

1996

Development of hydrologic modeling interface using Arcview and HEC-1

Nadella V.S. Narayana
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Civil and Environmental Engineering Commons](#), [Geographic Information Sciences Commons](#), and the [Hydrology Commons](#)

Recommended Citation

Narayana, Nadella V.S., "Development of hydrologic modeling interface using Arcview and HEC-1" (1996). *Retrospective Theses and Dissertations*. 16785.
<https://lib.dr.iastate.edu/rtd/16785>

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

Development of hydrologic modeling interface using Arcview and HEC-1

by

Nadella V.S. Narayana

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Department: Civil and Construction Engineering

Major: Civil Engineering (Geometronics)

Major Professor: K. Jeyapalan

Iowa State University

Ames, Iowa

1996

Copyright © Nadella V.S. Narayana, 1996. All rights reserved.

Graduate College
Iowa State University

This is to certify that the Master's thesis of
Nadella V.S. Narayana
has met the thesis requirements of Iowa State University

Signatures have been redacted for privacy

TABLE OF CONTENTS

ACKNOWLEDGMENTS		viii
1 INTRODUCTION		1
1.1	Land Cover Map	3
1.2	Hydrology and Flow Map	3
1.3	Subbasin Boundaries Map	3
1.4	Soil Map	7
1.5	Topography Map	7
2 GEOGRAPHIC INFORMATION SYSTEMS		11
2.1	Geographical Concepts	12
2.2	GIS Functional Elements	13
2.2.1	Data Acquisition	14
2.2.2	Data Management	14
2.2.3	Data Manipulation and Analysis	15
2.2.4	Product Generation	15
2.3	Data Structures	16
2.3.1	Raster Data Structure	16
2.3.2	Vector Data Structures	17
2.4	Map Projections and Coordinate Systems	18
2.4.1	Azimuthal Projections	20
2.4.2	Conic Projections	20

2.4.3	Cylindrical Projections	20
2.4.4	Conformal Projections	21
2.4.5	Equivalent Projection	21
2.4.6	Perspective Projection	21
2.5	Major programs in Arc/Info	23
3	HYDROLOGY: AN INTRODUCTION	25
3.1	Hydrologic Cycle	26
3.1.1	Precipitation	27
3.1.2	Evaporation	27
3.2	Factors Affecting Runoff	28
3.3	The Streamflow Hydrograph	28
3.3.1	Annual Hydrograph	29
3.3.2	Stormflow Hydrograph	29
3.4	Synthetic Unit Hydrograph	31
3.4.1	Snyder's Synthetic Unit Hydrograph	31
3.5	Land Surface Runoff Component	36
3.6	River Routing Component	38
3.7	Reservoir Component	39
3.8	Diversion Component	39
3.9	Pump Component	39
4	LINKING ARC/INFO AND HEC-1	40
5	SYSTEM IMPLEMENTATION	44
5.1	A Brief Discussion Of Each Menu Option In Icaria.apr	48
6	EXAMPLE APPLICATION AND RESULTS	52
6.1	Description of Example Application Area	52
6.2	Results	53

7	SUMMARY AND CONCLUSIONS	55
8	RECOMMENDATIONS FOR FUTURE RESEARCH	57
	BIBLIOGRAPHY	59
	APPENDIX	61

LIST OF TABLES

Table 6.1	Input Parameters For HEC-1 Input File	54
-----------	---	----

LIST OF FIGURES

Figure 1.1	Landcover Classification For Lake Icaria Watershed	4
Figure 1.2	Streams Coverage For Lake Icaria Watershed	5
Figure 1.3	Subbasin Delineation For Lake Icaria Watershed	6
Figure 1.4	Soils Coverage For Lake Icaria Watershed	8
Figure 1.5	Contours Coverage For Lake Icaria Watershed	9
Figure 4.1	A Schematic View of the Link Between Arcview and HEC-1	41
Figure 5.1	Screen Capture of The First Arcview Screen	46
Figure 5.2	Screen Capture of The Second Arcview Screen	47
Figure 5.3	Screen Capture of The Discharge Hydrograph at outlet	49

ACKNOWLEDGMENTS

I would like to thank my major professor, Dr. Kandiah Jeyapalan, for his input, inspiration and assistance in the completion of this project. I would also like thank Dr. Udoyara Sunday Tim and Dr. T. Al Austin, for their help in this project. I also wish to express my thanks to Mr. James Majure, Director, GIS Research and Support And Facility at Iowa State University, for his input and assistance at various aspects throughout the project. Finally, I wish to acknowledge my fellow graduate students Sarath Sarathchandra, Wu Yao and Jingfeng Kang for their support during the initial stages of the project.

1 INTRODUCTION

The flood of 1993 caused a great deal of damage to the state of Iowa. Property damage was extensive. The outlook for 1994 looked bleak because rainfall was expected to be higher than normal and the flood control structures had been weakened by the 1993 flood. In the event that the snowpack melted quickly and filled the storage capacity of the river basins, the summer 1994 rainfall would runoff quickly causing widespread damage. To avoid a repeat of the 1993 catastrophe, proper action must be taken at critical times to reduce the flood level of rivers. Such action must be taken at those locations where it is deemed to be most effective. In order to implement effective flood control measures, we need better methods of strategic planning and analysis based upon better information. The Geographic Information Systems (GIS) and the Hydrologic Modeling Systems (HMS) are two separate entities that can be used to improve hydrological simulation by more effective use of geographic information.

GIS facilitates the capture, storage, analysis, manipulation, retrieval and display of geographic data referenced to the earth surface. A GIS uses computer based tools to analyze and manage spatial data. These tools are arguably developed for use in cartography. Today, they are being used by urban engineers, resource managers, earth scientists, and civil and environmental engineers for inventory analysis, planning and modeling.

Surface hydrologic modeling includes processes like generation and transport of runoff sediments and pollutants from watersheds. HMSs are models that will compute the total runoff from a watershed area given the precipitation intensity and duration. GIS and

HMS require coupling to enable communication between them so as to improve hydrological simulation by a more effective use of geographic information. Thus, the coupling of GIS and HMS will enable better water resources planning, land development and flood control. The process of linking the HMS and GIS is as follows. The flow of water is modeled by hydrologists and depends upon the precipitation in sub basins. The stages of a river in a flood event are revealed by the hydrologic model. A Geographic Information System, namely Arc/Info is used to derive areally weighted hydrologic parameters for input to the hydrologic model, HEC-1. The necessary coverages of landuse/landcover, soiluse, basin boundaries are geo-processed using Arc/Info, and the required parameters that go as input in the HEC-1 hydrologic model are obtained from these coverages. The study area covered in this project is the Lake Icaria watershed, which is located in the northeast portion of Adams County, in southwest Iowa. The watershed is primarily un-gaged and is about 27 square miles in area. Water, forests and farmland are the major land covers for this area. The objective of this thesis is to develop a module linking HEC-1 with Arc/Info, so that the hydrologic parameters derived from Arc/Info, can be passed on to and processed by HEC-1 and the resultant hydrograph can be produced. The methodology used to carry out the objective stated above, is expressed in the terminology of the selected software and hardware systems.

The HEC-1 hydrologic model uses the following parameters as its input in determining the runoff due to a given precipitation event. The parameters are rainfall amount and distribution, time of concentration, which is based on flow path characteristics such as length, slope and roughness, loss rates, such as initial loss and either uniform or exponential losses, subbasin area, and curve number, which is related to soil type. Many of these parameters are areally based. That means an area is required in order to quantify the parameters. For this reason alone, GIS will be useful in obtaining an estimation of the above parameters. The coverages that will reside in a GIS in Arc/Info format, and are required to output the above stated parameters are shown below.

1.1 Land Cover Map

Satellite imagery in the form of Landsat Thematic Mapper (TM) data with 30 meter resolution using four of seven available bands is the raw data. ERDAS, an image processing software, is used to classify the Landsat TM data and produce the final landcover/landuse map. The landcover map thus produced, is the landcover coverage used for this project and helps us estimate the curve number, one of the important parameters that go into the HEC-1 input file. The landcover classification for Lake Icaria is shown in Figure 1.1.

1.2 Hydrology and Flow Map

The hydrology coverage is acquired from the TIGER/Line files that are available on a county basis. Programs written in the Arc Macro Language (AML) are used to extract the hydrologic features from the principal TIGER coverages. Since the TIGER coverages are available on a county basis, the TIGER coverage for Adams County will be obtained and the hydrology coverage for Lake Icaria watershed is clipped out from it. This clipped hydrology coverage for lake Icaria watershed is the coverage that is used in the project. See Figure 1.2 for the hydrological delineation of Lake Icaria watershed.

1.3 Subbasin Boundaries Map

The river boundaries and the subbasin boundaries are digitized manually from the existing topography (contour) coverage of the area, with the streams coverage in the background. See Figure 1.3 for the subbasin delineations.

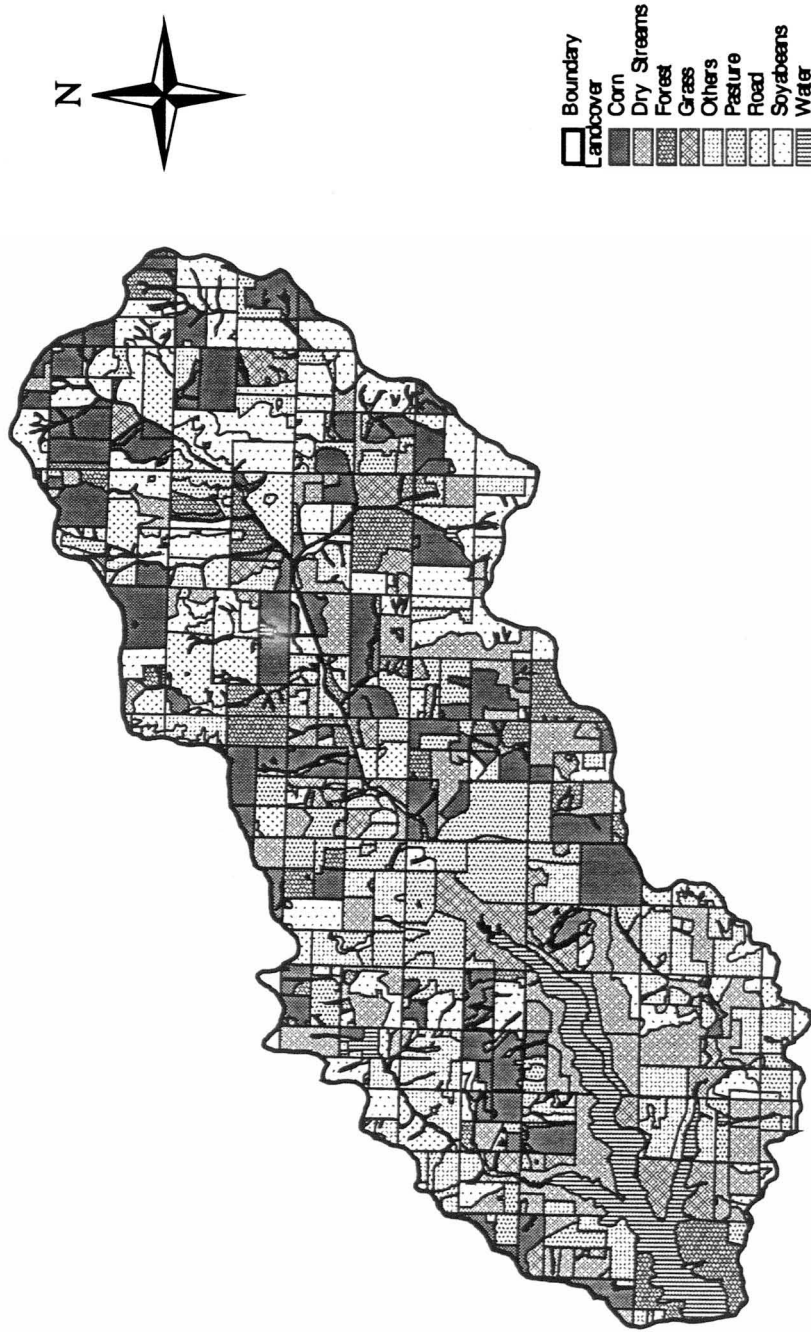


Figure 1.1 Landcover Classification For Lake Icaria Watershed

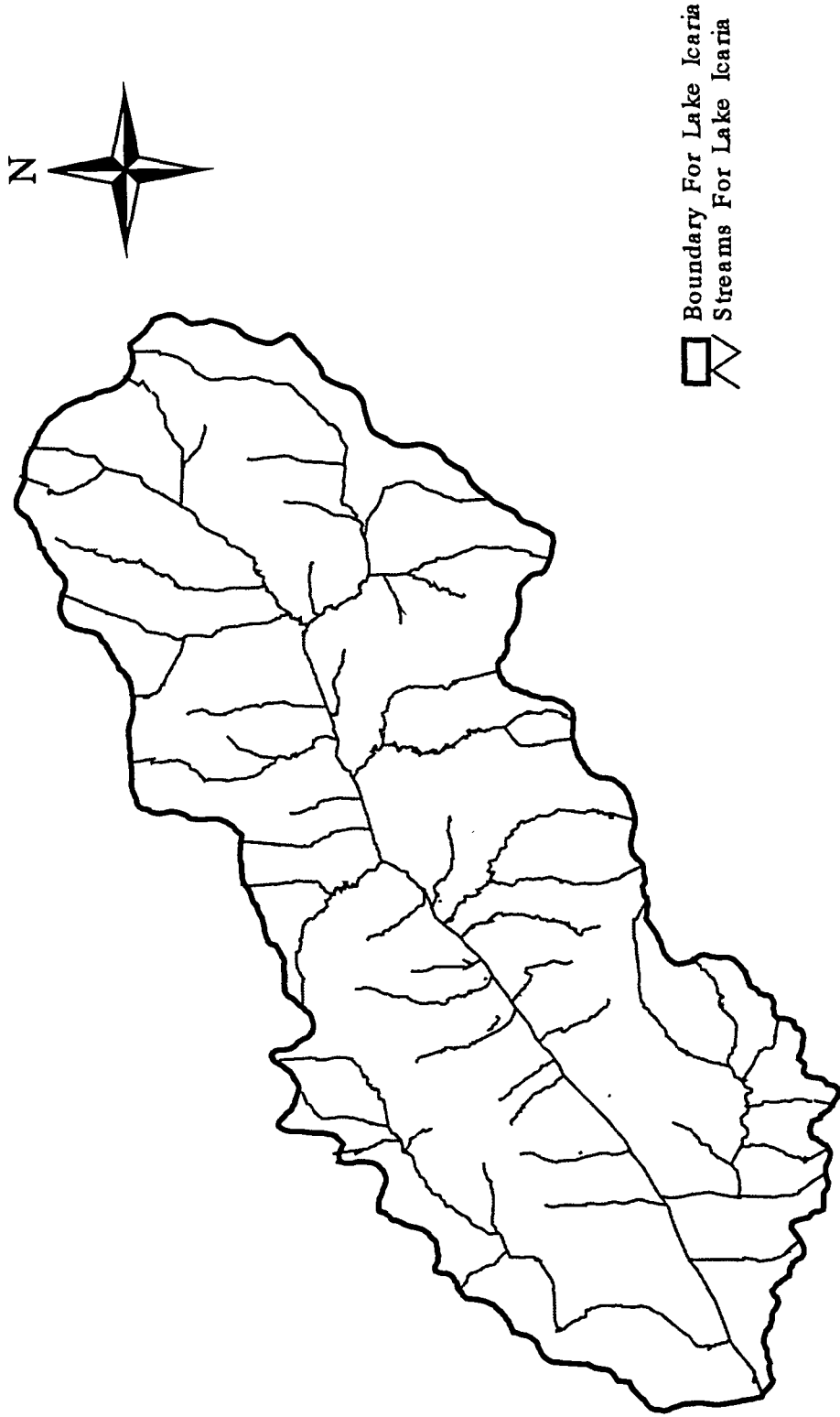


Figure 1.2 - Streams Coverage For Lake Icaria Watershed

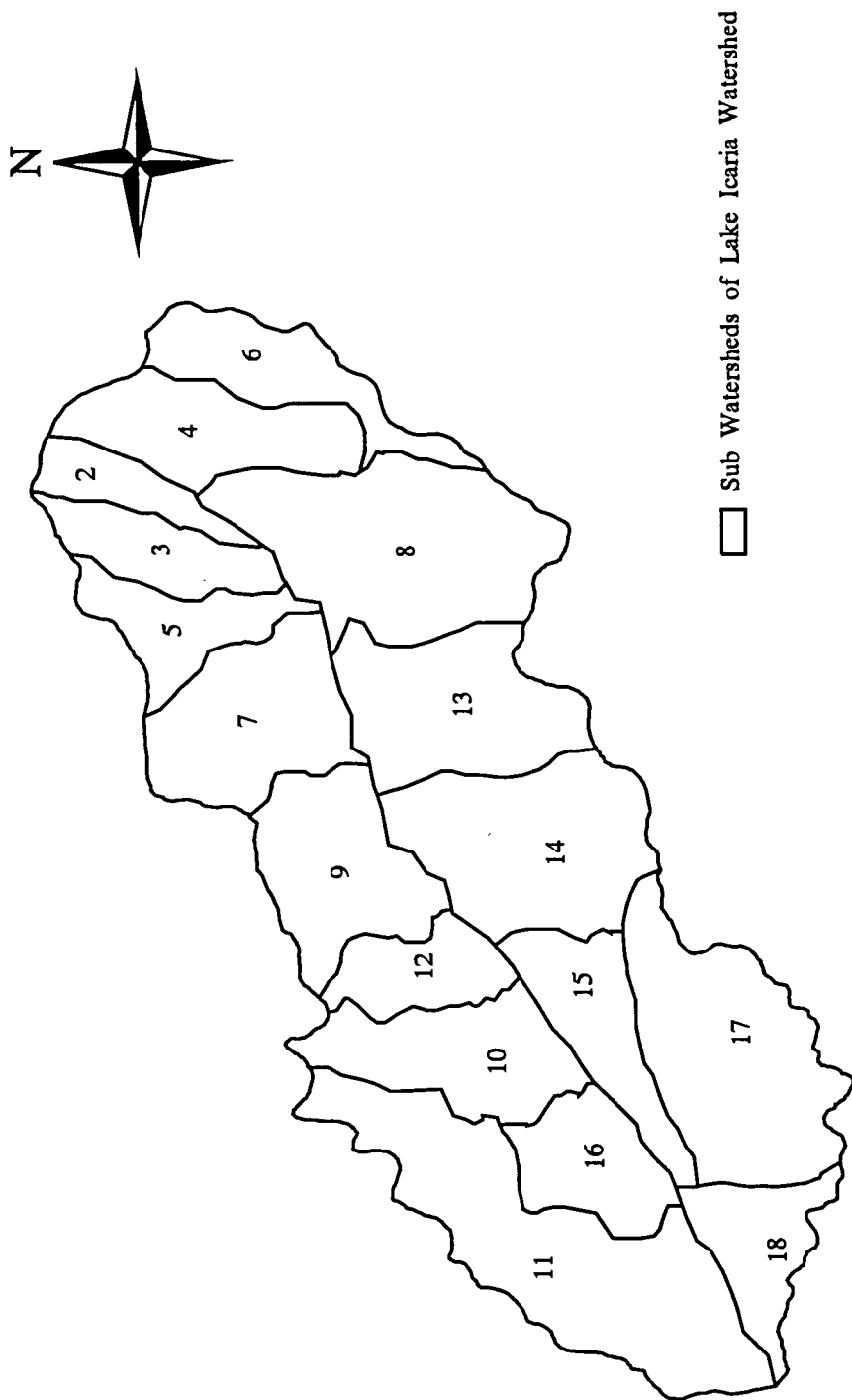


Figure 1.3 Subbasin Delineation For Lake Icaria Watershed

1.4 Soil Map

The soil map for Lake Icaria is obtained from the USDA Soil Conservation Service for Adams County. They have been scanned and converted to Arc/Info format and will be used as such in the project. The soil of Lake Icaria has been delineated according to its soil hydrologic group classification in Figure 1.4.

1.5 Topography Map

The contour coverage for Lake Icaria is obtained by scanning the 1:24000 scale quadrangle maps produced by the United States Geological Survey (USGS). These quadrangle maps are scanned and then rubber sheeted along the common border to obtain the contour coverages. See Figure 1.5 for the topography in Lake Icaria watershed.

The contour coverage along with the streams coverage is used to delineate the subbasins for the watershed. These subbasins are then digitized and stored in a separate coverage. The soils coverage is overlaid on the landuse/landcover coverage and the resultant composite coverage now contains a soil hydrologic group for every landuse type. These two parameters along with the assumed antecedent moisture conditions establish a curve number for each polygon in the composite coverage. From the streams coverage, a separate coverage for the channel flow in the watershed is developed, which is then used to compute the parameters that will be used to derive the runoff hydrograph using the Snyder's Unit Hydrograph method. Thus, Arc/Info is used to generate a set of input parameters to HEC-1, for each of the subbasins in the watershed, and these parameters are then structured in the HEC-1 input file, along with the precipitation values, to obtain a runoff hydrograph for a given intensity and duration of precipitation. The area under the hydrograph yields the cumulative runoff from the watershed. By varying the precipitation values, the user is able to notice the change in the shape of the output hydrograph, and the corresponding runoff values. At this stage, the link between

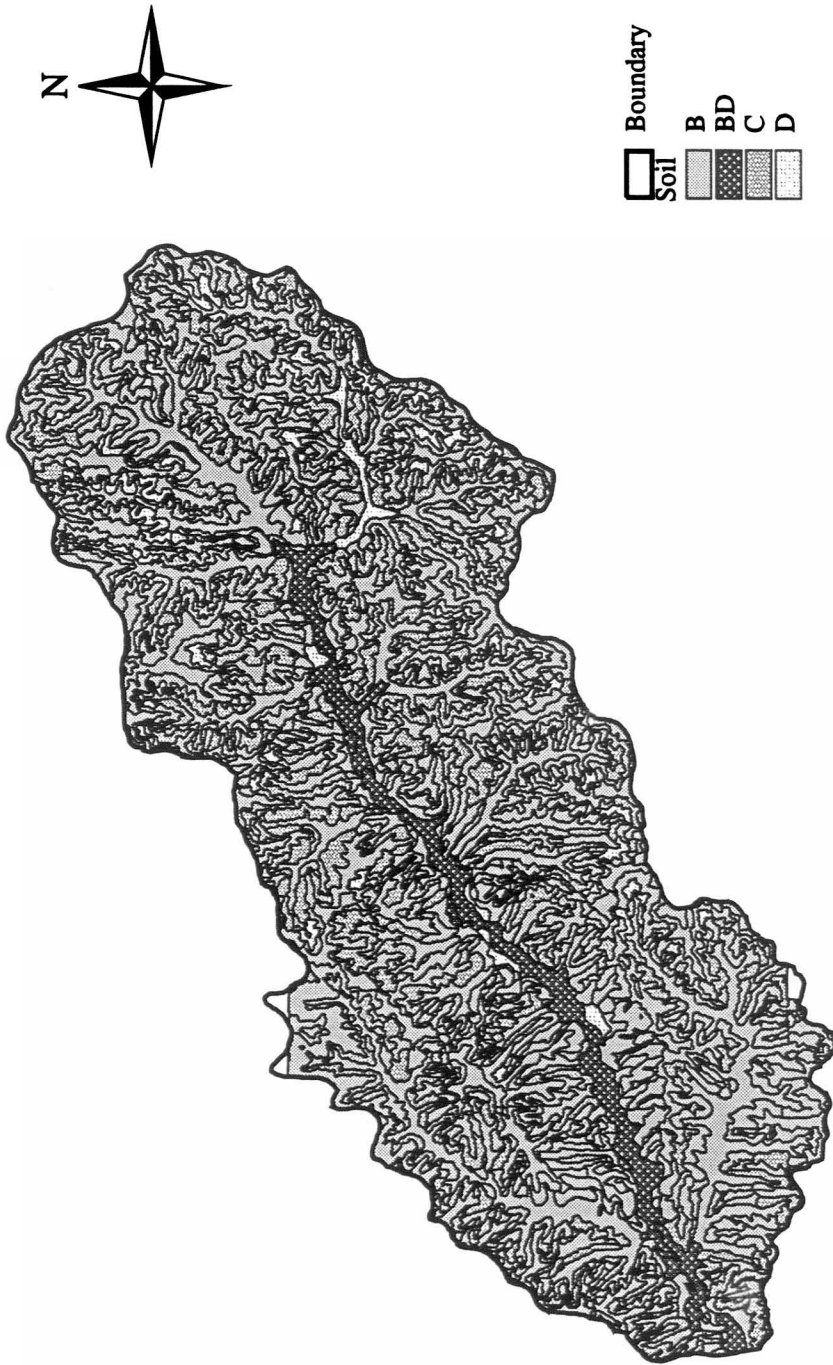


Figure 1.4 Soils Coverage For Lake Icaria Watershed

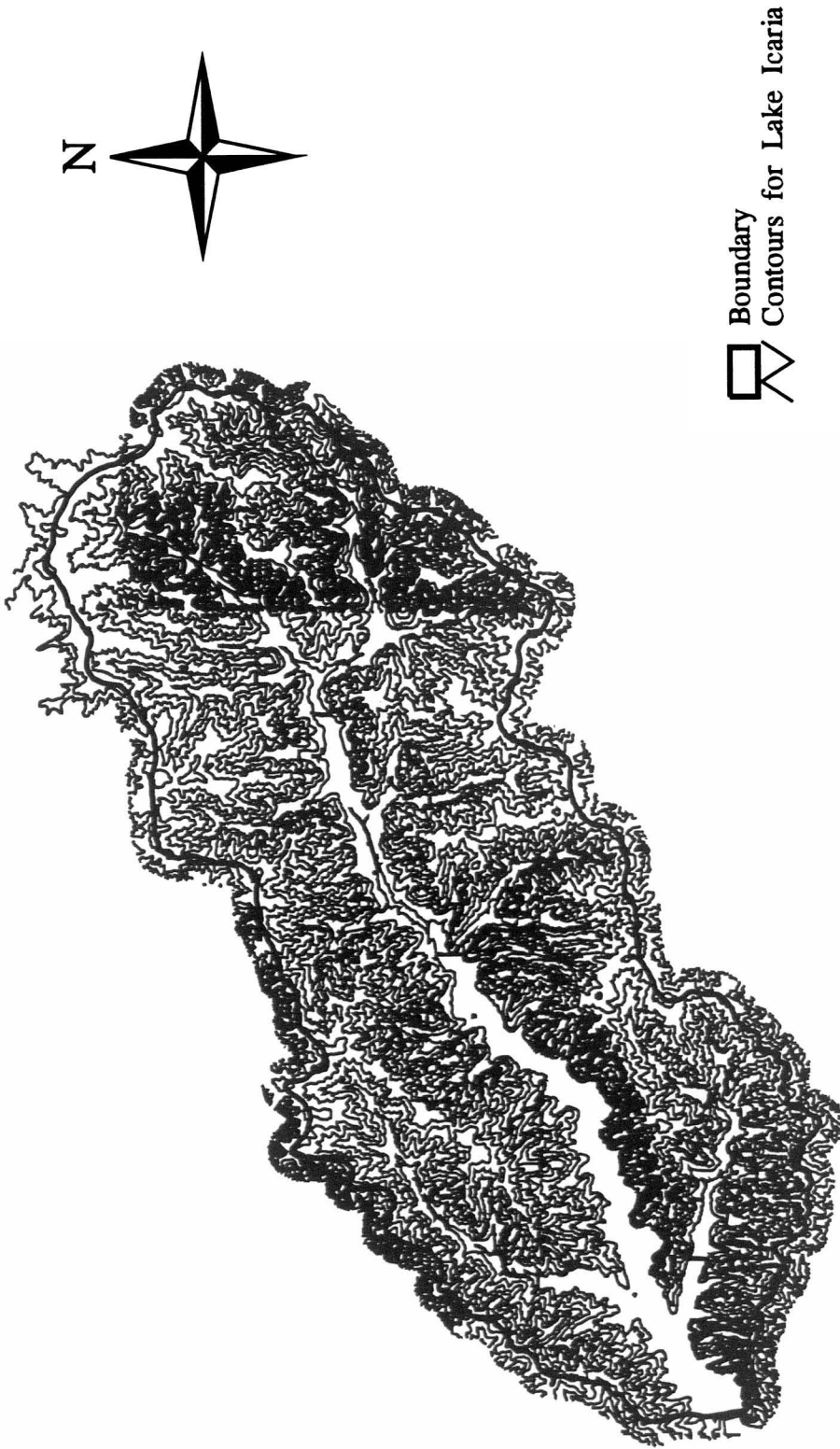


Figure 1.5 Contours Coverage For Lake Icaria Watershed

Arc/Info and HEC-1 is considered accomplished.

The procedure described above has been accomplished. Using Arc/Info, the required coverages of soil and landuse have been overlaid on top of one another and a composite coverage formed. The subbasins coverage has been formed and the required input parameters fed into the HEC-1 input file. The input file has been processed by HEC-1, to give the resultant output hydrograph.

Given below is a list and a brief description of the subsequent chapters that follow. The first chapter introduces and deals with the concept of Geographic Information Systems (GIS) and Hydrologic Modeling Systems (HMS) and how the two systems can be coupled to utilize the positive features of both. It also gives an overview of the project in terms of the idea pursued and the software packages used. The second chapter deals in more detail with the concept of GIS, by describing the different functional elements of GIS and the different data structures available and map projections and coordinate systems. The third chapter deals in more detail about the concept of hydrology and describes the hydrologic cycle, precipitation, evaporation, runoff and other topics that are relevant to an understanding of the project. It also deals with HEC-1 in more detail, with special emphasis on its different modeling components. In the fourth chapter, the linkage between Arc/Info and HEC-1 is described in its different component stages. The fifth chapter describes the process of data transfer between Arc/Info and HEC-1, using Avenue, the object oriented programming language in Arcview. Arcview is a querying and display software that is manufactured by ESRI, the manufacturers of Arc/Info. Avenue makes it easy to conduct the data transfer and has been used for that purpose essentially. The sixth chapter has the results along with an example application and the seventh chapter has the summary and conclusions and finally, the eighth chapter contains recommendations for future research that will further enhance the utility of this project.

2 GEOGRAPHIC INFORMATION SYSTEMS

A Geographic Information System (GIS) is a computer system for managing spatial data. The word “Geographic” implies that locations of the data items are known, or can be calculated, in terms of geographic coordinates. Most GISs are restricted to data in two spatial dimensions. However, data for a few systems can be in three dimensions. The word “Information” means that data in a GIS is organized to yield useful knowledge, such as colored maps, images, statistical graphs, tables and various on-screen responses to interactive queries. The word “System” implies that a GIS is made from several interrelated and linked components having different functions. Thus, a GIS has functional capabilities for data input, manipulation, transformation, visualization, combination, query, analysis, modeling and output. A GIS consists of a package of computer programs with a user interface, commonly known as a GUI, or by means of a command language, consisting of program statements that dictate the sequence and type of operations. GISs are computer tools for manipulating maps, digital images and tables of geocoded data items, such as the result of a geodetic survey. It is a system designed to bring together spatial data from diverse sources into a unified database, often employing a variety of digital data structures, and representing spatially varying phenomena as a series of data layers, all of which are in a spatial register, meaning that they overlap correctly at all locations.

GIS is designed to work with spatial or geographic coordinates. In other words, a GIS is both a database system with specific capabilities for spatially referenced data, and it is also a set of operations for working with those data. A GIS can be either

manual or automated. A manual GIS usually comprises several data elements including maps, sheets of transparent material used as overlays, aerial and ground photographs, statistical reports and field survey reports. These set of data are compiled and analyzed with such instruments such as stereo viewers and mechanical and electronic planimeters.

2.1 Geographical Concepts

Here, we shall introduce a few terms that are relevant to our discussion on GIS. These are terms of common usage in the world of GIS, and are used by the GIS professionals in their practice.

Spatial objects are delineated geographic areas, with a number of associated attributes or characteristics. These objects may be points, lines or polygons. A point is a spatial object with no area associated with it. It may however, have a host of attributes attached to it that define the point. For example, depending upon the scale of the map, a wastewater plant may be a point with a treatment capacity, location address, area and perimeter as its attributes. A line is a spatial object made up of a connected sequence of points. Lines have no width, and thus, a specified location must be on one side of the line or other, but never on the line itself. Attributes we could attach to that line are its length, location address, name of the entity the line represents, e.g. river, road, etc. Nodes are special kinds of points, usually indicating the junction between the lines or the end of line segments. A polygon is a closed area figure. Simple polygons are undivided areas, while complex polygons are divided into areas of similar characteristics, which in themselves are called simple polygons. An example of a simple polygon is the classroom on the second floor of the Town Engineering building. A complex polygon would be the Town Engineering building, which has several classrooms, restrooms, offices, etc. Chains are special kinds of line segments corresponding to a portion of the building edge of a polygon. For example, the boundary line that corresponds to the eastern edge of the

campus, can be considered as a chain. The above given terms are some of the commonly used object terms in GIS.

Scale and resolution are two terms associated with every cartographic product. By scale, we mean the ratio of the distances represented on a map or photograph to their true lengths on the earth's surface. Scale values are normally written as dimensionless numbers, indicating that the measurements on the map and the earth are in the same units. A scale of 1:2500, indicates that one unit on the map corresponds to 2500 of the same units on the ground. Scale always refers to the linear horizontal distances and not to measurements of elevations and areas. The terms large scale and small scale are in common use. Maps larger than 1:10000 are called large scale maps and maps smaller than 1:10000 are called small scale maps. Resolution is another concept that we are interested in. Most dictionaries refer to the term "resolution" in terms of distinguishing the individual parts of an object. But for the purpose of GIS, Tobler (1987) defines spatial resolution for geographic data as the content of the geometric domain divided by the number of observations, normalized by the spatial dimension. The domain, for two dimensional datasets like maps and photographs, is the area covered by the observations. Thus, for two-dimensional data, we need to take the square root of the ratio to normalize the value. When we have more information, the mean resolution element gets smaller, which means that we have a higher resolution set of data. Conversely, a lower resolution data will have fewer observations in an area, and thus, a larger mean resolution.

2.2 GIS Functional Elements

There are five functional elements that a GIS must contain. By common consensus, they are data acquisition, data preprocessing, data management, data manipulation and analysis, and product generation.

2.2.1 Data Acquisition

It is the process of identifying and gathering the data required for an application. This typically involves a lot of procedures. New data may be gathered by preparing large scale maps of natural vegetation from field observations, or by contracting for aerial photography. Other procedures of data acquisition may include locating and acquiring existing data, such as maps, aerial and ground photography, survey of many kinds, and documents from archives and depositories. Data preprocessing involves manipulating the data in several ways so that it may be entered into the GIS. Two of the principal tasks in data preprocessing include data format conversion, and identifying the location of objects in the original data in a systematic way. Converting the format of the original data often involves extracting information from maps, photographs and printed records and then recording this information in a computer database. This process is time consuming and a costly effort for organizations. A second key task of the preprocessing phase is to establish a consistent system for recording and specifying the location of objects in the datasets. When this task is completed, it is possible to determine the characteristics of any specified location in terms of the contents of any data layer in the system.

2.2.2 Data Management

Data Management functions govern the creation of, and access to, the database itself. These functions provide consistent methods for data entry, update, retrieval and deletion. Modern database management systems isolate the users from the details of data storage. When the operations of data management are executed well, the users usually do not notice the efficiency of the database system. However, when they are done poorly, everyone notices. The system is slow, cumbersome, and easy to disrupt. When poorly done, the smallest human and machine errors can create large problems for both the users and the system managers. Data management concerns include issues

of security. Procedures must be in place to provide different users with different kinds of access to the system and its database.

2.2.3 Data Manipulation and Analysis

These functions are often the central focus of attention for a user of the system. Analytic operators of GIS believe this module of GIS to be the only existing one. Since no single GIS can encompass the complete range of analytic operations that a user may need, we must be able to move data and information between systems. This kind of modularity where other data processing and analysis systems can be linked to a GIS, is very valuable in many circumstances, and permits the systems to be easily extended over time by pairing them with other analytical tools.

2.2.4 Product Generation

Finally Product Generation is the phase where final output from the GIS is created. This output might include statistical reports, maps and graphics of various kinds. Some of these may be soft copy images available on computer monitors, while others may be hard copies of the resultant graphics. Thus, the capability of taking the output of an analytic process and placing it back into the geographic database for future reference and analysis, is extremely important and useful.

GIS fulfills a very real need given the rapid growth of digital spatial data in geosciences. Many datasets are now being generated by government agencies, private companies and university researchers, and they would be ineffectively used and result in wasted resources without good systems of data management, like GIS. Having talked about the five basic functionalities of a GIS, let us now talk about the data structures available to us.

2.3 Data Structures

There are a number of different ways to organize data in an information system. The choice of a particular data structure is one of the important decisions in designing a geographic information system. Each different kind of spatial data or theme in a GIS is referred to as a data layer or a theme. The essential function of the spatial data that we store is to subdivide the earth's surface into meaningful entities or objects that can be characterized. Points, such as location of oil or water wells and lines, the centerlines of roadways or streams, are key elements of this breakdown. When we consider bounded regions, such as the borders of a subdivision or the edges of a reservoir, we often focus on the boundary lines, and call the enclosed region as a polygon. In spatial data processing, common usage relaxes the requirement that lines be straight. We use the term polygon even if the boundaries are curved. In this case, we try to think of the curved portion in terms of small straight lines. There are two different types of data structures that require special mention, since they are used very often in the world.

2.3.1 Raster Data Structure

One of the simplest data structures is a raster or cellular organization of spatial data. In a raster structure, a value for the parameter of interest, for example, elevation above datum in meters, landuse class from a specified list, plant biomass in grams per square meter, and so forth is developed for every cell in an array over space. The horizontal dimension of the simplest raster, along with the rows of the array, is often oriented parallel to the east and west direction for convenience. Following the conventional practice in image processing, raster elements in this direction along the rows of the array are sometimes called samples and are numbered starting from the top left boundary. This referencing of the cells is different from the more traditional georeferencing systems such as latitude and longitude in which one specific point on the earth's surface is the origin.

Its also different from the Universal Transverse Mercator (UTM) system, where (in the northern hemisphere), the origin of the system is in the lower left corner, which is similar to a Cartesian coordinate system. Raster datasets, in practice, can be very large in size. As an example, satellite remote sensing data is frequently used to distinguish categories of land cover over large areas. The standard view of the earth from the Landsat series of satellites covers approximately 30,000 square kilometers, which at the nominal size of 30 meters per pixel, corresponds to approximately 35 million raster cells or pixels (picture element). When dealing with such large datasets, there are several algorithms for compressing data, such as run-length encoding and chain coding.

2.3.2 Vector Data Structures

A mathematician may define a vector as a quantity with a starting coordinate, and an associated displacement and direction (or bearing). In a description of spatial data based on vectors, we can make an assumption that an element may be located at any location, without the positional constraints of a raster array. Vector data structures are based on elemental points whose locations are known to arbitrary precision, in contrast to a raster or cellular data structure. As an example, to store a circle in a raster data structure, we might find and encode all the raster cells whose locations correspond to the boundary of the circle. This might be called a low level description, on the other hand, might efficiently store a circle by recording a point location for the center of the circle, and specifying the radius. For spatial data in most geographic information systems, the coordinate data is encoded, and after input processing, is stored as combination of point, lines and areas or polygons. Several forms of vector data structures are in common use, both as representative database type within geographic information systems, as well as standards for data transfer between systems.

2.4 Map Projections and Coordinate Systems

Now, we shall talk about maps projections and coordinate systems. A map represents geographical features or other spatial phenomena by graphically conveying information about locations and attributes. Locational information describes the position of particular geographic features on the earth's surface, as well as spatial relationships between features, such as the shortest path from a fire station to a library, the proximity of competing businesses, and so on. Attribute information describes characteristics of the geographic features represented, such as the feature type, its name and number, and quantitative information such as its area and length. Maps are flat, but the surfaces they represent, are curved. Transforming three dimensional space into a two dimensional map is termed "projecting". The phenomena is called projection. Projection formulas are mathematical expressions which convert data from a geographical location (latitude and longitude) on a sphere or spheroid to a representative location on a flat surface. This process inevitably distorts at least one of the properties of shape, area, distance, direction and often, more. Although its actually a spheroid, the earth is sometimes treated as a sphere to make calculations easier. This assumption that the earth is a sphere can be used for small scale maps. At this scale, the difference between a sphere and a spheroid cannot be detected on a map. However, to maintain accuracy for large scale maps or larger, the earth has to be treated as a spheroid. As a sphere is based on a circle, so is a spheroid based on an ellipse. The shape of an ellipse can be defined by two different radii. The longer axis is called the major axis and the shorter one is called minor axis. The respective radius of each axis is termed as the semi-major and semi-minor axis. Just as rotating a circle about an axis defined by its diameter will create a sphere, rotating an ellipse about either its major or minor axis will produce an ellipsoid. An ellipsoid which approximates a sphere is called a spheroid. An ellipsoid which approximates the shape of the earth is formed by the rotation around the minor

axis. The earth has been surveyed many times to better understand its surface features and their peculiar irregularities. From these surveys, many spheroids of the earth have been defined. The semi-major and the semi-minor axes that best fit one geographical region are not necessarily the same ones that fit another region. Until recently, values determined by Clarke in 1866 have been used to describe the spheroid commonly used with the reference datum for North America. This is often referred to as the North American Datum 1927 (NAD 27).

Because it is difficult to make measurements in spherical coordinates, geographic data is projected into a planar coordinate system. Once the sphere or portion is projected onto a flat surface, locations are identified by x and y coordinates on a grid, with the origin at the center of the grid. Each position has two values which reference it to that central location, one specifying its horizontal position and the other its vertical location. These two values are called the x coordinate and the y coordinate respectively. Using this notation, the coordinates at the origin are x equals zero and y equals zero. On a gridded network constructed of equally spaced horizontal and vertical lines, the horizontal line in the center is called the x-axis and the vertical line in the center is called the y-axis. Equal spacing represents units consistent across the full range of x and y. Horizontal lines above the origin and vertical lines to the right of the origin are assigned positive values. Those below and to the left of the origin are assigned negative values. The four quadrants shown below represent the four possible combinations of positive and negative x and y coordinates. There are an infinite number of different map projections, since there are infinite number of ways to project the features of the earth's surface onto a plane.

Map projections are classified in different ways. A primary differentiation of classes of projections is based on the geometrical model of the projection. Based on this criterion, the different types of projections could be listed as follows.

2.4.1 Azimuthal Projections

In azimuthal projections, the map is constructed by placing a plane tangent to a point on the surface of the earth. Features on the map are generated by systematically transferring the locations of these features from the sphere to the plane. We can imagine a transparent earth, with an azimuthal plane tangent to the north pole, and a point light source at the south pole. The rays of light from the source may be traced in a straight line through the points on the earth's surface, to the tangent plane. Distortions are minimal in the immediate vicinity of the tangent point.

2.4.2 Conic Projections

These projections are based on a cone placed over the earth, oriented so that the intersection of cone and the earth forms one or two small circles. Polyconic projections involve a series of cones, each used to map small fraction of the surface of the globe. For cones oriented conventionally, the axis of the cone is the same as the axis of the earth's rotation. In this case, the small circles, at the intersection of the cone and the earth correspond to latitude lines. This family of projections is used for displaying areas that extend a great distance in the east-west direction than in the north-south direction, to take advantage of the areas on the map having minimal distortion.

2.4.3 Cylindrical Projections

A cylindrical projection is based on a cylinder placed over the earth. In the simplest case, the cylinder is tangent to the sphere, with the intersection forming a great circle. In the most common orientation, the intersection is the equator.

Another set of classifications of map projections involves the patterns of deformations or distortions in the map. Based on this criterion, the different types of projections are given below.

2.4.4 Conformal Projections

If a projection has no angular deformations, its called conformal. In order to maintain the accuracy of angles, shapes of the objects are distorted. Thus conformal projections are not a good choice for the measurement of distances, since the scale changes across the map. The smaller the area, the less important this problem is. The most well known of the conformal projections is the Mercator projection.

2.4.5 Equivalent Projection

This type of a projection displays relative areas correctly. Equivalence is an important consideration for data used as input to a geographic information system, so that inferences about the areas area correct. By modifying the features on a map so that the areas become correct, directions become unreliable. Common equivalent projections are Albers Conic, a frequent choice for maps that cover a large east-west distance, and Lambert Azimuthal equal area projections.

2.4.6 Perspective Projection

This type of a projection represents the way a camera views its world. They are based on straight lines that pass from the object at a distance through a specified point of intersection at a finite distance from the plane of projection. Perspective projections present many problems to the user of a GIS. One of the simplest problems, is that the scale in a perspective representation of the earth's surface varies as a function of the distance from the viewer. Objects of the same size will appear smaller when they are further away.

In practice, many geographic information system applications use plane coordinate systems such as the Universal Transverse Mercator (UTM) system. These have the advantage that distances recorded in the database correspond to the distances that

one might measure on the (hypothetically) flat ground. When applications require the accurate measurement of area over very large distances, coordinates are often stored in latitude longitude system, with the calculations based on an equal area projection of the globe.

Having discussed some of the fundamental concepts in GIS, we will now discuss some features of the GIS software that was used in this project. The software used in this project is Arc/Info. Arc/Info is a registered trademark product of the Environmental Systems Research Institute (ESRI), Redlands, California. It is a very popular software and is used all over the world in various fields of study. Arc/Info consists of five major programs, namely, Arc, Arcplot, Arcedit, Grid and Info/Tables. Arc/Info has a layered architecture. The foundation is the data engine used to access and manage the geographic database. At the next level, Arc/Info contains a powerful and flexible command language known as the Arc Macro Language (AML) that provides access to sophisticated geoprocessing tools which operate on the various data sources supported by Arc/Info. Commands are organized functionally into a series of programs for editing, mapping, analysis, table operations and data management. The Arc Macro Language provides the development environment in which sophisticated macro procedures are automated and customer user interfaces built. The facility of Inter application Communication (IAC) allows the user to call other application software from within Arc/Info. Thus Arc/Info can be used as a GIS data and process server. Arc/Info works with a number of data types, namely, coverages, grids, tables, TINs and images. These are the GIS data sources on which Arc/Info's geoprocessing tools operate. Datasets are organized into Arc/Info workspaces, directories that contain the data sources for a geographical area.

Coverages represent the fundamental source of data for Arc/Info. They are useful for representing geographic features such as points, lines and areas, and are highly suited for data automation, associating descriptive attributes with features, and for

many analytical operations. Coverages, along with tables, are the primary data sources managed by Arc/Info's spatial database manager, called ArcStorm. Grids are Arc/Info software's raster data structure used to represent categorical data such as soil types and to represent continuous surfaces such as elevation. Grids are highly suited for spatial modeling and raster editing. The GRID extension to Arc/Info is used for raster analysis of grid datasets. Tables are used to store descriptive attributes in rows. Each attribute is stored in a field or item. There is one record of attributes for each feature. In this way, the feature attribute tables can be related or linked to geographical features. This concept is the foundation for the georelational data model. Images store satellite data or photographs that have been scanned. Map images such as aerial photos and satellite imagery serve as backdrops in GIS and are used in image processing systems such as the ERDAS IMAGINE software package. Picture images such as photos and scanned documents can describe features much like table attributes do. TINs are used to represent surfaces requiring highly accurate definition, such as contour elevation for civil engineering and map quality contour generation. Themes provide a method for accessing and using all Arc/Info data sources. A theme is a collection of geographic features such as wells, streets, soil polygons, or parcels. A set of themes for the same map extent can be organized into a view for display and query purposes. Having discussed some of the fundamental features and sub systems of Arc/Info, let us now briefly discuss the major programs in Arc/Info, that make the geoprocessing of spatial and attribute data possible. These are actually subsystems in Arc.

2.5 Major programs in Arc/Info

Arc/Info is composed of a series of major programs that group key functionality into useful operating environments. Each program contains a set of commands to perform geoprocessing operations. The command prompt indicates which program you are using

at a particular point in a session. The following is a listing of the major programs in Arc/Info.

ARC: Used for workspace management, data conversion, maintenance and spatial analysis.

ARCEDIT: Arc/Info's interactive digitizing and editing system.

ARC PLOT: Used for map display and query as well as for performing many sophisticated spatial operations such as pathfinding, dynamic segmentation, surface analysis and spatial selection.

INFO and TABLES: Two programs used to operate on tabular data files and attribute files containing information about the arc coverages.

GRID: A program provided with the GRID software extension to Arc/Info that operates on grid datasets to perform raster analysis and display.

3 HYDROLOGY: AN INTRODUCTION

Water is the most abundant substance on earth, the principal constituent of all living things, and a major force constantly shaping the surface of the earth. It is also a key factor in the air conditioning of the earth for human existence and in influencing the progress of civilization. Hydrology, that deals with all phases of the earth's water, is a subject of great importance for people and their environment. Practical applications of hydrology are found in the design and operation of hydraulic structures, water supply, wastewater treatment and disposal, irrigation, drainage, hydro power generation, flood control, navigation, erosion and sediment control, salinity control, pollution abatement, recreational use of water, and fish and wildlife protection. The role of applied hydrology is to help analyze the problems involved in these tasks and to provide guidance for the planning and management of water resources. The hydrosociences deal with the waters of the earth, their distribution and circulation, their physical and chemical properties, and their interaction with the environment, including interaction with living things and in particular, human beings. Hydrology encompasses all the hydrosociences. Defined more strictly, hydrology is the study of the hydrologic cycle, that is, the endless circulation of water between the earth and its atmosphere. Thus, hydrology can be defined as the science which deals with the processes governing the depletion and replenishment of the water resources of the land areas of the earth. It is concerned with the transportation of water through the air, over the ground surface, and through the earth's strata. A knowledge of hydrology is important in practically all problems that involve the use and supply of water. Therefore hydrology is of value not only in the field of engineering but

also in forestry, agriculture, and other branches of natural science. Hydrology is broad in its scope. The topics in this chapter will be limited to only those aspects of hydrology relevant to the research topic (Hydrologic Modeling Using GIS).

3.1 Hydrologic Cycle

Water on earth exists in a space called the hydrosphere which extends about 15 km up into the atmosphere and about one kilometer down into the lithosphere, the crust of the earth. Water circulates in the hydrosphere through the maze of paths constituting the hydrologic cycle. The hydrologic cycle is the central focus of hydrology. The cycle has no beginning or end, and its many processes occur continuously. We shall pick it up at the evaporation stage. Water evaporates from the oceans and the land surface to become part of the atmosphere, water vapor is transported and lifted in the atmosphere until it condenses and precipitates on the land or the oceans, the precipitated water may be intercepted by vegetation, become overland flow over the ground surface, infiltrate into the ground, flow through the soil as subsurface flow, and discharge into streams as surface runoff. Much of the intercepted water and the surface runoff returns to the atmosphere through evaporation. The infiltrated water may percolate deeper to recharge groundwater, later emerging in springs or seeping into streams to form surface runoff, and finally flowing out to the sea or evaporating into the atmosphere to continue the cycle. Estimating the total amount of water on the earth and in the various processes of the hydrologic cycle has been a topic of scientific exploration since the second half of the nineteenth century. However, quantitative data are scarce, particularly over the oceans, and so the amounts of water in the various stages of the global hydrologic cycle are still not precisely known.

Of the many meteorological processes occurring continuously within the atmosphere, the processes of precipitation and evaporation, in which the atmosphere interacts with

the surface water, are the most important for hydrology. Much of the water precipitated on the land surface is derived from moisture evaporated from the oceans and transported long distances by atmospheric circulation. The two basic driving forces of atmospheric circulation result from the rotation of the earth and the transfer of heat energy between the equator and the poles.

3.1.1 Precipitation

Precipitation includes rainfall, snowfall, and other processes by which water falls to the land surface, such as hail or sleet. The formation of precipitation requires the lifting of an air mass in the atmosphere so that it cools and some of its moisture condenses. The three main mechanisms of air mass lifting are frontal lifting, where warm air is lifted over cooler air by frontal passage, orographic lifting, in which an air mass rises to pass over a mountain range, and convection lifting, where air is drawn upwards by convective action, such as in the center of a thunderstorm cell. Precipitation varies in space and time according to the general pattern of atmospheric circulation and according to local factors. Higher precipitation occurs near the coasts than inland because the oceans supply the bulk of the atmospheric moisture for precipitation. Pronounced seasonal variation in precipitation occurs where the annual oscillation in the atmospheric circulation changes the amount of moisture inflow over those regions.

3.1.2 Evaporation

The two main factors influencing evaporation from an open water surface are the supply of energy to provide the latent heat of vaporization and the ability to transport the vapor away from the evaporative surface. Solar radiation is the main source of heat energy. The ability to transport vapor away from the evaporative surface depends on the wind velocity over the surface and the specific humidity gradient in the air above it. Evaporation from the land surface comprises evaporation directly from the

soil and vegetation. The resultant flow after a portion of the precipitated water has been infiltrated and evaporated, is called runoff. The practical objective of Hydrology is to provide a means of determining the characteristics of the hydrograph that may be expected for a stream draining any particular basin. The basic principles of this science are applied to determine the maximum flood flow expected to occur with any stated frequency for a given basin, the minimum flow anticipated under a given set of conditions, the monthly, annual or average long term yield of runoff and the response of a watershed to changing scenarios, that is the sensitivity analysis of a model.

3.2 Factors Affecting Runoff

The flow of any stream is determined by two entirely different sets of factors, the one controlled by the climate with special reference to the precipitation, and the other, by the physical characteristics of the drainage basin. The influence of the first group depends upon the type of precipitation, rainfall intensity, duration of rainfall, distribution of rainfall on basin, direction of storm movement, antecedent precipitation and soil moisture. The influence of the second group is determined by the following characteristics of land use, types of soil, area, shape, elevation, slope, orientation, types of drainage net, extent of indirect drainage and artificial drainage.

3.3 The Streamflow Hydrograph

A streamflow or annual hydrograph is a graph or table showing the flow rate as a function of time at a given location on the stream. In effect, the hydrograph is an integral expression of the physiographic and climatic characteristics that govern the relations between rainfall and runoff of a particular drainage basin. In the above context, two types of hydrographs are very important. They are the annual hydrograph and the storm hydrograph.

3.3.1 Annual Hydrograph

The annual hydrograph, a plot of streamflow against time over a year, shows the long time balance of precipitation, evaporation and streamflow in a watershed.

3.3.2 Stormflow Hydrograph

Study of annual hydrographs shows that peak streamflows are produced infrequently, and are the result of storm rainfall alone or storm rainfall and snowmelt combined. The hydrograph showing the runoff resulting from storm rainfall only, is called a storm hydrograph.

This brings us to that topic of the unit hydrograph, the least common denominator, from which hydrographs of varying intensities can be developed and have been developed using various methods. A Unit Hydrograph is the unit pulse response function of a linear hydrologic system. It was first proposed by Sherman (1932), who defined the unit hydrograph of a watershed as a direct runoff hydrograph (DRH) resulting from one inch (usually taken as 1 centimeter in SI units) of excess rainfall generated uniformly over the drainage area at a constant rate for an effective duration. The word "unit" was originally used by Sherman to denote a unit of time, but since then, it has often been referred to as a unit depth of excess rainfall. The unit hydrograph was defined for only surface runoff, after Sherman divided the excess runoff into surface runoff and groundwater runoff. Thus, the unit hydrograph is a simple linear model that can be used to derive the hydrograph resulting from any excess rainfall. The following assumptions are inherent in the model. The excess rainfall has a constant intensity within the effective duration, the excess rainfall is evenly distributed throughout the entire drainage area, the base time of the Direct Runoff Hydrograph (DRH) resulting from an excess rainfall of given duration is constant, the ordinates of all the direct runoff hydrographs of a common base time are directly proportional to the total amount of direct runoff represented by

each hydrograph. For a given watershed, the hydrograph resulting from a given excess rainfall reflects the unchanging characteristics of the watershed.

Under natural conditions, the above requirements cannot be perfectly satisfied. However, when the hydrologic data to be used are carefully selected so that they come close to meeting the above assumptions, the results obtained by the unit hydrograph model are generally acceptable for practical purposes. Although the model was originally devised for large watersheds, it has been found applicable to small watersheds from less than 0.5 hectares to 2.5 square kilometers. Some cases do not support the use of the model because one or more of the assumptions are not satisfied. For such reasons, the model is considered inapplicable to runoff originating from snow or ice. The following constraints apply.

Concerning assumptions, the storms selected for analysis should be of short duration, since these will most likely produce an intense and nearly constant excess rainfall rate, yielding a well defined single peaked hydrograph of short time base.

The unit hydrograph may become inapplicable when the drainage area is too large to be covered by a nearly uniform distribution of rainfall. In such cases, the area must be subdivided and each subarea has to be analyzed for storms covering the whole subarea.

The base time of the DRH is generally uncertain, but depends on the method of baseflow separation. The base time is usually short if the direct runoff also includes subsurface runoff.

The principles of proportionality and superposition are assumed so that the ordinates Q_n of the DRH may be computed by the equation $Q_n = P_m U_{n-m+1}$, where n is the width of the time interval chosen and m is the initial time chosen, with M being the maximum time duration. Actual data are not truly linear. Hence when applying the above equation, the resulting hydrograph is only an approximation, which is satisfactory in many practical cases.

The unit hydrograph is considered unique for a given watershed and invariable with

respect to time. This is called the principle of time invariance which together with the principle of superposition and proportionality, is fundamental to the unit hydrograph model.

Usefulness of the unit hydrograph lies in the fact that once it has been obtained it may be applied to determine the direct runoff and streamflow hydrographs. A rainfall hyetograph is selected, the abstractions are estimated, and the excess rainfall hyetograph is calculated. The time interval used in defining the excess rainfall hyetograph ordinates must be the same as that for which the unit hydrograph was specified. The discrete convolution equation $Q_n = P_m U_{n-m+1}$ may then be used to yield the direct runoff hydrograph. By adding an estimated baseflow to the direct runoff hydrograph, the streamflow hydrograph can be obtained.

3.4 Synthetic Unit Hydrograph

The unit hydrograph developed from rainfall and streamflow data on a watershed applies only for that watershed and for the point on the stream where the streamflow data were measured. Synthetic unit hydrograph procedures are used to develop unit hydrographs for other locations on the stream in the same watershed, or for nearby watersheds with similar characteristics. There are three types of synthetic unit hydrographs. Those relating hydrograph characteristics (peak flow, rate, base time, etc.) to watershed characteristics, those based on a dimensionless unit hydrograph and those based on models of watershed storage. In this research project, we shall try and produce a Snyder's Synthetic hydrograph for the Lake Icaria watershed.

3.4.1 Snyder's Synthetic Unit Hydrograph

In a study of watersheds located mainly in the Appalachian highlands of the United States, and varying in size from about ten to ten thousand square miles, Snyder found

synthetic relations for some characteristics of a standard unit hydrograph. Additional such relations were found later on, and in modified form, are given below. From the relations, five characteristics of a required unit hydrograph for a given duration of excess rainfall may be calculated. These five characteristics are the peak discharge per unit of watershed area, $q_p R$, the basin lag time, t_{pR} (time difference between the centroid of the excess rainfall hyetograph and the unit hydrograph peak), the base time t_b , and the widths W (in time units) of the unit hydrographs at 50 and 75 percent of the peak discharge. Using these characteristics, the unit hydrograph can be drawn. Snyder defined a standard unit hydrograph as one whose rainfall duration t_r is related to the basin lag t_p by the relation $t_p = 5.5 * t_r$.

For a standard hydrograph, Snyder made the following deductions.

The basin lag is $t_p = C_1 * C_t * (L * L_c)$, where t_p is in hours, L is the length of the main stream in Kilometers (or miles) from the outlet to the upstream divide, L_c is the distance in kilometers (miles) from the outlet to a point on the stream nearest the centroid of the watershed area, C_1 is 0.75 (1.0 for the English system), and C_t is a coefficient derived from gaged watersheds in the same region.

The peak discharge per unit drainage area in $m^3/s/Km^2$ of the standard unit hydrograph is $q_p = C_2 * C_p/t_p$, where C_2 is 2.75 (640 for the English System) and C_p is a coefficient derived from the gaged watersheds in the same region. To compute C_t and C_p for a gaged watershed, the values of L and L_{ca} are measured from the basin map. From a derived unit hydrograph of the watershed are obtained values of its effective duration t_R in hours, its basin lag t_{pR} in hours, and its peak discharge per unit drainage area, q_{pR} , in $m^3/s/Km^2/cm$. If $t_{pR} = 5.5t_R$, then $t_R = t_r$, $t_{pR} = t_p$ and $q_{pR} = q_p$, and C_p and C_t are computed by the two equations given above. If t_{pR} is quite different from $5.5t_R$, then the standard basin lag is $t_p = t_{pR} + (t_r - t_R)/4$ and the above given equations are solved simultaneously for t_r and t_R . The values of C_t and C_p are then computed from the above equations with $q_{pR} = q_p$ and $t_{pR} = t_p$. When an ungaged watershed appears

to be similar to a gaged watershed, the coefficients C_t and C_p for the gaged watershed can be used in the above equations to derive the required synthetic unit hydrograph for the ungaged watershed. The relationship between q_p and the peak discharge per unit drainage area q_{pR} of the required unit hydrograph is $q_p = q_{pR} t_p / t_{pR}$.

The base time t_b in hours of the unit hydrograph can be determined using the fact that the area under the unit hydrograph is equivalent to a direct runoff of 1cm (1 inch in the English system). Assuming a triangular shape for the unit hydrograph, the base time may be estimated by $t_b = C_3 / q_{pR}$, where C_3 is 5.56 (1290 for the English System).

The width in hours of a unit hydrograph at a discharge equal to a certain percent of the peak discharge q_{pR} is given by $W = C_w q_{pR}^{-1.08}$, where C_w is 1.22 (440 for the English System) for the 75 percent width and 2.14 (770 for the English System) for the 50 percent width. Usually one third of this width is distributed before the unit hydrograph peak time and two-thirds after the peak.

As mentioned earlier, in this research project, we shall compute the synthetic unit hydrograph using Snyder's method. The hydrologic runoff model HEC-1 will be used to compute the synthetic unit hydrograph for the study area of Lake Icaria, a watershed in Adams County, in the southwest portion of Iowa. Since the hydrologic runoff model to be used in this project is HEC-1, we will devote a few lines to explaining the HEC-1 software package.

The HEC-1 Flood Hydrograph Package, a combination of computer programs was originally developed in 1967 by Leo R. Beard and other members of the Hydrologic Engineering Center (HEC) staff. The first version of the HEC-1 package program was published in October 1968. It was expanded and revised and published again in 1969 and 1970. The first package version represented a combination of several smaller programs which had previously been operated independently. These computer programs are still available at HEC as separate programs. In 1973, the 1970 version of the program underwent a major revision. The computational methods used by the program

remained basically unchanged, however the input and output formats were almost completely restructured. These changes were made in order to simplify input requirements and to make program output more meaningful and readable. In 1981, major revisions were made to the 1973 version of the program. The program input and output formats were completely revised and the computational capabilities of the dam-break (HEC-1DB), project optimization (HEC-1GS) and kinematic wave (HEC-1KW) special versions of HEC-1 were combined into one program. The new program included the powerful analysis features available in all the previous programs, together with some additional capabilities, in a single easy to use package. A microcomputer version (PC version) of the HEC-1 program was developed in late 1984. The PC version contained all the hydrologic and hydraulic computation capabilities of the mainframe, however, the flood damage and ogee spillway capabilities were not included because of microcomputer memory and compiler limitations at that time. The current version, HEC-1 1990, represents improvements and expansions to the hydrologic simulation capabilities together with interfaces to the HEC Data Storage System. The entire HEC-1 package, including the DSS interface, is available on the PC and HARRIS minicomputers. HEC-1 with the DSS package is not supported on any other computer system. The DSS capability allows storage and retrieval of data from or for other computer programs as well as the creation of report quality graphics and tables. New hydrologic capabilities include Green and Ampt infiltration, Muskingum Cunge flood routing, reservoir releases input over item, and improved numerical solution of kinematic wave equations. The Muskingum Cunge routing may also be used for the collector and main channels in a kinematic wave land surface runoff calculation. Current information about the program is available from the center. The HEC-1 model is designed to simulate the surface runoff response of a river basin to precipitation by representing the basin as an interconnected system of hydrologic and hydraulic components. Each component models an aspect of the precipitation runoff process within a portion of a basin, commonly referred to as

a subbasin. A component may represent a surface runoff entity, a stream channel, or a reservoir. Representation of a component requires a set of parameters which specify the particular characteristics of the component and mathematical relations which describe the physical processes. The result of the modeling process is the computation of streamflow hydrographs at the desired locations in the river basin.

There are some theoretical assumptions and limitations to the implementation of this mode. A river basin is represented as an interconnected group of subareas. The assumption is made that the hydrologic processes can be represented by model parameters which reflect average conditions within a subarea. If such averages are inappropriate for a subarea then it would be necessary to consider smaller subareas within which the average parameters apply. Model parameters represent temporal as well as spatial averages. Thus the time interval to be used should be small enough such that averages over the computation interval are applicable. There are several important limitations to the model. Simulations are limited to a single storm due to the fact that provision is not made for soil moisture recovery during periods of no precipitation. The model results are in terms of discharge and not stage, although stages can be printed out by the program based on a user specified rating curve.

The stream network simulation model capability is the foundation of the HEC-1 program. All other program computation options build on this option's capability to calculate flood hydrographs at the desired locations in a river basin. A river basin is subdivided into an interconnected system of stream network components using topographic maps and other geographic information. A basin schematic diagram of these components is developed by the following steps that are given below.

The study area watershed boundary is delineated first. In a natural or open area this can be done from a topographic map. However, supplementary information, such as municipal drainage maps, may be necessary to obtain an accurate depiction of an urban basin's extent.

Segmentation of the basin into a number of subbasins determines the number and types of stream network components to be used in the model. Two factors that are important for basin segmentation are the study purpose and the hydrometeorological variability throughout the basin. First, the study purpose defines the areas of interest in the basin, and hence, the points where the subbasin boundaries should occur. Second, the variability of the hydrometeorological processes and basin characteristics impacts on the number and location of subbasins. Each subbasin is intended to represent an area of the watershed, which on average, has same hydrologic or hydraulic properties. Further, the assumption of uniform precipitation and infiltration over a subbasin becomes less accurate as the subbasin becomes larger. Consequently, if the subbasins are chosen appropriately, the average parameters used in the components will accurately model the subbasins.

Each subbasin is to be represented by a combination of model components. Subbasin runoff, river routing, reservoir, diversion and pump components are available to the user.

The subbasins and their components are linked together to represent the connectivity of the river basin. HEC-1 has available a number of methods for combining or linking together outflow from different components. This step finalizes the basin schematic.

Since we are going to use HEC-1 as the hydrologic runoff model in our project, we will devote a few lines on HEC-1, with special emphasis on its different modeling components. HEC-1 has different modeling components that are used as and when appropriate. The following is a brief discussion on the different modeling components of HEC-1.

3.5 Land Surface Runoff Component

The subbasin land surface runoff component is used to represent the movement of water over the land surface and in stream channels. The input to this component is a precipitation hyetograph. Precipitation excess is computed by subtracting infiltration

and detention losses based on a soil water infiltration rate function. The resulting rainfall excesses are then routed by the unit hydrograph or kinematic wave techniques to the outlet of the subbasin producing a runoff hydrograph. The unit hydrograph technique produces a runoff hydrograph at the most downstream point in the subbasin. If that location for the runoff computation is not appropriate, it may be necessary to further subdivide the subbasin or use the kinematic wave method to distribute the local inflow. The kinematic wave rainfall excesses to runoff transformation allows for the uniform distribution of the land surface runoff along the length of the main channel. This uniform distribution of local inflow (subbasin runoff) is particularly important in areas where many lateral channels contribute flow along the length of the main channel. Base flow is computed relying on an empirical method and is combined with the surface runoff hydrograph to obtain flow at the subbasin outlet.

HEC-1 is a deterministic lumped parameter event simulation model, for modeling a single rainfall runoff event. The HEC-1 is probably the most widely used hydrologic event simulation model. The acronym HEC stands for Hydrologic Engineering Center, the U.S. Army of Corps of Engineers research facility in Davis, California, where this model was developed. HEC-1 is designed to simulate the surface runoff resulting from precipitation, by representing the basin as an interconnected system of components. Each component models an aspect of the rainfall-runoff process within a subbasin or subarea. Components include subbasin surface runoff, stream channels, and reservoirs. Each subbasin is represented by a set of parameters that specifies the particular characteristics of the component and the mathematical relations describing its physical processes. The end result of the modeling process is the computation of direct runoff hydrographs for various subareas and streamflow hydrographs at desired locations in the watershed. A subarea land surface runoff component is used to represent the movement of water over the land surface and into stream channels. The input to this component is a rainfall hyetograph. Excess rainfall is computed by subtracting infiltration and detention losses, based on an

infiltration function that may be chosen from several options, including the SCS curve number loss rate. Rainfall and infiltration are assumed to be uniformly distributed in time and space over the subbasin. The resulting rainfall excesses are then applied to the unit hydrograph to derive the subarea outlet runoff hydrograph. Unit hydrograph options include the Snyder's unit hydrograph, the SCS dimensionless unit hydrograph, etc. In our research project we will use the former method to determine the unit hydrograph, from a given precipitation event. For a given watershed, the connectivity of the stream network components is implied by the order in which the input data components are arranged. Simulation must always begin at the uppermost subarea in a branch of the stream network, and proceed downstream until a confluence is reached. Before simulating below the confluence, all flows above it must be routed. The flows are combined at the confluence and the combined flow is then routed downstream. This process is repeated, until the downstream most portion of the watershed has been reached. The HEC-1 input file which is created, contains the required parameters for each sub watershed and is provided as input to the HEC-1 program. Where appropriate the rainfall values are also provided. The output from the program is a hydrograph which is then studied. Simulation can be done, by changing one or a combination of parameters for a sub watershed and examining the response hydrograph.

3.6 River Routing Component

A river routing component, is used to represent flood wave movement in a river channel. The input to this component is an upstream hydrograph resulting from individual or combined contributions of subbasin runoff, river routings or diversions. If the kinematic wave method is used, the local subbasin distributed runoff is also input to the main channel and combined with the upstream hydrograph as it is routed to the end of the reach. The hydrograph is routed to a downstream point based on the characteristics

of the channel. The river routing component is one of the important components used in flood management and flood routing. It is often made use of to predict the amount of flood that will be caused due to a precipitation of given duration and intensity.

3.7 Reservoir Component

The reservoir component can be used to represent the storage outflow characteristics of a reservoir, lake, detention pond, highway, culvert, etc. The reservoir component functions by receiving upstream inflows and routing these inflows through a reservoir using storage routing methods. Reservoir outflow is solely a function of storage in the reservoir and not dependent on downstream controls.

3.8 Diversion Component

The diversion component is used to represent channel diversions, stream bifurcations, or any transfer of flow from one point of a river basin to another point in or out of the basin. The diversion component receives an upstream inflow and divides the flow according to a user prescribed rating curve.

3.9 Pump Component

The pump component can be used to simulate action of pumping plants used to lift runoff out of low lying ponding areas, such as behind levees. Pump operation data describes the number of pumps, their capacities, and “on” and “off” elevations.

4 LINKING ARC/INFO AND HEC-1

This chapter describes the interface between Arc/Info and HEC-1 to facilitate the passing of data and parameters from Arcview to HEC-1 and obtaining the output from HEC-1 and plotting it as a hydrograph (curve) in Arcview. Although the title suggests that the linking was done between Arc/Info and HEC-1, that is not the case. Arcview is a spatial display and query software that is also manufactured by ESRI, the same company that manufactures and markets Arc/Info. Arcview was used to establish a link between Arc/Info and HEC-1. Arcview is very easy to use and understand and comes with an object oriented programming language (OOPL) called Avenue. The whole stage of creating the interface happened in three stages, the Arc/Info stage, the Arcview stage and finally the HEC-1 stage. A schematic diagram showing the link is displayed in Figure 4.1.

The Arc/Info stage is described below.

First the soils coverage and the landuse coverage of Lake Icaria were obtained as described in Chapter 2, and were overlaid on top of each other, to create a composite coverage called "soiland". This facilitated the allocation of curve numbers, since curve numbers (CNs) are a function of the soil hydrologic group and the landuse type and the Antecedent Moisture Condition (AMC). The AMC in our case was assumed to be AMC II. Each of the polygons in the composite coverage obtained as described above, were assigned a curve number, based on the soil hydrologic group and the landuse type.

Next, the subwatershed boundaries coverage called "wsheds" was digitized, using the contours coverage and the rivers coverages in the background.

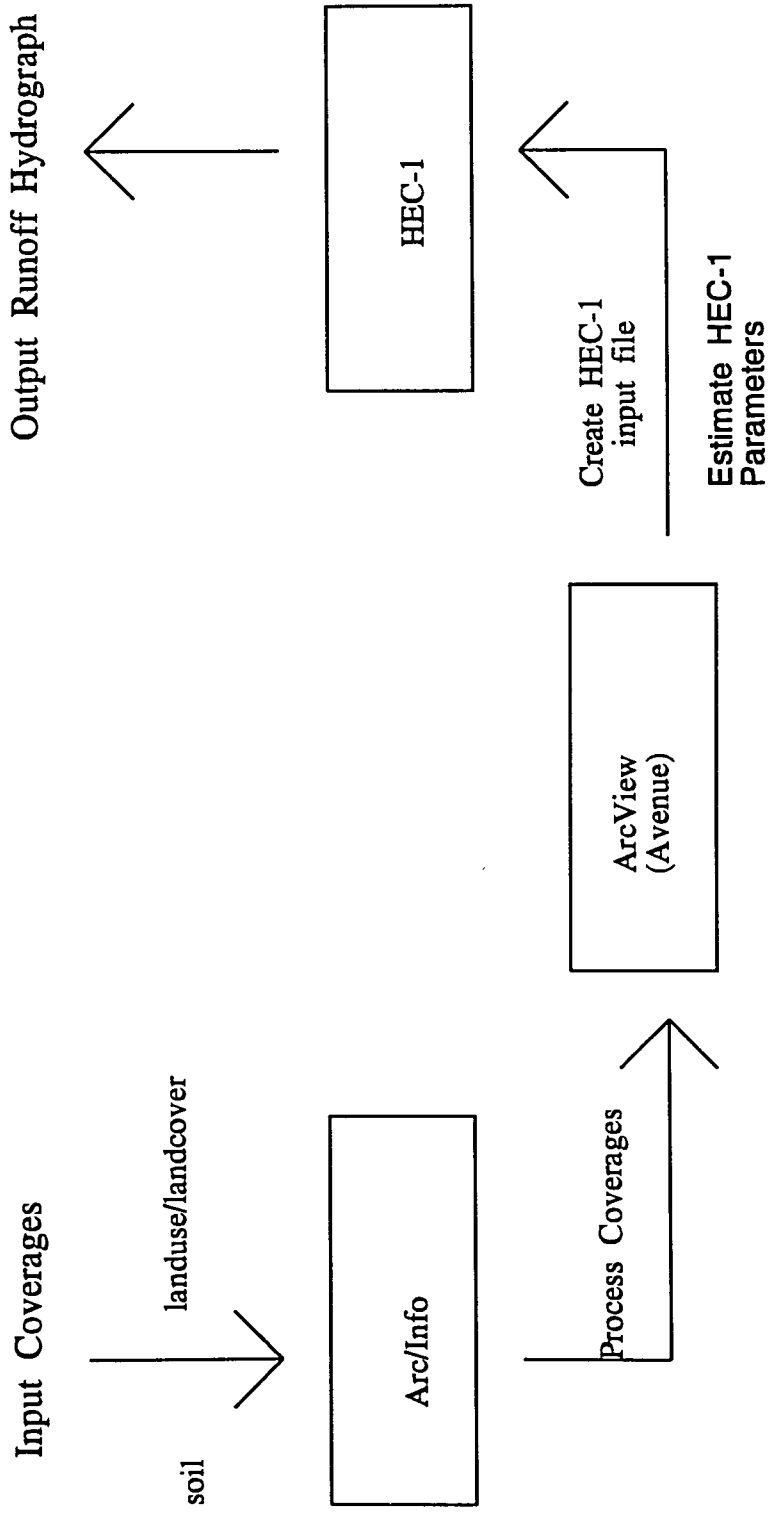


Figure 4.1 A Schematic View of the Link Between Arcview and HEC-1

Finally, the subwatersheds coverage “wsheds” and the composite soil landuse coverage “soiland” were overlaid on each other and the final composite coverage containing the subbasin boundaries, the soil type and the landuse type for each polygon was obtained. This composite coverage was called “soilandwsheds”. This coverage made it easy to calculate the area weighted curve number for each of the subbasins. These weighted curve numbers went into the HEC-1 input file as one of the several required parameters. The coding for extracting the area weighted curve numbers for each subbasin, was done in Avenue.

The hydrologic modeling was done by first starting at the furthestmost point in the furthestmost watershed from the outlet. The individual runoffs from various subbasins were then calculated using the SCS curve number technique and Snyder’s unit hydrograph technique. The flow from the subbasins above a confluence was then combined at the confluence and the combined flow was routed downstream. Care was taken to see to it that all the upstream combined flows were routed before proceeding downstream of a confluence. The Muskingum technique was used to route the combined flows.

The Arc/Info stage was followed by the Arcview stage.

First, in Arcview, a project called “Icaria.apr” was created, which contained the composite coverage “soilandwsheds”. The feature attribute table for this composite coverage was then processed, to obtain the area weighted curve number for each subbasin and stored in a separate info file, called “wsheds.sta”.

Then the entire HEC-1 input file called “Icaria.dat” was created using Avenue, inputting the appropriate parameters at the right stage of the file. Avenue, being an object oriented programming language made it easy to accomplish much with just little code. Avenue was particularly useful while formatting the parameters and inserting them at the right column values, since this was of importance when HEC-1 processes the input file. Some of the other parameters were calculated in Avenue, using known values. Other parameters, like the CNs for each subbasin, for example, were aggregated over

each subbasin and stored in a file named “wsheds.sta” and were then transferred from the “wsheds.sta” file to the HEC-1 input file, while other parameters like the precipitation data, were input directly using Avenue statements. Once the entire input file was created, it was then sent as input to HEC-1 for processing. This indicated the end of the Arcview state and the beginning of the HEC-1 stage.

The creation of the HEC-1 input file lead to the beginning of the HEC-1 stage.

In the HEC-1 stage, the input file “Icaria.dat” was processed by HEC-1 and the output was observed. Of interest were the coordinates of the runoff hydrograph, which determined the output runoff hydrograph, that we were looking for. These coordinates were then extracted from the file and a separate file was created, containing only the coordinates of the output discharge hydrograph. These coordinates were then imported into Arcview and were plotted as a discharge hydrograph.

The above stage concluded the project, whose objective was to do hydrologic modeling by creating an interface between Arc/Info (Arcview) and HEC-1 to facilitate the transfer of data between the two programs and thereby use the strengths of both the softwares, to achieve an application software, and more importantly, cut down on the hydrologic modeling time.

5 SYSTEM IMPLEMENTATION

The following is a step-by-step approach to using the interface for the Lake Icaria watershed and is designed to take the user through the steps of estimating the HEC-1 parameters, to formatting the input file for HEC-1 and finally, to plotting the output hydrograph. The steps appear as separate paragraphs, with each step shown in one paragraph.

At the Unix system prompt, type "Arcview Icaria.apr". Arcview then opens up the lake Icaria project and the first screen to appear is shown in the Figure 5.1. The initial **menu bar** consists of the options "File", "Project", "Window", "Scripts" and "Help" as shown in the Figure 5.1. At the end of this chapter, there will be a brief discussion of each menu option under each of these menu bar options. The Arcview project also contains a **tool bar** with two tools and a **project box**, with the various components "Views", "Tables", "Charts", "Scripts" and "Layouts".

Highlight the project box by clicking on it and then click on the "Views" option. The project box now shows a view titled "Lake Icaria" which is blacked. Double click the blacked view "Lake Icaria", and open up the view in the view area. At this stage the menu bar changes its options to reflect those for manipulating the view.

Enlarge the view area, so that the themes are well viewed.

The initial view opens up with the "Subbasin Delineation Coverage", the "Streams and Rivers Coverage" and the "Hydrography Coverage" for Lake Icaria, as shown in Figure 5.2. This view can be manipulated by turning on the themes the user wants and turning off the themes the user doesn't want. If however the user opens Arcview using

the command “Arcview ” at the system prompt, then the screen shown in Figure 5.1 is the initial screen. But for running this example application, it is recommended that the user open Arcview using the command “Arcview Icaria.apr”.

Now, click on the top bar of the project box and the menu bar changes to the one first visible when the project was opened. Refer Figure 5.2.

Go to the menu bar option “Scripts” and click on it. Four menu options, which are basically, the four scripts for this application will be revealed. **These scripts are to be used in the order in which they appear, from top down.**

Click on the first script (first menu option) titled “Estimate HEC-1 Parameters”. This will initiate the script behind the option, which will estimate the area weighted curve number for each subbasin and store them in a table. After the script has finished executing, there will be a message box output to the screen asking the user to proceed to the next script.

Now click on the next menu option titled “Create HEC-1 Input File/Run HEC-1”. The script behind this option will then open an input file “Icaria.dat” and format the input parameters for the HEC-1 input file and place them in proper columns for HEC-1 to process the file without any error messages. It will also make a system call to HEC-1, so that HEC-1 processes the file “Icaria.dat” and produces the output file “hec1Icaria.out” in the same working directory. At the end of this scripts’ execution, there will be a message output to the screen, directing the user to proceed to the next menu option titled “Find the Coordinate Data”.

Now click on the menu option titled “Find the Coordinate Data”. This option runs a script that searches through the output file “hec1Icariaout.dat”, until it finds the hydrograph coordinate data for the outlet discharge hydrograph, extracts that data and stores it in a comma delimited file called “final.txt” in the same working directory.

Now, to view the output discharge hydrograph at the outlet to the basin, click on the option titled “Plot the Output Hydrograph”. This option executes a script that creates

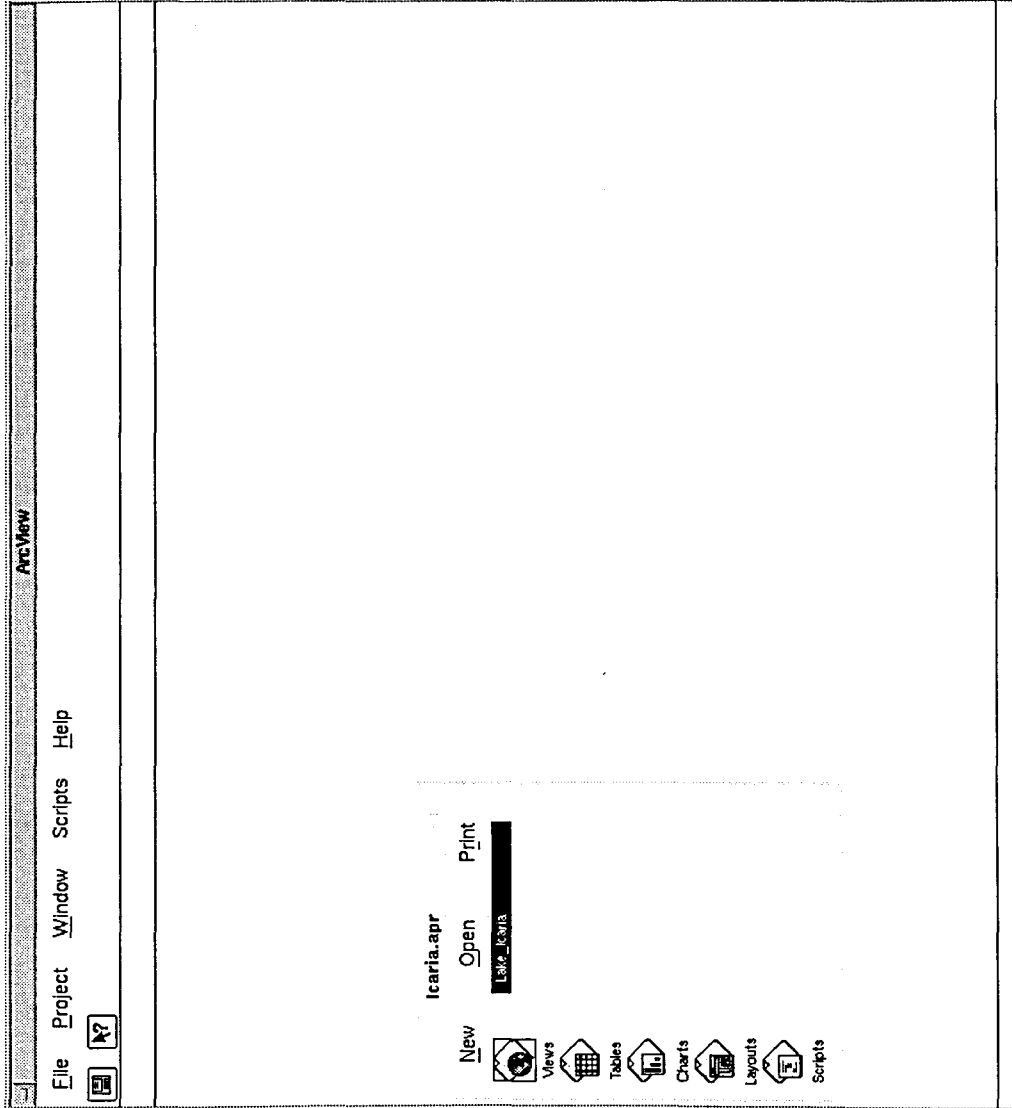


Figure 5.1 Screen Capture of The First Arcview Screen

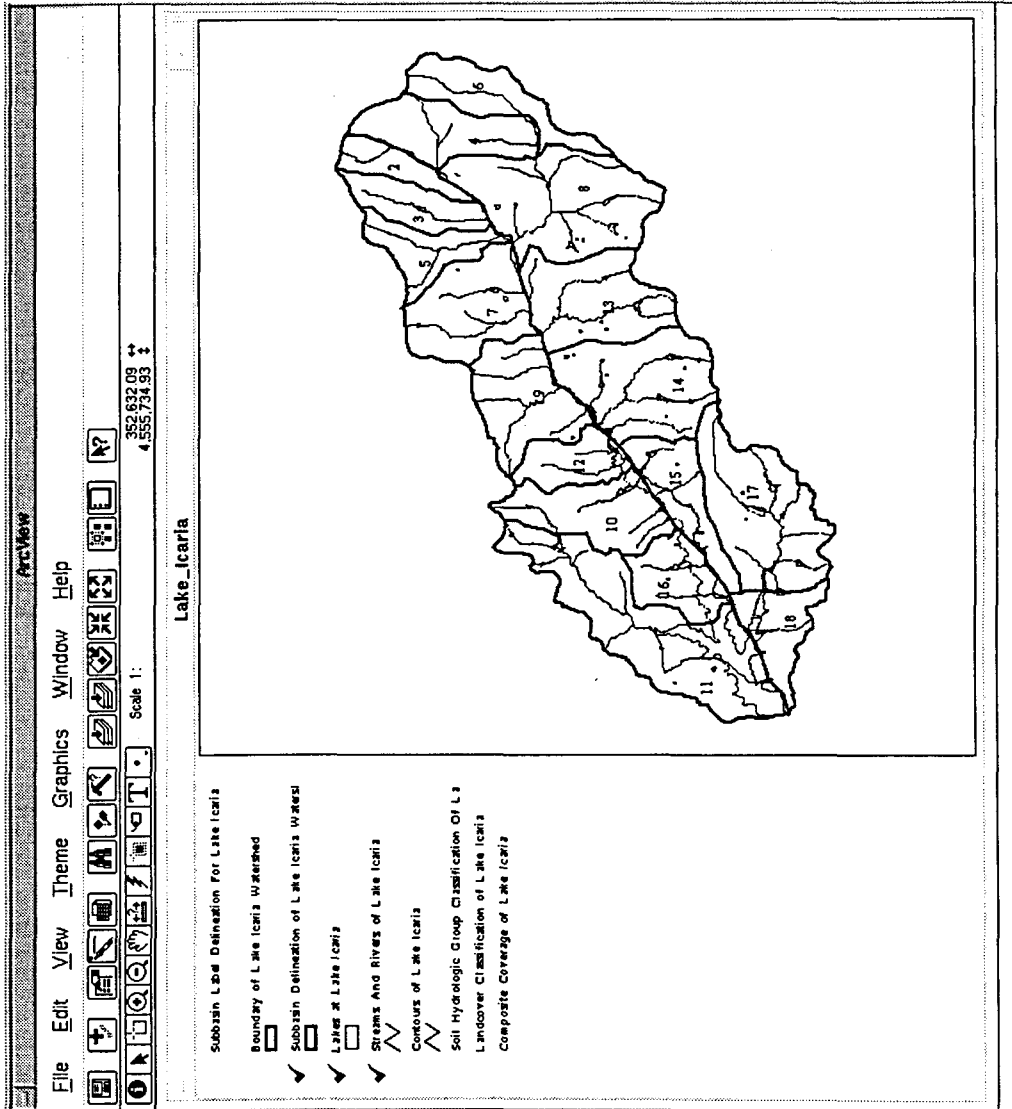


Figure 5.2 Screen Capture of The Second Arcview Screen

a chart of the hydrograph coordinate data and plots that chart as a line graph. The result is a discharge hydrograph at the outlet, as shown in Figure 5.3, that shows the time base in minutes, on the X axis and the discharge in cubic feet per second (cfs), on the Y axis. This output hydrograph can be printed, by clicking on the “Print” option under the menu bar option “File”.

Finally, to close the project, go to the project box and click on the top bar, thus changing the menu bar to the initial one. Choose “File” menu bar option and the option “Close Project”. The user will be prompted to save the project. Click on the “Yes” or “No” option and then click on the “Exit” option of the “File” menu bar option. This will cause Arcview to exit and the user is returned to the system prompt, in the directory from which he started Arcview.

5.1 A Brief Discussion Of Each Menu Option In Icaria.apr

The menu bar option “File” has the following menu options.

New Project: This option creates a new project. A project as explained earlier, is what Arcview uses, to hold all the components created by the user, e.g. views, themes, tables, scripts, layouts, charts, etc.

Open Project: This option opens an existing project that has already been created.

Close Project: This option allows the user to close the project and before closing the project, will prompt the user on whether he or she wants to save the project or not.

Save Project: This option allows the user to save the project without closing it.

Save Project As: This option allows the user to save a project under a particular name or assign a new name to an existing project. In the case of duplication, the old project is not overwritten.

Exit: This option allows the user to exit Arcview. Again, it prompts the user to save the project before exiting, if the user has not saved the project, before exiting, as

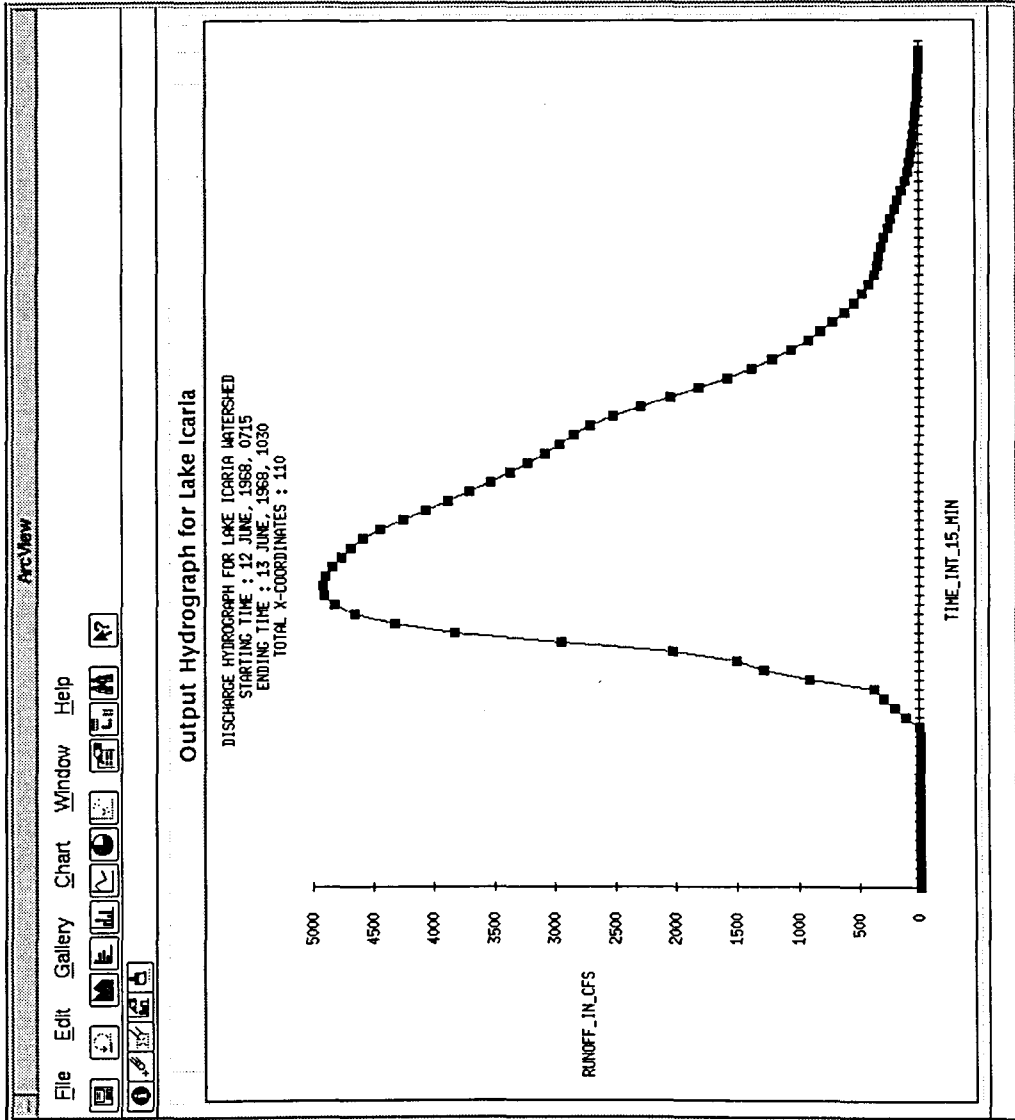


Figure 5.3 Screen Capture of The Discharge Hydrograph at outlet

a security measure.

The menu bar option “**Project**” has the following menu options.

Properties: This option allows the user to specify the properties of an Arcview project, more specifically, the startup script and the shutdown scripts associated with the project. As the names indicate, the startup and the shutdown scripts are executed when the project starts up and when it is closed respectively.

Customize: This option allows the user to customize the project by adding tool bars, menus or buttons and associating a script to run when each of these is activated.

Rename: This option allows the user to rename the active document in the project. A view, a table, a chart or a layout maybe examples of a document.

Delete: This option allows the user to delete the active document in the project.

Add Table: This option allows the user to add an external table (a comma delimited file, a D-Base file or a text file) to the active project.

Import: This option allows the user to import an external component into the active project. An external component may be another project by itself, or any of the project documents described above.

SQL Connect: This option allows the user connect to an external database that holds attribute data. This external database could be Oracle, INFO, or Microsoft Access.

The menu bar option “**Windows**” has the following options.

Tile: This options arranges the various open document windows in a tile inside the project window.

Cascade: This option arranges the document windows by cascading one on top of the other.

Arrange Icons: This options allows the user to arrange the various icons of the various project components, once the components have iconified.

Show Symbol Palette: This option opens up the symbol palette for the user to manipulate the text size, the pattern type, the pattern type, the thickness of lines, etc.

The menu bar option “**Scripts**” has the following options.

Estimate The HEC-1 Parameters: This script aggregates the area weighted curve numbers over an entire subbasin and enters the final curve number value in a table.

Create HEC-1 Input File/Run HEC-1: This script opens up an input file called “hec1Icaria.dat” in the current working directory. This script also formats the input parameters for HEC-1 so that HEC-1 can process the file. It then makes a system call to HEC-1 to process the input file, once its complete and creates an output file called “hec1Icaria.out” that holds the results of the HEC-1 processing.

Find The Coordinate Data: This script searches through the output file until it finds the output hydrograph coordinate data and extracts only that data into a separate file called “final.txt”.

Plot The Output Hydrograph: This option allows the user to create a chart from the coordinate data obtained from the file “final.txt”. This chart plots the coordinates as a line graph and the result is the discharge hydrograph at the outlet of the watershed as shown in Figure 5.3.

This concludes the explanation of the menu options under each menu bar item in the Arcview project “Icaria.apr”.

6 EXAMPLE APPLICATION AND RESULTS

6.1 Description of Example Application Area

This part of the chapter discusses the user interface to the system developed in Arcview. The interactive hydrologic modeling system was applied to the Lake Icaria watershed located in Adams County, Iowa, approximately 70 miles (112 Km) southwest of Des Moines. The watershed, which contributes flow to Lake Icaria, has an area of 17,482 acres (7075 ha). Lake Icaria, the major source of rural drinking water supply for the region, has an area of 700 acres (283 ha) with a maximum flood depth of 46 feet. It is part of the 1878-acre (760-ha) lake Icaria Recreational Area, providing facilities for boating, fishing, swimming, and camping. In addition, Lake Icaria provides water for domestic and industrial use within Corning and surrounding towns.

Soils in Lake Icaria watershed were prairie derived, developed primarily from loess pre-Wisconsin till or pre-Wisconsin till-derived paleosols and include Adair, Clarinda, Clearfield, Macksburg, Sharpsburg, and Shelby. These soils belong to the Sharpsburg-Adair (nearly level to moderately steep), Mackburg-Winterset (nearly level to gently sloping), and Shelby-Sharpsburg (moderately sloping to steep) associations. Average annual rainfall in the watershed is 33.1 inches with the greatest amount of 5.5 inches in June. The average frost free growing season is 160 days. These favorable climatic factors coupled with favorable physical and chemical characteristics of the soils make this a productive agricultural area.

Agricultural production in Lake Icaria watershed consists of row crops integrated with

livestock (hog, beef, poultry, sheep) production. Cropland and pasture land comprise 49 percent and 22.4 percent of the watershed area respectively, while 11.6 percent of the watershed is under cropland reserve program. About 4.6 percent of the watershed area is identified as idle land, which includes irregular-shaped tracts of land and parts of croplands that are either non-farmable or unsuitable for pasture. The remaining 12.5 percent of the Lake Icaria watershed includes water, farmsteads, roads and parkland.

6.2 Results

In this research, the link-up between HEC-1 and Arc/Info through Avenue was accomplished. In general, Arc/Info performed reasonably well in computing these needed spatially derived parameters. Since Arc/Info version 7.0.2 was used for the project, the computation of the centroids for each subbasin was easy. This made it easy to compute some of the length parameters that went as input to the HEC-1 input file. The three key coverages that contributed towards the computation and estimation of most of the input parameters for the HEC-1 input file were the streams coverage, the soil coverage and the landuse coverage. The streams coverage was digitized from the 1:24,000 scale USGS topographic maps, while the soils coverage was formed by scanning the soils maps provided by the Soil Conservation Service for Adams County, Iowa and the landuse coverages were formed by processing raw satellite data using the image processing software ERDAS. Using these three coverages and the attribute values stored within each one of the coverages the parameters for the HEC-1 input file were estimated or computed and input at the appropriate stage in the input file. Given below is a table of the parameters for each subbasin that went as input to the HEC-1 input file. The parameters for Snyder's synthetic unit hydrograph for the existing condition are $C_p = 0.25$ and $C_t = 0.38$. The values of L and L_{ca} , that were used for each subbasin for computing the Snyder unit hydrographs parameters t_p and C_p , are given in the table below. The two required

parameters, basin lag t_p and peak discharge per unit drainage area, Q_p , for Snyder's unit hydrograph were computed using the following equations, $t_p = C_1 C_t (LL_{ca})^{0.3}$ and $Q_p = C_2 C_p / t_p$.

The value of K (Muskingum coefficient) and X (a number showing the ratio between inflow and outflow) were 0.6 and 0.2 respectively. The following table shows the parameters that were input into the HEC-1 input file for each subbasin.

Table 6.1 Input Parameters For HEC-1 Input File

Subbasin Number	Area of the subbasin A (mi^2)	Length to the centroid L (mi)	Watershed length L_{ca} (mi)	SCS Curve number for the subbasin, CN
2	0.59	1.16	1.93	81
3	0.89	1.11	2.21	81
4	1.48	0.57	1.43	79
5	0.91	1.13	1.95	82
6	1.20	1.16	2.23	81
7	1.80	0.97	1.81	81
8	2.95	1.06	2.54	81
9	0.17	0.98	2.20	76
10	1.54	1.12	2.49	79
11	3.45	2.12	4.99	83
12	0.83	0.84	1.70	77
13	1.95	0.92	2.12	81
14	2.35	0.89	2.06	76
15	1.15	1.42	2.22	83
16	1.04	0.77	1.59	86
17	2.59	1.07	2.98	82
18	0.95	1.15	2.03	80

7 SUMMARY AND CONCLUSIONS

In this research, the Arc/Info GIS software package was utilized to generate needed spatially related parameters for HEC-1 execution. The above given method is one viable method of database accumulation and manipulation to achieve the purpose for linking a Geographic Information System with a Hydrologic Modeling System. In general, Arc/Info performed many tedious tasks and labor intensive determinations (calculating basin areas, average basin curve numbers, etc.) quite effectively. But the extensive costs of both equipment and software has to be considered prior to embarking upon a GIS to augment routine hydrologic models. While some of the products (color graphs, graphics, etc.) will look impressive, the real product, the unit hydrograph will likely be no more accurate than that obtained by more traditional means. The results are only as good as the accuracy of the data and in most cases the resolution of the data may not be high enough to draw conclusions from it. This limitation cannot be overcome by complex software, because it is inherent in the data or the database itself. But accuracy aside, the use of GIS for large projects requiring data from various sources may be quite justifiable. In addition, a GIS is perfectly suited for continual updating and rapid recomputation of alternate scenarios, a key factor to be considered in many water resources projects. Finally, one can question the use of spatially rich datasets and GIS analyses to drive a hydrologic modeling approach based upon spatially aggregated parameters, like average basin curve numbers. It is analogous to driving a nail with a sledgehammer, or perhaps performing word processing on a supercomputer. Traditional hydrologic modeling approaches are likely to stay with us for some time. However, the

growing presence of detailed spatial information and associated analytical tools will just as assuredly and effectively foster the continued development and implementation of more complex hydrologic models.

8 RECOMMENDATIONS FOR FUTURE RESEARCH

With the availability of sophisticated and efficient GIS software, the trend towards merging hydrologic models and integrating them with a geographic information system and thereby, capturing and utilizing the finer points of both the systems, is currently in vogue. Integration of groundwater models, pollution monitoring, surface water models with GIS softwares like Arc/Info, GRASS, etc. has been done and will continue to be done. But any conclusions drawn from such work can only be as good as the accuracy of the datasets used. So it is recommended that careful attention be paid to the spatial data accuracy of the datasets being used. Also it helps if the datasets are reasonably current. Some data like topography and soil may not change rapidly. But other data sources like the landuse data for example, might change rapidly. Thus, for landcover data, it is recommended that one use coverages derived from satellite data or from aerial photographs. Also the process of digitizing any coverage must be eliminated. This is because digitizing being a manual process is likely to create more errors. As far as possible, aerial photographs or satellite data must be used for deriving any coverage. But sometimes, this may not be possible, due to many reasons, like lack of time and money, lack of expertise, etc. In the attribute data that was entered into the HEC-1 input file, we can improve the precipitation data that is input as part of the HEC-1 input file. Using the National Weather Service (NWS) NEXRAD radar data would give more realistic values of precipitation. The National Weather Service has Doppler radar data converted into precipitation values. This data, may or may not be accurately representative of the watershed of interest, depending upon the distance of the watershed

from the NWS weather monitoring station. But since these are real data in both time and space, they could be used with reasonably good results. A better option would be to establish a link with the nearest NWS station, so that various scenarios of precipitation and runoff or flooding can be visualized in real time.

BIBLIOGRAPHY

- Bras, R. L. (1990). *Hydrology: An Introduction to Hydrologic Science*. Addison-Wesley Publishing: New York, N.Y.
- Chow, Van Te , Maidment, David R. and Mays, Larry W. (1988). *Applied Hydrology*. McGraw-Hill International Editions, Civil Engineering Series: Singapore.
- Clarke, K. C. (1986). *Advances in Geographic Information Systems. Computers, Environment, and Urban Systems, 10*, 175–184.
- Cline, T. J., Molinas, A., and Julien, P. Y. (1989). An AutoCAD Based Watershed Information System For The Hydrologic Model HEC-1. *Water Resources Bulletin, 25(3)(12)*, 641–652.
- ERDAS, Inc. (1991). *Erdas Field Guide*. ERDAS: Atlanta, Georgia.
- ESRI, Inc. (1994). *Understanding GIS: The Arc/Info Method*. Environmental Systems Research Institute, Inc.: Redlands, California.
- ESRI Inc. *Arc/Info Users Guide, Version 7.0.2* (1995). *ESRI, Inc.*: Redlands, California.
- Hydrologic Engineering Center. (1981). *HEC-1 Flood Hydrograph Package User's Manual*. Hydrologic Engineering Center, U.S. Army Corps of Engineers: Davis, California.
- Jensen, John R. 1986. *Introductory Digital Image Processing: A Remote Sensing Perspective*. Prentice Hall: Upper Saddle River, N.J.
- Lillesand, T. M., and Kiefer, R. W. (1987). *Remote Sensing and Image Interpretation*. John Wiley & Sons: New York, N.Y.
- Mark A. and Tara, Patrick D. (1993). Integrated Hydrologic Modeling With Geographic Information Systems. *Journal of Water Resources Planning And Management, 119(2)(11)*, 129–139.

Peuquet, Donna J. and Marble, Duane F. (1980). *Introductory Readings in Geographic Information Systems*. Taylor and Francis Ltd: New York, N.Y.

Warwick, J. J. and Hanes, S. J. An Insight Into GIS Application to Hydrologic Modeling. *Journal of Water Resources Management* 120(16), 366–381.

APPENDIX

Avenue Script To Estimate The HEC-1 Parameters

```

myProject = av.GetProject
myView = myProject.FindDoc("Lake_Icaria")
'If the view "testview" doesn't exist, exit from the program..
if (nil = myView) then
    MsgBox.Error ("The view Lake Icaria does not exist....", "Warning")
    MsgBox.Info ("Create the View Lake Icaria before
                running this module", "" '')
    exit
'if the view is found..
elseif (not (nil = myView)) then
    myThemesList = myView.GetThemes
    For each aTheme in myThemesList
        if (aTheme.GetName = "Composite Coverage of Lake Icaria") then
            myFTab = aTheme.GetFTab
            myFileName = FileName.Make ("/home/gis4/nara/info/wsheds.sta")
            myVTabSta = VTab.MakeNew (myFileName, INFO)
            myVTabSta.SetEditable(True)
            Field1 = Field.Make ("wsheds#", #Field_long, 2, 0)
            Field2 = Field.Make ("area", #Field_long, 16, 6)

```

```

Field3 = Field.Make ("areawcn", #Field_long, 2, 0)
myFieldsList = {Field1, Field2, Field3}
myVTabSta.AddFields (myFieldsList)
myVarString = ""
myResultString = ""
For each i in 1..18
    myExpression = "[wsheds#]= " ++ i.AsString
    mySelection = myFTab.GetSelection
    myFTab.Query (myExpression, mySelection, #VTAB_SELTYPE_NEW)
    myFTab.UpdateSelection
    valwshednum = 0
    valarea = 0.0
    valcn = 0
    valareawcn = 0.0
    totalnum = 0.0
    totalden = 0.0
    For each rec in mySelection
        valwshednum = myFTab.ReturnValue
            (myFTab.FindField("wsheds#") , rec)
        valarea = myFTab.ReturnValue
            (myFTab.FindField ("area"), rec)
        valcn = myFTab.ReturnValue
            (myFTab.FindField ("cn_comb"), rec)
        totalnum = totalnum + (valarea * valcn)
        totalden = totalden + valarea
        valareawcn = totalnum / totalden
        valareawcn = valareawcn.SetFormat("d")

```

```
        valarea = valarea.SetFormat("d.dd")
    end 'For each rec
    myResultString = valareaawcn.AsString ++ "is the area
        weighted  cn for wshed#" ++ i.AsString
        ++ "with an area of" ++ valarea.AsString
        ++ NL
    myVarString = myVarString + myResultString
    recnum = myVTabSta.AddRecord
    myVTabSta.SetValue (Field1, recnum, valwshednum)
    myVTabSta.SetValue (Field2, recnum, valarea)
    myVTabSta.SetValue (Field3, recnum, valareaawcn)
    myFTab.GetSelection.ClearAll
    myFTab.UpdateSelection
end 'For each i in....
    MsgBox.Report (myVarString, "Report")
end 'If (aTheme.GetName.....
end 'For each aTheme....
else
end
```


Avenue Script to Create The HEC-1 Input File and Run the HEC-1 Program

```

myInputFileName = FileName.Make ("/local/mnt/gis4/nara/heclinpIcaria")
av.ShowMsg ("Creating the HEC-1 Input File.....")
if (File.Exists (myInputFileName)) then
    File.Delete (myInputFileName)
    MsgBox.info ("The File" ++ myInputFileName.AsString ++
                "was found and deleted" ++ NL, "Info")
    myLineFile = LineFile.Make(myInputFileName, #FILE_PERM_WRITE)
else
    myLineFile = LineFile.Make(myInputFileName, #FILE_PERM_WRITE)
    MsgBox.Info ("The File" ++ myInputFileName.AsString ++ "was created"
                ++ NL, "Info")
end

myParamFileName = FileName.Make ("/home/gis4/nara/info/wsheds.sta")
myVTab = VTab.Make (myParamFileName, False, False)
myVTab.SetEditable (False)
Str1 = "ID" + av.Run( "MakeBuffer", {1}) + "LAKE ICARIA WATERSHED"
myLineFile.WriteElt (Str1)
Str2 = "ID" + av.Run ("MakeBuffer", {1}) + "CONSIDER EXISTING
                                           CONDITIONS, DEVELOPED
                                           CONDITIONS AND"
myLineFile.WriteElt (Str2)
Str3 = "ID" + av.Run ("MakeBuffer", {1}) + "DEVELOPED CONDITONS
                                           WITH IMPROVEMENTS"
myLineFile.WriteElt (Str3)

```

```

Str4 = "IT" + av.Run ("MakeBuffer", {4}) + "15" + av.Run ("MakeBuffer",{1})
      + "12JUN68" + av.Run ("MakeBuffer", {5}) + "715"
      + av.Run ("MakeBuffer", {5}) + "110"

myLineFile.WriteElt (Str4)

Str9 = "KK" + av.Run ("MakeBuffer", {2}) + "SUB4"

myLineFile.WriteElt (Str9)

Str10 = "KM" + av.Run ("MakeBuffer", {1}) + "RUNOFF COMPUTATIONS FOR
        SUBCATCHMENT 4"

myLineFile.WriteElt (Str10)

boundary = (70758701 / ( 1609 * 1609)).SetFormat ("d.dd")

Str11 = "PH" + av.Run ("MakeBuffer", {5}) + "1" + av.Run ("MakeBuffer",{3})
      + boundary.AsString + av.Run ("MakeBuffer", {4}) + "0.40"
      + av.Run ("MakeBuffer", {4}) + "0.65" + av.Run ("MakeBuffer",{4})
      + "1.00" + av.Run ("MakeBuffer", {4}) + "1.50"
      + av.Run ("MakeBuffer", {4}) + "1.90" + av.Run ("MakeBuffer",{4})
      + "2.90" + av.Run ("MakeBuffer", {4}) + "3.90"

myLineFile.WriteElt (Str11)

myExpression = "[wsheds#]= 4"

mySelection = myVTab.GetSelection

myVTab.Query (myExpression, mySelection, #VTAB_SELTYPE_NEW)

For each rec in mySelection

    wshedsnum = myVTab.ReturnValue (myVTab.FindField ("wsheds#"), rec)
    areawshed = (3827347.31405 / ( 1609 * 1609)).SetFormat("d.dd")
    areawshed = (myVTab.ReturnValue (myVTab.FindField ("area"), rec)
                / 1609).SetFormat("d.dd")

    wcn = (myVTab.ReturnValue (myVTab.FindField ("areawcn"), rec))
    wcn = wcn.SetFormat("d")

```

```

if (wshedsnum = 4) then
    Str12 = "BA" + av.Run ("MakeBuffer", {2}) + areawshed.AsString
    myLineFile.WriteElt (Str12)
else
    MsgBox.Info ("The wsheds# was not 4.. sorry:~)" ++ NL, "Info")
end

end

myVTab.GetSelection.ClearAll
myVTab.UpdateSelection

Str13 = "LS" + av.Run ("MakeBuffer", {5}) + "0"
        + av.Run ("MakeBuffer",{6}) + wcn.AsString
myLineFile.WriteElt (Str13)

tp4 = (0.38 * (((2302.50 * 922.09) / (1609 * 1609)) ^ 0.3))
tp4 = tp4.SetFormat ("d.dd")

Str14 = "US" + av.Run ("MakeBuffer", {2}) + tp4.AsString
        + av.Run ("MakeBuffer", {4}) + "0.25"

myLineFile.WriteElt (Str14)

Str37 = "KK" + av.Run ("MakeBuffer", {2}) + "SUB2"
myLineFile.WriteElt (Str37)

Str38 = "KM" + av.Run ("MakeBuffer", {1}) + "RUNOFF COMPUTATIONS FOR
                                                SUBCATCHMENT 2"

myLineFile.WriteElt (Str38)

myExpression = "[wsheds#]= 2"
mySelection = myVTab.GetSelection
myVTab.Query (myExpression, mySelection, #VTAB_SELTYPE_NEW)

For each rec in mySelection
    wshedsnum = myVTab.ReturnValue (myVTab.FindField ("wsheds#"), rec)

```

```

areawshed = (1545921.92268 / ( 1609 * 1609 )).SetFormat ("d.dd")
'areawshed = (myVTab.ReturnValue (myVTab.FindField ("area"), rec)
              / 1609).SetFormat("d.dd")

wcn = (myVTab.returnValue (myVTab.FindField ("areawcn"), rec))
wcn = wcn.SetFormat("d")

if (wshedsnum = 2) then
    Str39 = "BA" + av.Run ("MakeBuffer", {2}) + areawshed.AsString
    myLineFile.WriteElt (Str39)
else
    MsgBox.Info ("The wshed# was not 2.. sorry:~)" ++ NL, "Info")
end

end

myVTab.GetSelection.ClearAll
myVTab.UpdateSelection

Str40 = "LS" + av.Run ("MakeBuffer", {5}) + "0"
      + av.Run ("MakeBuffer", {6}) + wcn.AsString
myLineFile.WriteElt (Str40)

tp2 = (0.38 * ((( 3103.80 / 1609 ) * ( 1869.61 / 1609 )) ^ 0.3))
tp2 = tp2.SetFormat("d.dd")

Str41 = "US" + av.Run ("MakeBuffer", {2}) + tp2.AsString
      + av.Run ("MakeBuffer", {4}) + "0.25"

myLineFile.WriteElt (Str41)

Str42 = "KK" + av.Run ("MakeBuffer", {2}) + "OUT1"

myLineFile.WriteElt (Str42)

Str43 = "KM" + av.Run ("MakeBuffer", {1}) + "COMBINE SUBCATCHMENT 2
                                             RUNOFF WITH SUBCATCHMENT 4
                                             RUNOFF"

```

```

myLineFile.WriteElt (Str43)
Str44 = "HC" + av.Run ("MakeBuffer", {5}) + "2"
myLineFile.WriteElt (Str44)
Str45 = "KK" + av.Run ("MakeBuffer", {2}) + "SUB6"
myLineFile.WriteElt (Str45)
Str46 = "KM" + av.Run ("MakeBuffer", {1}) + "RUNOFF COMPUTATIONS
        FOR SUBCATCHMENT 6"
myLineFile.WriteElt (Str46)
myExpression = "[wsheds#]= 6"
mySelection = myVTab.GetSelection
myVTab.Query (myExpression, mySelection, #VTAB_SELTYPE_NEW)
For each rec in mySelection
    wshedsnum = myVTab.ReturnValue (myVTab.FindField ("wsheds#"), rec)
    areawshed = ( 3103043.80977 / ( 1609 * 1609 )).SetFormat ("d.dd")
    wcn = (myVTab.returnValue (myVTab.FindField ("areawcn"), rec))
    wcn = wcn.SetFormat("d")
    if (wshedsnum = 6) then
        Str47 = "BA" + av.Run ("MakeBuffer", {2}) + areawshed.AsString
        myLineFile.WriteElt (Str47)
    else
        MsgBox.Info ("The wshed# was not 6.. sorry:~)" ++ NL, "Info")
    end
end
myVTab.GetSelection.ClearAll
myVTab.UpdateSelection
Str48 = "LS" + av.Run ("MakeBuffer", {5}) + "0"
        + av.Run ("MakeBuffer", {6}) + wcn.AsString

```

```

myLineFile.WriteElt (Str48)

tp6 = (0.38 * ((( 3599.21 / 1609 ) * ( 1858.71 / 1609 )) ^ 0.3))

tp6 = tp6.SetFormat("d.dd")

Str49 = "US" + av.Run ("MakeBuffer", {2}) + tp6.AsString
      + av.Run ("MakeBuffer", {4}) + "0.25"

myLineFile.WriteElt (Str49)

Str50 = "KK" + av.Run ("MakeBuffer", {2}) + "SUB8"

myLineFile.WriteElt (Str50)

Str51 = "KM" + av.Run ("MakeBuffer", {1}) + "RUNOFF COMPUTATIONS FOR
      SUBCATCHMENT 8"

myLineFile.WriteElt (Str51)

myExpression = "[wsheds#]= 8"

mySelection = myVTab.GetSelection

myVTab.Query (myExpression, mySelection, #VTAB_SELTYPE_NEW)

For each rec in mySelection

    wshedsnum = myVTab.ReturnValue (myVTab.FindField ("wsheds#"), rec)
    areawshed = (7632702.19017 / (1609 * 1609)).SetFormat ("d.dd")
    wcn = (myVTab.returnValue (myVTab.FindField ("areawcn"), rec))
    wcn = wcn.SetFormat("d")

    if (wshedsnum = 8) then

        Str52 = "BA" + av.Run ("MakeBuffer", {2}) + areawshed.AsString
        myLineFile.WriteElt (Str52)

    else

        MsgBox.Info ("The wshed# was not 8.. sorry:):)" ++ NL, "Info")

    end

end

myVTab.GetSelection.ClearAll

```

```

myVTab.UpdateSelection
Str53 = "LS" + av.Run ("MakeBuffer", {5}) + "0"
      + av.Run ("MakeBuffer", {6}) + wcn.AsString
myLineFile.WriteElt (Str53)
tp8 = (0.38 * ((( 4093.67 / 1609 ) * ( 1709.51 / 1609 )) ^ 0.3))
tp8 = tp8.SetFormat("d.dd")
Str54 = "US" + av.Run ("MakeBuffer", {2}) + tp8.AsString
      + av.Run ("MakeBuffer", {4}) + "0.25"
myLineFile.WriteElt (Str54)
Str55 = "KK" + av.Run ("MakeBuffer", {2}) + "SUB3"
myLineFile.WriteElt (Str55)
Str56 = "KM" + av.Run ("MakeBuffer", {1}) + "RUNOFF COMPUTATIONS FOR
      SUBCATCHMENT 3"
myLineFile.WriteElt (Str56)
myExpression = "[wsheds#]= 3"
mySelection = myVTab.GetSelection
myVTab.Query (myExpression, mySelection, #VTAB_SELTYPE_NEW)
For each rec in mySelection
  wshedsnum = myVTab.ReturnValue (myVTab.FindField ("wsheds#"), rec)
  areawshed = (2297961.45034 / (1609 * 1609)).SetFormat ("d.dd")
  wcn = (myVTab.returnValue (myVTab.FindField ("areawcn"), rec))
  wcn = wcn.SetFormat("d")
  if (wshedsnum = 3) then
    Str57 = "BA" + av.Run ("MakeBuffer", {2}) + areawshed.AsString
    myLineFile.WriteElt (Str57)
  else
    MsgBox.Info ("The wsheds# was not 3.. sorry:):)" ++ NL, "Info")

```

```

end

end

myVTab.GetSelection.ClearAll

myVTab.UpdateSelection

Str58 = "LS" + av.Run ("MakeBuffer", {5}) + "0"
      + av.Run ("MakeBuffer", {6}) + wcn.AsString

myLineFile.WriteElt (Str58)

tp3 = 0.50

Str59 = "US" + av.Run ("MakeBuffer", {2}) + tp3.AsString
      + av.Run ("MakeBuffer", {4}) + "0.25"

myLineFile.WriteElt (Str59)

Str60 = "KK" + av.Run ("MakeBuffer", {2}) + "SUB5"

myLineFile.WriteElt (Str60)

Str61 = "KM" + av.Run ("MakeBuffer", {1}) + "RUNOFF COMPUTATIONS FOR
                                             SUBCATCHMENT 5"

myLineFile.WriteElt (Str61)

myExpression = "[wsheds#]= 5"

mySelection = myVTab.GetSelection

myVTab.Query (myExpression, mySelection, #VTAB_SELTYPE_NEW)

For each rec in mySelection

    wshedsnum = myVTab.ReturnValue (myVTab.FindField ("wsheds#"), rec)
    areawshed = ( 2355004.78002 / (1609 * 1609)).SetFormat ("d.dd")
    wcn = (myVTab.returnValue (myVTab.FindField ("areawcn"), rec))
    wcn = wcn.SetFormat("d")

if (wshedsnum = 5) then

    Str62 = "BA" + av.Run ("MakeBuffer", {2}) + areawshed.AsString
    myLineFile.WriteElt (Str62)

```



```

else
    MsgBox.Info ("The wsheds# was not 5.. sorry:):)" ++ NL, "Info")
end

end

myVTab.GetSelection.ClearAll
myVTab.UpdateSelection

Str63 = "LS" + av.Run ("MakeBuffer", {5}) + "0"
    + av.Run ("MakeBuffer",{6}) + wcn.AsString
myLineFile.WriteElt (Str63)

tp5 = (0.38 * ((( 3132.52 / 1609 ) * ( 1822.28 / 1609 )) ^ 0.3))
tp5 = tp5.SetFormat("d.dd")

Str64 = "US" + av.Run ("MakeBuffer", {2}) + tp5.AsString
    + av.Run ("MakeBuffer", {4}) + "0.25"
myLineFile.WriteElt (Str64)

Str65 = "KK" + av.Run ("MakeBuffer", {2}) + "OUT2"
myLineFile.WriteElt (Str65)

Str66 = "KM" + av.Run ("MakeBuffer", {1}) + "COMBINE SUB3,5,6,8 RUNOFF
AND OUT1 RUNOFF AT SUB4"
myLineFile.WriteElt (Str66)

Str67 = "HC" + av.Run ("MakeBuffer", {5}) + "5"
myLineFile.WriteElt (Str67)

Str68 = "KK" + av.Run ("MakeBuffer", {2}) + "SUB7"
myLineFile.WriteElt (Str68)

Str69 = "KM" + av.Run ("MakeBuffer", {1}) + "RUNOFF COMPUTATIONS FOR
SUBCATCHMENT 7"

myLineFile.WriteElt (Str69)

myExpression = "[wsheds#]= 7"

```

```

mySelection = myVTab.GetSelection
myVTab.Query (myExpression, mySelection, #VTAB_SELTYPE_NEW)
For each rec in mySelection
    wshedsnum = myVTab.ReturnValue (myVTab.FindField ("wsheds#"), rec)
    areawshed = (4660945.96333 / (1609 * 1609)).SetFormat("d.dd")
    wcn = (myVTab.returnValue (myVTab.FindField ("areawcn"), rec))
    wcn = wcn.SetFormat("d")
    if (wshedsnum = 7) then
        Str70 = "BA" + av.Run ("MakeBuffer", {2}) + areawshed.AsString
        myLineFile.WriteElt (Str70)
    else
        MsgBox.Info ("The wsheds# was not 7.. sorry:~)" ++ NL, "Info")
    end
end
end
myVTab.GetSelection.ClearAll
myVTab.UpdateSelection
Str70 = "LS" + av.Run ("MakeBuffer", {5}) + "0" + av.Run ("MakeBuffer",{6})
        + wcn.AsString
myLineFile.WriteElt (Str70)
tp7 = (0.38 * ((( 2919.17 / 1609 ) * ( 1564.61 / 1609 )) ^ 0.3))
tp7 = tp7.SetFormat("d.dd")
Str71 = "US" + av.Run ("MakeBuffer", {2}) + tp7.AsString
        + av.Run ("MakeBuffer", {4}) + "0.25"
myLineFile.WriteElt (Str71)
Str72 = "KK" + av.Run ("MakeBuffer", {2}) + "SUB9"
myLineFile.WriteElt (Str72)
Str73 = "KM" + av.Run ("MakeBuffer", {1}) + "RUNOFF COMPUTATIONS FOR

```

SUBCATCHMENT 9"

```

myLineFile.WriteElt (Str73)
myExpression = "[wsheds#]= 9"
mySelection = myVTab.GetSelection
myVTab.Query (myExpression, mySelection, #VTAB_SELTYPE_NEW)
For each rec in mySelection
    wshedsnum = myVTab.ReturnValue (myVTab.FindField ("wsheds#"), rec)
    areawshed = (4326258 / (1609 * 1609)).SetFormat("d.dd")
    wcn = (myVTab.returnValue (myVTab.FindField ("areawcn"), rec))
    wcn = wcn.SetFormat("d")
    if (wshedsnum = 9) then
        Str74 = "BA" + av.Run ("MakeBuffer", {2}) + areawshed.AsString
        myLineFile.WriteElt (Str74)
    else
        MsgBox.Info ("The wsheds# was not 9.. sorry:):)" ++ NL, "Info")
    end
end
end
myVTab.GetSelection.ClearAll
myVTab.UpdateSelection
Str75 = "LS" + av.Run ("MakeBuffer", {5}) + "0"
        + av.Run ("MakeBuffer", {6}) + wcn.AsString
myLineFile.WriteElt (Str75)
tp9 = (0.38 * ((( 3537.35 / 1609 ) * ( 1571.57 / 1609 )) ^ 0.3))
tp9 = tp9.SetFormat("d.dd")
Str76 = "US" + av.Run ("MakeBuffer", {2}) + tp9.AsString
        + av.Run ("MakeBuffer", {4}) + "0.25"
myLineFile.WriteElt (Str64)

```

```

Str77 = "KK" + av.Run ("MakeBuffer", {1}) + "SUB13"
myLineFile.WriteElt (Str77)
Str78 = "KM" + av.Run ("MakeBuffer", {1}) + "RUNOFF COMPUTATIONS FOR
                                         SUBCATCHMENT 13"
myLineFile.WriteElt (Str78)
myExpression = "[wsheds#]= 13"
mySelection = myVTab.GetSelection
myVTab.Query (myExpression, mySelection, #VTAB_SELTYPE_NEW)
For each rec in mySelection
    wshedsnum = myVTab.ReturnValue (myVTab.FindField ("wsheds#"), rec)
    areawshed = (5054471 / (1609 * 1609)).SetFormat("d.dd")
    wcn = (myVTab.returnValue (myVTab.FindField ("areawcn"), rec))
    wcn = wcn.SetFormat("d")
    if (wshedsnum = 13) then
        Str79 = "BA" + av.Run ("MakeBuffer", {2}) + areawshed.AsString
        myLineFile.WriteElt (Str79)
    else
        MsgBox.Info ("The wsheds# was not 13.. sorry:):)" ++ NL, "Info")
    end
end
myVTab.GetSelection.ClearAll
myVTab.UpdateSelection
Str79 = "LS" + av.Run ("MakeBuffer", {5}) + "0"
        + av.Run ("MakeBuffer", {6}) + wcn.AsString
myLineFile.WriteElt (Str79)
tp13 = (0.38 * ((( 3405.38 / 1609 ) * ( 1479.40 / 1609 )) ^ 0.3))
tp13 = tp13.SetFormat("d.dd")

```

```

Str80 = "US" + av.Run ("MakeBuffer", {2}) + tp13.AsString
      + av.Run ("MakeBuffer", {4}) + "0.25"
myLineFile.WriteElt (Str80)
Str81 = "KK" + av.Run ("MakeBuffer", {1}) + "SUB14"
myLineFile.WriteElt (Str81)
Str82 = "KM" + av.Run ("MakeBuffer", {1}) + "RUNOFF COMPUTATIONS FOR
      SUBCATCHMENT 14"
myLineFile.WriteElt (Str82)
myExpression = "[wsheds#]= 14"
mySelection = myVTab.GetSelection
myVTab.Query (myExpression, mySelection, #VTAB_SELTYPE_NEW)
For each rec in mySelection
  wshedsnum = myVTab.ReturnValue (myVTab.FindField ("wsheds#"), rec)
  areawshed = (6074800 / (1609 * 1609)).SetFormat("d.dd")
  wcn = (myVTab.returnValue (myVTab.FindField ("areawcn"), rec))
  wcn = wcn.SetFormat("d")
  if (wshedsnum = 14) then
    Str83 = "BA" + av.Run ("MakeBuffer", {2}) + areawshed.AsString
    myLineFile.WriteElt (Str83)
  else
    MsgBox.Info ("The wsheds# was not 14.. sorry:):)" ++ NL, "Info")
  end
end
end
myVTab.GetSelection.ClearAll
myVTab.UpdateSelection
Str84 = "LS" + av.Run ("MakeBuffer", {5}) + "0"
      + av.Run ("MakeBuffer", {6}) + wcn.AsString

```

```

myLineFile.WriteElt (Str84)

tp14 = (0.38 * ((( 3405.38 / 1609 ) * ( 1479.40 / 1609 )) ^ 0.3))

tp14 = tp14.SetFormat("d.dd")

Str85 = "US" + av.Run ("MakeBuffer", {2}) + tp14.AsString
      + av.Run ("MakeBuffer", {4}) + "0.25"

myLineFile.WriteElt (Str80)

Str86 = "KK" + av.Run ("MakeBuffer", {2}) + "OUT3"

myLineFile.WriteElt (Str86)

Str87 = "KM" + av.Run ("MakeBuffer", {1}) + "COMBINE SUB7,9,13,14
      RUNOFF AND ROUTE2"

myLineFile.WriteElt (Str87)

Str88 = "HC" + av.Run ("MakeBuffer", {5}) + "5"

myLineFile.WriteElt (Str88)

Str881 = "KK" + "ROUTE3"

myLineFile.WriteElt (Str881)

Str882 = "RM" + av.Run ("MakeBuffer", {5}) + "1"
      + av.Run ("MakeBuffer", {5}) + "0.6"
      + av.Run ("MakeBuffer", {5}) + "0.2"

myLineFile.WriteElt (Str882)

Str89 = "KK" + av.Run ("MakeBuffer", {1}) + "SUB10"

myLineFile.WriteElt (Str89)

Str90 = "KM" + av.Run ("MakeBuffer", {1}) + "RUNOFF COMPUTATIONS
      FOR SUBCATCHMENT 10"

myLineFile.WriteElt (Str90)

myExpression = "[wsheds#]= 10"

mySelection = myVTab.GetSelection

myVTab.Query (myExpression, mySelection, #VTAB_SELTYPE_NEW)

```

```

For each rec in mySelection
    wshedsnum = myVTab.ReturnValue (myVTab.FindField ("wsheds#"), rec)
    areawshed = (3979536 / (1609 * 1609)).SetFormat("d.dd")
    wcn = (myVTab.returnValue (myVTab.FindField ("areawcn"), rec))
    wcn = wcn.SetFormat("d")
    if (wshedsnum = 10) then
        Str91 = "BA" + av.Run ("MakeBuffer", {2}) + areawshed.AsString
        myLineFile.WriteElt (Str91)
    else
        MsgBox.Info ("The wsheds# was not 10.. sorry:~)" ++ NL, "Info")
    end
end

end

myVTab.GetSelection.ClearAll
myVTab.UpdateSelection
Str92 = "LS" + av.Run ("MakeBuffer", {5}) + "0"
        + av.Run ("MakeBuffer", {6}) + wcn.AsString
myLineFile.WriteElt (Str92)
tp10 = (0.38 * ((( 4000.88 / 1609 ) * ( 1798.23 / 1609 )) ^ 0.3))
tp10 = tp10.SetFormat("d.dd")
Str93 = "US" + av.Run ("MakeBuffer", {2}) + tp10.AsString
        + av.Run ("MakeBuffer", {4}) + "0.25"
myLineFile.WriteElt (Str93)
Str94 = "KK" + av.Run ("MakeBuffer", {1}) + "SUB12"
myLineFile.WriteElt (Str94)
Str95 = "KM" + av.Run ("MakeBuffer", {1}) + "RUNOFF COMPUTATIONS
                                                FOR SUBCATCHMENT 12"
myLineFile.WriteElt (Str95)

```

```

myExpression = "[wsheds#]= 12"
mySelection = myVTab.GetSelection
myVTab.Query (myExpression, mySelection, #VTAB_SELTYPE_NEW)
For each rec in mySelection
    wshedsnum = myVTab.ReturnValue (myVTab.FindField ("wsheds#"), rec)
    areawshed = (2151209 / (1609 * 1609)).SetFormat("d.dd")
    wcn = (myVTab.ReturnValue (myVTab.FindField ("areawcn"), rec))
    wcn = wcn.SetFormat("d")
    if (wshedsnum = 12) then
        Str96 = "BA" + av.Run ("MakeBuffer", {2}) + areawshed.AsString
        myLineFile.WriteElt (Str96)
    else
        MsgBox.Info ("The wsheds# was not 12.. sorry:~)" ++ NL, "Info")
    end
end
end
myVTab.GetSelection.ClearAll
myVTab.UpdateSelection
Str97 = "LS" + av.Run ("MakeBuffer", {5}) + "0"
        + av.Run ("MakeBuffer", {6}) + wcn.AsString
myLineFile.WriteElt (Str97)
tp12 = 0.51
Str98 = "US" + av.Run ("MakeBuffer", {2}) + tp12.AsString
        + av.Run ("MakeBuffer", {4}) + "0.25"
myLineFile.WriteElt (Str98)
Str99 = "KK" + av.Run ("MakeBuffer", {1}) + "SUB15"
myLineFile.WriteElt (Str99)
Str100 = "KM" + av.Run ("MakeBuffer", {1}) + "RUNOFF COMPUTATIONS

```


FOR SUBCATCHMENT 14"

```

myLineFile.WriteElt (Str100)
myExpression = "[wsheds#]= 15"
mySelection = myVTab.GetSelection
myVTab.Query (myExpression, mySelection, #VTAB_SELTYPE_NEW)
For each rec in mySelection
    wshedsnum = myVTab.ReturnValue (myVTab.FindField ("wsheds#"), rec)
    areawshed = (2980593 / (1609 * 1609)).SetFormat("d.dd")
    wcn = (myVTab.returnValue (myVTab.FindField ("areawcn"), rec))
    wcn = wcn.SetFormat("d")
    if (wshedsnum = 15) then
        Str101 = "BA" + av.Run ("MakeBuffer", {1}) + areawshed.AsString
        myLineFile.WriteElt (Str101)
    else
        MsgBox.Info ("The wsheds# was not 15.. sorry:~)" ++ NL, "Info")
    end
end
end
myVTab.GetSelection.ClearAll
myVTab.UpdateSelection
Str102 = "LS" + av.Run ("MakeBuffer", {5}) + "0"
        + av.Run ("MakeBuffer", {6}) + wcn.AsString
myLineFile.WriteElt (Str102)
tp15 = (0.38 * ((( 3316.90 / 1609 ) * ( 1430.91 / 1609 )) ^ 0.3))
tp15 = tp15.SetFormat("d.dd")
Str103 = "US" + av.Run ("MakeBuffer", {2}) + tp15.AsString
        + av.Run ("MakeBuffer", {4}) + "0.25"
myLineFile.WriteElt (Str103)

```

```

Str104 = "KK" + av.Run ("MakeBuffer", {1}) + "SUB17"
myLineFile.WriteElt (Str104)
Str105 = "KM" + av.Run ("MakeBuffer", {1}) + "RUNOFF COMPUTATIONS
                                         FOR SUBCATCHMENT 17"
myLineFile.WriteElt (Str105)
myExpression = "[wsheds#]= 17"
mySelection = myVTab.GetSelection
myVTab.Query (myExpression, mySelection, #VTAB_SELTYPE_NEW)
For each rec in mySelection
    wshedsnum = myVTab.ReturnValue (myVTab.FindField ("wsheds#"), rec)
    areawshed = (6692614 / (1609 * 1609)).SetFormat("d.dd")
    wcn = (myVTab.returnValue (myVTab.FindField ("areawcn"), rec))
    wcn = wcn.SetFormat("d")
    if (wshedsnum = 17) then
        Str106 = "BA" + av.Run ("MakeBuffer", {2}) + areawshed.AsString
        myLineFile.WriteElt (Str106)
    else
        MsgBox.Info ("The wsheds# was not 17.. sorry:):)" ++ NL, "Info")
    end
end
end
myVTab.GetSelection.ClearAll
myVTab.UpdateSelection
Str107 = "LS" + av.Run ("MakeBuffer", {5}) + "0"
        + av.Run ("MakeBuffer", {6}) + wcn.AsString
myLineFile.WriteElt (Str107)
tp17 = (0.38 * ((( 4789.95 / 1609 ) * ( 1723.90 / 1609 )) ^ 0.3))
tp17 = tp17.SetFormat("d.dd")

```

```

Str108 = "US" + av.Run ("MakeBuffer", {2}) + tp17.AsString
        + av.Run ("MakeBuffer", {4}) + "0.25"

myLineFile.WriteElt (Str108)

Str109 = "KK" + av.Run ("MakeBuffer", {1}) + "SUB16"

myLineFile.WriteElt (Str104)

Str110 = "KM" + av.Run ("MakeBuffer", {1}) + "RUNOFF COMPUTATIONS FOR
        SUBCATCHMENT 16"

myLineFile.WriteElt (Str110)

myExpression = "[wsheds#]= 16"

mySelection = myVTab.GetSelection

myVTab.Query (myExpression, mySelection, #VTAB_SELTYPE_NEW)

For each rec in mySelection

    wshedsnum = myVTab.ReturnValue (myVTab.FindField ("wsheds#"), rec)
    areawshed = (2684474 / (1609 * 1609)).SetFormat("d.dd")
    wcn = (myVTab.returnValue (myVTab.FindField ("areawcn"), rec))
    wcn = wcn.SetFormat("d")

    if (wshedsnum = 16) then

        Str111 = "BA" + av.Run ("MakeBuffer", {2}) + areawshed.AsString
        myLineFile.WriteElt (Str111)

    else

        MsgBox.Info ("The wsheds# was not 16.. sorry:):)" ++ NL, "Info")

    end

end

myVTab.GetSelection.ClearAll

myVTab.UpdateSelection

Str112 = "LS" + av.Run ("MakeBuffer", {5}) + "0"
        + av.Run ("MakeBuffer", {6}) + wcn.AsString

```

```

myLineFile.WriteElt (Str112)
tp16 = (0.38 * ((( 2574.88 / 1609 ) * ( 1246.41 / 1609 )) ^ 0.3))
tp16 = tp16.SetFormat("d.dd")
Str113 = "US" + av.Run ("MakeBuffer", {2}) + tp16.AsString
      + av.Run ("MakeBuffer", {4}) + "0.25"
myLineFile.WriteElt (Str113)
Str114 = "KK" + av.Run ("MakeBuffer", {2}) + "OUT4"
myLineFile.WriteElt (Str114)
Str115 = "KM" + av.Run ("MakeBuffer", {1}) + "COMBINE SUB10,12,15,16,17
      RUNOFF AND OUT3"
myLineFile.WriteElt (Str115)
Str116 = "HC" + av.Run ("MakeBuffer", {5}) + "6"
myLineFile.WriteElt (Str116)
Str117 = "KK" + av.Run ("MakeBuffer", {1}) + "SUB11"
myLineFile.WriteElt (Str117)
Str118 = "KM" + av.Run ("MakeBuffer", {1}) + "RUNOFF COMPUTATIONS FOR
      SUBCATCHMENT 11"
myLineFile.WriteElt (Str118)
myExpression = "[wsheds#]= 11"
mySelection = myVTab.GetSelection
myVTab.Query (myExpression, mySelection, #VTAB_SELTYPE_NEW)
For each rec in mySelection
  wshedsnum = myVTab.ReturnValue (myVTab.FindField ("wsheds#"), rec)
  areawshed = (8942919 / (1609 * 1609)).SetFormat("d.dd")
  wcn = (myVTab.returnValue (myVTab.FindField ("areawcn"), rec))
  wcn = wcn.SetFormat("d")
  if (wshedsnum = 11) then

```

```

Str119 = "BA" + av.Run ("MakeBuffer", {2}) + areawshed.AsString
myLineFile.WriteElt (Str119)

else
  MsgBox.Info ("The wsheds# was not 11.. sorry:~)" ++ NL, "Info")
end

end

myVTab.GetSelection.ClearAll

myVTab.UpdateSelection

Str120 = "LS" + av.Run ("MakeBuffer", {5}) + "0"
      + av.Run ("MakeBuffer", {6}) + wcn.AsString
myLineFile.WriteElt (Str120)

tp11 = (0.38 * ((( 8026.48 / 1609 ) * ( 3413.70 / 1609 )) ^ 0.3))
tp11 = tp11.SetFormat("d.dd")

Str121 = "US" + av.Run ("MakeBuffer", {2}) + tp11.AsString
      + av.Run ("MakeBuffer", {4}) + "0.25"
myLineFile.WriteElt (Str121)

Str122 = "KK" + av.Run ("MakeBuffer", {1}) + "SUB18"
myLineFile.WriteElt (Str122)

Str123 = "KM" + av.Run ("MakeBuffer", {1}) + "RUNOFF COMPUTATIONS FOR
      SUBCATCHMENT 18"

myLineFile.WriteElt (Str123)

myExpression = "[wsheds#]= 18"
mySelection = myVTab.GetSelection
myVTab.Query (myExpression, mySelection, #VTAB_SELTYPE_NEW)

For each rec in mySelection
  wshedsnum = myVTab.ReturnValue (myVTab.FindField ("wsheds#"), rec)
  areawshed = (2448895 / (1609 * 1609)).SetFormat("d.dd")

```

```

wcn = (myVTab.returnValue (myVTab.FindField ("areawcn"), rec))
wcn = wcn.SetFormat("d")
if (wshedsnum = 18) then
    Str124 = "BA" + av.Run ("MakeBuffer", {2}) + areawshed.AsString
    myLineFile.WriteElt (Str124)
else
    MsgBox.Info ("The wsheds# was not 18.. sorry:~)" ++ NL, "Info")
end
end
myVTab.GetSelection.ClearAll
myVTab.UpdateSelection
Str125 = "LS" + av.Run ("MakeBuffer", {5}) + "0"
    + av.Run ("MakeBuffer", {6}) + wcn.AsString
myLineFile.WriteElt (Str125)
tp18 = (0.38 * ((( 3260.80 / 1609 ) * ( 1847.13 / 1609 )) ^ 0.3))
tp18 = tp18.SetFormat("d.dd")
Str126 = "US" + av.Run ("MakeBuffer", {2}) + tp18.AsString
    + av.Run ("MakeBuffer", {4}) + "0.25"
myLineFile.WriteElt (Str126)
Str127 = "KK" + av.Run ("MakeBuffer", {2}) + "OUT5"
myLineFile.WriteElt (Str127)
Str128 = "KM" + av.Run ("MakeBuffer", {1}) + "COMBINE SUB11,18 AND OUT4"
myLineFile.WriteElt (Str128)
Str129 = "HC" + av.Run ("MakeBuffer", {5}) + "3"
myLineFile.WriteElt (Str129)
Str130 = "KK" + av.Run ("MakeBuffer", {3}) + "RES"
myLineFile.WriteElt (Str130)

```

```

Str131 = "KM" + av.Run ("MakeBuffer", {1}) + "ROUTE OUT5 THROUGH RESERVOIR"
myLineFile.WriteElt (Str131)
Str133 = "RS" + av.Run ("MakeBuffer", {5}) + "1" + av.Run ("MakeBuffer", {4})
      + "STOR" + av.Run ("MakeBuffer", {7}) + "0"
myLineFile.WriteElt (Str133)
Str134 = "SV" + av.Run("MakeBuffer", {5}) + "0"
      + av.Run("MakeBuffer", {6}) + "18"
      + av.Run ("MakeBuffer", {6}) + "25"
      + av.Run("MakeBuffer", {6}) + "40" + av.Run("MakeBuffer", {6}) + "50"
      + av.Run("MakeBuffer", {6}) + "65"
myLineFile.WriteElt (Str134)
Str135 = "SE" + av.Run("MakeBuffer", {1}) + "388.5"
      + av.Run("MakeBuffer", {3}) + "394.2" + av.Run("MakeBuffer", {3})
      + "398.2" + av.Run("MakeBuffer", {3}) + "404.8"
      + av.Run("MakeBuffer", {3}) + "410.8" + av.Run("MakeBuffer",{3})
      + "420.8"
myLineFile.WriteElt (Str135)
Str136 = "SL" + av.Run("MakeBuffer", {3}) + "391" + av.Run("MakeBuffer",
      {3}) + "19.63" + av.Run("MakeBuffer", {4})
      + "0.71" + av.Run("MakeBuffer", {5}) + "0.5"
myLineFile.WriteElt (Str136)
Str137 = "SS" + av.Run("MakeBuffer", {1}) + "401.8"
      + av.Run("MakeBuffer", {6}) + "30"
      + av.Run("MakeBuffer", {4}) + "2.86"
      + av.Run("MakeBuffer", {5}) + "1.5"
myLineFile.WriteElt (Str137)
Str140 = "ZZ"

```

```
myLineFile.WriteElt (Str140)
myLineFile.Close
if (File.Exists (myInputFileName) ) then
    av.ShowMsg ("Created the HEC-1 file....")
else
    MsgBox.Info ("The File" ++ myInputFileName.AsString ++ "does not
exist..Check it out!!" ++ NL, "Info")
end
myOutputFileName = FileName.Make ("/home/gis4/nara/hec1Icaria.out")
if (File.Exists (myOutputFileName)) then
    System.Execute("rm -f /home/gis4/nara/hec1Icaria.out")
else
end
System.Execute("hec1 < /home/gis4/nara/hec1inpIcaria
                > /home/gis4/nara/hec1Icaria.out")
if (File.Exists (myOutputFileName)) then
    MsgBox.Info ("The File" ++ myOutputFileName.AsString
                ++ "for input to HEC-1 was created"
                ++ NL ++ "Please Proceed to the Next Script''
                ++ ( Find the Coordinate Data)" ++ NL, "Info")
else
    MsgBox.Error ("The File" ++ myOutputFileName.AsString
                ++ "was not created" ++ NL, "Error")
end

av.ClearMsg
```


Avenue Script to Find The Hydrograph Data at the outlet

```

fileNameIn = "/home/gis4/nara/heclIcaria.out".AsFileName
fileNameOut = "/home/gis4/nara/heclIcaria.out".AsFileName
searchString1 = "HYDROGRAPH AT STATION      RES"
searchString2 = "*****"
searchString3 = "          "
searchString4 = "DA MON HRMN"
endString1 = "393.0"
endString2 = "393.1"
endString3 = "393.7"

fileIn = LineFile.Make( fileNameIn, #FILE_PERM_READ )
fileOut = LineFile.Make (fileNameOut, #FILE_PERM_WRITE)
numChanges = 0
counter = 0
total = fileIn.GetSize
done = false
av.ShowStopButton
av.ShowMsg("Searching...")
while (done.not)
    counter = counter + 1
    proceed = av.SetStatus( (counter / total) * 100 )
    if (proceed.Not) then
        av.ClearStatus
        av.ShowMsg("Stopped")
        exit

```

end

theString = fileIn.ReadElt

if ((theString <> nil)

 and (theString.Contains(searchString1))) then

 theString = fileIn.ReadElt

 while (Not (theString.Contains (endString1)

 or theString.Contains (endString2)

 or theString.Contains (endString3)))

 if ((theString.Contains (searchString2))

 or (theString.Contains (searchString3))

 or (theString.Contains (searchString4))) then

 theString = fileIn.ReadElt

 else

 newString = theString

 fileOut.WriteElt (newString)

 theString = fileIn.ReadElt

 end 'Corresponds to if (theString.Contains(searchString2))..

 end 'Corresponds to while (Not (...

fileOut.WriteElt (theString)

done = true

else

```

    newString = theString
end 'Corresponds to if (( theString <> nil) and ..
end 'Corresponds to while (done.not)....

av.ClearStatus

av.ShowMsg ("Stopped...")

' Close the files and report results...

fileIn.Close

fileOut.Close

myFileName3 = FileName.Make ("/home/gis4/nara/final.txt")

System.Execute ("rm -f /home/gis4/nara/junk1.hec")
System.Execute ("rm -f /home/gis4/nara/junk2.hec")
System.Execute ("rm -f /home/gis4/nara/junk3.hec")
System.Execute ("rm -f /home/gis4/nara/final.txt")
System.Execute ("cut -c16-17,21-25 hec11Icaria.out > junk1.hec")
System.Execute ("cut -c60-61,65-69 hec11Icaria.out > junk2.hec")
System.Execute ("cut -c103-105,110-113 hec11Icaria.out > junk3.hec")
System.Execute ("cat junk3.hec >> junk2.hec")
System.Execute ("cat junk2.hec >> junk1.hec")
System.Execute ( "cc -o /home/gis4/nara/txtformat /home/gis4/nara/txtformat.c")
System.Execute ("/home/gis4/nara/txtformat")
'System.Execute ("awk '{ print $1 ", " $2 }' junk1.hec > final.txt")
if ( File.Exists (myFileName3) ) then

    MsgBox.Info ("The File" ++ myFileName3.AsString ++ "Has been created"
                ++ NL ++ "Proceed to the next script'")

```

```
    ++ "(Plot the output hydrograph)", "Info")

elseif ( Not ( File.Exists (myFileName3))) then
    MsgBox.Info ("The File" ++ myFileName3.AsString ++ "Has Not Been created"
        ++ NL, "Info")

    exit

else
    exit

end
```

Avenue Script to Plot The Output Hydrograph

```
myFileName = FileName.Make ("/home/gis4/nara/final.txt")

if (File.Exists (myFileName) ) then
    myVTab = VTab.MakeNew ("/home/gis4/nara/info/coord.sta".AsFileName, INFO)
    Field1 = Field.Make ("Time Int 15 Min", #FIELD_SHORT, 4, 0)
    Field2 = Field.Make ("Runoff In CFS", #FIELD_SHORT, 5, 0)
    myFieldList = {Field1, Field2}
    myVTab.SetEditable (True)
    myVTab.Activate
    myVTab.AddFields (myFieldList)
    myInputFile = LineFile.Make (myFileName, #FILE_PERM_READ)
    totalRecs = myInputFile.GetSize
    xfield = myVTab.FindField ("Time Int 15 Min")
    yfield = myVTab.FindField ("Runoff In CFS")
    presentRec = 0

    av.ShowStopButton
    av.ShowMsg ("Converting the comma delimited file into a VTab" + ".....")

    while (true)
        buffer = myInputFile.ReadElt
        if (buffer = Nil) then
            break
        end
        buffertokens = buffer.AsTokens(",")
```

```

if ( (buffertokens.Count) = 2) then
    xcoordinate = buffertokens.Get(0)
    ycoordinate = buffertokens.Get(1)
    rec = myVTab.AddRecord
    myVTab.SetValueNumber (xfield, rec, xcoordinate.AsNumber)
    myVTab.SetValueNumber (yfield, rec, ycoordinate.AsNumber)
elseif ( (buffertokens.Count) <> 2) then
    MsgBox.Info ( "The List" ++ buffertokens.AsString ++
        "does not contain 2 elements" ++ NL, "Info")
    exit
end 'Corresponds to if ( (buffertokens.Count) = 2) then...
presentRec = presentRec + 1
progress = ( presentRec / TotalRecs ) * 100
proceed = av.SetStatus ( progress )
if (proceed.Not) then
    av.ClearStatus
    av.ShowMsg ( "Stopped...." )
    exit
end 'Corresponds to if (proceed.Not) then...
end 'Corresponds to while (true)

elseif ( Not ( File.Exists (myFileName) ) ) then
    MsgBox.Info ("The File" ++ myFileName.AsString ++ "does not Exist...")
    ++ ' ' Check it out" ++ NL, "Info")
    exit
end 'Corresponds to if (File.Exists (myFileName)) then

```

```
myChart = Chart.Make (myVTab, {Field2})
myChart.ShowGallery (#CHARTDISPLAY_LINE)
myDisplay = myChart.GetChartDisplay
myDisplay.SetType (#CHARTDISPLAY_LINE)
myDisplay.SetStyle (#CHARTDISPLAY_VIEW_SIDEBYSIDE)
myChart.SetName ("Output Hydrograph for Lake Icaria")
myChart.SetSeriesFromRecords(False)

myChart.GetXAxis.SetName(xfield.AsString.UCase)
myChart.GetXAxis.SetLabelVisible (True)
myChart.GetXAxis.SetTickLabelsVisible (False)

myChart.GetYAxis.SetName (yfield.AsString.UCase)
myChart.GetYAxis.SetLabelVisible (True)
myTitleString = "Discharge Hydrograph For Lake Icaria Watershed" ++ NL
                ++ "Starting Time : 12 June, 1968, 0715" ++ NL
                ++ "Ending Time : 13 June, 1968, 1030" ++ NL
                ++ "Total X-Coordinates : 110" ++ NL

myChart.GetTitle.SetName (myTitleString.UCase)
myChart.GetChartLegend.SetVisible (False)
av.GetProject.AddDoc(myChart)
myChartWin = myChart.GetWin
myChartWin.Resize (800,600)
MyChartWin.Open
myChartWin.Activate
```