IL)
          IL = IL + 1
c  Display table to determine if corrections are required
1 0 0       W R I T E ( * , 3 0 0 0 )
NAME1,NAME2,NAME3,CAP1,NAME4,NAME5,CAP2,NAME6,NAME7
     1          ,NAME8
      READ  (*,9002) INT
      IF ( INT .LE. 0 ) GO TO 200
      IF ( INT .GE. 99 ) GO TO 210
c  Correction to a column required
      CALL CORRIG (INT,I,NN,IX,IM,II,IS, IG, IR, IP, IL, IT,NAME3,NAME4)
c  INT = number of incorrect column
c    I = node number of current line
c   NN = total number of nodes
c   IX = code to skip replacement
c   IM = diversion TO number of current line
c   II = diversion FROM number of current line
c   IS = nospill number of current line
c   IG = irrigation number of current line
c   IR = reservoir number of current line
c   IP = power plant number of current line
c   IL = local inflow number of current line
c   IT = tributary number of current line
c NAME3, NAME4 = column names which may be redefined by CORRIG directly
      WRITE ( *,2000 )
      GO TO ( 110, 120, 130, 140, 100, 160, 170, 180, 190 ) INT
C  The local inflow has been corrected
      NAME8 = INFLOW(IL-1)
      GO TO 100
110   NAME1 = NNAME(I)
      IF ( IX .LT. 9 ) GO TO 100
      GO TO 200
120   NAME2 = 'PASS'
      GO TO 100
130    CAP1 = CCAPD(NDIV(1,1+1))
      GO TO 100
140    CAP1 = CCAPD(II)
      GO TO 100
160   NAME5 = WREQ(IG-1)
       CAP2 = CCAPI(IG-1)
      GO TO 100
170    CAP2 = CCAPI(IG-1)
      GO TO 100
180   NAME6 = EVAP(IR-1)
      GO TO 100
190   NAME7 = POWER(IP-1)
      GO TO 100
```

```
200   CONTINUE
      WRITE ( *,3001 )
      READ  ( *,9001 ) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 101
      NBEG = NEND + 1
205   CONTINUE
c  Amend total values in case there were changes
210          NI = 1 + NIRR(1)
             NP = 1 + NPOWER(1)
             NR = 1 + NRES(1)
             NS = 1 + NNOSPL(1)
             ND = 1 + NDIV(1,1)
             NT = 1 + NTRB(1,1)
             NF = 1 + NINFLO(1)
      WRITE (*,3003)
      READ  (*,9001) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 215
        GO TO 220
C  Write updated version of the configuration to the diskette
215   WRITE ( *,3004 ) FNAME1
      READ  ( *,9000 ) NAME1
      IF ( NAME1 .EQ. '            ' ) GO TO 216
        FNAME1 = NAME1
        WRITE ( *,3005 ) FNAME1
        READ  ( *,9100 ) ( FIRST(M), M=1,6 )
216   OPEN ( 10,FILE=FNAME1, STATUS='OLD')
      WRITE ( 10,9100 ) ( FIRST(M), M=1,6 )
      WRITE ( 10,9009 ) NI, NP, NR, NS, ND, NF, NT, NLIM(1), NN
      WRITE ( 10,9010 ) ( NIRR(K), CCAPI(K), WREQ(K), K=1,NI )
      WRITE ( 10,9012 ) ( NPOWER(K), POWER(K), K=1,NP )
      WRITE ( 10,9012 ) ( NRES(K), EVAP(K), K=1,NR )
      WRITE ( 10,9011 ) ( NNOSPL(K), K=1,NS )
      WRITE ( 10,9011 ) ( NDIV(1,K), NDIV(2,K), K=1,ND )
      WRITE ( 10,9013 ) ( CCAPD(K), K=1,ND )
      WRITE ( 10,9012 ) ( NINFLO(K), INFLOW(K), K=1,NF )
      WRITE ( 10,9011 ) ( NTRB(1,K), NTRB(2,K), K=1,NT )
      WRITE ( 10,9012 ) ( K, NNAME(K), K=1,NN )
      WRITE ( 10,9014 ) ( NLIM(K+1),FLOMIN(K),FLOMAX(K), K=1,NLIM(1)-1 )
      CLOSE ( 10 )
220   WRITE (*,3002)
      READ  (*,9001) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) CALL CONFIG ( 2, NI, NP, NR,
     1                                        NS, ND, NF, NT, NN )
c  2 = code for adding to an existing configuration
c NI = two more than the number of irrigation nodes
c NP = two more than the number of nodes with power plants
c NR = two more than the number of nodes with reservoirs
c NS = two more than the number of nodes which do not release to the
c      node numbered one higher
c ND = two more than the number of nodes with diversions
c NF = two more than the number of nodes with downstream flow
c      restrictions
```

```
c NT = two more than the number of nodes which are tributaries to
c      other nodes in the configuration
c NN = the number of nodes in the configuration
      GO TO 3
300    CONTINUE
C Read in Power coefficients and verify their accuracy
202   CALL DPOWER ( CT2 )
c  CT2 = integer code (1 or 0 ) indicating entry of power constants
c        complete
      GO TO 3
C  Read the reservoir data and verify that it is up to date
303   CALL DRES ( CT3 )
c  CT3 = integer code (1 or 0 ) indicating entry of reservoir data
c        complete
      GO TO 3
C  Read the irrigation data and verify that it is up to date
404   CALL DIRR ( CT4, KVAR )
c  CT4 = integer code (1 or 0 ) indicating entry of irrigation data
c        complete
c KVAR = integer code (1 or 0 ) indicating whether irrigation arrays
         have already been adjusted for current time steps
      GO TO 3
C  Enter canal data and verify that it is up to date
505   CALL DCANAL ( ND, NI, CT5 )
c   ND = number of inter-basin diversions
c   NI = number of irrigation diversions
c  CT5 = integer code (1 or 0 ) indicating entry of canal constants
c        complete
      GO TO 3
C  Enter starting date, but only if all constant data entered
606   IF ( MINCT .LT. 5 ) GO TO 3
      CALL PERIOD ( NN, I4, II, IDATE, CT6, KVAR )
c    NN = total number of nodes
c    I4 = number of time steps of a quarter month in the current run
c    II = total number of time steps in the current run
c IDATE = the order that the first quarter month of the current run is
c         in the season
c   CT6 = integer code (1 or 0 ) indicating entry of starting date and
c         conversions of flow rates to flow volumes is complete
c  KVAR = integer code (1 or 0 ) indicating that irrigation arrays have
c         already been adjusted for current time steps
      GO TO 3
C  Read the inflow data
707   IF ( CT7 .GT. 0 ) GO TO 710
      DO 705 I=1,II
      DO 705 N=1,NN
705     FLOW(I,N) = 0.0
710   CALL DINFLO ( I4, II, NN, CT7 )
c    I4 = number of time steps of a quarter month in the current run
c    II = total number of time steps in the current run
c    NN = total number of nodes
c   CT7 = integer code (1 or 0 ) indicating entry of inflow data is
```

```
c         complete
      GO TO 3
C  Read the irrigation return flow data
808   CALL RETQR  ( NN, I4, II, NNN, IDATE )
c     NN = total number of nodes
c     I4 = number of time steps of a quarter month in the current run
c     II = total number of time steps in the current run
c   NNN = number of instances of return flow
c IDATE = the order that the first quarter month of the current run is
c         in the season
      GO TO 3
C  Review or amend restrictions to the regulated downstream releases
809   WRITE ( *,1003 )
      WRITE ( *,1004 ) ( K, NNAME(NLIM(K+1)), FLOMIN(K), FLOMAX(K),
     1                       K=1,NLIM(1)-1 )
      WRITE ( *,1005 )
      READ  ( *,9002 ) ICH
      GO TO ( 810, 820, 830, 215 ) ICH
      GO TO 3
C  Additions, deletions, or changes are required to NLIM, FLOMAX, and FLOMIN
810   WRITE ( *,1006 )
      READ  ( *,9002 ) NX
       IF ( NX .LE. 0 ) GO TO 80
      WRITE ( *,1007 ) ( N, NNAME(N), N=1,NN )
         DO 812 N=1,NX
            WRITE ( *,1008 ) N
            READ  ( *,9002 ) NNEW
            WRITE ( *,1009 ) NNAME(NNEW)
            READ  ( *,9003 ) FLOMIN(NLIM(1))
            WRITE ( *,1010 ) NNAME(NNEW)
            READ  ( *,9003 ) FLOMAX(NLIM(1))
              IF ( FLOMAX(NLIM(1)) .LE. 0.05 ) FLOMAX(NLIM(1)) = 99999
            NLIM(1) = NLIM(1) + 1
812         NLIM(NLIM(1)) = NNEW
         GO TO 80
C  Deletions
820   WRITE ( *,1011 )
      READ  ( *,9002 ) NX
        IF ( NX .LE. 0 ) GO TO 80
         DO 826 N=1,NX
            WRITE ( *,1012 ) N
            READ  ( *,9002 ) KX
            DO 824 M=KX,NLIM(1)-2
              FLOMAX(M) = FLOMAX(M+1)
              FLOMIN(M) = FLOMIN(M+1)
824           NLIM(M+1) = NLIM(M+2)
826         NLIM(1) = NLIM(1) - 1
            GO TO 80
C  Corrections
830   WRITE ( *,1013 )
      READ  ( *,9002 ) NX
       IF ( NX .LE. 0 ) GO TO 80
```

```
          DO 835 N=1,NX
            WRITE ( *,1015 )
            READ  ( *,9002 ) KX
832          WRITE ( *,1016 ) KX
            READ  ( *,9000 ) NAME1
             IF ( NAME1 .EQ. '            ' ) GO TO 833
C  The correct min/max flow values were assigned to the wrong node
C  Find the node number of correct node
      DO 855 K=1,NN
        IF ( NNAME(K) .EQ. NAME1 ) GO TO 858
855   CONTINUE
      WRITE (*,1014) NAME1
      GO TO 832
C  Replace the node number, NLIM(KX), with the correct number, K
858     NLIM(KX+1) = K
833          WRITE ( *,1017 ) NNAME(NLIM(KX+1))
            READ ( *,9003 ) DUMMY
              IF ( DUMMY .GT. 0.005 ) FLOMIN(KX) = DUMMY
            WRITE ( *,1018 ) NNAME(NLIM(KX+1))
            READ ( *,9003 ) DUMMY
              IF ( DUMMY .GT. 0.005 ) FLOMAX(KX) = DUMMY
835   CONTINUE
      GO TO 80
C  Verify all data has been read and reviewed from the files
909   IF ( MINCT .LT. 7 ) GO TO 3
C   RELEASE.DATA is written to the diskette only once everything is
C  read in and ll known.
C           Enter current reservoir storage levels interactively
      WRITE ( *,4000 )
      DO 910 N=2,NR
        WRITE ( *,9004 ) NNAME(NRES(N))
        READ  ( *,9003 ) STO(NRES(N))
        IF ( STO(NRES(N)) .LE. 0 ) STO(NRES(N)) = SLO(N-1)
910   CONTINUE
913   WRITE ( *,4005 )
      WRITE ( *,4001 ) ( N, NNAME(NRES(N+1)), STO(NRES(N+1)),
     1                        N=1,NRES(1) )
      WRITE ( *,4002 )
      READ  ( *,9001 ) ANS
C  If any corrections are needed, correct
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 917
        GO TO 918
917   WRITE ( *,4003 )
      READ  ( *,9002 ) NX
          NX = NX+1
      WRITE ( *,4004 ) NNAME(NRES(NX))
      READ  ( *,9003 ) STO(NRES(NX))
      IF ( STO(NRES(NX)) .LE. 0 ) STO(NRES(NX)) = SLO(NX-1)
        GO TO 913
C  Add the current reservoir storage to the inflow in the first period
918   DO 919 N=2,NR
919     FLOW(1,NRES(N)) = FLOW(1,NRES(N)) + STO(NRES(N))
```

```
C  Enter final storage targets interactively
920   WRITE ( *,4010 )
      DO 922 N=2,NR
        WRITE ( *,9004 ) NNAME(NRES(N))
        READ  ( *,9003 ) ST12(NRES(N))
        IF ( ST12(NRES(N)) .LE. 0 ) ST12(NRES(N)) = SLO(N-1)
922   CONTINUE
925   WRITE ( *,4011 )
      WRITE ( *,4001 ) ( N, NNAME(NRES(N+1)), ST12(NRES(N+1)),
     1                         N=1,NRES(1) )
      WRITE ( *,4002 )
      READ  ( *,9001 ) ANS
C  If any corrections are needed, correct
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 927
        GO TO 930
927   WRITE ( *,4013 )
      READ  ( *,9002 ) NX
           NX = NX + 1
      WRITE ( *,4014 ) NNAME(NRES(NX))
      READ  ( *,9003 ) ST12(NRES(NX))
      IF ( ST12(NRES(NX)) .LE. 0 ) ST12(NRES(NX)) = SLO(NX-1)
        GO TO 925
C  Ask for cost to assign benefits of extra storage, RO3
930   WRITE ( *,4020 )
        RO3(NRES(1)) = 1
      DO 932 N=2,NR
        WRITE ( *,9004 ) NNAME(NRES(N))
        READ  ( *,9003 ) RO3(NRES(N))
        IF ( RO3(NRES(N)) .LE. 0 ) RO3(NRES(N)) = RO3(NRES(N-1))
932   CONTINUE
935   WRITE ( *,4021 )
      WRITE ( *,4001 ) ( N, NNAME(NRES(N+1)), RO3(NRES(N+1)),
     1                           N=1,NRES(1) )
      WRITE ( *,4002 )
      READ  ( *,9001 ) ANS
C  If any corrections are needed, correct
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 937
        GO TO 940
937   WRITE ( *,4023 )
      READ  ( *,9002 ) NX
            NX = NX + 1
      WRITE ( *,4024 ) NNAME(NRES(NX))
      READ  ( *,9003 ) RO3(NRES(NX))
      IF ( RO3(NRES(NX)) .LE. 0 ) RO3(NRES(NX)) = RO3(NRES(NX-1))
        GO TO 935
C  Enter RO1, RO2, RO4 and MINPOW interactively
940   WRITE ( *,4040 )
      READ  ( *,9003 ) RO1
      WRITE ( *,4041 )
      READ  ( *,9002 ) MINPOW
      IF ( MINPOW .LT. 3 ) GO TO 950
         WRITE ( *,4045 )
```

```
        READ  ( *,9003 ) RO4
C  Verify values entered correctly
950   WRITE ( *,4050 ) RO1, MINPOW, RO4
      READ  ( *,9001 ) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 940
C  If MINPOW > 0 and < 3 enter APOW
      IF ( MINPOW .LE. 0 .OR. MINPOW .GT. 2 ) GO TO 999
      WRITE ( *,4060 ) CHAR(MINPOW), I4
      DO 960 I=1,II
        WRITE ( *,9005 ) I
960     READ  ( *,9003 ) APOW(I)
963   WRITE ( *,4062 ) CHAR(MINPOW)
      WRITE ( *,4061 ) ( I, APOW(I), I=1,II )
      WRITE ( *,4002 )
      READ  ( *,9001 ) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 967
        GO TO 999
967   WRITE ( *,4063 )
      READ  ( *,9002 ) NX
      WRITE ( *,4064 ) CHAR(MINPOW), NX
      READ  ( *,9003 ) APOW(NX)
      GO TO 963
C  All set to write to b:RELEASE.DATA
999   CALL DWRITE ( NN, II, I4, NNN, MINPOW )
c     NN = total number of nodes
c     II = total number of time steps in the current run
c     I4 = number of time steps of a quarter month in the current run
c MINPOW = integer code for the variation of the formulation desired
c          ( 0 - allow large fluctuations
C             1 - meet minimum energy targets each time step
c             2 - limit the fluctuation to a maximum value
c             3 - minimize the fluctuations with objective function )
c
C
1000  FORMAT ( ' This program reads the datadisks and calls other subrou
     1tines to prepare the fileb:RELEASE.DATA  containing the SAS.OR spa
     2rsedata file  for a linear programming solution of the weekly  res
     3ervoir  releases  which  will  maximize  electrical  production.'/
     4' What is the  name  of  the file  containing  the  configuration?
     5  Enter b:FNAME,  where FNAME is the name in 10 or less letters. '
     6/ ' If you do not enter a name before RETURN you may enter an enti
     7rely new'/' configuration. ' )
1002  FORMAT ( ' What would you like to do?'/' Answer 0 or nothing to RE
     1-ENTER the current configuration. '/ 8X,'1 to review or amend rest
     2rictions to regulated downstream releases'/ 6X,I3,' to REVIEW the
     3current configuration, including updating or storing'/   6X,I3,' t
     3o enter or amend the file of power data'/ 6X,I3,' to enter or amen
     4d the file of reservoir data'/ 6X,I3,' to enter or amend the file
     5 of irrigation data'/  6X,I3,' to enter or amend the file of canal
     6 data '/  6X,I3,' to enter the starting date'/  6X,I3,' to enter o
     7r amend the file of inflow data'/  6X,I3,' to enter or amend the f
     8ile of irrigation return flow data'/8X,' 10 when all datafiles hav
```

```
     9e been entered and corrected. ' )
1003  FORMAT ( ' The following nodes have these restrictions on their'/'
     1 regulated downstream releases:'/' 99999.0 is for unlimited flow'
     2/15X,4HNAME,9X,9HMIN. FLOW,3X,9HMAX. FLOW/30X,4Hm3/s,8X,4Hm3/s/ )
1004  FORMAT ( I3,9X,A12,2X,F8.2,F13.1 )
1005  FORMAT ( /' Is this table correct?  Enter the code for the error:'
     1/' 0 or nothing to continue with these values'/
     2' 1 to add more nodes to the list'/
     3' 2 to remove some nodes from the list'/
     4' 3 to correct a value listed'/' 4 to save any corrections ' )
1006  FORMAT ( ' How many more nodes do you wish to add?  Enter an INTEG
     1ER value. ' )
1007  FORMAT ( I3,2X,A12, I3,2X,A12, I3,2X,A12, I3,2X,A12 )
1008  FORMAT ( ' What is the number of the ',I2,'st node which now has r
     1estrictions to the regulated releases?  Enter an INTEGER value ' )
1009  FORMAT ( ' What is the minimum flow in m3/s above which the regula
     1ted downstream releases  of ',A15,' must be? ' )
1010  FORMAT ( ' What is the maximum flow in m3/s below which the regula
     1ted downstream releases  of ',A15,' must be? ' )
1011  FORMAT ( ' How many nodes do you wish to remove?  Enter an INTEGER
     1 number. ' )
1012  FORMAT ( ' What is the number to the left of the node to be remove
     1d the ',I2,'st time?   Enter the latter numbers BEFORE the first n
     2umbers when deleting more than one node.'/' Enter an INTEGER numbe
     3r. ' )
1013  FORMAT ( ' How many nodes do you wish to correct? ' )
1014  FORMAT ( ' No  node of the name ',A15, ' was found.  Is it correct
     2ly spelt in full? ')
1015  FORMAT ( ' What is the number to the left of the line in which an
     1error occurs?  Enter an  INTEGER number. ' )
1016  FORMAT ( ' Enter the correct name for the node in line ',I3,'.  If
     1 the name is correct, enter nothing before RETURN. ' )
1017  FORMAT ( ' What is the correct minimum flow in m3/s above which th
     1e regulated downstream'/' releases of ',A15,' must be?'/' If the v
     2alue listed was correct, enter nothing before the RETURN. ' )
1018  FORMAT ( ' What is the correct maximum flow in m3/s below which th
     1e regulated downstream'/' releases of ',A15,' must be?'/' If the v
     2alue listed was correct, enter nothing before the RETURN. ' )
2000  FORMAT ( '    If the current line in the table has any incorrect e
     1ntries, enter the INTEGERabove one column which is not correct.  E
     2nter 0 or nothing before the RETURN to continue to the next node.'
     3/' Enter 99 if you wish to store what you have updated so far.'//
     4  5X,2H1.,9X,2H2.,
     54X,2H3.,6X,2H4.,6X,2H5.,4X,2H6.,8X,2H7.,6X,2H8.,4X,2H9.,4X,3H10.
     6  /4X,3HTHE,8X,4HDOWN,3X,13HDI- DIVERSION,3X,41HRE-  IRR.  IRRIGAT
     7ION  RES.  POWER  LOCAL/3X,5H NODE,5X,34HSTREAM  VERTS CAPACITY
CE
     8IVES REQ.,4X,28HCAPACITY NAME  COEF. INFLOW/14X,5H NODE,3X,3HTO:,
     95X,4Hm3/s,3X,10HFROM   NAME,7X,4Hm3/s,9X,10HNAME   NAME,/ )
3000  FORMAT ( 1X,A12,2X,A4,2X,A4,F10.3,2X,A4,2X,A4,2X,F10.3,2X,A4,2X,A4
     1,2X,A4,2H ? )
3001  FORMAT ( ' Do you wish to see the  nodes again?  Answer Y to revie
```

```
     1w the  nodes already dis-played. ' )
3002  FORMAT ( ' Do you wish to add  nodes to the configuration?   Answe
     1r Y if you do. ' )
3003  FORMAT ( ' Do you wish to store the updated version?  Answer Y if
     1you do: ' )
3004  FORMAT ( ' Do you  wish to save  the  configuration under ',A12,
     1 '?   You  may  change the  diskette in drive b: before entering.'
     2/ ' Enter b:FNAME, where FNAME is the name of an existing file in
     310 or less letters to change the name.'/ ' If you enter nothing be
     4fore RETURN, the name will stay the same.'/ )
3005  FORMAT ( ' Enter a title for the configuration file,',A12,', in 72
     1 characters or less'/ )
3006  FORMAT( ' The configuration file,',A12,' has the title:'/ 1X,6A12)
4000  FORMAT ( ' Please enter the CURRENT RESERVOIR STORAGE LEVELS (
MC
     1M ) for each reservoir as prompted.  The minimum reservoir level i
     2s assigned by default' )
4001  FORMAT ( I3,2X,A12,F7.3,4X,I3,2X,A12,F7.3,4X,I3,2X,A12,F7.3 )
4002  FORMAT ( ' Do you wish to change any value shown above?  Answer Y
     1to correct a value ' )
4003  FORMAT ( ' What is the number to the left of the reservoir whose c
     1urrent storage level is  incorrect?  Enter an INTEGER ' )
4004  FORMAT ( ' What is the correct volume currently in the reservoir a
     1t ',A12/' in MCM? ' )
4005  FORMAT ( ' These are the current reservoir storage volumes in each
     1 reservoir. ' )
4011  FORMAT ( ' These are the target storage volumes desired in each re
     1servoir at the end of theseason. ' )
4021  FORMAT ( ' These are the benefits ( Rs/MCM ) to assign to each res
     1ervoir with extra water  stored at the end of the season. ' )
4062  FORMAT ( ' These are the ',A47 )
4010  FORMAT ( ' Please enter the TARGET STORAGE VOLUMES desired at the
     1end of the growing seasonThe minimum reservoir level is assigned b
     2y default. ' )
4013  FORMAT ( ' What is the number to the left of the reservoir whose t
     1arget volume is incorrect?  Enter an INTEGER ' )
4014  FORMAT ( ' What is the correct target volume for the reservoir at
     1the node ',A12,' in MCM? ' )
4020  FORMAT ( ' Please enter the BENEFITS to assign to extra reservoir
     1storage volume ( Rs/MCM )as prompted.  The value of the preceeding
     2 reservoir is assigned by default'/' ( 1 Rs/MCM for the first ).')
4023  FORMAT ( ' What is the number to the left of the reservoir whose b
     1enefits are incorrect?     Enter an INTEGER ' )
4024  FORMAT ( ' What are the correct benefits to assign to extra storag
     1e at the ',A12/' reservoir? ' )
4040  FORMAT ( ' Please enter the price ( Rs/GW-hr ) to assign to the po
     1wer generated by power   plants in the project ' )
4041  FORMAT ( ' Please enter the code for minimizing the differences in
     1 weekly power production.Enter 0 if you wish to ignore this constr
     2aint completely'/7X,'1 if you wish to enter minimum power producti
     3on per period'/7X,'2 if you wish to limit the difference between p
     4eriods'/7x,'3 if you wish to minimize the occurence of any differe
```

```
     5nce in power '/' production from one period and the next ' )
4045  FORMAT ( ' Please enter the penalty cost ( Rs/GW-hr ) to assign th
     1e difference in weekly   power production. ' )
4050  FORMAT ( ' These are the values you have entered:'/' PRICE OF POWE
     1R: ',5X,F10.2/ 10X,' CODE:',5X,I8/ 6X,' PENALTY COST:',2X,F10.2,2X
     2,'( used when code = 3 )'/' Are any of the values incorrect?  Answ
     3er Y to re-enter. ' )
4060  FORMAT ( ' Please enter the ',A47,' as prompted.'/' Note that the
     1first', I3,' periods are one fourth as long as the others and in t
     2he  Maha season the last period is a half-month long. ' )
4061  FORMAT ( I3,2X,F10.3,2X,I3,2X,F10.3,2X,I3,2X,F10.3,2X,I3,2X,F10.3)
4063  FORMAT ( ' What is the number to the left of the incorrect value?
     1 Enter an INTEGER ' )
4064  FORMAT ( ' What is the correct ',A47,' for the',I3,'th period? ' )
9000  FORMAT ( A12 )
9001  FORMAT ( A2 )
9002  FORMAT ( I5 )
9003  FORMAT ( F10.0 )
9004  FORMAT ( 1X, A12,1X )
9005  FORMAT ( I5,3X )
9009  FORMAT ( 9I4 )
9010  FORMAT ( I5,F10.2,A15 )
9011  FORMAT ( 20I4 )
9012  FORMAT ( I5, A15, I5, A15, I5, A15, I5, A15 )
9013  FORMAT ( 8F10.3 )
9014  FORMAT ( I5,2F12.2,1X,I5,2F12.2 )
9100  FORMAT ( 6A12 )
9101  FORMAT ( 1X,A4,2F10.0,I3 )
9102  FORMAT ( 1X,12F6.0 )
9103  FORMAT ( 1X,A4,18I4 )
      STOP
      END
^Z
      SUBROUTINE CORRIG ( NX, I, NL, IX, IM, II, IS, IG, IR, IP, IL, IT,
     1                    NAME3, NAME4 )
c   NX = number of incorrect column
c    I = node number of current line
c   NL = total number of nodes
c   IX = code to skip replacement
c   IM = diversion TO number of current line
c   II = diversion FROM number of current line
c   IS = nospill number of current line
c   IG = irrigation number of current line
c   IR = reservoir number of current line
c   IP = power plant number of current line
c   IL = local inflow number of current line
c   IT = tributary number of current line
c NAME3, NAME4 = column names which may be redefined by CORRIG directly
c
C This subroutine corrects the entries in the REVIEW table, calling
C  other subroutines as required
      COMMON/SHR/ NPOWER(15), NIRR(21), NRES(25), NNOSPL(20), NDIV(2,15)
```

```
      COMMON/SHP/ NINFLO(30), NLIM(31), NTRB(2,10), NRF(3,10)
      COMMON/NEW/ FLOMIN(30), FLOMAX(30)
      COMMON/CAN/ CCAPI(21), CCAPD(15), CMINI(21), CMIND(15), CHLSS(15)
      COMMON/NOM/ NNAME(40), WREQ(21), EVAP(25), INFLOW(30), POWER(15)
      CHARACTER * 12 NNAME, WREQ, EVAP, FNAME, INFLOW, POWER
      CHARACTER * 4 NAME3, NAME4
      CHARACTER * 2 ANS
C
      GO TO ( 100, 200, 300, 400, 500, 600, 700, 800, 900, 1010 ) NX
      GO TO 1001
C  There was an error in the name of the  node or the user wishes to
C            delete this node entirely
100   WRITE (*,1002) NNAME(I)
      READ  (*,9001) ANS
       IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 105
        WRITE (*,1000)
        READ  (*,9000) NNAME(I)
          IX = 0
         GO TO 1001
C  Delete the node from all vectors
105       I = I - 1
      CALL DELETE ( I, NL )
          IX = 10
        GO TO 1001
C  There was an error in the downstream  node
200   WRITE (*,2000) NNAME(I+1)
      READ  (*,9002) IX
           GO TO ( 201, 230, 250, 290, 65 ) IX
           GO TO 1001
C  Delete the downstream  node
201    CALL DELETE ( I, NL )
           GO TO 1001
C  Insert  downstream  nodes
230   CALL ADD (I,NL,IM,II,IS,IG,IR,IP,IL,IT )
         GO TO 1001
C Remove this  node from the NNOSPL vector, first checking that it does not
C already flow to a  node on another stream.
250   DO 260 K=2,NTRB(1,1) + 1
       IF ( NTRB(1,K) .EQ. I ) GO TO 265
260   CONTINUE
      GO TO 275
265   WRITE ( *,2003 ) NNAME(I), NNAME(NTRB(2,K)), NNAME(I), NNAME(I+1)
      READ  ( *,9001 ) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' )  GO TO 200
C  A tributary did exist at this  node and is now deleted
       DO 270 KK=K,NTRB(1,1)
         KPLUS = KK + 1
      NTRB(1,KK) = NTRB(1,KPLUS)
270    NTRB(2,KK) = NTRB(2,KPLUS)
       NTRB(1,1) = NTRB(1,1) - 1
C  The  node now flows to the  node numbered one higher
275   DO 280 K=(IS-1),NNOSPL(1)
```

```
280     NNOSPL(K) = NNOSPL(K+1)
       NNOSPL(1) = NNOSPL(1) - 1
             IT = IT + 1
        GO TO 1001
C The  node belongs in the NOSPL vector and is not, or is,  but has no
C       trib  node listed which it should
290   IF ( NNOSPL(IS-1) .EQ. I ) GO TO 297
       NNOSPL(1) = NNOSPL(1) + 1
      DO 293 KK=2,NNOSPL(1)
       IF ( NNOSPL(KK) .GT. I ) GO TO 294
293    CONTINUE
294    CALL LOOPMK( NNOSPL(1), 1, KK, KTOP, KOLD, KSTOP )
     DO 296 K=1,KSTOP
296       NNOSPL(KTOP-K) = NNOSPL(KOLD-K)
            NNOSPL(KK) = I
                     IS =  IS + 1
C  Check to see if the  node, which does not flow to a  node numbered
C  one higher, may still flow to another  node
C  If it does not flow to another  node, remove that tributary
297   WRITE ( *,2001 ) NNAME(I)
     READ  ( *,9001 ) ANS
     IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' )  GO TO 64
      DO 298 K1 = 2,NTRB(1,1)+1
        IF ( NTRB(1,K1) .EQ. I ) GO TO 299
298   CONTINUE
       GO TO 1001
299    DO 202 KK=K1,NTRB(1,1)
          KPLUS = KK + 1
      NTRB(1,KK) = NTRB(1,KPLUS)
202    NTRB(2,KK) = NTRB(2,KPLUS)
       NTRB(1,1) = NTRB(1,1) - 1
            IT = IT - 1
        GO TO 1001
C  The  node is a new tributary
64    WRITE ( *,2002 )
     WRITE ( *,2004 ) ( K, NNAME(K), K=1, NL )
     WRITE ( *,3001 ) NNAME(I)
     READ  ( *,9002 ) JDUMMY
     IF ( JDUMMY .LE. 0 ) GO TO 200
       NTRB(1,1) = NTRB(1,1) + 1
       DO 165 JT = 2,NTRB(1,1)
         IF ( NTRB(1,JT) .GT. I ) GO TO 166
165     CONTINUE
166     CALL LOOPMK ( NTRB(1,1), 1, JT, KTOP, KOLD,KSTOP )
       DO 167 K=1,KSTOP
          NTRB(1,KTOP-K) = NTRB(1,KOLD-K)
167        NTRB(2,KTOP-K) = NTRB(2,KOLD-K)
            NTRB(1,JT) = I
            NTRB(2,JT) = JDUMMY
                 IT = IT + 1
        GO TO 1001
C  The  node was already defined as a tributary flow but to the wrong
```

```
C   node
C  Check that there already is a tributary flow from the node I
65    IF ( I .EQ. NTRB(1,IT-1) ) GO TO 651
        WRITE ( *,2005 ) NNAME(I)
        GO TO 200
651   WRITE ( *,2002 )
      WRITE ( *,2004 ) ( K, NNAME(K), K=1, NL )
      WRITE ( *,3001 ) NNAME(I)
      READ  ( *,9002 ) JDUMMY
      IF ( JDUMMY .LE. 0 ) GO TO 200
            NTRB(1,IT-1) = I
            NTRB(2,IT-1) = JDUMMY
         GO TO 1001
C There is a mistake in saying NNAME(I) diverts
C          - it does divert but is not listed
C          - it doesnt divert at all
C          - it diverts to a different  node from the one listed
300   WRITE (*,3000) NNAME(I)
      READ  (*,9000) FNAME
      IF ( FNAME .EQ. 'none' .OR. FNAME .EQ. 'NONE' ) GO TO 360
      DO 355 K=1,NL
        IF ( NNAME(K) .EQ. FNAME ) GO TO 358
355   CONTINUE
      WRITE (*,3002) FNAME
      GO TO 300
358   WRITE (*,3003)
      READ  (*,9001) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' )  GO TO 359
C  Diverts in a new diversion
      NDIV(1,NDIV(1,1)+2) = I
      NDIV(2,NDIV(1,1)+2) = K
      WRITE ( *,4000 )
      READ  ( *,9003 ) CCAPD(NDIV(1,1)+2)
      NDIV(1,1)=NDIV(1,1) + 1
      NAME3 = NNAME(K)
      GO TO 1001
C  Only diverts to a different  node
359   NDIV(2,II) = K
          NAME3 = NNAME(K)
       GO TO 1001
C  The  node does not even divert water to any other  node.
360   WRITE ( *,3004 ) NNAME(I)
      READ  ( *,9001 ) ANS
          NAME3 = '    '
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 1001
         NDIV(1,1) = NDIV(1,1) - 1
         DO 365 K=II,NDIV(1,1)
            KPLUS = K + 1
          CCAPD(K) = CCAPD(KPLUS)
         NDIV(1,K) = NDIV(1,KPLUS)
365       NDIV(2,K) = NDIV(2,KPLUS)
          NAME3 = '---'
```

```
            GO TO 1001
C  There was an error in the diversion capacity.
400   WRITE (*,4000)
      READ  (*,9003) CCAPD(II)
            GO TO 1001
C There is a mistake in saying NNAME(I) receives offstream water
C           - it does receive but is not listed
C           - it doesnt receive at all
C           - it receives from a different  node than the one listed
500   WRITE (*,5000) NNAME(I)
      READ  (*,9000) FNAME
      IF ( FNAME .EQ. 'none' .OR. FNAME .EQ. 'NONE' ) GO TO 560
      DO 553 K=1,NL
        IF ( NNAME(K) .EQ. FNAME ) GO TO 555
553   CONTINUE
      WRITE (*,3002) FNAME
      GO TO 500
555   WRITE (*,3003)
      READ  (*,9001) ANS
      IF ( ANS .EQ. 'N' .OR. ANS .EQ. 'n' )  GO TO 558
C  A new diversion has been entered
      NDIV(2,NDIV(1,1)+2) = I
      NDIV(1,NDIV(1,1)+2) = K
      WRITE ( *,4000 )
      READ  ( *,9003 ) CCAPD(NDIV(1,1)+2)
      NDIV(1,1)=NDIV(1,1) + 1
      NAME4 = NNAME(K)
      GO TO 1001
C  Only the  node from which water is diverted has changed
558   NDIV(1,IM) = K
      NAME4 = NNAME(K)
       GO TO 1001
C  The  node does not even divert water to any other  node.
560   WRITE ( *,3004 ) NNAME(I)
      READ  ( *,9001 ) ANS
      NAME4 = '    '
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 1001
          NDIV(1,1) = NDIV(1,1) - 1
          DO 565 K=IM,NDIV(1,1)
              KPLUS = K + 1
           CCAPD(K) = CCAPD(KPLUS)
          NDIV(1,K) = NDIV(1,KPLUS)
565        NDIV(2,K) = NDIV(2,KPLUS)
      NAME4 = '—'
            GO TO 1001
C  There was an error in the irrigation  node or file
600   WRITE (*,6000) NNAME(I)
      READ  (*,9002) IX
      GO TO ( 625, 650, 675 ) IX
      GO TO 1001
C  The name on the file needs changing
625   WRITE (*,6001) NNAME(I)
```

```
      READ  (*,9000) WREQ(IG-1)
          GO TO 1001
C  There really is no irrigation at this  node, delete from NIRR and WREQ
C     and CCAPI
650     DO 665 K=(IG-1),NIRR(1)
             KPLUS = K + 1
          CCAPI(K) = CCAPI(KPLUS)
           WREQ(K) = WREQ(KPLUS)
665         NIRR(K) = NIRR(KPLUS)
           NIRR(1) = NIRR(1) - 1
                IG = IG - 1
           GO TO 1001
C  There really IS irrigation at this  node
675       NIRR(1) = NIRR(1) + 1
       CALL LOOPMK( NIRR(1), 1, (IG-1), KTOP, KOLD, KSTOP )
      DO 676 K=1,KSTOP
          WREQ(KTOP-K) = WREQ(KOLD-K)
         CCAPI(KTOP-K) = CCAPI(KOLD-K)
676        NIRR(KTOP-K) = NIRR(KOLD-K)
             NIRR(IG) = I
      WRITE ( *,6002 ) NNAME(I)
      READ  ( *,9000 ) WREQ(IG)
      WRITE ( *,4000 )
      READ  ( *,9003 ) CCAPI(IG)
      IG = IG + 1
        GO TO 1001
C  There was an error in the irrigation capacity
700   WRITE (*,4000)
      READ  (*,9003) CCAPI(IG-1)
        GO TO 1001
C  There was an error in the reservoir file
800   WRITE (*,8000) NNAME(I)
      READ  (*,9002) IX
      GO TO ( 825, 850, 875 ) IX
      GO TO 1001
C  The name on the file needs changing
825   WRITE (*,8001) NNAME(I)
      READ  (*,9000) EVAP(IR-1)
          GO TO 1001
C  There really is no reservoir at this  node, delete from NRES and EVAP
850    DO 865 K=(IR-1),NRES(1)
             KPLUS = K + 1
           EVAP(K) = EVAP(KPLUS)
865         NRES(K) = NRES(KPLUS)
       NRES(1) = NRES(1) - 1
              IR = IR - 1
             GO TO 1001
C  There really IS a reservoir at this  node
875   NRES(1) = NRES(1) + 1
       CALL LOOPMK( NRES(1), 1, (IR-1), KTOP, KOLD, KSTOP )
      DO 876 K=1,KSTOP
          EVAP(KTOP-K) = EVAP(KOLD-K)
```

```
876       NRES(KTOP-K) = NRES(KOLD-K)
            NRES(IR) = I
      WRITE ( *,8002 )
      READ  ( *,9000 ) EVAP(IR)
         IR = IR + 1
        GO TO 1001
C  There was an error in the power coefficient file
900   WRITE (*,9900) NNAME(I)
      READ  (*,9002) IX
      GO TO ( 925, 950, 975 ) IX
      GO TO 1001
C  The file name needs changing
925   WRITE (*,9901) NNAME(I)
      READ  (*,9000) POWER(IP-1)
          GO TO 1001
C  There really is no power plant at this  node, delete from NPOWER and
c  POWER
950    DO 965 K=(IP-1),NPOWER(1)
            KPLUS = K + 1
          POWER(K) = POWER(KPLUS)
965       NPOWER(K) = NPOWER(KPLUS)
         NPOWER(1) = NPOWER(1) - 1
              IP = IP - 1
          GO TO 1001
C  There really IS a power plant here
975     NPOWER(1) = NPOWER(1) + 1
       CALL LOOPMK( NPOWER(1), 1, (IP-1), KTOP, KOLD, KSTOP )
      DO 976 K=1,KSTOP
         POWER(KTOP-K) = POWER(KOLD-K)
976      NPOWER(KTOP-K) = NPOWER(KOLD-K)
           NPOWER(IP) = I
      WRITE ( *,9901 ) NNAME(I)
      READ  ( *,9000 ) POWER(IP)
         IP = IP + 1
        GO TO 1001
C  There was an error in the local inflow file
1010  WRITE (*,1111) NNAME(I)
      READ  (*,9002) IX
      GO TO ( 1125, 1150, 1175 ) IX
      GO TO 1001
C  The name on the local inflow file is wrong
1125  WRITE (*,1101) NNAME(I)
      READ  (*,9000) INFLOW(IL-1)
          GO TO 1001
C  There really is no local inflow at this  node, delete from NINFLO and
c  INFLOW
1150   IL = IL - 1
         DO 1165 K=IL,NINFLO(1)
            KPLUS = K + 1
         INFLOW(K) = INFLOW(KPLUS)
1165      NINFLO(K) = NINFLO(KPLUS)
         NINFLO(1) = NINFLO(1) - 1
```

```
        GO TO 1001
C  There really IS local inflow to this  node
1175  NINFLO(1) = NINFLO(1) + 1
        CALL LOOPMK( NINFLO(1), 1, (IL-1), KTOP, KOLD, KSTOP )
      DO 1176 K=1,KSTOP
        INFLOW(KTOP-K) = INFLOW(KOLD-K)
1176    NINFLO(KTOP-K) = NINFLO(KOLD-K)
          NINFLO(IL) = I
      WRITE ( *,1101 ) NNAME(I)
      READ  ( *,9000 ) INFLOW(IL)
         IL = IL + 1
        GO TO 1001
C
1000  FORMAT ( ' What is the correct name for the last  node?  Enter a n
     1ame of 12 or less letters ' )
1002  FORMAT ( ' Do you wish to delete the node ',A12,' from the configu
     1ration or rename it?   To remove the node completely, answer Y ' )
2000  FORMAT ( ' Enter the code for the correction required: '/ ' 1 - de
     2lete the downstream node'/ ' 2 - insert one or more nodes between
     3this node and the next previously entered'/' downstream node'/ ' 3
     4 - list the flows from this node as flowing to node',A15/' 4 - lis
     5t this node as the last on a stream.  You may also indicate any no
     6de to'/' which its flows may naturally spill'/' 5 - list the tribu
     7tary flows from this node as spilling to not the node listed, but
     8to another'/' 0 or nothing  will return to the next line of the RE
     9VIEW table. ')
2001  FORMAT ( ' Do the spills or discharges from ',A15,' naturally flow
               1 to another node?  Enter Y to specify the node which receives any
     2such flows ' )
2002  FORMAT ( ' Here are the nodes previously entered in the configurat
     1ion: ' )
2003  FORMAT ( ' There is a tributary flowing from ',A15,' to ',A15,' .
     1 Answer Y to keep the tributary flow and return to the last menu.
     2 Otherwise ',A15,' will be considered as flowing to ',A15 )
2004  FORMAT ( I3,2X,A15,I3,2X,A15,I3,2X,A15,I3,2X,A15 )
2005  FORMAT ( ' The flows from ', A15,' do not spill to another stream
     1in the current'/' configuration.  Use option 4 to enter new tribut
     2ary flows. '/ )
3000  FORMAT ( ' What is the name of the  node to which ',A15,' does div
     1ert water?  If it does not actually divert water, nor enter anothe
     2r stream, or the  node to which it diverts water has not yet been
     3entered,  Answer NONE. ' )
3001  FORMAT ( ' Enter the  INTEGER for  the  node to  which the spills,
     1 turbine  discharges, or return flows from ',A15,' naturally flow.
     2'/' If you enter 0, you will return to the last menu where you may
     3 enter 4 to delete this  node from the list of last  nodes on a tr
     4ibutary or 0 to continue without any changes. ' )
3002  FORMAT ( ' No  node of the name ',A15, ' was found.  Is it correct
     2ly spelt in full? ')
3003  FORMAT ( ' Was a diversion listed in the table?  Answer Y to redir
     3ect the'/' diversion: ' )
3004  FORMAT ( ' Answer Y if there is a diversion  at ',A15,' otherwise
```

```
     1the diversion at this  node will be deleted! ' )
4000  FORMAT ( ' Enter the correct value for the channel capacity '/' (
     1 99999 is for unlimited capacity ) ' )
5000  FORMAT ( ' What is the name of the  node  which  diverts water to
     1',A15, ' ?  If the  node which diverts to it has not yet be entere
     3d,  Answer NONE. ' )
6000  FORMAT ( ' Enter the code for the correction required: '/' 1 - cha
     1nge the name on the file of estimated crop water requirements '/ '
     2 2 - delete ',A15, ' from the list of irrigation  nodes '/ ' 3 - a
     3dd this  node to the list of irrigation  nodes.  0 or nothing will
     4return  to the next line of the REVIEW table. ' )
6001  FORMAT ( ' What is the name on the file of crop water requirements
     1 for the district served by ',A15, '?  Enter a unique name of 4 or
     2 less letters. ' )
6002  FORMAT ( ' What is the name on the file of the weekly water requir
     1ements of the crops ser- viced by ',A15, ' ? ' )
8000  FORMAT ( ' Enter the code for the correction required: '/' 1 - cha
     1nge the file name '/ ' 2 - delete ',A15,' from the list of  nodes
     2with reservoirs '/ ' 3 - add this  node to the list of  nodes  wit
     3h  reservoirs.'/' 0 or nothing will return to the next line of the
     4 REVIEW table. ' )
8001  FORMAT ( ' What is the name on the reservoir file for ',A15,'  Ent
     1er a unique name of 4 or less letters. ' )
8002  FORMAT ( ' What is the name on the file of weekly evaporation esti
     1mated and storage eleva -tion table for the reservoir?  Enter a na
     2me of 4 or less letters. ' )
9900  FORMAT ( ' Enter the code for the correction required: '/ ' 1 - ch
     1ange the file name '/ ' 2 - delete ',A15,' from the list of  nodes
     2with power plants '/ ' 3 - add this  node to the list of  nodes wi
     3th  power plants.  0 or nothing will return to the next line of th
     4e REVIEW table. ' )
9901  FORMAT ( ' What is the name on the power coefficients file for the
     1 power plant at ',A15,'  Enter a unique name of 4 or less letters.
     2 ' )
1111  FORMAT ( ' Enter the code for the correction required: '/ ' 1 - ch
     1ange the file name '/ ' 2 - delete ',A15,' from the list of  nodes
     2 with local inflow estimates '/ ' 3 - add this  node to the list o
     3f  nodes with local inflow '/     ' 0 or nothing will return to the
     4 next line of the REVIEW table. ' )
1101  FORMAT ( ' What is the name on the file of local inflow estimated
     1for ',A15,' Enter a unique name of 4 or less letters. ' )
1102  FORMAT ( ' Is there signigicant unregulated inflow into ',A15, ' s
     1uch that the weekly local inflow to the  node has been estimated o
     2n a file stored on disk?  Answer Y/N ' )
9000  FORMAT ( A12 )
9001  FORMAT ( A2 )
9002  FORMAT ( I5 )
9003  FORMAT ( F10.0 )
C
1001    CONTINUE
      RETURN
      END
```

```
C
      SUBROUTINE LOOPMK ( N, NP, K, KT, KO, KS )
c   N = number of nodes in the vector where some nodes are to be added
c  NP = number of nodes to be added
c   K = place in vector after which a node will be added
c  KT = new final number of nodes in the vector, a number to be
c        calculated
   KO = old final number of nodes in the vector, a number to be
c        calculated
c  KS = the number of times nodes in the vector need to be shifted to
c        make space for the added nodes, a number to be calculated
C
c  This function subroutine defines three variables used to open up
c  vectors so that new nodes will be entered in ascending order in the
c  vector
              KT = N + 2
              KO = KT - NP
              KS = KO - K
       RETURN
       END
C
      SUBROUTINE ADD (I,NL,IM,II,IS,IG,IR,IP,IL,IT )
c   I = node number of current line
c  NL = total number of nodes
c  IM = diversion TO number of current line
c  II = diversion FROM number of current line
c  IS = nospill number of current line
c  IG = irrigation number of current line
c  IR = reservoir number of current line
c  IP = power plant number of current line
c  IL = local inflow number of current line
c  IT = tributary number of current line
C
C  This subroutine inserts downstream nodes after the current node and
c  before previously downstream nodes
      COMMON/SHR/ NPOWER(15), NIRR(21), NRES(25), NNOSPL(20), NDIV(2,15),
      COMMON/SHP/ NINFLO(30), NLIM(31), NTRB(2,10), NRF(3,10)
      COMMON/NEW/ FLOMIN(30), FLOMAX(30)
      COMMON/CAN/ CCAPI(21), CCAPD(15), CMINI(21), CMIND(15), CHLSS(15)
      COMMON/NOM/ NNAME(40), WREQ(21), EVAP(25), INFLOW(30), POWER(15)
      CHARACTER * 12 NNAME, WREQ, EVAP, FNAME, INFLOW, POWER
      CHARACTER * 2 ANS
C
330    WRITE (*,3001) NNAME(I), NNAME(I+1)
      READ  (*,9002) NPLUS
              NL = NL + NPLUS
C  First all  nodes will be renumbered in all the vectors
      DO 331 K=2, (NIRR(1) + 1)
       IF ( NIRR(K) .GT. I ) NIRR(K) = NIRR(K) + NPLUS
331    CONTINUE
      DO 332 K=2,(NPOWER(1) + 1)
       IF ( NPOWER(K) .GT. I ) NPOWER(K) = NPOWER(K) + NPLUS
```

```
332   CONTINUE
     DO 333 K=2,(NRES(1) + 1)
      IF ( NRES(K) .GT. I ) NRES(K) = NRES(K) + NPLUS
333   CONTINUE
     DO 334 K=2,(NNOSPL(1) + 1)
      IF ( NNOSPL(K) .GT. I ) NNOSPL(K) = NNOSPL(K) + NPLUS
334   CONTINUE
     DO 335 K=2,(NDIV(1,1) + 1)
      IF ( NDIV(1,K) .GT. I ) NDIV(1,K) = NDIV(1,K) + NPLUS
      IF ( NDIV(2,K) .GT. I ) NDIV(2,K) = NDIV(2,K) + NPLUS
335   CONTINUE
     DO 336 K=2,(NINFLO(1) + 1)
      IF ( NINFLO(K) .GT. I ) NINFLO(K) = NINFLO(K) + NPLUS
336   CONTINUE
     DO 338 K=2,(NLIM(1) + 1)
      IF ( NLIM(K) .GT. I ) NLIM(K) = NLIM(K) + NPLUS
338   CONTINUE
     DO 432 K=2, (NTRB(1,1) + 1)
      IF ( NTRB(1,K) .GE. I ) NTRB(1,K) = NTRB(1,K) + JPLUS
      IF ( NTRB(2,K) .GE. I ) NTRB(2,K) = NTRB(2,K) + JPLUS
432   CONTINUE
       CALL LOOPMK( (NL-1), NPLUS, (I+1), KTOP, KOLD, KSTOP )
       DO 337 K=1,KSTOP
337         NNAME(KTOP-K) = NNAME(KOLD-K)
C  If the upstream  node was at the end of the stream, the NNOSPL vector
C  will contain the latest new downstream  node
                 ICOMPR = I + NPLUS
                   ISMN = IS - 1
     IF ( NNOSPL(ISMN) .EQ. ICOMPR ) NNOSPL(ISMN) = ICOMPR + NPLUS
C
C  Second the various components for each new  node will be entered
     DO 500 N=1,NPLUS
C The names of the  nodes
340            I = I + 1
       WRITE ( *,3010 ) NNAME(I-1)
       READ  ( *,9000 ) NNAME(I)
       WRITE ( *,3015 ) NNAME(I)
       READ  ( *,9001 ) ANS
         IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 3405
           GO TO 3406
C  Any limits to minimum or maximum downstream releases
3405    WRITE ( *,3016 ) NNAME(I)
       READ  ( *,9003 ) FLOMIN(NLIM(1))
       WRITE ( *,3017 ) NNAME(I)
       READ  ( *,9003 ) FLOMAX(NLIM(1))
       IF ( FLOMAX(NLIM(1)) .LE. 0 ) FLOMAX(NLIM(1)) = 99999
        NLIM(1) = NLIM(1) + 1
        NLIM(NLIM(1)) = I
C  Any diversions
3406    WRITE ( *,4000 ) NNAME(I)
       READ  ( *,9001 ) ANS
       IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 341
```

```
        GO TO 10
341    WRITE ( *,4001 )
       WRITE ( *,9004 ) ( K, NNAME(K), K=1, (NL-1) )
       WRITE ( *,4002 ) NNAME(I)
       READ  ( *,9002 ) NDUMMY
       IF ( NDUMMY .LE. 0 ) GO TO 10
          NDIV(1,1) = NDIV(1,1) + 1
          NDIV(1,(NDIV(1,1)+1)) = NDUMMY
          NDIV(2,(NDIV(1,1)+1)) = I
          WRITE ( *,3032 )
          READ  ( *,9003 ) CCAPD((NDIV(1,1)+1))
C The local inflow file if any
10     WRITE ( *,2001 ) NNAME(I)
       READ  ( *,9000 ) FNAME
       IF ( FNAME .EQ. 'NONE' .OR. FNAME .EQ. 'none' ) GO TO 20
       IF ( FNAME .EQ. '      '.OR. FNAME .EQ. '      ') GO TO 20
        NINFLO(1) = NINFLO(1) + 1
        DO 343 KK=2,NINFLO(1)
         IF ( NINFLO(KK) .GT. I ) GO TO 345
343      CONTINUE
345      CALL LOOPMK( NINFLO(1), 1, KK, KTOP, KOLD, KSTOP )
        DO 346 K=1,KSTOP
          INFLOW(KTOP-K) = INFLOW(KOLD-K)
346        NINFLO(KTOP-K) = NINFLO(KOLD-K)
            INFLOW(KK) = FNAME
            NINFLO(KK) = I
                IL = IL + 1
C The reservoir file if any
20       WRITE ( *,2002 ) NNAME(I)
         READ  ( *,9000 ) FNAME
         IF ( FNAME .EQ. 'NONE' .OR. FNAME .EQ. 'none' ) GO TO 30
         IF ( FNAME .EQ. '      '.OR. FNAME .EQ. '      ') GO TO 30
        NRES(1) = NRES(1) + 1
        DO 353 KK=2,NRES(1)
         IF ( NRES(KK) .GT. I ) GO TO 355
353      CONTINUE
355      CALL LOOPMK( NRES(1), 1, KK, KTOP, KOLD, KSTOP )
        DO 356 K=1,KSTOP
            EVAP(KTOP-K) = EVAP(KOLD-K)
356          NRES(KTOP-K) = NRES(KOLD-K)
              EVAP(KK) = FNAME
              NRES(KK) = I
                 IR = IR + 1
C  Any irrigation district
30       WRITE ( *,2003 ) NNAME(I)
        READ  ( *,9000 ) FNAME
        IF ( FNAME .EQ. 'NONE' .OR. FNAME .EQ. 'none' ) GO TO 40
        IF ( FNAME .EQ. '      '.OR. FNAME .EQ. '      ') GO TO 40
       NIRR(1) = NIRR(1) + 1
        DO 363 KK=2,NIRR(1)
         IF ( NIRR(KK) .GT. I ) GO TO 364
363     CONTINUE
```

```
364     CALL LOOPMK( NIRR(1), 1, KK, KTOP, KOLD, KSTOP )
       DO 366 K=1,KSTOP
        WREQ(KTOP-K) = WREQ(KOLD-K)
       CCAPI(KTOP-K) = CCAPI(KOLD-K)
366      NIRR(KTOP-K) = NIRR(KOLD-K)
           NIRR(KK) = I
           WREQ(KK) = FNAME
        WRITE ( *,3032 )
        READ  ( *,9003 ) CCAPI(KK)
              IG = IG + 1
C  The power plant if any
40      WRITE ( *,2004 ) NNAME(I)
       READ  ( *,9000 ) FNAME
       IF ( FNAME .EQ. 'NONE' .OR. FNAME .EQ. 'none' ) GO TO 500
       IF ( FNAME .EQ. '      '.OR. FNAME .EQ. '      ') GO TO 500
      NPOWER(1) = NPOWER(1) + 1
       DO 323 KK=2,NPOWER(1)
        IF ( NPOWER(KK) .GT. I ) GO TO 325
323     CONTINUE
325     CALL LOOPMK( NPOWER(1), 1, KK, KTOP, KOLD, KSTOP )
       DO 326 K=1,KSTOP
        POWER(KTOP-K) = POWER(KOLD-K)
326     NPOWER(KTOP-K) = NPOWER(KOLD-K)
          NPOWER(KK) = I
           POWER(KK) = FNAME
              IP = IP + 1
C
500   CONTINUE
C
3001  FORMAT ( ' How many  nodes do you wish to insert between ',A15,' a
     1nd ',A15,'?  All the  nodes entered now would be on the same stre
     2am as the first  node.  You can add new offstream  nodes later.  E
     3nter an INTEGER value: ' )
3015  FORMAT ( ' Are there restrictions on the regulated downstream rel
     1eases at the node '/A12,', either minimum or maximum levels? ' )
3016  FORMAT ( ' What is the minimum flow level in m3/s above which the
     1 regulated downstream'/' releases of ',A12,' must be? ' )
3017  FORMAT ( ' What is the maximum flow level in m3/s below which the
     1 regulated downstream'/' releases of ',A12,' must be? ' )
3032  FORMAT ( ' What is the  capacity of the  diversion  canal in m3/s?
     1  If there is no limit, enter 99999: ' )
3010  FORMAT ( ' What is the name of the next  node downstream of ',A15,
     1'  Enter a unique name of 12 or less letters. ' )
2001  FORMAT ( ' What is the name  on the inflow  file?   If ',A15,' has
     1 only  regulated flows or an insignificant drainage  basin, and th
     2erefore no inflow file has beenprepared, enter NONE.  Enter a uniq
     3ue name in 4 or less letters. ' )
2002  FORMAT ( ' What is the name on the file containing the estimates o
     1f weekly evaporation and the  elevation  storage  table  for the r
     2eservoir?  If there is no  significant storage or evaporation at t
     3he  node ',A15, ' enter NONE.  Enter a unique name of 4 or less le
     4tters. ' )
```

```
2003  FORMAT ( ' What is the name on the file of the weekly water requir
     1ements of the crops ser- viced by ',A15, ' ?  If there is no irrig
     2ation at this  node, enter NONE.Enter a unique name of 4 or less l
     3etters.' )
2004  FORMAT ( ' What is the name on the file containing the weekly powe
     1r conversion factors for ',A15,' ?  If there is no power plant, en
     2ter NONE.  Enter  a unique nameof 4 or less letters. ')
4000  FORMAT ( ' Does the  node ',A15, ' receive water from OTHER than i
     1ts natural         drainage basin - offstream sources, diversions o
     2r previously entered tributaries   Answer Y if it does. ')
4001  FORMAT ( ' The following lists the  nodes previously entered. ' )
4002  FORMAT ( ' Enter the INTEGER to the left of the  node from which w
     1ater is diverted, returnsor naturally flows to the  node ',A15, '
     2.  If the source  node has not  yet been entered, enter 0. '  )
9000  FORMAT ( A12 )
9001  FORMAT ( A2 )
9002  FORMAT ( I5 )
9003  FORMAT ( F10.0 )
9004  FORMAT ( I5,A15,I5,A15,I5,A15,I5,A15 )
C
1001    CONTINUE
      RETURN
      END
C
      SUBROUTINE DELETE (I, NL)
c  I = node number of the current line
c NL = total number of nodes
C
c  This subroutines deletes the node(s) downstream of the node listed in
c  the current line of the REVIEW table
      COMMON/SHR/ NPOWER(15), NIRR(21), NRES(25), NNOSPL(20), NDIV(2,15),
      COMMON/SHP/ NINFLO(30), NLIM(31), NTRB(2,10), NRF(3,10)
      COMMON/NEW/ FLOMIN(30), FLOMAX(30)
      COMMON/CAN/ CCAPI(21), CCAPD(15), CMINI(21), CMIND(15), CHLSS(15)
      COMMON/NOM/ NNAME(40), WREQ(21), EVAP(25), INFLOW(30), POWER(15)
      CHARACTER * 12 NNAME, WREQ, EVAP, FNAME, INFLOW, POWER
      CHARACTER * 2 ANS
C
            IP1 = I + 1
      DO 203 K=2,NLIM(1)
        IF ( NLIM(K) .EQ. IP1 ) GO TO 205
203   CONTINUE
        GO TO 207
205    DO 206 KK=K,NLIM(1)
         FLOMIN(KK) = FLOMIN(KK+1)
         FLOMAX(KK) = FLOMAX(KK+1)
206         NLIM(KK) = NLIM(KK+1)
            NLIM(1) = NLIM(1) - 1
207   DO 303 K=2,NPOWER(1)
        IF ( NPOWER(K) .EQ. IP1 ) GO TO 305
303   CONTINUE
        GO TO 307
```

```
305    DO 306 KK=K,NPOWER(1)
         POWER(KK) = POWER(KK+1)
306      NPOWER(KK) = NPOWER(KK+1)
         NPOWER(1) = NPOWER(1) - 1
307   DO 308 K=2,NIRR(1)
        IF ( NIRR(K) .EQ. (IP1) ) GO TO 310
308   CONTINUE
        GO TO 313
310     DO 312 KK=K,NIRR(1)
            WREQ(KK) = WREQ(KK+1)
           CCAPI(KK) = CCAPI(KK+1)
312          NIRR(KK) = NIRR(KK+1)
     NIRR(1) = NIRR(1) - 1
313   DO 314 K=2,NRES(1)
        IF ( NRES(K) .EQ. (IP1) ) GO TO 316
314   CONTINUE
        GO TO 319
316     DO 318 KK=K,NRES(1)
          EVAP(KK) = EVAP(KK+1)
318        NRES(KK) = NRES(KK+1)
        NRES(1) = NRES(1) - 1
319   DO 320 K=2,NNOSPL(1)
        IF ( NNOSPL(K) .EQ. IP1 ) GO TO 322
320   CONTINUE
        GO TO 325
322      DO 324 KK=K,NNOSPL(1)
324        NNOSPL(KK) = NNOSPL(KK+1)
         NNOSPL(1) = NNOSPL(1) - 1
325   DO 326 K=2,NINFLO(1)
        IF ( NINFLO(K) .EQ. IP1 ) GO TO 328
326   CONTINUE
        GO TO 331
328     DO 330 KK=K,NINFLO(1)
          INFLOW(KK) = INFLOW(KK+1)
330        NINFLO(KK) = NINFLO(KK+1)
        NINFLO(1) = NINFLO(1) - 1
331    DO 826 K=2,NTRB(1,1)
        IF ( NTRB(1,K) .EQ. IP1 ) GO TO 828
        IF ( NTRB(2,K) .EQ. IP1 ) GO TO 888
826   CONTINUE
        GO TO 335
C There is natural flows from the  node to be deleted to another stream.
C What is to happen to the flows of the tributary.
828   WRITE ( *,1030 ) NNAME(IP1), NNAME(NTRB(2,K))
      READ  ( *,9001 ) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 829
      WRITE ( *,1001 ) NNAME(NTRB(2,K))
      READ  ( *,9000 ) FNAME
      DO 106 KK=1,NN
       IF ( NNAME(KK) .EQ. FNAME ) GO TO 108
106   CONTINUE
      WRITE (*,1002) FNAME
```

```
      GO TO 828
108   NTRB(1,K) = KK
      GO TO 335
C The TRIBUTARY is to be entirely deleted
829   DO 105 KK = K,NTRB(1,1)
            KPLUS = KK + 1
        NTRB(1,KK) = NTRB(1,KPLUS)
105      NTRB(2,KK) = NTRB(2,KPLUS)
         NTRB(1,1) = NTRB(1,1) - 1
         GO TO 335
C  There is a TRIBUTARY flowing to the  node to be deleted
888   WRITE ( *,2030 ) NNAME(IP1), NNAME(NTRB(1,K))
      READ  ( *,9001 ) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 829
      WRITE ( *,2001 ) NNAME(NTRB(1,K))
      READ  ( *,9000 ) FNAME
      DO 806 KK=1,NN
        IF ( NNAME(KK) .EQ. FNAME ) GO TO 808
806   CONTINUE
      WRITE (*,1002) FNAME
      GO TO 888
808   NTRB(2,K) = KK
335   DO 338 K=2,NDIV(1,1)
        IF ( NDIV(1,K) .EQ. IP1 ) GO TO 340
        IF ( NDIV(2,K) .EQ. IP1 ) GO TO 350
338   CONTINUE
      GO TO 400
C There is a diversion from the  node to be deleted.  Is the
C diversion  node served from another  node or is the diversion
C also to be deleted.
340   WRITE ( *,1000 ) NNAME(IP1), NNAME(NDIV(2,K))
      READ  ( *,9001 ) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 345
      WRITE ( *,1031 ) NNAME(NDIV(2,K))
      READ  ( *,9000 ) FNAME
      DO 146 KK=1,NL
        IF ( NNAME(KK) .EQ. FNAME ) GO TO 118
146   CONTINUE
      WRITE (*,1002) FNAME
      GO TO 340
118   NDIV(1,K) = KK
      WRITE ( *,1003 ) NNAME(NDIV(1,K)), NNAME(NDIV(2,K))
      READ  ( *,9003 ) CCAPD(K)
      GO TO 400
C The diversion is to be entirely deleted
345   DO 165 KK = K,NDIV(1,1)
            KPLUS = KK + 1
         CCAPD(KK) = CCAPD(KPLUS)
        NDIV(1,KK) = NDIV(1,KPLUS)
165      NDIV(2,KK) = NDIV(2,KPLUS)
         NDIV(1,1) = NDIV(1,1) - 1
         GO TO 400
```

```
C  There is a diversion to the  node to be deleted
350   WRITE ( *,2000 ) NNAME(IP1), NNAME(NDIV(1,K))
      READ  ( *,9001 ) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 345
      WRITE ( *,2001 ) NNAME(NDIV(1,K))
      READ  ( *,9000 ) FNAME
      DO 246 KK=1,NL
        IF ( NNAME(KK) .EQ. FNAME ) GO TO 218
246   CONTINUE
      WRITE (*,1002) FNAME
      GO TO 350
218   NDIV(2,K) = KK
      WRITE ( *,1003 ) NNAME(NDIV(1,K)), NNAME(NDIV(2,K))
      READ  ( *,9003 ) CCAPD(K)
400   NL = NL - 1
      DO 405 KK = IP1, NL
405   NNAME(KK) = NNAME(KK+1)
C  Now all the vectors have the I+1  node removed.  Next all  nodes
C  greater than I+1 will be renumbered one less.
      DO 410 K=2, NIRR(1)
       IF ( NIRR(K) .GT. I ) NIRR(K) = NIRR(K) - 1
410   CONTINUE
      DO 415 K=2, NLIM(1)
       IF ( NLIM(K) .GT. I ) NLIM(K) = NLIM(K) - 1
415   CONTINUE
      DO 420 K=2,(NPOWER(1) + 1)
       IF ( NPOWER(K) .GT. I ) NPOWER(K) = NPOWER(K) - 1
420   CONTINUE
      DO 430 K=2,(NRES(1) + 1)
       IF ( NRES(K) .GT. I ) NRES(K) = NRES(K) - 1
430   CONTINUE
      DO 440 K=2,(NNOSPL(1) + 1)
       IF ( NNOSPL(K) .GT. I ) NNOSPL(K) = NNOSPL(K) - 1
440   CONTINUE
      DO 450 K=2,(NDIV(1,1) + 1)
       IF ( NDIV(1,K) .GT. I ) NDIV(1,K) = NDIV(1,K) - 1
       IF ( NDIV(2,K) .GT. I ) NDIV(2,K) = NDIV(2,K) - 1
450   CONTINUE
      DO 460 K=2,(NTRB(1,1) + 1)
       IF ( NTRB(1,K) .GT. I ) NTRB(1,K) = NTRB(1,K) - 1
       IF ( NTRB(2,K) .GT. I ) NTRB(2,K) = NTRB(2,K) - 1
460   CONTINUE
      DO 470 K=2,(NINFLO(1) + 1)
       IF ( NINFLO(K) .GT. I ) NINFLO(K) = NINFLO(K) - 1
470   CONTINUE
C
1000  FORMAT ( ' There is a diversion from the  node to be deleted ',A15
     1,' to the  node ',A15,'. Is the second  node served from another
     2node previously enteredor is the diversion to be deleted completel
     3y?  Answer Y to destroy the diversionto the second  node. ')
1001  FORMAT ( ' What is the name of the previously entered  node which
     1now diverts to ',A15, '  Enter a name of 12 or less letters. ' )
```

```
1002 FORMAT ( ' No  node of the name ',A15, ' was found.  Is it correct
     2ly spelt? ')
1003 FORMAT ( ' What is the capacity in m3/s of the diversion canal fro
     1m the  node ',A15,' to the  node ',A15 )
1030 FORMAT ( ' The  node to be deleted ',A15,' naturally flows to the
     1 node ',A15,'.  Does the second  node receive flows from another
     2node previously enteredor is the tributary to be deleted completel
     3y?  Answer Y to destroy the tributaryto the second  node. ')
1031 FORMAT ( ' What is the name of the previously entered  node which
                    1now flows directly to ',A15, '  Enter a name of 12 or less letters
     2. ' )
2030 FORMAT ( ' There is a tributary to the  node to be deleted ',A15,'
     2 from the  node ',A15,'.  Does water from the second  node flow to
     3 another previously     entered  node or is the tributary to be del
     4eted completely?  Answer Y to destroy the tributary from the secon
     5d  node. ' )
2031 FORMAT ( ' What is the name of the previously entered  node which
     1now receives water from ',A15, ' Enter a name of 12 or less letter
     2s. ' )
2000 FORMAT ( ' There is a diversion to the  node to be deleted ',A15,'
     2 from the  node ',A15,'.  Does the second  node divert water to an
     3other previously entered  node or is the diversion to be deleted c
     4ompletely?  Answer Y to destroy the diversion from the second node
     5. ' )
2001 FORMAT ( ' What is the name of the previously entered  node which
     1now receives water from ',A15, ' Enter a name of 12 or less letter
     2s. ' )
9000 FORMAT ( A12 )
9001 FORMAT ( A2 )
9003 FORMAT ( F10.0 )
    RETURN
    END
C
    SUBROUTINE CONFIG ( NADD, NI, NP, NR, NS, ND, NF, NT, N )
c NADD = initialization code, 1 = starting afresh, 2 = adding to an
c        existing configuration
c NI = number of irrigation nodes plus 2
c NP = number of power plant nodes plus 2
c NR = number of reservoir nodes plus 2
c NS = number of nodes not spilling to node numbered one higher plus 2
c ND = number of diversions plus 2
c NF = number of local inflow nodes plus 2
c NT = number of tributaries plus 2
c  N = total number of nodes
c
C  This subroutine defines the configuration of the system, the
C  reservoirs, irrigation districts, hydro-electric plants etc.  The
C  vectors which contain this information can be saved on a diskette
C  for next time.
    COMMON/SHR/ NPOWER(15), NIRR(21), NRES(25), NNOSPL(20), NDIV(2,15),
    COMMON/SHP/ NINFLO(30), NLIM(31), NTRB(2,10), NRF(3,10)
    COMMON/NEW/ FLOMIN(30), FLOMAX(30)
```

```
      COMMON/CAN/ CCAPI(21), CCAPD(15), CMINI(21), CMIND(15), CHLSS(15)
      COMMON/NOM/ NNAME(40), WREQ(21), EVAP(25), INFLOW(30), POWER(15)
      CHARACTER * 12 NNAME, WREQ, EVAP, FNAME, INFLOW, POWER
      CHARACTER * 2 ANS
C
      WRITE ( *,1000 )
      READ  ( *,9001 ) ANS
      IF ( ANS .EQ. 'N' .OR. ANS .EQ. 'n' ) GO TO 110
C To add (continue from 80) or start from scratch:
      IF ( NADD .GT. 1 ) GO TO 80
             NP = 2
             NI = 2
             NR = 2
             NS = 2
             ND = 2
             NF = 2
             NT = 2
              N = 1
       CCAPD(1) = 0
       CCAPI(1) = 0
        NLIM(1) = 1
        EVAP(1) = '      '
       POWER(1) = '      '
      INFLOW(1) = '      '
        WREQ(1) = '      '
C First of new configuration:
      WRITE ( *,1001 )
      READ  ( *,9000 ) NNAME(N)
C Common set of entry questions:
10     WRITE ( *,2001 ) NNAME(N)
      READ  ( *,9000 ) INFLOW(NF)
      IF ( INFLOW(NF) .EQ. 'NONE' .OR. INFLOW(NF) .EQ. 'none' ) GO TO 20
      IF ( INFLOW(NF) .EQ. '      '.OR. INFLOW(NF) .EQ. '      ') GO TO 20
              NINFLO(NF)=N
                     NF = NF + 1
20       WRITE ( *,3015 ) NNAME(N)
        READ  ( *,9001 ) ANS
          IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 3405
            GO TO 3406
C  Any limits to minimum or maximum downstream releases
3405    WRITE ( *,3016 ) NNAME(N)
        READ  ( *,9003 ) FLOMIN(NLIM(1))
        WRITE ( *,3017 ) NNAME(N)
        READ  ( *,9003 ) FLOMAX(NLIM(1))
        IF ( FLOMAX(NLIM(1)) .LE. 0 ) FLOMAX(NLIM(1)) = 99999
        NLIM(1) = NLIM(1) + 1
        NLIM(NLIM(1)) = N
3406  WRITE ( *,2002 ) NNAME(N)
      READ  ( *,9000 ) EVAP(NR)
      IF ( EVAP(NR) .EQ. 'NONE' .OR. EVAP(NR) .EQ. 'none' ) GO TO 30
      IF ( EVAP(NR) .EQ. '      '.OR. EVAP(NR) .EQ. '      ') GO TO 30
               NRES(NR) = N
```

```
                NR = NR + 1
30    WRITE ( *,2003 ) NNAME(N)
      READ  ( *,9000 ) WREQ(NI)
      IF ( WREQ(NI) .EQ. 'NONE' .OR. WREQ(NI) .EQ. 'none' ) GO TO 40
      IF ( WREQ(NI) .EQ. '      '.OR. WREQ(NI) .EQ. '      ') GO TO 40
             NIRR(NI) = N
      WRITE ( *,2033 )
      READ  ( *,9003 ) CCAPI(NI)
                NI = NI + 1
40    WRITE ( *,2004 ) NNAME(N)
      READ  ( *,9000 ) POWER(NP)
      IF ( POWER(NP) .EQ. 'NONE' .OR. POWER(NP) .EQ. 'none' ) GO TO 50
      IF ( POWER(NP) .EQ. '      '.OR. POWER(NP) .EQ. '      ') GO TO 50
            NPOWER(NP)=N
                NP = NP + 1
C Ask if there is another  node downstream on the same stream.  If
C there is find out if it has offstream sources and then go through the
C loop again.  If not find out if it flows into a previosly entered  node
C or is the end of the configuration.
50    WRITE ( *,1002 ) NNAME(N)
      READ  ( *,9001 ) ANS
      IF ( ANS .EQ. 'N' .OR. ANS .EQ. 'n' ) GO TO 55
      WRITE ( *,2000 ) NNAME(N)
      GO TO 75
55    WRITE ( *,1003 ) NNAME(N)
      READ  ( *,9001 ) ANS
            NNOSPL(NS) = N
                NS = NS + 1
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 65
60    WRITE ( *,1004 )
      READ  ( *,9001 ) ANS
      IF ( ANS .EQ. 'N' .OR. ANS .EQ. 'n' ) GO TO 100
      GO TO 70
65    WRITE ( *,4001 )
      WRITE ( *,9004 ) ( K, NNAME(K), K=1, (N-1) )
      WRITE ( *,3001 ) NNAME(N)
      READ  ( *,9002 ) NTRB(2,NT)
      IF ( NTRB(2,NT) .LE. 0 ) GO TO 60
            NTRB(1,NT) = N
                NT = NT + 1
              GO TO 60
C The next downstream station or the new  node on another stream:
70    WRITE ( *,1005 )
75          N = N + 1
      READ  ( *,9000 ) NNAME(N)
      WRITE ( *,4000 ) NNAME(N)
      READ  ( *,9001 ) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 77
        GO TO 10
77      WRITE ( *,4001 )
        WRITE ( *,9004 ) ( K, NNAME(K), K=1, (N-1) )
        WRITE ( *,4002 ) NNAME(N)
```

```
      READ  ( *,9002 ) NDIV(1,ND)
      IF ( NDIV(1,ND) .LE. 0 ) GO TO 10
      WRITE ( *,2033 )
      READ  ( *,9003 ) CCAPD(ND)
          NDIV(2,ND) = N
               ND = ND + 1
              GO TO 10
C  The  nodes are to be added on to an existing configuration
80       NP = NP + 1
      NI = NI + 1
      NR = NR + 1
      NS = NS + 1
      ND = ND + 1
      NF = NF + 1
      NT = NT + 1
            GO TO 70
C Ready to return to PROJOP for verification
100             NIRR(1) = NI - 2
           NPOWER(1) = NP - 2
             NRES(1) = NR - 2
           NNOSPL(1) = NS - 2
           NINFLO(1) = NF - 2
           NDIV(1,1) = ND - 2
           NTRB(1,1) = NT - 2
               NI = NI - 1
               NP = NP - 1
               NR = NR - 1
               NS = NS - 1
               NF = NF - 1
               NT = NT - 1
               ND = ND - 1
C
1000  FORMAT ( ' This subroutine enters the configuration of reservoirs,
    1 power plants, irrigationschemes, canals and diversions,  or it ad
    2ds onto an existing configuration pre -viously  stored on the  dis
    3kette.  If you  wish to only insert or delete  nodes between  node
    4s already entered on the same stream or canal or you wish to add o
    5rdelete components of previously  entered  nodes then you can use
    6the correcting procedures of the REVIEW.'/'  Do you want to use th
    7is subroutine?   Answer N to return to the REVIEW '/' procedures.
    8' )
1001  FORMAT ( ' What is the name of the most upstream  node on the main
    1 stream?  Enter a unique name of 12 or less letters. ' )
1002  FORMAT ( ' Is there a new  node downstream of ',A14,' which you ha
    1ve not yet entered?  Answer N if this is the last  node on this st
    2ream or canal. ' )
1003  FORMAT ( ' Do the spills, turbine discharges, or return flows of '
    1, A15,' flow to a previously entered  node?  Answer Y to enter whi
    2ch  node. ' )
1004  FORMAT ( ' Do you have more  nodes to enter?  Answer N to return t
    1o the REVIEW procedures. ' )
1005  FORMAT ( ' What is the name of the most upstream  node on another
```

```
     1stream?  Enter a unique  name of 12 or less letters. ' )
2000  FORMAT ( ' What is the name of the next  node downstream of ',A15,
     1'  Enter a unique name of 12 or less letters. ' )
2001  FORMAT ( ' What is the name  on the inflow  file?   If ',A15,' has
     1 only  regulated flows or an insignificant drainage  basin, and th
     2erefore no inflow file has beenprepared, enter NONE.  Enter a uniq
     3ue name in 4 or less letters. ' )
2002  FORMAT ( ' What is the name on the file containing the estimates o
     1f weekly evaporation and the  elevation  storage  table  for the r
     2eservoir?  If there is no  significant storage or evaporation at t
     3he  node ',A15, ' enter NONE.  Enter a unique name of 4 or less le
     4tters. ' )
2003  FORMAT ( ' What is the name on the file of the weekly water requir
     1ements of the crops ser- viced by ',A15, ' ?  If there is no irrig
     2ation at this  node, enter NONE.Enter a unique name of 4 or less l
     3etters.' )
2033  FORMAT ( ' What is the  capacity of the  diversion  canal in m3/s?
     1   If there is no limit, enter 99999: ' )
2004  FORMAT ( ' What is the name on the file containing the weekly powe
     1r conversion factors for ',A15,' ?  If there is no power plant, en
     2ter NONE.  Enter  a unique nameof 4 or less letters. ')
3001  FORMAT ( ' Enter the  INTEGER for  the  node to  which the spills,
     1 turbine  discharges, or return flows from ',A15,' flow? ' )
3015  FORMAT ( ' Are there restrictions on the regulated downstream rele
     1ases at the node '/ A12,', either minimum or maximum levels?  Answ
     2er Y to specify the'/' restrictions. ' )
3016  FORMAT ( ' What is the minimum flow level in m3/s above which the
     1 regulated downstream'/' releases of ',A12,' must be? ' )
3017  FORMAT ( ' What is the maximum flow level in m3/s below which the
     1 regulated downstream'/' releases of ',A12,' must be? ' )
4000  FORMAT ( ' Does the  node ',A15, ' receive water from OTHER than i
     1ts natural         drainage basin - offstream sources, diversions o
     2r previously entered tributaries?  Answer Y if it does. ')
4001  FORMAT ( ' The following lists the  nodes previously entered. ' )
4002  FORMAT ( ' Enter the INTEGER to the left of the  node from which w
     1ater is diverted, returnsor naturally flows to the  node ',A15, '
     2  If the source  node has not  yet been entered, enter 0. '  )
9000  FORMAT ( A12 )
9001  FORMAT ( A2 )
9002  FORMAT ( I5 )
9003  FORMAT ( F10.0 )
9004  FORMAT ( I5, A15, I5, A15, I5, A15, I5, A15 )
C
110   CONTINUE
      RETURN
      END
^Z
      SUBROUTINE DIRR ( CT4, KVAR )
c  CT4 = integer code (1 or 0 ) indicating entry of irrigation data
c        complete
c KVAR = integer code (1 or 0 ) indicating whether irrigation arrays
         have already been adjusted for current time steps
```

```
C
C This subroutine reads the irrigation data from one or two files.
C  The constants used to linearize the costs due
C  to failures in meeting the crop water requirements are read first.
C  The crop water requirements may be read from the same file or from
C  another.  All the values may be reviewed, updated and stored for
C  a later time.
      COMMON/SHR/ NPOWER(15), NIRR(21), NRES(25), NNOSPL(20), NDIV(2,15)
      COMMON/CAP/ TMAX(15), TMIN(15), SUP(25), SLO(25)
      COMMON/NOM/ NNAME(40), WREQ(21)
      COMMON/IRR/ THETA(6,21,9), OMEGA(6,21,9), PHI(6,21,9), ID(6,21),
     1            KINI(6,21), DTHET(6,21,9), IC(21), CWR(26,21)
      INTEGER CT4
      CHARACTER * 12 NNAME, WREQ, FIRST(6), FNAME, DUMMY, SECTL(6),
     1                          DUMMYL(6), FNAME2
      CHARACTER * 2 ANS
C
      WRITE ( *,6001 )
      READ  ( *,9000 ) FNAME2
C  If a FNAME2 is given, read in new file, otherwise review data
      IF ( FNAME2 .EQ. '            ' ) GO TO 510
      OPEN ( 9,FILE=FNAME2, STATUS='OLD')
      READ  ( 9,9100 ) JI, ( FIRST(M), M=1,6 )
      IF ( NIRR(1) .EQ. JI ) GO TO 402
      WRITE ( *,6002 ) NIRR(1), JI, (FIRST(M), M=1,6)
       CLOSE (9)
       GO TO 900
402     DO 405 N=1,JI
          READ  ( 9,9101 ) DUMMY, IC(N), (ID(I,N),KINI(I,N), I=1,IC(N))
          IF ( DUMMY .NE. WREQ(N+1) ) GO TO 499
          READ  ( 9,9102 ) ( (THETA(I,N,K), K=1,KINI(I,N) ),I=1,IC(N) )
405        READ  ( 9,9102 ) ( (PHI (I,N,K), K=1,KINI(I,N) ),I=1,IC(N) )
       GO TO 510
C  The names of the irrigation districts in the datafile don't match the
C  names given in the system configuration.
499    WRITE ( *,6003 ) WREQ(N+1), N, DUMMY
       CLOSE ( 9 )
         GO TO 900
C  All the loss values read in, check for up-date
510    WRITE ( *,1004 )
      READ  ( *,9001 ) ICHOIX
      IF ( ICHOIX .LE. 0 ) GO TO 60
      DO 100 NN=1,JI
          J = NN
      IF ( ICHOIX .GE. 99 ) GO TO 25
      DO 23 J=1,NIRR(1)
        IF ( NIRR(J+1) .EQ. ICHOIX ) GO TO 25
23     CONTINUE
C  No irrigation was found at the node requested, try again
      WRITE ( *,1007 ) NNAME(ICHOIX), ICHOIX
      GO TO 510
25     DO  90 I=1,IC(J)
```

```
30      WRITE ( *,1010 ) I, WREQ(J+1), ID(I,J)
        DO 33 K=2,KINI(I,J)
33          OMEGA(I,J,K) = (  PHI(I,J,K) -  PHI(I,J,K-1) ) /
     1                     ( THETA(I,J,K) - THETA(I,J,K-1) )
            OMEGA(I,J,1) =    PHI(I,J,1) / THETA(I,J,1)
        WRITE ( *,1011 ) ( K, THETA(I,J,K), K=1,KINI(I,J) )
        WRITE ( *,1012 ) ( K,   PHI(I,J,K), K=1,KINI(I,J) )
        WRITE ( *,1013 ) ( K, OMEGA(I,J,K), K=1,KINI(I,J) )
        WRITE ( *,1014 )
        READ  ( *,9001 ) ICH
        GO TO ( 35, 40, 45, 80, 70 ) ICH
C  Nothing incorrect continue to next curve of cost-deficit function
      IF ( ICHOIX .GE. 99 ) GO TO 90
        GO TO 90
C  Phi limit of costs in interval incorrect
35    WRITE ( *,1030 )
      READ  ( *,9001 ) KX
      WRITE ( *,1031 ) THETA(I,J,KX)
      READ  ( *,9002 )  PHI(I,J,KX)
         GO TO 30
C  Theta deficit limit incorrect
40    WRITE ( *,1050 )
      READ  ( *,9001 ) KX
      WRITE ( *,1051 ) KX
      READ  ( *,9002 ) THETA(I,J,KX)
         GO TO 30
C  Kini(J) too many or not enough
45    WRITE ( *,1070 )
      READ  ( *,9001 ) KNEW
      NTIMES = KNEW - KINI(I,J)
      WRITE ( *,1071 )
      READ  ( *,9001 ) KAFT
      IF ( KINI(I,J) .GT. KNEW ) GO TO 55
C Increase the number of intervals
        KINI(I,J) = KNEW
        KAFTP1  = KAFT + 1
        CALL LOOPMK ( KINI(I,J), NTIMES, KAFTP1, KTOP, KOLD, KSTOP )
        DO 48 K=1,KSTOP
          THETA (I,J,KTOP-K) = THETA (I,J,KOLD-K)
            PHI (I,J,KTOP-K) =   PHI (I,J,KOLD-K)
48        OMEGA (I,J,KTOP-K) = OMEGA (I,J,KOLD-K)
        DO 50 N=1,NTIMES
              K = KAFT + N
          WRITE ( *,1072 )  K
          READ  ( *,9002 ) THETA(I,J,K)
          WRITE ( *,1073 ) THETA(I,J,K)
          READ  ( *,9002 )  PHI(I,J,K)
50        OMEGA(I,J,K) = (  PHI(I,J,K) -  PHI(I,J,K-1) ) /
     1                   ( THETA(I,J,K) - THETA(I,J,K-1) )
        GO TO 30
C Decrease the number of intervals
55      KINI(I,J) = KNEW
```

```
        DO 58 K=(KAFT+1),KINI(I,J)
                 KP = K - NTIMES
          THETA (I,J,K) = THETA (I,J,KP)
           PHI (I,J,K) =  PHI (I,J,KP)
58         OMEGA(I,J,K) = (  PHI(I,J,K) -  PHI(I,J,K-1) ) /
     1                    ( THETA(I,J,K) - THETA(I,J,K-1) )
        GO TO 30
C  Change the last date for which the curve applies
70     WRITE ( *,1075 )
       READ  ( *,9001 ) ID(I,J)
       GO TO 30
C Add on an unbounded upper interval
80     WRITE ( *,1080 )
-      READ  ( *,9002 ) OMEGA (I,J,KINI(I,J))
        GO TO 30
90     CONTINUE
       IF ( ICHOIX .LT. 99 ) GO TO 510
100    CONTINUE
C Everything correct, calculate constants for datafile
60     DO 200 J=1,JI
       DO 200 I=1,IC(J)
        DO 75 K=2,KINI(I,J)
         OMEGA(I,J,K) = (  PHI(I,J,K) -  PHI(I,J,K-1) ) /
     1                   ( THETA(I,J,K) - THETA(I,J,K-1) )
75     DTHET(I,J,K) = THETA(I,J,K) - THETA(I,J,(K-1) )
       OMEGA(I,J,1) = PHI(I,J,1)/THETA(I,J,1)
       DTHET(I,J,1) = THETA(I,J,1)
       IF ( ICH .LT. 4 ) GO TO 200
        DTHET(I,J,KINI(I,J)) = 9999
200    CONTINUE
C  If the values have been previously read in, skip to review
       IF ( FNAME2 .EQ. '            ' ) GO TO 205
C  All the crop loss curves read in for every irrigation node, ask if
C  crop water requirements follow on the same file or are in another
21     WRITE ( *,2001 )
       READ  ( *,9000 ) DUMMY
       IF ( DUMMY .EQ. '             ' ) GO TO 500
C  CWR are on a separate file
       CLOSE (9)
       FNAME = DUMMY
       OPEN ( 9,FILE=FNAME, STATUS='OLD')
500    READ  ( 9,9100 ) JI, ( SECTL(M), M=1,6 )
       IF ( NIRR(1) .EQ. JI ) GO TO 502
       WRITE ( *,6002 ) NIRR(1), JI, (SECTL(M), M=1,6)
        CLOSE (9)
        GO TO 900
502    DO 505 N=1,JI
           READ  ( 9,9101 ) DUMMY
           IF ( DUMMY .NE. WREQ(N+1) ) GO TO 499
505         READ  ( 9,9102 ) ( CWR(I,N), I=1,26 )
       CLOSE (9)
       CT4 = 1
```

```
C Review crop water requirements
205   WRITE ( *,1001 )
      READ  (*,9001) ICHOIX
      IF ( ICHOIX .LE. 0 ) GO TO 300
      DO 299 JJ=1,JI
          J = JJ
      IF ( ICHOIX .GE. 99 ) GO TO 220
      DO 210 J=1,NIRR(1)
        IF ( NIRR(J+1) .EQ. ICHOIX ) GO TO 220
210     CONTINUE
C  No irrigation was found at the node requested, try again
      WRITE ( *,1007 ) NNAME(ICHOIX), ICHOIX
          GO TO 205
220   WRITE ( *,1002 ) WREQ(J+1)
      WRITE ( *,1022 ) ( I, CWR(I,J), I=1,26 )
      WRITE ( *,1003 )
      READ  ( *,9001 ) NX
       IF ( NX .GT. 0 ) GO TO 250
       IF ( ICHOIX .LT. 99 ) GO TO 205
       GO TO 299
250      DO 275 N=1,NX
          WRITE ( *,1033 )
          READ  ( *,9001 ) IX
          WRITE ( *,1034 ) IX
275        READ  ( *,9002 ) CWR(IX,J)
       GO TO 220
299    CONTINUE
300    WRITE ( *,1005 )
      READ  (*,9003) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 315
        GO TO 900
C Write updated values to the diskette file, asking first how many files
C  are needed.
315    WRITE ( *,2002 )
      READ  ( *,9001 ) NFILES
      IF ( NFILES .LE. 1 ) GO TO 312
C  Writing to two files
      WRITE ( *,2003 ) FNAME
      READ  ( *,9000 ) DUMMY
       IF ( DUMMY .EQ. '            ' ) GO TO 417
          FNAME = DUMMY
417    OPEN ( 9,FILE=FNAME, STATUS='OLD')
      WRITE ( *,2004 ) FNAME, JI, ( SECTL(M), M=1,6 )
      READ  ( *,9103 ) ( DUMMYL(M), M=1,6 )
       IF ( DUMMYL(1) .EQ. '            ' ) GO TO 419
           DO 418 M=1,6
418            SECTL(M) = DUMMYL(M)
419    WRITE ( 9,9100 ) JI, ( SECTL(M), M=1,6 )
      DO 420 N=1,JI
        WRITE (9,9101 ) WREQ(N+1)
420      WRITE (9,9102 ) ( CWR(I,N), I=1,26 )
      CLOSE (9)
```

```
C  Next write the economic losses file
      WRITE ( *,2005 ) FNAME2
      READ  ( *,9000 ) DUMMY
        IF ( DUMMY .EQ. '            ' ) GO TO 421
           FNAME2 = DUMMY
421   OPEN ( 9,FILE=FNAME2, STATUS='OLD')
      WRITE ( *,2004 ) FNAME2, JI, ( FIRST(M), M=1,6 )
      READ  ( *,9103 ) ( DUMMYL(M), M=1,6 )
       IF ( DUMMYL(1) .EQ. '            ' ) GO TO 423
           DO 422 M=1,6
422          FIRST(M) = DUMMYL(M)
423   WRITE ( 9,9100 ) JI, ( FIRST(M), M=1,6 )
      DO 430 N=1,JI
        WRITE (9,9101 ) WREQ(N+1), IC(N), (ID(I,N),KINI(I,N), I=1,IC(N))
        WRITE (9,9102 ) ( (THETA(I,N,K), K=1,KINI(I,N) ),I=1,IC(N) )
430      WRITE (9,9102 ) ( (PHI (I,N,K), K=1,KINI(I,N) ),I=1,IC(N) )
        WRITE (9,9104 )
      CLOSE (9)
         GO TO 900
312   WRITE ( *,1006 ) FNAME2
      READ  ( *,9000 ) DUMMY
       IF ( DUMMY .EQ. '            ' ) GO TO 317
         FNAME2 = DUMMY
317   OPEN ( 9,FILE=FNAME2, STATUS='OLD')
      WRITE ( *,2004 ) FNAME2, JI, ( FIRST(M), M=1,6 )
      READ  ( *,9103 ) ( DUMMYL(M), M=1,6 )
       IF ( DUMMYL(1) .EQ. '            ' ) GO TO 319
           DO 318 M=1,6
318          SECTL(M) = DUMMYL(M)
319   WRITE ( 9,9100 ) JI, ( FIRST(M), M=1,6 )
      DO 320 N=1,JI
        WRITE (9,9101 ) WREQ(N+1), IC(N), (ID(I,N),KINI(I,N), I=1,IC(N))
        WRITE (9,9102 ) ( (THETA(I,N,K), K=1,KINI(I,N) ),I=1,IC(N) )
320      WRITE (9,9102 ) ( (PHI (I,N,K), K=1,KINI(I,N) ),I=1,IC(N) )
      WRITE ( 9,9100 ) JI, ( SECTL(M), M=1,6 )
      DO 330 N=1,JI
        WRITE (9,9101 ) WREQ(N+1)
330      WRITE (9,9102 ) ( CWR(I,N), I=1,26 )
      CLOSE (9)
900   KVAR = 0
C
1001  FORMAT ( ' Do you wish to review or update the quarter-monthly dis
     1tict crop water require- ments?'/' You may enter the INTEGER numbe
     2r of one specific district you wish to change'/' or 99 to see them
     3 all. '/'  0 or nothing continues with the values as read from the
     4 datafiles. ')
1002  FORMAT ( ' Here are the quarter-monthly crop water requirements in
     1 MCM for the district ',A5 )
1022  FORMAT ( I3,F7.1,5X,I3,F7.1,5X,I3,F7.1,5X,I3,F7.1 )
1003  FORMAT ( ' Do you wish to change any quantity for the current run?
     1  Enter 0 to continue'/' with these values or the INTEGER number o
     2f incorrect quantities. ' )
```

```
1033  FORMAT ( ' What is the INTEGER to the left of an incorrect quantit
     1y? ')
1034  FORMAT ( ' For the ',I3,'th quarter-month what should be the crop
     1water requirement in MCM for the district? ' )
1004  FORMAT ( ' Do you wish to review or update the linearization inter
     1vals of the irrigation   loss function?'/' You may enter the INTE
     2GER number of one specific district you wish to change'/' or 99 to
     3 see them all.'/'  0 or nothing continues with the values as read
     4from the datafiles. ' )
1005  FORMAT ( ' Do you wish to store updated values on a diskette?  Ans
     1wer Y to do so ')
1006  FORMAT ( ' Do you want to save under the name ',A12,'?  If you do,
     1 enter nothing '/' before the RETURN'/
     1' Enter b:FNAME where b: specifies the drive and FNAME the name of
     2 an existing'/' file in 10 or less letters ' )
1007  FORMAT ( ' There was no irrigation found at the node ',A12,', the
     1number ',I3/' Please try again. ' )
1010  FORMAT ( ' Here are the current values for the bounds of irrigatio
     1n deficits ( MCM ), the  corresponding  costs of lower yeilds ( Rs
     2 ) and  the slope of the straight line joining these points for th
     3e ',I3,' curve of district ',A5/ ' The curve applies until the ',
     4I3,'th quarter-month after the start of the growing  season'/ )
1011  FORMAT ( ' Deficit: ',I3,F10.1,I4,F10.1,I4,F10.1,I4,F10.1,I4,F10.1
     1 )
1012  FORMAT ( '  Cost  : ',I3,F10.1,I4,F10.1,I4,F10.1,I4,F10.1,I4,F10.1
     1 )
1013  FORMAT ( ' Slope:',I3,F10.4,I3,F10.4,I3,F10.4,I3,F10.4,I3,F10.4 )
1014  FORMAT ( /' Do you wish to change anything for the current run?'/'
     1 Answer 5 to change the last quarter-month for which the curve app
     2lies'/
     3   8X, '4 to directly give the slope of the last UNBOUNDED interva
     4l'/ 8X,'3 to increase or decrease the number of intervals'/  8X,
     5'2 to modify the bounds of a deficit range'/    8X,  '1 to modify t
     6he corresponding cost of lower yield'/     8X, '0 or nothing to con
     7tinue with these values. ' )
1030  FORMAT ( ' What is the INTEGER to the left of the cost which you w
     1ould like to change? ' )
1031  FORMAT ( ' What is the cost of lower yields ( Rs ) due to an irrig
     1ation deficit of ', F6.1/ )
1050  FORMAT ( ' What is the INTEGER to the left of the irrigation defic
     1it which you would like  to correct? ')
1051  FORMAT ( ' What is the correct value for the upper bound of the ',
     1I3,'th interval? ' )
1070  FORMAT ( ' How many intervals do you now want?  The maximum number
     1 is 9. ')
1071  FORMAT ( ' Where  would you like to insert or delete?  What is the
     1  INTEGER to the left of the interval after which you would like t
     4o insert or delete. ' )
1072  FORMAT ( ' What is the new upper bound of the ',I3,'th interval? '
     1 )
1073  FORMAT ( ' What is the cost of the lower yeilds due to an irrigati
     1on deficit of ',F6.1,' MCM? ' )
```

```
1075 FORMAT ( ' When ( how many quarter-months after the start of the g
    1rowing season ) does thiscurve last apply?  Enter an INTEGER ' )
1080 FORMAT ( ' What is the slope of the function in the unbounded last
    1 interval? ' )
2001 FORMAT ( ' Do the updated crop water requirements follow on the sa
    1me file?  If another filehas been prepared, enter b:FNAME, where b
    2: specifies the drive and FNAME is the name of the file in 10 or l
    3ess letters'/ )
2002 FORMAT ( ' How many files will you use?  Answer 1 or 2 ' )
2003 FORMAT ( ' Do you wish to save the crop water data in the file, ',
    1A12,'?'/' To save the data in a new data file, enter b:FNAME where
    2 b: specifies the drive and FNAME is the name of an already existi
    3ng file in 10 or less characters. '/ )
2004 FORMAT ( ' The current first line for the file, ',A12,' is:'/
    1 1X,I3,6A12/' Enter a new title if you wish, without entering the
    2number of irrigation demand points.  If the first six spaces are b
    3lank, no change in the title will occur '/ )
2005 FORMAT ( ' Do you wish to save the economic losses data in the fil
    1e ',A12/ ' Enter b:FNAME where b: specifies the drive and FNAME is
    2 the name of another'/' already existing file. ' )
6001 FORMAT ( ' What is the name of the file containing the irrigation
    1deficit economic losses?'/' Enter b:FNAME, where FNAME is the name
    2 in 10 or less letters and b: designates'/' the drive. If you ente
    3r nothing you may review any data already entered'/ )
6002 FORMAT ( ' The current configuration lists ',I3,' districts. The d
    1atafile on the disk has ',I3,' districts.  The first line of the
    2 file is:'/ 1X,6A12 /' Please check you have the correct file, dis
    3kette and drive.' )
6003 FORMAT ( ' The current configuration lists ',A6,' as the ',I2,' th
    1 district. The datafile lists ',A6,' at this  node.  Please check
    2you have the correct spelling, order,'/' and/or file.' )
9000 FORMAT ( A12 )
9001 FORMAT ( I2 )
9002 FORMAT ( F6.0 )
9003 FORMAT ( A2 )
9100 FORMAT ( 1X,I3,6A12 )
9101 FORMAT ( 1X,A4,18I4 )
9102 FORMAT ( 1X,11F7.2 )
9103 FORMAT ( 6A12 )
9104 FORMAT ( 80(1H ))
9031 FORMAT ( A2,I2 )
     RETURN
     END
C
     SUBROUTINE DRES ( CT3 )
c  CT3 = integer code (1 or 0 ) indicating entry of reservoir data
c        complete
c
C This subroutine reads the reservoir data from the file, the constants
C of the evaporation and seepage linearization and the monthly loss
C coefficient.  The data can be reviewed and updated and stored for use
C at a later date.
```

```
      COMMON/SHR/ JPOWER(15), JIRR(21), JRES(25), JNOSPL(20), JDIV(2,15)
      COMMON/NOM/ JNAME(40), WREQ(21), EVAP(25)
      COMMON/RES/ GAMMA(25,9), DELTA(25,9), EPSI(25,9), KIN(25),
     1            DGAMA(25,9), CLOSS(14,25)
      COMMON/CAP/ TMAX(15), TMIN(15), SUP(25), SLO(25)
      INTEGER CT3
      CHARACTER * 12 JNAME, FIRST(6), FNAME, WREQ, EVAP, DUMMY
      CHARACTER * 3 MON(12)
      CHARACTER * 2 ANS
      DATA MON / 'OCT','NOV','DEC','JAN','FEB','MAR','APR','MAY','JUN',
     1           'JUL','AUG','SEP'/
C
      WRITE ( *,5001 )
      READ  ( *,9000 ) FNAME
      IF ( FNAME .EQ. '             ' ) GO TO 21
      OPEN ( 9,FILE=FNAME, STATUS='OLD')
      READ  ( 9,9100 ) JR, ( FIRST(M), M=1,6 )
      IF ( JRES(1) .EQ. JR ) GO TO 305
      WRITE ( *,5002 ) JRES(1), JR, (FIRST(M), M=1,6)
       CLOSE (9)
       GO TO 900
305     DO 306 N=1,JR
          READ  ( 9,9101 ) DUMMY, SLO(N), SUP(N), KIN(N)
          IF ( DUMMY .NE. EVAP(N+1) ) GO TO 399
          READ  ( 9,9102 ) ( CLOSS(I,N), I=1,12 )
          READ  ( 9,9102 ) ( GAMMA(N,K), K=1,KIN(N) )
306        READ  ( 9,9102 ) ( EPSI (N,K), K=1,KIN(N) )
       CLOSE ( 9 )
       CT3 = 1
      GO TO 21
C
C  The names of the reservoirs in the datafile don't match the names
C  given in the system configuration.
399   WRITE ( *,5003 ) EVAP(N+1), N, DUMMY
       CLOSE ( 9 )
         GO TO 900
C  All the values read in, check for up-date
21    WRITE ( *,1004 )
      READ  ( *,9001 ) ICHOIX
      IF ( ICHOIX .LE. 0 ) GO TO 60
22    DO 100 JJ=1,JR
            J = JJ
      IF ( ICHOIX .GE. 99 ) GO TO 30
      DO 25 J=1,JRES(1)
        IF ( JRES(J+1) .EQ. ICHOIX ) GO TO 30
25    CONTINUE
C  No irrigation was found at the node requested, try again
      WRITE ( *,1007 ) JNAME(ICHOIX), ICHOIX
      GO TO 21
30    WRITE ( *,1010 ) EVAP(J+1)
      DO 33 K=2,KIN(J)
33          DELTA(J,K) = ( EPSI(J,K) - EPSI(J,K-1) ) /
```

```
     1                 ( GAMMA(J,K) - GAMMA(J,K-1) )
          DELTA(J,1) =  EPSI(J,1) / GAMMA(J,1)
      WRITE ( *,1011 ) ( K, GAMMA(J,K), K=1,KIN(J) )
      WRITE ( *,1012 ) ( K,  EPSI(J,K), K=1,KIN(J) )
      WRITE ( *,1013 ) ( K, DELTA(J,K), K=1,KIN(J) )
      WRITE ( *,1014 )
      READ  ( *,9001 ) ICH
        GO TO ( 35, 40, 45, 80 ) ICH
C  Nothing incorrect continue to next reservoir
      IF ( ICHOIX .GE. 99 ) GO TO 100
        GO TO 21
C  Epsilon limit of surface area in interval incorrect
35     WRITE ( *,1030 )
      READ  ( *,9001 ) KX
      WRITE ( *,1031 ) GAMMA(J,KX)
      READ  ( *,9002 ) EPSI(J,KX)
          GO TO 30
C  Gamma storage volume limit incorrect
40     WRITE ( *,1050 )
      READ  ( *,9001 ) KX
      WRITE ( *,1051 ) KX
      READ  ( *,9002 ) GAMMA(J,KX)
          GO TO 30
C  Kin(J) too many or not enough
45     WRITE ( *,1070 )
      READ  ( *,9001 ) KNEW
      NTIMES = KNEW - KIN(J)
      WRITE ( *,1071 )
      READ  ( *,9001 ) KAFT
      IF ( KIN(J) .GT. KNEW ) GO TO 55
C Increase the number of intervals
        KIN(J) = KNEW
        KAFTP1  = KAFT + 1
        CALL LOOPMK ( KIN(J), NTIMES, KAFTP1, KTOP, KOLD, KSTOP )
        DO 48 K=1,KSTOP
          GAMMA (J,KTOP-K) = GAMMA (J,KOLD-K)
           EPSI (J,KTOP-K) = EPSI (J,KOLD-K)
48         DELTA (J,KTOP-K) = DELTA (J,KOLD-K)
        DO 50 N=1,NTIMES
              K = KAFT + N
          WRITE ( *,1072 )  K
          READ  ( *,9002 ) GAMMA(J,K)
          WRITE ( *,1073 ) GAMMA(J,K)
          READ  ( *,9002 ) EPSI(J,K)
50         DELTA(J,K) = ( EPSI(J,K) - EPSI(J,K-1) ) /
     1                 ( GAMMA(J,K) - GAMMA(J,K-1) )
        GO TO 30
C Decrease the number of intervals
55       KIN(J) = KNEW
        DO 58 K=(KAFT+1),KIN(J)
                 KP = K - NTIMES
          GAMMA (J,K) = GAMMA (J,KP)
```

```
          EPSI (J,K) = EPSI (J,KP)
58         DELTA(J,K) = ( EPSI(J,K) - EPSI(J,K-1) ) /
     1                   ( GAMMA(J,K) - GAMMA(J,K-1) )
        GO TO 30
C Check the monthly reservoir losses
80    WRITE ( *,1080 )( I, MON(I), CLOSS(I,J), I=1,12 )
      WRITE ( *,1081 )
      READ  ( *,9001 ) IX
      IF ( IX .LE. 0 ) GO TO 30
      WRITE ( *,1087 ) MON(IX)
      READ  ( *,9002 ) CLOSS(IX,J)
        GO TO 80
100   CONTINUE
C Everything correct, calculate constants for datafile
60    DO 200 J=1,JR
        DO 75 K=2,KIN(J)
         DELTA(J,K) = ( EPSI(J,K) - EPSI(J,K-1) ) /
     1                   ( GAMMA(J,K) - GAMMA(J,K-1) )
75    DGAMA(J,K) = GAMMA(J,K) - GAMMA( J,(K-1) )
      DELTA(J,1) = EPSI(J,1)/GAMMA(J,1)
      DGAMA(J,1) = GAMMA(J,1)
200   CONTINUE
      WRITE ( *,1005 )
      READ  (*,9003) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 215
        GO TO 900
C write updated values to diskette file
215   WRITE ( *,1006 ) FNAME
      READ  ( *,9000 ) DUMMY
        IF ( DUMMY .EQ. '            ' ) GO TO 216
          FNAME = DUMMY
216   OPEN ( 9,FILE=FNAME, STATUS='OLD')
      WRITE ( 9,9100 ) JR, ( FIRST(M), M=1,6 )
      DO 220 J=1,JR
        WRITE ( 9,9101 ) EVAP(J+1), SLO(J), SUP(J), KIN(J)
        WRITE ( 9,9102 ) ( CLOSS(I,J), I=1,12 )
        WRITE ( 9,9102 ) ( GAMMA(J,K), K=1,KIN(J) )
220     WRITE ( 9,9102 ) ( EPSI (J,K), K=1,KIN(J) )
      CLOSE (9)
900   CONTINUE
C
1004  FORMAT ( ' Do you wish to review or update the linearization inter
     1vals or the monthly loss coefficients?'/'  You may enter the INTEG
     2ER number of one specific reservoir you wish to change'/' or 99 to
     3 see them all.'/'  0 or nothing continues with the values as read
     4from the datafiles. ' )
1005  FORMAT ( ' Do you wish to store updated values on a diskette?   An
     1swer Y if you do. ')
1006  FORMAT ( ' Do you wish to save the file under the name ',A12,'?'/
     1' Enter nothing to keep the same name or'/
     1' Enter b:FNAME where b: specifies the drive and FNAME the name of
     2 the file in 10 or less letters ' )
```

```
1007  FORMAT ( ' There was no reservoir  found at the node ',A12,', the
     1number ',I3/' Please try again. ' )
1010  FORMAT ( ' Here are the current values for the bounds of reservoir
     1 storage volumes ( MCM ), the  corresponding  surface areas, ( km2
     2 ) and  the slope of the straight line joining these points for th
     3e reservoir ',A5/ )
1011  FORMAT ( ' Storage: ',I3,F6.1,I4,F6.1,I4,F6.1,I4,F6.1,I4,F6.1,I4,F
     16.0,I4,F6.1 )
1012  FORMAT ( '  Area  : ',I3,F6.1,I4,F6.1,I4,F6.1,I4,F6.1,I4,F6.1,I4,F
     16.0,I4,F6.1 )
1013  FORMAT ( ' Slope:',I3,F6.4,I3,F7.4,I3,F7.4,I3,F7.4,I3,F7.4,I3,F7.4
     1,I3,F7.4 )
1014  FORMAT ( /' Do you wish to change anything for the current run?'/
     1' Answer 4 to review the monthly reservoir loss coefficients'/
     2 8X,  '3 to increase or decrease the number of intervals'/ 8X,
     3'2 to modify the bounds of a storage volume range'/
     48X,'1 to modify the corresponding surface area'/
     58X,'0 or nothing to continue with these values. ' )
1030  FORMAT ( ' What is the  INTEGER  to the  left of  the  surface are
     1a which you would like to change? ' )
1031  FORMAT ( ' What is the surface area ( km2 ) corresponding to the s
     1torage volume of ', F6.1/ )
1050  FORMAT ( ' What is the INTEGER to the left of the storage bound yo
     1u would like to correct? ')
1051  FORMAT ( ' What is the correct value for the upper bound of the ',
     1I3,'th interval? ' )
1070  FORMAT ( ' How many intervals do you now want?  The maximum number
     1 is 9. ')
1071  FORMAT ( ' Where  would you like to insert or delete?  What is the
     1  INTEGER to the left of the interval after which you would like t
     4o insert or delete. ' )
1072  FORMAT ( ' What is the new upper bound of the ',I3,'th interval? '
     1 )
1073  FORMAT ( ' What is the surface area which corresponds to a storage
     1 volume of ',F6.1,' MCM? ' )
1080  FORMAT ( ' Here are the monthly reservoir losses in mm/mo.'/ I3,A4
     1,F6.0,I3,A4,F6.0,I3,A4,F6.0,I3,A4,F6.0,I3,A4,F6.0,I3,A4,F6.0/ I3,A
     24,F6.0,I3,A4,F6.0,I3,A4,F6.0,I3,A4,F6.0,I3,A4,F6.0,I3,A4,F6.0 )
1081  FORMAT ( ' Are these all correct?  Enter the INTEGER of an incorre
     1ct month.  0 or nothing  continues with these values. ' )
1087  FORMAT ( ' What is the reservoir loss coeficient in mm during the
     1month of ',A5,'? ' )
5001  FORMAT ( ' What is the name of the file containing the reservoir d
     1ata?'/'  Enter b:FNAME, where FNAME is the name in 10 or less lett
     2ers and b: designates the drive. '/' If you enter nothing you may
     3 review data already there'/ )
5002  FORMAT ( ' The current configuration lists ',I3,' reservoirs. The
     1datafile on the disk has ',I3,' reservoirs.  The first line of the
     2 file is:'/ 1X,6A12 /' Please check you have the correct file, dis
     3kette and drive.' )
5003  FORMAT ( ' The current configuration lists ',A6,' as the ',I2,' th
     1 reservoir.  The datafile    lists ',A6,' at this  node.  Please che
```

```
    2heck you have the correct spelling, order,    and/or file.' )
9000  FORMAT ( A12 )
9001  FORMAT ( I2 )
9002  FORMAT ( F6.0 )
9003  FORMAT ( A2 )
9100  FORMAT ( 1X,I3,6A12 )
9101  FORMAT ( 1X,A4,2F10.1,I3 )
9102  FORMAT ( 1X,12F6.1 )
9031  FORMAT ( A2,I2 )
      RETURN
      END
C
      SUBROUTINE DPOWER ( CT2 )
c  CT2 = integer code (1 or 0 ) indicating entry of power constants
c        complete
c
c  This subroutine enters the turbine minimum and maximum flows and the
c  power coefficients, corrector-multipliers and refernce storage values
c  Values may be updated and saved for next time
      COMMON/SHR/ JPOWER(15), JIRR(21), JRES(25), JNOSPL(20), JDIV(2,15)
      COMMON/NOM/ NNAME(40), WREQ(21), EVAP(25), INFLOW(30), POWER(15)
      COMMON/POW/ H(14,15), C(14,15), STO(14,15)
      COMMON/CAP/ TUPP(15), TLOW(15)
      INTEGER CT2
      CHARACTER * 12 NNAME, FIRST(6), FNAME, POWER, WREQ, EVAP,
INFLOW
      CHARACTER * 12 DUMMY
      CHARACTER * 2 ANS
C
C Read in power file
      WRITE ( *,4001 )
      READ  ( *,9000 ) FNAME
      IF ( FNAME .EQ. '            ' ) GO TO 203
      OPEN ( 9,FILE=FNAME, STATUS='OLD')
      READ  ( 9,9100 ) NP, ( FIRST(M), M=1,6 )
      IF ( JPOWER(1) .EQ. NP ) GO TO 204
      WRITE ( *,4002 ) JPOWER(1), NP, (FIRST(M), M=1,6)
       CLOSE (9)
       GO TO 900
204     DO 206 N=1,NP
          READ  ( 9,9101 ) DUMMY, TLOW(N), TUPP(N)
          IF ( DUMMY .NE. POWER(N+1) ) GO TO 299
          READ  ( 9,9102 ) STO(1,N)
          READ  ( 9,9103 ) C(1,N)
206        READ  ( 9,9102 ) H(1,N)
       CLOSE ( 9 )
          CT2=1
      GO TO 203
C  The names of the power plants in the datafile don't match the names
C  given in the system configuration.
299   WRITE ( *,4003 ) POWER(N+1), N, DUMMY
      CLOSE ( 9 )
```

```
        GO TO 900
C  All the values read in, check for up-date
203   WRITE ( *,1004 )
      READ  ( *,9001 ) ICHOIC
      IF ( ICHOIC .LE. 0 ) GO TO 150
      DO 100 JJ=1,NP
         J = JJ
      IF ( ICHOIC .GE. 99 ) GO TO 30
      DO 25 J=1,JPOWER(1)
        IF ( JPOWER(J+1) .EQ. ICHOIC ) GO TO 30
25    CONTINUE
C  No power plant was found at the node requested, try again
      WRITE ( *,1007 ) NNAME(ICHOIC), ICHOIC
         GO TO 203
30    WRITE ( *,1010 ) POWER(J+1), H(1,J), C(1,J), STO(1,J)
      WRITE ( *,1012 )
      READ  ( *,9001 ) ICHOIX
        GO TO ( 35, 45, 40 ) ICHOIX
C  Nothing incorrect continue to next power plant
        IF ( ICHOIC .GE. 99 ) GO TO 100
         GO TO 203
C  H(1,J) constant head coefficient incorrect
35    WRITE ( *,1030 )
      READ  ( *,9003 ) ANS
      IF ( ANS .EQ. 'N' .OR. ANS .EQ. 'n' ) GO TO 37
        CALL CALC ( J, H )
c        J = power plant number of current node
c H(14,15) = the array of power coefficients for every time step for
c            every power plant.
         GO TO 30
37    WRITE ( *,1032 )
      READ  ( *,9002 ) H(1,J)
         GO TO 30
C  STo reference storage volume incorrect
40    WRITE ( *,1050 )
      READ  ( *,9002 ) STO(1,J)
         GO TO 30
C  C(1,j) corrector multiplier incorrect
45    WRITE ( *,1070 )
      READ  ( *,9002 ) C(1,J)
         GO TO 30
100   CONTINUE
C  Ask if save to diskette desired
150   WRITE ( *,1005 )
      READ  (*,9003) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 215
        GO TO 900
C Write updated data to disk
215   WRITE ( *,1006 ) FNAME
      READ  ( *,9000 ) DUMMY
      IF ( DUMMY .EQ. '            ' ) GO TO 216
         FNAME = DUMMY
```

```
216   OPEN ( 9,FILE=FNAME, STATUS='OLD')
      WRITE ( 9,9100 ) NP, ( FIRST(M), M=1,6 )
      DO 220 J=1,NP
        WRITE ( 9,9101 ) POWER(J+1), TLOW(J), TUPP(J)
        WRITE ( 9,9104 ) STO(1,J)
        WRITE ( 9,9103 ) C(1,J)
220     WRITE ( 9,9103 ) H(1,J)
      CLOSE ( 9 )
900   CONTINUE
C
1004  FORMAT ( ' Do you wish to review or update the power production fa
     1ctors used to calculate  the energy produced?'/' You may enter the
     2 INTEGER number of one specific power plant you wish to change or'
     3,' 99 to see them all.'/'  0 or nothing continues with the values
     4as read from the datafiles. ' )
1005  FORMAT ( ' Do you wish to store updated values on a diskette?  Ans
     1wer Y to do so ')
1006  FORMAT ( ' Do you wish to save under the name ', A12,'?'/' Enter n
     1othing to keep the same name or b:FNAME where b: specifies the dri
     2ve and FNAME the name of an existing file in 10 or less letters'/)
1007  FORMAT ( ' There was no power plant found at the node ',A12,', the
     1 number ',I3/' Please try again. ' )
1010  FORMAT ( ' Here are the current values for the constant head coeff
     1icient'/' ( 9.78*head*eff./3600 ), H'/' the corrector multiplier,
     3C, to be applied to the difference of the average'/' storage volum
     4e and the reference storage'/' and the reference storage volume, S
     5To, in Mm3, for the power plant at ', A12,/10X,1HH,10X,1HC,9X,3HST
     6o/3X,F9.4,3X,F9.6,3X,F7.1/' Power plants with constant head have C
     7= -1.0' )
1012  FORMAT ( ' Do you wish to change any of these values?  You will be
     1 able to vary the values through the season later.'/' Answer 3 to
     2modify the reference storage volume, STo'/8X,'2 to modify the corr
     3ector multiplier, C'/ 8X,'1 to modify the constant head coefficien
     4t, H'/5X,'or 0 or nothing to continue with these values. ' )
1030  FORMAT ( ' Do you wish to calculate the value for the coefficient,
     1 H?  If you can enter it directly, enter N: ' )
1032  FORMAT ( ' What is the constant head coefficient, H? ')
1050  FORMAT ( ' What is the reference storage volume, STo, in Mm3? ' )
1070  FORMAT ( ' What is the corrector multiplier, C?  Use -1 for power
     1plants with constant head ' )
4001  FORMAT ( ' What is the name of the file containing the power data?
     1'/'  Enter b:FNAME, where FNAME is the name in 10 or less letters
     2 and b: designates the drive.'/' If you enter nothing you may corr
     3ect data already there'/ )
4002  FORMAT ( ' The current configuration lists ',I3,' power plants. Th
     1e datafile on the disk has ',I3,' power plants.  The first line of
     2 the file is:'/ 1X,6A12 /' Please check you have the correct file,
     3 diskette and drive.' )
4003  FORMAT ( ' The current configuration lists ',A6,' as the ',I2,' th
     1 power plant.  The datafile  lists ',A6,' at this  node.  Please c
     2heck you have the correct spelling, order, and/or file.' )
9000  FORMAT ( A12 )
```

```
9001  FORMAT ( I2 )
9002  FORMAT ( F10.0 )
9003  FORMAT ( A2 )
9100  FORMAT ( 1X,I3,6A12 )
9101  FORMAT ( 1X,A4,2F10.2,I3 )
9102  FORMAT ( 1X,10F8.0 )
9103  FORMAT ( 1X,8F9.6 )
9104  FORMAT ( 1X,10F8.1 )
      RETURN
      END
C
      SUBROUTINE CALC ( J, H )
c         J = power plant number of current node
c H(14,15) = the array of power coefficients for every time step for
c              every power plant.
c
c  This subroutine calculates the power coefficient from the effective
c head and the turbine efficiency.
c
      DIMENSION H ( 14,15 )
      WRITE ( *,1000 )
      READ  ( *,9000 ) EFF
      WRITE ( *,1001 )
      READ  ( *,9000 ) HEAD
      H(1,J) = 9.777*EFF*HEAD / 3600
C
1000  FORMAT ( ' What is the representative efficiency (in decimals) at
     1this power plant and for this head? ' )
1001  FORMAT ( ' What is the constant head in meters? ' )
9000  FORMAT ( F6.0 )
      RETURN
      END
C
      SUBROUTINE DCANAL ( ND, NI, CT5 )
c   ND = number of inter-basin diversions
c   NI = number of irrigation diversions
c  CT5 = integer code (1 or 0 ) indicating entry of canal constants
c         complete
c
C  This subroutine reads in a constant value for the diversion chanal
C  losses as a percent of flow for all diversion canals, CHLSS, and
C  minimum canal flows for diversion (CMIND)
C  and irrigation flows (CMINI)
      COMMON/SHR/ NPOWER(15), NIRR(21), NRES(25), NNOSPL(20), NDIV(2,15)
      COMMON/CAN/ CCAPI(21), CCAPD(15), CMINI(21), CMIND(15), CHLSS(15)
      COMMON/NOM/ NNAME(40)
      INTEGER CT5
      CHARACTER * 12 FNAME, DUMMY, NNAME, FIRST(6)
      CHARACTER * 2 ANS
C
      WRITE ( *,1000 )
      READ  ( *,9000 ) FNAME
```

```
C  When no name given, user wants to only correct existing file
     IF ( FNAME .EQ. '              ' ) GO TO 200
C  Read in the canal file
     OPEN ( 9,FILE=FNAME, STATUS='OLD')
     READ ( 9,9100 ) NND, NNI, ( FIRST(K), K=1,6 )
     IF ( NND .EQ. ND .AND. NNI .EQ. NI ) GO TO 5
     WRITE ( *,1003 ) NND, ND, NNI, NI, ( FIRST(K), K=1,6 )
     CLOSE ( 9 )
     GO TO 500
5    READ ( 9,9010 ) ( CHLSS(K), CMIND(K), K=1,ND )
     READ ( 9,9010 ) ( CMINI(K), K=1,NI )
     CLOSE ( 9 )
     CT5=1
     WRITE ( *,1004 )
     READ  ( *,9001 ) ANS
     IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 200
     GO TO 500
C  Display irrigation canal minimum flows
200  WRITE ( *,2000 )
     WRITE ( *,2001 ) ( K-1, NNAME(NIRR(K)), CMINI(K), K=2,NI )
     WRITE ( *,2002 )
     READ  ( *,9002 ) KX
     IF ( KX .LE. 0 ) GO TO 300
C  Corrections are necessary
     DO 220 KTIME=1,KX
       WRITE ( *,2003 ) KTIME
       READ  ( *,9002 ) K
       WRITE ( *,2004 ) NNAME(NIRR(K+1))
220    READ  ( *,9003 ) CMINI(K+1)
     GO TO 200
C  Display diversion canal minimum flows and channel losses
300  WRITE ( *,3000 )
     WRITE ( *,3001 ) ( K-1, NNAME(NDIV(1,K)), NNAME(NDIV(2,K)),
    1                  CMIND(K), CHLSS(K)*100, K=2,ND )
     WRITE ( *,3002 )
     READ  ( *,9002 ) KX
     IF ( KX .LE. 0 ) GO TO 400
C  Corrections are necessary
     DO 350 KTIME=1,KX
       WRITE ( *,3003 ) KTIME
       READ  ( *,9002 ) K
       WRITE ( *,3004 ) K
       READ  ( *,9002 ) ICODE
C  Corrections to channel losses or to minimum flows?
       GO TO ( 320, 310 ) ICODE
         GO TO 300
C  Correct channel losses
310      WRITE ( *,3005 ) NNAME(NDIV(1,K+1)), NNAME(NDIV(2,K+1))
         READ  ( *,9003 ) CHLSS(K+1)
         GO TO 350
C  Correct diversion minimum flows
320      WRITE ( *,2004 ) NNAME(NDIV(1,K+1))
```

```
        READ  ( *,9003 ) CMIND(K+1)
350   CONTINUE
      GO TO 300
C  Allow corrections to be saved
400   WRITE (*,1001)
      READ  (*,9001) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 415
        GO TO 500
415   WRITE ( *,1002 ) FNAME
      READ  ( *,9000 ) DUMMY
      IF ( DUMMY .EQ. '            ' ) GO TO 416
        FNAME = DUMMY
416   OPEN ( 10,FILE=FNAME, STATUS='OLD')
      WRITE ( 10,9100 ) ND, NI, ( FIRST(K), K=1,6 )
      WRITE ( 10,9011 ) ( CHLSS(K), CMIND(K), K=1,ND )
      WRITE ( 10,9012 ) ( CMINI(K), K=1,NI )
      CLOSE ( 10 )
C
500   CONTINUE
C
1000  FORMAT ( ' What is the name of the file containing diversion canal
     1',   ' losses as a percent of  flow, and any minimum flow limits f
     2or diversion and irrigation canals?'/' Enter b:FNAME, where b: is
     3the drive and FNAME is the name in 10 or less lettersIf you do not
     4 enter a name, you may correct the previously entered file or ente
     5r new data. ' )
1001  FORMAT ( ' Do you wish to save corrections on a diskette?  Answer
     1Y to do so. ' )
1002  FORMAT ( ' Do you wish to store the correct version under the name
     1 ',A12,'?'/' Enter nothing before the return to keep the same name
     2, otherwise, enter b:FNAME, where b: is the drive and FNAME is a n
     3ame of an existing file in 10 or less'/' letters. ' )
1003  FORMAT ( ' The file lists ',I3,' diversion canals and the current
     1configuration lists ',I3  /     ' the file lists ',I3,' irrigation
     2canals and the current configuration lists ',I3/' The first line o
     3f the file is:'/ 6A12 )
1004  FORMAT ( ' Do you wish to see or correct the values as read?  Answ
     1er Y to do so ' )
2000  FORMAT ( ' These are the IRRIGATION canal minimum flow limits:'/
     13X,3HNo.,5X,15HIRRIGATION NODE,8X,20HMINIMUM FLOW IN m3/s, / )
2001  FORMAT ( I6,6X,A12,18X,F8.2 )
2002  FORMAT ( ' How many values for the minimum flow do you wish to cha
     1nge?  Enter an INTEGER   value or nothing to continue ' )
2003  FORMAT ( ' What is the number of the line you wish to change the '
     1,I3,'st time? ' )
2004  FORMAT ( ' What is the minimum flow limit for the canal from the n
     1ode ',A12,'   Enter the value in m3/s ' )
3000  FORMAT ( ' These are the DIVERSION canal minimum flow limits and t
     1he channel losses: '/ 3X,3HNo.,8X,9HFROM NODE,5X,7HTO NODE,6X,12HM
     2INIMUM FLOW,4X,12HCHANNEL LOSS/ 41X,9H1. (m3/s),8X,10H2. PERCENT
     3 / )
3001  FORMAT ( I6,6X,A12,3X,A12,4X,F8.2,9X,F8.4 )
```

```
3002  FORMAT ( ' How many values do you wish to change?  Enter an INTEGE
     1R or nothing to continue ' )
3003  FORMAT ( ' What is the line number you wish to change the ',I3,'st
     1 time?  Enter an INTEGER ' )
3004  FORMAT ( ' Do you wish to change a value in the first column or in
     2 the second for line'/' number ',I3,'?  Enter 1 or 2 ' )
3005  FORMAT ( ' What is the percentage of flow that is lost in the dive
     1rsion canal from '/ 1X,A15,' to ', A15,' Enter a decimal number ')
9000  FORMAT ( A12 )
9001  FORMAT ( A2 )
9002  FORMAT ( I2 )
9003  FORMAT ( F8.0 )
9010  FORMAT ( 10F8.0 )
9011  FORMAT ( F8.4,F8.2,F8.4,F8.2,F8.4,F8.2,F8.4,F8.2,F8.4,F8.2 )
9012  FORMAT ( 10F8.2 )
9100  FORMAT ( 2I4,6A12 )
      RETURN
      END
^Z
      SUBROUTINE PERIOD ( JJ, I4, II, IDATE, CT6, KVAR )
C  This subroutine determines how many time steps to use after reading the
C  starting date. The crop water requirement array is adjusted for
C  the correct time steps as are any other preparations relative to
C  the starting date that are necessary before writing B:RELEASE.DATA.
C      The parts which let the user vary the data by time-step and
c      read variations to the diskette are last and optional
      COMMON/SHR/ NPOWER(15), NIRR(21), NRES(25), NNOSPL(20), NDIV(2,15)
      COMMON/SHP/ NINFLO(30), NLIM(31)
      COMMON/NEW/ FLOMIN(30), FLOMAX(30), FMX(14,30), FMN(14,30)
      COMMON/CAN/ CCAPI(21), CCAPD(15), CMINI(21), CMIND(15), CHLSS(15)
      COMMON/CAP/ TMAX(15), TMIN(15), SUP(25), SLO(25)
      COMMON/VAR/ VLS(14,25),VMS(14,25),CIX(14,20),CIN(14,20),
     1         TMX(14,15), TMN(14,15), CDX(14,15), CDN(14,15), VL(14,15)
      COMMON/POW/ H(14,15), C(14,15), STO(14,15)
      COMMON/RES/ GAMMA(25,9), DELTA(25,9), EPSI(25,9), KIN(25),
     1            DGAMA(25,9), CLOSS(14,25)
      COMMON/IRR/ THETA(6,21,9), OMEGA(6,21,9), PHI(6,21,9), ID(6,21),
     1         KINI(6,21), DTHET(6,21,9), IC(21), CWR(26,21), IID(14,21)
      COMMON/NOM/ NNAME(40), WREQ(21)
      DIMENSION CLOS(5)
      INTEGER CT6
      CHARACTER * 13 CHAR(8)
      CHARACTER * 12 NNAME, WREQ, FNAME, FNAME1
      CHARACTER * 2 ANS
      CHARACTER * 3 MON(12)
      DATA MON / 'OCT','NOV','DEC','JAN','FEB','MAR','APR','MAY','JUN',
     1           'JUL','AUG','SEP'/
      DATA CHAR/  ' power plant ', ' restriction ', '  diversion   ',
     1            '  reservoir   ', ' power plant ', '  irrigation ',
     2            '  diversion   ', '  reservoir   '/
C
C  Display the months to ask for current month
```

```
2     WRITE ( *,1000 ) ( I, MON(I), I=1,12 )
      READ  ( *,9001 ) MONTH
      IF ( MONTH .GT. 12 .OR. MONTH .LT. 1 ) GO TO 2
      WRITE ( *,1001 )
      READ  ( *,9001 ) MON4
      IF ( MON4 .GT. 4 .OR. MON4 .LT. 1 ) GO TO 2
C Calculate the number of periods II and the number of quarter-month
C long periods I4
          II = 15 - MONTH - MON4
          I4 =  9 - MON4
       IDATE = 4*MONTH + MON4 - 4
      IF ( IDATE .LT. 18 ) GO TO 10
      IF ( IDATE .GE. 27 ) GO TO 5
          II = 27 - IDATE
          I4 = II
        GO TO 10
5          II = 20 - MONTH - MON4
      IF ( MONTH .LT. 12 ) GO TO 10
          II = 5 - MON4
          I4 = II
C  Verify II and I4
10    WRITE ( *,1002 ) II, I4, ( II-I4 )
      READ  ( *,9002 ) ANS
      IF ( ANS .EQ. 'N' .OR. ANS .EQ. 'n' ) GO TO 2
          CT6 = 1
C  Prepare the variation of the constants for the remaining periods in
C the growing season
       DO 12 I=1,II
         DO 121 J=1,NLIM(1)
        FMN(I,J) = FLOMIN(J) * 0.6588
        FMX(I,J) = FLOMAX(J) * 0.6588
        IF ( FLOMAX(J) .GE. 99999 ) FMX(I,J) = 99999
          IF ( I .LE. I4 ) GO TO 121
        FMN(I,J) = 4 * FMN(I,J)
        FMX(I,J) = 4 * FMX(I,J)
121       CONTINUE
         DO 122 J=1,NDIV(1,1)
122         VL(I,J) = 1.0
         DO 12  J=1,NRES(1)
           VLS(I,J) = 1.0
12          VMS(I,J) = 1.0
        DO 14 J=1,NPOWER(1)
           DO 14 I=2,II
            H(I,J) = H(1,J)
            IF ( C(1,J) .LT. 0 ) GO TO 14
                 C(I,J) = C(1,J)
               STO(I,J) = STO(1,J)
14       CONTINUE
      DO 20 J=1,NRES(1)
         CLOSR1 = CLOSS(MONTH,J)
         CLOSR2 = CLOSS(MONTH+1,J)
         CLOS(1) = CLOSS(MONTH+2,J)
```

```
         CLOS(2) = CLOSS(MONTH+3,J)
         CLOS(3) = CLOSS(MONTH+4,J)
         CLOS(4) = CLOSS(MONTH+5,J)
         CLOS(5) = CLOSS(MONTH+6,J)
      DO 22 I = 1,( 5-MON4 )
22       CLOSS(I,J) = CLOSR1/4
      DO 23 I = ( 6-MON4 ), I4
23       CLOSS(I,J) = CLOSR2/4
      DO 24 I = 1,(II - I4)
24       CLOSS(I+I4,J) = CLOS(I)
      IF ( IDATE .LE. 18 ) CLOSS(II,J)=CLOSS(II,J)/2
20    CONTINUE
C  Do the turbine max and min flows
      DO 30 J=1,NPOWER(1)
        DO 31 I=1,I4
          TMX(I,J) = TMAX(J) * 0.6588
31         TMN(I,J) = TMIN(J) * 0.6588
        DO 32 I=I4+1,II
          TMX(I,J) = TMAX(J) * 2.6352
32         TMN(I,J) = TMIN(J) * 2.6352
      IF ( IDATE .GT. 18 ) GO TO 30
        TMN(II,J)=TMN(II,J)/2
        TMX(II,J)=TMX(II,J)/2
30    CONTINUE
C  Check that CWR is as read in and not previously adjusted
      IF ( KVAR .LT. 1 ) GO TO 415
       WRITE ( *, 1222 )
       READ  ( *, 9001 ) KVAR
      IF ( KVAR .LT. 1 ) GO TO 70
C  Do the irrigation canal capacities
415    DO 40 J=1,NIRR(1)
       DO 41 I=1,I4
         CIX(I,J) = CCAPI(J+1) * .6588
41        CIN(I,J) = CMINI(J+1) * .6588
        DO 42 I=I4+1,II
          CIX(I,J) = CCAPI(J+1) * 2.6352
42         CIN(I,J) = CMINI(J+1) * 2.6352
      IF ( IDATE .GT. 18 ) GO TO 43
          CIN(II,J)=CIN(II,J)/2
          CIX(II,J)=CIX(II,J)/2
C  Determine what irrigation cost curves to use when and start CWR at
C  starting date  - ie, CWR(1,j) = CWR(IDATE,j) and so on
C                         IID(1,j) = the curve number to use for the 1st
C                                    period in this run
43     IS = IDATE
      IF ( IS .GT. 26 ) IS = IS - 26
      IF ( KVAR .GT. 1 ) GO TO 45
435    DO 44 I=1,I4
44       CWR(I,J) = CWR(IS+I-1,J)
      IF ( II .LE. I4 ) GO TO 45
      IF ( IDATE .GT. 26 ) GO TO 443
C   MAHA season with monthly seasons at the end
```

```
       DO 442 I=1,II-I4-1
442        CWR(I+I4,J) = CWR(IS+I4+(4*I)-1,J) + CWR(IS+I4+(4*I)-2,J) +
     1                 CWR(IS+I4+(4*I)-3,J) + CWR(IS+I4+(4*I)-4,J)
       CWR(II,J) = CWR(25,J) + CWR(26,J)
      GO TO 45
C   YALA season with monthly seasons at the end
443   DO 444 I=1,II-I4
444        CWR(I+I4,J) = CWR(IS+I4+(4*I)-1,J) + CWR(IS+I4+(4*I)-2,J) +
     1                 CWR(IS+I4+(4*I)-3,J) + CWR(IS+I4+(4*I)-4,J)
C  Determine what penalty cost to give to irrigation deficits in period I
45    DO 46 N=1,IC(J)
         IF ( ID(N,J) .GE. IS ) GO TO 47
46    CONTINUE
47    DO 48 I=1,I4
         IF ( ID(N,J) .LT. IS ) N = N + 1
            IID(I,J) = N
48                 IS = IS + 1
           IS = IS + 3
      DO 49 I=I4+1,II-1
        IF ( ID(N,J) .LT. IS ) N = N + 1
            IID(I,J) = N
49          IS = IS + 4
            IID(I,J) = IC(J)
40    CONTINUE
        KVAR = 1
C  Do the DIVERSION canal capacities
      DO 50 J=1,NDIV(1,1)
        DO 51 I=1,I4
         CDX(I,J) = CCAPD(J+1) * .6588
51        CDN(I,J) = CMIND(J+1) * .6588
        DO 52 I=I4+1,II
          CDX(I,J) = CCAPD(J+1) * 2.6352
52         CDN(I,J) = CMIND(J+1) * 2.6352
      IF ( IDATE .GT. 18 ) GO TO 50
          CDX(II,J)=CDX(II,J)/2
          CDN(II,J)=CDN(II,J)/2
50    CONTINUE
C  Find which constants, if any, vary with time
15    WRITE ( *,1004 )
      READ  ( *,9001 ) IND
      IF ( IND .LE. 0 ) GO TO 60
      IF ( IND .EQ. 1 ) GO TO 80
      WRITE ( *,1003 ) ( J, NNAME(J), J=1,JJ )
21    WRITE ( *,1104 ) CHAR(IND-1)
      READ  ( *,9001 ) INODE
      IF ( INODE .LE. 0 ) GO TO 15
      GO TO ( 80, 200, 250, 300, 400, 500, 600, 700, 100 ) IND
C  No more changes required
      GO TO 60
C  Calculate the difference in reservoir storage volumes in one period
C                            vs another
100   DO 199 J1=1,NRES(1)
```

```
          J = J1
      IF ( INODE .GE. 99 ) GO TO 130
      DO 125 J=1,NRES(1)
        IF ( NRES(J+1) .EQ. INODE ) GO TO 130
125     CONTINUE
C  No RESERVOIR was found at the node requested, try again
      WRITE ( *,1007 ) CHAR(IND-1), NNAME(INODE), INODE
       GO TO 21
130    WRITE ( *,1020 ) NNAME(NRES(J+1))
      WRITE ( *, 1008 ) ( I, SUP(J) * VMS(I,J), I=1,II )
      WRITE ( *, 1008 ) ( I, SLO(J) * VLS(I,J), I=1,II )
      WRITE ( *,1024 )
      READ  ( *,9001 ) ITIMES
      IF ( ITIMES .LE. 0 ) GO TO 135
      DO 123 I=1,ITIMES
         RED1 = 0.0
         RED2 = 0.0
         WRITE ( *,1006 ) I
         READ  ( *,9001 ) IIR
         WRITE ( *,1021 ) IIR
         READ  ( *,9003 ) RED1
         WRITE ( *,1022 )
         READ  ( *,9003 ) RED2
         VMS((IIR),J) = 1 + RED1
123       VLS((IIR),J) = 1 + RED2
      GO TO 130
135    IF ( INODE .LT. 99 ) GO TO 21
199    CONTINUE
      GO TO 15
C  Calculate the difference in power generated in one period vs another
200    DO 299 J1=1,NPOWER(1)
          J = J1
      IF ( INODE .GE. 99 ) GO TO 230
      DO 205 J=1,NPOWER(1)
        IF ( NPOWER(J+1) .EQ. INODE ) GO TO 230
205     CONTINUE
C  No POWER PLANT was found at the node requested, try again
      WRITE ( *,1007 ) CHAR(IND-1), NNAME(INODE), INODE
       GO TO 21
C  H(i,j) = 9.777 kN/m3 * eff. * constant head / 3600 s/hr
C  It gives energy production in GW-hr. when QTB(i,j) is in Mm3
C  When the head is not constant, the energy calculated is corrected by
C  multiplying a factor C(i,j) by the difference between the average
C  storage level during the time step and a reference level, STO(i,j)
C  These energy production constants may vary during the season.
230    WRITE ( *,1030 ) NNAME(NPOWER(J+1))
      WRITE ( *,1332 ) ( I, H(I,J), C(I,J), STO(I,J), I =1,II )
      WRITE ( *,1036 )
      READ  ( *,9001 ) ITIMES
        IF ( ITIMES .LE. 0 ) GO TO 290
          DO 233 I=1,ITIMES
            WRITE ( *,1031 ) I4
```

```
          READ  ( *,9001 ) IX
          WRITE ( *,1032 ) IX
          READ  ( *,9003 ) DUMMY
           IF ( DUMMY .GT. 0 )  H(IX,J) = DUMMY
             IF ( C(IX,J) .LE. 0 ) GO TO 233
                WRITE ( *,1033 ) IX
                READ  ( *,9003 ) DUMMY
                IF ( DUMMY .GT. 0 )  C(IX,J) = DUMMY
                WRITE ( *,1034 ) IX
                READ  ( *,9003 ) DUMMY
                IF ( DUMMY .GT. 0 )  STO(IX,J) = DUMMY
233       CONTINUE
         GO TO 230
290      IF ( INODE .LT. 99 ) GO TO 21
299   CONTINUE
       GO TO 15
C  Vary the flow release  restrictions
250   DO 259 J1=1,NLIM(1)-1
            J = J1
      IF ( INODE .GE. 99 ) GO TO 260
      DO 255 J=1,NLIM(1)-1
        IF ( NLIM(J+1) .EQ. INODE ) GO TO 260
255     CONTINUE
C  No FLOW RESTRICTIONS were found at the node requested, try again
      WRITE ( *,1007 ) CHAR(IND-1), NNAME(INODE), INODE
       GO TO 21
260   WRITE ( *,1250 ) NNAME(NLIM(J+1)), ( I, FMN(I,J), I, FMX(I,J),
     1                                         I=1,II )
      WRITE ( *,1253 )
      READ  ( *,9001 ) ICHOIX
      GO TO ( 270, 280 ) ICHOIX
      IF ( INODE .LT. 99 ) GO TO 21
      GO TO 259
C  There are differences in the minimum allowed flows
270   WRITE ( *,1015 )
      READ  ( *,9001 ) IX
      WRITE ( *,1251 ) IX
      READ  ( *,9003 ) FMN(IX,J)
        GO TO  260
C  There are differences in the maximum allowed flows
280   WRITE ( *,1015 )
      READ  ( *,9001 ) IX
      WRITE ( *,1252 ) IX
      READ  ( *,9003 ) FMX(IX,J)
        GO TO  260
259   CONTINUE
        GO TO 15
C Calculate the difference in diversion canal transportation losses from
C  one period to another
300   DO 399 J1=1,NDIV(1,1)
            J = J1
      IF ( INODE .GE. 99 ) GO TO 330
```

```
      DO 305 J=1,NDIV(1,1)
        IF ( NDIV(1,J+1) .EQ. INODE ) GO TO 330
305     CONTINUE
C  No DIVERSION was found at the node requested, try again
      WRITE ( *,1007 ) CHAR(IND-1), NNAME(INODE), INODE
       GO TO 21
330    J2 = NDIV(2,J+1)
      WRITE ( *,1040 ) NNAME(NDIV(1,J+1)), NNAME(J2)
      WRITE ( *,1008 ) ( I, CHLSS(J+1) * VL(I,J) * 100, I=1,II )
      WRITE ( *,1036 )
      READ  ( *,9001 ) ITIMES
      IF ( ITIMES .LE. 0 ) GO TO 335
      DO 343 I=1,ITIMES
         WRITE ( *,1041 ) I
         READ  ( *,9001 ) IIR
         WRITE ( *,1042 ) IIR
         READ  ( *,9003 ) RED
343      VL((IIR),J) = 1 + RED
      GO TO 330
335    IF ( INODE .LT. 99 )  GO TO 21
399    CONTINUE
       GO TO 15
C  Read the same variations as last time with the first period removed.
80     WRITE ( *,1051 )
      READ  ( *,9005 ) FNAME
           OPEN  ( 9,FILE=FNAME,STATUS='OLD' )
           READ (9,8000) MTHOLD, MN4OLD, IIOLD, I4OLD
           WRITE (*,1052) FNAME, MN4OLD, MON(MTHOLD), IIOLD, I4OLD
           READ  (*,9001) ICHOIX
              GO TO ( 80, 89 ) ICHOIX
           READ (9,8001) ( (H(I,J),C(I,J),STO(I,J),TMN(I,J),TMX(I,J),
     1                       I=1,IIOLD), J=1,NPOWER(1) )
           READ (9,8002) ( (VLS(I,J), VMS(I,J), CLOSS(I,J),
     1                       I=1,IIOLD), J=1,NRES(1) )
           READ (9,8002) ( (CIX(I,J),CIN(I,J), I=1,IIOLD), J=1,NIRR(1) )
           READ (9,8002) ((FMX(I,J),FMN(I,J), I=1,IIOLD), J=1,NLIM(1)-1)
           READ (9,8002) ( (CDX(I,J),CDN(I,J),VL(I,J), I=1,IIOLD),
     1                        J=1,NDIV(1,1) )
      CLOSE (9)
C  Advance the values by one time step
C  This cannot be done the first quarter-month of a new month
      WRITE (*,1053)
      READ  (*,9002) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' .OR. MON4 .LE. 1 ) GO TO 15
      DO 85 I=1,I4
       DO 82 J=1,NPOWER(1)
        TMN(I,J) = TMN(I+1,J)
82       TMX(I,J) = TMX(I+1,J)
       DO 83 J=1,NRES(1)
        VLS(I,J) = VLS(I+1,J)
83       VMS(I,J) = VMS(I+1,J)
       DO 84 J=1,NIRR(1)
```

```
       CIX(I,J) = CIX(I+1,J)
84      CIN(I,J) = CIN(I+1,J)
      DO 85 J=1,NDIV(1,1)
       CDX(I,J) = CDX(I+1,J)
85      CDN(I,J) = CDN(I+1,J)
89    CLOSE (9)
      WRITE ( *,1050 )
       GO TO 15
400   DO 499 J1=1,NRES(1)
           J = J1
      IF ( INODE .GE. 99 ) GO TO 430
      DO 405 J=1,NRES(1)
       IF ( NRES(J+1) .EQ. INODE ) GO TO 430
405    CONTINUE
C  No RESERVOIR was found at the node requested, try again
      WRITE ( *,1007 ) CHAR(IND-1), NNAME(INODE), INODE
       GO TO 21
430   WRITE ( *,1082 ) NNAME(NRES(J+1)), ( I, CLOSS(I,J), I=1,II )
      WRITE ( *,1083 )
      READ  ( *,9002 ) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 440
      IF ( INODE .GE. 99 ) GO TO 499
      GO TO 21
C  There are differences in the evaporation and infiltration loss
C  coefficient
440   WRITE ( *,1015 )
      READ  ( *,9001 ) IX
      WRITE ( *,1086 ) IX
      READ  ( *,9003 ) CLOSS(IX,J)
       GO TO  430
499   CONTINUE
       GO TO 15
C  Vary the turbine release restrictions
500   DO 599 J1=1,NPOWER(1)
           J = J1
      IF ( INODE .GE. 99 ) GO TO 530
      DO 505 J=1,NPOWER(1)
       IF ( NPOWER(J+1) .EQ. INODE ) GO TO 530
505    CONTINUE
C  No POWER PLANT was found at the node requested, try again
      WRITE ( *,1007 ) CHAR(IND-1), NNAME(INODE), INODE
       GO TO 21
530   WRITE ( *,1100 ) NNAME(NPOWER(J+1)), ( I, TMN(I,J), I, TMX(I,J),
     1                                       I=1,II )
      WRITE ( *,1103 )
      READ  ( *,9001 ) ICHOIX
      GO TO ( 550, 560 ) ICHOIX
      IF ( INODE .LT. 99 ) GO TO 21
      GO TO 599
C  There are differences in the minimum turbine flows
550   WRITE ( *,1015 )
      READ  ( *,9001 ) IX
```

```
390     CONTINUE
C  According to Yala or Maha season, prepare vector of which anticipated
C  irrigation flow will return to system when, where and what percentage
      DO 395 K=1,I4
           KP = IDATE + K - 1
395     C1(K,J) = C1(KP,J)
      IF ( IDATE .GT. 26 ) GO TO 450
        DO 398 K=I4+1,II-1
           KP = IDATE + K - 1
398     C1(K,J) = ( C1(KP,J) + C1(KP+1,J) + C1(KP+2,J) + C1(KP+3,J) )/4
       C1(II,J) = ( C1(25,J) + C1(26,J) ) / 2
        GO TO 500
450     IDATE1 = IDATE - 26
      DO 495 K=1,I4
           KP = IDATE1 + K - 1
495     C1(K,J) = C1(KP,J)
      DO 498 K=I4+1,II
           KP = IDATE1 + K - 1
498     C1(K,J) = ( C1(KP,J) + C1(KP+1,J) + C1(KP+2,J) + C1(KP+3,J) )/4
500   CONTINUE
550   CLOSE (9)
C
1000  FORMAT ( ' What is the name of the file which contains the informa
     1tion on the irrigation   return flows?'/   ' Enter b:FNAME, where
     2b: is the name of the drive and FNAME the name in 10 or'/' less le
     3tters.  If you enter nothing you may amend values already there. '
     4 )
1001  FORMAT ( ' Do you wish to save the updated version?  Answer Y to d
     1o so ' )
1002  FORMAT ( ' Do you wish to save under the same name,',A12/' Enter b
     1:FNAME, where b: is the drive and FNAME the name in 10 or less let
     2ters or nothing to use the same name. ' )
1003  FORMAT ( ' The file has ',I3,' occurences of return flow and the f
     1irst line of the file is '/ 6A12/ ' Do you wish to continue readin
     2g this file?  Answer N to return to the last menu ' )
2000  FORMAT ( ' This table shows the source and destination of irrigati
     1on return flows:'             / 4H No.,15X,10HIRRIGATION,15X,6HDRAI
     2NS,12X,8HTIME LAG/22X,4HFROM,20X,2HTO,8X,18H( quarter-months ) )
2001  FORMAT ( I5,15X,A12,5X,A12,15X,I3 )
2002  FORMAT ( ' How many values would you like to change?  Enter an INT
     1EGER.  Correct values may be saved later ' )
2006  FORMAT ( ' Enter the code for the error:'/ 10X,' 1 - correction of
     1 existing value'/ 10X,' 2 - deletion of an entire line'/ 10X,' 3 -
     2 addition of a  node which has return flows to another  node'/
     3' Enter an INTEGER. ' )
2007  FORMAT ( ' How many do you wish to add?  Enter an INTEGER value ')
2008  FORMAT ( ' What is the INTEGER to the left of the additional  node
     1, number ',I3,', whose      irrigation turnouts will return to ano
     2ther  node in the system? ' )
2009  FORMAT ( ' What is the INTEGER to the left of the  node which rece
     1ives the return flow from this  node ' )
2099  FORMAT ( ' How many quarter-months later does the irrigation flow
```

```
      WRITE ( *,1101 ) IX
      READ  ( *,9003 ) TMN(IX,J)
        GO TO  530
C  There are differences in the maximum turbine flows
560   WRITE ( *,1015 )
      READ  ( *,9001 ) IX
      WRITE ( *,1102 ) IX
      READ  ( *,9003 ) TMX(IX,J)
        GO TO  530
599   CONTINUE
        GO TO 21
600   DO 699 J1=1,NIRR(1)
            J = J1
      IF ( INODE .GE. 99 ) GO TO 625
      DO 605 J=1,NIRR(1)
        IF ( NIRR(J+1) .EQ. INODE ) GO TO 625
605     CONTINUE
C  No IRRIGATION was found at the node requested, try again
      WRITE ( *,1007 ) CHAR(IND-1), NNAME(INODE), INODE
       GO TO 21
625   WRITE ( *,1200 ) WREQ(J+1), ( I, CIN(I,J), I, CIX(I,J),
     1                                  I=1,II )
      WRITE ( *,1203 )
      READ  ( *,9001 ) ICHOIX
      GO TO ( 650, 660 ) ICHOIX
      IF ( INODE .LT. 99 ) GO TO 21
      GO TO 699
C  There are differences in the minimum canal flows
650   WRITE ( *,1015 )
      READ  ( *,9001 ) IX
      WRITE ( *,1201 ) IX
      READ  ( *,9003 ) CIN(IX,J)
        GO TO  625
C  There are differences in the maximum CANAL flows
660   WRITE ( *,1015 )
      READ  ( *,9001 ) IX
      WRITE ( *,1202 ) IX
      READ  ( *,9003 ) CIX(IX,J)
        GO TO  625
699   CONTINUE
        GO TO 21
700   DO 799 J1=1,NDIV(1,1)
            J = J1
      IF ( INODE .GE. 99 ) GO TO 725
      DO 705 J=1,NDIV(1,1)
        IF ( NDIV(1,J+1) .EQ. INODE ) GO TO 725
705     CONTINUE
C  No DIVERSION was found at the node requested, try again
      WRITE ( *,1007 ) CHAR(IND-1), NNAME(INODE), INODE
       GO TO 21
725   WRITE ( *,1300 ) NNAME(NDIV(1,J+1)), NNAME(NDIV(2,J+1))
      WRITE ( *,1301 ) ( I, CDN(I,J), I, CDX(I,J), I=1,II )
```

```
      WRITE ( *,1302 )
      READ  ( *,9001 ) ICHOIX
      GO TO ( 750, 760 ) ICHOIX
      IF ( INODE .LT. 99 ) GO TO 21
        GO TO 799
C  There are differences in the minimum canal flows
750   WRITE ( *,1015 )
      READ  ( *,9001 ) IX
      WRITE ( *,1201 ) IX
      READ  ( *,9003 ) CDN(IX,J)
        GO TO  725
C  There are differences in the maximum canal flows
760   WRITE ( *,1015 )
      READ  ( *,9001 ) IX
      WRITE ( *,1202 ) IX
      READ  ( *,9003 ) CDX(IX,J)
        GO TO  725
799   CONTINUE
        GO TO 21
C  Give the opportunity to save these variations for the run next week
60    WRITE ( *,1005 )
      READ  ( *,9002 ) ANS
      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 420
        GO TO 70
420   WRITE ( *,1400 ) FNAME
      READ ( *,9005 ) FNAME1
      IF ( FNAME1 .EQ. '              ' ) GO TO 475
        FNAME=FNAME1
475   OPEN (9,FILE=FNAME,STATUS='OLD')
          WRITE (9,8000) MONTH, MON4, II, I4
          WRITE (9,8001) ( (H(I,J),C(I,J),STO(I,J),TMN(I,J),TMX(I,J),
     1                    I=1,II), J=1,NPOWER(1) )
          WRITE (9,8002) ( (VLS(I,J), VMS(I,J), CLOSS(I,J),
     1                    I=1,II), J=1,NRES(1) )
          WRITE (9,8002) ( (CIX(I,J), CIN(I,J), I=1,II), J=1,NIRR(1) )
          WRITE (9,8002) ( (FMX(I,J), FMN(I,J), I=1,II), J=1,NLIM(1)-1)
          WRITE (9,8002) ( (CDX(I,J), CDN(I,J), VL(I,J), I=1,II),
     1                     J=1,NDIV(1,1) )
      CLOSE (9)
70    CONTINUE
C
1000  FORMAT ( I5,A5,I5,A5,I5,A5,I5,A5,I5,A5,I5,A5/I5,A5,I5,A5,I5,A5,I5,
     1A5,I5,A5,I5,A5/' What is the INTEGER to the left of the current mo
     2nth, i.e. the month from which the optimal operation modelling occ
     3urs? ' )
1001  FORMAT ( ' If the starting date is within  the first quarter-month
     1,  enter 1, 2 if  before the first half month, 3 if before the thr
     2ee-quarter  month and 4 if in the last quarter month. ' )
1002  FORMAT ( ' The current run has ',I3,' periods - ',I3,' quarter-mon
     1thly and ',I3,' monthly -  before the end of the current growing s
     2eason. In the Maha season, the last month period is a half-month l
     3ong'/' If the number of periods is not correct, answer N ' )
```

```
1003  FORMAT ( I3,5X,A12,I3,5X,A12,I3,5X,A12,I3,5X,A12 )
1004  FORMAT ( ' Do any of the sytem constants CHANGE from now to the en
     1d of the growing season?'/' Answer 1 to read the variations stored
     2 on the diskette from last time'/ 8X,'2 to review the power produc
     3tion v/s storage curves'/ 8X,'3 to reveiw the restrictions on flow
     4 releases'/    8X,  '4 to review the diversion canal losses over t
     4he growing season'/ 8X,'5 to review the reservoir losses'/ 8X,'6 t
     5o review turbine max. and min. flows'/ 8X,'7 to review the irrigat
     6ion canal max. and min. flows'/ 8X,'8 to review the diversion cana
     7l max. and min. flows'/ 8X,'9 to review the reservoir max. or min.
     9 storage levels'/ 8X,'0 or nothing if no changes are required ' )
1005  FORMAT ( ' Do you wish to save the time varying constants for next
     2 time?  Answer Y to do so' )
1007  FORMAT ( ' There was no ',A13,' found at the node ',A12,' the numb
     1er ',I3 )
1015  FORMAT ( ' What is the INTEGER to the left of the time step you wo
     1uld like to change? ' )
1020  FORMAT ( ' These are the minimum and maximum storage volumes allow
     1ed in the reservoir '/ 1X,A12,' during the season: ' )
1006  FORMAT ( ' What is the INTEGER to the left of the value requiring
     1the ', I3, ' revision? ' )
1021  FORMAT ( ' By how much percent will the storage levels increase or
     2 decrease in the ',I3,'th'/' time step?   Answer in decimals inclu
     3ding a minus sign for reductions, giving'/' the variation for the
     4MAXIMUM storage volume if any.'/' If you answer 0 or nothing, the
     5limit will remain unchanged. ')
1022  FORMAT ( ' Give the variation of the MINIMUM permissable storage v
     1olume in decimals in-'/' cluding a minus sign for reductions. ' )
1023  FORMAT ( ' No reservoir was found at the point ',A12,' Please chec
     1k that you have entered  the INTEGER correctly, or enter 0 to retu
     2rn to the main menu.' )
1008  FORMAT ( I5,F10.1,5X,I5,F10.1,5X,I5,F10.1,5X,I5,F10.1 )
1024  FORMAT ( ' How many of the storage volume boumds would you like to
     1 revise? ')
1030  FORMAT ( ' These are the power production constants at the ', A12,
     1' power plant:'/' PERIOD', 13X, 1HH,13X, 1HC, 8X, 10HSTO in Mm3 )
1031  FORMAT ( ' In what time step do any of the constants differ from t
     1hose shown?'/' The ',I3,'th time step is the last a quarter-month
     2long.  Answer an INTEGER number. ' )
1032  FORMAT( ' What should the constant head factor, H, be during the '
     1,I3,'th time step?'/' If no change is required, answer nothing befor
     2ore the RETURN ' )
1332  FORMAT ( I5,3X,F14.4,F14.6,F14.1 )
1033  FORMAT ( ' What should the corrector multiplier, C, be during the'
     1,I4,'th time step?'/' If no change is required, answer nothing befor
     2ore the RETURN ' )
1034  FORMAT ( ' What should the reference storage, STo, be during the '
     1,I4,'th time step?'/' If no change is required, answer nothing bef
     2ore the RETURN ' )
1035  FORMAT ( ' No power plant was found at the point ',A12,' Please ch
     1eck you have entered  the INTEGER correctly, or enter 0 to return
     2to the main menu.' )
```

```
1036 FORMAT ( ' How many need to be revised?  Answer an INTEGER number
     1 ' )
1040 FORMAT ( ' These are the transport losses ( % of flow ) which will
     1 be applied to the diver-sion canal from ',A12,' to ',A12 )
1042 FORMAT ( ' By how much percent will the canal transport losses be
     2greater or less in the '/I3,'th time step?  Answer in decimals inc
     3luding a minus sign for decreased'/' losses. ' )
1041 FORMAT ( ' In what time step, identified by the INTEGER to the lef
     1t, do the canal losses   differ the ',I3,'th time? ' )
1043 FORMAT ( ' No diversion canal was found at the point ',A12,' Pleas
     1e check you have'/' entered the INTEGER correctly, or enter 0 to r
     2eturn to the main menu.' )
1050 FORMAT ( ' The values were advanced one time step ' )
1051 FORMAT ( ' What is the drive and the name of the file which has th
     1e values from last time? Enter b:FNAME where b: specifies the driv
     2e and FNAME the name of the file in 10 or less letters ' )
1052 FORMAT ( ' The file, ',A12,' stores data for the ',I3,' quarter-mo
     1nth of ',A5,'.  There   are ',I3,' periods, ',I3,' of them quarter
     2 months.  Is this the file whose variations you want to use this t
     3ime?'/' Enter 1 to change the file'/ 6X,' 2 to return to the last
     4menu without reading this file'/ 3X,' any other INTEGER or nothing
     5 to continue reading the values in this file. ' )
1053 FORMAT ( ' Do you want to use the variation values as read in, tha
     1t is, you have the same  starting date, or do you wish to advance
     2the variation values one quarter-month period?  Answer Y to use th
     3e values as read, without advancing. ' )
1082 FORMAT ( ' These are the reservoir losses in mm/period until the e
     1nd of the season at the '/ 1X,A12,' reservoir.'/ I7,F6.0,I7,F6.0,I
     27,F6.0,I7,F6.0,I7,F6.0,I7,F6.0,/ I7,F6.0,I7,F6.0,I7,F6.0,I7,F6.0,I
     37,F6.0,I7,F6.0/ I7,F6.0,I7,F6.0 )
1083 FORMAT ( ' Do you wish to amend any value?  Answer Y to correct th
     1e figures. ' )
1086 FORMAT ( ' What is the reservoir loss coefficient in mm during the
     1 duration of the ',I3,'th'/' period? ' )
1100 FORMAT ( ' These are the minimum and maximum turbinable flows in M
     1CM/period ( quarter-month and month ) until the end of the season
     2at the ',A12,' power plant.'/ I3,F7.1,I3,F7.1,I3,F7.1,I3,F7.1,I3,
     3F7.1,I3,F7.1,I3,F7.1,I3,F7.1/ I3,F7.1,I3,F7.1,I3,F7.1,I3,F7.1,I3,F
     47.1,I3,F7.1,I3,F7.1,I3,F7.1/ I3,F7.1,I3,F7.1,I3,F7.1,I3,F7.1,I3,F7
     5.1,I3,F7.1,I3,F7.1,I3,F7.1/ I3,F7.1,I3,F7.1,I3,F7.1,I3,F7.1 )
1103 FORMAT ( ' Do you wish to amend any value due to plant maintenance
     1 or repairs?'/' Enter 2 to change a maximum flow'/ 7X,  '1 to chan
     2ge a minimum flow'/ 3X,' or 0 to continue with these values. ' )
1101 FORMAT ( ' What is the minimum turbinable flow in MCM/period ( mon
     1th or quarter-month )    during the ',I3,'th period? ' )
1102 FORMAT ( ' What is the maximum turbinable flow in MCM/period ( mon
     1th or quarter-month )    during the ',I3,'th period? ' )
1104 FORMAT ( ' What is the number for the specific ',A13,' you wish to
     1 review?'/' Enter 99 if you wish to see the values for every node
     2or one INTEGER'/'  If you enter 0 or nothing you will return to th
     3e main menu ' )
1200 FORMAT ( ' These are the minimum and maximum canal capacities'/' i
```

```
     1n MCM/period(quarter-month and month) until the end of the season
     2for the '/ 1X,A6,' district.'  / I3,F7.1,I3,F7.1,I3,F7.1,I3,F7.1,
     3I3,F7.1,I3,F7.1,I3,F7.1,I3,F7.1/ I3,F7.1,I3,F7.1,I3,F7.1,I3,F7.1,I
     43,F7.1,I3,F7.1,I3,F7.1,I3,F7.1/ I3,F7.1,I3,F7.1,I3,F7.1,I3,F7.1,I3
     5,F7.1,I3,F7.1,I3,F7.1,I3,F7.1/ I3,F7.1,I3,F7.1,I3,F7.1,I3,F7.1 )
1203  FORMAT ( ' Do you wish to amend any value due to seasonal higher l
     1osses?'/' Enter 2 to change a maximum channel capacity'/  6X, ' 1
     2to change a minimum'/ 4X,'or 0 to continue with these values. ' )
1201  FORMAT ( ' What is the minimum capacity flow in MCM/period (month
     1or quarter-month) during the ',I3,'th period? ' )
1202  FORMAT ( ' What is the maximum capacity flow in MCM/period (month
     1or quarter-month) during the ',I3,'th period? ' )
1222  FORMAT ( ' The crop water requirements have been adjusted to start
     1 at the previous startingdate.  Enter a number to continue with th
     2e values for that date'/' or nothing to re-adjust the crop water r
     3equirement data'/' ( You must re-enter the crop water requirements
     4 and the date anew '/' before adjusting them to a new starting dat
     5e )! ' )
1250  FORMAT ( ' These are the minimum and maximum flow restrictions in
     1MCM/period (quarter-monthand month) until the end of the season at
     2 the node ',A12/3(4(I3,F7.2,I3,F7.1)/),2(I3,F7.2,I3,F7.1) )
1251  FORMAT ( ' What is the minimum allowed flow in MCM/period ( month
     1or quarter-month ) duringthe ',I3,'th period? ' )
1252  FORMAT ( ' What is the maximum allowed flow in MCM/period ( month
     1or quarter-month ) duringthe ',I3,'th period? ' )
1253  FORMAT ( ' Do you wish to amend any value due to seasonal differen
     1ces?'/' Enter 2 to change a maximum flow'/ 7X,  '1 to change a min
     2imum flow'/ 3X,' or 0 to continue with these values. ' )
1300  FORMAT ( ' These are the minimum and maximum canal capacities in M
     1CM/period( quarter-month and month ) until the end of the season f
     2or the diversion from ',A12/' to ',A12 )
1301  FORMAT ( I3,F7.1,I3,F7.1,I3,F7.1,I3,F7.1,I3,F7.1,I3,F7.1,I3,F7.1,I
     13,F7.1/ I3,F7.1,I3,F7.1,I3,F7.1,I3,F7.1,I3,F7.1,I3,F7.1,I3,F7.1,I3
     2,F7.1/ I3,F7.1,I3,F7.1,I3,F7.1,I3,F7.1,  I3,F7.1,I3,F7.1,I3,F7.1,I
     33,F7.1/ I3,F7.1,I3,F7.1,I3,F7.1,I3,F7.1 )
1302  FORMAT ( ' Do you wish to amend any value due to seasonal higher l
     7osses?'/' Enter 2 to change a maximum channel capacity'/ 7X,'1 to
     8change a minimum'/ 3X,' or 0 to continue with these values. ' )
1400  FORMAT ( ' Do you wish to save the updated version under the same
     1filename, ',A15,' Enter b:FNAME where b: is the drive and FNAME is
     2 the new name or enter nothing to keep the same name. ' )
8000  FORMAT ( 4I4 )
8001  FORMAT ( 2(1X,F7.4,F9.6,3F8.1) )
8002  FORMAT ( 8F10.2 )
9001  FORMAT ( I2 )
9002  FORMAT ( A2 )
9003  FORMAT ( F8.0 )
9004  FORMAT ( 2F4.0 )
9005  FORMAT ( A12 )
      RETURN
      END
^Z
```

```
      SUBROUTINE DINFLO ( I4, II, NN, CT7 )
c     I4 = number of time steps of a quarter month in the current run
c     II = total number of time steps in the current run
c     NN = total number of nodes
c    CT7 = integer code (1 or 0 ) indicating entry of inflow data is
c          complete
c
c  This subroutine enters the predicted inflows for all local inflow
c      nodes for every time step.  Values may be corrected but not saved.
      COMMON/SHP/ NINFLO(30), NLIM(31), NTRB(2,10), NRF(3,10), C1(26,10)
      COMMON/INF/ FLOW(26,40)
      COMMON/NOM/ NNAME(40), WREQ(21), EVAP(25), INFLOW(30)
      INTEGER  CT7
      CHARACTER * 12 NAME1, NNAME, WREQ, EVAP, FNAME, INFLOW,
     1                 FIRST(6)
      CHARACTER * 2 ANS
C
      WRITE ( *,1000 ) I4, II
      READ  ( *,9000 ) FNAME
      IF ( FNAME .EQ. '            ' ) GO TO 300
C  Read in the INFLOW file, checking for the right place and number
c  of time periods.
      OPEN ( 9,FILE=FNAME, STATUS='OLD')
      READ  ( 9,9100 ) NNF, ( FIRST(M), M=1,6 )
      IF ( NINFLO(1) .EQ. NNF ) GO TO 102
      WRITE ( *,2002 ) NINFLO(1), NNF, (FIRST(M), M=1,6)
        CT7 = 0
       GO TO 999
102     DO 105 N=1,NNF
           READ  ( 9,9102 ) FNAME, I41, II1
           NP1 = N + 1
           IF ( FNAME .NE. INFLOW(NP1) ) GO TO 199
           IF ( I41 .NE. I4 ) GO TO 198
           IF ( II1 .NE. II ) GO TO 197
105         READ  ( 9,9101 ) ( FLOW(I,NINFLO(NP1)), I=1,II )
        CT7 = 1
       CLOSE ( 9 )
      GO TO 300
C  The total number of periods on the datafile is not the same as
C  calculated in DPERIOD
197    WRITE ( *,2003 ) INFLOW(NP1), II1, II
        CT7 = 0
        GO TO 999
C  The number of quarter-month periods on the datafile is not the same
C  as calculated in DPERIOD
198    WRITE ( *,2004 ) INFLOW(NP1), I41, I4
        CT7 = 0
      GO TO 999
C  The names of the local inflow basins in the datafile don't match the
C  names given in the system configuration.
199    WRITE ( *,2005 ) INFLOW(NP1), N, FNAME
        CT7 = 0
```

```
      GO TO 999
C  Display the local inflow data and ask for corrections, by screen
300   NBEG = 1
     NTIMES = NN/8 + 1
     DO 220 L=1,NTIMES
301     WRITE ( *,3000 ) ( I, I=1,II )
       NEND = 7 + NBEG
       IF ( NEND .GT. NN ) NEND = NN
     DO 200 N=NBEG, NEND
C  For every node N, list the local inflows the program will use
200   WRITE ( *,3001 )   N, NNAME(N), ( FLOW(I,N), I=1,II )
     WRITE ( *,3002 )
     READ  ( *,9002 ) KX
     IF ( KX .LE. 0 ) GO TO 215
C  Corrections need to be made
     DO 210 K1=1,KX
       WRITE ( *,3003 ) K1
       READ  ( *,9002 ) NX
       WRITE ( *,3004 ) NNAME(NX)
       READ  ( *,9002 ) IX
       WRITE ( *,3005 ) NNAME(NX), IX
210      READ  ( *,9003 ) FLOW(IX,NX)
     GO TO 301
215   NBEG = NEND + 1
220   CONTINUE
C
1000  FORMAT (      ' What is the name of the file you have prepared of
    2the local drainage to the'/' nodes listed in the configuration fil
    3e?  The predicted inflows must have ',I3/'  quarter-month periods
    4at the beginning and ',I3,' periods in all'/' Enter b:FNAME, where
    5 b: is the drive and FNAME is the name in 10 or less lettersIf you
    6 only wish to review the inflows already entered, enter nothing be
    7fore'/' RETURN ' )
2002  FORMAT ( ' The current configuration lists ',I3,' nodes for which
    1local inflow files have beenprepared.  The datafile lists ',I3,' a
    2s the number of nodes.  The first line of the datafile is '/ 8X,
    3 6A12  )
2003  FORMAT ( ' The node with the inflow name ',A6,' lists ',I3,' numbe
    1r of periods and not ',I3  )
2004  FORMAT ( ' The node with the inflow name ',A6,' lists ',I3,' quart
    1er-month long periods and    not ',I3 )
2005  FORMAT ( ' The node with the inflow name ',A6,', is the ',I3,'th n
    1ode with significant local  inflow in the configuration.  However,
    2 the name ',A6,' appears on the datafile.  Please check the spelli
    3ng, the order in the file, or the file. ' )
3000  FORMAT ( 30X,37HLOCAL INFLOW ( MCM ) IN THE TIME STEP/
    13X,3HNo.,3X,9HNODE NAME,11X,I1,7X,I1,7X,I1,7X,I1,7X,I1,7X,
    2I1,7X,I1/T20,7X,I1,7X,I1,6X,I2,6X,I2,6X,I2,6X,I2,6X,I2/ )
3001  FORMAT ( I5,2X,A15,2X,7F8.2/20X,7F8.2 )
3002  FORMAT ( ' How many values would you like to correct?  Enter an IN
    1TEGER number ' )
3003  FORMAT ( ' What is the NUMBER of the ',I3,'st node which you would
```

```
     1 like to correct? ' )
3004  FORMAT ( ' Enter the time step NUMBER above the incorrect inflow v
     1alue for the node ',A12,1H  )
3005  FORMAT ( ' Enter the correct local inflow for the node ',A12, ' du
     1ring the time step ',I3,1H  )
9000  FORMAT ( A12 )
9001  FORMAT ( A2 )
9002  FORMAT ( I5 )
9003  FORMAT ( F10.0 )
9100  FORMAT ( 1X,I3,6A12 )
9101  FORMAT ( 1X,12F6.0 )
9102  FORMAT ( 1X,A4,2I4 )
999   CLOSE (9)
      RETURN
      END
^Z
      SUBROUTINE RETQR ( NN, I4, II, IRT, IDATE )
c     NN = total number of nodes
c     I4 = number of time steps of a quarter month in the current run
c     II = total number of time steps in the current run
c    NNN = number of instances of return flow
c IDATE = the order that the first quarter month of the current run is
c          in the season
c
C This subroutine calculates the extra inflow due to return flow from
C          a) already delivered irrigation flows
C          b) anticipated irrigation flows
      COMMON/SHP/ NINFLO(30), NLIM(31), NTRB(2,10), NRF(3,10), C1(26,10)
      COMMON/INF/ FLOW(26,40)
      COMMON/NOM/ NNAME(40)
      CHARACTER * 12 NNAME, FNAME, NAME1, FIRST(6)
      CHARACTER * 2 ANS
C Read in the data file of flows whence, whither, when and what quantity
1     WRITE ( *,1000 )
      READ  ( *,9000 ) FNAME
      IF ( FNAME .EQ. '            ' ) GO TO 3
      OPEN ( 9,FILE=FNAME, STATUS='OLD')
      READ ( 9,9002 ) IRT, ( FIRST(N), N=1,6 )
      WRITE ( *,1003) IRT, ( FIRST(N), N=1,6 )
      READ  ( *,9001 ) ANS
      IF ( ANS .EQ. 'N' .OR. ANS .EQ. 'n' ) GO TO 550
      READ ( 9,9010 ) ( NRF(1,K), NRF(2,K), NRF(3,K), K=1,IRT )
      READ ( 9,9011 ) ( ( C1(I,K), I = 1,26 ), K=1,IRT )
      CLOSE ( 9 )
C Display and correct
3     WRITE ( *,2000 )
      WRITE ( *,2001 ) ( K, NNAME(NRF(1,K)), NNAME(NRF(2,K)), NRF(3,K),
     1                                    K=1,IRT )
331   WRITE ( *,2002 )
      READ  ( *,9002 ) NX
      IF ( NX .LE. 0 ) GO TO 340
C  Additions, deletions or corrections are required to the irrigation
```

```
C  district flows
      WRITE ( *,2006 )
      READ  ( *,9002 ) ICH
       GO TO ( 333, 322, 311 ) ICH
      GO TO 331
C Add more Turnouts which have return flows
311   WRITE ( *,2007 )
      READ  ( *,9002 ) NX
       IF ( NX .LE. 0 ) GO TO 3
      WRITE ( *,9004 ) ( N, NNAME(N), N=1,NN )
        DO 312 N=1,NX
          WRITE ( *,2008 ) N
          READ  ( *,9002 ) NRF(1,IRT+N)
          WRITE ( *,2009 )
          READ  ( *,9002 ) NRF(2,IRT+N)
          WRITE ( *,2099 ) NNAME(NRF(1,IRT+N)), NNAME(NRF(2,IRT+N))
312        READ  ( *,9002 ) NRF(3,IRT+N)
        IRT = IRT + NX
        GO TO 3
C  Deletions
322   WRITE ( *,2011 )
      READ  ( *,9002 ) NX
        IF ( NX .LE. 0 ) GO TO 3
        DO 324 N=1,NX
          WRITE ( *,2012 ) N
           READ ( *,9002 ) KX
            DO 324  M=KX,IRT
              IP1 = 1 + M
         NRF(3,M) = NRF(3,IP1)
         NRF(2,M) = NRF(2,IP1)
324       NRF(1,M) = NRF(1,IP1)
        IRT = IRT - 1
         GO TO 3
C  Corrections
333   WRITE ( *,2020 )
      READ  ( *,9002 ) NX
       IF ( NX .LE. 0 ) GO TO 3
        DO 339 N=1,NX
          WRITE ( *,2021 )
          READ ( *,9002 ) ICH
          WRITE ( *,2022 )
          READ ( *,9002 ) KX
          GO TO ( 334, 335, 337 ) ICH
             GO TO 3
C  Node of turnout number wrong
334   WRITE ( *,9004 ) ( N, NNAME(N), N=1,NN )
      WRITE ( *,2023 ) KX, NNAME(NRF(2,KX))
       READ ( *,9002 ) NRF(1,KX)
         GO TO 339
C Node of return wrong
335   WRITE ( *,9004 ) ( N, NNAME(N), N=1,NN )
      WRITE ( *,2024 ) KX, NNAME(NRF(1,KX))
```

```
      READ ( *,9002 ) NRF(2,KX)
        GO TO 339
C Lag-time of return flow wrong
337   WRITE ( *,2029 ) NNAME(NRF(1,KX)), NNAME(NRF(2,KX)), KX
      READ ( *,9002 ) NRF(3,KX)
339   CONTINUE
     GO TO 3
C Display percentages for each node for all quarter-months
C Vary with the time period if necessary
340   DO 360 J=1,IRT
345      WRITE ( *,3000 ) NNAME(NRF(1,J)), NNAME(NRF(2,J))
        CALL TABLE ( J, IDATE )
        WRITE ( *,2020 )
        READ  ( *,9002 ) NX
        IF ( NX .LE. 0 ) GO TO 360
C Percentage of irrigation wrong
        DO 350 N=1,NX
          WRITE ( *,3001 ) N
          READ  ( *,9002 ) IX
          WRITE ( *,3002 ) IX
350        READ ( *,9003 ) C1(IX,J)
       GO TO 345
360   CONTINUE
C write updates to disk
     WRITE (*,1001)
     READ  (*,9001) ANS
     IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'y' ) GO TO 365
       GO TO 380
365   WRITE ( *,1002 ) FNAME
     READ  ( *,9000 ) NAME1
     IF ( NAME1 .EQ. '            ' ) GO TO 375
       FNAME = NAME1
375   OPEN ( 10,FILE=FNAME, STATUS='OLD')
     WRITE ( 10,9002) IRT, ( FIRST(K), K=1,6 )
     WRITE ( 10,9010 ) ( NRF(1,K), NRF(2,K), NRF(3,K), K=1,IRT )
     WRITE ( 10,9011 ) ( ( C1(I,K), I = 1,26 ), K=1,IRT )
     CLOSE ( 10 )
C Ask for the actual quantity of previous irrigation, calculate what
C  will therfore return, and add to local inflow amount in array
380   DO 500 J=1,IRT
        C1LAST = C1(1,J)
     DO 390 K=1,NRF(3,J)
          KP = NRF(3,J) + 1 - K
         IFD = IDATE - K
       IF ( IFD .GT. 0 ) GO TO 385
       WRITE ( *,4001 ) NNAME(NRF(1,J)), NNAME(NRF(2,J)), K, C1LAST
       READ  ( *,9003 ) C1(IFD,J)
       C1LAST = C1(IFD,J)
       IF ( C1(IFD,J) .LE. 0 ) GO TO 390
385     WRITE ( *,4000 ) NNAME(NRF(1,J)), NNAME(NRF(2,J)), K
       READ  ( *,9003 ) QR
       FLOW(KP,NRF(2,J)) = FLOW(KP,NRF(2,J)) + C1(IFD,J)*QR
```

```
     1from ',A12,' reach  ',A12,'?  Enter an INTEGER value ' )
2011  FORMAT ( ' How many do you wish to delete?  Enter an INTEGER value
     1 ' )
2012  FORMAT ( ' What is the INTEGER of the ',I3,'th line you wish to de
     1lete? ' )
2020  FORMAT ( ' How many do you wish to correct?  Enter an INTEGER valu
     1e. ' )
2021  FORMAT ( ' What do you wish to change?' /' Enter 1 - change the  n
     1ode of the irrigation turnout'/ 7X,'2 - change the  node to where
     2they return'/ 7X,'3 - change the time lag ')
2022  FORMAT ( ' What is the number of the line you wish to change?  Ent
     1er the INTEGER to the    left of the line. ' )
2023  FORMAT ( ' What is the INTEGER of the  node in line ',I3,' of the
     1table whose irrigation      turnouts will return to ',A12,1X )
2024  FORMAT ( ' What is the INTEGER  of the  node in line ',I3,' to whi
     1ch  some of the irrigation  turnout of the  node ',A12,' does flow
     2? ' )
2029  FORMAT ( ' How many quarter-months later does a percentage of the
     1irrigation flows from the node ',A12,' reach the node ',A12,' of l
     2ine ',I3,' of the table?'/' Enter an INTEGER number ' )
3000  FORMAT ( ' This table shows the percentage of the irrigation flows
     1 delivered from the node ',A12,' which will eventually reach the n
     2ode ',A12,'.  Note that the   percentage may vary from one quarter
     3-month to the next' )
3001  FORMAT ( ' What is the INTEGER for the quarter-month above the ',I
     13,'st percentage you wish to correct? ' )
3002  FORMAT ( ' What is the correct percentage for the ',I3,' quarter-m
     1onth?  Enter in decimals ' )
4000  FORMAT ( ' What was the actual irrigation DELIVERY from ',A12,' (
     1some of which will  drain to ',A12,') that occured ',I3,' quarter-
     2months ago? ' )
4001  FORMAT ( ' What is the PERCENTAGE of the actual irrigation deliver
     1y from ',A12/' ( some of which will drain to ',A12,') that occured
     2 ',I3,' quarter-months ago?',' The last percentage was ',F6.3,' En
     3ter a decimal number ' )
9000  FORMAT ( A12 )
9001  FORMAT ( A2 )
9002  FORMAT ( I5,6A12 )
9003  FORMAT ( F10.0 )
9004  FORMAT ( I5,2X,A12,I5,2X,A12,I5,2X,A12,I5,2X,A12 )
9010  FORMAT ( 20I4 )
9011  FORMAT ( 8F10.3 )
200   CONTINUE
      RETURN
      END
C
      SUBROUTINE TABLE ( J, IDATE )
      COMMON/SHP/ NINFLO(30), NTRB(2,10), NRF(3,10), C1(26,10)
C This subroutine displays the proper heading and spacing according
C  to the season Yala or Maha
      IF ( IDATE .GT. 26 ) GO TO 405
C Display Maha season table
```

```
      WRITE ( *,3009 ) ( I, I=1,12 ), ( C1(I,J), I = 1, 12 ),
   1                   ( I, I=13,24 ), ( C1(I,J), I = 13,24 )
      WRITE ( *,3099 ) C1(25,J), C1(26,J)
    GO TO 500
C Display Yala season table
405    WRITE ( *,4009 ) ( I, I=1,10 ), ( C1(I,J), I = 1, 10 ),
   1                   ( I, I=11,22 ), ( C1(I,J), I = 11,22 )
C
500  CONTINUE
C
3009 FORMAT ( 12X,3HOCT,23X,5H  NOV,22X,5H  DEC/ 1X,I4,3I6,4X,4I6,4X,
    14I6/4F6.3,4X,4F6.3,4X,4F6.3/ 12X,3HJAN,23X,5H  FEB,22X,5H  MAR/
    11X,I4,3I6,4X,4I6,4X,4I6/4F6.3,4X,4F6.3,4X,4F6.3 )
3099 FORMAT ( 12X,3HAPR/ 3X,2H25,3X,2H26/ 2F6.3 )
4009 FORMAT ( 12X,3HAPR,23X,5H  MAY,22X,5H  JUN/ 12X,I4,I6,4X,4I6,4X,
    14I6/12X,2F6.3,4X,4F6.3,4X,4F6.3/ 12X, 3HJUL,23X,5H  AUG,22X,3HSEP/
    11X,I4,3I6,4X,4I6,4X,4I6/4F6.3,4X,4F6.3,4X,4F6.3 )
    RETURN
    END
^Z
    SUBROUTINE DWRITE ( NN, II, I4, NNN, MINPOW )
c     NN = total number of nodes
c     II = total number of time steps in the current run
c     I4 = number of time steps of a quarter month in the current run
c MINPOW = integer code for the variation of the formulation desired
c          ( 0 - allow large fluctuations
C            1 - meet minimum energy targets each time step
c            2 - limit the fluctuation to a maximum value
c            3 - minimize the fluctuations with objective function )
c
C This subroutine writes the B:Release datafile, calling on other
C subroutines as needed.  A vector of nodes which release downstream
C flows to the node numbered one higher is created, as is a vector of
C power plants with those that turbine flows which are diverted before the
C power plants which are not at diversion nodes or which turbine the flows
C delivered downstream to the node numbered one higher.
c  The constraints and variables specific to:
C              reservoir storage are written first,
C              then those for the power plants,
C              thirdly, those for irrigation
c             fourthly those for downstream and tributary flows
C              fifthly those variables defining diverted flows
C              lastly those involved in energy production levels.
c  There are subroutines to write the power, reservoir, and irrigation
variables and constraints, and general subroutines defining format lines
c
    COMMON/SHR/ NPOWER(15), NIRR(21), NRES(25), NNOSPL(20), NDIV(2,15)
    COMMON/SHP/ NINFLO(30), NLIM(31), NTRB(2,10), NRF(3,10), C1(10,10)
    COMMON/NEW/ FLOMIN(30), FLOMAX(30), FMX(14,30), FMN(14,30)
    COMMON/CAN/ CCAPI(21), CCAPD(15), CMINI(21), CMIND(15), CHLSS(15)
    COMMON/CAP/ TMAX(15), TMIN(15), SUP(25), SLO(25)
    COMMON/VAR/ VLS(14,25),VMS(14,25),CIX(14,20),CIN(14,20),
```

```
     1        TMX(14,15), TMN(14,15), CDX(14,15), CDN(14,15), VL(14,15)
      COMMON/STG/ ST0(40), ST12(40), APOW(14), ERHS(14)
      COMMON/POW/ H(14,15), C(14,15), STO(14,15)
      COMMON/RES/ GAMMA(25,9), DELTA(25,9), EPSI(25,9), KIN(25),
     1           DGAMA(25,9), CLOSS(14,25)
      COMMON/IRR/ THETA(6,21,9), OMEGA(6,21,9), PHI(6,21,9), ID(6,21),
     1        KINI(6,21), DTHET(6,21,9), IC(21), CWR(26,21), IID(14,21)
      COMMON/INF/ FLOW(26,40)
      COMMON/RSC/ RO1, RO2, RO3(40), RO4
      COMMON/NOM/ NNAME(40)
C
      DIMENSION N1(15), K1(15), NSPILL(40), FLOWN(14), NPOW(15)
      CHARACTER * 250 LF
      CHARACTER * 75 DOT1
      CHARACTER * 60 DOT2
      CHARACTER * 45 DOT3
      CHARACTER * 31 DOT4
      CHARACTER * 17 DOT5
      CHARACTER * 4 DOT6
      CHARACTER * 20 RHS, LE, GE, EQ
      CHARACTER * 16 UPB, LOW
      CHARACTER * 12 NNAME
      CHARACTER * 7 HDOT, ST, CNT, IFM(4), QT, QS, QD, POW, PDF, EN, EP
      CHARACTER * 4 HN
      CHARACTER * 2 ANS
C
      DATA RHS/' ( 2H .,6H _RHS_,1X,'/      EQ/' ( 3H EQ, 1X,1H.,1X,'/
      DATA LE/' ( 3H LE, 1X,1H.,1X,'/,      GE/' ( 3H GE, 1X,1H.,1X,'/
      DATA HDOT/'( 2H .,'/, ST/'4H  ST,'/, HN/'1HN,'/, CNT/'4H CNT,'/
      DATA UPB/'5H UPB ,F7.1,1X,'/, LOW/'5H LOW ,F5.1,1X,'/
      DATA QT/'4H QTB,'/, PDF/'4H PDF,'/, EN /'4H ENG,'/, EP /'4H  EP,'/
      DATA QS/'4H  QS,'/,  QD/'4H  QD,'/,  POW/'4H POW,'/,  NTIMES/0/
      DATA IFM/'  I1,  ',  '  I2,  ', ' 3H -1,', ' 3H 1 ,'/
C
      DATA DOT1/' 2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H
     1.,2X,1H.,2X,1H. )'/
      DATA DOT2/' 2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H
     1. ) '/
      DATA DOT3/' 2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H. ) '/
      DATA DOT4/' 2X,1H.,2X,1H.,2X,1H.,2X,1H. ) '/
      DATA DOT5/' 2X,1H.,2X,1H. ) '/
      DATA DOT6/'  ) '/
C
C  Divide CLOSS(i,j) in mm by 1000 to have in metres.  After
C   multiplication by surface area of a reservoir, in km2, the volume will
C   be in MCM, as desired.
        DO 250 I=1,II
        DO 250 J=1,NRES(1)
250       CLOSS(I,J) = CLOSS(I,J)/1000
C  Definition of NSPILL vector where spills and turbine flows
C  flow from the point j to the point j+1
          L = 2
```

```
          IN = 1
      DO 110 N=1,NN
        IF ( NNOSPL(L) .EQ. N ) GO TO 100
          NSPILL(IN) = N
                IN = IN + 1
          GO TO 110
100     L = L + 1
110     CONTINUE
C  Find the nodes where there is a diversion and a power plant.
105   DO 101 K=1,NPOWER(1)+1
101     NPOW(K) = NPOWER(K)
          LL = 1
      DO 120 N=2,NPOWER(1)+1
        DO 120 K=2,NDIV(1,1)+1
          IF ( NDIV(1,K) .EQ. NPOWER(N) ) GO TO 115
          GO TO 120
115        N1(LL) = N
          K1(LL) = K
             LL = LL + 1
120    CONTINUE
      IF ( LL .LE. 1 ) GO TO 200
C  Ask where the turbine flows go  - downstream
C                                  - or to the diversion
      WRITE (*,1001 )
      WRITE (*,1011 ) ( NNAME(NPOWER(N1(L))), NNAME(NDIV(2,K1(L))),
     1                             L=1,LL-1 )
      WRITE (*,1021 )
      READ  (*,9002 ) NTIMES
      IF ( NTIMES .LE. 0 ) GO TO 200
C  There are NTIMES nodes whose turbine flows go to a diversion.  Now
C  check that the first NTIMES nodes listed in the NPOWER array are the
C  nodes n whose turbine flows go to a diversion.
125    WRITE ( *,1002 ) ( L, NNAME(NPOW(L)), L=2,NPOW(1)+1 )
      WRITE ( *,1003 ) NTIMES
      READ  ( *,9002 ) LX
      IF ( LX .LE. 0 ) GO TO 200
      IF ( LX .GT. NPOW(1)+1) GO TO 125
        LX2 = NPOW(LX)
C  Move the node number of the node which does divert its turbine flow
C  to the second spot in the NPOWER vector, shifting others to its spot
      DO 130 N=1,LX-2
130        NPOW(LX-N+1) = NPOW(LX-N)
      NPOW(2) = LX2
      GO TO 125
C  Start to write to b:RELEASE.DATA
200   WRITE ( *,1000 )
      READ  ( *,9000 ) ANS
      OPEN ( 8, FILE='B:RELEASE.DAT', STATUS='OLD' )
      WRITE ( 8,2000 )
C  Write the lines defining the universal equations - UPB, LOW
C                                                    OBJ, INT
      WRITE ( 8,2001 )
```

```
      WRITE ( 8,2002 )
      WRITE ( 8,2003 )
      WRITE ( 8,2004 )
C  Write variables and equations for points with reservoirs
      DO 10 L=2,(NRES(1)+1)
           J = NRES(L)
      IF ( KIN(L-1) .GT. 1 ) CALL WRES( J,II,L )
c  J = node number of current node
c II = number of time steps in current run
c  L = reservoir number of current node
C  Write STiPj, storage volume variable, in the UPB, LOW, CNTiPj and
C     ( for every i, and every j )              CNTi+1Pj constraints
C  First the last period only
        JL=1
        IF ( J .GT. 9 ) JL=2
        IL=1
        IF ( II .GT. 9 ) IL=2
       IF ( KIN(L-1) .LE. 1 ) GO TO 3
        LF = HDOT//ST//IFM(IL)//HN//IFM(JL)//CNT//IFM(IL)//HN//IFM(JL)
     1       //IFM(4)//UPB//LOW//'2H R,'//IFM(IL)//HN//IFM(JL)//IFM(3)//
     2      '5H OBJ ,F8.2,'//DOT5
       IF ( IL .GE. 2  .OR.  JL .GE. 2 ) LF = HDOT//ST//IFM(IL)//HN//
     1                                  IFM(JL)//CNT//IFM(IL)//HN//IFM(JL)
     2                                  //IFM(4)//UPB//LOW//' /10X,2H R,'
     3                                  //IFM(IL)//HN//IFM(JL)//IFM(3)//
     4                                  ' 5H OBJ ,F8.2,'//DOT5
        WRITE ( 8,LF )  II,J, II,J, SUP(L-1), ST12(J), II,J, RO3(J)
              GO TO 5
C  The evaporation and seepage losses are a cste factor * closs * ST
3       IF ( KIN(L-1) .LE. 0 ) GO TO 4
         LF = HDOT//ST//IFM(IL)//HN//IFM(JL)//CNT//IFM(IL)//HN//IFM(JL)
     1       //'F7.3,'//UPB//LOW//' 4H OBJ,F7.2,'//DOT4
        IF ( J .GT. 9 .OR. II .GT. 9 ) LF = HDOT//ST//IFM(IL)//HN//
     1                   IFM(JL)//CNT//IFM(IL)//HN//IFM(JL)//'F8.3,'//
     2                   UPB//' / 10X,'//LOW//' 5H OBJ ,F8.2,'//DOT4
        WRITE ( 8,LF )  II,J, II,J, ( 1 + DELTA(L-1,1)*CLOSS(II,L-1) ),
     1                  VMS(II,L-1)*SUP(L-1), ST12(J), RO3(J)
              GO TO 5
C  There are no intervals on the reservoir loss curve, there are no
C  losses, the node is likely the sink node the SEA
4        LF = HDOT//ST//IFM(IL)//HN//IFM(JL)//CNT//IFM(IL)//HN//IFM(JL)
     1       //IFM(4)//DOT1
        WRITE ( 8,LF )  II,J, II,J
C  For all periods but the last
5       DO 10 I=1,II-1
        IL=1
        IF ( I .GT. 9 ) IL=2
        IP=1
        I2 = I + 1
        IF ( I2 .GT. 9 ) IP=2
        IF ( KIN(L-1) .LE. 1 ) GO TO 8
        LF = HDOT//ST//IFM(IL)//HN//IFM(JL)//CNT//IFM(IL)//HN//IFM(JL)
```

```
     1 //IFM(4)//CNT//IFM(IP)//HN//IFM(JL)//IFM(3)//UPB//LOW//'2H R,'//
     2 IFM(IL)//HN//IFM(JL)//IFM(3)//DOT5
       IF ( IL .GE. 2  .OR.  JL .GE. 2 ) LF = HDOT//ST//IFM(IL)//HN//
     1                               IFM(JL)//CNT//IFM(IL)//HN//IFM(JL)
     2                               //IFM(4)//CNT//IFM(IP)//HN//
     3                               IFM(JL)//IFM(3)//UPB//LOW//
     4                               ' /10X,2H R,'//IFM(IL)//HN//
     5                               IFM(JL)//IFM(3)//DOT5
        WRITE ( 8,LF )  I,J, I,J, I2,J, VMS(I,L-1)*SUP(L-1),
     1                                VLS(I,L-1)*SLO(L-1), I,J
             GO TO 10
C  The evaporation and seepage losses are a cste factor * closs * ST
8       IF ( KIN(L-1) .LE. 0 ) GO TO 9
        LF = HDOT//ST//IFM(IL)//HN//IFM(JL)//CNT//IFM(IL)//HN//IFM(JL)
     1       //' F8.3,'//CNT//IFM(IP)//HN//IFM(JL)//IFM(3)//UPB//LOW//
     2       DOT4
       IF ( IL .GT. 1 .OR. JL .GT. 1 ) LF = HDOT//ST//IFM(IL)//HN//
     1                             IFM(JL)//CNT//IFM(IL)//HN//IFM(JL)
     2                             //' F8.3,'//CNT//IFM(IP)//HN//IFM(JL)
     3                             //IFM(3)//' / 10X,'//UPB//LOW//DOT4
        WRITE ( 8,LF )  I,J, I,J, ( DELTA(L-1,1)*CLOSS(I,L-1) ) + 1,
     1                 I2,J, VMS(I,L-1)*SUP(L-1), VLS(I,L-1)*SLO(L-1)
             GO TO 10
C  There are no intervals on the reservoir loss curve, there are no
C  losses, the node is likely the sink node the SEA
9       LF = HDOT//ST//IFM(IL)//HN//IFM(JL)//CNT//IFM(IL)//HN//IFM(JL)
     1       //IFM(4)//CNT//IFM(IP)//HN//IFM(JL)//IFM(3)//DOT2
        WRITE ( 8,LF )  I,J, I,J, I+1,J
10    CONTINUE
C  Write the ENG variable and the type line for the EPiNn equation for
C    the nodes with power plants
      DO 16 L=1, NPOWER(1)
         J = NPOWER(L+1)
      CALL LFTYPE ( II, 10, EQ, ' 3H EP,', J )
c        II = number of time steps in current run
c        10 = code for inclusion of node number in constraint name
c        EQ = type of constraint, in this case an equality
c ' 3H EP,' = the piece of format line which will be used to write the
c             constraint name in the data line
c         J = node number of the current node
        JL = 1
        IF ( J .GT. 9 ) JL = 2
        DO 15 I=1,II
          IL = 1
          IF ( I .GT. 9 ) IL = 2
        IF ( C(I,L) .GT. 0 ) CALL WPOWER ( J, I, II, L )
c  J = node number of the current node
c  I = number of current time step
c II = number of time steps in the current run
c  L = power plant number in original NPOWER vector of the current node
c
C  Write the ENGiNn variable in EP, OBJ, and any other applicable
```

```
C   constraints  -  UPB, LOW, POW, etc
          IF ( MINPOW .LE. 0 ) GO TO 216
          IF ( MINPOW .LT. 2 ) GO TO 215
C MINPOW =2,3 per period power difference calculated
             IM = 1
             IF ( I-1 .GT. 9 ) IM = 2
             IF ( I .GE. II .OR. I .LE. 1 ) GO TO 221
     LF = HDOT//EN//IFM(IL)//HN//IFM(JL)//PDF//IFM(IL)//'I5,'
    1     //PDF//IFM(IM)//IFM(3)//'5H OBJ ,F6.2,'//EP//IFM(IL)
    2     //HN//IFM(JL)//IFM(3)//DOT4
             IF ( I4 .EQ. I ) GO TO 220
     WRITE (8,LF) I,J, I,1, I-1, RO1, I,J
          GO TO 15
220   WRITE (8,LF)  I,J,  I,4, I-1, RO1, I,J
          GO TO 15
C First and last periods are in only one difference calculation
221       IEQ = 1
         SIGN = 1
     IF ( I .LT. II ) GO TO 214
          IEQ = II - 1
         SIGN = -1
214    LF = HDOT//EN//IFM(IL)//HN//IFM(JL)//PDF//IFM(IM)//'F4.0,'//
    1       ' 5H OBJ ,F6.2,'//EP//IFM(IL)//HN//IFM(JL)//IFM(3)//DOT3
      WRITE (8,LF) I,J,  IEQ,SIGN,  RO1,  I,J
          GO TO 15
C MINPOW = 1, minimum power per period desired
215   LF = HDOT//EN//IFM(IL)//HN//IFM(JL)//POW//IFM(IL)//IFM(4)//
    1       ' 5H OBJ ,F6.2,'//EP//IFM(IL)//HN//IFM(JL)//IFM(3)//DOT3
         WRITE (8,LF)  I,J,   I,  RO1, I,J
          GO TO 15
C MINPOW = 0, ENG needs to be written in OBJ and EP only
216   LF = HDOT//EN//IFM(IL)//HN//IFM(JL)//'5H OBJ ,F6.2, T28,'//EP
    1     //IFM(IL)//HN//IFM(JL)//IFM(3)//DOT2
         WRITE (8,LF)  I,J,   RO1,  I,J
15    CONTINUE
C  Once the ENG variable and ST variables for plants with non-constant
C  head have been written for every time step, the RHS values for power
C  plants with non-constant head will be written to b:RELEASE.DATA
         IF ( C(1,L) .GT. 0 ) CALL LFRHS( II, 0, EP, J, ERHS )
c       II = number of time steps in the current run
c        0 = code to write the node number
c       EP = character variable of the piece of format line
c            used to write
c            the beginning of the constraint name
c        J = node number of current node to include in the
c            constraint name
c ERHS(14) = a vector of the constant values of the RHS of the
c            constraints for every time step
16    CONTINUE
C  Write the variable QTB in CNTiNj, CNTiNj+m, UPB, LOW and EPiNn
C    constraints, according to whether the turbined flows are diverted
C    or go downstream
```

```
C  First diverted flow:
     DO 20 L=1,NTIMES
       J = NPOW(L+1)
C  Find LJ, the ordinal number corresponding to the power plant at J
C       for its datafiles
        DO 155 LJ = 1,NPOWER(1)
          IF ( NPOWER(LJ+1) .EQ. J ) GO TO 160
155      CONTINUE
C  No power plant found at that number, redo ordering of NPOW
        WRITE (*,1005) J
        CLOSE (8)
        GO TO 105
160      JL = 1
        IF ( J .GT. 9 ) JL=2
C Find the ordinal value, K, in the diversion vector, NDIV, where
C   the node, J, where the turbined flows are diverted, is represented
        DO 17 K=2, NDIV(1,1)+1
          IF ( NDIV(1,K) .EQ. J ) GO TO 19
17       CONTINUE
C  No diversion found at the power plant node
        WRITE (*,1004) NNAME(J)
        CLOSE (8)
        GO TO 105
C  The diversion TO node found, the QTB variable for the power plants
c    on diversions will be written for all its constraints
19       JD = NDIV(2,K)
        JP = 1
        IF ( JD .GT. 9 ) JP=2
      DO 21 I=1,II
        IL=1
        IF ( I .GT. 9 ) IL=2
        IF ( TMX(I,LJ) .GT. CDX(I,K-1) ) TMX(I,LJ) = CDX(I,K-1)
        IF ( TMN(I,LJ) .LT. CDN(I,K-1) ) TMN(I,LJ) = CDN(I,K-1)
        LF = HDOT//QT//IFM(IL)//HN//IFM(JL)//CNT//IFM(IL)//HN//IFM(JL)
     1       //'3H 1 ,'//CNT//IFM(IL)//HN//IFM(JP)//'F7.3,'//UPB//
     2       ' /10X,'//LOW//'T35,'//EP//IFM(IL)//HN//IFM(JL)//'F7.4,'
     3       //DOT5
      WRITE ( 8,LF )  I,J, I,J, I,JD, VL(I,K)*CHLSS(K)-1, TMX(I,LJ),
     1                   TMN(I,LJ),   I,J,H(I,LJ)
21    CONTINUE
C  Remove the node just written from the diversion vectors DIV(1,n), DIV(2,n)
        DO 195 L2 = K,NDIV(1,1)+1
              NDIV(1,L2) = NDIV(1,L2+1)
              NDIV(2,L2) = NDIV(2,L2+1)
               CHLSS(L2) = CHLSS(L2+1)
            DO 195 I = 1,II
               CDN(I,L2-1) = CDN(I,L2)
195             CDX(I,L2-1) = CDX(I,L2)
                VL(I,L2-1) = VL(I,L2)
          NDIV(1,1) = NDIV(1,1) - 1
20    CONTINUE
C  Second, downstream flow
```

```
      DO 25 L=NTIMES+2, NPOWER(1)+1
        J = NPOW(L)
C  Downstream flow stations send turbined flow to the node numbered one
C  higher, or if they are in the NNOSPL vector, the flows arrrive to
C  the node listed in the NTRB vector
        JP1 = J + 1
      DO 821 K=2,NNOSPL(1)
        IF ( J .EQ. NNOSPL(K) ) GO TO 822
821   CONTINUE
        GO TO 830
C  The flows from node J were found to NOT flow to the node J+1
822   DO 823 K=2,NTRB(1,1)
        IF ( J .EQ. NTRB(1,K) ) GO TO 824
823   CONTINUE
C  The node J is not on a tributary which flows to another node, write
C  an error message to the screen and stop.
      WRITE ( *,1006 ) NNAME(J)
      STOP
C  The node J is on a tributary which flows to the node NTRB(2,K)
824   JP1 = NTRB(2,K)
C  Find LJ, the ordinal number corresponding to the power plant at J
C       for its datafiles
830     DO 22 LJ = 1,NPOWER(1)
          IF ( NPOWER(LJ+1) .EQ. J ) GO TO 24
22      CONTINUE
C  No power plant found at that number, redo ordering of NPOW
        WRITE (*,1005) J
        CLOSE (8)
        GO TO 105
24      JL = 1
        IF ( J .GT. 9 ) JL=2
        JP = 1
        IF ( JP1 .GT. 9 ) JP=2
      DO 25 I=1,II
        IL=1
        IF ( I .GT. 9 ) IL=2
        LF = HDOT//QT//IFM(IL)//HN//IFM(JL)//CNT//IFM(IL)//HN//IFM(JL)
     1       //'2H 1,'//CNT//IFM(IL)//HN//IFM(JP)//IFM(3)//UPB//LOW//
     2       EP//IFM(IL)//HN//IFM(JL)//'F7.4,'//DOT5
       IF ( J .GE. 9 .OR. I .GT. 9 ) LF = HDOT//QT//IFM(IL)//HN//
     1                              IFM(JL)//CNT//IFM(IL)//HN//IFM(JL)
     2                              //'3H 1 ,'//CNT//IFM(IL)//HN//
     3                              IFM(JP)//IFM(3)//' / 10X,'//UPB//
     4                              LOW//EP//IFM(IL)//HN//IFM(JL)
     5                              //'F7.4,'//DOT5
       WRITE ( 8,LF )  I,J, I,J, I,JP1, TMX(I,LJ), TMN(I,LJ), I,J,
     1                                                   H(I,LJ)
25    CONTINUE
c
C  Write variables and equations of the points with irrigation diversion
       DO 30 L = 2, NIRR(1)+1
           J = NIRR(L)
```

```
        CALL WIRR ( J, II, L, NNN )
c     J = node number of the current node
c    II = number of times steps in the current run
c     L = irragation number of the current node
c   NNN = number of instances of irrigation return flow
30      CONTINUE
C  Write the CNT variables of natural flow
39     DO 40 L = 1, IN-1
          J = NSPILL(L)
         JL = 1
       IF ( J .GT. 9 ) JL = 2
         JP = 1
       IF ( J+1 .GT. 9 ) JP = 2
       DO 40 I=1,II
          IL = 1
         IF ( I .GT. 9 ) IL = 2
         LF = HDOT//QS//IFM(IL)//HN//IFM(JL)//CNT//IFM(IL)//HN//IFM(JL)
     1        //IFM(4)//CNT//IFM(IL)//HN//IFM(JP)//IFM(3)//DOT2
40        WRITE ( 8,LF )  I, J, I, J,  I, J+1
C  Write any upper or lower bounds of QS variables
       DO 45 I = 1, II
         IL = 1
         IF ( I .GT. 9 ) IL = 2
       DO 45 L = 1, NLIM(1) - 1
        LP1 = L + 1
         JL = 1
         IF ( NLIM(LP1) .GT. 9 ) JL = 2
         IF ( FMN(I,L) .GT. 0 ) GO TO 43
         IF ( FMX(I,L) .GT. 99998 ) GO TO 45
C  QS has only an upperbound, FMX(I,L)
         LF = HDOT//QS//IFM(IL)//HN//IFM(JL)//UPB//DOT1
         WRITE ( 8,LF )  I, NLIM(LP1), FMX(I,L)
          GO TO 45
C  QS has a lowerbound and may have an upperbound too
43        IF ( FMX(I,L) .LT. 99999 ) GO TO 44
C  QS has only a lowerbound, FMN(I,L)
         LF = HDOT//QS//IFM(IL)//HN//IFM(JL)//LOW//DOT1
         WRITE ( 8,LF )  I, NLIM(LP1), FMN(I,L)
          GO TO 45
C  QS  has both upper and lower bounds
44        LF = HDOT//QS//IFM(IL)//HN//IFM(JL)//LOW//UPB//DOT2
         WRITE ( 8,LF )  I, NLIM(LP1), FMN(I,L),      FMX(I,L)
45     CONTINUE
C  Write the CNT variables of tributary flows
       DO 50 L = 2, NTRB(1,1)+1
          J = NTRB(1,L)
         JL = 1
       IF ( J .GT. 9 ) JL = 2
         JP = 1
       IF ( NTRB(2,L) .GT. 9 ) JP = 2
       DO 50 I=1,II
        IL = 1
```

```
      IF ( I .GT. 9 ) IL = 2
        LF = HDOT//QS//IFM(IL)//HN//IFM(JL)//CNT//IFM(IL)//HN//IFM(JL)
     1         //IFM(4)//CNT//IFM(IL)//HN//IFM(JP)//IFM(3)//DOT2
50       WRITE ( 8,LF )  I, J, I, J,  I, NTRB(2,L)
C  Write the CNT variables of DIVERSION flows in an explicit loop,
C  necessary because of branching for double diversions from one node
          L = 2
C  Start of loop to write a diversion variable for each diversion L
505    IDOUBL = 0
          J = NDIV(1,L)
         LM = L - 1
         L1 = L
         JL = 1
      IF ( J .GT. 9 ) JL = 2
         JP = 1
      IF ( NDIV(2,L) .GT. 9 ) JP = 2
        I = 1
C  At some nodes there is more than one diversion.  If there are two
C  unique diversions from the same node, assign Q1D and Q2D as the
C  variable name
        DO 51 LL=L+1, NDIV(1,1)+1
        IF ( NDIV(1,LL) .NE. J ) GO TO 51
          IDOUBL = 1
             QD = '4H Q1D,'
        GO TO 515
51       CONTINUE
C  Loop to write a diversion variable for each period I
515          IL = 1
      IF ( I .GT. 9 ) IL = 2
             IUP = 0
      IF ( CDX(I,LM) .GE. 99999 ) IUP = 1
      IF ( CDN(I,LM) .LE. 0 ) IUP = IUP + 2
        GO TO ( 52, 54, 56 ) IUP
        LF = HDOT//QD//IFM(IL)//HN//IFM(JL)//CNT//IFM(IL)//HN//IFM(JL)
     1         //IFM(4)//CNT//IFM(IL)//HN//IFM(JP)//'F7.3,2X,'//UPB
     2         //LOW//DOT4
      IF ( I .GT. 9  .OR.  J .GT. 9 ) LF = HDOT//QD//IFM(IL)//HN//
     1                               IFM(JL)//CNT//IFM(IL)//HN//IFM(JL)
     2                               //IFM(4)//CNT//IFM(IL)//HN//IFM(JP)
     3                               //'F7.3,2X/10X,'//UPB//LOW//DOT4
      WRITE ( 8,LF ) I, J, I, J, I, NDIV(2,L1), VL(I,LM)*CHLSS(L1)-1,
     1                     CDX(I,LM), CDN(I,LM)
        GO TO 57
52       LF = HDOT//QD//IFM(IL)//HN//IFM(JL)//CNT//IFM(IL)//HN//IFM(JL)
     1       //IFM(4)//CNT//IFM(IL)//HN//IFM(JP)//'F7.3,2X,'//LOW//DOT3
       WRITE ( 8,LF ) I, J, I, J, I, NDIV(2,L1), VL(I,LM)*CHLSS(L1)-1,
     1                               CDN(I,LM)
        GO TO 57
54       LF = HDOT//QD//IFM(IL)//HN//IFM(JL)//CNT//IFM(IL)//HN//IFM(JL)
     1       //IFM(4)//CNT//IFM(IL)//HN//IFM(JP)//'F7.3,2X,'//UPB//DOT3
       WRITE ( 8,LF ) I, J, I, J, I, NDIV(2,L1), VL(I,LM)*CHLSS(L1)-1,
     1                              CDX(I,LM)
```

```
      GO TO 57
56     LF = HDOT//QD//IFM(IL)//HN//IFM(JL)//CNT//IFM(IL)//HN//IFM(JL)
    1      //IFM(4)//CNT//IFM(IL)//HN//IFM(JP)//'F7.3,2X,'//DOT2
      WRITE ( 8,LF ) I, J, I, J, I, NDIV(2,L1), VL(I,LM)*CHLSS(L1)-1
57     I = I + 1
      IF ( I .LE. II ) GO TO 515
C  End of I loop for each period
C  If there is a second diversion also write II variables, and delete
C  the second diversion from the list
      IF ( IDOUBL .LE. 0 ) GO TO 60
      IF ( IDOUBL .GT. 1 ) GO TO 58
          IDOUBL = 2
             QD = '4H Q2D,'
             L1 = LL
             LM = L1 - 1
              I = 1
             JP = 1
      IF ( NDIV(2,L) .GT. 9 ) JP = 2
        GO TO 515
C  Delete second diversion from the vectors it was in, so it won't be
C  written again
58      DO 59 L2 = LL,NDIV(1,1)+1
           NDIV(1,L2) = NDIV(1,L2+1)
           NDIV(2,L2) = NDIV(2,L2+1)
            CHLSS(L2) = CHLSS(L2+1)
         DO 59 I = 1,II
            CDN(I,L2-1) = CDN(I,L2)
59          CDX(I,L2-1) = CDX(I,L2)
             VL(I,L2-1) = VL(I,L2)
       QD = '4H  QD,'
       NDIV(1,1) = NDIV(1,1) - 1
60       L = L + 1
      IF ( L .LE. NDIV(1,1) + 1 ) GO TO 505
C  End of L loop for each diversion in project
C  Define CNT and list its RHS constants
      DO 300 N=1,NN
      CALL LFTYPE ( II, 10, EQ, ' 3HCNT,', N )
c      II = number of time steps in current run
c      10 = code for inclusion of node number in constraint name
c      EQ = type of constraint, in this case an equality
c ' 3HCNT,' = the piece of format line which will be used to write the
c            beginning of the constraint name in the data line
c       N = node number of the current node
300  CONTINUE
      DO 325 J=1,NN
       DO 324 I=1,II
324    FLOWN(I) = FLOW(I,J)
       CALL LFRHS ( II, 0, CNT, J, FLOWN )
c       II = number of time steps in the current run
c        0 = code to write the node number
c      CNT = character variable of the piece of format line
c             used to write the beginning of the constraint name
```

```
c         J = node number of current node to include in the
c             constraint name
c FLOWN(14) = a vector of the constant values of the RHS of the
c             constraints for every time step
325   CONTINUE
      IF ( MINPOW .LE. 0 ) GO TO 999
C  Write the lines for firmpower
      IF ( MINPOW .GT. 1 ) GO TO 350
C  MINPOW = 1, CALCULATE TOTAL POWER EACH PERIOD
      CALL LFTYPE ( II, 0, GE, POW, 0 )
c        II = number of time steps in current run
c         0 = code for a constraint name without a node number
c        GE = type of constraint, in this case greater than or equal
c       POW = the piece of format line which will be used to write the
c             constraint name in the data line, in a character variable
c         0 = no node number needed
      CALL LFRHS  ( II, 10, POW, 0, APOW )
c        II = number of time steps in the current run
c        10 = code not to write the node number in the constraint name
c       POW = character variable of the piece of format line
c             used to write the beginning of the constraint name
c         0 = no node number needed
c APOW(14) = a vector of the constant values of the RHS of the
c             constraints for every time step
      GO TO 999
C  MINPOW > 1, the difference in power production calculated each period
c             the PDFi constraints and the DIFi, PDFPi and PDFMi variables
350   CALL LFTYPE ( II-1, 0, EQ, PDF, 0 )
c      II-1 = one less than the number of time steps in current run
c         0 = code for a constraint name without a node number
c        EQ = type of constraint, in this case an equality
c       PDF = the piece of format line which will be used to write the
c             constraint name in the data line, in a character variable
c         0 = no node number needed
      CALL LFTYPE ( II-1, 0, EQ, ' 3H PP,', 0 )
c      II-1 = one less than the number of time steps in current run
c         0 = code for a constraint name without a node number
c        EQ = type of constraint, in this case an equality
c ' 3H PP,' = the piece of format line which will be used to write the
c             constraint name in the data line
c         0 = no node number needed
      DO 360 I=1,MIN( 9,II-1)
        WRITE (8,3000) I, I, I, I, I, I
      IF ( MINPOW .GT. 2 ) GO TO 355
        WRITE (8,3001) I, I, APOW(I)
      GO TO 360
c             if MINPOW = 3, DIFi variable has a cost in OBJ
355     WRITE ( 8,3002) I, I, -RO4
360   CONTINUE
      IF ( II-1 .LT. 10 ) GO TO 999
      DO 370 I=10,II-1
        WRITE (8,3005) I, I, I, I, I, I
```

```
      IF ( MINPOW .GT. 2 ) GO TO 365
        WRITE (8,3006) I, I, APOW(I)
      GO TO 370
c            if MINPOW = 3, DIFi variable has a cost in OBJ
365     WRITE ( 8,3007) I, I, -RO4
370   CONTINUE
999   WRITE (8,2005)
C
1000  FORMAT ( ' Hit the RETURN when all is set to write the file RELEAS
     1E.DATA on drive b:' )
1001  FORMAT ( ' Here is a list of power plants in the current configura
     1tion which are at nodes  with diversions.  Where do the turbine fl
     2ows go?'/' Count the number of nodes where the turbine flows are t
     3he diverted flows, such  as from Polgolla to Bowatenne.  ' )
1011  FORMAT ( ' The node ',A12,' diverts to the node ',A12 )
1021  FORMAT ( ' Enter the NUMBER of power plants whose turbine flows ar
     1e diverted  to  another  stream ' )
1002  FORMAT ( ' Here is a list of the power plants in the current confi
     1guration:'/  I5,2X,A12,10X,I5,2X,A12,10X,I5,2X,A12/ I5,2X,A12,10X,
     2I5,2X,A12,10X,I5,2X,A12/ I5,2X,A12,10X,I5,2X,A12,10X,I5,2X,A12/
     3I5,2X,A12,10X,I5,2X,A12,10X,I5,2X,A12 )
1003  FORMAT ( ' The ',I3,' power plant(s) which divert all their turbin
     1e flow MUST be first in  the list.  Enter the number to the left o
     2f a power plant which needs to be moved to the top of the list.  N
     3OTE that the numbers are not the node numbers.'              /  ' En
     4ter 0 or nothing if the list needs no changes. ' )
1004  FORMAT ( ' The power plant at ',A12,' has no diversion listed.  Pl
     1ease correct the list of power plants. '/ )
1005  FORMAT ( ' I was unable to find a power plant ',I3,'.  Please redo
     1 the list of power plants. '/ )
1006  FORMAT ( ' The current configuration lists no place for the turbin
     1ed flows of ',A12 )
2000    FORMAT  (  14H  DATA  RELEASE;/23H  INPUT  _TYPE_  $  _COL_
$/9X,50H_ROW1_
     1 $ _COEF1_ _ROW2_ $ _COEF2_ _ROW3_ $ _COEF3_/9X,51H_ROW4_ $
_COEF4
     2_ _ROW5_ $ _COEF5_ _ROW6_ $ _COEF6_; /7H CARDS; )
2001  FORMAT ( 4H MAX,2X,1H.,5H  OBJ,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,
     12X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H. )
2002  FORMAT ( 8H INTEGER,2X,1H.,5H  INT,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,
     11H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H. )
2003  FORMAT ( 8H LOWERBD,2X,1H.,5H  LOW,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,
     11H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H. )
2004  FORMAT ( 8H UPPERBD,2X,1H.,5H  UPB,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,
     11H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H. )
2005   FORMAT  ( 2H  ;/63HPROC  LP  SPARSEDATA  TIME=1920  IMAXIT=5000
MAXIT1=5
     100 MAXIT2=500;/2H/* )
3000  FORMAT ( 2H .,6H  PDFM,I1,2X,3HPDF,I1,5H  1  ,2HPP,I1,5H  1  ,2X,
     11H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H./ 2H .,
     26H  PDFP,I1,2X,3HPDF,I1,5H -1  ,2HPP,I1,5H  1  ,2X,1H.,2X,1H.,2X,
     31H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H. )
```

```
3001  FORMAT ( 2H .,5H  DIF,I1,2X,2HPP,I1,5H -1  ,3HUPB,F9.3,2X,1H.,2X,
     11H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H. )
3002  FORMAT ( 2H .,5H  DIF,I1,2X,2HPP,I1,5H -1  ,3HOBJ,F9.3,2X,1H.,2X,
     11H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H. )
3005  FORMAT ( 2H .,6H  PDFM,I2,2X,3HPDF,I2,5H  1  ,2HPP,I2,5H  1  ,2X,
     11H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H./ 2H .,
     26H  PDFP,I2,2X,3HPDF,I2,5H -1  ,2HPP,I2,5H  1  ,2X,1H.,2X,1H.,2X,
     31H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H. )
3006  FORMAT ( 2H .,5H  DIF,I2,2X,2HPP,I2,5H -1  ,3HUPB,F9.3,2X,1H.,2X,
     11H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H. )
3007  FORMAT ( 2H .,5H  DIF,I2,2X,2HPP,I2,5H -1  ,3HOBJ,F9.3,2X,1H.,2X,
     11H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H. )
9000  FORMAT ( A2 )
9002  FORMAT ( I5 )
9998  format ( A250 )
9999  format ( 3I20 )
      RETURN
      END
^Z
      SUBROUTINE WPOWER ( J, I, II, L )
c  J = node number of the current node
c  I = number of current time step
c II = number of time steps in the current run
c  L = power plant number in original NPOWER vector of the current node
c
C  This subroutine corrects the energy production calculation for nodes
c    where the volume stored significantly affects the head, by
c    including the ST variable in the EPiNn constraint.  Also the RHS
c    of the EPiNn equation is calculated and passed to DWRITE
      COMMON/SHR/ NPOWER(15), NIRR(21), NRES(25), NNOSPL(20), NDIV(2,15)
      COMMON/STG/ ST0(40), ST12(40), APOW(14), ERHS(14)
      COMMON/POW/ H(14,15), C(14,15), STO(14,15)
      CHARACTER * 250 LF
      CHARACTER * 75 DOT1
      CHARACTER * 60 DOT2
      CHARACTER * 10 HDOT
      CHARACTER * 7 HN, HS, HE, IFM(3)
      DATA DOT1/  ' 2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,
     11H.,2X,1H.,2X,1H. )'/
      DATA DOT2/' 2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H
     1. )  '/
      DATA HDOT/'( 2H .,1X,'/
      DATA HN /' 1HN,  '/,  HS/' 2HST, '/,  HE/' 2HEP, '/
      DATA IFM/' I1,    ',       ' I2,    ',       '  1H1, '/
C
      JL = 1
      IF ( J .GT. 9 ) JL = 2
C  Determine how many digits are required to write the time step indicator
      IL = 1
      IF ( I .GT. 9 ) IL = 2
      IF ( I .EQ. II ) GO TO 10
      IP = 1
```

```
      IF ( I .GT. 8 ) IP = 2
         LF = HDOT//HS//IFM(IL)//HN//IFM(JL)//'2X,'//HE//IFM(IL)//HN//
     1         IFM(JL)//'2X,F8.6,1X,'//HE//IFM(IP)//HN//IFM(JL)//
     2         '2X,F8.6,1X,'//DOT2
          WRITE ( 8,LF ) I,J, I,J,C(I,L)/2,  I+1,J,C(I+1,L)/2
             GO TO 30
C  The last period is one digit
10       LF = HDOT//HS//IFM(IL)//HN//IFM(JL)//'2X,'//HE//IFM(IL)//HN//
     1         IFM(JL)//'2X,F8.6,'//DOT1
          WRITE ( 8,LF ) I,J, I,J,C(I,L)/2
C  Calculation of the ERHS vector
30     CONTINUE
          IF ( I .GT. 1 ) GO TO 40
         ERHS(I) = C(1,L) * ( STO(1,L) - STO(J)/2 )
             GO TO 50
40       ERHS(I) = STO(I,L) * C(I,L)
50     CONTINUE
      RETURN
      END
C
      SUBROUTINE WRES ( J, II, L )
c  J = node number of current node
c II = number of time steps in current run
c  L = reservoir number of current node
C
C  This subroutine writes the lines defining the reservoir linearization
c  equations, their RHS constants, and the variables which appear in
c  them only.  It calls on a separate subroutine, LFII, to construct the
c  necessary format line according to the number of periods II, where
C  the line or lines are written to b: RELEASE.DATA
      COMMON/SHR/ JPOWER(15), JIRR(21), JRES(25), JNOSPL(20), JDIV(2,15)
      COMMON/RES/ GAMMA(25,9), DELTA(25,9), EPSI(25,9), KIN(25),
     1            DGAMA(25,9), CLOSS(14,25)
      COMMON/CAP/ TMAX(15), TMIN(15), SUP(25), SLO(25)
      COMMON/VAR/ VP(10,12)
c
      CHARACTER * 250 LF
      CHARACTER * 60 DOT2
      CHARACTER * 45 DOT3
      CHARACTER * 20 RHS, LE, GE, EQ
      CHARACTER * 18 DOT5
      CHARACTER * 15 UPB, LOW
      CHARACTER * 10 DOT1, ONE, F6, F10, HDOT
      CHARACTER * 4 HN, HY, HT, HR, HE, HK, IFM(2)
c
      DATA DOT3/' 2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H. ) '/
      DATA DOT2/' 2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H
     1. )  '/
      DATA RHS/' ( 2H .,6H _RHS_,1X,'/      EQ/' ( 3H EQ, 1X,1H.,1X,'/
      DATA LE/' ( 3H LE, 1X,1H.,1X,'/,      GE/' ( 3H GE, 1X,1H.,1X,'/
      DATA DOT5/' 2X,1H.,2X,1H. )'/
      DATA UPB/'4HUPB ,F5.1,2X,'/
```

```
      DATA DOT1/'1X,1H.,1X,'/,  ONE/'1X,1H1,1X,'/,  F10/' F10.4,1X,'/
      DATA HDOT/'( 2H .,1X,'/
      DATA HN/'1HN,'/,  HY/'1HY,'/,  HT/'1HT,'/,  HE/'1HE,'/, HK/'1HK,'/
      DATA HR/'1HR,'/,              IFM/' I1,', ' I2,'/
C
C  Calculate the ordinal number, JL  of the linearization data for
C  the reservoir J
         LJ = L - 1
      IF ( KIN(LJ) .LE. 1 ) GO TO 50
         K = 0
C  Write the equations defining the storage interval variables
C  RiNn and  YiNn, and the RHS line of the YiNn equations:
      CALL LFII ( J, II, 0, EQ, HR, 'NONE', DOT1, 0.0, K )
      CALL LFII ( J, II, 0, LE, HY, 'NONE', DOT1, 0.0, K )
      CALL LFII ( J, II, 0, RHS, HY, 'NONE', ONE, 0.0, K )
c      J = node number of current node
c     II = number of time steps in current run
c      0 = no real numbers after the constraint names needed
c EQ, LE or RHS = the starting characters of the format line
c HR or HR = character variable of the piece of format which will
c           be used to write the constraint name
c 'NONE' = code not to write linearization interval number
c   DOT1, or ONE = character variable which will be written
c                 after each constraint name
c    0.0 = code to write TYPE lines
c      K = interval number, in this case 0
C  The equation which define the limits of the storage intervals, YiNnEk
C  and guarantee that variables in only one interval appear at a time
      DO 10 K=1,KIN(LJ)
         CALL LFII ( J, II, 0, GE, HY, HE, DOT1, 0.0, K )
c      J = node number of current node
c     II = number of time steps in current run
c      0 = no real numbers after the constraint names needed
c     GE = the starting characters of the format line
c     HY = character variable of the piece of format which will
c          be used to write the constraint name
c     HE = the character variable for the piece of format line
c          which will write the interval marker
c   DOT1 = character variable which will be written
c          after each constraint name
c    0.0 = code to write TYPE lines
c      K = interval number
10    CONTINUE
C  Now write the variables specific to the linearization of the reservoir
c losses.
C  There is a separate line for each I, and every item in the line is
C  quite different.  Therefore, the LF is built in this program unit
      JL = 1
      IF ( J .GT. 9 ) JL = 2
      DO 20 I=1,MIN(9,II)
         LF = HDOT//HY//IFM(1)//HN//IFM(JL)//HK//IFM(1)//'2X,'//HY//
     1        IFM(1)//HN//IFM(JL)//HE//IFM(1)//'F7.1,1X,'//HY//IFM(1)//
```

```
     2        HN//IFM(JL)//' 3H 1 ,3HINT,3H 1 ,3HUPB,3H 1 ,4H CNT,'
     3        //IFM(1)//HN//IFM(JL)//'F8.4,'//DOT5
          WRITE ( 8,LF ) I,J,1, I,J,1, DGAMA(LJ,1),  I,J,
     1                       I,J, CLOSS(I,LJ)*EPSI(LJ,1)
          LF = HDOT//HT//IFM(1)//HN//IFM(JL)//HK//IFM(1)//'2X,'//HR//
     1        IFM(1)//HN//IFM(JL)//' 5H  1  ,4H CNT,'//IFM(1)//HN
     2        //IFM(JL)//F10//HY//IFM(1)//HN//IFM(JL)//HE//IFM(1)//
     3        ' 5H -1  ,'//DOT3
          WRITE ( 8,LF ) I,J,1, I,J, I,J, DELTA(LJ,1)*CLOSS(I,LJ), I,J,1
          DO 20 K=2,KIN(LJ)
            LF = HDOT//HY//IFM(1)//HN//IFM(JL)//HK//IFM(1)//'2X,'//HY//
     1          IFM(1)//HN//IFM(JL)//HE//IFM(1)//'F7.1,1X,'//HY//IFM(1)
     2          //HN//IFM(JL)//' 3H 1 ,3HINT,3H 1 ,3HUPB,3H 1 ,'//HR//
     1          IFM(1)//HN//IFM(JL)//'F7.1,1X,3HCNT,'//IFM(1)//HN//
     2          IFM(JL)//'F8.4 ) '
            IF ( JL .GE. 2 ) LF = HDOT//HY//IFM(1)//HN//IFM(2)//HK//
     1                        IFM(1)//'2X,'//HY//IFM(1)//HN//IFM(2)
     2                        //HE//IFM(1)//'F7.1,1X,'//HY//IFM(1)//
     3        HN//IFM(2)//' 3H 1 /10X,3HINT,3H 1 ,3HUPB,3H 1 ,'//HR//
     4                        IFM(1)//HN//IFM(2)//'F7.1,4H CNT,'//
     5                        IFM(1)//HN//IFM(2)//'F8.4 ) '
           WRITE ( 8,LF ) I,J,K, I,J,K, DGAMA(LJ,K), I,J,
     1                I,J, GAMMA(LJ,K-1),   I,J, CLOSS(I,LJ)*EPSI(LJ,K)
             LF = HDOT//HT//IFM(1)//HN//IFM(JL)//HK//IFM(1)//'2X,'//HR//
     1          IFM(1)//HN//IFM(JL)//' 5H  1  ,4H CNT,'//IFM(1)//HN
     2          //IFM(JL)//F10//HY//IFM(1)//HN//IFM(JL)//HE//IFM(1)//
     3          ' 5H -1  ,'//DOT3
20         WRITE ( 8,LF ) I,J,K, I,J, I,J, DELTA(LJ,K)*CLOSS(I,LJ), I,J,K
C  When there are more than nine timesteps the format is changed for
c  the greater index.
         IF ( II .LE. 9 ) GO TO 50
        DO 25 I=10,II
          LF = HDOT//HY//IFM(2)//HN//IFM(JL)//HK//IFM(1)//'2X,'//HY//
     1        IFM(2)//HN//IFM(JL)//HE//IFM(1)//'F7.1,1X,'//HY//IFM(2)//
     3        HN//IFM(JL)//' 3H 1 ,3HINT,3H 1 ,3HUPB,3H 1 ,4H CNT,'
     4        //IFM(2)//HN//IFM(JL)//'F8.4,'//DOT5
           IF ( JL .GE. 2 ) LF = HDOT//HY//IFM(2)//HN//IFM(2)//HK//
     1                        IFM(1)//'2X,'//HY//IFM(2)//HN//IFM(2)
     2                        //HE//IFM(1)//'F7.1,1X,'//HY//IFM(2)//
     3        HN//IFM(2)//' 3H 1 / 9X,3HINT,3H 1 ,3HUPB,3H 1 ,4H CNT,'//
     4                          IFM(2)//HN//IFM(2)//'F8.4,'//DOT5
          WRITE ( 8,LF ) I,J,1, I,J,1, DGAMA(LJ,1),  I,J,
     1                       I,J, CLOSS(I,LJ)*EPSI(LJ,1)
          LF = HDOT//HT//IFM(2)//HN//IFM(JL)//HK//IFM(1)//'2X,'//HR//
     1        IFM(2)//HN//IFM(JL)//' 5H  1  ,4H CNT,'//IFM(2)//HN
     2        //IFM(JL)//F10//HY//IFM(2)//HN//IFM(JL)//HE//IFM(1)//
     3        ' 5H -1  ,'//DOT3
          WRITE ( 8,LF ) I,J,1, I,J, I,J,DELTA(LJ,1)*CLOSS(I,LJ), I,J,K
        DO 25 K=2,KIN(LJ)
          LF = HDOT//HY//IFM(2)//HN//IFM(JL)//HK//IFM(1)//'2X,'//HY//
     1        IFM(2)//HN//IFM(JL)//HE//IFM(1)//'F7.1,1X,'//HY//IFM(2)//
     2        HN//IFM(JL)//' 3H 1 /10X,3HINT,3H 1 ,3HUPB,3H 1 ,4H CNT,'
```

```
3        //IFM(2)//HN//IFM(JL)//'F8.4,1X,'//HR//IFM(2)//HN//IFM(JL)
4        //'F7.1 ) '
     WRITE ( 8,LF ) I,J,K,  I,J,K, DGAMA(LJ,K), I,J,
1                    I,J, CLOSS(I,LJ)*EPSI(LJ,K), I,J, GAMMA(LJ,K-1)
     LF = HDOT//HT//IFM(2)//HN//IFM(JL)//HK//IFM(1)//'2X,'//HR//
1        IFM(2)//HN//IFM(JL)//' 5H  1   ,4H CNT,'//IFM(2)//HN
2        //IFM(JL)//F10//HY//IFM(2)//HN//IFM(JL)//HE//IFM(1)//
3           `' 5H -1  ,'//DOT3
25      WRITE ( 8,LF ) I,J,K, I,J, I,J, DELTA(LJ,K)*CLOSS(I,LJ), I,J,K
C If the seepage and evaporation loss function is estimated as a straight
C line,  i.e. the losses = CLOSS(i,j) * cste factor * STiNn, this
C subroutine doesn't write anything.
50    CONTINUE
      RETURN
      END
C
      SUBROUTINE WIRR ( J, II, L, IRT )
c    J = node number of the current node
c   II = number of times steps in the current run
c    L = irragation number of the current node
c  IRT = number of instances of irrigation return flow
C
C  This subroutine writes the lines defining the irrigation linearization
c  equations, their RHS constants, and the variables which appear in
c  them only.  Because the linearization curves might have only one
c  interval, i.e. the loss function is a constant value times any size
c  deficit, eliminating the need for any linearization for the time period
C  the single interval curves apply, a special loop writes the RHS of XiNn,
C  the TYPE lines for XiNn and WiNn to b: RELEASE.DATA when KINI(i,j) > 1
      COMMON/SHR/ JPOWER(15), JIRR(21), JRES(25), JNOSPL(20), JDIV(2,15)
      COMMON/SHP/ NINFLO(30), NTRB(2,10), NRF(3,10), C1(26,10)
      COMMON/VAR/ VP(14,10),VLS(14,25),VMS(14,25),CIX(14,20),CIN(14,20)
      COMMON/IRR/ THETA(6,21,9), OMEGA(6,21,9), PHI(6,21,9), ID(6,21),
     1        KINI(6,21), DTHET(6,21,9), IC(21), CWR(26,21), IID(14,21)
c
      DIMENSION IS(6), I1(6), K1(6), CWRN(14)
      CHARACTER * 250 LF
      CHARACTER * 75 DOTS1
      CHARACTER * 60 DOTS2
      CHARACTER * 45 DOTS3
      CHARACTER * 35 DOTS4
      CHARACTER * 20 RHS, LE, GE, EQ
      CHARACTER * 18 DOTS5
      CHARACTER * 15 UPB, LOW
      CHARACTER * 10 DOT1, ONE, F6, F10, HDOT
      CHARACTER * 4 HN, HX, HD, HW, HE, HK, IFM(2)
c
      DATA DOTS1/ ' 2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,
     11H.,2X,1H.,2X,1H. )'/
      DATA DOTS2/' 2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1
     1H. )  '/
      DATA DOTS3/' 2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H. )'/
```

```
      DATA DOTS4/' 2X,1H.,2X,1H.,2X,1H.,2X,1H. )'/
      DATA DOTS5/' 2X,1H.,2X,1H. )'/
c
      DATA RHS/' ( 2H .,6H _RHS_,1X,'/      EQ/' ( 3H EQ, 1X,1H.,1X,'/
      DATA LE/' ( 3H LE, 1X,1H.,1X,'/,      GE/' ( 3H GE, 1X,1H.,1X,'/
      DATA UPB/'4HUPB ,F5.1,1X,'/          LOW/'4HLOW ,F5.1,1X,'/
      DATA DOT1/'1X,1H.,1X,'/,  ONE/'1X,1H1,1X,'/,  F10/' F10.3,1X,'/
      DATA HDOT/'( 2H .,1X,'/
      DATA HN/'1HN,'/,  HX/'1HX,'/,  HD/'1HD,'/,  HE/'1HE,'/, HK/'1HK,'/
      DATA HW/'1HW,'/,                 IFM/' I1,', ' I2,'/
C
C  Calculate the ordinal number, LJ  of the linearization data for
C  the district J
         LJ = L - 1
         JL = 1
         IF ( J .GT. 9 ) JL = 2
C  Small loop to print three lines defining the RHS, TYPE of WiNn and
C    XiNn for curves with more than one linearization interval,
C      6 at a time
        N = 0
        I = 0
60        I = I + 1
       IF ( I .GT. II ) GO TO ( 71, 72, 73, 74, 75 ) N
       IF ( I .GT. II .AND. N .LE. 0 ) GO TO 62
       IF ( KINI(IID(I,LJ),LJ) .LE. 1 ) GO TO 60
          N = N + 1
          IS(N) = I
          I1(N) = 1
      IF ( IS(N) .GT. 9 ) I1(N) = 2
      IF ( N .LT. 6 ) GO TO 60
        LF = RHS//HX//IFM(I1(1))//HN//IFM(JL)//ONE//HX//IFM(I1(2))//
     1        HN//IFM(JL)//ONE//HX//IFM(I1(3))//HN//IFM(JL)//ONE//
     2        HX//IFM(I1(4))//HN//IFM(JL)//ONE//HX//
     3        IFM(I1(5))//HN//IFM(JL)//ONE//HX//IFM(I1(6))//HN//
     4        IFM(JL)//ONE//'1H ) '
      WRITE ( 8,LF ) ( IS(L), J, M=1,6 )
        LF = LE//HX//IFM(I1(1))//HN//IFM(JL)//DOT1//HX//IFM(I1(2))//
     1        HN//IFM(JL)//DOT1//HX//IFM(I1(3))//HN//IFM(JL)//DOT1//
     2        HX//IFM(I1(4))//HN//IFM(JL)//DOT1//HX//
     3        IFM(I1(5))//HN//IFM(JL)//DOT1//HX//IFM(I1(6))//HN//
     4        IFM(JL)//DOT1//'1H ) '
      WRITE ( 8,LF ) ( IS(L), J, M=1,6 )
        LF = EQ//HW//IFM(I1(1))//HN//IFM(JL)//DOT1//HW//IFM(I1(2))//
     1        HN//IFM(JL)//DOT1//HW//IFM(I1(3))//HN//IFM(JL)//DOT1//
     2        HW//IFM(I1(4))//HN//IFM(JL)//DOT1//HW//
     3        IFM(I1(5))//HN//IFM(JL)//DOT1//HW//IFM(I1(6))//HN//
     4        IFM(JL)//DOT1//'1H ) '
      WRITE ( 8,LF ) ( IS(L), J, M=1,6 )
           N = 0
         GO TO 60
C  End of loop - at this point I=II and N number of lines
c               have yet to be written.  IS(n), and I1(n) are
```

```
C                 ready to write them
C One RHS constraint to go
71     LF = RHS//HX//IFM(I1(1))//HN//IFM(JL)//ONE//DOTS1
       WRITE ( 8,LF ) ( IS(L), J, M=1,N )
       LF = LE//HX//IFM(I1(1))//HN//IFM(JL)//DOT1//DOTS1
       WRITE ( 8,LF ) ( IS(L), J, M=1,N )
       LF = EQ//HW//IFM(I1(1))//HN//IFM(JL)//DOT1//DOTS1
          GO TO 77
C Two
72     LF = RHS//HX//IFM(I1(1))//HN//IFM(JL)//ONE//HX//IFM(I1(2))//
     1      HN//IFM(JL)//ONE//DOTS2
       WRITE ( 8,LF ) ( IS(L), J, M=1,N )
       LF = LE//HX//IFM(I1(1))//HN//IFM(JL)//DOT1//HX//IFM(I1(2))//
     1      HN//IFM(JL)//DOT1//DOTS2
       WRITE ( 8,LF ) ( IS(L), J, M=1,N )
       LF = EQ//HW//IFM(I1(1))//HN//IFM(JL)//DOT1//HW//IFM(I1(2))//
     1      HN//IFM(JL)//DOT1//DOTS2
          GO TO 77
73     LF = RHS//HX//IFM(I1(1))//HN//IFM(JL)//ONE//HX//IFM(I1(2))//
     1      HN//IFM(JL)//ONE//HX//IFM(I1(3))//HN//IFM(JL)//ONE//DOTS3
       WRITE ( 8,LF ) ( IS(L), J, M=1,N )
       LF = LE//HX//IFM(I1(1))//HN//IFM(JL)//DOT1//HX//IFM(I1(2))//
     1      HN//IFM(JL)//DOT1//HX//IFM(I1(3))//HN//IFM(JL)//DOT1//DOTS3
       WRITE ( 8,LF ) ( IS(L), J, M=1,N )
       LF = EQ//HW//IFM(I1(1))//HN//IFM(JL)//DOT1//HW//IFM(I1(2))//
     1      HN//IFM(JL)//DOT1//HW//IFM(I1(3))//HN//IFM(JL)//DOT1//DOTS3
       WRITE ( 8,LF ) ( IS(L), J, M=1,N )
          GO TO 77
74     LF = RHS//HX//IFM(I1(1))//HN//IFM(JL)//ONE//HX//IFM(I1(2))//
     1      HN//IFM(JL)//ONE//HX//IFM(I1(3))//HN//IFM(JL)//ONE//
     2      HX//IFM(I1(4))//HN//IFM(JL)//ONE//DOTS4
       WRITE ( 8,LF ) ( IS(L), J, M=1,N )
       LF = LE//HX//IFM(I1(1))//HN//IFM(JL)//DOT1//HX//IFM(I1(2))//
     1      HN//IFM(JL)//DOT1//HX//IFM(I1(3))//HN//IFM(JL)//DOT1//
     2      HX//IFM(I1(4))//HN//IFM(JL)//DOT1//DOTS4
       WRITE ( 8,LF ) ( IS(L), J, M=1,N )
       LF = EQ//HW//IFM(I1(1))//HN//IFM(JL)//DOT1//HW//IFM(I1(2))//
     1      HN//IFM(JL)//DOT1//HW//IFM(I1(3))//HN//IFM(JL)//DOT1//
     2      HW//IFM(I1(4))//HN//IFM(JL)//DOT1//DOTS4
       WRITE ( 8,LF ) ( IS(L), J, M=1,N )
          GO TO 77
75     LF = RHS//HX//IFM(I1(1))//HN//IFM(JL)//ONE//HX//IFM(I1(2))//
     1      HN//IFM(JL)//ONE//HX//IFM(I1(3))//HN//IFM(JL)//ONE//
     2      HX//IFM(I1(4))//HN//IFM(JL)//ONE//
     3      HX//IFM(I1(5))//HN//IFM(JL)//ONE//DOTS5
      WRITE ( 8,LF ) ( IS(L), J, M=1,N )
       LF = LE//HX//IFM(I1(1))//HN//IFM(JL)//DOT1//HX//IFM(I1(2))//
     1      HN//IFM(JL)//DOT1//HX//IFM(I1(3))//HN//IFM(JL)//DOT1//
     2      HX//IFM(I1(4))//HN//IFM(JL)//DOT1//
     3      HX//IFM(I1(5))//HN//IFM(JL)//DOT1//DOTS5
      WRITE ( 8,LF ) ( IS(L), J, M=1,N )
       LF = EQ//HW//IFM(I1(1))//HN//IFM(JL)//DOT1//HW//IFM(I1(2))//
```

```
     1        HN//IFM(JL)//DOT1//HW//IFM(I1(3))//HN//IFM(JL)//DOT1//
     2        HW//IFM(I1(4))//HN//IFM(JL)//DOT1//
     3        HW//IFM(I1(5))//HN//IFM(JL)//DOT1//DOTS5
77    WRITE ( 8,LF ) ( IS(L), J, M=1,N )
C  Because the number of intervals vary with the curve used for that
c  timestep a different loop is used to write the DiNnEk constraints
C  Small loop to print one line defining the DiNnEk constraints, 6 at a
c  time
62        N = 0
          I = 0
65           I = I + 1
             K = 0
70            K = K + 1
       IF ( I .GT. II )  GO TO 80
        KINIJ = KINI(IID(I,LJ),LJ)
       IF ( K .GT. KINIJ  .OR.  KINIJ .LE. 1 ) GO TO 65
             N = N + 1
             K1(N) = K
             I1(N) = I
             IS(N) = 1
             IF ( I .GT. 9 ) IS(N) = 2
             IF ( N .LT. 6 ) GO TO 70
        LF = GE//HD//IFM(IS(1))//HN//IFM(JL)//HE//IFM(1)//DOT1
     1          //HD//IFM(IS(2))//HN//IFM(JL)//HE//IFM(1)//DOT1
     2          //HD//IFM(IS(3))//HN//IFM(JL)//HE//IFM(1)//DOT1
     3          //HD//IFM(IS(4))//HN//IFM(JL)//HE//IFM(1)//DOT1
     4          //HD//IFM(IS(5))//HN//IFM(JL)//HE//IFM(1)//DOT1
     5          //HD//IFM(IS(6))//HN//IFM(JL)//HE//IFM(1)//DOT1//'1H ) '
      WRITE ( 8,LF ) ( I1(M), J, K1(M), M=1,6 )
              N = 0
          GO TO 70
80     GO TO ( 81, 82, 83, 84, 85 ) N
        GO TO 90
C  End of loop - at this point I=II, K=last k of last curve and N number
c                of DiNnEk have yet to be written
C                though K1(n), I1(n), and IS(n) are ready to write them
C One DiNnEk to go
81      LF = GE//HD//IFM(IS(1))//HN//IFM(JL)//HE//IFM(1)//DOT1//DOTS1
           GO TO 87
C Two
82      LF = GE//HD//IFM(IS(1))//HN//IFM(JL)//HE//IFM(1)//DOT1
     1         //HD//IFM(IS(2))//HN//IFM(JL)//HE//IFM(1)//DOT1//DOTS2
           GO TO 87
83     LF = GE//HD//IFM(IS(1))//HN//IFM(JL)//HE//IFM(1)//DOT1
     1        //HD//IFM(IS(2))//HN//IFM(JL)//HE//IFM(1)//DOT1
     2        //HD//IFM(IS(3))//HN//IFM(JL)//HE//IFM(1)//DOT1//DOTS3
```

```
85     LF = GE//HD//IFM(IS(1))//HN//IFM(JL)//HE//IFM(1)//DOT1
    1       //HD//IFM(IS(2))//HN//IFM(JL)//HE//IFM(1)//DOT1
    2       //HD//IFM(IS(3))//HN//IFM(JL)//HE//IFM(1)//DOT1
    3       //HD//IFM(IS(4))//HN//IFM(JL)//HE//IFM(1)//DOT1
    4       //HD//IFM(IS(5))//HN//IFM(JL)//HE//IFM(1)//DOT1//DOTS5
87     WRITE ( 8,LF ) ( I1(M), J, K1(M), M=1,N )
C
C  Now write the variables specific to the linearization of the IRRIGATION
c losses.  Also included are QDF, the irrigation deficit, and QR, the
c irrigation turnout.
C  There is a separate line for each I, and every item in the line is
C  quite different.  Therefore, the LF is built in this program unit
90     DO 20 I=1,MIN(9,II)
C  If a percentage of QR returns to another node, include it in CNT
         DO 11  K=1, IRT
           IF ( NRF(1,K) .EQ. J .AND. NRF(3,K)+I .LE. II ) GO TO 12
11        CONTINUE
         LF = HDOT//'3H QR,'//IFM(1)//HN//IFM(JL)//'4X,3HIRR,'//IFM(1)
    1     //HN//IFM(JL)//ONE//'3HCNT,'//IFM(1)//HN//IFM(JL)//ONE//DOTS2
         WRITE ( 8,LF ) I,J,  I,J, I,J
         GO TO 15
12        IP = 1
         IF ( I+NRF(3,K) .GT. 9 ) IP = 2
         JP = 1
         IF ( NRF(2,K) .GT. 9 ) JP = 2
         LF = HDOT//'3H QR,'//IFM(1)//HN//IFM(JL)//'4X,3HIRR,'//IFM(1)
    1       //HN//IFM(JL)//ONE//'3HCNT,'//IFM(1)//HN//IFM(JL)//ONE//
    2       ' 3HCNT,'//IFM(IP)//HN//IFM(JP)//F10//DOTS3
         WRITE ( 8,LF ) I,J,  I,J,   I,J, I+NRF(3,K), NRF(2,K), -C1(I,K)
C  The irrigation variable QDF
15        IF ( KINI(IID(I,LJ),LJ) .LE. 1 ) GO TO 19
         LF = HDOT//'3HQDF,'//IFM(1)//HN//IFM(JL)//'4X,3HIRR,'//IFM(1)
    1       //HN//IFM(JL)//' 5H  1  ,'//HW//IFM(1)//HN//IFM(JL)//
    2       ' 5H -1  ,'//DOTS2
         WRITE ( 8,LF )  I, J,  I, J,  I, J
         LF = HDOT//HX//IFM(1)//HN//IFM(JL)//HK//IFM(1)//'2X,'//HD//
    1       IFM(1)//HN//IFM(JL)//HE//IFM(1)//'F7.1,2X,'//HX//IFM(1)//
    2       HN//IFM(JL)//' 3H 1 ,3HINT,3H 1 ,3HUPB,3H 1 ,'//DOTS4
         WRITE ( 8,LF ) I,J,1, I,J,1, DTHET(IID(I,LJ),LJ,1),  I,J
         LF = HDOT//HD//IFM(1)//HN//IFM(JL)//HK//IFM(1)//'2X,'//HW//
    1       IFM(1)//HN//IFM(JL)//' 5H  1  ,4H OBJ,'//F10//HD//IFM(1)
    2       //HN//IFM(JL)//HE//IFM(1)//'4H -1 ,'//DOTS3
         WRITE ( 8,LF ) I,J,1,   I,J,  - OMEGA(IID(I,LJ),LJ,1), I,J,1
         DO 18 K=2,KINI(IID(I,LJ),LJ)
           LF = HDOT//HX//IFM(1)//HN//IFM(JL)//HK//IFM(1)//'2X,'//HD//
    1         IFM(1)//HN//IFM(JL)//HE//IFM(1)//'F7.1,2X,'//HX//IFM(1)
    2         //HN//IFM(JL)//' 3H 1 ,3HINT,3H 1 ,3HUPB,3H 1 ,'//HW//
    3         IFM(1)//HN//IFM(JL)//'F7.1,2X,3HOBJ,F7.1 ) '
          WRITE ( 8,LF ) I,J,K, I,J,K, DTHET(IID(I,LJ),LJ,K), I,J, I,J,
    1              THETA(IID(I,LJ),LJ,K-1), - PHI(IID(I,LJ),LJ,K-1)
          LF = HDOT//HD//IFM(1)//HN//IFM(JL)//HK//IFM(1)//'2X,'//HW//
    1          IFM(1)//HN//IFM(JL)//' 5H  1  ,4H OBJ,'//F10//HD//
```

```
     2            IFM(1)//HN//IFM(JL)//HE//IFM(1)//' 5H -1   ,'//DOTS3
        WRITE ( 8,LF ) I,J,K, I,J, - OMEGA(IID(I,LJ),LJ,K), I,J,K
18     CONTINUE
           GO TO 20
19       IF ( OMEGA(IID(I,LJ),LJ,1) .GT. 0.0001 ) GO TO 195
C  There is only one interval of linearization without any costs
        LF = HDOT//'3HQDF,'//IFM(1)//HN//IFM(JL)//'4X,3HIRR,'//IFM(1)
     1       //HN//IFM(JL)//' 5H  1    ,'//DOTS1
        WRITE ( 8,LF )  I, J,  I, J
         GO TO 20
C  There is only one interval of linearization, and there is a cost > 0.
195      LF = HDOT//'3HQDF,'//IFM(1)//HN//IFM(JL)//'4X,3HIRR,'//IFM(1)
     1       //HN//IFM(JL)//' 5H  1    ,4HOBJ ,'//F10//DOTS2
        WRITE ( 8,LF )  I, J,  I, J,  - OMEGA(IID(I,LJ),LJ,1)
20    CONTINUE
       IF ( II .LE. 9 ) GO TO 31
C  When there are more than nine timesteps the format is changed for
c  the greater index.
      DO 30 I=10,II
C  If a percentage of QR returns to another node, include it in CNT
        DO 22  K=1, IRT
         IF ( NRF(1,K) .EQ. J .AND. NRF(3,K)+I .LE. II ) GO TO 23
22       CONTINUE
        LF = HDOT//'3H QR,'//IFM(2)//HN//IFM(JL)//'4X,3HIRR,'//IFM(2)
     1       //HN//IFM(JL)//ONE//'3HCNT,'//IFM(2)//HN//IFM(JL)//ONE//
     2       DOTS2
        WRITE ( 8,LF ) I,J,  I,J,  I,J
        GO TO 25
23       JP = 1
        IF ( NRF(2,K) .GT. 9 ) JP = 2
        LF = HDOT//'3H QR,'//IFM(2)//HN//IFM(JL)//'4X,3HIRR,'//IFM(2)
     1       //HN//IFM(JL)//ONE//'3HCNT,'//IFM(2)//HN//IFM(JL)//ONE//
     2       ' 3HCNT,'//IFM(2)//HN//IFM(JP)//F10//DOTS3
        WRITE ( 8,LF ) I,J,  I,J,  I,J,  I+NRF(3,K), NRF(2,K), -C1(I,K)
25       IF ( KINI(IID(I,LJ),LJ) .LE. 1 )  GO TO 29
C  The irrigation variables QDF
        LF = HDOT//'3HQDF,'//IFM(2)//HN//IFM(JL)//'4X,3HIRR,'//IFM(2)
     1       //HN//IFM(JL)//' 5H  1   ,'//HW//IFM(2)//HN//IFM(JL)//
     2       ' 5H -1   ,'//DOTS2
        WRITE ( 8,LF )  I, J,  I, J,  I, J
        LF = HDOT//HX//IFM(2)//HN//IFM(JL)//HK//IFM(1)//'2X,'//HD//
     1       IFM(2)//HN//IFM(JL)//HE//IFM(1)//'F7.1,2X,'//HX//IFM(2)//
     2       HN//IFM(JL)//' 3H 1 ,3HINT,3H 1 ,3HUPB,3H 1 ,'//DOTS4
        WRITE ( 8,LF ) I,J,1, I,J,1, DTHET(IID(I,LJ),LJ,1),  I,J
        LF = HDOT//HD//IFM(2)//HN//IFM(JL)//HK//IFM(1)//'2X,'//HW//
     1       IFM(2)//HN//IFM(JL)//' 5H  1   ,4H OBJ,'//F10//HD//IFM(2)
     2       //HN//IFM(JL)//HE//IFM(1)//'4H -1 ,'//DOTS3
        WRITE ( 8,LF ) I,J,1,   I,J,  - OMEGA(IID(I,LJ),LJ,1), I,J,1
        DO 28 K=2,KINI(IID(I,LJ),LJ)
          LF = HDOT//HX//IFM(2)//HN//IFM(JL)//HK//IFM(1)//'2X,'//HD//
     1          IFM(2)//HN//IFM(JL)//HE//IFM(1)//'F7.1,2X,'//HX//IFM(2)
     2          //HN//IFM(JL)//' 3H 1 ,3HINT,3H 1 ,3HUPB,3H 1 ,'//HW//
```

```
    3          IFM(2)//HN//IFM(JL)//'F7.1,2X,3HOBJ,F7.1 ) '
        WRITE ( 8,LF ) I,J,K, I,J,K, DTHET(IID(I,LJ),LJ,K), I,J, I,J,
    1              THETA(IID(I,LJ),LJ,K-1), - PHI(IID(I,LJ),LJ,K-1)
          LF = HDOT//HD//IFM(2)//HN//IFM(JL)//HK//IFM(1)//'2X,'//HW//
    1          IFM(2)//HN//IFM(JL)//' 5H  1  ,4H OBJ,'//F10//HD//
    2          IFM(2)//HN//IFM(JL)//HE//IFM(1)//' 5H -1  ,'//DOTS3
       WRITE ( 8,LF ) I,J,K, I,J, - OMEGA(IID(I,LJ),LJ,K), I,J,K
28   CONTINUE
             GO TO 30
29      IF ( OMEGA(IID(I,LJ),LJ,1) .GT. 0.0001 ) GO TO 295
C  There is only one interval of linearization, and there is NO cost
       LF = HDOT//'3HQDF,'//IFM(2)//HN//IFM(JL)//'4X,3HIRR,'//IFM(2)
    1      //HN//IFM(JL)//' 5H  1  ,'//DOTS1
       WRITE ( 8,LF )  I, J,  I, J
             GO TO 30
C  There is only one interval of linearization, and there is a cost > 0.
295     LF = HDOT//'3HQDF,'//IFM(2)//HN//IFM(JL)//'4X,3HIRR,'//IFM(2)
    1      //HN//IFM(JL)//' 5H  1  ,4HOBJ ,'//F10//DOTS2
       WRITE ( 8,LF )  I, J,  I, J,   - OMEGA(IID(I,LJ),LJ,1)
30   CONTINUE
C
C  Definition of IRRiNn constraints and RHS
31   CALL LFTYPE ( II, 10,' ( 3H EQ, 1X,1H.,1X,',  ' 3HIRR,', J )
c       II = number of time steps in current run
c       10 = code for inclusion of node number in constraint name
c ' ( 3H EQ, 1X,1H.,1X,' = the piece of format line which will write
c                        the type of the constraint, in this case
c                        an equality
c ' 3HIRR,' = the piece of format line which will be used to write the
c              constraint name in the data line
c        J = node number of the current node
     DO 33 I=1,II
33   CWRN(I) = CWR(I,LJ)
     CALL LFRHS  ( II, 0, '4H IRR,', J, CWRN )
c       II = number of time steps in the current run
c        0 = code to write the node number
c '4H IRR,' = character variable of the piece of format line
c              used to write the beginning of the constraint name
c        J = node number of current node to include in the
c              constraint name
c  CWRN(14) = a vector of the constant values of the RHS of the
c              constraints for every time step
     RETURN
     END
^Z
     SUBROUTINE LFII ( J, IT, NUM, FIRST, RPEATR, INTVAL, FORM, R1, K )
c      J = node number of current node
c     IT = number of time steps in current run
c    NUM = code for writing real number values after the constraints
c  FIRST = the starting characters of the format line
c RPEATR = character variable of the piece of format which will
c          be used to write the constraint name
```

```
c INTVAL = code for use of linearization interval number
c   FORM = character variable which will be written
c          after each constraint name
c     R1 = real number value to write after every constraint name
c          using the format of FORM
c      K = interval number
c
C This subroutine constructs a format line of the form:
c  FIRST  +  ii * ( RPEATRiNjINTk + REAL VALUE )  +  END
C   where i is a time step i less than or equal to ii, the total number
C         j is the node in question, J
C       INT is an otional interval marker
C         k is the interval in question, K
C   Real Value, R1, is written using the format FORM
C                   FORM may contain a constant value )
C       END is the format of necessary dots to fill the line
C  The line is then written to b:RELEASE.DATA
      CHARACTER * 250 LF
      CHARACTER * 75 DOT1
      CHARACTER * 60 DOT2
      CHARACTER * 45 DOT3
      CHARACTER * 31 DOT4
      CHARACTER * 20 FIRST
      CHARACTER * 17 DOT5
      CHARACTER * 10 FORM
      CHARACTER * 7 NAME
      CHARACTER * 6 DOT6
      CHARACTER * 4  RPEATR, INTVAL, IFM(2), HN
c
      DATA DOT1/' 2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H
     1.,2X,1H.,2X,1H. )'/
      DATA DOT2/' 2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H
     1. ) '/
      DATA DOT3/' 2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H. ) '/
      DATA DOT4/' 2X,1H.,2X,1H.,2X,1H.,2X,1H. ) '/
      DATA DOT5/' 2X,1H.,2X,1H. ) '/
      DATA DOT6/' 1H ) '/
      DATA IFM/ ' I1,',  'I2,'/,  HN/'1HN,'/
C
      II = IT
      IS = 1
      JL = 1
      IF ( J .GT. 9 ) JL = 2
      IF ( INTVAL .EQ. 'NONE' ) GO TO 100
C There are intervals in the repeated name
          GO TO ( 10, 20, 30, 40, 50, 60, 10, 20, 30, 40, 50, 60 ) II
C II is equal to 1, 7, or 13 or more
10    LF = FIRST//RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//
     1     DOT1
      IF ( NUM .EQ. 0 ) GO TO 15
      WRITE ( 8,LF ) 1,J,K, R1
      IF ( II .LE. 1 ) GO TO 999
```

```
      IS = 2
      GO TO 60
c  write format line without a real value, ie TYPE lines
15    WRITE ( 8,LF ) 1,J,K
      IF ( II .LE. 1 ) GO TO 999
      IS = 2
      GO TO 60
C  II is equal to 2 or 8
20    LF = FIRST//RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//
     1   RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//DOT2
      IF ( NUM .EQ. 0 ) GO TO 25
      WRITE ( 8,LF ) ( I,J,K, R1, I=1,2 )
      IF ( II .LE. 2 )  GO TO 999
      IS = 3
      GO TO 60
25    WRITE ( 8,LF ) ( I,J,K, I=1,2 )
      IF ( II .LE. 2 ) GO TO 999
      IS = 3
      GO TO 60
C  II is equal to 3 or 9
30    LF = FIRST//RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//
     1    RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//
     2    RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//DOT3
      IF ( NUM .EQ. 0 ) GO TO 35
      WRITE ( 8,LF ) ( I,J,K, R1, I=1,3 )
      IF ( II .LE. 3 ) GO TO 999
      IS = 4
      GO TO 60
c  write format line without a real value, ie TYPE lines
35    WRITE ( 8,LF ) ( I,J,K, I=1,3 )
      IF ( II .LE. 3 ) GO TO 999
      IS = 4
      GO TO 60
C  II is equal to 4 or 10
40    LF = FIRST//RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//
     1    RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//
     2    RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//
     3    RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//DOT4
      IF ( NUM .EQ. 0 ) GO TO 45
      LF = FIRST//RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//
     1    RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//'/ 10X,'//
     2    RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//
     3    RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//DOT4
      WRITE ( 8,LF ) ( I,J,K, R1, I=1,4 )
      IF ( II .LE. 4 ) GO TO 999
      IS = 5
      GO TO 60
c  write format line without a real value, ie TYPE lines
45    WRITE ( 8,LF ) ( I,J,K, I=1,4 )
      IF ( II .LE. 4 ) GO TO 999
      IS = 5
      GO TO 60
```

```
C  II is equal to 5 or 11, use TWO lines, when there are numbers
50    IF ( NUM .EQ. 0 ) GO TO 55
      LF = FIRST//RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//
     1     RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//
     2     RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//'/ 10X,'//
     3     RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//
     4     FORM//RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM
     5     //DOT5
      WRITE ( 8,LF ) ( I,J,K, R1, I=1,5 )
      IF ( II .LE. 5 ) GO TO 999
        IS = 6
        GO TO 60
C  NUM equals 0, therefore there are no real numbers and 5 items
C  may be written on one line
55    LF = FIRST//RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//
     1     FORM//RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//
     2     FORM//RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//
     3     FORM//RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//
     4     FORM//RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//
     5     FORM//' 2X,1H.,2X,1H. )'
      WRITE ( 8,LF ) ( I,J,K, I=1,5 )
      IF ( II .LE. 5 ) GO TO 999
        IS = 6
        GO TO 60
C  II is greater or equal to 6
60    I3 = 1
      I4 = 1
      I5 = 1
      I6 = 1
      IF ( IS .LT. 5 ) GO TO 62
        I6 = 2
        IF ( IS .LT. 6 ) GO TO 62
          I5 = 2
          IF ( IS .LT. 7 ) GO TO 62
            I4 = 2
            IF ( IS .LT. 8 ) GO TO 62
              I3 = 2
62    IF ( NUM .EQ. 0 ) GO TO 65
      LF = FIRST//RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//
     1     RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//
     2     RPEATR//IFM(I3)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//DOT6
      WRITE ( 8,LF) ( I, J, K, R1, I = IS, IS+2 )
      LF = ' ( 9X,'//RPEATR//IFM(I4)//HN//IFM(JL)//INTVAL//IFM(1)//FORM
     4     //RPEATR//IFM(I5)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//
     5     RPEATR//IFM(I6)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//DOT6
      WRITE ( 8,LF ) ( I,J,K, R1, I=IS+3, IS+5 )
      IF ( II .LT. 12 ) GO TO 999
        IS = IS + 6
        II = 0
        GO TO 60
C  NUM equals 0, therefore no real numbers will be written
c  and 6 items may be written on one line
```

```
65    LF = FIRST//RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM
     1      //RPEATR//IFM(1)//HN//IFM(JL)//INTVAL//IFM(1)//FORM
     2      //RPEATR//IFM(I3)//HN//IFM(JL)//INTVAL//IFM(1)//FORM
     3      //RPEATR//IFM(I4)//HN//IFM(JL)//INTVAL//IFM(1)//FORM
     4      //RPEATR//IFM(I5)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//
     5      RPEATR//IFM(I6)//HN//IFM(JL)//INTVAL//IFM(1)//FORM//'1H ) '
70    WRITE ( 8,LF ) ( I,J,K, I=IS, IS+5 )
      IF ( II .LT. 12 ) GO TO 999
        IS = IS + 6
        II = 0
        GO TO 60
C
C  There are no interval markers in the repeated name
C
100       GO TO ( 110, 120, 130, 140, 150, 160,
     1              110, 120, 130, 140, 150, 160 ) II
C II is equal to 1, 7, or 13 or more
110   LF = FIRST//RPEATR//IFM(1)//HN//IFM(JL)//FORM//DOT1
      IF ( NUM .EQ. 0 ) GO TO 115
      WRITE ( 8,LF ) 1,J, R1
      IF ( II .LE. 1 ) GO TO 999
        IS = 2
        GO TO 160
115   WRITE ( 8,LF ) 1,J
      IF ( II .LE. 1 ) GO TO 999
        IS = 2
        GO TO 160
C  II is equal to 2 or 8
120   LF = FIRST//RPEATR//IFM(1)//HN//IFM(JL)//FORM//RPEATR//IFM(1)//
     1       HN//IFM(JL)//FORM//DOT2
      IF ( NUM .EQ. 0 ) GO TO 125
      WRITE ( 8,LF ) ( I,J, R1, I=1,2 )
      IF ( II .LE. 2 )  GO TO 999
        IS = 3
        GO TO 160
125   WRITE ( 8,LF ) ( I,J, I=1,2 )
      IF ( II .LE. 2 ) GO TO 999
        IS = 3
        GO TO 160
C  II is equal to 3 or 9
130   LF = FIRST//RPEATR//IFM(1)//HN//IFM(JL)//FORM//RPEATR//IFM(1)//
     1      HN//IFM(JL)//FORM//RPEATR//IFM(1)//HN//IFM(JL)//FORM//DOT3
      IF ( NUM .EQ. 0 ) GO TO 135
      WRITE ( 8,LF ) ( I,J, R1, I=1,3 )
      IF ( II .LE. 3 ) GO TO 999
        IS = 4
        GO TO 160
135   WRITE ( 8,LF ) ( I,J, I=1,3 )
      IF ( II .LE. 3 ) GO TO 999
        IS = 4
        GO TO 160
C  II is equal to 4 or 10
```

```
140   LF = FIRST//RPEATR//IFM(1)//HN//IFM(JL)//FORM//RPEATR//IFM(1)//HN
     1      //IFM(JL)//FORM//RPEATR//IFM(1)//HN//IFM(JL)//FORM//RPEATR//
     2      IFM(1)//HN//IFM(JL)//FORM//DOT4
      IF ( NUM .EQ. 0 ) GO TO 145
      WRITE ( 8,LF ) ( I,J, R1, I=1,4 )
      IF ( II .LE. 4 ) GO TO 999
        IS = 5
        GO TO 160
145    WRITE ( 8,LF ) ( I,J, I=1,4 )
      IF ( II .LE. 4 ) GO TO 999
        IS = 5
        GO TO 160
C  II is equal to 5 or 11, use TWO lines
150     IF ( NUM .EQ. 0 ) GO TO 155
C  NUM  >  0, real numbers will be written, TWO lines necessary
      LF = FIRST//RPEATR//IFM(1)//HN//IFM(JL)//FORM//RPEATR//IFM(1)//HN
     1      //IFM(JL)//FORM//RPEATR//IFM(1)//HN//IFM(JL)//FORM//DOT6
      WRITE ( 8,LF ) ( I,J, R1, I=1,3 )
      LF = ' ( 12X,'//RPEATR//IFM(IL)//HN//IFM(JL)//FORM//RPEATR//
     2      IFM(IL)//HN//IFM(JL)//FORM//DOT5
      WRITE ( 8,LF ) ( I,J, R1, I=4,5 )
      IF ( II .LE. 5 ) GO TO 999
         IS = 6
         GO TO 160
155   LF = FIRST//RPEATR//IFM(1)//HN//IFM(JL)//FORM//RPEATR//IFM(1)//
     1      HN//IFM(JL)//FORM//RPEATR//IFM(1)//HN//IFM(JL)//FORM//RPEATR
     2      //IFM(1)//HN//IFM(JL)//FORM//RPEATR//IFM(1)//HN//IFM(JL)//
     3      FORM//' 2X,1H.,2X,1H. )'
      WRITE ( 8,LF ) ( I,J, I=1,5 )
      IF ( II .LE. 5 ) GO TO 999
         IS = 6
         GO TO 160
C  II is greater or equal to 6
160   I3 = 1
      I4 = 1
      I5 = 1
      I6 = 1
      IF ( IS .LT. 5 ) GO TO 162
        I6 = 2
        IF ( IS .LT. 6 ) GO TO 162
          I5 = 2
          IF ( IS .LT. 7 ) GO TO 162
            I4 = 2
            IF ( IS .LT. 8 ) GO TO 162
              I3 = 2
162   IF ( NUM .EQ. 0 ) GO TO 165
      LF = FIRST//RPEATR//IFM(1)//HN//IFM(JL)//FORM//RPEATR//IFM(1)//
     1      HN//IFM(JL)//FORM//RPEATR//IFM(I3)//HN//IFM(JL)//FORM//DOT6
      WRITE ( 8,LF) ( I, J,  R1, I = IS, IS+2 )
      LF = ' ( 12X,'//RPEATR//IFM(I4)//HN//IFM(JL)//FORM//RPEATR//
     1      IFM(I5)//HN//IFM(JL)//FORM//RPEATR//IFM(I6)//HN//IFM(JL)//
     2      FORM//DOT6
```

```
      WRITE ( 8,LF ) ( I,J, R1, I=IS+3, IS+5 )
      IF ( II .LT. 12 ) GO TO 999
        IS = IS + 6
        II = 0
        GO TO 160
C  NUM equals zero, that is FORM contains the value, and 6 items may
c  be written on one line
165     LF = FIRST//RPEATR//IFM(1)//HN//IFM(JL)//FORM//RPEATR//IFM(1)
     1       //HN//IFM(JL)//FORM//RPEATR//IFM(I3)//HN//IFM(JL)//FORM
     2       //RPEATR//IFM(I4)//HN//IFM(JL)//FORM//RPEATR//IFM(I5)//HN//
     3       IFM(JL)//FORM//RPEATR//IFM(I6)//HN//IFM(JL)//FORM//' 1H )'
170   WRITE ( 8,LF ) ( I,J, I=IS, IS+5 )
      IF ( II .LT. 12 ) GO TO 999
        IS = IS + 6
        II = 0
        GO TO 160
999   CONTINUE
      RETURN
      END
C
      SUBROUTINE LFTYPE( IT, NUM, FIRST, NAME, N )
c          IT = number of time steps in current run
c         NUM = code for inclusion of node number in constraint name
c       FIRST = type of constraint
c        NAME = the piece of format line which will be used to write the
c               constraint name in the data line
c           N = node number of the current node
c
C This subroutine writes constraint TYPE lines to b:RELEASE.DATA
c depending on the number of time steps II (IT)
      CHARACTER * 250 LF
      CHARACTER * 75 DOT1
      CHARACTER * 60 DOT2
      CHARACTER * 45 DOT3
      CHARACTER * 31 DOT4
      CHARACTER * 20 FIRST
      CHARACTER * 17 DOT5
      CHARACTER * 10 FORM
      CHARACTER * 7 NAME
      CHARACTER * 6 DOT6
      CHARACTER * 4 IFM(2), HN
c
      DATA DOT1/' 2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H
     1.,2X,1H.,2X,1H. )'/
      DATA DOT2/' 2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H
     1. ) '/
      DATA DOT3/' 2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H. ) '/
      DATA DOT4/' 2X,1H.,2X,1H.,2X,1H.,2X,1H. ) '/
      DATA DOT5/' 2X,1H.,2X,1H. ) '/
      DATA DOT6/' 1H ) '/,              FORM / '1X,1H.,1X,'/
      DATA IFM/ ' I1,',  'I2,'/,  HN/'1HN,'/
C
```

```
            II = IT
            IS = 1
C  Define NL - 1 or 2 spaces required to write the integer N
         IF ( NUM .LE. 0 ) GO TO 5
            NL = 1
         IF ( N .GT. 9 ) NL = 2
5     GO TO ( 10, 20, 30, 40, 50, 60, 10, 20, 30, 40, 50, 60 ) II
C II is equal to 1, 7, or 13 or more
10    IF ( NUM .GT. 0 ) GO TO 11
C  TYPE line with one indice in repeated name
        LF = FIRST//NAME//IFM(1)//FORM//DOT1
      WRITE ( 8,LF ) 1
      IF ( II .LE. 1 ) GO TO 999
        IS = 2
        GO TO 60
C  TYPE line with two indices in repeated name
11      LF = FIRST//NAME//IFM(1)//HN//IFM(NL)//FORM//DOT1
        WRITE ( 8,LF ) 1, N
        IF ( II .LE. 1 ) GO TO 999
        IS = 2
        GO TO 60
C  II is equal to 2 or 8
20    IF ( NUM .GT. 0 ) GO TO 21
C  TYPE line with one indice in repeated name
        LF = FIRST//NAME//IFM(1)//FORM//NAME//IFM(1)//FORM//DOT2
      WRITE ( 8,LF ) 1, 2
      IF ( II .LE. 2 ) GO TO 999
        IS = 3
        GO TO 60
C  TYPE line with two indices in repeated name
21      LF = FIRST//NAME//IFM(1)//HN//IFM(NL)//FORM
     1          //NAME//IFM(1)//HN//IFM(NL)//FORM//DOT2
        WRITE ( 8,LF ) 1, N, 2, N
        IF ( II .LE. 2 ) GO TO 999
        IS = 3
        GO TO 60
C  II is equal to 3 or 9
30    IF ( NUM .GT. 0 ) GO TO 31
C  TYPE line with one indice in repeated name
        LF = FIRST//NAME//IFM(1)//FORM//NAME//IFM(1)//FORM
     1          //NAME//IFM(1)//FORM//DOT3
      WRITE ( 8,LF ) 1, 2, 3
      IF ( II .LE. 3 ) GO TO 999
        IS = 4
        GO TO 60
C  TYPE line with two indices in repeated name
31      LF = FIRST//NAME//IFM(1)//HN//IFM(NL)//FORM
     1          //NAME//IFM(1)//HN//IFM(NL)//FORM
     2          //NAME//IFM(1)//HN//IFM(NL)//FORM//DOT3
        WRITE ( 8,LF ) 1, N, 2, N, 3, N
        IF ( II .LE. 3 ) GO TO 999
        IS = 4
```

```
      GO TO 60
C  II is equal to 4 or 10
40    IF ( NUM .GT. 0 ) GO TO 41
C  TYPE line with one indice in repeated name
      LF = FIRST//NAME//IFM(1)//FORM//NAME//IFM(1)//FORM
     1         //NAME//IFM(1)//FORM//NAME//IFM(1)//FORM//DOT4
     WRITE ( 8,LF ) 1, 2, 3, 4
     IF ( II .LE. 4 ) GO TO 999
      IS = 5
      GO TO 60
C  TYPE line with two indices in repeated name
41     LF = FIRST//NAME//IFM(1)//HN//IFM(NL)//FORM
     1         //NAME//IFM(1)//HN//IFM(NL)//FORM
     2         //NAME//IFM(1)//HN//IFM(NL)//FORM
     3         //NAME//IFM(1)//HN//IFM(NL)//FORM//DOT4
      WRITE ( 8,LF ) 1, N, 2, N, 3, N, 4, N
      IF ( II .LE. 4 ) GO TO 999
      IS = 5
      GO TO 60
C  II is equal to 5 or 11
50    IF ( NUM .GT. 0 ) GO TO 51
C  TYPE line with one indice in repeated name
      LF = FIRST//NAME//IFM(1)//FORM//NAME//IFM(1)//FORM
     1         //NAME//IFM(1)//FORM//NAME//IFM(1)//FORM
     1         //NAME//IFM(1)//FORM//DOT5
     WRITE ( 8,LF ) 1, 2, 3, 4, 5
     IF ( II .LE. 5 ) GO TO 999
      IS = 6
      GO TO 60
C  TYPE line with two indices in repeated name
51     LF = FIRST//NAME//IFM(1)//HN//IFM(NL)//FORM
     1         //NAME//IFM(1)//HN//IFM(NL)//FORM
     2         //NAME//IFM(1)//HN//IFM(NL)//FORM
     3         //NAME//IFM(1)//HN//IFM(NL)//FORM
     4         //NAME//IFM(1)//HN//IFM(NL)//FORM//DOT5
      WRITE ( 8,LF ) ( I, N , I=1,5 )
      IF ( II .LE. 5 ) GO TO 999
      IS = 6
C  II is greater or equal to 6
60    I3 = 1
     I4 = 1
     I5 = 1
     I6 = 1
     IF ( IS .LT. 5 ) GO TO 62
      I6 = 2
      IF ( IS .LT. 6 ) GO TO 62
       I5 = 2
       IF ( IS .LT. 7 ) GO TO 62
        I4 = 2
        IF ( IS .LT. 8 ) GO TO 62
         I3 = 2
62    IF ( NUM .GT. 0 ) GO TO 65
```

```
C  TYPE line with one indice in repeated name
        LF = FIRST//NAME//IFM(1)//FORM//NAME//IFM(1)//FORM
     1          //NAME//IFM(I3)//FORM//NAME//IFM(I4)//FORM
     2          //NAME//IFM(I5)//FORM//NAME//IFM(I6)//FORM//DOT6
63    WRITE ( 8,LF ) ( I, I = IS,IS+5 )
      IF ( II .LE. 11 ) GO TO 999
        IS = IS + 6
        II = 0
        GO TO 60
C  TYPE line with two indices in repeated name
65      LF = FIRST//NAME//IFM(1)//HN//IFM(NL)//FORM
     1          //NAME//IFM(1)//HN//IFM(NL)//FORM
     2          //NAME//IFM(I3)//HN//IFM(NL)//FORM
     3          //NAME//IFM(I4)//HN//IFM(NL)//FORM
     4          //NAME//IFM(I5)//HN//IFM(NL)//FORM
     5          //NAME//IFM(I6)//HN//IFM(NL)//FORM//DOT6
67      WRITE ( 8,LF ) ( I, N, I=IS, IS+5 )
        IF ( II .LE. 11 ) GO TO 999
        IS = IS + 6
        II = 0
        GO TO 60
999   CONTINUE
      RETURN
      END
C
      SUBROUTINE LFRHS ( QUANT, ICODE, NAME, M, ARRAY )
c      QUANT = number of time steps in the current run
c      ICODE = code to write the node number
c       NAME = character variable of the piece of format line
c              used to write the beginning of the constraint name
c          M = node number of current node to include in the
c              constraint name
c ARRAY(14) = a vector of the constant values of the RHS of the
c              constraints for every time step
C
      INTEGER QUANT
      DIMENSION ARRAY(14), AR(6), IV(6), IL(6)
C  AR(6) vector used to hold RHS constant values until six non-zero
c        values are found to be written on a line in the data file
c  IV(6) vector used to hold the time step index until there are six
c        values to write
c  IL(6) vector used to hold the index for the format required (I1 or
c        I2), to write the time step index, until there are six values
c        to write
      CHARACTER * 250 LF
      CHARACTER * 75 DOT1
      CHARACTER * 60 DOT2
      CHARACTER * 45 DOT3
      CHARACTER * 31 DOT4
      CHARACTER * 20 RHS
      CHARACTER * 17 DOT5
      CHARACTER * 11 F10
```

```
      CHARACTER * 7 NAME
      CHARACTER * 6 DOT6
      CHARACTER * 4 IFM(2), HN
c
      DATA RHS /' ( 2H .,6H _RHS_,    '/,     F10 /'1X,F8.3,1X,'/
      DATA DOT1/' 2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H
     1.,2X,1H.,2X,1H. )'/
      DATA DOT2/' 2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H.,2X,1H
     1. )'/
      DATA DOT3/' 1X,1H.,1X,1H.,1X,1H.,1X,1H.,1X,1H.,1X,1H. )'/
      DATA DOT4/' 2X,1H.,2X,1H.,2X,1H.,2X,1H. )'/
      DATA DOT5/' 2X,1H.,2X,1H. )'/
      DATA DOT6/' 1H )'/
      DATA IFM/ ' I1,',  'I2,'/,  HN/'1HN,'/
C
      IF ( ICODE .LE. 0 ) GO TO 50
C  Small loop to print one line defining the NON-ZERO RHS 6 at a time
C    for NAMES with only one index, no Nn
5         N = 0
          I = 0
20          I = I + 1
        IF ( I .GT. QUANT ) GO TO ( 81, 82, 83, 84, 85 ) N
        IF ( I .GT. QUANT .AND. N .LE. 0 ) GO TO 999
        IF ( ARRAY(I) .LE. 0.0001 ) GO TO 20
            N = N + 1
            AR(N) = ARRAY(I)
            IV(N) = I
            IL(N) = 1
       IF ( IV(N) .GT. 9 ) IL(N) = 2
       IF ( N .LT. 6 ) GO TO 20
         LF = RHS//NAME//IFM(IL(1))//F10//'T27,'//NAME//IFM(IL(2))//F10
     1        //'T44,'//NAME//IFM(IL(3))//F10//'/ 8X,'//NAME//IFM(IL(4))
     2        //F10//'T27,'//NAME//IFM(IL(5))//F10//'T44,'//NAME//
     3        IFM(IL(6))//F10//DOT6
       WRITE ( 8,LF ) ( IV(L), AR(L), L=1,6 )
            N = 0
          GO TO 20
C  End of loop - at this point I=QUANT and N number of RHS constraints
c                 have yet to be written.  AR(n), IV(n), and IL(n) are
C                 ready to write them
C One RHS constraint to go
81      LF = RHS//NAME//IFM(IL(1))//F10//DOT1
           GO TO 87
C Two
82      LF = RHS//NAME//IFM(IL(1))//F10//'T27,'//NAME//IFM(IL(2))//F10
     1       //DOT2
           GO TO 87
83      LF = RHS//NAME//IFM(IL(1))//F10//'T27,'//NAME//IFM(IL(2))//F10
     1      //'T44,'//NAME//IFM(IL(3))//F10//DOT3
           GO TO 87
84      LF = RHS//NAME//IFM(IL(1))//F10//'T27,'//NAME//IFM(IL(2))//F10
     1      //' / 8X,'//NAME//IFM(IL(3))//F10//'T27,'//NAME//
```

```
     2      IFM(IL(4))//F10//DOT4
            GO TO 87
85      LF = RHS//NAME//IFM(IL(1))//F10//'T27,'//NAME//IFM(IL(2))//F10
     1       //'T44,'//NAME//IFM(IL(3))//F10//' / 8X,'//NAME//IFM(IL(4))
     2       //F10//'T27,'//NAME//IFM(IL(1))//F10//DOT5
87      WRITE ( 8,LF ) ( IV(L), AR(L), L=1,N )
            GO TO 999
C
C  ICODE equals 0, writing NAMEiNn constraint
C  Define ML - 1 or 2 spaces required to write the integer M
50          ML = 1
          IF ( M .GT. 9 ) ML = 2
C  Small loop to print one line defining the NON-ZERO RHS 6 at a time
          N = 0
          I = 0
70          I = I + 1
        IF ( I .GT. QUANT ) GO TO ( 71, 72, 73, 74, 75 ) N
        IF ( I .GT. QUANT .AND. N .LE. 0 ) GO TO 999
        IF ( ARRAY(I) .LE. 0.0001 ) GO TO 70
            N = N + 1
            AR(N) = ARRAY(I)
            IV(N) = I
            IL(N) = 1
        IF ( IV(N) .GT. 9 ) IL(N) = 2
        IF ( N .LT. 6 ) GO TO 70
       LF = RHS//NAME//IFM(IL(1))//HN//IFM(ML)//F10//'T27,'//NAME//
     1     IFM(IL(2))//HN//IFM(ML)//F10//'T44,'//NAME//IFM(IL(3))//HN//
     2     IFM(ML)//F10//' / 8X,'//NAME//IFM(IL(4))//HN//IFM(ML)//F10//
     3     'T27,'//NAME//IFM(IL(5))//HN//IFM(ML)//F10//'T44,'//NAME//
     4     IFM(IL(6))//HN//IFM(ML)//F10//DOT6
        WRITE ( 8,LF ) ( IV(L), M, AR(L), L=1,6 )
            N = 0
           GO TO 70
C  End of loop - at this point I=QUANT and N number of RHS constraints
c                have yet to be written.  AR(n), IV(n), and IL(n) are
C                ready to write them
C One RHS constraint to go
71      LF = RHS//NAME//IFM(IL(1))//HN//IFM(ML)//F10//DOT1
            GO TO 77
C Two
72      LF = RHS//NAME//IFM(IL(1))//HN//IFM(ML)//F10//'T27,'//NAME//
     1       IFM(IL(2))//HN//IFM(ML)//F10//DOT2
            GO TO 77
73      LF = RHS//NAME//IFM(IL(1))//HN//IFM(ML)//F10//'T27,'//NAME//
     1       IFM(IL(2))//HN//IFM(ML)//'1X,F7.3,1X,T44,'//NAME//
     2       IFM(IL(3))//HN//IFM(ML)//'1X,F7.3,1X,'//DOT3
            GO TO 77
74      LF = RHS//NAME//IFM(IL(1))//HN//IFM(ML)//F10//'T27,'//NAME//
     1       IFM(IL(2))//HN//IFM(ML)//F10//' / 8X,'//NAME//IFM(IL(3))//
     2       HN//IFM(ML)//F10//'T27,'//NAME//IFM(IL(4))//HN//IFM(ML)//
     3       F10//DOT4
            GO TO 77
```

```
75      LF = RHS//NAME//IFM(IL(1))//HN//IFM(ML)//F10//'T27,'//NAME//
    1        IFM(IL(2))//HN//IFM(ML)//F10//'T44,'//NAME//IFM(IL(3))//HN
    2        //IFM(ML)//F10//' / 8X, '//NAME//IFM(IL(4))//HN//IFM(ML)//
    3        F10//'T27,'//NAME//IFM(IL(5))//HN//IFM(ML)//F10//DOT5
77     WRITE ( 8,LF ) ( IV(L), M, AR(L), L=1,N )
999   CONTINUE
      RETURN
      END
```

APPENDIX C - LIST OF SYMBOLS

This appendix lists the decision variables and the constant parameters of PROJPLAN and PROJOP. First are the decision variables of PROJPLAN:

D_i = absolute difference between seasons of the quantity of land left idle in all the districts

D_1 = difference (ha) of the available land not cultivated in season 1 and season 2

D_2 = difference (ha) of the available land not cultivated in season 2 and season 3

D_3 = difference (ha) of the available land not cultivated in season 1 and season 3

DD_m = difference in the number of uncultivated hectares in the solution. If the difference is positive, the even index m is non-zero and the corresponding odd term is zero. Conversely, if the difference is negative, the odd index m is non-zero and the corresponding even term is zero.

H_{ijk} = decision variable of the number of hectares to be sown in crop j in district k in season i

$Q_{i\tau n}$ = regulated releases (Mm3) to the downstream node, n+1, or n-q (when n is the last node on a tributary flowing to the node n-q), during the sub-season τ of the season i

$QD_{i\tau(n-p)}$ = regulated diversion releases (Mm3) from a node, n-p, to the node n during the sub-season τ of season i

$QR_{i\tau n}$ = irrigation releases (Mm3) delivered to the districts served by the node n during the sub-season τ of the season i

$QR_{i\tau(n-m)}$ = irrigation releases (Mm3) from a node, n-m, of which a quantifiable percentage reaches node n during the same sub-season τ of season i

$S_{i\tau n}$ = storage (Mm3) at the node n at the end of the sub-season τ of the season i. The storage volume at the beginning of the first season should be included with the predicted inflows at the node n

$S_{i(\tau-1)n}$ = storage (Mm3) at the node n at the beginning of sub-season τ of the season i. When $\tau = 1$, the storage at the end of the preceding season enters into the calculation:

$$S_{i0n} = S_{(i-1)Ln}$$

where L = number of sub-seasons in season i-1

Secondly the constant parameters in PROJPLAN:

b_{ijk} = land (ha) required to grow a given level of crop j in district k in season i, may be 0

c_{ijk} = net benefit (Rs/ha) of the crop j in district k in season i

cc_1 = cost (Rs/ha) of land left uncultivated in season 1 v/s the amount left idle in season 2

cc_2 = cost (Rs/ha) of land left uncultivated in season 3 v/s the amount left idle in season 2

cc_3 = cost (Rs/ha) of land left uncultivated in season 3 v/s the amount left idle in season 1

cl_{ijk} = cost of labour (Rs/ha) associated with growing crop j in district k over the entire season i

$dmax_i$ = limit in hectares of the allowable difference between the season i and the season i+1 of idle land

f_{ijk} = fertilizer costs (Rs/ha) required to grow crop j in district k over the entire season i

hh_{ik} = total arable land (ha) in the district k in the season i

I = 3 consecutive growing seasons starting October 1 or April 15

$inf_{i\tau n}$ = predicted sub-seasonal inflows (Mm^3) from the drainage basin of the node n during the sub-season τ of the season i. The storage at the node n at the beginning of the first sub-season is included in inf1,1,n

J = number of prospective and possible crops, up to 50 dimensioned in PROJPLAN

J^1 = list of crops with growing seasons longer than 6 months

J^2 = list of crops with minimum production quotas in a given season

Jp = number of prospective crops, j, combined in a group

K = number of irrigation districts, up to 20 dimensioned in PROJPLAN, currently 14 districts operate in the Mahaweli Irrigation Project

K^1 = the list of irrigation districts k which obtain water from the node n, up to 5 irrigation districts form one node

$loss_{i\tau n}$ = estimated seasonal evaporation and seepage losses from the node n during the sub-season τ of the season i

N^1 = number of tributary nodes whose regulated releases reach the node n

N^2 = number of nodes whose diverted releases flow to the node n

N^3 = number of nodes whose irrigation return flows eventually reach the node n

p_{ijk} = price (Rs/ha) of crop j grown in district k in season i

$pcrop_{i,jp}$ = minimum production level (T) for that combination of crops jp in the season i

$qmin_{i\tau n}$ = minimum downstream sub-seasonal regulated release (Mm^3) allowed from the node n during the sub-season τ of the season i

$qmax_{i\tau n}$ = maximum downstream sub-seasonal regulated release (Mm^3) allowed from the node n during the sub-season τ of the season i

$qdmax_{i\tau n}$ = maximum regulated diversion canal releases (Mm^3) from node n during the sub-season τ of the season i, limited by the canal capacity

$smin_{i\tau n}$ = minimum allowed storage (Mm^3) at node n at the end of the sub-season τ of season i

$smax_{i\tau n}$ = maximum allowed storage (Mm^3) at the node n at the end of the sub-season τ of the season i

t_{ijk} = maximum land (ha) suitable for growing crop j in district k in season i.

u_{ijk} = water requirement (Mm^3/ha) of the crop j in district k over the entire season i, including delivery losses

$ul_{ijk\tau}$ = water requirement (Mm^3/ha) of the crop j in district k during the sub-season τ in the season i, including the delivery losses

w_{ik} = cost of water delivered (Rs/1000 m3) to district k in season i. This term may be used to minimize irrigation deliveries to certain districts by increasing their costs relative to the other districts

wd = cost (Rs) of disparate hectares left idle

Y_{ijk} = yield (T/ha) of the crop j in the district k in the season i

$ymin_{ij}$ = minimum production target (T) for the crop j in the season i, used to guarantee production of certain crops over all the districts

$\alpha_{i\tau(n-p)}$ = diversion canal transport loss as a percent of the seasonal flow, particular to the diversion from the node n-p to the node n, during the sub-season τ of season i

$\beta_{i\tau(n-m)}$ = irrigation return flow percentage of the irrigation turnout at the node n-m which flows to the node n during the sub-season τ of season i

Third the decision variables in PROJOP:

D_{ikn} = linearization variable, the incremental irrigation deficit (Mm3) occurring at the time step i at the irrigation node n in the linearization interval k

DIF_i = absolute difference (GW-hr) between the power produced in one time step and the next

$DIFM_i$ = negative difference between the total power produced at the time step i and that produced at the time step 1+1

$DIFP_i$ = positive difference between the total power produced at the time step 1 and that produced at the time step i+1

ENG_{in} = estimated energy produced (GW-hr) at the time step i at the node n

Q_{ikn} = linearization variable, the turbine release at time step i at the power plant n in a linearization interval k

QD_{in} = regulated diversion releases (Mm3) from node n at the time step i

$QD_{i(n-p)}$ = flow diverted to the noden from the node n-p at the time step i

QDF_{in} = irrigation deficit (Mm3) occurring at time step i at the node n

QR_{in} = Irrigation releases (Mm3) delivered to the irrigation district from the node n during the time step i

$QR_{(i-x)(n-m)}$ = the volume diverted for irrigation at node n-m at the time step i-x, a significant portion of which flows to the node n at the time step i

QS_{in} = regulated downstream releases (Mm3) at time step i at node n that was not used to generate electricity or if there is no power plant at n, the volume to the downstream node

$QS_{i(n-1)}$ = downstream flows from the node n-1 to node n at the time step i

$QS_{i(n-q)}$ = downstream flows from the node n-q, a tributary to node n, at time step i

QTB_{in} = turbine volume(Mm3) at the time step i at the node n

$QTB_{i(n-z)}$ = turbine volume from the power plant n-z which flows to the node n at the time step i

S_{ikn} = linearization variable, the incremental storage (average ST for the time step - minimum storage of the interval k) at power plant node n

$SHORT_i$ = decisision variable for the shortfall in target energy (GW-hr) from total energy suppied by all power plants in the time step i

ST_{in} = storage volume (Mm3) of the reservoir or tank at the node n at the end of time step i

STlln = desired storage volume (Mm3) of the reservoir or tank at node n at the end of the last time step II

T_{ikn} = linearization variable, the incremental storage (STin - minimum storage of the interval k) at the end of the time step i at the reservoir or tank at node n

X_{ikn} = Integer variable permitting only one interval k to have non-zero values at the time step i at irrigation node n

Y_{ikn} = integer variable used to select one interval k at the time step i at the node n for the linearization of the surface area - slope curve

Z_{ikn} = integer variable used to select one interval k at the time step i at the node n for the linearization of the electrical power curve

Finally the constants of PROJOP are these:

c_{in} = correcting factor used to adjust the estimate of energy production according to the average volume of water stored at the node n at time step i.

cwr_{in} = volume of water (Mm3) demanded by crops supplied by the node n for the time step i, including transportation losses, a given quantity based on the known cropping pattern and expected evapo-transportation rates for the remainder of the season

$difmax_i$ = maximum allowed difference (GW-hr) in energy generated between the time step i and i+1

h_{in} = power coefficient (GW-Hr/Mm3) for a representative head in the operating range of the reservoir n during time step i.

II = number of time steps left in the current season

inf_{in} = local involume(Mm3) in the natural basin of the node n, not including upstream regulated nor diverted flows. The inflows of the first time step include any reservoir or tank storage, therefore the current storage is added to the local inflow by the data preparation program

K = number of intervals used to linearize the power curve

K^1 = number of intervals used to linearize the irrigation losses curve.

K^2 = number of intervals used to linearize the irrigation losses curve using the second method

K^3 = number of intervals used to linearize the surface area - storage curve of the reservoir or tank

N^1 = list of nodes which divert to the node n

N^2 = list of nodes which are the last nodes on tributaries to the node n

N^3 = list of nodes where a percentage of their irrigation deliveries return to the node n at the time step i

Ni = number of nodes which deliver water to irrigation districts where the losses due to irrigation deficits are considered linear

Ni^1 = number of nodes which deliver water to irrigation districts where the losses due to irrigation deficits are linearized by mixed integer constraints

Ni^2 = number of nodes which deliver water to irrigation districts where the losses due to irrigation deficits are linearized by the second method

Np = number of hydro-electric power plants modeled in the project

Nr = number of nodes which have significant storage

$pmin_i$ = minimum total energy production (GW-hr) permitted at the time step i

$pmax_i$ = maximum total energy production (GW-hr) permitted at the time step i

$qdmin_{in}$ = minimum regulated diversion canal releases (Mm3) from node n at the time step i

$qdmax_{in}$ = maximum regulated diversion canal releases (Mm3) from node n at the time step i limited by the canal capacity

$qrmin_{in}$ = minimum irrigation releases (Mm3) from node n at the time step i

$qrmax_{in}$ = maximum irrigation releases (Mm3) from node n at the time step i, limited by canal capacities

$qsmin_{in}$ = minimum downstream releases (Mm3) from node n at the time step i

$qsmax_{in}$ = maximum downstream releases (Mm3) from node n at the time step i

$qtbmin_{in}$ = minimum turbine releases (Mm3) from the power plant at the node n at time step i

$qtbmax_{in}$ = maximum turbine releases (Mm3) from the power plant at the node n at time step i

$smax_{in}$ = maximum allowed storage (Mm3) at node n at the end of time step i

sf_{in} = target storage (Mm3) for the end of this monsoon season at node n

$smin_{in}$ = minimum allowed storage (Mm3) at the node n at the end of time step i

sto_{in} = representative storage volume (Mm3) of the reservoir at the node n for the time step i.

α_{kn} = maximum storage volume (Mm3) in interval k at the power plant node n

β_{ikn} = average power coefficient (GW-hr/Mm3) which calculates the amount of power generated at time step i at power plant n for the average storage in the interval k

Γ_{kn} = maximum storage volume (Mm3) of the interval k at the reservoir or tank n

¥ = 4, when i is the last quarter-month time step before the final monthly time steps when optimizing at the start of the season, otherwise, it is equal to 1

δ_{kn} = slope of the surface area - storage curve for the reservoir or tank at the node n in the linearization interval k

ϵ_{kn} = minimum surface area (km2) of the reservoir or tank at the node n in linearization interval k

θ_{ikn} = maximum irrigation deficit (Mm3) in interval k for time step i at the irrigation node n

μ_1 = unit price (Rs/GW-hr) of electrical production

μ_2 = penalty price (Rs/Mm3) of the reduced yields due to a unit volume of irrigation deficit

μ_{3n} = unit price (Rs /Mm3) which can be assigned to the agricultural or other benefits of a reservoir n storing more water at the end of the growing season than stipulated by PROJPLAN. When equal to zero, future benefits of additional water are ignored

μ_4 = unit price (Rs/GW-hr) of a difference in power production between one time step and another

σ_{in} = reservoir loss coefficient (mm) used to calculated the evaporation and seepage losses in the tank or reservoir at node n at time step i

$\pi_{in'}$ = percentage of irrigation release volume which remains after transporting water in the canal diversion n from the node n-p to the node n at the time step i

$\pi'_{in'}$ = percentage of irrigation flow, diverted at n-m (the n'), which will contribute to the regulated inflow of the node n during the time step i

Φ_{ikn} = the losses (Rs) due to a reduction of yield from a deficit of the magnitude of the minimum level for the linearization interval k at the node n at the time step i

Ω_{ikn} = the slope of the losses v/s water deficit curve for the crops of the irrigation district at the node n at the time step i due to a deficit of the magnitude of the linearization interval k