Rural Land Use Allocation for Flood Damage Mitigation Through Dynamic Programming Considering Multiple Annual Floods and Time Dependent Flood Damage Susceptibility

A thesis
presented to the University of Manitoba
in partial fulfillment of the
requirements for the degree of
Master of Science
in
Civil Engineering

by

T. C. Hannan

Winnipeg, Manitoba

RURAL LAND USE ALLOCATION FOR FLOOD DAMAGE MITIGATION THROUGH DYNAMIC PROGRAMMING CONSIDERING MULTIPLE ANNUAL FLOODS AND TIME DEPENDENT FLOOD DAMAGE SUSCEPTIBILITY

BY

T.C. HANNAN

A thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

ABSTRACT

A dynamic programming based model is used to determine a flood damage mitigation system for a small rural watershed with an agricultural economic base. The system combines both structural and non-structural approaches to flood damage mitigation, but the emphasis of the model is on the non-structural method of land use planning. The model objective is to maximize total expected net benefits from the crops grown for the watershed as a whole, given the expected damages from flooding. Two decisions are made by the model: 1) the location and volume of flood water detention in order to achieve maximum benefit, and 2) the best crop type to grow in each area.

The possibility of having more than one damaging flood in a single growing season is explicitly considered in the assessment of the expected flood damage. Also explicitly considered is the time dependent susceptibility of the crops to flood damage. The model is applied to the Wilson Creek watershed in western Manitoba for demonstration.

ACKNOWLEDGEMENTS

I would like to thank Dr. Ian Goulter for his guidance and support during the course of this study, and during my entire graduate program, but especially for the opportunity to do it in the first place.

TABLE OF CONTENTS

Appendix page

LIST OF TABLES

LIST OF FIGURES

Chapter I

INTRODUCTION

1.1 BACKGROUND

The predominant use of flood control structures for flood damage reduction in the urban environment is generally justified by the high cost of flood damages. In the rural environment, the land values, and the damage potential of the economic activity on the land, are much lower and the use of costly control structures may not be economically justified. The use of a plan involving reduced structural protection in combination with non-structural measures to reduce flood damages, which has been studied for urban watersheds, may be particularly applicable for rural watersheds.

Noneconomic aspects of flood control, such as loss of life and disruption of services, may make costly structural measures justifiable, even if not economically so. In the less populated rural areas, these concerns are greatly reduced, and where the watershed is small, risk to life is minimal. In spite of the apparent applicability of combinations of structural and non-structural measures to the rural watershed, very little research has been directed toward this.

In the rural watershed, due to the relative scarcity of domestic dwellings or commercial buildings on the flood plain, flood waters can be stored in the floodplain temporarily without significant damage to structures in order to reduce the intensity of the flood downstream. A structure of some sort is required to facilitate the storage. However, the decision to store water in a part of the watershed also implies land use decision, and therefore is a non-structural damage mitigation method, making this flood damage mitigation system a combination of structural and non-structural measures. The storage of water in a location has been given the term "hydrologic use" by Hopkins et al.(1978). The term is used to indicate that the land use decision is specifically water related, to differentiate them from other land use decisions which, while effecting the hydrologic regime, do so unintentionally.

In an agricultural area, where the predominant economic activity is cropping, there are two aspects to the land use decision process. One is the "hydrologic" use previously noted, and the other is the crop type most compatible to the hydrologic land use decision. In the urban flood plain, alternative land uses have varying effects on the runoff patterns from the areas for which the land use decision is made, and consequently on the hydrological conditions downstream. The result is an interdependence among reaches of the flood plain, based on all land use decisions, not only the "hydrologic" uses. In a rural watershed, if crop type is the only other land use decision available, downstream hydrologic conditions are affected only by the hydrologic use decisions, because the variety of crop does not have a significant effect on the downstream hydrologic conditions. This reduced dependence among areas of the basin is another difference between rural and urban watershed flood damage mitigation systems.

The preliminary research for developing a method of assessing flood damage control systems for small rural watersheds has been done by Goulter and Morgan (1983), whose approach is detailed in the literature review. This study is a continuation of that work, and a refinement of the model described in the paper.

1.2 PROBLEM DESCRIPTION

To assess a series of land use decisions, some measure of the costs and benefits from each decision, and from the group of decisions for the watershed as a whole, is required. Typically, this is done with the evaluation of expected damages. There are usually two aspects to an expected damage calculation, the damage caused by a flood of a known magnitude, and the probability of exceedence of that flood.

In most previous work, including the study by Goulter and Morgan (1983), the assumption is made that the flood damage caused by a given level and duration of flooding is constant and independent of the time of occurrence of the flood. This assumption is not valid in an area where cropping is the dominant economic activity. The susceptibility of the crop to flooding is dependent on the stage of growth of the plant when the event occurs (Leyshon and Sheard, 1974).

The inclusion of this complication creates differences in how other aspects of the evaluation are carried out. The calculation of expected damage must include the probability of the crop being at a certain stage of growth when the flood occurs. Therefore, the susceptibility to flood damage during the growing season will have its own probability distribution, based on meteorological conditions and variation in planting dates.

Another assumption typically made in expected damage calculations is that all annual damage is caused by the largest flood of the year. Though this may be a valid assumption in urban watersheds, it is not applicable to an agricultural region. Two or more floods could occur within the same growing season, and the amount of damage attributed to a single flood depends heavily on the amount of damage caused by previous events in the same growing season.

This study describes the development of a model capable of explicitly considering these issues.

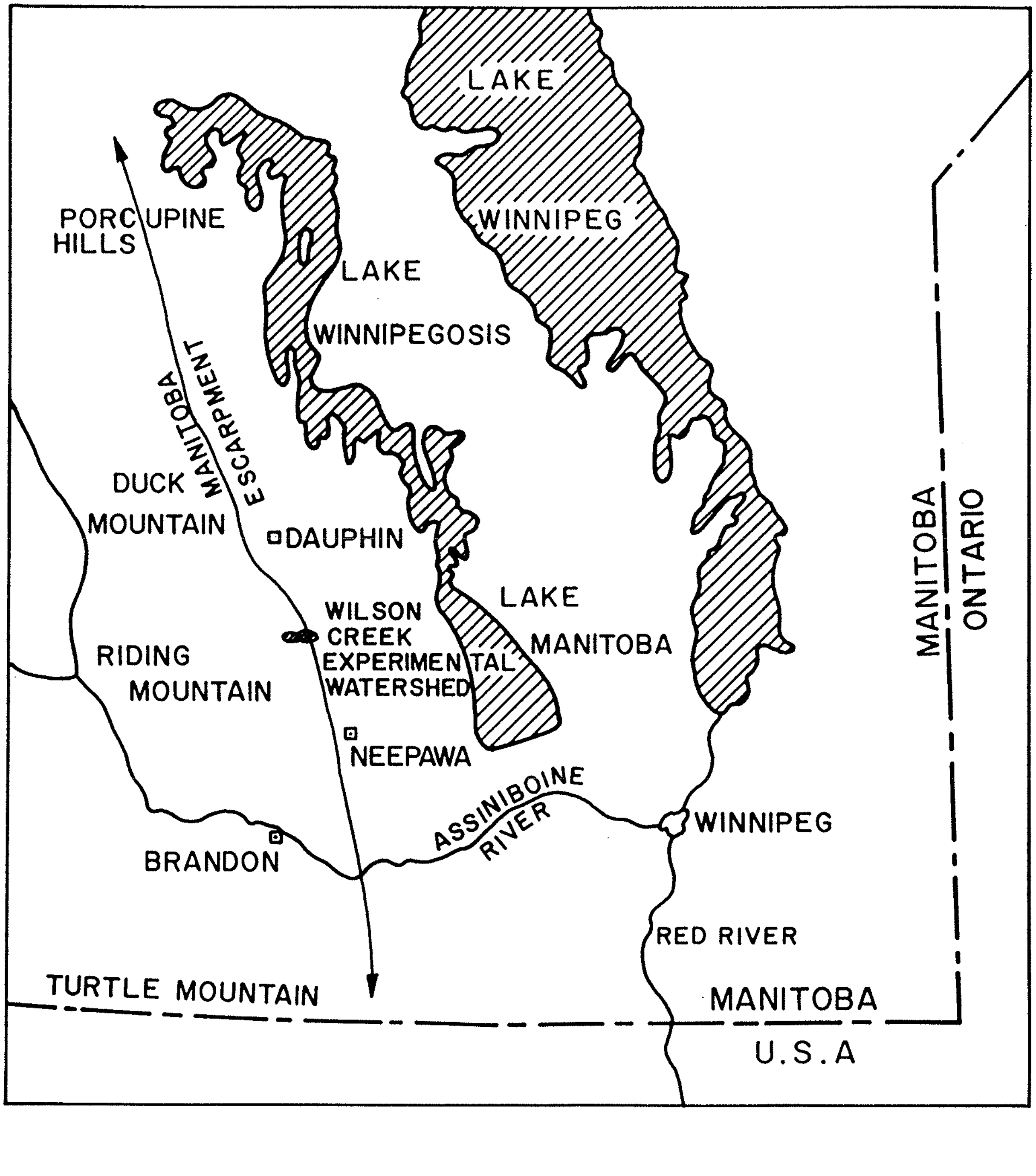
1.3 REGIONAL RELEVANCE

In Manitoba, most agricultural crop losses due to flooding on small rural watersheds are caused by short term, intense, summer convective or orographic rain storms. The frequency distributions of these storms, and the resulting floods, may vary over the growing season. The variation in the onset of spring causes variation in planting dates, and so a probability distribution also exists for the growth stage of the plant at various times during the growing season.

The watershed used to demonstrate the model was chosen because of its limited size, and the fact that it is representative of a large number of basins of similar size and physical features all along the Manitoba Escarpment (see Figure 1). Most summer rainfall in this area comes from high intensity thunderstorms, rather than prolonged showers. These types of storms can occur on the escarpment more than once in a year.

Shale deposits from years of flooding have reduced the value of escarpment watersheds for agricultural production, but this is still the economic mainstay of the region. Costly structural flood control measures would be unjustifiable economically, but a combination of smaller structures and appropriate land use decisions may be feasible.

Figure 1: Location of Wilson Creek Watershed (Scource: Goulter and Morgan, 1983)



The model developed is used to determine the location and volume of storage of the flood water, as well as the most beneficial crop type to be planted in the specific locations within the basin, in order to maximize expected net benefits from the watershed.

Chapter II

LITERATURE REVIEW

In the past twenty years, a considerable amount of research has been devoted to the study of flood damage mitigation measures other than the purely structural, and generally costly measures.

Lind (1967) discusses the merits of five potential flood damage mitigation measures. These are: 1) Structural riverbed transformation 2) Flood insurance 3) Flood warning 4) Flood proofing and 5) Flood zoning. Lind argues that structural measures and flood insurance are the best methods of reducing losses due to flooding, and that zoning is not particularly effective.

Krutilla (1966) suggests that flood insurance is not effective because it only distributes the losses in time and does not reduce them. Also, those who pay for it may not be the ones who benefit from it.

James (1965, 1967) discusses using combinations of structural, non-structural (land use decisions), and flood proofing measures in both urban and rural environments. In both papers James argues that while the primary index of flood damage has been the area inundated by floods at various depth intervals, other factors, such as duration, velocity, sediment content, and flood frequency by season may also contribute. James (1965) also discusses the possible need to synthesize damage information, due to its frequent paucity.

James (1965) also details factors involved in crop damages specifically. He states that crop damages tend to be independent of depth. James also suggests that while flood proofing is not possible for the crop itself, shifting the crop within the flood area could result in reduced losses. This, in effect, is a land use decision.

James (1967) elaborates on the earlier study by developing a model which determines the optimum combination of structural and non-structural flood damage mitigation measures for a watershed that has both urban and agricultural areas. Among the inputs to the agricultural aspect of the model are: flood damage parameters which include unit damages and market values, soil fertility classes, and land use control costs. Due to the difficulty in including a large number of flood levels, an optimum design flood frequency is determined, rather than using the range of possible flood levels.

Day (1970) used a recursive linear program in order to determine land uses that will maximize benefits for an urban area. Flood losses are treated as an additional operating cost for the community. This study does not permit the inclusion of any structural flood control measures.

Bialas and Loucks (1978) state that a structural flood control system can be rendered ineffective if it is not used with a land use plan. A model is developed which minimizes expected damage in an urban watershed. The decision variable is the type of land use zone each region should be comprised of, and how much of each zone type. Again, only one flood hydrograph is used for evaluation.

Ball et al. (1978) also promote the use of a combination of structural and non-structural flood control measures. A recursive model is used for evaluation in this study. An option, which includes possible land use decisions as well as structural devices, is picked and run through the model. More options are tried until an acceptable system is determined. A complicated routing approach is used which involves the use of the Muskingum routing method. Since the hydrograph must be re-evaluted for each combination of possible decisions, it can be very cumbersome, so again only one design flood is used.

A series of papers, Hopkins et al. (1976, 1977, 1978, 1980, 1981) address the use of flood plain management decisions as decision variables in recursive models used to maximize benefits from the flood plain. Total economic rent is the criterion by which the model is assessed. Hopkins et al. (1978) use the term "hydrologic use" as one of the possible land use types. This term means that some areas may be used to store water, which would reduce economic rent at that particular location, in order to have an increase in economic value within the basin as a whole.

Hopkins et al. (1978,1981) address the problem of mathematically intensive routing procedures by using a triangular hydrograph, which has similar physical properties to the more typical curvilinear hydrograph. The benefit of the triangular shape is that it requires the fewest number of coordinates to describe it, thus reducing the computational effort required. Reduction in computational effort achieved by using the triangular hydrograph is particularly important in recursive models such as the dynamic programming approach used by Hopkins et al. (1978,1981), due to the number of passes that must be made through the routing procedure. This routing model also allows for the inclusion of local inflow with relative ease. Even with this simplification, Hopkins uses only one design flood hydrograph for evaluation.

Like most of the previously cited studies, Hopkins et al. (1978,1981) refer only to urban watershed evaluation. There are ,however, two papers which address rural, agricultural flood control systems specifically, Lacewell and Eidman (1972), and Goulter and Morgan (1983).

Lacewell and Eidman (1972) develop a model to estimate agricultural flood damages for sample points, which are representative of an area surrounding them. This is a refinement of the more commonly used "composite acre" method of damage estimation. The composite acre is used to represent the entire reach under evaluation and the percentage of each available land use type allowed in the particular reach. This is useful for urban flood plains, but in agricultural settings no restrictions of this type are placed on the reaches. Also, the composite acre method does not allow for land use decisions to be made for areas smaller than the composite acre, and therefore none can be made for areas smaller than the reach itself. The sample point method does, however, allow for damage estimation of specific locations around the reach. Lacewell and Eidman use a damage estimator based on characteristics of a sample point, including land use, location, soil productivity, and depth of inundation. Note that the use of this last value is in contrast with James (1965) who claims that there tends to be no relationship between depth of flooding and amount of damage.

Lacewell and Eidman also use a series of discrete flood sizes to represent the complete distribution of flood levels. The damage factors are then weighted by a seasonal probability of the occurrence of a flood of that magnitude.

Goulter and Morgan (1983) refine the system used by Hopkins et al. (1981) for use in an agricultural environment. By using a combination of structural and non-structural damage mitigation systems net benefits for an agricultural watershed are maximized through a dynamic programming procedure. Each reach of the stream has available to it, three possible land use types. The concept of "hydrologic use" developed by Hopkins et al. (1981) is used here, to define the three land use types as follows. Some reaches can be used to store water to reduce flooding in downstream areas in order to maximize benefits in the watershed overall. The area designated to store water is defined as the planned flooding area. The area used for water storage will not necessarily involve the entire area associated with that particular reach of the stream. Therefore, an area is defined as one of unplanned flooding, which is flooding above the planned storage area. There will also be an area where flooding does not occur, defined as the non-flooded area. A variety of crop types from which can be selected the one providing the greatest benefit for each of the three hydrologically defined land use areas, is also available. These are implied land use decisions.

Goulter and Morgan (1983) also use the triangular hydrograph method to reduce computational problems, but add a refinement. Since depth and duration of flooding are considered factors in crop damage, and duration and depth are highly variable, especially in the planned storage area, the part of the hydrograph which constitutes the amount stored is divided into four parts of equal volume, which are used to calculate more precisely the depth and duration of flooding in the storage area.

The study by Goulter and Morgan uses the Wilson Creek Watershed to test the model. The present study is a refinement of that project, and uses the same watershed to demonstrate its application.

Chapter III

THE MODEL

The model used for evaluation of this system is a deterministic dynamic program which maximizes expected net benefits from crops grown in the flood plain and adjacent area. A dynamic program is particularly applicable to river systems where this kind of decision making is required, due to its sequential decision evaluating capability.

The dynamic programming model is applied in the following manner. The river is divided into a number of arbitrary reaches, which are discrete sections of the stream, and which together comprise the entire flood plain. Each reach has the potential to be a water storage area to varying degrees, depending on physical aspects of the reach. A decision is made for each reach of the stream involving the volume of flood water to be stored. Implicit within the decision is the most benefical crop type to grow in each area. The model is intended to determine 1) the most beneficial volume of water to store at each reach, and 2) the crops to grow in each area in order to provide the greatest net benefit. The net benefit calculations recognize the damage caused by the flooding of the crops, by either planned flooding (storage), or by unplanned flooding.

There are essentially three distinct features to the model: the dynamic program itself, the expected damage function, which is a part of the return function for the dynamic program, and the hydrograph routing algorithm. Each of these will be discussed in separate sections.

3.1 THE DYNAMIC PROGRAM

A dynamic program is used where optimal policy decisions made from a set of possible policy decisions, at various "locations" within the system are required. The "locations" may be in space or time, or any other quantity, depending on the definition of the problem. A series of terms unique to dynamic programming is necessary to define before discussing the dynamic program.

Dynamic Programming Terminology

Stage (n)	A specific segment or division of the entire problem which represents the "location" for which a set of policy decisions is evaluated, and an optimal policy decision made.
Policy Decision (x)	One of a set of specifically defined choices which is tested for its value according to the objective of the dynamic program.
State (s)	The possible state or condition that the system may be in at any stage. The state is a function of the cumulative policy decisions to the stage which the system is currently in.
Return Function	The function which evaluates the effect of a

policy decision (x) for a given state (s) at a specific stage (n), in accordance with the objective of the dynamic program.

Transformation Function	A function to transform the current state into an associated state at the next stage using the current state and policy decision.

3.1.1 General Description of a Dynamic Program

A dynamic program is a decision making model where the system under consideration is divided into a number of stages in time or space, in order to make a series of interrelated sequential decisions. Each stage has associated with it a state variable which defines the state or condition in which the system may find itself at a particular stage. Associated with the stage is a number of policy decisions, each of which is tested to determine its effect on the system in accordance with the objective of the model.

The combination of a decision together with a current state transforms the system, by means of a transformation function, into a new state associated with the next stage. The effect of the combination of state and policy decision is assessed by the return function of the model. The return function provides the means by which the decision, or set of decisions, is ranked with respect to other decisions, and defines the recursive nature of a dynamic program. The optimal policy for each state at each stage, is determined according to the return function. The system then moves into the next stage. The return function is unique to the dynamic program which it evalutes, but all follow a basic form.

The general form of a dynamic program is described mathematically (Hillier and Lieberman, 1980) as:

$$F_n^*(S_n) = \max \text{ or } \min \{F_n(S_n, X_n)\} \qquad (1)$$

Where: S_n = the current state in stage n

X_n = the policy decision

n = the current stage

$F_n^*(S_n)$ = the value of the optimal policy for the current state

$F_n(S_n, X_n)$ = a function of S, X_n, and $F_{n+1}^*(S)$

Where: $F_{n+1}^*(S_n)$ = the value of the optimal policy for the the previously evaluated stage in terms of the movement of calculation

$F_n(S_n, X_n)$ is calculated from a cost (or benefit) function which evaluates the effect of the combination of the state (S_n) and the decision or policy (X_n). This is added to the optimal value ($F_{n+1}(S_n)$), from the previous stage of calculation in state (S_n) from which the current state was derived.

Or: $$F_n(S_n, X_n) = CSX_n + F_{n+1}^*(S_n) \qquad (2)$$

Where: CSX_n = The cost (C) of the policy decision X_n at state S_n. This function is the return function of the dynamic program.

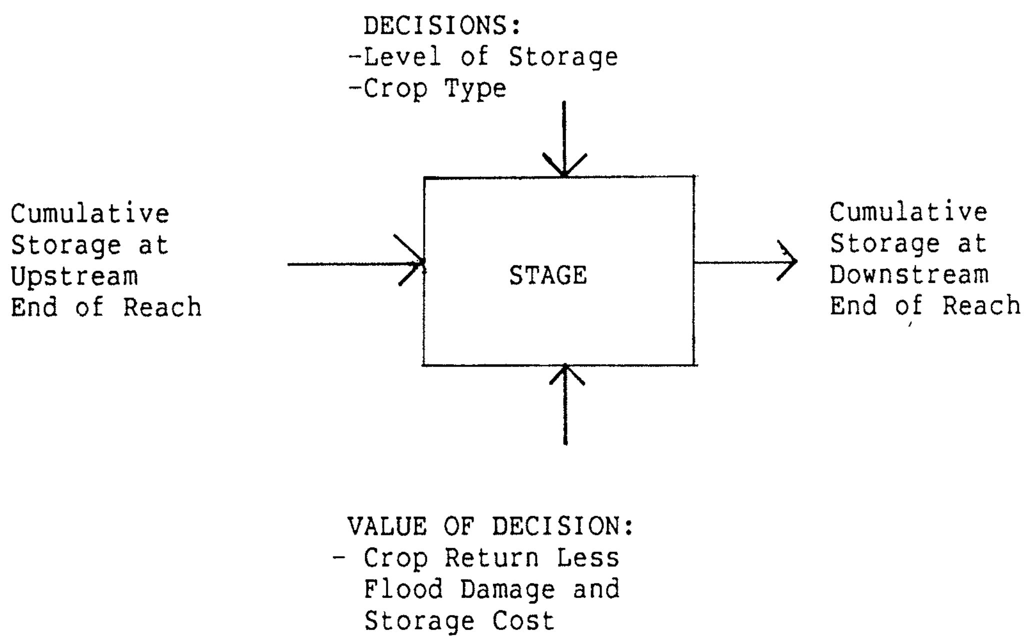
3.1.2 Specific Details of the Dynamic Program

The dynamic program developed in this thesis is based on the dynamic programming formulations of Hopkins et al. (1978,1981) and Goulter and Morgan (1983). For this dynamic program, the general dynamic program terms have the following specific definitions.

3.1.2.1 Stage

The stages in this dynamic program are reaches of the stream, incorporating the stream, the flood plain, and the adjacent area. Figure 2 shows a schematic representation of the stage with the associated inputs and outputs. At each stage a series of policy decisions are tested to determine the optimal policy at the stage, given the current state.

Figure 2: Stage Input and Output Diagram



In tabular form, the stage can be represented by a 'Stage/State Matrix' as in Table 1, with the columns representing decision variables, and the rows, state variables.

Each cell of the stage matrix is the value of a decision at the corresponding state.

TABLE 1

Stage/State Matrix

	Policy Decision Variables					
State	x=1	x=2	x=3	........x=X	$Fn^*(Xn)$	Xn^*
s=1	Fn(1,1)	....	....	....	Fn(1,x)	x
s=2	Fn(2,1)	....	....	....	Fn(2,x)	x
s=3	:	....	....	....	:	:
.	.	....	....	....	.	.
.	.	....	....	....	.	.
.	.	....	....	....	.	.
s=S	Fn(S,1)	....	....	........Fn(S,X)	Fn(S,X)	x

The optimal value of all decisions for a given state is listed in the Fn(Xn) column and the optimal decision is listed in the Xn column. The return function for the model developed in this thesis is detailed in section 3.4.

3.1.2.2 State Variable

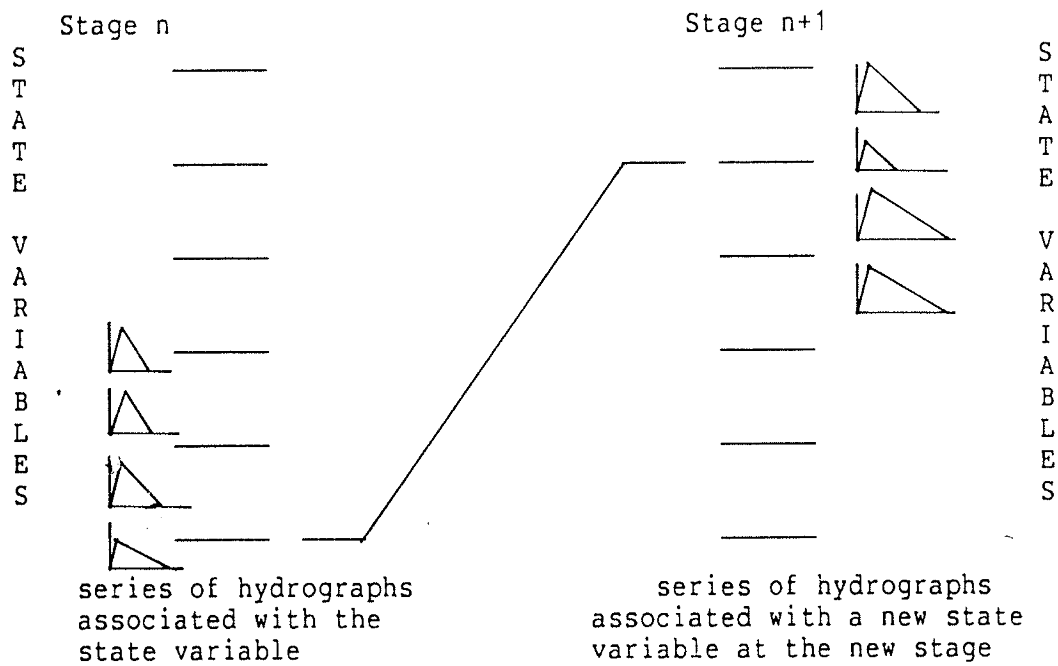
The state variable in this model is the volume of flood water previously stored upstream of the stage under consideration. Defining the state variable in this way allows for a direct relationship with the decision variable, which is the volume of water to be stored at any stage. The new state variable created at any stage is a function of the state variable for that stage, namely the cumulative upstream storage, and a decision (storage to be allocated) at the previous stage. The function which describes how the state and decision variables are combined is referred to as the transformation function, because it transforms the current state of the system into an associated state at the next stage. Figure 3 shows the transformation relationship between stages.

Since the objective of the model is to maximize net benefits, more than one flood hydrograph is required for evaluation. Therefore, each state variable has associated with it a series of hydrographs, each with its own probability of exceedence. In this respect, the model is similar to that developed by Goulter and Morgan (1983).

As the dynamic program progresses from one stage to the next, each of the hydrographs associated with the upstream cumulative storage levels is routed through the storage decision for the current stage, ie. it is transformed. The hydrograph is then stored with the appropriate cumulative storage variable at the downstream end of the stage. This also represents the upstream end of the following stage.

As this progression continues, each combination of state variable and decision variable creates a new state variable

Figure 3: Transformation Relationship Between Stages



*note: state variables are levels of cumulative storage

in the following stage. As the system progresses, this can cause the number of new state variables to become so large as to make the computational requirements impractical.

For example, if the number of initial decision variables is 6, then the number of state variables increases by a factor of approximately 6 at every new stage, except for the few occasions where more than one combination of state and decision result in exactly the same new state. After passing through only a few stages, the number of state variables becomes very large, resulting in an impractical computational burden. The number of state variables at any stage must therefore be limited. Each state variable is then representative of a range of possible state values, rather than an exact value.

Some precision is lost in using such a representative state variable. The degree of precision loss increases through the system toward the later stages. After a series of runs the results may indicate a problem with the state variable. In this case, the program can be rerun, in a similar manner to Discrete Differential Dynamic Programming (Chow et al., 1975). To do so, only the state ranges determined from the original optimal policy are used, and are divided into representative ranges. This can be continued until the actual optimal policy is converged upon. The results of the initial runs do in fact indicate that a problem exists, which may be a result of using the representative state variable. This will be discussed in the chapter dealing with the results.

3.1.2.3 Decision Variable

The decision variable is the volume of water to be stored at the current stage. The volume has a range from zero to a maximum imposed by the physical nature of the stage. Included in the decision is the crop type to be grown in the area. Since the area designated for flooding will be flooded more frequently, and to a greater extent for a longer duration, a crop which is least susceptible to flood damage should be the most beneficial one to be planted here. Other factors are included in the decision, however, which are crop values, soil quality, and frequency of flooding.

With a decision of the volume of water stored, the area designated to store it is defined. Implicitly, an area where water is not stored is also defined. This area has two parts: an area of unplanned flooding, and one which is free of flooding. The unplanned flooding area will be inundated when a flood larger than the cumulative storage capability up to and including this stage occurs. At stages where no storage is planned, for instance, especially in the upper stages, unplanned flooding is certain to occur. Each stage will have an area of no flooding, particulary if only the smaller floods are considered in the analysis. The size and existence of the non-flooded area is a function of the cumulative storage to this point. The crop allocated to this area will be that which has the highest return per hectare, since no damage is expected.

3.1.2.4 Decision Evaluation Process of the Model

The objective of this model is to maximize expected net benefits for the stage under consideration and all upstream stages. The return function calculates at each stage the expected net benefit for each storage decision for each upstream cumulative storage state. The expected net benefit is a function of crop type, crop value, area planted with the crop, soil fertility, and the expected damage from flooding, either by planned flooding (storage) or by unplanned flooding.

Figure 4 shows the decision evaluation process for a single decision assessment. This assessment is for a specific state at a given stage.

As a storage decision is chosen for evaluation, the hydrographs are modified by routing through that storage value. The flooded area is determined, along with the non-flooded area. The crop providing the best return for each area is determined, and the sum of the crop benefits for the three areas (planned flooding, unplanned flooding, and non-flooded) of the stage is the immediate or short term value of the decision.

The short term value of the decision is combined with the long range or cumulative return resulting from optimal upstream decisions and associated with the upstream cumulative storage state under consideration. For each cumulative storage level at the downstream end of the stage, the best combination of short term benefits, associated with the immediate decision, and long range returns, associated with the upstream cumulative storage state with which that decision is associated, is chosen. The routed hydrographs associated with the optimal upstream cumulative storage state and imme-

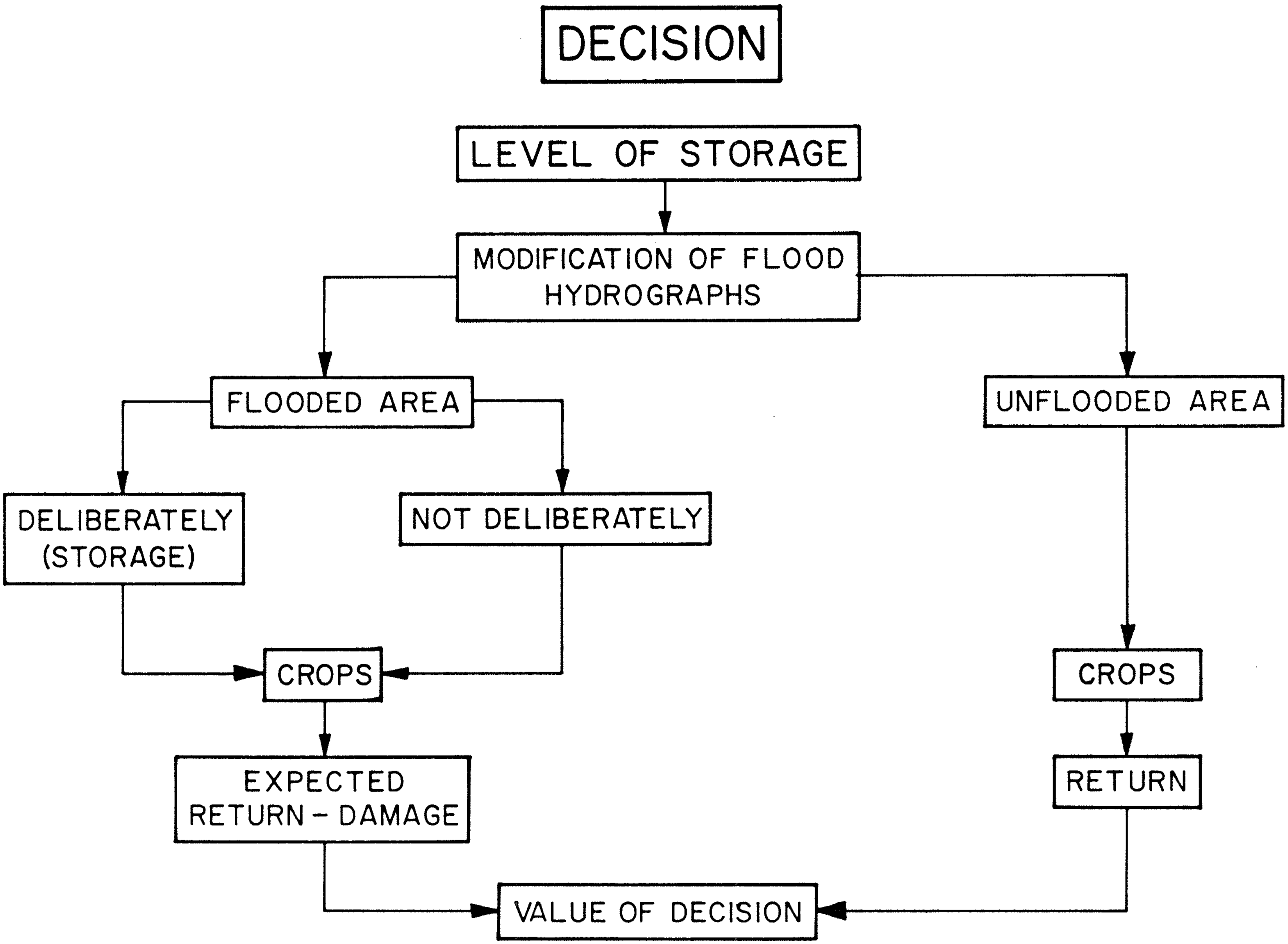


Figure 4: Decision Evaluation Process (Scource: Goulter and Morgan, 1983)

diate storage decision, are then stored with the particular downstream cumulative storage state under consideration.

3.2 THE EXPECTED DAMAGE FUNCTION

3.2.1 Theoretical Development

The traditional method of evaluating expected damages, for example that used by James and Lee (1971), is:

$$ED = V \sum_{x=1}^{X} \{ L(x)\ p(x) \} \tag{3}$$

Where : ED = the expected damage
V = the average value of the crop without flood damage ($)
L(x) = the proportion of the crop damaged as a result of a flood of magnitude x
p(x) = the probability of exceedence of a flood of magnitude x in a single flood season

This formulation provides the basis for the calculation of expected damages in the current study. It is modified, as described in the following sections, to include the refinements necessary to consider the range of conditions included in this model.

3.2.1.1 Variation of Flood Damage Susceptibility With Time

In order to incorporate the variation in flood damage susceptibility with time, two modifications to Equation 3 are required. The first is for the flood damage susceptibility itself, and the second is for the variation in the flood frequency curves over the growing season. The incorporation of these features is performed as follows.

The growing season is divided into separate periods. $L(x)$ is then replaced with another variable, $D(x,j)$, representing the proportion of flood damage caused by a flood of magnitude x in period j of the growing season. Equation (3) is changed to:

$$ED = V/N \sum_{j=1}^{J} \sum_{\substack{\text{flood} \\ \text{range}}} D(x,j)\ p(x) \tag{4}$$

Where: J = the number of periods in the growing season
j = the period in the growing season

The relationship in Equation (4) holds if the probabilities of exceedence of the floods are constant throughout the growing season, and all periods of the growing season are the same length. If the lengths of the periods are not constant, and / or the frequency curves vary, the $p(x)$ term in Equation (4) must be modified, resulting in:

$$ED = V \sum_{j=1}^{J} \sum_{\substack{\text{flood} \\ \text{range}}} D(x,j)\quad P(x,j) \tag{5}$$

Where: $P(x,j)$ = the probability of the exceedence of a flood of magnitude x, in period j.

The susceptibility to flood damage may have its own probability distribution, due to the variation in planting dates. This leads to the need to consider a joint probability distribution between that of the exceedence of a particular flood level and the plant being in a particular stage of growth. The incorporation of this issue requires the modification of Equation (5) to:

$$ED = V \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{\substack{\text{flood} \\ \text{range}}} D(x,k)\ Pj\ (x,k) \qquad (6)$$

Where: Pj (x,k) = the probability of exceedence of a flood of magnitude x, in period j, when the crop is in stage of growth k.

D(x,k) = the proportion of the crop damaged by a flood of magnitude x when the crop is in stage of growth k.

K = the number of stages of growth.

3.2.1.2 Multiple Damaging Floods in the Same Growing Season

The final modification to the expected damage function is the consideration of the possibility of the occurrence of more than one damaging flood in a single growing season. Including this feature requires the recognition that the damage resulting from a particular flood will vary depending on the amount of the crop already damaged by a previous event, or even by a series of previous events. The evaluation of expected damages must therefore consider both the change in damage susceptibility for second and subsequent floods, and the probabilities of the occurrence of previous floods over the range of flood magnitudes.

The actual incorporation of these two features is achieved through the use of conditional probabilities. To reduce the complexity of the problem, two assumptions were made. The first is that of independence of flood events in the same growing season. If independence exists, the conditional probability of the previous flood occuring, given the current flood under evaluation, is simply the probabilities of the two events. Independence was not tested for, but given the nature of the types of storms causing the floods, the assumption of independence is not unrealistic.

The second assumption is that there will be only a maximum of two events in a single season. In the region to which the model was applied, the probability of three events occuring will be very small, and when calculated as part of the conditional probability with two or more other floods, the individual effect of even a third flood on the entire calculation would be negligible. Therefore, this approach considers only the possibility of two floods in a single growing season. The inclusion of the possibility of two floods occurring modifies Equation 6 to:

$$ED = V \underbrace{\sum_{j=1}^{J} \sum_{\substack{\text{flood} \\ \text{range in} \\ \text{period } j}}}_{A} \underbrace{\sum_{l=1}^{j-1} \sum_{k=1}^{K} \sum_{\substack{\text{flood} \\ \text{range in} \\ \text{period } l}} D(x,k)\ Pl(x,k)}_{B} \qquad (7)$$

$$\underbrace{+\ D(x,k)\ \{ 1- \sum_{l=1}^{j-1} [\sum_{k=1}^{K} \sum_{\substack{\text{flood} \\ \text{range in} \\ \text{period } l}} D(x,k)\ Pl(x,k)]\}\ Pj(x,k)}_{C}$$

Where: $\sum_{l=1}^{j-1}$ = all periods previous to the period currently under assessment

Equation (7) is the sum of the cumulative previous flood expected damage and the expected damage caused by the current flood. It is the basis on which expected damage is calculated in this thesis. There are 3 identifiable parts to Equation (7). Firstly, expected damage from all previous floods is calculated by part B. Part C calculates expected damage caused by the current flood, given expected damage from previous floods. Parts B and C are summed, to give the damage level expected including current and previous damage, for the time period and flood level under consideration. The values are summed in part A over all flood levels and over all time periods, resulting in total expected damage.

3.2.2 Data Development for Practical Application

The return function of the dynamic program is a calculation of expected net benefits for the decision made for a particular state at any stage. Expected net benefit values are calculated for each crop type from the expectation of the return per hectare (in dollars), given expected flood damage. The value includes reductions due to the costs of production, and a factor that accounts for soil quality.

In this study, crop damage is calculated from damage due to duration of flooding only. Although it is reasonable to expect some damage owing to depth of flooding, no relationships between depth and damage for crops could be found or determined. James (1965) and Leyshon and Sheard (1974) both suggest that no relationship of this nature exists. Lacewell and Eidman (1972) do use a depth-damage relationship in their evaluation, but do not describe it in detail, or indicate its origin. While Goulter and Morgan (1983) use a depth-damage relationship, it is contrived solely for the purpose of testing that aspect of the model. It was therefore decided not to use a depth-damage relationship in the current study.

Four crops were used for the evaluation, wheat, barley, flax, and alfalfa. These crops were chosen on the recommendation of suitable crops for the region in Manitoba Department of Agriculture (1983).

3.2.2.1 Variation in Damage Susceptibility

Leyshon and Sheard (1974) developed a function relating duration of flooding to crop damage for barley (see Appendix A), for 21, 28, and 35 day old plants. These three points permitted the calculation of a piecewise linear relationship between the stage of growth and crop damage in terms of yield reduction. Since no other relationship was discovered for any of the other crops, the relationship developed for barley was also used for the other two grains. While it is likely that the duration-damage relationship for wheat and flax will be different, some form of useable relationship was necessary in order to properly assess the model. Substitution of more realistic values, if and when they become available, is very simple. Details of the calculations of the duration-damage functions are given in Appendix A.

The damage calculation for alfalfa was handled somewhat differently. Since the entire plant is valuable, rather than just the seed, and the damage to the grain crops is seed related, the assumption was made that no damage directly due to duration occurs. This assumption was made also by Goulter and Morgan (1983). However, some damage due indirectly to duration is possible in the form of reducing the probability of more than one cutting in a season. The plant will not grow when submerged, and will have to recover for a time after the flood has subsided, so time will be lost due to flooding. Since a long growing season is critical to getting two crops off the field, any reduction in the length of it will result in a reduced likelihood. This will have a tangible value in the expected damage evaluation.

Damage susceptibility is assumed to vary according to two parameters: crop type, and the stage of growth the plant is in. For each of the three grain crops, a joint probability matrix combining the probability of being in a particular time period with the probability of being in a particular stage of growth is developed. The value of a cell of the joint probability matrix is therefore the probability of being at the kth stage of growth in the jth time period.

For a known flood duration, damage due to flooding for each growth stage, in each time period is calculated. This is then multiplied by the joint probability of the plant being in that period and that stage of growth. The result is the duration-damage function. The details of the calculations of the joint probability matrices, and the matrices themselves are given in Appendix A.

3.2.2.2 Multiple Damaging Floods in the Same Growing Season

This feature is handled by assessing the expected damage from the flow level and time period currently under consideration, given the expected damage from all previous time periods. Since the storm events are independent, no conditional probability exists between the time periods in terms of the flood events. However, the damage itself at any time period is dependent on previous damage.

For simplification of calculation, it is assumed that only one flood will occur in the time period currently under consideration. This means, for instance, that if the first period is being assessed, then there will be no previous damage. Then, as the season progresses, the amount of expected previous damage will increase. At each time period, as each flood is assessed for damage capability, the expected amount of previous damage is included explicitly as part of the damage attributed to the current flood only, as well as the new value for total expected damage.

The necessity for this complication lies in the fact that the damage caused by a given flood depends on how much crop there is to damage. Damage tends to be measured in terms of reduced yield in the flooded area. If a previous flood has occurred, some of the crop is already damaged. Equation 8 is a simplified version of part of Equation 7 intended to illustrate this point.

$$\begin{array}{c}\text{EXPECTED}\\\text{TOTAL}\\\text{DAMAGE}\end{array} = \begin{array}{c}\text{Expected}\\\text{Damage from}\\\text{previous}\\\text{floods only}\end{array} + \begin{array}{l}\text{Expected}\\\text{Damage caused}\\\text{by the current}\\\text{flood given}\\\text{previous floods}\end{array} \qquad (8)$$

Where:

$$\begin{array}{c}\text{Expected}\\\text{Damage caused}\\\text{by the current}\\\text{flood given}\end{array} = \begin{array}{c}\text{Expected}\\\text{Damage from}\\\text{the current}\\\text{flood only}\end{array} * \quad 1 - \begin{array}{c}\text{Expected}\\\text{Damage caused}\\\text{by previous}\\\text{floods only}\end{array}$$

Equation (8), consists of two main parts, 1) damage caused by previous floods only and 2) damage caused by the current flood, given previous floods. These two parts refer directly to parts B and C of Equation (7), respectively. The latter of these is further divided into: 1) damage from the current flood only, and 2) 1 minus the damage caused by previous floods. Part 2) of this division is actually the expected amount of crop remaining after accounting for previous damage. The crop damage values are on a percentage basis, so it can be seen that 1 minus the percentage of crop damaged equals the percentage of crop remaining.

For instance, if an area has already been flooded, reducing the yield by 50%, and a flood of the same magnitude occurs again, the remainder of the crop will be reduced by 50%. The loss resulting from this second flood alone will be only 25%, but the total loss will be 75%.

3.3 THE DEVELOPMENT OF THE RETURN FUNCTION

As noted previously, the objective of the model is to maximize expected net benefits. The return function therefore includes expected damages as only part of the calculation. The expected damages are calculated as a percentage of the crop damaged, hence the expected net benefits are based on the percentage of crop remaining.

For the purpose of discussion of the expected damage function, it was necessary to include the flood probabilities with the damage calculations. In fact, the model is set up such that the benefits from a crop after a flood are calculated before the flood probability is included. Thus the net benefit equation used in the computer model is in fact:

$$NB(c) = DUR(x) * DDF(c) * V * SF * AREA * P(x,j) - SCOST \qquad (9)$$

Where:

NB(c) = Net benefit from crop (c)

DUR(x) = Duration of flooding from a flood of magnitude x .

DDF(c) = The duration-damage function for crop c , which includes the calculations with the joint probability matrices. The value that comes from this is a percentage of crop remaining.

V = Crop value per unit area

SF = A soil fertility factor, which is variable depending on the location

AREA = The area flooded, which could be planned or unplanned flooding, or even the area of no flooding, in which case there will be no damage, and DDF(c) will equal 1.0

P(x,j) = The probability of exceedence of a flood of magnitude x in period j .

SCOST = The cost of storage

This value is calculated for each crop type, for each flood magnitude, for each time period, and for the three types of "flood" area (planned, unplanned and not flooded). The NB(c) values are then summed over all flood area types, over all time periods, and over all flood magnitudes to give the actual expected net benefit value (ENB) for a particular crop. The crop which provides the greatest expected net benefit for the area is chosen.

The expected net benefit for a particular state (s) and decision (d) is then calculated by:

$$ENB(s,d) = \max(c) \text{ of }: \sum_{x=1}^{X} \sum_{j=1}^{J} DUR(x) * DDF(c) * V * SF * AREA * P(x,j) - SCOST \tag{10}$$

Where: ENB(s,d) = expected net benefit for cumulative storage state (s) and storage decision variable (d)

DUR(x) = duration of a flood of magnitude (x)

DDF(c) = duration - damage function function for crop (c)

J = all time periods

X = all flood levels

all other variables defined previously

Equation 10 calculates the expected net benefit for the state and decision variables currently being evaluated, with input from the current stage only. To this value is added the optimal value from the appropriate state from the previous stage. The final result is the expected net benefit for stage (n), for decision (d), when the system is in state (s), including the impact of the decision on the entire system to this point. Equation 11 shows the calculation of the final value. This is the calculation of one cell of the "stage matrix" as defined in section 3.1.2.1. The entire return is then defined as:

$$ENB(n,s,d) = ENB(s,d) + F^{*}_{n-1}(s) \qquad (11)$$

Where: ENB(n,s,d) = Expected net benefit for stage n, at cumulative storage state (s), and storage decision (d)

ENB(s,d) = expected net benefit for decision (d) in state (s) with input from stage n only

$F^{*}_{n-1}(s)$ = optimal result from stage n-1, at the state from which the current state in stage n is derived

The derived values for state (s) in stage (n) are maximized over all decisions (d), to arrive at the optimal decision for that state. This leads to the expression:

$$F^{*}_{n}(s) = \max_{d=1,D} \{ NB(j,x) + F^{*}_{n-1}(s) \} \qquad (12)$$

Where:

$F^{*}_{n}(s)$ = optimal value for the current state s and stage n

D = total number of decision variables ie. all levels of storage at the current stage

all other variables are defined previously

3.4 THE HYDROGRAPH ROUTING TECHNIQUE

In each of the stages in this system, there are several states and storage decision variables. To calculate expected benefits a number of floods must be evaluated. During the evaluation, each flood hydrograph must pass through a routing procedure once for each hydrograph for each state and decision variable combination, resulting in a large number of passes through the routing procedure. For example, if there are 5 hydrographs used for evaluation, with 6 states and 6 storage decisions, 5*6*6 = 180 passes through the routing procedure will be required. This makes it important to have a routing procedure which is simple in terms of the number of variables required to describe it, and in terms of computational efficiency. The basic triangular hydrograph as used by Hopkins et al. (1981) and modified by Goulter and Morgan (1983) is used in this study. It has, however, been further modified and refined to more precisely define the movement of the flood through the channel and the potential storage areas.

3.4.1 The Hydrographs

In order to develop the synthetic hydrographs, an attempt was made at finding a correlation between the peak flows and the time elements of the rising and descending limbs (Graham, personal communication) The correlations determined were poor, but were used to establish the required values.

Five hydrographs of annual return periods of 5, 10, 20, 40 and 50 years were used to be representative of the entire range of flood probabilities. Details of the development of the hydrographs are given in Appendix B.

3.4.2 The Routing Procedure

The routing procedure does not allow for local input from runoff or groundwater flow. The size and shape of the basin, and the short duration of the floods makes the extra computational effort required unwarranted. It is assumed, then, that if no storage decision is made at a stage, the outflow hydrograph from the stage is identical to the inflow hydrograph to the stage. The only time the hydrograph changes is when a storage decision is made.

The triangular hydrograph, if the flow level just prior to the storm is made equal to zero, requires only 4 values to describe it: the peak flow, time of peak, bankfull flow, and the time of the end of the recession limb. All other values required for routing can be calculated within the procedure. Details of the routing procedure and the resulting hydrograph shapes are given in Appendix B.

As the hydrograph moves downstream, its shape changes whenever a decision to store water is made. The theory of dynamic programming dictates that the state that the system is currently in must be independent of how that state at that stage was arrived at. This means that the state at any point must not vary with the route taken to arrive at that state. Therefore, the shape of the hydrograph at any point must be the same, whether it arrived at that shape through only one larger storage decision, or through a series of smaller ones. The routing technique developed in this thesis was tested for adherence to the dictate by a series of trials, and was found to comply.

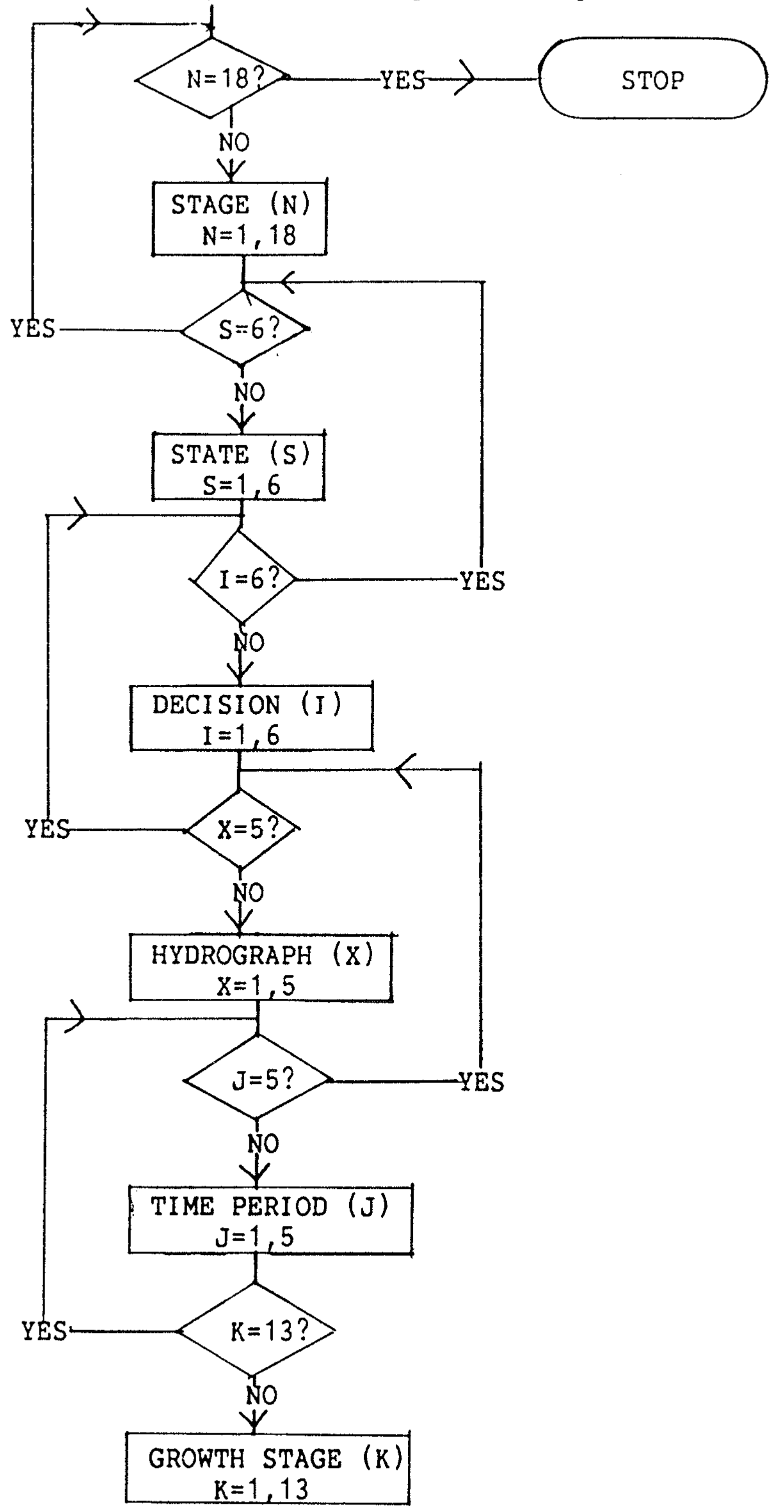
3.5 THE RECURSIVE PROCESS

Figure 5 shows the levels at which the model loops within itself in calculating the optimal values and decisions for the entire stream system.

As a stage is entered, calculation of the expected net benefits for the first combination of state variable (cumulative storage) and policy decision (volume of water to be stored at this point) is begun. For the series of hydrographs, durations of flooding for the planned flooding area and the unplanned flooding area are calculated, along with the respective areas of inundation, plus the area of the non-flooded section.

The area and duration of inundation in the planned flooding section is determined by discretizing the storage at that stage into four parts of equal storage volume, as detailed in Appendix B. For each of the 4 sections of the

Figure 5: Looping Diagram of the Dynamic Program



planned flooding area, and for the unplanned flooding area, the damage due to duration of inundation for the current hydrograph is calculated. The joint probability matrices (the probability of the crop being in growth stage k during time period j) are used in these evaluations.

From the duration - damage value, the net benefit for each crop is calculated for the planned and unplanned flooding areas. At this point, the net benefits for the non-flooded area are also calculated. The net benefit values for each area and crop are sorted to determine the most suitable crop for each of the three areas, and the values are summed over the areas, resulting in the net benefit value for the stage as a whole.

The probability levels for the hydrographs are then included in the calculation to determine expected net benefits. Given the current storage volume decision, the cost of storage is subtracted from the expected net benefits. The result is the value of the current decision at this cumulative storage state, for this stage only. To this, the optimal (maximum) value of the storage decision from the appropriate state in the previous stage is added. The result is the final, total value of the decision, including the upstream effect from the previous stages.

It is this interaction with the state(s) from previous stages which defines the recursive nature of a dynamic program. The value calculated here is the expected net benefit for the entire system to this point, given the current state, for this policy decision.

Within the same stage, and in the same state, all policy decisions are tested in the same manner as detailed above. Once all expected net benefits are calculated, they are sorted to find the maximum value. The maximum expected net benefit, and the policy decision for which it was derived, are stored for reference at the next stage.

The system then transfers to the next stage, and the process is repeated through all decisions variables, flood levels, etc.. Once all stages are evaluated, the traceback calculation determines the optimal "path", which is the optimal level of storage (for the system as a whole) at each stage, and the total expected net benefit derived.

The above describes the final equation:

$$MAX = \sum_{n=1}^{N} Fn(s)^{*} \qquad (13)$$

Where: MAX = maximum expected net benefit for the watershed

N = number of stages

$Fn(s)^{*}$ = optimal value of decision from each stage along the optimal "path"

Chapter IV

MODEL APPLICATION

This chapter outlines the application of the model to the Wilson Creek Watershed. Included in the discussion are physical aspects of the basin, crop value determination, dyke volumes and costs, time period determination and the flood probability levels.

Some assumptions pertaining to the input data were necessary in order to properly run and test the model. Therefore, this model has no operational significance in its present state. The assumptions are discussed in detail where appropriate.

4.1 THE STAGES

4.1.1 General Description

The Wilson Creek watershed was used for the evaluation of the model because it is of a type considered appropriate for analysis by this method. It is a small watershed (30 km^2) with no significant tributaries in the agricultural portion of the watershed, and has a predominantly agricultural economic base. Wilson Creek is typical of the watersheds which lie on the eastern face of the Manitoba Escarpment. The land is not prime farmland, and within the flood plain, shale deposits from flooding have reduced the land quality further.

The creek has its source at the top of the Manitoba Escarpment, some 450 m above the relatively flat farmland which makes up most of the flood plain. The part of the basin which lies on the escarpment is very steep, causing rapid flow response to storms. The change from the steep escarpment to the flat farmland is abrupt, contributing to the flood potential of the creek in the lower areas.

The lower part of the basin has been altered for drainage purposes. Before alteration, Wilson Creek emptied into a swamp at the foot of the escarpment. To increase the arable area, the swamp was drained by means of a narrow channel running due east to the Turtle River (See Figure 6). It is this channel, now included as part of Wilson Creek, which is the section of the stream of greatest importance to this study. Most of the farm land is located here, and it is this stretch of the creek that is most susceptible to flooding.

4.1.2 Specific Stage Descriptions

The basin is divided into 18 stages of varying dimension (see figure 6). Most stages are 81.67 hectares, approximating a quarter section (stages 7 through 17). Stage 18 is slightly larger, 97.68 hectares, to accomodate Wilson Creek turning northeast as it meets the Turtle River. Stage 1, which is 1200 hectares in size, is completely within Riding Mountain National Park. Since no storage would be allowed there, no decision is made at this stage, and it is only used to generate the hydrograph. Stages 2 through 6 vary in size, 248.9, 237.75, 101.36, 122.5, and 83.89 hectares respectively, due to variation in the physical characteristics of the watershed where the escarpment meets the plain.

At the base of the escarpment, the abrupt change in slope has resulted in a deep, wide channel cut into the soft shale of the alluvial fan. This channel has a cross-sectional area of as much as 4491 m^2, and runs through stages 2 and 3. The channel is large enough to store most of the flood water from all but the larger events, but the cost of a dam across the channel or any other structure capable of holding this volume of water is likely to be prohibitive. It was decided that for the preliminary model assessments, stages 2 and 3 would have no storage capacity until an approximate value for the cost of storage elsewhere could be determined.

At stage 18 there is no storage because there is no downstream benefit, yet there would be a storage cost and a loss of crop value in the storage area. This stage will have the potential for reduction in benefits due to unplanned flooding, depending on the decisions made upstream. For the preliminary investigation, this leaves stages 4 through 17 with potential for flood water storage.

For each stage, relationships between channel cross sections and flow volumes were calculated using channel cross

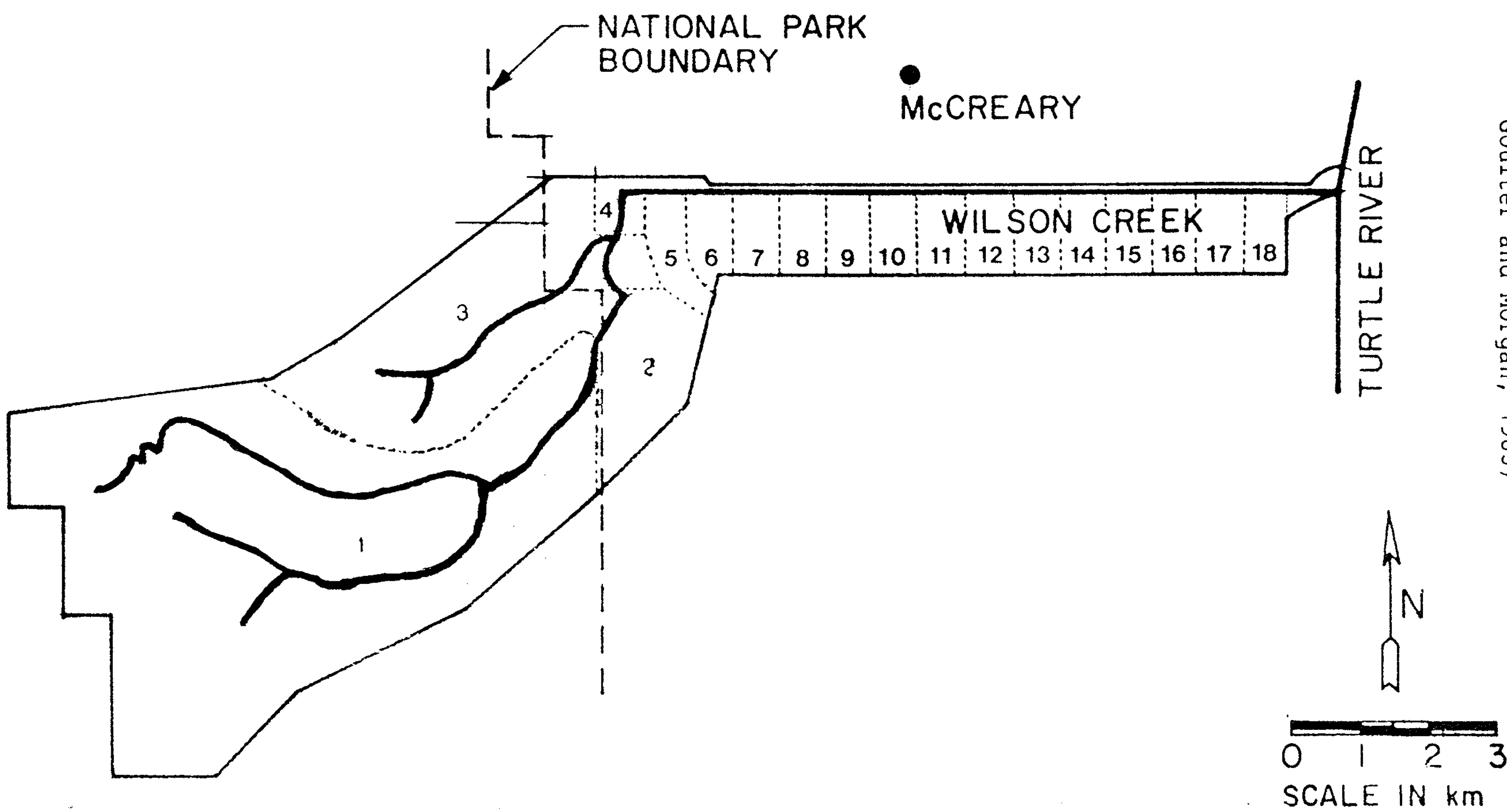


Figure 6: Schematic of Wilson Creek Watershed (Scource: Goulter and Morgan, 1983)

sections taken every 200 m. by the Manitoba Department of Natural Resources, Water Resources Branch, and using the standard Manning equation approach for calculation. Details of calculation are in Appendix C. These values were used to determine channel capacity.

Using streambed profiles and cross sections developed from aerial photographs and topographic maps, curves relating area flooded to storage volume, including a storage maximum, were determined for each stage. Details of the calculations are in Appendix C.

Soil productivity and crop returns from each soil type were obtained from the Manitoba Crop Insurance Corporation. Table 2 shows each stage, its size, maximum storage capacity, bankfull value and soil productivity value.

TABLE 2

Physical Descriptions of Stages

Stage	Size (hect.)	Bankfull Value (m^3/sec)	Storage Maximum (m^3)	Soil Productivity Index wheat (% of average yield)	barley	flax
1	1200	*	0.0	*	*	*
2	248.90	*	239250.	84	84	73
3	237.75	*	88150.	99	100	97
4	101.36	10.76	40000.	91	90	84
5	122.50	10.76	117800.	99	100	97
6	83.89	10.76	60000.	99	100	97
7	81.67	10.76	175175.	99	100	97
8	81.67	10.76	345950.	99	100	97
9	81.67	10.76	206790.	99	100	97
10	81.67	10.76	91200.	94	95	95
11	81.67	10.76	103845.	91	90	84
12	81.67	10.76	560000.	91	90	84
13	81.67	10.76	548000.	91	90	84
14	81.67	10.76	121900.	91	90	84
15	81.67	10.76	276000.	91	90	84
16	81.67	10.76	529000.	86	84	81
17	81.67	10.76	437000.	86	84	81
18	97.68	10.76	*	86	84	81

(* = no value calculated as none is required)

4.2 EXPECTED NET BENEFIT CALCULATIONS

4.2.1 Crop Values

Crop values in 1983 dollars per tonne, were taken from Manitoba Department of Agriculture (1983). Using average yield information for each crop a value of return per hectare was developed. Start-up costs were removed from this value leaving a net return for each crop type. These calculations are found in Appendix D. Table 3 shows these values.

TABLE 3

Crop Values

Crop	Gross Value ($/tonne)	Yield (tonne/hect)	Gross Return ($/hect)	Start Cost ($/hect)	Net Return ($/hect)
wheat	205.00	1.747	358.14	125.33	232.81
barley	180.00	2.218	399.24	118.31	280.93
flax	382.00	0.852	325.46	117.42	208.04
alfalfa	30.04	4.950	148.67	61.75	86.92

(note: Alfalfa values are estimated and based on one crop per season only. The possibility of two crops is included as part of the benefit function within the program.)

4.2.2 Dyke Volumes

On Wilson Creek, as well as on the other streams on the escarpment, each flood causes shale to be deposited in the flood plain as well as in the channel itself. Frequently the deposits are removed and piled along the north bank. This, along with the original channel excavation, has effectively resulted in a dyke paralleling the stream on the north side. In this study, for storage purposes, dykes perpendicular to the flow will be used at the stages where storage is desired. The combination of these dykes with the parallel dyke, in conjunction with the rise in elevation toward the south west and upstream from the storage location for each stage, form the reservoir boundaries.

The parallel dyke is not part of the decision process, because it already exists. Instead, it forms part of the restriction on the maximum storage for each stage. Adding to the height of this dyke in order to store more water is possible, but for this study only the height of the perpendicular dyke is variable.

A detailed study for the proper dyke dimensions was not carried out. A dyke shape is assumed only for calculation of the dyke volume, from which to calculate costs. The dyke cost is used as the cost of storage for the decision under assessment. From the assumed dyke shape and dimensions, an equation relating dyke height and dyke volume was determined. At each stage a relationship between storage volume and dyke height was determined, leading to a storage volume-dyke volume equation. Dyke costs are on a dollars per volume basis, making it easy to relate storage decisions to the cost of that storage. Details of the dyke calculations are found in Appendix F.

4.2.3 Time Periods

The total period of analysis was assumed to be the 153 days from May 1st to September 30th. The expected growing season is defined as the period from the average latest spring frost date to the average earliest fall frost date. These dates are May 16 and September 15, respectively. The choice of May 1 to September 30 as the period of study includes these dates with an approximately equal margin on each side of the expected growing season.

The growing season was divided into 5 periods of unequal lengths, based on growing periods of individual crops, average planting dates, earliest planting dates, and latest planting dates.

The first period runs from May 1 to May 19. During this time, wheat and barley can be reseeded if damaged by flooding. Flax is not normally planted until after May 19, and therefore if flax was planned for an area, risk of flood damage is virtually zero. Restarting alfalfa, which may or may not actually include seeding, is also possible.

The second period is from May 20 to June 20. Any crop can be reseeded up to June 20 and flax will be planted during this period. After June 20, reseeding of grain crops is not possible as there is not enough of the growing season remaining to mature a crop. Therefore, any losses due to flooding will be accepted as unrecoverable.

Alfalfa has a growing season of 60 days, making July 17 the latest date to begin a second crop. Thus June 21 and July 17 form the boundaries of the third period.

From July 18 to August 31, any damage to any crop is nonrecoverable. After August 31, crops may be mature. They may be cut and lying in the field, awaiting pick-up. If flooding occurs at this time, the crop will be more severely damaged. These dates therefore, are the boundaries of the fourth (July 18 to August 31) and fifth (September 1 to September 30) time periods.

4.3 FLOOD PROBABILITY LEVELS

4.3.1 Frequency Analysis

Frequency analyses were conducted for annual peaks, as well as for peaks for each of the five time periods of the growing season. The calculations for the frequency analyses are in Appendix F. Five hydrographs were used for evaluation of the model, which were derived from the analysis of annual peaks. The 5, 10, 20, 40 and 50 year floods were used. The flow levels for each of these were then used to determine the probabilities of exceedence of the floods in each of the five time periods. Table 4 shows the five annual return periods, and their respective probabilities in each of the five time periods.

TABLE 4

Flood Probability Levels

Annual Return Period	Peak Flow	Probability of Exceedence	Time Period Probabilities of Exceedence				
(yrs.)	(m^3/s)		Period 1	Period 2	Period 3	Period 4	Period 5
5	10.8	0.2	0.05	0.05	0.025	0.017	0.013
10	20.0	0.1	0.013	0.015	0.009	0.007	0.005
20	32.0	0.05	0.005	0.007	0.004	0.004	0.002
40	45.0	0.025	0.002	0.003	0.002	0.002	0.001
50	55.0	0.020	0.001	0.002	0.001	0.001	0.00008

As discussed earlier, independence must exist between the five time periods in terms of the frequency curves. Independence has been assumed for this study. This assumption is reasonable because the storms which cause the floods tend to be thunderstorms, which are usually not interrelated. Preliminary analysis of the flow records indicated that independence is likely to exist, as flows returned to the normal lower flow level after each event. No flow overlap from one event to the next was found.

Independence is necessary because conditional probabilities may become involved otherwise. If independence exists, the "conditional" probability between the events is non-existant because there is no intersection of their probability functions. Therefore the effect of a conditional probability need not be involved in the calculations. The probability of one event occurring, given the occurrence of another, is merely the product of the probabilities of each of the events. If the events are not independent, then the calculation of the conditional probabilities is more difficult. It does not, however, preclude the use of this kind of assessment. For this study it was decided, therefore, to assume independence.

4.3.2 Probability Levels Used For Model Input

An expected damage function relating damage from each flood level and the probability of exceedence of the levels must cover the entire spectrum of flood magnitudes. To reduce computational burden, only five flood levels were used for this study. These five levels must therefore be representative of the entire spectrum. The probability associated with each flood level is actually a range or band of probability levels with the calculated probability of exceedence at the approximate center. Table 5 shows these representative levels. During analysis, the probability used is the probability of a flood occuring between two flood levels, rather than the probability of exceedence.

The sum of the values of the probability bands must equal 1.0 in order to represent the full spectrum of flood events. Therefore, not only the bands must be calculated, but also the ranges below the smallest flood, and above the largest. The probability bands were calculated by determining the center point between each probability of exceedence level. The band extends from one center point to the next. At the low end of the curve, the flow level below which no damage can occur is known, (channel capacity). The probability level of this can be easily determined. In this case the value is 1.21, as can be seen in Table 5, for the "low end of frequency curve" value. This value is actually the probability

TABLE 5

Probability Bands

Peak Flow (m/s)	Range (Band) of Probability Period 1	Period 2	Period 3	Period 4	Period 5
10.8	0.5785	0.4675	0.2930	0.1980	0.2110
20.0	0.0225	0.0215	0.0105	0.0065	0.0055
32.0	0.0055	0.0060	0.0035	0.0025	0.0020
45.0	0.0020	0.0025	0.0015	0.0015	0.0006
55.0	0.0010	0.0010	0.0010	0.0010	0.0002
low end of frequency curve p(x)= 1.21 damage = 0%	0.3900	0.5000	0.6900	0.7900	0.7800
top end of frequency curve p(x)=inf damage = 100%	0.0005	0.0015	0.0005	0.0005	0.0007

of a flow of less than channel capacity occuring. Although this is rather a large probability value, no damage occurs below this flow level, resulting in an expected damage value of 0.00.

At the top end of the scale, (the top end of frequency curve value of infinity in Table 5) the band width is calculated such that the probability of exceedence of the largest flow (55.0 m^3/s) is the center point of the range. Note that, due to the nature of this type of banding, most probability of exceedence levels are not actually center points of the bands. Above this top band, the probability of exceedence is 1.0 minus the sum of all of the rest of the bands, including the value from the lower end. As can be seen on the above table, the probability is small, but the damage is assumed to be 100% .

In the course of the calculation of expected damages, the lower end value will not be used since no damage is attributed to it. This will leave six probability bands representing five hydrographs during the analysis. The probability bands will be combined with the joint probability matrices for the crops, to determine expected net benefits.

Chapter V

DISCUSSION OF RESULTS

This chapter is a discussion of the results of the model tests. Two types of decision are made by the model, 1) the volume of water to store at each stage, and 2) the type of crop to grow in each area. These decisions are discussed separately.

Ordinal numbers ranging from 1 to 6 are used to represent storage volume decisions and state variables, for easy reference during the discussion. The number 1 represents a zero storage decision or storage state. The other numbers (2 to 6) represent cumulatively another one fifth of the maximum storage capability for the stage or one fifth of the total cumulative storage state, depending on the context in which it is used. For example, when refering to the decision value of 2, this means that the decision is to store one fifth of the maximum capacity. If the number is 3, two fifths are being stored. When referring to the state variable, a 1 indicates that there is no previous storage, and 6 means that all the water that it is possible to store to this stage has been stored.

5.1 STORAGE VOLUME DECISIONS

One of the original expectations of this study is that the cost of a flood mitigation system on a small rural watershed, such as the Wilson Creek watershed, will be the critical aspect of the feasibility of the system. Because the cost of the dyke used for storage is the index of storage costs, the cost per volume of the dyke is varied. A value between \$1.00 and \$2.00 per m^3 for the cost of the dyke is reasonable, based on current cost information. Therefore \$2.00 has been used as a starting point. The cost was amortised over 50 years, at an interest rate of 8%. No operation and maintenance costs were added, as these costs would be limited if they existed at all.

At a dyke cost of \$2.00/m^3, three storage sites are chosen, as seen in Table 6.

TABLE 6

Optimal Path at \$2.00/m^3

Stage	Decision Number	Decision (m^3)	Dyke Cost (\$/m^3)
5	3	50000	1960.00
8	5	280000	7200.00
11	6	103845	400.00
total		433845	9560.00

ENB = \$623865

This result shows that the model chooses to store at the areas which have the lowest cost of storage, in terms of the storage volume capable of being stored with one unit of dyke volume. Table 7 lists the storage volume per dyke volume values for each stage.

TABLE 7

Dyke Volume per Storage Volume Ratios

Stage	Average Dyke Volume Per Storage Volume Ratios
4	1:7
5	1:44
6	1:24
7	1:45
8	1:67
9	1:17
10	1:12
11	1:459
12	1:36
13	1:33
14	1:15
15	1:22
16	1:44
17	1:34

Table 7 shows the dyke volume - storage volume ratios for each stage. A value of 1:67, for instance, means that 67 m^3 of water can be stored for each 1 m^3 of dyke. Because the cost of the dyke is the cost of the storage, as defined in this model, this ratio is an index of the cost of storage. The larger the number (ie. the denominator of the ratio) the cheaper the storage is, as a larger value means greater storage for the same dyke volume, and thus the same dyke cost.

At stage 11, the model chooses to store the maximum. Stage 11 has by far the best dyke volume - storage volume ratio (DSR), so it is chosen for storage of as much water as possible at that location. Stage 8 is also used for storage, as it has the second best DSR. However, it is not used to capacity before stage 5 is also chosen for storage. The model is trading off among factors other than just the cost of storage.

To investigate the trade offs between stages 5 and 8, the cost per dyke volume was increased to a point at which the model would choose one stage over the other. This occurred at $2.20/$m^3$, where stage 8 is chosen to store 140000m^3 (decision 3). At this point, no water is stored at stage 5. Here, the DSR has a greater effect on the decision than any other factor or combination of factors. However, as the cost is reduced toward $2.00 again, some storage is allocated to stage 5, and the storage volume at stage 5 continually increases, while stage 8 maintains the same volume until the value is at $2.00 again. It appears that there is some reluctance for stage 8 to store the maximum.

The DSR, although a comparison of volumes, is also an indication of the cost of storage at any location. Since the dyke volume translates directly into the dollar cost of storage, this ratio is also representative of a cost value. It is not unrealistic, therefore, to expect the model to select those locations possessing the best DSR values. The fact that storage is alloted to stage 5 before stage 8 is at capacity indicates that there is more to the decision than the DSR, or specific cost of storage value.

There are two other very important factors involved in the decision process: the soil quality at a storage location, and the location of the stage in terms of its relative upstream or downstream position. The poorer the quality of soil, the lower the crop value; and the lower the crop value, the lower the amount of damage that can be imposed on it. Generally, the soil quality decreases downstream (see Appendix C), so there is some tendancy for the model to store further downstream, on that basis alone.

At the same time, there is also a tendancy to store further upstream. The more water that is stored, as far upstream as possible, the greater the flood damage reduction downstream. There are two reasons for this. Storage upstream reduces the intensity of flooding at every downstream location, so the further upstream the storage is, the greater the number of reduced flooding locations. The farther downstream the model chooses as a storage location, the more stages there are which will be flooded as the flood passes through them unaltered. Secondly, the more storage there is upstream, the greater the chance of reducing the probability of flooding to a lower level, or even to prevent flooding entirely, at the more downstream stages.

There is another aspect which should have an effect on the storage decision, but it is very difficult to quantify. At each stage there is a unique relationship between the volume of storage and the area flooded by this volume. This is also true for the area flooded in unplanned flooding situations. On the surface it would appear that it would be more beneficial to store at those locations which flood the least area for a given volume. However, an increase in the area flooded by a given volume reduces the duration of flooding over that same area, and since the damage due to duration of flooding is the predominant factor in flood damages in this model, these two aspects may trade off to the point of nullifying their respective contributions.

Further runs were done to investigate the decision making capability of the model. Table 8 shows the decisions made for each stage for a number of dyke cost values. As discussed previously, the ordinal number values are the decisions, and represent a fraction of the total storage capacity of each stage. The maximum storage values vary from one stage to the next. Table 19, in Appendix C shows the actual storage volumes which these numbers represent. Table 9 shows the totals of the storage volumes chosen at each level of dyke cost, and the corresponding total dyke costs (non - amortised) and the total expected net benefit.

TABLE 8

Storage Decisions With Variation in Cost per Dyke Volume

Cost Per Dyke Vol ($\$/m^3$)	Stage													
	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2.00	1	3	1	1	5	1	1	6	1	1	1	1	1	1
1.80	1	3	1	1	5	1	1	6	1	1	1	1	4	1
1.50	1	3	1	2	5	1	1	6	1	1	1	1	4	1
1.40	1	4	1	6	4	1	1	6	1	1	6	1	4	1
1.30	1	4	2	6	4	1	1	6	3	1	5	1	3	1
1.10	1	4	2	6	4	1	1	6	3	2	5	1	4	1
1.00	1	4	2	6	4	1	1	6	4	1	5	1	4	1
0.75	1	4	6	6	4	2	1	6	4	1	5	6	5	1
0.50	1	6	6	6	6	1	6	6	4	1	1	6	5	1
0.30	1	6	6	6	6	6	6	6	6	5	6	6	5	6
0.25	1	6	6	6	6	6	6	6	6	6	6	6	6	6
0.10	6	6	6	6	6	6	6	6	6	6	6	6	6	6

Note: the values in the above are decision variable ordinal numbers (1-6) and represent actual storage decision values. Each increment is an increase in the storage volume by 1/5 of the maximum storage volume for the stage. For example, decision variable 2 is 1/3 of the maximum volume allowable at the particular stage.

Tables 8 and 9 show two important general patterns. The first is that as the cost per dyke volume is decreased, the total storage volume is increased. Also, the storage volume at any stage generally increases as the cost is reduced, but there are exceptions. Secondly, the upstream stages approach their storage maximums faster or earlier than do the stages further downstream. Again, this is a generality, and there are exceptions. It is these exceptions which indicate how the model is working.

TABLE 9

Actual Storage Volumes Chosen, With Dyke Costs and Total Expected Net Benefits

Cost Per Dyke Vol ($\$/m^3$)	Total Storage Volume (m^3)	Total Dyke Cost ($)	Total Expected Net Benefit ($)
2.00	433845	9560	623865
1.80	763845	21564	623922
1.50	798845	19300	624007
1.40	1015920	30006	624051
1.30	1120020	31920	624339
1.10	1340020	35860	624456
1.00	1342020	29800	624573
0.75	1816020	34268	625090
0.50	1945970	23475	625894
0.30	3375660	27045	626829
0.25	3572660	23038	627152
0.10	3612660	9615	628295

On Table 8, at the value of $2.00 the model trade offs are apparent. Stage 11 is chosen for the obvious advantage in its DSR. Stage 8 is also chosen on this basis. Stage 5, however, does not have the next best DSR value, although it is very close. Stages 5, 7, and 16 have similar values, (44, 45 and 44, respectively). It is more advantageous to choose stage 5 over 16, since stage 5 has the upstream advantage by a large margin. Stage 16 does have a substantially poorer soil type, (barley production is 16% lower at stage 16 than at 5) but it is not enough to counteract the upstream advantage. The choice of stage 5 over stage 7 must also be due to the fact that stage 5 is further upstream. There is almost no difference in the DSR; stage 7 is 45, and stage 5 is

44. The fact that the DSR for stage 7 is larger indicates that another variable is influencing the decision. Stages 5 and 7 also have the same soil type, so no decision is based on that factor.

The trade off between DSR and location may influence the decision to store some water at stage 5 before storing the maximum at stage 8. It is difficult to believe that this is the only contributing factor, as the discrepancy in DSR values between the two stages (44 at stage 5 and 67 at stage 8) is large, compared to the relatively small distance between the two stages, which also have the same soil type. There may be an influence from the area flooded for this volume. The effect of the difference in area flooded is not fully understood. It should be noted, however, that at a storage level of $50000m^3$, (the decision for stage 5) 18 hectares are flooded at stage 8 where only 13 hectares are flooded at stage 5. This may, in combination with the advantage of the upstream location of stage 5, be enough to result in the decision.

Stages 8, 12, 16 and 17 have the largest area flooded per storage volume among all the stages, and are much larger than any of the others. Throughout the model evaluation, these stages, with the exception of 17, are added to the optimal path at fairly high cost levels, but do not store at their maximums until the cost has been substantially reduced. The area flooded per volume also increases in these stages, as it does in all stages, as the storage level increases, but the effect is much greater where the potential for storage is very high. Again, it is apparent that there is some influence on the storage decision from the area flooded - storage volume relationships at each stage.

The reduction in dyke volume cost to $1.80 results in stage 16 being included in the optimal path, storing 330000m^3. This choice is not an obvious one, as stage 7 has a slightly larger DSR, and a more advantageous position. The only obvious reason for the choice of 16 is its poor soil quality. Investigation of the area - volume relationships shows that the storage of 175175 m^3 (the capacity of stage 7) at 16 floods an area of 70 hectares, compared with 31 hectares at stage 7. This is contradictory to the argument used comparing stages 8 and 5 in the discussion of the run at $2.00. The difference may lie in the actual volumes stored.

In the discussion of the trade offs between stages 5 and 8, the area flooded - storage volume relationship seemed to be influencing the decision to store at stage 5, rather than stage 8. The storage decision was only 50000 m^3, compared with 300000 m^3 under consideration in the decision to store at stage 16. Because 50000 m^3 is a relatively small volume, the duration of flooding will be short, regardless of the area - storage relationship. Damage due to flooding duration does not become extensive until the duration is quite long.

Therefore, when a small volume is under consideration, the difference in the area flooded between two stages may have a greater influence on the decision than the difference in duration of flooding does. With large storage decisions, as in the case of storing 300000 m^3 at stage 16, the durations of flooding are much longer, and cause more damage per unit time than do short durations. This may result in the duration of flooding being more critical to the storage decision than the area flooded is, for large volume decisions.

Reducing the dyke volume cost to $1.50 causes stage 7 to be included in the optimal path at 35000 m^3. Although this is a very low volume, it is only a step toward full capacity storage which occurs by reducing the cost by only $0.10, as is seen at the $1.40 level. Stage 7 is an obvious choice, as it has one of the better DSR values, and the best one of the stages yet to be chosen for storage. When the cost per dyke volume is reduced to $1.40, stage 7 moves to capacity storage. Being among the "cheapest" places to store, this is not unrealistic. The storage level at stage 5 is also increased, to 75000 m^3. At the same time, though, stage 8 is reduced by 70000 m^3 and stage 14 is included, at its maximum value of 121900 m^3.

The reduction in storage at stage 8 is a result of a change in the cumulative storage state variable at stage 8. For this model, the number of state variables at any stage is restricted to six. For each of these states, a storage decision is made. As the cost of storage is reduced, the model may choose to increase its decision volume at a particular cumulative storage state, or it may leave it the same, or it may even choose to reduce it. Each of the state variable levels is set before the model is run, and is not affected by the actual decisions during the model run. What is affected is the state variable level which the model currently is in. As more water is stored upstream of the stage, there is an increase in the cumulative storage state variable. Typically, the potential decisions at any stage are greater at the lower states than at the higher states. In other words, it is more beneficial to store water at a location if less has been stored to that point. With the change from \$1.50 to \$1.40, the potential decisions at each state do not change in stage 8. What changes is the state variable itself, because the cumulative storage has been increased by the inclusion of maximum storage at stage 7. At stage 8, as the cost of storage fluctuates, there is little or no change in the decision variable at each state. With an increase in the state variable, the actual decision for stage 8 changes, but it is a reaction to the change in state, not a change in policy due to the reduced dyke volume costs. Since more water is now stored upstream, the value of storage at stage 8 has decreased, thereby causing the storage volume choice to be reduced. This is likely due to the large area - volume relationship at this stage, and the resulting reluctance to store at its higher capacities.

The more perplexing decision addition at this cost level is at stage 14, which has been included at its maximum of 121900. Stage 14 has one of the poorest DSR values. Its location is also not highly desirable, as it is the 4th last stage in the decision process. It does have poor soil, but it is not the poorest. The only other advantage is that the area - volume relationship here is moderately low, compared to the other stages, but even in this aspect it is not among the best. The most likely explanation is the effect of the representative state variable. As discussed previously, the state values are not precise, but rather representative of a range of state variables. The actual state of the system is therefore not defined exactly. A small shift in state values will cause a change in the decisions made at stage 14, which is only chosen for storage in the lower states. This is illustrated by the reduction in storage at 14 as the cost is decreased further. It actually decreases to a zero storage level, before the cost is reduced to the point where the model will store anywhere it can. Stage 14 appears to be marginal in the optimal path decision process.

Stages 6 and 12 are introduced into the optimal path at $\$1.30/m^3$, while stages 14 and 16 show reductions in their storage volumes. Stage 12 has a DSR of 36, the next best after the group of stages 5, 7 and 16, and it has a fairly poor soil type. Stage 6 has a lower DSR than either stage 13 or 17, and a better soil type, so the decision here must be influenced more by its upstream location. The reduction at 16 is again only a matter of an increase in the state variable, rather than a major decision change. This is evident in the fact that reducing the cost to $1.10 results in stage 16 returning to its previous decision value. The only other difference at this cost level is the inclusion of state 13. It is apparent that stage 13 is not a strong candidate for inclusion, as it is dropped from the analysis at the next reduction in cost, implying that this may also be caused by a change in the state variable, and not a true change in the overall decision policy.

Explaining why stage 13 is not usually included in the optimal path at this cost level is more difficult. Other than this one occasion, water is not stored there until the cost gets so low that the model stores water virtually everywhere it can. Stage 13 has fairly poor soil, and it has a moderately good DSR, about the same as stage 12 and substantially better than stage 6, both already having been chosen as storage locations at this level. This does not appear to be a discrepancy caused by the representative state variable, because at virtually any state limited or no storage is chosen at this stage. In many respects it is similar to stage 12; the same soil type, a similar DSR, similar maximum storage volumes, similar area - volume relationships. The difference must lie in the fact that it lies downstream of stage 12. If storage is to take place, it should be stored at 12. Stage 13 is not seriously considered for storage until stage 12 is full.

Further reductions serve only to increase storage at all stages until at a cost of $0.10, all stages are at full storage capacity. It is evident from this very low cost requirement to facilitate storage that the cost of the storage outweighs the damage savings. Stages 9 and 10 are entered next, followed by 17 and, finally 4. Stage 14 also is returned to the optimal path.

The values to which the cost per volume must be reduced before the inclusion of stages 4 and 17, and the fact that these stages are at the extreme upstream and downstream ends of the section of the stream included in the decision process, illustrates the trade offs occurring here. Stage 4 has the worst DSR in the system, and by a fair margin. It has the lowest maximum capacity. It is also the stage furthest upstream. Stage 17, on the other hand, has a moderately good DSR (34) , one of the largest maximum capacities, the poorest soil, and is at the downstream end of the process. Neither of these are chosen until the cost of storage is very low. Stage 17 is not chosen due to its location. Virtually every other factor makes it a favorable location. Its relatively high DSR is not enough to counteract its location. Stage 4 is not chosen due to its very low DSR. The only other aspect not in its favour is its low maximum capacity. At only 40000 m^3, it would be better to fit this volume anywhere else. It would be almost unnoticeable included in stage 8, for instance, which is not very far upstream. These two stages illustrate the most important factors traded off in the evaluation of storage locations for this system, the cost of storage and the location of the stage in terms of its relative upstream or downstream position.

5.2 CROP TYPE DECISIONS

Included in the decision process is the crop type most suitable at each location within each stage. The first runs, with no changes to the damage values for each crop, yielded results that did not properly test the model. The model always chose barley for growing at every location at every stage. Investigation of the damage function in relation to the actual crop values shows the reason for this occurrence.

The amount of damage done to the crops is fairly small in most cases. Only the largest floods, and areas with high storage volume decisions, can cause a very great loss. The expected damage amounts to only about 20% of the crop value. The difference between the value of barley and that of wheat, the crop with the second best value, is actually greater than the expected damage to barley. Even after removing the average annual damage to barley, it has a greater return than does wheat, flax or alfalfa without removing flood damage. Therefore, no matter which situation occurs, the model will always choose barley to grow at each location.

This situation, in terms of the model, is due to the fact that the same duration - damage function that has been developed for barley was also used for wheat and flax. The lower net benefit for these crops results in their selection only if the expected damage for barley is greater than the difference in crop benefits, which never occurs. The model should be able to trade off between actual benefits (crop values) and the damage caused given the occurrence of a flood event. To properly test the model, the damage functions were varied by weighting the amount of damage due to duration for each crop. It is also important to vary these functions, because the actual duration - damage functions for wheat, flax and alfalfa are unknown, and those used in the model are assumed.

To test the model under the changing damage conditions, the model was run at the \$1.50 per dyke volume level. This value is used because it is mid-range, and a reasonable value. It also has an optimal path that is good for illustrative purposes as there is a variety in decisions over the entire system.

Table 10 shows the storage decisions with the dyke cost at \$1.50 and varying crop damages. There are two ways of considering this evaluation. The damage to barley can be varied, while leaving the other crops at the same level, or barley can be left and the other crops varied. Both of these were carried out, with the following results.

TABLE 10

Storage Decisions With Variation in Damage Values

Dyke cost = $1.50/m^3$

Stage	B=1 W=1 F=1 A=1	B=1 W=.5 F=.5 A=1	B=1 W=0 F=0 A=1	B=1 W=0 F=0 A=0	B=1.5 W=1 F=1 A=1	B=2 W=1 F=1 A=1	B=2 W=1 F=1 A=0	B=2 W=2 F=1 A=0	B=2 W=2 F=1 A=0'
4	1	1	1	1	1	1	1	1	1
5	3	3	4	4	3	1	1	1	1
6	1	1	2	2	1	1	1	1	1
7	2	2	6	6	1	1	1	1	1
8	5	5	4	4	5	5	5	1	6
9	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1
11	6	6	6	6	6	6	6	6	6
12	1	1	3	3	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1
14	1	1	5	5	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1
16	4	4	3	3	4	4	4	4	4
17	1	1	1	1	1	1	1	1	1

Note: B,W,F,A stand for Barley, Wheat, Flax and Alfalfa respectively. The number values associated with the crops in this table represent the amount by which the original damage function is altered. For instance, B=2 means that Barley is now damaged twice as much as in the original program design.
For the Alfalfa values, the last two columns have A=0 and A=0'. The difference is for A=0, the damage due to flooding is removed, but the probability of two crops is still reduced, and for A=0', the reduction of the probability of two crops is also removed.

To begin with, the damage due to flood duration for barley is doubled, with the functions for the other three crop types remaining unchanged. The results of this run are that for all areas of no flooding, the model chooses to grow barley, but in all areas where flooding occurs, whether in planned or unplanned flooding, the model chooses wheat. There is a reduction in total expected net benefit ($620990, reduced from $624007), which is due to the imposed greater losses for barley, as well as the choice of wheat in certain locations, which has a lower return than barley. The reduction in expected net benefits will also be a result of reduced storage. Stages 5 and 7 are removed completely from the optimal path, although at stages 8, 11 and 16, no reduction occurs. Increasing the damage to barley reduces the amount of barley grown, thereby reducing the total benefits, and decreasing the value of storage in the basin as a whole. Also, the increase results in the model choosing to store no water in some areas because it is now too susceptible to damage to grow it in a planned flooding area, and the return from wheat or either of the other crops does not pay for the storage at those locations.

It is expected that there is a trade off being weighed by the model between growing less of the most valuable crop, now that it is also the most heavily damaged by flooding, and growing as much as possible, but protecting it to a greater extent. At this damage level, it chooses to replace barley with another crop in the flooding areas.

To test for the possibility of the trade off between flood protection and barley reduction, the damage to barley was increased by 1.5 times its original value, rather than doubled. The result of this, as seen in Table 10, is to store more water (stage 7 returns to being a storage stage) but the cropping pattern is unchanged. Barley is still only grown in the non flooded areas. The flood damage to barley was then varied to find a point where this may change.

At a value of 1.35 times the original damage for barley, the model started choosing to grow wheat in some planned flooding areas. This occurred at stage 16, and would also occur at stage 17, if that stage were chosen for storage. At a factor of 1.40, this also ocurred, and to a greater extent. Stages 12, 14, 16, and 17 would grow wheat if there were planned storage at these locations. Stage 16 is the only stage among these to have a planned flooding area. As the damage potential is increased toward 1.5 again, more areas are used to grow wheat, with the final result at 1.5 of all planned and unplanned flooding areas growing wheat. However, the model never chooses to store more water in order to gain from the added benefits from barley. It is apparent that the return from barley is not great enough to offset the expense of water storage, even with the added return from wheat, which would be grown in the flooded areas.

In an attempt to force the model to increase storage for improved benefit, the damage weight for barley was increased to 1.75. This was not successful. No increase in storage over the runs with a lower weight placed on damage to barley was chosen. In fact, less storage is chosen at this level.

Again, the return from these crops given the average damage from flooding is not enough to offset the cost of storage.

As another attempt, the barley damage weight was returned to 2, while those of all other crops were reduced to 0. In this way, it should be more beneficial to store water because if a crop other than barley is grown in the planned flooding areas, full benefits will be realized from these areas. The run does, in fact, have this effect. Water is stored at more stages, and increased at most stages which already have some storage. Wheat is grown in all planned and unplanned flooding areas, and at many of the stages where storage is imposed, no non flooded area remains, so wheat is grown throughout the stage. Flax and alfalfa are never chosen, due to their inferior return, and because the damage due to flooding for these crops is the same as for wheat during this run (0).

Increasing the weight on wheat damage to 2, with flax at 1, barley remaining at 2 and alfalfa at 0, resulted in an increased tendancy toward growing flax in the planned flooding area, although this is never actually chosen in those stages in which storage is planned. In other words, the way the model works, a crop will be chosen for each of the three area types of the stage (planned flooding, unplanned flooding, and non flooded), even though the area of planned flooding may be 0. This would of course be the case if no storage is planned for the particular stage.

Comparing this run (column 8 in Table 10, with the previous one (column 7) shows a reduction in the total storage volume (stage 8 is dropped from the storage plan). The only difference between the two evaluations is the weight on wheat (it is doubled in the second run). When the weight is 1, flax is never chosen for the planned flooding area, and wheat is grown in all storage areas. Doubling the weight on wheat damage results in flax being considered for the planned flooding area, but not actually chosen. Thus the total net benefits are reduced, due to increased damage to wheat in the storage areas, forcing a reduction in the now unaffordable storage plan.

In the previous run, although alfalfa is assumed to be undamaged by flooding, the value of alfalfa includes the probability of getting two crops per year. By increasing the probability of getting two crops to 1, the yearly benefits are increased. The purpose in doing this is to attempt to force more storage, with the damage weights for all other crops remaining the same as for the previous run (barley, wheat = 2, flax = 1). The result, as seen in the final column of Table 10, is that more water is stored, by storing the maximum at stage 8 (an increase from storing none) with no other changes. Stage 8 is the cheapest location for storage since stage 11 is already at the maximum. Alfalfa has not been chosen at any location to this point, but now that it is effectively more valuable, and is the only crop with a

0 damage weight, it is chosen for all storage locations. Flax is no longer considered at any location.

The implication of this result is that a greater return is derived from alfalfa in its current "flood proof" state than it costs to store at stage 8, but not at other stages. Other crops, because their damage values are too great, do not make the increased storage feasible. This is evident in the choice of flax at some locations in the previous run, which was removed when the alfalfa damage was reduced.

Keeping the weight on barley damage at 1 and varying the weights on the other crops has little effect on the optimal path until the weights are reduced to 0. With wheat and flax at a weight of 0, and varying alfalfa from 0 to 1, (columns 3 and 4), there is no effect on the optimal path. This is due to the fact that since alfalfa is not chosen at either run, no change need occur. Alfalfa, no matter what its damage potential at the current weight level of the other crops, produces an expected benefit less than that of wheat, barley or flax, even including their damage potentials. In this run, wheat is grown at all planned and unplanned flooding areas, with barley at all non flooded areas.

With alfalfa and barley at a weight of 1, flax and wheat are increased toward 1. The optimal path is changed and the storage total is reduced, until at a weight of 0.5, the optimal path and total storage volume are the same as for the original run at \$1.50/m^3. At this weight level, the value of wheat, including losses due to flooding, is still lower than the value of barley, when flooding losses are included.

5.3 SUMMARY

From the tests described in this chapter, several conclusions can be made about this model. First, the model will choose to grow the crop which has the best overall expected net benefit for the particular location. The storage decision process is sensitive to crop damage and crop net benefit values. Secondly, the model is sensitive to inputs other than crop values. The storage level decisions appear to be based on a combination of: the soil quality, the relative location of the stage in terms of its upstream or downstream position, the cost of storage as it varies from stage to stage, and possibly the area flooded by volume of storage relationships at each stage, and the actual volume of storage itself.

Chapter VI

SUMMARY AND CONCLUSIONS

6.1 SUMMARY

A dynamic programming model has been developed for the purpose of determining a policy for flood damage mitigation in a small agricultural watershed, which involves land use decisions in combination with limited structural input. The land use decisions are of two types: 1) a "hydrologic use" decision, or the optimal locations for water storage and the volume of storage at those locations, and 2) a crop type decision, determining the best type of crop to grow at each hydrologic use type at each stage. At each stage, there are three types of hydrologic use decisions to choose from: 1) the planned flooding area, which is the area of storage, if a decision to store is made, 2) the unplanned flooding area, which exists at a stage if the cumulative storage up to and including the stage currently being evaluated is not sufficient to store the larger floods, or if there is no planned flooding, and 3) the non-flooded area, which will exist if the combination of previous decisions and the topography at the stage is such that the largest flood entering the system will be unable to inundate some part of the stage. The size of these areas is a function of the local topography, the volume of storage decision at the stage, and the cumulative storage decisions to this point.

The storage decisions are based on several factors, including the soil type at the stage, the cost of storage at the stage, and the location of the stage in terms of its relative upstream or downstream position. It also seems to be based on the area flooded per volume of storage, and the actual volume of storage itself. However, it is difficult to quantify the effect of these last two parameters. The predominant factors appear to be the cost of storage at the stage (dyke volume - storage volume ratios) and the location of the stage.

The model is sensitive to the cost per dyke volume, which is constant over the decision process, but it is not as sensitive as was anticipated at the beginning of the study, over the range of values ($1.50/m^3 to $2.00/m^3) considered to be realistic. Due to the total dyke costs resulting from the decisions within this range of dyke costs per volume, (highest is $21564 at $1.80/m^3) it was decided that stages 2 and 3, although capable of storing large volumes of water in channel, would be too costly to include in the decision process.

It is suspected that there may be a problem with using the representative state variable. In some cases the decisions at a stage are unrealistic based on the important parameters at the stage. Also, investigation of the stage matrix shows that decisions may be very random and show great variation from one state to the next. This problem can be reduced, and greater precision added to the model by using a discrete differential dynamic program. The problem, however, does not actually occur within the reasonable cost range, so no action was taken to correct it.

The model chooses only barley to grow at any location, regardless of its hydrologic use, or the volume decision. This is due to the high return from barley, even with the inclusion of the expected damage, being greater than that of the next best crop, wheat. This may be a result of incorrect assumptions in the assessment of crop damages, and though the model was tested under a number of damage conditions, the actual damage functions, especially for wheat, flax, and alfalfa, but also to an extent for barley, are unknown. It is shown, however, that the model will choose to grow the crop which provides the greatest expected net benefit, and that the crop decision also has an effect on the storage decision.

6.2 SUGGESTED MODEL IMPROVEMENTS

There are some improvements which can be made to this model. The two problems previously mentioned are among them, but not all of them. Greater precision could be achieved by making the model a discrete differential dynamic program.

This would vastly increase the already high memory requirement, and the expensive execution time, and this increase may not warrant the precision improvement. Certainly for use with the Wilson Creek watershed the improvement would not be worth the expense, as can be seen by the results in the range of dyke costs considered reasonable. If the model can be used more generally, the improvement may be worthwhile, and perhaps even necessary.

Another improvement which would also increase the precision of the model is to use a continuous series for floods and flood probabilities. In effect it would still be discrete, since for computer application it must be, but the precision would be much finer than using just five flood levels. Again, the increased precision may not be worthwhile, given the increased time and memory requirements.

There is also a problem which was not included in this study at all, but due to the direction in which operations research, among other disciplines, is going, could be included in future models of this sort. The problem is one of equity. No consideration is given to where the water is stored in terms of whose land it is on. It is possible, in fact very likely given the results of the decision process in this model, that all or most of the water is stored on one farmer's land. The particular farmer is losing benefit from his land by storing water (and perhaps by growing crops with a lower return) in order to increase benefits downstream, which may belong to another farmer. Including equity evolves the model into a multiobjective one, but the increased complication may not greatly affect the computer time and memory requirements.

6.3 CONCLUSIONS

The model developed in this thesis is capable of determining the best locations and volumes of water storage, and the type of crops to grow in each area, in order to have the maximum expected net benefit from the basin as a whole. The model chooses to store small volumes of water at a number of locations, rather than a large volume at one location. This is less expensive, even at the greatest total dyke cost, than other flood mitigation projects that have been attempted at Wilson Creek. Assuming that the model inputs are not too far wrong, given the assumptions which were necessary, the value of the combination of structural and non-structural flood mitigation methods appear to be greater than the more costly, purely structural methods.

REFERENCES

Bialas, W.F., and Loucks, D.P., Nonstructural Floodplain Planning, Water Resources Research, 14, 1, Feb. 1978, pp. 67-74.

Ball, M.O., Bialas, W.F. and Loucks, D.P., Structural Flood Control Planning, Water Resources Research, 14, 1, Feb. 1978, pp. 62-66.

Chow, V.T., Maidment, D.R., and Tauxe, G.W., Computer Time and Memory Requirements for DP and DDDP in Water Resources Systems Analysis, Water Resources Research, 11, 5, 1975, pp. 621-628.

Day, J.C., A recursive programming model for nonstructural flood damage control, Water Resources Research, 6, 5, Oct. 1970, pp. 1262-1271.

Faculty of Agriculture, University of Manitoba, Principles and Practices of Commercial Farming, 5th. ed., University of Manitoba, 1977.

Ford, D.T., Interactive Non-Structural Flood Control Planning, Journal of Water Resources Planning and Management Division, American Society of Civil Engineers, WR2, 107, Oct. 1981, pp. 351-363.

Goulter, I.C. and Morgan, D.R., Analyzing Alternative Flood Damage Reduction Measures on Small Rural Watersheds Using Multiple Return Period Floods, Water Resources Research, 19, 6, Dec. 1983, pp.1376-1382.

Hillier, F.S. and Lieberman, G.J., Introduction to Operations Research, 3rd. ed., Holden - Day, Ltd., Oakland, 1980, 820 pp..

Hopkins, L.D., Brill, E.D., Leibman, J.C. and Wenzel, H.G., Land Use Allocation Model for Flood Control, Journal of Water Resources Planning and Management Division, American Society of Civil Engineers, WR1, 104, Nov. 1978, pp. 93-104.

Hopkins, L.D., Brill, E.D., Jr., Kurtz, K.B., and Wenzel, H.G., Jr., Analyzing Floodplain Policies Using an Interdependent Land Use Allocation Model, Water Resources Research, 17, 3, June 1981, pp. 469-477.

James, D.L., Nonstructural Measures for Flood Control, Water Resources Research, 1, 1, 1965, pp. 9-24

James, D.L., Economic Analysis of Alternative Flood Control Measures, Water Resources Research, 3, 2, 1967, pp. 333-343

James, D.L. and Lee, R.R., Economics of Water Resources Planning, McGraw-Hill Co., New York, 1971, 615 pp..

Lacewell, R.D. and Eidman, V.R., General Model for Evaluating Agricultural Flood Plains, American Journal of Agricultural Economics, Feb. 1972, pp. 92-101.

Leyshon, A.J. and Sheard, R.W., Influence of Short Term Flooding on the Growth and Plant Nutrient Composition of Barley, Canadian Journal of Soil Science, 54, 4, Nov. 1974, pp. 463-473.

Canadian Grain Commission, Grain Statistics Weekly, Economic and Statistics Division, 21 Dec. 1973.

Manitoba Department of Agriculture, 1983 Field Crop Recommendations for Manitoba, 1983.

Committee on Headwater Flood and Erosion Control, Report on Activities in Wilson Creek Watershed, Series of Annual Reports on the Wilson Creek Watershed from 1959 to 1982 prepared for the Committee on Headwater and Flood and Erosion Control, Planning Division, Water Resources Branch, Department of Natural Resources, Winnipeg, Manitoba.

Appendix A

EXPECTED DAMAGE FUNCTION

The expected damage function includes two main parts, the calculation of damage due to a given flood level and the probability level assigned to each flood level. Both of these aspects are discussed in this appendix.

A.1 FLOOD DAMAGE

Damage for a given flood is based on a duration-damage function determined by Leyshon and Sheard (1974) and modified and extended in this study to include other crops. Other factors of importance are planting dates, area flooded, soil productivity, and crop value.

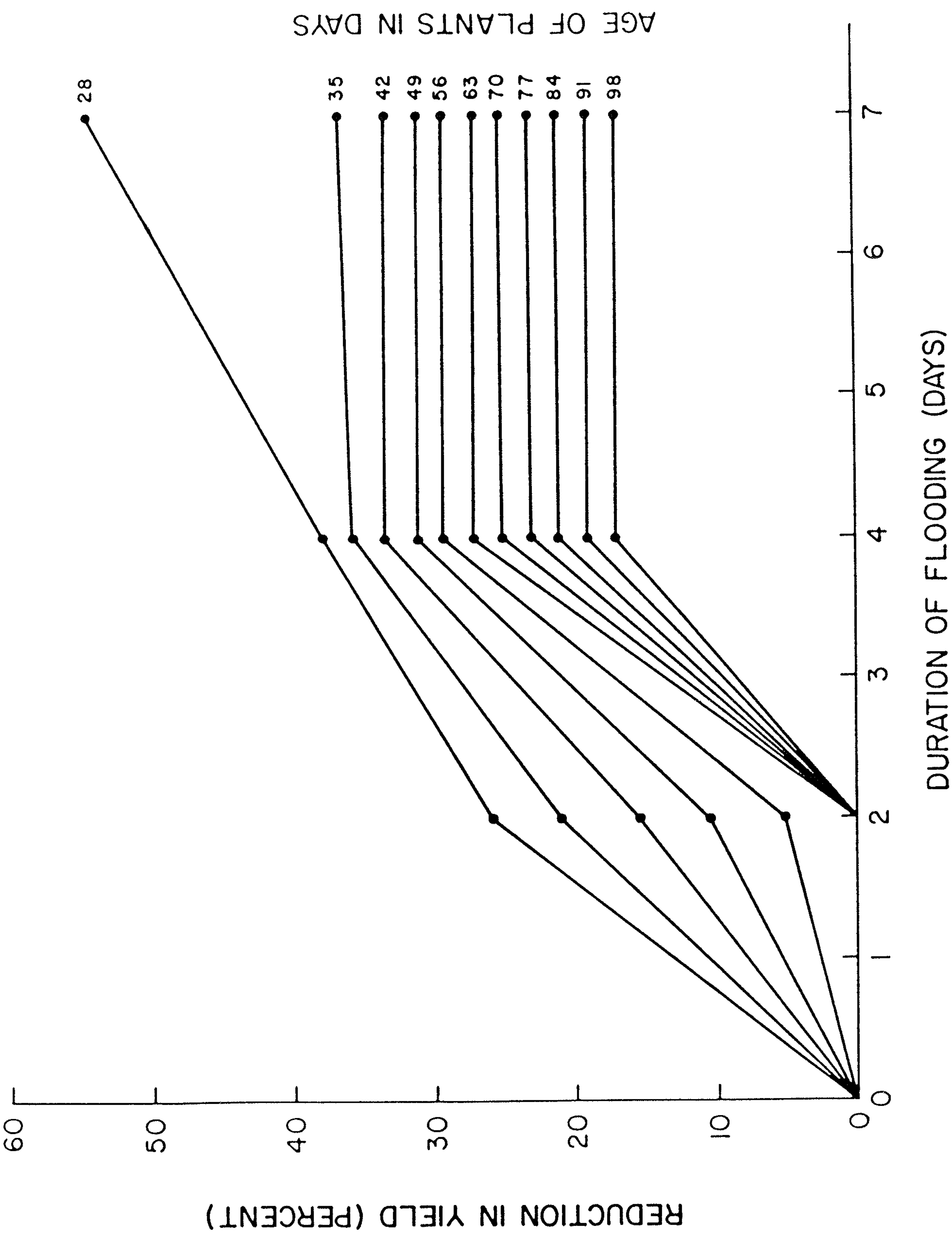
A.1.1 Damage Due to Duration of Flooding

Figure 7 shows an extended version of the duration/damage values for barley in various stages of growth (Leyshon and Sheard, 1974). Leyshon and Sheard calculated damage values for only 21, 28 and 35 day old plants. The 21 day old plant showed full recovery after flooding and is not shown on the figure for that reason. The curves for plant ages above 35 days were derived for this study by merely using the differences at the 4 day flood duration between each 7 day step as exists between the 28 and 35 day plant values. The other points of the curves were fit in what seemed the best positions given the information available. This form of extension was needed to provide data to demonstrate the function of the model.

For the other two grain crops, wheat and flax, the same relationship was used as for barley, as no such information could be found for these crops. Again, this was done to demonstrate the model. For crop ages of less than 28 days, no damage directly attributable to duration of flooding was assessed, according to the Leyshon and Sheard evaluation for barley. However, a recovery period of approximately 7 days was required before growth resumed. This is due to delayed emergence in younger plants, or in recovery in post emergent plants.

Figure 8 shows average yield versus seeding dates for wheat, flax and barley for the region which includes Wilson Creek. The information for this figure is from Manitoba Department of Agriculture (1983). It is assumed that delayed emergence and the recovery period of the plants younger than 28 days is essentially the same, in terms of reduced yield, as delayed seeding. Therefore, for the three grain crops, damage to the younger plants is the indirect damage related to delayed seeding. The total duration will be the duration of flooding plus the 7 day recovery period.

Figure 7: Variation in Flood Damage Susceptibility With Stage of Growth for Barley



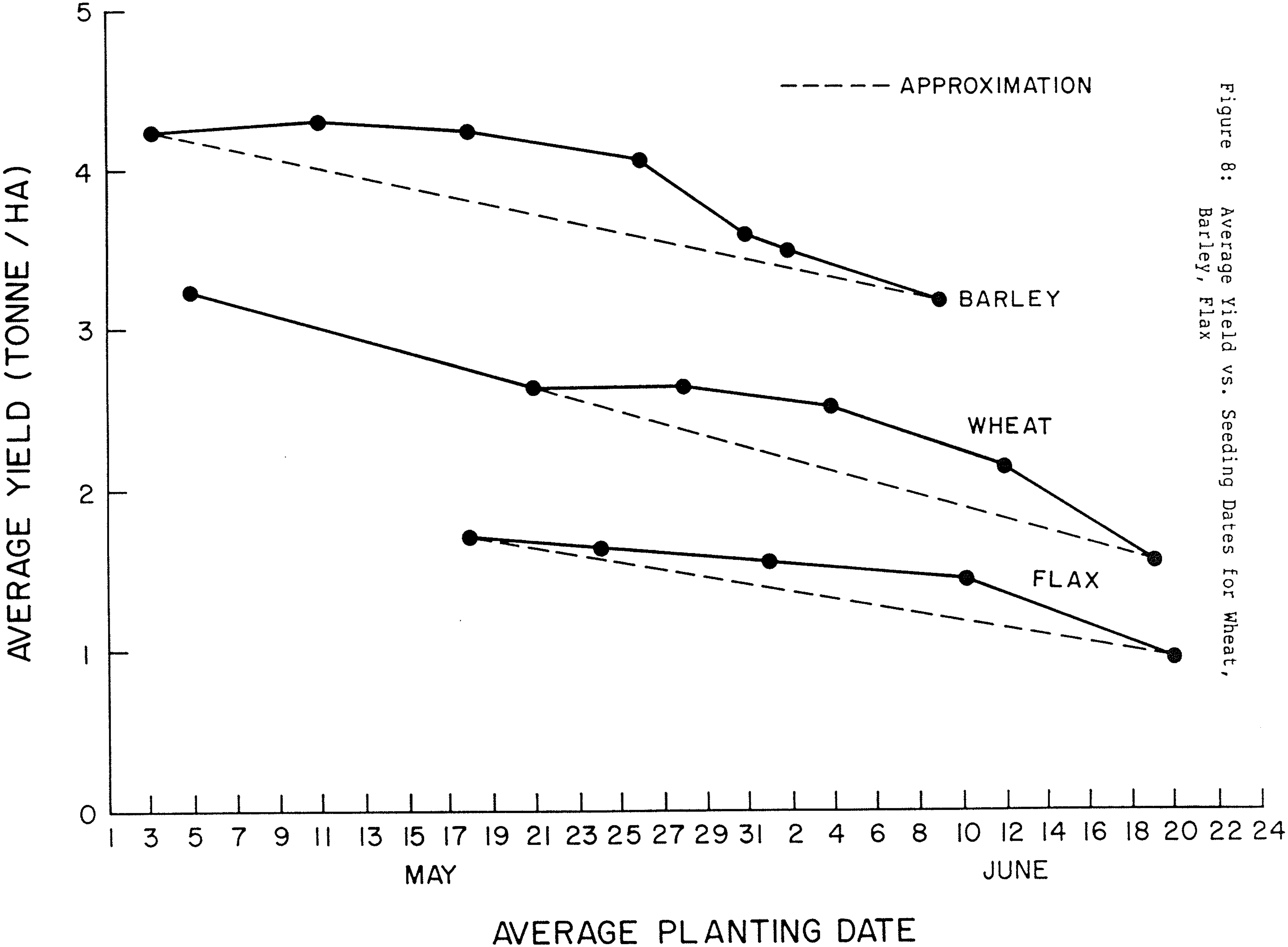


Figure 8: Average Yield vs. Seeding Dates for Wheat, Barley, Flax

For the fourth crop, alfalfa, it was assumed that no damage related directly to duration would occur. For this crop also, some indirect damage was assessed in the following manner. It is possible to get two crops off the field in a single growing season. A probability distribution can be developed for this situation. A contributing factor is the length of the growing season. Alfalfa takes 60 days to mature, or 120 days for two crops. To get two crops within the defined growing period, the first crop must be started by May 19. Since the average date of latest killing frost for this area is May 16, the probability of the crop being underway by May 19 is estimated to be 0.6. Any delay at any point during the growing season will result in reducing the probability of two crops. It is also assumed that the same recovery period established for the grain crops will be useable for alfalfa, resulting the delay of crop growth of 7 days plus the duration of inundation.

For alfalfa, the indirect loss due to flooding duration is:

Loss = 1440/(1608 + dur)

Where: 1440 = hours in 60 days

1608 = 1440 hours + hours in 7 days

dur = duration of inundation

Resulting in a total loss of:

Damage = 0.6 * loss * (173.84) + (1.0 - (0.6 * loss) * 86.92)

Where: 0.6 = probability of two crops

173.84 = value of two crops

86.92 = value of one crop

For the grain crops, from the average yield vs. seeding date values, the approximate linear equation for yield by hour of flooding:

$$y = -0.001189 * dur + 4.26 \qquad \text{for barley}$$

$$y = -0.001578 * dur + 3.238 \qquad \text{for wheat}$$

$$y = -0.000964 * dur + 1.67 \qquad \text{for flax}$$

Where y = yield in tonnes / hectare

These values, translated into loss per hour of inundation, in percentage of crop value is:

$$L = 0.0279107 * (dur + 168) \qquad \text{for barley}$$

$$L = 0.0487338 * (dur + 168) \qquad \text{for wheat}$$

$$L = 0.0577071 * (dur + 168) \qquad \text{for flax}$$

It is expected that even a short duration flood will cause this delay, so the shortest duration of inundation will result in 168 hours of delay and the resulting crop loss. Minimum crop losses given a flood will be:

Barley	4.69 %
Wheat	8.19 %
Flax	9.69 %

If a flood occurs early enough in the season, there is the possibility of reseeding. The cost of reseeding is equal to the start up costs. On a per hectare basis, the start costs for the crops are:

Barley	$118.31	
Wheat	$125.33	
Flax	$117.42	
Alfalfa	$ 61.75	(est.)

These values represent a reduction in value of the crop of:

Barley	42.11%
Wheat	53.83%
Flax	56.44%
Alfalfa	71.04%

Reseeding would only be undertaken if the losses due to flooding are greater than these values. This is explicitly considered in this model. Also, reseeding will be done only if the damage occurs early enough in the season.

Other than the complications of reseeding and loss due to delayed emergence, the duration-damage function follows the curves as seen in the damage susceptibility curve.

The damages for each growth period are shown for barley in Table 11, for wheat in Table 12, and for flax in Table 13.

TABLE 11

Linear Functions Relating Duration and Damage for Growth Stages for Barley

Growth Stage (days)	Flood Duration (hours)	Function
< 0		Y = 0.0279107 * (X + 168)
0 - 21		Y = min: 0.0279107 * (X + 168) 42.11 if in period 1 or 2
22 - 28	0 - 48	Y = 0.54167 X
	49 - 96	Y = 0.25000 X + 14.0
	> 96	Y = min: 0.23610 X + 15.3 42.11 if in period 1 or 2
29 - 35	0 - 48	Y = 0.43750 X
	49 - 96	Y = 0.31250 X + 6.0
	> 96	Y = min: 0.01380 X + 34.67 42.11 if in period 1 or 2
36 - 42	0 - 48	Y = 0.32290 X
	49 - 96	Y = 0.37920 X - 2.7
	> 96	Y = 33.7
43 - 49	0 - 48	Y = 0.21460 X
	49 - 56	Y = 0.43130 X - 10.4
	> 96	Y = 31.0
50 - 56	0 - 48	Y = 0.10420 X
	49 - 96	Y = 0.51040 X - 19.5
	> 96	Y = 29.5
57 - 63	0 - 48	Y = 4.69
	49 - 96	Y = 0.57290 X - 27.5
	> 96	Y = 27.5
64 - 70	0 - 48	Y = 4.69
	49 - 96	Y = 0.53130 X - 25.5
	> 96	Y = 25.5
71 - 77	0 - 48	Y = 4.69
	49 - 96	Y = 0.48960 X - 23.5
	> 96	Y = 23.5
77 - 87	0 - 48	Y = 4.69
	49 - 96	Y = 0.44790 X - 21.5
	> 96	Y = 21.5
> 87		Y = 50.0

TABLE 12

Linear Functions Relating Duration and Damage for Growth Stages for Wheat

Growth Stage (days)	Flood Duration (hours)	Function
< 0		Y = 0.0487338 * (X + 168)
0 - 21		Y = min: 0.0487338 * (X + 168)
		53.83 if in period 1 or 2
22 - 28	0 - 48	Y = 0.54167 X
	49 - 96	Y = 0.25000 X + 14.0
	> 96	Y = min: 0.23610 X + 15.3
		53.83 if in period 1 or 2
29 - 35	0 - 48	Y = 0.43750 X
	49 - 96	Y = 0.31250 X + 6.0
	> 96	Y = min: 0.01380 X + 34.67
		53.83 if in period 1 or 2
36 - 42	0 - 48	Y = 0.32290 X
	49 - 96	Y = 0.37920 X - 2.7
	> 96	Y = 33.7
43 - 49	0 - 48	Y = 0.21460 X
	49 - 56	Y = 0.43130 X - 10.4
	> 96	Y = 31.0
50 - 56	0 - 48	Y = 0.10420 X
	49 - 96	Y = 0.51040 X - 19.5
	> 96	Y = 29.5
57 - 63	0 - 48	Y = 8.19
	49 - 96	Y = 0.57290 X - 27.5
	> 96	Y = 27.5
64 - 70	0 - 48	Y = 8.19
	49 - 96	Y = 0.53130 X - 25.5
	> 96	Y = 25.5
71 - 77	0 - 48	Y = 8.19
	49 - 96	Y = 0.48960 X - 23.5
	> 96	Y = 23.5
77 - 91	0 - 48	Y = 8.19
	49 - 96	Y = 0.44790 X - 21.5
	> 96	Y = 21.5
> 87		Y = 50.0

TABLE 13

Linear Functions Relating Duration and Damage for Growth Stages for Flax

Growth Stage (days)	Flood Duration (hours)	Function
< 0		Y = 0.0577071 * (X + 168)
0 - 21		Y = min: 0.0577071 * (X + 168)
		56.44 if in period 2
22 - 28	0 - 48	Y = 0.54167 X
	49 - 96	Y = 0.25000 X + 14.0
	> 96	Y = min: 0.23610 X + 15.3
		56.44 if in period 2
29 - 35	0 - 48	Y = 0.43750 X
	49 - 96	Y = 0.31250 X + 6.0
	> 96	Y = min: 0.01380 X + 34.67
		56.44 if in period 2
36 - 42	0 - 48	Y = 0.32290 X
	49 - 96	Y = 0.37920 X - 2.7
	> 96	Y = 33.7
43 - 49	0 - 48	Y = 0.21460 X
	49 - 56	Y = 0.43130 X - 10.4
	> 96	Y = 31.0
50 - 56	0 - 48	Y = 0.10420 X
	49 - 96	Y = 0.51040 X - 19.5
	> 96	Y = 29.5
57 - 63	0 - 48	Y = 9.69
	49 - 96	Y = 0.57290 X - 27.5
	> 96	Y = 27.5
64 - 70	0 - 48	Y = 9.69
	49 - 96	Y = 0.53130 X - 25.5
	> 96	Y = 25.5
71 - 77	0 - 48	Y = 9.69
	49 - 96	Y = 0.48960 X - 23.5
	> 96	Y = 23.5
77 - 91	0 - 48	Y = 9.69
	49 - 96	Y = 0.44790 X - 21.5
	> 96	Y = 21.5
> 91		Y = 50.0

A.1.2 The Joint Probability Matrix (JPM)

Due to the variation in damage susceptibility with growth stage of the crop, a function is developed to determine the probability of a plant being in a particular stage of growth. Secondly, a probability function is developed to determine the probability of being within a particular time period as defined for this model. These two must be combined to form a joint probability function to quantify the probability of a plant being at a particular stage of growth within a certain time period. Combined with this will be the flood probability levels for each time period. Table 14, Table 15, and Table 16 show the joint probability matrices for barley, wheat and flax, respectively.

Each cell of the joint probability matrix is the probability of being in a certain stage of growth and in a certain time period. The probabilities of being at a particular stage of growth were calculated simply by the number of days in the range out of the total growing season. For example, the probability of being a 0 to 21 day old plant is 21/153 = 0.13725, where 153 is the total number of growing days.

TABLE 14

Joint Probability Matrix for Barley

Growth Stages (days)	Time Periods in Growing Season May 1 to May 19	May 19 to Jun 20	Jun 21 to Jul 17	Jul 18 to Aug 31	Sept 1 to Sept 30
<0	.0285004	.0695387	.0000000	.0000000	.0000000
0-21	.0336563	.0976823	.0059162	.0000000	.0000000
22-28	.0000000	.0373104	.0084411	.0000000	.0000000
29-35	.0000000	.0289277	.0168238	.0000000	.0000000
36-42	.0000000	.0176826	.0276016	.0004673	.0000000
43-49	.0000000	.0044563	.0377611	.0035342	.0000000
50-56	.0000000	.0000000	.0362881	.0094634	.0000000
57-63	.0000000	.0000000	.0274965	.0182550	.0000000
64-70	.0000000	.0000000	.0158425	.0299091	.0000000
71-77	.0000000	.0000000	.0030947	.0426569	.0000000
78-84	.0000000	.0000000	.0000000	.0457224	.0000292
85-87	.0000000	.0000000	.0000000	.0194075	.0002003
>87	.0000000	.0000000	.0000000	.1262627	.2070709

TABLE 15

Joint Probability Matrix for Wheat

Growth Stages (days)	Time Periods in Growing Season May 1 to May 19	May 19 to Jun 20	Jun 21 to Jul 17	Jul 18 to Aug 31	Sept 1 to Sept 30
<0	.0585358	.0395033	.0000000	.0000000	.0000000
0-21	.0419448	.0812108	.0140991	.0000000	.0000000
22-28	.0000000	.0341948	.0115568	.0000000	.0000000
29-35	.0000000	.0274256	.0175492	.0007768	.0000000
36-42	.0000000	.0191028	.0235416	.0031072	.0000000
43-49	.0000000	.0092264	.0295340	.0069912	.0000000
50-56	.0000000	.0000000	.0333229	.0124287	.0000000
57-63	.0000000	.0000000	.0263317	.0194198	.0000000
64-70	.0000000	.0000000	.0177870	.0279646	.0000000
71-77	.0000000	.0000000	.0076887	.0378092	.0002536
78-84	.0000000	.0000000	.0000000	.0438334	.0019182
85-91	.0000000	.0000000	.0000000	.0406152	.0051364
>91	.0000000	.0000000	.0000000	.0898560	.2173334

TABLE 16

Joint Probability Matrix for Flax

Growth Stages (days)	Time Periods in Growing Season May 1 to May 19	May 19 to Jun 20	Jun 21 to Jul 17	Jul 18 to Aug 31	Sept 1 to Sept 30
<0	.0519031	.0461360	.0000000	.0000000	.0000000
0-21	.0000000	.0941581	.0430967	.0000000	.0000000
22-28	.0000000	.0072794	.0384121	.0000597	.0000000
29-35	.0000000	.0009626	.0409388	.0038502	.0000000
36-42	.0000000	.0000000	.0322156	.0135359	.0000000
43-49	.0000000	.0000000	.0173862	.0283654	.0000000
50-56	.0000000	.0000000	.0060160	.0397356	.0000000
57-63	.0000000	.0000000	.0005414	.0452101	.0000000
64-70	.0000000	.0000000	.0000000	.0457516	.0000000
71-77	.0000000	.0000000	.0000000	.0442476	.0000688
78-84	.0000000	.0000000	.0000000	.0370886	.0086630
85-91	.0000000	.0000000	.0000000	.0240339	.0217177
>91	.0000000	.0000000	.0000000	.0516504	.2555393

The end points (the probabilities of not being planted, and that of being mature) were calculated in a different manner, as no range is available. The probability of not being seeded was chosen to be 15 out of 153 days. The average date of planting is May 16, with 15 days before, by definition of the growing season, and about 15 days after, depending on the crop. The probability of being a mature crop was determined by subtracting the sum of the probabilities of all other growth stages from 1.0, as the total sum must equal 1.0.

A.1.3 Combination of JPM with the Damage Functions

The damage functions and the JPM are combined by multiplying the corresponding JPM cell and damage function for each growth stage. The values are summed over all growth stages to arrive at a value for damage given a flood of known duration. The damage value must be subtracted from 100% to transpose it into a value of percent of crop remaining. By doing this the problem has been changed from one of expected damage to one of expected benefit.

A.2 CALCULATION OF EXPECTED NET BENEFITS

The percent of crop remaining is the value represented by the DUR (x) * DDF (c) calculation from Equation 9. The remaining percentage for each crop is then multiplied by the corresponding crop value (see Appendix D), the soil factor (SF), and the area of inundation (AREA), which is calculated with the duration values (see Appendices B and C).

If the net benefit values currently being calculated are for the planned flooding area, the total area has been divided into four units of equal flood volume. Each of these represents a specific area of inundation, and a duration. The damage values for the four areas are summed at this point to give a total area value. If the area under calculation is the unplanned flooding area, there is only the single area and duration, and thus one damage value. Any remaining area within the stage is area not flooded, and therefore subject to no damage, but is part of the net benefit calculation.

The net benefits for each of the three areas are determined for each of the four crop types, and the value for the crop which returns the maximum benefit for each area type is stored for calculation of expected net benefits.

A.2.1 Combination With Probability Levels

The net benefit values calculated to this point are then multiplied with the probability bands for each time period, and summed over all six probability levels for the flood level currently being assessed. The entire procedure is repeated, as described above, for each flood level, and summed over all six probability levels which represent the flood magnitude spectrum. The cost of storage for the policy decision (volume of storage) at the particular stage, state, and decision currently under assessment, is then subtracted from this value, finally resulting in the expected net benefit value. Details of the storage cost are in Appendix F.

Appendix B

DETERMINATION OF HYDROGRAPHS

Five storm hydrographs were used to be representative of the entire flood magnitude spectrum. The decision to use five hydrographs was somewhat arbitrary. The number had to be large enough to be reasonably representative of the spectrum, while being small enough to keep the computational efforts to a minimum.

It was decided, arbitrarily, to use the 5, 10, 20, 40, and 50 year return period peak flow values to cover the required range. The peak flow values were derived from the frequency analysis for annual peaks (see Appendix F).

B.1 HYDROGRAPH SYNTHESIS

To derive hydrographs with the determined peak sizes, an average time to peak and total time base value for each peak flow had to be calculated. To do this, all peak flow values, starting flow values, the time to peak, time of recession of the flow, and the final flow value were taken from the flow records.

Table 17 shows the peak flow, time to peak, recession limb duration, and the total time for all floods on the creek on record.

TABLE 17

Historical Storm Hydrographs From Wilson Creek

Date	Peak Flow (m/s)	Time to Peak (hrs)	Recession Limb (hrs)	Total Time (hrs)
JUNE 9, 1963	6.90	11	43	54
JUNE 30, 1963	7.17	2	2	4
MAY 5, 1965	3.68	10	35	45
SEPT 17, 1965	4.25	12	30	42
AUG 6, 1966	3.26	4	28	32
JUNE 26, 1969	19.80	18	44	62
JUNE 28, 1969	10.42	14	34	48
COMB. OF JUNE 26, 28 1969	19.8	18	92	110
MAY 2, 1970	8.39	124	95	219
JUNE 5, 1971	21.70	26	47	73
MAY 10, 1974	8.15	25	101	126
AUG 24, 1975	8.17	5	24	29
SEPT 18, 1975	44.75	25	101	126
JULY 11, 1977	15.4	11	28	39
MAY 14, 1979	10.4	19	70	89
AUG 4, 1980	4.93	4	12	16
AUG 20, 1980	6.11	8	18	26

Note: 1) The storms of June 26 and June 28, 1969 appear in the flow records to be separate, unrelated storms. They are, however, difficult to separate, and so are studied from seperate, as well as a combined point of view.

2) The storm of May 2, 1970 is such an unusual case, that it was considered an outlier and excluded from the regression analysis used to determine the relationship between peak flow and the time values.

Using these data, an attempt was made (Graham, personal communication) to establish a relationship between peak flow and time to peak and total time. A regression analysis was used with peak flow as the independent variable. The R^2 value in each case was poor (0.36 for time to peak, and 0.51 for total time), but the resulting prediction equations were used to derive the hydrographs, for lack of a better method. Table 18 shows the time to peak, recession limb time, and total time for the five synthetic hydrographs used in the analysis. The peak flows were determined from the annual peak frequency analysis (see Appendix F).

TABLE 18

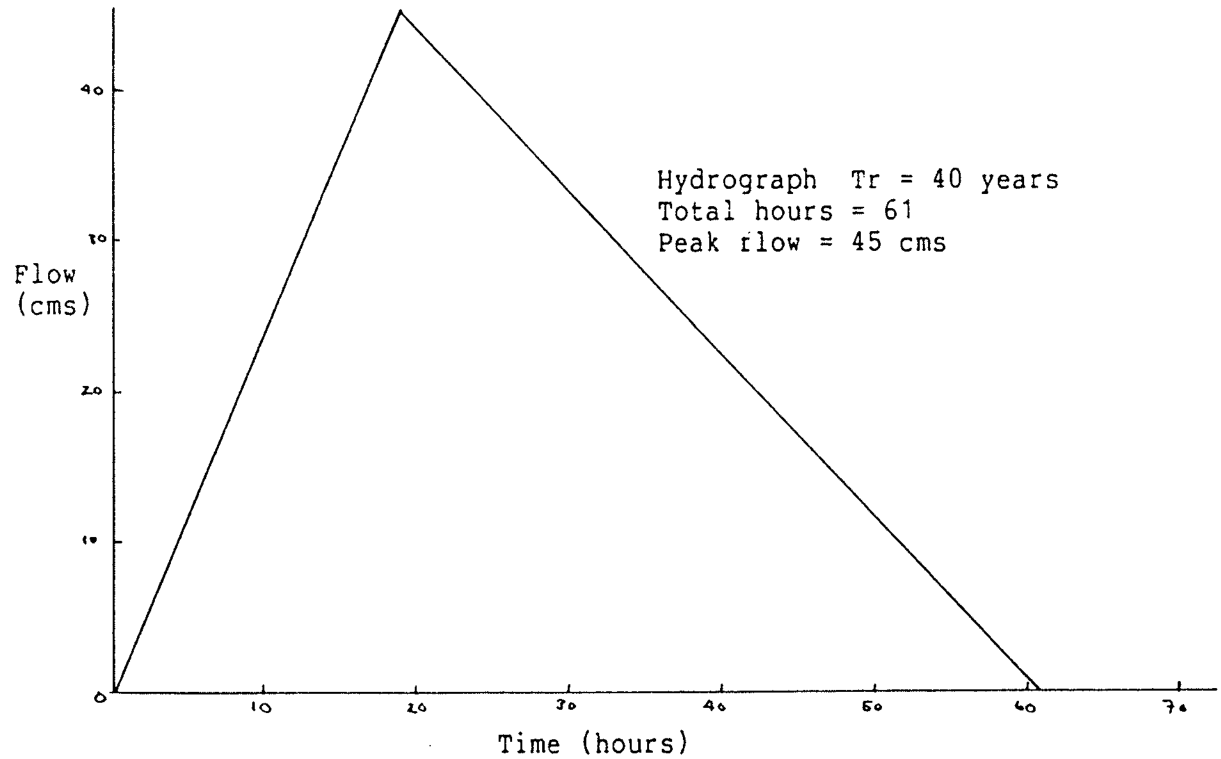
Hydrograph Peak Flow and Time Base Values

Peak Flow (m^3/s)	Time to Peak (hrs)	Total Time (hrs)	Recession Time (hrs)
10.8	10	34	34
20.0	12	41	29
32.0	16	52	36
45.0	19	61	42
55.0	22	70	48

Prediction Equations:
Time to peak = (peak flow * 0.281658) + 6.521421
Total time = (peak flow * 1.313545) + 20.819059
Recession time = Total time - Time to peak
Note: Tabled values are rounded off.

Figure 9 shows the five hydrographs used in the model evaluations.

Figure 9: Hydrographs Used in Model Evaluation



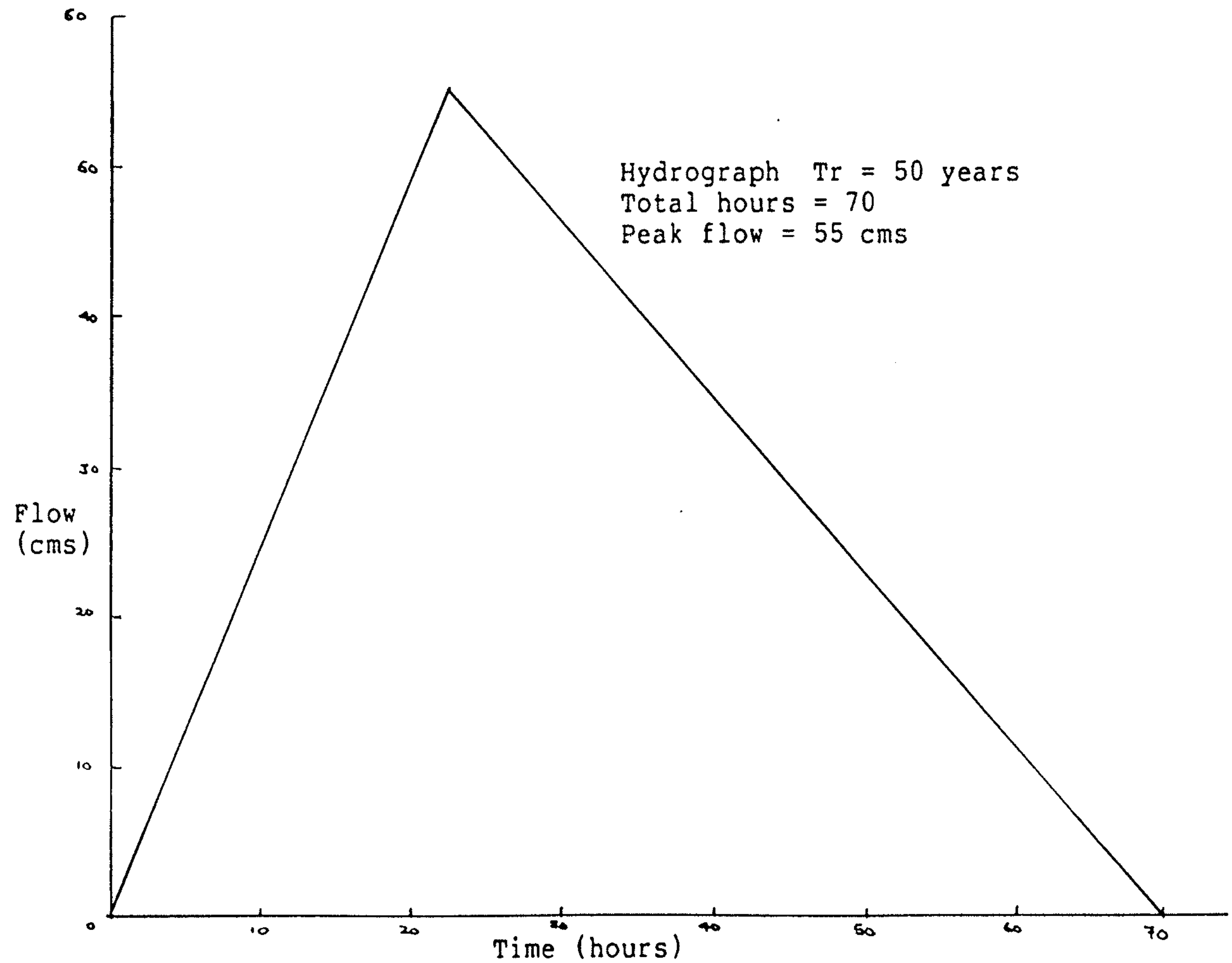
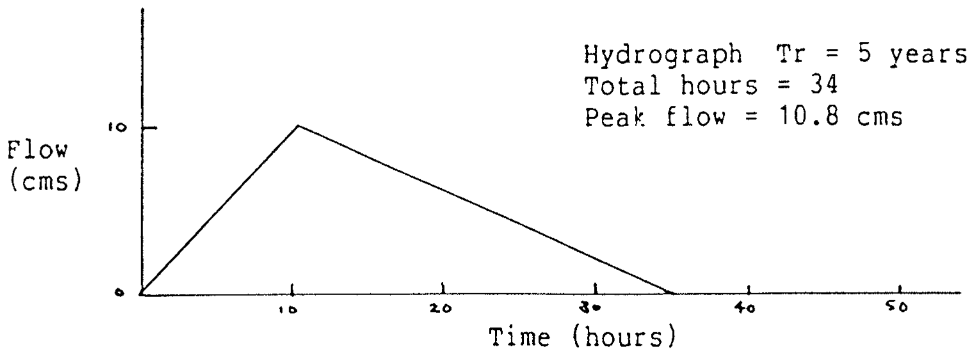
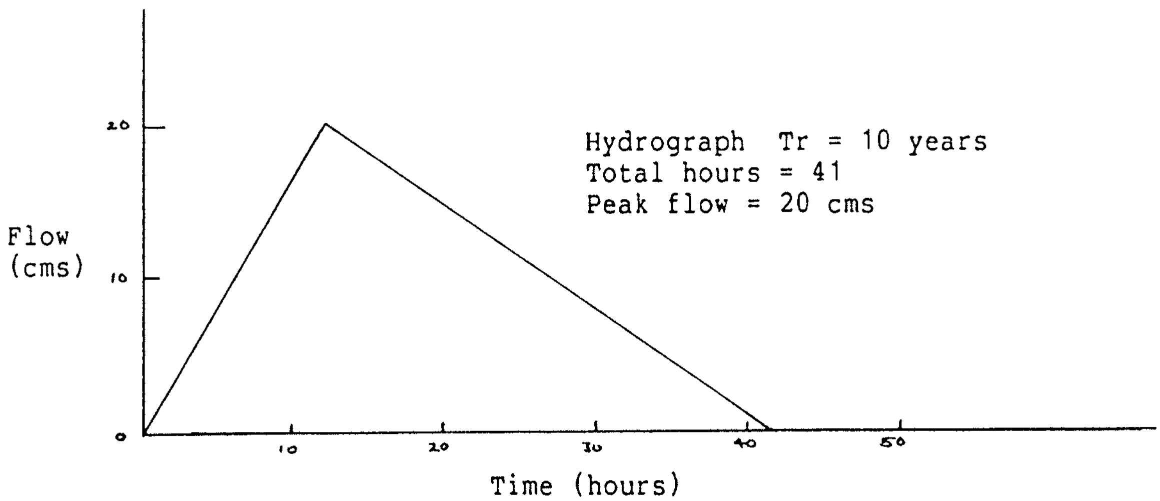
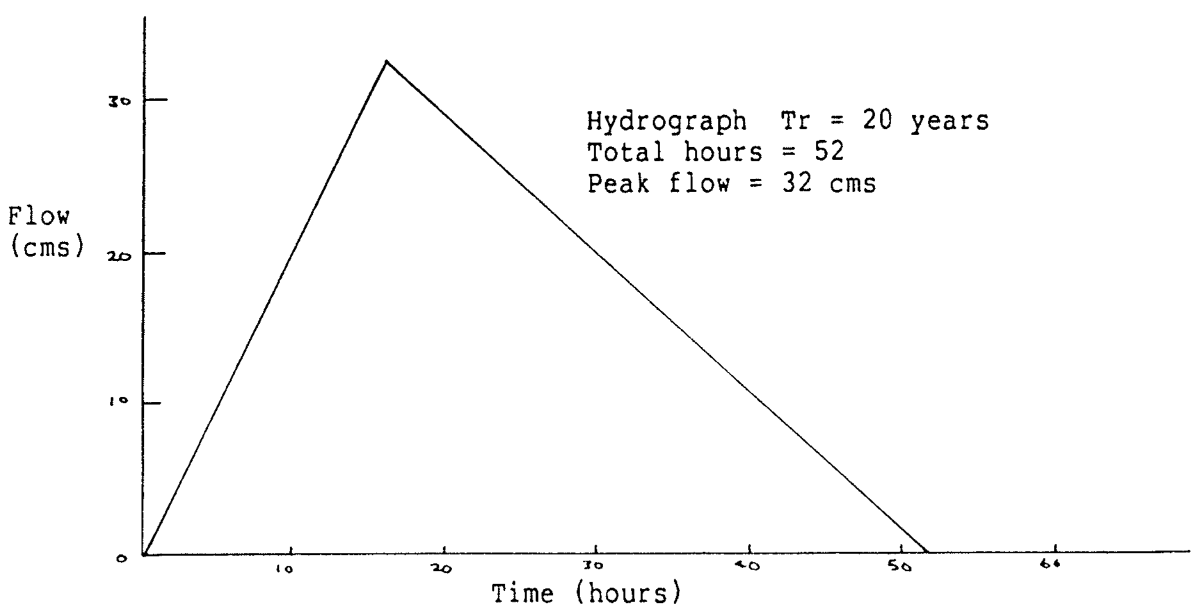


Figure 9 continued







B.2 THE ROUTING PROCEDURE

Figure 10 shows a simple plan view schematic of a stage, including a potential storage area. As a hydrograph enters the stage, its shape is dictated by the cumulative effect of the storage decisions upstream. The new hydrograph shape as it leaves the stage (ie. the outflow hydrograph) will be determined by the storage decision at this stage.

Until a decision to store some water is made, the hydrograph retains its original shape. Figure 11 shows this as the first of four hydrograph shapes which the routine must evaluate.

In order to facilitate storage at a stage, the stream is restricted by a simple control device which allows only the channel capacity to flow through. A dyke perpendicular to the stream creates the downstream end of the reservoir. In the storage locations, as the hydrograph begins to rise, the outflow hydrograph from the stage will be the same as inflow until channel capacity is reached. Once this point is passed, flow through the channel is restricted and water is stored behind the dyke. The outflow hydrograph stays constant at channel capacity. When the storage area is filled to capacity, water will spill over the dyke and into the next stage, adding to the channel capacity flow as input to the next stage. Depending on the size of the flow in rela-

Figure 10: Stage Schematic Plan View

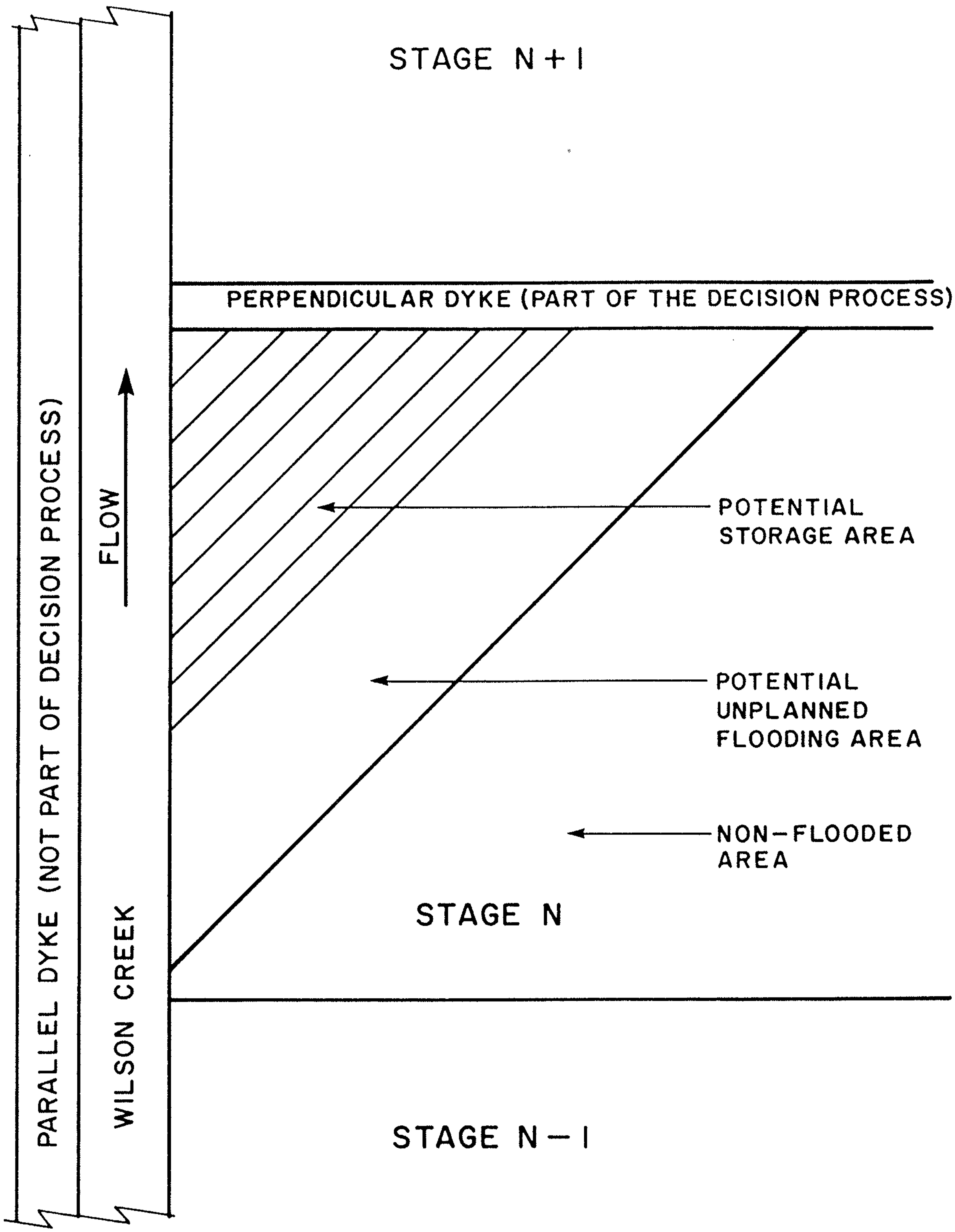
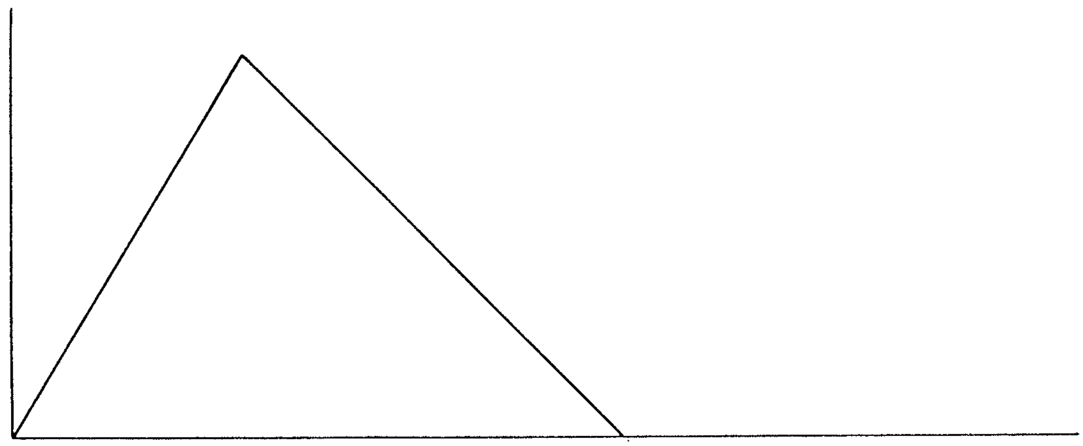


Figure 11: Hydrograph Shape 1



tion to the size of the reservoir, three other outflow shapes are possible. Figure 13 shows the first of these.

The first shape results from the reservoir filling before the peak is reached. Note on Figure 12 that the actual peak flow is not reduced if this occurs. However, since water is stored, peak reduction will be easier downstream, so a benefit is realized.

The second shape, Figure 13, occurs when the reservoir fills after the peak has passed, but before all the flood water is stored. This does result in a peak reduction.

Figure 12: Hydrograph Shape 2

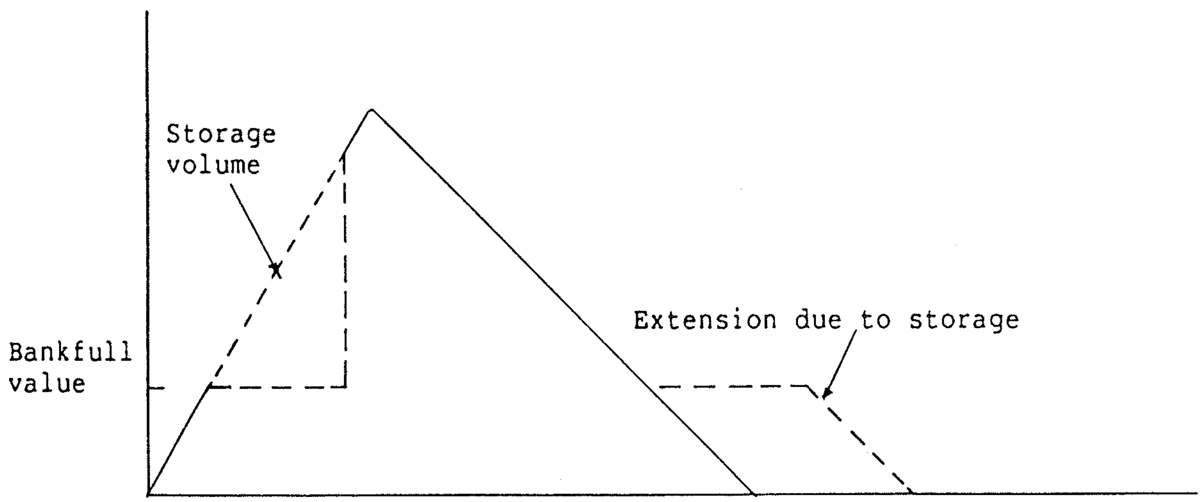
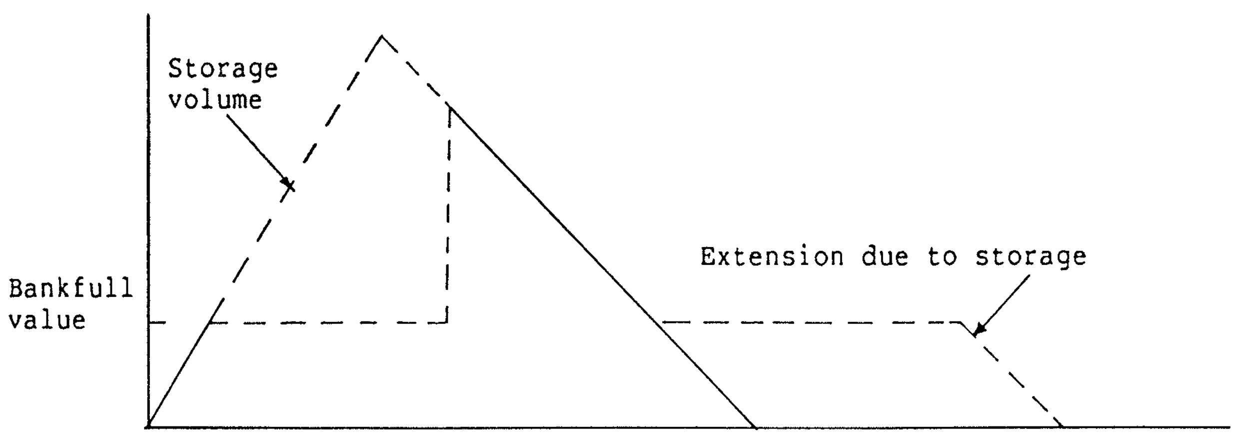
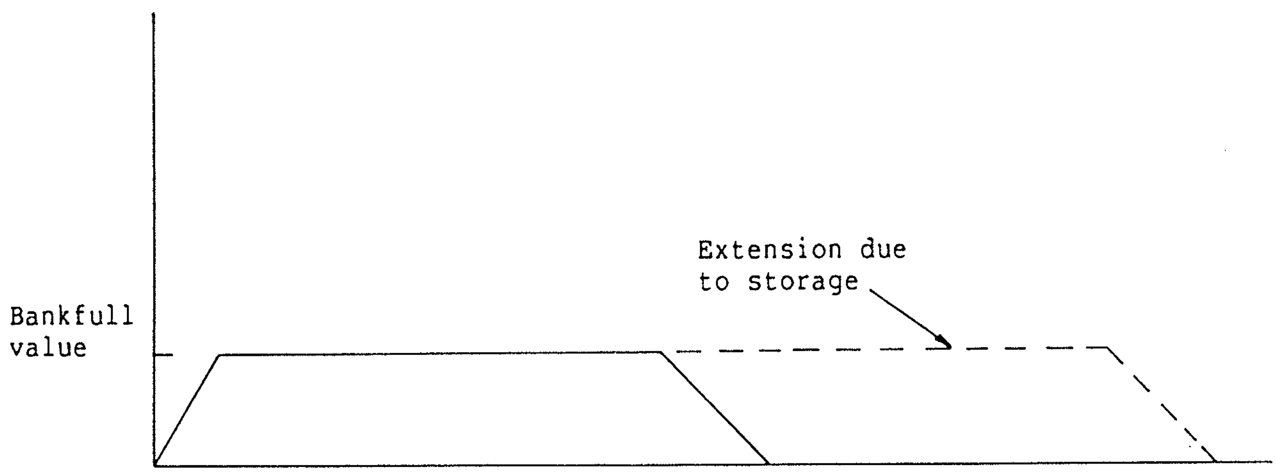


Figure 13: Hydrograph Shape 3



The final shape is left when the reservoir holds the whole of the remaining volume of water. See Figure 14.

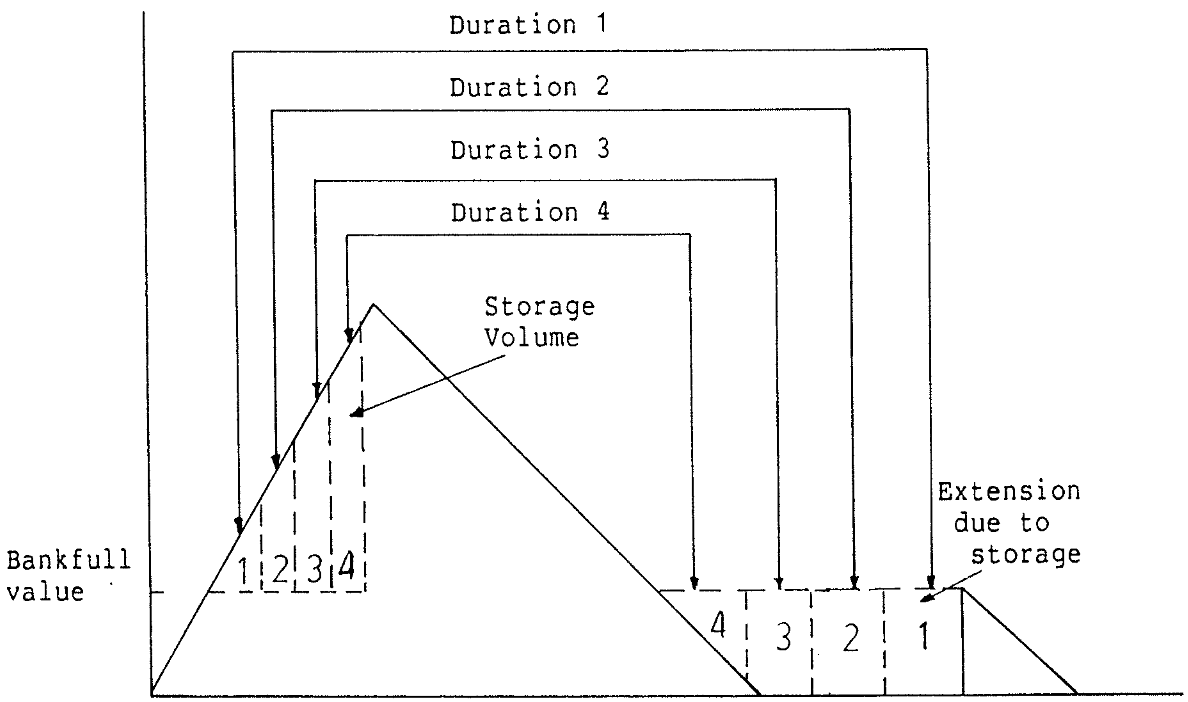
Figure 14: Hydrograph Shape 4



Reduction of the flow results in an increase in the time base of the hydrograph. Once the inflow is back to channel capacity, the outflow will remain at the outflow channel capacity until the reservoir is empty. Only then will the flow decline from bankfull. The volume (area under the curve) of the extension is equal to the volume of storage by definition.

The areas of the hydrograph which represents the storage volume, and the resulting hydrograph extension, are divided into 4 sub areas of equal volume, to more accurately determine the areas flooded, and the durations of flooding on those areas. Figure 15 shows these divisions and resulting durations.

Figure 15: Hydrograph Divisions for Duration Calculation



Each of the four equal volume divisions also represent 4 divisions of the area flooded, each with a duration calculated from these divisions. The area closest to the channel (division 1) will be the first to flood and the last to drain, thus having the longest duration of flooding. The area farthest from the channel will flood last and drain first, giving it the shortest duration of flooding. The durations are calculated from the distances from the centroids of each division in the main part of the hydrograph to the centroids of their respective corresponding divisions in the hydrograph extension.

The top line of the hydrograph extension also represents the channel capacity at that stage. On the outflow hydrograph, channel capacity is maintained until the flood water is completely drained. The area under the recession limb of the extension is therefore not part of the volume of storage, as no flooding is occuring past the start of the recession limb.

Appendix C

STAGE DESCRIPTIONS

Appendix C describes the calculations for determining the physical characteristics of the stages. Included are calculations of the area flooded per volume of storage, area flooded by volume of the unplanned flooding area, channel capacity, and any other aspect peculiar to the stage.

C.1 GENERAL DESCRIPTION OF THE STAGES

Table 19 shows the general physical descriptions of the stages.

The areal extents of the stages were determined by Goulter and Morgan (1983). Stages 7 through 17 approximate a quarter section and define an area which includes the flood plain. The other stages vary in size due to the physical nature of the specific stage or due to restrictions placed on them by the design of the system.

TABLE 19

Physical Descriptions of Stages

Stage	Size (hect.)	Channel Capacity (m^3/sec)	Storage Maximum (m^3)	Soil Type
1	1200	*	0.0	*
2	248.90	4491.1	239250.	G
3	237.75	2899.7	88150.	C
4	101.36	10.76	40000.	E
5	122.50	10.76	117800.	C
6	83.89	10.76	60000.	C
7	81.67	10.76	175175.	C
8	81.67	10.76	345950.	C
9	81.67	10.76	206790.	C
10	81.67	10.76	91200.	D
11	81.67	10.76	103845.	E
12	81.67	10.76	560000.	E
13	81.67	10.76	548000.	E
14	81.67	10.76	121900.	E
15	81.67	10.76	276000.	E
16	81.67	10.76	529000.	F
17	81.67	10.76	437000.	F
18	97.68	10.76	*	F

(* = no value calculated as none is required)

C.1.1 Channel Capacity Calculations

Cross sectional areas were calculated using cross sections from Manitoba Department of Natural Resources, Water Resources Branch, for several locations along the creek. Since most of the length of the creek is an unnatural channel, that part of it was expected to be fairly homogeneous in terms of its cross sectional area and slope. Two cross sections for this part of the channel (at stage 4 and stage 10) were calculated, and the channel capacity determined. The two channel capacity values were virtually identical.

The value of $10.76\ m^3/sec$ was used as the channel capacity for stages 4 through 18 inclusive.

For the section of the creek that is natural, homogeneity was not assumed, and channel capacities for both stages 2 and 3 were calculated. These values were substantially larger than the channel capacity of the creek downstream, and in fact are also larger than the largest flow on record. The capacity of stage 2 is $4491.2\ m^3$ and the capacity of stage is $2899.7\ m^3$. The calculations for channel capacity are given below. Figure 16 shows the cross sections for stages 2, 3, 4, and 10.

Channel Capacity Calculations

Using Manning's Equation:

$$Q = 1.49/n * R^{0.66} * S^{0.5} * A$$

Where: Q = channel capacity in cfs
n = a roughness coefficient
R = the hydraulic radius
= A / P (P = wetted perimeter)
S = slope
A = cross sectional area in ft^2

Stage 2:
slope = .0162
cross sectional area = 4405 ft^2
P = 250 ft
R = 4405 / 250 = 17.62
using an 'n' value of: 0.035

$$Q = 1.49 / 0.035 * 17.62^{0.66} * 0.0162^{0.5} * 4405.$$
$$= 158583.05\ ft^3/sec$$
$$= 4491.2\ m^3$$

Stage 3:
slope = 0.0154
cross sectional area = 930 ft^2
P = 100 ft
R = 930 / 100 = 93
using an 'n' value of 0.035

$$Q = 1.49 / 0.035 * 93^{0.66} * 0.0154^{0.5} * 930$$

$= 102387\ ft^3/sec$
$= 2899.66\ m^3/sec$

Stage 4:

slope = 0.0211
cross sectional area = 53.75 ft^2
P = 44 ft
R = 53.75 / 44 = 1.22
using an 'n' value of 0.035

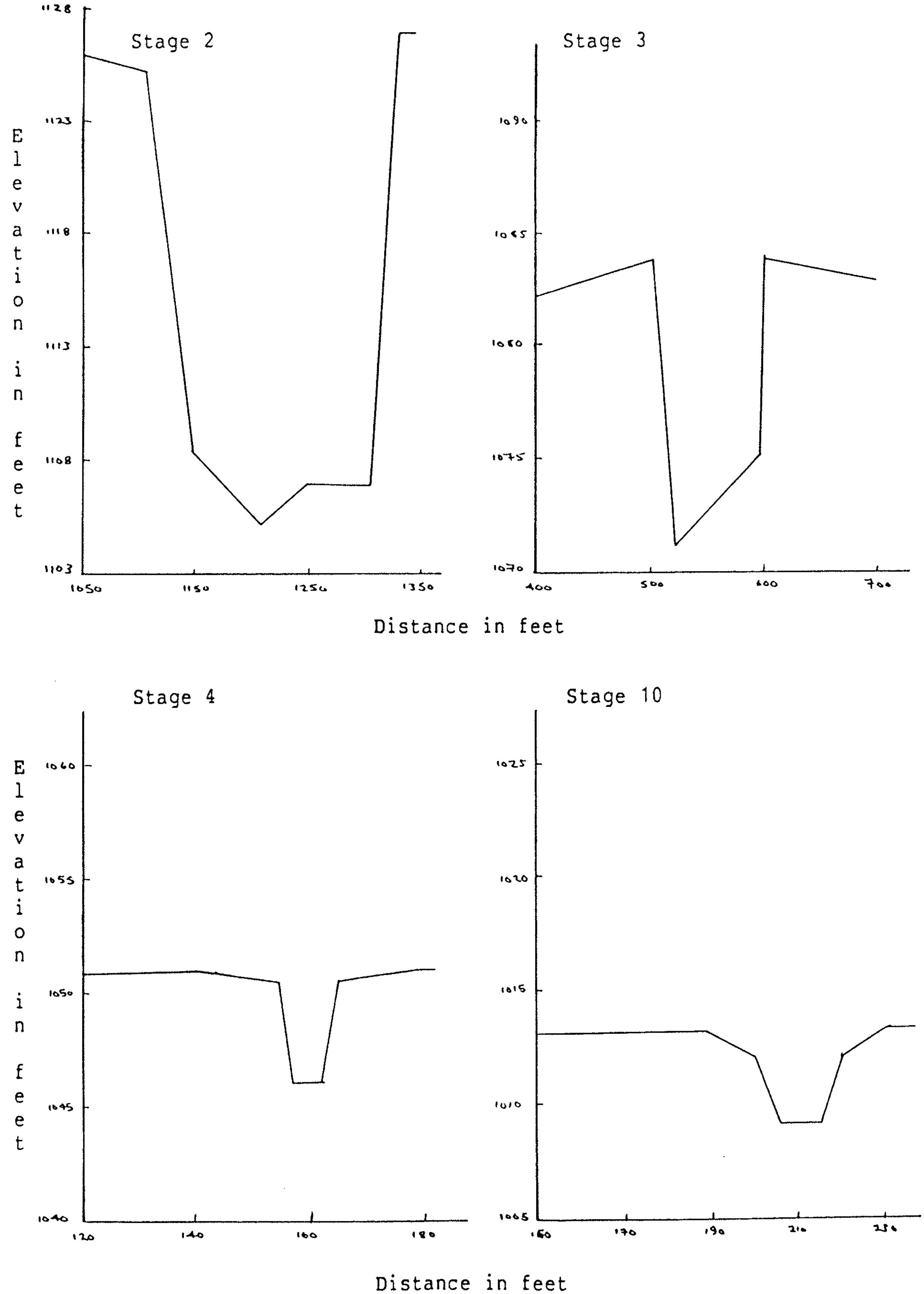
$Q = 1.49 / 0.035 * 1.22^{0.66} * 0.0211^{0.5} * 53.75$
$= 379.9\ ft^3/sec$
$= 10.76\ m^3/sec$

Stage 10:

slope = 0.0033
cross sectional area = 85 ft^2
P = 36
R = 85 / 36 = 2.361
using an 'n' value of 0.035

$Q = 1.49 / 0.035 * 2.361^{0.66} * 0.0033^{0.5} * 85$
$= 371.28\ ft^3/sec$
$= 10.52\ m^3/sec$

Figure 16: Channel Cross Sections



C.2 STORAGE VOLUME AND AREA FLOODED RELATIONSHIPS

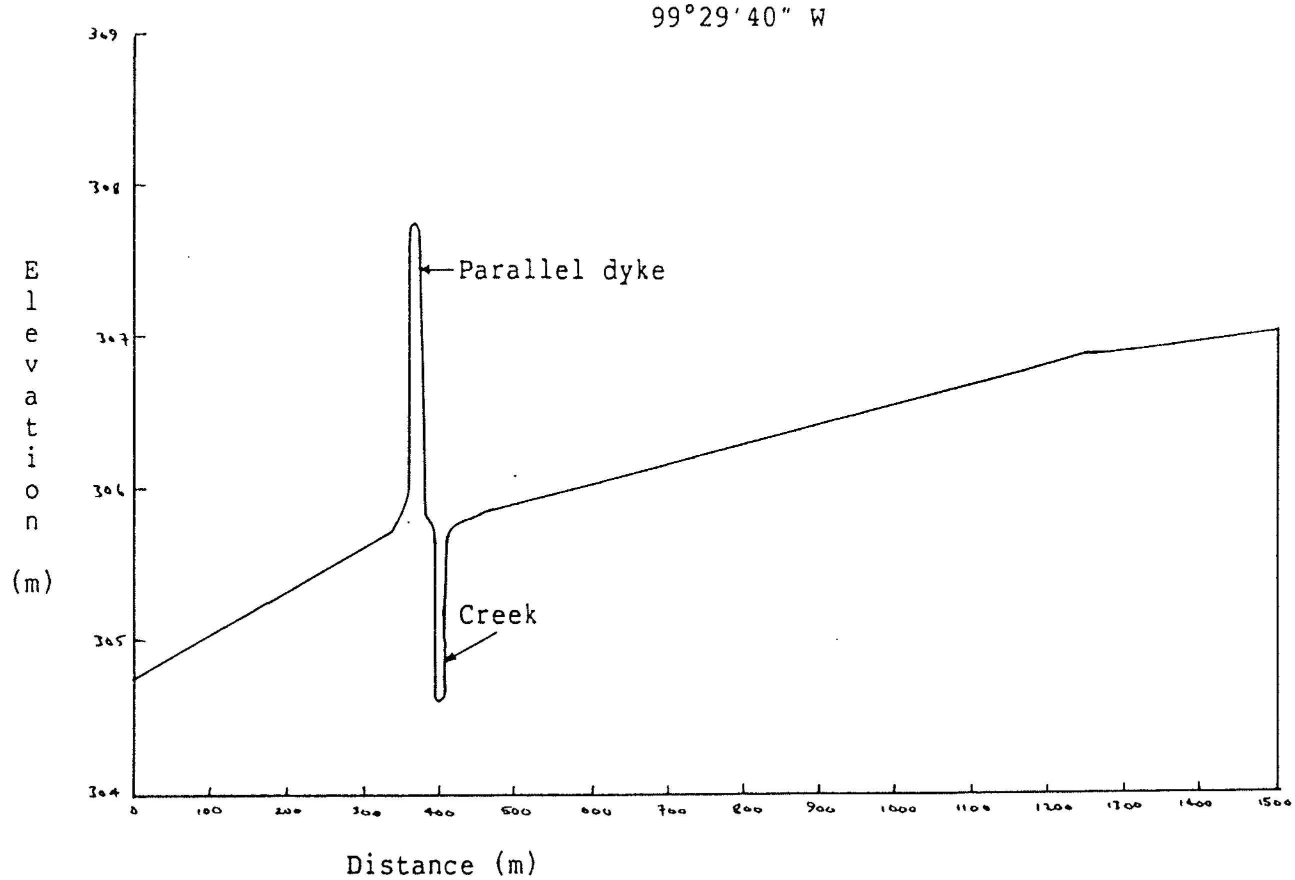
Cross sections and profiles for the entire stage area at each stage were developed from topographic maps and aerial photos. From these, a maximum storage volume was calculated based on the topographical features of the stage. In many cases, the limiting factor is a maximum height restriction, imposed by the dyke parallel to the channel on the north side.

From the stage cross sections, varying elevation (depth of flooding) values are chosen, with the upper boundary determined from topographical features, and the lower boundary being zero. A series of corresponding storage volumes are calculated using the depth of flooding values, the cross sections, and the profiles. Flooded areas are calculated in a similar manner. A series of linear relationships is developed between area flooded and storage volume for each stage. These relationships are used to determine area flooded for a particular storage decision, as well as for the area of unplanned flooding. Figure 17 shows a sample cross section and profile from which the maximum flooding depth and the area flooded - storage volume relationships are derived. Table 20 shows the derived depths of flooding, including the maximum depth, the area flooded, and the corresponding storage volumes.

Figure 17: Sample Cross Section and Profile

Stage 9: Cross-section

50°45'25" N
99°29'40" W



Stage 9: Profile

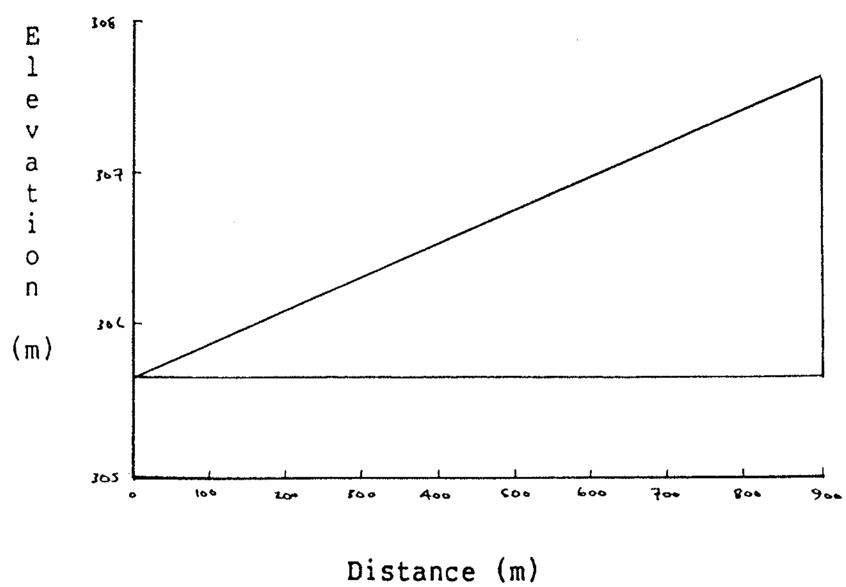


TABLE 20

Area Flooded - Storage Volume Relationships

Stage	Elevation (m)	Length (m)	Width (m)	Depth (avg) (m)	Cross -sect (m^2)	Area flooded (m^2)	Storage Volume (m^3)
2	342.9	290	660	2.5	825	191400	239250
	342.0	240	530	2.0	530	127200	127200
	341.0	180	370	1.6	296	53280	66600
	340.0	120	200	1.0	100	12000	24000
	339.5	100	120	0.8	48	4800	12000
3	328.0	215	410	2.0	410	88150	88150
	327.0	170	310	1.45	225	52700	38208
	326.0	100	210	0.9	77	21000	7650
	325.5	70	140	0.65	65	9800	4550
	325.0	40	120	0.4	24	4800	960
4	317.0	100	500	1.6	400	50000	40000
	316.5	70	360	0.35	63	25200	4410
	316.0	30	200	0.25	25	6000	750
5	315.8	760	310	1.0	155	235600	117800
	315.5	640	255	0.85	108	163200	69120
	315.0	450	190	0.6	57	85500	25650
	314.5	260	130	0.4	26	33800	6760
	314.0	70	45	0.1	2	3150	158
6	312.6	500	240	1.0	120	120000	60000
	312.0	350	170	0.7	60	59500	20825
	311.5	230	120	0.5	30	27600	6900
	311.0	100	70	0.2	7	7000	700
7	310.8	650	490	1.1	270	318500	175175
	310.5	520	380	0.9	171	197600	88920
	310.0	350	260	0.6	78	91000	27300
	309.5	200	140	0.4	28	28000	5600
8	309.2	850	740	1.1	407	629000	345950
	309.0	750	670	1.0	335	502500	251250
	308.5	350	470	0.6	141	258500	77550
	308.0	330	290	0.4	58	95700	19140
	307.5	120	110	0.1	6	13200	660
9	307.0	610	1130	0.6	339	689300	206790
	306.5	380	630	0.4	126	239400	47880
	306.0	150	250	0.1	13	37500	1875

Table 20 continued

10	304.8	320	1140	0.5	285	364800	91200
	304.5	210	640	0.4	128	134400	26880
	304.25	140	440	0.3	113	61600	18480
11	303.6	258	1150	0.7	403	296700	103845
	303.5	256	1100	0.6	330	281600	84480
	303.3	251	1000	0.4	200	251000	50200
	303.0	241	350	0.1	18	84350	4218
12	302.2	1000	1120	1.0	560	1120000	560000
	302.0	950	1100	0.9	495	1045000	470250
	301.5	670	790	0.6	237	529300	158790
	301.0	440	510	0.4	102	224200	37400
13	301.0	1000	1370	0.8	548	1370000	548000
	300.5	700	1150	0.6	345	805000	241500
	300.0	420	670	0.4	134	281400	56280
	299.5	120	210	0.1	11	25200	1260
14	299.2	530	1150	0.4	230	609500	121900
	299.0	420	940	0.3	141	394800	59220
	298.75	260	590	0.2	59	153400	15340
	298.6	150	380	0.1	19	57000	2850
15	298.0	600	1150	0.8	460	690000	276000
	297.5	420	670	0.5	168	281400	70350
	297.0	230	255	0.3	38	58650	8798
	296.5	50	80	0.1	2	4000	100
16	296.9	1150	1150	0.8	460	1322500	529000
	296.5	830	860	0.5	215	713800	178450
	296.0	460	490	0.3	74	225400	33810
17	295.7	950	1150	0.8	460	1092500	437000
	295.5	850	1040	0.7	364	884000	309400
	295.0	540	650	0.5	163	351000	87750
	294.5	220	300	0.2	30	66000	6600
18	295.0	950	1150	0.8	460	1092500	437000
	294.75	850	1040	0.7	364	884000	309400
	294.25	540	650	0.5	163	351000	87750
	294.0	220	300	0.2	30	66000	6600

The above values provided the basis for a set of linear equations for each stage relating area flooded and storage volume, and area flooded in the unplanned flooding area. Table 21 lists these equations.

TABLE 21

Linear Equations for Area – Volume Relationships

Stage	Storage Volume Range	Equation
2	< - 4800	Area = Vol * 0.035
	4800 - 12000	Area = Vol * 0.014167 + 100.0
	12000 - 53280	Area = Vol * 1.0319767 + 11616.3
	53280 - 127200	Area = Vol * 0.8198051 + 22920.8
	127200 - 239250	Area = Vol * 0.5729585 + 54319.7
3	< - 960	Area = Vol * 5.00
	960 - 4550	Area = Vol * 1.3927576 + 3463.0
	4550 - 7650	Area = Vol * 3.6129032 - 6638.7
	7650 - 38208	Area = Vol * 1.0373715 - 13064.1
	38208 - 88150	Area = Vol * 0.7098233 + 25579.1
4	< - 750	Area = Vol * 8.00
	750 - 4410	Area = Vol * 5.2459016 + 2065.6
	4410 - 40000	Area = Vol * 0.6968249 + 22127.0
5	< - 158	Area = Vol * 19.94
	158 - 6760	Area = Vol * 4.6425325 + 2416.48
	6760 - 25650	Area = Vol * 2.7368978 + 15298.6
	25650 - 69120	Area = Vol * 1.7874396 + 39652.2
	69120 - 117800	Area = Vol * 1.4872637 + 60400.3
6	< - 700	Area = Vol * 10.0
	700 - 6900	Area = Vol * 3.3225806 + 4674.2
	6900 - 20825	Area = Vol * 2.2908438 + 11793.2
	20825 - 60000	Area = Vol * 1.5443522 + 27338.9
7	< - 5600	Area = Vol * 5.0
	5600 - 27300	Area = Vol * 2.9032258 + 11741.9
	27300 - 88920	Area = Vol * 1.7299578 + 43772.2
	88920 - 175175	Area = Vol * 1.4016578 + 72964.6
8	< - 660	Area = Vol * 20.0
	660 - 19140	Area = Vol * 4.4642857 + 10253.6
	19140 - 77550	Area = Vol * 2.7871939 + 42353.1
	77550 - 251250	Area = Vol * 1.4047207 + 149564.0
	251250 - 345950	Area = Vol * 1.3357972 + 166881.0
9	< - 1875	Area = Vol * 20.0
	1875 - 47880	Area = Vol * 4.3886534 + 29271.3
	47880 - 97500	Area = Vol * 3.0350665 + 94081.0
	97500 - 206790	Area = Vol * 2.7385854 + 122988.0
10	< - 18480	Area = Vol * 3.33
	18480 - 26880	Area = Vol * 8.6666667 - 98560.0
	26880 - 91200	Area = Vol * 3.5820895 + 38113.4

Table 21 continued

11	< - 4218 4218 - 50200 50200 - 84480 84480 - 103845	Area = Vol * 19.997 Area = Vol * 3.6242442 + 69063.0 Area = Vol * 0.8926487 + 206189.0 Area = Vol * 0.7797572 + 215726.1
12	< - 1870 1870 - 44880 44880 - 158790 158790 - 470250 470250 - 560000	Area = Vol * 20.0 Area = Vol * 4.347826 + 29269.6 Area = Vol * 2.6766745 + 104270.9 Area = Vol * 1.6557503 + 266383.5 Area = Vol * 0.8356545 + 652033.5
13	< - 1260 1260 - 56280 56280 - 241500 241500 - 548000	Area = Vol * 20.0 Area = Vol * 4.654885 + 19332.8 Area = Vol * 2.8269085 + 122301.6 Area = Vol * 1.8433931 + 359820.6
14	< - 2850 2850 - 15340 15340 - 59220 59220 - 121900	Area = Vol * 20.0 Area = Vol * 7.7181745 + 35003.2 Area = Vol * 5.5013673 + 69009.0 Area = Vol * 3.425335 + 191951.7
15	< - 100 100 - 8798 8798 - 70350 70350 - 276000	Area = Vol * 40.0 Area = Vol * 6.2830535 + 3371.7 Area = Vol * 3.6188913 + 26811.0 Area = Vol * 1.9868708 + 141623.7
16	< - 33810 33810 - 178450 178450 - 529000	Area = Vol * 6.6666667 Area = Vol * 3.3766592 + 111235.2 Area = Vol * 1.7364142 + 403936.9
17	< - 6600 6600 - 87750 87750 - 309400 309400 - 437000	Area = Vol * 10.0 Area = Vol * 3.5120147 + 42820.7 Area = Vol * 2.404692 + 139988.3 Area = Vol * 1.6340125 + 378436.6
18	< - 6600 6600 - 87750 87750 - 309400 309400 - 437000	Area = Vol * 10.0 Area = Vol * 3.5120147 + 42820.7 Area = Vol * 2.404692 + 139988.3 Area = Vol * 1.6340125 + 378436.6

C.3 STORAGE DECISION LEVELS

Using the available storage capacity range (zero to storage maximum) six storage levels, which are the potential storage decisions, were calculated simply by dividing the range into five and rounding the values to the nearest thousand. The decision of no storage makes up the sixth potential decision. Table 22 shows these values.

TABLE 22

Storage Levels Available for Decision Variables

Stage	Storage Levels (m^3)					
4	0	8000	16000	24000	32000	40000
5	0	25000	50000	75000	100000	117800
6	0	12000	24000	36000	48000	60000
7	0	35000	70000	105000	140000	175275
8	0	70000	140000	210000	280000	345950
9	0	40000	80000	120000	160000	206790
10	0	18000	36000	54000	72000	91200
11	0	20000	40000	60000	80000	103845
12	0	112000	224000	336000	448000	560000
13	0	110000	220000	330000	440000	548000
14	0	25000	50000	75000	100000	121900
15	0	55000	110000	165000	220000	176000
16	0	110000	220000	330000	440000	529000
17	0	87000	174000	261000	348000	437000
18	0	0	0	0	0	0

At stage 11, the dyke configuration and the storage area is somewhat different from the other stages. This is due to the Canadian National Railway mainline meeting Highway 5 at stage 11, creating a set of barriers capable of holding water for the duration necessary for the types of floods on this creek.

Appendix D

CROP VALUES

Appendix D includes information pertaining to crop values including crop types, crop returns, production costs, and effects of soil type differences.

D.1 CROP TYPES

Based on soil type, drainage qualities and climatological information, Manitoba Department of Agriculture (1983) classifies the agricultural regions of Manitoba by their Agricultural Capability Class. Based on class type, a variety of crops are recommended for farming in each region. The crops recommended for the area which includes Wilson Creek are wheat, feed grains, oil seeds, legumes and grasses. From these, wheat, barley, flax and alfalfa were chosen. To some extent the choices were arbitrary, but barley was chosen because some information on damage variation due to duration of flooding variation was available. Otherwise, the crops are common in the region and are representative of the suggested crop types.

D.1.1 Crop Values

From crop value information from Manitoba Department of Agriculture (1983) and University of Manitoba (1983), crop returns were determined using average yield, start-up costs and crop prices. Table 23 shows average yields for wheat, barley and flax in kg/acre for fallow and stubble fields.

TABLE 23

Average Crop Yields

Crop	Yield (kg/acre) Fallow	Stubble
wheat	675	805
barley	849	869
flax	325	721

Assuming the practice of 1 fallow year in 4, the average yield over a 10 year period (including 2 fallow and 8 stubble years) is listed in Table 24. Crop prices in \$ per tonne, from Canadian Grain Commission (1983) are listed in Table 25. From Tables 24 and 25, the gross returns per hectare for each crop are calculated. These are listed in Table 26.

TABLE 24

Average Crop Yields Over 10 Years

Crop	Yield kg/acre	tonnes/hect
wheat	708	1.747
barley	898	2.218
flax	345	0.852
alfalfa	-	4.950

* (average yield)

TABLE 25

Crop Prices

Crop	Price ($/tonne)
wheat	205.00
barley	180.00
flax	382.00
alfalfa	30.04

TABLE 26

Gross Returns

Crop	Gross Returns ($1983) ($/hect)
wheat	358.14
barley	399.24
flax	325.46
alfalfa	148.67

From the gross returns, the start costs are removed.

Start costs for a ten year average period are taken from University of Manitoba (1983). They include machinery costs (11% depreciation), costs of fuel and oil, seed, fertilizers and expenses during fallow years. Table 27 shows the start costs.

TABLE 27

Start Costs

Crop	Start Costs ($1983/acre)	
	Fallow	Stubble
wheat	46.5	51.8
barley	45.3	48.6
flax	46.5	47.8

Table 28 shows the start costs for average over 10 years, and converted to $/hect. These values are then removed from the gross values, as listed in Table 29.

TABLE 28

Start Costs

Crop	$/acre	$/hect
wheat	50.74	125.33
barley	47.90	118.31
flax	47.54	117.42
alfalfa	25.00	61.75

*note: alfalfa is an assumed value based on the reduced use of seed in this perennial.

TABLE 29

Net Crop Returns

Crop	Gross Return ($/hect)	Start Costs ($/hect)	Net Returns ($/hect)
wheat	358.14	125.33	232.81
barley	399.24	118.31	280.93
flax	325.46	117.42	208.04
alfalfa	148.67	61.75	86.92

*note: alfalfa value based on assumed values, for one cutting in the season. Increased values due to two cuttings are considered within the model.

D.2 REDUCTION IN VALUE DUE TO SOIL QUALITY

The above crop values are averages for southern Manitoba. There is also some reduction in value due to poor soil quality. Soil types, as detailed for each stage in Appendix C, are derived from Manitoba Crop Insurance Corporation (1980), as are the relative productivities for each type. Table 30 shows the variation in yield (in kg/acre) for the soil types in the region around Wilson Creek for various levels of seed coverage.

TABLE 30

Yield Variation Due to Soil Type

Soil Type	Coverage 50%			60%			70%		
	w	b	f	w	b	f	w	b	f
B09	368	362	148	441	435	178	514	508	208
C09	362	362	144	435	435	173	508	508	202
D09	347	347	142	416	416	170	485	485	198
E09	333	327	125	400	392	150	467	457	175
F09	315	305	121	378	366	145	441	427	169
G09	308	305	108	370	366	130	432	427	152
H09	263	281	99	316	337	119	369	393	139
I09	254	254	78	305	305	94	356	356	110
J09	113	182	44	136	218	53	159	254	62

* w= wheat b= barley f= flax

Regardless of the coverage option, the relative differences between yields from one soil quality to another is the same. This relationship exists for all three crops. Using these values, the assumption is made that since soil type B09 is of the best quality, it represents a 100% yield. The remaining soil types are then some value less than 100% and can be represented by a value of less than 1.0 to facilitate a reduction in crop net returns by multiplying the soil index by the crop net return. The soil quality types vary with the stages, and therefore the actual calculation for reduction in net returns due to soil quality is carried out as the net benefit calculations are performed at each stage. See Appendix A for these calculations. The yield variations with soil quality types are shown in Table 31.

TABLE 31

Soil Zone Yield Variations

Soil	Crop		
	Wheat	Barley (% of maximum)	Flax
B09	100	100	100
C09	99	100	97
D09	94	95	95
E09	91	90	84
F09	86	84	81
G09	84	84	73

No data was available from which to determine a relationship between soil type and reduction in yield for alfalfa. Therefore, it is assumed that there is no variation in alfalfa yields due to soil type differences.

Appendix E

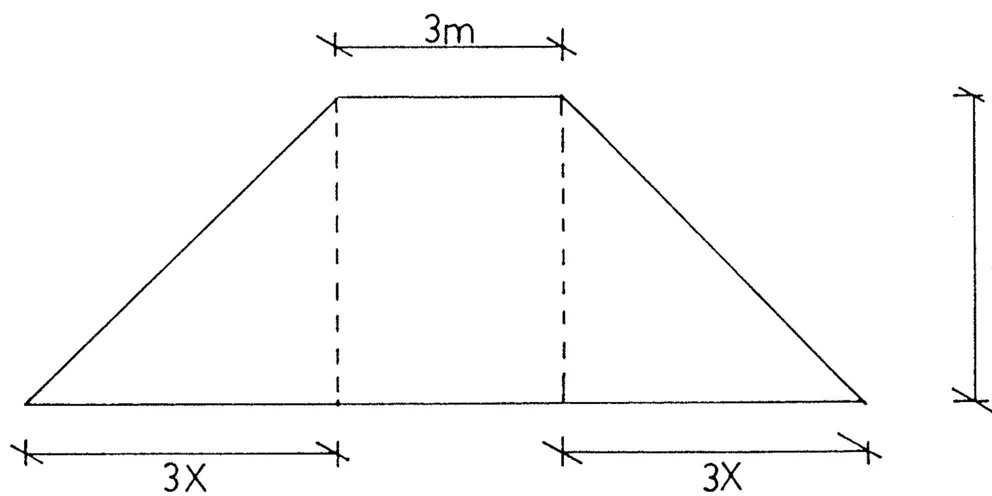
DYKE CALCULATIONS

Figure 18 shows the dyke dimensions used in this thesis. The shape and cross sectional dimensions were assumed so that a dyke volume value could be determined, in order to establish dyke costs. The dyke costs are also used as storage costs.

The dyke is designed to be 3m in width at the top, with a side slope length of three times the dyke height. A freeboard value of 0.5m is used. When a storage decision is made, the required height and area of the storage area is calculated from cross sections of the stages, as detailed in Appendix C. The length of dyke required, and the height required are therefore known. The dyke design then allows for the calculation of total dyke volume, based on the required height and length and the average cross sectional area of the dyke.

Since the dyke is perpendicular to the flow, the dyke length is related to the required width of the cross section of the stage as determined in Appendix C. A series of dyke heights is calculated from a range of possible water elevations between the minimum (zero) and the maximum water storage elevation for each stage. From these, linear equations

Figure 18: Dyke Design



X = Average dyke height + freeboard

relating storage volumes and dyke volumes are determined. The dyke volumes are used to calculate dyke costs, which are used for the storage cost values. Table 32 lists the elevations from which the dyke heights are determined, and the corresponding storage volumes used for calculating dyke volumes.

The values for the dyke volume and storage volume relationships were used to calculate linear equations for dyke volume - storage volume calculations. From these, values for required dyke volumes for each potential storage decision at each stage were determined. Table 33 lists the dyke volumes by storage decision.

The cost per volume for the dykes was not given a specific value. It is expected that dyke costs will range between \$1.50 and \$2.00 per m^3, which is used in the analysis. Dyke costs are also expected to be a major factor in the decision process. Therefore the per volume value of the dyke must be variable in order to test some aspects of the model, as described in Chapter 5.

TABLE 32

Dyke Volume - Storage Volume Relationships

Stage	Elevation (m)	Dyke Width (m)	Dyke Height (m)	Dyke Cross Sect (m^2)	Storage Volume (m^3)	Dyke Volume (m^3)
2	342.9	660	2.00	18.00	239250	11880
	342.0	530	1.60	12.48	127200	6614
	341.0	370	1.10	6.93	53280	1247
	340.0	200	0.50	2.25	12000	270
	339.5	120	0.40	1.68	4800	168
3	328.0	410	2.20	21.12	88150	8660
	327.0	310	1.80	15.12	38208	4687
	326.0	210	1.30	8.97	7650	1884
	325.5	140	0.90	5.13	4550	718
	325.0	120	0.65	3.22	960	386
4	317.0	500	0.80	7.80	50000	3900
	316.5	360	0.40	5.40	25200	1944
	316.0	200	0.25	4.50	6000	900
5	315.8	310	1.38	9.85	117800	3054
	315.5	255	0.83	4.56	69120	1162
	315.0	190	0.75	3.94	25650	748
	314.5	130	0.25	0.93	6760	122
	314.0	45	0.18	0.63	158	29
6	312.6	240	1.15	7.42	60000	1376
	312.0	170	0.95	5.56	20825	657
	311.5	120	0.70	3.57	6900	270
	311.0	70	0.40	1.68	700	82
7	310.8	490	0.97	5.73	175175	3302
	310.5	380	0.94	5.47	88920	2079
	310.0	260	0.62	3.01	27300	783
	309.5	140	0.29	1.12	5600	157
8	309.2	740	0.96	5.65	345950	4181
	309.0	670	0.88	4.96	251250	3325
	308.5	470	0.73	3.78	77550	1781
	308.0	290	0.43	1.84	19140	535
	307.5	110	0.20	0.72	660	80
9	307.0	1130	1.30	8.97	206790	10136
	306.75	780	1.20	7.92	97500	6178
	306.5	630	0.90	5.13	47880	3232
	306.0	250	0.70	3.57	1875	893

Table 32 continued

10	304.8 304.5 304.25	1140 640 440	1.00 0.90 0.70	6.00 5.13 3.57	91200 26880 18480	6848 3283 1571
11	303.6 303.5 303.3 303.0	46 46 46 46	0.80 0.70 0.50 0.20	4.32 3.57 2.25 0.72	103845 84480 50200 4218	199 165 104 33
12	302.2 302.0 301.5 301.0 300.5	1120 1100 790 510 220	1.05 1.40 1.10 1.00 0.60	11.25 10.08 6.93 6.00 2.88	560000 470250 158790 44880 1870	12600 11088 5475 3060 634
13	301.0 300.5 300.0 299.5	1370 1150 670 210	1.40 1.10 0.90 0.60	10.08 6.93 5.13 2.88	548000 241500 56280 1260	13810 7970 3437 605
14	299.2 299.0 298.75 298.6	1150 940 590 380	0.95 0.80 0.70 0.60	5.56 4.32 3.57 2.88	121900 59220 15340 2850	6391 4061 2106 1094
15	298.0 297.5 297.0 296.5	1150 670 255 80	1.30 1.10 0.80 0.60	8.97 6.93 4.32 2.88	276000 70350 8798 100	10316 4643 1102 230
16	296.9 296.5 296.0	1150 860 490	1.20 1.05 0.80	7.92 6.45 4.32	529000 178450 33810	9108 5553 2117
17	295.7 295.5 295.0 294.5	1150 1040 650 300	1.30 1.20 1.00 0.70	8.97 7.92 6.00 3.57	437000 309400 87750 6600	10316 8237 3900 1071

TABLE 33

Dyke Volumes for Potential Storage Volumes

Stage	Storage Volumes (m^3)	Dyke Volumes (m^3)
	0	0
2	50000	1170
	100000	4640
	150000	7690
	200000	10040
	239250	11880
	0	0
3	20000	3020
	40000	4830
	60000	6420
	80000	8010
	88150	8660
	0	0
4	8000	2100
	16000	2600
	24000	3000
	32000	3500
	40000	4000
	0	0
5	25000	730
	50000	980
	75000	1390
	100000	2360
	117800	3050
	0	0
6	12000	410
	24000	1430
	36000	1650
	48000	1870
	60000	2100
	0	0
7	35000	940
	70000	1680
	105000	2310
	140000	2800
	175175	3300

Table 33 continued

8	0 70000 140000 210000 280000 345950	0 1600 2300 3000 3600 4200
9	0 40000 80000 120000 160000 206790	0 2800 5100 7000 8400 10000
10	0 18000 36000 54000 72000 91200	0 1500 3800 4800 5800 6800
11	0 20000 40000 60000 80000 103845	0 60 90 120 160 200
12	0 112000 224000 336000 448000 560000	0 4500 6700 8700 11000 12600
13	0 110000 220000 330000 440000 548000	0 4800 7400 9700 11800 13800
14	0 25000 50000 75000 100000 121900	0 2500 3700 4600 5600 6400

Table 33 continued

15	0 55000 110000 165000 220000 276000	0 3800 5700 7300 8800 10300
16	0 110000 220000 330000 440000 529000	0 4000 6000 7200 8300 9100
17	0 87000 174000 261000 348000 437000	0 3900 5600 7300 8900 10300

Note: Dyke volume values are rounded to the nearest thousand

Appendix F

FREQUENCY ANALYSIS

Using the standard log normal method, frequency analyses were carried out on annual peaks, as well as on the peak flows from each of the 5 time periods. Twenty three data points were available for each time period, from 1959 to 1981 inclusive. The available data is in standard units, so the analysis was carried using these units, and converted to metric later. Table 34 shows the data used for the six frequency analyses.

Using these data, frequency analysis were carried out. Table 35 shows the annual peak analysis. Tables 36 through 40 show the analyses for the five time periods.

Figure 19 presents the frequency analysis in frequency curve form.

TABLE 34

Annual Peak Flows and Peak Flows for the Five Time Periods

Year	Annual Peaks (ft^3/s)	Per 1 Peaks (ft^3/s)	Per 2 Peaks (ft^3/s)	Per 3 Peaks (ft^3/s)	Per 4 Peaks (ft^3/s)	Per 5 Peaks (ft^3/s)
1959	52.9	52.9	29.6	17.9	1.4	28.4
1960	115.0	79.5	115.0	6.3	4.1	1.4
1961	41.0	41.0	3.8	10.2	0.2	0.6
1962	160.0	6.2	160.0	12.3	15.4	4.9
1963	253.0	35.3	211.9	253.0	23.3	12.9
1964	63.0	63.0	6.3	11.4	0.6	8.8
1965	126.5	126.5	68.5	3.9	9.4	111.6
1966	118.0	118.0	39.0	18.0	114.1	1.6
1967	70.0	70.0	35.1	10.5	2.6	0.1
1968	166.0	64.3	19.1	69.7	166.0	29.6
1969	700.0	18.9	39.0	700.0	23.0	17.0
1970	360.0	360.0	75.5	22.7	58.0	15.7
1971	737.0	10.4	737.0	46.6	52.3	20.9
1972	29.0	29.0	5.2	3.4	5.0	3.7
1973	63.0	35.7	22.8	63.0	26.3	35.5
1974	288.0	288.0	50.1	5.8	2.4	4.1
1975	1580.0	99.6	92.9	14.0	260.0	1580.0
1976	22.1	22.1	19.3	18.4	2.8	0.8
1977	540.0	25.7	19.3	540.0	8.7	58.0
1978	36.4	36.4	5.7	6.0	0.5	32.5
1979	355.0	355.0	116.2	19.7	4.0	11.9
1980	215.7	5.6	2.7	11.4	215.7	5.7
1981	39.9	3.9	39.9	24.4	1.9	6.6

TABLE 35

Frequency Analysis for Annual Peaks

Peak (cfs)	Log Peak	Log of Square of Peak
1580.0	3.198657	10.231406
737.0	2.867467	8.222366
700.0	2.845098	8.094582
540.0	2.732394	7.465976
360.0	2.556302	6.534679
355.0	2.550228	6.503662
288.0	2.459392	6.048609
253.0	2.403120	5.774985
216.0	2.333931	5.447233
166.0	2.220108	4.928879
160.0	2.204120	4.858144
126.5	2.102090	4.418782
118.0	2.071882	4.292695
115.0	2.060698	4.246476
70.0	1.845098	3.404386
63.0	1.799340	3.237624
63.0	1.799340	3.237624
41.0	1.612784	2.601072
39.9	1.600973	2.563114
36.4	1.560743	2.435910
29.6	1.471292	2.164700
27.0	1.431525	2.049260
22.1	1.344392	1.807380
	Σ = 49.07097	Σ = 110.56954

mean of log peak values = 2.1335206
standard deviation of
log peak values = 0.5167912
mean + / - standard deviation = (+) 2.6503118 = 12.66 m^3/sec
(-) 1.6167294 = 1.17 m^3/sec

TABLE 36

Frequency Analysis for Period One Peaks

Peak (cfs)	Log Peak	Log of Square of Peak
360.0	2.5563	6.5347
355.0	2.5502	6.5037
288.0	2.4594	6.0486
120.5	2.1021	4.4188
118.0	2.0719	4.2927
99.6	1.9981	3.9925
79.5	1.9001	3.6104
70.0	1.8451	3.4044
64.3	1.8079	3.2686
63.0	1.7993	3.2376
52.9	1.7235	2.9703
41.0	1.6128	2.6011
36.4	1.5607	2.4359
35.7	1.5527	2.4108
35.3	1.5479	2.3960
29.0	1.4625	2.1390
28.7	1.4099	1.9879
22.1	1.3444	1.8074
18.9	1.2771	1.6311
10.4	1.0179	1.0361
6.2	0.7952	0.6323
5.6	0.7466	0.5575
3.9	0.5855	0.3428
	Σ = 37.72700	Σ = 68.260200

mean of log peak values = 1.6403086
standard deviation of
log peak values = 0.5383523
mean + / - standard deviation = (+) 2.6503118 = 4.27 m^3/sec
(-) 1.6167294 = 0.36 m^3/sec

TABLE 37

Frequency Analysis for Period Two Peaks

Peak (cfs)	Log Peak	Log of Square of Peak
737.0	2.8675	8.2224
211.9	2.3260	5.4105
160.0	2.2041	4.8581
116.2	2.0651	4.2646
115.0	2.0607	4.2465
92.9	1.9682	3.8738
75.5	1.8779	3.5267
68.5	1.8357	3.3698
50.1	1.6998	2.8894
39.9	1.6010	2.5631
39.0	1.5911	2.5315
39.0	1.5911	2.5315
35.1	1.5433	2.3880
29.6	1.4713	2.1647
22.8	1.3579	1.8440
19.3	1.2856	1.6527
19.3	1.2856	1.6527
19.1	1.2810	1.6410
6.3	0.7993	0.6384
5.7	0.7520	0.5656
5.2	0.7160	0.5127
3.8	0.5809	0.3375
2.7	0.4281	0.1833
	Σ = 35.1912	Σ = 61.869

mean of log peak values = 1.5300520
standard deviation of
log peak values = 0.6039501
mean + / - standard deviation = (+) 2.1340021 = 3.89 m^3/sec
(-) 0.9261019 = 0.24 m^3/sec

TABLE 38

Frequency Analysis for Period Three Peaks

Peak (cfs)	Log Peak	Log of Square of Peak
700.0	2.8451	8.0946
540.0	2.7324	7.4660
253.0	2.4031	5.7750
69.7	1.8432	3.3975
63.0	1.7993	3.2376
46.6	1.6684	2.7835
24.4	1.3876	1.9224
22.7	1.3568	1.8409
19.7	1.2945	1.6736
18.4	1.2655	1.6016
18.0	1.2553	1.5757
17.9	1.2529	1.5696
14.0	1.1461	1.3136
12.3	1.0906	1.1894
11.4	1.0584	1.1203
11.4	1.0570	1.1170
10.5	1.0191	1.0386
10.2	1.0086	1.0173
6.3	0.7993	0.6389
6.0	0.7810	0.6100
6.2	0.7952	0.6323
5.8	0.7657	0.5862
3.9	0.5888	0.3467
	Σ = 30.9493	Σ = 50.201

mean of log peak values = 1.3456217
standard deviation of
log peak values = 0.6235875
mean + / - standard deviation = (+) 1.9692092 = 2.64 m^3/sec
(-) 0.7220342 = 0.15 m^3/sec

TABLE 39

Frequency Analysis for Period Four Peaks

Peak (cfs)	Log Peak	Log of Square of Peak
260.0	2.4150	5.8321
215.7	2.3339	5.4472
166.0	2.2201	4.9289
114.1	2.0573	4.2324
58.0	1.7634	3.1097
52.3	1.7185	2.9532
26.3	1.4200	2.0163
23.3	1.3674	1.8697
23.0	1.3617	1.8543
15.4	1.1875	1.4102
9.6	0.9708	0.9424
8.7	0.9370	0.8780
5.0	0.6990	0.4886
4.1	0.6085	0.3703
4.0	0.6010	0.3612
2.8	0.4393	0.1930
2.6	0.4116	0.1694
2.4	0.3876	0.1501
1.9	0.2878	0.0828
1.4	0.1492	0.0223
0.6	-0.2518	0.0634
0.5	-0.2757	0.0760
0.2	-0.7696	0.5992
	Σ = 22.0393	Σ = 38.0437

mean of log peak values = 0.9582304
standard deviation of
log peak values = 0.7693
mean + / - standard deviation = (+) 1.7275304 = 1.51 m^3/sec
(-) 0.1889304 = 0.04 m^3/sec

TABLE 40

Frequency Analysis for Period Five Peaks

Peak (cfs)	Log Peak	Log of Square of Peak
1580.0	3.1987	10.2314
111.6	2.0475	4.1925
58.0	1.7634	3.1097
35.5	1.5505	2.4040
32.5	1.5122	2.2866
29.6	1.4713	2.1647
28.4	1.4530	2.1112
20.9	1.3201	1.7428
17.0	1.2304	1.5140
13.7	1.1959	1.4302
12.9	1.1106	1.2334
11.9	1.0741	1.1537
8.8	0.9420	0.8874
6.6	0.8176	0.6684
5.7	0.7574	0.5736
4.9	0.6902	0.4764
4.1	0.6128	0.3753
3.7	0.5694	0.3242
1.6	0.1903	0.0362
1.4	0.1399	0.0196
0.8	-0.1079	0.0116
0.6	-0.2218	0.0492
0.1	-0.8534	0.7291
	Σ = 22.4937	Σ = 37.7254

mean of log peak values = 0.9779869
standard deviation of
log peak values = 0.8454975
mean + / - standard deviation = (+) 1.8234844 = 1.89 m^3/sec
(-) 0.1324894 = 0.04 m^3/sec

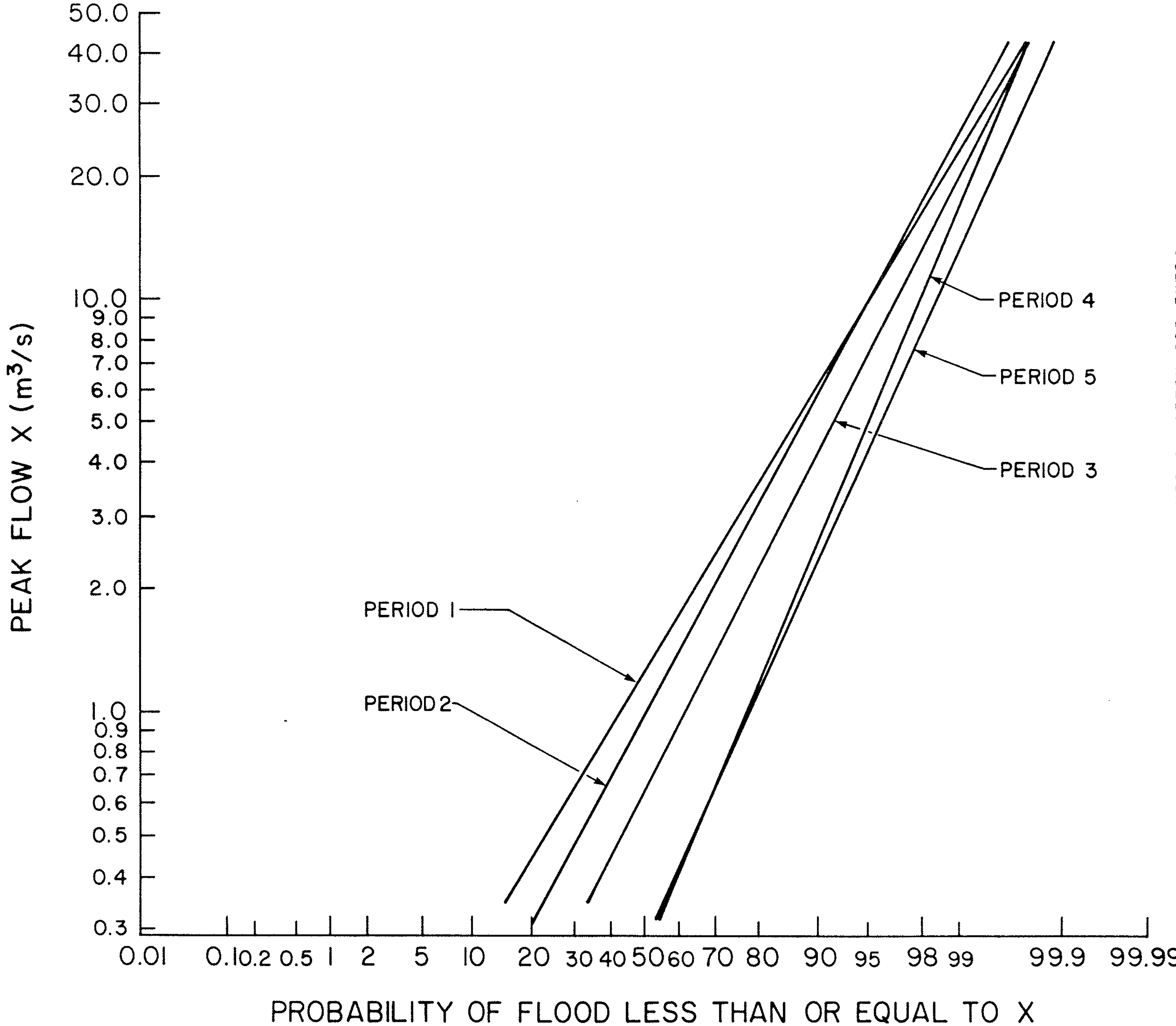


Figure 19: Flow Frequency for Annual Peaks, Periods 1 - 5 Peaks for Wilson Creek

F.1 PROBABILITY BANDS

From the frequency curve for annual peaks, the flow values for the 5, 10, 20, 40 and 50 year return period floods were determined. These peak flows were used to derive the probabilities of exceedence of floods of the same magnitudes for each of the five time periods of the growing season. Table 41 shows the peak flow values, and the probabilities of exceedence for each time period.

TABLE 41

Probabilities of Exceedence for the Design Floods for the Time Periods

Annual Return Period (yrs.)	Peak Flow (m^3/s)	Probability of Exceedence	Time Period Probabilities of Exceedence				
			Per 1	Per 2	Per 3	Per 4	Per 5
5	10.8	0.2	0.05	0.05	0.025	0.017	0.013
10	20.0	0.1	0.013	0.015	0.009	0.007	0.005
20	32.0	0.05	0.005	0.007	0.004	0.004	0.002
40	45.0	0.025	0.002	0.003	0.002	0.002	0.001
50	55.0	0.020	0.001	0.002	0.001	0.001	0.00008

To properly assess the expected damage, the five hydrographs must be representative of the full spectrum of flood magnitudes. Bands of probability for each flood level were derived to facilitate this. These were calculated by determining the midpoint between each probability of exceedence, which are the upper and lower extremes of each band. Table 42 shows these calculations.

TABLE 42

Probability Band Calculations

Time Periods	Probabilities of Exceedence	Range (upper and lower bounds)	Band Width
		0.0000	
			0.39
		0.61	
1	0.05		0.5785
		0.0315	
	0.013		0.0225
		0.0090	
	0.005		0.0055
		0.0035	
	0.002		0.0020
		0.0015	
	0.001		0.0010
		0.0005	
			0.0005
		infinity	
		0.0000	
			0.50
		0.50	
2	0.05		0.4675
		0.0325	
	0.015		0.0215
		0.0110	
	0.007		0.0060
		0.0050	
	0.003		0.0025
		0.0025	
	0.002		0.0010
		0.0015	
			0.0015
		infinity	

(continued next page)

Table 42 continued

3	0.025 0.009 0.004 0.002 0.001	0.0000 0.31 0.0170 0.0065 0.0030 0.0015 0.0005 infinity	0.69 0.2930 0.0105 0.0035 0.0015 0.0010 0.0005
4	0.017 0.007 0.004 0.002 0.001	0.0000 0.21 0.1200 0.0055 0.0030 0.0015 0.0005 infinity	0.79 0.1980 0.0065 0.0025 0.0015 0.0010 0.0005
5	0.013 0.005 0.002 0.001 0.0008	0.0000 0.22 0.0090 0.0035 0.0015 0.0009 0.0007 infinity	0.78 0.2110 0.0055 0.0020 0.0006 0.0002 0.0007

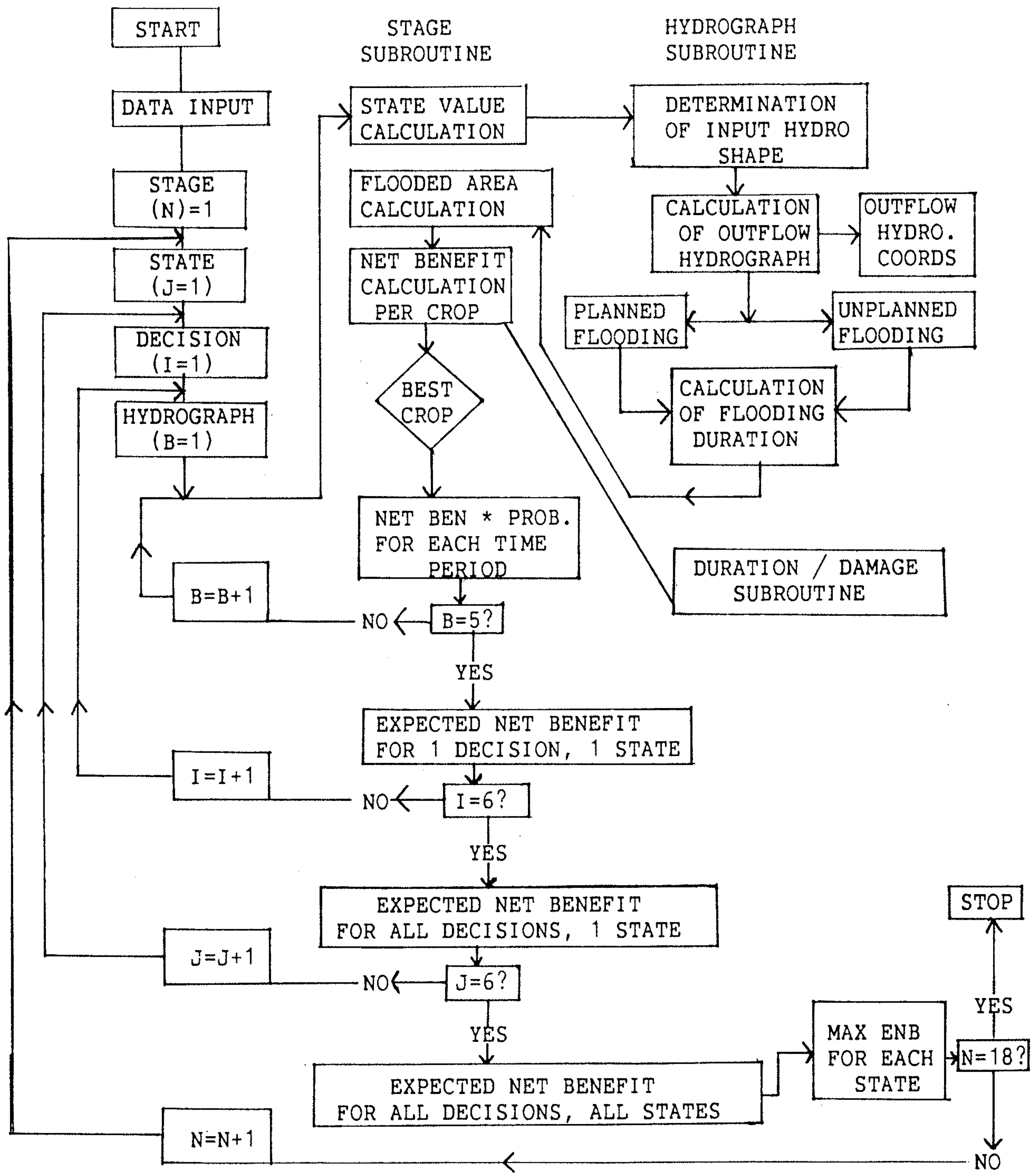
Appendix G

THE PROGRAM

The program is written in the Watfiv dialect of Fortran. The program is listed in its entirety on the following pages. Comments have been included in appropriate places to make it easy to follow.

Figure 20 is a flow chart of the program.

Figure 20: Flow Chart for the Model Program



```
//HANNAN JOB '1103028,,T=1M,I=2,L=150','HANNAN' MSGLEVEL=(1,1), CLASS=x
// EXEC WATFIV,SIZE=512K
//SYSIN DD *
$JOB  WATFIV HANNAN,NOEXT
      INTEGER N,J,A,B,I,K,LENG,C,R,DECIS,D,Z,Q,POSN,INC,O
C ****************************************************************
C N= SUBSCRIPT FOR STAGE R=1,18
C J=SUBSCRIPT FOR STATE VAR R=1,6
C I= SUBSCRIP FOR STORAGE DEC.,(SVOL) R=1,6
C A= SUB FOR GROWING SEASON DIVISIONS R=1,5
C B= SUB FOR HYDROGRAPH R=1,6
C K= SUB FOR BUILDING THE STAGE MATRIX
C LENG= A VALUE FOR THE SORT PROGRAM TO TELL HOW MANY VALUES TO SORT
C******************************************************************
      REAL STAT(18,6),BESTF,BESTU,BESTN,DYKE(18),D1(18,6),
     *     DCOST,ADCOS,P(6,5),NB(5),ENB(6),SVOL(18,6),BEN,MAX,
     *     STG(18,6,8),JPB(5,13),JPW(5,13),JPF(5,13),SUM,
     *     ADD(6),MADD,LINC,GINC,PREDEC,POSS
C *************************************************************
C X(5,2)= X COORDS OF THE HYDRO. 5 HYDROS, 2 X-COORDS
C Y(5)= Y COORD OF HYDRO. 5 HYDROS, 1 COORD
C STAT(18,6)= STATE VARIABLE. 18 STAGES, 6 STATES PER.
C BESTF= BEST VALUE FOR THE PLANNED FLOODING AREA.
C BESTU= SAME FOR AREA OF UNPLANNED FLOODING
C BESTN= SAME FOR AREA OF NO FLOODING
C       NOTE, THESE ARE CALCULATED IN EACH STAGE SUBROUTINE
C D1(18,6)= CALCULATED DYKE VOLUME REQUIRED FOR SVOL
C DCOST= COST OF DYKE
C P(6,5)= PROB OF HYDRO(1 TO 6) IN TIME(1 TO 5)
C NB(5)= NET BENEFIT VAL FOR THE STAGE FOR 1 TIME PER
C NB= SUM OF ALL NB(A) VALUES OVER ALL TIME PERS
C ENB= EXPECTED NET BEN¬, IE. SUMMED OVER ALL HYDROS
C SVOL(18,5)= STORAGE DECISION, FOR 18 STAGES,5(?) DEC VAR
C MAX= BEST DECISION VALUE
C DECIS= BEST DECISION
C STG(18,6,3)= A VAR FOR STORING VALUES IN THE STAGE MATRICES
C JPB(A,C)= JOINT PROB MAT FOR BARLEY
C JPW(A,C)= JOINT PROB MAT FOR WHEAT
C JPF(A,C)= JOINT PROB MAT FOR FLAX
      READ 80, ((JPB(A,C),C=1,13),A=1,5)
C     PRINT 81
      READ 80, ((JPW(A,C),C=1,13),A=1,5)
C     PRINT 83
      READ 80, ((JPF(A,C),C=1,13),A=1,5)
C     PRINT 84
      READ 85, ((SVOL(N,I),I=1,6),N=1,18)
C     PRINT 86
      READ 88, ((P(B,A),A=1,5),B=1,6)
C     PRINT 89
80    FORMAT(7F9.7/6F9.7)
C 81  FORMAT(' ', 25X, 'JOINT PROBABILITY MATRIX FOR BARLEY')
C 83  FORMAT(' ', 25X, 'JOINT PROBABILITY MATRIX FOR WHEAT')
C 84  FORMAT(' ', 25X, 'JOINT PROBABILITY MATRIX FOR FLAX')
85    FORMAT(6F9.1)
C 86  FORMAT(' ', 25X, 'STORAGE DECISIONS')
```

```
88     FORMAT(5F7.4)
C 89   FORMAT(' ',25X, 'FLOOD PROBABILITY RANGES')
       READ 99, ((D1(N,I),I=1,6),N=1,18)
 99    FORMAT(6F8.1)
C *************************************************************
C              THIS LOOPS THRU STAGES
C *************************************************************
        PRINT, 'DCOST = 1.50'
        PRINT, 'BARLEY = 0.00'
        PRINT, 'WHEAT = 0.0'
        PRINT, 'FLAX = 0.0'
        PRINT, 'ALFALFA = 0'
       DO 30  N=1,18
C ***********************************************************
C              THIS LOOPS THRU STATES
C *************************************************************
       DO 40  J=1,6
C *************************************************************
C              THIS LOOPS THRU STORAGE DECISIONS
C *************************************************************
       DO 50  I=1,6
       BEN=0.0
C ***************************************************************
C              THIS LOOPS THRU HYDROGRAPHS
C *******************************************************************
       DO 20  B=1,5
C ***************************************************************
C    THIS PART SENDS PROGRAM TO NECESSARY PART TO
C    CALCULATE RETURN FUNCTION VALUES FOR EACH STAGE.
C ***************************************************************
       IF(N.LE.3) THEN DO
         SVOL(N,I)=0.0
         DYKE(N)=0.0
         STAT(N,J)=0.0
         GO TO 39
       END IF
       IF(N.GE.4.AND.N.LT.18) THEN DO
         GO TO 4
       END IF
       IF(N.EQ.18) THEN DO
         GO TO 18
       END IF
C ************************************************************
C THIS SENDS THE PROGRAM TO THE STAGE SUBROUTINE FOR CALCULATION
C OF VALUES UNIQUE TO THE STAGE CURRENTLY UNDER CONSIDERATION
C **************************************************************
4      CONTINUE
       IF(N.EQ.4) THEN DO
         IF(J.GT.1) THEN DO
           GO TO 39
         END IF
       END IF
       CALL STAG(SVOL,BESTF,BESTU,BESTN,STAT,JPB,JPW,JPF,I,J,N,B,INC)
       IF(I.EQ.1) THEN DO
         DYKE(N)=D1(N,I)
```

```
      END IF
      IF(I.EQ.2) THEN DO
        DYKE(N)=D1(N,I)
      END IF
      IF(I.EQ.3) THEN DO
        DYKE(N)=D1(N,I)
      END IF
      IF(I.EQ.4) THEN DO
        DYKE(N)=D1(N,I)
      END IF
      IF(I.EQ.5) THEN DO
        DYKE(N)=D1(N,I)
      END IF
      IF(I.EQ.6) THEN DO
        DYKE(N)=D1(N,I)
      END IF
      GO TO 70
18    CONTINUE
      SVOL(N,I)=0.0
      DYKE(N)=0.0
      CALL STAG(SVOL,BESTF,BESTU,BESTN,STAT,JPB,JPW,JPF,I,J,N,B,INC)
      GO TO 70
C ***********************************************************
C                THIS PART CALCULATES ALL THE
C                RETURN FUNCTION VALUES FOR
C                ANY STAGE.
C ***********************************************************
70    CONTINUE
      ADCOS=DYKE(N)* 1.50
      DCOST=ADCOS*0.08174
C *****************************************************************
C     HERE IT LOOPS THRU SEASONS
C *****************************************************************
      SUM=0.0
      DO 60  A=1,5
      NB(A)=(BESTF+BESTU+BESTN)*P(B,A)+SUM
C     PRINT, 'PROB=',P(B,A),'FOR A=',A
      SUM=NB(A)
60    CONTINUE
      A=5
      BEN=BEN+NB(A)+((BESTF+BESTU+BESTN)*P(6,A))
      ENB(I)=BEN
C     PRINT, 'ENB BEFORE DYKE REM=',ENB(I)
C *****************************************************************
C      TO THIS POINT, THE NET BEN FOR 1 HYDRO IS
C      CALCULATED. ONCE THIS NEXT LOOP IS CALCULATED
C      1 CELL OF THE STAGE MATRIX IS CALCULATED.
C *****************************************************************
20    CONTINUE
      ENB(I)=ENB(I)-DCOST
C      PRINT, 'ENB=',ENB(I),' FOR I=',I
      PRINT, 'DYKE COST=',ADCOS
      K=I
      IF(N.EQ.4) THEN DO
        STG(N,J,K)=ENB(I)
```

```
      END IF
      IF(N.EQ.5) THEN DO
        ENB(I)=ENB(I)+STG(N-1,1,7)
        STG(N,J,K)=ENB(I)
        PREDEC=STG(N-1,1,8)
      END IF
      IF(N.GT.5) THEN DO
        IF(J.EQ.1) THEN DO
          ENB(I)=ENB(I)+STG(N-1,1,7)
          STG(N,J,K)=ENB(I)
          PREDEC=STG(N-1,1,8)
        END IF
        IF(J.GT.1) THEN DO
          DO 15 Z=1,6
          DO 16 Q=1,6
          POSS=STAT(N-1,Z)+SVOL(N-1,Q)
          LINC=STAT(N,Z)-(INC/2.0)
          GINC=STAT(N,Z)+(INC/2.0)
          IF(POSS.GT.LINC.AND.POSS.LE.GINC) THEN DO
            ADD(Q)=STG(N-1,Z,7)
          ELSE DO
            ADD(Q)=0.0
          END IF
16        CONTINUE
15        CONTINUE
          D=6
          CALL SORT2(MADD,ADD,POSN,D)
          ENB(I)=ENB(I)+MADD
          STG(N,J,K)=ENB(I)
        END IF
      END IF
C *****************************************************************
C      ONCE THIS LOOP IS COMPLETE, THE FIRST ROW OF
C      THE STAGE MATRIX IS CALCULATED.
C      THE FOLLOWING SET OF STATEMENTS ALSO CALCULATE
C      THE BEST DECISION VALUE, BEST DECISION, AND PUT
C      THESE IN A MATRIX WITH THE CORRESPONDING STATE VAR.
C *****************************************************************
50    CONTINUE
C      PRINT 21
C      PRINT 22, (ENB(I), I=1,6)
C 21   FORMAT(' ','CURRENT BENEFIT VALUES FOR STATE(J) FOR ALL SVOL')
C 22   FORMAT(' ',6(1X,F10.2/))
      LENG=6
      CALL SORT2(MAX,ENB,DECIS,LENG)
      STG(N,J,7)=MAX
      STG(N,J,8)=DECIS
      PRINT, 'DECIS=',STG(N,J,8)
      PRINT, 'MAX=',STG(N,J,7)
40    CONTINUE
      PRINT, 'STAGE=',N
      PRINT, '     STORAGE     DECISIONS      '
      PRINT 23, (SVOL(N,I),I=1,6)
23    FORMAT(' ','STATES  ',6(F7.0,1X),' F*XN     X*N')
      DO 14 O=1,6
```

```
      PRINT 24, STAT(N,O),(STG(N,O,K),K=1,8)
24    FORMAT(' ',F8.0,8(1X,F7.0))
14    CONTINUE
C ******************************************************************
C      AT THE END OF THIS LOOP, THE J*3 MATRIX FOR 1 STAGE
C      IS COMPLETE.
39    CONTINUE
30     CONTINUE
C      AT END OF THIS LOOP THERE ARE MATRICES FOR EACH STAGE.
C *************************************************************
C *********************************************************
C         THIS IS A SUBROUTINE WHICH CALCULATES AREA FLOODED FOR
C         PLANNED STORAGE, UNPLANNED FLOODING, AND THE UNFLOODED
C         AREA. THEN IT CALCULATES NET BENEFIT FOR
C         EACH DECISION, FOR EACH CROP TYPE, FOR THE
C         STORAGE AREA, THE UNPLANNED FLOODING AREA, AND
C         THE AREA OF NO FLOODING. IT ALSO SORTS THESE TO
C         FIND THE BEST CROP TYPE FOR EACH AREA FOR EACH
C         DECISION. THESE ARE SENT TO THE MAIN PROGRAM.
C
C         IT ALSO CALLS SUB. ROUT, AND DURATION/DAMAGE
C         SUBPROGRAMS.
C ****************************************************************
      STOP
      END
C ****************************************************************
      SUBROUTINE STAG(SVOL,BIGF,BIGU,BIGN,STAT,JPB,
     *JPW,JPF,I,J,N,B,INC)
      REAL SVOL(18,6),GOODF,GOODU,GOODN,STAT(18,6),V2,VL,AREA,
     *     AC,VA,BENF1,BENF2,BENF3,BENF4,DUR,BENU1,BENU2,DURU,
     *     BENU3,BENU4,BENN1,BENN2,BENN3,BENN4,JPW(5,13),JPB(5,13),
     *     JPF(5,13),ARE(4),DUDW,DUDB,DUDF,DUDA,BENF(4),BENU(4),
     *     BENN(4),BIGF,BIGU,BIGN,VOLR,ARUF,DURN,FURA
      REAL TOAR(18),VT(18,6),AS(18,6),AI(18,6),PCV(18,3)
      INTEGER D,J,X,A,C,N,B,I,INC,E,CROP,R(18),P
      IF(N.EQ.2) THEN DO
        BF=4491.2
      END IF
      IF(N.EQ.3) THEN DO
        BF=2899.66
      END IF
      IF(N.GE.4) THEN DO
        BF=10.76
      END IF
      IF(N.EQ.4) THEN DO
      STAT(N,J)=0.0
      END IF
      IF(N.EQ.5) THEN DO
        STAT(N,J)=SVOL(N-1,J)
      END IF
      IF(N.GT.5) THEN DO
      STAT(N,1)=0.0
      STAT(N,6)=STAT(N-1,6)+SVOL(N-1,6)
      INC=STAT(N,6)/5
      STAT(N,2)=INC
```

```
      STAT(N,3)=STAT(N,2)+INC
      STAT(N,4)=STAT(N,3)+INC
      STAT(N,5)=STAT(N,4)+INC
      END IF
      IF(N.EQ.18) THEN DO
        SVOL(N,I)=0.0
      END IF
      BENF1=BENF2=BENF3=BENF4=0.0
      BENU1=BENU2=BENU3=BENU4=0.0
      BENN1=BENN2=BENN3=BENN4=0.0
      CALL ROUT(SVOL,STAT,VL,DUR1,DUR2,DUR3,DUR4,N,J,I,B,VOLR,DURX,BF)
C     PRINT 303, SVOL(N,I),STAT(N,J),I,J
C  303FORMAT(' ',10X,'SVOL= ',F10.2,3X,'STAT= ',F10.2,1X,'I= ',I2,
C    *1X,'J= ',I2)
C     PRINT 304, VOLR,DUR1,DUR2,DUR3,DUR4,DURX
C  304FORMAT(' ','VOLR= ',F10.1,1X,'DUR1= ',F10.1,1X,'DUR2= ',F10.1 /
C    *       ' ','DUR3= ',F10.2,1X,'DUR4= ',F10.1,1X,'DURX= ',F10.2)
C         PRINT, 'VL=  ',VL
      IF(I.EQ.1 .AND.J.EQ.1) THEN DO
      IF(B.EQ.1) THEN DO
      READ, TOAR(N),R(N)
      NN=R(N)
      READ, (VT(N,L),L=1,NN)
      M=NN-1
      DO 17 K=1,M
      READ, AS(N,K),AI(N,K)
17    CONTINUE
      READ, PCV(N,1),PCV(N,2),PCV(N,3)
      END IF
      END IF
      AC=0.0
      AREA=0.0
      IF(SVOL(N,I).NE.0.0) THEN DO
      V2=VL
C       PRINT, 'V2=  ',V2
      VA=V2
      DO 510 P=1,4
      DO 500 K=1,M
      L=K+1
      IF(V2.GE.VT(N,K).AND.V2.LT.VT(N,L)) THEN DO
        AREA=V2*AS(N,K)+AI(N,K)
        IF(AREA.GE.TOAR(N)) THEN DO
          AREA=TOAR(N)
        END IF
      END IF
 500  CONTINUE
      V2=V2+VA
        ARE(P)=AREA-AC
        AC=ARE(P)
C       PRINT, 'AC=  ',AC
      IF(P.EQ.1) THEN DO
        DUR=DUR1
      END IF
      IF(P.EQ.2) THEN DO
        DUR=DUR2
```

```
      END IF
      IF(P.EQ.3) THEN DO
        DUR=DUR3
      END IF
      IF(P.EQ.4) THEN DO
        DUR=DUR4
      END IF
C     PRINT 305, ARE(P)
C  305FORMAT(' ','ARE(P)= ',F10.1)
      BENF1=DUDW(DUR,JPW)*232.81*ARE(P)*PCV(N,1)+BENF1
      BENF2=DUDB(DUR,JPB)*280.93*ARE(P)*PCV(N,2)+BENF2
      BENF3=DUDF(DUR,JPF)*208.04*ARE(P)*PCV(N,3)+BENF3
      BENF4=DUDA(DUR)*ARE(P)+BENF4
C     BENF4=ARE(P)*173.84+BENF4
C     PRINT 306, BENF1,BENF2,BENF3,BENF4,AREA
C  306FORMAT(' ','BENF1= ',F10.1,1X,'BENF2= ',F10.1,1X,'BENF3= ',
C    *F10.1,1X,'BENF4= ',F10.1,1X,'AREA=',F6.2)
510   CONTINUE
      END IF
      DURU=DURX
      DO 550 K=1,M
      IF(VOLR.GE.VT(N,K) .AND.VOLR.LT.VT(N,K+1)) THEN DO
        ARUF=VOLR*AS(N,K)+AI(N,K)
      END IF
 550  CONTINUE
      FURA=TOAR(N)-AREA
      IF(ARUF.GE.FURA) THEN DO
        ARUF=FURA
      END IF
C     PRINT 313, ARUF,P,DURU,VOLR
C  313FORMAT(' ','ARUF= ',F6.2,' P= ',I2,' DURU= ',
C    *(F8.2,2X),'VOLR= ',F10.2)
      BENU1=DUDW(DURU,JPW)*232.81*ARUF*PCV(N,1)
      BENU2=DUDB(DURU,JPB)*280.93*ARUF*PCV(N,2)
      BENU3=DUDF(DURU,JPF)*208.04*ARUF*PCV(N,3)
      BENU4=DUDA(DURU)*ARUF
C     BENU4=ARUF*173.84
C     PRINT 307, BENU1,BENU2,BENU3,BENU4
C  307FORMAT(' ','BENU1= ',F10.1,1X,'BENU2= ',F10.1,1X,'BENU3= ',
C    *F10.1,1X,'BENU4= ',F10.1)
      ARNF=TOAR(N)-(ARUF+AREA)
      IF(ARNF.LE.0.0) THEN DO
        ARNF=0.0
      END IF
C     PRINT, 'ARNF=',ARNF
      DURN=0.0
      BENN1=232.81*ARNF*PCV(N,1)
      BENN2=280.93*ARNF*PCV(N,2)
      BENN3=208.04*ARNF*PCV(N,3)
      BENN4=ARNF*((0.6*173.84)+(0.4*86.92))
C     BENN4=ARNF*173.84
C     PRINT 308, BENN1,BENN2,BENN3,BENN4
C  308FORMAT(' ','BENN1= ',F10.1,1X,'BENN2= ',F10.1,1X,'BENN3= ',
C    *F10.1,1X,'BENN4= ',F10.1)
      BENF(1)=BENF1
```

```
      BENF(2)=BENF2
      BENF(3)=BENF3
      BENF(4)=BENF4
      BENU(1)=BENU1
      BENU(2)=BENU2
      BENU(3)=BENU3
      BENU(4)=BENU4
      BENN(1)=BENN1
      BENN(2)=BENN2
      BENN(3)=BENN3
      BENN(4)=BENN4
      E=4
      CALL SORT1(BIGF,BENF,CROP,E)
      PRINT, 'BIGF=',BIGF,'CROP=',CROP
      CALL SORT1(BIGU,BENU,CROP,E)
      IF(B.EQ.5) THEN DO
      PRINT, 'BIGU=',BIGU,'CROP=',CROP
      END IF
      CALL SORT1(BIGN,BENN,CROP,E)
      IF(B.EQ.5) THEN DO
      PRINT, 'BIGN=',BIGN,'CROP=',CROP
      END IF
      RETURN
      END
C ################################################################
C THIS SUBROUTINE ROUTES EACH HYDRO TO THE END OF THE
C NEXT STAGE. INPUT TO THIS CAN BE ONE OF 3 POSSIBLE SHAPES
C DEPENDING ON THE CURRENT STATE.
C ###########################################################
      SUBROUTINE ROUT(SVOL,STAT,VL,DUR1,DUR2,DUR3,DUR4,N,J,I,B,
     *VOLR,DURX,BF)
      INTEGER B,Z,N,J,I,F,D,E
      REAL SA,SR,VOLH,SVOL(18,6),VOLR,LOV,VL,VLA,VL2,
     *     VOL,DUR1,DUR2,DUR3,DUR4,G,H,STAT(18,6),V(18,6,6,5,6),
     *     W(18,6,6,5,2),DURX,BF,LO,RLOV(18,6,6,5)
      REAL*8 X(5,50),Y(5,50)
C ################################################################
C VARIABLES: INTEGER: B= LOOPING COUNTER TO READ HYDROGRAPH COORDS
C                        B REPRESENTS A HYDROGRAPH
C                     Z= THE SUBSCRIPT FOR THE ACTUAL HYDRO COORDS
C                     SVOL= STORAGE DECISION.
C                        N= STAGE COUNTER
C                        J= STATE COUNTER
C                        I= SVOL COUNTER
C             REAL: X(B,1)= X COORD OF PEAK OF HYDRO.
C                   X(B,2)= X COORD OF END POINT OF HYDRO
C                   X(B,3)= POINT AT WHICH BANKFULL IS INITIALLY
C                           REACHED
C                   X(B,4)= X COORD OF POINT AT WHICH BANKFULL IS
C                           RETURNED TO AFTER PEAK
C                   X(B,5)= END OF EXTENSION OF HYDRO, BANKFULL
C                   X(B,6)= END OF EXTENSION OF HYDRO, ABSOLUTE
C                  X(B,12)= END POINT OF SVOL
C                           THIS MAY BE NEW "PEAK"
C                  X(B,13),X(B,14),X(B,15)= DIVISIONS OF SVOL
```

```
C                 X(B,16),X(B,17),X(B,18)= DIVISIONS OF EXTN.
C                 X(B,?)= COORDS USED IN ITERATION TO FIND ALL
C                          ABOVE NOTED POINTS
C                  Y(B,1)= PEAK
C                  Y(B,2)= BANKFULL
C                  Y(B,3)= START, FINISH
C                 Y(B,12)= END POINT OF SVOL
C                          MAY BE NEW "PEAK"
C                 Y(B,13),Y(B,14),Y(B,15)= SAME AS FOR X
C                 Y(B,16),Y(B,17),Y(B,18)= SAME AS FOR X
C                 Y(B,?)= SAME AS FOR X
C                  SA=SLOPE OF RISING LIMB FOR B HYDRO
C                  SR=SLOPE OF RECESSION LIMB FOR SAME
C                  VOLH= INITIAL HYDRO VOLUME
C                  SVOL= CHOSEN STORAGE VOLUME
C                  VOLR= VOLUME REMAINING AFTER SVOL IS REMOVED
C                  LOV = A VOLUME USED FOR ITERATION
C                  VL  = .25*SVOL USED FOR ITERATION
C                  VL2 = .50*SVOL USED FOR ITERATION
C                  VLA = A VOLUME USED FOR ITERATION
C                  VOL = A VOLUME USED FOR ITERATION
C                  DUR1= DURATION OF FLOODING OF FIRST AREA
C                  DUR2,DUR3,DUR4= SAME AS FOR DUR1
C                  G   = VALUE USED FOR ITERATION IN EXTN
C                  H   = SAME
C                  STAT= STATE
C                  V(J,I,B,Z)= ARRAY TO STORE VALUES OF X(B,Z)
C                  W(J,I,B,Z)= ARRAY TO STORE VALUES OF Y(B,Z)
C ###############################################################
C THIS FIRST PART CALCULATES THE STORAGE VOLUME OF THE
C HYDROGRAPH, IF THE CURRENT STATE IS 0, AND THE STORAGE
C VOLUME IS NOT 0.  FOLLOWING SECTIONS WILL CALCULATE
C HYDROGRAPHS FROM DIFFERENT INPUT HYDRO'S.
C #################################################################
      IF(N.EQ.4) THEN DO
      IF(I.EQ.1.AND.J.EQ.1) THEN DO
       IF(B.EQ.1) THEN DO
        READ 91, ((X(L,Z), Y(L,Z), Z=1,3),L=1,5)
       END IF
C       PRINT 92
C       PRINT 93, (X(B,Z), Y(B,Z), Z=1,3)
91      FORMAT(6F7.2)
C 92    FORMAT(' ', 25X, 'HYDROGRAPH COORDINATES')
C 93    FORMAT(' ', 6(1X,F4.1))
        Y(B,12)=0.0
        LO=0.0
        D=I
        F=J
        E=N
        GO TO 21
      END IF
      END IF
C     PRINT 94, N,J,I,B
C 94  FORMAT(' ',10X,'N= ',I2,'J=  ',I2,'I=  ',I2,'B=  ',I2)
      IF(N.EQ.4) THEN DO
```

```
          E=N
          IF(J.EQ.1) THEN DO
            D=1
            F=J
            VOLR=0.0
          ELSE DO
            F=1
            D=J
          END IF
        ELSE DO
          E=N-1
          IF(J.EQ.1) THEN DO
            D=1
            F=J
            VOLR=0.0
          ELSE DO
            F=1
            D=J
          END IF
        END IF
        X(B,1)=DBLE(V(E,F,D,B,1))
        X(B,3)=DBLE(V(E,F,D,B,2))
        X(B,4)=DBLE(V(E,F,D,B,3))
        X(B,5)=DBLE(V(E,F,D,B,4))
        X(B,6)=DBLE(V(E,F,D,B,5))
        X(B,12)=DBLE(V(E,F,D,B,6))
        Y(B,1)=DBLE(W(E,F,D,B,1))
        Y(B,2)=BF
        Y(B,12)=DBLE(W(E,F,D,B,2))
        VOLR=RLOV(E,F,D,B)
        D=I
        F=J
        E=N
        IF(SVOL(N,I).EQ.VOLH) THEN DO
          SVOL(N,I)=LO
C         PRINT, 'SVOL=  ',SVOL(N,I)
        END IF
        GO TO 22
21      CONTINUE
        Y(B,2)=BF
        SA=(Y(B,1)-Y(B,3))/X(B,1)
        X(B,3)=(Y(B,2)-Y(B,3))/SA
        SR=(Y(B,3)-Y(B,1))/(X(B,2)-X(B,1))
        X(B,4)=(Y(B,2)-Y(B,1))/SR+X(B,1)
        X(B,5)=X(B,4)
        X(B,6)=X(B,2)
        X(B,12)=0.0
        VL=0.0
        DURX=((X(B,4)-X(B,1))/2.0+X(B,1))-((X(B,1)-X(B,3))/2.0+X(B,3))
        VOLH=(X(B,1)-X(B,3))*(Y(B,1)-Y(B,2))/2.0+(X(B,4)-X(B,1))*
     *      (Y(B,1)-Y(B,2))/2.0
        VOLH=VOLH*3600.0
        PRINT, 'VOLH=',VOLH
        VOLR=VOLH
22      CONTINUE
```

```
      SA=(Y(B,1)-Y(B,3))/X(B,1)
      X(B,3)=(Y(B,2)-Y(B,3))/SA
      SR=(Y(B,3)-Y(B,1))/(X(B,2)-X(B,1))
      X(B,4)=(Y(B,2)-Y(B,1))/SR+X(B,1)
      IF(J.EQ.1) THEN DO
        VL=0.0
        DURX=((X(B,4)-X(B,1))/2.0+X(B,1))-((X(B,1)-X(B,3))/2.0+X(B,3))
      END IF
      IF(Y(B,12).EQ.Y(B,2)) THEN DO
        VOLR=0.0
        VL=0.0
      END IF
      IF(Y(B,12).NE.Y(B,2)) THEN DO
        IF(SVOL(N,I).EQ.0.0) THEN DO
          IF(J.GT.1) THEN DO
            DURX=((X(B,4)-X(B,12))/2.0+X(B,12))-X(B,12)
          END IF
        END IF
      IF(SVOL(N,I).NE.0.0) THEN DO
        IF(STAT(N,J).EQ.0.0) THEN DO
          VOLH=(X(B,1)-X(B,3))*(Y(B,1)-Y(B,2))/2.0+(X(B,4)-X(B,1))*
     *    (Y(B,1)-Y(B,2))/2.0
          VOLH=VOLH*3600.0
          IF(SVOL(N,I).GE.VOLH) THEN DO
            Y(B,12)=Y(B,2)
            X(B,12)=X(B,4)
            VOLR=0.0
            LO=SVOL(N,I)
            SVOL(N,I)=VOLH
            GO TO 112
          END IF
          VOLR=VOLH-SVOL(N,I)
          VOL=((X(B,1)-X(B,3))*(Y(B,1)-Y(B,2)))/2.0
          VOL=VOL*3600.0
          IF(IFIX(.5+VOL).EQ.IFIX(.5+SVOL(N,I))) THEN DO
C ################################################################
C ONCE SVOL IS FOUND ON THE HYDROGRAPH, EXECUTION GOES TO
C THE PART WHICH CALCULATES FOUR DIVISIONS WITHIN THE SVOL
C SECTION. THIS IS FOR CALCULATION OF DURATION AND CORRESPONDING
C FLOODED AREA. 110 IS FOR SVOL STORING UP TO THE PEAK, EXACTLY.
C ##################################################################
          Y(B,12)=Y(B,1)
          X(B,12)=X(B,1)
          X(B,20)=X(B,3)
            GO TO 110
          END IF
          IF(IFIX(.5+VOL).GT.IFIX(.5+SVOL(N,I))) THEN DO
            Y(B,12)=(Y(B,1)-Y(B,2))/2.0+Y(B,2)
            X(B,12)=(Y(B,12)-Y(B,2))/SA+X(B,3)
            VOL=((X(B,12)-X(B,3))*(Y(B,12)-Y(B,2)))/2.0
            VOL=VOL*3600.0
            IF(IFIX(.5+VOL).EQ.IFIX(.5+SVOL(N,I))) THEN DO
              X(B,20)=X(B,3)
              GO TO 110
            END IF
```

```
            IF(IFIX(.5+VOL).GT.IFIX(.5+SVOL(N,I))) THEN DO
              Y(B,22)=Y(B,2)
              Y(B,32)=Y(B,12)
              Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
              X(B,12)=(Y(B,12)-Y(B,2))/SA+X(B,3)
100           CONTINUE
              VOL=((X(B,12)-X(B,3))*(Y(B,12)-Y(B,2)))/2.0
              VOL=VOL*3600.0
              IF(IFIX(.5+VOL).GT.IFIX(.5+SVOL(N,I))) THEN DO
                Y(B,32)=Y(B,12)
                Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
                X(B,12)=(Y(B,12)-Y(B,2))/SA+X(B,3)
                GO TO 100
              END IF
              IF(IFIX(.5+VOL).LT.IFIX(.5+SVOL(N,I))) THEN DO
                Y(B,22)=Y(B,12)
                Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
                X(B,12)=(Y(B,12)-Y(B,2))/SA+X(B,3)
                GO TO 100
              END IF
              GO TO 110
            END IF
            IF(IFIX(.5+VOL).LT.IFIX(.5+SVOL(N,I))) THEN DO
              Y(B,22)=Y(B,12)
              Y(B,32)=Y(B,1)
              Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
              X(B,12)=(Y(B,12)-Y(B,2))/SA+X(B,3)
101           CONTINUE
              VOL=((X(B,12)-X(B,3))*(Y(B,12)-Y(B,2)))/2.0
              VOL=VOL*3600.0
              IF(IFIX(.5+VOL).GT.IFIX(.5+SVOL(N,I))) THEN DO
                Y(B,32)=Y(B,12)
                Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
                X(B,12)=(Y(B,12)-Y(B,2))/SA+X(B,3)
                GO TO 101
              END IF
              IF(IFIX(.5+VOL).LT.IFIX(.5+SVOL(N,I))) THEN DO
                Y(B,22)=Y(B,12)
                Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
                X(B,12)=(Y(B,12)-Y(B,2))/SA+X(B,3)
                GO TO 101
              END IF
              GO TO 110
            END IF
          END IF
          IF(IFIX(.5+VOL).LT.IFIX(.5+SVOL(N,I))) THEN DO
            Y(B,12)=(Y(B,1)-Y(B,2))/2.0+Y(B,2)
            X(B,12)=(Y(B,12)-Y(B,2))/SR+X(B,4)
            LOV=((X(B,4)-X(B,12))*(Y(B,12)-Y(B,2)))/2
            LOV=LOV*3600.0
            VOL=VOLH-LOV
            IF(IFIX(.5+VOL).EQ.IFIX(.5+SVOL(N,I))) THEN DO
              X(B,20)=X(B,3)
C ####################################################################
C IF SVOL IS LARGE ENOUGH TO STORE PAST THE PEAK, IT GOES TO
```

```
C 112 FOR DIVISIONAL CALCULATION.
C ###################################################################
               GO TO 112
             END IF
             IF(IFIX(.5+VOL).GT.IFIX(.5+SVOL(N,I))) THEN DO
               Y(B,22)=Y(B,12)
               Y(B,32)=Y(B,1)
               Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
               X(B,12)=(Y(B,12)-Y(B,2))/SR+X(B,4)
102            CONTINUE
               LOV=((X(B,4)-X(B,12))*(Y(B,12)-Y(B,2)))/2
               LOV=LOV*3600.0
               VOL=VOLH-LOV
               IF(IFIX(.5+VOL).GT.IFIX(.5+SVOL(N,I))) THEN DO
                 Y(B,22)=Y(B,12)
                 Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
                 X(B,12)=(Y(B,12)-Y(B,2))/SR+X(B,4)
                 GO TO 102
               END IF
               IF(IFIX(.5+VOL).LT.IFIX(.5+SVOL(N,I))) THEN DO
                 Y(B,32)=Y(B,12)
                 Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
                 X(B,12)=(Y(B,12)-Y(B,2))/SR+X(B,4)
                 GO TO 102
               END IF
             ELSE DO
               Y(B,32)=Y(B,12)
               Y(B,22)=Y(B,2)
               Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
               X(B,12)=(Y(B,12)-Y(B,2))/SR+X(B,4)
103            CONTINUE
               LOV=((X(B,4)-X(B,12))*(Y(B,12)-Y(B,2)))/2
               LOV=LOV*3600.0
               VOL=VOLH-LOV
               IF(IFIX(.5+VOL).GT.IFIX(.5+SVOL(N,I))) THEN DO
                 Y(B,22)=Y(B,12)
                 Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
                 X(B,12)=(Y(B,12)-Y(B,2))/SR+X(B,4)
                 GO TO 103
               END IF
               IF(IFIX(.5+VOL).LT.IFIX(.5+SVOL(N,I))) THEN DO
                 Y(B,32)=Y(B,12)
                 Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
                 X(B,12)=(Y(B,12)-Y(B,2))/SR+X(B,4)
                 GO TO 103
               END IF
             END IF
           END IF
           DURX=((X(B,4)-X(B,12))/2.0+X(B,12))-X(B,12)
           GO TO 112
C ###################################################################
C THIS PART CALCULATES X(B,13,14,15) AND Y(B,13,14,15), WHICH
C ARE THE COORDINATES OF THE DIVIDING LINES BETWEEN THE TIME
C SEGMENTS. 110 IS FOR SVOL .LE. PEAK, AND CALCULATED
C THE SAME WAY, AND 112 IS FOR SVOL.GT.PEAK.
```

```
C ####################################################################
110       CONTINUE
          Y(B,22)=Y(B,12)
          X(B,22)=X(B,12)
          VL=SVOL(N,I)/4.0
          Y(B,14)=(Y(B,12)-Y(B,2))/2.0+Y(B,2)
          X(B,14)=(Y(B,14)-Y(B,2))/SA+X(B,3)
          VLA=((X(B,14)-X(B,3))*(Y(B,14)-Y(B,2)))/2.0*3600.0
          VL2=VL*2.0
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
            Y(B,34)=Y(B,14)
            Y(B,24)=Y(B,2)
            Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
            X(B,14)=(Y(B,14)-Y(B,2))/SA+X(B,3)
113         CONTINUE
            VLA=((X(B,14)-X(B,3))*(Y(B,14)-Y(B,2)))/2.0*3600.0
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
              Y(B,34)=Y(B,14)
              Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
              X(B,14)=(Y(B,14)-Y(B,2))/SA+X(B,3)
              GO TO 113
            END IF
            IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
              Y(B,24)=Y(B,14)
              Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
              X(B,14)=(Y(B,14)-Y(B,2))/SA+X(B,3)
              GO TO 113
            END IF
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
            Y(B,24)=Y(B,14)
            Y(B,34)=Y(B,12)
            Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
            X(B,14)=(Y(B,14)-Y(B,2))/SA+X(B,3)
114         CONTINUE
            VLA=((X(B,14)-X(B,3))*(Y(B,14)-Y(B,2)))/2.0*3600.0
            IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
              Y(B,24)=Y(B,14)
              Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
              X(B,14)=(Y(B,14)-Y(B,2))/SA+X(B,3)
              GO TO 114
            END IF
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
              Y(B,34)=Y(B,14)
              Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
              X(B,14)=(Y(B,14)-Y(B,2))/SA+X(B,3)
              GO TO 114
            END IF
          END IF
C    ################### KNOW X(B,14), Y(B,14) ####################
C ################### IF SVOL.LE.PEAK    ##########################
          Y(B,13)=(Y(B,14)-Y(B,2))/2.0+Y(B,2)
          X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
          VLA=((X(B,13)-X(B,3))*(Y(B,13)-Y(B,2)))/2.0*3600.0
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
```

```
            Y(B,33)=Y(B,13)
            Y(B,23)=Y(B,2)
            Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
            X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
115         CONTINUE
            VLA=((X(B,13)-X(B,3))*(Y(B,13)-Y(B,2)))/2.0*3600.0
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
              Y(B,33)=Y(B,13)
              Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
              X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
              GO TO 115
            END IF
            IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
              Y(B,23)=Y(B,13)
              Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
              X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
              GO TO 115
            END IF
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
            Y(B,23)=Y(B,13)
            Y(B,33)=Y(B,14)
            Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
            X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
116         CONTINUE
            VLA=((X(B,13)-X(B,3))*(Y(B,13)-Y(B,2)))/2.0*3600.0
            IF(IFIX(.5+VLA).GT.IFIX(VL)) THEN DO
              Y(B,33)=Y(B,13)
              Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
              X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
              GO TO 116
            END IF
            IF(IFIX(.5+VLA).LT.IFIX(VL)) THEN DO
              Y(B,23)=Y(B,13)
              Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
              X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
              GO TO 116
            END IF
          END IF
C ************* KNOW X(B,13), Y(B,13) **********************
C ************* IF SVOL.LE.PEAK      *************************
          Y(B,15)=(Y(B,12)-Y(B,14))/2.0+Y(B,14)
          X(B,15)=(Y(B,15)-Y(B,14))/SA+X(B,14)
          VLA=((X(B,15)-X(B,14))*(Y(B,14)-Y(B,15)))/2.0+(X(B,15)-
     *        X(B,14))*(Y(B,14)-Y(B,2))
          VLA=VLA*3600.0
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
            Y(B,25)=Y(B,15)
            Y(B,35)=Y(B,12)
            Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
            X(B,15)=(Y(B,15)-Y(B,2))/SA+X(B,3)
117         CONTINUE
            VLA=((X(B,15)-X(B,14))*(Y(B,14)-Y(B,15)))/2.0+(X(B,15)-
     *      X(B,14))*(Y(B,14)-Y(B,2))
            VLA=VLA*3600.0
```

```
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
              Y(B,35)=Y(B,15)
              Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
              X(B,15)=(Y(B,15)-Y(B,2))/SA+X(B,3)
              GO TO 117
            END IF
            IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
              Y(B,25)=Y(B,15)
              Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
              X(B,15)=(Y(B,15)-Y(B,2))/SA+X(B,3)
              GO TO 117
            END IF
          END IF
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
            Y(B,35)=Y(B,15)
            Y(B,25)=Y(B,14)
            Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
            X(B,15)=(Y(B,15)-Y(B,2))/SA+X(B,3)
145         CONTINUE
            VLA=((X(B,15)-X(B,14))*(Y(B,14)-Y(B,15)))/2.0+(X(B,15)-
     *      X(B,14))*(Y(B,14)-Y(B,2))
            VLA=VLA*3600.0
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
              Y(B,35)=Y(B,15)
              Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
              X(B,15)=(Y(B,15)-Y(B,2))/SA+X(B,3)
              GO TO 145
            END IF
            IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
              Y(B,25)=Y(B,15)
              Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
              X(B,15)=(Y(B,15)-Y(B,2))/SA+X(B,3)
              GO TO 145
            END IF
          END IF
C ############# KNOW ALL TIME DIV COORDS IF SVOL<PEAK#########
          GO TO 200
C ############# 200 TAKES TO THE PART WHERE THE EXTENTION IS###
C ############# DIVIDED INTO 4 SEGMENTS                    ####
112       CONTINUE
C ######### 112 IS FOR CALCULATING THE 4 DIVISIONS###############
C ######### IF SVOL.GT.PEAK                      ###############
          X(B,20)=X(B,3)
          Y(B,20)=Y(B,2)
          VL=SVOL(N,I)/4.0
          VL2=VL*2.0
          Y(B,14)=Y(B,1)
          X(B,14)=X(B,1)
          VLA=((X(B,14)-X(B,3))*(Y(B,14)-Y(B,2)))/2.0*3600.0
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
            Y(B,34)=Y(B,14)
            Y(B,14)=(Y(B,14)-Y(B,2))/2.0+Y(B,2)
            X(B,14)=(Y(B,14)-Y(B,2))/SA+X(B,3)
            VLA=((X(B,14)-X(B,3))*(Y(B,14)-Y(B,2)))/2.0*3600.0
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
```

```
              Y(B,34)=Y(B,14)
              Y(B,24)=Y(B,2)
              Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
              X(B,14)=(Y(B,14)-Y(B,2))/SA+X(B,3)
118           CONTINUE
              VLA=((X(B,14)-X(B,3))*(Y(B,14)-Y(B,2)))/2.0*3600.0
              IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
                Y(B,34)=Y(B,14)
                Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
                X(B,14)=(Y(B,14)-Y(B,2))/SA+X(B,3)
                GO TO 118
              END IF
              IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
                Y(B,24)=Y(B,14)
                Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
                X(B,14)=(Y(B,14)-Y(B,2))/SA+X(B,3)
                GO TO 118
              END IF
            END IF
            IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
              Y(B,24)=Y(B,14)
              Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
              X(B,14)=(Y(B,14)-Y(B,2))/SA+X(B,3)
119           CONTINUE
              VLA=((X(B,14)-X(B,3))*(Y(B,14)-Y(B,2)))/2.0*3600.0
              IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
                Y(B,34)=Y(B,14)
                Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
                X(B,14)=(Y(B,14)-Y(B,2))/SA+X(B,3)
                GO TO 119
              END IF
              IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
                Y(B,24)=Y(B,14)
                Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
                X(B,14)=(Y(B,14)-Y(B,2))/SA+X(B,3)
                GO TO 119
              END IF
            END IF
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
            Y(B,34)=Y(B,14)
            Y(B,24)=Y(B,12)
            VLB=VLA
            Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
            X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
            VLA=(((X(B,14)-X(B,1))*(Y(B,1)-Y(B,14)))/2.0+(X(B,14)-
     *      X(B,1))*(Y(B,14)-Y(B,2)))*3600.0+VLB
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
              Y(B,24)=Y(B,14)
              Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
              X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
120           CONTINUE
              VLA=(((X(B,14)-X(B,1))*(Y(B,1)-Y(B,14)))/2.0+(X(B,14)-
     *            X(B,1))*(Y(B,14)-Y(B,2)))*3600.0+VLB
              IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
```

```
              Y(B,34)=Y(B,14)
              Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
              X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
              GO TO 120
            END IF
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
              Y(B,24)=Y(B,14)
              Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
              X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
              GO TO 120
            END IF
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
            Y(B,34)=Y(B,14)
            Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
            X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
121         CONTINUE
            VLA=(((X(B,14)-X(B,1))*(Y(B,1)-Y(B,14)))/2.0+(X(B,14)-
     *          X(B,1))*(Y(B,14)-Y(B,2)))*3600.0+VLB
            IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
              Y(B,34)=Y(B,14)
              Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
              X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
              GO TO 121
            END IF
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
              Y(B,24)=Y(B,14)
              Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
              X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
              GO TO 121
            END IF
          END IF
        END IF
C      ############## KNOW X(B,14), Y(B,14) ######################
C      ##############################################################
C      FROM HERE X,Y(B,13) IS CALCULATED, FIRST FOR X(B,14)<X(B,1)
C      THEN FOR X(B,14)>X(B,1).
C      ##############################################################
        IF(X(B,14).LE.X(B,1)) THEN DO
          Y(B,13)=(Y(B,14)-Y(B,2))/2.0+Y(B,2)
          X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
          VLA=(X(B,13)-X(B,3))*(Y(B,13)-Y(B,2))/2.0*3600.0
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
            Y(B,33)=Y(B,13)
            Y(B,23)=Y(B,2)
            Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
            X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
122         CONTINUE
            VLA=(X(B,13)-X(B,3))*(Y(B,13)-Y(B,2))/2.0*3600.0
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
              Y(B,33)=Y(B,13)
              Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
              X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
              GO TO 122
            END IF
```

```
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
            Y(B,23)=Y(B,13)
            Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
            X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
            GO TO 122
          END IF
        END IF
        IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
          Y(B,23)=Y(B,13)
          Y(B,33)=Y(B,14)
          Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
          X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
123       CONTINUE
          VLA=(X(B,13)-X(B,3))*(Y(B,13)-Y(B,2))/2.0*3600.0
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
            Y(B,33)=Y(B,13)
            Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
            X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
            GO TO 123
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
            Y(B,23)=Y(B,13)
            Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
            X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
            GO TO 123
          END IF
        END IF
      END IF
C ########### Y(B,13), X(B,13) KNOWN IF X(B,14).LT. PEAK #######
      IF(X(B,14).GT.X(B,1)) THEN DO
        Y(B,13)=Y(B,1)
        X(B,13)=X(B,1)
        VLA=(X(B,13)-X(B,3))*(Y(B,13)-Y(B,2))/2.0*3600.0
        IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
          Y(B,33)=Y(B,13)
          Y(B,13)=(Y(B,13)-Y(B,2))/2.0+Y(B,2)
          X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
          VLA=(X(B,13)-X(B,3))*(Y(B,13)-Y(B,2))/2.0*3600.0
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
            Y(B,33)=Y(B,13)
            Y(B,23)=Y(B,2)
            Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
            X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
124         CONTINUE
            VLA=(X(B,13)-X(B,3))*(Y(B,13)-Y(B,2))/2.0*3600.0
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
              Y(B,33)=Y(B,13)
              Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
              X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
              GO TO 124
            END IF
            IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
              Y(B,23)=Y(B,13)
              Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
              X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
```

```
          GO TO 124
        END IF
      END IF
      IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
        Y(B,23)=Y(B,13)
        Y(B,33)=Y(B,1)
        Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
        X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
125     CONTINUE
        VLA=(X(B,13)-X(B,3))*(Y(B,13)-Y(B,2))/2.0*3600.0
        IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
          Y(B,33)=Y(B,13)
          Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
          X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
          GO TO 125
        END IF
        IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
          Y(B,23)=Y(B,13)
          Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
          X(B,13)=(Y(B,13)-Y(B,2))/SA+X(B,3)
          GO TO 125
        END IF
      END IF
    END IF
    IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
      Y(B,33)=Y(B,13)
      X(B,33)=X(B,13)
      Y(B,23)=Y(B,14)
      X(B,23)=X(B,14)
      Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
      X(B,13)=(Y(B,13)-Y(B,14))/SR+X(B,14)
      VLA=(X(B,14)-X(B,13))*(Y(B,13)-Y(B,14))/2.0+(X(B,14)-
*     X(B,13))*(Y(B,14)-Y(B,2))
      VLA=VLA*3600.0
      IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
        Y(B,33)=Y(B,13)
        Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
        X(B,13)=(Y(B,13)-Y(B,14))/SR+X(B,14)
126     CONTINUE
        VLA=(X(B,14)-X(B,13))*(Y(B,13)-Y(B,14))/2.0+(X(B,14)-
*      X(B,13))*(Y(B,14)-Y(B,2))
        VLA=VLA*3600.0
        IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
          Y(B,33)=Y(B,13)
          Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
          X(B,13)=(Y(B,13)-Y(B,14))/SR+X(B,14)
          GO TO 126
        END IF
        IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
          Y(B,23)=Y(B,13)
          Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
          X(B,13)=(Y(B,13)-Y(B,14))/SR+X(B,14)
          GO TO 126
        END IF
      END IF
```

```
              IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
                Y(B,23)=Y(B,13)
                Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
                X(B,13)=(Y(B,13)-Y(B,14))/SR+X(B,14)
127             CONTINUE
                VLA=(X(B,14)-X(B,13))*(Y(B,13)-Y(B,14))/2.0+(X(B,14)-
     *         X(B,13))*(Y(B,14)-Y(B,2))
                VLA=VLA*3600.0
                IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
                  Y(B,33)=Y(B,13)
                  Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
                  X(B,13)=(Y(B,13)-Y(B,14))/SR+X(B,14)
                  GO TO 127
                END IF
                IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
                  Y(B,23)=Y(B,13)
                  Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
                  X(B,13)=(Y(B,13)-Y(B,14))/SR+X(B,14)
                  GO TO 127
                END IF
              END IF
            END IF
          END IF
C ############# KNOW Y(B,13), X(B,13) #########################
C ############# THIS CALCULATES X,Y(B,15), FIRST FOR X(B,14)>X(B,1)
C ######### THEN FOR X(B,14)<X(B,1) #######################
          IF(X(B,14).GE.X(B,1)) THEN DO
            Y(B,15)=(Y(B,14)-Y(B,2))/2.0+Y(B,2)
            X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
            VLA=(X(B,4)-X(B,15))*(Y(B,15)-Y(B,2))/2.0*3600.0
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
              Y(B,35)=Y(B,15)
              Y(B,25)=Y(B,2)
              Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
              X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
128           CONTINUE
              VLA=(X(B,4)-X(B,15))*(Y(B,15)-Y(B,2))/2.0*3600.0
              IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
                Y(B,35)=Y(B,15)
                Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
                X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
                GO TO 128
              END IF
              IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
                Y(B,25)=Y(B,15)
                Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
                X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
                GO TO 128
              END IF
            END IF
            IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
              Y(B,25)=Y(B,15)
              Y(B,35)=Y(B,14)
              Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
              X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
```

```
129             CONTINUE
                VLA=(X(B,4)-X(B,15))*(Y(B,15)-Y(B,2))/2.0*3600.0
                IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
                  Y(B,35)=Y(B,15)
                  Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
                  X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
                  GO TO 129
                END IF
                IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
                  Y(B,25)=Y(B,15)
                  Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
                  X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
                  GO TO 129
                END IF
              END IF
            END IF
C ########### Y(B,15), X(B,15) KNOWN IF X(B,14).GT. PEAK #######
            IF(X(B,14).LT.X(B,1)) THEN DO
              Y(B,15)=Y(B,1)
              X(B,15)=X(B,1)
              VLA=(X(B,4)-X(B,15))*(Y(B,15)-Y(B,2))/2.0*3600.0
              IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
                Y(B,35)=Y(B,15)
                Y(B,15)=(Y(B,15)-Y(B,2))/2.0+Y(B,2)
                X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
                VLA=(X(B,4)-X(B,15))*(Y(B,15)-Y(B,2))/2.0*3600.0
                IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
                  Y(B,35)=Y(B,15)
                  Y(B,25)=Y(B,2)
                  Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
                  X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
130               CONTINUE
                  VLA=(X(B,4)-X(B,15))*(Y(B,15)-Y(B,2))/2.0*3600.0
                  IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
                    Y(B,35)=Y(B,15)
                    Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
                    X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
                    GO TO 130
                  END IF
                  IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
                    Y(B,25)=Y(B,15)
                    Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
                    X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
                    GO TO 130
                  END IF
                END IF
                IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
                  Y(B,25)=Y(B,15)
                  Y(B,35)=Y(B,1)
                  Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
                  X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
131               CONTINUE
                  VLA=(X(B,4)-X(B,15))*(Y(B,15)-Y(B,2))/2.0*3600.0
                  IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
                    Y(B,35)=Y(B,15)
```

```
              Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
              X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
              GO TO 131
            END IF
            IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
              Y(B,25)=Y(B,15)
              Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
              X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
              GO TO 131
            END IF
          END IF
        END IF
        IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
          Y(B,35)=Y(B,15)
          X(B,35)=X(B,15)
          Y(B,25)=Y(B,14)
          X(B,25)=X(B,14)
          Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
          X(B,15)=(Y(B,15)-Y(B,14))/SA+X(B,14)
          VLA=(X(B,15)-X(B,14))*(Y(B,15)-Y(B,14))/2.0+(X(B,15)-
     *    X(B,14))*(Y(B,14)-Y(B,2))
          VLA=VLA*3600.0
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
            Y(B,35)=Y(B,15)
            Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
            X(B,15)=(Y(B,15)-Y(B,14))/SA+X(B,14)
132         CONTINUE
            VLA=(X(B,15)-X(B,14))*(Y(B,15)-Y(B,14))/2.0+(X(B,15)-
     *      X(B,14))*(Y(B,14)-Y(B,2))
          VLA=VLA*3600.0
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
              Y(B,35)=Y(B,15)
              Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
              X(B,15)=(Y(B,15)-Y(B,14))/SA+X(B,14)
              GO TO 132
            END IF
            IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
              Y(B,25)=Y(B,15)
              Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
              X(B,15)=(Y(B,15)-Y(B,14))/SA+X(B,14)
              GO TO 132
            END IF
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
            Y(B,25)=Y(B,15)
            Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
            X(B,15)=(Y(B,15)-Y(B,14))/SA+X(B,14)
133         CONTINUE
            VLA=(X(B,15)-X(B,14))*(Y(B,15)-Y(B,14))/2.0+(X(B,15)-
     *      X(B,14))*(Y(B,14)-Y(B,2))
          VLA=VLA*3600.0
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
              Y(B,35)=Y(B,15)
              Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
              X(B,15)=(Y(B,15)-Y(B,14))/SA+X(B,14)
```

```
              GO TO 133
            END IF
            IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
              Y(B,25)=Y(B,15)
              Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
              X(B,15)=(Y(B,15)-Y(B,14))/SA+X(B,14)
              GO TO 133
            END IF
          END IF
        END IF
      END IF
      GO TO 200
C ######### ALL COORDINATES ARE KNOWN FOR DIVISIONS #############
C ######### FOR THE MAIN HYDRO PART. THE EXTENSION IS LATER #####
C ######### IE. X,Y(13,14,15,12)                    #############
C ######### NEXT PART IS FOR STAT.NE.0  ########################
    ELSE DO
      VOLH=VOLR
      IF(VOLH.LE.SVOL(N,I)) THEN DO
        VOLR=0.0
        Y(B,12)=Y(B,2)
        X(B,12)=X(B,4)
        LO=SVOL(N,I)
        SVOL(N,I)=VOLH
        GO TO 160
      END IF
      VOLR=VOLH-SVOL(N,I)
      IF(X(B,12).NE.X(B,1)) THEN DO
        VOL=(X(B,1)-X(B,12))*(Y(B,1)-Y(B,12))/2.0+(X(B,1)-X(B,12))*
     *      (Y(B,12)-Y(B,2))
        VOL=VOL*3600.0
        VLB=VOL
        Y(B,20)=Y(B,12)
        X(B,20)=X(B,12)
        IF(IFIX(.5+VOL).EQ.IFIX(.5+SVOL(N,I))) THEN DO
          Y(B,12)=Y(B,1)
          X(B,12)=X(B,1)
          GO TO 160
C ######### 160 IS FOR SEGMENT DIVISION FOR STAT.NE.0 #########
C ######### AND Y(B,12)=Y(B,1)                       ###########
        END IF
        IF(IFIX(.5+VOL).GT.IFIX(.5+SVOL(N,I))) THEN DO
          Y(B,22)=Y(B,12)
          X(B,22)=X(B,12)
          Y(B,12)=(Y(B,1)-Y(B,12))/2.0+Y(B,12)
          X(B,12)=(Y(B,12)-Y(B,1))/SA+X(B,1)
          VOL=(X(B,12)-X(B,20))*(Y(B,12)-Y(B,20))/2.0+(X(B,12)-
     *          X(B,20))*(Y(B,20)-Y(B,2))
          VOL=VOL*3600.0
          IF(IFIX(.5+VOL).GT.IFIX(.5+SVOL(N,I))) THEN DO
            Y(B,32)=Y(B,12)
            Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
            X(B,12)=(Y(B,12)-Y(B,1))/SA+X(B,1)
140         CONTINUE
            VOL=(X(B,12)-X(B,20))*(Y(B,12)-Y(B,20))/2.0+(X(B,12)-
```

```
     *             X(B,20))*(Y(B,20)-Y(B,2))
                 VOL=VOL*3600.0
                 IF(IFIX(.5+VOL).GT.IFIX(.5+SVOL(N,I))) THEN DO
                   Y(B,32)=Y(B,12)
                   Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
                   X(B,12)=(Y(B,12)-Y(B,1))/SA+X(B,1)
                   GO TO 140
                 END IF
                 IF(IFIX(.5+VOL).LT.IFIX(.5+SVOL(N,I))) THEN DO
                   Y(B,22)=Y(B,12)
                   Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
                   X(B,12)=(Y(B,12)-Y(B,1))/SA+X(B,1)
                   GO TO 140
                 END IF
               END IF
               IF(IFIX(.5+VOL).LT.IFIX(.5+SVOL(N,I))) THEN DO
                 Y(B,22)=Y(B,12)
                 Y(B,32)=Y(B,1)
                 Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
                 X(B,12)=(Y(B,12)-Y(B,1))/SA+X(B,1)
 141             CONTINUE
                 VOL=(X(B,12)-X(B,20))*(Y(B,12)-Y(B,20))/2.0+(X(B,12)-
     *             X(B,20))*(Y(B,20)-Y(B,2))
                 VOL=VOL*3600.0
                 IF(IFIX(.5+VOL).GT.IFIX(.5+SVOL(N,I))) THEN DO
                   Y(B,32)=Y(B,12)
                   Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
                   X(B,12)=(Y(B,12)-Y(B,1))/SA+X(B,1)
                   GO TO 141
                 END IF
                 IF(IFIX(.5+VOL).LT.IFIX(.5+SVOL(N,I))) THEN DO
                   Y(B,22)=Y(B,12)
                   Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
                   X(B,12)=(Y(B,12)-Y(B,1))/SA+X(B,1)
                   GO TO 141
                 END IF
               END IF
               GO TO 160
             END IF
             IF(IFIX(.5+VOL).LT.IFIX(.5+SVOL(N,I))) THEN DO
               Y(B,12)=(Y(B,1)-Y(B,2))/2.0+Y(B,2)
               X(B,12)=(Y(B,12)-Y(B,2))/SR+X(B,4)
               LOV=(X(B,12)-X(B,1))*(Y(B,1)-Y(B,12))/2.0+(X(B,12)-
     *             X(B,1))*(Y(B,12)-Y(B,2))
               LOV=LOV*3600.0
               VOL=VLB+LOV
               IF(IFIX(.5+VOL).LT.IFIX(.5+SVOL(N,I))) THEN DO
                 Y(B,22)=Y(B,2)
                 Y(B,32)=Y(B,12)
                 Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
                 X(B,12)=(Y(B,12)-Y(B,2))/SR+X(B,4)
 142             CONTINUE
                 LOV=(X(B,12)-X(B,1))*(Y(B,1)-Y(B,12))/2.0+(X(B,12)-
     *             X(B,1))*(Y(B,12)-Y(B,2))
                 LOV=LOV*3600.0
```

```
              VOL=VLB+LOV
              IF(IFIX(.5+VOL).LT.IFIX(.5+SVOL(N,I))) THEN DO
                Y(B,32)=Y(B,12)
                Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
                X(B,12)=(Y(B,12)-Y(B,2))/SR+X(B,4)
                GO TO 142
              END IF
              IF(IFIX(.5+VOL).GT.IFIX(.5+SVOL(N,I))) THEN DO
                Y(B,22)=Y(B,12)
                Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
                X(B,12)=(Y(B,12)-Y(B,2))/SR+X(B,4)
                GO TO 142
              END IF
            END IF
            IF(IFIX(.5+VOL).GT.IFIX(.5+SVOL(N,I))) THEN DO
              Y(B,32)=Y(B,1)
              Y(B,22)=Y(B,12)
              Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
              X(B,12)=(Y(B,12)-Y(B,2))/SR+X(B,4)
143           CONTINUE
              LOV=(X(B,12)-X(B,1))*(Y(B,1)-Y(B,12))/2.0+(X(B,12)-
     *            X(B,1))*(Y(B,12)-Y(B,2))
              LOV=LOV*3600.0
              VOL=VLB+LOV
              IF(IFIX(.5+VOL).GT.IFIX(.5+SVOL(N,I))) THEN DO
                Y(B,22)=Y(B,12)
                Y(B,12)=(Y(B,32)-Y(B,22))/2.00000+Y(B,22)
                X(B,12)=(Y(B,12)-Y(B,2))/SR+X(B,4)
                GO TO 143
              END IF
              IF(IFIX(.5+VOL).LT.IFIX(.5+SVOL(N,I))) THEN DO
                Y(B,32)=Y(B,12)
                Y(B,12)=(Y(B,32)-Y(B,22))/2.00000+Y(B,22)
                X(B,12)=(Y(B,12)-Y(B,2))/SR+X(B,4)
                GO TO 143
              END IF
            END IF
          END IF
          GO TO 160
        END IF
        IF(X(B,12).EQ.X(B,1)) THEN DO
          VOL=(X(B,4)-X(B,1))*(Y(B,1)-Y(B,2))/2.0
          VOL=VOL*3600.0
          Y(B,20)=Y(B,12)
          X(B,20)=X(B,12)
          IF(IFIX(.5+VOL).LE.IFIX(.5+SVOL(N,I))) THEN DO
            GO TO 180
          END IF
          IF(IFIX(.5+VOL).GT.IFIX(.5+SVOL(N,I))) THEN DO
            Y(B,32)=Y(B,1)
            Y(B,22)=Y(B,2)
            Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
            X(B,12)=(Y(B,12)-Y(B,2))/SR+X(B,4)
144         CONTINUE
            VOL=(X(B,12)-X(B,1))*(Y(B,1)-Y(B,12))/2.0+(X(B,12)-
```

```
     *             X(B,1))*(Y(B,12)-Y(B,2))
                 VOL=VOL*3600.0
                 IF(IFIX(.5+VOL).GT.IFIX(.5+SVOL(N,I))) THEN DO
                   Y(B,22)=Y(B,12)
                   Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
                   X(B,12)=(Y(B,12)-Y(B,2))/SR+X(B,4)
                   GO TO 144
                 END IF
                 IF(IFIX(.5+VOL).LT.IFIX(.5+SVOL(N,I))) THEN DO
                   Y(B,32)=Y(B,12)
                   Y(B,12)=(Y(B,32)-Y(B,22))/2.0+Y(B,22)
                   X(B,12)=(Y(B,12)-Y(B,2))/SR+X(B,4)
                   GO TO 144
                 END IF
               END IF
             END IF
             GO TO 180
           END IF
C ####################################################################
C     SHOULD HAVE SVOL AREA FIGURED HERE, REGARDLESS OF INPUT
C       SHAPE. NOW GO TO AREAS WHERE SEGMENTS ARE DIVIDED.
C ####################################################################
150          CONTINUE
             Y(B,20)=Y(B,12)
             X(B,20)=X(B,12)
             DURX=((X(B,4)-X(B,12))/2.0+X(B,12))-X(B,12)
             VL=SVOL(N,I)/4.0
             VL2=VL*2
             Y(B,14)=(Y(B,12)-Y(B,20))/2.0+Y(B,20)
             X(B,14)=(Y(B,14)-Y(B,20))/SA+X(B,20)
             VLA=((X(B,14)-X(B,20))*(Y(B,14)-Y(B,20)))/2.0+(X(B,14)-
     *           X(B,20))*(Y(B,20)-Y(B,2))
                 VLA=VLA*3600.0
             IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
               Y(B,34)=Y(B,14)
               Y(B,24)=Y(B,20)
               Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
               X(B,14)=(Y(B,14)-Y(B,20))/SA+X(B,20)
151            CONTINUE
               VLA=((X(B,14)-X(B,20))*(Y(B,14)-Y(B,20)))/2.0+(X(B,14)-
     *             X(B,20))*(Y(B,20)-Y(B,2))
                 VLA=VLA*3600.0
               IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
                 Y(B,34)=Y(B,14)
                 Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
                 X(B,14)=(Y(B,14)-Y(B,20))/SA+X(B,20)
                 GO TO 151
               END IF
               IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
                 Y(B,24)=Y(B,14)
                 Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
                 X(B,14)=(Y(B,14)-Y(B,20))/SA+X(B,20)
                 GO TO 151
               END IF
             END IF
```

```
         IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
           Y(B,24)=Y(B,14)
           Y(B,34)=Y(B,12)
           Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
           X(B,14)=(Y(B,14)-Y(B,20))/SA+X(B,20)
152        CONTINUE
           VLA=((X(B,14)-X(B,20))*(Y(B,14)-Y(B,20)))/2.0+(X(B,14)-
     *         X(B,20))*(Y(B,20)-Y(B,2))
             VLA=VLA*3600.0
           IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
             Y(B,24)=Y(B,14)
             Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
             X(B,14)=(Y(B,14)-Y(B,20))/SA+X(B,20)
             GO TO 152
           END IF
           IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
             Y(B,34)=Y(B,14)
             Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
             X(B,14)=(Y(B,14)-Y(B,20))/SA+X(B,20)
             GO TO 152
           END IF
         END IF
C    ################### KNOW X(B,14), Y(B,14) ####################
C ################### IF SVOL.LE.PEAK    ##########################
         Y(B,13)=(Y(B,14)-Y(B,20))/2.0+Y(B,20)
         X(B,13)=(Y(B,13)-Y(B,20))/SA+X(B,20)
         VLA=((X(B,13)-X(B,20))*(Y(B,13)-Y(B,20)))/2.0+(X(B,13)-
     *       X(B,20))*(Y(B,20)-Y(B,2))
             VLA=VLA*3600.0
         IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
           Y(B,33)=Y(B,13)
           Y(B,23)=Y(B,20)
           Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
           X(B,13)=(Y(B,13)-Y(B,20))/SA+X(B,20)
153        CONTINUE
           VLA=((X(B,13)-X(B,20))*(Y(B,13)-Y(B,20)))/2.0+(X(B,13)-
     *         X(B,20))*(Y(B,20)-Y(B,2))
             VLA=VLA*3600.0
           IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
             Y(B,33)=Y(B,13)
             Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
             X(B,13)=(Y(B,13)-Y(B,20))/SA+X(B,20)
             GO TO 153
           END IF
           IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
             Y(B,23)=Y(B,13)
             Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
             X(B,13)=(Y(B,13)-Y(B,20))/SA+X(B,20)
             GO TO 153
           END IF
         END IF
         IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
           Y(B,23)=Y(B,13)
           Y(B,33)=Y(B,14)
           Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
```

```
          X(B,13)=(Y(B,13)-Y(B,20))/SA+X(B,20)
154       CONTINUE
          VLA=((X(B,13)-X(B,20))*(Y(B,13)-Y(B,20)))/2.0+(X(B,13)-
     *       X(B,20))*(Y(B,20)-Y(B,2))
            VLA=VLA*3600.0
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
            Y(B,33)=Y(B,13)
            Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
            X(B,13)=(Y(B,13)-Y(B,20))/SA+X(B,20)
            GO TO 154
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
            Y(B,23)=Y(B,13)
            Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
            X(B,13)=(Y(B,13)-Y(B,20))/SA+X(B,20)
            GO TO 154
          END IF
        END IF
C ************* KNOW X(B,13), Y(B,13) **********************
C ************* IF SVOL.LE.PEAK       *************************
        Y(B,15)=(Y(B,12)-Y(B,14))/2.0+Y(B,14)
        X(B,15)=(Y(B,15)-Y(B,14))/SA+X(B,14)
        VLA=((X(B,15)-X(B,14))*(Y(B,15)-Y(B,14)))/2.0+(X(B,15)-
     *       X(B,14))*(Y(B,14)-Y(B,2))
             VLA=VLA*3600.0
        IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
          Y(B,25)=Y(B,15)
          Y(B,35)=Y(B,12)
          Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
          X(B,15)=(Y(B,15)-Y(B,2))/SA+X(B,3)
155       CONTINUE
          VLA=((X(B,15)-X(B,14))*(Y(B,15)-Y(B,14)))/2.0+(X(B,15)-
     *    X(B,14))*(Y(B,14)-Y(B,2))
            VLA=VLA*3600.0
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
            Y(B,35)=Y(B,15)
            Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
            X(B,15)=(Y(B,15)-Y(B,25))/SA+X(B,15)
            GO TO 155
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
            Y(B,25)=Y(B,15)
            Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
            X(B,15)=(Y(B,15)-Y(B,25))/SA+X(B,15)
            GO TO 155
          END IF
        END IF
        IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
          Y(B,35)=Y(B,15)
          Y(B,25)=Y(B,14)
          Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
          X(B,15)=(Y(B,15)-Y(B,25))/SA+X(B,25)
156       CONTINUE
          VLA=((X(B,15)-X(B,14))*(Y(B,15)-Y(B,14)))/2.0+(X(B,15)-
     *    X(B,14))*(Y(B,14)-Y(B,2))
```

```
                VLA=VLA*3600.0
              IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
                Y(B,35)=Y(B,15)
                Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
                X(B,15)=(Y(B,15)-Y(B,25))/SA+X(B,15)
                GO TO 156
              END IF
              IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
                Y(B,25)=Y(B,15)
                Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
                X(B,15)=(Y(B,15)-Y(B,25))/SA+X(B,15)
                GO TO 156
              END IF
            END IF
C ############# KNOW ALL TIME DIV COORDS IF SVOL<PEAK#########
            GO TO 200
C ########NEXT PART CALCULATES DIVISIONS IF SVOL.GT.PEAK #######
160         CONTINUE
            DURX=((X(B,4)-X(B,12))/2.0+X(B,12))-X(B,12)
            VL=SVOL(N,I)/4.0
            VL2=VL*2.0
            Y(B,14)=Y(B,1)
            X(B,14)=X(B,1)
            Y(B,24)=Y(B,20)
            X(B,24)=X(B,20)
            VLA=((X(B,14)-X(B,3))*(Y(B,14)-Y(B,20)))/2.0+(X(B,14)-
     #          X(B,20))*(Y(B,20)-Y(B,2))
                VLA=VLA*3600.0
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
              Y(B,34)=Y(B,14)
              Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
              X(B,14)=(Y(B,14)-Y(B,20))/SA+X(B,20)
              VLA=((X(B,14)-X(B,3))*(Y(B,14)-Y(B,20)))/2.0+(X(B,14)-
     #            X(B,20))*(Y(B,20)-Y(B,2))
                VLA=VLA*3600.0
              IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
                Y(B,34)=Y(B,14)
                Y(B,24)=Y(B,20)
                Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
                X(B,14)=(Y(B,14)-Y(B,20))/SA+X(B,20)
161             CONTINUE
                VLA=((X(B,14)-X(B,3))*(Y(B,14)-Y(B,20)))/2.0+(X(B,14)-
     #              X(B,20))*(Y(B,20)-Y(B,2))
                VLA=VLA*3600.0
                IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
                  Y(B,34)=Y(B,14)
                  Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
                  X(B,14)=(Y(B,14)-Y(B,20))/SA+X(B,20)
                  GO TO 161
                END IF
                IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
                  Y(B,24)=Y(B,14)
                  Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
                  X(B,14)=(Y(B,14)-Y(B,20))/SA+X(B,20)
                  GO TO 161
```

```
              END IF
            END IF
            IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
              Y(B,24)=Y(B,14)
              Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
              X(B,14)=(Y(B,14)-Y(B,20))/SA+X(B,20)
162           CONTINUE
              VLA=((X(B,14)-X(B,3))*(Y(B,14)-Y(B,20)))/2.0+(X(B,14)-
     #            X(B,20))*(Y(B,20)-Y(B,2))
              VLA=VLA*3600.0
              IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
                Y(B,34)=Y(B,14)
                Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
                X(B,14)=(Y(B,14)-Y(B,20))/SA+X(B,20)
                GO TO 162
              END IF
              IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
                Y(B,24)=Y(B,14)
                Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
                X(B,14)=(Y(B,14)-Y(B,20))/SA+X(B,20)
                GO TO 162
              END IF
            END IF
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
            Y(B,34)=Y(B,14)
            Y(B,24)=Y(B,12)
            Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
            X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
            VLA=((X(B,12)-X(B,14))*(Y(B,14)-Y(B,12)))/2.0+(X(B,12)-
     *          X(B,14))*(Y(B,12)-Y(B,2))
              VLA=VLA*3600.0
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
              Y(B,34)=Y(B,14)
              Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
              X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
163           CONTINUE
              VLA=((X(B,12)-X(B,14))*(Y(B,14)-Y(B,12)))/2.0+(X(B,12)-
     *            X(B,14))*(Y(B,12)-Y(B,2))
              VLA=VLA*3600.0
              IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
                Y(B,24)=Y(B,14)
                Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
                X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
                GO TO 163
              END IF
              IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
                Y(B,34)=Y(B,14)
                Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
                X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
                GO TO 163
              END IF
            END IF
            IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
              Y(B,24)=Y(B,14)
```

```
          Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
          X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
164       CONTINUE
          VLA=((X(B,12)-X(B,14))*(Y(B,14)-Y(B,12)))/2.0+(X(B,12)-
     *        X(B,14))*(Y(B,12)-Y(B,2))
          VLA=VLA*3600.0
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
            Y(B,24)=Y(B,14)
            Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
            X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
            GO TO 164
          END IF
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
            Y(B,34)=Y(B,14)
            Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
            X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
            GO TO 164
          END IF
        END IF
      END IF
C     ############## KNOW X(B,14), Y(B,14) ######################
C     ################################################################
C     FROM HERE X,Y(B,13) IS CALCULATED, FIRST FOR X(B,14)<X(B,1)
C     THEN FOR X(B,14)>X(B,1).
C     ################################################################
      IF(X(B,14).LE.X(B,1)) THEN DO
        Y(B,13)=Y(B,14)-Y(B,20)/2.0+Y(B,20)
        X(B,13)=(Y(B,13)-Y(B,20))/SA+X(B,20)
        VLA=(X(B,13)-X(B,20))*(Y(B,13)-Y(B,20))/2.0+(X(B,13)-
     #        X(B,20))*(Y(B,20)-Y(B,2))
          VLA=VLA*3600.0
        IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
          Y(B,33)=Y(B,13)
          Y(B,23)=Y(B,20)
          Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
          X(B,13)=(Y(B,13)-Y(B,20))/SA+X(B,20)
165       CONTINUE
          VLA=(X(B,13)-X(B,20))*(Y(B,13)-Y(B,20))/2.0+(X(B,13)-
     #        X(B,20))*(Y(B,20)-Y(B,2))
          VLA=VLA*3600.0
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
            Y(B,33)=Y(B,13)
            Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
            X(B,13)=(Y(B,13)-Y(B,20))/SA+X(B,20)
            GO TO 165
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
            Y(B,23)=Y(B,13)
            Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
            X(B,13)=(Y(B,13)-Y(B,20))/SA+X(B,20)
            GO TO 165
          END IF
        END IF
        IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
          Y(B,23)=Y(B,13)
```

```
                Y(B,33)=Y(B,14)
                Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
                X(B,13)=(Y(B,13)-Y(B,20))/SA+X(B,20)
166             CONTINUE
                VLA=(X(B,13)-X(B,20))*(Y(B,13)-Y(B,20))/2.0+(X(B,13)-
     #              X(B,20))*(Y(B,20)-Y(B,2))
                VLA=VLA*3600.0
                IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
                  Y(B,33)=Y(B,13)
                  Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
                  X(B,13)=(Y(B,13)-Y(B,20))/SA+X(B,20)
                  GO TO 166
                END IF
                IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
                  Y(B,23)=Y(B,13)
                  Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
                  X(B,13)=(Y(B,13)-Y(B,20))/SA+X(B,20)
                  GO TO 166
                END IF
              END IF
C ########### Y(B,13), X(B,13) KNOWN IF X(B,14).LT. PEAK #######
            END IF
            IF(X(B,14).GT.X(B,1)) THEN DO
              Y(B,13)=Y(B,1)
              X(B,13)=X(B,1)
              Y(B,23)=Y(B,20)
              X(B,23)=X(B,20)
              VLA=(X(B,13)-X(B,20))*(Y(B,13)-Y(B,20))/2.0+(X(B,13)-
     #            X(B,20))*(Y(B,20)-Y(B,2))
                VLA=VLA*3600.0
              IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
                Y(B,33)=Y(B,13)
                Y(B,13)=(Y(B,13)-Y(B,23))/2.0+Y(B,23)
                X(B,13)=(Y(B,13)-Y(B,23))/SA+X(B,23)
                VLA=(X(B,13)-X(B,20))*(Y(B,13)-Y(B,20))/2.0+(X(B,13)-
     #              X(B,20))*(Y(B,20)-Y(B,2))
                VLA=VLA*3600.0
                IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
                  Y(B,33)=Y(B,13)
                  Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
                  X(B,13)=(Y(B,13)-Y(B,20))/SA+X(B,20)
167               CONTINUE
                  VLA=(X(B,13)-X(B,20))*(Y(B,13)-Y(B,20))/2.0+(X(B,13)-
     #                X(B,20))*(Y(B,20)-Y(B,2))
                VLA=VLA*3600.0
                  IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
                    Y(B,33)=Y(B,13)
                    Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
                    X(B,13)=(Y(B,13)-Y(B,20))/SA+X(B,20)
                    GO TO 167
                  END IF
                  IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
                    Y(B,23)=Y(B,13)
                    Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
                    X(B,13)=(Y(B,13)-Y(B,20))/SA+X(B,20)
```

```
                GO TO 167
              END IF
            END IF
            IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
              Y(B,23)=Y(B,13)
              Y(B,33)=Y(B,1)
              Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
              X(B,13)=(Y(B,13)-Y(B,20))/SA+X(B,20)
168           CONTINUE
              VLA=(X(B,13)-X(B,20))*(Y(B,13)-Y(B,20))/2.0+(X(B,13)-
     #            X(B,20))*(Y(B,20)-Y(B,2))
            VLA=VLA*3600.0
              IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
                Y(B,33)=Y(B,13)
                Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
                X(B,13)=(Y(B,13)-Y(B,20))/SA+X(B,20)
                GO TO 168
              END IF
              IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
                Y(B,23)=Y(B,13)
                Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
                X(B,13)=(Y(B,13)-Y(B,20))/SA+X(B,20)
                GO TO 168
              END IF
            END IF
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
            Y(B,33)=Y(B,13)
            X(B,33)=X(B,13)
            Y(B,23)=Y(B,14)
            X(B,23)=X(B,14)
            Y(B,13)=(Y(B,33)-Y(B,23)/2.0+Y(B,23))
            X(B,13)=(Y(B,13)-Y(B,14))/SR+X(B,14)
            VLA=(X(B,14)-X(B,13))*(Y(B,13)-Y(B,14))/2.0+(X(B,14)-
     *          X(B,13))*(Y(B,14)-Y(B,2))
            VLA=VLA*3600.0
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
              Y(B,33)=Y(B,13)
              Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
              X(B,13)=(Y(B,13)-Y(B,14))/SR+X(B,14)
169           CONTINUE
              VLA=(X(B,14)-X(B,13))*(Y(B,13)-Y(B,14))/2.0+(X(B,14)-
     *            X(B,13))*(Y(B,14)-Y(B,2))
            VLA=VLA*3600.0
              IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
                Y(B,33)=Y(B,13)
                Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
                X(B,13)=(Y(B,13)-Y(B,14))/SR+X(B,14)
                GO TO 169
              END IF
              IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
                Y(B,23)=Y(B,13)
                Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
                X(B,13)=(Y(B,13)-Y(B,14))/SR+X(B,14)
                GO TO 169
```

```
                END IF
              END IF
              IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
                Y(B,23)=Y(B,13)
                Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
                X(B,13)=(Y(B,13)-Y(B,14))/SR+X(B,14)
170             CONTINUE
                VLA=(X(B,14)-X(B,13))*(Y(B,13)-Y(B,14))/2.0+(X(B,14)-
     *              X(B,13))*(Y(B,14)-Y(B,2))
              VLA=VLA*3600.0
                IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
                  Y(B,33)=Y(B,13)
                  Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
                  X(B,13)=(Y(B,13)-Y(B,14))/SR+X(B,14)
                  GO TO 170
                END IF
                IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
                  Y(B,23)=Y(B,13)
                  Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
                  X(B,13)=(Y(B,13)-Y(B,14))/SR+X(B,14)
                  GO TO 170
                END IF
              END IF
            END IF
          END IF
C ############# KNOW Y(B,13), X(B,13) #########################
C ############# THIS CALCULATES X,Y(B,15), FIRST FOR X(B,14)>X(B,1)
C ######### THEN FOR X(B,14)<X(B,1) ######################
          IF(X(B,14).GT.X(B,1)) THEN DO
            Y(B,15)=Y(B,14)-Y(B,12)/2.0+Y(B,12)
            X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
            VLA=(X(B,12)-X(B,15))*(Y(B,15)-Y(B,12))/2.0+(X(B,12)-
     *          X(B,15))*(Y(B,12)-Y(B,2))
              VLA=VLA*3600.0
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
              Y(B,35)=Y(B,15)
              Y(B,25)=Y(B,12)
              Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
              X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
171           CONTINUE
              VLA=(X(B,12)-X(B,15))*(Y(B,15)-Y(B,12))/2.0+(X(B,12)-
     *            X(B,15))*(Y(B,12)-Y(B,2))
              VLA=VLA*3600.0
              IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
                Y(B,35)=Y(B,15)
                Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
                X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
                GO TO 171
              END IF
              IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
                Y(B,25)=Y(B,15)
                Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
                X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
                GO TO 171
              END IF
```

```
           END IF
           IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
             Y(B,25)=Y(B,15)
             Y(B,35)=Y(B,14)
             Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
             X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
172          CONTINUE
             VLA=(X(B,12)-X(B,15))*(Y(B,15)-Y(B,12))/2.0+(X(B,12)-
     *           X(B,15))*(Y(B,12)-Y(B,2))
             VLA=VLA*3600.0
             IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
               Y(B,35)=Y(B,15)
               Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
               X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
               GO TO 172
             END IF
             IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
               Y(B,25)=Y(B,15)
               Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
               X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
               GO TO 172
             END IF
           END IF
         END IF
C ########### Y(B,15), X(B,15) KNOWN IF X(B,14).GT. PEAK #######
         IF(X(B,14).LT.X(B,1)) THEN DO
           Y(B,15)=Y(B,1)
           X(B,15)=X(B,1)
           VLA=(X(B,12)-X(B,15))*(Y(B,15)-Y(B,12))/2.0+(X(B,12)-
     *         X(B,15))*(Y(B,12)-Y(B,2))
             VLA=VLA*3600.0
           IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
             Y(B,35)=Y(B,15)
             Y(B,15)=(Y(B,15)-Y(B,12))/2.0+Y(B,12)
             X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
             VLA=(X(B,12)-X(B,15))*(Y(B,15)-Y(B,12))/2.0+(X(B,12)-
     *           X(B,15))*(Y(B,12)-Y(B,2))
             VLA=VLA*3600.0
             IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
               Y(B,35)=Y(B,15)
               Y(B,25)=Y(B,12)
               Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
               X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
173            CONTINUE
               VLA=(X(B,12)-X(B,15))*(Y(B,15)-Y(B,12))/2.0+(X(B,12)-
     *             X(B,15))*(Y(B,12)-Y(B,2))
             VLA=VLA*3600.0
               IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
                 Y(B,35)=Y(B,15)
                 Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
                 X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
                 GO TO 173
               END IF
               IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
                 Y(B,25)=Y(B,15)
```

```
            Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
            X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
            GO TO 173
          END IF
        END IF
        IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
          Y(B,25)=Y(B,15)
          Y(B,35)=Y(B,1)
          Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
          X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
174       CONTINUE
          VLA=(X(B,12)-X(B,15))*(Y(B,15)-Y(B,12))/2.0+(X(B,12)-
     *         X(B,15))*(Y(B,12)-Y(B,2))
        VLA=VLA*3600.0
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
            Y(B,35)=Y(B,15)
            Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
            X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
            GO TO 174
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
            Y(B,25)=Y(B,15)
            Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
            X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
            GO TO 174
          END IF
        END IF
      END IF
      IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
        Y(B,35)=Y(B,15)
        X(B,35)=X(B,15)
        Y(B,25)=Y(B,14)
        X(B,25)=X(B,14)
        Y(B,15)=(Y(B,35)-Y(B,25)/2.0+Y(B,25))
        X(B,15)=(Y(B,15)-Y(B,14))/SA+X(B,14)
        VLA=(X(B,15)-X(B,14))*(Y(B,15)-Y(B,14))/2.0+(X(B,15)-
     *       X(B,14))*(Y(B,14)-Y(B,2))
        VLA=VLA*3600.0
        IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
          Y(B,35)=Y(B,15)
          Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
          X(B,15)=(Y(B,15)-Y(B,14))/SA+X(B,14)
175       CONTINUE
          VLA=(X(B,15)-X(B,14))*(Y(B,15)-Y(B,14))/2.0+
     *         (X(B,15)-X(B,14))*(Y(B,14)-Y(B,2))
        VLA=VLA*3600.0
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
            Y(B,35)=Y(B,15)
            Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
            X(B,15)=(Y(B,15)-Y(B,14))/SA+X(B,14)
            GO TO 175
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
            Y(B,25)=Y(B,15)
            Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
```

```
                X(B,15)=(Y(B,15)-Y(B,14))/SA+X(B,14)
                GO TO 175
              END IF
            END IF
            IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
              Y(B,25)=Y(B,15)
              Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
              X(B,15)=(Y(B,15)-Y(B,14))/SA+X(B,14)
176           CONTINUE
              VLA=(X(B,15)-X(B,14))*(Y(B,15)-Y(B,14))/2.0+
     *            (X(B,15)-X(B,14))*(Y(B,14)-Y(B,2))
            VLA=VLA*3600.0
              IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
                Y(B,35)=Y(B,15)
                Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
                X(B,15)=(Y(B,15)-Y(B,14))/SA+X(B,14)
                GO TO 176
              END IF
              IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
                Y(B,25)=Y(B,15)
                Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
                X(B,15)=(Y(B,15)-Y(B,14))/SA+X(B,14)
                GO TO 176
              END IF
            END IF
          END IF
        END IF
        GO TO 200
C ########### NOW HAVE ALL DIVISIONS FOR FIRST 2 INPUT #########
C ########### TYPES. NEXT SECTION CALCULATES FOR 3RD INPUT #####
C ########### TYPE.       #####################################
180     CONTINUE
        DURX=((X(B,4)-X(B,12))/2.0+X(B,12))-X(B,12)
        VL=SVOL(N,I)/4.0
        VL2=VL*2.0
        Y(B,14)=(Y(B,12)-Y(B,2))/2.0+Y(B,2)
        X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
        VLA=((X(B,12)-X(B,14))*(Y(B,14)-Y(B,12))/2.0)+((X(B,12)-
     *      X(B,14))*(Y(B,12)-Y(B,2)))
        VLA=VLA*3600.0
        IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
          Y(B,34)=Y(B,14)
          X(B,34)=X(B,14)
          Y(B,24)=Y(B,2)
          X(B,24)=X(B,4)
          Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
          X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
181       CONTINUE
          VLA=((X(B,12)-X(B,14))*(Y(B,14)-Y(B,12))/2.0)+((X(B,12)-
     *        X(B,14))*(Y(B,12)-Y(B,2)))
          VLA=VLA*3600.0
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
            Y(B,34)=Y(B,14)
            X(B,34)=X(B,14)
            Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
```

```
            X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
            GO TO 181
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
            Y(B,24)=Y(B,14)
            X(B,24)=X(B,14)
            Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
            X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
            GO TO 181
          END IF
        END IF
        IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
          Y(B,34)=Y(B,12)
          X(B,34)=X(B,12)
          Y(B,24)=Y(B,14)
          X(B,24)=X(B,14)
          Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
          X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
182       CONTINUE
          VLA=((X(B,12)-X(B,14))*(Y(B,14)-Y(B,12))/2.0)+((X(B,12)-
     *        X(B,14))*(Y(B,12)-Y(B,2)))
          VLA=VLA*3600.0
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL2)) THEN DO
            Y(B,34)=Y(B,14)
            X(B,34)=X(B,14)
            Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
            X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
            GO TO 182
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL2)) THEN DO
            Y(B,24)=Y(B,14)
            X(B,24)=X(B,14)
            Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
            X(B,14)=(Y(B,14)-Y(B,2))/SR+X(B,4)
            GO TO 182
          END IF
        END IF
C ##################X,Y(B,14) KNOWN FOR THIS INPUT #############
C ################## FOR X,Y(B.,13)......          #############
        Y(B,13)=(Y(B,12)-Y(B,14))/2.0+Y(B,2)
        X(B,13)=(Y(B,13)-Y(B,2))/SR+X(B,4)
        VLA=(X(B,14)-X(B,13))*(Y(B,13)-Y(B,14))/2.0+(X(B,14)-
     *      X(B,13))*(Y(B,14)-Y(B,2))
        VLA=VLA*3600.0
        IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
          Y(B,33)=Y(B,13)
          X(B,33)=X(B,13)
          Y(B,23)=Y(B,14)
          X(B,23)=X(B,14)
          Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
          X(B,13)=(Y(B,13)-Y(B,2))/SR+X(B,4)
183       CONTINUE
          VLA=(X(B,14)-X(B,13))*(Y(B,13)-Y(B,14))/2.0+(X(B,14)-
     *        X(B,13))*(Y(B,14)-Y(B,2))
          VLA=VLA*3600.0
```

```
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
            Y(B,33)=Y(B,13)
            X(B,33)=X(B,13)
            Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
            X(B,13)=(Y(B,13)-Y(B,2))/SR+X(B,4)
            GO TO 183
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
            Y(B,23)=Y(B,13)
            X(B,23)=X(B,13)
            Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
            X(B,13)=(Y(B,13)-Y(B,2))/SR+X(B,4)
            GO TO 183
          END IF
        END IF
        IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
          Y(B,33)=Y(B,12)
          X(B,33)=X(B,12)
          Y(B,23)=Y(B,14)
          X(B,23)=X(B,14)
          Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
          X(B,13)=(Y(B,13)-Y(B,2))/SR+X(B,4)
184       CONTINUE
          VLA=(X(B,14)-X(B,13))*(Y(B,13)-Y(B,14))/2.0+(X(B,14)-
     *        X(B,13))*(Y(B,14)-Y(B,2))
          VLA=VLA*3600.0
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
            Y(B,33)=Y(B,13)
            X(B,33)=X(B,13)
            Y(B,13)=(Y(B,33)-Y(B,23))/2.0+Y(B,23)
            X(B,13)=(Y(B,13)-Y(B,2))/SR+X(B,4)
            GO TO 184
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
            Y(B,23)=Y(B,13)
            X(B,23)=X(B,13)
            Y(B,14)=(Y(B,34)-Y(B,24))/2.0+Y(B,24)
            X(B,13)=(Y(B,13)-Y(B,2))/SR+X(B,4)
            GO TO 184
          END IF
        END IF
C ############# X,Y(B,13) KNOWN  ################################
C ############# NOW X,Y(B,15)    ################################
        Y(B,15)=(Y(B,14)-Y(B,2))/2.0+Y(B,2)
        X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
        VLA=((X(B,12)-X(B,15))*(Y(B,15)-Y(B,12))/2.0)+((X(B,12)-
     *      X(B,15))*(Y(B,12)-Y(B,2)))
        VLA=VLA*3600.0
        IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
          Y(B,35)=Y(B,15)
          X(B,35)=X(B,15)
          Y(B,25)=Y(B,2)
          X(B,25)=X(B,4)
          Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
          X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
```

```
185         CONTINUE
            VLA=((X(B,12)-X(B,15))*(Y(B,15)-Y(B,12))/2.0)+((X(B,12)-
     *          X(B,15))*(Y(B,12)-Y(B,2)))
            VLA=VLA*3600.0
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
              Y(B,35)=Y(B,15)
              X(B,35)=X(B,15)
              Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
              X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
              GO TO 185
            END IF
            IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
              Y(B,25)=Y(B,15)
              X(B,25)=X(B,15)
              Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
              X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
              GO TO 185
            END IF
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
            Y(B,35)=Y(B,14)
            X(B,35)=X(B,14)
            Y(B,25)=Y(B,15)
            X(B,25)=X(B,15)
            Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
            X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
186         CONTINUE
            VLA=((X(B,12)-X(B,15))*(Y(B,15)-Y(B,12))/2.0)+((X(B,12)-
     *          X(B,15))*(Y(B,12)-Y(B,2)))
            VLA=VLA*3600.0
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
              Y(B,35)=Y(B,15)
              X(B,35)=X(B,15)
              Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
              X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
              GO TO 186
            END IF
            IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
              Y(B,25)=Y(B,15)
              X(B,25)=X(B,15)
              Y(B,15)=(Y(B,35)-Y(B,25))/2.0+Y(B,25)
              X(B,15)=(Y(B,15)-Y(B,2))/SR+X(B,4)
              GO TO 186
            END IF
          END IF
C ############# X,Y(B,15) KNOWN  ###############################
C ############# NOW ALL COORDINATES KNOWN ######################
C ############# FOR DIVISIONS REGARDLESS OF INPUT ##############
C ############# OR STORAGE DECISION ############################
C ##### FROM HERE EXTENSION DIVISIONS AND DURATIONS ############
C ##### ARE CALCULATED.            ############################
200     CONTINUE
          VL2=VL2/3600.0
          VAL=SVOL(N,I)/3600.0
        IF(STAT(N,J).EQ.0.0) THEN DO
```

```
          IF(SVOL(N,I).NE.0.0) THEN DO
            G=X(B,4)
            H=X(B,2)
            X(B,5)=H
            VLA=((X(B,5)-G)*Y(B,2))/2.0
            VLA=VLA*3600.0
            IF(IFIX(VLA).LT.IFIX(SVOL(N,I))) THEN DO
              X(B,5)=(VAL-(((H-G)*Y(B,2))/2.0)/Y(B,2))+H
              X(B,6)=X(B,5)-G+H
            END IF
            IF(IFIX(VLA).GE.IFIX(SVOL(N,I))) THEN DO
              X(B,5)=(X(B,5)-G)/2.0+G
              VLA=(X(B,5)-G)*(Y(B,2)-(SR*(X(B,5)-G)+Y(B,2)))
     *            /2.0*3600.0
              IF(IFIX(.5+VLA).GT.IFIX(.5+SVOL(N,I))) THEN DO
                X(B,45)=X(B,5)
                X(B,44)=G
                X(B,5)=(X(B,5)-G)/2.0+G
201             CONTINUE
                VLA=(X(B,5)-G)*(Y(B,2)-(SR*(X(B,5)-G)
     *            +Y(B,2)))/2.0*3600.0
                IF(IFIX(.5+VLA).GT.IFIX(.5+SVOL(N,I))) THEN DO
                  X(B,45)=X(B,5)
                  X(B,5)=(X(B,45)-X(B,44))/2.0+X(B,44)
                  GO TO 201
                END IF
                IF(IFIX(.5+VLA).LT.IFIX(.5+SVOL(N,I))) THEN DO
                  X(B,44)=X(B,5)
                  X(B,5)=(X(B,45)-X(B,44))/2.0+X(B,44)
                  GO TO 201
                END IF
              END IF
              IF(IFIX(.5+VLA).LT.IFIX(.5+SVOL(N,I))) THEN DO
                X(B,45)=H
                X(B,44)=X(B,5)
                X(B,5)=(X(B,45)-X(B,44))/2.0+X(B,44)
202             CONTINUE
                VLA=(X(B,5)-G)*(Y(B,2)-(SR*(X(B,5)-G)
     *            +Y(B,2)))/2.0*3600.0
                IF(IFIX(.5+VLA).GT.IFIX(.5+SVOL(N,I))) THEN DO
                  X(B,45)=X(B,5)
                  X(B,5)=(X(B,45)-X(B,44))/2.0+X(B,44)
                  GO TO 202
                END IF
                IF(IFIX(.5+VLA).LT.IFIX(.5+SVOL(N,I))) THEN DO
                  X(B,44)=X(B,5)
                  X(B,5)=(X(B,45)-X(B,44))/2.0+X(B,44)
                  GO TO 202
                END IF
              END IF
            END IF
            X(B,6)=X(B,5)-G+H
          END IF
        END IF
        IF(STAT(N,J).NE.0.0) THEN DO
```

```
      IF(SVOL(N,I).NE.0.0) THEN DO
        G=X(B,5)
        H=X(B,6)
        X(B,5)=H
        VLA=((X(B,5)-G)*Y(B,2))/2.0
        VLA=VLA*3600.0
        IF(IFIX(VLA).LT.IFIX(SVOL(N,I))) THEN DO
          X(B,5)=(VAL-(((H-G)*Y(B,2))/2.0))/Y(B,2)+H
          X(B,6)=X(B,5)-G+H
        END IF
        IF(IFIX(VLA).GE.IFIX(SVOL(N,I))) THEN DO
          X(B,5)=(X(B,5)-G)/2.0+G
          VLA=(X(B,5)-G)*(Y(B,2)-(SR*(X(B,5)-G)+Y(B,2)))
     *        /2.0*3600.0
          IF(IFIX(.5+VLA).GT.IFIX(.5+SVOL(N,I))) THEN DO
            X(B,45)=X(B,5)
            X(B,44)=G
            X(B,5)=(X(B,5)-G)/2.0+G
203         CONTINUE
            VLA=(X(B,5)-G)*(Y(B,2)-(SR*(X(B,5)-G)
     *        +Y(B,2)))/2.0*3600.0
            IF(IFIX(.5+VLA).GT.IFIX(.5+SVOL(N,I))) THEN DO
              X(B,45)=X(B,5)
              X(B,5)=(X(B,45)-X(B,44))/2.0+X(B,44)
              GO TO 203
            END IF
            IF(IFIX(.5+VLA).LT.IFIX(.5+SVOL(N,I))) THEN DO
              X(B,44)=X(B,5)
              X(B,5)=(X(B,45)-X(B,44))/2.0+X(B,44)
              GO TO 203
            END IF
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+SVOL(N,I))) THEN DO
            X(B,45)=H
            X(B,44)=X(B,5)
            X(B,5)=(X(B,45)-X(B,44))/2.0+X(B,44)
204         CONTINUE
            VLA=(X(B,5)-G)*(Y(B,2)-(SR*(X(B,5)-G)
     *        +Y(B,2)))/2.0*3600.0
            IF(IFIX(.5+VLA).GT.IFIX(.5+SVOL(N,I))) THEN DO
              X(B,45)=X(B,5)
              X(B,5)=(X(B,45)-X(B,44))/2.0+X(B,44)
              GO TO 204
            END IF
            IF(IFIX(.5+VLA).LT.IFIX(.5+SVOL(N,I))) THEN DO
              X(B,44)=X(B,5)
              X(B,5)=(X(B,45)-X(B,44))/2.0+X(B,44)
              GO TO 204
            END IF
          END IF
        END IF
        X(B,6)=X(B,5)-G+X(B,6)
      END IF
    END IF
    IF(X(B,5).GE.H) THEN DO
```

```
          X(B,17)=X(B,5)-VL2/Y(B,2)
          X(B,18)=(X(B,5)-X(B,17))/2.0+X(B,17)
          X(B,16)=H
          VLA=(X(B,16)-G)*(Y(B,2)-Y(B,3))/2.0*3600.0
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
            X(B,16)=(H-G)/2.0+G
            VLA=(Y(B,2)-(SR*(X(B,16)-G)+Y(B,2)))*(X(B,16)-G)/2.0*3600.0
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
              X(B,36)=X(B,16)
              X(B,26)=G
              X(B,16)=(X(B,36)-X(B,26))/2.0+X(B,26)
205           CONTINUE
              VLA=(Y(B,2)-(SR*(X(B,16)-G)+Y(B,2)))*(X(B,16)-G)/2.0
              VLA=VLA*3600.0
              IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
                X(B,36)=X(B,16)
                X(B,16)=(X(B,36)-X(B,26))/2.0+X(B,26)
                GO TO 205
              END IF
              IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
                X(B,26)=X(B,16)
                X(B,16)=(X(B,36)-X(B,26))/2.0+X(B,26)
                GO TO 205
              END IF
            END IF
            IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
              X(B,36)=H
              X(B,26)=X(B,16)
              X(B,16)=(X(B,36)-X(B,26))/2.0+X(B,26)
206           CONTINUE
              VLA=(Y(B,2)-(SR*(X(B,16)-G)+Y(B,2)))*(X(B,16)-G)/2.0
              VLA=VLA*3600.0
              IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
                X(B,36)=X(B,16)
                X(B,16)=(X(B,36)-X(B,26))/2.0+X(B,26)
                GO TO 206
              END IF
              IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
                X(B,26)=X(B,16)
                X(B,16)=(X(B,36)-X(B,26))/2.0+X(B,26)
                GO TO 206
              END IF
            END IF
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
            X(B,16)=(X(B,17)-X(B,16))/2.0+X(B,16)
            VLA=(X(B,16)-H)*(Y(B,2)-Y(B,3))+(H-G)*(Y(B,2)-Y(B,3))/2.0
            VLA=VLA*3600.0
            IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
              X(B,36)=X(B,16)
              X(B,26)=H
              X(B,16)=(X(B,36)-X(B,26))/2.0+X(B,26)
207           CONTINUE
              VLA=(X(B,16)-H)*(Y(B,2)-Y(B,3))+(H-G)*(Y(B,2)-Y(B,3))/2.0
              VLA=VLA*3600.0
```

```
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
            X(B,36)=X(B,16)
            X(B,16)=(X(B,36)-X(B,26))/2.0+X(B,26)
            GO TO 207
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
            X(B,26)=X(B,16)
            X(B,16)=(X(B,36)-X(B,26))/2.0+X(B,26)
            GO TO 207
          END IF
        END IF
        IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
          X(B,36)=X(B,17)
          X(B,26)=H
          X(B,16)=(X(B,36)-X(B,26))/2.0+X(B,26)
208       CONTINUE
          VLA=(X(B,16)-H)*(Y(B,2)-Y(B,3))+(H-G)*(Y(B,2)-Y(B,3))/2.0
          VLA=VLA*3600.0
          IF(IFIX(.5+VLA).GT.IFIX(.5+VL)) THEN DO
            X(B,36)=X(B,16)
            X(B,16)=(X(B,36)-X(B,26))/2.0+X(B,26)
            GO TO 208
          END IF
          IF(IFIX(.5+VLA).LT.IFIX(.5+VL)) THEN DO
            X(B,26)=X(B,16)
            X(B,16)=(X(B,36)-X(B,26))/2.0+X(B,26)
            GO TO 208
          END IF
        END IF
      END IF
    END IF
    IF(X(B,5).LT.H) THEN DO
      X(B,17)=(X(B,5)-G)/SQRT(2.0)+G
      X(B,16)=(X(B,5)-G)/2.0+G
      X(B,18)=(X(B,5)-G)/SQRT(1.33)+G
    END IF
      DUR1=((X(B,16)-G)/2.0+G)-((X(B,12)-X(B,15))/2.0+X(B,15))
      DUR2=((X(B,17)-X(B,16))/2.0+X(B,16))-((X(B,15)-X(B,14))/
*          2.0+X(B,14))
      DUR3=((X(B,18)-X(B,17))/2.0+X(B,17))-((X(B,14)-X(B,13))/
*          2.0+X(B,13))
      DUR4=((X(B,5)-X(B,18))/2.0+X(B,18))-((X(B,13)-X(B,3))/
*          2.0+X(B,3))
   IF(SVOL(N,I).GE.VOLH) THEN DO
     SVOL(N,I)=LO
   END IF
   END IF
   END IF
   V(E,F,D,B,1)=SNGL(X(B,1))
   V(E,F,D,B,2)=SNGL(X(B,3))
   V(E,F,D,B,3)=SNGL(X(B,4))
   V(E,F,D,B,4)=SNGL(X(B,5))
   V(E,F,D,B,5)=SNGL(X(B,6))
   V(E,F,D,B,6)=SNGL(X(B,12))
   W(E,F,D,B,1)=SNGL(Y(B,1))
```

```
      W(E,F,D,B,2)=SNGL(Y(B,12))
      RLOV(E,F,D,B)=VOLR
C     PRINT, 'X1=',V(E,F,D,B,1),'X3= ',V(E,F,D,B,2),'X4=',
C    *        V(E,F,D,B,3)
C     PRINT, 'X5=',V(E,F,D,B,4),'X6=',V(E,F,D,B,5),'X12=',
C    *        V(E,F,D,B,6)
C     PRINT, 'Y1=',W(E,F,D,B,1),'Y2=',BF,'Y12=',
C    *        W(E,F,D,B,2)
C     PRINT, 'VOLR=',VOLR
      IF(SVOL(N,I).EQ.0.0) THEN DO
      DUR1=0.0
      DUR2=0.0
      DUR3=0.0
      DUR4=0.0
      END IF
      IF(VOLR.EQ.0.0) THEN DO
        DURX=0.0
      END IF
      RETURN
      END
      REAL FUNCTION DUDW(DUR,JPW)
      INTEGER A,C
      REAL JPW(5,13),DUR3,DUR4,DUR,DUD
      A=1
      DUDW=0.0
      IF(DUR.NE.0.0) THEN DO
      WHILE(A.LE.5) DO
        IF(DUR.LE.48) THEN DO
          DUD=(DUR+168)*0.0487338*(JPW(A,1)+JPW(A,2))+
     *DUR*0.54167*JPW(A,3)+DUR*0.4375*JPW(A,4)+DUR*0.3229*
     *JPW(A,5)+DUR*0.2146*JPW(A,6)+DUR*0.1042*JPW(A,7)+
     *8.19*JPW(A,8)+8.19*JPW(A,9)+8.19*JPW(A,10)+8.19*JPW(A,11)
     *+8.19*JPW(A,12)+50.0*JPW(A,13)
        END IF
        IF(DUR.GT.48.AND.DUR.LE.96) THEN DO
          DUD=(DUR+168)*0.0487338*(JPW(A,1)+JPW(A,2))+(DUR*0.25
     *+14.0)*JPW(A,3)+(DUR*0.3125+6.0)*JPW(A,4)+(DUR*0.3792-2.7)
     **JPW(A,5)+(DUR*0.4313-10.4)*JPW(A,6)+(DUR*0.5014-19.5)*
     *JPW(A,7)+(DUR*0.5729-27.5)*JPW(A,8)+(DUR*0.5313-25.5)*
     *JPW(A,9)+(DUR*0.4896-23.5)*JPW(A,10)+(DUR*0.4479-21.5)*
     *JPW(A,11)+(DUR*0.4479-21.5)*JPW(A,12)+50.0*JPW(A,13)
        END IF
        IF(DUR.GT.96) THEN DO
          DUR3=DUR*0.2361+15.3
          DUR4=DUR*0.0381+34.67
          IF(A.LE.2) THEN DO
            IF(DUR3.GT.53.83) THEN DO
               DUR3=53.83
            END IF
            IF(DUR4.GT.53.83) THEN DO
               DUR4=53.83
            END IF
          END IF
          DUD=(DUR+168)*0.0487338*(JPW(A,1)+JPW(A,2))+DUR3*JPW(A,3)+
     *DUR4*JPW(A,4)+33.7*JPW(A,5)+31.0*JPW(A,6)+29.5*JPW(A,7)+
```

```
     *27.5*JPW(A,8)+25.5*JPW(A,9)+23.5*JPW(A,10)+21.5*JPW(A,11)+
     *21.5*JPW(A,12)+50.0*JPW(A,13)
        END IF
        A=A+1
C       DUDW=DUDW+DUD
C       DUDW=DUDW+(DUD*0.00)
      END WHILE
      END IF
        DUDW=1.0-(DUDW*0.01)
C       PRINT, 'DUDW=  ',DUDW
      RETURN
      END
C ***************************************************************
      REAL FUNCTION DUDB(DUR,JPB)
      INTEGER A,C
      REAL JPB(5,13),DUR,DUR3,DUR4,DUD
      A=1
      DUDB=0.0
      IF(DUR.NE.0.0) THEN DO
      WHILE(A.LE.5) DO
        IF(DUR.LE.48) THEN DO
          DUD=(DUR+168)*0.0279107*(JPB(A,1)+JPB(A,2))+
     *DUR*0.54167*JPB(A,3)+DUR*0.4375*JPB(A,4)+DUR*0.3229*
     *JPB(A,5)+DUR*0.2146*JPB(A,6)+DUR*0.1042*JPB(A,7)+
     *4.69*JPB(A,8)+4.69*JPB(A,9)+4.69*JPB(A,10)+4.69*JPB(A,11)
     *+4.69*JPB(A,12)+50.0*JPB(A,13)
        END IF
        IF(DUR.GT.48.AND.DUR.LE.96) THEN DO
          DUD=(DUR+168)*0.0279107*(JPB(A,1)+JPB(A,2))+(DUR*0.25
     *+14.0)*JPB(A,3)+(DUR*0.3125+6.0)*JPB(A,4)+(DUR*0.3792-2.7)
     **JPB(A,5)+(DUR*0.4313-10.4)*JPB(A,6)+(DUR*0.5014-19.5)*
     *JPB(A,7)+(DUR*0.5729-27.5)*JPB(A,8)+(DUR*0.5313-25.5)*
     *JPB(A,9)+(DUR*0.4896-23.5)*JPB(A,10)+(DUR*0.4479-21.5)*
     *JPB(A,11)+(DUR*0.4479-21.5)*JPB(A,12)+50.0*JPB(A,13)
        END IF
        IF(DUR.GT.96) THEN DO
          DUR3=DUR*0.2361+15.3
          DUR4=DUR*0.0381+34.67
          IF(A.LE.2) THEN DO
            IF(DUR3.GT.42.11) THEN DO
               DUR3=42.11
            END IF
            IF(DUR4.GT.42.11) THEN DO
               DUR4=42.11
            END IF
          END IF
          DUD=(DUR+168)*0.0279107*(JPB(A,1)+JPB(A,2))+DUR3*JPB(A,3)+
     *DUR4*JPB(A,4)+33.7*JPB(A,5)+31.0*JPB(A,6)+29.5*JPB(A,7)+
     *27.5*JPB(A,8)+25.5*JPB(A,9)+23.5*JPB(A,10)+21.5*JPB(A,11)+
     *21.5*JPB(A,12)+50.0*JPB(A,13)
        END IF
        A=A+1
C       DUDB=DUDB+DUD
        DUDB=DUDB+(DUD*0.00)
      END WHILE
```

```
      END IF
        DUDB=1.0-(DUDB*0.01)
C       PRINT, 'DUDB=  ',DUDB
      RETURN
      END
C ************************************************************
      REAL FUNCTION DUDF(DUR,JPF)
      INTEGER A,C
      REAL JPF(5,13),DUR,DUR3,DUR4,DUD
      A=1
      DUDF=0.0
      IF(DUR.NE.0.0) THEN DO
      WHILE(A.LE.5) DO
        IF(DUR.LE.48) THEN DO
          DUD=(DUR+168)*0.0577071*(JPF(A,1)+JPF(A,2))+
     *DUR*0.54167*JPF(A,3)+DUR*0.4375*JPF(A,4)+DUR*0.3229*
     *JPF(A,5)+DUR*0.2146*JPF(A,6)+DUR*0.1042*JPF(A,7)+
     *9.69*JPF(A,8)+9.69*JPF(A,9)+9.69*JPF(A,10)+9.69*JPF(A,11)
     *+9.69*JPF(A,12)+50.0*JPF(A,13)
        END IF
        IF(DUR.GT.48.AND.DUR.LE.96) THEN DO
          DUD=(DUR+168)*0.0577071*(JPF(A,1)+JPF(A,2))+(DUR*0.25
     *+14.0)*JPF(A,3)+(DUR*0.3125+6.0)*JPF(A,4)+(DUR*0.3792-2.7)
     **JPF(A,5)+(DUR*0.4313-10.4)*JPF(A,6)+(DUR*0.5014-19.5)*
     *JPF(A,7)+(DUR*0.5729-27.5)*JPF(A,8)+(DUR*0.5313-25.5)*
     *JPF(A,9)+(DUR*0.4896-23.5)*JPF(A,10)+(DUR*0.4479-21.5)*
     *JPF(A,11)+(DUR*0.4479-21.5)*JPF(A,12)+50.0*JPF(A,13)
        END IF
        IF(DUR.GT.96) THEN DO
          DUR3=DUR*0.2361+15.3
          DUR4=DUR*0.0381+34.67
          IF(A.LE.2) THEN DO
            IF(DUR3.GT.56.44) THEN DO
               DUR3=56.44
            END IF
            IF(DUR4.GT.56.44) THEN DO
               DUR4=56.44
            END IF
          END IF
          DUD=(DUR+168)*0.0577071*(JPF(A,1)+JPF(A,2))+DUR3*JPF(A,3)+
     *DUR4*JPF(A,4)+33.7*JPF(A,5)+31.0*JPF(A,6)+29.5*JPF(A,7)+
     *27.5*JPF(A,8)+25.5*JPF(A,9)+23.5*JPF(A,10)+21.5*JPF(A,11)+
     *21.5*JPF(A,12)+50.0*JPF(A,13)
        END IF
        A=A+1
C       DUDF=DUDF+DUD
        DUDF=DUDF+(DUD*0.0)
      END WHILE
      END IF
        DUDF=1.0-(DUDF*0.01)
C       PRINT, 'DUDF=  ',DUDF
      RETURN
      END
C ************************************************************
      REAL FUNCTION DUDA(DUR)
```

```
      REAL DUR,DAM
      DAM=1440/(1608+DUR)
C     DUDA=(0.6*DAM*173.84)+(1.0-(0.6*DAM)*86.92)
C     DUDA=(0.6*173.84)+(0.4*86.92)
      DUDA=173.84
C     PRINT, 'DUDA=   ',DUDA
      RETURN
      END
C *******************************************************
      SUBROUTINE SORT1(LARGE,Z,POSN,C)
      REAL LARGE,Z(4)
      INTEGER C,POSN,SUB
      SUB=1
      POSN=1
      LARGE=Z(1)
      WHILE(SUB.LE.C) DO
        IF(LARGE.LE.Z(SUB)) THEN DO
          LARGE=Z(SUB)
          POSN=SUB
        END IF
      SUB=SUB+1
      END WHILE
      RETURN
      END
      SUBROUTINE SORT2(LARGE,Z,POSN,C)
      REAL LARGE,Z(6)
      INTEGER C,POSN,SUB
      SUB=1
      POSN=1
      LARGE=Z(1)
      WHILE(SUB.LE.C) DO
        IF(LARGE.LE.Z(SUB)) THEN DO
          LARGE=Z(SUB)
          POSN=SUB
        END IF
      SUB=SUB+1
      END WHILE
      RETURN
      END
C ################################################################
C                    DATA
C ################################################################
$ENTRY
 .0285004 .0336563 .0000000 .0000000 .0000000 .0000000 .0000000
 .0000000 .0000000 .0000000 .0000000 .0000000 .0000000
 .0695389 .0976823 .0373104 .0289277 .0176826 .0044563 .0000000
 .0000000 .0000000 .0000000 .0000000 .0000000 .0000000
 .0000000 .0059161 .0084411 .0168238 .0276016 .0377611 .0362881
 .0274965 .0158425 .0030947 .0000000 .0000000 .0000000
 .0000000 .0000000 .0000000 .0000000 .0004673 .0035342 .0094634
 .0182550 .0299091 .0426569 .0457224 .0194075 .1262627
 .0000000 .0000000 .0000000 .0000000 .0000000 .0000000 .0000000
 .0000000 .0000000 .0000000 .0000292 .0002003 .2070709
 .0585358 .0419448 .0000000 .0000000 .0000000 .0000000 .0000000
 .0000000 .0000000 .0000000 .0000000 .0000000 .0000000
```

```
.0395033 .0812108 .0341948 .0274256 .0191028 .0092264 .0000000
.0000000 .0000000 .0000000 .0000000 .0000000 .0000000
.0000000 .0140991 .0115568 .0175492 .0235416 .0295340 .0333229
.0263317 .0177870 .0076887 .0000000 .0000000 .0000000
.0000000 .0000000 .0000000 .0000768 .0031072 .0069912 .0124287
.0194198 .0279646 .0378092 .0438334 .0406152 .0898560
.0000000 .0000000 .0000000 .0000000 .0000000 .0000000 .0000000
.0000000 .0000000 .0002536 .0019182 .0051364 .2173334
.0519031 .0000000 .0000000 .0000000 .0000000 .0000000 .0000000
.0000000 .0000000 .0000000 .0000000 .0000000 .0000000
.0461360 .0941581 .0072794 .0009626 .0000000 .0000000 .0000000
.0000000 .0000000 .0000000 .0000000 .0000000 .0000000
.0000000 .0430967 .0384121 .0409388 .0322156 .0173862 .0060160
.0005414 .0000000 .0000000 .0000000 .0000000 .0000000
.0000000 .0000000 .0000597 .0038502 .0135359 .0283654 .0397356
.0452101 .0457516 .0442476 .0370886 .0240339 .0516504
.0000000 .0000000 .0000000 .0000000 .0000000 .0000000 .0000000
.0000000 .0000000 .0000688 .0086630 .0217177 .2555393
000000.0 000000.0 000000.0 000000.0 000000.0 000000.0
000000.0  50000.0 100000.0 150000.0 200000.0 239250.0
000000.0  20000.0  40000.0  60000.0  80000.0  88150.0
000000.0   8000.0  16000.0  24000.0  32000.0  40000.0
000000.0  25000.0  50000.0  75000.0 100000.0 117800.0
000000.0  12000.0  24000.0  36000.0  48000.0  60000.0
000000.0  35000.0  70000.0 105000.0 140000.0 175175.0
000000.0  70000.0 140000.0 210000.0 280000.0 345950.0
000000.0  40000.0  80000.0 120000.0 160000.0 206790.0
000000.0  18000.0  36000.0  54000.0  72000.0  91200.0
000000.0  20000.0  40000.0  60000.0  80000.0 103845.0
000000.0 112000.0 224000.0 336000.0 448000.0 560000.0
000000.0 110000.0 220000.0 330000.0 440000.0 548000.0
000000.0  25000.0  50000.0  75000.0 100000.0 121900.0
000000.0  55000.0 110000.0 165000.0 220000.0 276000.0
000000.0 110000.0 220000.0 330000.0 440000.0 529000.0
000000.0  87000.0 174000.0 261000.0 348000.0 437000.0
000000.0 000000.0 000000.0 000000.0 000000.0 000000.0
0.5785 0.4675 0.2930 0.1980 0.2110
0.0225 0.0215 0.0105 0.0065 0.0055
0.0055 0.0060 0.0035 0.0025 0.0020
0.0020 0.0025 0.0015 0.0015 0.0006
0.0010 0.0010 0.0010 0.0010 0.0002
0.0005 0.0015 0.0005 0.0005 0.0007
    0.0      0.0      0.0      0.0      0.0      0.0
    0.0      0.0      0.0      0.0      0.0      0.0
    0.0      0.0      0.0      0.0      0.0      0.0
    0.0   2100.0   2600.0   3000.0   3500.0   4000.0
    0.0    730.0    980.0   1390.0   2360.0   3050.0
    0.0    410.0   1430.0   1650.0   1870.0   2100.0
    0.0    940.0   1680.0   2310.0   2800.0   3300.0
    0.0   1600.0   2300.0   3000.0   3600.0   4200.0
    0.0   2800.0   5100.0   7000.0   8400.0  10000.0
    0.0   1500.0   5100.0   4800.0   5800.0   6800.0
    0.0     60.0     90.0    120.0    160.0    200.0
    0.0   4500.0   6700.0   8700.0  11000.0  12600.0
    0.0   4800.0   7400.0   9700.0  11800.0  13800.0
```

```
    0.0  2500.0  3700.0  4600.0  5600.0  6400.0
    0.0  3800.0  5700.0  7300.0  8800.0 10300.0
    0.0  4000.0  6000.0  7200.0  8300.0  9100.0
    0.0  3900.0  5600.0  7300.0  8900.0 10300.0
    0.0     0.0     0.0     0.0     0.0     0.0
 10.00  10.80  34.00  17.87  0.000  0.00
 12.00  20.00  41.00  17.87  00.00  0.00
 16.00  32.00  52.00  17.87  00.00  0.00
 19.00  45.00  61.00  17.87  00.00  0.00
 22.00  55.00  70.00  17.87  00.00  0.00
 101.36    5
0.0   750.0   4410.0    40000.0  10000000.0
 0.00080       0.0
 0.0005259     0.206557
 0.0000697     2.22127
 0.0000697     2.22127
0.91   0.90   0.84
 122.50    6
0.0   158.0   6760.0   25260.0   69120.0  10000000.0
 0.0019937     0.0
 0.0004643     0.24165
 0.0002737     1.5298572
 0.0001787     3.96522
 0.0001487     6.040034
0.99   1.00   0.97
  83.89    5
0.0   700.0   6900.0    20825.0  10000000.0
 0.00100       0.0
 0.0003323     0.4674
 0.0002291     1.1793187
 0.0001554     2.733887
0.99   1.00   0.97
  81.67    5
0.0  5600.0  27300.0    88920.0  10000000.0
 0.00050       0.0
 0.0002903     1.11741
 0.0001723     4.3772160
 0.0001400     7.296460
0.99   1.00   0.97
  81.67    6
0.0   660.0  19140.0    77550.0  251250.0  10000000.0
 0.00020       0.0
 0.0004464     1.2536
 0.0002787     4.2353120
 0.0001400    14.9564
 0.0001336    16.688096
0.99   1.00   0.97
  81.67    5
0.0  1875.0  47880.0    97500.0  10000000.0
 0.00200       0.0
 0.0004389     2.927128
 0.0003035     9.4081020
 0.0002740    12.298793
0.99   1.00   0.97
  81.67    4
```

```
0.0  18480.0    26880.0  10000000.0
 0.0003333    0.0
 0.0008667   -9.856
 0.00035821    3.8113440
0.94   0.95   0.95
  81.67    5
0.0  4218.0  50200.0    84480.0  10000000.0
 0.001998     0.0
 0.0003624    6.9063
 0.0000893   20.6189040
 0.0000780   21.57260
0.91   0.90   0.84
  81.67    6
0.0  1870.0  44880.0  158790.0  470250.0  10000000.0
 0.0020000    0.0
 0.0004348    2.927
 0.0002677   10.4270
 0.0001650   26.2384
 0.0000836   65.20335
0.90   0.90   0.84
  81.67    5
0.0  1260.0  56280.0   241500.0  10000000.0
 0.00200      0.0
 0.0004656    1.933283
 0.0002827   12.230160
 0.0001840   35.98206
0.91   0.90   0.84
  81.67    5
0.0  2850.0  15340.0    59220.0  10000000.0
 0.00200      0.0
 0.0007718    3.500321
 0.0005501    6.9009030
 0.0003430   19.195167
0.91   0.90   0.84
  81.67    5
0.0   100.0   8798.0    70350.0  10000000.0
 0.00400      0.0
 0.0006283    0.3372
 0.0003619    2.6811000
 0.0001987   14.162366
0.91   0.90   0.84
  81.67    4
0.0   33810.0   178450.0  10000000.0
 0.000667     0.0
 0.0003377   11.123517
 0.0001736   40.3936900
0.86   0.84   0.81
  81.67    5
0.0   6600.0  87750.0  309400.0  10000000.0
 0.001000     0.0
 0.0003512    4.2821
 0.0002400   13.9998830
 0.0001630   37.84366
0.86   0.84   0.81
  97.68    5
```

```
 0.0    6600.0   87750.0   309400.0   10000000.0
  0.001000      0.0
  0.0003512     4.2821
  0.0002400    13.998830
  0.0001630    37.84366
 0.86   0.84    0.81
/*
//
```