IMPROVING UPON A GIS-BASED SPATIALLY DISTRIBUTED RAINFALL-RUNOFF MODEL

by

James F. Emerick

B.S. in Civil Engineering, University of Pittsburgh, 2001

Submitted to the Graduate Faculty of

The School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science in Civil Engineering

University of Pittsburgh

2003

UNIVERSITY OF PITTSBURGH

SCHOOL OF ENGINEERING

This thesis was presented

by

James F. Emerick

It was defended on

02/28/2003

and approved by

Dr. C. L. Chiu, Professor, Civil and Environmental Engineering

Dr. T. K. Hung, Professor, Civil and Environmental Engineering

Dr. J. S. Lin, Professor, Civil and Environmental Engineering

Thesis advisor: Dr. R. G. Quimpo, Professor, Civil and Environmental Engineering

ABSTRACT

IMPROVING UPON A GIS-BASED SPATIALLY DISTRIBUTED RAINFALL-RUNOFF MODEL

James Emerick, M.S.

University of Pittsburgh, 2003

The modeling of small dendritic watersheds has always been a difficult undertaking. The lack of streamflow data and reliable rainfall data, and the inability to model distributed properties have all been reasons for this difficulty. With improvements in computer technology, advancements in radar rainfall measurement, and the innovations in Geographical Information Systems (GIS), it is possible to make improvements to existing modeling techniques. This, along with increasing environmental concerns, has sparked new interests in the development of watershed models.

This goal of this research has been to improve upon a Geographical Information System based rainfall-runoff model. A major improvement to the model has been the implementation of spatially distributed incremental rainfall data obtained from gage adjusted radar measurements. The significance of the radar data is the representation of highly defined spatial and temporal distribution in the rainfall measurements. The current model also makes use of distributed properties of the watershed such as elevations, slopes, soil types, land cover, and stream networks. The direct runoff is calculated using a distributed Curve Number Method. The hydrograph is then developed by the use of a distributed map of isochrones. The model was tested for a small watershed where streamflow data was available. The goal is to test and calibrate the model so that it can be used in watersheds where there is no streamflow data available.

DESCRIPTORS

Curve Number	DEM
Distributed	GIS
Hydrograph	IDRISI
Runoff	SCS
Watershed	

ACKNOWLEDGMENTS

I would like to especially thank Dr. Rafael G. Quimpo for his guidance and dedication to my research. His commitment to my learning was a continuous driving force in my research efforts. Also I would like to acknowledge Dr. Chao-Lin Chiu, Dr. Jeen-Shang Lin, and Dr. Tin-Kan Hung for serving on my advising committee.

I would like to thank Khalid Maseem Khan for his assistance and prior thesis research. I would also like to thank Timothy D. Prevost at the Three Rivers Wet Weather Demonstration Program for sharing its rainfall data.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

1.0 INTRODUCTION

Mankind has always been dependent on water as a requirement for survival. Throughout history the survival of civilizations depended greatly on the availability of a constantly replenished water source. For this reason many early civilizations developed along the banks of rivers and streams. Rivers and streams were a good source of clean water because of the constant flow. Rivers and streams were also used to carry away waste. Because of the dynamics of river systems there were two major concerns, those being flooding and draught. These concerns lead to an interest in understanding what caused changes in streamflow.

As far back as the 1st century observations were made by Vitruvious that the rainfall contributed to streamflow and that some of the precipitated water was infiltrated into the soil. It was not until the 17th century that attempts were made to prove by measurement that rainfall had an effect on streamflow by men such as Perrault, Mariotte, and Halley. The eighteenth century brought about great advancements in hydraulic theory and instrumentation by Bernoulli and Chezy, but it was not until the early twentieth century that a theoretical, quantitative approach to the relationship of rainfall and streamflow was established. Sherman (1932) [(1)] developed the unit hydrograph which is the still the basis for much of the hydrologic principles used today.

Hydrology can be defined as the scientific study of the occurrence, movement, properties, and distribution of the earth's water throughout all phases of the hydrologic cycle. The relationship between rainfall and runoff has been one of the biggest concerns in the field of hydrology. Many models have been developed to determine the resulting streamflow hydrograph due to a given rainfall event. The methods used to estimate a storm hydrograph usually required a long period of rainfall and streamflow record for the area of study in order to develop a reliable model. This becomes problematic in most small watersheds that have little or no record of streamflow and rainfall. The following model is a rainfall-runoff model that was developed to meet that particular need. Using remote sensing and GIS tools a spatially distributed model was developed to predict runoff due to a rainfall event.

2.0 BACKGROUND AND LITERATURE REVIEW

2.1 Hydrologic Cycle

The hydrologic cycle is the path through which earth's water circulates within the lithosphere. Water is evaporated from the earth's surfaces and condenses as it rises. As the evaporated water further condenses it forms droplets and falls as precipitation over land and water bodies. The precipitation over the land goes through several processes before a portion of that water becomes surface runoff. Some of the precipitation over the land is infiltrated into the soil. This infiltrated water may become subsurface flow eventually feeding into streams or the water may percolate deeper into the ground to recharge the ground water. Yet another portion of the precipitation can be intercepted by vegetation or evaporated. The remaining water can then become ponded. When the water has exceeded the ponding capacity it then become runoff as overland flow. This water is considered excess precipitation or direct runoff. The overland flow moves along the steepest path and eventually combines with flow from other paths to form channel flow in rivers and streams. The water in the rivers and streams flows into larger water bodies such as lakes and oceans where evaporation occurs at a considerable rate as the hydrologic cycle continues.

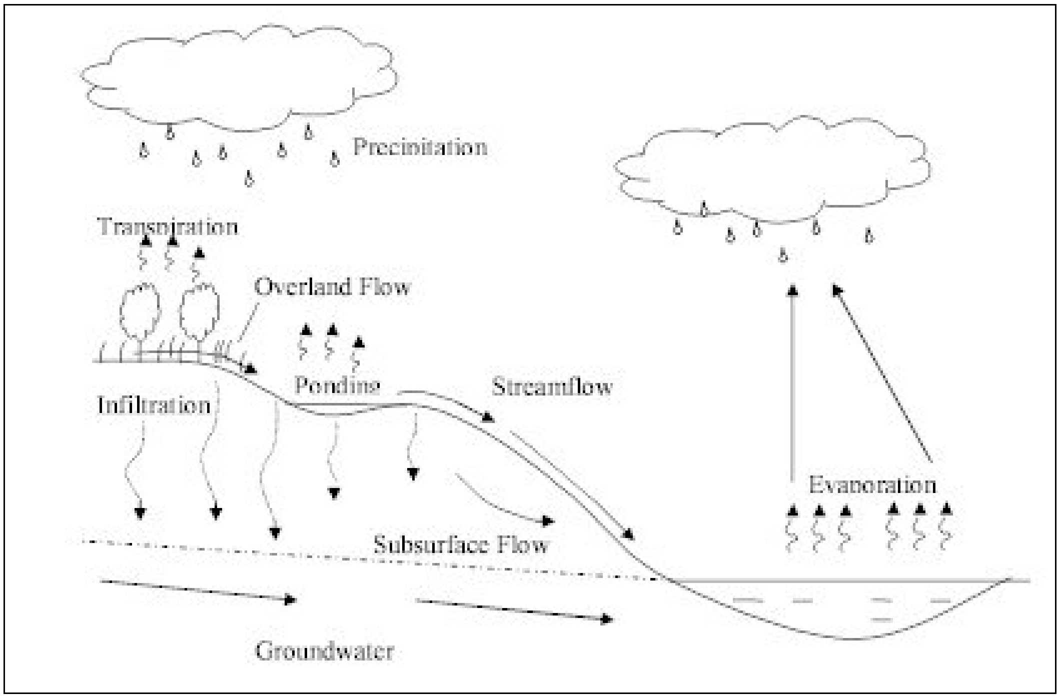


Figure 2-1 Hydrologic Cycle

2.2 Surface Water Modeling

The prediction of the quantity of water within a particular component of the hydrologic cycle as it is transferred from another component remains a difficult task for the hydrologist. Although surface water is a very small component of the hydrologic cycle it has been one of the main concerns of hydrologists. There are several reasons for the interest in this part of the hydrologic cycle. Surface water is responsible for much of the world's water supply as well as being responsible for changes in the Earth's topography, transportation of pollutants, and destructive flooding. Hydrology began with an understanding that precipitation has an effect on streamflow, but because there are so many variables involved in the transformation of rainfall to runoff it has been difficult to model quantitatively.

Before the evolution of computer technology, models had to be simplified in order to maintain a reasonable number of the hand calculations. With the advent of computers, models became more representative of the processes involved in the rainfall-runoff transformation. This was because the increase in processing speed allowed for many more calculations to be performed in a shorter amount of time in comparison to the limited amount of calculations that could previously be done by hand. The hydrologic models will continue to evolve with advancements in computer technology. Software development in GIS, increases in computer processor speed and data storage, and advancements in remote sensing and meteorological data collection make it necessary for modelers to constantly improve upon computer simulated rainfall-runoff models. The improvement of these models is a continuous process. With all of the advancements made there are still some assumptions that need to be considered in order to make the modeling process feasible.

2.3 System Approach Modeling

Natural phenomena are often very complex and it can be unrealistic and even impossible to attempt to model in detail all of the physical processes involved. This is especially true when dealing with a large scale, complex system as the rainfall-runoff process. A systems approach is often applied in such cases. The system is defined in space by conceptual boundaries which make up a control volume. The control volume is chosen such that the laws of conservation are applicable to the type of input and output of the system. The system concept considers the output of the system to be a function of an input being applied. The difference of the input (*I(t)*) and output (*O(t)*) is equal to the change in storage of the system per time ($\frac{dS}{dt}$).

$$I(t) - O(t) = \frac{dS}{dt} \tag{2-1}$$

The goal of the system concept is not to model all of the processes occurring within the control volume in detail, but rather determine a function that replicates the overall transformation of the input to the output.

2.3.1 Linear and Nonlinear Systems

Non-linear systems are much more difficult to model than linear systems. Most natural phenomena do not truly behave linearly, so it is sometimes necessary to make this assumption to allow for a natural event to be represented by a linear model making the modeling process much simpler. A system is assumed to be linear when the system's output is directly proportional to the input and the laws of superposition are applicable.

$$Y=f(X) \text{ then } 2Y=f(2X) \tag{2-2}$$

This allows for the use of a unit impulse in order to determine the response function of the system. The response can then be scaled and combined using superposition.

2.3.2 Time Variant and Time Invariant Systems

Time invariance is another assumption commonly used in system modeling to simplify the model. Assuming a system is time invariant means that the transformation function is independent of time. Therefore an output $O(t)$ due to an input $I(t)$ is equal to an output $O(t+T)$ due to an input $I(t+T)$.

2.3.3 Systems Approach Applied to Hydrology

In surface water hydrologic modeling the control volume is defined by the watershed.

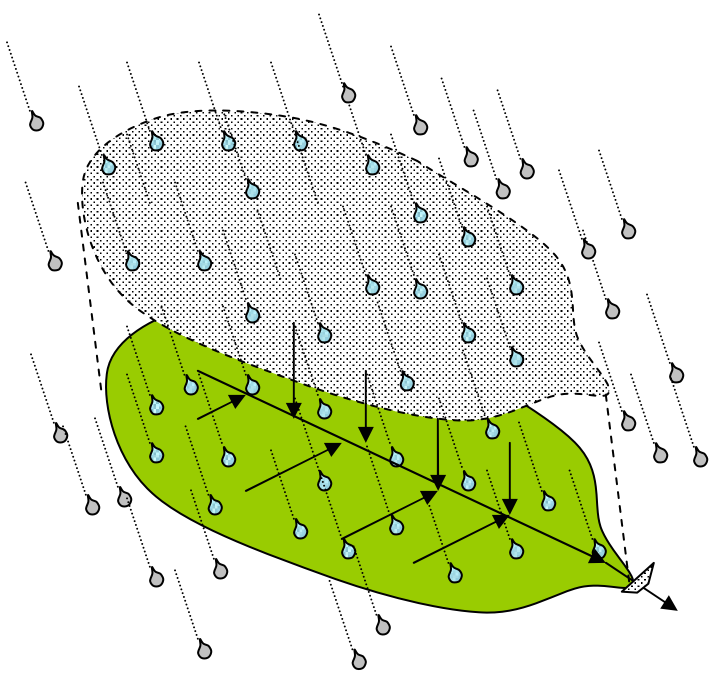


Figure 2-2 Watershed Control Volume

A watershed is an area defined such that all of the surface water drains to a single outlet. In viewing the watershed as a system the input to the system would be the rainfall intensity as a function of time. The system's output would then be the resulting flow through the outlet of the watershed as a function time. The objective is to determine a response function relating the streamflow (Q) to the rainfall (I).

$$Q = f(I) \tag{2-3}$$

The rainfall-runoff process is a very complex system. There are several ways to simplify the model. One way to do this is to process the data so that what is being modeled is only the direct effect of the rainfall on the streamflow during the time period being considered. It is also often necessary to break a complex system into several subsystems. In the modeling of systems with variable input it is sometimes necessary to begin with a simplified input in order to determine the response of the system. If assumptions such as linearity and time invariance are reasonable, a simple unit input can be used to determine the response function. A response function to a unit input can be developed, studied, and then applied to more complex cases assuming laws of superposition. In hydrology the unit hydrograph is the unit response function.

2.4 The Unit Hydrograph Model

One of the most commonly used system approaches in hydrology is the use unit hydrograph. The unit hydrograph can be defined as a response function for a specific watershed due to an input of a unit depth of excess precipitation distributed uniformly over the entire watershed in space and time. The unit hydrograph is typically described by the time period in which the rainfall occurred. A 4-hour unit hydrograph would represent the watershed's response to a 4-hour storm with an excess precipitation intensity of ¼ inch per hour resulting in a storm total excess precipitation of 1 inch. The concepts derived from unit hydrograph theory have been, and continue to be used in many applications in hydrology.

Assumptions of Unit Hydrograph Theory

1. Uniform Depth of Excess Precipitation
2. Uniform Intensity
3. Time Invariance and Linearity

These assumptions are almost never completely true in an actual event, but they can be reasonable in certain situations. The data from the watershed must be carefully selected when deriving the unit hydrograph in order for these assumptions to be valid. It is well known that over large areas there is typically a great deal of variability in the depth of precipitation during an event. For small areas there is usually less variability in the rainfall distribution, therefore it may be necessary to subdivide large watersheds into sub-watersheds in order to make the first assumption valid. The assumption of a constant intensity is also a consideration when choosing data from which to develop the unit hydrograph. Storms with long durations tend to have largely varying intensities. Data from a storm with a short duration are fairly consistent with the assumption of constant intensity. Another consideration is that of linearity and time invariance. Unit hydrograph theory is not applicable for watersheds with considerable storage effects such as detention ponds, reservoirs, and flood plains. All of these things must be considered when choosing data from which to develop a unit hydrograph.

2.5 Synthetic Unit Hydrographs

A synthetic unit hydrograph is a unit hydrograph developed from data that are ether from another location in the same watershed or from another watershed with similar characteristics. Synthetic unit hydrographs are often useful in small watersheds since small watersheds are often not gaged or do not have a long enough period of record to develop a reliable unit hydrograph.

2.5.1 Snyder Synthetic Unit Hydrograph

Snyder (1938)[2] developed a synthetic unit hydrograph approach based on empirical data from watersheds located in the Appalachian highlands in the United States ranging in size from 10 to 10,000mi^2. The method uses the following equations to develop the unit hydrograph,

$$t_l = C_t (LL_c)^{0.3} \tag{2-4}$$

t_l = time of the centroid of the rainfall to the peak of the unit hydrograph in hours

C_t = a constant based on the slope and storage, ranges from 1.8 to 2.2

L = length of the mainstream in miles

L_c = length along the mainstream from a point nearest the centroid to the outlet in miles

$$Q_p = 640\frac{C_p A}{t_l} \tag{2-5}$$

Q_p = peak discharge in cubic feet per second

C_p = coefficient based on the retention and storage, ranges from 0.4 to 0.8

A = the area in square miles

$$t_r = \frac{t_l}{5.5} \tag{2-6}$$

t_r = the duration of effective rainfall

$$T = 3 + \frac{t_l}{8} \tag{2-7}$$

T = base time

To develop hydrographs of different duration the following equation is used,

$$t_{lR} = t_l + 0.25(t_R - t_r) \tag{2-8}$$

t_{lR} = the adjusted time to peak in hours

t_R = the desired rainfall duration in hours

2.5.2 SCS Synthetic Unit Hydrograph

The Soil Conservation Service (SCS), now the Natural Resources Conservation service, developed a procedure for estimating the unit hydrograph using a dimensionless unit hydrograph based on empirical data from watersheds throughout the US. The method suggests that the parameters q_p and T_p can be estimated from the following formulas based on a triangular hydrograph as seen in Figure 2-3.

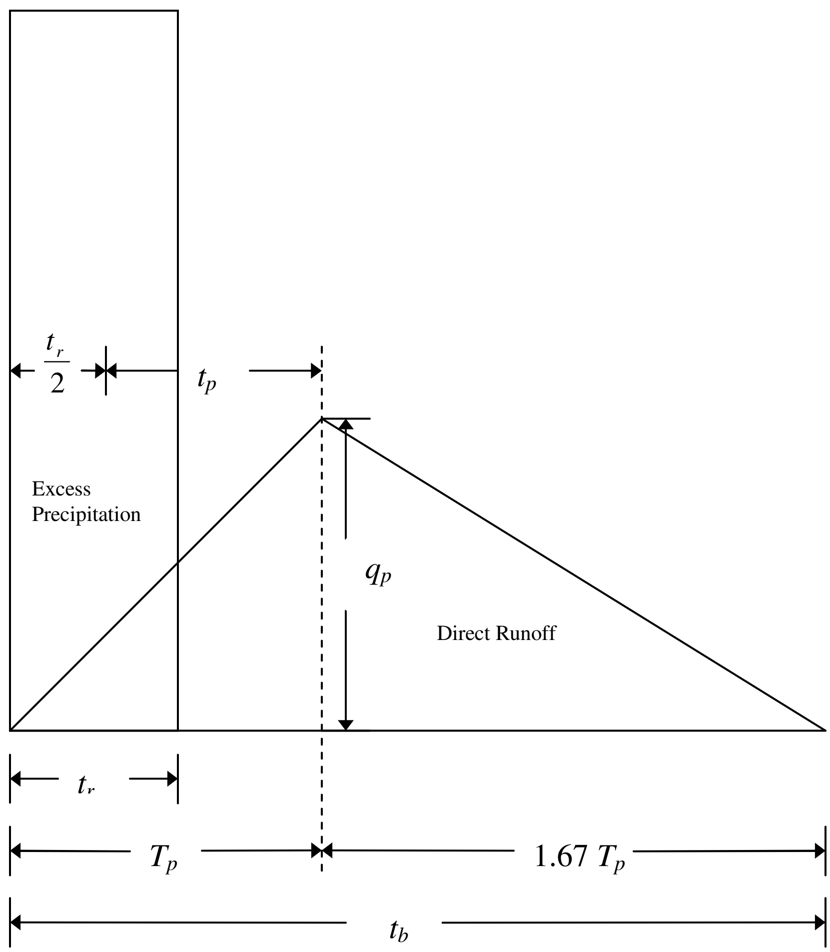


Figure 2-3 SCS Triangular Hydrograph

$$T_p = \frac{t_r}{2} + t_p \tag{2-9}$$

T_p = time to peak

t_p = time from the centroid of rainfall to the peak of the hydrograph approximately equal to $0.6T_c$

t_r = duration of effective rainfall

$$T_r = 1.67T_p \tag{2-10}$$

$$T_b = T_p + T_r = 2.67T_p \tag{2-11}$$

T_r = the recession time

T_b = the base time

$$q_p = \frac{CA}{T_p} \tag{2-12}$$

The following empirical formula was developed by the SCS for the calculation of the time of concentration,

$$T_c = \frac{100L^{0.8}[(1000/CN) - 9]^{0.7}}{1900S^{0.5}} \tag{2-13}$$

L = the longest flow path in ft

CN = the composite curve number

S = average slope in %

These values calculated for q_p and T_p are then multiplied by ordinates of the dimensionless unit hydrograph (t/T_p , q/q_p) to obtain the ordinates of the synthetic unit hydrograph. The dimensionless unit hydrograph can be seen in Figure 2-4.

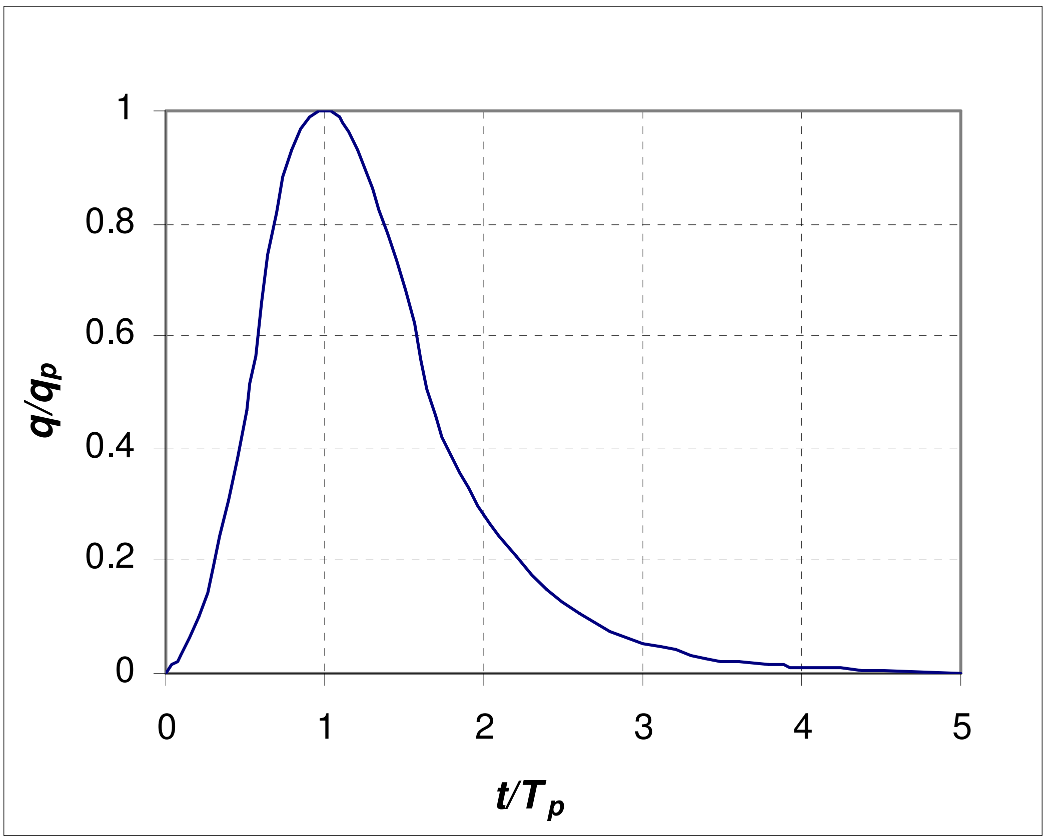


Figure 2-4 Dimensionless Unit Hydrograph

2.5.3 Clark Unit Hydrograph

The unit hydrograph method proposed by Clark (1945) [3] is based in a theory that the watershed storage and diffusion can be represented by routing the time area curve occurring at the outlet of the watershed through a linear reservoir. The travel time for areas within the watershed must first be determined using physical characteristics of the watershed such as length, slope, and roughness for the overland and channel flow. Several methods have been introduced to calculate the travel times. The time area curve is then developed by assuming an excess precipitation of 1 inch evenly distributed over the entire watershed. Linear routing is then applied to the time area curve in order simulate the storage effects of the watershed. The value of the storage constant must be determined. It is often stated that the storage constant is equal to the time of concentration (T_c). The resulting hydrograph is an instantaneous unit hydrograph for the particular watershed. The IUH can then be used to determine a unit hydrograph of any duration by applying the convolution equation as explained in section 2.5.5.

2.5.4 Nash Unit Hydrograph

Nash (1957) [4] developed a unit hydrograph method that is very similar to the Clark method but rather than routing through a single reservoir, the time area graph is routed through several identical reservoirs each providing diffusion and storage. The identical reservoirs are assumed to have the same value of storage constant.

2.5.5 Instantaneous Unit Hydrographs

The instantaneous unit hydrograph, sometimes called an impulse response function, is a hydrograph that results from an instantaneous application of one inch of excess precipitation evenly distributed over the enter watershed. Although the IUH is not physically possible, it can be very useful in many applications using unit hydrograph theory. The IUH can be used to determine a hydrograph for any given time period. This is done by the use of the convolution equation. By defining the step response function or S-curve as an infinite summation of unit hydrographs of a given duration it can than be represented by the following integration of the impulse response function.

$$g(t) = \int_0^t u(t-\tau)d\tau \qquad (2\text{-}14)$$

This also implies,

$$u(t-\tau) = \frac{d}{dt}g(t) \qquad (2\text{-}15)$$

The T-hour unit hydrograph can be represented by,

$$h(t) = \frac{1}{T}\left[g(t) - g(t-T)\right] \quad \text{where } T = \Delta t \qquad (2\text{-}16)$$

The impulse response function can be estimated using the finite difference method.

The backward difference solution:

$$u(t) = \frac{d}{dt}g(t) = \frac{g(t) - g\left(t - \frac{T}{2}\right)}{\frac{T}{2}} \tag{2-17}$$

The forward difference solution:

$$u(t-T) = \frac{d}{dt}g(t) = \frac{g\left(t - \frac{T}{2}\right) - g(t-T)}{\frac{T}{2}} \tag{2-18}$$

If the average of the forward and backward difference is taken the result is:

$$\frac{u(t) + u(t-T)}{2} = \frac{1}{T}\left[g(t) - g(t-T)\right] \tag{2-19}$$

Therefore,

$$h(t) = \frac{u(t) + u(t-T)}{2} \tag{2-20}$$

This implies that to obtain a T-hour unit hydrograph the ordinates of the impulse response function at the beginning and end of the time interval T are averaged to obtain the ordinate at the end of the time interval for T-hour unit hydrograph. Another way that this can be explained is by using two IUHs. One of the IUHs is lagged by the time T. The ordinates are then averaged to obtain the T-hour unit hydrograph.

2.6 The Direct Runoff Hydrograph and Baseflow Separation

A direct runoff hydrograph is a hydrograph that represents the flow directly resulting from the excess precipitation. This is a way of processing the data so that the changes in streamflow are due to the event being studied rather than effects caused by things such as ground water and subsurface flow which are usually effects caused by previous storms. The direct runoff hydrograph is often used in calibration of single event models. The direct runoff hydrograph can be determined by subtracting the baseflow from the streamflow hydrograph. This is usually done using a one of the three graphical methods.

2.6.1 Straight Line Method

The straight line method is the simplest method for separation of the runoff from the streamflow hydrograph. The straight line method is accomplished by drawing a line from the point on the hydrograph where the direct runoff begins to the point where the baseflow recession limb begins.

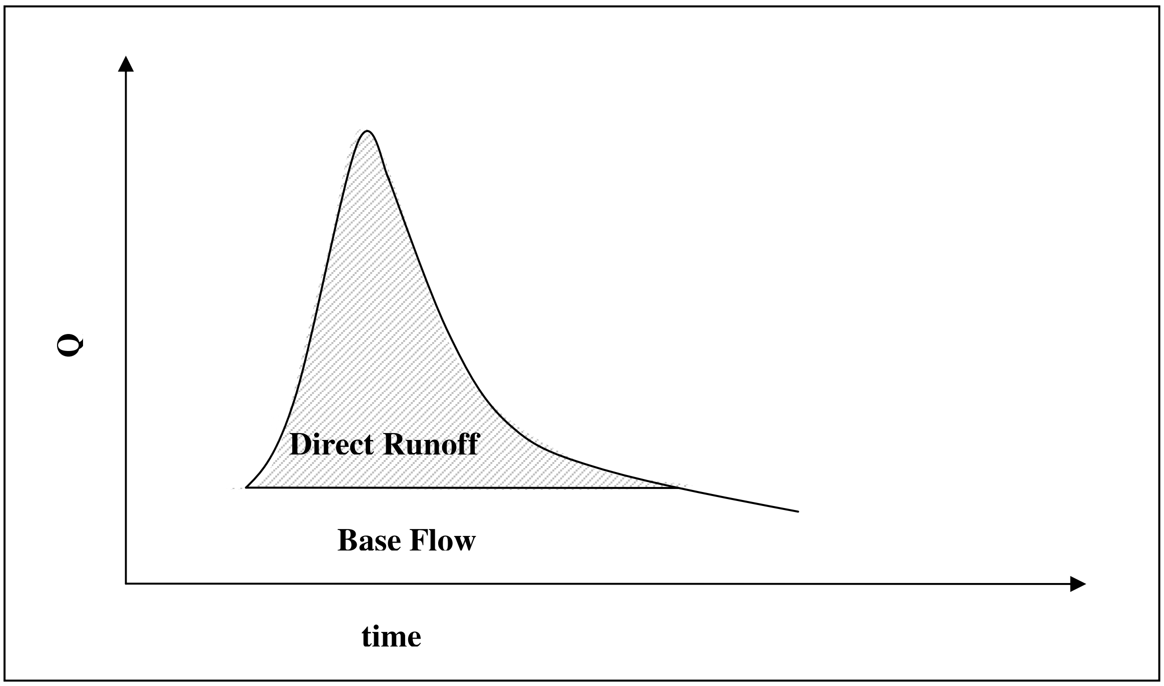


Figure 2-5 Straight Line Method

2.6.2 Fixed Base Method

The fixed base method assumes that the direct runoff ends at a fixed time after the hydrograph peak. In this technique the baseflow recession before the direct runoff is projected to the time of peak. A line is then drawn from that point to the point on the hydrograph at time T_p+N.

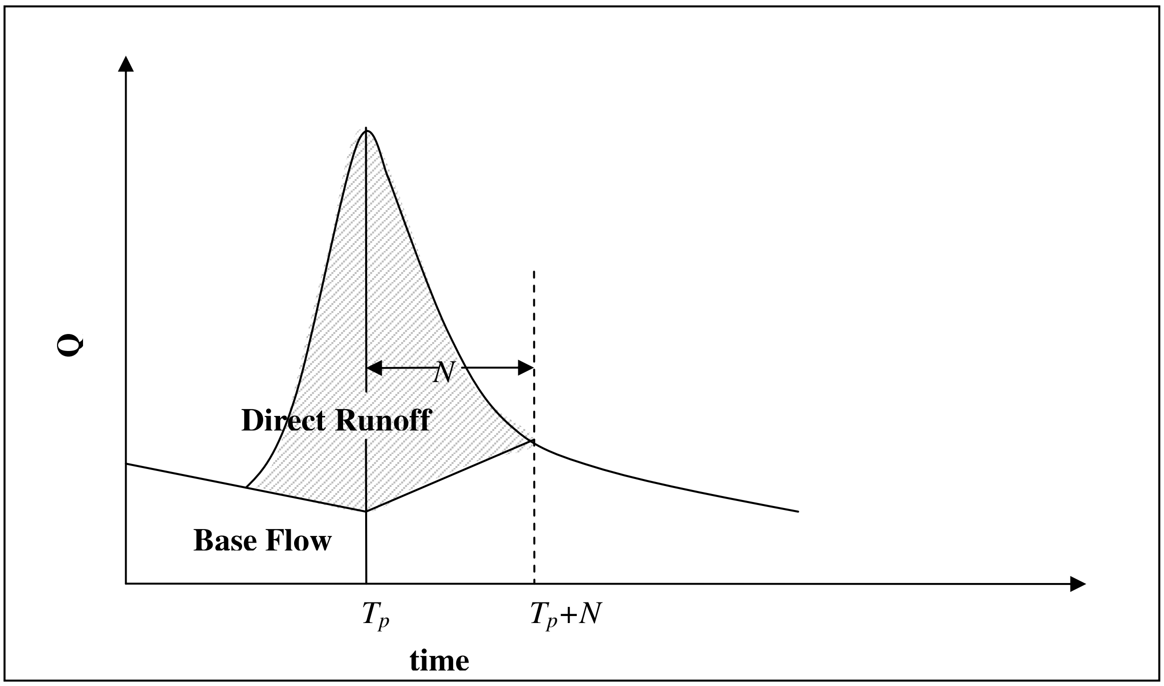


Figure 2-6 Fixed Base Method

2.6.3 Variable Slope Method

In the variable slope method the baseflow recession before the direct runoff is extrapolated forward to the T_p. The baseflow recession after the direct runoff is extrapolated back to the time at which the point of inflection on falling limb of the hydrograph occurs. The two lines are then connected.

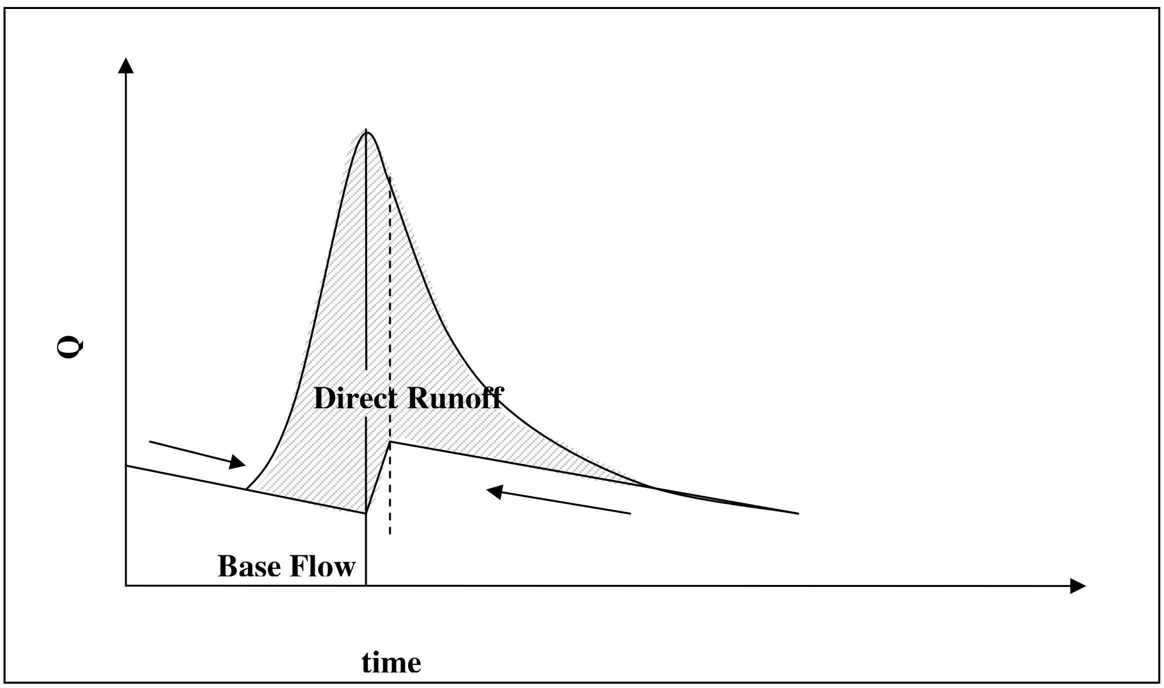


Figure 2-7 Straight Line Method

It should be noted that these methods are ways to obtain the *estimated* direct runoff. They are not exact and can be quite subjective. This should be a consideration when using the resulting data in the calibration of a model. The straight line method is usually the most practical method when using real data.

2.7 Single Event Modeling vs. Continuous Modeling

In modeling a single storm there are several components of the transformation of rainfall to streamflow that can be assumed insignificant and therefore may be excluded from the model. This is because the period of time considered in single event modeling is usually less than 24 hours. This is a relatively small time period for some of the components to have a considerable effect on the streamflow. Evaporation, transpiration, subsurface flow, and groundwater effects due to the rainfall for the event can be assumed insignificant in most single event modeling. In continuous modeling these are not valid assumptions. In long term modeling these effects are more considerable and it is unreasonable to exclude them. In single event modeling infiltration has the greatest effect on the amount of precipitation that becomes surface runoff.

2.8 Lumped vs. Distributed Modeling

Before the use of high speed computers in hydrologic modeling, the lumping of data was necessary in order to make the handling and storage of data more reasonable and limit the calculations performed in the modeling process. As computers processors become faster and larger storage also became available the trend in modeling has moved toward distributed modeling rather then lumped modeling.

In distributed modeling the watershed is broken into smaller areas by applying a grid. All of the cells in the grid are of equal size and represent an area that can be determined from the resolution of the grid. The common resolution is 30 meters. This is because this is a common resolution of the digital elevation model which is often used in distributed hydrologic models. The model described in this report began in many aspects a lumped model and has been improved by the introduction of spatial distribution in watershed characteristics as well as meteorological data. Spatial distribution is represented by the use of unique values for each cell in the grid.

2.9 Rainfall Runoff Process

Initially when rainfall occurs on the ground surface the water is infiltrated into the soil (assuming that the surface is not entirely impervious). The amount of water that can be infiltrated into the soil depends on several factors such as the porosity, hydraulic conductivity, and the existing moisture condition of the soil. Runoff does not occur until either the soil becomes completely saturated and can no longer retain water or the infiltration rate is exceeded by the intensity of the rainfall. The first case occurs with storms having a moderate intensity lasting over a considerable time period. The second case occurs with storms having a high intensity usually over a smaller time period.

2.10 Determination of Runoff from Rainfall

2.10.1 Rational Method

The Rational Method may be one of the simplest methods used to estimate the peak runoff due to a rainfall event. It is because of the simplicity of this method that it has been used extensively. It is assumed that the runoff is a linear function of the intensity, area, and the runoff coefficient.

$$Q_p = CiA \tag{2-21}$$

or

$$Q_p = i\sum_{j=1}^{m} C_j A_j \tag{2-22}$$

Q_p = the peak runoff

C = the runoff coefficient

i = the intensity of the rainfall

A = the area of the watershed

It should be noted that in order for this to be applicable the intensity must be constant and the storm duration must be longer than the time of concentration. If the storm duration does not exceed the time of concentration then at no time is the total area contributing to the outflow and therefore this equation is not relevant.

2.10.2 Runoff Coefficients

A runoff coefficient can be described as the ratio of the peak rate of direct runoff to the average intensity of a storm. Because rainfall intensity is often highly variable, a more practical definition for the runoff coefficient would be the ratio of the excess precipitation (P_e) to the total precipitation (P) for a given period.

$$C = \frac{P_e}{P} \tag{2-23}$$

The runoff coefficient can be used to estimate the excess precipitation for a given rainfall. This method requires a good record of rainfall and runoff. As mentioned previously the methods used to estimate the runoff from the streamflow data can be subjective. This should be considered when using the runoff coefficient.

2.10.3 Ø-index Method

The Ø-index method for abstractions assumes the abstractions to occur at a constant rate (Ø) throughout the entire storm. The method requires the rainfall hyetograph and the streamflow hydrograph for the particular event. The abstraction rate (Ø) can be determined from the data using the following equation.

$$r_d = \sum_{m=1}^{M} (R_m - \text{Ø}\Delta t) \tag{2-24}$$

R_m = observed rainfall contributing to runoff

r_d = total excess rainfall over the entire watershed (this is obtained from the streamflow record)

M = number of intervals of rainfall that contribute to the direct runoff

2.10.4 SCS Curve Number Method

The curve number method for abstractions is a widely used method for estimating the excess precipitation from rainfall. This method was developed by the Soil Conservation Service (now known as the Natural Resources Conservation Service) in 1972. The method assumes that the ratio of the actual water retained (F_a) to a maximum potential retention (S) is equal to the ratio of the excess rainfall (P_e) to the actual rainfall minus the initial abstractions ($P - I_a$).

$$\frac{F_a}{S} = \frac{P_e}{P - I_a} \tag{2-25}$$

The total rainfall (P) can be accounted for by summing the initial abstractions (I_a) the retention (F_a) and the excess rainfall (P_e).

$$P = P_e + I_a + F_a \tag{2-26}$$

This can be seen from the following graph.

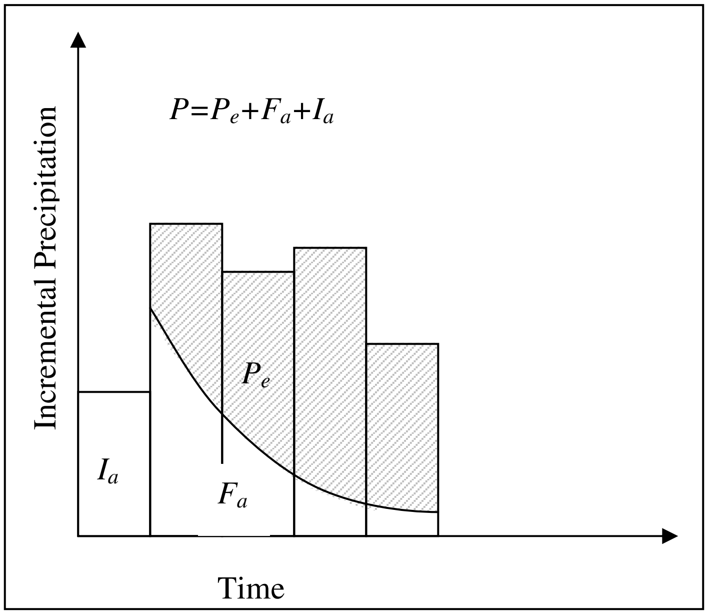


Figure 2-8 Hyetograph Showing Accounting of Precipitation

Therefore it can be said that,

$$F_a = (P - I_a) - P_e \quad (2\text{-}27)$$

The Soil Conservation Service (SCS) found using empirical data from experimentation involving small watersheds that the initial abstractions (I_a) is approximately equal to twenty percent of the maximum potential retention (S).

$$I_a \cong 0.2S \quad (2\text{-}28)$$

By replacing I_a and F_a the following formula can be derived for the excess precipitation.

$$P_e = \frac{(P - 0.2S)^2}{P + 0.8S} \qquad (2\text{-}29)$$

By plotting the precipitation versus the excess precipitation for many different watersheds the SCS developed a set of curves. It was found that these curves could be representative of the watershed retention characteristics. The SCS then defined a number called a curve number for each of these curves. The curve number is a number related to the maximum retention of the watershed based on the soil and land use type. The following relation was developed for the maximum retention and the curve number.

$$S = \frac{1000}{CN} - 10 \qquad (2\text{-}30)$$

The SCS published tables of curve numbers bases on the hydrologic soil groups and land use types. The curve numbers range from 0 to 100. The higher the curve number, the higher is the runoff potential for that area. An area that is totally impervious or an area such as a water body would have a curve number of 100. In accounting for the land use the curve number represents the imperviousness of an area. If the area has been classified as high density residential the area would consist of a great deal of impervious surfaces such as roofs and driveways. This would then be represented by a higher curve number and would therefore have the potential to produce a high runoff. In accounting for the soil type the curve number is representative of the potential infiltration given that the surface is not completely impervious. Given that the soil and land use data is reliable for a particular area the curve number can give a good representation of the potential runoff that can be generated.

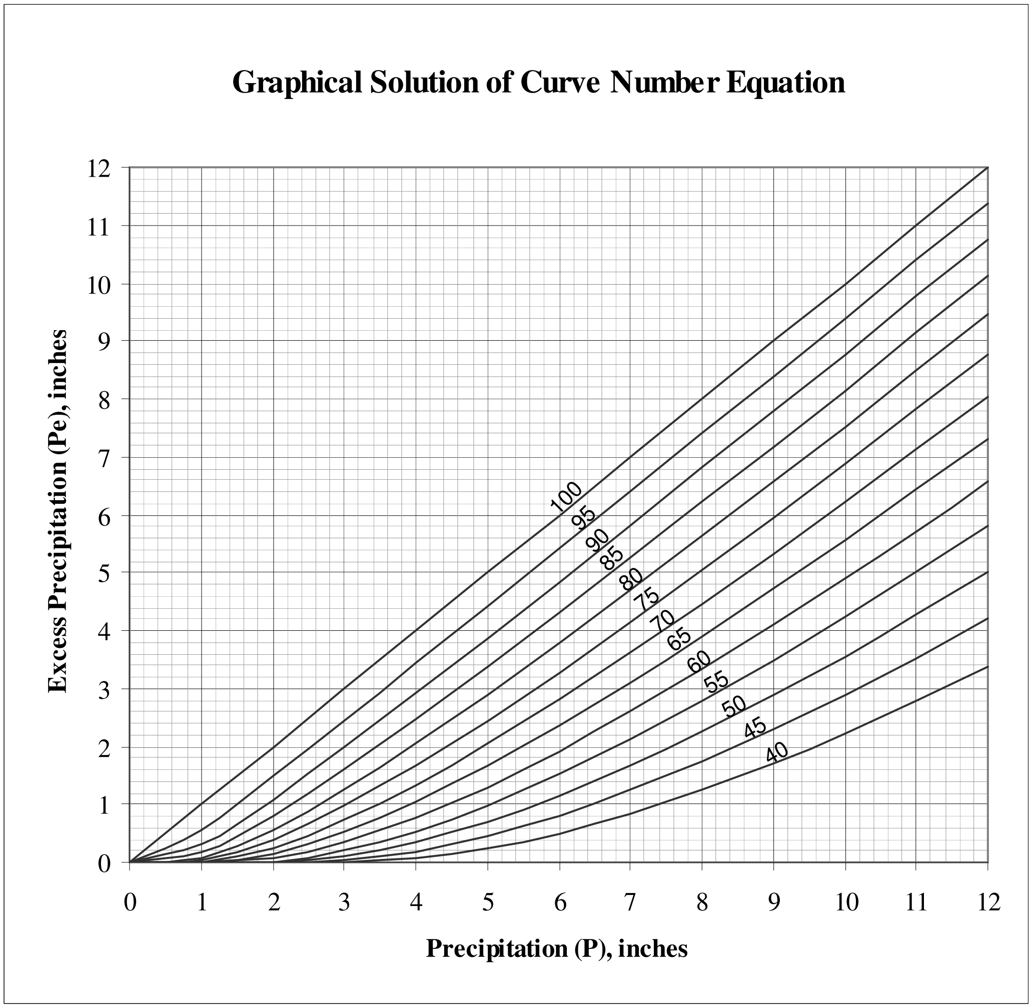


Figure 2-9 Solution of Curve Number Equations

2.10.5 Green-Ampt Method

The Green-Ampt method is a method used to determine the infiltration rate or the cumulative infiltration. This method is often used in single event modeling where infiltration is the only significant abstraction. It is a physical based model proposed by Green and Ampt (1911) [(5)]. The method considers a control volume, in which the conservation principles of continuity and momentum are applied to obtain the following equations for the infiltration,

$$F(t) = Kt + \psi\Delta\theta \ln\left(1 + \frac{F(t)}{\psi\Delta\theta}\right) \tag{2-31}$$

where,

$\Delta\theta = (1 - s_e)\theta_e$

F(t) = the cumulative depth of water infiltrated into the soil in cm

K = the permeability of the soil in cm/hr

θ_e = effective porosity

2.11 Rainfall Measurements

2.11.1 Raingage Data

Raingages have been used for the measurement of precipitation since the earliest attempts to quantify rainfall. Raingages still prove to be the most accurate method for measuring rainfall intensity and depth at a point where the gage is located. The one major setback with the measurement of precipitation with raingages is the lack of representation of the spatial distribution of a storm. Rainfall measurements become much less representative as the distance between the gage increases, therefore to obtain a sufficient representation of areal rainfall distribution for storms of non-uniform depth there must be a high density of rain gages. If more than one gage is available the point sources can be used to estimate an areal average. This can be done by a simple arithmetic average value or by a weighted average as in the Thiessen polygon and Isohyetal method. When modeling small watersheds it is rare to have even one raingage that falls inside of the watershed. Using a single point gage to represent the rainfall for an entire watershed can lead to highly inaccurate results.

2.11.2 Radar Data

Radar rainfall data has become more popular in hydrologic modeling with the shift towards spatially distributed models. Radar rainfall data can give high resolution in spatially and temporally distributed rainfall measurements as opposed to the point data from a raingage. The measurements can be made for up to 1 km^2 resolution of gridded data. Incremental measurements can be received in real time at a time interval as small as 5 minutes. The radar data can be significantly less accurate than measurements taken from a point gage if the radar is not properly calibrated. The use of raingages in calibration can greatly improve the accuracy of radar measurement.

Radar rainfall measurement works by emitting an electromagnetic signal from a transmitter and analyzing response signals that are reflected back to the receiver by the target. When electromagnetic waves encounter a target (raindrop, dust, bird, etc) they are scattered. Some of the scattered waves are reflected back to the receiver. These waves have less power than the transmitted waves. The average power of the return waves is proportional to the diameter of the rainfall drop raised to the sixth power.

$$\overline{P}_r = \frac{C}{r^2} \sum d^6 \qquad (2\text{-}32)$$

C=constant that is dependant on the radar design such as power, antenna size, beam width, and wave length

r=range from target to the antenna (2 to 230 km)

d=diameter of the raindrops

The reflectivity factor can be defined by the following equation relating it to the average power,

$$Z = \int D^6 N(D) dD \qquad (2\text{-}33)$$

The reflectivity factor can then be related to the rainfall rate R,

$$Z = aR^b \tag{2-34}$$

By the integration of the rate of mass precipitation,

$$P(D) = \rho_w N(D) \frac{\pi}{6} D^3 \left[V_T(D) - V \right] \tag{2-35}$$

Therefore if the power of the return waves are measured the rainfall rate can be estimated.

The major difficulty is in the determination of the parameters *a* and *b*. Linsley et al. (1982) said *a* can range from 15 to 1000 and *b* from 1.2 to 3.2. Marshall and Palmer (1948) [(6)] suggested average values of 200 and 1.6 for *a* and *b* respectively by empirical analysis. The National Weather Service (NWS) uses a=300 and b=1.4 for its WSR-88D radar network. The values suggested by the NWS have been shown to be a reasonable value for use in the Eastern United States.

There are several reasons for discrepancies between radar and raingage measurements. It should be recognized that some of the differences are due to the differences in the way that measurements are taken and what is being measured. Radar can miss precipitation that is formed at low levels. Also precipitation that is detected at higher levels may not fall on the ground directly below where the measurements were taken. This can be due to evaporation or horizontal movement due to wind. Vertical variation of intensity and precipitation type as well as anomalous propagation can also be sources of discrepancy.

The raindrop size distribution model implied by Marshall and Palmer, although widely used and accepted, may not be the best model. In a study conducted by Quimpo and Brohi [(7)] it is shown that the normal or Gaussian distribution is inadequate. It is instead suggested that a lognormal distribution results in a better fit.

The transmitted wavelength can have an effect on the quality of the data. Radar systems with shorter wavelengths tend to be less expensive. Shorter wavelengths are better for detection of small particles. Longer wavelengths work better at long distances and are better at detecting the spatial distribution of a storm because there is less absorption. Less absorption allows for the detection of a large storm located at the distance behind a smaller storm that is closer to the radar. Because most hydrologic models are concerned with larger storms, longer wavelengths tend to be more desirable.

Grayman and Eagleson (1971) [(8)] defined error in radar rainfall measurement as large and small scale error. Large scale error is error that occurs with a large area and a large time scale, such as with a storm total.

$$x_t = \frac{\text{radar measurement of storm total}}{\text{true storm total}} \tag{2-36}$$

x_t is log-normally distributed

where,

$\mu_{xt} = 1.0, \sigma_{xt} = 0.5$

Small scale error is error that occurs with radar measurements of rainfall intensity and radar reflectivity.

$$x = \frac{\text{radar measurement}}{\text{raingage measurement}} \qquad (2\text{-}37)$$

where,

$\mu_{xt} = 1.0, \ \log_{10} \sigma_x = C_1 + C_2 \log_{10} t + C_3 \log_{10} a + C_4 r$

The variance is a function of

1. Average time of echoes, t

2. Area over which the echoes are averaged

3. Distance from radar to the storm, r

4. Spatial characteristics of the storm

2.11.3 Calibrated Radar Data

One of the most effective ways of reducing the variance associated with radar rainfall measurement is by calibration of the radar data using raingages. The variance is reduced as the density of the gage network increases. The ratio of the variance after calibration (σ_{yA}) to the variance before calibration (σ_y) can be calculated using the following equation.

$$\frac{\sigma_{yA}}{\sigma_y} = 1 - \frac{R_c}{2.3}\left(\frac{\pi}{A}\right)^{1/2}\left[1 - e^{-(2.31R_c)(A/\pi)^{1/2}}\right] \qquad (2\text{-}38)$$

A = Area per raingage

R_c = distance at which the spatial correlation of rainfall begins oscillating

Raingage calibrated radar rainfall data is a good source of data for hydrologic modeling because of the representation of the spatial distribution of the precipitation. It has been found to be useful in the implementation in distributed GIS based models such as this one.

2.12 Geographic Information Systems (GIS)

Geographic Information Systems are software tools that are used to link attributes to spatial relations defined in a common coordinate system. Attributes can be grouped by their particular data type in a file called a layer. All of the layers in complex system must be defined using the same coordinate system and projection for the information to be effectively related. Data used in GIS can come from a variety of sources such as maps, text, and satellite and radar imagery. Most GIS software includes a graphical interface where the user can view the layers either separately or in a combined overly. Often computational tools, database management tools, and image processing tools are also included in the GIS software package.

2.12.1 Projections and Coordinate Systems

Maps are developed on a two dimensional surface to represent the earth which is a spherical object having a curved surface. A map projection is required to perform this transformation. *A map projection is a mathematical model that transforms (or projects) locations from the curved surface of the earth onto a flat sheet or two dimensional surface in accordance to certain rules (Shamsi 2002)* [(9)]. The two most commonly used projections are Universal Transverse Mercator (UTM) and State Plane Coordinate (SPC) projection. The UTM projection is usually used in national data sets. State Plane Projection is most commonly used in local data sets for states and counties. Another descriptor in the projection is the datum. A common definition for datum is a point, line, or surface used as a basic reference in mapping or surveying. The datum used in GIS is a spheroid. The two most commonly used datum are North American Datum of 1927 (NAD1927) and the North American Datum of 1983 (NAD1983). Data obtained from various sources can have different projections and coordinate systems. Some GIS software programs include projection modules that can be used to be used to re-project the data into a chosen projection to match other data.

2.12.2 Raster Files

A raster file is an image file made up of small equal sized independent cells or pixels arranged in rows and columns to form a grid. In a GIS program a pixel represents an area that is defined by the resolution described in the file's metadata. Attributes in a digital form are assigned to each of the individual pixels. When the file is displayed a specific color is assigned to each value or range of values of the digital attributes. Raster files are commonly used in applications where high resolution, spatial variability is an important consideration. Raster files work well where many computations are to be made based on the attributes. One of the problems with raster files is that they are not good for viewing multiple layers overlaid in a single view. It is often necessary to convert one of the layers to a vector polygon file so that the outline of the common attributes can be overlaid onto the raster file or visa versa. Digital Elevation Models (DEMs) are represented in the form of a raster file. Elevations are identified for each cell in the grid. Because the elevations are given at equally spaced intervals the calculation of things such slope and flow direction from a raster file is relatively easy.

2.12.3 Vector Files

Vector files are image files that are defined by collections of points or nodes. The nodes can be connected to make lines. A group of lines can be connected to enclose an area forming a polygon. In a GIS program attributes can be assigned to the nodes, points, lines, and areas defined by the polygons. Vector files work well in situations where multiples layers are overlaid onto a base map. A Triangular Irregular Network (TIN) would be an example of a vector file. A

TIN is a type of elevation model where elevations are connected to form triangles representing areas of constant slope.

2.12.4 Metadata Files

A metadata file contains information about the GIS file. These files commonly contain information such as map units, projection and coordinate system, file format, resolution, and information about the author. Metadata files can differ with GIS software. With some GIS programs metadata files are text files written by the person collecting the data. Other types of metadata files are actually written and referenced by the software program itself and may be required in order to view the data. The advantage of the file being written by the program is that it is standardized. Also changes such as reclassification or re-projection performed on data files are automatically recorded in the metadata file. This way the metadata information always accompanies the data file and is always updated. Data without a metadata file can often be useless when things such as units and coordinate system are not known. Attempts have been made by government agencies to set standards for metadata formatting.

2.13 GIS Based Hydrologic Modeling

GIS has become an invaluable tool in hydrologic modeling. It gives the modeler the ability to combine large amounts of data from several sources and the tools necessary to manage and view the data effectively. Because hydrologic systems are very complex and require a great deal of data to model, GIS has greatly improved hydrologic modeling. GIS has introduced the visual representation of the models. Models prior to the use of GIS were usually represented by text files. These files were cumbersome and limited the amount of data that could be effectively viewed. The early models also required a great deal of experience in order to understand input and output formatting. GIS offers the data in a graphical form that can be much easier to understand even by those without a great deal of experience. This can make the presentation and conveyance of details of the model much easier. Therefore the appreciation of hydrologic modeling is no longer limited to the hydrologic modeler. Information can be passed on much easier to other parties such as the management or the client.

2.13.1 HEC-HMS

HEC-HMS stands for Hydrologic Engineering Center Hydrologic Modeling System. HEC-HMS is a software package developed for the Army Corps of Engineers to supercede HEC-1. The HMS software is designed for the modeling of dendritic watersheds.

There are three basic components of the software. The first component is the basin model. In the basin model the elements such as the sub-basins, reaches, junctions, reservoirs, diversions, sources, and sinks are defined. There is a graphical display in which the components can be placed and viewed. Parameters for each of these components can be set such as the types of losses, the channel routing, and the type of runoff transformation to be used. The second component of HMS is the Meteorological model. In this component the precipitation and evaporation is defined. The user can chose from simulated rainfall such as SCS design hydrographs or real-time data can be used as input. The third component is the control specifications. This is where the dates, times, and time increments are specified.

An extension to HMS has been developed called HEC-GeoHMS. HEC-GeoHMS works along with Arc View 3.2. It is designed to use GIS information to prepare input for the HMS program. These programs are publicly available at no charge.

2.13.2 WMS

WMS stands for Watershed Modeling System. It is a GIS based model designed for hydrologic modeling. Unlike HEC-GeoHMS it has stand alone GIS software. It does not require the use of GIS program such as ArcView. It allows for the use of several methods for estimating runoff and flood modeling. It can also be used in the same manner as HEC-GeoHMS to prepare input data for HEC-1 and HEC-HMS.

2.13.3 Previously Research on this Model

The research described in this thesis is a continuation of research conducted from 1994 until the present by Dr. Quimpo and previous graduate students at the University of Pittsburgh. The following sections will outline the work done by other students under Dr. Quimpo's guidance and supervision.

2.13.3.1 Mark Michelini. The first student to work on this project with Dr. Quimpo was Mark Michelini [10]. Michelini's research consisted of the development of a GIS based automated rainfall-runoff model. The model used many of the same concepts currently being used in this research. The model calculated the travel time for each cell in the watershed and developed the time area ordinates from which the unit hydrograph could be developed using the Clark unit hydrograph method as explained in section 2.5.3. Michelini's model used a single value for the curve number for the calculation of the excess precipitation. This curve number was assumed by the user. He tested two methods for calculating the travel time for the cells. The first method used the SCS equation for the time of concentration to calculate the travel time. The second method used the Kerby equation for the calculation of the overland flow and the Ramser equation for the calculation of the channel flow. It was determined that the method that accounted for the travel times of the overland flow and the channel flow separately was the better method. Michelini finished his research in 1994.

2.13.3.2 Randal Bodnar. Randal Bodnar [11] was the next student to work on this project. Bodnar's work consisted of improving the model by examining different methods of calculating the time of concentration, and verification and calibration of the model using field data. The calibration was done by adjusting the values of the storage factor K. At this time the model was still using a composite curve number that was assumed by the modeler. Bodnar finished his research in 1995.

2.13.3.3 Jaber Hamad Al-Medeij. Jaber Hamad Al-Medeij [12] worked on the research following Bodnar. Al-Medeij's work was focused on reducing the subjectivity involved in the assumption of the basin curve number. He was the first student in this research to develop a method of automatically generating the curve number for the basin. This was done by digitizing soil and land use maps and then creating a program which used these maps to calculate a composite curve number for the basin. Al-Medeij finished work on his thesis in 1998.

2.13.3.4 Khalid Naseem Khan. Khalid Naseem Khan [13] was the last student to work on this research. His contributions to the model were the introduction of spatially distributed curve numbers, the integration of the SCS overland flow velocity formulas, and the investigation of the effect of breaking the watershed into sub-watersheds for analysis. Khan developed a program which calculated the curve number for each of the cells in the watershed and wrote them to a raster file that could be used in the calculation of the excess precipitation. At the time of Khan's research GIS was becoming more widely used. He also mentioned the possible use of spatially distributed radar rainfall data. Khan finished work on his research in 2001.

3.0 OBJECTIVES

3.1 Refine Methodology and Investigate Alternative Methods

Dr. Chiu, as well as many others involved in modeling, has emphasized that modeling process should not end after the conception of the model. The modeler should seek to make improvements by repeating the modeling process. The process begins by making observations of the event and collecting data. The data is then organized and analyzed. The conception and development of the model are the next steps in the process. The model must then be tested. Observations are then made between predictions of the model and the data collected. The data can be compared graphically or statistically. When comparing the predicted values to the observed values one should also consider the reliability of the observed data. It should be noted how the data was collected and the accuracy and precision of the measurements. The differences between the predicted values and the observed values are then studied to determine how the model can be improved by calibration or by use of alternative methods. Changes are then made to the model and observation begins again. A modeler should always look to improve upon the model by repeating this process.

The model described in this thesis has been improved upon several times by several different students. When a student begins work on the model, the first thing necessary is an understanding of how the model works. To do this it is necessary to understand the work each student has done by reading the thesis written by each of the students. This is critical because this gives an idea of the intent of the research. The next step is to investigate the model in order to see that what was intended is indeed carried out correctly. The intent of this investigation is not to discredit the previous student's work. The programs should be tested without bias. This is often better accomplished by the student doing this follow up work. Methods used should be investigated, along with alternative methods, to determine if changes should be made that would improve existing processes before any new work is done to the model.

3.2 Refine Data Collection

The collection of soil, land use, and precipitation data for use in this modeling program has been researched extensively for this phase of the project. This was not as essential during the development phase of the programs where the main emphasis was on programming. In this phase of the project the model is being compared with real data, therefore the data used as input into the model should be the best data available in order to expect reasonable results.

3.3 Implementation of Gridded Precipitation Data

Because the objective of this model is to make use of spatial distribution it is only reasonable to try to make use gridded rainfall data from calibrated radar data. This data has been made available by Three Rivers Wet Weather Demonstration Program (3RWWDP). Implementation of this data in this phase of the research, where the model is being tested against actual data, meant that the data had to be available for a small watershed that has an active stream gage with recent data. Unfortunately the only small watershed (under 30 square miles) in

Allegheny County that still has an active stream gage is Little Pine Creek. This watershed extends out of the area that was covered by the gridded precipitation at the time that this research began. The coverage area was expanded in June to the entire Allegheny County which included the entire Little Pine Creek watershed. The 3RWWDP is receiving these data but they have not been published on their website at this time. Because the data have only been available for a short period of time during the summer months the rainfall data is limited. The 3RWWDP supplied data for the testing of this model for the months of June through August.

3.4 General Testing of Programs

In order for a computer model to be accepted by the engineering community it must be tested extensively. It should be tested not only to ensure that the programs work correctly, but so that the results are reasonable for any given situation. This type of testing involves the modeler attempting to debunk his own work. It is the modeler's responsibility to test the programs without bias to ensure the credibility of the model.

3.5 Model Calibration and Verification

In order to calibrate a model that is going to be used for the determination of runoff in ungaged watersheds it is first necessary to compare the results obtained from the model using actual rainfall data with the corresponding runoff data before it is assumed that the results from the model are reasonable. This was done in this study by selecting a small watershed in

Allegheny County with an existing active streamgage. The selection was limited to a single watershed. The Little Pine Creek watershed was the best candidate. The USGS maintains a streamgage at the outlet of the Little Pine Creek watershed. The data from the stream gage was compared by the time to peak (T_p), the peak flow (Q_p), and the total volume of runoff.

The watershed storage coefficient (K) used in the reservoir routing applied to the time area curve must be calibrated by analyzing the model output and the actual streamflow record for a particular storm. The typical suggestions for the values of the storage constant are based on the time of concentration (T_c). Suggested values range from $0.75T_c$ to T_c. This research attempts to determine an acceptable value of the storage coefficient to use in the model.

4.0 METHODOLOGY

The curve number method, time-area method, and Muskingum routing method are by no means the most sophisticated methods that could have been chosen. The simplicity of these methods is one of the appealing characteristics of this model. More sophisticated methods tend to involve more parameters which need to be estimated. The estimation of these parameters can be quite subjective. This subjectivity can lead to a great deal of error in their estimation and in turn this error can then be magnified in the resulting output. This is especially true with spatially distributed models. With a spatially distributed model it is almost necessary to keep the model from becoming too complex because of the numerous calculations involved.

4.1 Geographical Information Systems

The GIS software used in this model was IDRISI. IDRISI was developed by the Graduate School of Geography at Clark University. It began as a raster based GIS. Therefore IDRISI was originally chosen because of the software's ability to effectively handle raster files. Raster files are used for the representation of the spatial distribution of the watershed and meteorological characteristics. ArcView was used for the initial processing of the soil and land use coverage because these files were obtained in the form of shapefiles. The shape files were then converted into IDRISI raster files.

4.2 Time Area Method

The time area method is used in this model to relate travel times for runoff from each cell in the watershed to the outlet. The first step in the time area method is the development of a map of isochrones. Figure 4-1 shows an example of an isochrone map. Isochrones are contours of equal travel time. The isochrones therefore delineate areas in the watershed with travel times that fall between specific time increments. A time-area histogram can be developed from the map of isochrones.

In this model the travel time was calculated for each cell within the watershed. This was done by calculating the travel time for both the overland and channel flow sections of the flow path separately. The flow path is determined by the direction file. The overland flow and channel flow are designated by the stream file in the form of either a 1 representing channel flow, or a 0 representing overland flow. Because of the scale of this model and the automation involved using remote sensing it is not possible to calculate the travel time using boundary value hydrodynamic equations. Many of the parameters such as the hydraulic radius and roughness of the channels are not available from the GIS data. Therefore more versatile, simplified equations were used in the calculation of the travel time of the cells.

Figure 4-1 Map Showing Isochrones

4.2.1 Travel Time for Overland Flow

The SCS equations for the overland flow velocity were used to calculate the travel time for water flowing from a particular cell to the point at which it enters the stream and becomes channel flow. These equations use the average slope along the flow path and the curve number in the calculation of the velocity. The curve number could be thought of as a representation of the roughness as it is a function of the soil and land cover. The travel time is then calculated by simply dividing the length of the flow path by the estimated velocity. Table 4-1 shows the velocity formulas for different values of the curve number. These were taken from *The SCS National Engineering Handbook, Section 4, Hydrology.*

Table 4-1 Overland Velocity Formulas

Curve Number	Velocity Formula
0 - 50	$y = 0.041e^{0.2951x}$
51 - 60	$y = 0.0813e^{0.2836x}$
61 - 70	$y = 0.1217e^{0.2861x}$
71 - 75	$y = 0.1686e^{0.2882x}$
76 - 85	$y = 0.2533e^{0.2925x}$
86 - 100	$y = 0.3378e^{0.2885x}$

4.2.2 Travel Time for Channel Flow

The Ramser equation was used to calculate the travel times for a particular cell from the point at which it enters the channel to the outlet of the watershed or sub-watershed. This equation is of the following form:

$$T_c = 0.02 L_c^{0.77} S_c^{-0.385} \tag{4-1}$$

Where L_c is the length of the channel in meters, and S_c is the average slope of the channel. This program only considers the length and slope therefore it assumes that the velocity does not change with changes in flow.

4.2.3 Total Travel Time

The total travel time for a particular cell is found by adding the travel time of the overland portion of the flow path and the travel time for the channel portion of the flow path. This time is then written to the cell in the isochrone file.

4.3 SCS Method for Abstractions

The Soil Conservation Service Curve Number Method was used to account for abstractions in order to estimate the excess precipitation. The method was employed by generating a curve number raster file for the watershed. The curve number file is a GIS layer containing unique values of the curve number for each individual cell in the watershed. The Runoff program uses the data in this file as input to the SCS equation for calculating the excess precipitation. This method was chosen because it is a simple yet effective method for estimating runoff and it works well with the application of time area method.

Prior to this research the runoff was generated from a single value of precipitation. In this research the programs were modified so that incremental precipitation could be used in the model. Because the rainfall intensity varies with time, a cumulative method was used to calculate the abstractions. This is done by calculating the cumulative excess precipitation at each time increment using the cumulative precipitation. The incremental excess precipitation for a particular increment is then calculated by deducting the cumulative excess precipitation for the previous increment. This accounts for the reduction in the abstractions as the rainfall continues to wet the basin and saturate the soil.

4.4 Reservoir Routing

Reservoir routing was previously used in the model to simulate the effects of the storage in the watershed. The way this is done is by routing the time-area series through a linear reservoir. This is described in the Clark Unit Hydrograph method. The storage factor K must be estimated. Suggested values range from $0.75T_c$ *to* T_c.

4.5 Channel Routing

The method used to route the hydrographs from an upstream outlet to the subsequent downstream outlet is the Muskingum Routing method. Once a hydrograph is routed to the downstream outlet it is then added to the hydrograph at that particular outlet. This process is repeated until the outlet for the entire watershed was reached. The value of the storage constant (K) that was used in the channel routing was equal to the travel time from the upstream outlet to the downstream outlet. The value of K can be found from the isochrone file by looking at the cell just downstream of the outlet from which the hydrograph is being routed downstream. The value of X used in the Muskingum routing must be chosen. The difficulty, in this type of model, is that little detail is known about the channel characteristics besides the slope. Chow [14] states that in natural streams X can vary from 0 to 0.3. It is furthermore suggested that the value of X need not be determined with great accuracy because it has a relatively small effect on the results.

4.6 Instantaneous Hydrograph Conversion

The hydrographs that were generated from the incremental precipitation files are instantaneous hydrographs. The runoff program converts these instantaneous hydrographs to hydrographs of duration equal to the precipitation time increment by averaging the ordinates at the beginning and end of each time increment. The averaged value is then made the new ordinate at the end of the time increment. This is as explained in section 2.5.5.

4.7 Programming

All of the programming in this research as well as past research has been done in FORTRAN. Programming for this research was done in a Force 2.0 FORTRAN compiler and editor. This is a free compiler that is available for download from the internet. It is available at the following address:

http://www.forceproject.hpg.ig.com.br/index.html

Help files for this compiler are available at the following address:

http://www.forceproject.hpg.ig.com.br/index.html

FORTRAN was chosen for several reasons. It is a fairly commonly used language and relatively easy to learn. It also has the ability to handle raster files easily. The way the raster files are viewed in FORTRAN is as a two dimensional array. Therefore raster files can be read by using a double Do loop. Some of the methods commonly used in the programming are included in the manual written for this model.

5.0 DATA COLLECTION

Data collected for this model was obtained from several sources. Most of the data used was available through public domain and therefore could be obtained with no charge. There were some restrictions on the use of the data such as the land cover files, such that they were only for educational used and not to be used commercially. The DEM raster files were originally purchased from the USGS, but are now available on the internet for free download. One should be aware that when data are obtained at no charge the quality of the may not be as reliable as data which are purchased from a vendor where the data usually is guarantied. With this in mind the data should be examined for noticeable errors before it is used.

5.1 Digital Elevation Model (DEM)

The Digital Elevation Model, commonly referred to as a DEM, is one of the most important data sets in almost any GIS based hydrologic model. A DEM is a digital representation of cartographic information in a raster form. The DEM raster files used in this model were obtained from the USGS. The same USGS DEMs can also be obtained on the Pennsylvania Spatial Data Access (PASDA) website:

www.pasda.psu.edu

The projection of the USGS DEMs is in North American Datum of 1927 Universal Transverse Mercator Zone 17N. The resolution of the DEMs used is 30 meters. This means that each cell has an area of 0.22 acres. DEMs are created either by digitizing the USGS topographic maps or more recently they are created directly from the stereographic photography.

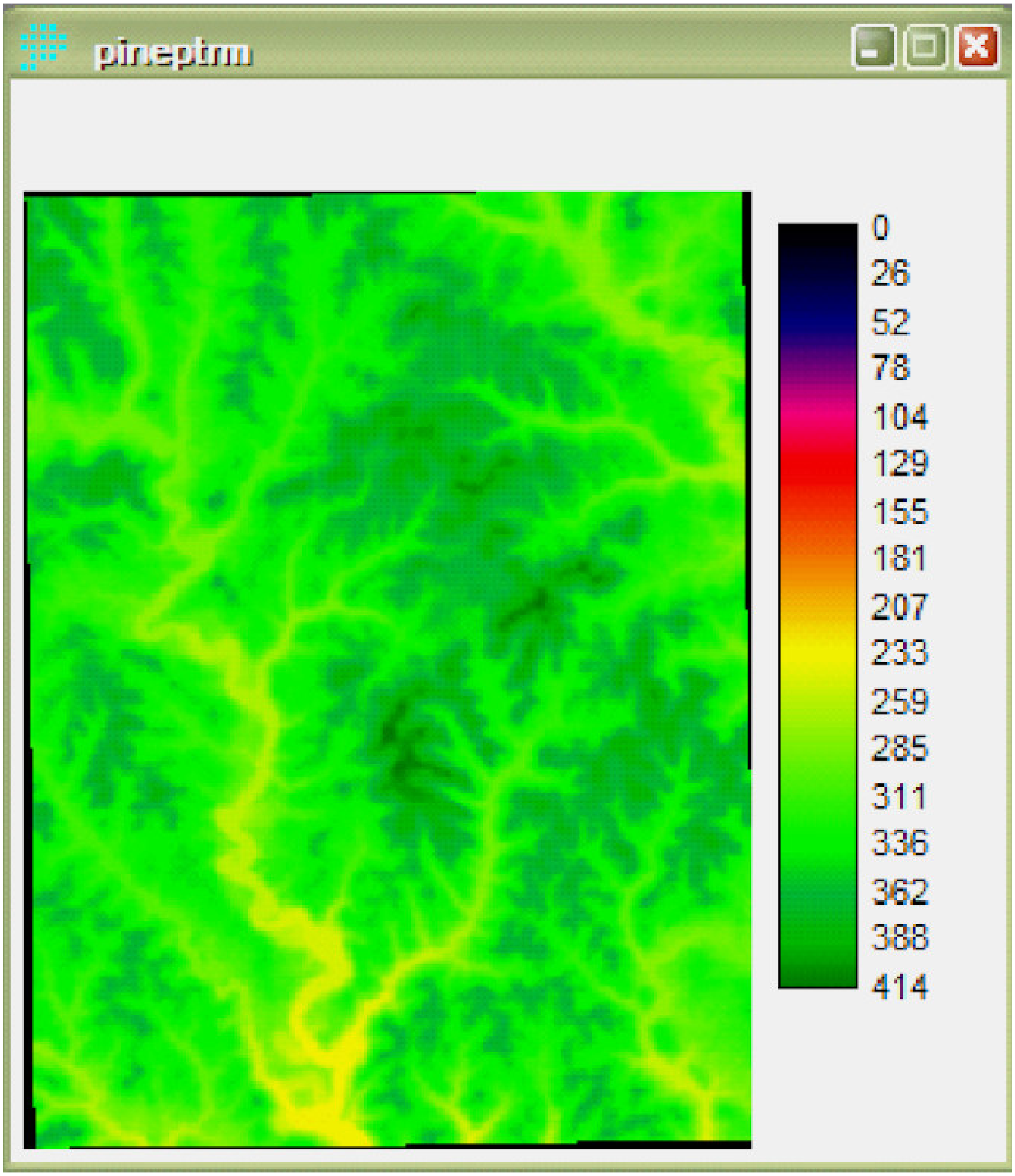


Figure 5-1 Digital Elevation Model

5.2 Soil Data

The soil data were obtained from the US Department of Agriculture Natural Resources Conservation Service (NRCS) SSURGO website. There is a link to this data on the PASDA website mentioned previously. The soil data were obtained in the form of a shapefile. The shapefile covers the entire Allegheny County area. Figure 5-2 shows the NRCS soil survey shapefile. The projection of this file is also North American Datum of 1927 Universal Transverse Mercator Zone 17N. The NRCS, formerly known as the Soil Conservation Service, is the agency that conducted the original soils survey. These files were created by digitizing the soil survey maps from the original survey. The original survey was published in 1981. It was digitized and published in GIS form in 1999.

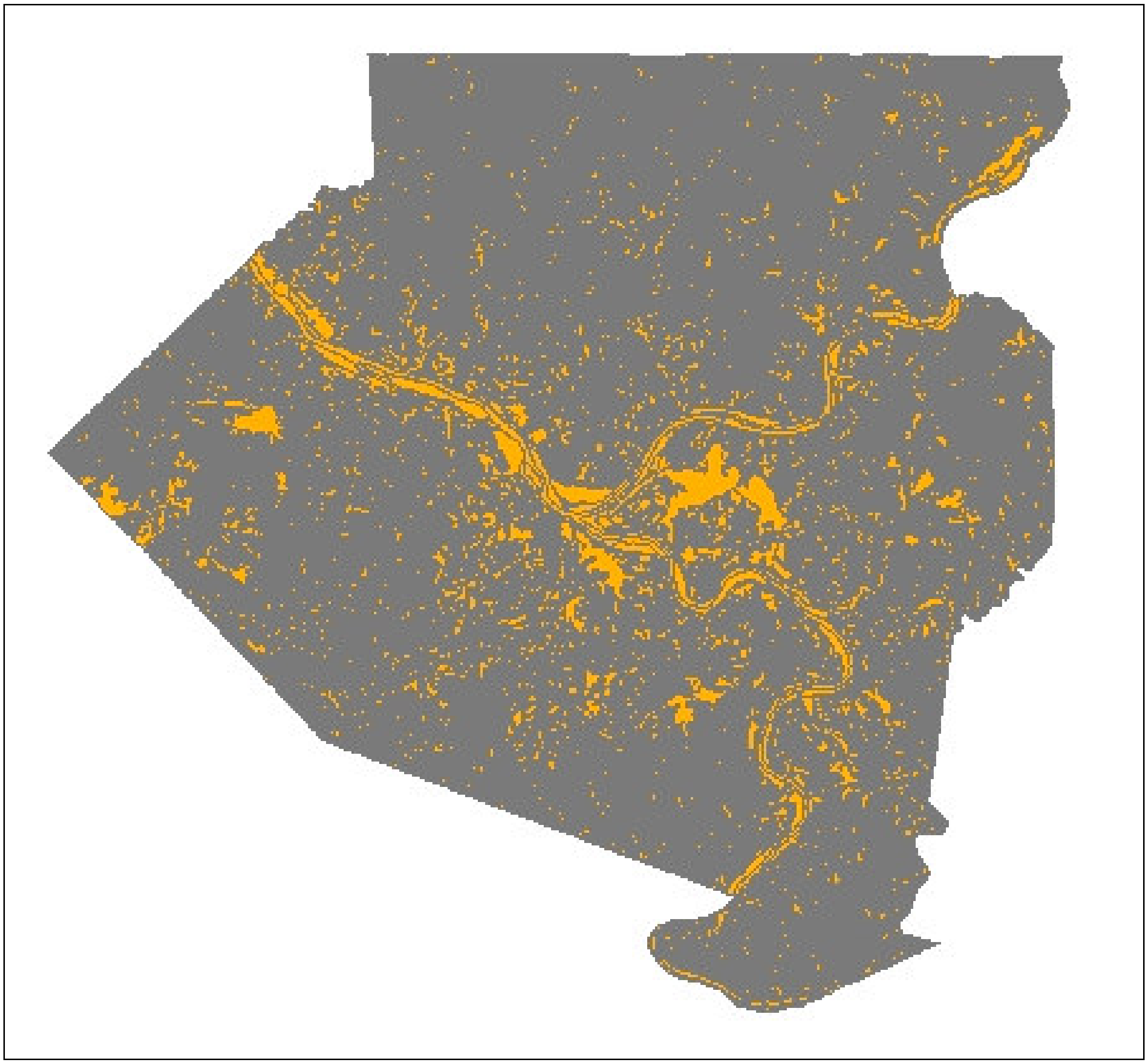


Figure 5-2 Allegheny Soil Survey Shapefile

5.3 Land Cover Data

The land cover data were obtained from the Government Publications in Hillman Library. These data were in the form of three shapefiles (North, South, and East) covering all of Allegheny County. Figure 5-3 shows the three land use shape files. The projection of these files was in North American Datum of 1983 Pennsylvania State Plane South projection. The land cover survey was conducted by the Southwest Pennsylvania Commission. The data was collected using Landsat imagery. The survey was conducted in 1992.

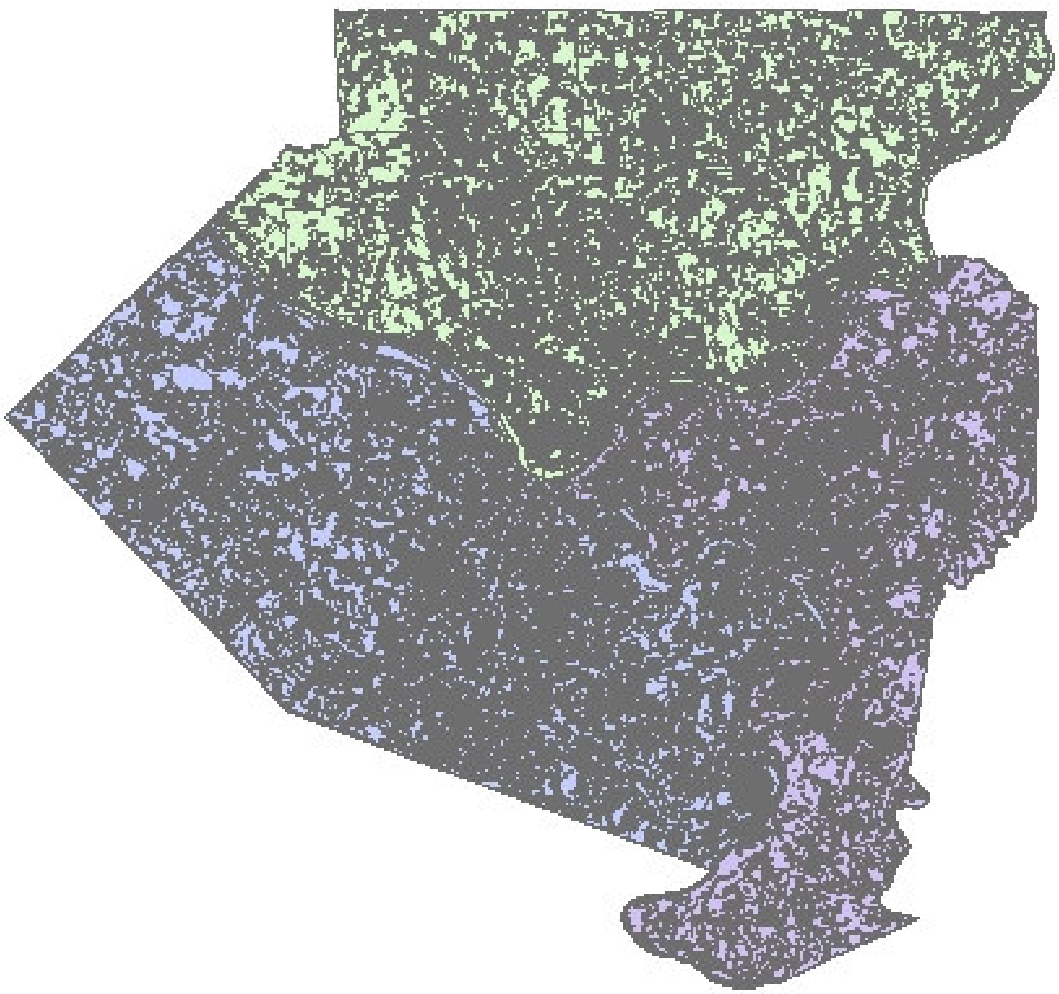


Figure 5-3 Allegheny County Land Cover Shapefiles

5.4 Precipitation Data

The precipitation data for this study were obtained from Three Rivers Wet Weather Demonstration Program. The 3RWWDP web site contains both a point gage network consisting of 21 gages and gage adjusted radar precipitation data for the ALCOSAN service area as shown outlined in red in Figure 5-4. The point gages that are used to calibrate the radar data can be seen in Figure 5-5. The radar data coverage will soon be expanded to the entire Allegheny County area. This will include the addition of more point gages for the calibration.

The gridded precipitation is from gage adjusted radar rainfall data in the form of a grid where each cell has 1 kilometer resolution. The grid can be seen in Figure 5-6 overlaid onto Allegheny County. Incremental precipitation data can be downloaded for each cell in the grid. This allows for the representation of the spatial distribution as well as temporal distribution of each rainfall event. The gridded precipitation is available from April 2000 to the present. The rainfall data will eventually be available in real-time format. The radar rainfall data is collected and calibrated by Vieux & Associates, Inc.

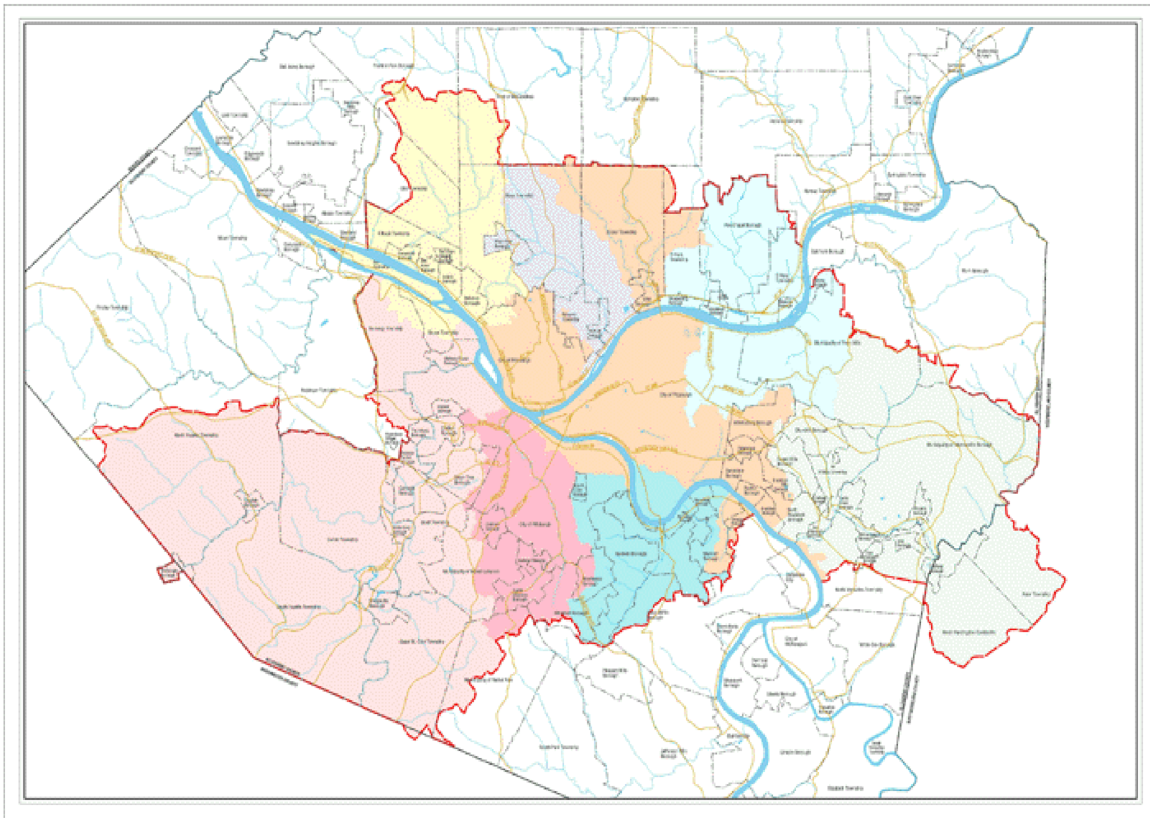


Figure 5-4 Current Coverage of Gridded Precipitation Data

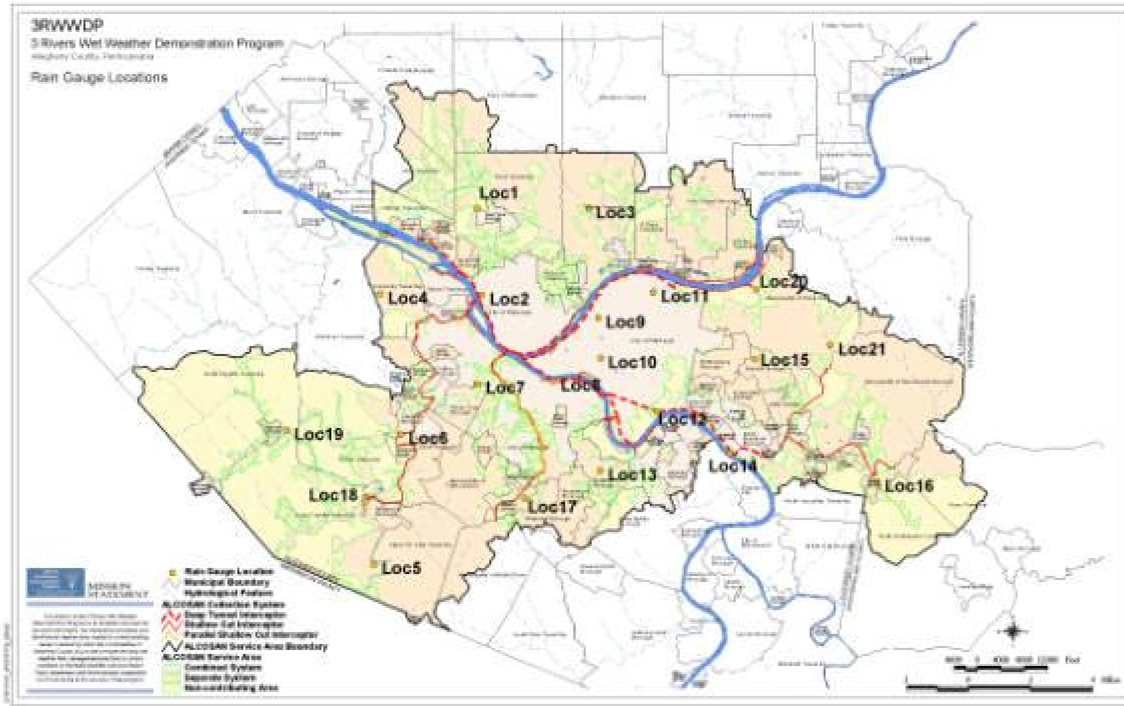


Figure 5-5 Gage Network for Radar Calibration

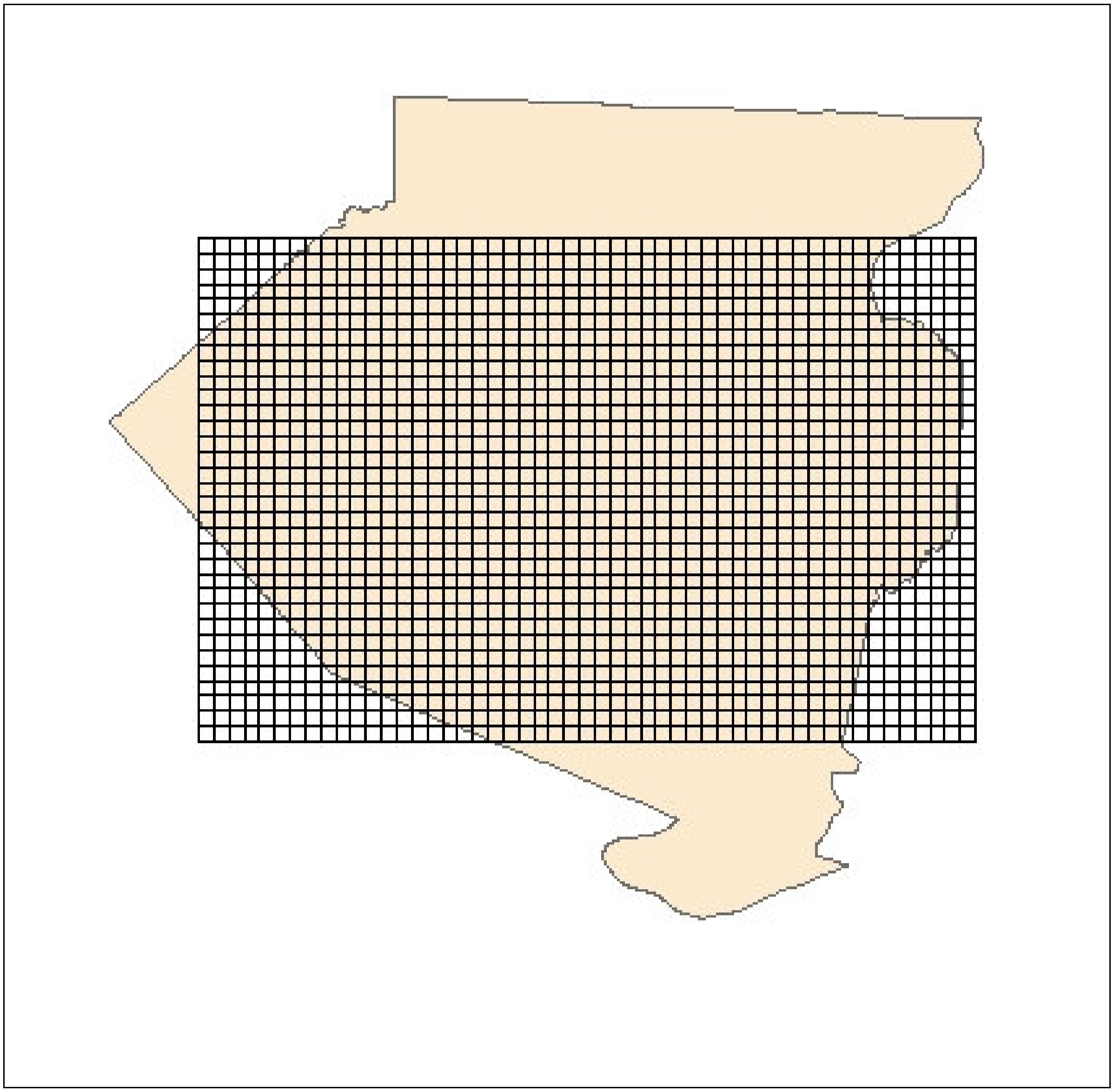


Figure 5-6 Grid Overlay

5.5 Streamflow Data

The streamflow data were obtained from Ray Siwicki of the United States Geological Survey (USGS). The hourly historical data are not available on the USGS website. The data must be obtained by contacting the USGS and requesting the data for the period of interest. The hourly streamflow data are recorded by a pressure transducer. The pressure transducer measures the stage of the stream. Measurements of the stream flow are taken at various stages. The flow measurements are then plotted against the stage to develop a rating curve. The rating curve is later used to determine the flow at a particular stage given by the pressure transducer. The accuracy of the rating curve can be affected by changes in control conditions produced by scour-fill. Therefore it is necessary for the USGS to make discharge measurements to update the rating curve periodically. This is done approximately every seven weeks. The accuracy of the gage can also be affected by leaves and ice. The USGS rated their records as fair for the last water year.

6.0 DATA PROCESSING AND COMPUTER METHODS

6.1 Digital Elevation Model (DEM)

The DEM is used by the hydrologic model to create raster files such as the direction file, stream file, and isochrone file, which represent properties or attributes based on calculations made from the elevation data. These raster files are developed for two basic reasons. One reason is that some of the attributes are used by more than one program. Rather than repeatedly calculating the attributes from the DEM a raster file is created in which the attributes are stored in a manner such that they can be referenced by their spatial assignment. This makes the programs run faster making the entire process less time consuming and better organized. These files are also created so that the user can view the attributes using a GIS graphical interface such as IDRISI. The visual representation of the attributes gives the modeler an improved perception of the characteristics of the watershed. This makes it easier for the modeler to notice problems with the input and output as well as giving a better understanding of the data.

6.1.1 DEM Correction

When DEMs are created sometimes there are errors created within the file. These errors are in the form of either pits or spikes. Pits are cells within the DEM which are lower than the surrounding cells. They are usually cells that were mistakenly assigned a zero elevation by the digitizing software when the DEM was created. Alternatively spikes are cells that are significantly higher than the surrounding cells. Depending on nature of the model in which the DEM is used these errors can have varying levels of significance. In watershed modeling, the pits tend to be the most problematic because they tend to terminate the flow path. Spikes tend to cause fewer problems because flow paths are diverted rather than discontinued. When a program evaluates the flow path the program recognizes water flowing into a pit but the water has no way of flowing out. The pits can be filled using the PIT REMOVE module found in IDRISI. In the use of DEMs in other types of modeling such as the study of waves in coastal areas the spikes are the more relevant error because they tend to signal to the model that the wave should break when it encounters the spike. The PIT REMOVAL module fills the pits by a method of interpolation.

6.1.2 Generating the Flow Direction File

The direction file is a raster file which contains integers that represent the direction of flow for each individual cell within the area covered by the DEM. The direction integers are assigned when the direction program evaluates the DEM one cell at a time. The program determines the steepest path from a particular cell to one of the eight adjacent bordering cells. Figure 6-1 shows a diagram showing the representation the direction associated with each integer in the direction file. The direction file is used in the delineation of the watershed and in the creation of the isochrones file. The newest version of IDRISI includes a module that does essentially the same thing but using different integers in the representation of the directions.

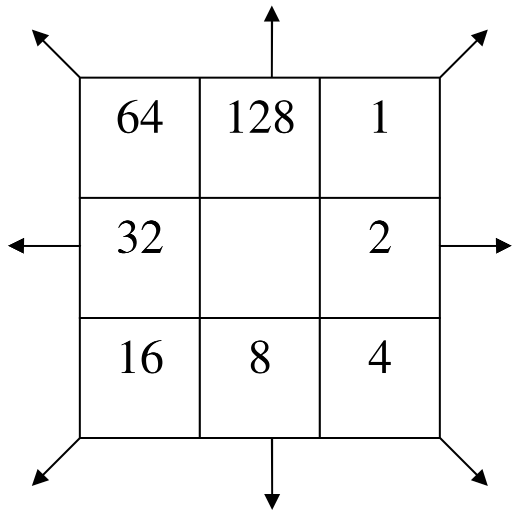


Figure 6-1 Direction Integers

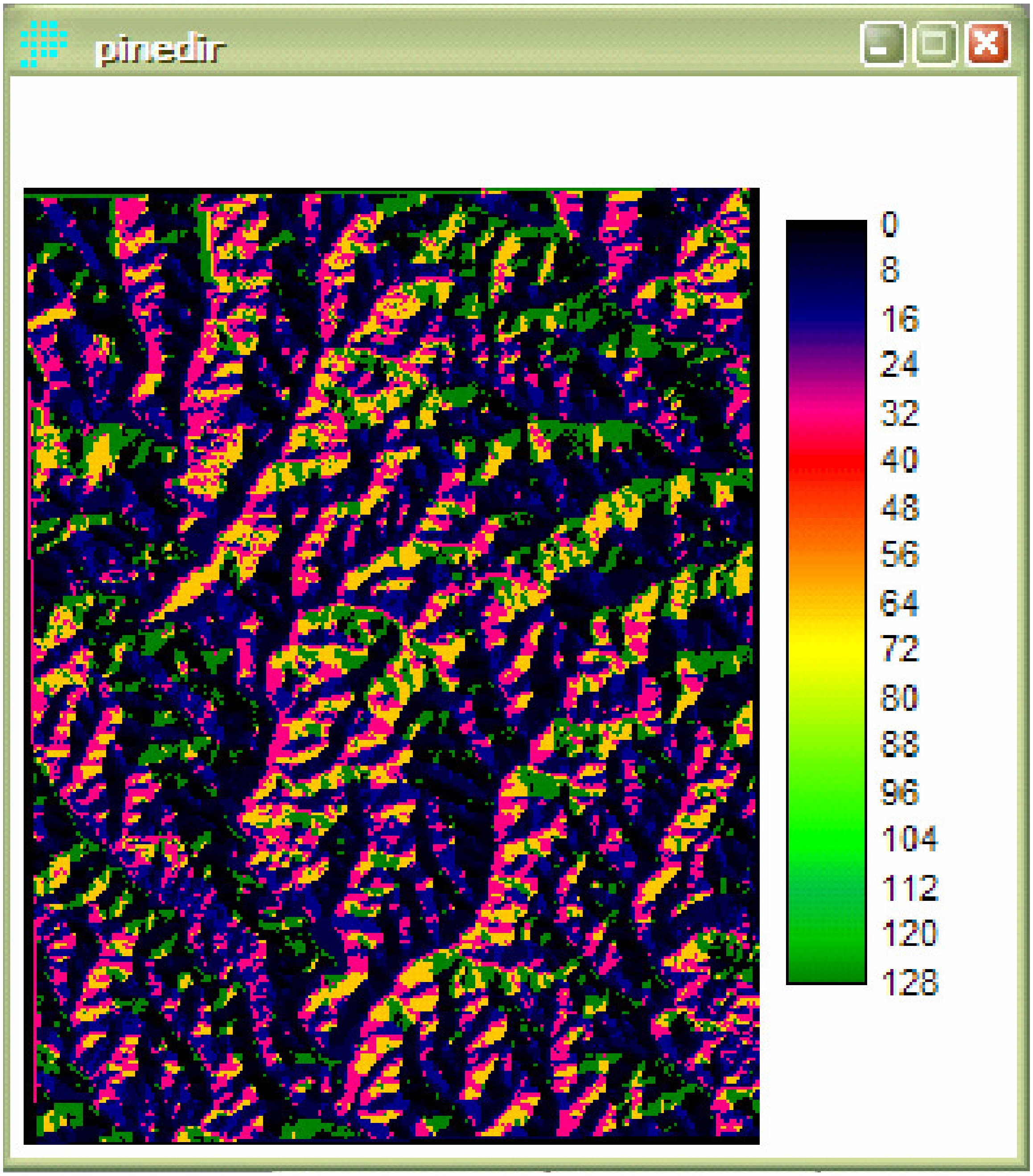


Figure 6-2 Direction File

6.1.3 Generating the Stream File

The stream file is a raster file which represents the stream network within the area of the DEM. The stream file is created using the RUNOFF module in IDRISI. The RUNOFF module determines the number of cells that flow into each individual cell and then writes that number to the corresponding cell in the stream raster file. The user then defines channel flow by designating a minimum number of cells contributing to the flow at a particular cell. If the user chooses 100 cells (22 acres) as the point at which flow becomes channel flow, all of the flow in cells that have less than 100 contributing cells is considered overland flow. All of the flow in cells where there is greater than 100 cells contributing to the flow is channel flow. The channel flow is then represented by a 1 being assigned to the cell. Overland flow is represented by a 0 assigned to the cell. The stream file then is used by the model to distinguish overland flow from channel flow in the generation of the isochrones file. It is also used as a visual aid when determining the placement of the outlet of the watershed. The outlets are defined by renumbering the cell in the stream file where the outlet is to be placed to a number of 2 or above. This identifies to the watershed program and the isochrone program where the outlet is located.

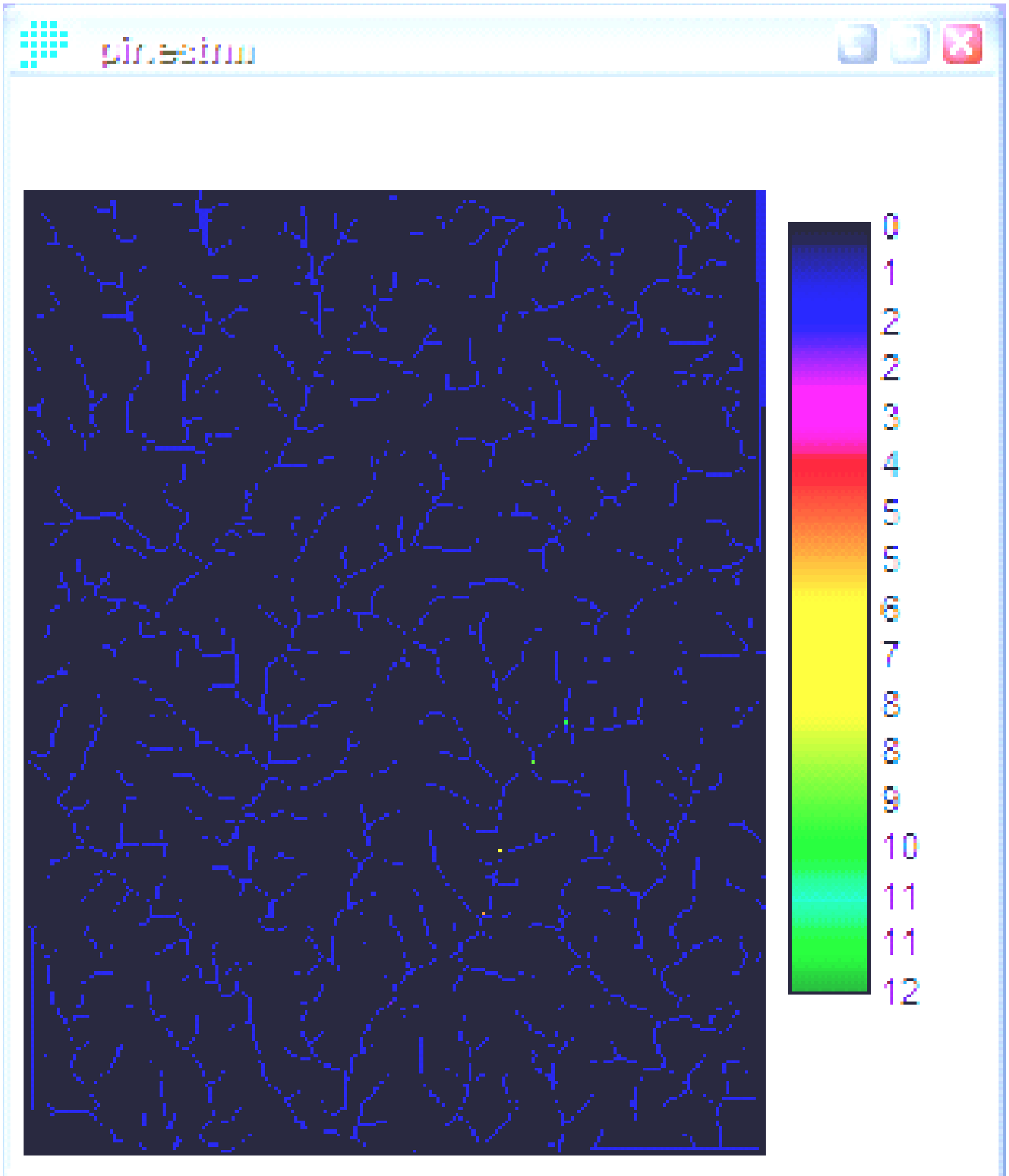


Figure 6-3 Stream File

6.1.4 Delineating Watersheds and Sub Watersheds

The watershed file is a raster file representing the watershed or sub watersheds that are being studied. The watershed file is created by the watershed program. Before the watershed program is run, the user must designate the outlets for each watershed in the steam file as mentioned in the previous section. The watershed program then uses the direction file to determine which cells flow into each outlet. The program designates the cells within a watershed by assigning each of the cells the same number as the outlet chosen for that particular watershed. For example, if the outlet is the number 6 outlet then the watershed is made up of cells containing the number 6. The background cells contain zeros.

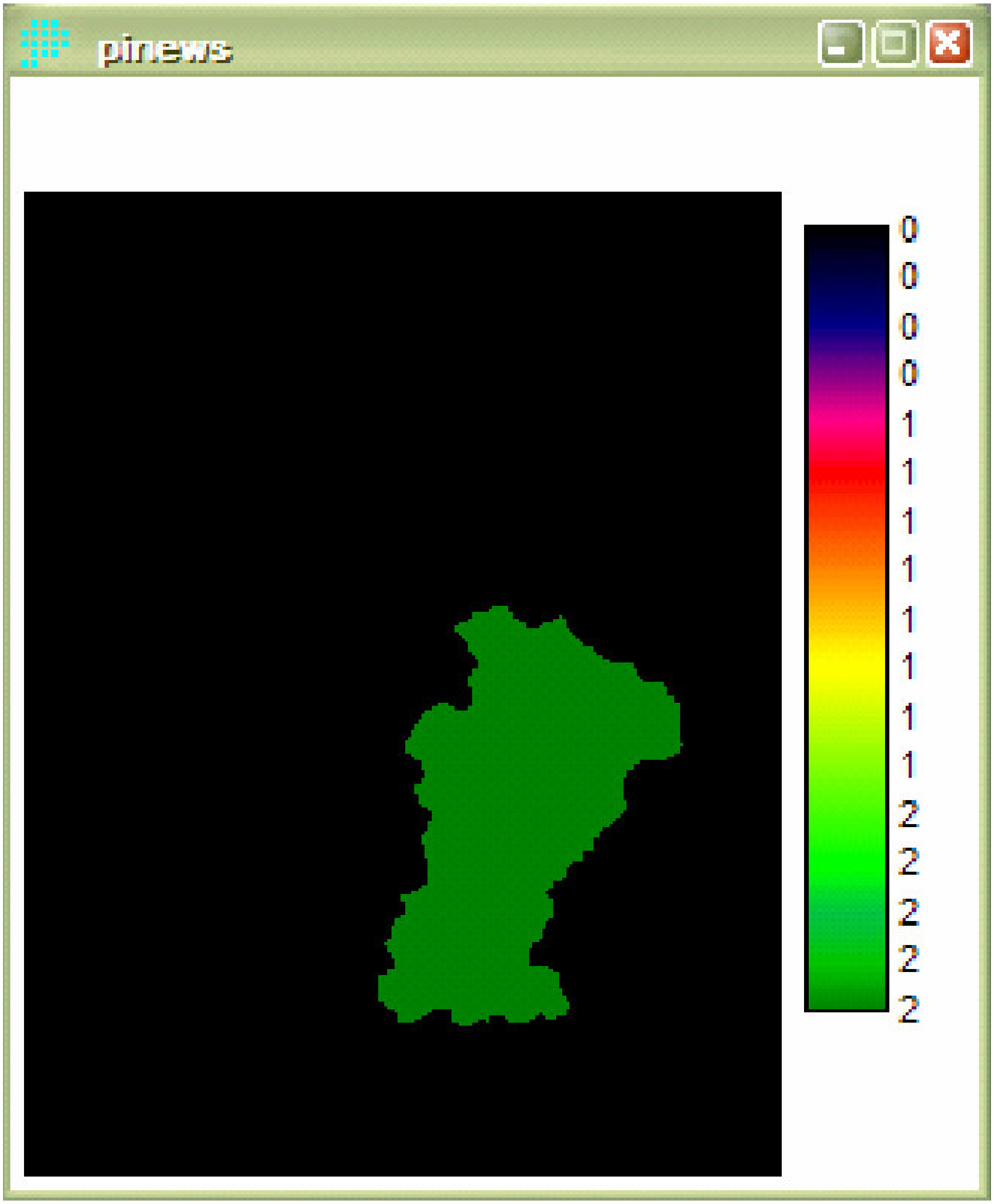


Figure 6-4 Little Pine Creek Watershed

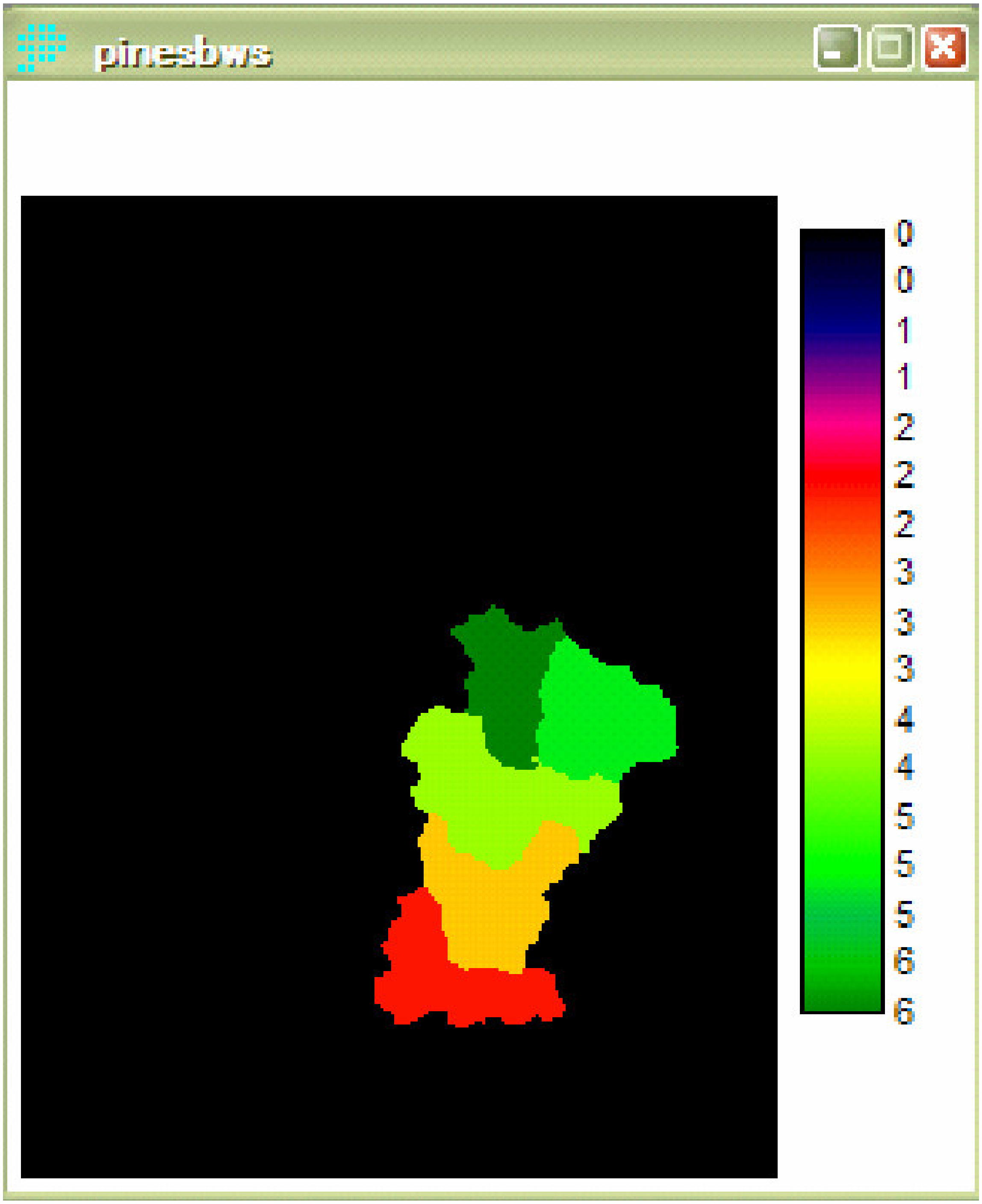


Figure 6-5 Example of Sub-watersheds

6.2 Processing the Curve Number Data

6.2.1 Creating the Soil Raster File

The soil raster file is created using the NRCS digital soil survey. The soil types in the soil survey are reclassified using by the hydrologic soil groups as given in *The SCS National Engineering Handbook, Section 4, Hydrology*. A description of each hydrologic soil groups can be seen in Table 6-1. The complete list of soil types and their corresponding hydrologic soil groups for Allegheny County can be seen in the Table 6-2. Once the soils have been reclassified the soil layer is clipped using an outline of the watershed that is being studied. The result is a raster file representing the hydrologic soil group for each cell in the watershed. The soil file and the land use files are used by the CURVENUM program to create the curve number raster file.

Table 6-1 Hydrologic Soils Group Descriptions

Hydrologic Soil Group	Description
A	Sand, loamy sand, or sandy loam. Soils in this group are made up of deep sands or gravels which are well drained to excessively drained with a high infiltration rate even when fully wetted. Consequently these soils have a low runoff potential.
B	Silt loam or loam. Soils in this group have a moderate infiltration rate. They consist mainly of moderately deep to deep, moderately drained to well drained sands or gravels.
C	Sandy clay loam. This group is made up of soils either having a layer that impedes the downward movement of water or soils consisting of a moderately fine to fine texture. These soils have a slow rate of infiltration.
D	Clay loam, silty clay loam, sandy clay, silty clay, or clay. Soils in this group have a low infiltration rate when fully wetted. This group consist of clays with a high swelling potential, soils existing in area with a high water table, soils with a clay pan or clay layer near the surface, and shallow sols over nearly impervious material. The soils in this group have a high runoff potential.

Table 6-2 Allegheny Soils Types and Corresponding Hydrologic Soil Groups

Map Symbol	Digital Number Used	Hydrologic Code	Mapping Unit
AgB	2	B	Allegheny silt loam, coarse subsoil variant, 2 to 8 percent slopes
AgC	2	B	Allegheny silt loam, coarse subsoil variant, 8 to 15 percent slopes
At	4	D	Atkins silt loam
BrB	4	D	Brinkerton silt loam, 2 to 8 percent slopes
BrC	4	D	Brinkerton silt loam, 8 to 15 percent slopes
CaB	3	C	Cavode silt loam, 2 to 8 percent slopes
CaC	3	C	Cavode silt loam, 8 to 15 percent slopes
CeB	3	C	Caneadea silt loam, 3 to 5 percent slopes
CkB	3	C	Clarksburgh silt loam, 3 to 8 percent slopes
CkC	3	C	Clarksburgh silt loam, 8 to 15 percent slopes
CmB	2	B	Clymer silt loam, 3 to 8 percent slopes
CmC	2	B	Clymer silt loam, 8 to 15 percent slopes
CmD	2	B	Clymer silt loam, 15 to 25 percent slopes
CoB	3	C	Cooksport silt loam, 3 to 8 percent slopes
CoD	3	C	Cooksport silt loam, 15 to 25 percent slopes
CuB	2	B	Culleoka silt loam, 3 to 8 percent slopes
CuC	2	B	Culleoka silt loam, 8 to 15 percent slopes
CuD	2	B	Culleoka silt loam, 15 to 25 percent slopes
CwB	2	B	Culleoka-Weikert shaly silt loam, 3 to 8 percent slopes
CwC	2	B	Culleoka-Weikert shaly silt loam, 8 to 15 percent slopes
CwD	2	B	Culleoka-Weikert shaly silt loam, 15 to 25 percent slopes
DoB	3	C	Dormont silt loam, 2 to 8 percent slopes
DoC	3	C	Dormont silt loam, 8 to 15 percent slopes
DoD	3	C	Dormont silt loam, 15 to 25 percent slopes
DoE	3	C	Dormont silt loam, 25 to 35 percent slopes
Du	2	B	Dumps, coal waste
Dw	3	C	Dumps, Industrial waste
ErB	3	C	Ernest silt loam, 3 to 8 percent slopes
ErC	3	C	Ernest silt loam, 8 to 15 percent slopes
ErD	3	C	Ernest silt loam, 15 to 25 percent slopes
EvB	3	C	Ernest-Vandergrift silt loam, 3 to 8 percent slopes
EvC	3	C	Ernest-Vandergrift silt loam, 8 to 15 percent slopes
EvD	3	C	Ernest-Vandergrift silt loam, 15 to 25 percent slopes
GlB	3	C	Gilpin silt loam, 3 to 8 percent slopes
GlC	3	C	Gilpin silt loam, 8 to 15 percent slopes
GlD	3	C	Gilpin silt loam, 15 to 25 percent slopes
GmD	3	-	-
GnB	3	-	-
GpB	3	C	Gilpin-Upshur silt loam, 3 to 8 percent slopes
GpC	3	C	Gilpin-Upshur silt loam, 8 to 15 percent slopes

Table 6-2 (continued)

GpD	3	C	Gilpin-Upshur silt loam, 15 to 25 percent slopes
GQF	3	C	Gilpin-Upshur complex, very steep
GrE	3	C	Gilpin-Vandergrift silt loam,slumped 15 to 35 percent slopes
GSF	3	C	Gilpin, Weikert, and Culleoka shaly silt loams, very steep
GuB	3	C	Guernsey silt loam, 3 to 8 percent slopes
GuC	3	C	Guernsey silt loam, 8 to 15 percent slopes
GuD	3	C	Guernsey silt loam, 15 to 25 percent slopes
GvB	3	C	Guernsey-Vandergrift silt loam, 3 to 8 percent slopes
GvC	3	C	Guernsey-Vandergrift silt loam, 8 to 15 percent slopes
GvD	3	C	Guernsey-Vandergrift silt loam, 15 to 25 percent slopes
GwC	3	-	-
GwD	3	-	-
Gx	3	C	Gullied Land
HaB	2	B	Hazleton loam, 3 to 8 percent slopes
HaC	2	B	Hazleton loam, 8 to 15 percent slopes
HaD	2	B	Hazleton loam, 15 to 25 percent slopes
HTE	2	B	Hazleton loam, steep
Hu	2	B	Huntington silt loam
LbB	4	D	Library silty clay loam, 3 to 8 percent slopes
LbC	4	D	Library silty clay loam, 8 to 15 percent slopes
LbD	4	D	Library silty clay loam, 15 to 25 percent slopes
Ln	3	C	Lindside silt loam
Ne	3	C	Newark silt loam
Qu	4	D	Quarries
Ph	2	B	Philo silt loam
RaA	3	C	Rainsboro silt loam, 0 to 3 percent slopes
RaB	3	C	Rainsboro silt loam, 3 to 8 percent slopes
RaC	3	C	Rainsboro silt loam, 8 to 15 percent slopes
RyB	2	B	Rayne silt loam, 2 to 8 percent slopes
RyC	2	B	Rayne silt loam, 8 to 15 percent slopes
SmB	3	C	Strip mines, 0 to 8 percent slopes
SmD	3	C	Strip mines, 8 to 25 percent slopes
SmF	3	C	Strip mines, 25 to 75 percent slopes
TaB	3	C	Tipson silt loam, 3 to 8 percent slopes
UaB	4	D	Upshur silty clay loam, 3 to 8 percent slopes
UaC	4	D	Upshur silty clay loam, 8 to 15 percent slopes
UB	4	D	Urban land
UCB	4	D	Urban land-Culleoka complex, gently sloping
UCD	4	D	Urban land-Culleoka complex, moderately steep
UCE	4	D	Urban land-Culleoka complex, steep
UGB	4	D	Urban land-Guernsey complex, gently sloping
UGD	4	D	Urban land-Guernsey complex, moderately steep
URB	4	D	Urban land-Rainsboro complex, gently sloping
URC	4	D	Urban land-Rainsboro complex, sloping

Table 6-2 (continued)

UWB	4	D	Urban land-Wharton complex, gently sloping
UWD	4	D	Urban land-Wharton complex, moderately steep
VcB	3	C	Vadergrift-cavode silt loam, 3 to 8 percent slopes
VcC	3	C	Vadergrift-cavode silt loam, 8 to 15 percent slopes
VcD	3	C	Vadergrift-cavode silt loam, 15 to 25 percent slopes
W	4	D	Water
WEF	3	C/D	Weikert-Rock outcrop complex, very steep
WhB	3	C	Wharton silt loam, 2 to 8 percent slopes
WhC	3	C	Wharton silt loam, 8 to 15 percent slopes
WhD	3	C	Wharton silt loam, 15 to 25 percent slopes

Highlighted numbers were soils not found in the NEH. Because their coverage was a very small area they were classified as the Most common coverage Group C

6.2.2 Creating the Land Use Raster File

The land use raster file is a raster file which represents the type of land use for each cell in the watershed. It is created using a digital land cover survey that was conducted by the Southwest Pennsylvania Commission. Land use data for the entire US are also available from the NRCS. The land use data from the Southwest Pennsylvania Commission were chosen because locally collected data are usually more detailed than national data sets. Each type of land cover is reclassified by a digital land use number. Table 6-3 shows the land use types and the corresponding digital land use numbers.

Table 6-3 Digital Land Use Numbers

		LU_NUM
Cultivated Land:	without conservation treatment	**1**
	with conservation treatment	**2**
Pasture or range land:	poor condition	**3**
	good condition	**4**
Meadow:	good condition	**5**
Wood or forest land:	thin stand, poor cover, no mulch	**6**
	good cover	**7**
Open spaces, lawns, parks, golf courses, cemeteries, etc.	good condition (75% grass cover)	**8**
	fair condition (50-75% grass cover)	**9**
Commercial and business areas	(85% impervious)	**10**
Industrial districts	(72% impervious)	**11**
Residential	1/8 ac or less (65% impervious)	**12**
	1/4 ac (38% impervious)	**13**
	1/3 ac (65% impervious)	**14**
	1/2 ac (65% impervious)	**15**
	1 ac (65% impervious)	**16**
Paved parking lots, roofs, driveways, etc.		**17**
Streets and roads:	Paved with curbs and storm sewers	**18**
	Gravel	**19**
	Dirt	**20**
Bare soil		**21**
Water		**22**

6.2.3 Generating the Curve Number File

The curve number file is a file which represents the curve number values for each cell in the watershed. The curve number program creates this file by the use of a table lookup. The program examines the soil and land use attributes associated with each cell in the watershed. It the program contains a series of if statements such that given a particular digital hydrologic soil group number and digital land use number for a cell the program assigns a curve number to the corresponding cell in the curve number file. A text file is also created showing the number of cells for each curve number and the calculated composite curve number for each sub-watershed as well as for the total watershed. Table 6-4 shows how the curve numbers are calculated from the digital hydrologic soil group number and digital land use number.

Table 6-4 Calculation of Curve Numbers

	Hydrologic Soil Group			
	A	**B**	**C**	**D**
	1	**2**	**3**	**4**
LU_NUM	**Curve Number**			
1	72	81	88	91
2	62	71	78	81
3	68	79	86	89
4	39	61	74	80
5	30	58	71	78
6	45	66	77	83
7	25	55	70	77
8	39	61	74	80
9	49	69	79	84
10	89	92	94	95
11	81	88	91	93
12	77	85	90	92
13	61	75	83	87
14	57	72	81	86
15	54	70	80	85
16	51	68	79	84
17	98	98	98	98
18	98	98	98	98
19	76	85	89	91
20	72	82	87	89
21	77	86	91	94
22	100	100	100	100

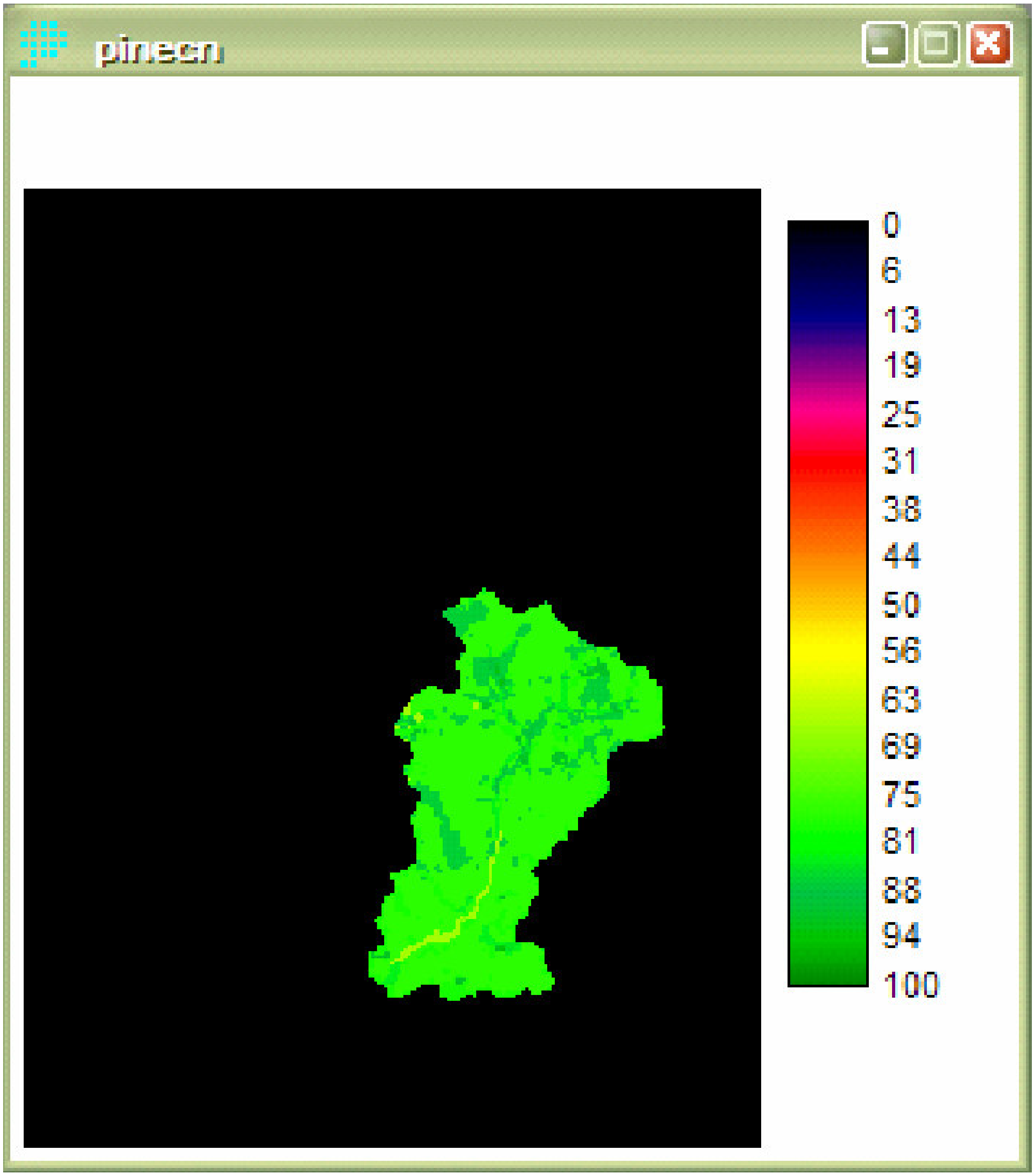


Figure 6-6 Little Pine Creek Curve Number File

6.3 Generating the Isochrones File

The isochrones raster file is created by the ISOCHRONE program. This program works by reading the first cell in the watershed file to determine if the particular cell falls within the watershed or sub-watershed. If the cell in the watershed raster file contains a 0 it is not in the watershed. The cell is then skipped and the next cell is evaluated. When the program finds a cell within the watershed the program then reads from the stream file to determine if the flow from that cell to the next cell is overland flow or channel flow and reads from the direction file to determine the distance. The program stores the distance of either overland or channel flow in the appropriate variable. In the travel from one cell to the next cell there are two possible distances. One is that of a horizontal or vertical flow path (in the plane of the image) which is equal to the resolution. In this case that would be 30 meters. The other distance is that of a diagonal flow path. This is equal to the square root of two times the square of the resolution. The travel time is then calculated by either the formula for overland or channel flow using the slope, curve number, and length of the flow path. The calculated travel time is stored as a cumulative travel time variable for the first cell in the flow path. Next the direction of flow is read from the direction file to determine the next cell in the flow path. That cell is evaluated in the same manner and again adding the travel time to the cumulative travel time. This is repeated until the outlet is reached. Once the outlet is reached the cumulative travel time is written to the first cell in the flow path. The program then moves on to the next cell in the watershed and follows the flow path to the outlet again calculating the travel time. If there are several sub-watersheds the program begins with the highest numbered sub-watershed and creates the isochrones separately for each sub-watershed one at a time and writing it to the isochrone file. When the isochrone program is finished with a sub-watershed it writes descriptive data about the sub-watershed to the watershed data file. The watershed data file is a text file. An example of the data file can be seen in Table 6-4. The isochrone program continues by moving on to the next sub-watershed until all of the watersheds have been processed.

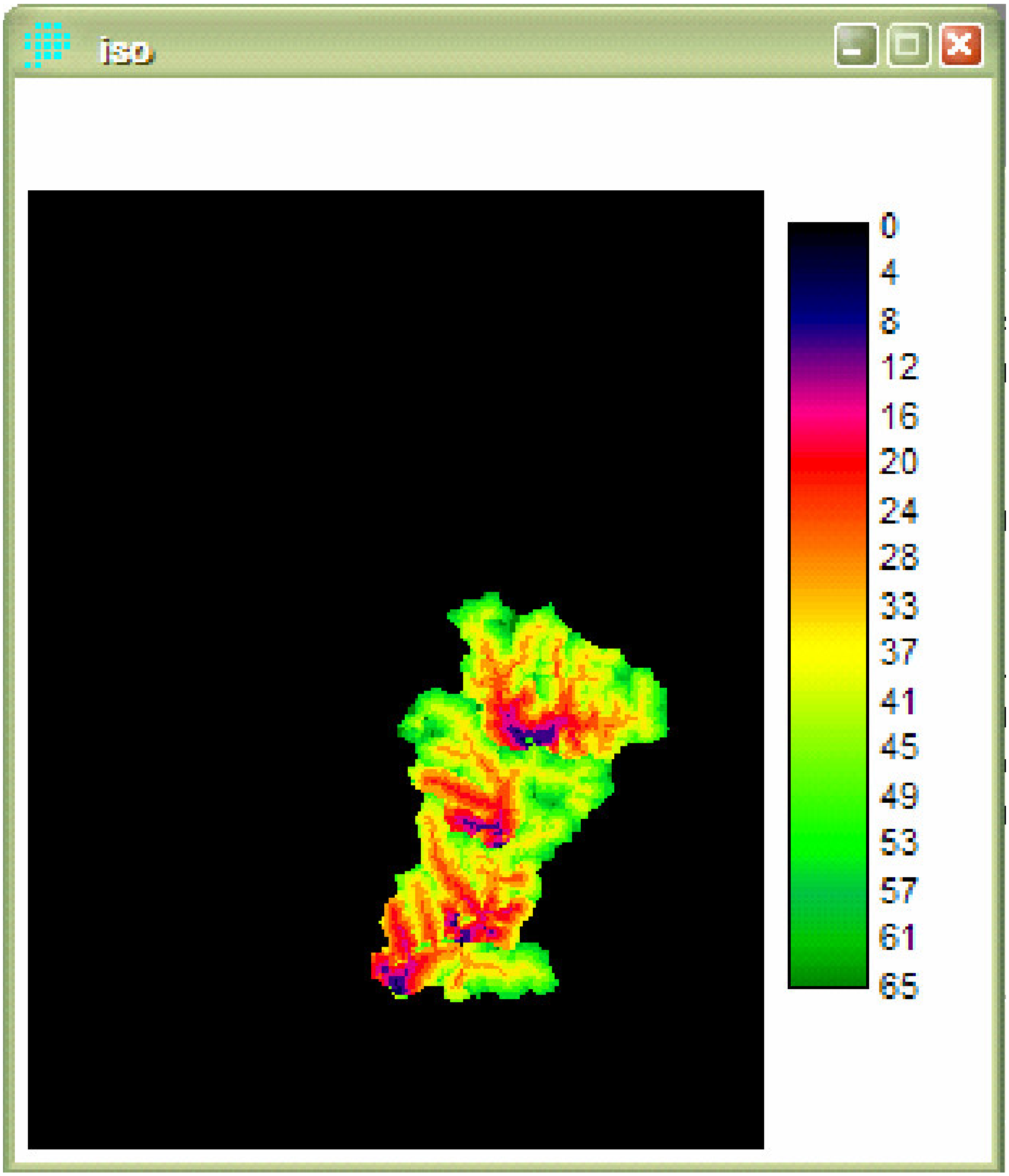


Figure 6-7 Isochrone File

Table 6-5 Data File

WATERSHED NUMBER 6
NUMBER OF CELLS = 2414
TC: 70.0
LONGEST FLOWPATH = 2845.22 METERS
TC CHANNEL: 37.4
LENGTH OF MAINSTREAM = 2333.09 METERS
AVERAGE OVERLAND SLOPE: .09947267
AVERAGE CHANNEL SLOPE: .02525495

6.4 Generating Incremental Precipitation Raster Files

The precipitation raster files are a representation of the data obtained from the Three Rivers Wet Weather Demonstration Program in a GIS raster image format. The data must first be downloaded for the cells that fall within the watershed of interest. This is done by entering the cells for which the precipitation is to be downloaded. These data are copied into a text file and saved. The gridded precipitation program reads the data and writes them to a separate raster file for each time increment. This can be done for time increments as small as 15 minutes. The files are displayed with a color gradient showing the depth of precipitation over the specific area. These files are then used by the runoff program.

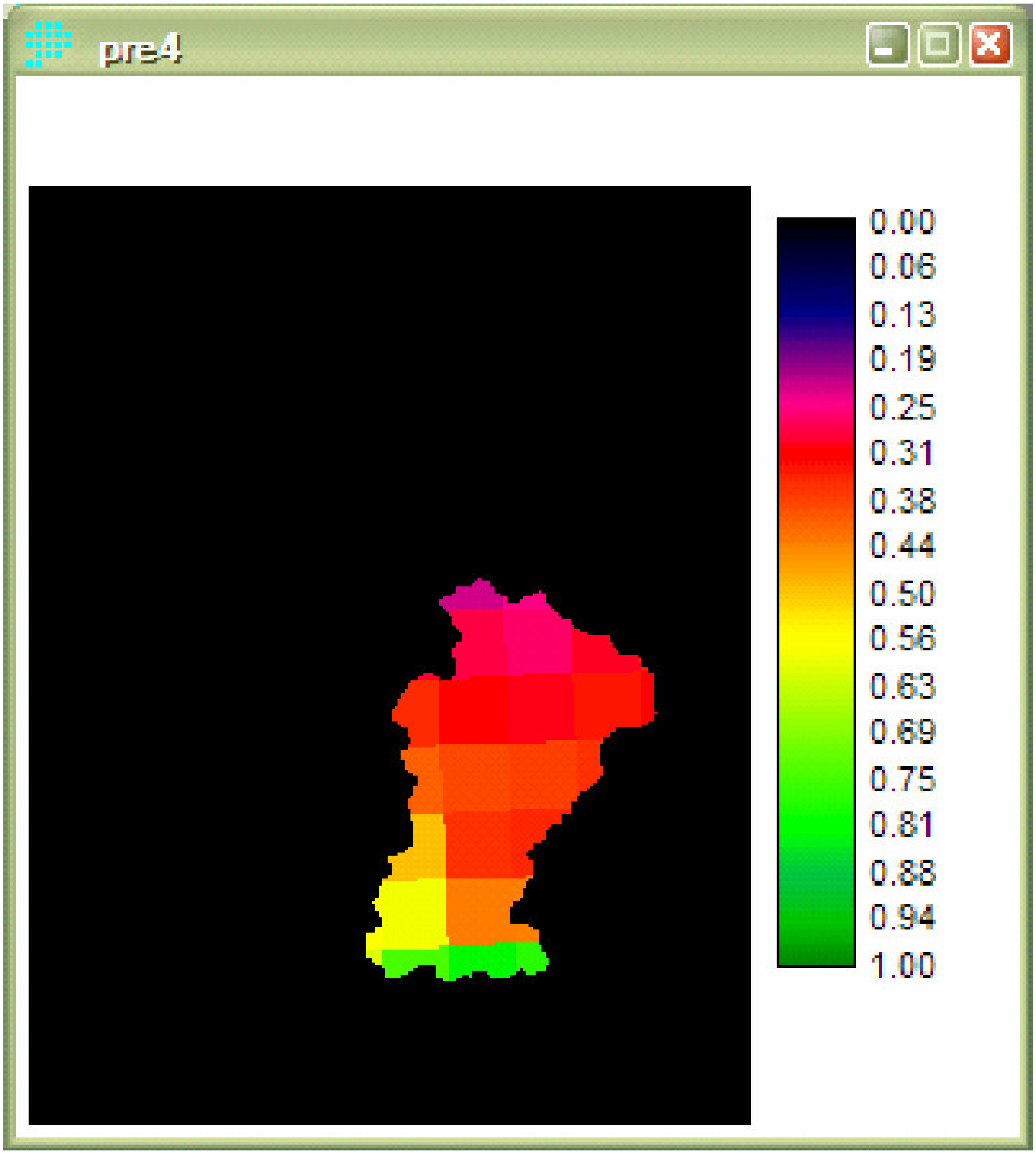


Figure 6-8 Incremental Precipitation Raster File

6.5 Generating Runoff Time Series

Before this research began, the runoff was calculated in the isochrone program. The isochrone program used only one precipitation file with a single value of precipitation for each watershed. This meant the program could only be used with the storm total precipitation. The isochrone raster file was not used in the calculation of the runoff. It was only written for display purposes. Changes were made in the way that the program ran in order to incorporate the use of incremental precipitation. The formulas used for the calculation of the travel time in the isochrone program are not a function of the flow or time therefore the model assumes that the travel time does not change during the storm. This means that the isochrone file can be created one time only and then used for the generation of the runoff time series for each incremental precipitation. It is one of the more time consuming processes. A separate program for the calculation of the runoff was then written so that it could use the gridded incremental precipitation.

The runoff program written for this research generates the runoff by evaluating the cells in the precipitation file, isochrone file, and curve number file. The values of precipitation for each cell is read from the precipitation raster file and the SCS curve number method is applied using data from the curve number file to calculate the excess precipitation which is then converted to runoff and arranged in a time series. The isochrone file is used by the program to organize the runoff generated by each cell in the time series. The process is then repeated for the next precipitation time increment. These files are then converted to hydrographs of duration equal to that of the given time increment and combined to obtain the hydrograph for the entire storm. This is done individually for each sub-basin. The result is a hydrograph at each outlet in the watershed for the entire storm.

6.6 Simulating Basin Storage Using Reservoir Routing

As described in the Clark method the watershed storage can be accounted for by routing the excess precipitation or time area curve through a linear reservoir. This is done by the reservoir routing program. This program uses the linear routing equation to develop a hydrograph for the watershed or sub-watershed. The parameter would be the storage constant (K).

6.7 Channel Routing

If the watershed is broken into several sub-watersheds the hydrographs for the upper watersheds must be routed downstream to achieve a hydrograph occurring at the outlet of the entire watershed. The best way that was found to do this was to route the hydrograph at the uppermost outlet to the next downstream outlet and then add the routed hydrograph to the hydrograph for the watershed pertaining to that outlet. This is done until the final outlet is reached. The value used for the storage constant (K) in the channel routing is equal to the travel time along the reach in the mainstream. There are programs written for the routing of the hydrograph and for the adding of the routed hydrographs.

7.0 CASE STUDY

7.1 Study Area

The area chosen for the testing and calibration of this model was the Little Pine Creek watershed. The selection criteria for a study area were the following:

1. Small watershed within Allegheny County
2. Gridded radar rainfall data available for the watershed
3. Stream flow data available for the same time period as the rainfall data
4. No significant detention ponds or storage

The area of the Little Pine Creek watershed is 5.78 square miles. The location of the gage is Latitude 40º31'13" and Longitude 79º56'18". The datum of the gage is at 775.26 feet above sea level. The hydrologic unit is 05010009. The watershed is approximately 3,000 meters wide and 6,000 miles long. This watershed has a stream gage maintained by the USGS. The period of record is from October 1962 until the current year. The Little Pine Creek watershed is located in the northern portion of Allegheny County. It drains into Pine Creek which is a tributary of the Allegheny River.

The most common soil type in the Little Pine Creek watershed is GQF. A histogram of the soil types is shown in Figure 7-1. The most common hydrologic soil group is group C this can be seen from the histogram in Figure 7-2. The most common land use in the Little Pine

Creek watershed is forest. Figure 7-3 shows the land use histogram for the watershed. The average curve number which is representative of the hydrologic soil group and the land use is 77. Figure 7-4 shows a histogram of the curve numbers calculated for the basin.

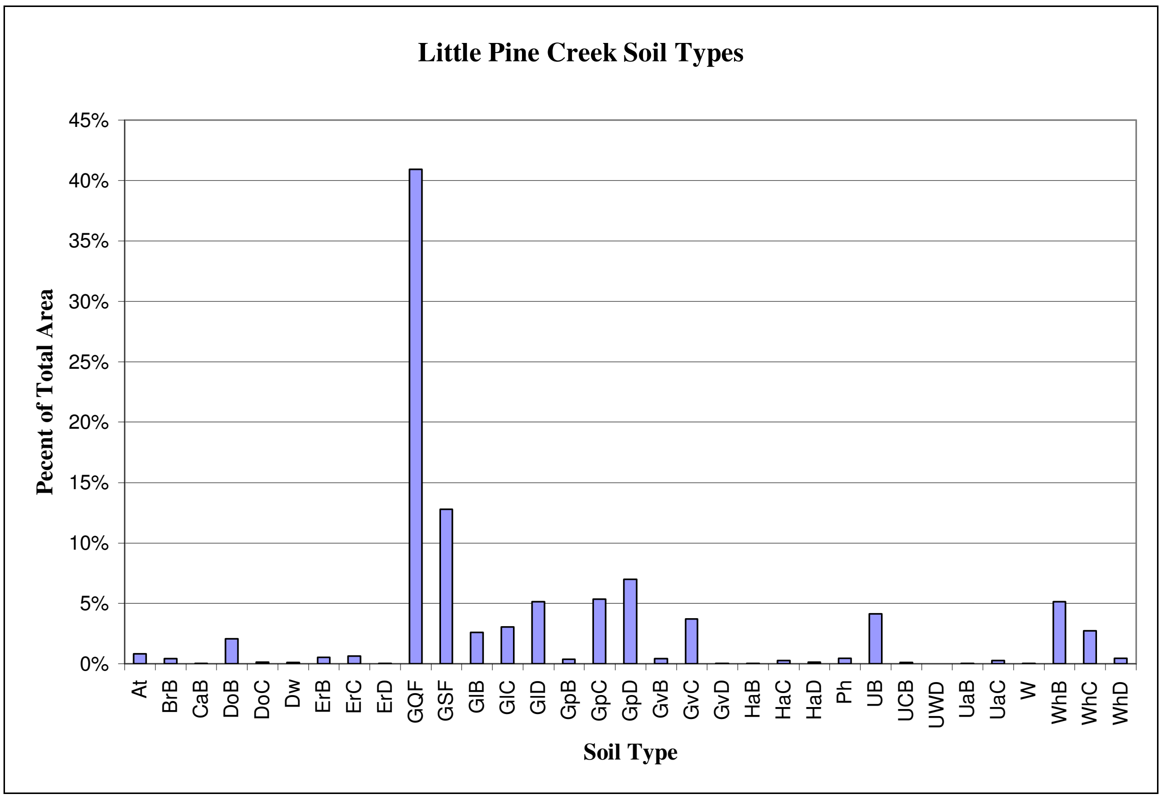


Figure 7-1 Soil Types Little Pine Creek

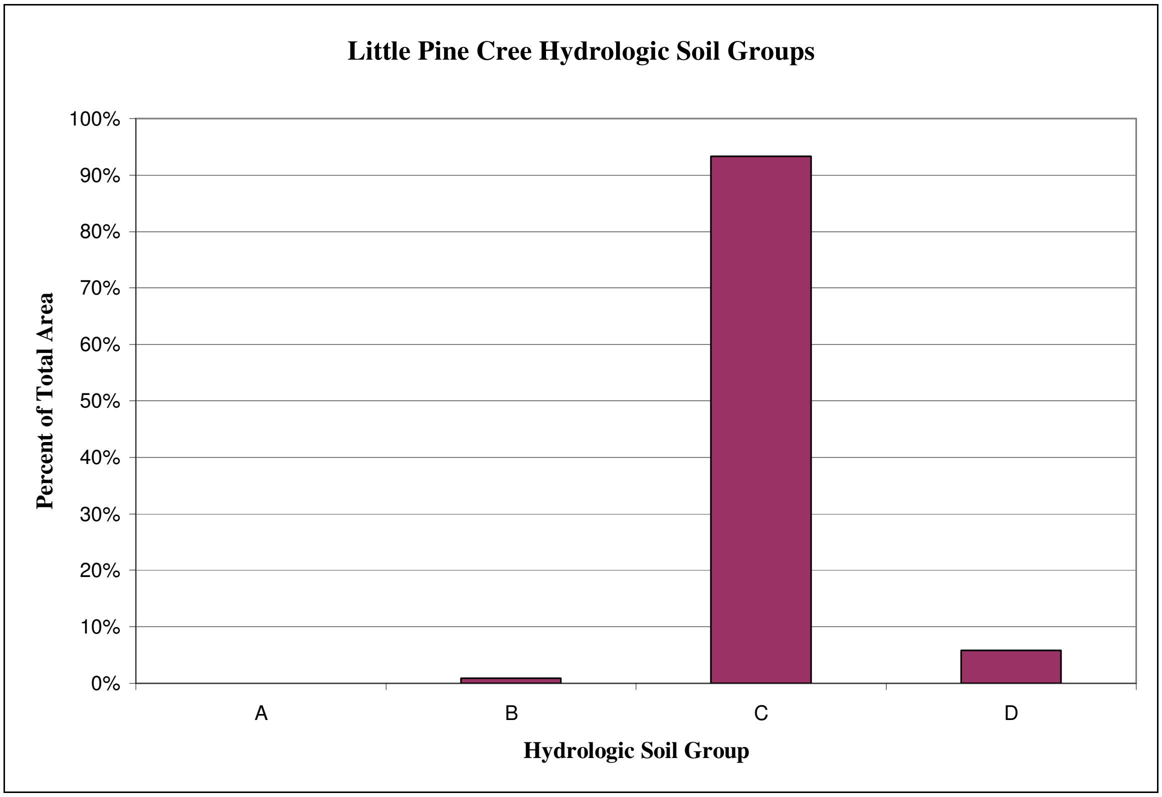


Figure 7-2 Hydrologic Soil Groups Little Pine Creek

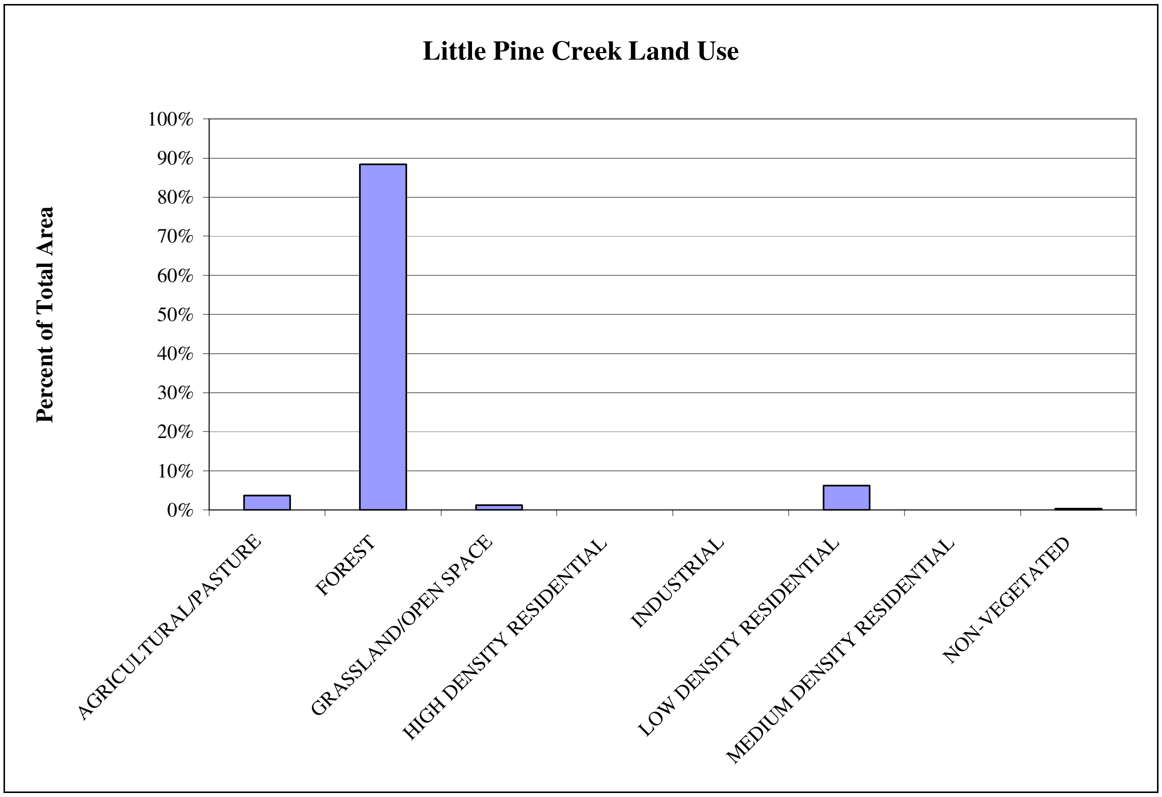


Figure 7-3 Land Use Little Pine Creek

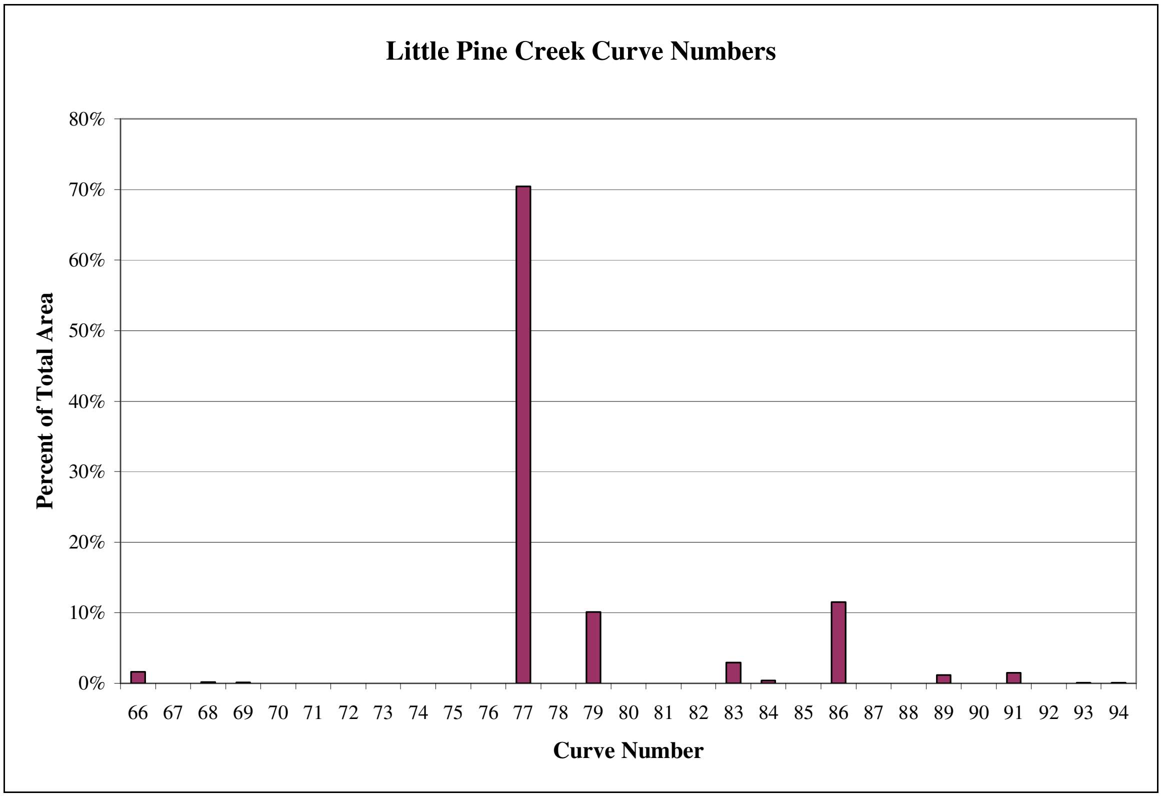


Figure 7-4 Curve Numbers Little Pine Creek

7.2 Event 1

7.2.1 Storm Description

The first event that was used in the verification and calibration of the model was a storm that occurred on August 17, 2002. The first detectable precipitation began at 1:00 AM and lasted for approximately 5 hours. This event was chosen because the storm was of a fairly short duration with considerable intensity. Figure 7-5 shows the maximum, minimum, and average 1 km^2 gridded precipitation measurements from the gage adjusted radar and the measurements taken from the nearest point gage. From this figure it is apparent the importance of the gridded radar rainfall data. In this storm there was a great deal of spatial and temporal variation in the rainfall measured by the radar. It can be seen that using an average value for the entire basin or using data from a point gage would give much different results. Figure 7-6 shows the incremental precipitation raster files generated for this event. These images show the movement of the storm. These images should again help to illustrate the importance of the spatially distributed rainfall measurements. As the storm begins to occur the northern portion of the watershed receives the most intense rainfall. Towards the end of the occurrence of the storm the southern portion is receiving the highest intensities. This obviously has an effect on shape of the hydrograph generated because of the time of travel of the runoff that is occurring. Therefore the predicted hydrograph generated by this model is not only representative of highly defined watershed characteristics but also the characteristics of spatially and temporally distributed meteorological data.

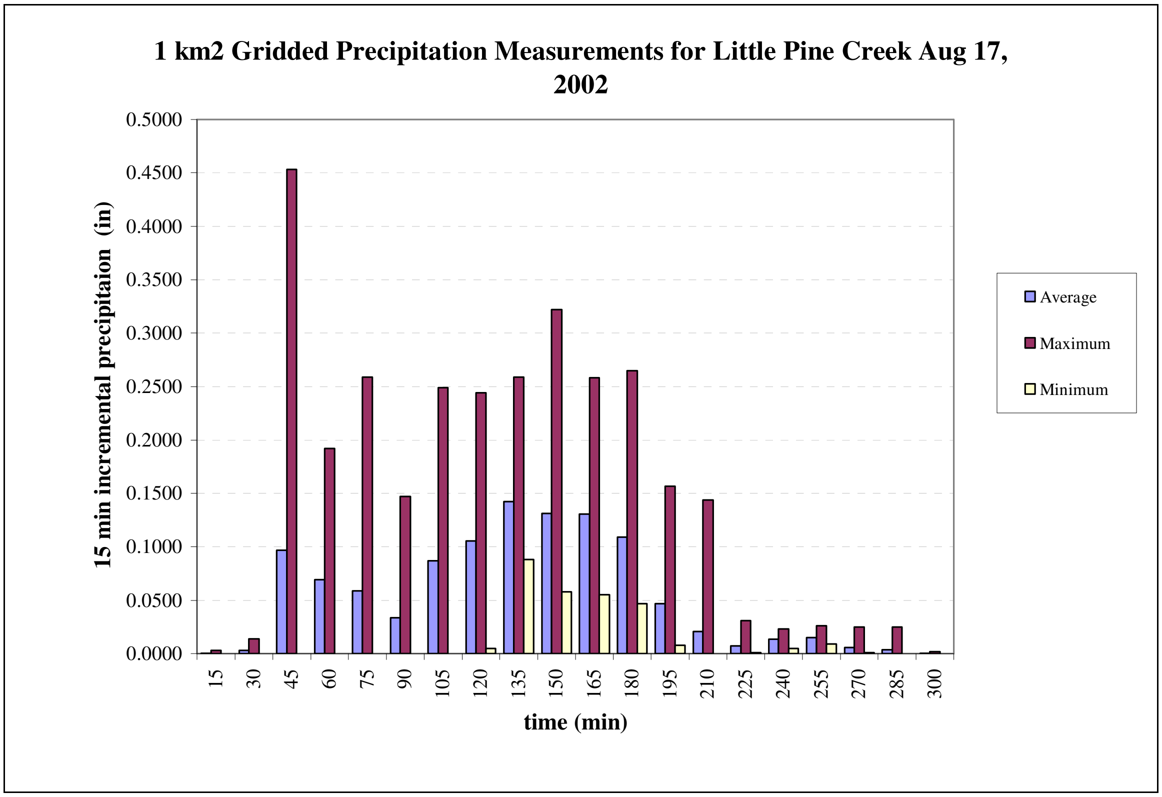
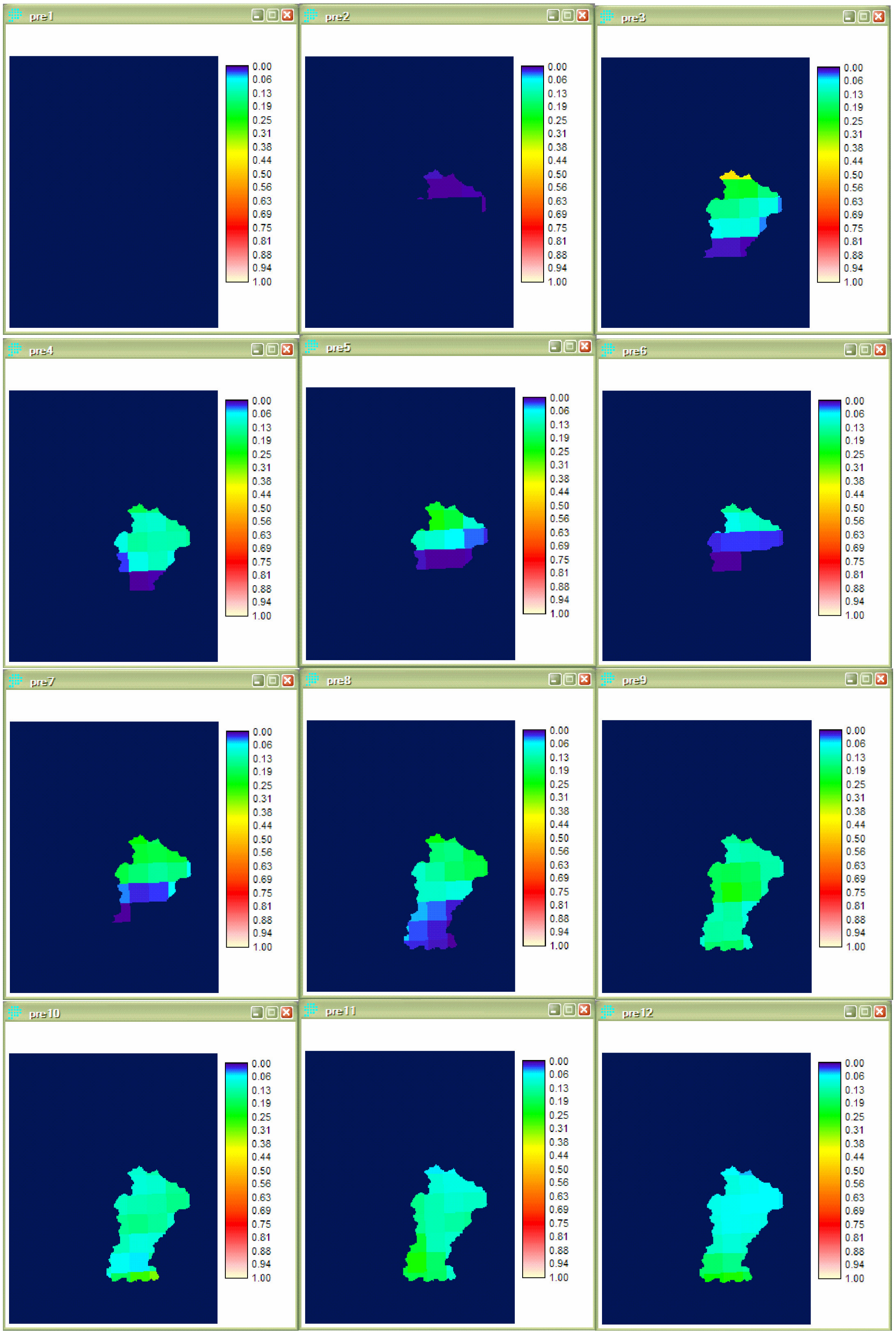


Figure 7-5 Variation in Precipitation


pre1
0.00 0.06 0.13 0.19 0.25 0.31 0.38 0.44 0.50 0.56 0.63 0.69 0.75 0.81 0.88 0.94 1.00
pre2
0.00 0.06 0.13 0.19 0.25 0.31 0.38 0.44 0.50 0.56 0.63 0.69 0.75 0.81 0.88 0.94 1.00
pre3
0.00 0.06 0.13 0.19 0.25 0.31 0.38 0.44 0.50 0.56 0.63 0.69 0.75 0.81 0.88 0.94 1.00
pre4
0.00 0.06 0.13 0.19 0.25 0.31 0.38 0.44 0.50 0.56 0.63 0.69 0.75 0.81 0.88 0.94 1.00
pre5
0.00 0.06 0.13 0.19 0.25 0.31 0.38 0.44 0.50 0.56 0.63 0.69 0.75 0.81 0.88 0.94 1.00
pre6
0.00 0.06 0.13 0.19 0.25 0.31 0.38 0.44 0.50 0.56 0.63 0.69 0.75 0.81 0.88 0.94 1.00
pre7
0.00 0.06 0.13 0.19 0.25 0.31 0.38 0.44 0.50 0.56 0.63 0.69 0.75 0.81 0.88 0.94 1.00
pre8
0.00 0.06 0.13 0.19 0.25 0.31 0.38 0.44 0.50 0.56 0.63 0.69 0.75 0.81 0.88 0.94 1.00
pre9
0.00 0.06 0.13 0.19 0.25 0.31 0.38 0.44 0.50 0.56 0.63 0.69 0.75 0.81 0.88 0.94 1.00
pre10
0.00 0.06 0.13 0.19 0.25 0.31 0.38 0.44 0.50 0.56 0.63 0.69 0.75 0.81 0.88 0.94 1.00
pre11
0.00 0.06 0.13 0.19 0.25 0.31 0.38 0.44 0.50 0.56 0.63 0.69 0.75 0.81 0.88 0.94 1.00
pre12
0.00 0.06 0.13 0.19 0.25 0.31 0.38 0.44 0.50 0.56 0.63 0.69 0.75 0.81 0.88 0.94 1.00

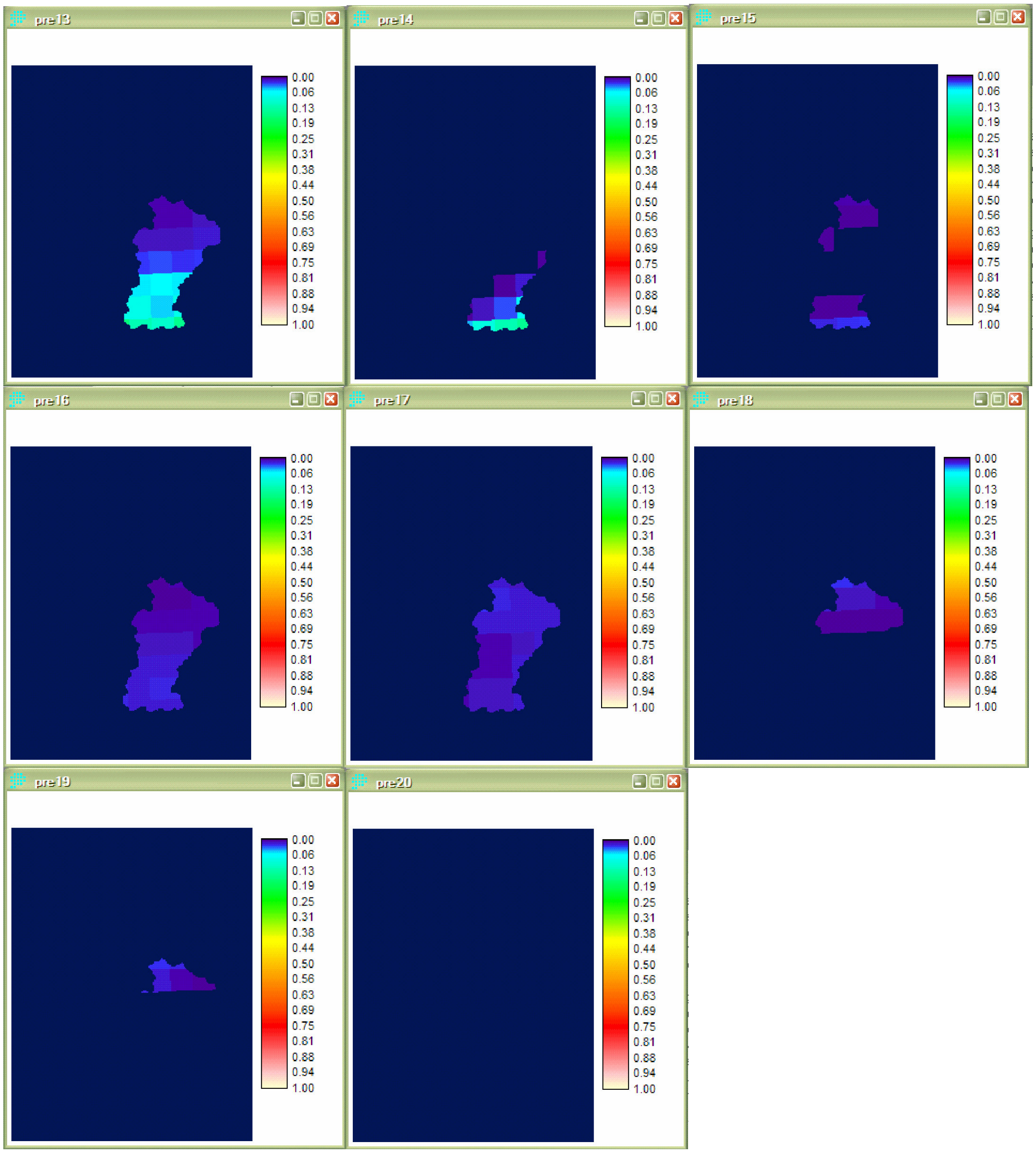


Figure 7-6 15 Minute Incremental precipitation for Event 1

7.2.2 Stream Conditions

The baseflow during the August 17, 2002 event studied was estimated using a straight line method. The baseflow was estimated at 0.48 cubic feet per second.

7.3 Event 2

7.3.1 Storm Description

The second event was a storm which occurred on July 29, 2002. This storm was much less intense than the first event. The storm for this event began at 11:00 PM and lasted approximately 1.5 hours. This storm was used to test the model for an event with a low runoff knowing that the curve number method has been criticized for being less accurate in this type of situation.

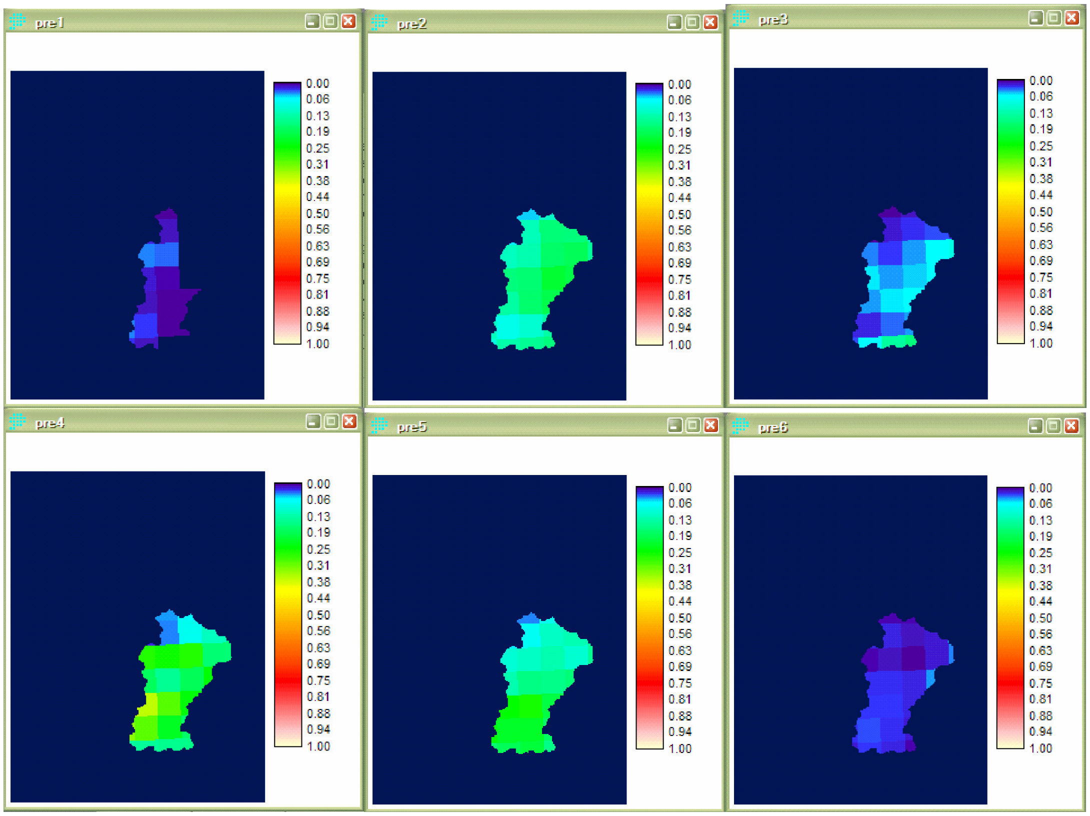


Figure 7-7 15 Minute Incremental Precipitation for Event 2

7.3.2 Stream Conditions

The base flow during the July 29, 2002 event studied was estimated using a straight line method. The baseflow was estimated at 1.8 cubic feet per second.

8.0 RESULTS

8.1 Accuracy of the Automatically Generated Watershed and Stream Files

In GIS based hydrologic modeling watershed delineation and generation of the stream network is a very important to the reliability of the model. The shape and area of the watershed and stream network automatically generated by this model were compared with data from other sources such as published data and data and data generated from other GIS based hydrologic software. Discrepancies that were found and the possible implications they would have on the model are discussed in this section. Possible causes for these differences are also examined.

Figure 8-1 shows the shapefile obtained from the PASDA website containing all of the watersheds in Allegheny County. The data for this file were taken from the Pennsylvania Gazetteer of Streams. The Little Pine Creek watershed is shown highlighted in Figure 8-1.

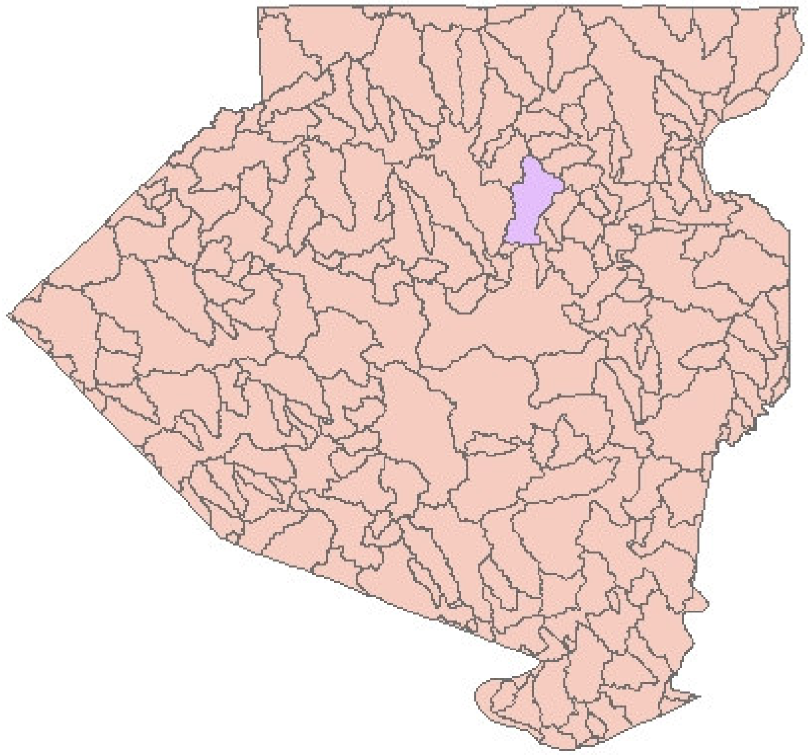


Figure 8-1 Allegheny County Watersheds

The area that the USGS lists for this watershed is 5.78 mi^2. The area found from the Allegheny County watershed shapefile in Figure 8-2 was 5.71 mi^2. The area calculated from the watershed created from by this model is 5.53 mi^2. It can be seen in Figures 8-2 and 8-3 that there is a slight difference in northwest portion of the watersheds.

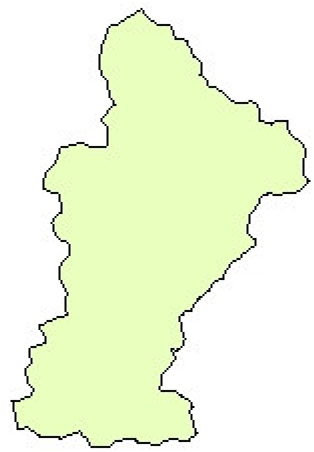


Figure 8-2 Watershed from PASDA Website

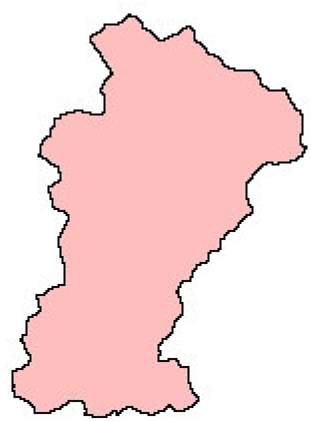


Figure 8-3 Watershed Generated from Watershed Program

A shapefile of the streams in Allegheny County was also obtained from the PASDA website. The data for this file were obtained from the Department of Transportation. The stream shapefile from the Department of Transportation were compared with the stream file generated by this model. It is noted that there are also some slight discrepancies in the same general location as was a problem in the watershed file. This can be seen in Figure 8-4. The watershed generated by the model is shown in red. The streams generated for this model are shown in light blue. They are overlaid onto the watershed in green and streams in dark blue from the published data. The streams in this file are more defined, but the general shape of the mainstreams is quite similar. The only real discrepancy is that the stream in the northwest section of the model's stream file from the model seems to have a break at the point where there is a discrepancy in the delineation of the watershed.

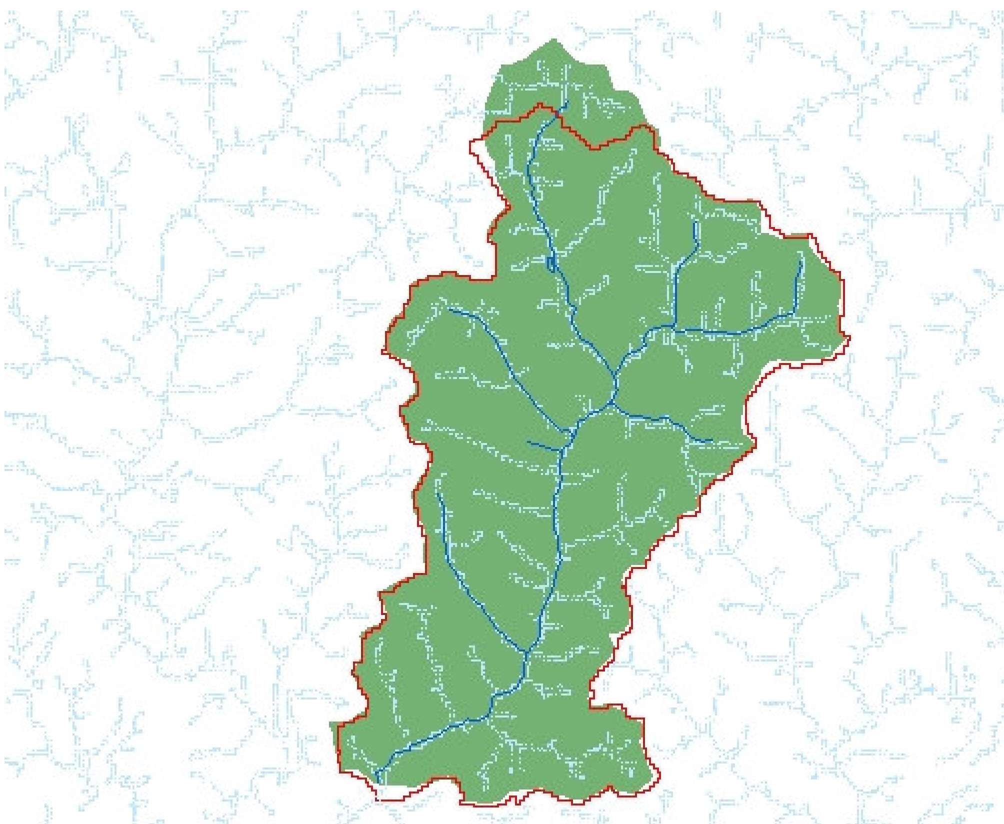


Figure 8-4 Comparison of Stream Generation

Because the watershed file is generated from the flow direction file which is created from the DEM data and the stream file is created directly from the corrected DEM, it is obvious that the discrepancy originates in the DEM itself or in the methods used to correct the DEM. To determine which of these was the cause of the error, the same uncorrected DEM was used in Geo-HMS to delineate the Little Pine Creek watershed. The correction of the DEM was done using Geo-HMS. The results from this can be seen in Figure 8-5. It can be seen that this delineation closely matches the published data. This would infer that the problem lies in the method of interpolation used by IDRISI to correct the DEM.

Figure 8-5 Watershed Generated Using Geo-HMS

The missing area in the watershed raster file in this model could cause the total volume of runoff generated by the storm to be slightly less than the actual. In the results seen with the hydrographs generated by the model the volume is indeed slightly lower, but it was still was within reason

8.2 Event 1: Watershed Subdivision

This section illustrates the hydrographs generated by the model using 5 sub-watersheds and 11 sub-watersheds. These results were obtained without using the reservoir routing that is normally suggested by the Clark method. The direct runoff that occurred at each outlet of the sub-watersheds was instead simply routed downstream from one outlet to the next until the most downstream outlet was reached. The value of K used in the Muskingum routing for each stream section was found from the isochrone map to determine the longest travel time through the reach. The value of X used in the Muskingum equation for the routing of the hydrographs through the reaches was 0.05.

8.2.1 Direct Runoff Hydrograph Using 5 Sub-watersheds

The watershed subdivision resulting in 5 sub-watersheds can be seen in Figure 8-6. The information from the data file for the 5 sub-watersheds can be seen in Table 8-1. The average size of the sub-watersheds was 1.1 mi^2. The isochrone file from this subdivision can be seen in Figure 8-7. Figure 8-8 shows the hydrograph generated with this particular subdivision for the Little Pine Creek watershed using rainfall data from the storm occurring on August 17, 2002. Table 8-2 compares the hydrograph parameters of the predicted hydrograph and the hydrograph from the USGS gage.

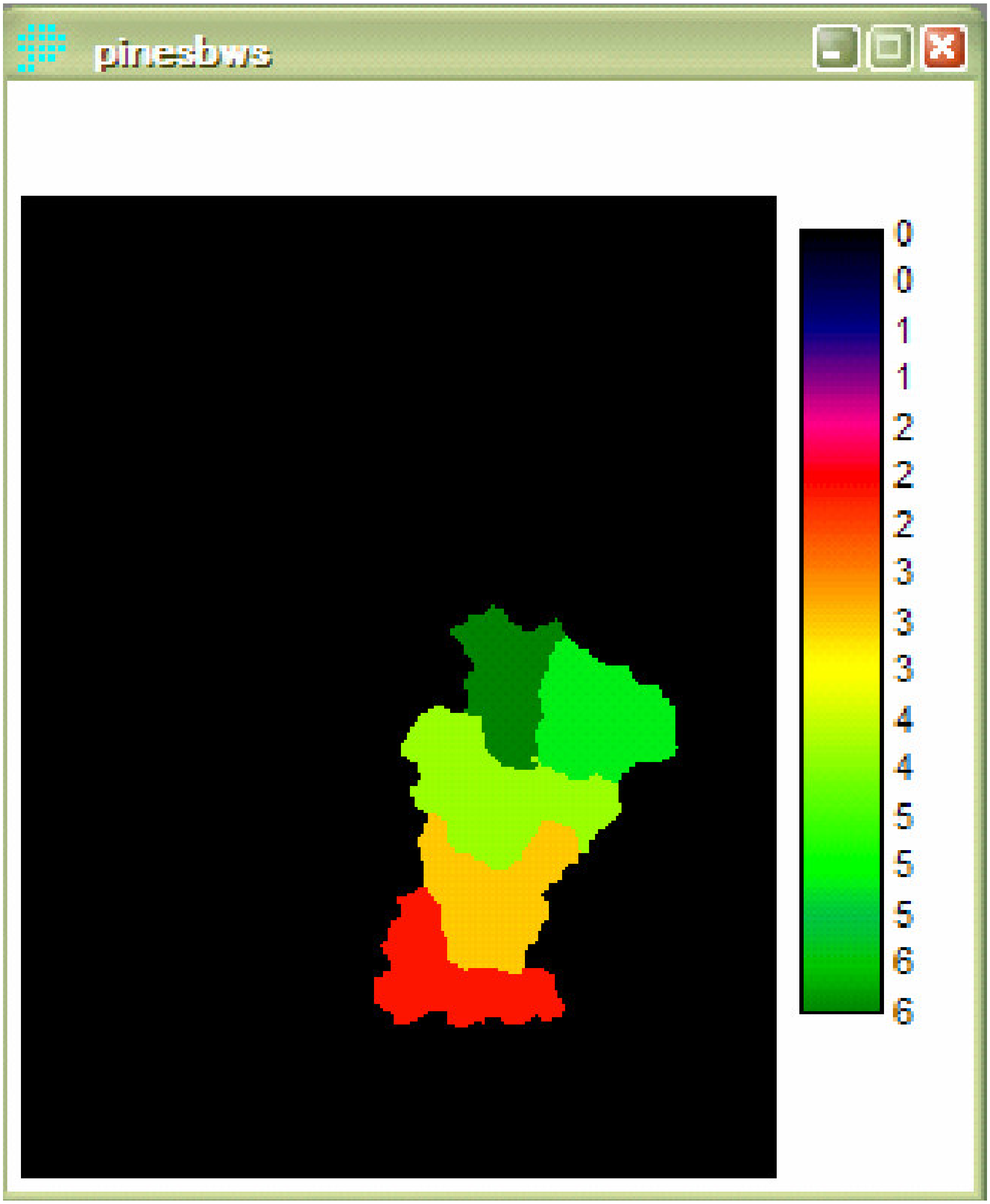


Figure 8-6 Watershed File

Table 8-1 Data File 5 Watersheds

Watershed Number	Number of Cells	Area (mi^2)	Longest Flowpath (m)	Time of Concentration (min)	Length of Mainstream (m)	Average Overland Slope	Average Channel Slope
6	2414	0.8	2845.2	70	2333.1	10%	3%
5	3177	1.1	2633.1	65	2313.4	9%	2%
4	4122	1.4	3404.0	90	2957.1	15%	2%
3	3234	1.1	3102.8	65	2758.2	15%	3%
2	2947	1.0	3207.4	65	2852.5	16%	3%

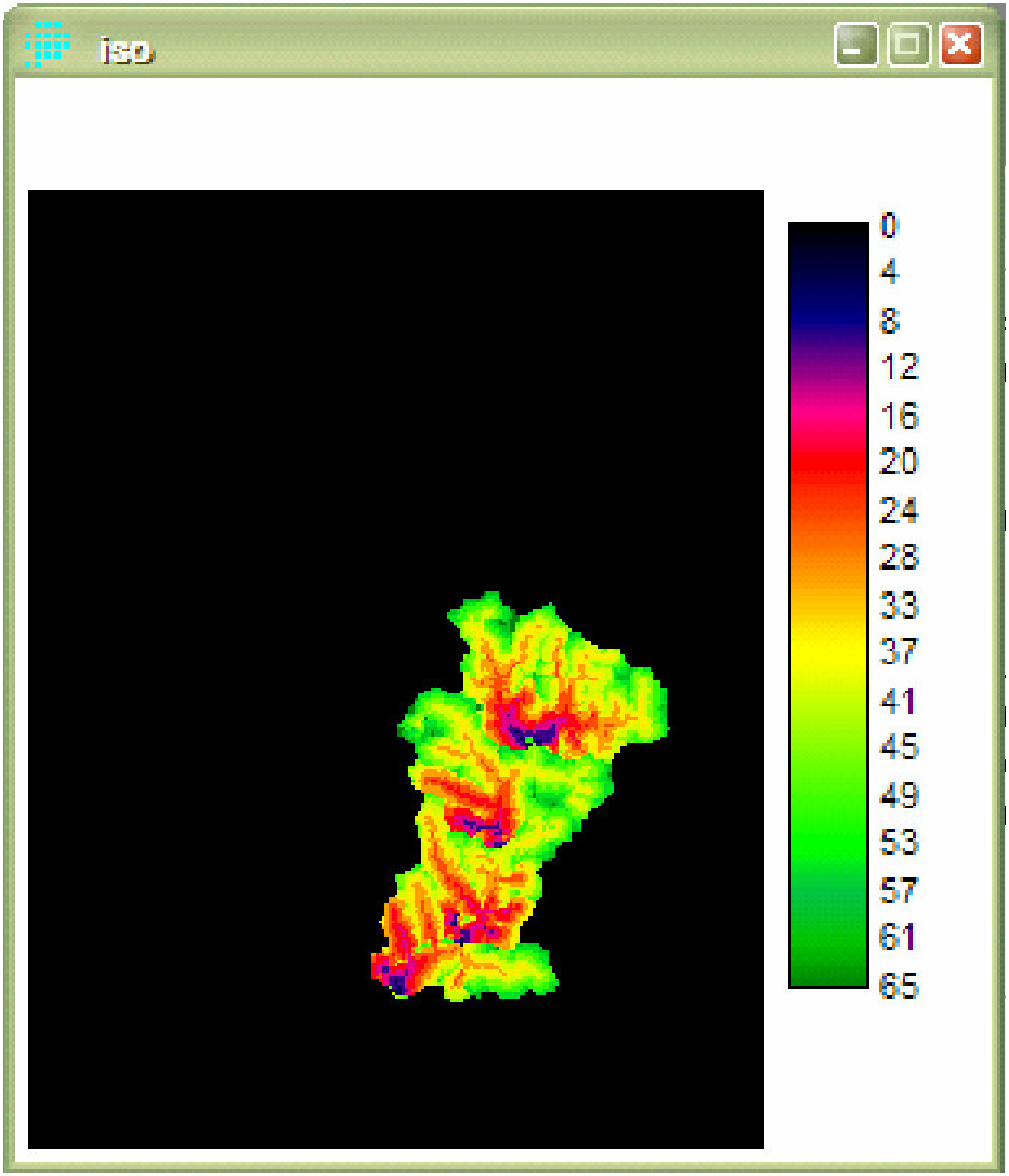


Figure 8-7 Isochrone File using 5 Watersheds

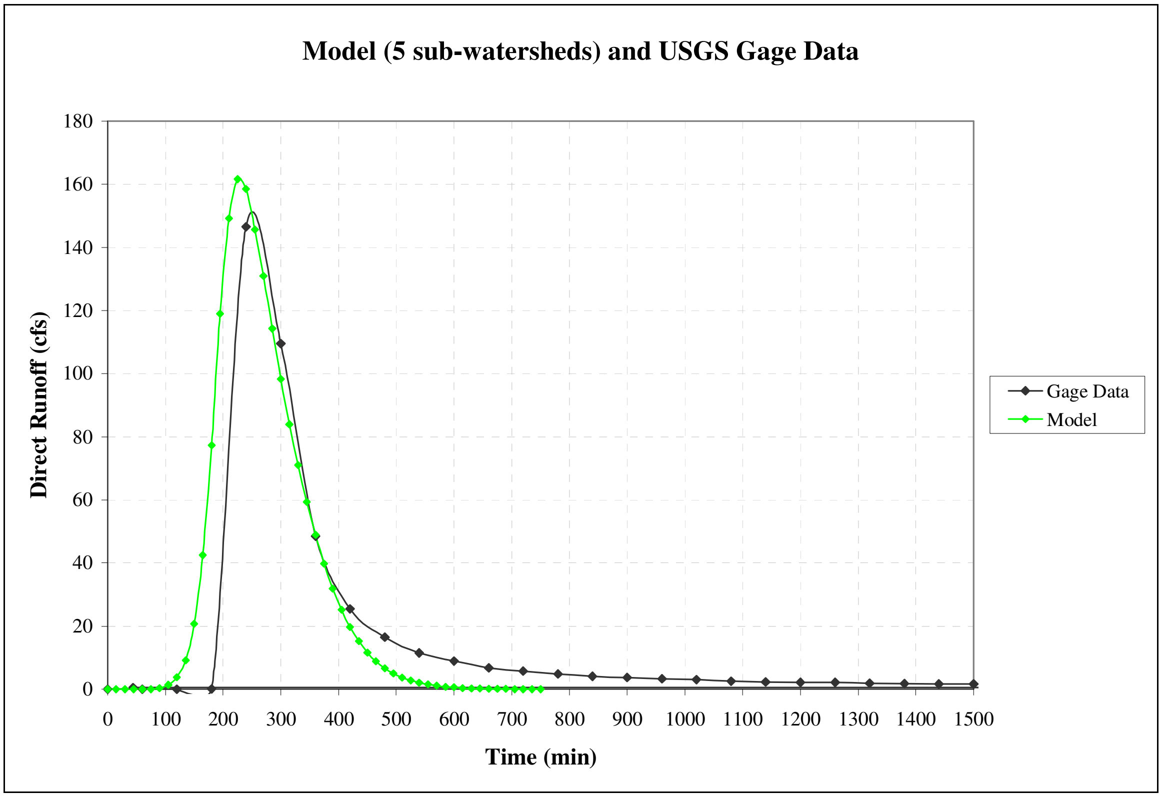


Figure 8-8 Predicted Hydrograph vs. DRH Estimated from USGS Gage

Table 8-2 Comparison of Hydrograph Parameters

	Gage Data	Model (5 subws)	Error
Total Volume (ft^3)	1561068	1506332	4%
Q_p (ft^3/s)	150	162	8%
T_p (s)	255	225	12%

8.2.2 Direct Runoff Hydrograph Using 11 Sub-watersheds

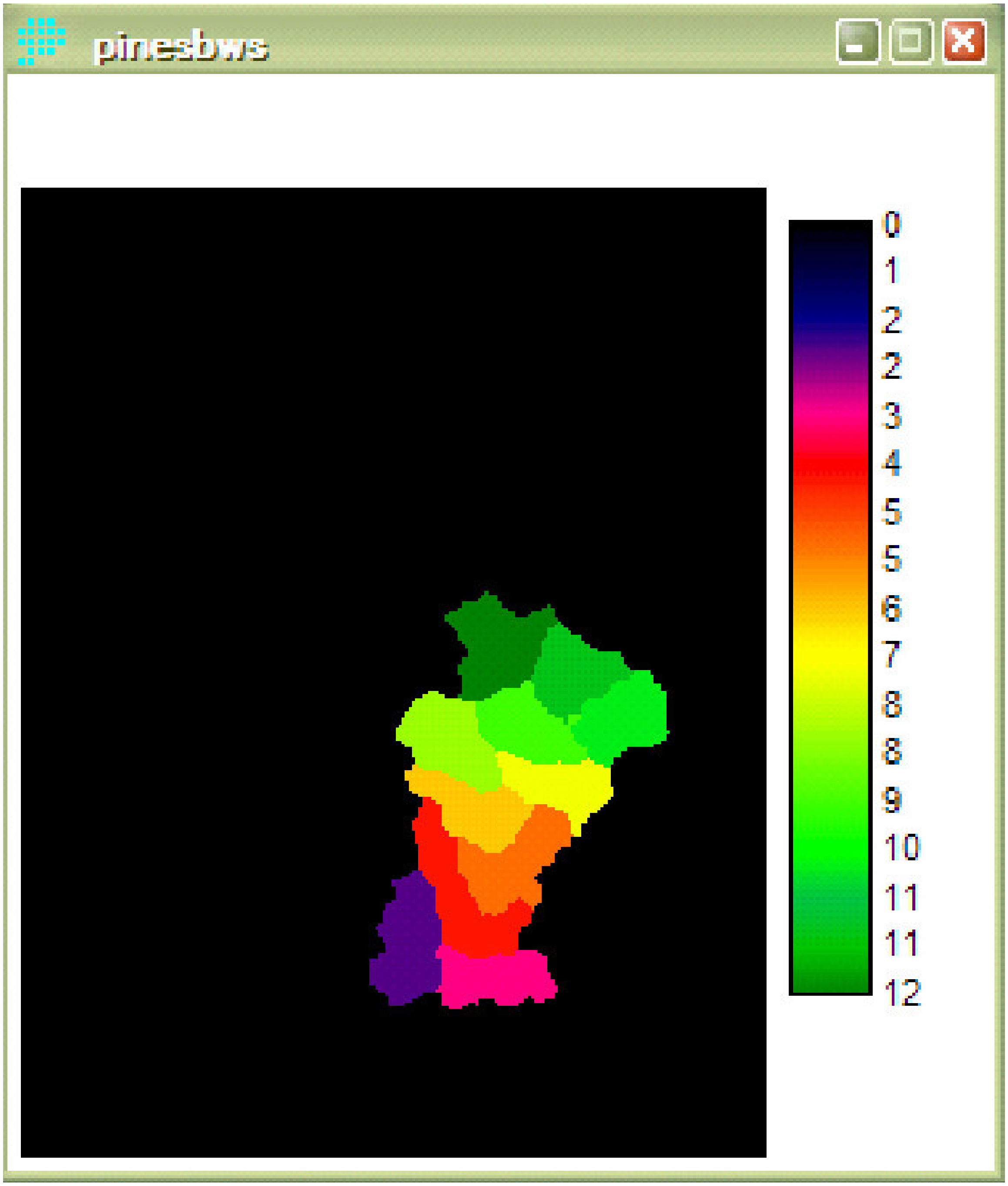


Figure 8-9 Watershed File

Table 8-3 Data File 11 Watersheds

Watershed Number	Number of Cells	Area (mi^2)	Longest Flowpath (m)	Time of Concentration (min)	Length of Mainstream (m)	Average Overland Slope	Average Channel Slope
12	1681	0.6	1788.8	60	1276.7	9%	2%
11	1292	0.5	1903.7	55	1474.3	9%	2%
10	1370	0.5	1878.8	55	1559.1	9%	2%
9	1334	0.5	1551.0	70	1211.5	11%	2%
8	1515	0.5	2344.6	80	1897.6	15%	4%
7	1195	0.4	1963.7	50	1596.4	13%	2%
6	1326	0.5	2230.7	55	1856.1	17%	4%
5	1563	0.6	2325.8	55	1981.3	14%	3%
4	1732	0.6	2859.8	60	2395.2	15%	4%
3	1292	0.5	2138.5	50	1783.7	14%	4%
2	1594	0.6	2562.8	55	2170.7	17%	4%

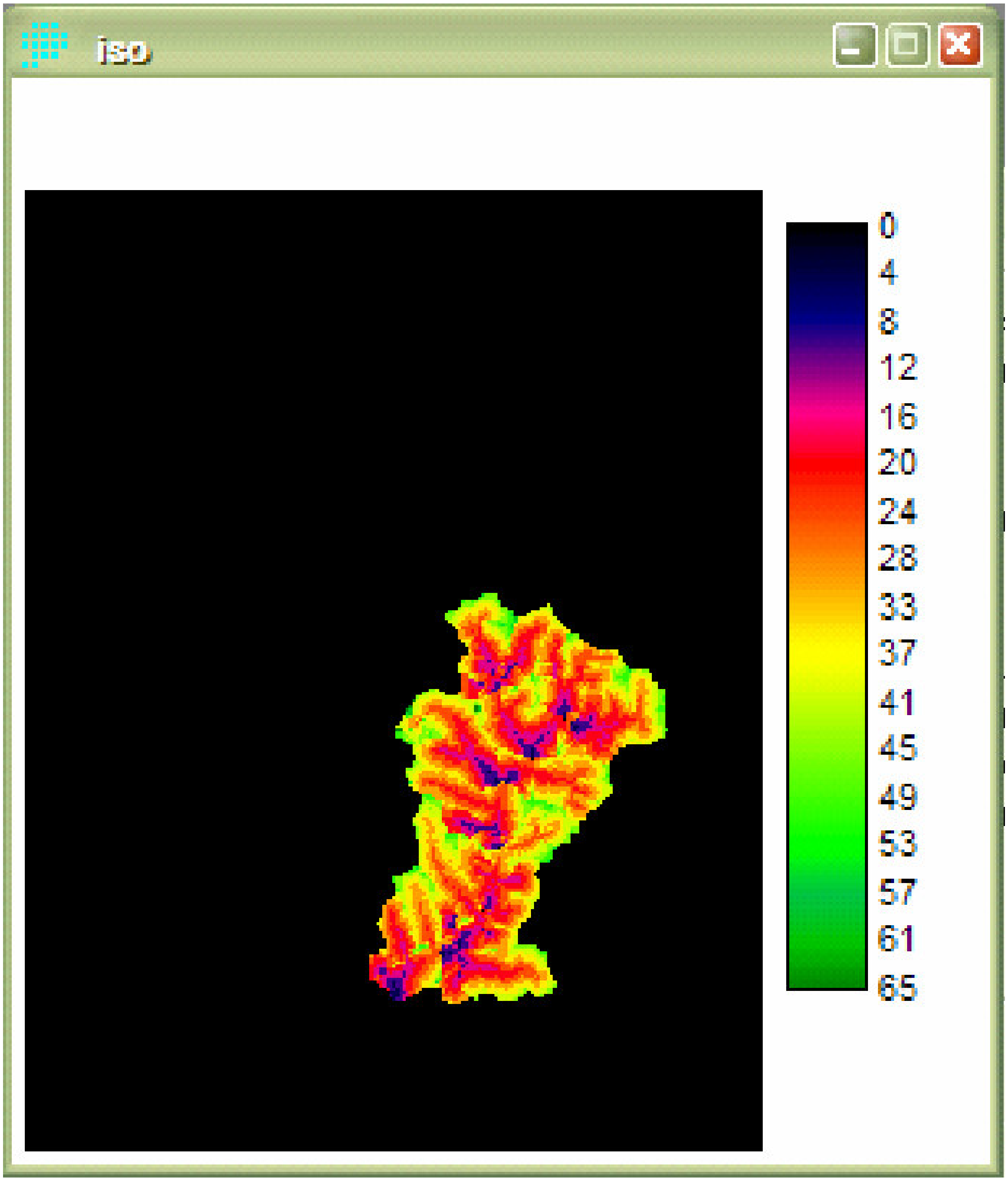


Figure 8-10 Isochrone File using 11 Watersheds

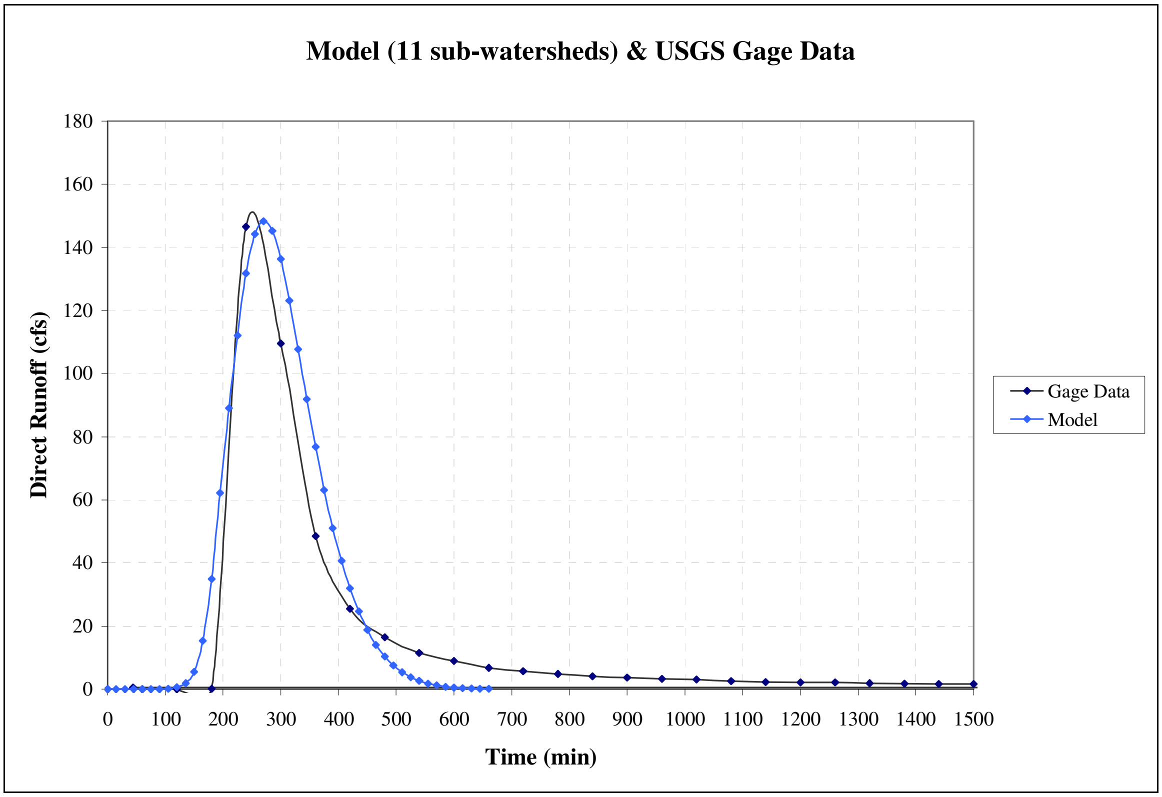


Figure 8-11 Predicted Hydrograph vs. DRH Estimated from USGS Gage

Table 8-4 Comparison of Hydrograph Parameters

	Gage Data	Model (11 subws)	Error
Total Volume (ft^3)	1561068	1535657	2%
Q_p (ft^3/s)	150	148	1%
T_p (s)	255	270	6%

8.3 Event 1: Effect of Isochrone Time Increment

This section shows hydrographs generated using 15 minutes time step when creating the isochrone file for the 11 sub-watersheds. This was attempted because 15 minute isochrones would allow for faster processing than 5 minute isochrones. The model could be run with 1 minute isochrones, but this would seem unreasonable given the precision of the data used and the equations for the calculation of the travel time in the model.

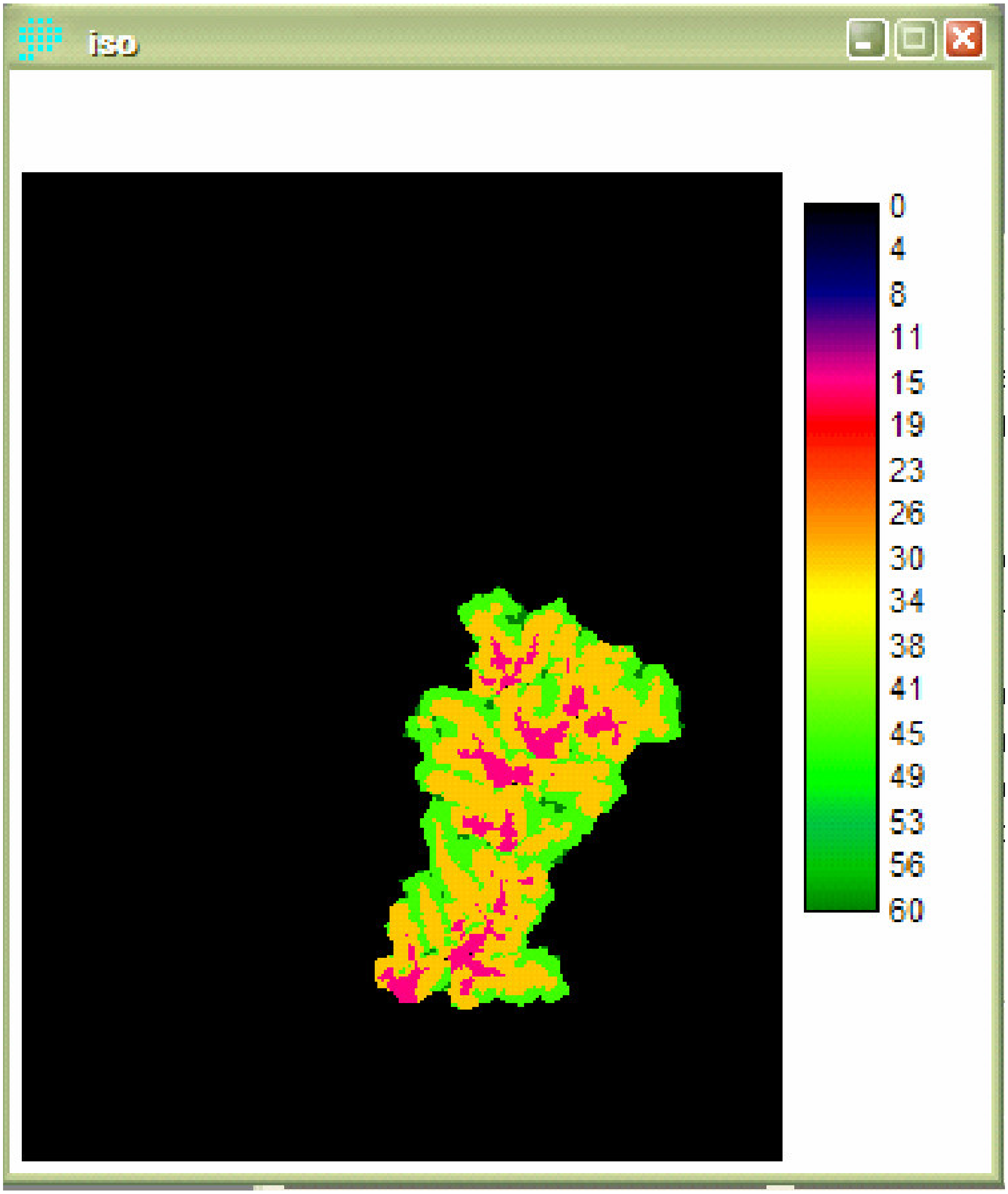


Figure 8-12 15 Minute Isochrone File

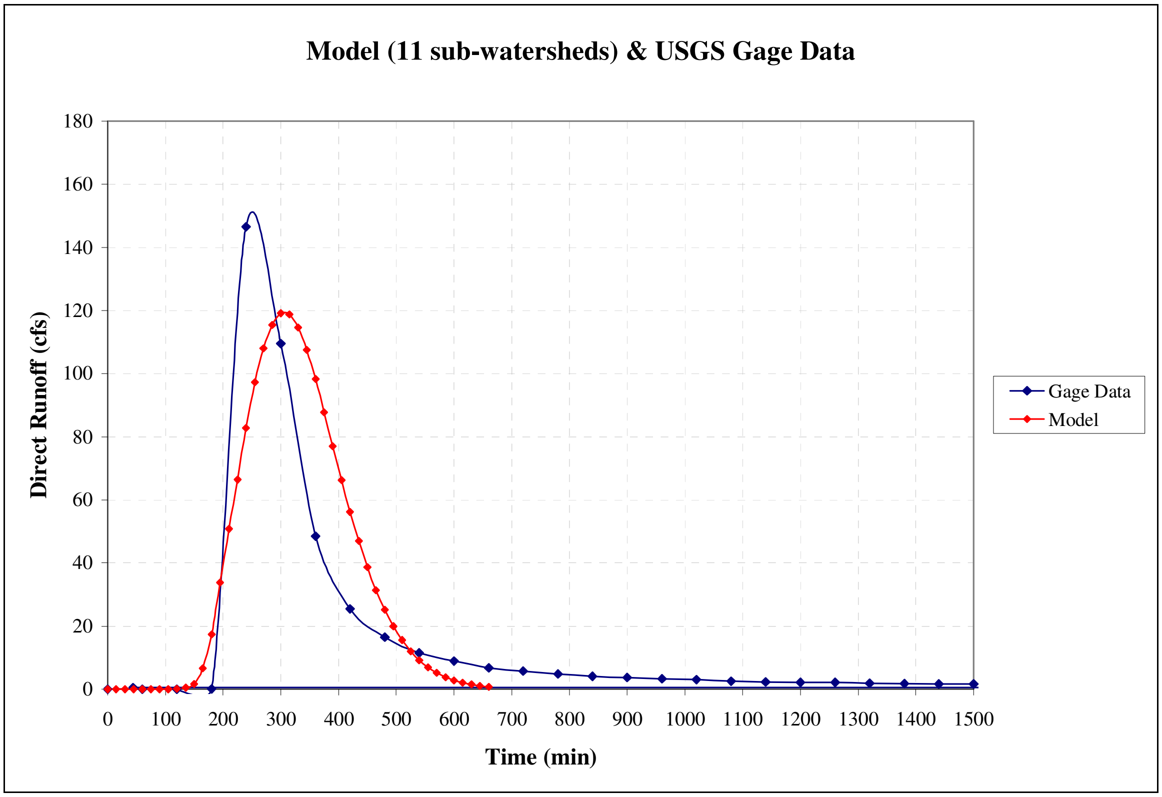


Figure 8-13 Predicted Hydrograph vs. DRH Estimated from USGS Gage

Table 8-5 Comparison of Hydrograph Parameters

	Gage Data	Model (11 subws)	Error
Total Volume (ft^3)	1561068	1485756	5%
Q_p (ft^3/s)	150	119	21%
T_p (s)	255	300	18%

8.4 Event 1: Effect of the Wedge Parameter (X) in Muskingum Routing

This section shows the hydrograph generated using a value of 0.15 for X in the Muskingum routing. The 11 sub-watershed division was used.

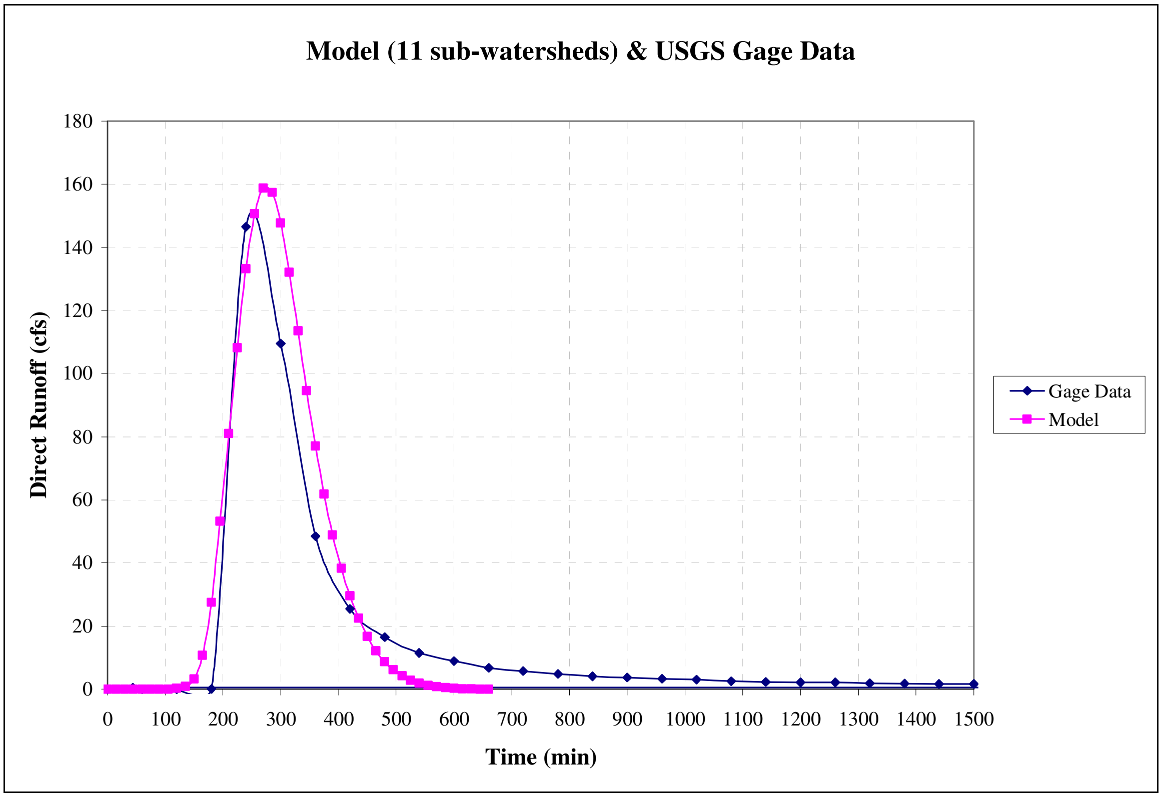


Figure 8-14 Predicted Hydrograph vs. DRH Estimated from USGS Gage

Table 8-6 Comparison of Hydrograph Parameters

	Gage Data	Model (11 subws)	Error
Total Volume (ft^3)	1561068	1535810	2%
Q_p (ft^3/s)	150	159	6%
T_p (s)	255	270	6%

8.5 Event 2: Low Flow Event

This event was chosen to test the model against a storm with low intensity of rainfall.

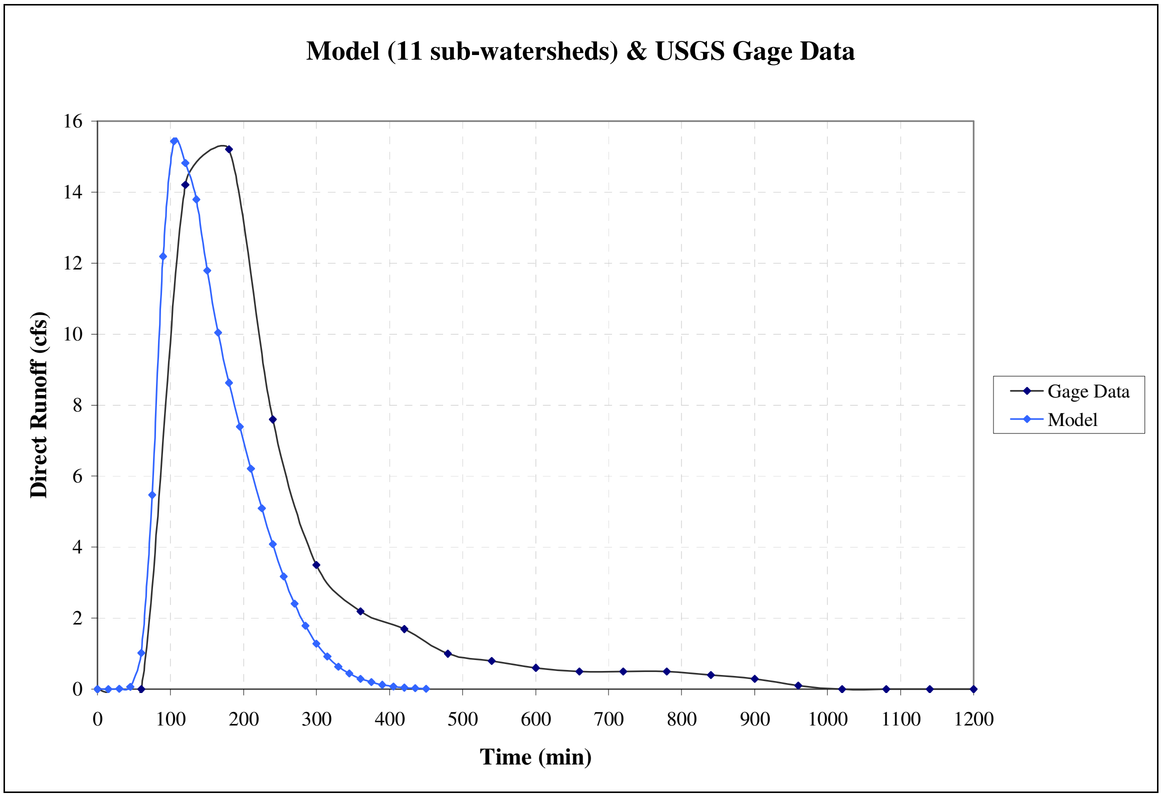


Figure 8-15 Predicted Hydrograph vs. DRH Estimated from USGS Gage

Table 8-7 Comparison of Hydrograph Parameters

	Gage Data	Model (11 subws)	Error
Total Volume (ft^3)	176760	114738	35%
Q_p (ft^3/s)	15	15	1%
T_p (s)	180	105	42%

9.0 CONCLUSIONS

9.1 Watershed Subdivision and Basin Routing

The Clark Unit Hydrograph Method suggests that in order to simulate the storage effects of the watershed, linear routing should be applied to the time area curve. It was found in this study, by breaking the watershed into sub-watersheds, that when the sub-watershed size becomes small enough the linear routing that was applied in previous studies to the runoff or the time area histogram for the entire watershed was no longer necessary. The hydrograph shape instead came from the channel routing imposed on the hydrographs as they were routed through the mainstream to the main outlet for the entire watershed using the Muskingum method.

The value of K in the Muskingum equation was easily determined from the isochrone map. The model worked best when using 5 minute isochrones. It is believed that the fifteen minute isochrones gave too high of a K value when routing the hydrographs downstream.

The hydrograph more closely matched the USGS data when using a low value for X such as 0.05 in the Muskingum routing. It is shown that by using a value of 0.15 that there is not much of a change in the predicted hydrograph except for a slight increase in the value of the Q_p. Error in the estimation of x has little effect in comparison with error in the estimation of the value of K used previously in the linear reservoir routing.

9.2 Accuracy of the Model

Before making a comparison of the model's predicted direct runoff hydrographs with the direct runoff hydrographs estimated from the USGS streamgage there are several things that should be taken into consideration. The first thing to consider is the method by which the measurements were taken, the assumptions that were made, and the reliability of that method to make accurate measurements of streamflow. It is well known that a streamgage works by making measurements of the stage of the stream at a given section. A stage-discharge curve is used to convert the measured stage to a discharge. The stage-curve is developed from measurements of streamflow and stage that are taken periodically to update the curve from changes due to scour and fill. What should be noted is that the stage-discharge relationship is developed assuming steady flow. Therefore it may not be entirely accurate during unsteady flow. Chow shows a comparison of a rating curve from the stage-discharge relationship with a rating curve obtained using dynamic and diffusion models. It is shown that the rating curve from the stage-discharge relationship over estimates the values of discharge during the rising limb of the hydrograph and underestimates the values during the recession. Therefore to compare the predicted DRH and the DRH estimated through the stage discharge relationship this should be considered.

The accuracy of the model increased as the sub-watershed size was reduced from approximately 1 mi^2 to 0.5 mi^2. When the sub-watershed size was reduced the peak was shifted forward and the hydrograph became slightly flattened, more closely matching the actual hydrograph from the USGS data.

There does seem to be a slight inaccuracy in the falling limb of the hydrograph generated by the model for Event 1 using 11 sub-watersheds. The falling limb of the model hydrograph lags that of the actual hydrograph such that the rate of recession seems to be somewhat slower. It can also be seen that the time to peak for the predicted hydrograph is slightly longer than that of the actual hydrograph. For the predicted hydrograph generated by the second event (more of a low flow event) the opposite is true. In this case the predicted hydrograph for the second event the falling limb seems to recede faster than the actual hydrograph. A fast rise and recession is also seen in the portion of the predicted hydrograph for the first event that is below 20 cubic feet per second.

A reasonable explanation for these differences could be that the Ramser equation which is used to calculate the travel time of the water through the stream does not take into account changes in velocity that may result from increasing flow. This is why it was possible to use a constant value for the isochrones. In the first event the velocities after the flow has increased near the maximum flow could have been underestimated by the Ramser equation which was used to generate the isochrones. This would account for the longer time to peak and slower recession of the predicted hydrograph. In the second event which the flow is much lower the opposite may have been true. The Ramser equation may be overestimating the velocities used to create the isochrones. This would explain the early peak and faster recession of the predicted hydrograph for the second event. Obviously some of the error in the second event is due to the inaccuracy of the curve number method for events with low runoff. This can be seen in the comparison the total volume. Looking at data from the streamflow measurements taken by the USGS at the location of the streamgage it can be seen that with higher flows the mean velocity estimated from the measurements is indeed higher. This would suggest that an improvement to this model would be to develop a way to account for the changes in velocity due to the increasing flow.

9.3 Gridded Incremental Precipitation

It was shown in this research that the use of gridded precipitation in this rainfall-runoff model is a good improvement. The implementation of this data was possible because of the efforts of the Three Rivers Wet Weather Demonstration Program in purchasing the data and publishing it on the web. The processing of gage adjusted radar rainfall data is still a relatively new field. There are only a few companies that process this data and it is rather expensive to obtain for selected events such as the event studied in this research. The public availability of this data made it economically feasible for the use in this model.

10.0 RECOMMENDATIONS

Because the gridded precipitation and stream flow data were only limited to two storms it is recommended that the model by tested against more data. This will need to be done as data are collected. Data should become available with the rainfall in the spring of 2003.

An assumption of the model is that the isochrones remain constant even when the flow has increased. It could be argued that often the average velocity increases as the flow increases. Therefore as the stream flow increases during the storm the isochrones would change. This is an interesting area to investigate, but it may prove difficult to incorporate into the model. The data available for this type of investigation could be found from the streamflow measurements taken by the USGS at the gage location. If a factor could be added to the velocity equation that would account for the change in discharge the model would have to be changed so that the isochrones are no longer constant during the storm.

One of the major difficulties in attempting to develop a model for small watersheds is that there is often not enough data collected. As environmental concerns with combined sewer overflows and sanitary sewer overloading increase due to federal regulation, there is a renewed interest in the study of small watersheds and sewersheds. The increasing concerns should not only be with the quality aspects but also the quantitative aspect. It is hopeful that this new interest will lead to great improvements in hydrologic measurement and modeling.

APPENDIX

Watershed Program

```
C     ****************************************************************
C     THIS PROGRAM HAS BEEN MODIFIED JAMES EMERICK
C     10/12/02
C     1. READ FROM PROJECT INFO FILE
C     2. CREATES WATERSHEDS BY READING OUTLETS DIRECTLY FROM STREAM
FILE
C     3. CREATES METADATA (.RDC) FILE AUTOMATICALLY
C     ****************************************************************

      IMPLICIT INTEGER*2 (A-Z)
C      INTEGER MXO
      CHARACTER*80 ELFILE,ELRDC,DIRFILE,DIRRDC,SFILE,SRDC,WSRDC,ISORDC
      CHARACTER*80 WSFILE,SOILFILE,LANDFILE,CNFILE,D,CNRDC,ISOFILE

      REAL MN_X,MX_X,MN_Y,MX_Y
      LOGICAL DONE,ACTIV
      DIMENSION DIR(20000,3),MASK(20000,3),SBUF(20000,3),MBUF(20000,3)
      COMMON DIR,MASK

C      MXO=15
C      PRINT 1
C1     FORMAT(/,'*** ENTER LENGTH AND WIDTH OF FILE ***')
C      READ(*,*)NL,NS
C      PRINT 2
C2     FORMAT('*** ENTER DIRECTION FILE ***')
C      READ (5,1000)DIFILE
C      PRINT 3
C3     FORMAT('*** ENTER MASK FILE ***')
C      READ (5,1000)MAFILE
C      PRINT 100
C100   FORMAT('*** ENTER STREAM FILE ***')
C      READ (5,1000)STFILE
      PRINT 7
7     FORMAT('*** ENTER HIGHEST OUTLET NUMBER ***')
      READ (5,*)MXO
1000  FORMAT(A80)
C      THIS IS A TEXT FILE WHERE THE FILE INFO IS STORED TO AND READ FROM
C      Z IS FOR INF NOT USED IN THIS PARTICULAR PROGRAM
       OPEN(3,FILE='PROJINFO.TXT',STATUS='UNKNOWN')
       READ(3,*)NL
       READ(3,*)NS
       READ(3,*)ELFILE
```

```
      READ(3,*)ELRDC
      READ(3,*)DIRFILE
      READ(3,*)DIRRDC
      READ(3,*)SFILE
      READ(3,*)SRDC
      READ(3,*)WSFILE
      READ(3,*)WSRDC
      READ(3,*)CNFILE
      READ(3,*)CNRDC
      READ(3,*)D
      READ(3,*)D
      READ(3,*)MN_X
      READ(3,*)MX_X
      READ(3,*)MN_Y
      READ(3,*)MX_Y
      CLOSE(3,STATUS='KEEP')
C-- DFILE IS THE SDIR FILE THAT IS MADE BY DOSDIR, I*2
      DFILE=13
C-- MFILE IS THE STARTER FILE,0 FOR MASK,-1 FOR NO LABEL YET,
C    AND A NUMBER WHERE THE USER WISHES A WATERSHED TO TERMINATE
      MFILE=12
      STFILE=14
C
C     This section is for opening files under UNIX and DOS
C
      OPEN(13,STATUS='OLD',RECL=NS*2,
     *   FORM='UNFORMATTED',ACCESS='DIRECT',
     *   FILE=DIRFILE)
      OPEN(12,STATUS='NEW',FORM='UNFORMATTED',
     *   ACCESS='DIRECT',FILE=WSFILE,RECL=NS*2)
      OPEN(14,STATUS='OLD',FORM='UNFORMATTED',
     *   ACCESS='DIRECT',FILE=SFILE,RECL=NS*2)

      DO 711 I=1,NL
      DO 712 J=1,NS
      MBUF(J,1)=-1
      WRITE (12,REC=I) (MBUF(J,1))
712     CONTINUE
711     CONTINUE

      DO 145 I=1,NL
      READ (14,REC=I) (SBUF(J,1),J=1,NS)
      DO 144 J=1,NS
      IF(SBUF(J,1).LT.2) THEN
        SBUF(J,1)=-1
        ENDIF
```

```
144   CONTINUE
      WRITE (12,REC=I) (SBUF(J,1),J=1,NS)
145   CONTINUE

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      UPDOWN=-1
      PASS=0
 33   I1=1
      I2=2
      I3=3
      UPDOWN=-UPDOWN
      DONE=.TRUE.
      PASS=PASS+1
      PRINT *,'PASS',PASS
      DO 4 II=1,NS+2
      INDEX=1
      IF(UPDOWN.EQ.-1)INDEX=3
 4    MASK(II,INDEX)=0
      DO 5 I=1,3
      MASK(1,I)=0
 5    MASK(NS+2,I)=0
      INDEX2=0
      IF(UPDOWN.EQ.-1)INDEX2=NL-2
      INDEX1=1
      IF(UPDOWN.EQ.-1)INDEX1=0
      DO 6 I=1,2
      READ(DFILE,REC=I+INDEX2)(DIR(II,I+INDEX1),II=2,NS+1)
 6    READ(MFILE,REC=I+INDEX2)(MASK(II,I+INDEX1),II=2,NS+1)
      DO 10 I=1,NL
 15   ACTIV=.FALSE.
      DO 20 J=2,NS+1
      IF(MASK(J,I2).NE.-1)GOTO 20
      DIRX=DIR(J,I2)
      IF(DIRX.NE.0) GOTO 16
      MASK(J,I2) = 0
      ACTIV = .TRUE.
      GOTO 20
 16   IF(MASK(J+1,I1).LT.0.OR.DIRX.NE.1)GOTO 21
      MASK(J,I2)=MASK(J+1,I1)
      ACTIV=.TRUE.
      GOTO 20
 21   IF(MASK(J+1,I2).LT.0.OR.DIRX.NE.2)GOTO 22
      MASK(J,I2)=MASK(J+1,I2)
      ACTIV=.TRUE.
```

```
      GOTO 20
 22   IF(MASK(J+1,I3).LT.0.OR.DIRX.NE.4)GOTO 23
      MASK(J,I2)=MASK(J+1,I3)
      ACTIV=.TRUE.
      GOTO 20
 23   IF(MASK(J,I3).LT.0.OR.DIRX.NE.8)GOTO 24
      MASK(J,I2)=MASK(J,I3)
      ACTIV=.TRUE.
      GOTO 20
 24   IF(MASK(J-1,I3).LT.0.OR.DIRX.NE.16)GOTO 25
      MASK(J,I2)=MASK(J-1,I3)
      ACTIV=.TRUE.
      GOTO 20
 25   IF(MASK(J-1,I2).LT.0.OR.DIRX.NE.32)GOTO 26
      MASK(J,I2)=MASK(J-1,I2)
      ACTIV=.TRUE.
      GOTO 20
 26   IF(MASK(J-1,I1).LT.0.OR.DIRX.NE.64)GOTO 27
      MASK(J,I2)=MASK(J-1,I1)
      ACTIV=.TRUE.
      GOTO 20
 27   IF(MASK(J,I1).LT.0.OR.DIRX.NE.128)GOTO 20
      MASK(J,I2)=MASK(J,I1)
      ACTIV=.TRUE.
      GOTO 20
 20   IF(ACTIV)DONE=.FALSE.
      IF(ACTIV)GOTO 15
      INDEX=I
      IF(UPDOWN.EQ.-1)INDEX=NL-I+1
      WRITE(MFILE,REC=INDEX)(MASK(II,I2),II=2,NS+1)
      IF(I.EQ.NL)GOTO 10
      IF(UPDOWN.EQ.-1)GOTO 18
      ITEMP=I1
      I1=I2
      I2=I3
      I3=ITEMP
      IF(I.NE.NL-1)GOTO 65
      DO 64 II=1,NS+2
 64   MASK(II,I3)=0
      GOTO 10
 65   READ(MFILE,REC=I+2)(MASK(II,I3),II=2,NS+1)
      READ(DFILE,REC=I+2)(DIR(II,I3),II=2,NS+1)
      GOTO 10
C--DOING A BOTTOM TO TOP PASS
 18   ITEMP=I3
      I3=I2
```

```
      I2=I1
      I1=ITEMP
      IF(I.NE.NL-1)GOTO 19
      DO 17 II=1,NS+2
 17   MASK(II,I1)=0
      GOTO 10
 19   READ(MFILE,REC= NL-I-1)(MASK(II,I1),II=2,NS+1)
      READ(DFILE,REC=NL-I-1)(DIR(II,I1),II=2,NS+1)
 10   CONTINUE
      IF(.NOT.DONE)GOTO 33
*******************************************************
      OPEN(2,FILE=WSRDC,STATUS='UNKNOWN')
       PRINT 110
       WRITE(2,110)
110    FORMAT(0X,'file format : IDRISI Raster A.1')
       PRINT 111
       WRITE(2,111)
111    FORMAT(0X,'file title  : ')
       PRINT 112
       WRITE(2,112)
112    FORMAT(0X,'data type   : integer')
       PRINT 113
       WRITE(2,113)
113    FORMAT(0X,'file type   : binary')
       PRINT 114, NS
       WRITE(2,114) NS
114    FORMAT(0X,'columns     : ',I3)
       PRINT 115,NL
       WRITE(2,115) NL
115    FORMAT(0X,'rows        : ',I3)
       PRINT 116
       WRITE(2,116)
116    FORMAT(0X,'ref. system : us27tm17')
       PRINT 117
       WRITE(2,117)
117    FORMAT(0X,'ref. units  : m')
       PRINT 118
       WRITE(2,118)
118    FORMAT(0X,'unit dist.  : 1.0000000')
       PRINT 119, MN_X
       WRITE(2,119) MN_X
119    FORMAT(0X,'min. X      : ',F9.0)
       PRINT 120, MX_X
       WRITE(2,120) MX_X
120    FORMAT(0X,'max. X      : ',F9.0)
       PRINT 121, MN_Y
```

```
      WRITE(2,121) MN_Y
121   FORMAT(0X,'min. Y      : ',F9.0)
      PRINT 122, MX_Y
      WRITE(2,122) MX_Y
122   FORMAT(0X,'max. Y      : ',F9.0)
      PRINT 123
      WRITE(2,123)
123   FORMAT(0X,"pos'n error : unknown")
      PRINT 124
      WRITE(2,124)
124   FORMAT(0X,'resolution  : 30.0273323')
      PRINT 125
      WRITE(2,125)
125   FORMAT(0X,'min. value  : 0')
      PRINT 126, MXO
      WRITE(2,126)MXO
126   FORMAT(0X,'max. value  : ',I3)
      PRINT 127
      WRITE(2,127)
127   FORMAT(0X,'display min : 0')
      PRINT 128, MXO
      WRITE(2,128)MXO
128   FORMAT(0X,'display max : ',I3)
      PRINT 129
      WRITE(2,129)
129   FORMAT(0X,'value units : m')
      PRINT 130
      WRITE(2,130)
130   FORMAT(0X,'value error : unknown')
     PRINT 131
      WRITE(2,131)
131   FORMAT(0X,'flag value  : 0')
      PRINT 132
      WRITE(2,132)
132   FORMAT(0X,"flag def'n  : background")
      PRINT 133
      WRITE(2,133)
133   FORMAT(0X,'legend cats : 0')


     STOP
     END
```

Curve Number Program

```
C
******************************************************************************
C       PROGRAM CURVE_NUMBER_KNK FOR THE CALCULATION OF THE CURVE
NUMBER OF THE WATERSHED
C       UNDER STUDY
C       AUTHOR: KHALID KHAN
C       DATE: 07/08/2001
C
******************************************************************************
**
C     THIS PROGRAM HAS BEEN MODIFIED BY JAMES EMERICK
C     10/12/02
C     1. TEXT OUTPUT FOR EACH WATERSHED
C     2. READ FROM PROJECT INFO FILE
C     3. GENERATE RDC FILE AUTOMATICALLY
C
******************************************************************************
**
C
C
C    INTITIALIZING THE DIFFERENT ARRAYS
C    WSBUF STORES THE READ IN DATA FROM THE WATERSHED FILE
C    SOILBUF STORES THE READ IN DATA FROM THE SOIL RASTER FILE
C    LANDBUF STORES THE READ IN DATA FROM THE LAND USE RASTER FILE
C    CNBUF STORES THE WRITTEN CURVE NUMBERS IN THE RASTER FILE
C
C
C
      INTEGER*2 WSBUF(1500,1500),SOILBUF(1500,1500),LANDBUF(1500,1500)
      INTEGER*2 CNBUF(1500,1500)
      INTEGER*2 CN15(10000),CN14(10000),CN13(10000),CN12(10000),
     *CN11(10000)
      INTEGER*2 CN10(10000),CN9(10000),CN8(10000),CN7(10000),CN6(10000)
      INTEGER*2 CN5(10000),CN4(10000),CN3(10000),CN2(10000),CN1(10000)
      INTEGER NUM_CELLS15,NUM_CELLS14,NUM_CELLS13,NUM_CELLS12,
     *NUM_CELLS11
      INTEGER
NUM_CELLS10,NUM_CELLS9,NUM_CELLS8,NUM_CELLS7,NUM_CELLS6
      INTEGER NUM_CELLS5,NUM_CELLS4,NUM_CELLS3,NUM_CELLS2,NUM_CELLS1
      INTEGER CURVE,N,SOIL,LAND,Q,NUM_CELLS,WSNUM
      INTEGER CCN1,CCN2,CCN3,CCN4,CCN5,CCN6,CCN7,CCN8,CCN9,CCN10
      INTEGER CCN11,CCN12,CCN13,CCN14,CCN15
      CHARACTER*80 ELFILE,ELRDC,DIRFILE,DIRRDC,SFILE,SRDC,WSRDC
```

```
      CHARACTER*80 WSFILE,SOILFILE,LANDFILE,CNFILE,D,CNRDC
      REAL MN_X,MX_X,MN_Y,MX_Y

C     GETTING THE INFORMATION ABOUT THE FILES FROM THE USER
C
C      PRINT 1
C1     FORMAT(/,'*** ENTER LENGTH AND WIDTH OF FILE***')
C      READ(*,*)NL,NS
C      PRINT 2
C2     FORMAT('*** ENTER WATERSHED FILE***')
C      READ(5,1000)WSFILE
***C   PRINT 3
***C3 FORMAT('*** ENTER THE WATERSHED NUMBER ***')
***C      READ(*,*)Q
      PRINT 1
1     FORMAT('*** ENTER THE SOIL RASTER FILE ***')
      READ(5,1000)SOILFILE
      PRINT 2
2     FORMAT('*** ENTER THE LAND USE RASTER FILE ***')
      READ(5,1000)LANDFILE
C      PRINT 3
C3     FORMAT('*** ENTER THE CURVE NUMBER FILE ***')
C      READ(5,1000)CNFILE
C      PRINT 4
C4     FORMAT('*** ENTER THE CURVE NUMBER METADATA FILE ***')
C      READ(5,1000)CNRDC
1000  FORMAT (A80)
C
C      THIS IS A TEXT FILE WHERE THE FILE INFO IS STORED TO AND READ FROM
C
       OPEN(3,FILE='PROJINFO.TXT',STATUS='UNKNOWN')
       READ(3,*)NL
       READ(3,*)NS
       READ(3,*)ELFILE
       READ(3,*)ELRDC
       READ(3,*)DIRFILE
       READ(3,*)DIRRDC
       READ(3,*)SFILE
       READ(3,*)SRDC
       READ(3,*)WSFILE
       READ(3,*)WSRDC
       READ(3,*)CNFILE
       READ(3,*)CNRDC
       READ(3,*)D
       READ(3,*)D
       READ(3,*)MN_X
```

```
       READ(3,*)MX_X
       READ(3,*)MN_Y
       READ(3,*)MX_Y
      CLOSE(3,STATUS='KEEP')



C
C     OPENING ALL THE FILES OLD AND NEW
C
      OPEN(11,FILE=WSFILE,STATUS='OLD',FORM='UNFORMATTED',
     * ACCESS='DIRECT',RECL=NS*2)
      OPEN(12,FILE=SOILFILE,STATUS='OLD',FORM='UNFORMATTED',
     * ACCESS='DIRECT',RECL=NS*2)
      OPEN(13,FILE=LANDFILE,STATUS='OLD',FORM='UNFORMATTED',
     * ACCESS='DIRECT',RECL=NS*2)
      OPEN(14,FILE=CNFILE,STATUS='UNKNOWN',FORM='UNFORMATTED',
     * ACCESS='DIRECT',RECL=NS*2)
C
C     INITIALIZING THE DIFFERENT COUNTERS AND THE NEW RASTER CN FILE TO
0s
C
       SUM=0
          NUM_CELLS=0
          DO 11 I=1,NL
          DO 12 J=1,NS
          CNBUF(J,1)=0
          WRITE (14,REC=I) (CNBUF(J,1))
12        CONTINUE
11        CONTINUE
          N=2
          DO 30 I=N,NL-1
          READ(11,REC=I) (WSBUF(K,1),K=1,NS)
          READ(12,REC=I) (SOILBUF(K,1),K=1,NS)
          READ(13,REC=I) (LANDBUF(K,1),K=1,NS)
          DO 25 K=2,NS-1
C
C     PROCESSING ONLY THE CELLS THAT BELONG TO THE WATERSHED
C
          IF(WSBUF(K,1).EQ.0)GOTO 25
       WSNUM=WSBUF(K,1)
          SOIL=SOILBUF(K,1)
          LAND=LANDBUF(K,1)
C
C     CALLING THE SUBROUTINE CURVE_NUM TO GET THE CURVE NUMBER FOR
THE CELL
```

```
C
         CALL CURVE_NUM(CURVE,SOIL,LAND)
        WSBUF(K,1)=CURVE
        IF(WSNUM.EQ.15)THEN
        NUM_CELLS15=NUM_CELLS15+1
        CN15(CURVE)=CN15(CURVE)+1
        ENDIF
        IF(WSNUM.EQ.14)THEN
        NUM_CELLS14=NUM_CELLS14+1
        CN14(CURVE)=CN14(CURVE)+1
        ENDIF
        IF(WSNUM.EQ.13)THEN
        NUM_CELLS13=NUM_CELLS13+1
        CN13(CURVE)=CN13(CURVE)+1
        ENDIF
        IF(WSNUM.EQ.12)THEN
        NUM_CELLS12=NUM_CELLS12+1
        CN12(CURVE)=CN12(CURVE)+1
        ENDIF
        IF(WSNUM.EQ.11)THEN
        NUM_CELLS11=NUM_CELLS11+1
        CN11(CURVE)=CN11(CURVE)+1
        ENDIF
        IF(WSNUM.EQ.10)THEN
        NUM_CELLS10=NUM_CELLS10+1
        CN10(CURVE)=CN10(CURVE)+1
        ENDIF
        IF(WSNUM.EQ.9)THEN
        NUM_CELLS9=NUM_CELLS9+1
        CN9(CURVE)=CN9(CURVE)+1
        ENDIF
        IF(WSNUM.EQ.8)THEN
        NUM_CELLS8=NUM_CELLS8+1
        CN8(CURVE)=CN8(CURVE)+1
        ENDIF
        IF(WSNUM.EQ.7)THEN
        NUM_CELLS7=NUM_CELLS7+1
        CN7(CURVE)=CN7(CURVE)+1
        ENDIF
        IF(WSNUM.EQ.6)THEN
        NUM_CELLS6=NUM_CELLS6+1
        CN6(CURVE)=CN6(CURVE)+1
        ENDIF
        IF(WSNUM.EQ.5)THEN
        NUM_CELLS5=NUM_CELLS5+1
        CN5(CURVE)=CN5(CURVE)+1
```

```
        ENDIF
        IF(WSNUM.EQ.4)THEN
        NUM_CELLS4=NUM_CELLS4+1
        CN4(CURVE)=CN4(CURVE)+1
        ENDIF
        IF(WSNUM.EQ.3)THEN
        NUM_CELLS3=NUM_CELLS3+1
        CN3(CURVE)=CN3(CURVE)+1
        ENDIF
        IF(WSNUM.EQ.2)THEN
        NUM_CELLS2=NUM_CELLS2+1
        CN2(CURVE)=CN2(CURVE)+1
        ENDIF
        NUM_CELLS1=NUM_CELLS1+1
        CN1(CURVE)=CN1(CURVE)+1

C
C     CREATING THE CURVE NUMBER RASTER FILE
C

25    CONTINUE
      WRITE (14,REC=I) (WSBUF(J,1),J=1,NS)
30    CONTINUE
     CLOSE(11)
     CLOSE(12)
     CLOSE(13)

C     CREATING THE METADATA FILE
     OPEN(2,FILE=CNRDC,STATUS='UNKNOWN')
      PRINT 110
      WRITE(2,110)
110   FORMAT(0X,'file format : IDRISI Raster A.1')
      PRINT 111
      WRITE(2,111)
111   FORMAT(0X,'file title  : ')
      PRINT 112
      WRITE(2,112)
112   FORMAT(0X,'data type   : integer')
      PRINT 113
      WRITE(2,113)
113   FORMAT(0X,'file type   : binary')
      PRINT 114, NS
      WRITE(2,114) NS
114   FORMAT(0X,'columns     : ',I3)
      PRINT 115, NL
      WRITE(2,115) NL
```

```
115  FORMAT(0X,'rows        : ',I3)
     PRINT 116
     WRITE(2,116)
116  FORMAT(0X,'ref. system : us27tm17')
     PRINT 117
     WRITE(2,117)
117  FORMAT(0X,'ref. units  : m')
     PRINT 118
     WRITE(2,118)
118  FORMAT(0X,'unit dist.  : 1.0000000')
     PRINT 119, MN_X
     WRITE(2,119) MN_X
119  FORMAT(0X,'min. X      : ',F9.0)
     PRINT 120, MX_X
     WRITE(2,120) MX_X
120  FORMAT(0X,'max. X      : ',F9.0)
     PRINT 121, MN_Y
     WRITE(2,121) MN_Y
121  FORMAT(0X,'min. Y      : ',F9.0)
     PRINT 122, MX_Y
     WRITE(2,122) MX_Y
122  FORMAT(0X,'max. Y      : ',F9.0)
     PRINT 123
     WRITE(2,123)
123  FORMAT(0X,"pos'n error : unknown")
     PRINT 124
     WRITE(2,124)
124  FORMAT(0X,'resolution  : 30.0273323')
     PRINT 125
     WRITE(2,125)
125  FORMAT(0X,'min. value  : 0')
     PRINT 126
     WRITE(2,126)
126  FORMAT(0X,'max. value  : 100')
     PRINT 127
     WRITE(2,127)
127  FORMAT(0X,'display min : 0')
     PRINT 128
     WRITE(2,128)
128  FORMAT(0X,'display max : 100')
     PRINT 129
     WRITE(2,129)
129  FORMAT(0X,'value units : m')
     PRINT 130
     WRITE(2,130)
130  FORMAT(0X,'value error : unknown')
```

```
      PRINT 131
      WRITE(2,131)
131   FORMAT(0X,'flag value  : 0')
      PRINT 132
      WRITE(2,132)
132   FORMAT(0X,"flag def'n  : background")
      PRINT 133
      WRITE(2,133)
133   FORMAT(0X,'legend cats : 0')


C     CREATING THE CURVE NUMBER TEXT FILE WHICH GIVES THE DISTRIBUTION
C     OF THE CURVE NUMBERS AND CALCULATES THE AVERAGE CURVE NUBER
FOR THE WATERSHED
C
     OPEN(1,FILE='CNOUTPUT.TXT',STATUS='UNKNOWN')
         PRINT 100
         WRITE(1,100)
100   FORMAT(//,30X,'THE CURVE-NUMBER CALCULATION FILE')
     PRINT 101
         WRITE(1,101)
101   FORMAT(30X,'------------------------------------')
C        PRINT 102,Q
C        WRITE(1,102)Q
C102   FORMAT(///,10X,'WATERSHED NUMBER:',I3)
     PRINT 103,NUM_CELLS1
         WRITE(1,103)NUM_CELLS1
103   FORMAT(///,10X,'NUMBER OF CELLS IN THE WATERSHED:',I10)
         PRINT 104
         WRITE(1,104)
104   FORMAT(/////,2X,'CN',15X,'NUMBER OF CELLS')
     PRINT 105
     WRITE(1,105)
105   FORMAT(2X,'----',3X,
    *'------------------------------------------------------------',
    *'------------------------------------------------')
     PRINT 106
     WRITE(1,106)
106   FORMAT(5X,'',3X,'SBWS2',2X,'SBWS3',2X,'SBWS4',2X,'SBWS5',2X
    *,'SBWS6',2X,'SBWS7',2X,'SBWS8',2X,'SBWS9',2X,'SBWS10',2X
    *,'SBWS11',2X,'SBWS12',2X,'SBWS13',2X,'SBWS14',2X,'SBWS15',4X
    *,'TOTAL')
     PRINT 105
     WRITE(1,105)

     DO 31 I=25,100
```

```
      SUM1=SUM1+I*CN1(I)
        SUM2=SUM2+I*CN2(I)
      SUM3=SUM3+I*CN3(I)
      SUM4=SUM4+I*CN4(I)
      SUM5=SUM5+I*CN5(I)
      SUM6=SUM6+I*CN6(I)
      SUM7=SUM7+I*CN7(I)
      SUM8=SUM8+I*CN8(I)
      SUM9=SUM9+I*CN9(I)
      SUM10=SUM10+I*CN10(I)
      SUM11=SUM11+I*CN11(I)
      SUM12=SUM12+I*CN12(I)
      SUM13=SUM13+I*CN13(I)
      SUM14=SUM14+I*CN14(I)
      SUM15=SUM15+I*CN15(I)
        IF(CN1(I).EQ.0)GOTO 31
        PRINT 107,I,CN2(I),CN3(I),CN4(I),CN5(I),CN6(I),CN7(I),CN8(I)
     *,CN9(I),CN10(I),CN11(I),CN12(I),CN13(I),CN14(I),CN15(I),CN1(I)
        WRITE(1,107)I,CN2(I),CN3(I),CN4(I),CN5(I),CN6(I),CN7(I),CN8(I)
     *,CN9(I),CN10(I),CN11(I),CN12(I),CN13(I),CN14(I),CN15(I),CN1(I)
31    CONTINUE
107   FORMAT(2X,I3,3X,I5,2X,I5,2X,I5,2X,I5,2X,I5,2X,I5,2X,I5,2X,I5
     *,3X,I5,3X,I5,3X,I5,3X,I5,3X,I5,3X,I5,4X,I5)
      CCN1=INT(SUM1/NUM_CELLS1)+1
      CCN2=INT(SUM2/NUM_CELLS2)+1
      CCN3=INT(SUM3/NUM_CELLS3)+1
      CCN4=INT(SUM4/NUM_CELLS4)+1
      CCN5=INT(SUM5/NUM_CELLS5)+1
      CCN6=INT(SUM6/NUM_CELLS6)+1
      CCN7=INT(SUM7/NUM_CELLS7)+1
      CCN8=INT(SUM8/NUM_CELLS8)+1
      CCN9=INT(SUM9/NUM_CELLS9)+1
      CCN10=INT(SUM10/NUM_CELLS10)+1
      CCN11=INT(SUM11/NUM_CELLS11)+1
      CCN12=INT(SUM12/NUM_CELLS12)+1
      CCN13=INT(SUM13/NUM_CELLS13)+1
      CCN14=INT(SUM14/NUM_CELLS14)+1
      CCN15=INT(SUM15/NUM_CELLS15)+1
      PRINT 105
      WRITE(1,105)
      PRINT 108,CCN2,CCN3,CCN4,CCN5,CCN6,CCN7,CCN8,CCN9,CCN10
     *,CCN11,CCN12,CCN13,CCN14,CCN15,CCN1
      WRITE(1,108)CCN2,CCN3,CCN4,CCN5,CCN6,CCN7,CCN8,CCN9,CCN10
     *,CCN11,CCN12,CCN13,CCN14,CCN15,CCN1
108   FORMAT(0X,'AVE CN',4X,I3,4X,I3,4X,I3,4X,I3,4X,I3,4X,I3,4X,I3,4X,I3
     *,5X,I3,5X,I3,5X,I3,5X,I3,5X,I3,5X,I3,6X,I3)
```

```
      PRINT 109,CCN1
      WRITE(1,109)CCN1
109   FORMAT(///,10X,'CURVE NUMBER FOR THE ENTIRE WATERSHED:',I3)
      STOP
      END

          SUBROUTINE CURVE_NUM(C,S,L)
          INTEGER C,S,L
          IF((S.EQ.1).AND.(L.EQ.1))THEN
              C=72
          ELSEIF((S.EQ.1).AND.(L.EQ.2))THEN
              C=62
          ELSEIF((S.EQ.1).AND.(L.EQ.3))THEN
              C=68
          ELSEIF((S.EQ.1).AND.(L.EQ.4))THEN
              C=39
          ELSEIF((S.EQ.1).AND.(L.EQ.5))THEN
              C=30
          ELSEIF((S.EQ.1).AND.(L.EQ.6))THEN
              C=45
          ELSEIF((S.EQ.1).AND.(L.EQ.7))THEN
              C=25
          ELSEIF((S.EQ.1).AND.(L.EQ.8))THEN
              C=39
          ELSEIF((S.EQ.1).AND.(L.EQ.9))THEN
              C=49
          ELSEIF((S.EQ.1).AND.(L.EQ.10))THEN
            C=89
          ELSEIF((S.EQ.1).AND.(L.EQ.11))THEN
              C=81
          ELSEIF((S.EQ.1).AND.(L.EQ.12))THEN
              C=77
          ELSEIF((S.EQ.1).AND.(L.EQ.13))THEN
              C=61
          ELSEIF((S.EQ.1).AND.(L.EQ.14))THEN
              C=57
          ELSEIF((S.EQ.1).AND.(L.EQ.15))THEN
              C=54
          ELSEIF((S.EQ.1).AND.(L.EQ.16))THEN
              C=51
          ELSEIF((S.EQ.1).AND.(L.EQ.17))THEN
              C=98
          ELSEIF((S.EQ.1).AND.(L.EQ.18))THEN
            C=98
          ELSEIF((S.EQ.1).AND.(L.EQ.19))THEN
            C=76
```

```
    ELSEIF((S.EQ.1).AND.(L.EQ.20))THEN
         C=72
ELSEIF((S.EQ.1).AND.(L.EQ.21))THEN
         C=77
ELSEIF((S.EQ.1).AND.(L.EQ.22))THEN
         C=100
    ELSEIF((S.EQ.2).AND.(L.EQ.1))THEN
         C=81
    ELSEIF((S.EQ.2).AND.(L.EQ.2))THEN
         C=71
    ELSEIF((S.EQ.2).AND.(L.EQ.3))THEN
         C=79
    ELSEIF((S.EQ.2).AND.(L.EQ.4))THEN
         C=61
    ELSEIF((S.EQ.2).AND.(L.EQ.5))THEN
         C=58
    ELSEIF((S.EQ.2).AND.(L.EQ.6))THEN
          C=66
    ELSEIF((S.EQ.2).AND.(L.EQ.7))THEN
         C=55
    ELSEIF((S.EQ.2).AND.(L.EQ.8))THEN
         C=61
    ELSEIF((S.EQ.2).AND.(L.EQ.9))THEN
         C=69
    ELSEIF((S.EQ.2).AND.(L.EQ.10))THEN
         C=92
    ELSEIF((S.EQ.2).AND.(L.EQ.11))THEN
         C=88
    ELSEIF((S.EQ.2).AND.(L.EQ.12))THEN
         C=85
    ELSEIF((S.EQ.2).AND.(L.EQ.13))THEN
         C=75
    ELSEIF((S.EQ.2).AND.(L.EQ.14))THEN
         C=72
    ELSEIF((S.EQ.2).AND.(L.EQ.15))THEN
         C=70
    ELSEIF((S.EQ.2).AND.(L.EQ.16))THEN
         C=68
    ELSEIF((S.EQ.2).AND.(L.EQ.17))THEN
         C=98
    ELSEIF((S.EQ.2).AND.(L.EQ.18))THEN
         C=98
    ELSEIF((S.EQ.2).AND.(L.EQ.19))THEN
         C=85
    ELSEIF((S.EQ.2).AND.(L.EQ.20))THEN
         C=82
```

```
ELSEIF((S.EQ.2).AND.(L.EQ.21))THEN
        C=86
ELSEIF((S.EQ.2).AND.(L.EQ.22))THEN
        C=100
   ELSEIF((S.EQ.3).AND.(L.EQ.1))THEN
        C=88
   ELSEIF((S.EQ.3).AND.(L.EQ.2))THEN
        C=78
   ELSEIF((S.EQ.3).AND.(L.EQ.3))THEN
        C=86
   ELSEIF((S.EQ.3).AND.(L.EQ.4))THEN
        C=74
   ELSEIF((S.EQ.3).AND.(L.EQ.5))THEN
        C=71
   ELSEIF((S.EQ.3).AND.(L.EQ.6))THEN
        C=77
   ELSEIF((S.EQ.3).AND.(L.EQ.7))THEN
        C=70
   ELSEIF((S.EQ.3).AND.(L.EQ.8))THEN
        C=74
   ELSEIF((S.EQ.3).AND.(L.EQ.9))THEN
        C=79
   ELSEIF((S.EQ.3).AND.(L.EQ.10))THEN
        C=94
   ELSEIF((S.EQ.3).AND.(L.EQ.11))THEN
        C=91
   ELSEIF((S.EQ.3).AND.(L.EQ.12))THEN
        C=90
   ELSEIF((S.EQ.3).AND.(L.EQ.13))THEN
        C=83
   ELSEIF((S.EQ.3).AND.(L.EQ.14))THEN
        C=81
   ELSEIF((S.EQ.3).AND.(L.EQ.15))THEN
        C=80
   ELSEIF((S.EQ.3).AND.(L.EQ.16))THEN
        C=79
   ELSEIF((S.EQ.3).AND.(L.EQ.17))THEN
        C=98
   ELSEIF((S.EQ.3).AND.(L.EQ.18))THEN
         C=98
   ELSEIF((S.EQ.3).AND.(L.EQ.19))THEN
        C=89
   ELSEIF((S.EQ.3).AND.(L.EQ.20))THEN
        C=87
ELSEIF((S.EQ.3).AND.(L.EQ.21))THEN
        C=91
```

```
ELSEIF((S.EQ.3).AND.(L.EQ.22))THEN
        C=100
   ELSEIF((S.EQ.4).AND.(L.EQ.1))THEN
        C=91
   ELSEIF((S.EQ.4).AND.(L.EQ.2))THEN
        C=81
   ELSEIF((S.EQ.4).AND.(L.EQ.3))THEN
        C=89
   ELSEIF((S.EQ.4).AND.(L.EQ.4))THEN
        C=80
   ELSEIF((S.EQ.4).AND.(L.EQ.5))THEN
        C=78
   ELSEIF((S.EQ.4).AND.(L.EQ.6))THEN
        C=83
   ELSEIF((S.EQ.4).AND.(L.EQ.7))THEN
        C=77
   ELSEIF((S.EQ.4).AND.(L.EQ.8))THEN
        C=80
   ELSEIF((S.EQ.4).AND.(L.EQ.9))THEN
        C=84
   ELSEIF((S.EQ.4).AND.(L.EQ.10))THEN
        C=95
   ELSEIF((S.EQ.4).AND.(L.EQ.11))THEN
        C=93
   ELSEIF((S.EQ.4).AND.(L.EQ.12))THEN
        C=92
   ELSEIF((S.EQ.4).AND.(L.EQ.13))THEN
        C=87
   ELSEIF((S.EQ.4).AND.(L.EQ.14))THEN
        C=86
   ELSEIF((S.EQ.4).AND.(L.EQ.15))THEN
        C=85
   ELSEIF((S.EQ.4).AND.(L.EQ.16))THEN
        C=84
   ELSEIF((S.EQ.4).AND.(L.EQ.17))THEN
        C=98
   ELSEIF((S.EQ.4).AND.(L.EQ.18))THEN
        C=98
   ELSEIF((S.EQ.4).AND.(L.EQ.19))THEN
        C=91
   ELSEIF((S.EQ.4).AND.(L.EQ.20))THEN
        C=89
ELSEIF((S.EQ.4).AND.(L.EQ.21))THEN
        C=94
ELSEIF((S.EQ.4).AND.(L.EQ.22))THEN
        C=100
```

```
        ENDIF
    RETURN
    END
```

Isochrone Program

```
C
******************************************************************************
*****
C    THIS PROGRAM WAS ORRIGINALLY WRITTEN BY MICHELLINI AND HAS BEEN
MODIFIED
C    BY KHAN AND EMERICK
C
******************************************************************************
******
C
C
******************************************************************************
******
C
C
******************************************************************************
******
      INTEGER*2 BUFFER(1500,3),DIR(1500,3),NEWBUF(1500,3),ELEVS(1500,3)
      INTEGER*2 ISOBUF(1500,3),WBUF(1500,3),CNBUF(1500,3),EL1(1500,3)
      INTEGER*2 EL2(1500,3)
      CHARACTER*80 ELFILE,ELRDC,DIRFILE,DIRRDC,SFILE,SRDC,WSRDC,ISORDC
      CHARACTER*80 WSFILE,SOILFILE,LANDFILE,CNFILE,D,CNRDC,ISOFILE
      REAL SIZE,TC,MAXSUM,MXCSUM,MXSLOP,RET,SLOPE,SUM_PE,AVE_PE,CNN
      REAL S1,S2,E1,E2,SAV,DIST,XDIST,SC1,SC2,SCAV
      REAL SAVB,SCAVB,SB,SCB
      REAL MN_X,MX_X,MN_Y,MX_Y
      INTEGER*2 TT
      INTEGER Q,INTERV,TOT,CN,RO,T,W,E,Z,CELLNUM
      INTEGER TA(9000)
      LOGICAL ACTIV
      LUNIT=7
      PUNIT=12
      DUNIT=14
      ISOUNIT=18
      CNUNIT=19


C     THIS IS A TEXT FILE WHERE THE FILE INFO IS STORED TO AND READ FROM
C     Z IS FOR INF NOT USED IN THIS PARTICULAR PROGRAM
      OPEN(3,FILE='PROJINFO.TXT',STATUS='UNKNOWN')
      READ(3,*)NL
      READ(3,*)NS
```

```
      READ(3,*)ELFILE
      READ(3,*)ELRDC
      READ(3,*)DIRFILE
      READ(3,*)DIRRDC
      READ(3,*)SFILE
      READ(3,*)SRDC
      READ(3,*)WSFILE
      READ(3,*)WSRDC
      READ(3,*)CNFILE
      READ(3,*)CNRDC
      READ(3,*)ISOFILE
      READ(3,*)ISORDC
      READ(3,*)MN_X
      READ(3,*)MX_X
      READ(3,*)MN_Y
      READ(3,*)MX_Y
     CLOSE(3,STATUS='KEEP')




C     PRINT 10
C10   FORMAT(/,'---- OUTPUT FILE IS NAMED "DATA" ----')
C     PRINT 1
C1    FORMAT(/,'*** ENTER LENGTH AND WIDTH OF FILE ***')
C     READ (*,*)NL,NS
C     PRINT 2
C2    FORMAT('*** ENTER ELEVATION FILE ***')
C     READ (5,1000)ELFILE
C     PRINT 3
C3    FORMAT('*** ENTER WATERSHED FILE ***')
C     READ (5,1000)WFILE
C     PRINT 4
C4    FORMAT('*** ENTER STREAM FILE ***')
C     READ (5,1000)SFILE
C     PRINT 5
C5    FORMAT('*** ENTER DIRECTION FILE ***')
C     READ (5,1000)DFILE
C     PRINT 13
C13   FORMAT('*** ENTER ISOCHRONES FILE ***')
C     READ (5,1000)ISOFILE
C     PRINT 44
C44   FORMAT('*** ENTER ISOCHRONES METADATA FILE ***')
C     READ (5,1000)ISORDC
```

```
C        PRINT 41
C41    FORMAT('*** ENTER PRECIPITATION FILE ***')
C      READ (5,1000)PRECFILE
C        PRINT 42
C42    FORMAT('*** ENTER CURVE NUMBER FILE ***')
C      READ (5,1000)CNFILE
C      PRINT 6
C6     FORMAT('*** ENTER THE GRID SIZE (METERS) ***')
C      READ (5,*)SIZE
      SIZE=30
      PRINT 7
7     FORMAT('*** ENTER HIGHEST WATERSHED NUMBER ***')
      READ (5,*)Q
C      PRINT 8
C8     FORMAT('*** ENTER AVERAGE VALUE OF CN FOR THE WATERSHED ***')
C      READ (5,*)CN
C      PRINT 9
C9     FORMAT('*** ENTER THE ROW CONTAINING THE OUTLET ***')
C      READ (5,*)RO
      PRINT 43
43     FORMAT('*** ENTER TIME INTERVAL IN MINUTES ***')
      READ (5,*)INTERV

1000  FORMAT(A80)

C      OPEN(3,FILE='PROJINFO.TXT',STATUS='UNKNOWN')
C      WRITE(3,*)NL
C      WRITE(3,*)NS
C      WRITE(3,*)ELFILE
C      WRITE(3,*)ELRDC
C      WRITE(3,*)DIRFILE
C      WRITE(3,*)DIRRDC
C      WRITE(3,*)SFILE
C      WRITE(3,*)SRDC
C      WRITE(3,*)WSFILE
C      WRITE(3,*)WSRDC
C      WRITE(3,*)CNFILE
C      WRITE(3,*)CNRDC
C      WRITE(3,*)ISOFILE
C      WRITE(3,*)ISORDC
C      WRITE(3,*)MN_X
C      WRITE(3,*)MX_X
C      WRITE(3,*)MN_Y
C      WRITE(3,*)MX_Y
C      CLOSE(3,STATUS='KEEP')
C
```

```
C    OPENING FILES UNDER UNIX OR DOS
C
10    OPEN(7,STATUS='OLD',RECL=NS*2,
     * FORM='UNFORMATTED',ACCESS='DIRECT',FILE=ELFILE)
      OPEN(8,STATUS='OLD',RECL=NS*2,
     * FORM='UNFORMATTED',ACCESS='DIRECT',FILE=ELFILE)
      OPEN(12,STATUS='OLD',RECL=NS*2,
     * FORM='UNFORMATTED',ACCESS='DIRECT',FILE=WSFILE)
      OPEN(16,STATUS='OLD',RECL=NS*2,
     * FORM='UNFORMATTED',ACCESS='DIRECT',FILE=SFILE)
      OPEN(14,STATUS='OLD',ACCESS='DIRECT',FILE=DIRFILE,
     * RECL=NS*2,FORM='UNFORMATTED')
      OPEN(18,STATUS='UNKNOWN',ACCESS='DIRECT',FILE=ISOFILE,
     * RECL=NS*2,FORM='UNFORMATTED')
      OPEN(19,STATUS='OLD',ACCESS='DIRECT',FILE=CNFILE,
     * RECL=NS*2,FORM='UNFORMATTED')
C
C    INITIALIZING THE ISOCHROMES FILE
C
      TT=0
         DO 711 I=1,NL
         DO 712 J=1,NS
         ISOBUF(J,1)=0
         WRITE (18,REC=I) (ISOBUF(J,1))
712      CONTINUE
711      CONTINUE
C
C    FINDING THE BOTTOM ELEVATION
C
      S1=0.0
      S2=0.0
      DIST=SIZE*1.0
      XDIST=SQRT(2.0*DIST*DIST)
      I=1
      J=1
      K=0
      Z=1
      W=1
      E=0
      N=0
      M=0
      KOUNT=0
      KOUNT1=0
      SUM_PE=0.0
      T_CO=0
      T_CC=0
```

```
      OPEN(3,FILE='RECORD.TXT',STATUS='UNKNOWN')
      OPEN(4,FILE='RECORD2.TXT',STATUS='UNKNOWN')
      DO 15 I=1,NL
      READ (16,REC=I) (BUFFER(J,1),J=1,NS)
      READ (7,REC=I) (ELEVS(J,1),J=1,NS)
      DO 14 J=1,NS
      IF(BUFFER(J,1).EQ.Q) THEN
          BOTELV=ELEVS(J,1)
          GO TO 17
          ENDIF
14    CONTINUE
15    CONTINUE
C
C    INITIALIZE TIMES--INTERV = INTERVAL,(TOTAL OF 250)
C
C
C     TMAX1 IS ACTUAL TIME OF CONCENTRATION
C
17    TMAX=0
      TMAX1=0.
      MAXSUM=0.
      MXCSUM=0.
      N=2
      ROW=1
C
C     READING THE FILES
C
60    KOUNT=KOUNT+1
      KOUNT1=KOUNT1+1
      SUM=0.
      CSUM=0.
      I1=1
      I2=2
      I3=3
      READ (12,REC=N-1) (WBUF(K,1),K=1,NS)
      READ (12,REC=N) (WBUF(K,2),K=1,NS)
      READ (12,REC=N+1) (WBUF(K,3),K=1,NS)
      READ (16,REC=N-1) (BUFFER(K,1),K=1,NS)
      READ (16,REC=N) (BUFFER(K,2),K=1,NS)
      READ (16,REC=N+1) (BUFFER(K,3),K=1,NS)
      READ (18,REC=N-1) (NEWBUF(K,1),K=1,NS)
      READ (18,REC=N) (NEWBUF(K,2),K=1,NS)
      READ (18,REC=N+1) (NEWBUF(K,3),K=1,NS)

      READ (8,REC=N-1) (EL2(K,1),K=1,NS)
      READ (8,REC=N) (EL2(K,2),K=1,NS)
```

```
      READ (8,REC=N+1) (EL2(K,3),K=1,NS)
      M=N
      ACTIV=.TRUE.
C
C    IDENTIFYING THE CELLS THAT FALL IN THE INCUMBENT WATERSHED
C
      DO 30 I=N,NL-1
      ROW=ROW+1
      DO 25 K=0,NS
      IF(WBUF(K,I2).NE.Q)GOTO 25
      CELLNUM=CELLNUM+1
      IF(BUFFER(K,I2).LE.0)THEN
C          NEWBUF(K,I2)=TT
          WBUF(K,I2)=0
          Z=K
          W=I2
          E=N
          GO TO 23
          ENDIF
      IF (BUFFER(K,I2).EQ.1)THEN
C          NEWBUF(K,I2)=TT
          WBUF(K,I2)=0
          ACTIV=.FALSE.
          Z=K
          W=I2
          E=N
          GO TO 23
          ENDIF
      IF (BUFFER(K,I2).EQ.Q)THEN
C          NEWBUF(K,I2)=TT
          WBUF(K,I2)=0
          WRITE (12,REC=M) (WBUF(L,I2),L=1,NS)
          GO TO 60
          ENDIF
23    READ (LUNIT,REC=M)(ELEVS(J,1),J=1,NS)
      MIDELV=ELEVS(K,1)
      TOPELV=ELEVS(K,1)
      J=K
      GO TO 35

25    CONTINUE
      ITEMP=I1
      I1=I2
      I2=I3
      I3=ITEMP
      N=N+1
```

```
      M=M+1
      IF(I.EQ.NL-1)GO TO 200
      READ (12,REC=M+1) (WBUF(K,I3),K=1,NS)
      READ (18,REC=M+1) (NEWBUF(K,I3),K=1,NS)
      READ (16,REC=M+1) (BUFFER(K,I3),K=1,NS)
      READ (8,REC=M+1) (EL2(K,I3),K=1,NS)
30    CONTINUE
35    SUM=0.
      READ (16,REC=M) (BUFFER(K,I2),K=1,NS)
      READ (14,REC=M) (DIR(K,I2),K=1,NS)
      READ (19,REC=M) (CNBUF(K,I2),K=1,NS)
      WRITE (12,REC=M) (WBUF(K,I2),K=1,NS)
      READ (8,REC=M) (EL2(K,I2),K=1,NS)
      READ (8,REC=M) (EL1(K,I2),K=1,NS)

      CN=CNBUF(J,I2)
      CNN=CN*1.0

66    IF(ACTIV)GO TO 75
      GO TO 185
75    IF(BUFFER(J-1,I2).LE.0.AND.DIR(J,I2).EQ.32)THEN
          SUM=SUM+DIST
          ELV2=EL2(J-1,I2)
          ELV1=EL1(J,I2)
          E1=ELV1*1.0
          E2=ELV2*1.0
          S1=(E1-E2)/DIST
          IF(S1.EQ.0)S1=0.01
          S2=S1+S2
          ST=ST+1
C           WRITE(4,*)SAV,S1,S2,E1,E2


          J=J-1
          GO TO 66
          ENDIF
      IF(BUFFER(J,I1).LE.0.AND.DIR(J,I2).EQ.128)THEN
          SUM=SUM+DIST

          ELV2=EL2(J,I1)

          ELV1=EL1(J,I2)
          E1=ELV1*1.0
          E2=ELV2*1.0
          S1=(E1-E2)/DIST
          IF(S1.EQ.0)S1=0.01
```

```
      S2=S1+S2
      ST=ST+1

C       WRITE(4,*)SAV,S1,S2,E1,E2


      GO TO 160
      ENDIF
    IF(BUFFER(J+1,I2).LE.0.AND.DIR(J,I2).EQ.2)THEN
      SUM=SUM+DIST


      ELV2=EL2(J+1,I2)

      ELV1=EL1(J,I2)
      E1=ELV1*1.0
      E2=ELV2*1.0
      S1=(E1-E2)/DIST
      IF(S1.EQ.0)S1=0.01
      S2=S1+S2
      ST=ST+1

C       WRITE(4,*)SAV,S1,S2,E1,E2


      J=J+1
      GO TO 66
      ENDIF
    IF(BUFFER(J,I3).LE.0.AND.DIR(J,I2).EQ.8)THEN
      SUM=SUM+DIST

      ELV2=EL2(J,I3)

      ELV1=EL1(J,I2)
      E1=ELV1*1.0
      E2=ELV2*1.0
      S1=(E1-E2)/DIST
      IF(S1.EQ.0)S1=0.01
      S2=S1+S2
      ST=ST+1

C       WRITE(4,*)SAV,S1,S2,E1,E2


      GO TO 140
      ENDIF
```

```
      IF(BUFFER(J-1,I1).LE.0.AND.DIR(J,I2).EQ.64)THEN
          SUM=SUM+XDIST

          ELV2=EL2(J-1,I1)

          ELV1=EL1(J,I2)
          E1=ELV1*1.0
          E2=ELV2*1.0
          S1=(E1-E2)/XDIST
          IF(S1.EQ.0)S1=0.01
          S2=S1+S2
          ST=ST+1

C           WRITE(4,*)SAV,S1,S2,E1,E2


          J=J-1
          GO TO 160
          ENDIF
      IF(BUFFER(J+1,I1).LE.0.AND.DIR(J,I2).EQ.1)THEN
          SUM=SUM+XDIST

          ELV2=EL2(J+1,I1)

          ELV1=EL1(J,I2)
          E1=ELV1*1.0
          E2=ELV2*1.0
          S1=(E1-E2)/XDIST
          IF(S1.EQ.0)S1=0.01
          S2=S1+S2
          ST=ST+1

C           WRITE(4,*)SAV,S1,S2,E1,E2


          J=J+1
          GO TO 160
          ENDIF
      IF(BUFFER(J+1,I3).LE.0.AND.DIR(J,I2).EQ.4)THEN
          SUM=SUM+XDIST

          ELV2=EL2(J+1,I3)

          ELV1=EL1(J,I2)
          E1=ELV1*1.0
          E2=ELV2*1.0
```

```
      S1=(E1-E2)/XDIST
      IF(S1.EQ.0)S1=0.01
      S2=S1+S2
      ST=ST+1

C       WRITE(4,*)SAV,S1,S2,E1,E2


      J=J+1
      GO TO 140
      ENDIF
    IF(BUFFER(J-1,I3).LE.0.AND.DIR(J,I2).EQ.16)THEN
      SUM=SUM+XDIST

      ELV2=EL2(J-1,I3)

      ELV1=EL1(J,I2)
      E1=ELV1*1.0
      E2=ELV2*1.0
      S1=(E1-E2)/XDIST
      IF(S1.EQ.0)S1=0.01
      S2=S1+S2
      ST=ST+1

C       WRITE(4,*)SAV,S1,S2,E1,E2


      J=J-1
      GO TO 140
      ENDIF
    GO TO 180
140  ITEMP=I1
    I1=I2
    I2=I3
    I3=ITEMP
    M=M+1
    READ (14,REC=M) (DIR(K,I2),K=1,NS)
    READ (16,REC=M+1) (BUFFER(K,I3),K=1,NS)
    READ (8,REC=M) (EL1(K,I2),K=1,NS)
    READ (8,REC=M+1) (EL2(K,I3),K=1,NS)
    GO TO 66
160  M=M-1
    ITEMP=I3
    I3=I2
    I2=I1
    I1=ITEMP
```

```
      READ (14,REC=M)(DIR(K,I2),K=1,NS)
      READ (16,REC=M-1)(BUFFER(K,I1),K=1,NS)
      READ (8,REC=M) (EL1(K,I2),K=1,NS)
      READ (8,REC=M-1) (EL2(K,I1),K=1,NS)
      GO TO 66
180   CSUM=0.
      IF(BUFFER(J-1,I2).EQ.1.AND.DIR(J,I2).EQ.32)THEN
         SUM=SUM+DIST


         ELV2=EL2(J-1,I2)

         ELV1=EL1(J,I2)
         E1=ELV1*1.0
         E2=ELV2*1.0
         S1=(E1-E2)/DIST
         IF(S1.EQ.0)S1=0.01
         S2=S1+S2
         ST=ST+1


C          READ(LUNIT,REC=M)(ELEVS(K,1),K=1,NS)
C          MIDELV=ELEVS(J-1,1)
         J=J-1
         ACTIV=.FALSE.
         GO TO 66
         ENDIF
      IF(BUFFER(J,I1).EQ.1.AND.DIR(J,I2).EQ.128)THEN
         SUM=SUM+DIST


         ELV2=EL2(J,I1)

         ELV1=EL1(J,I2)
         E1=ELV1*1.0
         E2=ELV2*1.0
         S1=(E1-E2)/DIST
         IF(S1.EQ.0)S1=0.01
         S2=S1+S2
         ST=ST+1

C          READ(LUNIT,REC=M-1)(ELEVS(K,1),K=1,NS)
C          MIDELV=ELEVS(J,1)
         ACTIV=.FALSE.
         GO TO 160
         ENDIF
```

```
      IF(BUFFER(J+1,I2).EQ.1.AND.DIR(J,I2).EQ.2)THEN
        SUM=SUM+DIST


        ELV2=EL2(J+1,I2)

        ELV1=EL1(J,I2)
        E1=ELV1*1.0
        E2=ELV2*1.0
        S1=(E1-E2)/DIST
        IF(S1.EQ.0)S1=0.01
        S2=S1+S2
        ST=ST+1

C         READ(LUNIT,REC=M)(ELEVS(K,1),K=1,NS)
C         MIDELV=ELEVS(J+1,1)
        J=J+1
        ACTIV=.FALSE.
        GO TO 66
        ENDIF
      IF(BUFFER(J,I3).EQ.1.AND.DIR(J,I2).EQ.8)THEN
        SUM=SUM+DIST

        ELV2=EL2(J,I3)

        ELV1=EL1(J,I2)
        E1=ELV1*1.0
        E2=ELV2*1.0
        S1=(E1-E2)/DIST
        IF(S1.EQ.0)S1=0.01
        S2=S1+S2
        ST=ST+1

C         READ(LUNIT,REC=M+1)(ELEVS(K,1),K=1,NS)
C         MIDELV=ELEVS(J,1)
        ACTIV=.FALSE.
        GO TO 140
        ENDIF
      IF(BUFFER(J-1,I1).EQ.1.AND.DIR(J,I2).EQ.64)THEN
        SUM=SUM+XDIST

        ELV2=EL2(J-1,I1)

        ELV1=EL1(J,I2)
        E1=ELV1*1.0
        E2=ELV2*1.0
```

```
      S1=(E1-E2)/XDIST
      IF(S1.EQ.0)S1=0.01
      S2=S1+S2
      ST=ST+1

C         READ(LUNIT,REC=M-1)(ELEVS(K,1),K=1,NS)
C         MIDELV=ELEVS(J-1,1)
      J=J-1
      ACTIV=.FALSE.
      GO TO 160
      ENDIF
   IF(BUFFER(J+1,I1).EQ.1.AND.DIR(J,I2).EQ.1)THEN
      SUM=SUM+XDIST


      ELV2=EL2(J+1,I1)

      ELV1=EL1(J,I2)
      E1=ELV1*1.0
      E2=ELV2*1.0
      S1=(E1-E2)/XDIST
      IF(S1.EQ.0)S1=0.01
      S2=S1+S2
      ST=ST+1

C         READ(LUNIT,REC=M-1)(ELEVS(K,1),K=1,NS)
C         MIDELV=ELEVS(J+1,1)
      J=J+1
      ACTIV=.FALSE.
      GO TO 160
      ENDIF
   IF(BUFFER(J+1,I3).EQ.1.AND.DIR(J,I2).EQ.4)THEN
      SUM=SUM+XDIST

      ELV2=EL2(J+1,I3)

      ELV1=EL1(J,I2)
      E1=ELV1*1.0
      E2=ELV2*1.0
      S1=(E1-E2)/XDIST
      IF(S1.EQ.0)S1=0.01
      S2=S1+S2
      ST=ST+1

C         READ(LUNIT,REC=M+1)(ELEVS(K,1),K=1,NS)
C         MIDELV=ELEVS(J+1,1)
```

```
      J=J+1
      ACTIV=.FALSE.
      GO TO 140
      ENDIF
    IF(BUFFER(J-1,I3).EQ.1.AND.DIR(J,I2).EQ.16)THEN
      SUM=SUM+XDIST

      ELV2=EL2(J-1,I3)

      ELV1=EL1(J,I2)
      E1=ELV1*1.0
      E2=ELV2*1.0
      S1=(E1-E2)/XDIST
      IF(S1.EQ.0)S1=0.01
      S2=S1+S2
      ST=ST+1

C       READ(LUNIT,REC=M+1)(ELEVS(K,1),K=1,NS)
C       MIDELV=ELEVS(J-1,1)
      J=J-1
      ACTIV=.FALSE.
      GO TO 140
      ENDIF
185   IF(BUFFER(J-1,I2).EQ.1.AND.DIR(J,I2).EQ.32)THEN
      CSUM=CSUM+DIST


      ELV2=EL2(J-1,I2)

      ELV1=EL1(J,I2)
      E1=ELV1*1.0
      E2=ELV2*1.0
      SC1=(E1-E2)/DIST
      IF(SC1.EQ.0)S1=0.01
      SC2=SC1+SC2
      SCT=SCT+1



      J=J-1
      GO TO 66
      ENDIF
    IF(BUFFER(J,I1).EQ.1.AND.DIR(J,I2).EQ.128)THEN
      CSUM=CSUM+DIST

      ELV2=EL2(J,I1)
```

```
     ELV1=EL1(J,I2)
     E1=ELV1*1.0
     E2=ELV2*1.0
     SC1=(E1-E2)/DIST
     IF(SC1.EQ.0)S1=0.01
     SC2=SC1+SC2
     SCT=SCT+1


     GO TO 160
     ENDIF
IF(BUFFER(J+1,I2).EQ.1.AND.DIR(J,I2).EQ.2)THEN
     CSUM=CSUM+DIST

     ELV2=EL2(J+1,I2)

     ELV1=EL1(J,I2)
     E1=ELV1*1.0
     E2=ELV2*1.0
     SC1=(E1-E2)/DIST
     IF(SC1.EQ.0)S1=0.01
     SC2=SC1+SC2
     SCT=SCT+1


     J=J+1
     GO TO 66
     ENDIF
IF(BUFFER(J,I3).EQ.1.AND.DIR(J,I2).EQ.8)THEN
     CSUM=CSUM+DIST

     ELV2=EL2(J,I3)

     ELV1=EL1(J,I2)
     E1=ELV1*1.0
     E2=ELV2*1.0
     SC1=(E1-E2)/DIST
     IF(SC1.EQ.0)S1=0.01
     SC2=SC1+SC2
     SCT=SCT+1

     GO TO 140
     ENDIF
IF(BUFFER(J-1,I1).EQ.1.AND.DIR(J,I2).EQ.64)THEN
     CSUM=CSUM+XDIST
```

```
    ELV2=EL2(J-1,I1)

    ELV1=EL1(J,I2)
    E1=ELV1*1.0
    E2=ELV2*1.0
    SC1=(E1-E2)/XDIST
    IF(SC1.EQ.0)S1=0.01
    SC2=SC1+SC2
    SCT=SCT+1

    J=J-1
    GO TO 160
    ENDIF
IF(BUFFER(J+1,I1).EQ.1.AND.DIR(J,I2).EQ.1)THEN
    CSUM=CSUM+XDIST


    ELV2=EL2(J+1,I1)

    ELV1=EL1(J,I2)
    E1=ELV1*1.0
    E2=ELV2*1.0
    SC1=(E1-E2)/XDIST
    IF(SC1.EQ.0)S1=0.01
    SC2=SC1+SC2
    SCT=SCT+1

    J=J+1
    GO TO 160
    ENDIF
IF(BUFFER(J+1,I3).EQ.1.AND.DIR(J,I2).EQ.4)THEN
    CSUM=CSUM+XDIST

    ELV2=EL2(J+1,I3)

    ELV1=EL1(J,I2)
    E1=ELV1*1.0
    E2=ELV2*1.0
    SC1=(E1-E2)/XDIST
    IF(SC1.EQ.0)S1=0.01
    SC2=SC1+SC2
    SCT=SCT+1

    J=J+1
```

```
        GO TO 140
        ENDIF
      IF(BUFFER(J-1,I3).EQ.1.AND.DIR(J,I2).EQ.16)THEN
        CSUM=CSUM+XDIST


        ELV2=EL2(J-1,I3)

        ELV1=EL1(J,I2)
        E1=ELV1*1.0
        E2=ELV2*1.0
        SC1=(E1-E2)/XDIST
        IF(SC1.EQ.0)S1=0.01
        SC2=SC1+SC2
        SCT=SCT+1

        J=J-1
        GO TO 140
        ENDIF
      IF(BUFFER(J,I2).EQ.1)GO TO 190
      GO TO 191
190   IF(BUFFER(J-1,I2).EQ.Q.AND.DIR(J,I2).EQ.32)THEN
        CSUM=CSUM+DIST

        ELV2=EL2(J-1,I2)

        ELV1=EL1(J,I2)
        E1=ELV1*1.0
        E2=ELV2*1.0
        SC1=(E1-E2)/DIST
        IF(SC1.EQ.0)S1=0.01
        SC2=SC1+SC2
        SCT=SCT+1

        GO TO 195
        ENDIF
      IF(BUFFER(J,I1).EQ.Q.AND.DIR(J,I2).EQ.128)THEN
        CSUM=CSUM+DIST

        ELV2=EL2(J,I1)

        ELV1=EL1(J,I2)
        E1=ELV1*1.0
        E2=ELV2*1.0
        SC1=(E1-E2)/DIST
        IF(SC1.EQ.0)S1=0.01
```

```
      SC2=SC1+SC2
      SCT=SCT+1

      GO TO 195
      ENDIF
   IF(BUFFER(J+1,I2).EQ.Q.AND.DIR(J,I2).EQ.2)THEN
      CSUM=CSUM+DIST

      ELV2=EL2(J+1,I2)

      ELV1=EL1(J,I2)
      E1=ELV1*1.0
      E2=ELV2*1.0
      SC1=(E1-E2)/DIST
      IF(SC1.EQ.0)S1=0.01
      SC2=SC1+SC2
      SCT=SCT+1

      GO TO 195
      ENDIF
   IF(BUFFER(J,I3).EQ.Q.AND.DIR(J,I2).EQ.8)THEN
      CSUM=CSUM+DIST

      ELV2=EL2(J,I3)

      ELV1=EL1(J,I2)
      E1=ELV1*1.0
      E2=ELV2*1.0
      SC1=(E1-E2)/DIST
      IF(SC1.EQ.0)S1=0.01
      SC2=SC1+SC2
      SCT=SCT+1


      GO TO 195
      ENDIF
   IF(BUFFER(J-1,I1).EQ.Q.AND.DIR(J,I2).EQ.64)THEN
      CSUM=CSUM+XDIST

      ELV2=EL2(J-1,I1)

      ELV1=EL1(J,I2)
      E1=ELV1*1.0
      E2=ELV2*1.0
      SC1=(E1-E2)/XDIST
      IF(SC1.EQ.0)S1=0.01
```

```
    SC2=SC1+SC2
    SCT=SCT+1

    GO TO 195
    ENDIF
IF(BUFFER(J+1,I1).EQ.Q.AND.DIR(J,I2).EQ.1)THEN
    CSUM=CSUM+XDIST

    ELV2=EL2(J+1,I1)

    ELV1=EL1(J,I2)
    E1=ELV1*1.0
    E2=ELV2*1.0
    SC1=(E1-E2)/XDIST
    IF(SC1.EQ.0)S1=0.01
    SC2=SC1+SC2
    SCT=SCT+1

    GO TO 195
    ENDIF
IF(BUFFER(J+1,I3).EQ.Q.AND.DIR(J,I2).EQ.4)THEN
    CSUM=CSUM+XDIST

    ELV2=EL2(J+1,I3)

    ELV1=EL1(J,I2)
    E1=ELV1*1.0
    E2=ELV2*1.0
    SC1=(E1-E2)/XDIST
    IF(SC1.EQ.0)S1=0.01
    SC2=SC1+SC2
    SCT=SCT+1

    GO TO 195
    ENDIF
IF(BUFFER(J-1,I3).EQ.Q.AND.DIR(J,I2).EQ.16)THEN
    CSUM=CSUM+XDIST

    ELV2=EL2(J-1,I3)

    ELV1=EL1(J,I2)
    E1=ELV1*1.0
    E2=ELV2*1.0
    SC1=(E1-E2)/XDIST
    IF(SC1.EQ.0)S1=0.01
    SC2=SC1+SC2
```

```
      SCT=SCT+1

      GO TO 195
      ENDIF

C     **************************************************
191   CSUM=0
    IF(BUFFER(J-1,I2).EQ.Q.AND.DIR(J,I2).EQ.32)THEN
      SUM=SUM+DIST

      ELV2=EL2(J-1,I2)

      ELV1=EL1(J,I2)
      E1=ELV1*1.0
      E2=ELV2*1.0
      S1=(E1-E2)/DIST
      IF(S1.EQ.0)S1=0.01
      S2=S1+S2
      ST=ST+1

      GO TO 195
      ENDIF
    IF(BUFFER(J,I1).EQ.Q.AND.DIR(J,I2).EQ.128)THEN
      SUM=SUM+DIST


      ELV2=EL2(J,I1)

      ELV1=EL1(J,I2)
      E1=ELV1*1.0
      E2=ELV2*1.0
      S1=(E1-E2)/DIST
      IF(S1.EQ.0)S1=0.01
      S2=S1+S2
      ST=ST+1

      GO TO 195
      ENDIF
    IF(BUFFER(J+1,I2).EQ.Q.AND.DIR(J,I2).EQ.2)THEN
      SUM=SUM+DIST

      ELV2=EL2(J+1,I2)

      ELV1=EL1(J,I2)
      E1=ELV1*1.0
      E2=ELV2*1.0
```

```
      S1=(E1-E2)/DIST
      IF(S1.EQ.0)S1=0.01
      S2=S1+S2
      ST=ST+1

      GO TO 195
      ENDIF
IF(BUFFER(J,I3).EQ.Q.AND.DIR(J,I2).EQ.8)THEN
      SUM=SUM+DIST

      ELV2=EL2(J,I3)

      ELV1=EL1(J,I2)
      E1=ELV1*1.0
      E2=ELV2*1.0
      S1=(E1-E2)/DIST
      IF(S1.EQ.0)S1=0.01
      S2=S1+S2
      ST=ST+1

      GO TO 195
      ENDIF
IF(BUFFER(J-1,I1).EQ.Q.AND.DIR(J,I2).EQ.64)THEN
      SUM=SUM+XDIST

      ELV2=EL2(J-1,I1)

      ELV1=EL1(J,I2)
      E1=ELV1*1.0
      E2=ELV2*1.0
      S1=(E1-E2)/XDIST
      IF(S1.EQ.0)S1=0.01
      S2=S1+S2
      ST=ST+1

      GO TO 195
      ENDIF
IF(BUFFER(J+1,I1).EQ.Q.AND.DIR(J,I2).EQ.1)THEN
      SUM=SUM+XDIST

      ELV2=EL2(J+1,I1)

      ELV1=EL1(J,I2)
      E1=ELV1*1.0
      E2=ELV2*1.0
      S1=(E1-E2)/XDIST
```

```
          IF(S1.EQ.0)S1=0.01
          S2=S1+S2
          ST=ST+1

          GO TO 195
          ENDIF
      IF(BUFFER(J+1,I3).EQ.Q.AND.DIR(J,I2).EQ.4)THEN
          SUM=SUM+XDIST

          ELV2=EL2(J+1,I3)

          ELV1=EL1(J,I2)
          E1=ELV1*1.0
          E2=ELV2*1.0
          S1=(E1-E2)/XDIST
          IF(S1.EQ.0)S1=0.01
          S2=S1+S2
          ST=ST+1

          GO TO 195
          ENDIF
      IF(BUFFER(J-1,I3).EQ.Q.AND.DIR(J,I2).EQ.16)THEN
          SUM=SUM+XDIST

          ELV2=EL2(J-1,I3)

          ELV1=EL1(J,I2)
          E1=ELV1*1.0
          E2=ELV2*1.0
          S1=(E1-E2)/XDIST
          IF(S1.EQ.0)S1=0.01
          S2=S1+S2
          ST=ST+1

          GO TO 195
          ENDIF
C195   CDROP=MIDELV-BOTELV
C     IF(CDROP.EQ.0)CSLOPE=0.01
C     IF(CSUM.GT.0.AND.CDROP.GT.0)CSLOPE=CDROP/CSUM

C     TDROP=TOPELV-BOTELV
195     TSUM=SUM+CSUM
C     TSLOPE=TDROP/TSUM
C     IF(TSLOPE.LT.0.01)TSLOPE=0.01
C     ODROP=TDROP-CDROP
C     IF(SUM.GT.0)OSLOPE=ODROP/SUM
```

```
      IF(TSUM.GT.MAXSUM)THEN
         MAXSUM=TSUM
C           SLOPE=TSLOPE
          ENDIF
      IF(CSUM.GT.MXCSUM)THEN
          MXCSUM=CSUM
C           MXSLOP=CSLOPE
          ENDIF

C     GETTING THE AVERAGE SLOPE FOR THE CELL
C

       SAV=S2/(ST*1.0)
       IF(ST.GT.0.AND.S2.EQ.0)SAV=0.001
       IF(ST.LE.0.)SAV=0
       SCAV=SC2/(SCT*1.0)
       IF(SCT.GT.0.AND.SC2.EQ.0)SCAV=0.001
       IF(SCT.EQ.0)SCAV=0


       SB=SB+SAV
       IF(SUM.GT.0)OPATH=OPATH+1
       SCB=SCB+SCAV
       IF(CSUM.GT.0)CPATH=CPATH+1
C     GETTING THE VALUE OF THE VELOCITY FOR THE CELL
C

        IF(CNN.EQ.0)THEN
        T=0
        TT=0
        T_C=0
        GOTO 196
        ENDIF
          CALL VELO(VEL,SAV,CNN)
          IF(VEL.EQ.0)T_CO=0
        IF(VEL.GT.0)T_CO=SUM/(VEL*60)

C       CALL VELC(VEL_C,CSLOPE)
        IF(SCAV.GT.0)T_CC=0.0195*((CSUM**0.77)/(SCAV**0.385))
        IF(T_CC.GT.T_CCMX)T_CCMX=T_CC
        T_C=(T_CO+T_CC)
        T_C=T_C/INTERV
        TT=INT(T_C)
        TT=(TT+1)*INTERV
        T=TT
        T_C=T_C*INTERV
```

```
      IF(T.GT.TMAX)TMAX=T
      COL=Z-1
      ROW=ROW-1
      NEWBUF(Z,W)=TT

C       WRITE (18,REC=E) (NEWBUF(K,W),K=1,NS)
C       WRITE (12,REC=E) (NEWBUF(K,W),K=1,NS)
196     WRITE (18,REC=E) (NEWBUF(K,W),K=1,NS)
      WRITE(3,*)COL,ROW,SUM,CSUM,SCAV,SAV
      WRITE(4,*)ST,S2,SAV
      SAV=0.0
      SCAV=0.0
      S1=0.0
      S2=0.0
      SC1=0.0
      SC2=0.0
      ST=0.0
      SCT=0.0

C       CNN,VEL,T_CO,T_CC,T_C,TT,T




     GO TO 170
170   IF(KOUNT.EQ.50)THEN
     PRINT*,'CELL',KOUNT1
     KOUNT=0
     ENDIF
198   GO TO 60
200   KOUNT1=KOUNT1-2
     CLOSE(12)
     CLOSE(14)
C     AVERAGE BASIN SLOPES
     SAVB=SB/(OPATH*1.0)
     SCAVB=SCB/(CPATH*1.0)

     OPEN(60,FILE='DATA.TXT',STATUS='UNKNOWN')
     WRITE(60,201)Q
     PRINT 201,Q
201   FORMAT(//,'WATERSHED NUMBER',I3)
     PRINT 202,CELLNUM
     WRITE (60,202)CELLNUM
202   FORMAT(//,'NUMBER OF CELLS =',I6)
     PRINT 203,TMAX
```

```
      WRITE(60,203)TMAX
203   FORMAT(//,'TC:',F5.1)
      PRINT 204,MAXSUM
      WRITE(60,204)MAXSUM
204   FORMAT(//,'LONGEST FLOWPATH =',F9.2,' METERS')
      PRINT 205,T_CCMX
      WRITE(60,205)T_CCMX
205   FORMAT(//,'TC CHANNEL:',F5.1)
      PRINT 206,MXCSUM
      WRITE(60,206)MXCSUM
206   FORMAT(//,'LENGTH OF MAINSTREAM =',F9.2,' METERS')

      PRINT 207,SAVB
      WRITE(60,207)SAVB
207   FORMAT(//,'AVERAGE OVERLAND SLOPE:',2X,F9.8)
      PRINT 208,SCAVB
      WRITE(60,208)SCAVB
208   FORMAT(//,'AVERAGE CHANNEL SLOPE:',2X,F9.8)
      PRINT 209
      WRITE(60,209)
209   FORMAT('-----------------------------------------------------')

      SAVB=0.0
      SCAVB=0.0
      SB=0
      SCB=0
      OPATH=0
      CPATH=0
      CELLNUM=0
      Q=Q-1
      IF(Q.NE.1)THEN




      GOTO 10
      ENDIF


       OPEN(2,FILE=ISORDC,STATUS='UNKNOWN')
        PRINT 110
        WRITE(2,110)
110   FORMAT(0X,'file format : IDRISI Raster A.1')
        PRINT 111
        WRITE(2,111)
```

```
111   FORMAT(0X,'file title  : ')
      PRINT 112
      WRITE(2,112)
112   FORMAT(0X,'data type   : integer')
      PRINT 113
      WRITE(2,113)
113   FORMAT(0X,'file type   : binary')
      PRINT 114, NS
      WRITE(2,114) NS
114   FORMAT(0X,'columns     : ',I3)
      PRINT 115,NL
      WRITE(2,115) NL
115   FORMAT(0X,'rows        : ',I3)
      PRINT 116
      WRITE(2,116)
116   FORMAT(0X,'ref. system : us27tm17')
      PRINT 117
      WRITE(2,117)
117   FORMAT(0X,'ref. units  : m')
      PRINT 118
      WRITE(2,118)
118   FORMAT(0X,'unit dist.  : 1.0000000')
      PRINT 119, MN_X
      WRITE(2,119) MN_X
119   FORMAT(0X,'min. X      : ',F9.0)
      PRINT 120, MX_X
      WRITE(2,120) MX_X
120   FORMAT(0X,'max. X      : ',F9.0)
      PRINT 121, MN_Y
      WRITE(2,121) MN_Y
121   FORMAT(0X,'min. Y      : ',F9.0)
      PRINT 122, MX_Y
      WRITE(2,122) MX_Y
122   FORMAT(0X,'max. Y      : ',F9.0)
      PRINT 123
      WRITE(2,123)
123   FORMAT(0X,"pos'n error : unknown")
      PRINT 124
      WRITE(2,124)
124   FORMAT(0X,'resolution  : 30.0273323')
      PRINT 125
      WRITE(2,125)
125   FORMAT(0X,'min. value  : 0')
      PRINT 126,TMAX
      WRITE(2,126) TMAX
126   FORMAT(0X,'max. value  :',F5.1)
```

```
      PRINT 127
      WRITE(2,127)
127   FORMAT(0X,'display min : 0')
      PRINT 128,TMAX
      WRITE(2,128) TMAX
128   FORMAT(0X,'display max :',F5.1)
      PRINT 129
      WRITE(2,129)
129   FORMAT(0X,'value units : m')
      PRINT 130
      WRITE(2,130)
130   FORMAT(0X,'value error : unknown')
     PRINT 131
      WRITE(2,131)
131   FORMAT(0X,'flag value  : 0')
      PRINT 132
      WRITE(2,132)
132   FORMAT(0X,"flag def'n  : background")
      PRINT 133
      WRITE(2,133)
133   FORMAT(0X,'legend cats : 0')

C     OPEN(1,FILE='DATA.TXT',STATUS='UNKNOWN')
C     WRITE(1,201)Q
C     PRINT 201,Q
C201   FORMAT(//,'WATERSHED NUMBER',I3)
C     PRINT 202,KOUNT1
C     WRITE (1,202) KOUNT1
C202   FORMAT(//,'NUMBER OF CELLS =',I7)
C     PRINT 203,CN
C     WRITE (1,203) CN
C203   FORMAT(//,'CURVE NUMBER =',I3)
C     PRINT 204,AVE_PE
C        WRITE (1,214) AVE_PE
C214   FORMAT(//,'AVERAGE EXCESS RAINFALL FOR WATERSHED = ',F5.2)
C     PRINT 204,MAXSUM,SLOPE
C     WRITE(1,204)MAXSUM,SLOPE
C204   FORMAT(//,'LONGEST FLOWPATH =',F9.2,' METERS',4X,'SLOPE = ',F7.4)
C     PRINT 205,MXCSUM,MXSLOP
C     WRITE(1,205)MXCSUM,MXSLOP
C205   FORMAT('LENGTH OF MAINSTREAM =',F9.2,' METERS',4X,'SLOPE =',F7.4)
C     PRINT 206,TMAX1
C     WRITE (1,206) TMAX1
C206   FORMAT('TIME OF CONCENTRATION =',F9.2,' MINUTES')
C     PRINT 208,INTERV
C     WRITE (1,208)INTERV
```

```
C208   FORMAT('TIME INTERVAL =',I3,' MINUTES',/)
C      PRINT 210,INTERV
C      WRITE(1,210)INTERV
C210   FORMAT('NO. OF CELLS PER TIME INTERVAL, STARTING AT',I3,' MIN')
C      PRINT *,(TA(T), T=INTERV,TMAX,INTERV)
C      WRITE (1,*) (TA(T), T=INTERV,TMAX,INTERV)
C      WRITE(2,213)TOT
C      WRITE(2,213)INTERV
C     WRITE(5,213)TOT
C      WRITE(5,213)INTERV
C     OPEN(2,FILE='TA.TXT',STATUS='UNKNOWN')
C      DO 211 T=INTERV,TMAX,INTERV
C      WRITE(2,213)(TA(T))
C      WRITE(5,212)(EX_PE(T))
C211   CONTINUE
C211   WRITE(2,213)(TA(T))
C212   FORMAT(F15.5)
C213   FORMAT(I5)

210   STOP
      END

C
C
C     CALCUTATING THE VELOCITY FOR THE CELL
C     BASED ON THE SLOPE OF THE CELL USING THE
C     USING A FORMULA DEVELOPED FROM THE USDA 1986
C     GRAPH WHICH RELATES THE SLOPE AND THE
C     VELOCITY OF SHEET FLOW IN METERS PER SEC
C
C
      SUBROUTINE VELO(V,I,CNN)
          REAL I,V,CNN


      IF((CNN.GE.0).AND.(CNN.LE.50)) THEN
      V=0.041*EXP(0.2951*I)
      ELSEIF((CNN.GE.51).AND.(CNN.LE.60)) THEN
      V=0.0813*EXP(0.2836*I)
      ELSEIF((CNN.GE.61).AND.(CNN.LE.70)) THEN
      V=0.1217*EXP(0.2861*I)
      ELSEIF((CNN.GE.71).AND.(CNN.LE.75)) THEN
      V=0.1686*EXP(0.2882*I)
      ELSEIF((CNN.GE.76).AND.(CNN.LE.85)) THEN
      V=0.2533*EXP(0.2925*I)
      ELSEIF((CNN.GE.86).AND.(CNN.LE.100)) THEN
```

```
      V=0.3378*EXP(0.2885*I)
      ENDIF
      RETURN
      END
C
C
C    ESTIMATING CHANNEL VELOCITY BASED ON SLOPE
C
C     SUBROUTINE VELC(VC,SC)
C     REAL VC,SC
C     VC=((1.49/0.05)*(0.438)*(SQRT(SC)))*0.3048
C     RETURN
C     END
C
```

Gridded Precipitation Program

```
C       ************************************************************
C       CREATED BY JAMES EMERICK
C
C       THIS PROGRAM USES A TEXT FILE CONTAINING INCREMENTAL PRECIP
C       FOR CELLS IN THE WATERSHED TO GENERATE PRECIP RASTER FILES
C      ************************************************************
C


      CHARACTER*80 WFILE,PFILE,MFILE
      CHARACTER*80 ELFILE,ELRDC,DIRFILE,DIRRDC,SFILE,SRDC,WSRDC,ISORDC
      CHARACTER*80 WSFILE,SOILFILE,LANDFILE,CNFILE,D,CNRDC,ISOFILE

      INTEGER DUR,INC,PN,F,NL,NS,R,Z,NC,C
      REAL P(20,40),P2(50)
      REAL MXP,MAXDISP
      REAL MN_X,MX_X,MN_Y,MX_Y
      INTEGER*2 WBUF(1500,1)
      REAL*4 PRECBUF(1500,1)
      PRINT 1
1      FORMAT(/,'*** ENTER THE DURATION OF THE STORM (MIN)***')
      READ(5,*)DUR
      PRINT 2
2      FORMAT('*** ENTER PRECIP TIME INCREMENT (MIN)***')
      READ (5,*)INC
      PRINT 4
4      FORMAT('*** ENTER THE NUMBER OF PRECIP CELLS ***')
      READ (5,*)NC
      PRINT 5
5      FORMAT('*** ENTER THE MAXIMUM DISPLAY VALUE ***')
      READ (5,*)MAXDISP
C       NL=468
C       NS=359
      PN=DUR/INC
C       NC=27

      OPEN(3,FILE='PROJINFO.TXT',STATUS='UNKNOWN')
       READ(3,*)NL
       READ(3,*)NS
       READ(3,*)ELFILE
       READ(3,*)ELRDC
       READ(3,*)DIRFILE
       READ(3,*)DIRRDC
```

```
       READ(3,*)SFILE
       READ(3,*)SRDC
       READ(3,*)WSFILE
       READ(3,*)WSRDC
       READ(3,*)CNFILE
       READ(3,*)CNRDC
       READ(3,*)D
       READ(3,*)D
       READ(3,*)MN_X
       READ(3,*)MX_X
       READ(3,*)MN_Y
       READ(3,*)MX_Y
      CLOSE(3,STATUS='KEEP')




C1000  FORMAT(A80)
C
C     READ INC PRECIP
C
      OPEN(UNIT=4,STATUS='OLD',FORM='UNFORMATTED',
     *ACCESS='DIRECT',FILE='GRID.RST',RECL=NS*2)
      OPEN(3,FILE='INCPREC.TXT',STATUS='UNKNOWN')

      DO 6 I=1,PN
      READ(3,*) (P(I,J),J=1,NC)
6     CONTINUE
C     R IS THE LINE NUMBER EACH TIME THROUGH
C     F IS THE TIME OF THE PRECIP FILE BEING WRITTEN
      R=10
      F=0
10    F=F+1
      IF(F.EQ.1)THEN
      PFILE="PRE1.RST"
      MFILE="PRE1.RDC"
      GOTO 40
      ENDIF
      IF(F.EQ.2)THEN
      PFILE="PRE2.RST"
      MFILE="PRE2.RDC"
      GOTO 40
      ENDIF
      IF(F.EQ.3)THEN
      PFILE="PRE3.RST"
      MFILE="PRE3.RDC"
```

```
GOTO 40
ENDIF
IF(F.EQ.4)THEN
PFILE="PRE4.RST"
MFILE="PRE4.RDC"
GOTO 40
ENDIF
IF(F.EQ.5)THEN
PFILE="PRE5.RST"
MFILE="PRE5.RDC"
GOTO 40
ENDIF
IF(F.EQ.6)THEN
PFILE="PRE6.RST"
MFILE="PRE6.RDC"
GOTO 40
ENDIF
IF(F.EQ.7)THEN
PFILE="PRE7.RST"
MFILE="PRE7.RDC"
GOTO 40
ENDIF
IF(F.EQ.8)THEN
PFILE="PRE8.RST"
MFILE="PRE8.RDC"
GOTO 40
ENDIF
IF(F.EQ.9)THEN
PFILE="PRE9.RST"
MFILE="PRE9.RDC"
GOTO 40
ENDIF
IF(F.EQ.10)THEN
PFILE="PRE10.RST"
MFILE="PRE10.RDC"
GOTO 40
ENDIF
IF(F.EQ.11)THEN
PFILE="PRE11.RST"
MFILE="PRE11.RDC"
GOTO 40
ENDIF
IF(F.EQ.12)THEN
PFILE="PRE12.RST"
MFILE="PRE12.RDC"
GOTO 40
```

```
      ENDIF
      IF(F.EQ.13)THEN
      PFILE="PRE13.RST"
      MFILE="PRE13.RDC"
      GOTO 40
      ENDIF
      IF(F.EQ.14)THEN
      PFILE="PRE14.RST"
      MFILE="PRE14.RDC"
      GOTO 40
      ENDIF
      IF(F.EQ.15)THEN
      PFILE="PRE15.RST"
      MFILE="PRE15.RDC"
      GOTO 40
      ENDIF
      IF(F.EQ.16)THEN
      PFILE="PRE16.RST"
      MFILE="PRE16.RDC"
      GOTO 40
      ENDIF
      IF(F.EQ.17)THEN
      PFILE="PRE17.RST"
      MFILE="PRE17.RDC"
      GOTO 40
      ENDIF
      IF(F.EQ.18)THEN
      PFILE="PRE18.RST"
      MFILE="PRE18.RDC"
      GOTO 40
      ENDIF
      IF(F.EQ.19)THEN
      PFILE="PRE19.RST"
      MFILE="PRE19.RDC"
      GOTO 40
      ENDIF
      IF(F.EQ.20)THEN
      PFILE="PRE20.RST"
      MFILE="PRE20.RDC"
      GOTO 40
      ENDIF
40    MXP=0
      I=0
      K=0
      OPEN(1,STATUS='UNKNOWN',ACCESS='DIRECT',FILE=PFILE,
     * RECL=NS*4,FORM='UNFORMATTED')
```

```
C    WRITE A BLANK RASTER FILE

      DO 43 I=1,NL
      DO 42 K=1,NS
      PRECBUF(K,1)=0
      WRITE (UNIT=1,REC=I) (PRECBUF(K,1))
42    CONTINUE
43    CONTINUE
C
C    FINDING WATERSHED CELLS AND WRITING PREC
C
C

      DO 44 I=1,NL
      READ (4,REC=I) (WBUF(K,1),K=1,NS)
      DO 45 K=1,NS
C      WBUF(K,1)=C
C      IF(C.EQ.0)PRECBUF(K,1)=0
C      IF(C.GT.0)PRECBUF(K,1)=P(F,C)
      IF(WBUF(K,1).EQ.0)PRECBUF(K,1)=0
      IF(WBUF(K,1).EQ.1)PRECBUF(K,1)=P(F,1)
      IF(WBUF(K,1).EQ.2)PRECBUF(K,1)=P(F,2)
      IF(WBUF(K,1).EQ.3)PRECBUF(K,1)=P(F,3)
      IF(WBUF(K,1).EQ.4)PRECBUF(K,1)=P(F,4)
      IF(WBUF(K,1).EQ.5)PRECBUF(K,1)=P(F,5)
      IF(WBUF(K,1).EQ.6)PRECBUF(K,1)=P(F,6)
      IF(WBUF(K,1).EQ.7)PRECBUF(K,1)=P(F,7)
      IF(WBUF(K,1).EQ.8)PRECBUF(K,1)=P(F,8)
      IF(WBUF(K,1).EQ.9)PRECBUF(K,1)=P(F,9)
      IF(WBUF(K,1).EQ.10)PRECBUF(K,1)=P(F,10)
      IF(WBUF(K,1).EQ.11)PRECBUF(K,1)=P(F,11)
      IF(WBUF(K,1).EQ.12)PRECBUF(K,1)=P(F,12)
      IF(WBUF(K,1).EQ.13)PRECBUF(K,1)=P(F,13)
      IF(WBUF(K,1).EQ.14)PRECBUF(K,1)=P(F,14)
      IF(WBUF(K,1).EQ.15)PRECBUF(K,1)=P(F,15)
      IF(WBUF(K,1).EQ.16)PRECBUF(K,1)=P(F,16)
      IF(WBUF(K,1).EQ.17)PRECBUF(K,1)=P(F,17)
      IF(WBUF(K,1).EQ.18)PRECBUF(K,1)=P(F,18)
      IF(WBUF(K,1).EQ.19)PRECBUF(K,1)=P(F,19)
```

```
      IF(WBUF(K,1).EQ.20)PRECBUF(K,1)=P(F,20)
      IF(WBUF(K,1).EQ.21)PRECBUF(K,1)=P(F,21)
      IF(WBUF(K,1).EQ.22)PRECBUF(K,1)=P(F,22)
      IF(WBUF(K,1).EQ.23)PRECBUF(K,1)=P(F,23)
      IF(WBUF(K,1).EQ.24)PRECBUF(K,1)=P(F,24)
      IF(WBUF(K,1).EQ.25)PRECBUF(K,1)=P(F,25)
      IF(WBUF(K,1).EQ.26)PRECBUF(K,1)=P(F,26)
      IF(WBUF(K,1).EQ.27)PRECBUF(K,1)=P(F,27)
      IF(WBUF(K,1).EQ.28)PRECBUF(K,1)=P(F,28)
      IF(WBUF(K,1).EQ.29)PRECBUF(K,1)=P(F,29)
      IF(WBUF(K,1).EQ.30)PRECBUF(K,1)=P(F,30)
      IF(WBUF(K,1).EQ.31)PRECBUF(K,1)=P(F,31)
      IF(WBUF(K,1).EQ.32)PRECBUF(K,1)=P(F,32)
      IF(WBUF(K,1).EQ.33)PRECBUF(K,1)=P(F,33)
      IF(WBUF(K,1).EQ.34)PRECBUF(K,1)=P(F,34)
      IF(WBUF(K,1).EQ.35)PRECBUF(K,1)=P(F,35)
      IF(WBUF(K,1).EQ.36)PRECBUF(K,1)=P(F,36)
      IF(WBUF(K,1).EQ.37)PRECBUF(K,1)=P(F,37)
      IF(WBUF(K,1).EQ.38)PRECBUF(K,1)=P(F,38)
      IF(WBUF(K,1).EQ.39)PRECBUF(K,1)=P(F,39)


      IF(PRECBUF(K,1).GT.MXP) MXP=PRECBUF(K,1)

45    CONTINUE
      WRITE (UNIT=1,REC=I) (PRECBUF(K,1),K=1,NS)
44    CONTINUE


      OPEN(2,FILE=MFILE,STATUS='UNKNOWN')
       PRINT 110
       WRITE(2,110)
110    FORMAT(0X,'file format : IDRISI Raster A.1')
       PRINT 111
       WRITE(2,111)
111    FORMAT(0X,'file title  : ')
       PRINT 112
       WRITE(2,112)
112    FORMAT(0X,'data type   : real')
       PRINT 113
       WRITE(2,113)
113    FORMAT(0X,'file type   : binary')
```

```
      PRINT 114, NS
      WRITE(2,114) NS
114   FORMAT(0X,'columns     : ',I3)
      PRINT 115, NL
      WRITE(2,115) NL
115   FORMAT(0X,'rows        : ',I3)
      PRINT 116
      WRITE(2,116)
116   FORMAT(0X,'ref. system : us27tm17')
      PRINT 117
      WRITE(2,117)
117   FORMAT(0X,'ref. units  : m')
      PRINT 118
      WRITE(2,118)
118   FORMAT(0X,'unit dist.  : 1.0000000')
      PRINT 119, MN_X
      WRITE(2,119) MN_X
119   FORMAT(0X,'min. X      : ',F9.0)
      PRINT 120, MX_X
      WRITE(2,120) MX_X
120   FORMAT(0X,'max. X      : ',F9.0)
      PRINT 121, MN_Y
      WRITE(2,121) MN_Y
121   FORMAT(0X,'min. Y      : ',F9.0)
      PRINT 122, MX_Y
      WRITE(2,122) MX_Y
122   FORMAT(0X,'max. Y      : ',F9.0)
      PRINT 123
      WRITE(2,123)
123   FORMAT(0X,"pos'n error : unknown")
      PRINT 124
      WRITE(2,124)
124   FORMAT(0X,'resolution  : 30.0273323')
      PRINT 125
      WRITE(2,125)
125   FORMAT(0X,'min. value  : 0')
      PRINT 126,MXP
      WRITE(2,126) MXP
126   FORMAT(0X,'max. value  :',F7.4)
      PRINT 127
      WRITE(2,127)
127   FORMAT(0X,'display min : 0')
      PRINT 128,MAXDISP
      WRITE(2,128) MAXDISP
128   FORMAT(0X,'display max :',F7.4)
      PRINT 129
```

```
      WRITE(2,129)
129   FORMAT(0X,'value units : m')
      PRINT 130
      WRITE(2,130)
130   FORMAT(0X,'value error : unknown')
      PRINT 131
      WRITE(2,131)
131   FORMAT(0X,'flag value  : 0')
      PRINT 132
      WRITE(2,132)
132   FORMAT(0X,"flag def'n  : background")
      PRINT 133
      WRITE(2,133)
133   FORMAT(0X,'legend cats : 0')

     R=R+1
     CLOSE(UNIT=1,STATUS='KEEP')
     CLOSE(UNIT=2,STATUS='KEEP')
     REWIND(UNIT=1)
     REWIND(UNIT=2)
     IF(F.LT.PN)GO TO 10

136   END
```

Runoff Program

```
C     ****************************************************************
C      This program was written by James Emerick
C      the program uses the curve number method to determine
C      the amount of runoff generated for each cell the
C      isochrone file isthen used to order the values in a time series
C     ****************************************************************

      INTEGER*2 ISOBUF(1500,3),WBUF(1500,3),CNBUF(1500,3)
      REAL*4 PRECBUF1(1500,3),PRECBUF2(1500,3),PRECBUF3(1500,3)
      REAL*4 PRECBUF4(1500,3),PRECBUF5(1500,3),PRECBUF6(1500,3)
      REAL*4 PRECBUF7(1500,3),PRECBUF8(1500,3),PRECBUF9(1500,3)
      REAL*4 PRECBUF10(1500,3),PRECBUF11(1500,3)
      REAL*4 PRECBUF12(1500,3),PRECBUF13(1500,3)
      REAL*4 PRECBUF14(1500,3),PRECBUF15(1500,3)
      REAL*4 PRECBUF16(1500,3),PRECBUF17(1500,3)
      REAL*4 PRECBUF18(1500,3),PRECBUF19(1500,3)
      REAL*4 PRECBUF20(1500,3)
      REAL RET,CNN,SIZE
      REAL P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,P11,P12
      REAL P13,P14,P15,P16,P17,P18,P19,P20
      REAL PE1,PE2,PE3,PE4,PE5,PE6,PE7,PE8,PE9,PE10,PE11,PE12
      REAL PE13,PE14,PE15,PE16,PE17,PE18,PE19,PE20
      REAL PECUM2,PECUM3,PECUM4,PECUM5,PECUM6
      REAL PECUM7,PECUM8,PECUM9,PECUM10,PECUM11,PECUM12
      REAL PECUM13,PECUM14,PECUM15,PECUM16,PECUM17,PECUM18
      REAL PECUM19,PECUM20
      REAL PCUM2,PCUM3,PCUM4,PCUM5,PCUM6
      REAL PCUM7,PCUM8,PCUM9,PCUM10,PCUM11,PCUM12
      REAL PCUM13,PCUM14,PCUM15,PCUM16,PCUM17,PCUM18
      REAL PCUM19,PCUM20
      INTEGER MAXT,MAXINT,N,Z,NMAXINT,O
      INTEGER Q,INTERV,CN,T,R,ISO,TMAX,E,PN,PINC,PDUR
      REAL SUM_PE1(9000),SUM_PE2(9000),SUM_PE3(9000)
      REAL SUM_PE4(9000),SUM_PE5(9000),SUM_PE6(9000)
      REAL SUM_PE7(9000),SUM_PE8(9000),SUM_PE9(9000)
      REAL SUM_PE10(9000),SUM_PE11(9000),SUM_PE12(9000)
      REAL SUM_PE13(9000),SUM_PE14(9000),SUM_PE15(9000)
      REAL SUM_PE16(9000),SUM_PE17(9000),SUM_PE18(9000)
      REAL SUM_PE19(9000),SUM_PE20(9000)

      REAL SUM_Q1(9000),SUM_Q2(9000),SUM_Q3(9000)
      REAL SUM_Q4(9000),SUM_Q5(9000),SUM_Q6(9000)
```

```
      REAL SUM_Q7(9000),SUM_Q8(9000),SUM_Q9(9000)
      REAL SUM_Q10(9000),SUM_Q11(9000),SUM_Q12(9000)
      REAL SUM_Q13(9000),SUM_Q14(9000),SUM_Q15(9000)
      REAL SUM_Q16(9000),SUM_Q17(9000),SUM_Q18(9000)
      REAL SUM_Q19(9000),SUM_Q20(9000)
      REAL MN_X,MX_X,MN_Y,MX_Y
      CHARACTER*80 ELFILE,ELRDC,DIRFILE,DIRRDC,SFILE,SRDC,WSRDC,ISORDC
      CHARACTER*80 WSFILE,SOILFILE,LANDFILE,CNFILE,D,CNRDC,ISOFILE
      INTEGER NL,NS,AMC
******************************************************************
      REAL I1(900),I2(900),I3(900),I4(900),
     * I5(900),I6(900),I7(900),I8(900),I9(900),
     * I10(900),I11(900),I12(900),I13(900),I14(900),
     * I15(900),I16(900),I17(900),I18(900),I19(900),
     * I20(900),I21(900),I22(900),I23(900),I24(900),
     * TOT(900)
      REAL U1(900),U2(900),U3(900),U4(900),
     * U5(900),U6(900),U7(900),U8(900),U9(900),
     * U10(900),U11(900),U12(900),U13(900),U14(900),
     * U15(900),U16(900),U17(900),U18(900),U19(900),
     * U20(900),U21(900),U22(900),U23(900),U24(900)
      CHARACTER*20 TOTAL,NO


C       OPEN(70,FILE='PROJINFO.TXT',STATUS='OLD')
C       READ(70,*)NL
C       READ(70,*)NS
C       READ(70,*)ELFILE
C       READ(70,*)DIFILE
C       READ(70,*)SFILE
C       READ(70,*)WSFILE
C       READ(70,*)SOILFILE
C       READ(70,*)LANDFILE
C       READ(70,*)CNFILE
C       READ(70,*)ISOFILE

       OPEN(70,FILE='PROJINFO.TXT',STATUS='OLD')
       READ(70,*)NL
       READ(70,*)NS
       READ(70,*)ELFILE
       READ(70,*)ELRDC
       READ(70,*)DIRFILE
       READ(70,*)DIRRDC
       READ(70,*)SFILE
       READ(70,*)SRDC
       READ(70,*)WSFILE
```

```
      READ(70,*)WSRDC
      READ(70,*)CNFILE
      READ(70,*)CNRDC
      READ(70,*)ISOFILE
      READ(70,*)ISORDC
      READ(70,*)MN_X
      READ(70,*)MX_X
      READ(70,*)MN_Y
      READ(70,*)MX_Y


      E=0




      PRINT 10
10    FORMAT('*** ENTER ANTICEEDENT MOISTURE CONDITION (1,2,OR 3)')
      READ (5,*)AMC
      PRINT 11
11    FORMAT('*** ENTER THE WATERSHED NUMBER ***')
      READ (5,*)Q
      SIZE=30.0
12    PRINT 13
13    FORMAT('*** ENTER TIME INTERVAL FOR ISOCHRONES IN MINUTES  ***')
      READ (5,*)INTERV
14    PRINT 15
15    FORMAT('*** ENTER TIME INCREMENT FOR PRECIPITAION IN MINUTES ***')
      READ (5,*)PINC
16    PRINT 17
17    FORMAT('*** ENTER STORM DURATION IN MINUTES ***')
      READ (5,*)PDUR
      PRINT 31
31    FORMAT(/,'***DELETE INCREMENTAL FILES? 1=YES, 2=NO***')
      READ(5,*)DEL
1000  FORMAT(A80)


*********************************************************
C     AMC=2
C     INTERV=5
C     PINC=15
C     PDUR=90
*********************************************************
      PN=PDUR/PINC
```

```
C     OPENING FILES UNDER UNIX OR DOS

      DO 35 E=1,PN
      OPEN(60,STATUS='UNKNOWN',ACCESS='DIRECT',FILE=ISOFILE,
     * RECL=NS*2,FORM='UNFORMATTED')
      OPEN(61,STATUS='OLD',ACCESS='DIRECT',FILE=CNFILE,
     * RECL=NS*2,FORM='UNFORMATTED')
      OPEN(62,STATUS='UNKNOWN',ACCESS='DIRECT',FILE=WSFILE,
     * RECL=NS*2,FORM='UNFORMATTED')
      IF(E.EQ.1)OPEN(1,STATUS='OLD',ACCESS='DIRECT',FILE="pre1.rst",
     * RECL=NS*4,FORM='UNFORMATTED')
      IF(E.EQ.2)OPEN(2,STATUS='OLD',ACCESS='DIRECT',FILE="pre2.rst",
     * RECL=NS*4,FORM='UNFORMATTED')
      IF(E.EQ.3)OPEN(3,STATUS='OLD',ACCESS='DIRECT',FILE="pre3.rst",
     * RECL=NS*4,FORM='UNFORMATTED')
      IF(E.EQ.4)OPEN(4,STATUS='OLD',ACCESS='DIRECT',FILE="pre4.rst",
     * RECL=NS*4,FORM='UNFORMATTED')
      IF(E.EQ.5)OPEN(5,STATUS='OLD',ACCESS='DIRECT',FILE="pre5.rst",
     * RECL=NS*4,FORM='UNFORMATTED')
      IF(E.EQ.6)OPEN(6,STATUS='OLD',ACCESS='DIRECT',FILE="pre6.rst",
     * RECL=NS*4,FORM='UNFORMATTED')
      IF(E.EQ.7)OPEN(7,STATUS='OLD',ACCESS='DIRECT',FILE="pre7.rst",
     * RECL=NS*4,FORM='UNFORMATTED')
      IF(E.EQ.8)OPEN(8,STATUS='OLD',ACCESS='DIRECT',FILE="pre8.rst",
     * RECL=NS*4,FORM='UNFORMATTED')
      IF(E.EQ.9)OPEN(9,STATUS='OLD',ACCESS='DIRECT',FILE="pre9.rst",
     * RECL=NS*4,FORM='UNFORMATTED')
      IF(E.EQ.10)OPEN(10,STATUS='OLD',ACCESS='DIRECT',FILE="pre10.rst",
     * RECL=NS*4,FORM='UNFORMATTED')
      IF(E.EQ.11)OPEN(11,STATUS='OLD',ACCESS='DIRECT',FILE="pre11.rst",
     * RECL=NS*4,FORM='UNFORMATTED')
      IF(E.EQ.12)OPEN(12,STATUS='OLD',ACCESS='DIRECT',FILE="pre12.rst",
     * RECL=NS*4,FORM='UNFORMATTED')
      IF(E.EQ.13)OPEN(13,STATUS='OLD',ACCESS='DIRECT',FILE="pre13.rst",
     * RECL=NS*4,FORM='UNFORMATTED')
      IF(E.EQ.14)OPEN(14,STATUS='OLD',ACCESS='DIRECT',FILE="pre14.rst",
     * RECL=NS*4,FORM='UNFORMATTED')
      IF(E.EQ.15)OPEN(15,STATUS='OLD',ACCESS='DIRECT',FILE="pre15.rst",
     * RECL=NS*4,FORM='UNFORMATTED')
      IF(E.EQ.16)OPEN(16,STATUS='OLD',ACCESS='DIRECT',FILE="pre16.rst",
     * RECL=NS*4,FORM='UNFORMATTED')
```

```
      IF(E.EQ.17)OPEN(17,STATUS='OLD',ACCESS='DIRECT',FILE="pre17.rst",
     * RECL=NS*4,FORM='UNFORMATTED')
      IF(E.EQ.18)OPEN(18,STATUS='OLD',ACCESS='DIRECT',FILE="pre18.rst",
     * RECL=NS*4,FORM='UNFORMATTED')
      IF(E.EQ.19)OPEN(19,STATUS='OLD',ACCESS='DIRECT',FILE="pre19.rst",
     * RECL=NS*4,FORM='UNFORMATTED')
      IF(E.EQ.20)OPEN(20,STATUS='OLD',ACCESS='DIRECT',FILE="pre20.rst",
     * RECL=NS*4,FORM='UNFORMATTED')




C
C        STARTING THE PROCESSING
C    INITIALIZING THE CONSTANTS
C
C      SUM_PE(R)=0
C
C   INITIALIZE TIMES--INTERV = INTERVAL,(TOTAL OF 250)
C
C17    MAXINT=INTERV*250
C      DO 22 T=INTERV,MAXINT,INTERV
C
      IF (E.EQ.1)OPEN(21,FILE='RUNOFF1.TXT',STATUS='UNKNOWN')
      IF (E.EQ.2)OPEN(21,FILE='RUNOFF2.TXT',STATUS='UNKNOWN')
      IF (E.EQ.3)OPEN(21,FILE='RUNOFF3.TXT',STATUS='UNKNOWN')
      IF (E.EQ.4)OPEN(21,FILE='RUNOFF4.TXT',STATUS='UNKNOWN')
      IF (E.EQ.5)OPEN(21,FILE='RUNOFF5.TXT',STATUS='UNKNOWN')
      IF (E.EQ.6)OPEN(21,FILE='RUNOFF6.TXT',STATUS='UNKNOWN')
      IF (E.EQ.7)OPEN(21,FILE='RUNOFF7.TXT',STATUS='UNKNOWN')
      IF (E.EQ.8)OPEN(21,FILE='RUNOFF8.TXT',STATUS='UNKNOWN')
      IF (E.EQ.9)OPEN(21,FILE='RUNOFF9.TXT',STATUS='UNKNOWN')
      IF (E.EQ.10)OPEN(21,FILE='RUNOFF10.TXT',STATUS='UNKNOWN')
      IF (E.EQ.11)OPEN(21,FILE='RUNOFF11.TXT',STATUS='UNKNOWN')
      IF (E.EQ.12)OPEN(21,FILE='RUNOFF12.TXT',STATUS='UNKNOWN')
      IF (E.EQ.13)OPEN(21,FILE='RUNOFF13.TXT',STATUS='UNKNOWN')
      IF (E.EQ.14)OPEN(21,FILE='RUNOFF14.TXT',STATUS='UNKNOWN')
      IF (E.EQ.15)OPEN(21,FILE='RUNOFF15.TXT',STATUS='UNKNOWN')
      IF (E.EQ.16)OPEN(21,FILE='RUNOFF16.TXT',STATUS='UNKNOWN')
      IF (E.EQ.17)OPEN(21,FILE='RUNOFF17.TXT',STATUS='UNKNOWN')
      IF (E.EQ.18)OPEN(21,FILE='RUNOFF18.TXT',STATUS='UNKNOWN')
      IF (E.EQ.19)OPEN(21,FILE='RUNOFF19.TXT',STATUS='UNKNOWN')
      IF (E.EQ.20)OPEN(21,FILE='RUNOFF20.TXT',STATUS='UNKNOWN')
```

```
      SUM_PE1(T)=0.0
      SUM_PE2(T)=0.0
      SUM_PE3(T)=0.0
      SUM_PE4(T)=0.0
      SUM_PE5(T)=0.0
      SUM_PE6(T)=0.0
      SUM_PE7(T)=0.0
      SUM_PE8(T)=0.0
      SUM_PE9(T)=0.0
      SUM_PE10(T)=0.0
      SUM_PE11(T)=0.0
      SUM_PE12(T)=0.0
      SUM_PE13(T)=0.0
      SUM_PE14(T)=0.0
      SUM_PE15(T)=0.0
      SUM_PE16(T)=0.0
      SUM_PE17(T)=0.0
      SUM_PE18(T)=0.0
      SUM_PE19(T)=0.0
      SUM_PE20(T)=0.0

      SUM_Q1(T)=0.0
      SUM_Q2(T)=0.0
      SUM_Q3(T)=0.0
      SUM_Q4(T)=0.0
      SUM_Q5(T)=0.0
      SUM_Q6(T)=0.0
      SUM_Q7(T)=0.0
      SUM_Q8(T)=0.0
      SUM_Q9(T)=0.0
      SUM_Q10(T)=0.0
      SUM_Q11(T)=0.0
      SUM_Q12(T)=0.0
      SUM_Q13(T)=0.0
      SUM_Q14(T)=0.0
      SUM_Q15(T)=0.0
      SUM_Q16(T)=0.0
      SUM_Q17(T)=0.0
      SUM_Q18(T)=0.0
      SUM_Q19(T)=0.0
      SUM_Q20(T)=0.0
C
C
C
```

```
C
       TMAX=INTERV

       I=1
       DO 30 I=1,NL
      READ (60,REC=I) (ISOBUF(J,1),J=1,NS)
      READ (61,REC=I) (CNBUF(J,1),J=1,NS)
      READ (62,REC=I) (WBUF(J,1),J=1,NS)
      IF (E.GE.1)READ (1,REC=I) (PRECBUF1(J,1),J=1,NS)
      IF (E.GE.2)READ (2,REC=I) (PRECBUF2(J,1),J=1,NS)
      IF (E.GE.3)READ (3,REC=I) (PRECBUF3(J,1),J=1,NS)
      IF (E.GE.4)READ (4,REC=I) (PRECBUF4(J,1),J=1,NS)
      IF (E.GE.5)READ (5,REC=I) (PRECBUF5(J,1),J=1,NS)
      IF (E.GE.6)READ (6,REC=I) (PRECBUF6(J,1),J=1,NS)
      IF (E.GE.7)READ (7,REC=I) (PRECBUF7(J,1),J=1,NS)
      IF (E.GE.8)READ (8,REC=I) (PRECBUF8(J,1),J=1,NS)
      IF (E.GE.9)READ (9,REC=I) (PRECBUF9(J,1),J=1,NS)
      IF (E.GE.10)READ (10,REC=I) (PRECBUF10(J,1),J=1,NS)
      IF (E.GE.11)READ (11,REC=I) (PRECBUF11(J,1),J=1,NS)
      IF (E.GE.12)READ (12,REC=I) (PRECBUF12(J,1),J=1,NS)
      IF (E.GE.13)READ (13,REC=I) (PRECBUF13(J,1),J=1,NS)
      IF (E.GE.14)READ (14,REC=I) (PRECBUF14(J,1),J=1,NS)
      IF (E.GE.15)READ (15,REC=I) (PRECBUF15(J,1),J=1,NS)
      IF (E.GE.16)READ (16,REC=I) (PRECBUF16(J,1),J=1,NS)
      IF (E.GE.17)READ (17,REC=I) (PRECBUF17(J,1),J=1,NS)
      IF (E.GE.18)READ (18,REC=I) (PRECBUF18(J,1),J=1,NS)
      IF (E.GE.19)READ (19,REC=I) (PRECBUF19(J,1),J=1,NS)
      IF (E.GE.20)READ (20,REC=I) (PRECBUF20(J,1),J=1,NS)

      DO 25 J=1,NS
      IF(WBUF(J,1).NE.Q)GOTO 25
      IF(ISOBUF(J,1).LT.0)GOTO 25
      IF(ISOBUF(J,1).GE.0)THEN
      ISO=ISOBUF(J,1)
      IF(ISO.GT.MAX)THEN
      MAX=ISO
      MAXINT=MAX/INTERV
      ENDIF
      T=ISO/INTERV
      IF(T.GT.TMAX)THEN
       TMAX=T
       END IF

C       OPEN(69,FILE='TEST.TXT',STATUS='NEW')

       IF(E.GE.1)THEN
```

```
      P1=PRECBUF1(J,1)
      PCUM1=P1
        CN=CNBUF(J,1)
      CNN=CN*1.0
      IF(AMC.EQ.1)CNN=(4.2*CNN)/(10-0.058*CNN)
      IF(AMC.EQ.3)CNN=(23*CNN)/(10-0.13*CNN)
        RET=(1000/CNN)-10
        CALL EXCESS(PE1,P1,RET)
        SUM_PE1(T)=SUM_PE1(T)+PE1
      SUM_Q1(T)=SUM_PE1(T)*(SIZE**2)*(3.28**2)/(12.0*60.0*INTERV)
C       WRITE(69,200)E,T,ISO,CNN,RET,P1,PE1
      END IF



     IF(E.GE.2)THEN
      P2=PRECBUF2(J,1)
      PCUM2=P1+P2
      CALL EXCESS(PECUM2,PCUM2,RET)
      PE2=PECUM2-PE1
      IF(PE2.LT.0)PE2=0
      SUM_PE2(T)=SUM_PE2(T)+PE2
      SUM_Q2(T)=SUM_PE2(T)*(SIZE**2)*(3.28**2)/(12.0*60.0*INTERV)

C      WRITE(69,200)E,T,ISO,CNN,RET,PCUM2,PE2
      END IF



     IF(E.GE.3)THEN
      P3=PRECBUF3(J,1)
      PCUM3=PCUM2+P3
      CALL EXCESS(PECUM3,PCUM3,RET)
      PE3=PECUM3-PECUM2
      IF(PE3.LT.0)PE3=0
      SUM_PE3(T)=SUM_PE3(T)+PE3
      SUM_Q3(T)=SUM_PE3(T)*(SIZE**2)*(3.28**2)/(12.0*60.0*INTERV)

C       WRITE(69,200)E,T,ISO,CNN,RET,PCUM3,PE3
      END IF



C200   FORMAT(I1,1X,I2,1X,I2,1X,F10.1,1X,F10.5,1X,F10.5,1X,F10.5)

     IF(E.GE.4)THEN
```

```
          P4=PRECBUF4(J,1)
          PCUM4=PCUM3+P4
          CALL EXCESS(PECUM4,PCUM4,RET)
          PE4=PECUM4-PECUM3
          IF(PE4.LT.0)PE4=0
          SUM_PE4(T)=SUM_PE4(T)+PE4
          SUM_Q4(T)=SUM_PE4(T)*(SIZE**2)*(3.28**2)/(12.0*60.0*INTERV)
C         TEST
C          WRITE(69,200)E,T,ISO,CNN,RET,PCUM4,PE4
          END IF



        IF(E.GE.5)THEN
          P5=PRECBUF5(J,1)
          PCUM5=PCUM4+P5
          CALL EXCESS(PECUM5,PCUM5,RET)
          PE5=PECUM5-PECUM4
          IF(PE5.LT.0)PE5=0
          SUM_PE5(T)=SUM_PE5(T)+PE5
          SUM_Q5(T)=SUM_PE5(T)*(SIZE**2)*(3.28**2)/(12.0*60.0*INTERV)
          END IF

        IF(E.GE.6)THEN
          P6=PRECBUF6(J,1)
          PCUM6=PCUM5+P6
          CALL EXCESS(PECUM6,PCUM6,RET)
          PE6=PECUM6-PECUM5
          IF(PE6.LT.0)PE6=0
          SUM_PE6(T)=SUM_PE6(T)+PE6
          SUM_Q6(T)=SUM_PE6(T)*(SIZE**2)*(3.28**2)/(12.0*60.0*INTERV)
          END IF

        IF(E.GE.7)THEN
          P7=PRECBUF7(J,1)
          PCUM7=PCUM6+P7
          CALL EXCESS(PECUM7,PCUM7,RET)
          PE7=PECUM7-PECUM6
          IF(PE7.LT.0)PE7=0
          SUM_PE7(T)=SUM_PE7(T)+PE7
          SUM_Q7(T)=SUM_PE7(T)*(SIZE**2)*(3.28**2)/(12.0*60.0*INTERV)
          END IF

        IF(E.GE.8)THEN
          P8=PRECBUF8(J,1)
          PCUM8=PCUM7+P8
```

```
    CALL EXCESS(PECUM8,PCUM8,RET)
    PE8=PECUM8-PECUM7
    IF(PE8.LT.0)PE8=0
    SUM_PE8(T)=SUM_PE8(T)+PE8
    SUM_Q8(T)=SUM_PE8(T)*(SIZE**2)*(3.28**2)/(12.0*60.0*INTERV)
    END IF

   IF(E.GE.9)THEN
    P9=PRECBUF9(J,1)
    PCUM9=PCUM8+P9
    CALL EXCESS(PECUM9,PCUM9,RET)
    PE9=PECUM9-PECUM8
    IF(PE9.LT.0)PE9=0
    SUM_PE9(T)=SUM_PE9(T)+PE9
    SUM_Q9(T)=SUM_PE9(T)*(SIZE**2)*(3.28**2)/(12.0*60.0*INTERV)
    END IF

   IF(E.GE.10)THEN
    P10=PRECBUF10(J,1)
    PCUM10=PCUM9+P10
    CALL EXCESS(PECUM10,PCUM10,RET)
    PE10=PECUM10-PECUM9
    IF(PE10.LT.0)PE10=0
    SUM_PE10(T)=SUM_PE10(T)+PE10
    SUM_Q10(T)=SUM_PE10(T)*(SIZE**2)*(3.28**2)/(12.0*60.0*INTERV)
    END IF

   IF(E.GE.11)THEN
    P11=PRECBUF11(J,1)
    PCUM11=PCUM10+P11
    CALL EXCESS(PECUM11,PCUM11,RET)
    PE11=PECUM11-PECUM10
    IF(PE11.LT.0)PE11=0
    SUM_PE11(T)=SUM_PE11(T)+PE11
    SUM_Q11(T)=SUM_PE11(T)*(SIZE**2)*(3.28**2)/(12.0*60.0*INTERV)
    END IF

   IF(E.GE.12)THEN
    P12=PRECBUF12(J,1)
    PCUM12=PCUM11+P12
    CALL EXCESS(PECUM12,PCUM12,RET)
    PE12=PECUM12-PECUM11
    IF(PE12.LT.0)PE12=0
    SUM_PE12(T)=SUM_PE12(T)+PE12
    SUM_Q12(T)=SUM_PE12(T)*(SIZE**2)*(3.28**2)/(12.0*60.0*INTERV)
    END IF
```

```
IF(E.GE.13)THEN
 P13=PRECBUF13(J,1)
 PCUM13=PCUM12+P13
 CALL EXCESS(PECUM13,PCUM13,RET)
 PE13=PECUM13-PECUM12
 IF(PE13.LT.0)PE13=0
 SUM_PE13(T)=SUM_PE13(T)+PE13
 SUM_Q13(T)=SUM_PE13(T)*(SIZE**2)*(3.28**2)/(12.0*60.0*INTERV)
 END IF

IF(E.GE.14)THEN
 P14=PRECBUF14(J,1)
 PCUM14=PCUM13+P14
 CALL EXCESS(PECUM14,PCUM14,RET)
 PE14=PECUM14-PECUM13
 IF(PE14.LT.0)PE14=0
 SUM_PE14(T)=SUM_PE14(T)+PE14
 SUM_Q14(T)=SUM_PE14(T)*(SIZE**2)*(3.28**2)/(12.0*60.0*INTERV)
 END IF

IF(E.GE.15)THEN
 P15=PRECBUF15(J,1)
 PCUM15=PCUM14+P15
 CALL EXCESS(PECUM15,PCUM15,RET)
 PE15=PECUM15-PECUM14
 IF(PE15.LT.0)PE15=0
 SUM_PE15(T)=SUM_PE15(T)+PE15
 SUM_Q15(T)=SUM_PE15(T)*(SIZE**2)*(3.28**2)/(12.0*60.0*INTERV)
 END IF

IF(E.GE.16)THEN
 P16=PRECBUF16(J,1)
 PCUM16=PCUM15+P16
 CALL EXCESS(PECUM16,PCUM16,RET)
 PE16=PECUM16-PECUM15
 IF(PE16.LT.0)PE16=0
 SUM_PE16(T)=SUM_PE16(T)+PE16
 SUM_Q16(T)=SUM_PE16(T)*(SIZE**2)*(3.28**2)/(12.0*60.0*INTERV)
 END IF

IF(E.GE.17)THEN
 P17=PRECBUF17(J,1)
 PCUM17=PCUM16+P17
 CALL EXCESS(PECUM17,PCUM17,RET)
 PE17=PECUM17-PECUM16
```

```
      IF(PE17.LT.0)PE17=0
      SUM_PE17(T)=SUM_PE17(T)+PE17
      SUM_Q17(T)=SUM_PE17(T)*(SIZE**2)*(3.28**2)/(12.0*60.0*INTERV)
      END IF

     IF(E.GE.18)THEN
      P18=PRECBUF18(J,1)
      PCUM18=PCUM17+P18
      CALL EXCESS(PECUM18,PCUM18,RET)
      PE18=PECUM18-PECUM17
      IF(PE18.LT.0)PE18=0
      SUM_PE18(T)=SUM_PE18(T)+PE18
      SUM_Q18(T)=SUM_PE18(T)*(SIZE**2)*(3.28**2)/(12.0*60.0*INTERV)
      END IF

     IF(E.GE.19)THEN
      P19=PRECBUF19(J,1)
      PCUM19=PCUM18+P19
      CALL EXCESS(PECUM19,PCUM19,RET)
      PE19=PECUM19-PECUM18
      IF(PE19.LT.0)PE19=0
      SUM_PE19(T)=SUM_PE19(T)+PE19
      SUM_Q19(T)=SUM_PE19(T)*(SIZE**2)*(3.28**2)/(12.0*60.0*INTERV)
      END IF

     IF(E.GE.20)THEN
      P20=PRECBUF20(J,1)
      PCUM20=PCUM19+P20
      CALL EXCESS(PECUM20,PCUM20,RET)
      PE20=PECUM20-PECUM19
      IF(PE20.LT.0)PE20=0
      SUM_PE20(T)=SUM_PE20(T)+PE20
      SUM_Q20(T)=SUM_PE20(T)*(SIZE**2)*(3.28**2)/(12.0*60.0*INTERV)
      END IF



     END IF
25    CONTINUE
30    CONTINUE
     IF(E.EQ.1)THEN
     WRITE(21,213)MAXINT
     WRITE(21,213)INTERV
     DO 50 T=1,MAXINT
     WRITE(21,212)(SUM_Q1(T))
50    CONTINUE
```

```
      GO TO 35
      END IF


      IF(E.EQ.2)THEN
      WRITE(21,213)MAXINT
      WRITE(21,213)INTERV
      DO 51 T=1,MAXINT
      WRITE(21,212)(SUM_Q2(T))
51    CONTINUE
      GO TO 35
      END IF

      IF(E.EQ.3)THEN
      IF(PN.EQ.2)GO TO 35
      WRITE(21,213)MAXINT
      WRITE(21,213)INTERV
      DO 52 T=1,MAXINT
      WRITE(21,212)(SUM_Q3(T))
52    CONTINUE
      GO TO 35
      END IF

      IF(E.EQ.4)THEN
      WRITE(21,213)MAXINT
      WRITE(21,213)INTERV
      DO 53 T=1,MAXINT
      WRITE(21,212)(SUM_Q4(T))
53    CONTINUE
      GO TO 35
      END IF

      IF(E.EQ.5)THEN
      WRITE(21,213)MAXINT
      WRITE(21,213)INTERV
      DO 54 T=1,MAXINT
      WRITE(21,212)(SUM_Q5(T))
54    CONTINUE
      GO TO 35
      END IF

      IF(E.EQ.6)THEN
      WRITE(21,213)MAXINT
      WRITE(21,213)INTERV
      DO 55 T=1,MAXINT
      WRITE(21,212)(SUM_Q6(T))
```

```
55    CONTINUE
      GO TO 35
      END IF

      IF(E.EQ.7)THEN
      WRITE(21,213)MAXINT
      WRITE(21,213)INTERV
      DO 56 T=1,MAXINT
      WRITE(21,212)(SUM_Q7(T))
56    CONTINUE
      GO TO 35
      END IF

      IF(E.EQ.8)THEN
      WRITE(21,213)MAXINT
      WRITE(21,213)INTERV
      DO 57 T=1,MAXINT
      WRITE(21,212)(SUM_Q8(T))
57    CONTINUE
      GO TO 35
      END IF

      IF(E.EQ.9)THEN
      WRITE(21,213)MAXINT
      WRITE(21,213)INTERV
      DO 58 T=1,MAXINT
      WRITE(21,212)(SUM_Q9(T))
58    CONTINUE
      GO TO 35
      END IF

      IF(E.EQ.10)THEN
      WRITE(21,213)MAXINT
      WRITE(21,213)INTERV
      DO 59 T=1,MAXINT
      WRITE(21,212)(SUM_Q10(T))
59    CONTINUE
      GO TO 35
      END IF

      IF(E.EQ.11)THEN
      WRITE(21,213)MAXINT
      WRITE(21,213)INTERV
      DO 60 T=1,MAXINT
      WRITE(21,212)(SUM_Q11(T))
60    CONTINUE
```

```
      GO TO 35
      END IF

      IF(E.EQ.12)THEN
      WRITE(21,213)MAXINT
      WRITE(21,213)INTERV
      DO 61 T=1,MAXINT
      WRITE(21,212)(SUM_Q12(T))
61    CONTINUE
      GO TO 35
      END IF

      IF(E.EQ.13)THEN
      WRITE(21,213)MAXINT
      WRITE(21,213)INTERV
      DO 62 T=1,MAXINT
      WRITE(21,212)(SUM_Q13(T))
62    CONTINUE
      GO TO 35
      END IF

      IF(E.EQ.14)THEN
      WRITE(21,213)MAXINT
      WRITE(21,213)INTERV
      DO 63 T=1,MAXINT
      WRITE(21,212)(SUM_Q14(T))
63    CONTINUE
      GO TO 35
      END IF

      IF(E.EQ.15)THEN
      WRITE(21,213)MAXINT
      WRITE(21,213)INTERV
      DO 64 T=1,MAXINT
      WRITE(21,212)(SUM_Q15(T))
64    CONTINUE
      GO TO 35
      END IF

      IF(E.EQ.16)THEN
      WRITE(21,213)MAXINT
      WRITE(21,213)INTERV
      DO 65 T=1,MAXINT
      WRITE(21,212)(SUM_Q16(T))
65    CONTINUE
      GO TO 35
```

```
      END IF

      IF(E.EQ.17)THEN
      WRITE(21,213)MAXINT
      WRITE(21,213)INTERV
      DO 66 T=1,MAXINT
      WRITE(21,212)(SUM_Q17(T))
66    CONTINUE
      GO TO 35
      END IF

      IF(E.EQ.18)THEN
      WRITE(21,213)MAXINT
      WRITE(21,213)INTERV
      DO 67 T=1,MAXINT
      WRITE(21,212)(SUM_Q18(T))
67    CONTINUE
      GO TO 35
      END IF

      IF(E.EQ.19)THEN
      WRITE(21,213)MAXINT
      WRITE(21,213)INTERV
      DO 68 T=1,MAXINT
      WRITE(21,212)(SUM_Q19(T))
68    CONTINUE
      GO TO 35
      END IF

      IF(E.EQ.20)THEN
      WRITE(21,213)MAXINT
      WRITE(21,213)INTERV
      DO 69 T=1,MAXINT
      WRITE(21,212)(SUM_Q20(T))
69    CONTINUE
      GO TO 35
      END IF

C     OPEN(69,FILE='TEST.TXT',STATUS='NEW')
C     WRITE(69,212)

212   FORMAT(F15.5)
213   FORMAT(I5)
35    CONTINUE
```

```
******************************************************************************
      CLOSE (1,STATUS='KEEP')
      CLOSE (2,STATUS='KEEP')
      CLOSE (3,STATUS='KEEP')
      CLOSE (4,STATUS='KEEP')
      CLOSE (5,STATUS='KEEP')
      CLOSE (6,STATUS='KEEP')
      CLOSE (7,STATUS='KEEP')
      CLOSE (8,STATUS='KEEP')
      CLOSE (9,STATUS='KEEP')
      CLOSE (10,STATUS='KEEP')
      CLOSE (11,STATUS='KEEP')
      CLOSE (12,STATUS='KEEP')
      CLOSE (13,STATUS='KEEP')
      CLOSE (14,STATUS='KEEP')
      CLOSE (15,STATUS='KEEP')
      CLOSE (16,STATUS='KEEP')
      CLOSE (17,STATUS='KEEP')
      CLOSE (18,STATUS='KEEP')
      CLOSE (19,STATUS='KEEP')
      CLOSE (20,STATUS='KEEP')
      CLOSE (21,STATUS='KEEP')
      CLOSE (60,STATUS='KEEP')
      CLOSE (61,STATUS='KEEP')
      CLOSE (62,STATUS='KEEP')
      N=0
       O=Q
       IF(O.EQ.2)TOTAL="TOTAL_2.TXT"
       IF(O.EQ.3)TOTAL='TOTAL_3.TXT'
       IF(O.EQ.4)TOTAL='TOTAL_4.TXT'
       IF(O.EQ.5)TOTAL='TOTAL_5.TXT'
       IF(O.EQ.6)TOTAL='TOTAL_6.TXT'
       IF(O.EQ.7)TOTAL='TOTAL_7.TXT'
       IF(O.EQ.8)TOTAL='TOTAL_8.TXT'
       IF(O.EQ.9)TOTAL='TOTAL_9.TXT'
       IF(O.EQ.10)TOTAL='TOTAL_10.TXT'
       IF(O.EQ.11)TOTAL='TOTAL_11.TXT'
       IF(O.EQ.12)TOTAL='TOTAL_12.TXT'
       IF(O.EQ.13)TOTAL='TOTAL_13.TXT'
       IF(O.EQ.14)TOTAL='TOTAL_14.TXT'
       IF(O.EQ.15)TOTAL='TOTAL_15.TXT'
       IF(O.EQ.16)TOTAL='TOTAL_16.TXT'
```

```
      OPEN (30,STATUS='OLD',FILE='RUNOFF1.TXT')
      READ(30,*)MAXINT
      READ(30,*)INTERV
      MAXT=MAXINT*INTERV
C      zero variables
      DO 70 Z=INTERV,(MAXT+(PN*PINC)),INTERV
      I1(Z)=0
      I2(Z)=0
      I3(Z)=0
      I4(Z)=0
      I5(Z)=0
      I6(Z)=0
      I7(Z)=0
      I8(Z)=0
      I9(Z)=0
      I10(Z)=0
      I11(Z)=0
      I12(Z)=0
      I13(Z)=0
      I14(Z)=0
      I15(Z)=0
      I16(Z)=0
      I17(Z)=0
      I18(Z)=0
      I19(Z)=0
      I20(Z)=0

70      CONTINUE
      CLOSE(30)

      IF(PN.GE.1)OPEN (1,STATUS='OLD',FILE='RUNOFF1.TXT')
      IF(PN.GE.2)OPEN (2,STATUS='OLD',FILE='RUNOFF2.TXT')
      IF(PN.GE.3)OPEN (3,STATUS='OLD',FILE='RUNOFF3.TXT')
      IF(PN.GE.4)OPEN (4,STATUS='OLD',FILE='RUNOFF4.TXT')
      IF(PN.GE.5)OPEN (5,STATUS='OLD',FILE='RUNOFF5.TXT')
      IF(PN.GE.6)OPEN (6,STATUS='OLD',FILE='RUNOFF6.TXT')
      IF(PN.GE.7)OPEN (7,STATUS='OLD',FILE='RUNOFF7.TXT')
      IF(PN.GE.8)OPEN (8,STATUS='OLD',FILE='RUNOFF8.TXT')
      IF(PN.GE.9)OPEN (9,STATUS='OLD',FILE='RUNOFF9.TXT')
      IF(PN.GE.10)OPEN (10,STATUS='OLD',FILE='RUNOFF10.TXT')
      IF(PN.GE.11)OPEN (11,STATUS='OLD',FILE='RUNOFF11.TXT')
      IF(PN.GE.12)OPEN (12,STATUS='OLD',FILE='RUNOFF12.TXT')
      IF(PN.GE.13)OPEN (13,STATUS='OLD',FILE='RUNOFF13.TXT')
      IF(PN.GE.14)OPEN (14,STATUS='OLD',FILE='RUNOFF14.TXT')
      IF(PN.GE.15)OPEN (15,STATUS='OLD',FILE='RUNOFF15.TXT')
      IF(PN.GE.16)OPEN (16,STATUS='OLD',FILE='RUNOFF16.TXT')
```

```
      IF(PN.GE.17)OPEN (17,STATUS='OLD',FILE='RUNOFF17.TXT')
      IF(PN.GE.18)OPEN (18,STATUS='OLD',FILE='RUNOFF18.TXT')
      IF(PN.GE.19)OPEN (19,STATUS='OLD',FILE='RUNOFF19.TXT')
      IF(PN.GE.20)OPEN (20,STATUS='OLD',FILE='RUNOFF20.TXT')

      OPEN (21,STATUS='UNKNOWN',FILE=TOTAL)

C       INTERV=5
C       MAXINT=5


    U15(Z)=0
C      READ(1,*)MAXINT
C      READ(1,*)INTERV
C      DO 10 Z=INTERV,MAXT,INTERV
C      READ (1,*) I1(Z)
C10     CONTINUE

      DO 72 N=0,PN-1
      READ((N+1),*)MAXINT
      READ((N+1),*)INTERV
      DO 71 Z=(INTERV+(N*PINC)),(MAXT+(N*PINC)),INTERV
      IF(N.EQ.0)READ((N+1),*)I1(Z)
      IF(N.EQ.1)READ((N+1),*)I2(Z)
      IF(N.EQ.2)READ((N+1),*)I3(Z)
      IF(N.EQ.3)READ((N+1),*)I4(Z)
      IF(N.EQ.4)READ((N+1),*)I5(Z)
      IF(N.EQ.5)READ((N+1),*)I6(Z)
      IF(N.EQ.6)READ((N+1),*)I7(Z)
      IF(N.EQ.7)READ((N+1),*)I8(Z)
      IF(N.EQ.8)READ((N+1),*)I9(Z)
      IF(N.EQ.9)READ((N+1),*)I10(Z)
      IF(N.EQ.10)READ((N+1),*)I11(Z)
      IF(N.EQ.11)READ((N+1),*)I12(Z)
      IF(N.EQ.12)READ((N+1),*)I13(Z)
      IF(N.EQ.13)READ((N+1),*)I14(Z)
      IF(N.EQ.14)READ((N+1),*)I15(Z)
      IF(N.EQ.15)READ((N+1),*)I16(Z)
      IF(N.EQ.16)READ((N+1),*)I17(Z)
      IF(N.EQ.17)READ((N+1),*)I18(Z)
      IF(N.EQ.18)READ((N+1),*)I19(Z)
      IF(N.EQ.19)READ((N+1),*)I20(Z)
71    CONTINUE

72    CONTINUE
    Z=0
```

```
      DO 73 Z=PINC,(MAXT+(PN*PINC)),PINC

      U1(Z+PINC)=(I1(Z+PINC)+I1(Z))/2
      U2(Z+PINC)=(I2(Z+PINC)+I2(Z))/2
      U3(Z+PINC)=(I3(Z+PINC)+I3(Z))/2
      U4(Z+PINC)=(I4(Z+PINC)+I4(Z))/2
      U5(Z+PINC)=(I5(Z+PINC)+I5(Z))/2
      U6(Z+PINC)=(I6(Z+PINC)+I6(Z))/2
      U7(Z+PINC)=(I7(Z+PINC)+I7(Z))/2
      U8(Z+PINC)=(I8(Z+PINC)+I8(Z))/2
      U9(Z+PINC)=(I9(Z+PINC)+I9(Z))/2
      U10(Z+PINC)=(I10(Z+PINC)+I10(Z))/2
      U11(Z+PINC)=(I11(Z+PINC)+I11(Z))/2
      U12(Z+PINC)=(I12(Z+PINC)+I12(Z))/2
      U13(Z+PINC)=(I13(Z+PINC)+I13(Z))/2
      U14(Z+PINC)=(I14(Z+PINC)+I14(Z))/2
      U15(Z+PINC)=(I15(Z+PINC)+I15(Z))/2
      U16(Z+PINC)=(I16(Z+PINC)+I16(Z))/2
      U17(Z+PINC)=(I17(Z+PINC)+I17(Z))/2
      U18(Z+PINC)=(I18(Z+PINC)+I18(Z))/2
      U19(Z+PINC)=(I19(Z+PINC)+I19(Z))/2
      U20(Z+PINC)=(I20(Z+PINC)+I20(Z))/2
73    CONTINUE
C      ADD TIME SERIES
      DO 74 Z=INTERV,(MAXT+(PN*PINC)),INTERV


      TOT(Z)=U1(Z)+U2(Z)+U3(Z)+U4(Z)+U5(Z)+U6(Z)+U7(Z)+
     *  U8(Z)+U9(Z)+U10(Z)+U11(Z)+U12(Z)+U13(Z)+U14(Z)+U15(Z)+U16(Z)+
     *  U17(Z)+U18(Z)+U19(Z)+U20(Z)

74     CONTINUE
      NMAXINT=(MAXT+(PN-1)*PINC)/PINC
C       DO 30 Z=INTERV,(MAXT+(PN*PINC)),INTERV
      WRITE (21,*) NMAXINT
      WRITE (21,*) PINC
      DO 75 Z=PINC,(MAXT+(PN-1)*PINC),PINC
      WRITE (21,*) TOT(Z)
75     CONTINUE




      IF(DEL.EQ.2)GOTO 215
      CLOSE (1,STATUS='DELETE')
      CLOSE (2,STATUS='DELETE')
```

```
      CLOSE (3,STATUS='DELETE')
      CLOSE (4,STATUS='DELETE')
      CLOSE (5,STATUS='DELETE')
      CLOSE (6,STATUS='DELETE')
      CLOSE (7,STATUS='DELETE')
      CLOSE (8,STATUS='DELETE')
      CLOSE (9,STATUS='DELETE')
      CLOSE (10,STATUS='DELETE')
      CLOSE (11,STATUS='DELETE')
      CLOSE (12,STATUS='DELETE')
      CLOSE (13,STATUS='DELETE')
      CLOSE (14,STATUS='DELETE')
      CLOSE (15,STATUS='DELETE')
      CLOSE (16,STATUS='DELETE')
      CLOSE (17,STATUS='DELETE')
      CLOSE (18,STATUS='DELETE')
      CLOSE (19,STATUS='DELETE')
      CLOSE (20,STATUS='DELETE')


215   STOP
      END
C
C     CALCULATING THE RAINFALL EXCESS
C
      SUBROUTINE EXCESS(A,C,B)
        REAL A,B,C
        IF(C.LT.0.2*B)THEN
        A=0.0
        ELSEIF(C.GE.0.2*B)THEN
        A=(C-0.2*B)**2/(C+0.8*B)
       ENDIF
       RETURN
        END
```

Reach Routing Program

```
C*************************************************************************
C       THIS PROGRAM WAS ORIGINALLY WRITTEN BY KHAN
C       IT WAS MODIFIED BY JAMES EMERICK
C
C*************************************************************************
C
C     THIS SECTION IS FOR INPUT DATA
C
      INTEGER  Z,J,SIZE
      REAL X,T,C1,K,C0,C2,SUM,Q(1000),I(1000)
      CHARACTER*20 TOTAL,HYDRO
C     PRINT*, '*** THE STORAGE CONSTANT K IS CALCULATED AS ***'
C     PRINT*, '                                    '
C     PRINT*, 'K=DELTA_X/C   WHERE'
C     PRINT*, '             DELTA_X=REACH LENGTH'
C     PRINT*, '             C=FLOOD WAVE CELERITY GIVEN BY C=M*V'
C     PRINT*, '             M=CONSTANT WITH VALUE 5/3'
C     PRINT*, '             V=AVERAGE VELOCITY AT BANKFUL DISCHARGE'
C     PRINT*, '                                         '
C     PRINT*, '*** IF THE ABOVE MENTIONED METHOD IS NOT EMPLOYED THEN'
      PRINT*, '*** ENTER THE WATERSHED NUMBER***'
      READ*, O
      PRINT*, '    K CAN BE APPROXIMATED AS EQUAL TO THE TRAVEL TIME'
      PRINT*, '    IN THE REACH OR THE TIME OF CONCENTRATION ***'
      PRINT*, '                                    '
      PRINT*, '*** ENTER STORAGE CONSTANT---K IN MINUTES***'
      READ (5,*) K
C     PRINT*, '*** THE WEDGE PARAMETER X IS ESIMATED AS ***'
C     PRINT*, 'X=0.5(1-(Q0/(S0*C*DELTA_X))) WHERE'
C     PRINT*, '                    Q0=DISCHARGE PER UNIT '
C     PRINT*, '                    WIDTH OF THE CHANNEL AT'
C     PRINT*, '                    PEAK FLOW RATE'
C     PRINT*, '                    S0=SLOPE OF CHANNEL'
C     PRINT*, '                    C=FLOOD WAVE CELERITY'
C     PRINT*, '                    DELTA_X=REACH LENGTH'
C     PRINT*, '                                        '
C     PRINT*, '*** IF THE ABOVE MENTIONED METHOD IS NOT EMPLOYED THEN'
C     PRINT*, '    PUT X = 0.25 ***'
C     PRINT*, '                                    '
      PRINT*, '*** ENTER WEDGE PARAMETER----X ***'
      READ (5,*) X
C     PRINT*, '*** ENTER THE INITIAL OUTFLOW (CFS) ***'
C      READ*, Q(0)
```

```
C          PRINT*, '*** ENTER THE GRID SIZE ***'
C          READ*, SIZE

      IF(O.EQ.2)THEN
      TOTAL='HYDRO_2.TXT'
      HYDRO='RHYDRO_2.TXT'
      ENDIF
      IF(O.EQ.3)THEN
      TOTAL='HYDRO_3.TXT'
      HYDRO='RHYDRO_3.TXT'
      ENDIF
      IF(O.EQ.4)THEN
      TOTAL='HYDRO_4.TXT'
      HYDRO='RHYDRO_4.TXT'
      ENDIF
      IF(O.EQ.5)THEN
      TOTAL='HYDRO_5.TXT'
      HYDRO='RHYDRO_5.TXT'
      ENDIF
      IF(O.EQ.6)THEN
      TOTAL='HYDRO_6.TXT'
      HYDRO='RHYDRO_6.TXT'
      ENDIF
      IF(O.EQ.7)THEN
      TOTAL='HYDRO_7.TXT'
      HYDRO='RHYDRO_7.TXT'
      ENDIF
      IF(O.EQ.8)THEN
      TOTAL='HYDRO_8.TXT'
      HYDRO='RHYDRO_8.TXT'
      ENDIF
      IF(O.EQ.9)THEN
      TOTAL='HYDRO_9.TXT'
      HYDRO='RHYDRO_9.TXT'
      ENDIF
      IF(O.EQ.10)THEN
      TOTAL='HYDRO_10.TXT'
      HYDRO='RHYDRO_10.TXT'
      ENDIF
      IF(O.EQ.11)THEN
      TOTAL='HYDRO_11.TXT'
      HYDRO='RHYDRO_11.TXT'
      ENDIF
      IF(O.EQ.12)THEN
      TOTAL='HYDRO_12.TXT'
      HYDRO='RHYDRO_12.TXT'
```

```
      ENDIF
      IF(O.EQ.13)THEN
      TOTAL='HYDRO_13.TXT'
      HYDRO='RHYDRO_13.TXT'
      ENDIF
      IF(O.EQ.14)THEN
      TOTAL='HYDRO_14.TXT'
      HYDRO='RHYDRO_14.TXT'
      ENDIF
      IF(O.EQ.15)THEN
      TOTAL='HYDRO_15.TXT'
      HYDRO='RHYDRO_15.TXT'
      ENDIF
      L=0
4      OPEN (10,STATUS='OLD',FILE=TOTAL)
       SUM=0.0
       DO 5 Z=0,500
       I(Z)=0
       Q(Z)=0
5      CONTINUE
       READ (10,*) J
       READ (10,*) T
       INTERV=T
       DO 10 Z=1,J
       READ (10,*) I(Z)

       SUM=SUM+I(Z)
10     CONTINUE
       IF (SUM.EQ.0.0) THEN
       PRINT*, 'NO DATA VALUES IN THE TOTALRUNOFF.TXT FILE'
       END IF
C        QT(0)=Q(0)/2.0
C
C         THIS SECTION IS TO CALCULATE THE COEFFICIENTS
C
       C0=(0.5*T-K*X)/(K*(1.0-X)+0.5*T)
       C1=(0.5*T+K*X)/(K*(1.0-X)+0.5*T)
       C2=(K*(1.0-X)-0.5*T)/(K*(1.0-X)+0.5*T)
C
C         THIS SECTION IS TO ROUTE AND PRINT OUT
C       THE RESULTS
C
       DO 20 Z=1,J
       Q(Z)=C0*I(Z)+C1*I(Z-1)+C2*Q(Z-1)
C        QM(Z)=C0*I(Z+1)+C1*Q(Z)
C        QT(Z+1)=(QM(Z)+Q(Z))/2.0
```

```
C       QT(1)=(Q(1)+Q(0))/2.0

C          PRINT *,I(Z),Q(Z)
20     CONTINUE
       IF(L.EQ.0)OPEN (20,FILE=HYDRO,STATUS='UNKNOWN')
C       WRITE (20,*)
C       WRITE (20,*)
C       WRITE (20,*) '       Time', '        HYDRO'
C       WRITE (20,*) '       (min)', '        (cfs)'
C       WRITE (20,*)
       TI=0.0
       DO 30 Z=1,J
       WRITE (20,1) Q(Z)
1       FORMAT (2X,F10.2)
       TI=TI+T
30     CONTINUE
       Z=J
       TI=TI-T
40     TI=TI+T
       I(Z+1)=0.0
       I(Z+2)=0.0
       Q(Z+1)=C0*I(Z+1)+C1*I(Z)+C2*Q(Z)
       IF ((Q(Z+1).LT.0.01).OR.(TI.GE.5000)) THEN
       GOTO 50
       ENDIF
       WRITE (20,2) Q(Z+1)
2       FORMAT (2X,F10.2)
       Z=Z+1
       GOTO 40
50     IF(L.EQ.1)GOTO 51
       L=1
       REWIND 20
       WRITE (20,3) Z
       WRITE (20,3) INTERV
3       FORMAT (2X,I10)
       GOTO 4
51     STOP
       END
```

Hydrograph Adding Program

```
C     ************************************************************
C      THIS PROGRAM WAS WRITTEN BY JAMES EMERICK
C      THE PROGRAM IS USED TO ADD HYDROGRAPHS AT AN OUTLET
C      AFTER ROUTING
C
C
C     ************************************************************

      REAL Q1(1000),Q2(1000),Q3(1000),Q4(1000),Q5(1000),Q6(1000)
      REAL Q7(1000),Q8(1000),Q9(1000),Q10(1000),QT(1000)
      CHARACTER*20
HYDRO,RHYDRO_A,RHYDRO_B,RHYDRO_C,RHYDRO_D,THYDRO
      CHARACTER*20 RHYDRO_E,RHYDRO_F,RHYDRO_G,RHYDRO_H,RHYDRO_I
      INTEGER INTERV,MAXT,PINC,MAXINT,N,Z,NMAXINT,NH,O,H
     H=0
     PRINT 1
1    FORMAT(/,'*** ENTER THE NUMBER OF THE OUTLET***')
     READ(5,*)O
     PRINT 2
2    FORMAT(/,'**ENTER THE NUMBER OF ROUTED HYRDOGRAPHS**')
     READ(5,*)NH
     PRINT 3
3    FORMAT(/,'**ENTER THE NUMBER OF THE FIRST ROUTED HYDROGRAPH**')
     READ(5,*)H1
     H=H+1
     IF(H.EQ.NH)GOTO 5
     PRINT 4
4    FORMAT(/,'**ENTER THE NUMBER OF THE NEXT ROUTED HYDROGRAPH**')
     READ(5,*)H2
     H=H+1
     IF(H.EQ.NH)GOTO 5
     PRINT 4
     READ(5,*)H3
     H=H+1
     IF(H.EQ.NH)GOTO 5
     PRINT 4
     READ(5,*)H4
     H=H+1
     IF(H.EQ.NH)GOTO 5

5     IF(O.EQ.2)HYDRO='HYDRO_2.TXT'
      IF(O.EQ.3)HYDRO='HYDRO_3.TXT'
      IF(O.EQ.4)HYDRO='HYDRO_4.TXT'
```

```
IF(O.EQ.5)HYDRO='HYDRO_5.TXT'
IF(O.EQ.6)HYDRO='HYDRO_6.TXT'
IF(O.EQ.7)HYDRO='HYDRO_7.TXT'
IF(O.EQ.8)HYDRO='HYDRO_8.TXT'
IF(O.EQ.9)HYDRO='HYDRO_9.TXT'
IF(O.EQ.10)HYDRO='HYDRO_10.TXT'
IF(O.EQ.11)HYDRO='HYDRO_11.TXT'
IF(O.EQ.12)HYDRO='HYDRO_12.TXT'
IF(O.EQ.13)HYDRO='HYDRO_13.TXT'
IF(O.EQ.14)HYDRO='HYDRO_14.TXT'
IF(O.EQ.15)HYDRO='HYDRO_15.TXT'


IF(H1.EQ.3)RHYDRO_A='RHYDRO_3.TXT'
IF(H1.EQ.4)RHYDRO_A='RHYDRO_4.TXT'
IF(H1.EQ.5)RHYDRO_A='RHYDRO_5.TXT'
IF(H1.EQ.6)RHYDRO_A='RHYDRO_6.TXT'
IF(H1.EQ.7)RHYDRO_A='RHYDRO_7.TXT'
IF(H1.EQ.8)RHYDRO_A='RHYDRO_8.TXT'
IF(H1.EQ.9)RHYDRO_A='RHYDRO_9.TXT'
IF(H1.EQ.10)RHYDRO_A='RHYDRO_10.TXT'
IF(H1.EQ.11)RHYDRO_A='RHYDRO_11.TXT'
IF(H1.EQ.12)RHYDRO_A='RHYDRO_12.TXT'
IF(H1.EQ.13)RHYDRO_A='RHYDRO_13.TXT'
IF(H1.EQ.14)RHYDRO_A='RHYDRO_14.TXT'
IF(H1.EQ.15)RHYDRO_A='RHYDRO_15.TXT'

IF(H2.EQ.3)RHYDRO_B='RHYDRO_3.TXT'
IF(H2.EQ.4)RHYDRO_B='RHYDRO_4.TXT'
IF(H2.EQ.5)RHYDRO_B='RHYDRO_5.TXT'
IF(H2.EQ.6)RHYDRO_B='RHYDRO_6.TXT'
IF(H2.EQ.7)RHYDRO_B='RHYDRO_7.TXT'
IF(H2.EQ.8)RHYDRO_B='RHYDRO_8.TXT'
IF(H2.EQ.9)RHYDRO_B='RHYDRO_9.TXT'
IF(H2.EQ.10)RHYDRO_B='RHYDRO_10.TXT'
IF(H2.EQ.11)RHYDRO_B='RHYDRO_11.TXT'
IF(H2.EQ.12)RHYDRO_B='RHYDRO_12.TXT'
IF(H2.EQ.13)RHYDRO_B='RHYDRO_13.TXT'
IF(H2.EQ.14)RHYDRO_B='RHYDRO_14.TXT'
IF(H2.EQ.15)RHYDRO_B='RHYDRO_15.TXT'

IF(H3.EQ.3)RHYDRO_C='RHYDRO_3.TXT'
IF(H3.EQ.4)RHYDRO_C='RHYDRO_4.TXT'
IF(H3.EQ.5)RHYDRO_C='RHYDRO_5.TXT'
IF(H3.EQ.6)RHYDRO_C='RHYDRO_6.TXT'
IF(H3.EQ.7)RHYDRO_C='RHYDRO_7.TXT'
```

```
IF(H3.EQ.8)RHYDRO_C='RHYDRO_8.TXT'
IF(H3.EQ.9)RHYDRO_C='RHYDRO_9.TXT'
IF(H3.EQ.10)RHYDRO_C='RHYDRO_10.TXT'
IF(H3.EQ.11)RHYDRO_C='RHYDRO_11.TXT'
IF(H3.EQ.12)RHYDRO_C='RHYDRO_12.TXT'
IF(H3.EQ.13)RHYDRO_C='RHYDRO_13.TXT'
IF(H3.EQ.14)RHYDRO_C='RHYDRO_14.TXT'
IF(H3.EQ.15)RHYDRO_C='RHYDRO_15.TXT'

IF(H4.EQ.3)RHYDRO_D='RHYDRO_3.TXT'
IF(H4.EQ.4)RHYDRO_D='RHYDRO_4.TXT'
IF(H4.EQ.5)RHYDRO_D='RHYDRO_5.TXT'
IF(H4.EQ.6)RHYDRO_D='RHYDRO_6.TXT'
IF(H4.EQ.7)RHYDRO_D='RHYDRO_7.TXT'
IF(H4.EQ.8)RHYDRO_D='RHYDRO_8.TXT'
IF(H4.EQ.9)RHYDRO_D='RHYDRO_9.TXT'
IF(H4.EQ.10)RHYDRO_D='RHYDRO_10.TXT'
IF(H4.EQ.11)RHYDRO_D='RHYDRO_11.TXT'
IF(H4.EQ.12)RHYDRO_D='RHYDRO_12.TXT'
IF(H4.EQ.13)RHYDRO_D='RHYDRO_13.TXT'
IF(H4.EQ.14)RHYDRO_D='RHYDRO_14.TXT'
IF(H4.EQ.15)RHYDRO_D='RHYDRO_15.TXT'


IF(O.EQ.2)THYDRO='HYDRO_2.TXT'
IF(O.EQ.3)THYDRO='HYDRO_3.TXT'
IF(O.EQ.4)THYDRO='HYDRO_4.TXT'
IF(O.EQ.5)THYDRO='HYDRO_5.TXT'
IF(O.EQ.6)THYDRO='HYDRO_6.TXT'
IF(O.EQ.7)THYDRO='HYDRO_7.TXT'
IF(O.EQ.8)THYDRO='HYDRO_8.TXT'
IF(O.EQ.9)THYDRO='HYDRO_9.TXT'
IF(O.EQ.10)THYDRO='HYDRO_10.TXT'
IF(O.EQ.11)THYDRO='HYDRO_11.TXT'
IF(O.EQ.12)THYDRO='HYDRO_12.TXT'
IF(O.EQ.13)THYDRO='HYDRO_13.TXT'
IF(O.EQ.14)THYDRO='HYDRO_14.TXT'
IF(O.EQ.15)THYDRO='HYDRO_15.TXT'

DO 20 Z=1,1000
Q1(Z)=0
Q2(Z)=0
Q3(Z)=0
Q4(Z)=0
Q5(Z)=0
Q6(Z)=0
```

```
      Q7(Z)=0
      Q8(Z)=0
      Q9(Z)=0
20    CONTINUE

      OPEN(1,STATUS='OLD',FILE=HYDRO)
      IF(NH.GE.1)OPEN(2,STATUS='OLD',FILE=RHYDRO_A)
      IF(NH.GE.2)OPEN(3,STATUS='OLD',FILE=RHYDRO_B)
      IF(NH.GE.3)OPEN(4,STATUS='OLD',FILE=RHYDRO_C)
      IF(NH.GE.4)OPEN(5,STATUS='OLD',FILE=RHYDRO_D)

      READ(1,*)MAXINT1
      READ(1,*)INTERV
      DO 6 Z=1,MAXINT1
      READ(1,*)Q1(Z)
6     CONTINUE

      IF(NH.GE.1)THEN
      READ(2,*)MAXINT2
      READ(2,*)INTERV
      DO 7 Z=1,MAXINT2
      READ(2,*)Q2(Z)
7     CONTINUE
      ENDIF

      IF(NH.GE.2)THEN
      READ(3,*)MAXINT3
      READ(3,*)INTERV
      DO 8 Z=1,MAXINT3
      READ(3,*)Q3(Z)
8     CONTINUE
      ENDIF

      IF(NH.GE.3)THEN
      READ(4,*)MAXINT4
      READ(4,*)INTERV
      DO 9 Z=1,MAXINT4
      READ(4,*)Q4(Z)
9     CONTINUE
      ENDIF

      IF(NH.GE.4)THEN
      READ(5,*)MAXINT5
      READ(5,*)INTERV
      DO 10 Z=1,MAXINT5
      READ(5,*)Q5(Z)
```

```
10     CONTINUE
      ENDIF

      MAXINT=MAXINT1
      IF(MAXINT2.GT.MAXINT)MAXINT=MAXINT2
      IF(MAXINT3.GT.MAXINT)MAXINT=MAXINT3
      IF(MAXINT4.GT.MAXINT)MAXINT=MAXINT4
      IF(MAXINT5.GT.MAXINT)MAXINT=MAXINT5

C     ADDING HYDROGRAPHS
      DO 11 Z=1,MAXINT
      QT(Z)=Q1(Z)+Q2(Z)+Q3(Z)+Q4(Z)
11     CONTINUE

      OPEN(10,STATUS='UNKNOWN',FILE=THYDRO)
      WRITE (10,*) MAXINT
      WRITE (10,*) INTERV
      DO 30 Z=1,MAXINT
      WRITE (10,*) QT(Z)
30     CONTINUE
      STOP
      END
```

BIBLIOGRAPHY

BIBLIOGRAPHY

REFERENCES CITED

1. Sherman, L.K., "Streamflow from Rainfall by the Unit-Graph Method," Eng. News Record, Vol. 108, (April, 1932), pp. 501-505.

2. Snyder, F.F., "Synthetic Unit Graphs," Trans. Am. Geophys. Union, Vol. 19,(1938), pp. 447-454.

3. Clark, C.O., "Storage and the Unit Hydrograph," Trans. Am. Soc, Civ. Eng., Vol. 110

4. Nash, J. E., "The Form of the Instantaneous Unit Hydrograph," Int. Assos. Sci. Hydrol., Pub. 45, 3 (1957): 14-121.

5. Green, W. H. and G.A. Ampt. "Studies on Soil Physics:1. Flow of Air and Water Though Soils," Journal of Agricultural Science, Vol. 4 (1911) pp. 1-24.

6. Marshall, J.S., and Palmer, W.M., "The Distribution of Raindrops with Size," Journal of Meteorology, Vol. 5, (1948).

7. Quimpo R. G., "On the Frequency Distribution of Raindrop Sizes" Journal of Irrigation and Drainage Engineering, Vol. 112, No.2, (1986), pp. 119-129.

8. Grayman, W. M. and Eagleson, P.S. A Review of the Accuracy of Radar and Raingauges for Precipitation Measurements, Cambridge, MA.: MIT Department of Civil Engineering, Ralph M. Parkson Laboratory, 1970. (Technical Report No. 119).

9. Shamsi, U.M. GIS Tools for Water, Wastewater, and Stormwater Systems. ASCE, Reston, VA 2002

10. Michelini, Mark, "Automatic Generation of Unit Hydrographs Using a Digital Elevation Model"(unpublished M.S. thesis, School of Engineering, University of Pittsburgh, 1994)

11. Bodnar, Randal F., "Verification of Automatically Generated Unit Hydrographs from DEM's with Field Data" (unpublished M.S. thesis, School of Engineering, University of Pittsburgh, 1995)

12. Al-Medeij, Jaber H., "Automatic Generation of Runoff Curve Numbers With Geographic Information Systems" (unpublished M.S. thesis, School of Engineering, University of Pittsburgh, 1998)

13. Khalid Naseem Khan "A Geographic Information Systems Based Spatially Distributed Rainfall-Runoff Model" (unpublished M.S. thesis, School of Engineering, University of Pittsburgh, 2001)

14. Chow, Ven T., Handbook of Applied Hydrology. New York: McGraw-Hill, Inc., 1964.

REFERENCES NOT CITED

Bras, Rafael. Hydrology An Introduction to Hydrologic Science. Addison-Wesley Publishing Company, Reading, MA 1990.

Quimpo, Rafael G., Hydrology and Hydraulics. University of Pittsburgh, 1996.

Bedient and Huber. Hydrology and FloodPlain Analysis. 3rd edition. Prentice Hall Inc., Upper Saddle River, NJ 2002.

Getting to Know ArcView the Geographic Information System for Everyone. ESRI Redlands, CA. 1996

Dingman, Lawrence S., Physical Hydrology. 2nd edition. Prentice Hall. Upper Saddle River, NJ 2002

GIS Modules and Distributed Models of the Watershed. ASCE Reston, VA 1999

Hydrology, Section 4, National Engineering Handbook (NEH-4) Natural Resources Conservation Service, 2001.