COMPUTER TOOLS FOR DEVELOPMENT OF PARAMETERS

FOR AN URBAN WATERSHED USING DESKTOP GIS

By

NAZIR AHMED HAWARY

Bachelor of Science
University of Engineering and Technology
Lahore, Pakistan
1982

Master of Science
California State University
Long Beach, California
1993

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF PHILOSOPHY
May, 2000

COMPUTER TOOLS FOR DEVELOPMENT OF PARAMETERS

FOR AN URBAN WATERSHED USING DESKTOP GIS

Thesis Approved:

_____
Thesis Adviser

_____

_____

_____

_____
Dean of the Graduate College

ACKNOWLEDGMENTS

A doctoral degree is the epitome of human effort in formal education and its realization is certainly not possible due only to individual struggle. In this long and arduous endeavor, the assistance and advice of many are included. Before I mention a few of my benefactors, my eternal gratitude and all thanks go to Allah, the Merciful, for his Compassion and Mercy.

I am indebted to my adviser, Dr. Avdhesh Tyagi, for providing the motivation and opportunity to conduct this research. He gave his time to test the software developed during this research, formulate various methodologies of the software, conduct analyses for verification of designed tools, and review numerous drafts of the manuscript.

I especially acknowledge the intellectual contributions, thoughtful reviews, and comments of my committee members--Dr. John Veenstra, Dr. Vernon Mast, Dr. James Lawler, and Dr. Allen Finchum. I will always remember Dr. Veenstra for his collaboration, for our technical discussions, and for his ever-open door policy extended to me. He is a modern day saint and worthy of absolute respect.

I express my deep respect to Dr. Allen Finchum for his fully involved service and guidance on my committee. This dissertation is the result of his technical assistance and involvement. He was always there to guide me in unfolding the complexities of GIS and solving hardware- and software-related problems that arose while interfacing two completely different software in this research.

I am deeply indebted to Dr. Robert Dickinson and Mr. Tony Kuch of XP Software Inc. for their technical advice and hospitality. They were always generous with their time, technical advice and with free copies of XP–SWMM software and its upgrades for this research. In particular, thanks are extended to Mr. Jeff Thompson, President of XP Software Inc., for his visit at OSU and personal involvement in this research. He attended many of my presentations and spent much of

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

According to numerous physiological studies, perceptions of most modelers change with time due to fatigue, mood, and repetitious work (Henning, 1996). Multiple manual area measurements of the same watershed by the same modeler may differ each time, and perceptions of the different modelers performing the same task also differ and depend on experience, skill, and vision. These traditional measurement techniques lack reproducibility—the same set of parameters developed by two groups of expert mappers might produce different values for the same parameters.

The development of large quantity of input data is the very characteristic of lumped watershed models that often render them inefficient for everyday operational hydrology. This application provides a difference in the tedium of hand measurements from maps by using geographical information systems (GIS) which provides evidence of superior GIS results to more traditional methods. In this study new procedures are developed that enable GIS technology to extend its capabilities in the water resources field for surface water quantity and transport modeling. This application links ArcView 3.1 (ESRI 1998) and XP-SWMM 32, a privately enhanced version of EPA's Surface Water Management Model (SWMM) (XP-SWMM, 1998), and owned by CaiCE Software Corporation (CaiCE 1998).

This research demonstrates the application of ArcView 3.1 and one of its extensions, Spatial Analyst, to change the way the water resources community uses XP-SWMM, prepares its input or performs modeling runoff quantity estimation, and presents the simulated results graphically. The code to interface these two software is written in Avenue, which is the script language of ArcView.

1

1.1 Background

Since the introduction of desktop personal computers (PCs) in the 1980s, computer systems have continuously eased the work of water resources professionals through a broad range of state-of-the-art hardware and software. This technology has caused an evolution in hydrologic/hydraulic (H/H) modeling practices, such as data development, forecasting sewer and channel flows, and pollution assessment. Many models for sewer and runoff collection systems have been developed which simulate the flow and movement of precipitation and pollutants from the ground surface through channel networks to the receiving waters.

These H/H models are continuously updated for predicting flows, pollution concentrations, urban runoff problems and various abatement options. The variability of problems in a watershed and their abatement options dictate the extent and complexity of input data requirements of these models. The greater complexity of the model requires a greater volume of input, especially if the subject watershed has a higher number of problems (Navulur and Engel, 1996).

Preparation of input and interpretation of output required by ever-changing H/H computer models have become complex and tedious. The nature of the input in a watershed is related to rainfall characteristics, generation of runoff, and watershed geomorphology. Geomorphic characteristics are the channel (pipe) network and surrounding landscapes that translate the rainfall into an output hydrograph at the watershed outlet. The development of topographic data for these H/H models by traditional manual map measurements is time-consuming, which needs to be enhanced by taking advantages of the sophisticated computer software and powerful hardware.

The introduction of spatial (terrain) analysis software has eased the use of digital elevation models (DEM) for extraction of intelligent information about sewage and runoff collection and conveying networks. With the introduction of various powerful programming languages and operating systems, PCs can combine spatial analysis software—i.e., GIS—to any surface water management software.

GIS software, originally developed for storage and analysis of spatially distributed data, can also support a variety of applications in water resources management, hydrology, and environmental analysis. The knowledge of terrain and land cover features (geomorphic characteristics) of a watershed is improved by GIS (Maidment, 1993). Various input requirements

2

—topographic slope, land use, soil characteristics, flowpath, and watershed limits—of water models can be developed through the spatial analytical tools of GIS as they are suitable to deal with spatial variability of runoff parameters (Jenson and Domingue, 1988).

The combination of GIS and H/H models mixes spatial data storing, managing, analyzing, mapping, and visualizing capabilities of the GIS with flow quantity and quality simulation (estimation) capabilities of the H/H models. Various integrated spatial analysis tools in GIS provide spatial parameters needed for H/H models. This combination is suitable for continual updating, for operations needing reproducibility, and for rapid recomputation of alternative scenarios as is typically required in water resources planning and management projects.

The approach developed in this thesis focuses on front- and back-end applications of GIS by introducing a two-way link between ArcView and XP-SWMM. Front-end applications include the computation of watershed parameters for a runoff block of XP-SWMM in ArcView environment (i.e., using GIS as the platform to develop data); and back-end applications include the cartographic display of the computed H/H output from XP-SWMM in ArcView.

Currently, most GIS and hydrological modeling software use either an all-raster or an all-vector approach; but in this interface both vector and raster data are used to take advantage of the best features of ArcView and one of its extension, Spatial Analyst. This extension provides tools to perform integrated analysis using feature- and cell-based (grid) themes (i.e. DEM). To calculate slope, aspect mapping, delineation of watershed limits, and other derived parameters, the DEM of an urban watershed is digitized and used for the spatial analysis operations.

The XP-SWMM is a lumped-parameters H/H model and its latest version can use WIN95 or NT or DOS as operating system (CaiCE, 1998). It is a large and complex model that simulates the movement of sewage, runoff, and pollutants from the ground surface through sewer lines, channel networks, storage treatment units, and finally to receiving waters or treatment plants. This software is being used effectively in various water resources-related applications. But like many other H/H software, it lacks the advantages of GIS features. Incorporating the GIS interface developed in this study with XP-SWMM will allow utilization of certain features of GIS, such as, spatial data storage, spatial data intelligence, and processing systems (data storage and logic between spatially-related features). As the knowledge of terrain and land cover features is

3

improved by GIS, it will be used by XP-SWMM to analyze watershed delineation, topographic slope, flow length, land use, imperviousness, pipe/channel length, soil characteristics and Manning's roughness of a watershed etc.

ArcView 3.2 is a user-friendly desktop mapping tool that can utilize raster (DEM) as well as vector data (pipe/conduit map). Its extension, Spatial Analyst, and its script language, Avenue, have several built-in hydrological functions that are extensively utilized in this research for automatic generation of input parameters for XP-SWMM. The Avenue enables modelers to create custom applications and user interfaces within ArcView environment. It consists of tools to create, query, analyze cell-base data; and to perform integrated analysis on DEM, i.e. use feature- and cell-based (grid) themes. Hydrological analysis can be executed to calculate terrain slope, aspect, flowpath, and delineation of watershed boundaries, etc. which are basic input data required by most of the H/H models (ESRI, 1999).

1.2 Objectives

Sanitary sewer systems and stormwater collection systems consist of a network of manholes, inlets, and gutters (nodes) connected by pipes (links). These networks are installed to collect runoff and wastewater and transmit them mainly as gravity flow to treatment plants or to outfalls. The hydraulic design of runoff conveyance systems and sewer lines depends on the determination of runoff quantity for all inlets, discharge rates, pipe sizes, slopes, roughness coefficients, and invert elevations. The tedious and lengthy requirements for hand calculation, especially for larger networks, limit the development and evaluation of alternative designs. Although most H/H calculation procedures are now available in computers programs and their use has substantially reduced the mathematical effort involved. However, a substantial effort is required to establish and manipulate the data required for input into H/H models (Maidment, 1993).

The traditional estimation of runoff parameters for XP-SWMM and for many other H/H models has been performed manually. For example, delineation and measurement of drainage area and other runoff parameters may require piecing together topographic, soil, and land use maps for which a designer must establish drainage boundaries by first interpreting elevation contours and then developing slope, slope length, roughness and infiltration, etc. The development of these parameters is tedious, expensive, and lacks reproducibility—two groups of hydrologists might

produce slightly or substantially different watershed boundaries, thus affecting the subsequent parameterization process. Measurement of these parameters involves the use of planimeters or digitization on computer-aided design (CAD) systems. But the sophistication in computer graphics and advances in desktop GIS have made the preparation of input data, construction of model, and interpretation of results, an easier process.

The main objective of this research is to demonstrate the application of desktop GIS—ArcView 3.1 and its extension Spatial Analyst—to change the way hydrologists prepare the input and consequently perform the XP-SWMM modeling. The GUI developed in this study consists of designing and programming an integrated set of ArcView routines (Avenue scripts) and associated data, and provides a user with the capability to establish some of the runoff parameters used in XP-SWMM. This interface incorporates a menu-driven system within which users can develop, edit, or import a runoff collection network. By incorporating Spatial Analyst, the runoff parameters (such as area, slope, and width etc.), of a delineated subwatershed around a manhole (inlet) are automatically determined from the DEM of the watershed.

The effectiveness of this interface is evaluated after developing the link between ArcView and XP-SWMM. This effectiveness is determined by automatically generating the preceding three parameters for XP-SWMM's runoff block used for modeling the Storm-water Collection System for the City of Ann Arbor, Michigan, and comparing them with the values of these parameters estimated manually. Following are the four specific objectives to be achieved by this study:

1. Evaluate the applicability of GIS technology for deriving urban watershed characteristics that are required as input data for XP-SWMM.

2. Develop a methodology for automatic extraction of four parameters for runoff block, i.e., area, slope, width, and imperviousness of a subwatershed.

3. Construct ArcView Extensions using Avenue to exchange information between ArcView and XP-SWMM. Develop new attribute coding format in ArcView to import the output files from XP-SWMM into the ArcView database. Show results (surcharging manholes, pipes exceeding design capacity, surcharging manholes, and inundated areas) graphically within the ArcView platform.

4. Evaluate and verify the procedures by conducting a comparative study with an existing data set developed by the traditional engineering methods.

5

In this research, the newly developed methodologies to generate runoff parameters for XP-SWMM from digital maps are based on the overlapping of criteria (Boolean Algebra) and not on the analysis of flow (i.e., the values of these parameters are created using ArcView GIS, Spatial Analyst and a DEM). The GIS spatial analysis does not involve any physical laws as this analysis does not consider the dynamics of water flow. This interface only automates the overlapping routines to generate watershed runoff parameters. These routines have procedural similarities to the manual approaches.

1.3 Statement of the Problem

Surface water and sewage drainage studies using XP-SWMM require significant efforts in terms of data organization, development, calibration of parameters, and presentation of results. When land use patterns in the study area change rapidly, the problem becomes even greater because new plans need H/H modeling of areas becoming urbanized. This change in terrain due to development requires requantification of runoff parameters. To counter these difficulties, an application is needed to take advantage of the spatial analysis capabilities of the GIS technology. The abilities of GIS to organize, store, conduct spatial analysis and display spatial data (maps) and nonspatial (relational data) for any part of landscape can be integrated with XP-SWMM to assemble input and store output from it for analysis and display.

Currently most engineers who deal with water resources create much of the input physically by generating manual overlays of paper maps and manual measurements of length, width, and area using map wheels and planimeters to estimate parameters of H/H models (i.e., STORM, HEC-1, SWMM, MOUSE, and XP-SWMM). Traditionally these parameters have been produced manually; although they can be accurate but are tedious and expensive (especially updating procedures) as they lack reproducibility.

6

The interface introduced in this dissertation is the first endeavor to use XP-SWMM, ArcView GIS, and Spatial Analyst which are PC based software and can use WIN-95 operating system. This interface addresses the following four problems:

1. Assisting water resources professionals to develop and evaluate the impact of different growth scenarios in developing watersheds more rapidly.

2. Generating parameters automatically using GIS spatial analysis capabilities—without manual involvement of the modeler—will eliminate human subjectivity in estimating physical model input parameters of the watershed.

3. Reducing the efforts required for map manipulation, table referencing, and repetitious computations for determination of hydrologic parameters. This saving in human energy will result in the increased ability of hydrologists to make responsible decisions based on the most detailed and intelligent data available.

4. Currently, several simulated results of XP-SWMM runoff modeling cannot be displayed efficiently and are difficult to be interpreted by decision makers, especially with non-engineering background. But interfacing it with ArcView and using the graphical capabilities of GIS, several options can be displayed. The impacts of future urbanization on the existing runoff collection system, surcharging manholes, under-over-capacity pipes could be identified. The extent of detrimental localized flooding by surcharging manholes could be displayed.

7

CHAPTER 2

LITERATURE REVIEW

Recently, geographic information systems (GIS) have become prominent tools in H/H model preparation and evaluation of modeling results. The most common method of surface water modeling with GIS has been to couple an existing model code with a spatial data set via external files. Typically, GIS is used as a data preprocessor to obtain, organize, and clean data sets of runoff, soils, roughness coefficient, land use, infiltration and precipitation rate, etc. Once these data sets are prepared, GIS exports these files (after reformating) which are subsequently used as input to the modeling program. After this external program has completed simulation, its input is reformatted into a form that can be exported to and understood by GIS for further spatial analysis and display. Burrough et al. (1988) categorize model types used in water resources and state the need for connecting mathematical models, database, and graphics systems for a given area. They provide a list of requirements that must be satisfied to effectively integrate GIS (as a spatial database) and models. These requirements deal primarily with accuracy and appropriateness of data stored in GIS and data required by the program. They concluded that neither GIS nor mathematical models are sufficiently developed to provide seamless integration, and much research is needed before integration can be achieved.

GIS can link land cover data to other information about processes and properties related to a particular geographic location. When applied to hydrologic systems, information can include descriptions of soil, land use, and ground cover. It provides ease and accuracy in surface terrain representation, watershed delineation, precipitation data compilation related to environmental processes, surface water modeling predictions, agricultural chemical concentration estimations, and water balance forecasting (Mizgalewicz, 1993)

Part of the research described in this study is a contribution toward understanding how GIS can be used in modeling urban storm-water drainage and sewer networks. The literature search was conducted based on the following categories:

1.   XP—SWMM or SWMM interfaces with expert systems

2.   Interfaces between various H/H models and geographic information systems.

3.   Interfaces between ARC/INFO, SWMM, and H/H models.

4.   Interfaces between ArcView and various H/H models

2.1 SWMM and Expert Systems

The basic H/H model, SWMM, cited in this dissertation has been utilized in numerous studies. Several studies of SWMM with various expert systems and GISs are performed to check its ability for real-time runoff and for better urbanization planning, e.g. proper decision on dimensions and sites of storm drains.   Aldrich and Roesner (1986) and Liong et. al (1991) developed knowledge-based systems for the Runoff Block.   One of these expert systems is known as a Knowledge-Based Stormwater Management Model (KBSWMM) that contains: (1) a pre-processor developed with the aid of the X windows system; (2) a knowledge-based component, for SWMM calibration, implemented with the help of Nexpert, an object-based expert system development tool; and (3) an enhanced graph presentation capability compared to that of SWMM.   Milles (1986) developed a preprocessor that was based on an application of Lotus 1-2-3 to SWMM and he obtained reasonably good agreement between the values suggested by the expert system and a professional user.

Baffaut and Delleur (1989) developed another expert system for calibrating SWMM through use of an expert-system-shell called Knowledge Engineering System (KES). The automation and methodical approach in performing the calibration required less time and experience of the user and achieved, in some cases, better results than the traditional approach to calibrate SWMM.

2.2 GIS Interfaces With Environmental and Hydrologic Models

In this section, a literature search is presented which focuses on interfaces developed between various GISs and environmental models. A GIS in conjunction with the H/H models was used by Gupta and Solomon (1977).   They employed a regular raster system to store stream information for use in modeling runoff and sediments. White et al. (1985) described the application

of a grid cell data system in conjunction with the SCS TR-55 lumped parameter model to extract the watershed delineation by examining drainage paths over grid elevation data. Evans and Miller (1988) applied raster-based GIS technology to target nonpoint pollution from agricultural lands.

Stuebe and Johnston (1990) used GIS to delineate the spatial extent of hydrologic response units, estimate the runoff volume utilizing the Soil Conservation Service (SCS) method to represent factors in Universal Soil Loss Equation (USLE), and determine soil erosion. Pansuka et al. (1990) interfaced a grid-based terrain analysis method and grid-based DEMs with the AGNPS model to generate terrain-based model parameters which represent about one-third of the input parameters for the model. Use of grid-structures for these models has a major benefit, as it allows pixel-based remotely-sensed data—such as vegetation types and cover (canopy) characteristics— to be used for estimating model parameters in each element or cell because of the inherent compatibility of the two structures.

Sasowsky and Gardner (1991) used a raster-based GIS, ERDAS, to parameterize a quasi-physically based surface-runoff model. For similar applications, Srinivasan et al (1993) developed interfaces between AGNPS, a pollutant runoff model, and GRASS and used it for a wide range of nutrient and sediment variables. He et al. (1993) studied environmental impacts of phosphorus transport by integrating Agricultural Non-Point Pollution Model (AGNPS) with GRASS, to predict runoff, display phosphorus and sediment yield on maps. They showed various management practices and alternatives to decrease the adverse impacts

Trybus (1994) used Desk Top Mapping (DTM), a sub set of GIS, for managing a sanitary sewer collection system. The DTM, although not as powerful as GIS, offers a more functional means of integrating AutoCAD files with existing sewer system evaluation survey (SSES) data. The methodology for incorporating HYDRA, CYBERNET, and SSES software programs is presented by Gray and Peel (1994). In another report, Mckibben (1994) discussed the use of GIS for estimating wastewater flows and capacities of wastewater collection systems, while Majure and Eash (1994) developed an automated method to quantify various runoff parameters by GIS.

Barber et al (1994) integrated SWMM and Intergraph Microstation PC and Oracle database. The input data were organized to describe the drainage basin in a database; this was accomplished by creating a logical database format and form with an Oracle database to input,

store, and display RUNOFF and TRANSPORT Blocks information. The datafiles of SWMM were created by developing an application of Microstation Development Language (MDL) to export data from GIS to SWMM datafiles. Finally, simulated results from SWMM were displayed graphically within GIS by showing conveyance system capacities, design flows, and inadequacies on the drainage schematic. Digital maps containing watershed planimetrics and contour information are used as base maps for developing the graphic files. The spatial analysis is executed by using Intergraph's Modular Graphics Environment (MGE) software on graphic files to provide input data.

Julian et al. (1995) used a raster-based GIS to manage large databases describing the detailed spatial configuration of a land surface. Raster GIS and radar rainfall data were combined with a two-dimensional physically-based rainfall-runoff model CASC2D that simulates spatially-varied surface runoff. They obtained topographic data such as elevation, soil type, and land use by satellite imagery and photogrammetry. It typically employs field-measured or remotely-sensed values describing the spatially-varied nature of watershed topography, soils, vegetation, drain-age networks, and rainfall. These variable are used as input to numerical algorithms based on infiltration physics and overland/channel flow to model the response of a watershed.

Van Gelder and Miller (1996) presented an interactive tool consisting of MapInfo GIS and XP-SWMM. MapInfo GIS is used to produce a records management system, generate thematic maps, record and track information. The system is represented as a link-node network to facilitate incorporation into XP-SWMM. XP-SWMM results are used to evaluate existing conditions, determine necessary improvements, and develop a regional stormwater master plan.

Ball and Lok (1998) used spline surfaces within GIS and produced robust and accurate estimates of rainfall and enabled real-time estimation of alternative hyetographs for different locations within the catchement. Kull and Feldman (1998) discussed the unit hydrograph method by incorporation of spatially distributed rainfall data and travel times by using radar data and GIS. The effects of various types of pollution on the degradation of surface and groundwater within the state of Pennsylvania were assessed by Evan and Nizeyimana (1988). By using GIS and runoff models, they computed annual loads for nitrogen and phosphorous for each watershed including agricultural runoff, nonagricultural fertilization, and urban runoff.

2.3 Environmental Models and ARC/INFO

Systems like ARC/INFO GIS are capable of accessing large amounts of spatially-varying or spatially-related data. In this way, geographic features (roads, topography, hydrography) can be digitized and made available for independent analyses or in some aggregated fashion via overlaying. ARC/INFO technology has already been applied to stormwater modeling scenarios by Terstriep and Lee (1989) and DeBarry and Carrington (1990).

Warwick and Hanes (1994) and Cera et al. (1995) identified the role of professional judgment in the application of GIS technology to evaluate the efficacy of the ARC/INFO GIS applications to predict storm water quantity and its ability to accurately derive spatially-related information required to compute runoff hydrographs. They demonstrated a viable method of data construction and showed working of the ARC/INFO system by performing tedious and time consuming tasks of spatial averaging (basin areas, average runoff curve numbers) quite well. They quantified specific inaccuracies associated with the application of GIS to H/H modeling.

Curtis (1995) developed SWMMDUET software for the integration of SWMM with ARC/INFO and was written in ARC/INFO's Macro Language (AML). It creates a computing environment that does not require knowledge of SWMM and ARC/INFO. SWMMDUET incorporates both pre- and post-processors and thus provides interfaces for creating SWMM's output file. This interface has significantly simplified management of a vast amount of hydrologic data and allows hydrologists to concentrate on hydrologic matters.

Adamus et al. (1995) used spatial processing capabilities of ARC/INFO GIS to create large data sets required as model input. They introduced a screening-level nonpoint source pollution model—Pollution Load Screening Model (PLSM)—written entirely with a GIS software package. The model input data consist of spatial data layers, runoff coefficients, mean runoff concentrations, and stormwater treatment efficiencies. The entire model is contained in GRID; is a cell-based geoprocessing system with analytical capabilities. The model is processed on a cell-by-cell basis and can screen vast areas of land with respect to potential pollution load problems caused by stormwater runoff using a relatively simple computational algorithm.

Shamsi (1996, and 1997) used both vector and raster systems to take advantage of the best features of each and developed a front-end application of PC-based ARC/INFO GIS and ERDAS.

12

In this interface an integration of the Penn State Runoff Model (PSRM), a lumped parameter hydrologic model, was presented with the ARC/INFO, PC version. The primary objectives of GIS analyses were derivation of subbasin physical parameters and implementation of a watershed-wide stormwater management plan.

Robbins and Phipps (1996) at Woolpert Consultants ran GIS to prepare data for HEC-1 and HEC-2, and obtained better modeling results. The authors presented ARC/INFO tools and Woolpert's customized techniques to model 2- through 100-year storm events for floodplain management and stormwater infrastructure design. GIS base mapping is used to create digital orthophotos; a Digital Terrain Model (DTM) is used to obtain surface models. Contours with topographic data within the GIS platform streamline the modeling process, and GIS calculates model parameters, cuts channel cross sections, and performs tasks that otherwise would be done manually. These routines replaced manual calculations or the use of complicated spreadsheets for organizing and generating data required for modeling. GIS produces watershed, basin, and subbasin boundaries needed as a reference for both HEC-1 and HEC-2 models from DTMs and contours, and then uses these boundaries to generate input data for the HEC-1 model, i.e. SCS number, time of concentration, and lag time. Watershed boundaries are automatically and consistently determined with the ARC/INFO GRID module that produces a three-dimensional lattice surface model from DTM.

Mercado (1996) applied aerial photography, ARC/INFO GIS, and ERDAS Imaging Processing in conjunction with the XP-SWMM. The georectified image was used as a background to other project GIS overlay data, and a two-foot contour-elevation data set was used to create a DEM of the project area. Thematic overlay of elevation contour data, streams, and image backgrounds allows identification of existing surface runoff detention areas. The slope-optimization analysis technique determined high erosion potential areas, prevalent gradients, and accurate subbasin delineation; TIN and GRID modules of ARC/INFO GIS were used with project DEM. The DEM was modeled with an ARC/INFO grid module to perform slope optimization. Automation of GIS data and the XP-SWMM linkage process via script files or ARC/INFO AML language was not developed. GIS/DEM derived data were pre-processed for manual input into XP-SWMM, and parameters for Runoff and Extran Blocks were manually processed to perform management of H/H

13

modeling. The results can be displayed on project DEM and/or scanned imagery. The use of GIS and scanned aerial photo imagery for stormwater management modeling has proven very useful parameter generation and for display of intermediate and final results.

Bishop et al. (1998) evaluated maps of runoff with a 5-km and 10-km resolution from parameter-elevation regressions on independent slopes model (PRISM) by utilizing GIS and found less interpolation errors in the 5-km resolution. Wang and Hjelmfelt (1998) developed a physically based runoff model to route overland flows from flat watersheds, using spatially-varied data of soils, crops, land slopes and aspects, which were extracted from GIS database and DEM of the watershed. The overland flow was described by diffusion wave model due to the presence of very flat slopes on the watershed. The errors in observed discharges and computed discharges ranged between 4% and 47%, whereas the average error for peak discharge was 13.78 %.

Hourani (1998) developed physical parameters from a DEM (generated from satellite imagery). The values of these GIS generated parameters varied less than 1% from measured values. Lieberman et al. (1998) and Hussain and Schwartz (1998) applied GIS tools to the estimation and prediction of stormwater flows. The data layers included USGS digital and scanned image maps, SCS soils maps, and engineering drawings for the watersheds. Bryant et al. (1998) instituted a comprehensive stormwater infrastructure and conveyance system inventory to be used as a citywide preventive maintenance program, in addition to stormwater and watershed applications.

Nelson et al. (1999) used a hybrid algorithm for TIN creation and implemented in the Watershed Modeling System (WMS). They have made some comparisons with DEM utilization. The TIN needs less data space with respect to DEM. They stress that while developing TIN model, an important criteria for accurately defining boundaries is that triangle edges of the TIN honor linear drainage features such as streams or roads.

2.4 ArcView and Environmental Models

This section deals primarily with the application of ArcView and its interfaces with various environmental models. A few studies have been conducted for the applications of this desktop GIS with various models because it is comparatively newer in market.

Vasarhelyi (1996) discussed an integrated approach in drainage master planning projects that provided solutions for multi-objective needs. He applied GIS technology through two projects

14

and showed that use of GIS applications in drainage master planning projects has proven to be very efficient, especially if the necessary data are readily available. The author studied closed conduit storm drains by using GIS innovative solutions for analysis of engineered drainage facilities and their integration into drainage system. The author used SWMM to evaluate and identify deficiencies in various management practices such as flood control, and erosion and stormwater quality control in existing and future systems. GIS is used to develop hydrologic parameters for the SWMM model such as percent imperviousness, surface slope, surface roughness, soil infiltration rate, and tributary areas. GIS also developed pollutant generation and pollutant parameters for the water quality model and for potential detention basin sites. Data sets used for the development of runoff quality and quantity input parameters include topographic coverages (contour, digital elevation, or digital terrain models); soil coverage with attribute table; land use; natural water systems (rivers, streams, and lakes); streets, highways and other transportation corridors; drainage facilities; closed conduit coverage; and drainage boundaries. The evaluated results from SWMM are presented graphically through GIS. The results of this application from two sites show the consistency of manually computed and developed hydrologic model parameters. This application shows the benefits of interfacing GIS with SWMM, as the overall quality of H/H modeling results has increased significantly with the use of GIS.

Xue et al. (1996) developed a Best Management Practices Assessment Model (BMPAM) to assess best management practice performance based on runoff and pollutant removal mechanisms. The model is linked to ArcView software using ArcView macro language, Avenue. This integrated tool consists of the BMPAM model, input/output data, pre- and post- processors, and the GIS–BMPAM interface. ArcView provides needed information to other components. The input data component links with the pre-processor via the GIS user interface and data from the pre-processor are fed into the BMPAM model. The post-processor component displays model simulation results in various tabular and graphic formats through the GIS user interface. Files generated by the pre-processor and the BMPAM model are stored by the system for reuse or modification through the GIS user.

Di Luzio et al. (1997) used ArcView GIS to develop an interface with SWAT. This SWAT-GIS tool was developed to help water resources managers to improve the efficiency of analysis for nonpoint and point pollution assessment and management plans on a regional scale. SWAT is used to estimate sediment loadings under different management practices within a watershed and the input information for the model was derived from available spatial databases.

Shamsi et al. (1998) discuss ArcView GIS and XP-SWMM and described their applications in four typical activities associated with the management of stormwater and wastewater collection systems: mapping, monitoring, modeling, and maintenance. The authors demonstrated that due to its low cost, ease of use, and compatibility with the ARC/INFO file format, ArcView is an effective tool for routine display and plotting of collection system maps, querying the GIS database, developing sewer system hydraulic models, and conducting computerized maintenance management. The authors used ArcView to develop a Computer Mapping Program (CMP) for data collection, data management services and infrastructure mapping.

The Danish Hydraulic Institute (DHI) (1998) has an interface between ArcView and MOUSE, called MOUSE GIS. MOUSE is a numerical sewer modeling system that can calculate surface runoff, water quality, and sediment transport in urban catchments and sewer systems. This interface works in the ArcView environment and consists of two parts—Network Editor and Results Presentation. The network simplification process is based on such criteria as maximum change in pipe diameter, and maximum change in slope. This interface does not incorporate Spatial Analyst.

Cedra Corporation Inc. (1998) has introduced ArcView and SWMM integration software called AVSand. The program is used to model sanitary sewers but offers limited SWMM modeling capability. It uses a simplified implementation of SWMM's TRANSPORT Block only and does not use other blocks of SWMM. Storm and combined sewers are modeled using rational methods or surface hydrographs instead of SWMM. AVSand accepts model input via a series of dialogue boxes, and provides model development and editing tools. It also does not use Spatial Analyst.

Bonta (1998) compared runoff data from two small, adjacent similar agricultural watersheds but with dissimilar slopes, soil and landuses and showed that annual runoff can be much higher on one watershed compared with the other using DEM and GIS spatial capabilities. The results

16

suggested that deterministic and random spatial variability of soil information within soil map units (at small scales) would be necessary for adequately modeling watershed runoff. Fort Worth, Texas, implemented a low-budget GIS to evaluate field data and to develop recommendations for structural rehabilitation for a study area of approximately 80 miles of the city's sanitary sewer collection system (Hegwald et al. 1998). Martin et al. (1998) developed an implementation and response system based on GIS data to deal with pollution. Moeller et al. (1998) reviewed the application of a desktop GIS to the East Baton Rouge City, Parish (Louisiana) Sanitary Sewer Overflow Correction Action Plan. The key functionality of the GIS was in the areas of overlay analysis and network tracing.

Small et al. (1998) quantified several flooding problems of the Norfolk Naval Base in Norfolk, Virginia, by surveying and inventorying the existing storm drainage network. The hydrological model parameters were generated using GIS, and were subsequently utilized in the Extran Block of SWMM. Barnett and Fulcher (1998) integrated HSPF model with ArcView GIS to facilitate hydrologic simulation with respect to water quality, urban storm runoff, and flood damage modeling for Dardenne Creek Watershed in Missouri.

Miller and Sias (1998) described the use of simple numerical models of site hydrology, groundwater flow, and slope stability for estimating the effects of timber harvest on landslide stability. Backhaus and Braun (1998) developed and demonstrated a GIS concept based on a landuse/vegetation classification derived from landsat data, a digital elevation model (DEM), and a relief-dependent water distributin model (WDM).

Huang et al. (1999) incorporated ArcView GIS to develop a hybrid GIS-supported Watershed Modeling System (GIS—WMS) for water pollution controls. This integrated system consists of GIS-supported database, and simulation model. The GIS technology is used for managing spatial and non-spatial data, linking models, and providing interfaces between the models and user. This model was able to incorporate a variety of system components and the related submodels within a general framework, and quantify system information with GIS throughout the modeling processes.

CHAPTER 3

SOFTWARE

ArcView 3.2, Spatial Analyst and Avenue (script language for ArcView) are appropriate tools for analysis of spatial data due to a rich set of spatial functions. The inherent presence of various hydrological functions assists in delineation of watersheds, a laborious computation to be practical but a task well suited to GIS. Spatial Analyst is comparatively new for watershed delineation. It is a matter of time only when its role will increase in the water resources community like that of ArcView GIS.

Although XP-SWMM is built on SWMM computing engine but it has several unique capabilities which EPA did not develop for SWMM. XP-SWMM uses a unique XP environment to create drainage network graphically over real-world topographical background while its knowledge-based rules prevent incorrect data entry and inconsistent network. The aforementioned software and pertinent data needed to develop this interface are discussed in this chapter.

3.1 Surface Water Management Model (XP-SWMM)

Urban storm water management is dependent on the spatial variability of urban watershed characteristics. Because of these characteristics, watershed models with spatially-distributed, physically-based, lumped parameters are preferred. There are several such models, some of which are maintained by EPA and are well respected in the engineering community; Storm Water Management Model (SWMM) is one of them. SWMM is capable of representing urban stormwater runoff and combined sewer overflow phenomena, and investigating quantity and quality problems.

The original SWMM was a DOS-based, batch-processed (lumped parameter) FORTRAN program. Its successful execution depends upon correctly organizing the input. Although many modifications and improvements have been added, it still lacks the advantages of more recent computer operating systems such as Windows and information processing advances such as GIS.

18

Therefore, several software vendors in the private sector have contributed to updating SWMM by writing various graphical user interfaces (GUI).

XP Software Inc. in Australia developed a GUI named XP-SWMM, which is currently marketed by CaiCE Software Corporation (CaiCE, 1998). This GUI is a modified method of data entry and data output interpretation tool, acting as pre- and post-processors to SWMM. XP-SWMM is based on SWMM's FORTRAN engine and has the same computational procedures employed in SWMM but consists of several enhancements. It utilizes all computational and service blocks used by SWMM (Runoff, Extran, Transport, Storage, Treatment, and Receive). This research uses only Runoff Block and its nonlinear Runoff routing method to develop its various spatially-related input parameters.

3.1.1 SWMM and XP-SWMM Development History

SWMM is a comprehensive water quantity and quality simulation model used primarily for urban areas and developed for the U.S. Environmental Protection Agency in 1969-1971, through the combined endeavors of Metcalf and Eddy Inc., the University of Florida, and Water Resources Engineers Inc. Extran, one of the modules used by SWMM, had its origin in the early 1960s as a model of the San Francisco Bay. Additional capabilities were added to this block to simulate the upland area's contributing stormwater Runoff. In 1974, EPA acquired this model and incorporated it into the SWMM package, calling it the Extran, Roesner and Shubinski (1992)

The majority of the modifications after 1994 have been carried out by WP Software Ltd., Australia. The XP-SWMM-32 Versions (5.2 or higher), run under DOS or Windows 95 or Windows NT operating systems (XP-SWMM User's Manual 1998).

3.1.2 XP-SWMM Description

XP-SWMM is a link-node model with the capabilities of SWMM to simulate surface Runoff and Transport through the drainage network. As noted earlier, Runoff is one of six blocks used only for Runoff estimation. This block generates surface and subsurface Runoff based on arbitrary rainfall (and/or snowmelt) hydrographs, antecedent conditions, land use, and topography. Dry-weather flow and infiltration into the sewer system may be optionally generated using the Transport Block. The Runoff, TRANSPORT and Extended Transport (Extran) Blocks route flows through the sewer or drainage system and sophisticated hydraulic routing may be performed with Extran. The hydraulic

and pollutant routings are only available in the Transport Block. The Storage/Treatment Block characterizes the effects of control devices upon flow and quality. The Receive Block analyzes pollution effects in a receiving water body.

The graphical EXPERT Environment used in XP-SWMM is a graphics-based environment and its EXPERT shell acts as an interpreter between the user and the model. The shell incorporates both pre- and post-processors which use expert knowledge to filter the input data and create a valid model of the system. XP-SWMM has added several new features which are not available in SWMM, such as new Runoff methods in its Runoff Block. The Runoff Block consists of nine methods for computing system Runoff hydrographs: (1) Nonlinear Reservoir Runoff Routing (USEPA Runoff), (2) SCS Unit Hydrographs using a curve number with a curvilinear or triangular unit hydrograph, (3) Kinematics Wave, (4) Clarke Hydrograph, (5) Snyder Hydrograph, (6) Nash Hydrograph, (7) Rational Method, (8) Laurenson's Nonlinear Runoff Routing (RAFTS), and (9) Santa Barbara Urban Hydrograph (XP-SWMM User's Manual, 1998).

XP-SWMM's unique Expert environment helps the user to create the drainage network interactively on the screen using a mouse and toolstrip. The data are stored in a database and entered through graphical dialog boxes or optional text files. The knowledge-based rules built into the XP-SWMM continuously filter user input and prevent incorrect or inconsistent network structures. Any conflict in the data is detected at the time of data entry, before running the model, or when a full relational data check is performed. XP-SWMM has the capability to draw drainage networks on the screen over real-world topographical backgrounds or imported from a database. Output may be viewed on the screen at different scales or sent to a DXF file for further CAD manipulation.

The global database contains design and measured storm events, infiltration data, pollutant data, and other data required to run XP-SWMM. This database is connected to all of the computational blocks (Runoff, Extran, Transport, and Storage/Treatment, etc.). This database stores the commonly used catchment parameters so they can be recalled and reused throughout the XP-SWMM system. The selected data can be imported or exported to XP-SWMM through the XPX Export file which is an ASCII text file. Some data from XP-SWMM can be exported/stored in this XPX Export file.

3.1.3 XP-SWMM Runoff Block

The objective of the Runoff Block is to simulate both quantity and quality of Runoff from a drainage basin. Data for the given drainage basin are accepted by the Runoff Block in the form of aggregation of subcatchments and channels or pipes. The block utilizes an arbitrary rainfall hydrograph to make a step-by-step accounting of infiltration losses in pervious areas, surface detention, overland and channel flows, leading to calculations of inlet hydrographs.

Overland flow hydrographs can be generated by nonlinear reservoir routing using Manning's equations and lumped continuity equations with depression storage and pervious/ impervious area parameters. Urban, suburban, and rural areas of any size may be simulated using nonlinear routing. Subcatchments are modeled as idealized rectangular areas with the slope of the catchment perpendicular to the width. Each subcatchment is classified into three or four subareas. Flow from each subarea moves directly to a node and does not travel over any other subarea. The width of the pervious subareas is the entire subcatchment width; the width of the impervious subareas is in proportion to the ratio of their area to the total impervious area. Subcatchments are analyzed as spatially-lumped nonlinear reservoirs.

3.1.4 Spatial Runoff Node Data

Physical and hydrological parameters are required to produce hydrographs. Physical parameters include the area of the subwatershed and its slope. Their values are relatively easy to estimate from topographic maps, aerial photographs, and city maps and records. Hydrological parameters include width of the subwatershed, Manning's coefficients, depression storage, and infiltration rates or coefficients. These parameters are more difficult to estimate than physical ones. In conclusion, 11 parameters per subwatershed are required to simulate only the quantity of the Runoff. For each pipe or gutter that connects these subwatersheds, an additional minimum of five parameters are required. The user must also specify the meteorological data. For development of this interface between ArcView and XP-SWMM, only spatial parameters for the RUNOFF Block will be incorporated utilizing the Grid Hydrological Functions and Spatial Analyst.

3.2 Geographic Information System (GIS)

The Geographic Information System (GIS) describes various features by providing exact locations and attributes that distinguish them from their adjacent features or surroundings. GIS links land cover data to topographic (terrain modeling) and to non-topographic data (information about various processes and attributes) related to a geographic location. The nontopographic information, when applied to hydrologic systems, may consist of soils, land use, ground cover, and groundwater conditions. The advent of digital maps in GIS provides one of the latest methods of storing and retrieving topographic features and their various characteristics. This digital representation of features is stored in the computer according to some form of locational referencing system.

3.2.1 GIS Data Structure

The geometry and attributes of the data (nonspatial data) are important for the creation of a theme. The data storage requirements of nonspatial data depend on the type of data structure being used. Both data storage volume and processing efficiency of GIS depend on the data structure. There are at least two possible ways of representing topological data in GIS: raster and vector representation as described in the next paragraphs.

3.2.2 GIS Format Structure

At least three types of geometrical entities—points, line, and polygons or planes—are needed for any digital representation (data layer or theme) in a GIS. But while linking GIS and hydrological modeling, the GIS may possess six data structures. These data may be divided into two categories—basic and derived structures (Maidment, 1993). The first includes the afore-mentioned geometrical entities; the second includes grid, lattice, triangular irregular network and network. A point in a GIS is the simplest kind of geographic feature without any dimension and its only property is its location. Manholes, wells, and sampling points represent points. A line or arc is a set of connected line segments beginning and ending at a node. Sewers, links, conduits, or open channels are represented as links. The polygon is a set of connected lines which forms its boundary. Examples of polygons are the extent of watershed boundary, types of soils, and land uses in a study area. The spatial variation of hydrological parameters may be represented by grid or TIN structures. Polygons as a data structure form the basis of a lumped, spatially-variable hydrological model.

22

3.2.3 Raster Data Structures

The term raster is a German word (meaning screen) which implies one row followed by others. It is similar to the set of parallel lines making up a television picture. Raster data structure describes a matrix composed of distinct units called pixels or cells. This matrix (cellular organization) stores data by dividing the area under study into cells. Each cell represents a certain specified portion of the terrain in any area unit, i.e. square foot, acres, square miles, etc. Each pixel or cell stores a numeric value. Each cell is given a value corresponding to the feature or characteristic that is located at or describes the site, such as land use or soil type. This data structure represents single cells or cells with common attributes (cluster observations in groups of similar characters). It provides coordinates for each cell, i.e. location of a cell and not its attributes.

The raster-based systems, while representing spatial data, have limited ability to specify a location in space due to the size of the raster cells. The exact position of any location or feature is difficult to know within a raster cell because geographic specificity is limited. Entities are represented implicitly and thus raster structures are simple. Areal attribute data developed for raster-based systems may be stored in two ways: as a tag (classification) that describes each spatial unit as a whole based on the dominant attribute within it, or as a count (inventory) that lists frequency of occurrence of certain phenomena as a whole.

Several GISs use GRID, an implementation of the generic raster data structure. The locational referencing in GRID is provided by a grid referencing system which allows the information to be numerically located in space. Raster images and raster maps can be stored in a GRID. ArcView's extension, Spatial Analyst, is designed to store and manipulate raster maps, i.e., a digital elevation model (DEM) that is similar to GRID (ESRI 1999).

3.2.4 Vector Data Structures

Vector data structures are based on elemental points whose locations are known to arbitrary precision, i.e., geographic specificity is not limited. In any vector-based GIS, coordinate data for spatial data are encoded and stored as some combination of points, lines, and areas or polygons after input processing. There are several types of vector data structures—whole polygon, dual independent map encoding (DIME) file, arc-node, relational, and digital line graph (DLG). In the arc-node data structure,

23

points are elemental or basic components and arcs are individual line segments that are defined by a series of x-y coordinate pairs. Nodes are the arc ends and form the points of intersection between arcs. Polygons are areas that are completely bounded by a set of arcs.

The topological information in vector-based structures is often coded explicitly in the database. Line segments within DIME files, for example, have identification and codes for the polygon on either side. When required topological relationships are not explicitly coded in vector data structures, it can be expensive and time-consuming for the system to establish them. In the arc-node data structure, attribute values are stored together with topological information; but in the case of the relational data structure, attribute information is stored separately. This design strategy is used by ArcView GIS. Topological data in a relational structure are organized in such a way that a row in a table represents a single data value and a column represents a different field. Comparison of data volume between two structures (raster and vector) is entirely dependent on the database contents as well as consideration of accuracy and precision. An elevation data set is generally stored as a complete cellular array in a raster (DEM), and as line segments storing line locations of constant elevation (TIN, i.e. contour lines) in a vector model (Theobald and Goodchild 1990).

These are fundamentally different views of the underlying spatial information. The raster is quasi-continuous while the vector is clearly discrete. The raster representation may be considered denser than the vector because unique values are stored. Topography of a landscape can be described by many techniques in a GIS which could be used for hydrologic modeling (Maidment 1993).

The digital elevation model (DEM) is one of the techniques used in ArcView to describe terrain. The spatial attributes, connection, and inter-relation of various features on each other and on topologic data are needed to fully describe the elevation data. This research uses DEM for developing hydrological parameters for XP-SWMM as described further in the next section.

3.2.5 Digital Elevation Model (DEM)

Topographic data for an urban drainage network mainly relate to the direction of water movement and describe the ability of a watershed to store and convey Runoff. Hydrological modeling depends on a representation of land surface and subsurface (topographic data) because this is the environment through which water flows. The various parameters for hydrological modeling may be determined by

24

analysis of terrain and land cover features. The average slope, drainage path, time of concentration, flow potential energies, flow attenuation, and area of watershed contributing to a particular manhole are related to topographically derived, topologic attributes. These attributes can be extracted from raster. As noted earlier, the derived structures consist of three terrain representations: rectangular grid, TIN, and topographic contours. Although contour lines are the most common terrain representations in GIS coverages, they must be converted to grids or TINs before an automated analysis can be performed. These aforementioned attributes can be calculated by utilizing a DEM, since DEM is well suited to the extraction of such attributes (DeVantier and Feldman, 1993).

A DEM can be defined as an ordered array of numbers representing the spatial distribution of elevations above some arbitrary datum in a landscape. This spatial distribution may consist of elevations sampled at discrete points or the average elevation over a specified segment of the landscape. The DEM structure may be different if it is used for a hydrologic model than if it is used to determine only the topographic attributes of the terrain.

The most commonly used application of GIS in hydrologic modeling is the rectangular cell, whereas the grid cell is made up of regularly spaced lines and the enclosed area of each rectangular is described in terms of its center coordinates. There can be different grid scales for attributes of the terrain. For attributes that are largely homogeneous, the use of the grid resolution necessary for a DEM would require storage of large amounts of redundant data. An inherent problem in hydrologic modeling with grid DEM data is the production of nonphysical depressions due to noise in the elevation data affecting interpolation schemes used to describe variation in elevation between raster points, resulting in unwanted termination of drainage paths in pits. This problem is acute for relatively flat areas. Various techniques demonstrated for locating and removing depressions in grid DEM data by O'Callaghan and Mark (1984) and Jenson (1991) are incorporated by ESRI while developing Spatial Analyst Extension, which is used in this research.

The square grid networks cannot easily handle abrupt changes in elevation. The grid mesh size affects the outcome of results and computational efficiency. Mesh size is one reason precision is lacking in the definition of specific catchment areas. Also the computed upslope paths used in the hydrologic analysis are sometimes unrealistic because they tend to meander. Since regular grids

must be adjusted to the roughest terrain, redundancy can be significant in sections with smooth terrain (Panuska et al., 1990).

The input parameters (topographic attributes) required for the RUNOFF block of the XP-SWMM for a particular watershed (subwatershed area and roughness, slope, aspect, catchment area width, percent imperviousness, and other spatial global parameters) can be derived from DEMs. The most efficient DEM structure for estimation of these attributes is generally the grid-based method. However, Mark (1983) noted that grid structures for spatially partitioning topographic data are not appropriate for many hydrological applications. But Maidment (1993) describes that with TIN structures, there can be difficulties in determining the upslope connection of a facet and the irregularity of the TIN makes the computation of attributes more difficult than for grid-based methods.

3.3 ArcView

Traditionally, GIS has been used primarily to create maps and display spatial data. But due to spatial analysis capabilities of GIS, water resource professionals have started adopting it as a part of their own software applications. Watershed and sewer network modeling are spatially related applications and these models can also utilize spatial input data from GIS instead of maps using manual (traditional) methods. ArcView GIS has more future potential in the water resources field as it is a PC-based user friendly software.

ArcView has been continuously upgraded and many tools have been added, resulting in enhanced capabilities and its 3.1 Version was released in 1998. The introduction of its many extensions has made it a powerful and sophisticated program. These extensions include Spatial Analyst, Network Analyst, Dialogue Designer, CAD Reader, and 3D Analyst. Spatial Analyst is used in this research for developing various hydrological functions and procedures and is described in Section 3.4. ArcView is designed to work on various spatial databases and focuses on providing a mechanism for spatial query processes. It displays multiple tables of spatial data known as themes which are user-defined sets of geographic features such as grids, images, coverage, and shapefiles. For graphical features, graphical and geographical information (coordinates) are attached to each row in the relational database table.

ArcView has the ability to link documents, images, tables, text, graphs, spreadsheets, maps multimedia, and CAD drawings. It is compatible with ARC/INFO, and ArcCAD file formats, and can use AutoCAD drawings (DWGs) or interchange files (DXF files).

The internal environment of ArcView follows a similar style and concept of the Multiple Document Interface (MDI) standard adopted by other software such as Microsoft Word or Excel. The data used directly by ArcView are dBase or INFO. It uses dBASE to format files for data management and can access information from FoxPro, LOTUS 1-2-3, and Microsoft Excel applications. It can link map information to Structured Query Language (SQL) databases which are common in local governments (ORACLE, INGRES, and INFORMIX). Similar to most Windows applications, ArcView has the ability to display multiple graphic Windows, but no updating can be done on the graphical objects.

ArcView uses a spatial data file format known as shapefile. A shapefile is a digital version of a map making the basic unit of vector/raster data whereas the unit represents a set of associated data. It stores geographic features such as arc, nodes, and polygons. Label points and data editing functions are supported by shapefiles and their attributes are held in a dBASE file format.

Each feature in a shapefile represents a single feature/theme and its attributes. Shapefile stores nontopological geometry and attribute information for the spatial features in a data set, and the geometry for a feature is stored as a shape comprising of vector coordinates.

The shapefile is divided into main, index, and dBASE files. Main file is a variable-record-length file with direct access and each record describes a shape with a list of its vertices. Index file stores the index of the feature geometry, and each record contains the offset of the corresponding main file record from the start of the main file. dBASE file stores the attribute information of features. When a shapefile is added as a theme to a view, this file is displayed as a feature table.

3.4 Spatial Analyst

Spatial Analyst (ESRI 1999) is designed to extend ArcView capabilities to display and analyze raster maps. It can perform analysis on a digital elevation model. It Spatial Analyst has the capability to create and analyze grid themes representing continuous surfaces to determine the relationship between soil, terrain, runoff flow, slope, etc. For a detailed look at the relief of the landscape, Spatial Analyst can create slope, an aspect that contains important factors in determining the conveyance and direction of runoff. Spatial Analyst discovers spatial relationships in data and solves a wide range of problems in water resources field due to its capability to model raster by incorporating its built-in set of tools. This built-in set of has the potential for generating surfaces and analyzing characteristics such as slope, aspect, watershed delineation, and modeling drainage basins.

3.4.1 Integer and Floating-Point Grids

Spatial Analyst utilizes grid themes that are based upon a grid data source. Grid data sets are raster data that may consist of either integer or floating-point grid data sets. Integer grid data sets are usually used to represent features that have discrete values. With discrete data it is easy to define precisely the starting and ending point of an object. But the floating-point grid data sets represent features that have continuous values such as elevation. In a floating-point grid theme, there is continuous change in values from one location to the next. The integer data set is smaller and faster to operate on than the floating-point data set.

In the case of a grid theme based on an integer, an associated table stores the list of unique cells. When a variable with an integer is mapped, an integer grid is created which has an associated value attribute table (VAT). The VAT stores a single record for each unique value in the grid, as well as the number of cells of that value and additional attributes.

The map projection and coordinate system play an important role when analyzing the grid data. Grid themes have several dissimilarities with feature themes while considering the projection process. Grid themes cannot be projected into a different coordinate system as can feature themes. The displaying coordinate system and storing coordinate system should be the same for a grid theme. If feature themes are in decimal degrees and grid themes are in

28

projected coordinates, projection for the view must be set to the projection of grid themes. Feature themes will be projected to the same coordinates as that of grid themes. The projection and alignment of coordinates of feature and grid themes allow the performance of an integrated vector and raster analysis. If both themes have their projection in decimal degrees, the analysis can be performed without setting the projection for the view. Otherwise the feature theme is projected but not the grid theme, causing the data to appear in two separate places. This appearance and separation of data in two places prevent the performance of integrated vector and raster analysis.

3.4.2 Surface and Hydrologic Functions

Spatial Analyst has some built-in functions which use a grid theme to represent elevation. The discussion presented in this section is taken from ESRI (1999) documents related to Spatial Analyst. The assumption behind the surface-analysis functions is that additional data can be derived by analyzing DEM data and identifying patterns in existing surfaces.

The Spatial Analyst uses grid themes to support two types of surface functions—Surface Interpolators and Surface Analyzers. The first type creates a continuous surface from sampled input points values, i.e. Spline, Kriging, etc. The second type performs specified calculations resulting in different representations of a surface not apparent in the original surface, i.e. slope.

The surface function, *Slope*, identifies the slope or maximum rate of change from each cell to its neighbors. An output slope grid theme can be calculated as either a percentage of slope (% slope) or a degree of slope (degree° slope). With this function all flat areas in a hilly terrain or all areas that have the greatest risk for landslide, erosion, flooding, or the direction of Runoff flow can be shown. The Avenue Request *Slope* is used for this function.

Hydrologic functions in Spatial Analyst provide methods for describing the physical components of a surface. Using DEM as input, a drainage system can be automatically delineated and subsequently its characteristics can be quantified. Some of the applications, which can be developed by the built-in hydrologic functions are: (1) delineating a watershed and defining a pipe or channel network that crosses a surface; (2) determining the watershed

29

within a study area so that management units can be defined; and (3) estimating water Runoff for flood control purposes.

The Spatial Analyst utilizes several hydrological functions originally introduced in ARC/INFO GRID for delineating a watershed. A Flow Direction Request determines the direction of water flow. A *Sink* Request identifies any sinks in the original DEM. A *Fill* Request fills depressions. A *Watershed* Request delineates the watersheds around the specified inlets (manholes). A *Flow Accumulation* Request calculates upslope cells draining into a manhole. A *Flow Length* Request determines flow path length within a given watershed which is used to calculate the runoff travel time and runoff quantity (ESRI, 1999). Several of the above mentioned Surface and Hydrologic Functions are incorporated into various procedures developed for this research.

3.5 Avenue

Avenue is the scripting language for ArcView. It is an Object-Oriented Program (OOP). OOP is equipped with a library of classes representing objects found in ArcView and modules which work with ArcView. Avenue can execute such development tasks as developing a new interface for ArcView, customizing an existing one, automating repetitive tasks, or writing complete query analysis applications. It has three basic building blocks: classes, objects, and requests (ESRI, 1999).

A class is the most basic concept in OOP. It is a block that allows the structure to create objects and combine objects having common characteristics. An object (class instances) is a tool that can be a person, place, or thing that knows the values of its attributes and has the capability to perform operations (methods/services) on those attributes. A request will only be executed by an object if the request has a corresponding service of the same name and parameters required by the service are specified. Avenue controls an object by sending it a request which can also carry parameters. An object inherits a service from its class or superclass and a request can be made to a class for creating or obtaining information about objects (ESRI 1999).

30

CHAPTER 4

DATA DEVELOPMENT FOR GIS AND RUNOFF MODELING

The procedure developed in this dissertation uses XP-SWMM and ArcView 3.1 and these H/H and GIS applications operate independently and are linked through mutually shared exchange files, as mentioned earlier in Chapter 1. ArcView GIS is used to generate several physical and hydrological parameters of a subwatershed—area, slope, width, and imperviousness—by utilizing digitized drainage plan (runoff collection and conveying system) soil and land-use maps and the digital elevation model (DEM) of the catchment area. Manual or traditional methods were previously employed by Black & Veatch to estimate these parameters as well as other pertinent data for runoff modeling. The spatial analysis procedures of GIS cannot extract all the parameters required by watershed runoff modeling because they cannot be developed strictly from maps or GIS coverages. They are mostly indicated or input manually in the data base (GIS or other data base) and consequently exported to XP-SWMM, such as pipe shapes, dimensions, invert elevations, and kinds of pipe-material are indicated manually.

4.1 Data Sources and Development for GIS and Watershed Runoff

The input data generation by desktop GIS for hydrological and hydraulic models is effectively augmenting the traditional engineering practices of data development. The storage, preprocessing of data and spatial analytical capabilities of GIS have made it an accepted and mature technology for water resources field as it has been successfully applied for the computation of surface flow characteristics of urban and non-urban watersheds. Most of the urban catchments consist of multiple land-uses and soils causing unequal runoff contribution in terms of quantity and quality to the collection system. XP-SWMM can simulate the dynamic runoff response of urban watersheds and provides a basis for various impacts of urban stormwater management strategies. But these capabilities of XP-SWMM also demand increased efforts to satisfy its spatial database, which can

31

be partially supported by GIS technology. To meet the extensive data development requirements for XP-SWMM and to utilize the spatial analytical capabilities of ArcView, a graphical user interface (GUI) is introduced which facilitates the data entry.

The software developed in this research is applied to the Allen Creek Watershed, City of Ann Arbor, Michigan. The case study is well suited for bench marking GIS-facilitated urban watershed modeling. Because previous flood management studies have been conducted for this watershed that employed traditional manual methods to develop the input data for runoff modeling.

4.1.1 Data Development Procedures

The spatial, physical and hydraulic data needed to run GIS and XP-SWMM for verification purposes and to test the code (developed in this dissertation) are provided in this section. Also the data development procedures to construct the digitized data, (subsequently derived data) and manually estimated data are briefly discussed herein, which involves the following steps:

1. Acquisition and development of base maps, topographic maps (hydrologic/contours) soil maps, land-use maps, sewer network map (manholes, pipes, invert elevations, and other hydraulic and hydrologic data about the collection system), DEM data layers and other coverages required for urban storm water modeling.

2. Manually estimating and processing of runoff input data, parameters and development of runoff techniques suitable for input of collection system information into XP-SWMM.

3. Preprocessing of model input data, parameters and development of GIS techniques suitable for input of spatial information into XP-SWMM.

4. Interfacing ArcView GIS to XP-SWMM through development of Avenue programs for converting GIS-derived coverages to XP-SWMM input parameters.

4.1.2 Data Development and Sources for GIS and Runoff Modeling

From the base maps, the City of Ann Arbor Utilities Department developed information in ARC/INFO as derived maps. All necessary data were furnished by the City of Ann Arbor Engineering Section, which is responsible for development, storage, and updating of the data for watershed, land use, sewer, and runoff conveyance systems. The Information Services

Department of the City of Ann Arbor and Black & Veatch Environmental Engineers (1996) were the other two major contributors for the digitization process of all GIS coverage used in this study. Table 4.1 presents a summary of sources and a brief description of the contents of all information collected for the data utilized in this research.

TABLE 4.1. SOURCES OF DATA USED IN GIS AND RUNOFF MODELING

Source	Title of Documents	Information
Black & Veatch	SEWER INDEX MAPS	Street maps, water bodies, drainage boundaries,
City of Ann Arbor Utilities Department	WATER INDEX MAPS of Ann Arbor Township	Surface and invert elevations for sewers
	City of Ann Arbor, Base Map, Land Uses	
Soil Conservation Service (SCS)	Soil Survey of Washtenaw County, Michigan	Soil Survey of Ann Arbor area
U.S. Geological Survey	Ann Arbor East and West Quadrangles to develop Digital elevation model (DEM)	Quadrangle maps of Ann Arbor area

4.2 Software and Data Requirements

In this section the data requirements for XP-SWMM and ArcView are discussed. The Section 4.2.1 provides the sources of the GIS and XP-SWMM input data used in this study. The Section 4.2.2 discusses the manually developed data related to watershed and collection system, which are needed by XP-SWMM to estimate urban runoff quantity. Section 4.2.3 considers information about various coverages (maps) required by ArcView GIS to derive the necessary runoff data. ArcView provides the capability to create several runoff parameters by incorporating Spatial Analyst, its script language (Avenue), digitized maps, and DEM of the watershed.

4.2.1 XP-SWMM and Overland Flow Parameters

The Runoff Block simulates quantity and quality of runoff from a drainage basin. The mathematical details and various parameters required for the runoff simulation are provided in Chapter 5. Calculated rainfall excess is routed as surface flow by the kinematic wave approximation. The data for the given watershed is accepted by the Runoff Block in the form of aggregation of subcatchments and channels or pipes. It utilizes an arbitrary rainfall hydrograph to make a step by step accounting of infiltration losses in pervious areas, surface detention, overland flow and channel flow, leading to the calculations of inlet hydrographs. Overland flow hydrographs are generated by non-linear reservoir routing method coupling Manning's equations and continuity equation and the change in reservoir volume is the result of net inflow, as shown in Equation 4.1:

$$\frac{\partial (V)}{\partial t} = A \frac{\partial d}{\partial t} = AR_{excess} - Q \tag{4.1}$$

where

V = volume of water (cubic-feet);

t = time (seconds);

d = depth of water (foot);

A = Surface Area (square-foot);

R_{excess} = rainfall excess (foot/second); and

Q = overland flow (cubic-foot/second) from a subcatchment.

Overland flow (runoff) Q generated can be calculated by the use of Manning's equation. The momentum equation is approximated by assuming bed slope equals friction slope, with the latter governed by Manning's equation, as follows in Equation 4.2:

$$Q = \frac{1.49 \; W}{n} (d - d_p)^{5/3} S^{1/2} \tag{4.2}$$

where

d_p = depth of depression storage, foot;

S = Slope of the subcatchment, foot/ foot;

n = Manning's coefficient, and

W = is one-dimensional, overland flow width.

If Equations 4.1 and 4.2 are combined, Equation 4.3 can be obtained in following format:

34

$$\frac{d\,(d)}{dt} = R_{excess} - C(d - dp)^{5/3} \qquad (4.3)$$

where

$$C = \frac{1.49\ W\ S^{1/2}}{A\ n} = \text{Routing Parameter} \qquad (4.4)$$

4.2.1.1 Routing Parameter and Spatial Data Requirements. The Routing Parameter (C) is a term introduced in EPA's SWMM Manual to define a single parameter consisting of four variables, W, S, n, and A of a subcatchment. Equivalent changes can be caused by appropriate alteration of any one of the four parameters while conducting the calibration process. The reader is referred to Huber et al. (1988) for further studies.

As shown in Equation 4.4, the mathematical representation of Routing Parameter C is described by four physical and hydrological parameters which are required to produce the overland flow or hydrographs. Physical parameters include the area of the sub-watershed, and its slope. Their values are relatively easy to estimate from topographic maps, aerial photographs, city maps, and records. Hydrologic parameters include the width of the subwatershed, and Manning's roughness. These parameters are more difficult to estimate than the physical ones. Three of these variables can be derived from GIS coverages but Manning's roughness cannot be developed strictly from maps and plans or GIS coverages.

4.2.1.2 Data for Allen Creek Watershed and Its Runoff Conveyance System. The values of three variables—area, width, and slope—of the Routing Parameter were estimated manually for this test watershed so that the values of these variables by GIS tools could be verified. The watershed data were developed to conform to input requirements of XP-SWMM.

Black and Veatch used a labeling scheme, while developing hydrologic and hydraulic data for the Allen Creak Watershed and its stormwater conveyance system. This labeling scheme adheres to the system presently used for the City of Ann Arbor's sanitary sewer system. The scheme consists of a unique 10-digit alphanumeric label for each conduit and node as shown in Table 4.2 and Figure 4.1. Labels are ten-digit alphanumeric codes, such as **09292004RO,** which describe the structure and conduit downstream from the structure. The first five digits identify the location of the

structure (township, section, and quarter section) and the next three digits provide a structure number. Fields 9 and 10 are added to comply with the need of the XP-SWMM model.

TABLE 4.2. LABELING SCHEME FOR CONVEYANCE SYSTEM

Serial No.	Digit	Description
1	1–2	Township Number
2	3–4	Section Number
3	5	1 / 4 (Quarter) Section Number
4	6–8	Structure Number
5	9	Type of Structure
6	10	Identifier for Manholes and "Dummy Nodes"

The Allen Creek Watershed is one of the watersheds in the City of Ann Arbor. The data for this watershed were developed manually by Black & Veatch and subsequently arranged for input in XP-SWMM. The stormwater conveyance system modeled generally contiguous segments of 36-inch and larger underground system facilities and open channels. This conveyance system consist of subbasins, conduits (pipes), nodes (manholes). The final runoff exit point for this conveyance system falls into the Huron River. The Allen Creek Watershed has an area of 3, 398 acres and it consists of 46 subbasins as shown in Figure 4.2.

Ten-Digit Labeling Scheme for Nodes and Conduits		
09292	004	RO

Five-Digit Identifier for:
 	
Township (1st 2 Digits);
 	
Section (3rd and 4th Digits); and
 	
1/4 Section (5th Digit) Where Structure is Located
	Three-Digit Structure Number	Identifier Code Indicating Structure Type and Number as Follows:
		R – Denotes Manhole Number
		D – Denotes Dummy
		     Structure Number
		C – Denotes Conduit Downstream
		     From Structure Number
		1–9 Identifier Number for
		     Dummy Structure

Figure 4.1. Labeling Scheme for Stormwater Conveyance System

The conveyance system (manholes and pipes) and contributing subwatersheds (runoff collection areas around a manhole), described in Figure 4.2, for the Allen Creek Watershed were developed manually by hydrologists at the Black & Veatch Environmental Office. The delineation process employed on the selected subbasins was based on the traditional methods adopted by the hydrologic community (Black & Veatch, 1996). From hydrologic maps, subcatchment areas were computed and then slope and width parameters were estimated using traditional mathematical techniques. The manually estimated values for area, slope, and width of the 46 subwatersheds for the Allen Creek Watershed are given in Table 4.3. Additional data (explained in Section 5.1.2 and Table 5.1) for the watersheds and structures were collected to aid in development of the stormwater model. These data consist of land use, soil characteristics, depression storage, Manning's roughness and pertinent hydraulic data for the network and are given in Table 4.4.
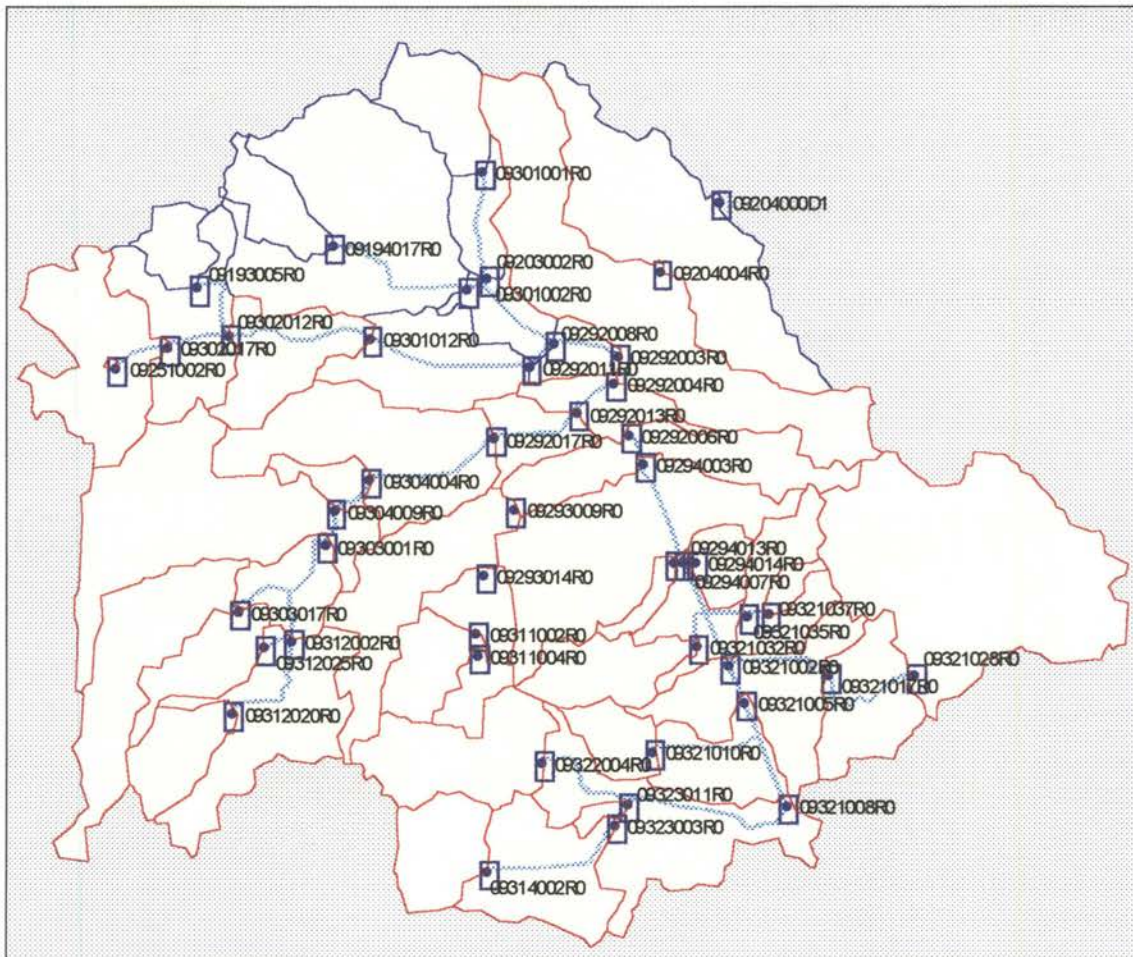
Figure 4.2. Allen Creek Watershed 46 Manually-Drawn Subcatchments,
Runoff Collection (Inlets) and Conveyance (Pipes) System.

TABLE 4.3. MANUALLY-DEVELOPED DATA OF 46 SUB-BASINS OF ALLEN CREEK WATERSHED

Serial No.	Node Name	Area (A_{man}) (ac)	Width (W_{MAN}) (ft)	Slope (S_{MAN}) (%)
1	09193005R0	26.17	1462	0.04241
2	09194017R0	59.20	2500	0.06597
3	09203002R0	109.08	2520	0.07620
4	09203013R0	52.30	1875	0.08295
5	09204000D1	227.50	2840	0.07158
6	09204004R0	193.12	2844	0.05574
7	09251002R0	30.01	1160	0.06535
8	09292003R0	49.10	1000	0.06455
9	09292004R0	40.24	1840	0.05866
10	09292006R0	49.63	5670	0.05572
11	09292008R0	60.66	3500	0.06253
12	09292011R0	69.09	4590	0.07451
13	09292013R0	38.19	3000	0.06870
14	09292017R0	114.50	2913	0.07062
15	09293009R0	78.47	1625	0.06590
16	09293014R0	47.82	2400	0.06995
17	09294003R0	185.35	3000	0.04962
18	09294007R0	105.41	2660	0.05280
19	09294013R0	20.03	1330	0.07372
20	09294014R0	33.28	575	0.03822
21	09301002R0	126.08	4950	0.08171
22	09301012R0	68.07	3840	0.08481
23	09302012R0	46.27	2000	0.05460
24	09302017R0	98.25	1530	0.04676
25	09303001R0	66.83	3420	0.07353
26	09303017R0	89.18	6318	0.07041
27	09304004R0	45.92	1360	0.06359
28	09304009R0	201.51	2000	0.05892
29	09311002R0	61.95	1200	0.05578
30	09311004R0	15.01	1494	0.04629
31	09312002R0	90.79	3420	0.05099
32	09312020R0	79.11	2000	6.62600
33	09312025R0	22.60	4600	0.04914
34	09314002R0	51.31	2700	0.04957
35	09321002R0	84.06	3230	0.03508
36	09321005R0	52.17	2700	0.07777
37	09321008R0	86.81	3800	0.06552
38	09321010R0	31.24	2900	0.05610
39	09321017R0	79.39	3800	0.04198
40	09321028R0	101.96	5890	0.03240
41	09321032R0	27.94	3000	0.07625
42	09321035R0	7.67	330	0.06420
43	09321037R0	22.95	1346	0.04753
44	09322004R0	99.50	5160	0.04693
45	09323003R0	104.16	4725	0.04591
46	09323011R0	34.75	2554	0.04858

TABLE 4.4. MANUALLY-DEVELOPED INPUT DATA FOR RUNOFF BLOCK OF XP-SWMM FOR 46 SUBCATCHMENTS OF ALLEN CREEK WATERSHED

```
A1 'Allen Watershed - 10yr/24hr Storm Event',
B1 0, 0, 1, 0, 0, 0, 0, 0, 1, 7, 95,
B2 0,1, 0,
B3 60., 120., 1440., 2, 24,
B4 0.0,0.0,
D1 2,
E1 0, 10, 0, 0, 0, 1, 96, 15, 0,
E3 .0099, .0099, .0099, .0099, .0099, .0099, .0099, .0099, .0117, .0117,
E3 .0117, .0117, .0117, .0117, .0117, .0117, .0144, .0144, .0144, .0144,
E3 .0144, .0144, .0144, .0144, .0162, .0162, .0162, .0162, .0198, .0198,
E3 .0198, .0198, .0234, .0234, .0255, .0255, .0285, .0285, .0324, .0324,
E3 .0414, .0414, .0558, .0558, .0864, .0864, .2664, 1.1016, .1296, .1296,
E3 .0666, .0666, .0486, .0486, .0378, .0378, .027, .027, .027, .027,
E3 .027, .027, .027, .027, .0162, .0162, .0162, .0162, .0162, .0162,
E3 .0162, .0162, .0162, .0162, .0162, .0162, .0162, .0162, .0162, .0162,
E3 .0108, .0108, .0108, .0108, .0108, .0108, .0108, .0108, .0108, .0108,
E3 .0108, .0108, .0108, .0108, .0108, .0108,
H1 1, '09311004R0#1', '09311004R0', 25.0, 1494., 15.01, 35., .04629, 0.02, 0.3, 0.06, 0.2, 2., .6, .00115, 0,
H1 1, '09311002R0#1', '09311002R0', 25., 1200., 61.95, 36.71, .05578, .02, .3, .06, .2, 2.0, 0.6, .00115, 0,
H1 1, '09293014R0#1', '09293014R0', 25., 2400., 47.82, 33.78, .06995, .02, .3, .06, .2, 2.0, 0.60, .00115, 0,
H1 1, '09293009R0#1', '09293009R0', 25., 1625., 78.47, 29.846, .0659, .02, .3, .06, .2, 2.0, 0.60, .00115, 0,
H1 1, '09294014R0#1', '09294014R0', 25., 575., 33.28, 41.1076, .03822, .02, .3, .06, .2, 1.9998, .5999, .00115, 0,
H1 1, '09294013R0#1', '09294013R0', 25., 1330., 20.03, 66.0041, .07372, .02, .3, .06, .2, 2., .6, .00115, 0,
H1 1, '09321037R0#1', '09321037R0', 25., 1346., 22.95, 23.0916, .04753, .02, .3, .06, .2, 2.0, 0.60, .00115, 0,
H1 1, '09321032R0#1', '09321032R0', 25.0, 3000., 27.94, 52.13, .07625, 0.02, 0.3, 0.06, 0.2, 2.0, .6, 0.00115, 0,
H1 1, '09321035R0#1', '09321035R0', 25., 330., 7.67, 54.9036, .0642, .02, .3, .06, .2, 2.002, .6001, .00115, 0,
H1 1, '09314002R0#1', '09314002R0', 25., 2700., 51.31, 32.412, .04957, .02, .3, .06, .2, 1.1860, .3674, .00115, 0,
H1 1, '09323003R0#1', '09323003R0', 25., 4725., 104.16, 16.384, .04591, .02, .3, .06, .2, .9898, .3969, .00115, 0,
H1 1, '09322004R0#1', '09322004R0', 25., 5160., 99.5, 35.664, .04693, .02, .3, .06, .2, 1.7318, .5234, .00115, 0,
H1 1, '09323011R0#1', '09323011R0', 25., 2554., 34.75, 25.9045, .04858, .02, .3, .06, .2, 1.2406, .3830, .00115, 0,
H1 1, '09321008R0#1', '09321008R0', 25.0, 3800., 86.81, 16., .06552, 0.02, 0.3, 0.06, 0.2, 1.9444, .6989, .00115, 0,
H1 1, '09321010R0#1', '09321010R0', 25.0, 2900., 31.24, 42.0693, .0561, 0.02, 0.3, 0.06, 0.2, 1.9711, .5917, .00115, 0,
H1 1, '09321005R0#1', '09321005R0', 25.0, 2700., 52.17, 24.4166, .07777, 0.02, 0.3, 0.06, 0.2, 1.9877, .5965, .00115, 0,
H1 1, '09321028R0#1', '09321028R0', 25., 5890., 101.96, 37.5379, .0324, .02, .3, .06, .2, 2., .6, .00115, 0,
H1 1, '09321017R0#1', '09321017R0', 25., 3800., 79.39, 57.4565, .04198, .02, .3, .06, .2, 2., .6, .00115, 0,
H1 1, '09321002R0#1', '09321002R0', 25., 3230., 84.06, 38.3618, .03508, .02, .3, .06, .2, 2., .6, .00115, 0,
H1 1, '09294007R0#1', '09294007R0', 25., 2660., 105.41, 47.0957, .0528, .02, .3, .06, .2, 2.0, 0.60, .00115, 0,
H1 1, '09294003R0#1', '09294003R0', 25., 3000., 185.35, 47.3057, .04962, .02, .3, .06, .2, 2.1202, .6421, .00115, 0,
H1 1, '09292006R0#1', '09292006R0', 25., 5670., 49.63, 61.0449, .05572, .02, .3, .06, .2, 2.3104, .7086, .00115, 0,
H1 1, '09303017R0#1', '09303017R0', 25., 6318., 89.18, 42.9784, .07041, .02, .3, .06, .2, 2.1902, .8335, .00115, 0,
H1 1, '09312020R0#1', '09312020R0', 25.0, 2000., 79.11, 37., 6.626, 0.02, 0.3, 0.06, 0.2, 3.0, 1.0, .00115, 0,
H1 1, '09312025R0#1', '09312025R0', 25.0, 4600., 22.6, 67.2552, .04914, 0.02, 0.3, 0.06, 0.2, 2.5, .8, .00115, 0,
H1 1, '09312002R0#1', '09312002R0', 25.0, 3420., 90.79, 54.9633, .05099, 0.02, 0.3, 0.06, 0.2, 1.8132, .544, 0.00115, 0,
H1 1, '09303001R0#1', '09303001R0', 25., 3420., 66.83, 56.5172, .07353, .02, .3, .06, .2, 1.9998, .5999, .00115, 0,
H1 1, '09304009R0#1', '09304009R0', 25., 2000., 201.51, 66.9916, .05892, .02, .3, .06, .2, 1.925, .6041, .00115, 0,
H1 1, '09304004R0#1', '09304004R0', 25., 1360., 45.92, 36.6267, .06359, .02, .3, .06, .2, 2.1195, .6080, .00115, 0,
H1 1, '09292017R0#1', '09292017R0', 25., 2913., 114.50, 33.1999, .07062, .02, .3, .06, .2, 2.041, .6027, .00115, 0,
H1 1, '09292013R0#1', '09292013R0', 25., 3000., 38.19, 32.6698, .0687, .02, .3, .06, .2, 2., .6, .00115, 0,
H1 1, '09292004R0#1', '09292004R0', 25., 1840., 40.24, 73.9208, .05866, .02, .3, .06, .2, 2.7284, .8549, .00115, 0,
H1 1, '09203013R0#1', '09203013R0', 25., 1875., 52.3, 37.02, .08295, .02, .3, .06, .2, 2., .6, .00115, 0,
H1 1, '09203002R0#1', '09203002R0', 25., 2520., 109.08, 33.542, .07620, .02, .3, .06, .2, 1.9998, .5999, .00115, 0,
H1 1, '09194017R0#1', '09194017R0', 25., 2500., 59.20, 32.232, .06597, .02, .3, .06, .2, 1.9998, .5999, .00115, 0,
H1 1, '09301002R0#1', '09301002R0', 25., 4950., 126.08, 31.8168, .08171, .02, .3, .06, .2, 2.9, 1.57, .00115, 0,
H1 1, '09193005R0#1', '09193005R0', 25., 1462., 26.17, 32.574, .04241, .02, .3, .06, .2, 2.00, 0.60, .00115, 0,
H1 1, '09251002R0#1', '09251002R0', 25., 1160., 30.01, 49.5176, .06535, .02, .3, .06, .2, 1.6947, .4728, .00115, 0,
H1 1, '09302017R0#1', '09302017R0', 25., 1530., 98.25, 55.5691, .04676, .02, .3, .06, .2, 1.4666, .3982, .00115, 0,
H1 1, '09302012R0#1', '09302012R0', 25., 2000., 46.27, 30.3598, .0546, .02, .3, .06, .2, 1.309, .3927, .00115, 0,
H1 1, '09301012R0#1', '09301012R0', 25., 3840., 68.07, 34.1977, .08481, .02, .3, .06, .2, 1.9244, .5773, .00115, 0,
H1 1, '09292011R0#1', '09292011R0', 25., 4590., 69.09, 39.1306, .07451, .02, .3, .06, .2, 2., .6, .00115, 0,
H1 1, '09292008R0#1', '09292008R0', 25., 3500., 60.66, 33.9715, .062525, .02, .3, .06, .2, 2., .6, .00115, 0,
H1 1, '09292003R0#1', '09292003R0', 25., 1000., 49.10, 57.5677, .06455, .02, .3, .06, .2, 2.2308, .6808, .00115, 0,
H1 1, '09204004R0#1', '09204004R0', 25., 2844., 193.12, 51.2422, .05574, .02, .3, .06, .2, 2.3338, .7168, .00115, 0,
H1 1, '09204000D1#1', '09204000D1', 25., 2840., 227.5, 52.1376, .07158, .02, .3, .06, .2, 2.6720, .8105, .00115, 0,
M1 0, 0,
```

4.2.2 ArcView GIS Coverages Required As Input Data For Spatial Analysis

In this study, ArcView's environment is primarily intended to provide interactive control of the GIS rather than to serve as modeling platforms. The capabilities of ArcView for importing DEM, sewer maps (manholes, and conduits or open channels), soil, and land use maps are exploited to develop runoff parameters.

4.2.2.1 Digitized Data for ArcView for Derivation of Runoff Parameters. The four runoff parameters for a subcatchment generated in this study by ArcView GIS are: (1) Area, (2) Width, (3) Percent Slope, and (4) Percent Imperviousness. The following five GIS base coverages, shown in Figures 4.3, 4.4, 4.5, 4.6 and 4.7, are the minimum input maps to furnish the above mentioned watershed runoff parameters:

1. Digital elevation model (DEM) of the watershed (Figure 4.3).
2. Drainage network consisting of two coverages (pipes, and manholes) (Figures 4.4 & 4.5).
3. Land use map (roads, houses, parks, waterbodies) of watershed (Figure 4.6), and
4. Soil coverage of the watershed (Figure 4.7).

Some of the above mentioned data were developed in ARC/INFO by the City of Ann Arbor Utilities Department. But the Information Services Department of the City of Ann Arbor and Black & Veatch Environmental Engineers are the two major contributors for the digitization process of GIS coverages used in this study. Black & Veatch prepared the DEM for Allen Creek Watershed, shown in Figure 4.3. The DEM was imported into ArcView and saved as Grid file and subsequently utilized in this research.

This DEM for Allen Creek Watershed consists of 500 by 476 cells and each of the cells has vertical data. This DEM was developed from a topographic map having a contour interval of 10 feet and at a scale of 1:24,000. The final product is a digital representation of the elevation over the entire Allen Creek Watershed, projected to 476 x 500 cells with a cell size of 42.08 x 42.08 foot-square. Modeling the Ann Arbor Information Services Division provided the land use coverage presented in Figure 4.6. The soil data coverage presented in Figure 4.7 was obtained from the Michigan Department of Natural Resources and the Soil Conservation Service in digital format and was converted to ARC/INFO GIS coverage.
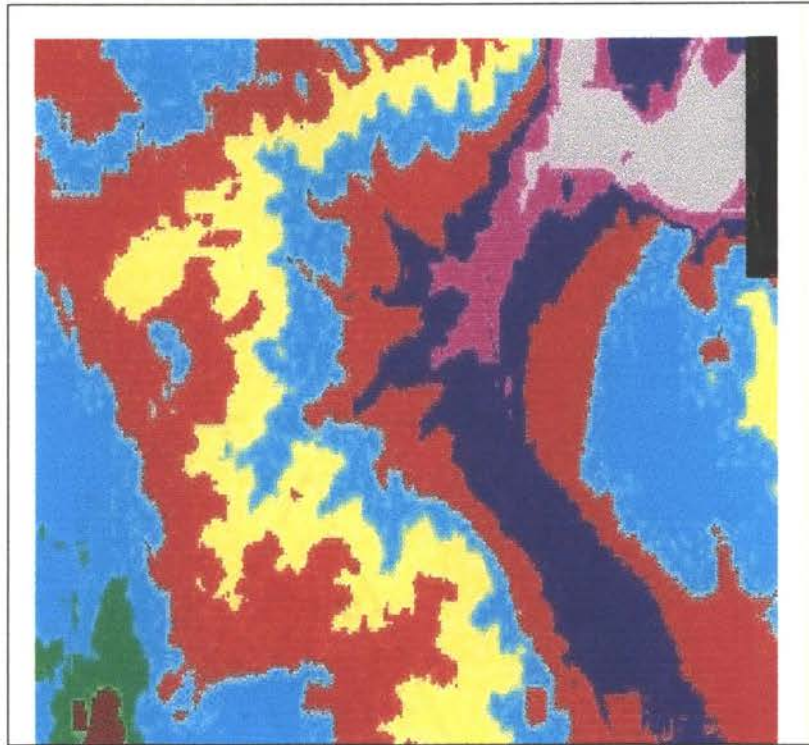
Figure 4.3. Primary Input Data Theme Representing Digital
Elevation Model (DEM) of Allen Creek Watershed
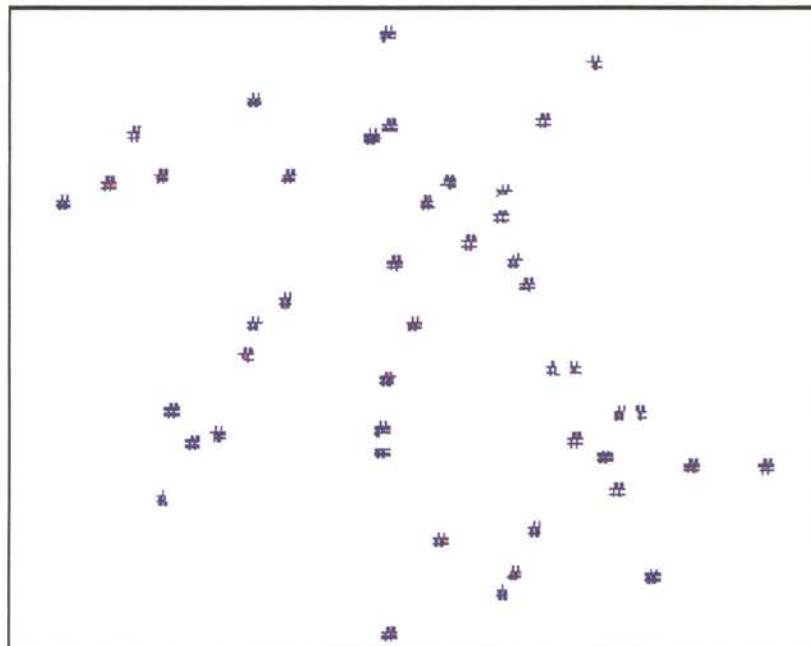
Figure 4.4. Primary Input Data Theme for Nodes Showing
46 Manholes for Allen Creek Watershed

41
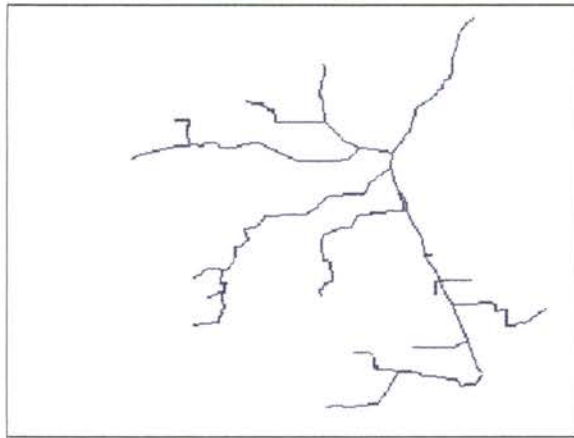
Figure 4.5. Primary Input Data Theme of Links/Pipes
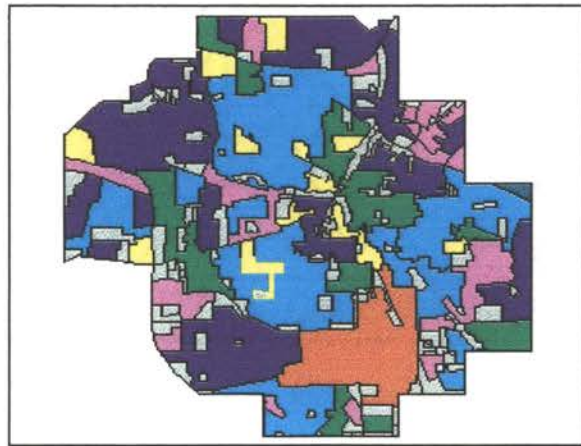
Figure 4.6. Primary Input Data Theme of Land Use
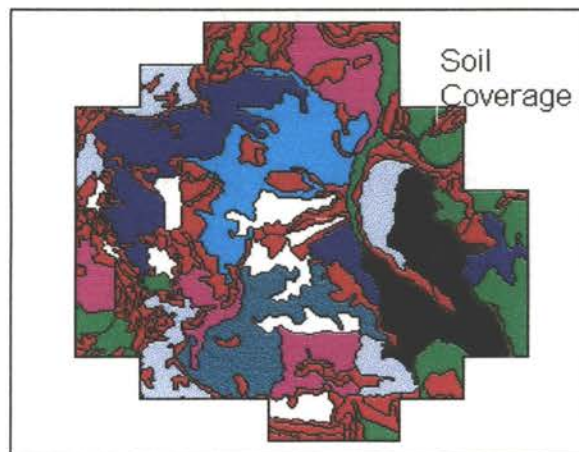
Figure 4.7. Primary Input Data Theme of Soils

CHAPTER 5

METHODOLOGY AND PROCEDURES

The principal task of this research is to use ArcView 3.1, a desktop GIS, to generate an input data group related to the catchment surface characteristics. If the necessary input data for GIS, such as DEM, soil and land-use maps and drainage plan of the watershed are digitized, and if the procedures developed in this GUI are utilized, it is possible to have GIS correctly calculate runoff parameters of any urban or non urban watershed.

This GUI exploits the capabilities of Avenue to convert ArcView's *shape files* into XP-SWMM import files. The data from XP-SWMM, after numerical modeling, is sent back to ArcView via *XPX* export files, which create various *Shape* Files. The *XPX* output files read network geometry, parameters, simulation results and subsequently writes them into shape's attribute tables. Ultimately, data export procedures display the output in the form of charts, GIS themes, tables, and profiles. The Operational Manual and Avenue Codes developed in this study, are listed in the Appendixes A and B, at the end of this thesis.

5.1 Description of Runoff Block

As described earlier in the preceding two chapters, XP-SWMM simulates urban surface runoff and transports it through the drainage network. Its Runoff Block is used for runoff estimation and is described in Section 5.1. Several of the hydrologic parameters required by Runoff Block are calculated from the DEM. Water accumulates at the ground surface when the precipitation rate exceeds the infiltration rate. Runoff occurs after surface irregularities (static surface storage) are filled and gravitational force on that water overcomes surface traction, keeping it in static equilibrium. Surface runoff is diverted by the terrain into discrete stream channels and man-made conduits such as rivers reaches, diversions ditches, sewer pipe networks, CSOs and runoff collection and conveyance system that transport the water downstream.

43

The Runoff Block uses a deterministic modeling strategy to account for influences on the hydrologic cycle (terrain slope, surface roughness, and infiltration) and consequently requires information about the environment. The model upon which this block is based provides results at different accuracies. Accurate simulations usually need enhanced data than do less accurate simulations (Huber et al., 1988).

The use of SWMM requires that much of the calibration occur in the Runoff Block because it is the first block that generates hydrographs on each subwatershed. The estimation of various parameters is often tedious because of the lack or uncertainty of maps, plans, projections, data, measurements, records, and time. There are five hydrograph generation techniques available in the Runoff Block. Several of the runoff parameters are common for these techniques. But this study uses only the Runoff Nonlinear Reservoir Method and some of its input parameters will be generated through GIS spatial analysis using a digital elevation model (DEM).

5.1.1 Runoff Block Analysis

The Runoff Nonlinear Reservoir Technique models a watershed catchment as a reservoir. The net inflow (precipitation, snowmelt, and groundwater flow) creates the reservoir volume; evaporation, infiltration, and stream flow from the outlet of the catch-ment create reservoir outflows. The catchment is treated as if all surface storage is spatially lumped in a tank with flows not dependent upon spatial location. Such a single-reservoir watershed is depicted in Figure 5.1.

A watershed comprised of multiple subcatchments can be treated as a network of interconnected reservoirs. Within each subcatchment, parameters describing the flow mechanisms are assumed uniform (i.e., roughness, soils, infiltration, etc). This lumped approach admits only variation between subcatchments. The Runoff Block flow routing is modeled after a system on interconnected reservoirs, as shown in Figure 5.2.

For the reservoir of Figure 5.1, the change in reservoir volume is the result of net inflow Q as shown in Equation 5.1:

$$\frac{d(V)}{dt} = A\frac{\partial d}{\partial t} = A R_{excess} - Q \qquad\qquad 5.1$$

and

$$R_{excess} = E_{ps} - e - f \qquad\qquad 5.2$$

where

V	= volume of the reservoir (cubic feet);
A	= surface area of the catchment (sq ft);
t	= time (sec);
R_{excess}	= rainfall excess = rainfall intensity minus evaporation-infiltration rate (ft/sec);
E_{ps}	= precipitation and snowmelt;
e	= evaporation;
f	= denotes infiltration and seepage; and
Q	= shows outflow from reservoir.
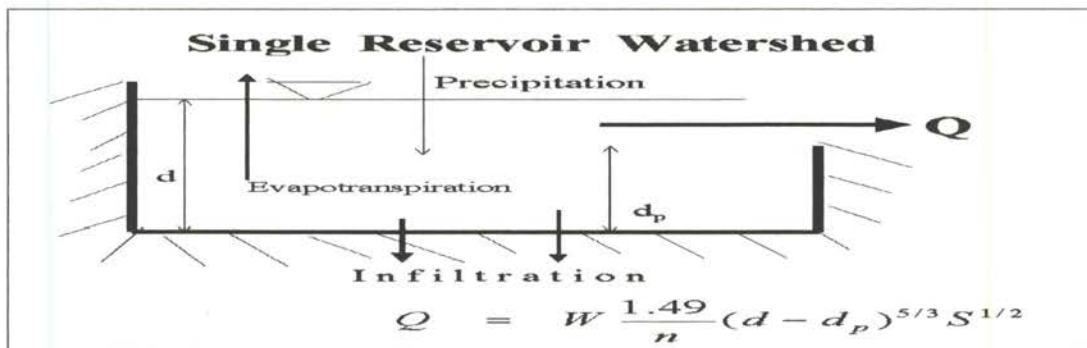
Figure 5.1. Nonlinear Reservoir Representation of a Subwatershed
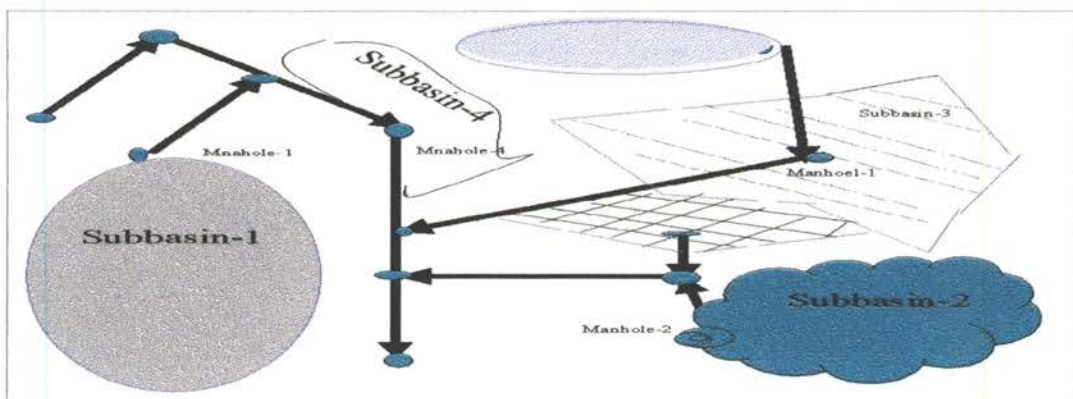
Figure 5.2. Multiple Watersheds and Runoff Collection Network Consisting of Manholes (Nodes) and Pipes or Branches (Conduits)

The Runoff Block generates overland flow from each of the subcatchment by approximating it as nonlinear reservoir. The nonlinear reservoir flow is derived by coupling the continuity equation with Manning's equation, assuming a wide channel configuration for overland flow:

$$\frac{d(d)}{d(t)} = R_{excess} - \frac{1.49\, W\, S^{1/2}}{A\, n} (d - d_p)^{5/3} \qquad 5.3$$

where

d = water depth (ft);

W = subcatchment width (ft);

n = Manning's roughness coefficient;

d_p = depth of depression storage (ft); and

S = subcatchment slope.

Equation 5.3 is solved at every time step by means of a finite difference technique. The rainfall excess R_{excess} is given in the program as a time-step average. If subscripts i-1 and i denote the beginning and end of a time step, then Equation 5.3 is approximated by the implicit, non-linear finite difference equation as 5.4:

$$\frac{d_{i-1} - d_i}{\Delta t} = R_{excess} + \frac{1.49\, W\, S^2}{A\, n} \left[d_i + \frac{1}{2}(d_{i-1} - d_i) - d_p \right]^{\frac{5}{3}} \qquad 5.4$$

where $\Delta t = t_2 - t_1$ is time step (sec). Equation 5.4 is then solved for d_2 using a Newton-Raphson iteration scheme. The outflow is modeled using a variation of Manning's equation for overland flow. Given d_2, the instantaneous outflow at the end of each time step is computed using Manning's equation as shown in Equation .5.5:

$$Q = V A = \frac{1.49\, W\, S^{\frac{1}{2}}}{n} (d - d_p)^{\frac{5}{3}} \qquad 5.5$$

where Q is runoff flow rate (cu ft/sec), and V is velocity of flow (ft/sec). Given the initial conditions, output and depth of flow can be determined from the numerical solution of the finite difference equations. Flow through pipes, gutters, and other manmade or natural channels are handled in a commensurate fashion. Figure 5.2 shows a network of reservoirs. The flow description is based on the solution of a stream of finite difference equations. Reader is referred to Huber et al. (1988) for details.

5.1.2 Data Requirements and Development for Runoff Block

The Runoff Block simulates both the quantity and quality of runoff from a drainage basin. Data for a given drainage basin are accepted by it in the form of an aggregation of subcatchments and channels and pipes. It utilizes an arbitrary rainfall of hydrographs to make a step-by-step accounting of infiltration losses in pervious areas, surface detention, overland and channel flows.

Data preparation for this block is divided into four types of data: (1) general input and control, (2) precipitation, (3) surface quantity, and (4) print control. Precipitation data, in the form of hydrographs for the duration of the simulation, can be obtained from on-site gages or from the nearest National Weather Service stations. Surface quantity or flow routing data are usually acquired from topographic maps, aerial photos, and drainage systems plans. Table 5.1 shows a sequential listing of input data required to run the Runoff Block.

TABLE 5.1. SUMMARY OF INPUT DATA FOR
RUNOFF BLOCK FOR FLOW ANALYSIS

Item No.	Data Group	Description
1	SW	Number of blocks to be run and JIN and JOUT interface file unit numbers
2	MM	Scratch file unit numbers
3	$RUNOFF	Starts Runoff simulation
4	A1	Descriptive titles–2 lines
5	B1–B4	Control parameters
6	D1	Precipitation control
7	E1–E3	Precipitation data
8	G1	Channel or pipe data
9	H1	Subcatchment surface data
10	M1–M3	Print control input
11	$ENDPROGRAM	Ends SWMM simulation

Descriptions of the input data guidelines can be found in the SWMM or XP-SWMM User's Manual (XP Software, 1998). The subcatchment data, identified by Group H1, is the main focus of this study and is necessary for the Input file of the Runoff Block. These data (Group H1), described in Table 5.2, contain the following 14 parameters and are repeated for each subcatchment. The description of these parameters is given in the User's Manual for XP-SWMM.

47

TABLE 5.2. DESCRIPTION OF SUBCATCHMENT GROUP IDENTIFIER H1
KNOWN AS SUBCATCHMENT SURFACE DATA

Item No.	Parameter	Description
1	JK	Hydrographs
2	NAMEW	Subcatchment number (1-9999)
3	NGTO	Channel or pipe or inlet number for drainage. A maximum of five different catchments may feed to single node
4	WW1	Width of subcatchment (overland flow width) in feet.
5	WAREA	Area of subcatchment, acres
6	WW3	Percent hydraulically effective impervious area of the subcatchment
7	WSLOPE	Ground slope (ft/ft)
8	WW5	Impervious area Manning's roughness
9	WW6	Pervious area Manning's roughness
10	WW7	Impervious area depression storage (in.)
11	WW8	Pervious area depression storage, (in.)
12	WW9	Infiltration parameter (Horton, or Green-Ampt)
13	WW10	Infiltration parameter (0 = minimum infiltration rate or saturated hydraulic conductivity = 1)
14	WW11	Decay rate in Horton equation or initial moisture deficit for soil or volume of air/volume voids

As shown in Tables 5.1 and 5.2, the Runoff Block requires two types of parameters to produce the hydrographs—physical and hydrological. Physical parameters include subwatershed area and its slope, pipe length, and diameter. These values are relatively easy to estimate from topographic maps, aerial photographs, city maps and records. Hydrologic parameters include subwatershed width, Manning's coefficients, depression storage, and infiltration rates or coefficients. These parameters are more difficult to estimate than physical ones. Finally, 11 parameters per subwatershed are required to simulate only the runoff quantity (in the case of H1 data, Item Nos. 3 to 14 are given in Table 5.2). For each pipe or gutter that connects these subwatersheds, an additional minimum of 5 parameters are required (G1 data as shown in Table 5.1), and the user must specify the meteorological data (data provided in Table 4.5).

In the case of the Runoff Block, the input data are needed in the following format: Subcatchment Flag; Subcatchment Area; Subcatchment Percent Impervious; Subcatchment Width

and Subcatchment Slope, etc. Figure 5.3 (Courtesy of XP Software Inc.) shows the dialog box which allows the entry of some of the subcatchment surface data (four parameters only) to calculate runoff at each node. In this study, these four parameters are generated automatically from DEM, network (node & conduit), soil and land use themes. The description of these four subcatchment parameters (area, percent imperviousness, width, and slope) is provided in subsections 5.1.2.1 through 5.1.2.4.
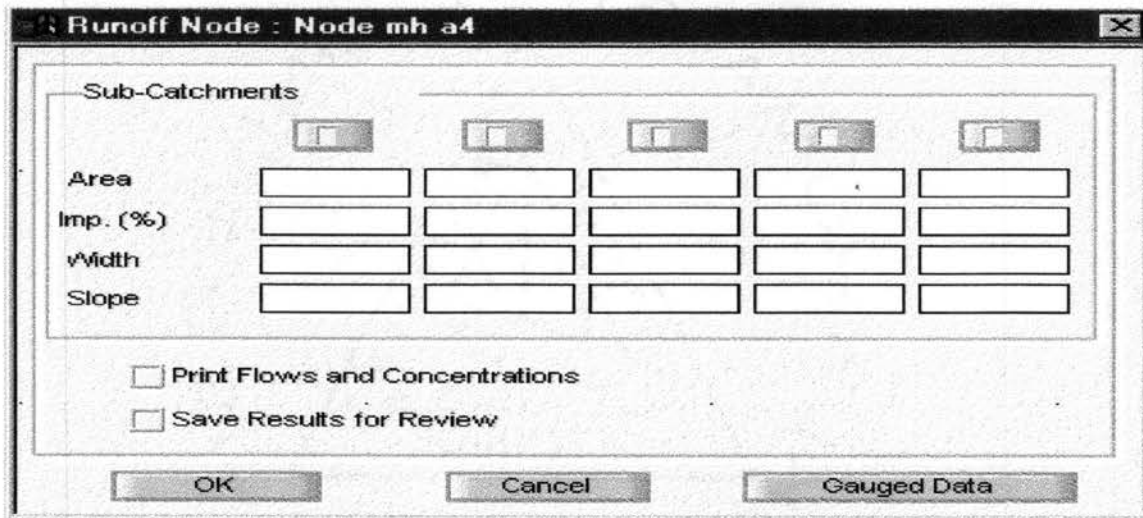
Figure 5.3. Subcatchment Surface Data Required by Each Node Generated by GIS Spatial Analysis Tools Written in This Research for Runoff Calculations

5.1.2.1 Subcatchment Area (H1–WAREA). This parameter shown in Figure 5.3 represents the area of the subcatchment (acres). It is traditionally developed manually from topographic maps by using planimeteric or other area estimating instruments. In this research, it is generated using GIS spatial analytical procedures applied on digital elevation model (DEM).

5.1.2.2 Subcatchment Percent Impervious (H1–WW3). This parameter provides the percentage of the subcatchment area that is impervious. Impervious areas must be hydraulically (directly) connected to the drainage system. In this thesis, after the subcatchment area is delineated by GIS spatial analysis technique, the corresponding area in the soil and land-use Shape Files (maps) are read by overlapping. The values obtained are used as percent imperviousness (WW3) for a particular subcatchment.

49

5.1.2.3. Subcatchment Width (H1–WW1). This parameter denotes the width of a subcatchment, (ft). If overland flow is visualized as running down-slope off an idealized rectangular catchment, then subcatchment width is the physical width of overland flow. The XP-SWMM manual recommends a good estimate for width that is calculated by dividing the quantity of subcatchment area by the average path length of the overland flow.

The manually estimated width parameter (W_{Man}) for subbasin used by Black & Veatch is based on this method. The GIS generated width ($W1_{GIS}$) also considers the subcatchments as rectangular with properties of symmetry, although these types of subbasins do not exit, especially in urban environments.

In this study, however, two other methods ($W2_{GIS}$ and $W3_{GIS}$) are proposed to calculate width parameter by using GIS spatial analysis techniques. The descriptions about the development of these width parameters by GIS spatial analysis are provided in Section 5.2.2.

5.1.2.4 Subcatchment Slope (H1–WSLOPE). This parameter considers average slope of a subcatchment (ft/ft). The subcatchment slope reflects the average slope along the pathway of overland flow to inlet locations. The calculation considers elevation difference between two points divided by the flow length by considering the elevation of each cell along the flow direction and then weighting all the slope values with the length of the flow path.

5.1.3 Global Data for Subcatchment

In XP-SWMM, the majority of the subcatchment data is entered as global data and simply referenced from within the dialog shown in Figure 5.4 (Courtesy of XP Software Inc.). This makes data entry easier when a number of subcatchments share common data.
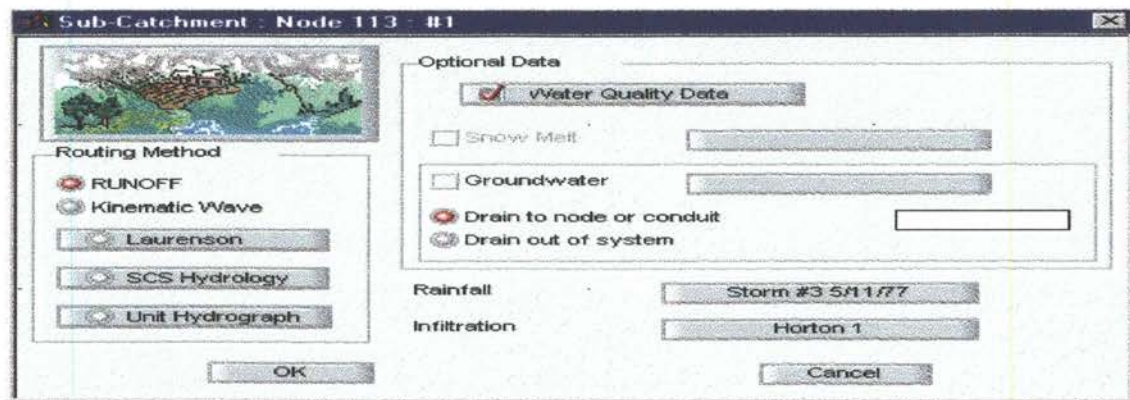
Figure 5.4. Dialog Box for Global and Common Data for Nodes in System

Also, several other data groups for Runoff Block, shown in Tables 4.2 and 5.2 (time, steps, dates, rainfall), have to be input manually. Nevertheless, data groups G1 and H1, are the groups that consume major efforts for data preparation. If a GIS generates these data, considerable labor savings will result and less calibration efforts will be required for runoff modeling.

The principal task of this study is the use of ArcView (GIS) to generate four parameters for the H1 Data Group described in Figure 5.3. GIS and spatial analysis programs are used to develop these input data that describe the catchment surface (Group H1) characteristics.

Clearly, if a DEM and drainage coverage of the catchment are available, the GIS tools developed in this study make it possible to calculate areas, lengths, slopes etc. Just as clearly, some of the parameters cannot be developed strictly from maps and plans (i.e., invert elevations, pipe shapes and dimensions) must be indicated manually.

5.2 GIS Procedures for Development of Derived Data

The basic concept of raster-based hydrologic modeling is the flow direction indication number. For example, ArcView-Spatial Analyst and ARC/INFO GRID denote the next cell in the flow path by one of eight numbers taken for computational purposes from the binary series 2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7: 1 indicates flow from the cell into East neighbor cell, 2 into Southeast cell, 4 into South cell, 8 into Southwest cell, and so on. Figure 5.5 shows eight directional flow indicators.

32	64	128
16		1
8	4	2

Figure 5.5. Spatial Analyst's Flow Direction Indicators

5.2.1 Subcatchment Delineation Procedures

The flow direction number assigned is based on the steepest calculated slope between adjacent cells. Figure 5.6 presents an *Elevation Grid* (digital elevation model (DEM)) and derives the *Flow Direction Grid*. The flow accumulation process counts the number of cells "upstream" to a cell (cells that contribute flow to the cell using the flow direction grid). Figure 5.7 visualizes the *Flow Accumulation Grid* calculated from the flow direction grid shown in Figure 5.6. By selecting cells of the Flow Accumulation Grid in which the value is greater than the arbitrary selected threshold value, a stream grid can be determined (ESRI, 1996).

27	28	28	27	28	28	29	28	28	26
27	27	29	28	29	30	31	32	32	26
26	29	28	26	27	28	27	26	28	28
27	27	26	24	25	25	24	24	26	25
26	27	25	20	23	20	19	24	26	25
25	26	23	19	18	20	19	23	25	24
24	26	23	20	16	18	18	20	25	23
24	25	21	19	19	14	16	23	20	20
22	23	20	18	14	12	14	18	16	16
12	14	16	14	14	11	12	12	12	12

32	16	1	64	16	64	1	64	1	128
4	8	2	4	8	8	4	4	1	1
16	16	2	4	8	4	4	4	8	4
64	2	2	4	8	4	4	8	16	1
4	2	1	2	4	8	4	16	8	4
4	1	1	2	4	8	4	16	8	4
16	2	1	1	2	4	8	8	16	4
4	1	2	2	4	4	8	16	4	4
4	4	8	4	2	4	8	4	4	4
8	16	1	4	1	4	16	4	4	2

(a) Elevation Grid   (b) Flow Direction
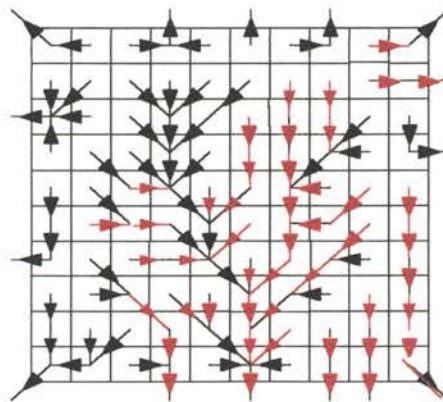
(c) Flow Direction Arrows

Figure 5.6. Elevation and Derived Flow Direction Grid

52

Shaded cells in Figure 5.7 form an exemplary stream grid created by selecting cells from the

Flow Accumulation Grid in which the value is greater than 6. Subcatchments are delineated using

Flow Direction Grid. Each subcatchment is defined by its outfall point (manhole of the subbasin).

1	0	0	2	0	0	0	1	0	1
0	0	0	0	0	0	0	0	0	1
4	0	0	3	1	0	1	1	0	0
0	0	0	7	0	1	2	4	0	1
0	0	1	12	0	2	9	0	0	0
1	0	2	3	17	0	12	1	0	1
2	0	0	1	25	0	13	2	0	2
0	0	2	0	0	41	4	0	0	3
1	0	0	3	2	47	0	0	1	4
5	2	0	5	0	54	0	1	2	5

Figure 5.7. Runoff Path Created by Flow Accumulation Grid

The area of each subcatchment can be determined by multiplying the number of cells constituting the given subcatchment by the area of single cell. This operation is described by the following formula given in Equation 5.8:

$$A_i = N_i X^2 \qquad\qquad 5.8$$

where A_i is area of *ith* subcatchment; N_i is number of cells that represents *ith* sub-catchment and X is cell width. Average slope of a subcatchment is calculated by first determining slope for each cell from the elevation grid, and then by calculating the average value for each delineated subcatchment. Equation 5.9 describes this operation:

$$S_i = \frac{\sum_{k=1}^{n_i} s_{k,i}}{n_i} \qquad\qquad 5.9$$

where S_i is average slope of *ith* subcatchment, and $s_{k,i}$ is slope of the *kth* cell of *ith* subcatchment of the watershed. Figure 5.8 is an example of three subcatchments delineated from the *Flow Direction Grid* and from user-specified points located on the channel/pipe network.
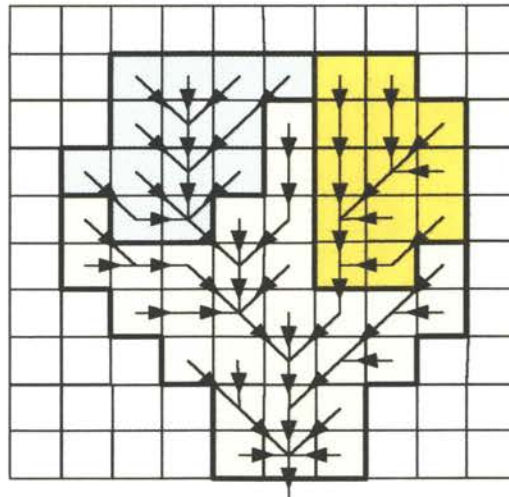
Figure 5.8. Three Delineated Subcatchments

5.2.2 GIS-Based Derivation Procedures for Width Parameter (W_{GIS})

This interface uses Avenue to write procedures for Width Variable (W_{GIS}). These programs are written to mimic anticipated urban surface water modeling and three different levels of watershed spatial abstractions are introduced and investigated. Figures 5.9 and 5.10 depict several successive levels of subcatchment *width* conceptualizations. These methods of estimating subcatchment *Width* parameter are: (1) the subcatchment area divided by the maximum length of the flow path (W_{Man} and $W1_{GIS}$); (2) subcatchment area divided by the diameter of the circle contained within the subwatershed ($W2_{GIS}$); *and* (3) diameter of the circle contained within the subcatchment ($W3_{GIS}$). In the case of various Width(W_{Man}) schematics shown in Figure 5.9, the subcatchment is assumed to take the shape of a rectangle, and the slope of the subbasin is parallel to the overland (gravity) flow.

The traditional width parameter (W_{Man}) generation procedure is similar to the one provided in XP-SWMM manual, whereas the subbasin has a rectangular shape. Its side perpendicular to the flow direction is considered as the width of the subbasin, and its value is computed as total subbasin area divided by the runoff flow length. One of the procedures to generate width parameter by GIS spatial analysis, introduced in this research, is based on the traditional methods of width estimation. The Width-1 ($W1_{GIS}$) is generated by computing the area of the delineated subbasin and then averaging this area with the length of the runoff path (L_{GIS}) within the subbasin.

54
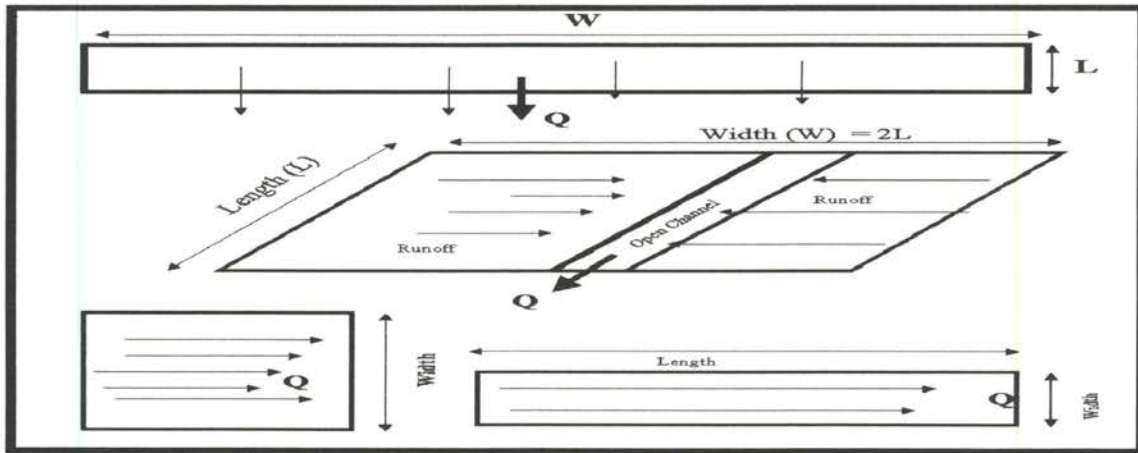
Figure 5.9. Schematics Depicting Traditional Subbasin Width Parameter (W_{MAN})

Figure 5.10 describes three GIS-based schematics introduced in this research for Width-1 ($W1_{GIS}$), *Width-2 ($W2_{GIS}$)* and *Width-3 ($W3_{GIS}$)*. $W2_{GIS}$ is estimated by dividing the GIS generated area of a subbasin *(A_{GIS})* by the diameter of a circle (D_{GIS}) contained within the boundary of a particular subbasin. In the case of Width-3, the diameter *(D_{GIS})* of the circle contained within the subbasin represents $W3_{GIS}$.
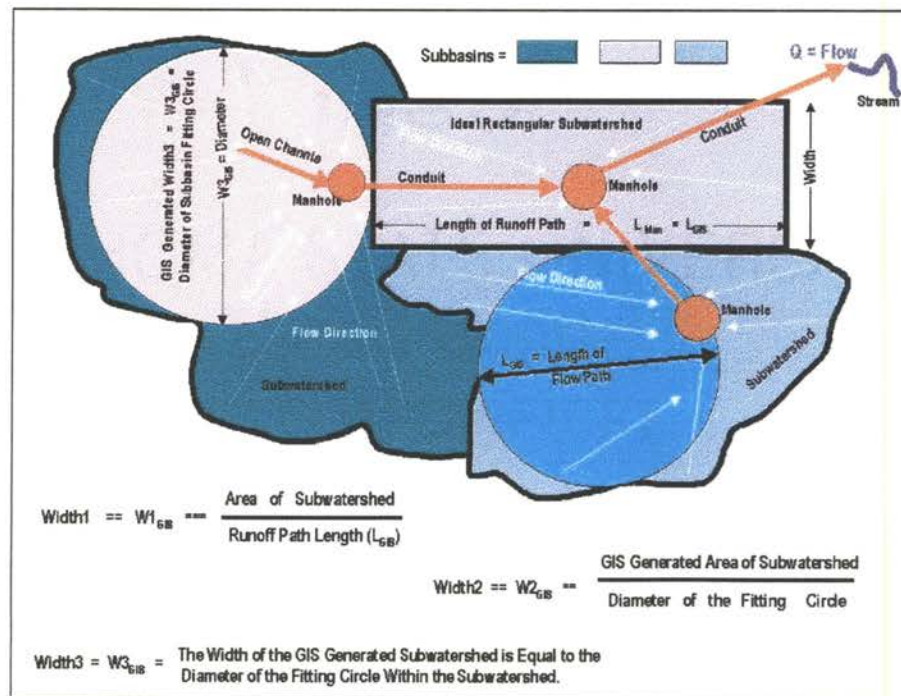
Figure 5.10. Three Schemes Depicting GIS Generated Subwatershed Widths

Figure 5.11 shows an example of the *Flow-Length Grid*. The maximum flow length within each subcatchment is the difference between maximum value of flow length and minimum value. Thus for subcatchment 1, shown in Figure 5.11, the maximum flow length is 406 – 245 = 161 map units. Its value for subcatchment 2 is 372 – 186 = 186 units and for subcatchment 3 its value is 329 – 0 = 329 units.

0	42	42	0	42	0	42	0	42	0
42	60	389	372	389	406	354	372	42	0
0	42	347	329	347	329	312	329	347	42
42	347	305	287	305	287	270	287	329	0
84	305	287	245	228	245	228	270	287	210
42	287	245	203	186	203	186	228	221	168
0	161	228	186	144	126	144	161	203	126
84	144	102	119	102	84	102	144	84	84
42	84	102	42	60	42	60	42	42	42
0	42	42	0	42	0	42	0	0	0

Figure 5.11. Example of Flow Length Grid

Figure 5.11 does not show the decimal places due to simplifying purposes. The cell width is 42.0842 feet as the map units are in feet. Subcatchment diameter is the diameter of the largest circle that can be drawn within the subcatchment area without including cells outside the drainage zone. Table 5.3 compares subcatchment widths estimated by different methods.

TABLE 5.3. SUBCATCHMENT AREA AND WIDTH ESTIMATED
BY DIFFERENT GIS SPATIAL ANALYSIS METHODS

Subcatchment	Area	Maximum Flow Length	Area/Maximum Flow Length	Circle Diameter	Area/Circle Diameter
1	23024	161	143	126	182
2	23024	186	124	126	182
3	51361	329	156	144	357

5.3 Description of ArcView's Extensions Linking XP-SWMM

In this study, a graphical user interface (GUI) is developed between ArcView GIS and XP-SWMM, which is depicted in Figure 5.12. This GUI introduces three ArcView extensions: (1) **WATERSHED (XP-DATA)**, (2) **NETWORK**, and (3) **PRESENTATIONS (XP-Results and Zone)**. These extensions support data exchange between ArcView GIS and XP-SWMM software. These extensions and their various components are depicted in Figure 5.12.
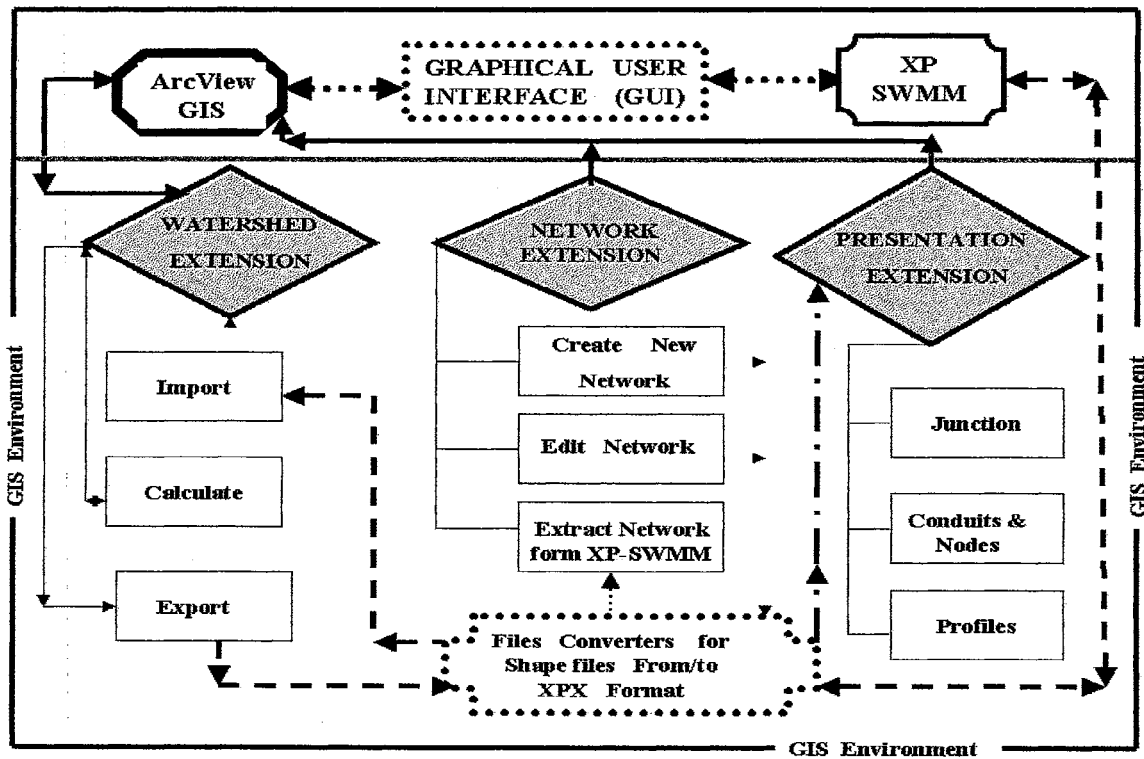
Figure 5.12. ArcView's Three Extensions, WATERSHED, NETWORK, and PRESENTATION

5.3.1 WATERSHED (XP-Data) Extension

The **WATERSHED (XP-Data)**, depicted schematically in Figure 5.13. It constructs a *Flow Direction Grid*, delineates subbasins, calculates subbasin parameters (such as area, width, slope, and percent impervious area). It creates XP export file to allow the transfer of GIS generated parameters for subbasins into the XP-SWMM. This extension adds a new menu that contains three items: (1) **Input**, (2) **Calculate**, and (3) **Export**.

57

The *Input* Item takes at least five theme; DEM, Pipes, Manholes, Land-use and Soil to generated subbasin parameters. The **Calculate** Item estimates Area, Width, Slope, and % Imperviousness of the subbasins. The **Export** Item exports all the calculations done by this Watershed Extension to XP-SWMM.
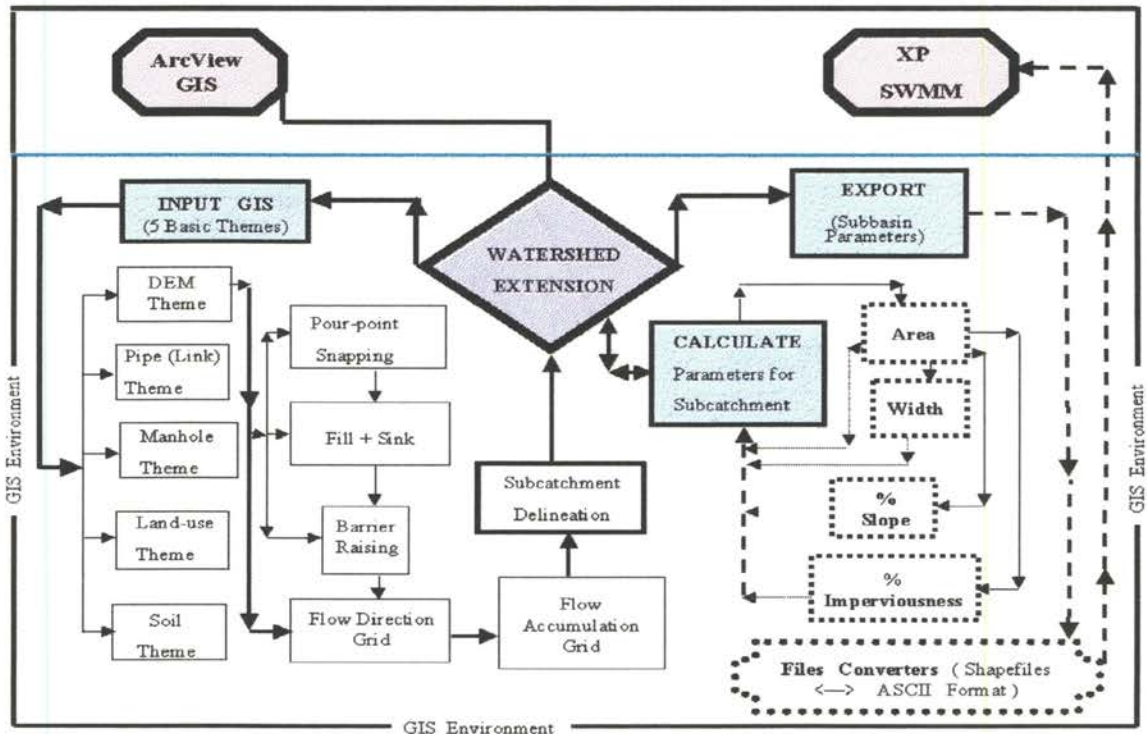
Figure 5.13. Menu Items INPUT, CALCULATE, and EXPORT for WATERSHED Extension

Menu item **Input** develops the *Grid of Flow Direction* from a *DEM* and delineates subbasins. *Flow Direction Theme* and *Point Theme* are required to calculate flow direction and to delineate subbasins. *Point Theme* represents subbasin manholes and *Grid Theme* shows elevations. If adjustment of the *Elevation Grid (DEM)* before Flow Direction is calculated, then a *Line Theme* is required which represents pipes or streams. This procedure also needs a *Line Theme* of *Flow Barriers* to be specified. For the adjustment of elevations of the cells of *DEM*, several algorithms are written, such as *"burn-in stream"* procedure, which adjusts DEM by forcing the flow direction calculated from *DEM* to follow natural flow paths. This procedure works by lowering the elevation

of those cells of the *DEM* that represent flow path (stream) or by raising the elevation of those cells that do not represent the stream. This procedure actually creates artificial *"ditches"* in *DEM* by raising the elevations of all cells in *DEM Grid* not representing real streams by an arbitrary value. A *Flow Barrier* procedure is also incorporated in this software. Although it is similar to the *"burn in stream"* procedure, as it raises the elevations of the cells of the DEM that do represent the barriers to prevent the flow to go through them. *Fill* algorithms are incorporated to ensure the total watershed area contributes to the generation of runoff. The total areal contribution is attained by removing all sinks from the surface of DEM.

The concept in the watershed delineation is the use of DEM to the direction of flow over the surface of the watershed. The basis of this concept is the application of the D8 "eight direction pour point model" as discussed in Section 5.2. Once the direction of the overland runoff is established, known as Flow Direction Grid, the Spatial Analyst accumulates the flow down to a given manhole lying within a particular delineated subbasin by counting the number of cells upstream discharging into the manhole.

The *Flow Direction Grid* is used to delineate different sets of subbasins. Since the subbasins are defined by their corresponding manholes, to delineate different subbasins, different themes (maps) of subbasin manholes are created and used for subcatchment delineation, using the same *Flow Direction Grid* for all program runs.

A routine executes an Avenue function *SnapPourPoint*. This utilizes the "snap- outlets" option by ensuring the location of subcatchment manholes on the major flow paths and snaps pour points to the cell of a maximum value in the weight grid, such as *Flow Accumulation Grid*, within a specified snap distance. After the subbasins are delineated, Spatial Analyst executes various operations on selected themes, which are required to calculate subcatchment parameters.

As described in Figure 5.13, the DEM *Theme,* (Figure 4.3), *Node Theme* (Figure 4.4) *Link Theme* (Figure 4.5), *Land-use Theme* Figure 4.6), and *Soil Theme* (Figure 4.7), are the major input themes are required to delineate subcatchments and subsequently extract their parameters, such as area, slope, width and imperviousness etc.

5.3.2 NETWORK Extension

The **NETWORK Extension,** depicted in Figure 5.14, creates and edits a basic drainage network (nodes and links) in ArcView environment, and then saves it in the XP-SWMM export file. This extension adds menu **XP-Network** to the View's menu bar and can extract a drainage network from the XP-SWMM export file. It consists of three tools for: (1) **Network Creation**, (2) **Network Import** and (3) **Editing of nodes**, and **links** in an existing runoff or sewer network.
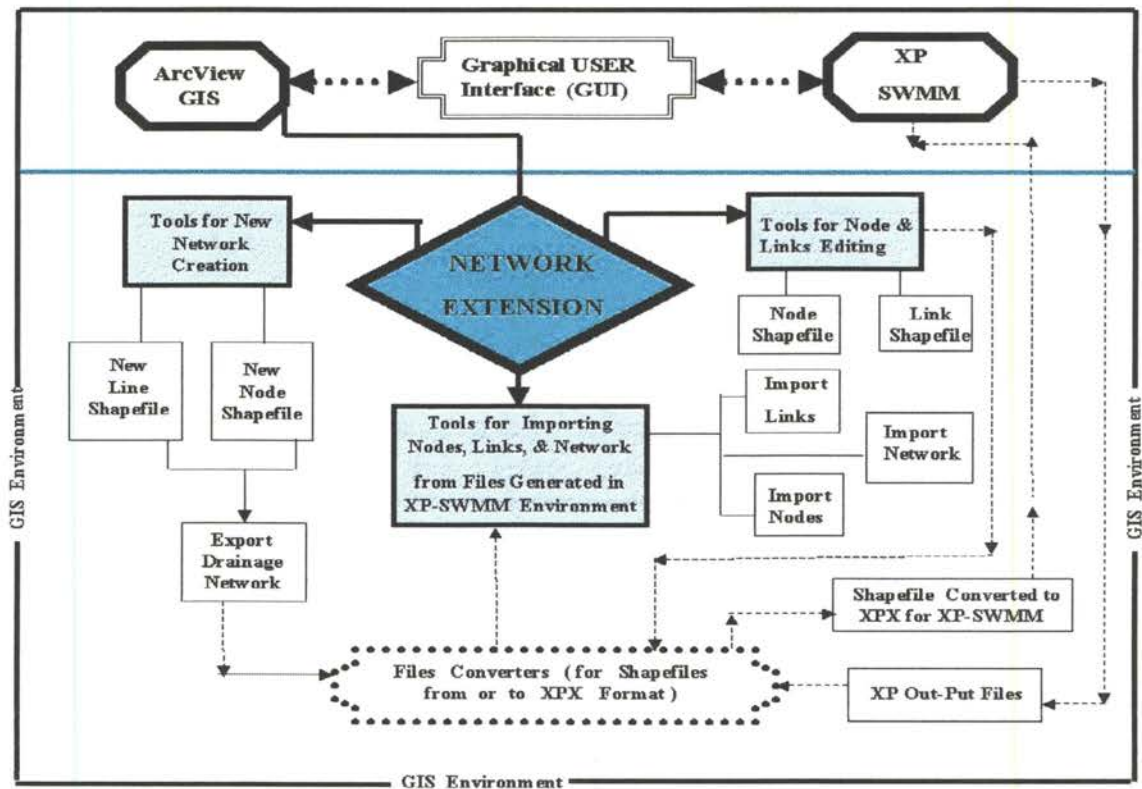
Figure 5.14. NETWORK Extension Showing Tools for Generation, Editing and Import/Export.

The NETWORK Extension also converts and stores ArcView's *shapes* representing *nodes* and *links* into an XP-SWMM readable file. The *"New Nodes"* menu item creates a new *Point Shape* of points, which represents nodes as manholes, junctions, or outlets. The *"New Links"* menu item creates a new *Line Shape* of *conduits*, ditches, or streams. Menu items *"Add Nodes"* and *"Add Links"* allow to add nodes and links to existing shape files. The drainage network is saved in XPX export files before importing it into ArcView and the information about network geometry is read by selecting one of the menu items.

60

5.3.3 PRESENTATION (XP-Results and XP-Zone) Extension

The PRESENTATION Extension, also known as XP-Results and Zone Extension, depicted in Figure 5.15, presents results from XP-SWMM in the form of maps, profiles, surcharging manholes and flood prone zones. It adds a new menu to the menu bar that contains three items: (1) *Junctions*, (2) *Conduits*, and (3) *Profiles*.
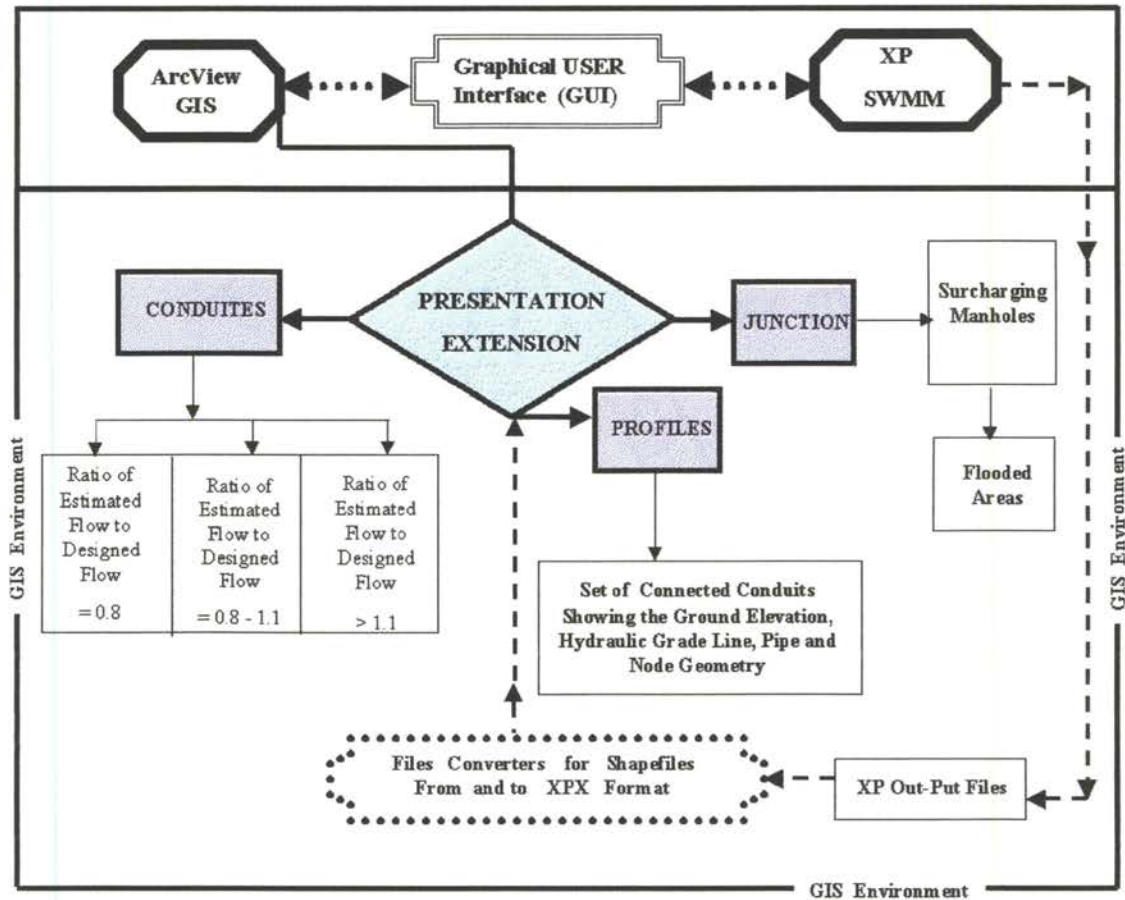
Figure 5.15. PRESENTATION Extension Showing Three Menu Items
(Junctions, Conduits, and Profiles) Developed to Display Output

Junction imports the results of hydrologic and hydraulic calculations of XP-SWMM into ArcView. The menu item *Conduits* imports the results of XP program calculations into ArcView. A *Line Shape* is created in a similar way as that of the *Point Shape* for Junction. *Profile* shows a map of nodes and conduits created using information stored in the XPX export file. The procedures and methods utilized for these extensions are described in detail in the following sections.

61

5.4 Description of Procedures for WATERSHED Extension

A new menu is added to the menu bar by loading the WATERSHED (XP-Data) Extension into ArcView. This menu contains four items—Runoff, Input, Calculate, and Export. Figure 5.16 shows these items. The Input, Calculate, and Export are the integral parts of WATERSHED Extension. The Runoff is optional, which is not necessary to generate any data for XP-SWMM.
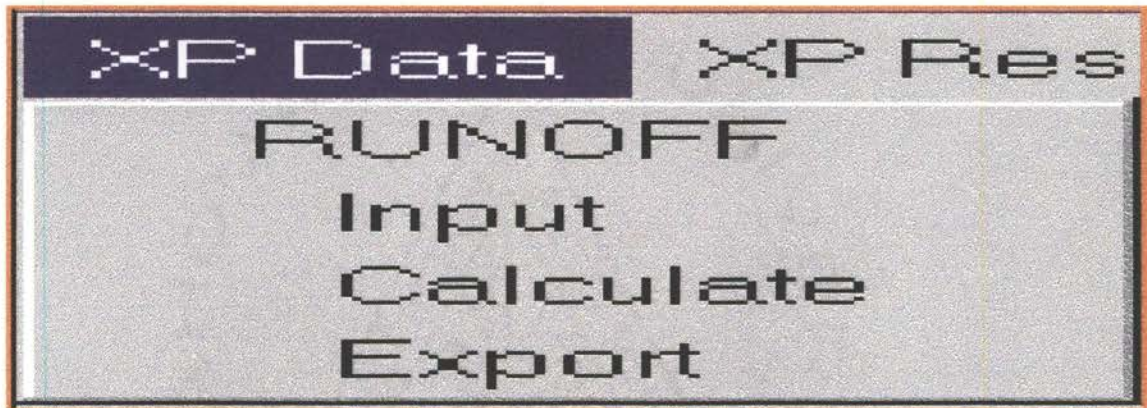
Figure 5.16. Four Menu Items; Input, Calculate, and Export, for the WATERSHED (XP-DATA) Extension

5.4.1 Flow Direction Development and Subcatchment Delineation

Menu item *INPUT* is designed to develop the *Flow Direction Grid* and delineate subcatchments. A dialog box shown in Figure 5.17 opens up whenever *Input Item* is selected. Three ArcView themes are required to calculate flow direction and delineate subcatchments: (1) *Point Theme* of subcatchment outlets; (2) *Grid Theme* of elevations (*DEM*) and (3) if adjustment of the elevation grid before flow direction is calculated, a *Line Theme* of the streams and/or a *Line Theme* of *Flow Barriers* must be specified.

Adjustment of flow path is crucial—because points (manholes) identifying subcatchment runoff collection points must be located on the major natural flow path. Unfortunately, these points are usually located off the channel determined from DEM due to imperfection of the digital terrain representation or its insufficient resolution (both horizontal and vertical) and software limitations.
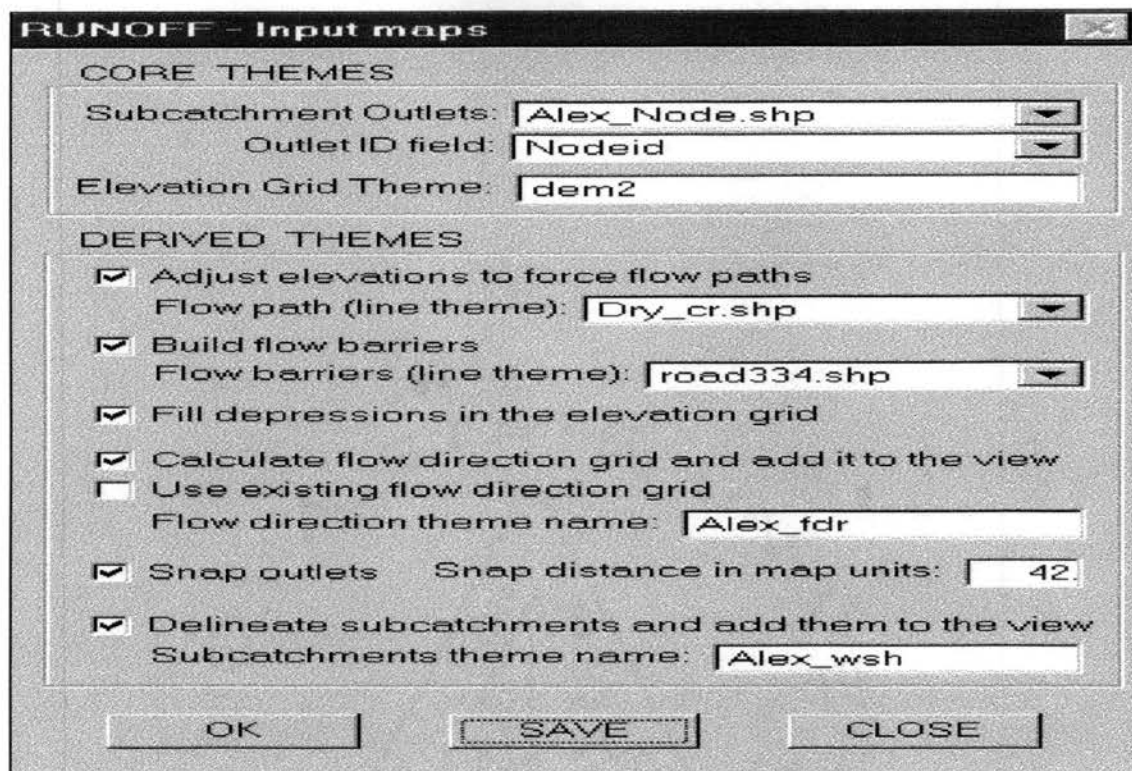
Figure 5.17. Dialog Box for Input Data Representing Core Themes
and Derived Themes for the Delineation of Subcatchments

Mizgalewicz (1996) introduced and tested a simple method to adjust DEM to force the flow direction calculated from DEM to follow natural stream flow paths. By simply lowering elevation of DEM's cells that represent streams or by raising elevation of cells that do not represent streams, artificial "ditches" are created in DEM. Such a grid can be used only to calculate the Flow Direction Grid, which is a crucial grid for subcatchment delineation, and for the calculation of drainage area and the length of the longest flow path.

During this *"Burn-In Streams"* process, the WATERSHED Extension converts the link/channel map into a grid representation. The elevations of all cells in the DEM grid that do not represent real channels are increased by an arbitrary value. To allow water to be drained from the subject area, the channel network must extend outside the DEM boundaries. This sometimes requires adding to the stream map a reach that travels from the most downstream point of the stream network to a point located out of DEM. If the stream network does not extend outside DEM, the *FILL* procedure will fill *"Burned-In"* streams and the effort to adjust flow direction for real streams will be futile. The process of building *Flow Barriers* —executed by the second Check Box in Figure 5.17

"*Build Flow barriers, Flow Barriers (line Theme)*"—is similar to the "*Burn-In-Stream*" procedure. Although a small difference exists whereas the DEM cell elevations, which represent barriers, are raised to prevent flow through them. The priority of "*Burning-In-Streams*" is higher than building *Flow Barriers*—if the *Flow Barrier* is crossed by the *Burn-In Stream*, the flow will pass through the *Barrier* at a point where both of them intersect.

5.4.2 DEM Depression Filling Procedures

The Check Box, "*Fill depressions in the elevation grid*'", Figure 5.17, represents a derived Item which is incorporated to fill the sinks or depressions in a DEM. The sink is a topological feature in which water flows into a point that has no out-fall. In a DEM, the sink is a cell surrounded by cells with higher elevations. Sinks cause discontinuities in flow paths and thus reduce the number of cells that should be contributing to the drainage area.

For detailed analysis, it is necessary to differentiate between natural and artificial sinks. Jenson, S.K. and J.O. Domingue (1988) recommend the removal of all sinks so that the whole watershed contributes to runoff generation. The ArcView procedure *FILL* performs this job and removes the sinks from the watershed terrain.

5.4.3 Flow Direction Grid Creation Procedures

The fourth Check Box in Figure 5.17, "*Calculate flow direction grid and add it to the view*" executes procedures to calculate the *Flow Direction Grid* and adds it to the View. However, after the *Flow Direction Grid* for a watershed is created. This derived grid can be used to delineate different sets of subcatchments. Since the drainage area is defined by its outlet point (manhole), different themes (maps) of subcatchment outlets may be created and employed for subcatchment delineation by using the same *Flow Direction Grid* for all program runs.

Instead of creating new subcatchment outlet themes, one *Point Theme* can be used by selecting a different set of points each time. These points can be chosen by selecting points displayed in the View or by selecting rows in the theme's attribute table using the standard tools of ArcView 3.1.

5.4.4 Subcatchment Delineation Procedures

The selection of the last Check Box in Figure 5.17 starts the delineation process of the subcatchments. The name of the theme for subcatchments to be delineated should be provided prior to the start of the procedures. The location of the inlets of subcatchments on major flow paths is ensured by introducing the "*Snap Outlets*" option that uses ArcView-Avenue function *SnapPourPoint*. This function snaps pour points to the cell of a maximum value in weight grid (such as a *Flow Accumulation Grid*) within a specified snap distance. It searches within a snap distance around each pour point for the cell of highest accumulated flow and moves the subcatchment outlet to that location. Snap distance can be specified in map units such as foot or meter or, any negative number, i.e., if a value of 130 is used, the snap distance of 130 map units will be used. The negative number represents the value of the snap distance in terms of number of cells. For example, if a number 3 with a negative value, i.e., (-3) is used, the snap distance of three times the width of he cell will be applied.

The "**CLOSE**" button, shown in Figure 5.17, closes the dialog box and subsequently no action is performed. The "**SAVE**" button saves information entered by the user in View's tag—when this dialog box is reopened, all previously entered information is displayed. The "*OK*" button performs all calculations selected in the dialog box.

5.4.5 Calculation of Subcatchment Parameters

The dialog boxes, shown in Figures 5.18 and 5.19, "*Area Conversion Factors*" and "*Runoff Subcatchment Parameters*" are designated to enter data and select themes that are required to calculate subcatchment parameters and output fields are specified. In Figure 5.19, the major input theme is the grid that represents subcatchments. Another input theme is a polygon map of land use with information about percentage of impervious area. To estimate average subcatchment impervious area percentage, the elevation grid calculates subcatchment slope and the flow direction grid estimates longest flow path (L_{GIS}).

The estimated parameters are stored in an attribute table of the point theme. The user must specify ID fields of both watershed grid and output theme to link the input table of subcatchments with the output table of the point theme. If the output theme does not contain a field specified as

an "Output Field," a new field will be created in the output feature table. If the field exists, the existing values will be replaced by new ones in all or in selected records, according to the table editing rules of ArcView.

The user can select a conversion factor to change the units of the subcatchment area, for example, change square feet to acres or enter any value in the input box, as shown in Figure 5.18. Selected area conversion factors are also listed in Table 5.4.
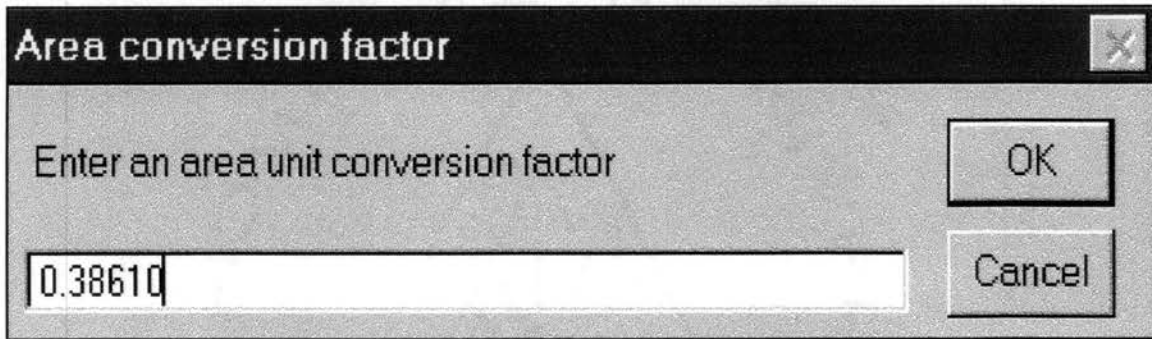
Figure 5.18. Entering User-Defined Area Conversion Factor

The subcatchment width multiplier is introduced to adjust the estimated values, for example, to account for subcatchment shape, or one- or two-sided stream inflow. The purpose of command buttons *OK*, *SAVE*, and *CLOSE* is the same as buttons shown in the Dialog Box "Runoff input maps" of Figure 5.17

TABLE 5.4. LIST OF AREA CONVERSION FACTORS USED IN INTERFACE

Selection		Conversion Factor	Conversion Result
2.2957e-05	ft^2 to ac,	2.30E-05	Square feet to acres
3.5870e-08	ft^2 to mi^2,	3.59E-08	Square feet to square miles
0.092903	ft^2 to m^2,	0.092903	Square feet to square meters
9.2903e-06	ft^2 to ha,	9.29E-06	Square feet to hectares
9.2903e-08	ft^2 to km^2,	9.29E-08	Square feet to square kilometers
10.764	m^2 to ft^2,	10.764	Square meters to square feet
2.4711e-04	m^2 to ac,	2.47E-04	Square meters to acres
0.0001	m^2 to ha,	0.0001	Square meters to hectares
0.38610	km^2 to mi^2,	0.3861	Square kilometers to square miles
Other			User defined
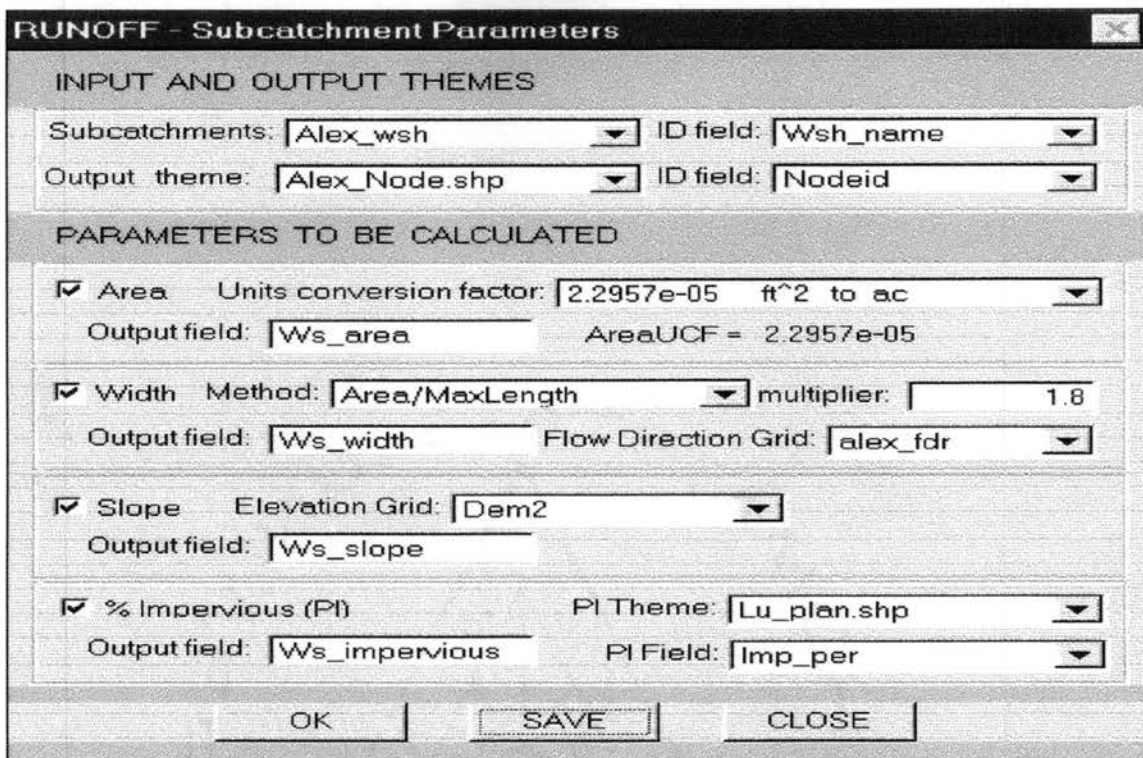
Figure 5.19. Dialog Box for Calculating Subcatchment Parameters

5.4.6 Exporting Subcatchment Parameters

The dialog box, given in Figure 5.20 describes "*Export Subcatchment Parameters*" for runoff as shown by four Check Boxes. It is designated to store sub-catchment parameters in a file that can be read by XP-SWMM. The user must specify the theme that contains subcatchment parameters, the theme's attribute table field that stores subcatchment IDs, and fields that hold parameters to be exported.

Figure 5.20 also describes three options of exporting data, i.e., *Append*, *Replace All*, and *Replace by ID*. The "*Append*" option writes subcatchment parameters in lines added at the end of the existing file (if the file already does not exist, a new file is created). If "*Replace All*" is selected, the export procedure deletes all lines that are related to selected parameters from the existing file, and adds estimated parameters at the end of the export file. The "*Replace by ID*" option is similar to the "*Replace All*" option, except the procedure deletes only those lines from the export, which are related to selected parameters and have the same subcatchment ID as IDs of the subcatchment for which the parameters were estimated.

67

Figure 5.20. Dialog Box for Exporting Four Runoff parameters for Subcatchment

Table 5.5 provides four sample code lines for the four parameters shown in Figure 5.20 (four check boxes). These lines are examples of exported subcatchment area, width, slope, and percentage impervious in the XP-SWMM's export format:

TABLE 5.5. EXAMPLES OF CODE EXPORT FORMAT USED FROM XP-SWMM

DATA R_WAREA "102" 0 5 .352 "" "" "" ""

DATA R_WIDTH "102" 0 5 200. "" "" "" ""

DATA R_WSLOPE "102" 0 5 .04 "" "" "" ""

DATA R_WIMP "102" 0 5 31. "" "" "" ""

The subcatchment parameters are exported according to the general ArcView export rules. Only data for selected records in the attribute table are exported. If no record or feature is selected, parameters for all subcatchments are exported. This allows the user to pass data to XP-SWMM only for selected subcatchments.

Figure 5.20. Dialog Box for Exporting Four Runoff parameters for Subcatchment

Table 5.5 provides four sample code lines for the four parameters shown in Figure 5.20 (four check boxes). These lines are examples of exported subcatchment area, width, slope, and percentage impervious in the XP-SWMM's export format:

TABLE 5.5. EXAMPLES OF CODE EXPORT FORMAT USED FROM XP-SWMM

```
DATA R_WAREA "102" 0 5 .352 "" "" "" ""

DATA R_WIDTH "102" 0 5 200. "" "" "" ""

DATA R_WSLOPE "102" 0 5 .04 "" "" "" ""

DATA R_WIMP "102" 0 5 31. "" "" "" ""
```

The subcatchment parameters are exported according to the general ArcView export rules. Only data for selected records in the attribute table are exported. If no record or feature is selected, parameters for all subcatchments are exported. This allows the user to pass data to XP-SWMM only for selected subcatchments.

5.5.1 Creating Map of Nodes

The "*New Nodes*" menu item, shown in Figure 5.21, allows the user to create a new shape of points which represents such network nodes as manholes, junctions, or outlets. When this item is selected, user is asked to specify a name of *Point Shape File* to be created as in Figure 5.23.
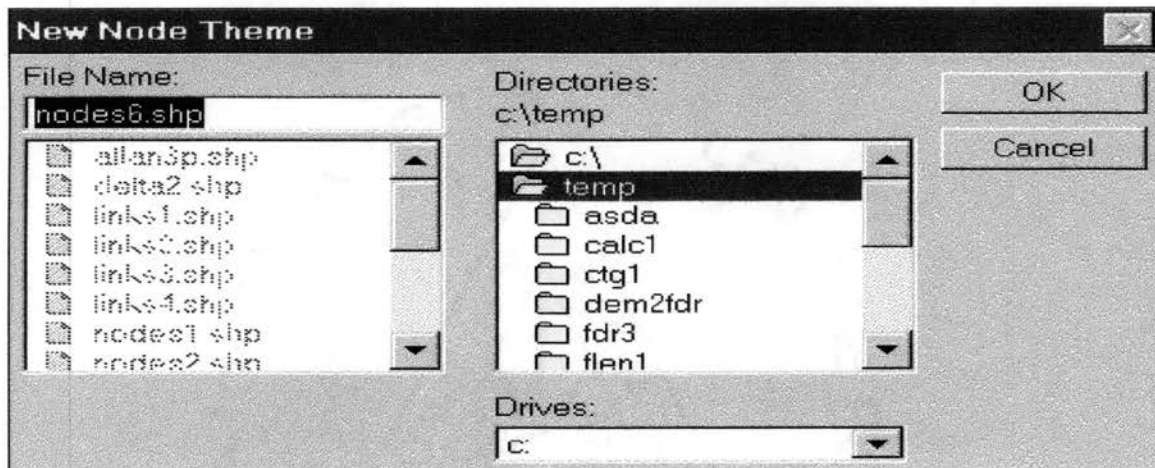
Figure 5.23. Dialog Box for Entering Node Shape File Name

After the name is entered, the program makes tool button "*N*" active. The user can click the mouse within active View in the location where a node/manhole is to be created. The user is then prompted for basic information required to transfer geographic information to XP-SWMM, i.e. node name and node type, as shown in Figure 5.24. The process of building the map of nodes can be terminated by pressing the tool button "*S*".
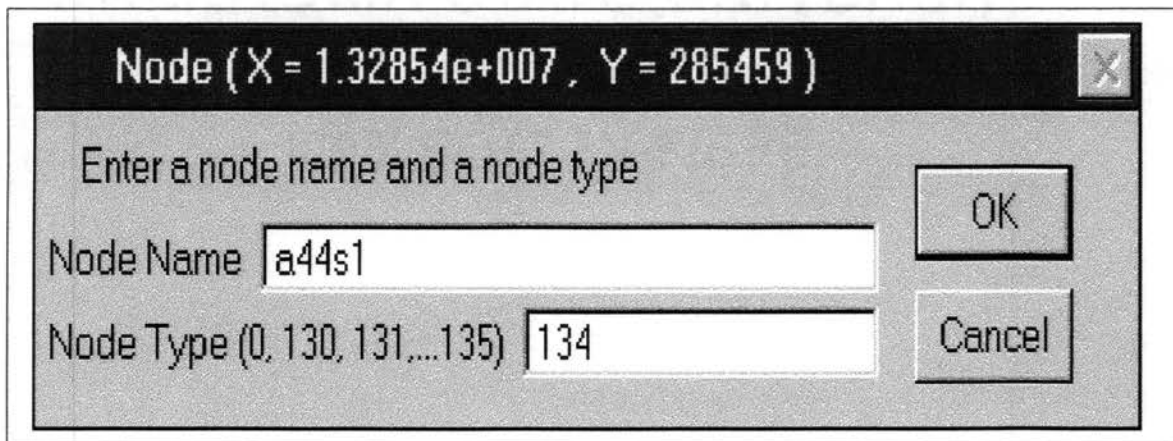
Figure 5.24. Dialog Box for Entering Name and New Node Type

5.5.2 Creating Map of Conduits

The "*New Links*" menu item allows the user to create a new line shape of conduits, ditches, or streams. When this item is selected, the user is asked to specify the name of node theme on which the link theme will be constructed as shown in Figure 5.25. To create a map of conduits, a map of manholes must exist. The user can then enter a name of the line shape file to be created as shown in Figure 5.25.
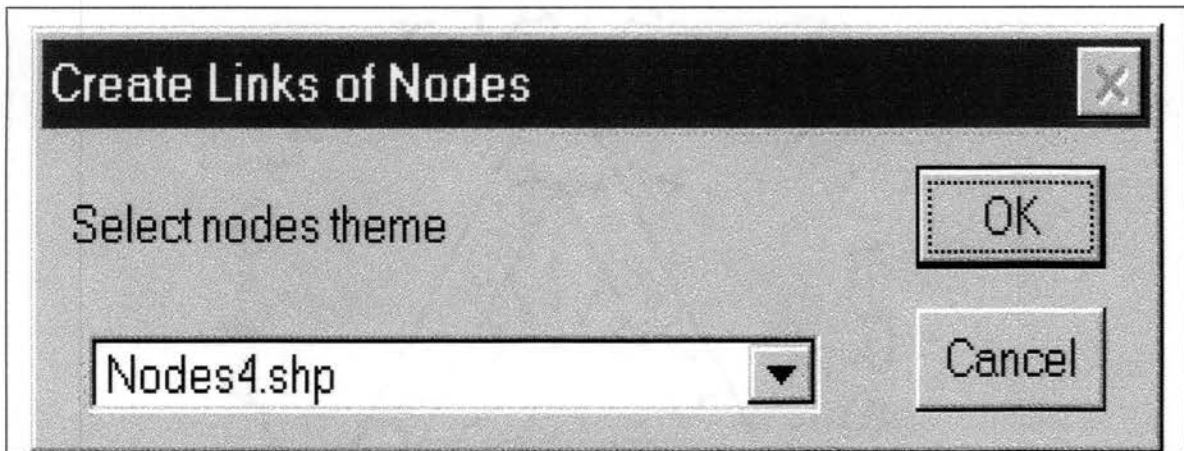
Figure 5.25. Selecting Node Theme to Create Map of Conduits

After entering the name of the link theme, the tool button "*L*" gets active. The upstream node of a link is selected by clicking the mouse on the selected point. A confirmation is displayed about the upstream node selection. The selection can be accepted about the upstream node by clicking on "*Yes*" or "*No*" buttons, as shown in Figure 5.27 and subsequently the program is ready to guide the user to make another selection.
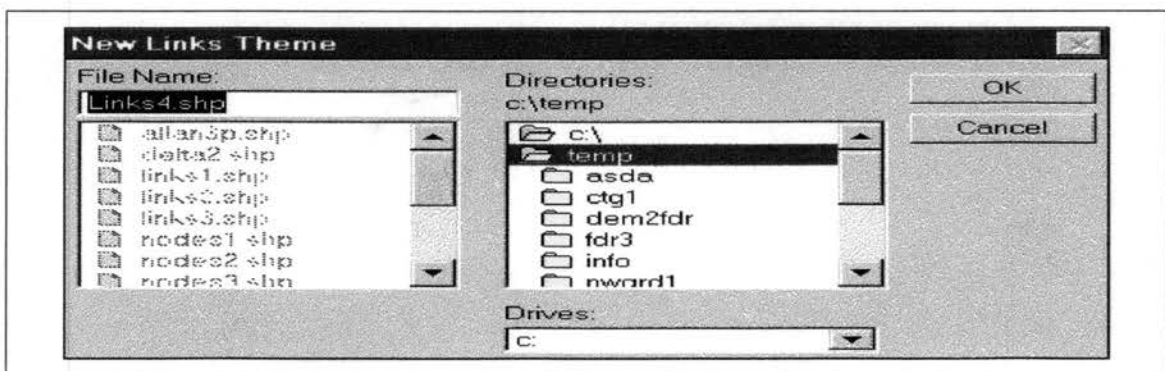
Figure 5.26. Dialog Box for Entering Conduit Shape File Name

71
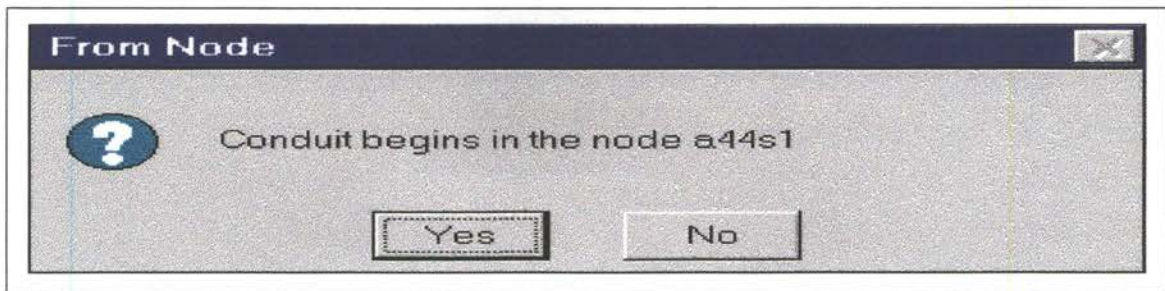
Figure 5.27. Accepting Conduit Upstream Node

After the upstream node is accepted, the user must select the downstream node. It is accepted in the same manner as selecting the upstream node as in Figure 5.28.
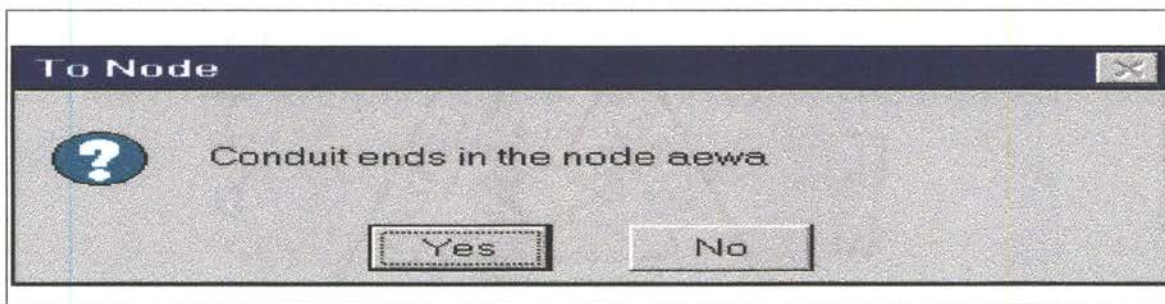
Figure 5.28. Accepting Conduit Downstream Node

The dialog box shown in Figure 5.28 allows the user to enter basic information required by the XP-SWMM export file format. The process of building the map of links can be terminated by pressing tool button "*S*".
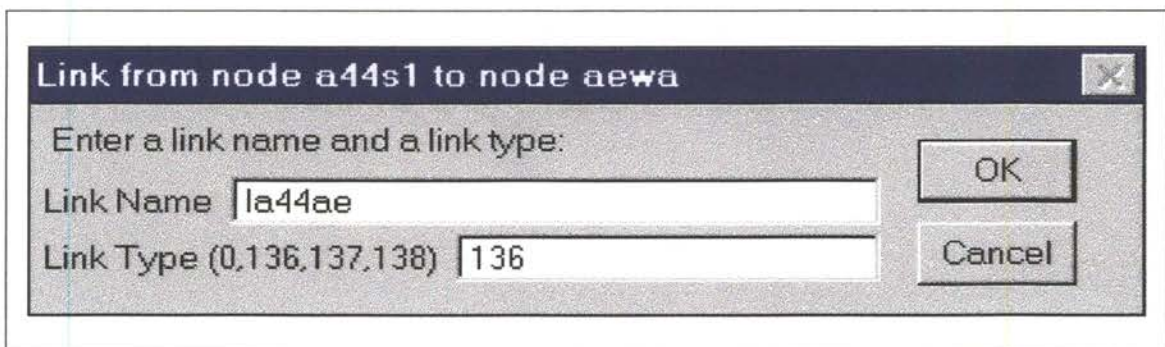
Figure 5.29. Dialog Box for Entering Name and Type of New Link

72

5.5.3 Adding Nodes and Links to Existing Shapes

Menu items "**Add Nodes**" and "**Add Links**" shown in Figure 5.21, allow the user to add nodes and links to existing shapes. The new nodes or links can be added only to the active point or line theme. In addition, this theme attribute table must contain a string item named "**Node_name**" if nodes are added or "**Link_name**" if links are added. The editing process is similar to that of developing maps of nodes and links, except no new shape file is created. Figure 5.30 shows an example of a drainage network map created using an orthophoto of a small residential area.
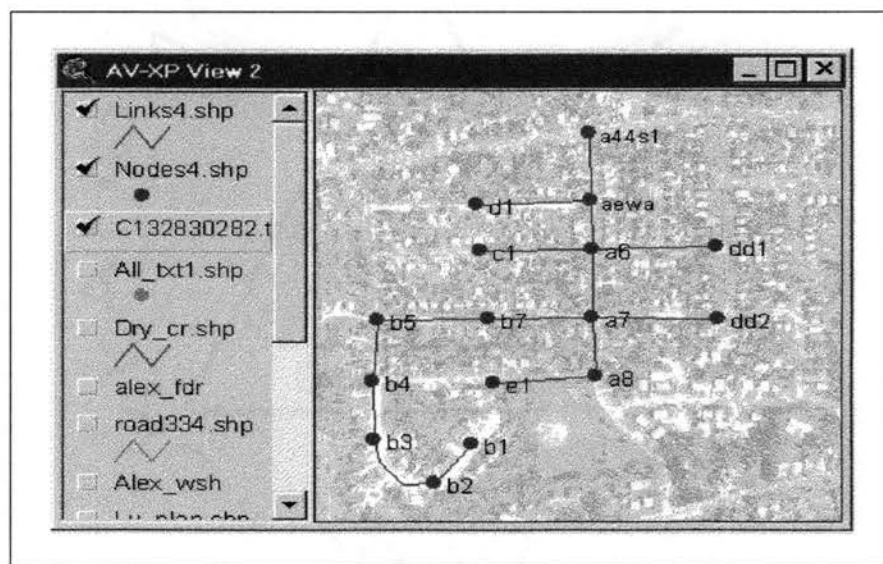
Figure 5.30. An Application of NETWOR Extension—Map of Drainage

5.5.4 Import / Export Tools for Drainage Network Between XP-SWMM and ArcView

Before importing the map of a particular drainage network into ArcView, this network must first be saved in the XPX export format using XP-SWMM tools. Information about network geometry is then read by selecting one of the menu items "**Import Nodes**", "**Import Links**", or "**Import Nodes+Links**" and subsequently appropriate shape files are created. The user is asked to select an XPX export file from which the network geometry information should be read.

The menu item "Export Nodes+Links" runs the procedure that converts ArcView shape files into XP-SWMM export file. The user must specify node theme, link theme, and XPX file names to be created. Figures 5.31, 5.32, and 5.32 show dialog boxes for selecting **Node Theme, Link Theme**, and **Export File**, respectively.

73
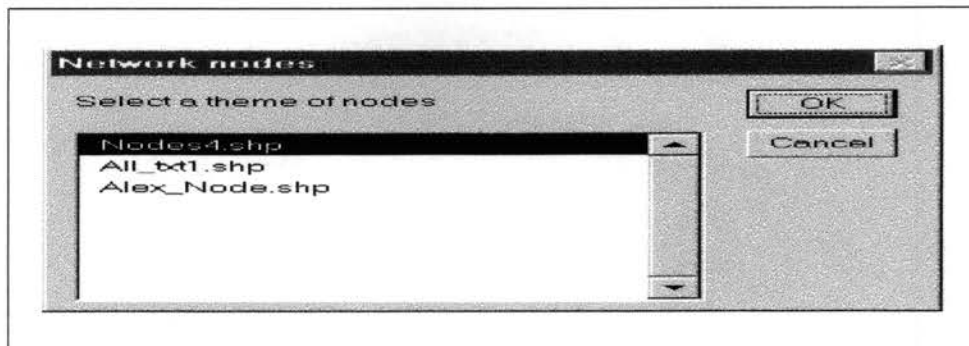
Figure 5.31. Selecting Node Theme for Exporting Files Into XP-SWMM Program
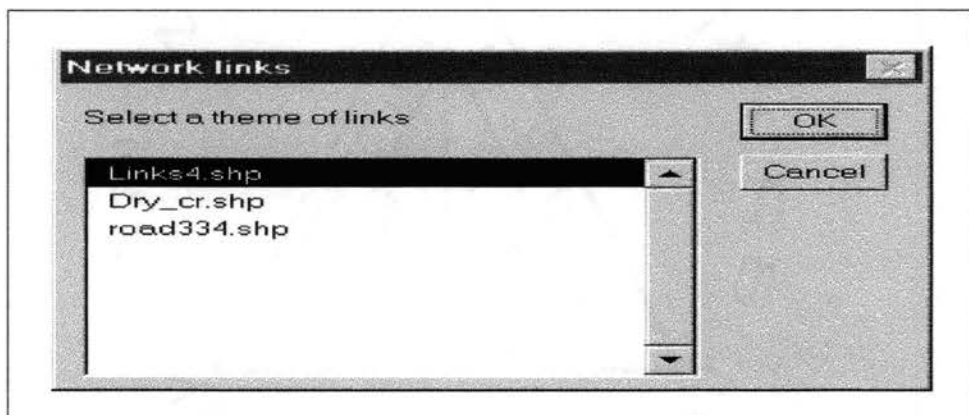
Figure 5.32. Selecting Link Theme for Exporting Files Into XP-SWMM Program
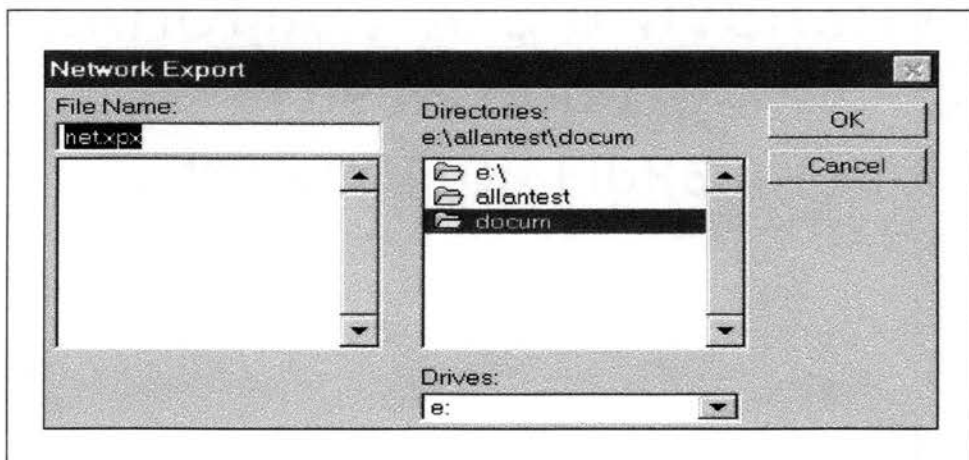
Figure 5.33. Specifying XP-SWMM Export File

Figure 5.34 presents a network that was imported into XP-SWMM using the XPX export file created by the XP-Network extension. Table 5.6 provides data detail lines, which are extracted from the XPX-file about the network given in Figure 5.34.
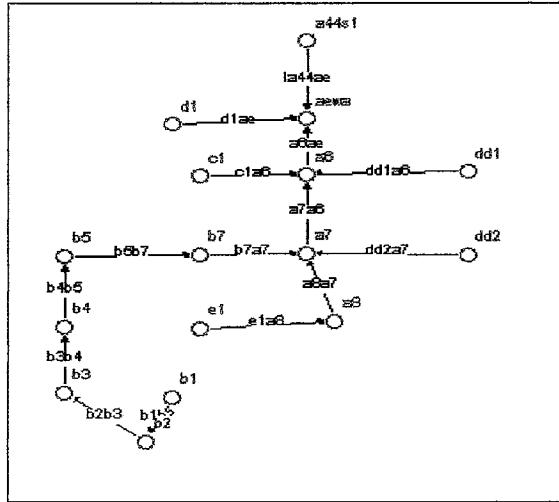
Figure 5.34. Drainage Network Created Using NETWORK Extension
and Imported Into XP-SWMM Program

TABLE 5.6. EXTRACTED DATA FOR NODES AND LINKS
FOR NETWORK SHOWN IN FIGURE 5.34

Item Number	Node Data Extracted from Figure 5.34	Conduit Data Extracted from Figure 5.34
1	NODE 134 "a44s1" 13285400 285459	
2	NODE 134 "aewa" 13285400 285110	LINK 136 "la44ae" "a44s1" "aewa"
3	NODE 134 "a6" 13285400 284859	LINK 136 "b5b7" "b5" "b7"
4	NODE 134 "a7" 13285400 284506	LINK 136 "b7a7" "b7" "a7"
5	NODE 134 "a8" 13285500 284203	LINK 136 "e1a8" "e1" "a8"
6	NODE 134 "b1" 13284900 283857	LINK 136 "b1b2" "b1" "b2"
7	NODE 134 "b2" 13284800 283656	LINK 136 "b2b3" "b2" "b3"
8	NODE 134 "b3" 13284500 283879	LINK 136 "b3b4" "b3" "b4"
9	NODE 134 "b4" 13284500 284174	LINK 136 "b4b5" "b4" "b5"
10	NODE 134 "b5" 13284500 284491	LINK 136 "a8a7" "a8" "a7"
11	NODE 134 "b7" 13285000 284499	LINK 136 "a7a6" "a7" "a6"
12	NODE 134 "c1" 13285000 284852	LINK 136 "a6ae" "a6" "aewa"
13	NODE 134 "d1" 13284900 285089	LINK 136 "d1ae" "d1" "aewa"
14	NODE 134 "dd1" 13286000 284873	LINK 136 "c1a6" "c1" "a6"
15	NODE 134 "dd2" 13286000 284499	LINK 136 "dd2a7" "dd2" "a7"
16	NODE 134 "e1" 13285000 284167	LINK 136 "dd1a6" "dd1" "a6"

75

5.6 PRESENTATION Extension

The **PRESENTATION** Extension consists of two modules: (1) *Results* and (2) *Zones*, depicted in Figure 5.15. It adds a new menu to the menu bar of ArcView 3.1. This menu contains three tools—*Junctions, Conduits,* and *Profiles*. The applications of these three tools are discussed in this section.

The XP-Zone adds *Z* button to the View's tool bar, which is shown in Figure 5.22. The *Z* application shows the possible areas, which can be flooded or those locations where runoff might create hazard to the public. The *Z* button should be selected prior to conducting a spatial analysis to investigate the potential of some urban areas to get flooded. This tool works only when the DEM theme and Z button are selected simultaneously. The *Junctions* tool will provide the simulated hydraulic results for the runoff collections system from XP-SWMM and it will display the surcharging manholes (in red color). If any surcharging manhole is clicked on, the program will guide the user to select various options to develop the possible extent of the flooded area.

For example, if a user likes to analyze the affects of a surcharging manhole within a distance of 500 feet from its location. The *Z* tool will delineate a circular area having its boundary at a distance of 500 from this particular surcharging manhole. Subsequently, it will execute the spatial analysis on the *DEM* theme. The potentially affected locations are displayed in a color, which is different than the color shown by the adjacent non-flooded areas. The hydrologist is required to overlap this extent of potentially hazardous area on a street theme. Consequently, the exact location and distance of inundated streets and properties can be mapped.

Figures 5.35 and 5.36 show the *XP-Results* module and two of its tools, *Junctions* and *Conduits*. The menu item *Junctions* imports the results of XP program calculations into ArcView. Figure 5.36 shows the dialog box which asks the user to select the XPX export file to read node coordinates to create a shape file of points and the XP output file to write results into the node shape's attribute table.

5.6.1 Junction and Node Data

Figure 5.36 shows a View that guides the user to enter the node shape file name to be created and the View name on which the nodes should be displayed. Figure 5.37 presents the information stored in the node attribute table. The items (fields) are described in Tables 5.7 and 5.8. The menu item Conduits allows the user to import results of the XP program calculations into ArcView. A line shape is created in a manner similar to the point shape for nodes. The user is

76

asked to select the XPX export file to read node coordinates to create a shape file of links, and to select the XP output file to read parameters and simulation results related to nodes and conduits. These are then written into the line (conduits) shape's attribute table.

Figure 5.35. Menu of PRESENTATION Extension
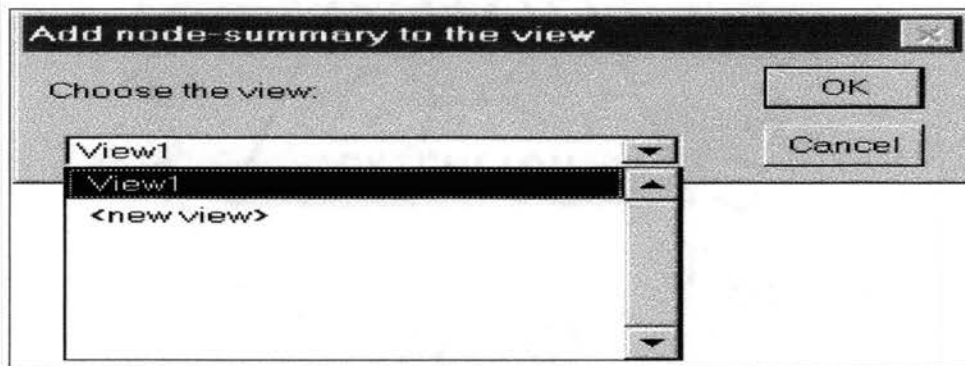
Figure 5.36. Adding Node Theme for Extracting Results From XP-SWM

TABLE 5.7. DATA IMPORTED FROM "JUNCTION DATA" ASCI FILE (*.OUT) (FIELD = NAME OF A FIELD IN THE POINT-SHAPE ATTRIBUTE TABLE; COLUMN = COLUMN NAME FROM THE TABLE "JUNCTION DATA")

Field	Column Name
Inp_Num	Inp Num
node_name	Junction Name
Ground_el	Ground Elevation
Crown_el	Crown Elevation
Invert_el	Invert Elevation
Qinst	Qinst Cfs
Ini_depth	Initial Depth (ft)
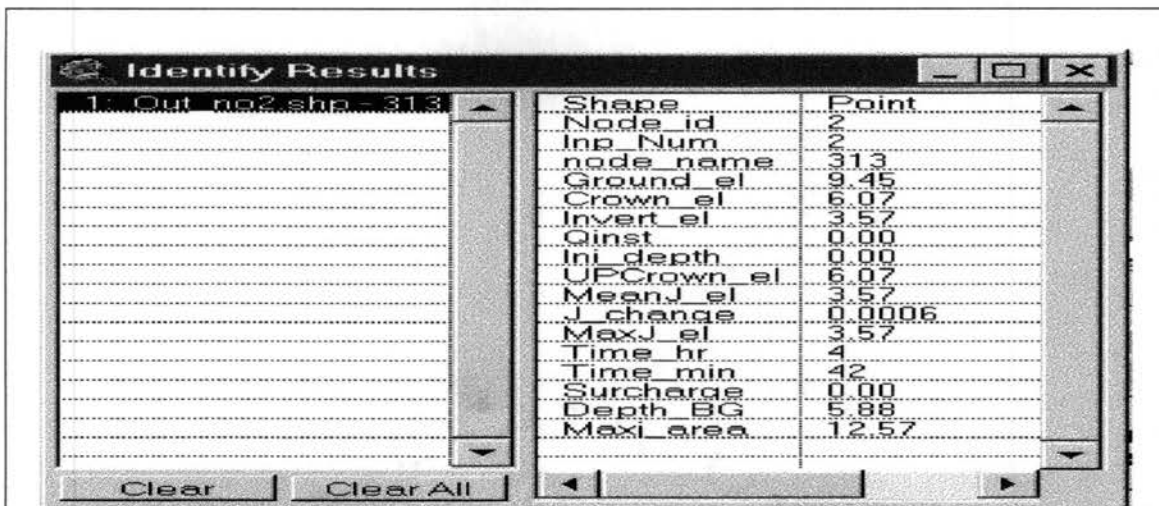
Figure 5.37. XP-SWMM Results Stored in Node Shape Attribute Table—
Message Box Displayed by ArcView "Identify" Tool

TABLE 5.8. DATA IMPORTED FROM "JUNCTION SUMMARY
STATISTICS" (FIELD = NAME OF A FIELD IN THE ATTRI-
BUTE TABLE; COLUMN = COLUMN NAME FROM TABLE
"JUNCTION SUMMARY STATISTICS")

Field	Column Name
Node_Name	Junction Name
Ground_El	Ground Elevation (ft)
Upcrown_El	Uppermost Pipe Crown Elevation (ft)
Meanj_El	Mean Junction Elevation (ft)
J_Change	Junction Average Percentage Change
Max._El	Maximum Junction Elevation (ft)
Time_Hr	Time of Occurrence (Hour)
Time_Min	Time of Occurrence (Minute)
Surcharge	Feet of Surcharge at Maximum Elevation
Depth_BG	Feet of Max. Depth Below Ground Elevation
Max._Area	Maximum Junction Area ft^2

The user is also asked to enter the node shape file name to be created as well as the View name on which the nodes should be displayed. Figure 5.38 shows items of the created conduits attribute table as displayed by ArcView's tool "Identify". These items (fields) are described in Tables 5.9 and 5.10. Values of items such as ground_up, ground_down, invert_up, and invert_up were extracted from tables related to nodes.

78
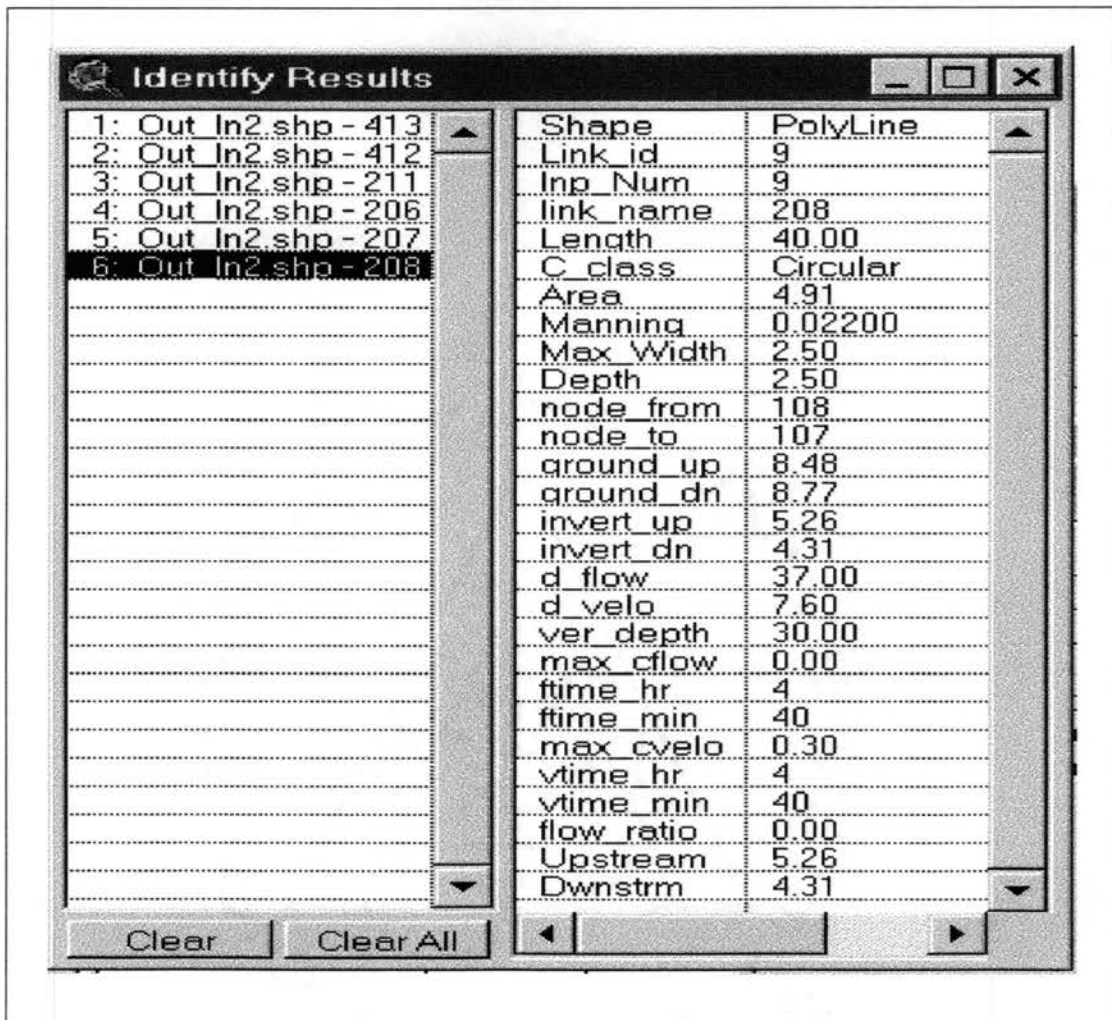
Figure 5.38. Results of XP-SWMM Stored in Conduit or Line-Shape Attribute Table
(Message Box Displayed by Arc-View "Identify" Tool)

TABLE 5.9. DATA IMPORTED FROM "CONDUIT DATA" (FIELD = NAME OF FIELD IN
ATTRIBUTE TABLE COLUMN = COLUMN NAME FROM TABLE "CONDUIT DATA")

Field	Column Name
Inp_Num	Inp Num
link_name	Conduit Number
Length	Length (ft)
C_class	Conduit Class
Area	Area (ft^2)
Manning	Manning coef.
Max_Width	Max Width (ft)
Depth	Depth (ft)
Node_from	Junctions at Ends
Node_to	Junctions at Ends

TABLE 5.10. DATA IMPORTED FROM "CONDUIT SUMMARY STATISTICS"
(FIELD = NAME OF FIELD IN ATTRIBUTE TABLE COLUMN = COLUMN
NAME FROM TABLE "CONDUIT SUMMARY STATISTICS")

Field	Column Name
Link_Name	Conduit_name
D_Flow	Design Flow (cfs)
D_Velo	Design Velocity (ft/s)
Ver_Depth	Conduit Vertical Depth (in.)
Max_Cflow	Maximum Computed Flow (cfs)
Ftime_Hr	Time of Occurrence (hr)
Ftime_Min	Time of occurrence (min)
Max_Cvelo	Maximum Computed Velocity (ft/s)
Vtime_Hr	Time of Occurrence (hr)
Vtime_Min	Time of Occurrence (min)
Flow_Ratio	Ratio of Maximum to Design Flow
Upstream	Maximum Depth > at Pipe Ends Upstream (ft)
Downstream	Maximum Depth > at Pipe Ends Downstream (ft)

5.6.2 Hydraulic Capacity Display of Runoff Collection Network

Figures 5.39 and 5.40 show node and conduit maps created using information stored in the XP-export and XP-simulation output files. Based on the relation of head to ground level (BGL— Below Ground Level), nodes are divided into two groups. Figure 5.39 shows two sets of manholes. One group contains nodes in which the calculated head is below ground elevation (displayed in blue color); the second group is composed of flooded nodes, i.e. manholes where the head equals ground elevation (displayed in red).

Figure 5.40 shows three sets of conduits with different hydraulic capacities. Each conduit set is classified according to the ratio of calculated discharge to the design discharge (item flow ratio). The following three thresholds are used: (1) flow ratio <0.8; (2) flow ratio within interval from 0.8 to 1.1; and (3) flow ratio >1.1.
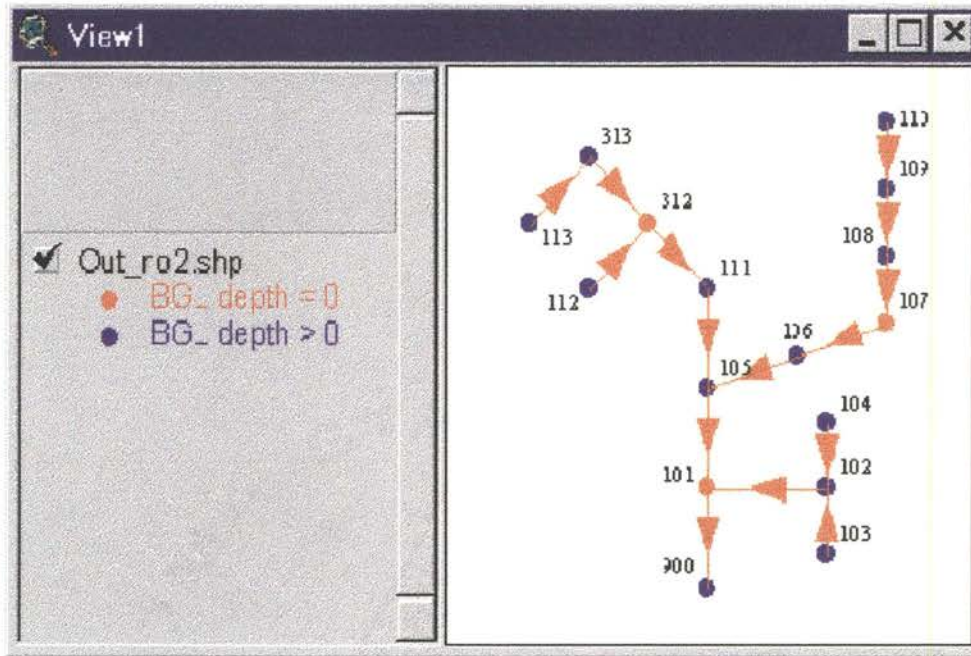
Figure 5.39. PRESENTATION Extension Showing Surcharging Manholes
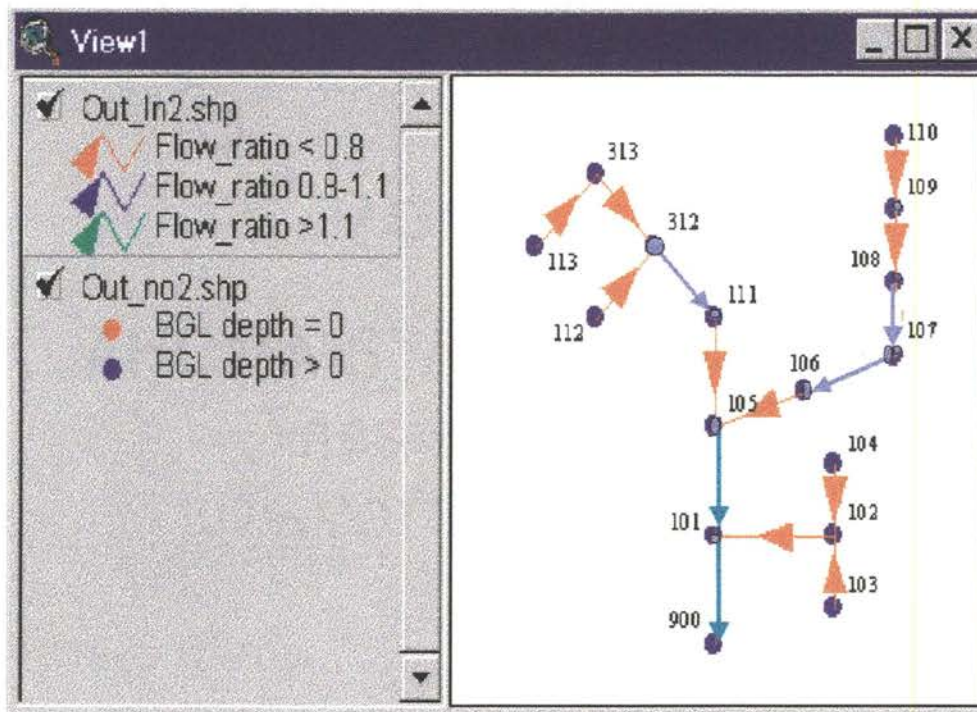
Figure 5.40. PRESENTATION Extension—Map of Drainage System Showing Three
Sets of Conduits Created From Results of XP-SWMM

When drainage system calculations are imported and stored in ArcView's form, ArcView tools can be used to manipulate imported data and present graphic results, support analysis, and design process. For example, water head values are used, as a label of manholes and conduits would significantly enhance speed of spatial verification of the drainage system.

Before a profile is created, a set of connected conduits must be selected. This can be done by using the selection tool located on ArcView's tool bar; menu item "**Profiles**". The procedure reads necessary information from the conduit's attribute table and draws the profile. Figure 5.41 shows an exemplary profile extracted from the network shown in Figures 5.39 and 5.40, for conduits which go from node 313 to node 900.
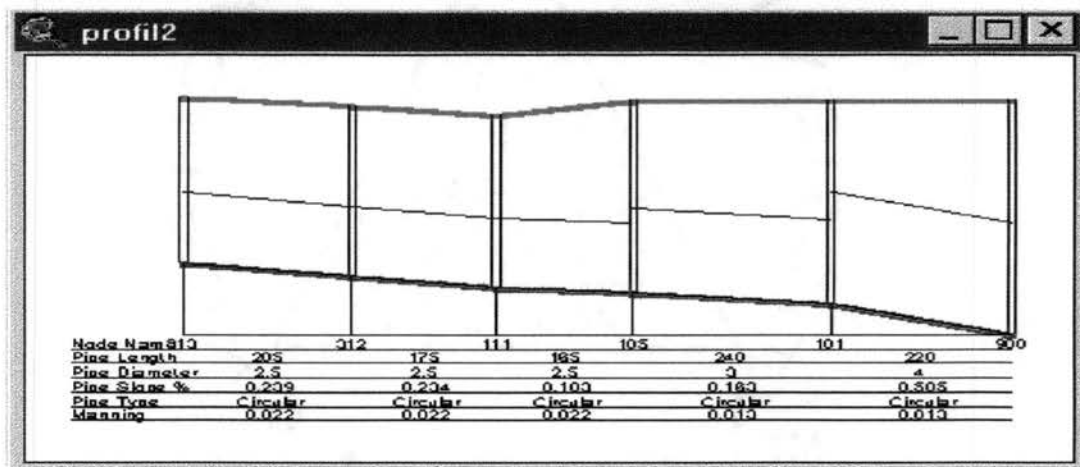
Figure 5.41 Longitudinal Profile of Network Displayed by PRESENTATION Extension

5.6.3 Flooded Areas Caused By Surcharging Manhole

As shown in Figure 5.40, the simulated results from XP-SWMM can describe the surcharging manhole according to set rules based on the hydraulic flow grade line. The PRESENTATION Extension's Zone Tool adds a button to the tool bar. This button when active allows one to specify a circle on the display and the region of elevations lower than the center of the circle is then drawn within the circle. A digital elevation model (grid of elevations), must be the active theme, although it does not have to be the theme displayed in the View window. Figures 5.42 and 5.43 show the application of the "**Z**" button to identify potential regions for flooding from a particular surcharging manhole. This "**Z**" button is shown in Figure 5.22
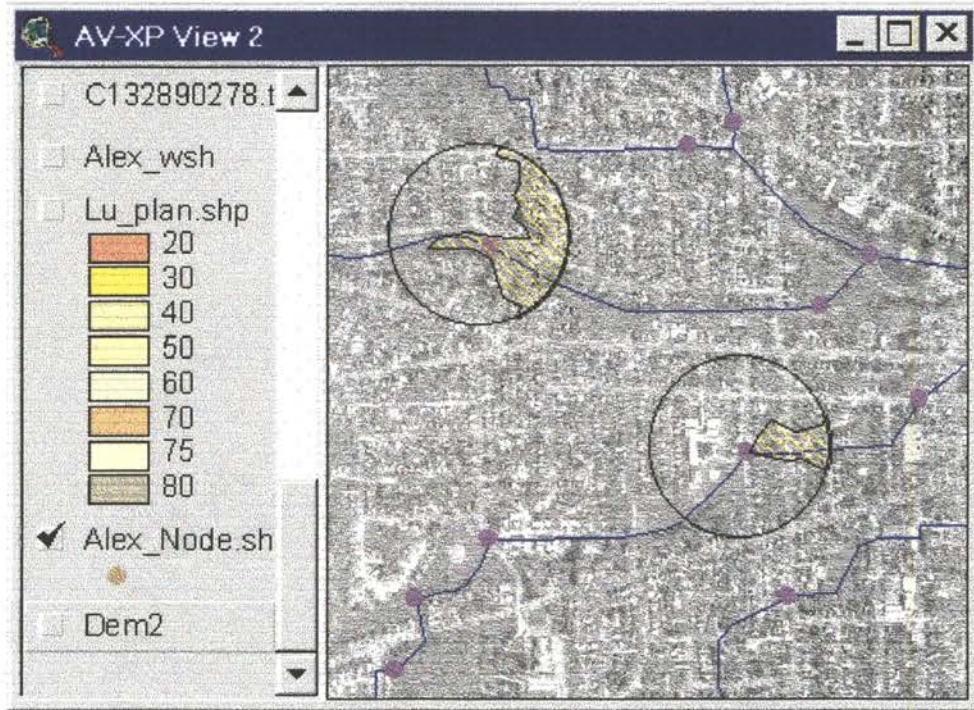
Figure 5.42. PRESENTATION (XP-Zones) Extension— "**Z**"
Tool Button Showing Flood-Prone Regions
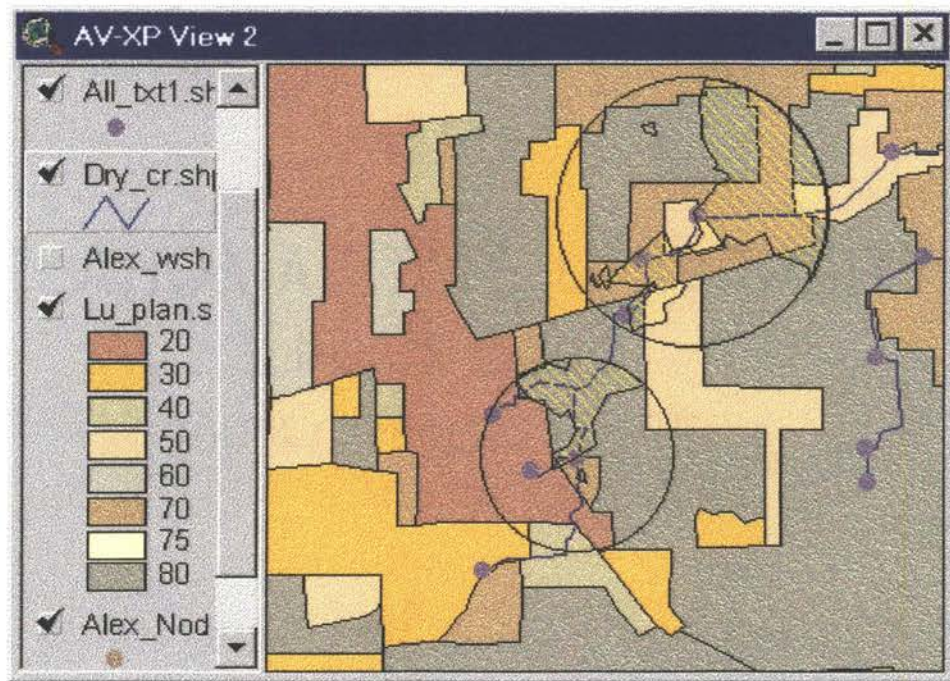
Figure 5.43. PRESENTATION (XP-Zones) Extension Showing
Application of "**Z**" Tool Button for Flood-Prone Regions

CHAPTER 6

VALIDATION OF RUNOFF RESULTS AND PARAMETERS GENERATED BY

MANUAL AND GIS PROCEDURES FOR URBAN WATERSHED

When a model is developed, such as generation of Runoff Block parameters from coverages of node, link, soil, land-use and DEM, the accuracy of the outcome is undoubtedly one of the most important measures of the model's worth. In this chapter, in order to make informed decisions and evaluations, derived values of the GIS generated variables (extracted from five GIS layers shown in Figure 6.1) were compared with manually-estimated values in a meaningful and quantitative way. A 10-year storm events was simulated by XP-SWMM using two sets of runoff input data for the same urban watershed. The runoff quantities for each subbasin using manually-estimated parameters were compared with the runoff quantities at the same subbasin utilizing GIS-extracted runoff parameters.

The manually-estimated data was developed by Black and Veatch Engineers (BV), as described earlier in Chapter 4. The statistical analysis performed in this section is presented in two parts: (1) comparison of values of GIS generated variables with manually-estimated variables; (2) comparison of discharges simulated by XP-SWMM using manually and GIS derived input data.

In Section 6.1, the manual topographic map measurements of the watershed characteristics are compared in order to verify the accuracy of the urban watershed characteristics quantified by the use of this interface between ArcView and XP-SWMM. Values of manually-estimated physical characteristics of the subbasins of Allen Creek Watershed are given in Table 6.1 and the values for the same variables extracted by GIS procedures—developed in this dissertation—are given in Table 6.2. Shape of each subbasin was delineated for the GIS-derived theme shown in Figure 6.2 and then subsequent three parameters (area, % slope, and width) were derived according to the methods described in Chapter 5. The comparisons between GSI-generated and manually-estimated values for these three parameters are provided in Table 6.3, 6.4, 6.5, 6.6 and 6.7.

Section 6.2 compares the discharges simulated by XP-SWMM using manually estimated data and GIS-derived data. The input data for Runoff Block of XP-SWMM, prepared by manually estimated procedures are given in Table 4.5. The two sets of discharges simulated for all subbasins for Allen Creek Watershed utilizing manually estimated input data and GIS-generated input data are given in Table 6.8.

6.1 Comparison Between GIS- and Manually-Estimated Parameters

The Allen Creek Watershed was used to validate the performance of the routines developed in this study. A DEM with a cell size of 42.08 x 42.08 sq.ft. was used to delineate drainage boundaries and to calculate subcatchment parameters. The major flow paths were forced to follow the direction defined by the lines of drainage system (bra1x.shp). The point map, Allnpp.shp, was used to define subcatchment outlets. The snap distance for subcatchment outlets of one cell size (42.0842 feet) was selected. Figure 6.1 shows primary GIS coverages used in this study.

Comparison analysis executed between manually estimated data made from hard maps and the results of GIS-derived data from ArcView coverages are shown in Tables 6.3 to 6.7. Figures 6.3, 6.4, 6.5, 6.6, and 6.7 show statistical analysis between manually-drawn parameters and GIS-generated parameters for the 46 subbasins of the Allen Creek Watershed. The manually estimated parameters are: Area (A_{Man}), Slope (S_{Man}) and Width (W_{Man}) and GIS-generated parameters are written as Area (A_{GIS}), Slope (S_{GIS}) and Width (W_{GIS}). The GIS-generated set of width parameters consists of three types: $W1_{GIS}$, $W2_{GIS}$, and $W3_{GIS}$, as described in Chapter 5.

6.1.1 Area

The data for the Allen Creek Watershed was obtained from the records of the City of Ann Arbor Utilities Department. This office lists 3398 acres as the total actual area of this watershed. The total manually-drawn watershed area (A_{Man}), given in Table 6.1, for all 46 subbasins is 3384.63 acres. The total GIS-generated watershed area (A_{GIS}), given in Table 6.2, is 3528.9 acres for 46 subbasins delineated by GIS spatial analysis tools. The total GIS-generated area (A_{GIS}) of the watershed is more than manually-estimated area (A_{Man}) by 144.27 acres. The difference between GIS-generated area and manually-estimated values is 4.0%. This size of difference between the magnitudes of A_{GIS} and A_{Man} is not very consequential to the runoff quantity estimation for an urban watershed. Figure 6.3 shows a regression analysis between manually estimated values for area (A_{Man}) and GIS-extracted quantities for area (A_{GIS}) for the Allen Creek Watershed.

GIS Input Data theme for Nodes (Manholes) for Allen Creek Watershed Showing 46 Runoff Collection Points (Allnpp.shp).

GIS Input Data Theme of Pipes (Links) for Allen Creek Watershed (bra1.shp)

Digital Elevation Model (DEM) for Allen Creek Watershed

GIS Coverage of Land Use for Allen Creek Watershed

GIS Coverage for Type of Soil for Allen Creek Watershed.
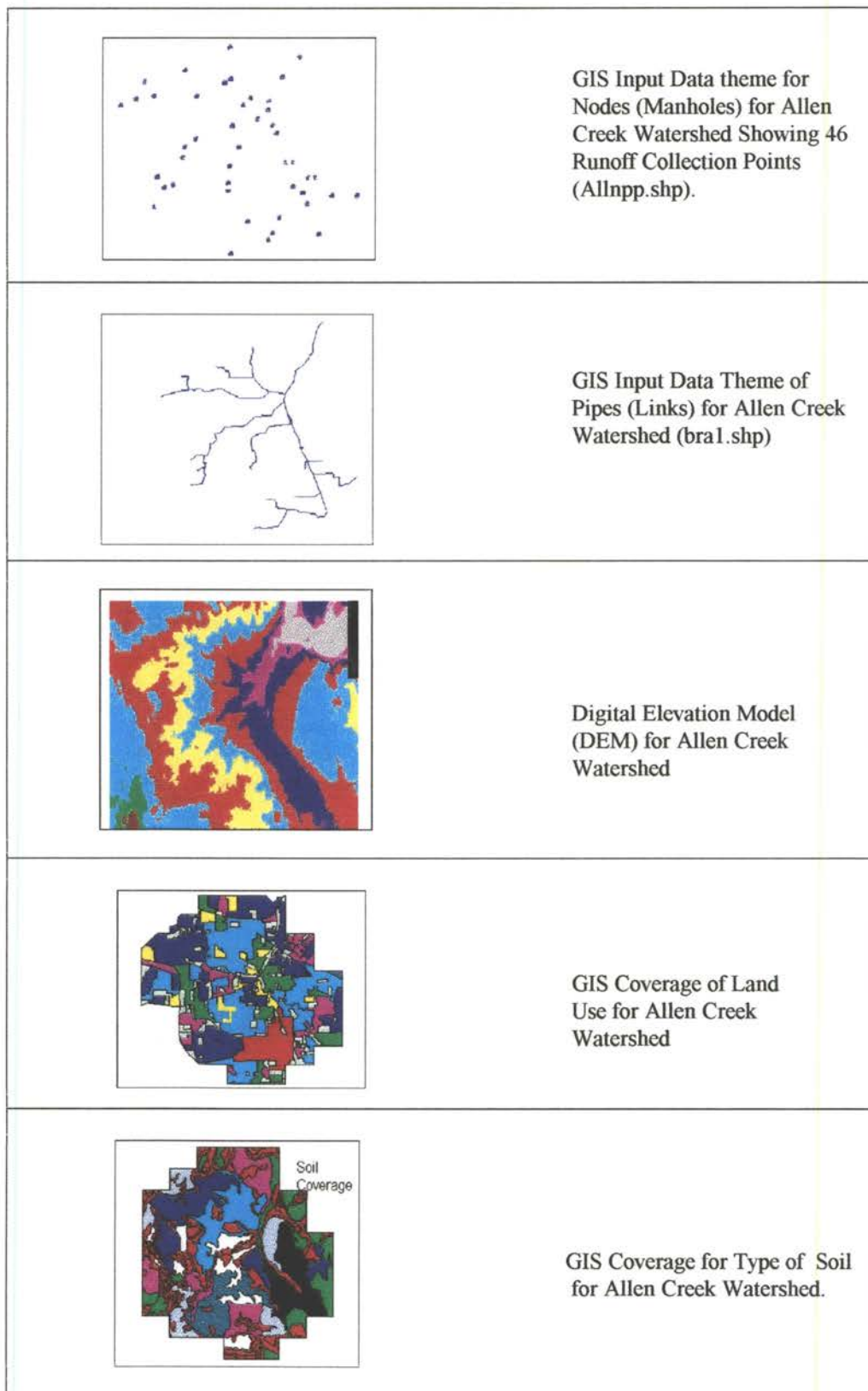
Figure 6.1. Five Basic GIS Data Themes for Allen Creek Watershed Used for the Generation of Watershed Parameters
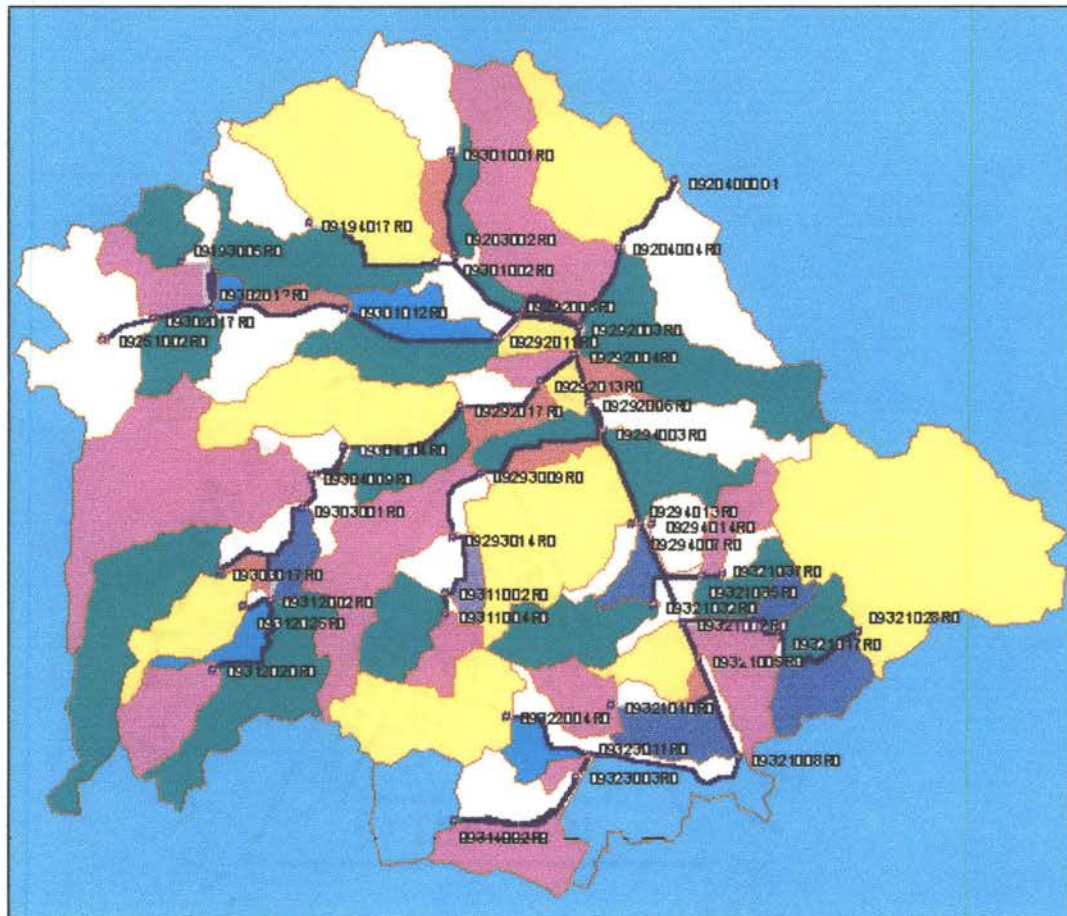
Figure 6.2. GIS Delineated 46 Subbasins, Manholes, and Pipes
Network for Allen Creek Watershed

The advantages of GIS parameter estimation lie in variables involving areas of subbasins and the only restriction on the accuracy is the digitization processes (including availability) of accurate maps and plans. The algorithms used in this study provide useful results about subbasin areas and other watershed variables. Half the number of subbasins had a difference between 1% and 30%, while 22 subbasins had a range of difference between 31% to 200%.

Two subbasins had a quantitatively major difference between their A_{GIS} and A_{Man} values. In case of Subbasin No. 09251002RO (item # 7), the manual estimation techniques generated 30.01 acres as its area. This manually-estimated value was 44 times higher than its GIS-estimated value of 0.69 acres. But in the case of Subbasin No. 09321035RO (item # 42), the GIS procedures generated 56.19 acres as its area, which was higher by an order of 2 than its manually-estimated value of 7.67 acres.

87
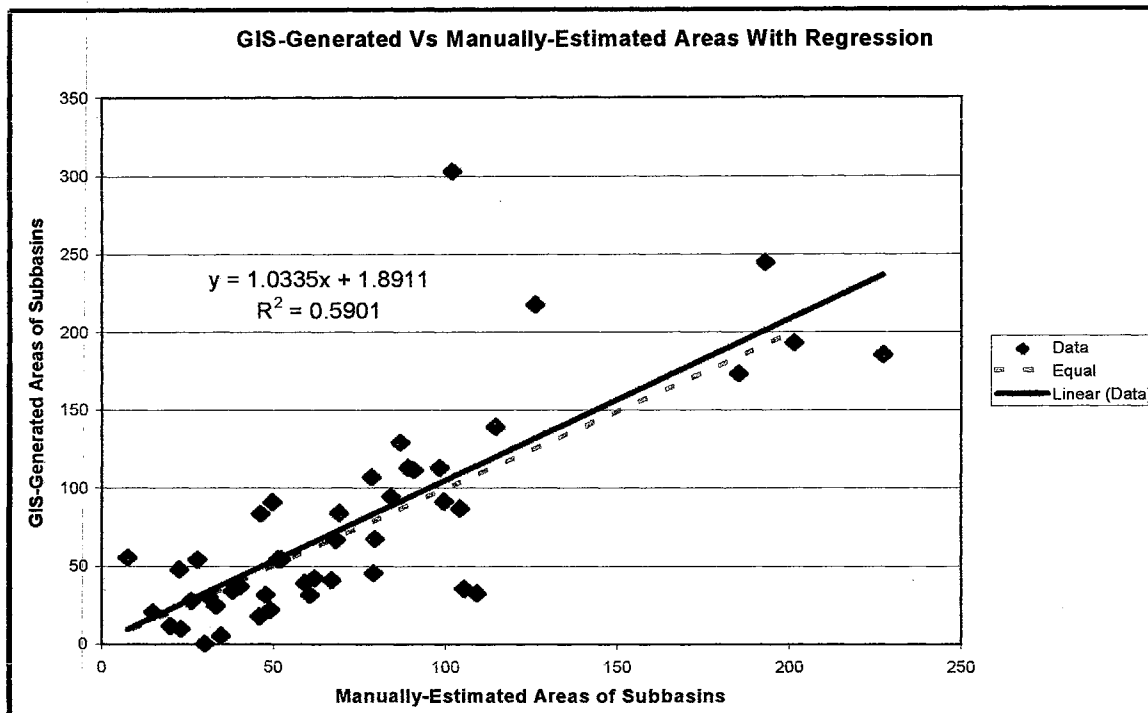
Figure 6.3. Comparison of Manually Drawn Subbasin Area (A_Man)
With GIS-Extracted Subbasin Area (A_GIS)

This type of difference is common in urban environments. Hydrologists can visually inspect the topography and altered drainage patterns of the subbasin. DEM, however, does not have this type of data incorporation facility and consequently will analyze and delineate all cells having higher elevations (runoff contributing cells) than the elevation of the manhole residing in a particular subbasin. The cause of these differences in the measurements of both types of areas might be also due to the incorporation of total area of a DEM cell, which partially resides on the delineated boundary of a subbasin or the watershed. The GIS procedures and methods of utilization of topographic information are different than the standard manual data development procedures for the estimation of runoff parameters. This disparity might be caused due to the imperfections of the digital terrain representation or its insufficient resolution (both horizontal and vertical), software limitations or subjective and incorrect traditional estimation techniques used by Black and Veatch. The traditional and GIS methods have inherent errors in their parameter generation processes and both techniques need calibration for runoff modeling.

The traditional procedures employed by Black & Veatch to prepare area from the contour map of the Allen Creek Watershed at a 10-foot interval and scale of 1:24000, might have produced unsatisfactory results. This urban watershed consists of berms, elevated roads, parks, rails, building, swales, culverts and several artificial runoff transferring systems. The manual delineation of subcatchments having sizes of 1 to 228 acres needs detailed field surveys, especially the smaller subbasins. In urban settings, a hydrologist can never be sure about the geographical location of the line dividing two subbasins, or subbasins with multiple slopes.

ArcView GIS estimates the subbasin area by counting the number of cells flowing through a subcatchment towards a particular manhole. While delineating the subbasin boundary, every cell of the DEM is counted even if it resides partially in the delineated boundary. The spatial analysis techniques consider this feature (dividing boundary/line) as having two subcatchments while hydrologist might consider it only a single subcatchment. There are differences in procedures and utilization of information by GIS and the hydrologist. The inclusion of total area—rather than partial area—of the saddling cell (common cell with multiple slope directions) might increase the overall area of a delineated subbasin. But type of inclusion results in decreasing the area of a neighboring subbasin or all subbasins. The difference might be pronounced when the actual areas of the subbasins consist of few acres in urban settings, and especially when the DEM has a relatively bigger cell size.

Man-made structures are mostly not digitized in an ordinary DEM. The user has to make some adjustments to the data by creating artificial paths, especially in urban settings. If the water is flowing from the one side of a road structure (underneath it) to a particular manhole located on the other side, the hydrologist will estimate a higher value of the subbasin area. Under this type of spatial analytical conditions, DEM will normally provide a less area as it will not consider the flow from other side of the road. It cannot automatically detect that subcatchment consists of two or more than two runoff contributing areas connected by pipes, culverts etc. This type of analysis cannot be done solely either from topographic maps, or by hydrologist or by GIS techniques. It needs a field survey. Although the hydrologist is better equipped to utilized the point-flow data, i.e., points through which runoff collected from one watershed is transferred to the other side of an man made interventions through underground conveyance (culverts) or open channels.

6.1.2 Slope

A comparison analysis was conducted between GIS-generated slope parameter (S_{GIS}) and manually-developed slope parameter (S_{Man}), which are listed in Table 6.4 and the regression analysis between them is shown in Figure 6.4. The average difference is about 50% between the manually-estimated and GIS-generated slope variables, i.e., quantitatively the manually-estimated slope parameter is approximately double than the GIS-generated slope parameter. Measurement differences for the subbasin slope ranged from 22.6% to 79.1%, with an average GIS underestimation of 50.0% for 44 drainage subbasins. The slope for two subbasins was not used in the analysis. The GIS-generated slope value for 092033013RO (item No. 4) was not available; and the manually-developed slope (S_{Man}) value for subbasin 09312020RO (item No. 32) had a non-practical value and was not utilized in statistical analysis.

This difference in slope magnitudes might be due to several reasons; such as, difference in estimation techniques used by Black and Veatch and GIS spatial analysis procedures or due to the use of different types of basic maps and information. A brief description is given which will assist in understanding the significance of difference between the slope values generated by the GIS spatial analysis procedures form a DEM and slope values traditionally estimated from a contour map of a watershed.

The terrain modifications in urban settings are the main causes to divert the flow direction whereas the surface runoff does not follow natural flow direction. In urban environments terrain is altered along the natural flow paths and runoff is forced prematurely into manholes. This type of surface features cannot be analyzed by a regular DEM. But hydrologist can analyze the situation from the site records and by conducting a site survey.

Manually developed overland-flow slope—the most troublesome part of a measurement on a topographic map—subjectively chooses the contour lengths (normal to the direction of flow). The runoff slope for a subbasin is obtained by calculating and averaging the elevation differences between high points and subbasin manhole and then averaging this elevation difference by distances along the flow direction toward the manhole. But the slope of a subbasin is calculated by estimating the slope of each connected cell of a DEM in the flow path and then averaging.

The manual procedures employed by the water resources community for the estimation of slope parameter are comparatively conservative and subjective, as less mathematical or geometrical details are considered. The hydrologist, while developing overland-flow slope procedure, subjectively chooses the contour lengths (normal to the direction of flow) and checks the contours of the subcatchment and estimates the vertical distances between contour lines by interpolation. The selection of locations of all points representing subbasin boundary and flow direction while manually developing the slope parameter is very tedious and subjective. The hydrologist considers that all the area between two contours has the same elevation, but in real life it is not true, i.e., each cell of a DEM with respect to its neighboring cells (especially with a 42x42 ft cell size) will not have the same elevation. This selection of a particular point and analyzing its consequences on the runoff direction depends on the experience of the hydrologist and the details provided on the topographic map. The BV-estimated slope parameters were developed from a dense urban map of a 10-foot contour interval at 1:24000 horizontal scale. The runoff path and consequently its length generated from such a map will generate slope vecters, which are inconsistent and non-reproducible. In an urban setting such as the Allen Creek Watershed, flow path definition and length calculation are subjective to the experience of the hydrologist and consequently prone to errors in slope parameter evaluation.

The difference in slope magnitudes can also be attributed to inadequate DEM resolution, and noise in data. Also GIS spatial analysis procedures generate the land surface slope by considering the directions of water flow. These trajectories represent the movement of water using a multiple-direction approach, which allows the accumulated upslope area from any cell to be distributed among 8 flow directions by weighted proportions based on slope angle and its aspect. Raster coverage for subbasin slope (*Flow Direction Grid*) is created by overlapping the raster coverages for subbasins and DEM. Percent slope is generated by fitting a plane to a pixel elevation and its eight neighboring, pixel elevations. The difference in elevation between the low and high points is divided by the horizontal distance, and multiplied by 100 to compute the percent slope for the pixel. Also the runoff length within a subbasin is obtained by calculating distances along the flow direction toward the manhole and then averaging. The GIS spatial analysis technique considers every cell of the DEM and consequently generates meandering flow path

direction. This sinuosity of flow path will always generate longer flow length and consequently a lower value of slope parameter.

The manual procedure mostly produces inconsistent results, when using a map of greater contour interval, which might be accurate, but definitely different from the GIS procedure. GIS results represent a real life situation while developing the DEM through the digitization process or aerial photogrammetry procedures. The GIS spatial analysis techniques are consistent in their selection of points due to the use of spatially designed features and algorithms.
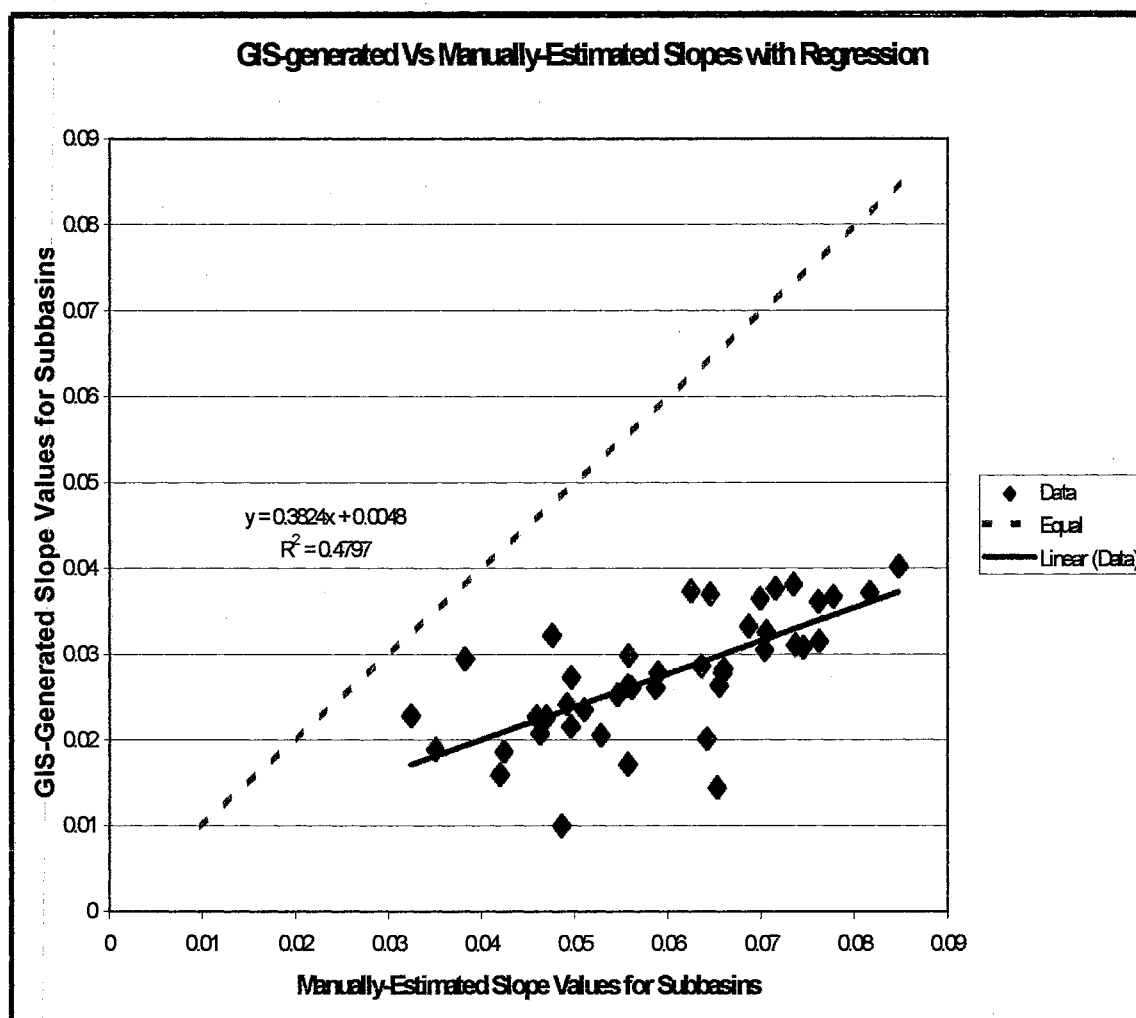
Figure 6.4. Analysis of Manually-Drawn Slope Variables (S_{Man}) and GIS-Generated Slope Variables (S_{GIS}) for 46 Subbasins of the Allen Creek Watershed.

The DEM for Allen Creek Watershed was built using a 10-foot contour map, and the cell size for this digitized DEM was 42 ft X 42 ft. It is recommended that for an urban setting these data might be augmented by digitizing 1-foot contour elevation maps of the watershed. The subbasin slope comparisons indicate that the 1:24,000-scale map with a 10-ft contour interval used to create DEM was incapable of reproducing the sinuosity of the elevation contour topographic maps. Subsequently the total length of the elevation contours are underestimated in the spatial analysis procedures for calculation of the subbasin slope.

An inherent problem in hydrologic modeling with grid DEM data is the production of nonphysical depressions due to noise in the elevation data that affect interpolation schemes used to describe raster point variations in elevation paths. The result is an undesirable termination of drainage paths in pits. Structural developments (buildings, parking lots, streets), graded or flat areas and other manmade features can change the natural topography of an urban watershed and alter surface elevations and consequently the surface slope.

6.1.3 Width

As described in Chapter 5, three procedures were written in Avenue to generate width parameters ($W1_{GIS}$), ($W2_{GIS}$), ($W3_{GIS}$) of a subbasin by GIS spatial analysis techniques. The first procedure has some similarity to the traditional techniques as described in the XP-SWMM manual. The traditional methods consider the subbasin with a rectangular shape and the runoff traveled distance is utilized as the length of the rectangular subbasin. The width of this ideal rectangle is used as the width parameter (W_{Man}) of the subbasin. This manual procedure was used by Black and Veatch to estimate the width W_{Man}) values for 46 subbasins, listed in Tables 6.1.

The GIS-generated Width variable ($W1_{GIS}$) was generated by averaging the GIS-estimated area of a subbasin by the length of the runoff path (L_{GIS}). The values for $W1_{GIS}$ for all 46 subbasins of the Allen Creek Watershed are provided in Table 6.2.

The second Width variable ($W2_{GIS}$) estimated by GIS spatial analysis techniques was calculated by dividing the area of the subbasin by the diameter of a circle. This circle must be fit (contained) within the GIS delineated boundary of the subbasin. The GIS-estimated values for $W2_{GIS}$ parameter for all subbasins are provided in Table 6.2.

93

TABLE 6.1. MANUALLY-DEVELOPED DATA OF 46 SUB-BASINS OF ALLEN CREEK WATERSHED

Serial No.	Node Name	Area (A_{Man}) (ac)	Width (W_{MAN}) (ft)	Slope (S_{MAN}) (%)
1	09193005R0	26.17	1462	0.04241
2	09194017R0	59.20	2500	0.06597
3	09203002R0	109.08	2520	0.07620
4	09203013R0	52.30	1875	0.08295
5	09204000D1	227.50	2840	0.07158
6	09204004R0	193.12	2844	0.05574
7	09251002R0	30.01	1160	0.06535
8	09292003R0	49.10	1000	0.06455
9	09292004R0	40.24	1840	0.05866
10	09292006R0	49.63	5670	0.05572
11	09292008R0	60.66	3500	0.06253
12	09292011R0	69.09	4590	0.07451
13	09292013R0	38.19	3000	0.06870
14	09292017R0	114.50	2913	0.07062
15	09293009R0	78.47	1625	0.06590
16	09293014R0	47.82	2400	0.06995
17	09294003R0	185.35	3000	0.04962
18	09294007R0	105.41	2660	0.05280
19	09294013R0	20.03	1330	0.07372
20	09294014R0	33.28	575	0.03822
21	09301002R0	126.08	4950	0.08171
22	09301012R0	68.07	3840	0.08481
23	09302012R0	46.27	2000	0.05460
24	09302017R0	98.25	1530	0.04676
25	09303001R0	66.83	3420	0.07353
26	09303017R0	89.18	6318	0.07041
27	09304004R0	45.92	1360	0.06359
28	09304009R0	201.51	2000	0.05892
29	09311002R0	61.95	1200	0.05578
30	09311004R0	15.01	1494	0.04629
31	09312002R0	90.79	3420	0.05099
32	09312020R0	79.11	2000	6.62600
33	09312025R0	22.60	4600	0.04914
34	09314002R0	51.31	2700	0.04957
35	09321002R0	84.06	3230	0.03508
36	09321005R0	52.17	2700	0.07777
37	09321008R0	86.81	3800	0.06552
38	09321010R0	31.24	2900	0.05610
39	09321017R0	79.39	3800	0.04198
40	09321028R0	101.96	5890	0.03240
41	09321032R0	27.94	3000	0.07625
42	09321035R0	7.67	330	0.06420
43	09321037R0	22.95	1346	0.04753
44	09322004R0	99.50	5160	0.04693
45	09323003R0	104.16	4725	0.04591
46	09323011R0	34.75	2554	0.04858

TABLE 6.2. GIS EXTRACTED DATA FOR 46 SUB-BASINS OF ALLEN CREEK WATERSHED.

Serial #	Node Name	(A_{GIS})	(L_{GIS})	(S_{GIS})	($W1_{GIS}$)	($W2_{GIS}$)	($W3_{GIS}$)
1	09193005R0	28.26	2289	0.01876	538	1357	907
2	09194017R0	39.64	3564	0.02842	484	1845	936
3	09203002R0	33.06	2678	0.03624	538	1861	774
4	09203013R0	—	—	—	—	—	—
5	09204000D1	186.18	6151	0.03777	1318	4362	1859
6	09204004R0	245.46	7892	0.02461	1355	7420	1441
7	09251002R0	0.69	186	0.01462	162	238	126
8	09292003R0	22.48	1833	0.03708	534	1225	800
9	09292004R0	37.41	2415	0.02620	675	1868	872
10	09292006R0	91.52	5267	0.01725	757	4511	884
11	09292008R0	32.20	2600	0.03749	540	1635	858
12	09292011R0	84.69	3916	0.03095	942	2365	1560
13	09292013R0	34.52	2289	0.03339	657	1596	942
14	09292017R0	140.07	5702	0.03268	1070	3603	1694
15	09293009R0	107.66	6115	0.02796	767	4224	1110
16	09293014R0	32.08	1962	0.03659	712	1418	985
17	09294003R0	174.06	6151	0.02741	1233	3601	2106
18	09294007R0	36.06	3083	0.02068	510	2127	739
19	09294013R0	12.20	2443	0.03121	218	1403	379
20	09294014R0	25.21	1504	0.02960	730	1134	968
21	09301002R0	218.38	5515	0.03731	1725	4292	2216
22	09301012R0	67.49	3768	0.04031	780	2274	1293
23	09302012R0	84.08	3105	0.02536	1180	3223	1136
24	09302017R0	113.52	4626	0.02250	1069	3885	1273
25	09303001R0	41.47	2418	0.03821	747	1578	1145
26	09303017R0	113.56	7437	0.03066	665	3783	1308
27	09304004R0	18.46	2415	0.02878	333	1142	704
28	09304009R0	193.78	8924	0.02799	946	4343	1943
29	09311002R0	42.77	2527	0.02997	737	1771	1052
30	09311004R0	21.06	1974	0.02087	465	1480	620
31	09312002R0	112.22	5373	0.02366	910	2973	1644
32	09312020R0	46.19	2845	0.03448	707	1812	1110
33	09312025R0	48.42	3518	0.02431	600	1961	1075
34	09314002R0	54.93	2653	0.02172	902	1702	1406
35	09321002R0	95.30	4148	0.01899	1001	3654	1136
36	09321005R0	55.09	3105	0.03687	773	2232	1075
37	09321008R0	129.70	5737	0.02649	985	3813	1482
38	09321010R0	30.66	2563	0.02624	521	1283	1041
39	09321017R0	68.02	3049	0.01611	972	2012	1473
40	09321028R0	303.64	5201	0.02296	2543	4487	2947
41	09321032R0	54.73	3544	0.03164	673	2337	1020
42	09321035R0	56.19	2952	0.02027	829	3340	733
43	09321037R0	10.12	1868	0.03229	236	953	463
44	09322004R0	92.17	3887	0.02286	1033	3176	1264
45	09323003R0	87.62	3466	0.02286	1101	2469	1546
46	09323011R0	5.90	1254	0.01014	205	872	295

Note: (GIS Derived Subbasin Variables): Area = A_{GIS}; Maximum Length = L_{GIS}; Slope = S_{GIS} ; Width-1 = A_{GIS} / L_{GIS} = $W1_{GIS}$; Width-2 = $A_{GIS} / Diameter_{max}$ = $W2_{GIS}$; Width-3 = $Diameter_{max}$ = $W3_{GIS}$

TABLE 6. 3: COMPARISON OF MANUALLY AND GIS GENERATED AREAL DATA FOR 46 SUBBASINS OF ALLEN CREEK WATERSHED.

Serial #	Node Name	Area-Manual A_{Man} (ac)	Area by GIS A_{GIS} (ac)	% Difference = 100 (A_{GIS} − A_{MAN}) ÷ A_{MAN}
1	09193005R0	26.17	28.26	+ 8.0
2	09194017R0	59.20	39.64	- 33.0
3	09203002R0	109.08	33.06	-69.7
4	09203013R0	52.30	—	NA
5	09204000D1	227.50	186.18	-18.2
6	09204004R0	193.12	245.46	+27.6
7	09251002R0	30.01	0.69	NA
8	09292003R0	49.10	22.48	-54.2
9	09292004R0	40.24	37.41	-7.0
10	09292006R0	49.63	91.52	+84.4
11	09292008R0	60.66	32.20	-46.9
12	09292011R0	69.09	84.69	+22.6
13	09292013R0	38.19	34.52	-9.6
14	09292017R0	114.50	140.07	+22.3
15	09293009R0	78.47	107.66	+37.2
16	09293014R0	47.82	32.08	-32.9
17	09294003R0	185.35	174.06	-6.1
18	09294007R0	105.41	36.06	-65.8
19	09294013R0	20.03	12.20	-39.1
20	09294014R0	33.28	25.21	-24.2
21	09301002R0	126.08	218.38	+73.2
22	09301012R0	68.07	67.49	-0.9
23	09302012R0	46.27	84.08	+81.7
24	09302017R0	98.25	113.52	+15.5
25	09303001R0	66.83	41.47	-37.9
26	09303017R0	89.18	113.56	+27.3
27	09304004R0	45.92	18.46	-59.8
28	09304009R0	201.51	193.78	-3.8
29	09311002R0	61.95	42.77	-36.0
30	09311004R0	15.01	21.06	+40.3
31	09312002R0	90.79	112.22	+23.6
32	09312020R0	79.11	46.19	-41.6
33	09312025R0	22.60	48.42	+114.2
34	09314002R0	51.31	54.93	+7.1
35	09321002R0	84.06	95.30	+13.4
36	09321005R0	52.17	55.09	+5.6
37	09321008R0	86.81	129.70	+49.4
38	09321010R0	31.24	30.66	-1.9
39	09321017R0	79.39	68.02	-14.3
40	09321028R0	101.96	303.64	+197.8
41	09321032R0	27.94	54.73	+95.9
42	09321035R0	7.67	56.19	+632.6
43	09321037R0	22.95	10.12	-55.9
44	09322004R0	99.50	92.17	-7.4
45	09323003R0	104.16	87.62	-15.9
46	09323011R0	34.75	5.90	-83.0

Total Area of the Watershed (Manually Developed) = A_{Man} = 3384.6 acres

Total Area of the Watershed (GIS Generated) = A_{GIS} = 3528.9 acres

Percent Difference = 100 (A_{GIS} — A_{Man}) ÷ A_{Man} = 100 (3528.9 − 3384.6) ÷ 3528.9 = 4.0%

TABLE 6.4. MANUALLY- AND GIS-GENERATED SLOPE DATA
FOR ALLEN CREEK WATERSHED

Serial #	Node Name	Slope Manually Developed (S_{Man}) (%)	Slope GIS Generated (S_{GIS}) (%)	% Difference
1	09193005R0	0.04241	0.01876	-55.8
2	09194017R0	0.06597	0.02842	-56.9
3	09203002R0	0.07620	0.03624	-52.4
4	09203013R0	0.08295	—	NA
5	09204000D1	0.07158	0.03777	-47.2
6	09204004R0	0.05574	0.02461	-55.84
7	09251002R0	0.06535	0.01462	-77.6
8	09292003R0	0.06455	0.03708	-42.6
9	09292004R0	0.05866	0.02620	-55.3
10	09292006R0	0.05572	0.01725	-69.0
11	09292008R0	0.06253	0.03749	-40.0
12	09292011R0	0.07451	0.03095	-58.5
13	09292013R0	0.06870	0.03339	-51.4
14	09292017R0	0.07062	0.03268	-53.7
15	09293009R0	0.06590	0.02796	-57.6
16	09293014R0	0.06995	0.03659	-47.7
17	09294003R0	0.04962	0.02741	-44.8
18	09294007R0	0.05280	0.02068	-60.8
19	09294013R0	0.07372	0.03121	-57.7
20	09294014R0	0.03822	0.02960	-22.6
21	09301002R0	0.08171	0.03731	-54.3
22	09301012R0	0.08481	0.04031	-52.5
23	09302012R0	0.05460	0.02536	-53.6
24	09302017R0	0.04676	0.02250	-51.9
25	09303001R0	0.07353	0.03821	-48.0
26	09303017R0	0.07041	0.03066	-56.5
27	09304004R0	0.06359	0.02878	-54.7
28	09304009R0	0.05892	0.02799	-52.5
29	09311002R0	0.05578	0.02997	-46.3
30	09311004R0	0.04629	0.02087	-54.9
31	09312002R0	0.05099	0.02366	-53.6
32	09312020R0	6.62600	0.03448	-NA
33	09312025R0	0.04914	0.02431	-50.5
34	09314002R0	0.04957	0.02172	-56.2
35	09321002R0	0.03508	0.01899	-46.9
36	09321005R0	0.07777	0.03687	-52.6
37	09321008R0	0.06552	0.02649	-59.6
38	09321010R0	0.05610	0.02624	-53.2
39	09321017R0	0.04198	0.01611	-61.6
40	09321028R0	0.03240	0.02296	-29.1
41	09321032R0	0.07625	0.03164	-58.5
42	09321035R0	0.06420	0.02027	-68.4
43	09321037R0	0.04753	0.03229	-32.1
44	09322004R0	0.04693	0.02286	-51.3
45	09323003R0	0.04591	0.02286	-50.2
46	09323011R0	0.04858	0.01014	-77.1

TABLE 6.5. COMPARISON OF MANUALLY DEVELOPED WIDTH (W_{MAN}) VALUES AND GIS-GENERATED WIDTH-1 ($W1_{GIS}$) DATA OF ALLEN CREEK WATERSHED.

Serial #	Node Name	Manually Developed Width (W_{Man}) (ft)	GIS Generated $W1_{GIS} = (A_{GIS}) \div (L_{GIS})$	% Difference
1	09193005R0	1462	538	- 63.2
2	09194017R0	2500	484	- 80.6
3	09203002R0	2520	538	- 78.7
4	09203013R0	1875	—	NA
5	09204000D1	2840	1318	- 53.6
6	09204004R0	2844	1355	- 52.4
7	09251002R0	1160	162	- 86.0
8	09292003R0	1000	534	- 46.6
9	09292004R0	1840	675	- 63.3
10	09292006R0	5670	757	- 86.6
11	09292008R0	3500	540	- 84.6
12	09292011R0	4590	942	- 79.5
13	09292013R0	3000	657	- 78.1
14	09292017R0	2913	1070	- 63.3
15	09293009R0	1625	767	- 52.8
16	09293014R0	2400	712	- 70.0
17	09294003R0	3000	1233	- 58.9
18	09294007R0	2660	510	- 80.8
19	09294013R0	1330	218	- 83.6
20	09294014R0	575	730	+ 27.0
21	09301002R0	4950	1725	- 65.2
22	09301012R0	3840	780	- 79.7
23	09302012R0	2000	1180	- 41.0
24	09302017R0	1530	1069	- 30.1
25	09303001R0	3420	747	- 78.2
26	09303017R0	6318	665	- 89.5
27	09304004R0	1360	333	- 75.5
28	09304009R0	2000	946	- 52.7
29	09311002R0	1200	737	- 38.6
30	09311004R0	1494	465	- 68.9
31	09312002R0	3420	910	- 73.4
32	09312020R0	2000	707	- 64.7
33	09312025R0	4600	600	- 87.0
34	09314002R0	2700	902	- 66.6
35	09321002R0	3230	1001	- 69.0
36	09321005R0	2700	773	- 71.3
37	09321008R0	3800	985	- 74.1
38	09321010R0	2900	521	- 82.0
39	09321017R0	3800	972	- 74.4
40	09321028R0	5890	2543	- 56.8
41	09321032R0	3000	673	- 77.6
42	09321035R0	330	829	+ 151.2
43	09321037R0	1346	236	- 82.5
44	09322004R0	5160	1033	-80.0
45	09323003R0	4725	1101	- 76.7
46	09323011R0	2554	205	- 92.0

TABLE 6.6. COMPARISON OF MANUALLY DEVELOPED WIDTH (W_{MAN}) AND GIS-GENERATED WIDTH ($W2_{GIS}$) BY WIDTH-2 PROCEDURE

Serial #	Node Name	Manually Developed Width (W_{Man}) (ft)	$W2_{GIS} = A_{GIS} \div$ Fitting Circle Diameter (ft)	% Difference
1	09193005R0	1462	1357	- 7.2
2	09194017R0	2500	1845	- 26.2
3	09203002R0	2520	1861	- 26.1
4	09203013R0	1875	—	NA
5	09204000D1	2840	4362	+ 53.6
6	09204004R0	2844	7420	+ 160
7	09251002R0	1160	238	- 79.5
8	09292003R0	1000	1225	+ 22.5
9	09292004R0	1840	1868	+ 1.5
10	09292006R0	5670	4511	- 20.4
11	09292008R0	3500	1635	- 53.3
12	09292011R0	4590	2365	- 40.5
13	09292013R0	3000	1596	-46.8
14	09292017R0	2913	3603	- 23.7
15	09293009R0	1625	4224	+ 160.0
16	09293014R0	2400	1418	- 40.9
17	09294003R0	3000	3601	+ 20.0
18	09294007R0	2660	2127	- 20.0
19	09294013R0	1330	1403	+ 5.5
20	09294014R0	575	1134	+ 97.2
21	09301002R0	4950	4292	- 13.3
22	09301012R0	3840	2274	- 40.8
23	09302012R0	2000	3223	+ 61.2
24	09302017R0	1530	3885	+ 154.0
25	09303001R0	3420	1578	- 53.9
26	09303017R0	6318	3783	- 40.1
27	09304004R0	1360	1142	- 16.0
28	09304009R0	2000	4343	+ 117.1
29	09311002R0	1200	1771	+ 47.6
30	09311004R0	1494	1480	- 0.9
31	09312002R0	3420	2973	- 13.0
32	09312020R0	2000	1812	- 9.4
33	09312025R0	4600	1961	- 57.4
34	09314002R0	2700	1702	- 37.0
35	09321002R0	3230	3654	- 13.1
36	09321005R0	2700	2232	- 17.3
37	09321008R0	3800	3813	+ 0.3
38	09321010R0	2900	1283	- 55.8
39	09321017R0	3800	2012	- 47.1
40	09321028R0	5890	4487	- 23.8
41	09321032R0	3000	2337	- 22.1
42	09321035R0	330	3340	NA
43	09321037R0	1346	953	- 29.2
44	09322004R0	5160	3176	- 38.4
45	09323003R0	4725	2469	- 47.7
46	09323011R0	2554	872	- 65.9

99

TABLE 6.7. COMPARISON OF MANUALLY-DEVELOPED WIDTH (W_{MAN}) AND GIS-GENERATED WIDTH ($W3_{GIS}$) BY WIDTH-3 PROCEDURE.

Serial #	Node Name	Width Manually Developed W_{Man} (ft)	$W3_{GIS}$ = Contained Circle Diameter (ft)	% Difference
1	09193005R0	1462	907	- 38.0
2	09194017R0	2500	936	- 62.6
3	09203002R0	2520	774	- 69.3
4	09203013R0	1875	—	—
5	09204000D1	2840	1859	- 34.5
6	09204004R0	2844	1441	- 49.3
7	09251002R0	1160	126	- 89.1
8	09292003R0	1000	800	- 20.0
9	09292004R0	1840	872	- 52.6
10	09292006R0	5670	884	- 84.4
11	09292008R0	3500	858	- 75.5
12	09292011R0	4590	1560	- 66.6
13	09292013R0	3000	942	- 68.6
14	09292017R0	2913	1694	- 41.8
15	09293009R0	1625	1110	- 31.7
16	09293014R0	2400	985	- 59.0
17	09294003R0	3000	2106	- 29.8
18	09294007R0	2660	739	- 72.2
19	09294013R0	1330	379	- 71.5
20	09294014R0	575	968	+ 68.3
21	09301002R0	4950	2216	- 55.2
22	09301012R0	3840	1293	- 66.3
23	09302012R0	2000	1136	- 43.2
24	09302017R0	1530	1273	- 16.8
25	09303001R0	3420	1145	- 66.5
26	09303017R0	6318	1308	- 79.3
27	09304004R0	1360	704	- 48.2
28	09304009R0	2000	1943	- 3.0
29	09311002R0	1200	1052	- 9.6
30	09311004R0	1494	620	- 58.5
31	09312002R0	3420	1644	- 51.9
32	09312020R0	2000	1110	- 44.5
33	09312025R0	4600	1075	- 76.6
34	09314002R0	2700	1406	- 47.9
35	09321002R0	3230	1136	- 64.8
36	09321005R0	2700	1075	- 60.2
37	09321008R0	3800	1482	- 61.0
38	09321010R0	2900	1041	- 64.1
39	09321017R0	3800	1473	- 61.2
40	09321028R0	5890	2947	- 49.7
41	09321032R0	3000	1020	- 66.0
42	09321035R0	330	733	+ 122.0
43	09321037R0	1346	463	- 65.6
44	09322004R0	5160	1264	- 75.5
45	09323003R0	4725	1546	- 67.3
46	09323011R0	2554	295	- 88.4

In the third case, the diameter of the circle contained within the delineated boundary of the subbasin was used as the width parameter ($W3_{GIS}$) of the subbasin. The GIS-estimated values for $W3_{GIS}$ parameter for all subbasins are provided in Table 6.2. One subbasin 09203013R0 (Item #4) out of 46 subbasins is not used in the statistical analysis. This subbasin was not incorporated into the analysis as its GIS data could not be generated.

The measurement differences between $W1_{GIS}$ and W_{Man} for subbasins are listed in Table 6.5 and a comparison analysis is shown in Figure 6.5. The Width parameter values $W1_{GIS}$ for 43 subbasins generated by GIS analysis were less than the respective manually-generated values (W_{Man}) and they varied between 30.1% and 92%. Only in case of two subbasins, (Item # 20, & 42), the values of $W1_{GIS}$ were higher than the values of W_{Man}, which were 27.0% and 151.2% respectively. The average value difference for $W1_{GIS}$ was 70% less than W_{Man}. This comparison difference between manual and GIS measurements is due mainly to the differences in the procedures involved in the width measurement. The manual procedures for W_{Man} estimation consider subbasin shape as an ideal rectangle. But GIS spatial analysis techniques do not consider it a rectangle as the subbasins have generally an irregular shape.

Figure 6.5. Comparison of GIS-Extracted Width Variable ($W1_{GIS}$)
with Manually-Developed Width Variable (W_{Man}).

In the third case, the diameter of the circle contained within the delineated boundary of the subbasin was used as the width parameter ($W3_{GIS}$) of the subbasin. The GIS-estimated values for $W3_{GIS}$ parameter for all subbasins are provided in Table 6.2. One subbasin 09203013R0 (Item #4) out of 46 subbasins is not used in the statistical analysis. This subbasin was not incorporated into the analysis as its GIS data could not be generated.

The measurement differences between $W1_{GIS}$ and W_{Man} for subbasins are listed in Table 6.5 and a comparison analysis is shown in Figure 6.5. The Width parameter values $W1_{GIS}$ for 43 subbasins generated by GIS analysis were less than the respective manually-generated values (W_{Man}) and they varied between 30.1% and 92%. Only in case of two subbasins, (Item # 20, & 42), the values of $W1_{GIS}$ were higher than the values of W_{Man}, which were 27.0% and 151.2% respectively. The average value difference for $W1_{GIS}$ was 70% less than W_{Man}. This comparison difference between manual and GIS measurements is due mainly to the differences in the procedures involved in the width measurement. The manual procedures for W_{Man} estimation consider subbasin shape as an ideal rectangle. But GIS spatial analysis techniques do not consider it a rectangle as the subbasins have generally an irregular shape.

Figure 6.5. Comparison of GIS-Extracted Width Variable ($W1_{GIS}$) with Manually-Developed Width Variable (W_{Man}).

In the third case, the diameter of the circle contained within the delineated boundary of the subbasin was used as the width parameter ($W3_{GIS}$) of the subbasin. The GIS-estimated values for $W3_{GIS}$ parameter for all subbasins are provided in Table 6.2. One subbasin 09203013R0 (Item #4) out of 46 subbasins is not used in the statistical analysis. This subbasin was not incorporated into the analysis as its GIS data could not be generated.

The measurement differences between $W1_{GIS}$ and W_{Man} for subbasins are listed in Table 6.5 and a comparison analysis is shown in Figure 6.5. The Width parameter values $W1_{GIS}$ for 43 subbasins generated by GIS analysis were less than the respective manually-generated values (W_{Man}) and they varied between 30.1% and 92%. Only in case of two subbasins, (Item # 20, & 42), the values of $W1_{GIS}$ were higher than the values of W_{Man}, which were 27.0% and 151.2% respectively. The average value difference for $W1_{GIS}$ was 70% less than W_{Man}. This comparison difference between manual and GIS measurements is due mainly to the differences in the procedures involved in the width measurement. The manual procedures for W_{Man} estimation consider subbasin shape as an ideal rectangle. But GIS spatial analysis techniques do not consider it a rectangle as the subbasins have generally an irregular shape.

Figure 6.5. Comparison of GIS-Extracted Width Variable ($W1_{GIS}$) with Manually-Developed Width Variable (W_{Man}).

The width parameter is the first trial number and its value is adjusted later on by the calibration process during the simulations of XP-SWMM. It derives its value either through measurement and/or interpretation of available information. In manual procedures, the value assigned to width parameter represents macro-effects with a single number, so their derivation is open to debate. Sources of information in arriving at a single number representing a width parameter are; for instance, topographic maps, aerial photographs, and land use maps. It is practically difficult to survey the width parameter for an irregular shaped watershed.

So it is a futile effort to make a scientific comparison between an adjusted width parameter and the GIS generated parameter. The values of the width parameter generated by GIS routine might be better than the manually estimated values as neither one of them is in a better mathematical position than the other. A better value can only be verified through validation process of the model. If an automated procedure is incorporated for the generation of input data for watershed runoff modeling, it might be advisable to use the GIS generated values for width parameter and any subsequent modification will render to the validation process.

6.2 Peak Flows Analysis for GIS Versus Manual Runoff Estimation Scenarios

This software toolbox was tested by incorporating a hydrologic study for the Allen Creek Watershed consisting of 46 subcatchments. Based on Ann Arbor Design Standards, a 10-year design storm was used as a basis for conveyance system analysis and SCS 24 hour distribution was selected as input to XP-SWMM for the estimation of runoff quantities for all 46 subcatchments while using manually estimated parameters. The same single event rainfall stormwater was chosen for all the 46 collection points (subcatchments) while developing the GIS scenario.

6.2.1 Variability in Runoff Estimation by GIS and Manual Procedures

The purpose of the GIS model verification was to achieve a level of accuracy in the spatial analysis—GIS—computations consistent with the level of detail required for manual estimation. Peak runoff for each subwatershed was verified by comparing the XP-SWMM generated subareas flows (using GIS input data) with previous runoff calculations using the traditional procedures.

6.2.1.1. Differences in Flow Values—Individual Manholes. Table 6.8 shows both sets of flows (Q_{GIS} and Q_{Man}) simulated for each of the 46 subcatchments. The average flow difference was 26.5 cfs (the difference is defined as manually-estimated Q_{MAN} minus GIS-estimated Q_{GIS}.). The Root Mean Square Difference (RMSD) was 42.4 cfs. The much lower magnitude of the average difference than the RMSD indicates that positive differences flow values for some subcatchments tend to offset the negative differences in flow values for other subcatchments. The absolute maximum and minimum differences in flow values were 155 cfs and 1 cfs, respectively.

On a percent difference basis, the average percent difference was 20.9. There were two subbasins whose results were significantly worse with respect to the results for other 44 subbasins. The GIS techniques did not generate any area for Subbasin (092030013R0) but they generated (A_{GIS}) an area of 56.7 acres for Subbasin No. 09321035RO, which was higher by an order of 2 from the manually-drawn area (A_{Man}).

So, if only the runoff simulated (Q_{Man}) by using manually-estimated input data is considered, then Subbasin (092030013R0) had a difference of 100% (as Q_{GIS} was zero and Q_{Man} was 118 cfs for this subbasin). The Subbasin No. 09321035RO had a 407% difference between Q_{GIS} (38 cfs) and Q_{Man} (193 cfs). Not including these two, the absolute maximum and minimum percent difference were 20 % and 26%, respectively.

This type of difference is common in urban environments. Hydrologists can visually inspect the topography and the altered drainage patterns of the subbasin. DEM, however, does not have this type of data interpretation tools and consequently will analyze and delineate all cells having higher elevations than the elevation of the manhole residing in a particular subbasin, and it will consider them as runoff contributing cells.

Figure 6.8 shows that majority of the differences in flow quantities were relatively small, except for the two above mentioned subbasins (Item No.4 (Subbasin 092030013R0) and Item # 42 (Subbasin No. 09321035RO). The distribution of differences between individual subbasins is shown in the Figure 6.8. The positive differences are the most undesirable, since underestimating flows would result in an inadequate system. The plot shows there is a very low incidence of

105

positive differences. Figure 6.9 shows 63% of all differences are between ±20 cfs and 43% of all differences are between minus 20 cfs and zero.

6.2.1.2 Difference in Overall Flow. To determine the overall difference, the percent difference in the flow at the outlet of the Allen Creek Watershed was determined. The total discharge (Q_{MAN}) simulated by using manually-estimated parameters was 7,684 cfs and .the total discharge (Q_{GIS}) simulated by using GIS-estimated parameters was 8,457 cfs. The total difference in runoff quantities for Allen Creek watershed between Q_{GIS} and Q_{Man} was 773 cfs. The average percent difference between both types of runoff estimation techniques was 20.9 cfs.

The total flow difference between Q_{GIS} and Q_{Man} was 10%. The fact that this difference is smaller overall than the differences in individual watersheds again shows that the individual differences tend to offset each other. Since the flows are overestimated, a design based on this GIS model would be a conservative one.

6.2.1.3 Regression Analysis. Figure 6.8 shows the results of the regression. A linear regression of GIS-generated Vs manually-observed watershed discharges showed a small systematic multiplicative difference, which indicates the GIS-based model tends to over-predict the flows. The linear regression for the runoff for 46 subcatchments gives a correlation coefficient R of 0.96. An $R^2 = 0.93$ implied that approximately 96% of the variability is accounted for by the given variables with the remainder of 7% variability being unaccountable.

According to the results, the GIS predicted runoff volumes are highly correlated with manually measured runoff quantities for this 10-year storm and the GIS model performance for the Allen Creek Watershed is acceptable. In addition, the 7% unaccountable variability may be attributed to a number of causes such as input data error, logic error of the model, or incorrect estimates of parameter values.

6.2.2 Validity of Subcatchment Flow Generated by GIS-Derived Data

Calibration results indicate the GIS model developed for the Allen Creek Watershed correlates well with previous estimated flow data. Results for 10-year GIS- and manually-simulated subcatchment runoff quantities are summarized in Table 6.8. The total accumulated GIS-generated flow (Q_{GIS}) for Allen Creek Watershed is within 10% of the previously estimated flow (Q_{Man}) (generated by using traditional estimation procedures).

TABLE 6.8. COMPARISON OF PEAK DISCHARGES SIMULATED BY XP-SWMM UTILIZING INPUT DATA GENERATED BY GIS AND TRADITIONAL METHODS.

Subbasin	Manually Estimated	GIS Extracted	Comparison of Subbasin Flow Quantities	
No.	Flow (Q_{Man}) cfs	Flow (Q_{GIS}) cfs	Flow Difference (cfs) = (Q_{Man} − Q_{GIS})	Difference %
1	114	125	-11	-9
2	506	630	-124	-20
3	200	252	-52	-21
4	118	0	118	+100
5	82	87	-5	-6
6	144	158	-14	-9
7	63	75	-12	-16
8	23	26	-3	-12
9	73	74	-1	-1
10	220	253	-33	-13
11	74	78	-4	-5
12	142	154	-12	-8
13	112	108	+4	-4
14	431	502	-71	-14
15	209	248	-39	-16
16	135	138	-3	-2
17	54	62	-8	-13
18	237	272	-35	-13
19	92	103	-11	-11
20	159	153	+5	+4
21	131	141	-10	-7
22	179	156	+23	+15
23	210	237	-27	-11
24	92	83	+8	+11
25	494	575	-81	-14
26	177	158	+19	+12
27	93	102	-9	-9
28	396	421	-25	-6
29	225	246	-21	-9
30	127	148	-21	-14
31	39	42	-3	-7
32	248	270	-22	-8
33	210	196	+14	+7
34	104	118	-14	-12
35	175	189	-14	-7
36	238	254	-16	-6
37	232	219	+13	+6
38	114	129	-15	-12
39	259	305	-46	-15
40	95	87	+8	+9
41	89	78	+11	+14
42	38	193	155	+ 407
43	237	286	-49	-17
44	89	87	+2	+2
45	95	114	-19	-17
46	110	125	-15	-12

Total Manually Estimated Discharge for the Allen Creek Watershed (Q_{Man})   = 7,684 cfs
Total GIS Generated Discharge for the Allen Creek Watershed (Q_{GIS})   = 8,457 cfs
Total Difference in Discharge Quantity Generated by both Techniques   = 773   cfs
Total percent Difference in Discharge Quantity Generated by both Techniques   = 10   %
Average Difference (absolute values) = 26.5 cfs   Average % Difference (absolute)   = 20.9   %
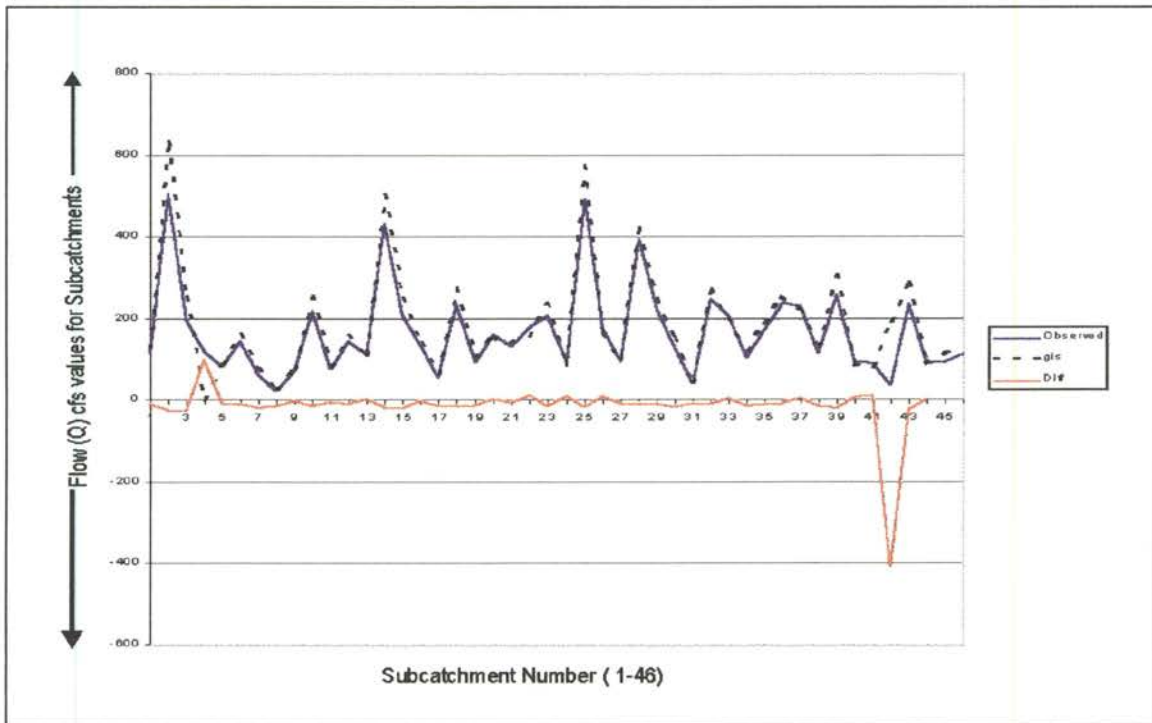
Figure 6.8. Differences in Runoff Quantities for 46 Subcatchments Estimated by GIS and Manual Procedures for the Allen Creek Watershed
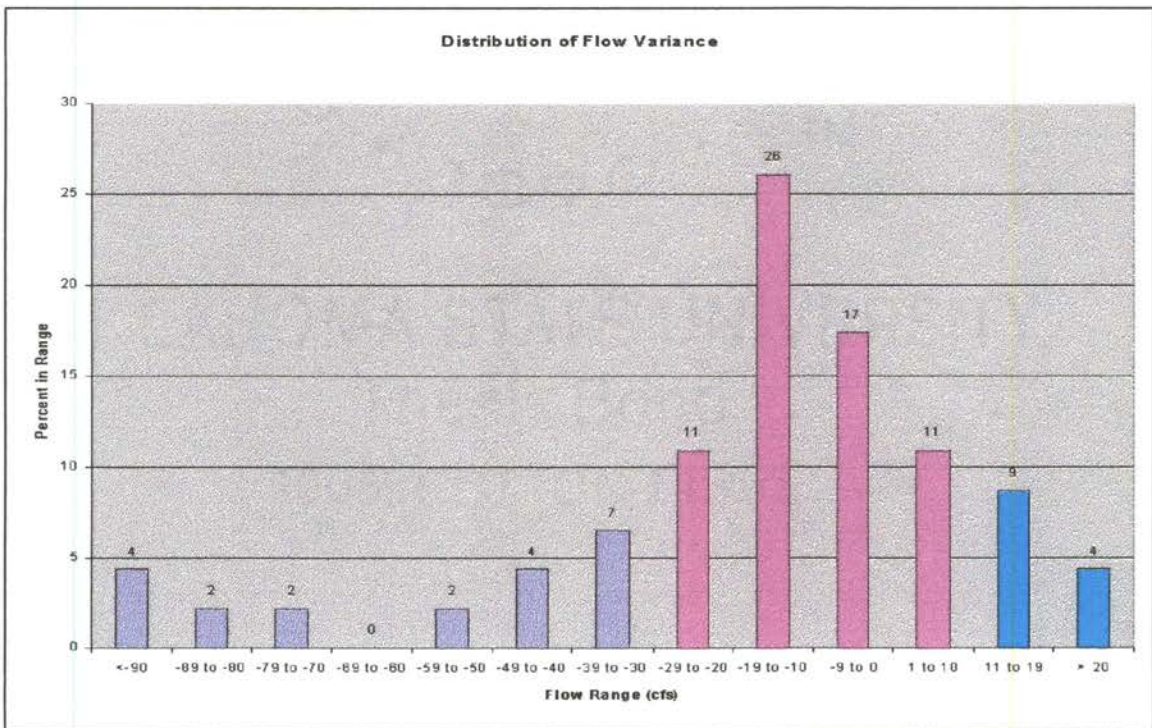
Figure 6.9.　Percent Range Distribution of Flow Differences for 46 Subbasins
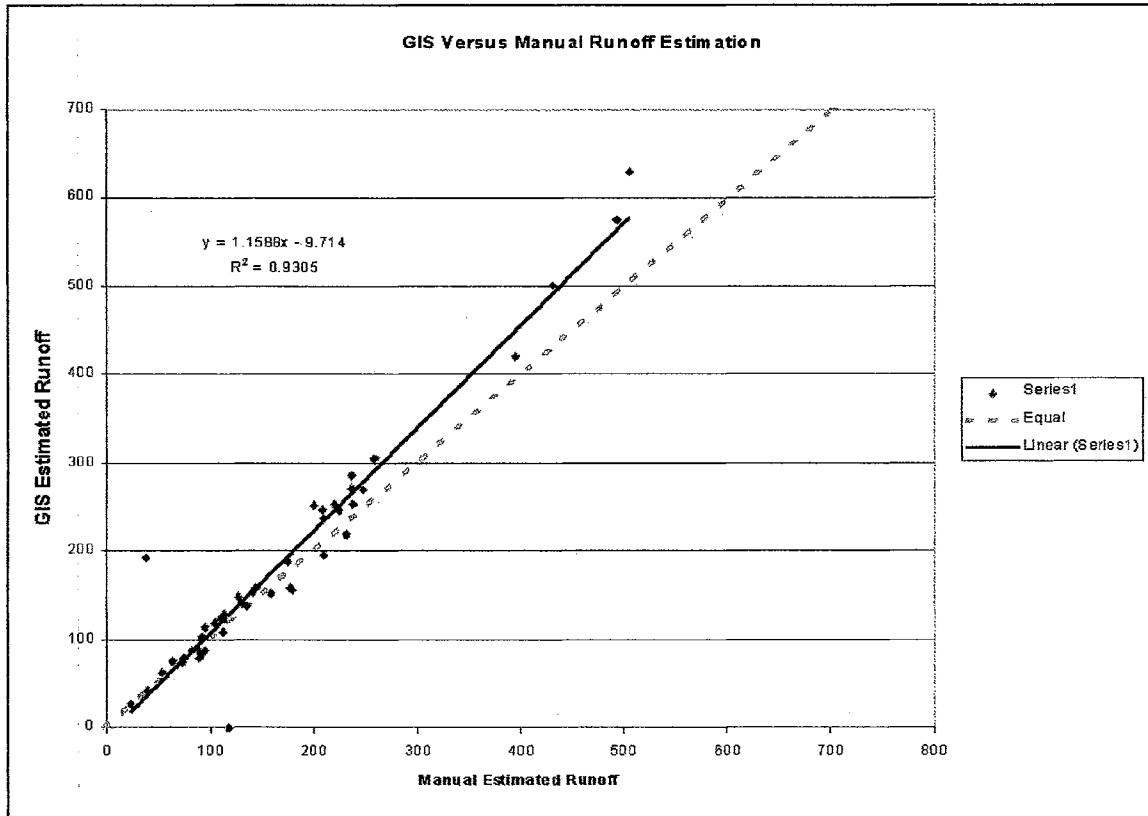
Figure 6.10. Regression Analysis Between Runoff Generated
by GIS and Traditional (Manual) Procedures

The coefficient of determination (R^2 = 0.93) is computed between GIS flow and manual flow as a measure of variability. Comparing the subwatershed runoff derived from digital elevation data using this integration tool and subwatershed manually delineated from topographic maps shows a close agreement. This analysis suggests that the area, slope, width, and other surface cover classifications for the sub-watershed are accurate and are the primary factors affecting the volume of runoff. The regression analysis also indicates the model performs quite well with GIS spatial parameters when compared with traditionally produced data.

However, GIS procedures slightly overestimated runoff. This analysis demonstrated comparison between two alternative runoff estimation scenarios, but it is difficult to speculate which scenario is optimal, as enough comparative data are not available.

CHAPTER 7

DISCUSSION OF RESULTS

This chapter presents conclusions, a general discussion about software interface procedures, results, and recommendations for further research. This research was intended to illustrate the coupling of a widely used urban surface water model XP-SWMM with a PC-based GIS, ArcView, and it provides guidelines to enhance the usefulness of both software packages. The linkage created by this GUI between XP-SWMM and ArcView successfully performs spatial analysis on the necessary GIS input data required for watershed parameterization and supports the data exchange between both software packages. This interface primarily offers front- and back-end applications of GIS to XP-SWMM. The front-end application includes the computation of watershed parameters for the Runoff Block of XP-SWMM. The back-end application includes the cartographic display of computed hydrologic data generated by XP-SWMM.

The Allen Creek Watershed, City of Ann Arbor, Michigan was used as the study area in this dissertation for the generation of GIS-based watershed physical parameters and for verification and validation of the accuracy of the software procedures. The derived watershed parameters by software tools are sufficiently accurate as their utilization in XP-SWMM modeling generated although higher runoff quantities than the observed discharges, but the difference is within a reasonable range.

7.1 ArcView Extensions and Data Development Procedures

ArcView is a desktop GIS capable of pre-processing, spatial analyzing, manipulating, displaying, and post-processing geographic data, as well as simulated results from XP-SWMM of any urban watershed, stormwater or sewer system. An ArcView map, termed as "*shape*" is managed by a database that is spatial in orientation. This spatial data base keeps track of map features such as points (manholes), lines (pipes) or polygons (types of soils, land uses etc) by generating topographic

and topological data known as feature's attributes (location, length, area, elevation, and neighboring or connecting features).

The spatial analysis capabilities of ArcView form the basis of the methodologies developed in this dissertation. A combination of ArcView, Spatial Analyst, Avenue and ARC/INFO was used in this dissertation. The ARC/INFO was used only to digitize hydrological map, sewer, and land use maps etc., which were used as base coverages (input data) by ArcView. All of the routines developed in this GUI are written in Avenue. Spatial Analyst Extension performs most of the spatial analysis, and it operates on DEM that provides modeling, and displays capabilities for terrain surface, and it is the basic coverage for the delineation of subcatchment and slope estimation.

The grid-based watershed delineation process produces a digital representation of areas draining to manholes. The concept in the watershed delineation is the use of DEM to the direction of flow over the surface of the watershed. The basis of this concept, is the application of the "eight direction pour point model (D-8)". The D-8 model is based on the gravity theory that a raindrop falling on a cell can move towards any one of the eight directions with the steepest slope. Once this direction of the overland runoff is established, known as Flow Direction Grid, the Spatial Analyst accumulates the flow down to a given manhole lying within a subbasin by counting the number of cells upstream that discharge into that particular manhole.

In this study, three ArcView extensions; (1) **WATERSHED** (2) **NETWORK**, and (3) **PRESNETATION** are built. They are the three main tools which generate, support data exchange between ArcView GIS and XP-SWMM software, and display the results. A brief summary is provided about each of them in the next paragraphs.

7.1.1 WATERSHED Extensions

The **WATERSHED** adds a new menu, which contains three items: (1) **Input**, (2) **Calculate**, and (3) **Export**. This extension constructs a *Flow Direction Grid*, delineates subbasins, calculates subbasin parameters (such as area, width, slope, and percent area impervious), and creates XP export file to allow the transfer of GIS generated parameters for subbasins into the XP-SWMM.

Menu item **Input** develops the *Grid of Flow Direction* and delineates subbasins. The *Elevation Grid (DEM)*, and *Point Theme* are required to calculate flow direction and to delineate subbasins. The *Point Theme* represents subbasin manholes and *Grid Theme* shows elevations. The adjustment of the *Elevation Grid (DEM)* before *Flow Direction* is calculated. For the adjustment of elevation of the

111

cells of *DEM*, several algorithms are written, which generate *Flow Direction Grid* to incorporate those features into flow analysis which are usually not represented by raw *DEM*. The *"Burn-In Stream"* procedure was written which adjusts DEM by forcing the flow direction calculated from *DEM* to follow the natural flow paths. This procedure lowers the elevation of those cells of the *DEM* that represent flow path or raises the elevation of those cells that do not represent the stream. This procedure actually creates artificial "ditches" in *DEM* by raising the elevations of all cells in *DEM Grid* not representing real streams by an arbitrary value. A *Flow Barrier* procedure is also incorporated in this software. Although it is similar to the *"Burn-In-Stream"* procedure, but it raises the elevation of the particular cells of the DEM which represent *Flow Barriers* and prevent the passage of flow through them. Another algorithm incorporates ArcView's procedure *Fill* and this function removes all the sinks from DEM surface to ensure that the whole area under study contributes to the generation of runoff.

The *Flow Direction Grid* is a necessary procedure for the delineation of subbasins. The subbasins are defined by their corresponding manholes. For the delineation of different set of subbasins within the same *DEM Grid*, different themes (maps) of subbasin manholes are to be created and used for subcatchment delineation. Before the derivation of *Flow Accumulation Grid*, a routine executes *SnapPourPoint* Function, which utilizes the *"Snap-Outlets"* option by ensuring the location of manholes on the major flow paths. This procedure snaps pour points to the cell of a maximum value in the weight grid (*Flow Accumulation Grid*) within a specified snap distance. This procedure searches—within the specified snap distance around each manhole—the cell of the highest accumulated flow and moves the subbasin manhole to that particular location. The recommended value of the snap distance is between 1 to 3 to the cell size of the DEM for urban watersheds.

After the subbasins are delineated, spatial analysis routines start procedures to develop area, width, slope and other subcatchment parameters by executing various operations on selected themes. The *DEM, Link* and *Node Themes* are the major input themes that are required for subcatchment parameter estimation procedures. The *Elevation Grid (DEM)* as well as *Flow Direction Grid* are required for the estimation of subcatchment slope parameter. The *DEM* provides elevation for each cell and the *Flow Direction Grid* estimates the longest flow path. The code written for the conversion of file format and data take over after the derivation of surface runoff parameters. These routines store subcatchment parameters in .XPX files, which are subsequently read by XP-SWMM.

7.1.2 NETWORK Extension

The NETWORK Extension adds XP-Network to the View's menu. It creates and edits a basic drainage network (nodes and links) in ArcView environment, which is subsequently saved in an export file. Also it converts and stores ArcView's *shapes* representing *nodes* and *links* into an XP-SWMM readable file. The "New Nodes" menu item creates a new *Point Shape* of points, which represents nodes as manholes, junctions, or outlets. The "*New Links*" menu item creates a new *Line Shape* of *conduits*, ditches, or streams.

This extension also extracts a drainage network from XP-SWMM export file and its data, network geometry and other pertinent information are read by selecting either one of the menu items "*Import Nodes*," "*Import Links*," or "*Import Nodes+Links*." The menu item "*Export Nodes+Links*" executes a procedure, which converts shape files into an XP-SWMM export file.

7.1.3 PRESENTATION Extension

This extension presents results from XP-SWMM in the form of maps, profiles, and flood prone zones. This extension adds three items to the menu bar: (1) *Junctions*, (2) *Conduits*, and (3) *Profiles*.

The first two items, *Junctions*, and *Conduits* import the results of hydrologic and hydraulic calculations of XP-SWMM into ArcView. Once the drainage system calculations are imported from XP-SWMM and stored in ArcView's format, the tools developed in this research manipulate the imported data and display them as graphical presentations of results.

The *Profile* item shows a map of nodes and conduits created using information stored in the XPX export file and the XPX simulation output file. The manholes are classified into two groups. One group contains manholes in which the calculated hydraulic head is below the ground elevation. The second group is composed of over-flowing manholes. Conduits are classified into three groups. Classification of each conduit is based on the ratio of simulated discharge by XP-SWMM to the design discharge.

This extension also introduces a Flood-Prone Area Identification Procedure. The Z Button in the menu bar identifies this application and displays the region having lower elevation than a specified surcharging manhole. It identifies and delineates the potential areas, which can be inundated.

7.2 Validation of Urban Watershed Runoff and Its Parameters

In this study, the previously manually-estimated watershed parameters and runoff data set were compared with the corresponding GIS-generated data set, were for the Allen Creek Watershed. This watershed contains 46 subcatchments and it is representative of the drainage network for any urban settings. It provides the information needed to create a detailed description of realistic urban environment on which the developed routines were tested.

The GIS procedures and methods of utilization of topographic information are different than the standard manual data development procedures for the estimation of runoff parameters. Some of the man-made structures cannot be digitized in DEMs, especially if the DEM is developed from remotely sensed data or by automatic procedures. Runoff flowing from one side of a road structure (underneath it) to a particular manhole located on the other side, results in increased quantity of runoff of the subcatchment. Under this type of topographical condition, DEM will normally provide a less area as it will not consider the flow from other side of the road. In urban settings, many subcatchments have two or more than two runoff contributing subareas connected by pipes, culverts etc. This type of surface flow does not follow a natural path, rather it follows an altered paths. This is opposite to the spatial analysis techniques performed on a DEM.

Subbasin shape is very important for runoff volume estimation. ArcView GIS estimates the subbasin area by counting the number of cells flowing through a subbasin towards a particular manhole. While delineating the subbasin boundary, every cell of the DEM is counted even if it resides partially in the delineated boundary. This inclusion of total cell area might increase the overall area of a particular delineated subbasin, which causes a relative decrease in the areas of the adjacent subbasins. The percent difference might be pronounced when the affected subbasin has an area of few acres. The comparison between manually- and GIS-generated subbasin parameters indicated several similar situations in Allen Creek Watershed, as discussed in Chapter 6.

7.2.1. Area

A verification study between GIS- and manually-estimated quantities for subbasin area was conducted. The comparison showed that the GIS-generated total area of the subbasins and the actual surveyed area of the watershed has only a difference of 4.0%. Three-fourth of subbasins had a difference between 1 and 50%, while the other subbasins had a difference ranging between 51% to

200%. This disparity might be caused due to the imperfections of the digital terrain representation or its insufficient resolution (both horizontal and vertical), software limitations or subjective and incorrect manual estimation techniques. The traditional and GIS methods have inherent errors in their parameter generation processes, and both of them need calibration for runoff modeling.

The manually-estimated data from the contour map of the Allen Creek Watershed with a 10-foot interval and a scale of 1:24000, might produce unsatisfactory results. This watershed consists of roads, parks, rails, gullies, swales, culverts and other artificial runoff transferring and obstruction systems. The manual delineation of subcatchments having sizes of 1 to 200 acres needs detailed field surveys. In urban settings, a hydrologist can never be sure about the geographical location of the line dividing two subbasins, or subbasins with multiple slopes.

But the spatial analysis on a DEM considers this feature (dividing boundary/line) as belonging to two subcatchments while hydrologist might consider it only a single subcatchment. There is difference of procedure and utilization of information by GIS and the hydrologist. This inclusion of total area—rather than partial area—of the saddling cell (common cell with multiple slope directions) might increase the overall area of a delineated subbasin. But this inclusion results in decreasing the areas of the neighboring subbasins. The difference might be pronounced when actual areas of affected subbasins consist of few acres and a DEM with a relatively bigger cell size is used.

Man-made structures are mostly not digitized in an ordinary DEM. The user has to make some adjustments to the data by creating artificial paths. If the water is flowing from the one side of a road structure (underneath it) to a particular manhole located on the other side, the hydrologist will estimate a higher value of the subbasin area. Under this type of spatial analytical conditions, DEM will normally provide a less area as it will not consider the flow from other side of the road. It cannot automatically detect that the subcatchment undergoing a spatial analysis consists of two or more than two runoff contributing subareas connected by pipes, culverts etc.

This type of analysis cannot be done solely either from topographic maps, or by hydrologist or by GIS techniques. It needs a field survey. Although the hydrologist is better equipped to utilized the point-flow data, i.e., points through which runoff collected from one watershed is transferred to the other side of man made interventions through underground conveyance systems (culverts) or open structures (open channels).

7.2.2 Slope

The average value of the manually-estimated slope parameter was approximately double than the GIS-generated slope parameter. The measurement differences for the GIS subbasin slope ranged from 22.6% to 77.6%, with an average underestimation of 50%. This difference in slope magnitude might be due to the difference of estimation techniques involved by the manual and GIS procedures.

The manual procedures to estimate the slope parameter are very conservative and subjective as less mathematical or geometrical details are considered. This difference in slope magnitude can also be attributed to inadequate DEM resolution, and digitizing error. As described earlier in Section 7.2.1, the manually-estimated slope parameters were developed from a dense urban map with a 10-foot contour interval and a 1:24000 horizontal scale. Because in an urban setting such as the Allen Creek Watershed, flow path definition and length calculation are subjective to the experience of the hydrologist and consequently prone to errors in slope data evaluation.

The runoff path and subsequently its length generated by DEM are different than the one created by traditional methods. The GIS spatial analysis technique considers every cell of the DEM and consequently generates meandering natural runoff flow path direction. This sinuosity of flow path will always generate longer flow length and consequently a lower value of the slope parameter. GIS spatial analysis procedures generate the land surface slope by considering the directions of water flow from a cell to an adjacent cell. These trajectories represent the movement of water using a multiple-direction approach, which allows the accumulated upslope area from any cell to be distributed among 8 flow directions by weighted proportions based on slope angle and its aspect.

Raster coverage for subbasin slope (*Flow Direction Grid*) is created by overlapping the raster coverages for subbasins and DEM. Percent slope is generated by fitting a plane to a pixel elevation and its eight neighboring, pixel elevations. The difference in elevation between the low and high points is divided by the horizontal distance, and multiplied by 100 to compute the percent slope for the pixel. Also the runoff length within a subbasin is obtained by calculating distances along the flow direction toward the manhole and then averaging. The subcatchment slope is calculated by estimating the slope of each connected cell in each flow path in a subcatchment and then averaging.

While, manually developed overland-flow slope procedure subjectively chooses the contour lengths (normal to the direction of flow). The hydrologist checks the contours of the subcatchment and estimates the vertical distances. Mostly he will use interpolation, especially estimating the vertical

116

distances between contour lines. While developing the slope data manually, the selection of the location representing runoff boundary point is very subjective. The hydrologist considers that all the area between two contours has the same elevation, but physically and topographically it is not true. This selection of a particular point and analyzing its consequences on the runoff direction depends on the experience of the hydrologist and the details provided on the topographic map.

The manual procedures mostly produce inconsistent results, which might be accurate but definitely different from the results provided by GIS procedures. GIS results represent a real life situation while developing the DEM through the digitization process, automatic or photogrammetry procedures. The GIS spatial analysis techniques are consistent in their selection of points due to the use of spatially designed features and algorithms.

7.2.3 Width

In this study, three new methodologies were developed to generate width parameters of a subbasin by GIS spatial analysis techniques which were different than the traditional width estimation procedure. The traditional methods consider the subbasin with a rectangular shape and its width is used as the width parameter (W_{Man}).

One of the GIS-based width variable, ($W1_{GIS}$), averages the GIS-estimated area of a subbasin by the length of the runoff path (L_{GIS}). The second GIS width variable, ($W2_{GIS}$), calculates width by dividing the area of the subbasin by the diameter of a circle. This circle is contained within the GIS delineated boundary of the subbasin.   In the third case, the diameter of the circle contained within the delineated boundary of the subbasin is used as the width parameter ($W3_{GIS}$) of the subbasin.

The width parameter values $W1_{GIS}$ for subbasins generated by GIS analysis were less than the respective manually-generated values (W_{Man}) and the average value difference $W1_{GIS}$ was 70% less than W_{Man}. Comparison performed between GIS-generated width parameters ($W2_{GIS}$) and the width parameter estimated manually ($W2_{GIS}$) showed an average difference of 41%. However, the second GIS technique to generate width parameter ($W2_{GIS}$) provided better results than $W1_{GIS}$. The third GIS generated width parameters ($W3_{GIS}$) had an average difference of 58 % from the W_{Man}. This comparative study showed similarity of results for GIS width parameter, $W3_{GIS}$ and $W1_{GIS}$.

The width parameters (W_{GIS}) generated by GIS spatial analysis did not correlate well with the previously manually-estimated width (W_{Man}) parameters. The difference in width parameter measurement between manual and GIS estimations is due to the differences in the methodologies of

117

the measurement of width. The manually developed width parameters were estimated according to the traditional procedures and information provided in the XP-SWMM's Manual. The manual procedure to estimate width considers the shape of the subbasin as an ideal rectangular shape.

The width parameter is the first trial number and its value is adjusted later on by the calibration process during the simulations of XP-SWMM's Runoff Block. It derives its value either through measurement and/or interpretation of available information. In manual procedures, the value assigned to width parameter represents macro-effects with a single number, so their derivation is open to debate. Sources of information in arriving at a single number representing a width parameter are; for instance, topographic maps, aerial photographs, and land use maps. It is practically difficult to survey the width parameter for an irregular shaped watershed.

The values of the width parameter generated by GIS routine might be better than the manually estimated values as neither one of them is in a better mathematical position than the other. A better value can only be verified through validation process of the model. If an automated procedure is incorporated for the generation of input data for watershed runoff modeling, it might be advisable to use the GIS generated values for width parameter and any subsequent modification will render to the validation process.

7.2.4. Runoff

The runoff analysis demonstrates two alternative runoffs estimation scenarios (GIS versus Manual estimation procedures), but it is difficult to speculate which scenario is optimal. The linear regression for runoff gives a Coefficient Correlation (R) of 0.96. The Coefficient of Determination (R^2) is computed between GIS flows and manual flows as a measure of variability and a value of R^2 = 0.93 implies that approximately 96% of the variability is accounted for by the given variables with the remaining 7% variability being unaccountable. This suggests that area, slope, width and other surface cover classifications for subbasin are accurate and are the primary influence affecting the volume of the runoff from a catchment.

The validation results indicate the runoff, for Allen Creek Watershed, generated by GIS correlates well with the previously observed flow data which used manually-estimated input parameters. The total accumulated GIS-generated flow is within 10% of the previously observed watershed runoff quantity. This shows a close agreement, although the GIS runoff procedures overestimate the runoff quantity with respect to the previously estimated flow value for this watershed.

118

CHAPTER 8

CONCLUSION AND RECOMMENDATIONS

The main objective of this research was to investigate how a Graphical User Interface (GUI) can be created between ArcView GIS and a Surface Water Management Model (XP-SWMM). Both of these desktop—PC based—software packages were chosen as they are widely used by the water resources and engineering community.

The principal task of this dissertation was to use ArcView to generate an input data group related to the catchment surface characteristics. If the necessary input data for GIS, such as DEM, drainage plan, soil and land-use maps of the watershed are digitized, and if the algorithms provided in this GUI are utilized, it is possible to have ArcView correctly calculate area, length, width, slope, and other topographic attributes of subbasins contributing runoff to surface water collection system.

8.1 Conclusion

This interface exploits the capabilities of Avenue for developing the code for subbasin delineation, extraction of runoff parameters, conversion of ArcView's *shape files* into XP-SWMM export files and transferring the GIS generated parameters for subbasins into XP-SWMM for hydrological and hydraulic modeling. The data from XP-SWMM are transferred to ArcView via *XPX* export files. Finally, ArcView executes various manipulations on the imported output and displays it in the form of charts, themes, tables, and profiles.

The interface is able to accurately generate the terrain attributes of the watershed by using GIS spatial analysis techniques performed on digital elevation model (DEM). The validation study conducted on two sets of watershed parameters—GIS-derived versus manually-estimated— for Allen Creek Watershed, Ann Arbor, Michigan. Both set of parameters were not calibrated and the original data sets were used for both types of runoff estimations. The comparison conducted on runoff quantities generated by both sets of parameters showed an average difference of 10 percent, which

119

is quantitatively not significant for a hydrological study performed on an urban watershed.

The use of GIS is recommended for the generation of watershed parameters and for runoff estimation, because GIS spatial analysis techniques are less subjective and produce consistent results. Also, slope and width parameters are basically calibration parameters used by the SWMM users, and their use as the first pre-calibration numbers should be preferred. Their values can be adjusted with subsequent simulation in the validation process. Anyhow, the GIS generated parameter values are not final, but due to a non-subjective estimation process rendered by GIS techniques, they might be better numbers than the manual estimated values to start the validation process of urban runoff modeling by XP-SWMM.

This GUI has successfully established an initial link between ArcView GIS and XP-SWMM. This linkage, once fully interfaced, will provide a toolbox to support the dynamic task of watershed management. The complete work will develop all runoff related input data by GIS spatial analysis, which will definitely improve efficiencies of cost and time more than the traditional methods. Future research, and further refinement in this link are proposed.

8.2 Recommendations

As the initial connection to the XP-SWMM model is established, the possibility for further refinement and advancement in the link is completely possible. The next goal could be to develop the pollutant loading parameters for the Runoff Block and to run the Extran Block of XP-SWMM so that impacts on land uses and engineering practices to the receiving waters could be observed directly through ArcView. This proposed research should develop computer software to meet the following potential objectives:

The first objective will be to develop computer programs so that GIS and SWMM can be interfaced and data can be interchanged between them. The second objective will be to estimate pollutant loadings at individual sewersheds to identify critical sewersheds, determine total pollutant loadings from individual land use polygons, and obtain the total amount of nonpoint sources in each sewershed. The third objective will be to propose an effective mitigation strategy to decrease the amount of nonpoint pollutants from critical sewersheds.

The final objective of future research will be to establish a GIS model for watershed parameter generation and pollutants loading, with a link to SWMM or XP-SWMM so that overall connection can be used as a powerful watershed management tool.

REFERENCES

Adamus, C.L. and J.B. Martinus, 1995. Estimating nonpoint source pollution loads with a GIS screening model. *Water Resources Bulletin* 31(4):647-655.

Aldrich, J.A. and L.A. Roesner, 1986. PC-RUNOFF and DBASE III for storm data management, planning, and design. *Proc. ASCE 4th Conference on Computing in Civil Engineering*, 134-147, ASCE, New York.

Backhaus, R., and G. Braun (1998). Integration of remotely sensed and model data to provide the spatial information basis for sustainable landuse. *Acta Astronautica, 42, 541.*

Baffaut, C. and J.W. Delleur, 1989. Expert system for calibrating SWMM. *Journal of Water Resources Planning and Management*, ASCE, 115(3), 278-298.

Bammet, Y.Z., and C. Fulcher (1998). An interactive tool or watershed information management and water quality assessment. *Proc, Watershed management--moving from theory to implement*, Denver, CO. 1199-1204.

Ball, J.E., and K.C. Lok (1998). Modeling of spatial Variability of rainfall over a catchment. *Journal of hydrological Engineering*, 3 (2) 122-128.

Barber, J., K. Lage, and Carlon, P., 1994. Stormwater management and modeling: Integrat-ing SWMM and GIS. *URISA 1994 Annual Conference Proceedings*, Milwaukee, WI, Vol. 1, p. 310.

Bishop, G.D., M.R. Church, and C. Daly (1998). Effects of improved precipitation estimates on automated runoff mapping: Easter United States. *J. Am. Water Resour. Assoc.*, 34 (1), 159.

Black and Veatch Engineers, (1996). 8400 ward Parkway, Kansas City, Missouri 64114

Bonta, J.V. (1998). Spatial variability of runoff and soil properties on small watersheds in similar soil-map units. *Trans. Am. Soc. Agric. Eng.*, 41-575.

Bryant, S.D., T. Young, and J. Nicholson (1998). Stormwater Infrastructure Inventory and GIS database design. Proc, Watershed management--moving from theory to implementation, Denver, CO, 1183.

Burrough, P.A., W. van Deursen, and G.H. Heuvelink, 1988. Linking spatial process models and GIS: A marriage of convenience or blossoming partnership. *GIS/LIS* 88:598-607.

CaiCE Software Corporation, 1998. XP-SWMM 32 and Visual Hydra. 410 Ware Blvd, Suite 1200, Tampa, FL 33619-4439. Phone: 813/620-1444, Fax: 813/620-9019.

Cedra Corporation, 1999. AvSand Software for Runoff Estimation, 65 West Broad Street, 7th Floor, Rochester, NY 14614,

Cera, T. B. T.K. Remwel, and R.W. Burleson, Sept. 1996. Use of ARC/INFO, EPA-SWMM, and UNIX text processing tools to determine flood extent. *Symposium Proceedings, GIS and water resources,* American Water Resources Association, pp. 285-294.

Curtis, G.T., 1994. SwmmDuet, an AML to facilitate hydrologic modeling for the EPA storm water management model (SWMM). *Proceedings of the Fourteenth ESRI User Conference,* May 1994.

Danish Hydraulic Institute (DHI), 1999. MOUSE Interface with ArcView GIS. Agern Alle 5, DK – 2970 Horsholm, Denmark, e-mail: mouse@dhi.dk

Debarry, P.A., and Carrington, J.T., July 1990. Computer watersheds. *Civil Engineering Magazine,* ASCE, pp. 68-70.

DeVantier, B. A., and Feldman, A. D.,1993. Review of GIS application in hydrologic modeling. *J. Water resources planning and Management,* ASCE, 119(2):246-261.

DeVries, J.J. and T.V. Hromdka, 1993. Chapter 21: Computer models for surface water. *Handbook of Hydrology.* D. R. Maidment, Ed. McGraw Hill, New York, NY, 23.1-23.24.

Di Luzio, M., R. Srinivasan, and J.G. Arnold, 1997. Integrated user interface for SWAT using ArcView and Avenue. *American Society of Agricultural Engineers* (2), 16.

ESRI, 1998. *ArcView 3.1. Spatial Analyst, Extension, and Avenue,* ESRI, Redlands, CA.

ESRI, 1996. *Cell-based modeling with GRID 6.1: Supplement—hydrologic and distance modeling tools.* ESRI, Redlands, CA.

Evans, B. M. and D.A. Miller, 1988. Modeling nonpoint pollution at the watershed level with the aid of a geographic information system. Nonpoint pollution: 1988—policy, economy, management, and appropriate technology. *American Water resources Association,* Milwaukee, WI.

Evans, B.M. and E. Nizeyimana, 1998. GIS-based quantification of statewide NPS nutrient loads within Penn. Proc., Watershed management--moving from theory to implementation, Denver, CO, 841.

Gray, J.P. and M. Peel, 1994. Effective integration of GIS and modeling software for the Cheyenne, Wyoming. *Proc. 67th Annual Conference of Water Environ. Fed.,* Chicago, IL, 3, 305.

Gupta, S.K., and S.I. Solomon, 1977. Distributed numerical model for estimating runoff and sediment of ungauged rivers. *Water Resources Research,* 13(3):613-636.

He, Chansheng, J.F. Riggs, and Y.T. Kang, 1993. Integration of geographic information systems and a computer model to evaluate impacts of agricultural runoff on water quality. *Water Resources Bulletin* 29(6):891-900.

Hegwald, D.R., K. Fu, D.J. Zabolio, J.W. Bergin, and R.H. Chao, 1998. I/I evaluation and rehabilitation program simplified with cost-effective GIS implementation. *Proc, Adv. Urban Wet Weather Pollut. Reduc.*, Cleveland, OH, 195.

Henning, R.A. and S.L. Sauter, Feb. 1996. Work-physiological synchronization as a determinant of performance in repetitive computer work. *Journal of Biological Psycho-logy*, 42(3):269-286.

Hourani, N.M., 1998. Watershed modeling using remote sensing. Water Resour. Urban Environment. *Proc, 25th Water Resour. Planning Management Conf.*, Chicago, IL, ASCE, 757.

Huang, G.H., L. Liu, A. Chakma, and S.M. Wu, 1999. A hybrid GIS-supported watershed modeling system: Application to the Lake Erhai Basin, China. *Journal of Hydrological Sciences*, 44 (4), August 1999, 597-609.

Huber, W.C. and R.E. Dickinson, 1988. Stormwater water management model. *User's Manual, Ver. 4.* Environmental Research Laboratory, U.S. EPA, Athens, GA.

Hussain, M., and F.W. Schwartz, 1998. Extracting watershed characteristics from digital data using GIS: A case study of the Great Miami River basin. *Computing in Civil Engineering,* 54. 8. 515-521

Jenson, S.K. and J.O. Domingue, 1988. Extracting topographic structure from digital elevation data for geographic information system analysis. *Photogrammetric Engineering and Remote Sensing*, 54(11):1593-1600.

Jenson, S.K., 1991. Applications of hydrologic information automatically extracted from digital elevation models. *Hydrologic Processes*, 5(1).

Julien P.Y., S. Bahram, and F.L. Ogden, 1995. Raster-based hydrologic modelling of spatially-varied surface runoff. *Water Resources Bulletin* 29(6):5231-5536.

Kull, D.W., and A.D. Feldman, 1998. Evaluation of Clarks unit hydrographs method to spatially-distributed runoff. *Journal of Hydrologic Eng.*, 3(1), 9.

Lieberman, J., R. Gaulin, and J. Fulton, 1998. Application of GIS to watershed management: Stormwater pollution prevention and prediction of stormwater runoff. Proc., Watershed management--moving from theory to implementation, Denver, CO, 213.

Liong, S.Y., W.T. Chan, and L.H. Lum, 1991. Knowledge-based system for SWMM's runoff component calibration. *Journal of Water Resources Planning and Management*, 11(5):507-523.

Maidment, D.R., 1993. Chapter 13. GIS and hydrologic modelling. *Geographic Information Systems and Environmental Modeling.* M.F. Goodrich, B.O. Parks, and L.F Steeyaert, Eds. Oxford University Press, New York, pp. 147-167.

Majure, J.J., and D.A. Eash, 1994. An automated method to quantify physical basin characteristics. U.S. Geological Survey Toxic Substances Hydrology Program. G.E. Mallard and D.A. Aronson, Eds. Proceedings of the Technical Meeting, Monterey, California, March 11-15, 1991. *U.S. Geological Suvey Water-Resources Investigation Report* 91-4034, pp. 558-561.

Mark, D.M., 1983. Automated detection of drainage networks from digital elevation models. *Proc. Auto-Carto Automated Cartography: International Perspectives on Achievement and Challenges*, Ottawa, Ontario, pp. 288-298.

Martin, L., J. Castillo, and D. Ackerman, 1998. Technology in the sewer business. *Proc., Adv. Urban Wet Weather Pollut. Reduc.*, Cleveland, OH, 187.

McKibben, J.W. 1994. Wastewater flows model calibration with GIS for the City of Los Angeles. *Proc., 67th Annual Conference of Water Environ. Fed.* Chicago, IL, Vol. 3, p. 315.

Mercado, R.M. 1996. Geographic information systems (GIS) and image processing for stormwater management modeling using XP-SWMM in Tallahassee, FL.. *Proc., AWRA*, Sept. 1996, 305-313.

Milles, S. W., J.P. Heaney, W.C. Huber, and T.C. Potter, 1986. Application of Lotus 1-2-3 to SWMM preprocessor. *Proc. ASCE 4th Conference on Computing in Civil Engineering*, New York.

Miller D.J., and J. Sias, 1998. Deciphering large landslides: Linking hydrological groundwater and slope stability models through GIS. *Hydrological Processes*, 12, 93.

Mizgalewicz, P.J., W.S. White, D.R. Maidment, and K.M. Ridd, 1996. GIS modeling and visulization of the water balance during the 1993 midwest floods. *Proc., AWRA Annual Symposium. GIS and Water Resources.* Fort Lauderdale, FL, September 22-26, 1996, pp. 437-456.

Moeller, R., M. Watson, K. Mudd, and J. Sapia, 1998. A desktop GIS/Hydraulic modeling application for SSO reduction. *Proc., Adv. Urban Wet Weather Pollut. Reduc.*, Cleve-land, OH, 221.

Navulur, K. and B. Engel, 1996. Predicting spatial distribution of vulnerability of indiana state aquifer systems to nitrate leaching using a GIS. In *Proc., Third Intl Conference/ Workshop on Integrating GIS and Environmental Modeling.* Santa Fe, NM, January 21-26, 1996. California National Center for Geographic Information and Analysis, Santa Barbara, CA..

Nelson, E.J., L.J. Jones, and R.J. Berrett, 1999. Adaptive tessellation method for creating TINS from GIS data. *Journal of Hydrologic Engineering*, pp. 2-9

O'Callaghan, J.E., and D.M. Mark, 1984. The extraction of drainage networks from digital elevation data. *Computer Vision, Graphics and Image Processing*, 28:323-344.

Pansuka, J.C., I.D. Moore, and L.A. Kramer, 1990. Terrain analysis integration into the agricultural nonpoint pollution (AGNPS) model. *Journal of Soil and Water Conservation* 34:230-241.

124

Robbins, C. and S.P. Phipps, 1996. GIS/Water resources tools for performing floodplain management modeling analysis. *Proceedings AWRA Annual Symposium. GIS and Water Resources*. Fort Lauderdale, FL, Sept. 22-26, 1996, pp. 275-284.

Roesner, L.A., R.P. Shubinski, and J.A. Aldrich, 1981. *Storm water management model user's manual, version 3: Addendum I, EXTRAN*. EPA-600/2-84-109b (NTIS PB84-198431), Environmental Protection Agency, Cincinnati, OH, Nov. 1981.

Saskowsky, K.C. and T.W. Gardner, 1991. Watershed configuration and geographical information systems parametrization for SPUR model hydrologic simulation. *Water Resources Bulletin* 27(1):7-18.

Shamsi, U.M., 1998. ArcView applications in SWMM modeling. Chapter 11, *Advances in modeling: The management of stormwater impacts*. William James, Ed. CHI Publica-tions, Guelph, Ontario.

Shamsi, U.M., 1997. SWMM Graphics. Chapter 7, *Advances in modeling: The management of stormwater Impacts*. William James, Ed., CHI Publications, Guelph, Ontario, pp. 129-*153*.

Shamsi, U.M., 1996. Storm-water management implementation through modeling and GIS. *Journal of Water Resources Planing and management*, 122(2):114-127.

Small, A.B., R. Cosby, and D. Jenkins, 1998. Development of a comprehensive stormwater improvement plan for the Norfolk Naval Base. Proc., Watershed management--moving from theory to implementation, Denver, CO, 1303.

Steube, M.M., and D.M. Johnston, 1990. Runoff volume estimation using GIS techniques. *Water Resources Bulletin*, 26(4):611-620.

Srinivasan, R., J. Arnold, B.A. Engel, 1993. A continuous time, grid cell watershed model. In *Application of advanced information technologies for management of natural resources*. American Society of Agricultural Engineers, Spokane, WA, June 17-19, 1993.

Terstriep, M.L and M.T. Lee, 1989. Regional stormwater modeling: Q-ILLUDAS and ARC/ INFO. *Proc., ASCE 6th National Conference on Computing in Civil Engineering. ASCE*, New York, pp. 338-345.

Theobald, D.M., and M.F. Goodchild, 1990. Artefacts of TIN-based surface flow modeling. *GIS/LIS* 90:955-967.

Trybus, T.W., 1994. Using desk top mapping with microsoft windows, GIS without floating a bond issue. *Proc., 67th Annual Conf. of Water Environ. Fed.*, Chicago, IL., 3, p. 297.

Van Gelder, P. and M. Miller, 1996. GIS as an aid in the evaluation of drainage facilities at the Albany County Airport, New York. *GIS and Water Resources, Symposium Proceedings, American Water Resources Association*, Sept. 1996, pp. 295-301.

Vasarhelyi, M. and T. Gabor, 1996. An integrated approach and GIS applications in drainage master planning. *American Water Resources Association, Symposium Proceedings, GIS and Water Resources*, Sept. 22-26, 1996, Fort Lauderdale, FL, pp. 251-258.

Wang, M., and A.T. Hjelmfelt, 1998. DEM-based overland flow routing mode. *Journal of Hydrologic Engineering.* 3(1), 1-19.

Warwick, J.J., S.J. Hanes, 1994. Efficacy of ARC/INFO GIS application to hydrologic modeling. *Journal of water Resources Planning and management,* 120(3):366-381.

White, D., 1985. Grid-based application of runoff curve numbers, *J. Water Resources, Planning and Management Div., ASCE,* 114:601-612.

Xue, R.Z., T.J. Bechtel, and Z. Chen, 1996. Developing a user-friendly tool for BMP assessment model using a GIS. *GIS and Water Resources, Symposium Proceedings, American Water Resources Association,* Sept., 1996, pp. 285-294.

XP-SWMM Software. 1998. *User's Manual.* .XP- Software Inc., USA Office. 4747 West waters Ave # 33614, Tampa, FL., USA. Phone: (813) 886-72, FAX: (813) 885-4198.

APPENDIX A: SOURCE CODE LISTINGS
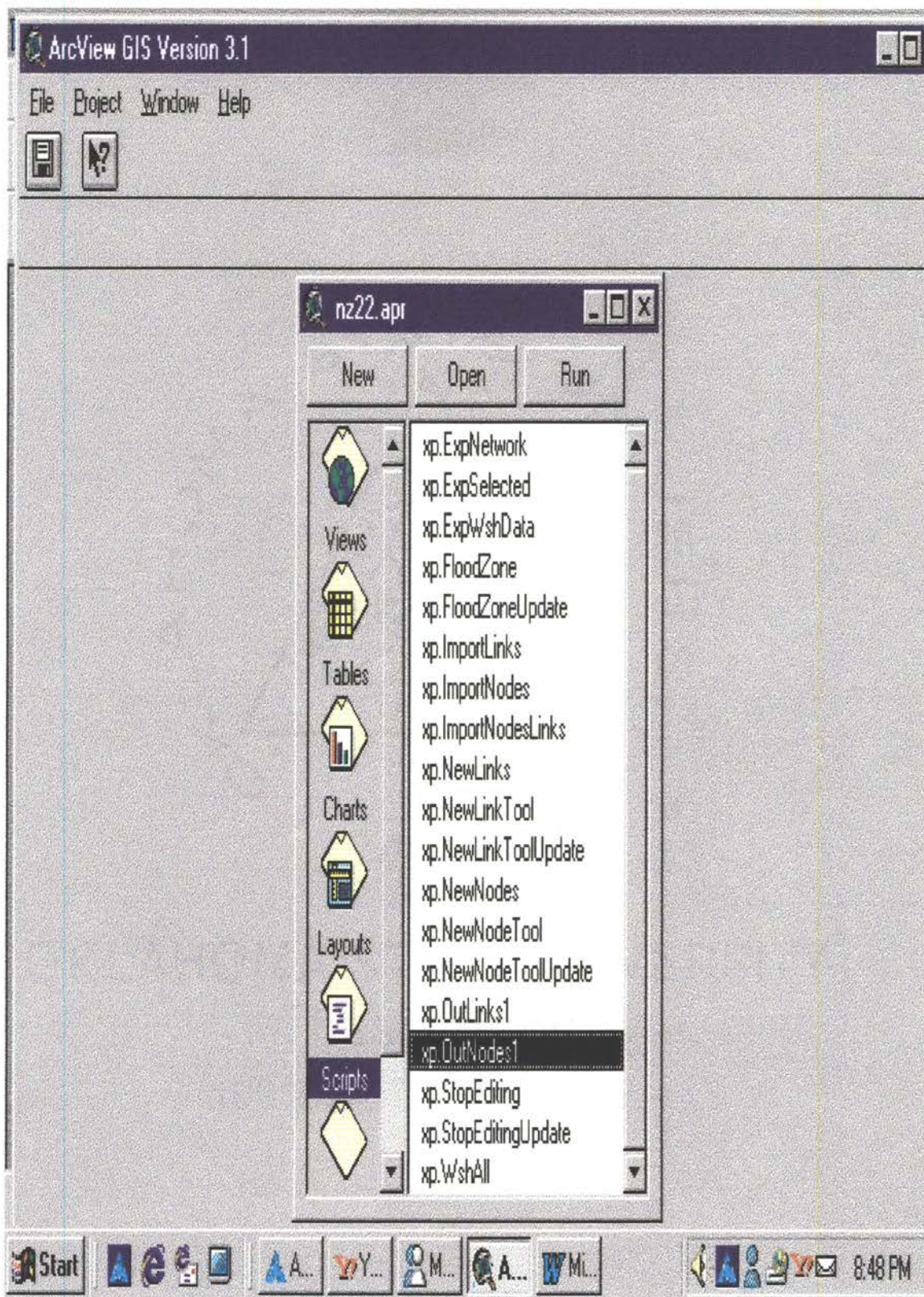
Figure 1A. (Appendix-A). List of Avenue Codes for Three ArcView's
Extensions: WATERSHED, NETWORK, and PRESENTATION

APPENDIX A

```
1 A.    'XP.Network
theView = av.GetActiveDoc
allthemes = theView.getThemes
pointThemeList = list.Make
nodeFields = list.make
lineThemeList = list.Make
linkFields = list.Make

for each t in allthemes
  if (t.Is(GTHEME)) then
    continue
  end
  xx = t.getftab.findfield("shape")
  if (xx.gettype = #FIELD_SHAPELINE) then
    lineThemeList.add(t)
    continue
  end
  if (xx.gettype = #FIELD_SHAPEPOINT) then
    pointThemeList.add(t)
    continue
  end
' select a point theme
if(pointThemeList.count = 0) then
  msgbox.error(theView.asstring++"does not contain any point theme","")
  return nil
end
ntheme = msgbox.list(pointThemeList,
      "Select a theme of nodes ", "Network nodes")
if(ntheme = nil) then
  return nil
end
NodeFtab = ntheme.getFtab
for each pf in NodeFtab.GetFields
  nodefields.add(pf)
end
' find fields:
nnamefd = NodeFtab.FindField("node_name")
if ( nnamefd = nil) then
  nnamefd = msgbox.list(nodefields, "Select the node_name field",
    "Name of the node")
  if( nnamefd = nil ) then
    return nil
  end
end
ntypefd = NodeFtab.FindField("node_type")
if ( ntypefd = nil) then
ntypefd = msgbox.list(nodefields, "Select the node_type field",
    "Type of the node")
  if( ntypefd = nil ) then
    return nil
  end
end
nxfd = NodeFtab.FindField("node_x")
if ( nxfd = nil) then
  nxfd = msgbox.list(nodefields, "Select the node_x field",
    "X coordinate of the node")
  if( nxfd = nil ) then
    return nil
  end
end
nyfd = NodeFtab.FindField("node_y")
if ( nyfd = nil) then
  nyfd = msgbox.list(nodefields, "Select the node_y field",
    "Y coordinate of the node")
  if( nyfd = nil ) then
    return nil
  end
end

' select a line theme
if(lineThemeList.count = 0) then
  msgbox.error(theView.asstring++"does not contain any line theme","")
  return nil
end
ltheme = msgbox.list(lineThemeList,
      "Select a theme of links ", "Network links")
if(ltheme = nil) then
  return nil
end
linkFtab = ltheme.getFtab
for each lf in LinkFtab.GetFields
  Linkfields.add(lf)
end
' find fields:
lnamefd = LinkFtab.FindField("link_name")
if ( lnamefd = nil) then
  lnamefd = msgbox.list(linkfields, "Select the link_name field",
    "Name of the link")
  if( lnamefd = nil ) then
    return nil
  end
end
ltypefd = LinkFtab.FindField("link_type")
if ( ltypefd = nil) then

  ltypefd = msgbox.list(linkfields, "Select the link_type field",
    "Type of the link")
  if( ltypefd = nil ) then
```

```
      return nil
    end
  end
lfromfd = LinkFtab.FindField("node_from")
if ( lfromfd = nil) then
    lfromfd = msgbox.list(linkfields, "Select the node_from field",
    "Beginning node of the link")
    if( lfromfd = nil ) then
      return nil
    end
  end
ltofd = LinkFtab.FindField("node_to")
if ( ltofd = nil) then
    ltofd = msgbox.list(linkfields, "Select the node_to field",
    "End node of the link")
    if( ltofd = nil ) then
      return nil
    end
  end


'everythig is ready
aFN = FileDialog.put("net.xpx".asFileName,"*.xpx", "Network Export")
if(aFN = nil) then
    return nil
end
alinefile = LineFile.Make(aFN, #FILE_PERM_WRITE )
for each rec in NodeFtab
    xn1 = NodeFtab.ReturnValueNumber(ntypefd,rec).asstring
    xn2 = NodeFTab.ReturnValue(nnamefd,rec)
    xn3 = NodeFTab.ReturnValue(nxfd,rec).asstring
    xn4 = NodeFTab.ReturnValue(nyfd,rec).asstring
    xline = "NODE"++xn1++""""+xn2+""""++xn3++xn4
    alinefile.WriteElt(xline)
end
for each rec in LinkFtab
    xl1 = LinkFtab.ReturnValueNumber(ltypefd,rec).asstring
    xl2 = LinkFtab.ReturnValue(lnamefd,rec)
    xl3 = LinkFTab.ReturnValue(lfromfd,rec)
    xl4 = LinkFTab.ReturnValue(ltofd,rec)
    xline = "LINK"++xl1++""""+xl2+""""++""""+xl3+""""++""""+xl4+""""
    alinefile.WriteElt(xline)
end
alineFile.close

2 A. XP.Floodzone

av.UseWaitCursor
theView = GetActiveDoc
theDisplay = theView.GetDisplay
theGrid = theView.GetActiveThemes.Get(0).getGrid
```

```
p = theDisplay.ReturnUserPoint
aPrj =- Prj.MakeNull
elev = theGrid.cellvalue(p, aPrj)
msgbox.INfo(elev.asstring, "elevation")

zones1 = (theGrid <= elev.asGrid).con().asGrid, 1.asgrid)
zones1b = zones1.setNull(1.asgrid)
zones2 =zones1b.RegionGroup(TRUE, FALSE), -1)

zone_id = zones2.cellvalue(p, aPrj)
'here the only zone that contains the point is selected
' ...(zones2 = zone id.asgrid) .con(...), and ...zone3.setNull(...)
aFN = av.GetProject.GEtWorkDir.MakeTmp("zone","shp")
theZone = zone2.AsPolygonFTab(aFN,TRUE,aPrj)
theFTheme = FTheme.Make(theZone)
theFtheme.SetName("Elev < "++elev.asstring)
theView.AddTheme(theFTheme)


3 A.   'XP.Selected

theView = av.GetActiveDoc
sp = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
ln1 = "nodes . . . . . . . . . . . . . . . . . . ."
ln2 = "links . . . . . . . . . . . . . . . . . . ."
ln2a = "Catchment Parameters              ."
ln3 = "       Area. . . . . . . . . . . . . . . ."
ln4 = "       Width/Length . . . . . . . . . . . "
ln5 = "       Average Slope . . . . . . . . . . . "
ln6 = "       Percent of Imprevious . . . . . ."

labls = {ln1,ln2,ln2a,ln3,ln4,ln5,ln6}
defts = {"1","1"," ","1","1","1","1"}
tytul = "     Select data for export  ( 1 = YES, 0 = NO )"
xlist = msgBox.Multiinput(tytul, "Exporting Data", labls, defts )
if (xlist.count = 0) then
    return nil
end
if ( xlist.get(0) <> "0") then
    no = TRUE
else
    no = FALSE
end
if ( xlist.get(1) <> "0") then
    li = TRUE
else
    li = FALSE
end
```

```
if ( xlist.get(3) <> "0") then
  wa = TRUE
else
  wa = FALSE
end
if ( xlist.get(4) <> "0") then
  ww = TRUE
else
  ww = FALSE
end
if ( xlist.get(5) <> "0") then
  ws = TRUE
else
  ws = FALSE
end
if ( xlist.get(6) <> "0") then
  wi = TRUE
else
  wi = FALSE
end
allthemes = theView.getThemes
pointThemeList = list.Make
nodeFields = list.make

lineThemeList = list.Make
linkFields = list.Make

for each t in allthemes
  if (t.Is(GTHEME)) then
    continue
  end
  xx = t.getftab.findfield("shape")
  if (xx.gettype = #FIELD_SHAPELINE) then
    lineThemeList.add(t)
    continue
  end
  if (xx.gettype = #FIELD_SHAPEPOINT) then
    pointThemeList.add(t)
    continue
  end
end

if (no or wa or ww or ws or wi) then
  ' select a point theme
  if(pointThemeList.count = 0) then
    msgbox.error(theView.asstring++"does not contain any point
theme","")
    return nil
```

```
end
ntheme = msgbox.list(pointThemeList,
    "Select a theme of nodes ", "Network nodes")
if(ntheme = nil) then
  return nil
end
NodeFtab = ntheme.getFtab
for each pf in NodeFtab.GetFields
    nodefields.add(pf)
end

' find fields:
nnamefd = NodeFtab.FindField("node_name")
if ( nnamefd = nil) then
  nnamefd = msgbox.list(nodefields, "Select the node_name field",
"Name of the node")
    if( nnamefd = nil ) then
      return nil
    end
end
if(no) then
    ntypefd = NodeFtab.FindField("node_type")
    if ( ntypefd = nil) then
      ntypefd = msgbox.list(nodefields, "Select the node_type
field",
        "Type of the node")
      if( ntypefd = nil ) then
        return nil
end
    end
    nxfd = NodeFtab.FindField("node_x")
    if ( nxfd = nil) then
      nxfd = msgbox.list(nodefields, "Select the node_x field",
       "X coordinate of the node")
      if( nxfd = nil ) then
        return nil
      end
    end
    nyfd = NodeFtab.FindField("node_y")
    if ( nyfd = nil) then
      nyfd = msgbox.list(nodefields, "Select the node_y field",
       "Y coordinate of the node")
      if( nyfd = nil ) then
        return nil
      end
    end
end
if(wa) then
```

```
    wareafd = NodeFtab.FindField("cat_area")
    if ( wareafd = nil) then
       wareafd = msgbox.list(nodefields, "Select the cat_area
field",
       "Catchment area")
       if( wareafd = nil ) then
          return nil
       end
  if(ww) then
       widthfd = NodeFtab.FindField("wsh_width")
       if ( widthfd = nil) then
          widthfd = msgbox.list(nodefields, "Select the wsh_width
field",
          "Watershed width (length)")
          if( widthfd = nil ) then
             return nil
          end
       end
    end
    if(ws) then
       wslopefd = NodeFtab.FindField("wsh_slope")
       if ( wslopefd = nil) then
          wslopefd = msgbox.list(nodefields, "Select the wsh_slope
field",
          "Watershed slope")
          if( wslopefd = nil ) then
             return nil
          end
       end
    end
    if(wi) then
       wimpfd = NodeFtab.FindField("wsh_imp")
    if ( wimpfd = nil) then
          wimpfd = msgbox.list(nodefields, "Select the wsh_imp field",
          "Percentage of watershed imprevious")
          if( wimpfd = nil ) then
             return nil
          end
       end
    end
end

if(li) then
  ' select a line theme
  if(lineThemeList.count = 0) then
    msgbox.error(theView.asstring++"does not contain any line
theme","")
       return nil
```

```
end
ltheme = msgbox.list(lineThemeList,
        "Select a theme of links ", "Network links")
if(ltheme = nil) then
   return nil
end
linkFtab = ltheme.getFtab
for each lf in LinkFtab.GetFields
    Linkfields.add(lf)
end
' find fields:
lnamefd = LinkFtab.FindField("link_name")
if ( lnamefd = nil) then
   lnamefd = msgbox.list(linkfields, "Select the link_name
field",
      "Name of the link")
   if( lnamefd = nil ) then
      return nil
   end
   ltypefd = LinkFtab.FindField("link_type")
   if ( ltypefd = nil) then
      ltypefd = msgbox.list(linkfields, "Select the link_type
field",
      "Type of the link")
      if( ltypefd = nil ) then
         return nil
      end
 lfromfd = LinkFtab.FindField("node_from")
   if ( lfromfd = nil) then
      lfromfd = msgbox.list(linkfields, "Select the node_from
field",
      "Beginning node of the link")
      if( lfromfd = nil ) then
         return nil
      end
   end

ltofd = LinkFtab.FindField("node_to")
   if ( ltofd = nil) then
      ltofd = msgbox.list(linkfields, "Select the node_to field",
      "End node of the link")
      if( ltofd = nil ) then
         return nil
      end
   end
```

```
end

'everythig is ready
aFN = FileDialog.put("datt.xpx".asFileName,"*.xpx", "XP Data
Export")
if(aFN = nil) then
   return nil
end
alinefile = LineFile.Make(aFN, #FILE_PERM_WRITE )
if(no) then
   for each rec in NodeFtab
      xn2 = NodeFTab.ReturnValue(nnamefd,rec)
      xn1 = NodeFTab.ReturnValueNumber(ntypefd,rec).asstring
      xn3 = NodeFTab.ReturnValue(nxfd,rec).asstring
      xn4 = NodeFTab.ReturnValue(nyfd,rec).asstring
      xline = "NODE"++xn1++""""+xn2+""""++xn3++xn4
      alinefile.WriteElt(xline)
   end
end
if (li) then
   for each rec in LinkFtab
      xl1 = LinkFtab.ReturnValueNumber(ltypefd,rec).asstring
      xl2 = LinkFTab.ReturnValue(lnamefd,rec)
      xl3 = LinkFTab.ReturnValue(lfromfd,rec)
      xl4 = LinkFTab.ReturnValue(ltofd,rec)
      xline = "LINK"++xl1++""""+xl2+""""++""""+xl3+""""++""""+xl4+""""
      alinefile.WriteElt(xline)
   end
end
if (wa or ww or ws or wi) then
xstrange = """"+""""++""""+""""++""""+""""++""""+""""
   for each rec in NodeFtab
      xw2 = NodeFTab.ReturnValue(nnamefd,rec)
      if (wa) then
         xw3 = NodeFTab.ReturnValue(wareafd,rec).asstring
         xline = "DATA R_WAREA"++""""+xw2+""""++"0 5"++xw3++xstrange
         alinefile.WriteElt(xline)
      end

   if (wi) then
         xw6 = NodeFTab.ReturnValue(wimpfd,rec).asstring
         xline = "DATA R_WIMP"++""""+xw2+""""++"0 5"++xw6++xstrange
         alinefile.WriteElt(xline)
      end
   if (ww) then
         xw4 = NodeFTab.ReturnValue(widthfd,rec).asstring
         xline = "DATA R_WIDTH"++""""+xw2+""""++"0 5"++xw4++xstrange
         alinefile.WriteElt(xline)
```

```
   end
   if (ws) then
      xw5 = NodeFTab.ReturnValue(wslopefd,rec).asstring
      xline = "DATA R_WSLOPE"++""""+xw2+""""++"0 5"++xw5++xstrange
      alinefile.WriteElt(xline)
   end
   end
end
alineFile.close


4 A.  'XP.FloodZoneUpdate

theView = av.GetActiveDoc
enabled = FALSE

' enable if 1) one, and only one,
'            2) Theme is active, and
'            3) the Theme is a GTheme
for each atheme in theView.GetActiveThemes
   if (atheme.Is(GTHEME)) then
      enabled = TRUE
   end
end
if (theView.GetActiveThemes.Count > 1) then
   enabled = FALSE
end
SELF.SetEnabled(enabled)

5 A   'XP.ImportLinks

'Select the file to read data *.xpx
xpxFN = FileDialog.Show("*.xpx","XP eXchange Files",
            "Create a map of links from xpx export file")

if ( xpxFN = nil ) then
   return nil

end

xpxFile = LineFile.Make(xpxFN, #FILE_PERM_READ)
xpxLines = xpxFile.GetSize

tempName    = xpxFN.GetBaseName.AsTokens(".").Get(0)
tempDir     = xpxFN.ReturnDir.AsString
defLinkName = FileName.Make( tempDir ).MakeTmp("links", "shp" )
shpLinkName = FileDialog.Put(defLinkName,"*.shp","Create Shape
```

```
                File of Links")

                if ( shpLinkName = nil ) then
                    return nil

                end

                shpLinkFTab = FTab.MakeNew(shpLinkName, Polyline)

                fieldsLink = List.Make
                fieldsLink.add(Field.Make("Link_id", #FIELD_SHORT, 4, 0 ))
                fieldsLink.add(Field.Make("link_name", #FIELD_VCHAR, 12, 0 ))
                fieldsLink.add(Field.Make("link_type", #FIELD_SHORT, 4, 0 ))
                fieldsLink.add(Field.Make("node_from", #FIELD_VCHAR, 12, 0 ))
                fieldsLink.add(Field.Make("node_to", #FIELD_VCHAR, 12, 0 ))
                shpLinkFTab.AddFields(fieldsLink)

                lshpfd = shpLinkFtab.FindField("Shape")
                lidfd = shpLinkFtab.FindField("Link_id")
                lnamefd = shpLinkFtab.FindField("link_name")
                ltypefd = shpLinkFtab.FindField("link_type")
                lfromfd = shpLinkFtab.FindField("node_from")
                ltofd = shpLinkFtab.FindField("node_to")

                buf = xpxFile.ReadElt

                if (xpxFile.IsAtEnd ) then
                    MsgBox.Error( " File" + xpxFile.asstring ++"is empty.","")
                    return nil
                end
                if (buf = nil ) then

                MsgBox.Error( " Can not open " + xpxFile.asstring ,"")
                    return nil

                end

                av.ShowMsg ( "Reading Nodes and Links")
                av.ShowStopButton
                recNum = 1
                linecount = 1
                node_no = 0
                dn_xy = dictionary.make(50)
                link_no = 0
                while ( TRUE)

                if (xpxFile.IsAtEnd ) then
                    break
```

```
            end

            if ( buf.extract(0).ucase = "NODE" ) then
                node_no = node_no + 1
                'extract node name and remove it from the buffer
                'if node name is NIL ("") then insert a dummy string
                buf = buf.substitute("""""", """dummystring156""")
                node_name = buf.AsTokens("""").get(1).asstring.trim
                buf2 = buf.substitute("""""+node_name+""""", "")

                if( node_name = "dummystring156") then
                    node_name = NIL

            end

                n_type = buf2.extract(1).AsNumber
                n_x = buf2.extract(2).AsNumber
                n_y = buf2.extract(3).AsNumber
                thenode = n_x@n_y
                dn_xy.add(node_name, thenode)

            end

            if ( buf.extract(0).ucase = "LINK" ) then

            link_no = link_no + 1
                'extract link name and node names.

            buf = buf.substitute("""""", """dummystring156""")
                link_name = buf.AsTokens("""").get(1).asstring.trim
                nfrom = buf.AsTokens("""").get(3).asstring.trim
                nto = buf.AsTokens("""").get(5).asstring.trim
                l_type = buf.extract(1).AsNumber
                if( link_name = "dummystring156") then
                    link_name = NIL

            end

            x1 =  dn_xy.get(nfrom)
                if (x1 = nil) then

            msgbox.error( " link: F-node "++nfrom ++ "not defined !", "")
                    return nil
                end
                x2 =  dn_xy.get(nto)

            if (x2 = nil) then
                    msgbox.error( " link: T-node "++nto ++ "not defined !", "")
```

```
        return nil

    end


    thePolyline = polyline.Make({{x1, x2}})

    lrec = shpLinkFTab.AddRecord

        shpLinkFTab.SetValue(lshpFd, lrec, thePolyline)
        shpLinkFTab.SetValueNumber(lidFd, lrec, link_no)
        shpLinkFTab.SetValue(lnameFd, lrec, link_name)
        shpLinkFTab.SetValueNumber(ltypeFd, lrec, l_type)
        shpLinkFTab.SetValue(lfromFd, lrec, nfrom)
        shpLinkFTab.SetValue(ltoFd, lrec, nto)

    end

lsview.add("<new view>")

buf = xpxFile.ReadElt

end

if (MsgBox.YesNo("Display the shapefile as a theme on a view?",
    "XPX IMPORT",true)) then
    lsview = list.make
    for each d in av.GetProject.GetDocs

if (d.is(view) and d.getwin.isopen )then
        lsview.add(d)

end

end

ViewChoice = MsgBox.ChoiceAsString( lsview, "Choose the view:",
"Add node-shape to the view" )


if (ViewChoice = nil) then
    return nil

end

if(ViewChoice = "<new view>") then
    newview = view.make
    newView.GetWin.Open
    newView.AddTheme(FTheme.Make(shpLinkFTab))
```

```
    'bring the View to the front
    newView.GetWin.Activate
    else
    ViewChoice.AddTheme(FTheme.Make(shpLinkFTab))
    end
        end

6 A.    'XP.WshData

theView = av.GetActiveDoc
allthemes = theView.getThemes
pointThemeList = list.Make
nodeFields = list.make

for each t in allthemes
    if (t.Is(GTHEME)) then
        continue
    end
    xx = t.getftab.findfield("shape")
    if (xx.gettype = #FIELD_SHAPEPOINT) then
        pointThemeList.add(t)
        continue
    end
end


' select a point theme
if(pointThemeList.count = 0) then
    msgbox.error(theView.asstring++"does not contain any point
theme","")
    return nil
end
ntheme = msgbox.list(pointThemeList,
        "Select a theme of nodes ", "Network nodes")
if(ntheme = nil) then
    return nil
end
NodeFtab = ntheme.getFtab
for each pf in NodeFtab.GetFields
    nodefields.add(pf)
end

' find fields:
nnamefd = NodeFtab.FindField("node_name")
if ( nnamefd = nil) then
    nnamefd = msgbox.list(nodefields, "Select the node_name field",
    "Name of the node")
```

```
if( nnamefd = nil ) then
  return nil
end
end

wareafd = NodeFtab.FindField("cat_area")
if ( wareafd = nil) then
  wareafd = msgbox.list(nodefields, "Select the cat_area field",
  "Catchment area")
  if( wareafd = nil ) then
    return nil
  end
end
widthfd = NodeFtab.FindField("wsh_width")
if ( widthfd = nil) then
  widthfd = msgbox.list(nodefields, "Select the wsh_width field",
  "Watershed width (length)")
  if( widthfd = nil ) then
    return nil
  end
end
wslopefd = NodeFtab.FindField("wsh_slope")
if ( wslopefd = nil) then
  wslopefd = msgbox.list(nodefields, "Select the wsh_slope field",
  "Watershed slope")
  if( wslopefd = nil ) then
    return nil
  end
end
wimpfd = NodeFtab.FindField("wsh_imp")
if ( wimpfd = nil) then
  wimpfd = msgbox.list(nodefields, "Select the wsh_imp field",
  "Percentage of watershed imprevious")
  if( wimpfd = nil ) then
    return nil
  end
end

'everythig is ready
aFN = FileDialog.put("catdat.xpx".asFileName,"*.xpx", "Catchment
Parameters Export")
if(aFN = nil) then
  return nil
end
alinefile = LineFile.Make(aFN, #FILE_PERM_WRITE )
```

```
xstrange = """"+""""++""""+""""++""""+""""++""""+""""
for each rec in NodeFtab
  xw2 = NodeFTab.ReturnValue(nnamefd,rec)
  xw3 = NodeFTab.ReturnValue(wareafd,rec).asstring
  xw4 = NodeFTab.ReturnValue(widthfd,rec).asstring
  xw5 = NodeFTab.ReturnValue(wslopefd,rec).asstring
  xw6 = NodeFTab.ReturnValue(wimpfd,rec).asstring
  xline = "DATA R_WAREA"++""""+xw2+""""++"0 5"++xw3++xstrange
  alinefile.WriteElt(xline)
  xline = "DATA R_WIMP"++""""+xw2+""""++"0 5"++xw6++xstrange
  alinefile.WriteElt(xline)
  xline = "DATA R_WIDTH"++""""+xw2+""""++"0 5"++xw4++xstrange
  alinefile.WriteElt(xline)
  xline = "DATA R_WSLOPE"++""""+xw2+""""++"0 5"++xw5++xstrange
  alinefile.WriteElt(xline)
end
alineFile.close

7 A.   'XP.ImportNodes

'Select the file to read data *.xpx
xpxFN = FileDialog.Show("*.xpx","XP eXchange Files",
        "Create a map of nodes from xpx export file")
if ( xpxFN = nil ) then
  return nil
end
xpxFile = LineFile.Make(xpxFN, #FILE_PERM_READ)
xpxLines = xpxFile.GetSize
tempName     = xpxFN.GetBaseName.AsTokens(".").Get(0)
tempDir      = xpxFN.ReturnDir.AsString
defNodeName  = FileName.Make( tempDir ).MakeTmp("nodes", "shp" )
  Specify the output shapefiles:
shpNodeName = FileDialog.Put(defNodeName,"*.shp","Create Shape File
of Nodes")
if ( shpNodeName = nil ) then
  return nil
end
shpNodeFTab = FTab.MakeNew(shpNodeName, Point)
fieldsNode = List.Make
fieldsNode.add(Field.Make("Node_id", #FIELD_SHORT, 4, 0 ))
fieldsNode.add(Field.Make("node_name", #FIELD_VCHAR, 12, 0 ))
fieldsNode.add(Field.Make("node_type", #FIELD_SHORT, 4, 0 ))
fieldsNode.add(Field.Make("node_x", #FIELD_DOUBLE, 20, 8 ))
fieldsNode.add(Field.Make("node_y", #FIELD_DOUBLE, 20, 8 ))
shpNodeFTab.AddFields(fieldsNode)
nshpfd = shpNodeFtab.FindField("Shape")
```

```
nidfd = shpNodeFtab.FindField("Node_id")
nnamefd = shpNodeFtab.FindField("node_name")
ntypefd = shpNodeFtab.FindField("node_type")
nxfd = shpNodeFtab.FindField("node_x")
nyfd = shpNodeFtab.FindField("node_y")
buf = xpxFile.ReadElt
if (xpxFile.IsAtEnd ) then
  MsgBox.Error( " File" + xpxFile.asstring ++"is empty.","")
  return nil
end
if (buf = nil ) then
  MsgBox.Error( " Can not open " + xpxFile.asstring ,"")
  return nil
end
av.ShowMsg ( "Reading Nodes")
av.ShowStopButton
recNum = 1
linecount = 1
node_no = 0
dn_xy = dictionary.make(50)
link_no = 0
while ( TRUE)
  if (xpxFile.IsAtEnd ) then
    break
while ( TRUE)
  if (xpxFile.IsAtEnd ) then
    break
  end
  if ( buf.extract(0).ucase = "NODE" ) then
    node_no = node_no + 1
    'extract node name and remove it from the buffer
    'if node name is NIL ("") then insert a dummy string
    buf = buf.substitute("""""", """dummystring156""")
    node_name = buf.AsTokens("""").get(1).asstring.trim
    buf2 = buf.substitute(""""+node_name+"""", "")
    if( node_name = "dummystring156") then
      node_name = NIL
    end
    n_type = buf2.extract(1).AsNumber
    n_x = buf2.extract(2).AsNumber
    n_y = buf2.extract(3).AsNumber
    thenode = n_x@n_y
    dn_xy.add(node_name, thenode)
    nrec = shpNodeFTab.AddRecord
    shpNodeFTab.SetValue(nshpFd, nrec, thenode)
```

```
      shpNodeFTab.SetValueNumber(nidFd, nrec, node_no)
      shpNodeFTab.SetValue(nnameFd, nrec, node_name)
      shpNodeFTab.SetValueNumber(ntypeFd, nrec, n_type)
      shpNodeFTab.SetValue(nxFd, nrec, n_x)
      shpNodeFTab.SetValue(nyFd, nrec, n_y)
  end
  buf = xpxFile.ReadElt
end
if (MsgBox.YesNo("Display the shapefile as themes on a view?",
   "XPX IMPORT",true)) then
  lsview = list.make
  for each d in av.GetProject.GetDocs
    if (d.is(view) and d.getwin.isopen )then
      lsview.add(d)
    end
  lsview.add("<new view>")
  ViewChoice = MsgBox.ChoiceAsString( lsview, "Choose the view:",
"Add node-shape to the view" )
  if (ViewChoice = nil) then
    return nil
  end
  if(ViewChoice = "<new view>") then
    newview = view.make
    newView.GetWin.Open
    newView.AddTheme(FTheme.Make(shpNodeFTab))
    'bring the View to the front
    newView.GetWin.Activate
  else
    ViewChoice.AddTheme(FTheme.Make(shpNodeFTab))
  end

8 A.  'XP.NewLinks

theView = av.GetActiveDoc
'If a theme in the view is being edited, Stop Editing it before
creating new theme
editThm = theView.GetEditableTheme
if (editThm <> nil) then
  doSave = MsgBox.YesNoCancel("Save edits to
"+editThm.GetName+"?","Stop Editing",true)
  if (doSave = nil) then
    return nil
  end
  if (editThm.StopEditing(doSave).Not) then
    MsgBox.Info("Unable to Save Edits to "
                  + editThm.GetName +
                  ", please use the Save Edits As option", "")
```

```
        return nil                                      fld3 = Field.Make("link_type", #FIELD_SHORT, 4, 0)
    else                                                fld4 = Field.Make("node_from", #FIELD_CHAR, 16, 0)
        theView.SetEditableTheme(NIL)                   fld5 = Field.Make("node_to", #FIELD_CHAR, 16, 0)
    end
end                                                     fld1.SetVisible( FALSE )
                                                        tbl.AddFields({fld1,fld3,fld2,fld4,fld5})
                                                        tbl.SetEditable(False)
'ask user to select the point (nodes) theme            theTheme = FTheme.Make(tbl)
nodethemelist = List.make                               theView.AddTheme(theTheme)
themelist = av.getActiveDoc.getThemes                   theTheme.SetActive(TRUE)
for each xth in themelist                               theTheme.SetVisible(TRUE)
    if (xth.is(Gtheme)) then continue end               theView.SetEditableTheme(theTheme)
    if (xth.getftab.findfield("shape").gettype = #FIELD_SHAPEPOINT)  av.GetProject.SetModified(true)
then                                                    for each x in av.GetActiveGUI.GetToolBar
        nodethemelist.add(xth)                              if (x.is(Tool)) then
    end                                                         if(x.HasScript("xp.NewLinkTool")) then
end                                                                 x.Select
                                                                    x.setObjectTag(nil)
_nodeth = msgbox.choice(nodethemelist, "Select nodes theme",         return nil
"Create Links of Nodes" )                                       end
if (_nodeth = nil) then                                     end
    exit                                                end
end
    thtype = _nodeth.getftab.FindField("Shape").GetType
        if ( thtype <> #FIELD_SHAPEPOINT) then
msgbox.error("Theme"++_nodeth.asstring++"is not a point theme", "") 9 A.   'XP.NewLinkToolUpdate
    exit
end                                                     theView = av.GetActiveDoc
'_nodeftab = nodeth.getftab                             theTheme = theView.GetEditableTheme
                                                        if (theTheme = nil) then
_namefield = _nodeth.getftab.findField("Node_name")       SELF.SetEnabled(False)
                                                        else
def = av.GetProject.MakeFileName("Links", "shp")          SELF.SetEnabled( (theTheme.GetFTab.FindField("Shape").GetType =
def = FileDialog.Put(def, "*.shp", "New Links Theme")   #FIELD_SHAPELINE)
        if (def <> nil) then                              and (theTheme.GetFTab.FindField("Link_name") <> nil )
                                                          and (theTheme.GetFTab.FindField("Link_type") <> nil )
                                                          and (theTheme.GetFTab.FindField("Node_from") <> nil )
    tbl = FTab.MakeNew(def, Polyline)                     and (theTheme.GetFTab.FindField("Node_to") <> nil ) )
    if (tbl.HasError) then                              end
        if (tbl.HasLockError) then
            MsgBox.Error("Unable to acquire Write Lock for file " +
def.GetBaseName, "")                                    10A  'XP.LinkTool
        else
            MsgBox.Error("Unable to create " + def.GetBaseName, "") theView = av.GetActiveDoc
        end                                             pt = theView.GetDisplay.ReturnUserPoint
        return nil                                      fnn0 = ""
    end                                                 nodeftab = _nodeth.getftab
    fld1 = Field.Make("ID", #FIELD_DECIMAL, 8, 0)       if (_nodeth.CanFindByPoint.not) then
    fld2 = Field.Make("link_name", #FIELD_CHAR, 16, 0)    return nil
                                                        end
```

```
keys = _nodeth.FindByPoint(pt)
if (keys.count <> 0 ) then
  fnn0 = nodeftab.returnvaluestring(_namefield, keys.get(0))
else
  return nil
end

thisTool =  av.GetActiveGUI.GetToolBar.getactive
aName = thisTool.getObjectTag

if( aName = nil) then
  answer = msgbox.yesno("Conduit begins in the node"++fnn0,
       " From Node ", True)
  if(answer) then
     thisTool.setObjectTag(fnn0)
     'thisTool.setTag(fnn0)
  end
  return nil
end

if( aName <> nil) then
  answer = msgbox.yesno("Conduit ends in the node"++fnn0,
       " To Node ", True)
  if(answer) then
     ' fnn0 = thisTool.getTag
     tnn0 = fnn0
     fnn0 = thisTool.getObjectTag
     thisTool.setObjectTag(nil)
     ' thisTool.setTag("")
  else
     return nil
  end
end

ln0 = ""
lt0 = 136.asstring
while(true)
  entered = msgbox.multiinput( "Enter link name and link type:",
  "Link from node"++fnn0++"to node"++tnn0,
  {"Link Name", "Link Type (0,136,137,138)"}, {ln0, lt0} )
if (entered.count = 0) then
thisTool.setObjectTag(nil)
    return nil
  end
  ln0 = entered.get(0)

lt0 = entered.get(1)
```

```
  if ( lt0.isnumber.not) then
msgbox.Warning (" Link Type must be a number ", "Link Type")
    continue
  end
  lt = lt0.asNumber

  found = false
  for each rec in nodeftab
    xx = nodeftab.returnvaluestring(_namefield, rec)
    if (fnn0.trim = xx ) then
         found = true
      fp = nodeftab.returnvalue(nodeftab.findfield("shape"), rec)
         break
    end
  end
         if (found.not) then
msgbox.warning ("From node:"++fnn0++"not found","Wrong node name")
    continue
  end

  found = false
  for each rec in nodeftab
    xx = nodeftab.returnvalueString(_namefield, rec)
         if (tnn0.trim = xx ) then
      found = true
 tp = nodeftab.returnvalue(nodeftab.findfield("shape"), rec)
    break
    end
  end
         if (found.not) then
msgbox.warning (" To node:"++tnn0++"not found", "Wrong node name")
    continue
  end

  if (ln0.trim = "") then
    again = msgbox.yesno("No link name entered"+NL
    +"Do you want to enter a name ?", "Link name", True)
           if (again.not) then
      thisTool.setObjectTag(nil)
           return nil
    else
continue
    end
  end
 break
end
```

```
' now create a link (polyline)
theTheme = theView.GetEditableTheme
theline = polyline.make({{fp,tp}})

    if (theTheme <> nil) then
      theTheme.GetFtab.BeginTransaction
'       thePrj = theView.GetProjection
'       if (thePrj.IsNull.Not) then
'         theline = theline.ReturnUnprojected(thePrj)
'       end


      theField = theTheme.GetFTab.FindField("Shape")
      theline = polyline.make({{fp,tp}})
      theF1 = theTheme.GetFTab.FindField("Link_type")
      theF2 = theTheme.GetFTab.FindField("Link_name")
      theF3 = theTheme.GetFTab.FindField("Node_from")
      theF4 = theTheme.GetFTab.FindField("Node_to")
      rec = theTheme.GetFTab.AddRecord
      theTheme.GetFTab.SetValue(theField, rec, theline)
      theTheme.GetFTab.SetValueString(theF2, rec, ln0)
      theTheme.GetFTab.SetValueString(theF3, rec, fnn0)
      theTheme.GetFTab.SetValueString(theF4, rec, tnn0)
      theTheme.GetFTab.SetValue(theF1, rec, lt)
'       theTheme.GetFTab.GetSelection.ClearAll
'       theTheme.GetFTab.GetSelection.Set(rec)
'       theTheme.GetFTab.UpdateSelection
      theTheme.GetFtab.EndTransaction
    else
      gl = GraphicShape.Make({fp,tp})
      theView.GetGraphics.UnselectAll
      gl.SetSelected(TRUE)
      theView.GetGraphics.Add(gl)
    end
  av.GetProject.SetModified(true)
'end


11 A.  'XP.NewNodes

theView = av.GetActiveDoc
'If a theme in the view is being edited, Stop Editing it before
creating new theme
editThm = theView.GetEditableTheme
if (editThm <> nil) then
  doSave = MsgBox.YesNoCancel("Save edits to

"+editThm.GetName+"?","Stop Editing",true)
  if (doSave = nil) then
```

```
    return nil
  end
  if (editThm.StopEditing(doSave).Not) then
    MsgBox.Info("Unable to Save Edits to "
                    + editThm.GetName +
                    ", please use the Save Edits As option", "")
    return nil
  else
    theView.SetEditableTheme(NIL)
  end
end
'def = av.GetProject.GetWorkDir.MakeTmp("nodes", "shp")
def = av.GetProject.MakeFileName("nodes", "shp")

def = FileDialog.Put(def, "*.shp", "New Node Theme")
if (def <> nil) then
  tbl = FTab.MakeNew(def, Point)
  if (tbl.HasError) then
    if (tbl.HasLockError) then
      MsgBox.Error("Unable to acquire Write Lock for file " +
def.GetBaseName, "")
    else
      MsgBox.Error("Unable to create " + def.GetBaseName, "")
    end
    return nil
  end
  fld1 = Field.Make("ID", #FIELD_DECIMAL, 8, 0)
  fld2 = Field.Make("node_name", #FIELD_CHAR, 16, 0)
  fld3 = Field.Make("node_type", #FIELD_SHORT, 4, 0)
  fld1.SetVisible( FALSE )
  tbl.AddFields({fld1,fld3,fld2})
  tbl.SetEditable(False)
  theTheme = FTheme.Make(tbl)
  theView.AddTheme(theTheme)
  theTheme.SetActive(TRUE)
  theTheme.SetVisible(TRUE)
  theView.SetEditableTheme(theTheme)
  av.GetProject.SetModified(true)
  'find GUI's tool and make it active ...
  for each x in av.GetActiveGUI.GetToolBar
    if (x.is(Tool)) then
      if(x.HasScript("xp.NewNodeTool")) then
        x.Select
        return nil
      end
    end
  end
        end
end
```

```
12 A.   'XP.NewNodeToolUpdate

theView = av.GetActiveDoc
theTheme = theView.GetEditableTheme

if (theTheme = nil) then
  SELF.SetEnabled(False)
else
  SELF.SetEnabled( (theTheme.GetFTab.FindField("Shape").GetType =
#FIELD_SHAPEPOINT)
    and (theTheme.GetFTab.FindField("Node_name") <> nil )
    and (theTheme.GetFTab.FindField("Node_type") <> nil) )
end

13 A.   'XP.OutNodes1

( 1 = takes geometry from XPX file, 0 - takes geo from a Theme)
'Select the file to read results *.out
xoutFN = FileDialog.Show("*.out","XP Output File (*.out)",
         "Select XP output file")
if ( xoutFN = nil ) then
  exit
end
'create a shape
'Select the file to read data *.xpx
xpxFN = FileDialog.Show("*.xpx","XP eXchange File (*.xpx)",
         "Select xpx export file")
if ( xpxFN = nil ) then
  exit
end
xpxFile = LineFile.Make(xpxFN, #FILE_PERM_READ)
xpxLines = xpxFile.GetSize
tempName    = xpxFN.GetBaseName.AsTokens(".").Get(0)
tempDir     = xpxFN.ReturnDir.AsString
defNodeName = FileName.Make( tempDir ).MakeTmp("out_nod", "shp" )
      ' Specify the output shapefile:
shpNodeName = FileDialog.Put(defNodeName,"*.shp","Create Shape File
of Nodes")
if ( shpNodeName = nil ) then
  exit
end
shpNodeFTab = FTab.MakeNew(shpNodeName, Point)
fieldsNode = List.Make
'fieldsNode.add(Field.Make("Node_id", #FIELD_SHORT, 4, 0 ))
fieldsNode.add(Field.Make("node_name", #FIELD_VCHAR, 12, 0 ))
fieldsNode.add(Field.Make("ground_el", #FIELD_FLOAT, 12, 2 ))
fieldsNode.add(Field.Make("UPCrown_el", #FIELD_FLOAT, 12, 2 ))
fieldsNode.add(Field.Make("MeanJ_el", #FIELD_FLOAT, 12, 2 ))
fieldsNode.add(Field.Make("J_change", #FIELD_FLOAT, 12, 4 ))
fieldsNode.add(Field.Make("MaxJ_el", #FIELD_FLOAT, 12, 2 ))
fieldsNode.add(Field.Make("Time_hr", #FIELD_short, 4, 0 ))
fieldsNode.add(Field.Make("Time_min", #FIELD_short, 4, 0 ))
fieldsNode.add(Field.Make("Surcharge", #FIELD_FLOAT, 12, 2 ))
fieldsNode.add(Field.Make("Depth_BG", #FIELD_FLOAT, 12, 2 ))
fieldsNode.add(Field.Make("Maxj_area", #FIELD_FLOAT, 12, 2 ))
'the following fields could be added:
'fieldsNode.add(Field.Make("N_type", #FIELD_SHORT, 4, 0 ))
'fieldsNode.add(Field.Make("N_x", #FIELD_DOUBLE, 20, 8 ))
'fieldsNode.add(Field.Make("N_y", #FIELD_DOUBLE, 20, 8 ))
shpNodeFtab.AddFields(fieldsNode)
nshpfd = shpNodeFtab.FindField("Shape")
'nidfd = shpNodeFtab.FindField("Node_id")
nnamefd = shpNodeFtab.FindField("node_name")
nge = shpNodeFtab.FindField("ground_el")
nupce = shpNodeFtab.FindField("UPCrown_el")
nmeje = shpNodeFtab.FindField("MeanJ_el")

nmxje = shpNodeFtab.FindField("MaxJ_el")
njac = shpNodeFtab.FindField("J_change")
ntoch = shpNodeFtab.FindField("Time_hr")
ntocm = shpNodeFtab.FindField("Time_min")
nfsme = shpNodeFtab.FindField("Surcharge")
nfmdbg = shpNodeFtab.FindField("Depth_BG")
nmja = shpNodeFtab.FindField("Maxj_area")
buf = xpxFile.ReadElt
if (xpxFile.IsAtEnd ) then
  MsgBox.Error( " File" + xpxFile.asstring ++"is empty.","")
  exit
end
if (buf = nil ) then
  MsgBox.Error( " Can not open " + xpxFile.asstring ,"")
        exit
end
av.ShowMsg ( "Reading Nodes")
av.ShowStopButton
recNum = 1
linecount = 1
node_no = 0
while ( TRUE)
  if (xpxFile.IsAtEnd ) then
    break
  end
            if ( buf.extract(0).ucase = "NODE" ) then
            node_no = node_no + 1
  'extract node name and remove it from the buffer
  'if node name is NIL ("") then insert a dummy string
  buf = buf.substitute("""""", """dummystring156""")
  node_name = buf.AsTokens("""").get(1).asstring.trim
```

```
        buf2 = buf.substitute(""""+node_name+"""", "")          end
        if( node_name = "dummystring156") then                  xsx = buf.middle(19,11).trim
            node_name = NIL                                      if(xsx.IsNumber) then
    end                                                             upce = xsx.AsNumber
        n_type = buf2.extract(1).AsNumber                       else
        n_x = buf2.extract(2).AsNumber                             upce = -999
        n_y = buf2.extract(3).AsNumber                          end
        thenode = n_x@n_y                                       xsx = buf.middle(30,10).trim
        nrec = shpNodeFTab.AddRecord                            if(xsx.IsNumber) then
        shpNodeFTab.SetValue(nshpFd, nrec, thenode)               meje = xsx.AsNumber
        shpNodeFTab.SetValue(nnameFd, nrec, node_name)          else
    end                                                           meje = -999
        buf = xpxFile.ReadElt                                   end
end                                                             xsx = buf.middle(40,10).trim
xoutFile = LineFile.Make(xoutFN, #FILE_PERM_READ)              if(xsx.IsNumber) then
buf = xoutFile.ReadElt                                           jac = xsx.AsNumber
if (xoutFile.IsAtEnd ) then                                    else
    MsgBox.Error( " File" + xoutFile.asstring ++"is empty.","")   jac = -999           .
    exit                                                       end
end                                                            xsx = buf.middle(50,9).trim
                                                              if(xsx.IsNumber) then
if (buf = nil ) then                                            mxje = xsx.AsNumber
    MsgBox.Error( " Can not open " + xoutFile.asstring ,"")
    exit                                                       else
end                                                             mxje = -999
xloop = TRUE                                                   end
while ( xloop )                                                xsx = buf.middle(59,6).trim
    if (xoutFile.IsAtEnd ) then                                if(xsx.IsNumber) then
        break                                                   toch = xsx.AsNumber
    end                                                        else
        if ( buf.contains("J u n c t i o n   S u m m a r y")) then    toch = -999
    'skip 10 lines                                             end
    for each i in 1..10                                        xsx = buf.middle(65,5).trim
        buf = xoutFile.ReadElt                                 if(xsx.IsNumber) then
    end                                                         tocm = xsx.AsNumber
            while (xloop)                                      else
        buf = xoutFile.ReadElt                                   tocm = -999
        'if empty line then stop reading                       end
        if(buf.trim = "") then                                 xsx = buf.middle(70,11).trim
            xloop = FALSE                                      if(xsx.IsNumber) then
            break                                               fsme = xsx.AsNumber
        end                                                    else
        name = buf.middle(0,8).trim                              fsme = -999
        xsx = buf.middle(9,10).trim                            end
        if(xsx.IsNumber) then                                  xsx = buf.middle(81,14).trim
            ge = xsx.AsNumber                                  if(xsx.IsNumber) then
        else                                                    fmdbg = xsx.AsNumber
            ge = -999                                          else
```

```
        fmdbg = -999
    end
    xsx = buf.middle(95,14).trim
    if(xsx.IsNumber) then
        mja = xsx.AsNumber
    else
        mja = -999
    end
            for each rec in shpNodeFTab
        name1 = shpNodeFTab.ReturnValue(nnameFd, rec)
        if ( name = name1 ) then
            shpNodeFTab.SetValue(nge, rec, ge)
            shpNodeFTab.SetValue(nupce, rec, upce)
            shpNodeFTab.SetValue(nmeje, rec, meje)
            shpNodeFTab.SetValue(njac, rec, jac)
            shpNodeFTab.SetValue(nmxje, rec, mxje)
            shpNodeFTab.SetValue(ntoch, rec, toch)
            shpNodeFTab.SetValue(ntocm, rec, tocm)
            shpNodeFTab.SetValue(nfsme, rec, fsme)
            shpNodeFTab.SetValue(nfmdbg, rec, fmdbg)
            shpNodeFTab.SetValue(nmja, rec, mja)
            break
        end
    end
  end
end


        buf = xoutFile.ReadElt
end
if (MsgBox.YesNo("Add junctions to a view?",
    "Junction Summary Statistics",true)) then
  lsview = list.make
  for each d in av.GetProject.GetDocs
    if (d.is(view) and d.getwin.isopen )then
      lsview.add(d)
    end
  end

lsview.add("<new view>")
  ViewChoice = MsgBox.ChoiceAsString( lsview, "Choose the view:",
"Add node-summary to the view" )
  if (ViewChoice = nil) then
    exit
  end

theTheme = FTheme.Make(shpNodeFTab)
  if(ViewChoice = "<new view>") then
    newview = view.make
```

```
        newView.GetWin.Open
        newView.AddTheme(theTheme)
        'bring the View to the front
        newView.GetWin.Activate

    else
        ViewChoice.AddTheme(theTheme)
    end

' theField = theTheme.getFtab.findField("Depth_BG")
' if(theField = nil) then exit end
  theLgd = TheTheme.getLegend
  theLgd.SetLegendTYpe(#Legend_type_color)
  thelgd.Natural(theTheme,"Depth_BG",2)
  cls0 = theLgd.getclassifications.get(0)
  cls0.setLabel("BGL depth = 0")
  cls0.setMinimum( -0.000001 )
  cls0.setMaximum(0.000001)

  cls1 = theLgd.getclassifications.get(1)
  cls1.setLabel("BGL depth > 0")
  cls1.setMinimum( 0.000002 )
  cls1.setMaximum(1000000000)
  theLgd.GetSymbols.get(0).setColor(color.getred)
  theLgd.GetSymbols.get(1).setColor(color.getblue)

  theTheme.UpdateLegend
end
exit

14 A.  'XP.StopEditing

                (based on toggle-editing  script)

theView = av.GetActiveDoc
editThm = theView.GetEditableTheme
doSave = MsgBox.YesNoCancel("Save Edits to "+editThm.GetName+
    "?", "Stop Editing", true)
if (doSave = nil) then
    return nil
end

if (editThm.StopEditing(doSave).Not) then
' save failed, remain editing this theme
    MsgBox.Info ("Unable to Save Edits to "
                + editThm.GetName +
                ", please use the Save Edits As option", "")
    return nil
```

```
end
` save succeeded
theView.SetEditableTheme(NIL)

  15 A.   STOP-EDITING UPDATE

theView = av.GetActiveDoc
if (theView.GetEditableTheme <> nil) then
  SELF.SetEnabled(TRUE)

else
  SELF.SetEnabled(FALSE)

end

16 A.  `XP.OutLinks1

( 1 = takes geometry from XPX file, 0 - takes geometry from a
Theme)
`Select the file to read results *.out

xoutFN = FileDialog.Show("*.out","XP Output File (*.out)",
        "Select XP output file")
if ( xoutFN = nil ) then
  return nil
`create a shape (or from existing Theme)
`Select the file to read data *.xpx
xpxFN = FileDialog.Show("*.xpx","XP eXchange File (*.xpx)",
        "Select xpx export file")
if ( xpxFN = nil ) then
  return nil
end
xpxFile = LineFile.Make(xpxFN, #FILE_PERM_READ)
xpxLines = xpxFile.GetSize
tempName     = xpxFN.GetBaseName.AsTokens(".").Get(0)
tempDir      = xpxFN.ReturnDir.AsString
defLinkName      = FileName.Make( tempDir ).MakeTmp("out_lnk",
"shp" )
`ask for the name of the shape file
shpLinkName = FileDialog.Put(defLinkName,"*.shp","Create Shape File
of Links")
if ( shpLinkName = nil ) then
   return nil
      shpLinkFTab = FTab.MakeNew(shpLinkName, Polyline)
fieldsLink = List.Make
fieldsLink.add(Field.Make("Link_id", #FIELD_SHORT, 4, 0 ))
fieldsLink.add(Field.Make("link_name", #FIELD_VCHAR, 12, 0 ))
fieldsLink.add(Field.Make("d_flow", #FIELD_FLOAT, 12, 2 ))
fieldsLink.add(Field.Make("d_velo", #FIELD_FLOAT, 12, 2 ))
```

```
fieldsLink.add(Field.Make("ver_depth", #FIELD_FLOAT, 12, 2 ))
fieldsLink.add(Field.Make("max_cflow", #FIELD_FLOAT, 12, 2 ))
fieldsLink.add(Field.Make("ftime_hr", #FIELD_short, 4, 0 ))
fieldsLink.add(Field.Make("ftime_min", #FIELD_short, 4, 0 ))
fieldsLink.add(Field.Make("max_cvelo", #FIELD_FLOAT, 12, 2 ))
fieldsLink.add(Field.Make("vtime_hr", #FIELD_short, 4, 0 ))
fieldsLink.add(Field.Make("vtime_min", #FIELD_short, 4, 0 ))
fieldsLink.add(Field.Make("flow_ratio", #FIELD_FLOAT, 12, 2 ))
fieldsLink.add(Field.Make("Upstream", #FIELD_FLOAT, 12, 2 ))
fieldsLink.add(Field.Make("Dwnstrm", #FIELD_FLOAT, 12, 2 ))
`the following fields could be added:
`fieldsLink.add(Field.Make("link_type", #FIELD_SHORT, 4, 0 ))
`fieldsLink.add(Field.Make("node_from", #FIELD_VCHAR, 12, 0 ))
`fieldsLink.add(Field.Make("node_to", #FIELD_VCHAR, 12, 0 ))
shpLinkFTab.AddFields(fieldsLink)
lshpfd = shpLinkFtab.FindField("Shape")
lnamefd = shpLinkFtab.FindField("link_name")
ldfl = shpLinkFtab.FindField("d_flow")

ldve = shpLinkFtab.FindField("d_velo")

lvdh = shpLinkFtab.FindField("ver_depth")
lmcf = shpLinkFtab.FindField("max_cflow")
lfth = shpLinkFtab.FindField("ftime_hr")
lftm = shpLinkFtab.FindField("ftime_min")
lmcv = shpLinkFtab.FindField("max_cvelo")
lvth = shpLinkFtab.FindField("vtime_hr")
lvtm = shpLinkFtab.FindField("vtime_min")
lfro = shpLinkFtab.FindField("flow_ratio")
lups = shpLinkFtab.FindField("Upstream")
ldwn = shpLinkFtab.FindField("Dwnstrm")
` ========== reading nodes from xpx file
buf = xpxFile.ReadElt
if (xpxFile.IsAtEnd ) then
  MsgBox.Error( " File" + xpxFile.asstring ++"is empty.","")
  return nil
end
if (buf = nil ) then
  MsgBox.Error( " Can not open " + xpxFile.asstring ,"")
  return nil
end
av.ShowMsg ( "Reading Nodes and Links")
av.ShowStopButton
recNum = 1
linecount = 1
node_no = 0
dn_xy = dictionary.make(50)
link_no = 0
```

```
while ( TRUE)
  if (xpxFile.IsAtEnd ) then
    break
  end
  if ( buf.extract(0).ucase = "NODE" ) then
    node_no = node_no + 1
    'extract node name and remove it from the buffer
    'if node name is NIL ("") then insert a dummy string
    buf = buf.substitute("""""", """dummystring156""")
    node_name = buf.AsTokens("""").get(1).asstring.trim
    buf2 = buf.substitute(""""+node_name+"""", "")
          if( node_name = "dummystring156") then
      node_name = NIL
    end
    n_type = buf2.extract(1).AsNumber
    n_x = buf2.extract(2).AsNumber
    n_y = buf2.extract(3).AsNumber
    thenode = n_x@n_y
    dn_xy.add(node_name, thenode)
  end
          if ( buf.extract(0).ucase = "LINK" ) then
    link_no = link_no + 1
    'extract link name and node names.


    xsx = buf.middle(63,5).trim
    if(xsx.IsNumber) then
      vth = xsx.AsNumber
    else
      vth = -999
    end
    xsx = buf.middle(68,5).trim
    if(xsx.IsNumber) then
      vtm = xsx.AsNumber
    else
      vtm = -999
    end
      xsx = buf.middle(73,9).trim
    if(xsx.IsNumber) then
      fro = xsx.AsNumber
    else
      fro = -999
    end
     xsx = buf.middle(82,9).trim
    if(xsx.IsNumber) then
      ups = xsx.AsNumber
    else
      ups = -999
```

```
  end
    xsx = buf.middle(91,9).trim
    if(xsx.IsNumber) then
      dwn = xsx.AsNumber
    else
      dwn = -999
    end
          for each rec in shpLinkFTab
    name1 = shpLinkFTab.ReturnValue(lnameFd, rec)
        if ( name = name1 ) then
          shpLinkFTab.SetValue(ldfl, rec, dfl)
          shpLinkFTab.SetValue(ldve, rec, dve)
          shpLinkFTab.SetValue(lvdh, rec, vdh)
          shpLinkFTab.SetValue(lmcf, rec, mcf)
          shpLinkFTab.SetValue(lfth, rec, fth)
          shpLinkFTab.SetValue(lftm, rec, ftm)
          shpLinkFTab.SetValue(lmcv, rec, mcv)
          shpLinkFTab.SetValue(lvth, rec, vth)
          shpLinkFTab.SetValue(lvtm, rec, vtm)
          shpLinkFTab.SetValue(lfro, rec, fro)
          shpLinkFTab.SetValue(lups, rec, ups)
          shpLinkFTab.SetValue(ldwn, rec, dwn)
          break
        end
      end
    end
  end

buf = xoutFile.ReadElt
end
if (MsgBox.YesNo("Add junctions to a view?",
    "Conduit Summary Statistics",true)) then
  lsview = list.make
  for each d in av.GetProject.GetDocs
    if (d.is(view) and d.getwin.isopen )then
      lsview.add(d)
    end
  end

lsview.add("<new view>")
  ViewChoice = MsgBox.ChoiceAsString( lsview, "Choose the view:",
"Add Conduit-summary to the view" )
  if (ViewChoice = nil) then
    exit
  end

  theTheme = FTheme.Make(shpLinkFTab)
  if(ViewChoice = "<new view>") then
```

```
    newview = view.make
    newView.GetWin.Open
    newView.AddTheme(theTheme)
    'bring the View to the front
    newView.GetWin.Activate
  else
    ViewChoice.AddTheme(theTheme)
  end


' theField = theTheme.getFtab.findField("flow_ratio")
' if(theField = nil) then exit end
  theLgd = TheTheme.getLegend
  theLgd.SetLegendTYpe(#Legend_type_color)
  thelgd.Interval(theTheme,"Flow_ratio",6)
  theColorRamp =
SymbolList.GetPreDefined(#SYMLIST_TYPE_COLORRAMP).Get(4)
  theLgd.GetSymbols.RampSavedColors(theColorRamp)
' cls0 = theLgd.getclassifications.get(0)
' cls0.setLabel("Flow_ratio = 0")
' cls0.setMinimum( -0.000001 )
' cls0.setMaximum(0.000001)
' cls1 = theLgd.getclassifications.get(1)
' cls1.setLabel("Flow_ratio > 0")
' cls1.setMinimum( 0.000002 )
' cls1.setMaximum(1000000000)
' theLgd.GetSymbols.get(0).setColor(color.getred)
' theLgd.GetSymbols.get(1).setColor(color.getblue)
  theTheme.UpdateLegend
end
exit


17 A.   'XP.Watershed-All

theView = av.GetActiveDoc
aPrj = theView.GetProjection
allthemes = theView.getThemes
pointThemeList = list.Make
lineThemeList = list.Make
gridThemeList = list.Make
polyThemeList = list.Make
for each t in allthemes
  if (t.Is(GTHEME)) then
    gridThemeList.add(t)
    continue
  end
  xx = t.getftab.findfield("shape")
```

```
  if (xx.gettype = #FIELD_SHAPELINE) then
    lineThemeList.add(t)
    continue
  end
  if (xx.gettype = #FIELD_SHAPEPOINT) then
    pointThemeList.add(t)
    continue
  end
  if (xx.gettype = #FIELD_SHAPEPOlY) then
    polyThemeList.add(t)
  end
end
        ' select a point theme
          if(pointThemeList.count = 0) then
  msgbox.error(theView.asstring++"does not contain any point theme","")
  return nil
end
ntheme = msgbox.list(pointThemeList,
      "Map of catchment outlets: ", "Watershed Outlets")
if(ntheme = nil).then
  return nil
end
pointFtab = ntheme.getFtab
        'make a list of points (needed only for version with
agrid.extractbypoint)
pointlist = list.make
pshfield = pointFtab.findfield("shape")
for each rec in pointFtab
  pointlist.add(pointftab.returnvalue(pshfield , rec) )
end
'make a list of fields
    'choose the ID field


pointIDfield = msgbox.listasstring(pointFtab.getfields ,
 "Select the field that contains point ID", "Node ID field")
if (pointIDfield = nil) then
  return nil
end
      idfdwidth = pointIDfield.getWidth
      idfdprec = pointIDfield.getPrecision
      idfdenum = pointIDfield.getType

'choose a field for drainage area
nfields = {"< new >"}.merge(pointFtab.getfields)
narea = msgbox.listasstring(nfields , "Select a field in which value of
the drainage area will be stored"++
NL+"existing values will be owerwritten!", "Catchment area field")
      if (narea = nil) then
```

```
        return nil
end
    if (narea = "< new >" ) then
    dafname = msgbox.input("Enter the name of catchment area field",
        "Catchment Area" , "Cat_Area")
    if (dafname = nil) then
        return nil
    end
    if ( pointFtab.isEditable.not) then
        pointFtab.SetEditable(true)
    end
    if ( pointFtab.CanAddFields) then
        f1 = Field.Make(dafname, #Field_float, 16, 5)
        narea = f1
        pointFtab.addfields({f1})
    else
        msgbox.warning("Can not add a field to"++pointFtab.asstring,
            "Can not edit node table")
        return nil
    end
    if ( pointFtab.isEditable) then
        pointFtab.SetEditable(false)
    end
end
foottoacre = 0.000022957
while(true)
    foottoacre = msgbox.input("Enter an area unit conversion factor, for
example,"+NL+
        "1 sq. ft (map unit is ft) = 0.000022957 ac","Area conversion
factor" , foottoacre.asstring)
    if(foottoacre = nil) then
        return nil
    end

            18 A.  Watershed  Area
***************** ****** ******* ******* *** **** ***** * ***
'make a grid of pour points:
flowaccgrid = flowdirgrid.flowaccumulation(NIL)
' create a theme
theGTheme22 = GTheme.Make(flowaccgrid)
' set name of theme
theGTheme22.SetName("fac1"++nTheme.GetName)
' add theme to the view
theView.AddTheme(theGTheme22)
        mPoint = MultiPoint.Make(pointlist)
'theSrcGrid= elevGrid.ExtractByPoints(mPoint, Prj.MakeNull, FALSE)
'better way is to make the outlet grid from FTab:
theOutGrid = Grid.MakeFromFtab(pointFtab, Prj.MakeNull, pointIdfield,
{thecellsize, theextent})
```

```
'since env. is defined instead of {thecellsize, theextent}  NIL can be
used
            'define the snap distance
snapcells = 1
while(TRUE)
    snapcells = msgbox.input("Snap distance to the major flow path in
cells"
    +NL+"1 cell ="++thecellsize.asstring++"map units)",
    "Snapping points to major flow routes", snapcells.asstring )
    if(snapcells = NIL) then snapcells = 1.asstring end
    if (snapcells.isnumber) then break end
        end
snapdistance = snapcells.asnumber * thecellsize

wshGrid = flowDirGrid.Watershed(theoutGrid.SnapPourPoint(flowaccgrid,
snapdistance))
'join point Ftab with grid outVtab and then with wshVtab

'check watershed outlets grid:
if (theOutGrid.getVTab <> Nil ) then
   OutVtab = theOutGrid.getvtab
   if (pointIDfield.IsTypeNumber) then
      Jfield = OutVtab.FindField("Value")
   else
      Jfield = OutVtab.FindField("S_Value")
   End
    else
msgbox.error("Error in converting point coverage to a grid ", "")
   return nil
end
'check watershed grid:
if(wshGrid.getVTab <> Nil ) then
      WshVtab = wshGrid.getVtab
   J2field = WshVtab.FindField("Value")

Countfd = WshVtab.FindField("Count")

Countfd.setalias("wshcount")

else
   msgbox.error("Watershed delineation error", "")
   return nil
end        .
        'joining outgrid with wshgrid
OutVtab.Join(OutVtab.FindField("Value"), WshVtab, J2field)
'add a field
        if(outVtab.isEditable.not) then
            outVtab.SetEditable(true)
   end
```

```
        if(outVtab.CanAddFields) then
    fdadd = field.make("wshtempfield", countfd.gettype, countfd.getwidth,
countfd.getprecision)
    outVtab.Addfields({fdadd})
        else
    msgbox.error("Can not add a field to watershed outlet vtab","")
    return nil
            end
xxx = outVtab.calculate("[wshcount]", fdadd )
if(outVtab.IsEditable) then
    outVtab.SetEditable(false)
        end
outVtab.unjoinall
        'joining point coverage with outgrid and calculating the watershed
area
xxx1 = foottoacre.asnumber.setformat("dddddddd.ddddddddddddddddd")
exp = xxx1.asstring+"*[wshtempfield]"
        pointFtab.Join(pointIDfield, OutVtab, Jfield)
'calculate area:
        if(pointFtab.isEditable.not) then
    pointFtab.SetEditable(true)
end
    xxx = pointFtab.calculate(exp, narea )
        if(pointFtab.IsEditable) then
    pointFtab.SetEditable(false)
end
    pointFtab.unjoinall
' rename data set

'aFN = av.GetProject.GetWorkDir.MakeTmp("Watersheds", "")
'wshGrid.Rename(aFN)
' create a theme
theGTheme = GTheme.Make(wshGrid)

' set name of theme
theGTheme.SetName("wsh1"++nTheme.GetName)
' add theme to the view
theView.AddTheme(theGTheme)


' remove all grids used for area that are no longer needed


19 A  Watershed  Width/Length
*** **** **** **** ******** ****** ********* ***** ***** ********
' SWMM uses 2 * river length. Here the watershed width is
' 2 * the longest flow path in a given subcatchment.
' The watershed width (here the longest flow path) is calculated
' utilizing the elevation grid created for calculating
' the catchment area.
```

```
' the longest flow length can be multiplied by a coefficient
' to make the flow length compatible with the XP's watershed width
' Here the multiplier is equal to one (as string)

multiplier = "1"
' choose a field for watershed width:
nwidth = msgbox.listasstring(nfields , "Select a field for the catch-
ment width"++
    NL+"existing values in selected records will be owerwritten!",
"Watershed width field")
            if (nwidth = nil) then
    return nil
end

if (nwidth = "< new >" ) then
wwfname = msgbox.input("Enter the name of watershed width field",
        "Watershed width" , "wsh_width")
    if (wwfname = nil) then
        return nil
    end
    if ( pointFtab.isEditable.not) then
        pointFtab.SetEditable(true)
    end
    if ( pointFtab.CanAddFields) then
        f1 = Field.Make(wwfname, #Field_float, 16, 2)
        nwidth = f1
        pointFtab.addfields({f1})
    else
        msgbox.warning("Can not add a field to"++pointFtab.asstring,
            "Can not edit node table")
        return nil

    if ( pointFtab.isEditable) then
        pointFtab.SetEditable(false)
    end
end


19 A  calculate watershed width

' elevgrid = filled elevation grid
' flowdirgrid = flow direction grid
' wshgrid = catchments (watersheds)
flgGrid = flowdirgrid.FlowLength(NIL, FALSE)

' check if output is ok
if (flgGrid.HasError) then

msgbox.error("can not calculate watershed width ", "")
    return nil
```

```
end

'the longest zonal length = zonal_max - zonal_min
' zoneField = field from wshgrid (zoneobject) =

aFNstat = av.GetProject.GetWorkDir.MakeTmp("Wshstat", "dbf")
'LongestGrid =
flgGrid.ZonalStats(#GRID_STATYPE_RANGE,wshgrid,Prj.makenull,J2Field,FALSE)
 WstatVtab =
flgGrid.ZonalStatsTable(wshgrid,Prj.makenull,J2Field,FALSE,aFNstat)

'the statistics table can be visible ...
'xTable = Table.make(WstatVtab)
'xtable.getwin.activate
if (WstatVtab.HasError) then
  msgbox.error("can not create a wshd statistics table  ", "")
  return nil
end

'add node-ID field to the WstatVtab
if(WstatVtab.isEditable.not) then
  WstatVtab.SetEditable(true)
        end
if ( WstatVtab.CanAddFields) then
  f3 = Field.Make(pointIDfield.asstring, idfdenum, idfdwidth, idfdprec)
  WstatVtab.addfields({f3})
else
  msgbox.warning("Can not add a field to"++WstatVtab.asstring,
      "Bad watershed statistics table")
  return nil
end
WstatVtab.Join(WstatVtab.findfield("Value"), OutVtab,
OutVtab.findfield("Value"))
exp3 = "["+Jfield.asstring+"]"
xxx4 = WstatVtab.calculate(exp3, f3 )
if(WstatVtab.IsEditable) then
  WstatVtab.SetEditable(false)
        end
WstatVtab.unjoinall

'join WstatVtab with point Table
' find a field which name contains the string "range"
rangefield = NIL
for each f in WstatVtab.getfields
  if ( f.asstring.contains("_ran") ) then
    rangefield = f
    break
  end
end
```

```
if (rangefield = NIL) then
  msgbox.warning("Can not calculate watershed length",
      "Bad watershed statistics table")
  return nil
end
pointFtab.Join(pointIDfield, WstatVtab, f3)

'calculate (copy) watershed length:
if(pointFtab.isEditable.not) then
  pointFtab.SetEditable(true)
end
exp4 = multiplier+"*["+rangefield.asstring+"]"
xxx2 = pointFtab.calculate(exp4, nwidth )
if(pointFtab.IsEditable) then
  pointFtab.SetEditable(false)
end
pointFtab.unjoinall

WstatVtab.deactivate
' WstatVtab.isbase TRUE
' File.candelete(aFNstat) TRUE

WstatVtab = NIL
av.purgeobjects
File.delete(aFNstat)   'ERROR CODE 13

'*********************************************************
'* 21 A.  Watershed  average  slope        *
'*********************************************************
' First, the grid of slopes is created and then the average
' value is calculated for each watershed.

'choose a field for watershed average slope:
nslope = msgbox.listasstring(nfields , "Select a field for the average
watershed slope"++
    NL+"existing values will be owerwritten!", "Watershed slope field")

if (nslope = nil) then
  return nil
end

if (nslope = "< new >" ) then
wsfname = msgbox.input("Enter the name of watershed slope field",
      "Watershed slope" , "wsh_slope")
  if (wsfname = nil) then
    return nil
  end
  if ( pointFtab.isEditable.not) then
```

```
      pointFtab.SetEditable(true)
    end
  if ( pointFtab.CanAddFields) then
    f4 = Field.Make(wsfname, #Field_float, 16, 8)
    nslope = f4
    pointFtab.addfields({f4})
  else
    msgbox.warning("Can not add a field to"++pointFtab.asstring,
      "Can not edit node table")
    return nil
  end
  if ( pointFtab.isEditable) then
    pointFtab.SetEditable(false)
  end
end


' ========  raw elevation grid or filled one =========
if(msgbox.yesno("Slope can be calculated from a grid which contains
depressions"+NL+
                "Do you want to fill depressions ?", "Filling Sinks",
True) )then

' filling all sinks is based on an ESRII example skrips
  elevGrid = ELgrid
  sinkCount = 0
  numSinks = 0
  while (TRUE)

    flowDirGrid = elevGrid.FlowDirection(FALSE)
    sinkGrid = flowDirGrid.Sink
    if (sinkGrid.GetVTab = NIL) then
      ' check for errors
      if (sinkGrid.HasError) then
msgbox.warning("Error in the Sink Grid. Skipfilling process.","")
      break
      end
      sinkGrid.BuildVAT
    end
    ' check for errors
    if (sinkGrid.HasError) then
msgbox.warning("Error in the Sink Grid. Skip filling process.","")
      break
    end

  if (sinkGrid.GetVTab <> NIL) then
      theVTab = sinkGrid.GetVTab
                  numClass = theVTab.GetNumRecords
 newSinkCount = theVTab.ReturnValue(theVTab.FindField("Count"),0)
    else
```

```
      numClass = 0
      newSinkCount = 0
    end
    if (numClass < 1) then
      break
    elseif ((numSinks = numClass) and (sinkCount = newSinkCount))
then
      break
    end
    waterGrid = flowDirGrid.Watershed(sinkGrid)
    zonalFillGrid = waterGrid.ZonalFill(elevGrid)
    fillGrid = (elevGrid <
(zonalFillGrid.IsNull.Con(0.AsGrid,zonalFillGrid))).Con(zonalFillGrid,elevGrid)
    elevGrid = fillGrid
    numSinks = numClass
    sinkCount = newSinkCount
  end
else
  elevgrid = el2grid
end
slpGrid = elevgrid.slope(NIL, TRUE)
' true - slope = a percent rise
' check if output is ok
if (slpGrid.HasError) then
  msgbox.error("can not calculate watershed slope ", "SLOPE")
  return nil
end

22 A.   TEST - ADD SLOPE GRID TO VIEW

' rename data set
'aFN = av.GetProject.GetWorkDir.MakeTmp("Wshslp", "")
'slpGrid.Rename(aFN)
' create a theme
'theGTheme4 = GTheme.Make(slpGrid)
' set name of theme
'theGTheme4.SetName("slp1"++nTheme.GetName)
' add theme to the view
'theView.AddTheme(theGTheme4)

aFNstat2 = av.GetProject.GetWorkDir.MakeTmp("Wshstat", "dbf")
WstatVtab2=slpGrid.ZonalStatsTable(wshgrid,Prj.makenull,J2Field,FALSE,aFNstat2)

if (WstatVtab2.HasError) then
msgbox.error("can not create a wshd statistics table  ", "SLOPE")
  return nil
end

'add node ID field to the WstatVtab2
```

```
if(WstatVtab2.isEditable.not) then
   WstatVtab2.SetEditable(true)
end
      if ( WstatVtab2.CanAddFields) then
'f3 = Field.Make(pointIDfield.asstring, idfdenum, idfdwidth, idfdprec)
   WstatVtab2.addfields({f3})
else
   msgbox.warning("Can not add a field to"++WstatVtab2.asstring,
        "Bad watershed statistics table")
   return nil
end
WstatVtab2.Join(WstatVtab2.findfield("Value"), OutVtab,
OutVtab.findfield("Value"))
exp3 = "["+Jfield.asstring+"]"
xxx4 = WstatVtab2.calculate(exp3, f3 )
if(WstatVtab2.IsEditable) then
   WstatVtab2.SetEditable(false)
end
WstatVtab2.unjoinall

'join WstatVtab2 with point Table
' find a field which name contains the string "mean"
meanfield = NIL
for each f in WstatVtab2.getfields
   if ( f.asstring.contains("_mea") ) then
      meanfield = f
      break
   end
end

if (meanfield = NIL) then
   msgbox.warning("Can not calculate watershed slope",
        "Bad watershed statistics table")
   return nil
end
pointFtab.Join(pointIDfield, WstatVtab2, f3)

'calculate watershed slope:
if(pointFtab.isEditable.not) then
   pointFtab.SetEditable(true)
end
exp4 = "["+meanfield.asstring+"]/100"
xxx2 = pointFtab.calculate(exp4, nslope )
if(pointFtab.IsEditable) then
   pointFtab.SetEditable(false)
end
pointFtab.unjoinall

WstatVtab2.deactivate
```

```
WstatVtab2 = NIL
av.purgeobjects
File.delete(aFNstat2)    'ERROR CODE 13


   23 A.   Percent of Watershed  Impervious

'This version requires a polygon map of Land use. One of
'the items should contain percentage of impervious area.

' choose a field for impervious:
nimp = msgbox.listasstring(nfields , "Select a field for the % impervi-
ous area"++
   NL+"existing values in selected records will be owerwritten!",
"Impervious (%) area field")

if (nimp = nil) then
   return nil
end

if (nimp = "< new >" ) then
   wifname = msgbox.input("Enter the name of % Impervious area field",
        "Impervious area" , "wsh_imp")
   if (wifname = nil) then
      return nil
   end
   if ( pointFtab.isEditable.not) then
      pointFtab.SetEditable(true)
   end
   if ( pointFtab.CanAddFields) then
      f5 = Field.Make(wifname, #Field_float, 16, 3)
      nimp = f5
      pointFtab.addfields({f5})
   else
      msgbox.warning("Can not add a field to"++pointFtab.asstring,
         "Can not edit node table")
      return nil
   end
   if ( pointFtab.isEditable) then
      pointFtab.SetEditable(false)
   end
end

' Select the LU coverage and the item
if(polyThemeList.count = 0) then
   msgbox.error(theView.asstring++"does not contain any polygon
theme","")
   return nil.
end
```

```
lutheme = msgbox.list(polyThemeList,
     "Map of Land Use: ", "Map of Impervious Area")
if(lutheme = nil) then
   return nil
end

luFtab = lutheme.getFtab

'make a list of fields

'choose the ID field
impfield = msgbox.listasstring(luFtab.getfields ,
 "Select the field that contains % impervious", "% impervious field")
if (impfield = nil) then
   return nil
end
' calculate % of watershed that is impervious

' wshgrid = catchments (watersheds)
' LU map needs to be converted into raster format:
luFtab.getSelection.clearall
luFtab.UpdateSelection
impgrid = Grid.MakeFromFtab(luFtab, Prj.MakeNull, impfield, Nil)
' check if output is ok
if (impGrid.HasError) then
  msgbox.error("can not calculate impervious area ", "")
   return nil
      end
aFNstat3 = av.GetProject.GetWorkDir.MakeTmp("Wshstat", "dbf")
WstatVtab3=impGrid.ZonalStatsTable(wshgrid,Prj.makenull,J2Field,FALSE,aFNstat3)
     if (WstatVtab3.HasError) then
        msgbox.error("can not create a wshd statistics table  ", "")
   return nil
end

'add node ID field to the WstatVtab3
if(WstatVtab3.isEditable.not) then
  WstatVtab3.SetEditable(true)
end
       if ( WstatVtab3.CanAddFields) then
    f3 = Field.Make(pointIDfield.asstring, idfdenum, idfdwidth,
idfdprec)
          WstatVtab3.addfields({f3})
else
  msgbox.warning("Can not add a field to"++WstatVtab3.asstring,
       "Bad watershed statistics table")
   return nil
end
WstatVtab3.Join(WstatVtab3.findfield("Value"), OutVtab,
```

```
OutVtab.findfield("Value"))
exp3 = "["+Jfield.asstring+"]"
xxx4 = WstatVtab3.calculate(exp3, f3 )
if(WstatVtab3.IsEditable) then
  WstatVtab3.SetEditable(false)
End

WstatVtab3.unjoinall

'join WstatVtab3 with point Table
' find a field which name contains the string "mean"
meanfield = NIL
for each f in WstatVtab3.getfields
   if ( f.asstring.contains("_mea") ) then
     meanfield = f
     break
   end
end
if (meanfield = NIL) then
   msgbox.warning("Can not calculate percentage of impervious",
        "Bad watershed statistics table")
   return nil
end
pointFtab.Join(pointIDfield, WstatVtab3, f3)

'copy values of imperv.:
if(pointFtab.isEditable.not) then
  pointFtab.SetEditable(true)
end
exp6 = "["+meanfield.asstring+"]"
xxx4 = pointFtab.calculate(exp6, nimp )
if(pointFtab.IsEditable) then
  pointFtab.SetEditable(false)
end
pointFtab.unjoinall

WstatVtab3.deactivate
WstatVtab3 = NIL
av.purgeobjects
File.delete(aFNstat3)

'*********************************************************
'*                    E N D    02/22/2000               *
'*********************************************************
```

APPENDIX B: README FILE

APPENDIX B

README FILE

This Readme File assists the user to use this software in ArcView 3.1 environment while utilizing its Spatial Analyst Extension. The principal task of this research is to use ArcView to generate an input data group for XP-SWMM. Any watershed runoff model can use these generated data. The software introduced in Appendix B consists of three main tools, which are written, in Avenue, the script language of ArcView. These codes are copied to ArcView program and work as extensions to it.

The software presented in this Appendix B consists of 21 script files. They should be copied to the ArcView subdirectory. The user should locate the directory for ArcView program. While installing the ArcView, if default options were used, then ArcView will be in ESRI Directory. The user should open ArcView Subdirectory and copy all these scripts in EXT32 Subdirectory. If ESRI Directory is located in C drive, then user can use the following path for copying code copying: *C:\ESRI\ArcView\Ext32*

After starting the ArcView program, user should open file button and click on Extensions and check the boxes for Spatial Analyst, WATERSHED, NETWORK and PRESENTATION. The Spatial Analyst and the WATERSHED extensions are pre-request for developing the watershed parameters. If the user wants to draw a runoff collection system (pipes and manholes in ArcView environment), he should select NETWORK extension also. The PRESENTATION extension is used for displaying the simulated results from XP-SWMM. If five themes for ArcView, such as DEM, soil and land-use maps, node and pipe (drainage plan of a watershed) are available, and if the above said four extensions are used, it is possible to have GIS calculate runoff parameters of any urban or non urban watershed.

This program consists of three ArcView extensions: (1) **WATERSHED**, (2) **NETWORK**, and (3) **PRESENTATION**. They delineate subbasins, develop subbasin parameters, display simulated data from XP-SWMM and support data exchange between ArcView and XP-SWMM. The summary of GIS Input, Output and Presentation themes are provided in Figure 1B.

154

1 B. WATERSHED

The **WATERSHED Extension** constructs a *Flow Direction Grid*, delineates subbasins, calculates subbasin parameters (such as area, width, slope, and percent impervious area). It creates XP export file to allow the transfer of GIS generated parameters for subbasins into the XP-SWMM. This extension adds a new menu that contains three items: (1) *Input*, (2) *Calculate*, and (3) *Export*. The *Input* Item takes at least five themes; DEM, Pipes, Manholes, Land-use and Soil to generate subbasin parameters. The *Calculate* Item estimates Area, Width, Slope, and % Imperviousness of the subbasins. The *Export* Item exports all the calculations done by this Watershed Extension to XP-SWMM.
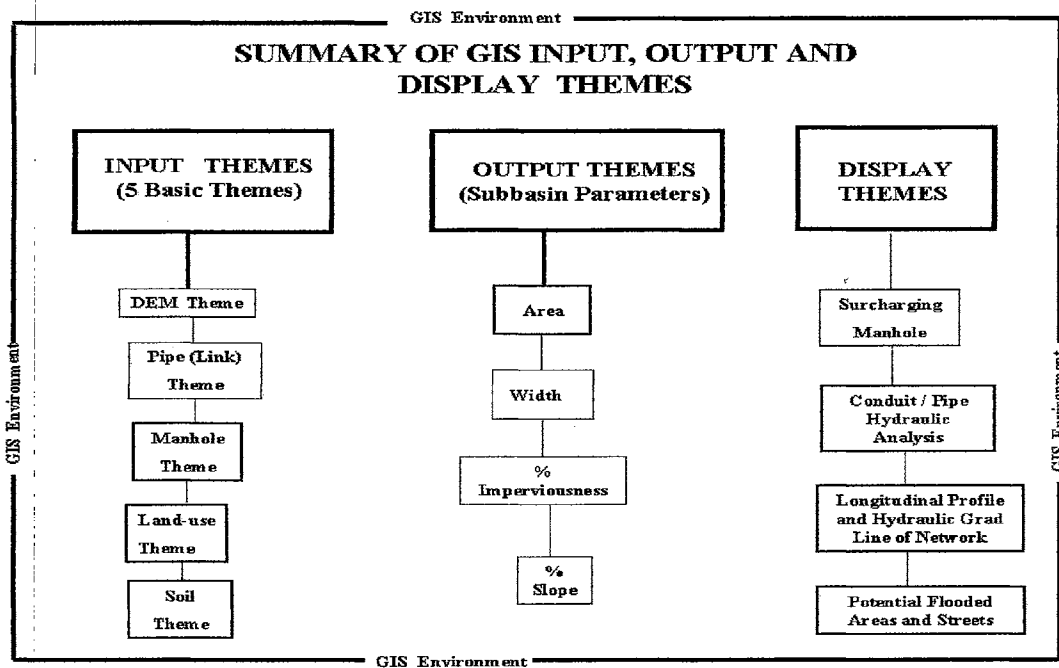
Figure 1B. (Appendix-B) List of Input, Output and Presentation Themes

2 B. NETWORK

The **NETWORK Extension,** creates and edits a basic drainage network (nodes and links) in ArcView environment. Also this extension extracts a drainage network from XP-SWMM. It consists of three tools for: (1) *Network Creation*, (2) *Network Import* and (3) *Editing of nodes*, and *links* in an existing runoff or sewer network.

3 B. PRESENTATION Extension

This extension, consists of two tools: *XP-Results* and *Zones*, Its displays results from XP-SWMM in the form of maps, profiles, surcharging manholes and flood prone zones. It adds a new menu to the menu bar containing three items: (1) *Junctions*, (2) *Conduits*, and (3) *Profiles*. *Junction* imports the results of hydrologic and hydraulic calculations of XP-SWMM into ArcView. The menu item *Conduits* imports the results of XP program calculations into ArcView. *Profile* shows a map of nodes and conduits using information imported form XP-SWMM.

4 B. Subcatchment Delineation

A new menu is added to the menu bar by loading the *WATERSHED* Extension into ArcView. This menu contains three items— *Input, Calculate*, and *Export.* Menu item *INPUT* develops the *Flow Direction Grid* to delineate subcatchments. A dialog box shown in Figure 2B opens up whenever *Input Item* is selected. The various names used in figures in this Readme File for various themes, shape files and data are only for narrative purposes. The user can have any name for themes and shape files as long as he follows ArcView's file creation and naming rules.

Two *CORE THEMES* and seven *DERIVED THEMES* make the basis for the whole process of subbasin delineation, as provided in Figure 2B. Minimum two *CORE THEMES* are required as Input maps to delineate subcatchments: (1) *Point Theme* of subcatchment outlets; (such as *Alex_Node.shp*, it can be any name) (2) *Grid Theme of elevations (dem2)*, which is actually a digital elevation model (*DEM*). The ID of the manhole should also be specified, i.e., Outlet ID field (*Nodeid*). and (3) if adjustment of the elevation grid before flow direction is calculated, a *Line Theme of the conduits (pipes)* and/or a *Line Theme of Flow Barriers* must be specified.

Adjustment of Flow Path is important for the correct delineation of subcatchments—because points (manholes) identifying subcatchment runoff collection points must be located on the major natural flow path. These points are usually located off the channel determined from DEM due to imperfection of the digital terrain representation or its insufficient resolution (both horizontal and vertical) and software limitations. The first *DERIVED THEME*, shown by the first Check Box must be selected to execute the "*Adjust elevations to force flow paths*" command and to make a grid theme of adjusted elevations. A name should be entered for *Flow path (line theme),* (such as *Dry_cr.shp*) as shown in the fifth line in Figure 2B.
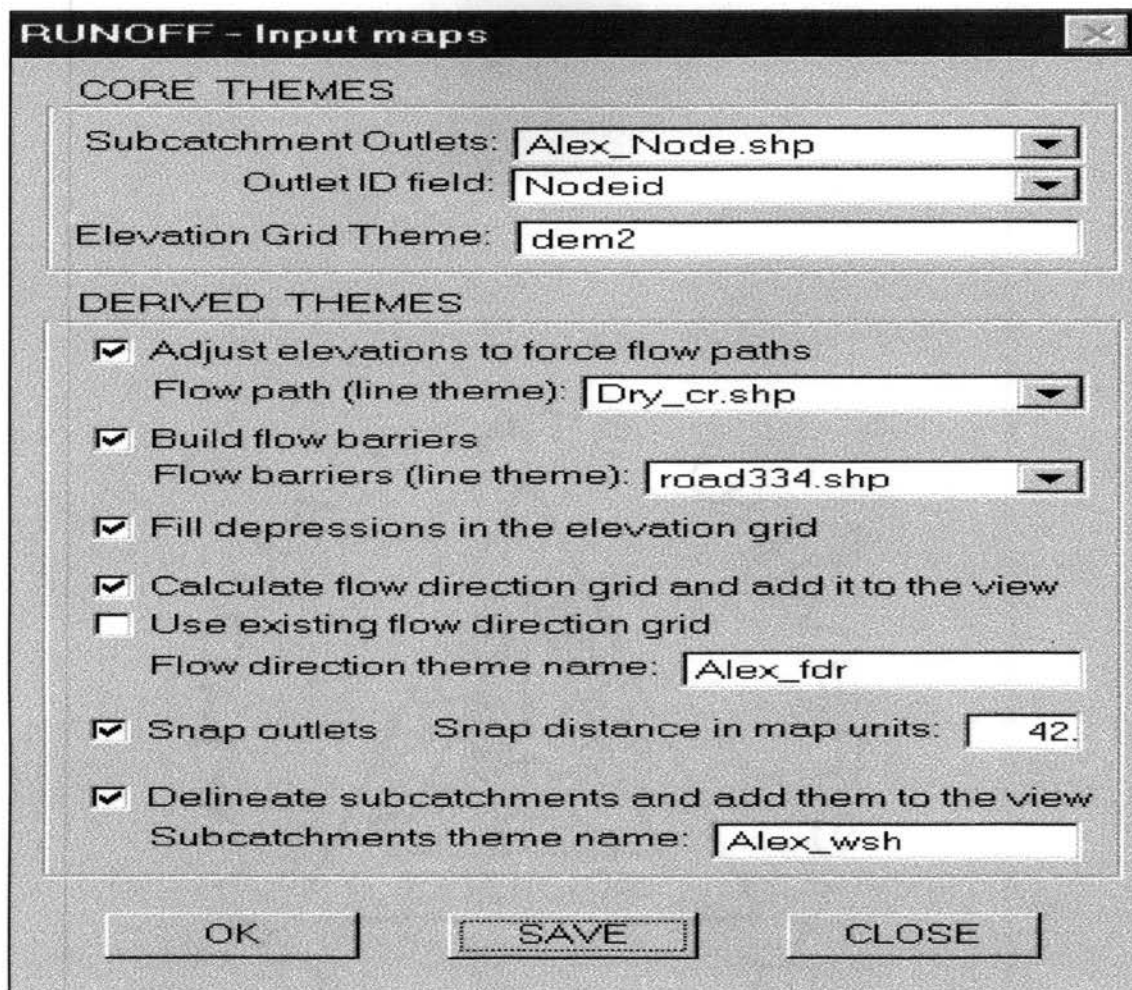
Figure 2B. (Appendix-B) Dialog Box for Input Data Representing Core Themes
and Derived Themes for the Delineation of Subcatchments

A method to adjust DEM to force the flow direction calculated from DEM to follow natural stream flow paths is incorporated in the code. The "Artificial Ditches" are created by simply lowering elevation of DEM's cells that represent conduits or stream or by raising elevation of cells that do not represent streams. This grid calculates the **Flow Direction Grid**, which is a crucial grid for subcatchment delineation, for calculation of drainage area and the length of longest flow path.

During this "*Burn-In Streams*" process, the WATERSHED Extension converts the link/channel map into a grid representation. The process of building **Flow Barriers** is executed by selecting the second Check Box in Figure 2B. "**Build Flow barriers, Flow Barriers (line Theme)**". If the user wants to use this tool, then he should enter the name of Flow Barrier Grid, such as **road334.shp**, as given in Figure 2B, and select this Check box.

The Check Box, *"Fill depressions in the elevation grid"*, Figure 2B, represents a derived Item which is incorporated to fill the sinks or depressions in a DEM. The sink is a topological feature in which water flows into a point that has no out-fall. In a DEM, the sink is a cell surrounded by cells with higher elevations. As sinks cause discontinuities in flow paths and any terrain having this type of surface features should be smoothed.

The fourth Check Box in Figure 2B, calculates the *Flow Direction Grid*. This Tool consists of two options: "**Calculate flow direction and add it to the view**" and "**Use existing flow direction grid**". The user should give a name for this theme, if it is new, (such as **Alex_fdr**.). Otherwise he can enter the name of the old, previously calculated *Flow Direction Grid* This derived theme is used to delineate different sets of subcatchments. Since the drainage area is defined by its manhole, different themes (maps) of subcatchment outlets may be created and employed for subcatchment delineation by using the same *Flow Direction Grid* for all program runs.

The location of the manhole (inlet) of subcatchment on major flow path can be ensured by introducing the "*Snap Outlets*" option. This function snaps pour points to the cell of a maximum value in weight grid (such as a *Flow Accumulation Grid*) within a specified snap distance. It searches within a snap distance around each pour point (manhole) for the cell of highest accumulated flow and moves the subcatchment outlet to that location. Snap distance can be specified in map units such as foot or meter or, any negative number. If user enters a value of 82 in the box of "*Snap distance in map units*", a snap distance of 82 map units (meter or foot) will be used for the analysis. The negative number represents the value of the snap distance in terms of number of cells. For example, if a number 2 with a negative value, i.e., (*-2*) is used, a snap distance of twice the width of the cell will be applied during the spatial analysis.

The selection of the last Check Box, Figure 2B, starts the delineation process of the subcatchments. The last Check Box, "*delineate subcatchments and add them to the view*" will work only when all the other check boxes are selected. It needs to have the name of the subcatchments which are to be delineated (*Alex_wsh*" given in the last line of Figure 2B). The "**CLOSE**" button, Figure 2B, closes the dialog box and subsequently no action is performed. The "**SAVE**" button saves information entered by the user in View's tag—when this dialog box is reopened, all previously entered information is displayed. The "*OK*" button performs all calculations selected in the dialog box.

5 B. Calculation of Subcatchment Parameters

The dialog box, for *Calculate* Tool, Figures 3B, consists of two parts. The first one requires "*INPUT AND OUTPUT THEMES*" The second part describes the "*PARMETERS TO BE CALCULATED*". The First part requires one Input Theme (*Subcatchments*) and its *ID Field*. The Input Theme and its ID Field are shown in the first line of the Dialog Box. The user should provide the name of the theme for delineated subcatchments, (the name used in Figure 1B, on the last line; *Alex_wsh*). The Output theme and its ID field are provided in the second line in the Dialog Box, Figure 3B. User should provide the name of the Node theme (*Alex_Node.shp*), which will be created and its ID Field (*Nodeid*). The Node (manhole) theme will have all the parameters connected to it.

The second part of the *Calculate* Tool, estimates the parameters, given in Figure 1B, i.e., area (*Ws_area*), Width (*Ws_width*), Slope (*Ws_Slope*) and % Imperviousness (*Ws_inpervious*). Figure 3B consists of all of these watershed parameters are in. The user must select the Check Box in front of any parameter to be estimated. "*Runoff Subcatchment Parameters*" are designated to enter data and select themes that are required to calculate subcatchment parameters and output fields. The major input theme is the DEM grid that represents subcatchments. Another input theme is a polygon map of land use with information about percentage of impervious area. The user can select a conversion factor to change the units of the subcatchment area, for example, change square feet to acres or enter any value in the input box.

For the calculation of subbasin area (*Ws_area*), the user also enter the appropriate *Units conversion factor* (such as 2.2957^{-05} *ft² to acres*) The Width parameter needs four types of data from the user. User should select the *Method* of estimating the width value. The width *Method* consists of three scenarios shown in Figure 4B for the estimation of subcatchment w*idth*: (1) the subcatchment area divided by the maximum length of the flow path; (2) subcatchment area divided by the diameter of the circle contained within the subwatershed; *and* (3) diameter of the circle contained within the subcatchment.

Also the value of the *multiplier* must be given, and its value can be anywhere between 1 and 2 (*multiplier = 1.8* , Figure 3B).   The user is advised to check the SWMM Manual for further details about the multiplier. The *Output field* should be entered (*Ws-width*) and the name of the

159

Flow Direction Grid (Figure 2B, fourth check box) should be provided. This *Flow Direction Grid* (*alex_fdr*) is necessary to calculate the elevation difference to get the gravity flow. The subcatchment width multiplier is introduced to adjust the estimated values, to account for subcatchment shape, or one- or two-sided stream inflow.

Figure 3B. (Appendix-B). Dialog Box for Calculating Subcatchment Parameters

6 B. Exporting Subcatchment Parameters

The dialog box, given in Figure 5B, describes "*Export Subcatchment Parameters*" for runoff as shown by four Check Boxes. It is designed to store sub-catchment parameters in a file that can be read by XP-SWMM. The user must specify the theme that contains subcatchment parameters, theme's attribute table field that stores subcatchment IDs, and the fields that hold parameters.

Figure 5B also describes three options of exporting data, i.e., *Append*, *Replace All*, and *Replace by ID*. The "*Append*" option writes subcatchment parameters in lines added at the end of the existing file. If "*Replace All*" is selected, the export procedure deletes all lines that are related to selected parameters from the existing file, and adds estimated parameters at the end of the export file. The "*Replace by ID*" option is similar to the "*Replace All*"
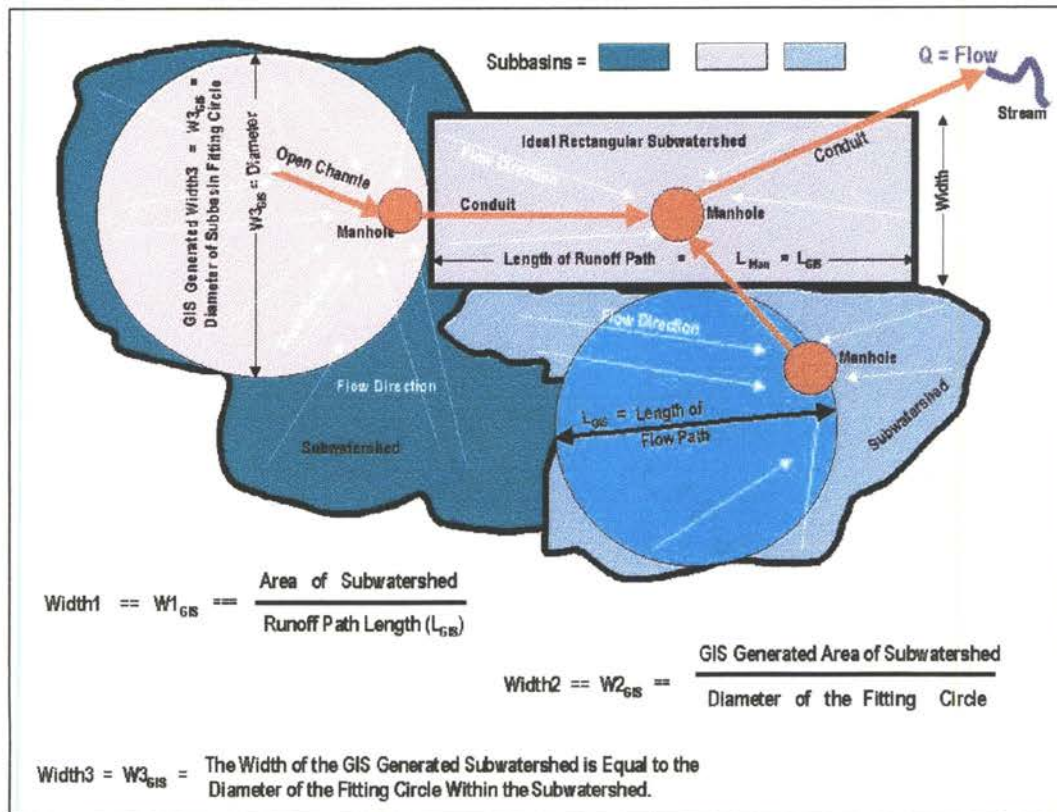
Figure 4B. (Appendix-B) Three Methods For Generation of GIS-Based Subwatershed Width

Figure 5B. (Appendix-B). Dialog Box for Exporting Parameters to XP-SWMM

7 B. Network Construction

The network construction can be drawn by using four tools of **NETWORK Extension:** (1) *New Nodes and Links*, (2) *Add Nodes and Links*, (3) *Import Nodes and Links*, (4) *Import* and *Export Nodes + Links*. This extension creates a simple map of a drainage network within the ArcView environment, extracts a drainage network from the XP-SWMM export file, converts and stores ArcView's shapes representing nodes and links into an XP-SWMM readable file. Actually any user familiar with XP-SWMM program will not have any trouble using this NETWORK Tool.

This extension adds menu *XP-Network* to the view's menu bar as shown in Figure 6B. It also adds three buttons, *N, L,* and *S,* to the View's tool bar.

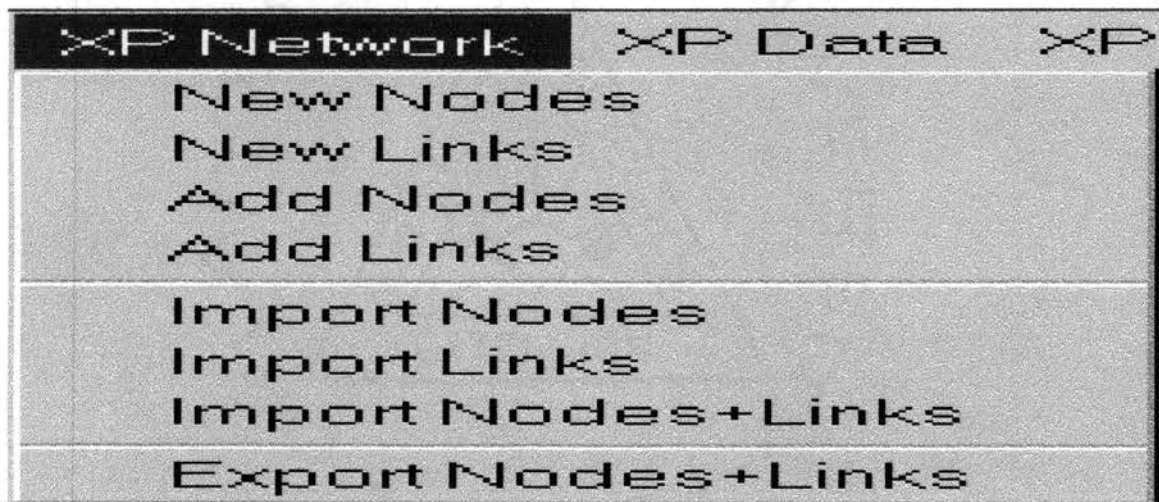Figure 6B. (Appendix-B). Menu of NETWORK Extension (XP-Network)

The "*New Nodes*" menu item, shown in Figure 6B, allows the user to create a new shape of points which represents such network nodes as manholes, junctions, or outlets. When this item is selected, user is asked to specify a name of *Point Shape File*. After the name is entered, the program makes tool button "*N*" active. The user can click the mouse within active View in the location where a node/manhole is to be created. The user is then prompted for basic information required to transfer geographic information to XP-SWMM, i.e. node name and node type. The process of building the map of nodes can be terminated by pressing the tool button "*S*".

The "*New Links*" menu item allows the user to create a new line shape of conduits, When this item is selected, the user is asked to specify the name of node theme on which the link theme

will be constructed. To create a map of conduits, a map of manholes must exist. The user can then enter a name of the line shape file to be created. After entering the name of the link theme, the tool button "*L*" gets active. The upstream node of a link should be selected by clicking the mouse on the selected point. A confirmation is displayed about the upstream node selection. The selection can be accepted about the upstream node by clicking on "**Yes**" or "**No**" buttons and subsequently the program is ready to guide the user to make another selection.

After the upstream node is accepted, the user must select the downstream node. It is accepted in the same manner as selecting the upstream node. The process of building the map of links can be terminated by pressing tool button "*S*".

Menu items "**Add Nodes**" and "**Add Links**" allow the user to add nodes and links to existing shapes. The new nodes or links can be added only to the active point or line theme. The editing process is similar to that of developing maps of nodes and links, except no new shape file is created. Before importing any map of drainage network from XP-SWMM into ArcView, this network must first be saved in the XPX export format using XP-SWMM tools. Information about network geometry is then read by selecting one of the menu items "**Import Nodes**", "**Import Links**", or "**Import Nodes+Links**". The user is asked to select an XPX export file. The menu item "**Export Nodes+Links**" runs the procedure that converts ArcView shape files into XP-SWMM export file. The user must specify node theme, link theme, and XPX file names.

8 B. Results Display

The **PRESENTATION** Extension consists of two modules: (1) **Results** and (2) **Zones**. It adds a new menu to the menu bar of ArcView 3.1. This menu contains three tools—**Junctions, Conduits,** and **Profiles**.

These tools guide the user to enter the name of the node shape file to be created and the name of the **View** on which the nodes should be displayed. The menu item **Conduits** allows the user to import results of the XP program calculations into ArcView. A **line shape** is created in a manner similar to the **point shape** for nodes. The user is asked to select the XPX export file to

read node coordinates to create a shape file of links, and to select the XP output file to read parameters and simulation results related to nodes and conduits.

Based on the relation of head to ground level (*BGL—Below Ground Level*), nodes are divided into two groups. One group contains nodes in which the calculated head is below ground elevation (displayed in blue color); the second group is composed of flooded nodes, i.e. manholes where head equals ground elevation (displayed in red). Also three sets of conduits with different hydraulic capacities can be analyzed. Each conduit set is classified according to the ratio of calculated discharge to the design discharge according to following three thresholds: (1) flow ratio <0.8; (2) flow ratio 0.8 to 1.1; and (3) flow ratio >1.1.

Before the *Profiles* Tool is executed to display a longitudinal X-section of the network, a set of connected conduits must be selected. This can be done by using the selection tool located on menu item "*profiles*". The procedure reads necessary information from the conduit's attribute table and draws the profile.

The *Zones* module adds *Z* button to the View's tool bar. The *Z* application shows the possible areas, which can be flooded or those locations where runoff might create hazard to the public. The *Z* button should be selected prior to conducting a spatial analysis to investigate the flooding potential of some urban areas. This tools works only when the *DEM* theme and *Z* button are selected simultaneously. The *Junctions* tool will provide the simulated hydraulic results for the runoff collections system from XP-SWMM and it will display the surcharging manholes (in red color). If any surcharging manhole is clicked on, the program will guide the user to select various options to develop the possible extent of the flooded area.

For example, if a user likes to analyze the affects of a surcharging manhole within a distance of 500 feet from its location. The *Z* tool will delineate a circular area having its boundary at a distance of 500 from this particular surcharging manhole. Subsequently, it will execute the spatial analysis on the *DEM* theme. The potentially affected locations are displayed in a color, which is different than the color shown by the adjacent non-flooded areas. The hydrologist is required to overlap this extent of potentially hazardous area on a street theme. Consequently, the exact location and distance of inundated streets and properties can be mapped.

164

VITA

Nazir Ahmed Hawary

Candidate for the Degree of

Doctor of Philosophy

Thesis: COMPUTER TOOLS FOR DEVELOPMENT OF PARAMETERS
FOR AN URBAN WATERSHED USING DESKTOP GIS

Major Field: Civil Engineering

Biographical:

Personal Data: Born in Bahawalnagar, Punjab, Islamic Republic of Pakistan, the son of
Ghulam Hussain Jakhar and Amina Jakhar.

Education: Received the Bachelor of Science degree in Civil Engineering at the University of
Engineering and Technology, Lahore, Pakistan, in 1982; received the Master of Science
degree in Civil Engineering at California State University, Long Beach, California, USA,
in 1993; completed requirements for the Doctor of Philosophy degree in May, 2000.

Professional Experience: GIS Manager, Warner Geomatics, Houston, May 1998 to present;
Manager Information Technology (IT), Al-Swailem Engineering Group (MASCO), Riyadh,
Saudi Arabia, January 1997 to April 1998; Environmental Engineer, Accurate
Environmental and Laboratory Services, Stillwater, Oklahoma, April 1993 to November
1996; Civil Engineer, Global Enviroteq Engineering Services, Long Beach, California,
February 1988 to November 1992; Design/Site Engineer, Mohammed Ali Al-Swailem
Establishment for Contracting, Riyadh, Saudi Arabia, March 1983 to December 1987.