CRWR Online Report 99-1

Floodplain Mapping Using HEC-RAS and ArcView GIS

by

Eric Tate, M.S.E.

Graduate Research Assistant

and

David Maidment, PhD.

Principal Investigator

May 1999

CENTER FOR RESEARCH IN WATER RESOURCES

Bureau of Engineering Research • The University of Texas at Austin
J.J. Pickle Research Campus • Austin, TX 78712-4497

This document is available online via World Wide Web at
http://www.ce.utexas.edu/centers/crwr/reports/online.html

Acknowledgements

First I would like to thank my advisor, Dr. David Maidment, for his support throughout this project. I am also grateful to Dr. Francisco Olivera for his constant assistance throughout this research. The study presented in this report was funded by the Texas Department of Transportation. Their support is gratefully acknowledged.

I would also like to thank the City of Austin Watershed Protection Department, which helped by providing key hydraulic and GIS data for this project. Specifically, I offer thanks to David Walker and Ellen Wadsworth.

I have much appreciated the help and support I received from other members of Dr. Maidment's research team, especially Seann Reed, Brian Adams, Patrice Melancon, Ben Bigelow, and Seth Ahrens.

Finally, I'd like to thank my wife Mercy, for her patience and support over the past two years.

May 7, 1999

Abstract

Floodplain Mapping Using HEC-RAS and ArcView GIS

Eric Christopher Tate, M.S.E.
The University of Texas at Austin, 1999

Supervisor: David Maidment

A significant deficiency of most computer models used for stream floodplain analysis, is that the locations of structures impacted by floodwaters, such as bridges, roads, and buildings, cannot be effectively compared to the floodplain location. This research presents a straightforward approach for processing output of the HEC-RAS hydraulic model, to enable two- and three-dimensional floodplain mapping and analysis in the ArcView geographic information system. The methodology is applied to a reach of Waller Creek, located in Austin, Texas. A planimetric floodplain view is developed using digital orthophotography as a base map. A digital terrain model is synthesized from HEC-RAS cross-sectional coordinate data and a digital elevation model of the study area. The resulting surface model provides a good representation of the general landscape and contains additional detail within the stream channel. Overall, the results of the research indicate that GIS is an effective environment for floodplain mapping and analysis.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

The Texas Department of Transportation is responsible for thousands of drainage control structures along highways throughout the State of Texas. These include facilities such as storm drains, culverts, bridges, and water quality and quantity-control structures. An important design component of these facilities involves hydraulic analyses to determine conveyance capacity. Computer models play a pivotal role in these analyses by aiding in the determination of water surface profiles associated with different flow conditions. Unfortunately, a consistent deficiency of these programs has been their inability to connect the information describing the water profiles with their physical locations on the land surface. Often the computed water surface elevations are manually plotted on paper maps in order to delineate floodplains. Automating this manual plotting would result in significant savings of both time and resources. Geographic information systems (GISs) offer the ideal environment for this type of work. The research reported herein was supported by the Texas Department of Transportation, in order to help improve their hydraulic design capabilities.

This thesis presents a GIS approach for automated floodplain mapping to aid in the design of drainage facilities. The approach establishes a connection between the HEC-RAS hydraulic model and ArcView GIS, allowing for improved visualization and analysis of floodplain data. It also permits GIS to function as an effective planning tool by making hydraulic data easily transferable to floodplain management, flood insurance rate determination, economic impact analysis, and flood warning systems.

1.1 OBJECTIVES

In essence, this research project involves connecting hydraulic modeling and GIS. The primary research objectives are twofold:

1. Develop a procedure to take computed water surface profiles generated from the HEC-RAS hydraulic model and draw a map of the resulting floodplain in ArcView GIS,
2. Synthesize a terrain model from high resolution HEC-RAS data describing the stream channel and comparatively lower resolution digital elevation model data. The result is a continuous landscape surface that contains additional detail within the stream channel.

Attaining these objectives requires translating hydraulic modeling output from HEC-RAS to ArcView. The difficulty stems from the fact that each program uses entirely different coordinate systems to define its data. HEC-RAS is a one-dimensional model, intended for hydraulic analysis of river channels. In HEC-RAS, the stream morphology is represented by a series of cross-sections called river stations. Proceeding from downstream to upstream, the river station numbering increases. The distance between adjacent cross-sections is termed the reach length. Figure 1-1 shows a part of a typical HEC-RAS stream schematic.

The numbers on the figure denote the river station and the polygons indicate river stations containing a bridge or culvert.

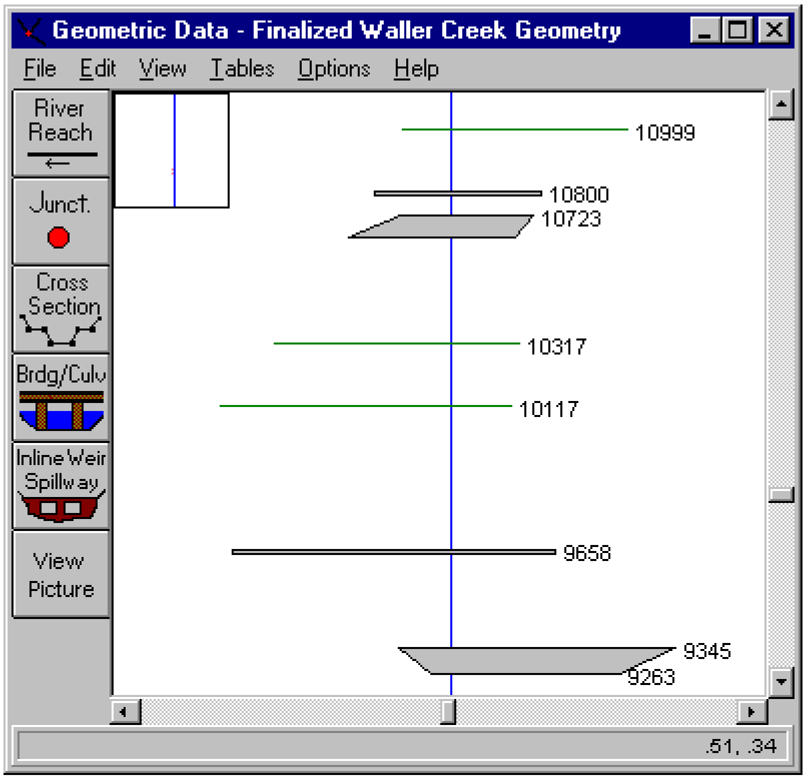


Figure 1-1. HEC-RAS one-dimensional stream schematic

Each cross-section is defined by a series of lateral and elevation coordinates, which are typically obtained from land surveys. The numbering of the lateral coordinates begins at the left end of the cross-section, (looking downstream) and increases until reaching the right end. The value of the starting lateral coordinate is arbitrary, only the distance between points is important. For example, the lateral coordinates numbering for one cross-section may begin at

1000, whereas it may begin at a value of 800 in an adjacent cross-section. The result is that in effect, each cross-section has its own local coordinate system (Figure 1-2).

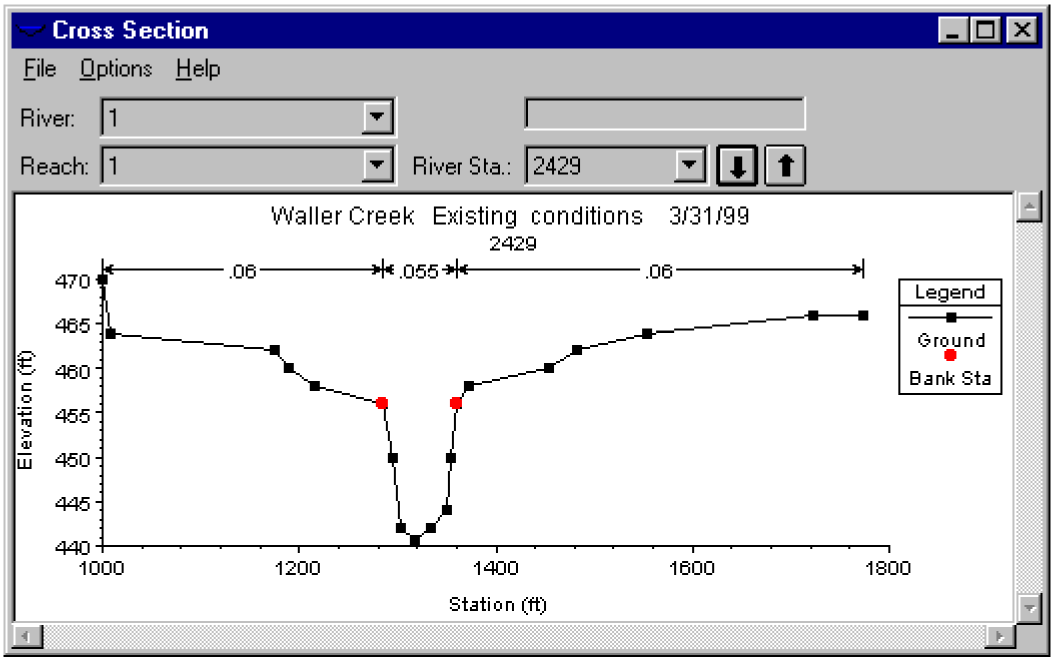


Figure 1-2. HEC-RAS cross-section coordinates.

In the HEC-RAS coordinate system, the coordinate of any given point is based on its river station along a one-dimensional stream centerline, location along the cross-section line, and elevation. In contrast, data in ArcView are attributed with real-world map coordinates; the location of a given point in space is based on its easting (x-coordinate), northing (y-coordinate), and elevation (z-coordinate). Where HEC-RAS represents the stream as a straight-line in model coordinates, ArcView represents it as a curved line in map coordinates (Figure 1-

3). In order to map the hydraulic modeling output in GIS, the differences between the HEC-RAS and ArcView coordinate systems must be resolved. This is the fundamental problem that this research solves.

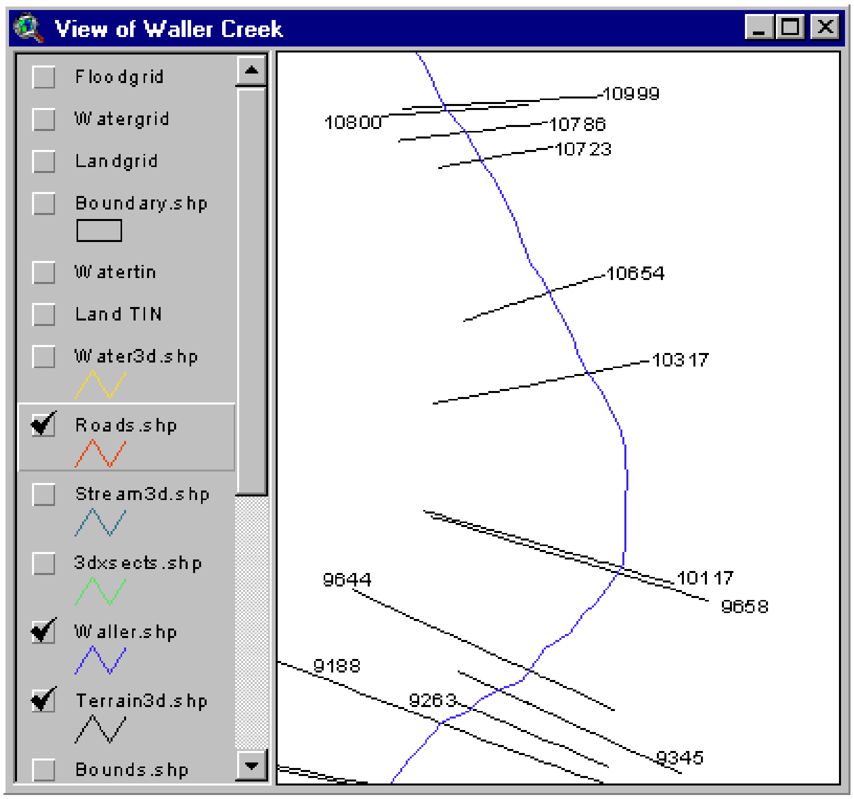


Figure 1-3. GIS two-dimensional stream schematic.

1.2 Study Area

Waller Creek located in Austin, Texas was selected as the study area for this project. Waller Creek is an urban stream that flows south through the University of Texas and downtown Austin (Figure 1-4). Due to its proximity to numerous school buildings, homes, and businesses, the location of Waller Creek's

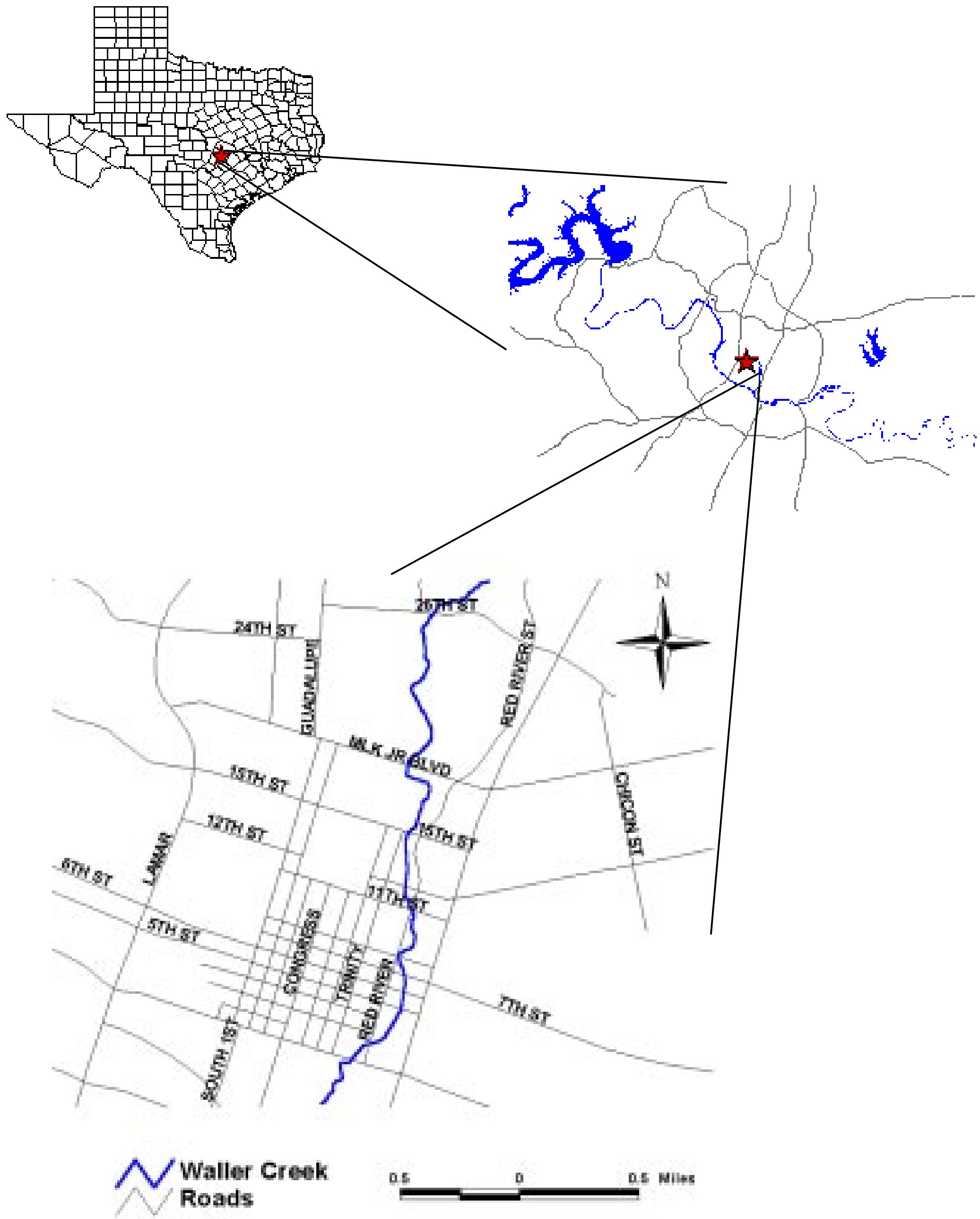


Figure 1-4. Waller Creek study area in Austin, Texas

floodplain is of great interest to city planners, developers, and property owners. As such, the City of Austin has expended a great deal of effort developing detailed HEC-RAS model data describing the stream flow and channel geometry. These model data were made available for use on this research project.

1.3 OUTLINE

The research detailed in this thesis provides an approach for processing output of the HEC-RAS hydraulic model, to enable automated floodplain mapping in ArcView GIS. The thesis is divided into six chapters. Chapter 2 is a review of related literature. Chapter 3 provides a discussion of the research methodology, pertaining to open channel flow, HEC-RAS, GIS, and digital photogrammetry. Chapter 4 details the procedure of application for the processing of HEC-RAS output data, terrain modeling, and floodplain delineation. Chapter 5 includes the results and discussion of the research. Chapter 6 presents conclusions and recommendations.

An important by-product of this research is a GIS program that can be used for floodplain mapping and terrain model development. Tutorial exercises for using both this program and HEC-RAS are given in Appendix A. A data dictionary describing the digital data used in the project is provided as Appendix B. The computer programs developed through the research are included in Appendix C.

Chapter 2: Literature Review

In 1968, Congress established the National Flood Insurance Program (NFIP) in response to rising costs associated with taxpayer funded disaster relief for flood victims (FEMA, 1999a). Through the NFIP, federally backed flood insurance became available in communities that enacted and enforced floodplain management ordinances aimed at reducing flood damage. The need to delineate floodplains for the NFIP spawned a massive national floodplain-mapping project in the 1970s. The floodplain mapping process generally consisted of three steps (Jones, *et al*, 1998):

1. The streamflow associated with the 100-year flood is estimated based on peak flow data or hydrologic modeling
2. The 100-year flood elevation profiles are computed using hydraulic modeling
3. Areas inundated by floodwaters are delineated on paper maps

Until the last few years, GIS applications to floodplain mapping and terrain modeling have been relatively limited. With the rapid advances in GIS in the 1980s, GIS began to be used to represent the flow of water on the land surface. Much of the initial work dealt with the analysis of digital elevation models (DEMs), square grids of regularly spaced elevation data, for hydrologic applications. O'Callaghan and Mark (1984) and Jenson and Domingue (1988) developed methods to fill DEM depressions, and create flow direction and flow accumulation grids using a DEM as the sole input. These advances allowed for the automation of watershed and drainage network delineation. Unfortunately, the use of DEM surfaces is generally not suitable for large-scale terrain representation required for the hydraulic analysis of river channels. Because they cannot vary in spatial resolution, DEMs may poorly define the land surface in areas of complex relief (Carter, 1988). For hydraulic modeling of river channels, the triangular irregular network (TIN) model is preferred (Figure 2-1).

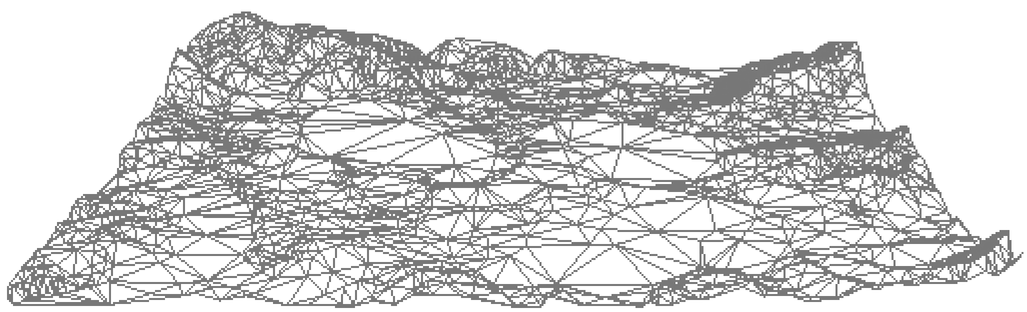


Figure 2-1. TIN surface model.

A TIN is a triangulated mesh constructed on the (x,y,z) locations of a set of data points. The TIN model allows for a dense network of points where the land surface is complex and detailed, such as river channels, and for a lower point density in flat or gently sloping areas (Carter, 1988). Djokic and Maidment (1990) used TINs to model storm drainage in an urban setting. The TIN surface was found to be effective for the determination of parameters for design flow calculations.

Beavers (1994) performed some of the first work connecting hydraulic modeling of river channels and GIS. A GIS-based tool, named ARC/HEC2, was developed to assist hydrologists in floodplain analysis. ARC/HEC2 consists of a compilation of Arc/Info Macro Language scripts (AMLs) and C programs used to pre- and post-process terrain and floodplain data used for the HEC-2 hydraulic model. The programs extract terrain information from contour coverages, insert user-supplied information (e.g., Manning's roughness values, channel contraction/expansion coefficients), and format the information so that it can be imported into HEC-2.

Following the execution of HEC-2, ARC/HEC2 can retrieve the output (in the form of water elevations at each cross-section) and create an Arc/Info TIN coverage of the floodplain (HEC-2 post-processing). This process generates a floodplain that can be used in conjunction with other Arc/Info coverages. ARC/HEC2 requires that a terrain surface be generated so that accurate cross-section profiles are provided to HEC-2. These terrain surfaces, in the format of TINs or grids, are created within Arc/Info based on contour lines, survey data, or other means of establishing terrain relief. The accuracy of the HEC-2 floodplain calculations depends heavily upon the accuracy of the surface representation.

In the years following the release of ARC/HEC2, many hydrologists have switched from using HEC-2 to the Windows based HEC-RAS hydraulic model. HEC-RAS differs from the HEC-2 model in that the capability to import and export GIS data was included in the program. Version 2 of HEC-RAS gives the user the option to import and utilize three-dimensional river reach and cross-sectional data from a general-purpose data exchange file. Terrain information stored in a GIS can be translated to conform to the HEC-RAS data exchange file format (Figure 2-2).

```
BEGIN STREAM NETWORK:
  ENDPOINT: 476132.66, 65291.86, 155.28, 1
  ENDPOINT: 478144.53, 64296,61, 123.72, 2

  REACH:
    STREAM ID: Waller Creek
    REACH ID: Hemphill Branch
    FROM POINT: 1
    TO POINT: 2
    CENTERLINE:
      476132.66, 65291.86, 155.28, 23.13
      476196.08, 65196.61, 154.47
        lines omitted
      478144.53, 64296.61, 123.72, 22.41
    END:
END STREAM NETWORK:
```

Figure 2-2. HEC-RAS data exchange file.

In 1997, Tom Evans of HEC improved on the Beavers work with the release of a set of AMLs that serve both as a pre- and post-processor for HEC-RAS. The preprocessing AMLs create a data exchange file consisting of stream geometry descriptions extracted from a triangular irregular network (TIN) model of the land surface. In HEC-RAS, the user is required to provide additional data such as Manning's n, contraction and expansion coefficients, geometric descriptions of any hydraulic structures (e.g., bridges, culverts) in the cross-sections, and bank stations and reach lengths (if they are not included in the exchange file). After running the model, HEC-RAS can export the output data into the same digital exchange file format. A TIN of the water surface can then be created from the exchange file using the AML post-processing macros. The process is illustrated in Figure 2-3.

Pre-Processing:

TIN Data Extracted

Cross-sections

Stream centerline

Streambank lines

Flow lines

Post-Processing:

GIS Themes Created

Cross-section cut lines

Bounding polygons

Water surface TINs

Floodplain polygons

Figure 2-3. Evans' pre- and post-processing for HEC-RAS

In function, Evans' work is quite similar to the work completed by Beavers and Djokic, with a primary difference being that Evans' programs work with the HEC-RAS model rather than HEC-2. The AML code provides a set of utilities that allows preparation of GIS data for HEC-RAS input and formatting of model output for GIS display.

In 1998, the Environmental Systems Research Institute (ESRI) had a limited release of an ArcView extension called AVRAS. Whereas, the previous

GIS software operated in the Arc/Info environment, AVRAS was designed to use ArcView as the pre- and post-processing environment for hydraulic modeling in HEC-RAS. AVRAS was initially created by translating Evans' AML code into Avenue (ArcView's scripting language), but several additional utilities were later added to make the program more user friendly. In 1999, AVRAS was released as a commercial product by Dodson & Associates, Inc., under the trade name of GIS Stream Pro.

Other computer software packages that connect GIS and river hydraulic modeling are also available. The Danish Hydraulic Institute developed MIKE 11 GIS, a system that connects the one-dimensional MIKE 11 hydraulic model and ArcView GIS. Using MIKE 11 GIS, floodplain topography can be extracted from a DEM and imported into MIKE 11. After running the model, MIKE 11 GIS can prepare three types of floodplain maps for display and analysis in ArcView: depth/area inundation, duration, and comparison/impact (DHI, 1999). In addition, MIKE 11 GIS can produce output graphs of water level time series data, terrain and water level profiles, and flood zone statistics.

In cooperation with several governmental agencies, the Environmental Modeling Research Laboratory developed the Surfacewater Modeling System (SMS). SMS is a graphical user interface designed to serve as a pre- and post-processor for hydraulic modeling of river channels. SMS is compatible with the one-dimensional WSPRO backwater calculation model and the two-dimensional

FESWMS, RMA2, RMA4, and HIVEL2D hydrodynamic hydraulic models (EMRL, 1999). As a preprocessor for one-dimensional hydraulic modeling, SMS can be used to extract stream cross-sections from a TIN surface. For two-dimensional modeling, SMS is used to develop finite element meshes. The modeling output can be exported from the SMS interface to either Arc/Info or ArcView for display and analysis.

The approaches described thus far are all rather sophisticated methods for linking hydraulic modeling and GIS. Each shares a common theme: hydraulic modeling parameters are extracted from a terrain model, and imported into a hydraulic model. After executing the model, the output is processed for display and analysis in a GIS. Each of the techniques requires a DEM or TIN digital terrain model as the source of input cross-section descriptions. Unfortunately, terrain models with a density of stream channel points sufficient for hydraulic modeling are generally not available. However, a wealth of detailed cross-sectional data has been developed for hydraulic modeling, typically from field surveys and topographic maps. The problem is that these high-resolution data are often stored in hydraulic model coordinate systems, a format incompatible with GIS. Currently, there are no available methods with which to integrate hydraulic model data with GIS, if an input terrain model is not the original source of the cross-section descriptions used for the hydraulic modeling. The research presented in the following sections offers an approach to resolve this deficiency.

Chapter 3: Methodology

This chapter introduces concepts and associated terminology that are often referred to from this point forward. The discussion includes open channel flow theory, the HEC-RAS hydraulic model, GIS, and aerial photogrammetry.

3.1 OPEN CHANNEL FLOW

Open channel flow is defined as the flow of a free surface fluid within a defined channel. Typical examples are flow in natural streams, constructed drainage canals, and storm sewers. The development of effective floodplain management plans requires that engineers understand the hydraulics of open channel flow, which depend upon the flow classification, flow and conveyance, and the energy equation.

3.1.1 Flow Classification

Open channel flow is classified based on time, space, and flow regime:

3.1.1.1 Time

Steady flow describes conditions in which depth and velocity at a specific channel location do not change with time. In contrast, *unsteady flow* refers to flow conditions that change with time at a given location.

3.1.1.2 Space

The term *uniform flow* denotes fluid flow in which depth and velocity are constant with distance. Uniform flow conditions require the channel to be straight, with constant cross-sectional geometry, and a water surface that is parallel to the base of the channel. In *varied flow*, water depth and velocity change with distance along the channel.

3.1.1.3 Flow regime

The dimensionless Froude number is used to classify flow type:

$$Fr = \frac{V}{\sqrt{gy}} \qquad \text{(Eq. 3-1)}$$

Where: Fr = Froude number

V = mean fluid velocity (m/s)

g = gravitational acceleration (m/s^2)

y = water depth (m)

Subcritical flow occurs when the Froude number is less than 1; when the Froude number exceeds 1, s*upercritical flow* conditions exist. *Critical flow, critical depth, and critical velocity* are defined at the point where the total energy head is a minimum. At critical conditions, the Froude number equals one.

In order to determine the water surface elevations at different cross-sections in a channel, the flow rate and velocity must be known or calculated. For river hydraulic analysis, a steady, gradually varied flow assumption is often used for both subcritical and supercritical flow regimes. Steady, gradually varied flow applies to flow in which changes in flow depth and velocity occur gradually over a considerable length of channel (Prasuhn, 1992).

3.1.2 Flow and Conveyance

The continuity equation for steady flow states that flow must be conserved between adjacent cross-sections:

$$Q = V_1A_1 = V_2A_2 \quad \text{(Eq. 3-2)}$$

Where: Q = flow rate/discharge (m^3/s)

V_n = average velocity at cross-section n (m/s)

A_n = area at cross-section n (m^2)

For open channel flow, the momentum equation is used in the form of the Manning equation:

$$Q = K\sqrt{S_f} \quad \text{(Eq. 3-3)}$$

$$K = \frac{1}{n}AR^{2/3} \quad \text{(Eq. 3-4)}$$

Where: R = hydraulic radius (m)

n = Manning roughness coefficient

K = conveyance ($m^{5/3}$)

$\overline{S_f}$ = average friction slope between adjacent cross-sections

The hydraulic radius is calculated by dividing the cross-sectional area by the wetted perimeter. The Manning coefficient is a parameter that measures the effect of channel roughness on the flow of water through it. The values for the Manning coefficient vary based on channel terrain, and are published in most hydraulic engineering books. Some typical values of the Manning coefficient are shown in Table 3-1 (Prasuhn, 1992).

Table 3-1. Manning Coefficients for Open Channels

Channel Type and Description	**Value**
Smooth Concrete	0.012 - 0.013
Unfinished Concrete	0.013 - 0.016
Earthen, smooth, no weeds	0.02
Firm gravel	0.02
Earthen, some stones & weeds	0.025
Earthen, unmaintained, winding natural streams	0.035
Mountain streams	0.04 - 0.05

For determination of conveyance, the cross-section is subdivided based on Manning coefficient (Figure 3-1) into the left overbank, main channel, and right overbank. The conveyance for each subdivision is then calculated (Eq. 3-4). The total conveyance for the cross-section is obtained by summing the individual subdivision conveyances (HEC, 1997)

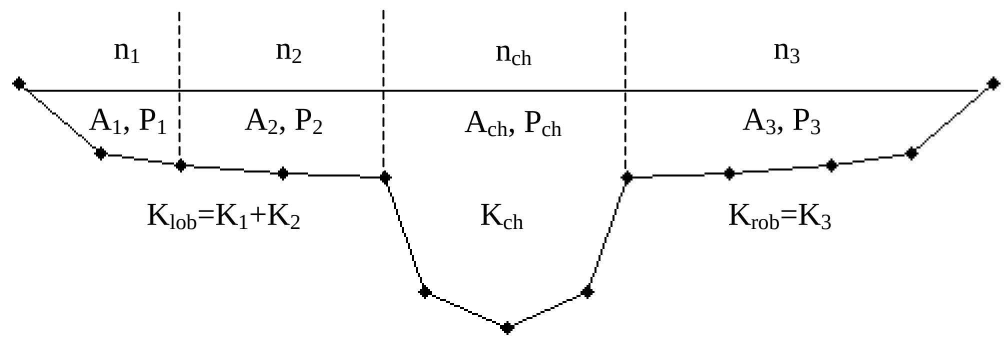


Figure 3-1. Conveyance calculation parameters

For prismatic river channels, the flow rate is typically known based on either hydrologic modeling or flood frequency analyses. With the flow and conveyance known, the average friction slope between two adjacent cross-sections can be calculated:

$$\overline{S_f} = \left(\frac{Q_1 + Q_2}{K_1 + K_2} \right)^2 \qquad \text{(Eq. 3-5)}$$

When the Manning equation is applied to uniform flow, the average friction slope is replaced by channel bed slope (S_o).

3.1.3 Energy Equation

For open channel flow, the total energy per unit weight (energy head) has three components: elevation head, pressure head, and velocity head (Figure 3-2):

$$H = Z + Y + \frac{\alpha V^2}{2g} \qquad \text{(Eq. 3-6)}$$

Where: H = energy head (m)

Z = channel bed elevation above a datum (m)

Y = pressure head/water depth (m)

α = velocity weighting coefficient

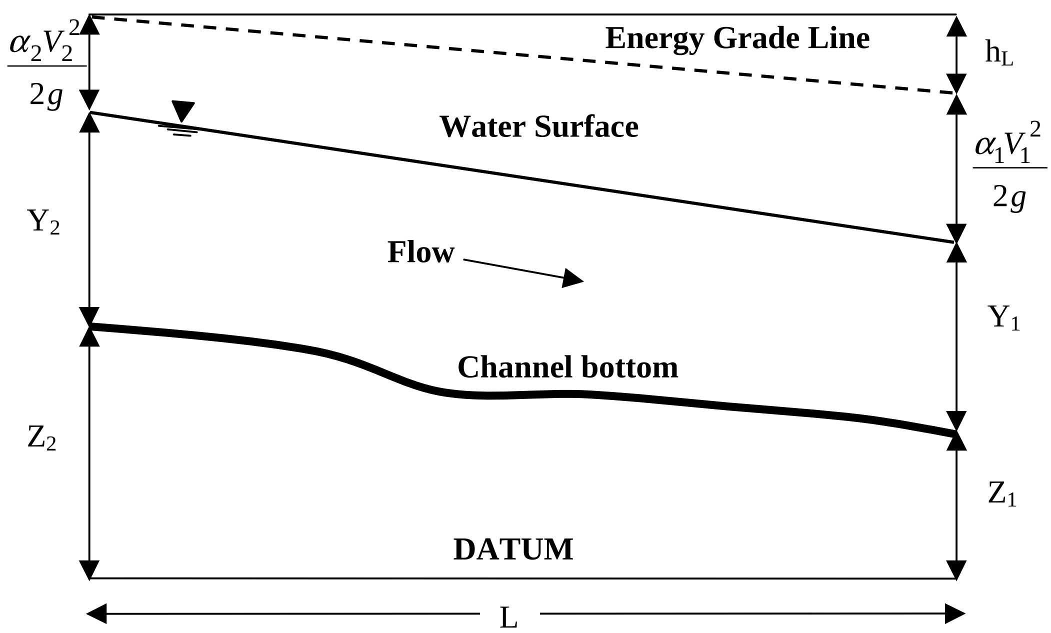


Figure 3-2. Energy equation parameters for gradually varied flow.

For a given water surface elevation, the mean velocity head is obtained by computing a flow weighted velocity head over the cross-section. Based on channel geometry, channel expansions and contractions, and flow obstructions such as bridges and piers, the flow velocity can vary from one end of the cross-section to the other. The velocity-weighting coefficient (α) accounts for the error in using the average velocity instead of a velocity distribution. The magnitude of the velocity coefficient depends on the type of channel. Some typical values are shown in Table 3-2 (CES, 1998).

Table 3-2. Velocity-Weighting Coefficients

	Value of α		
Channel Type	**Minimum**	**Average**	**Maximum**
Regular channels, flumes, spillways	1.10	1.15	1.20
Natural Streams	1.15	1.30	1.50
Rivers under ice cover	1.20	1.50	2.00
River valleys, overflooded	1.50	1.75	2.00

Based on the energy equation parameters shown in Figure 3-2, the water surface elevation is the sum of Y and Z. The change in energy head between adjacent cross-sections is equal to the head loss:

$$H_2 = H_1 + h_L \qquad \text{(Eq. 3-7)}$$

Where: H_1 = energy head at cross-section 1 (m)

H_2 = energy head at cross-section 2 (m)

h_L = energy head loss (m)

The head loss between the two cross-sections is the sum of friction head loss and flow contraction/expansion head loss. Friction losses result from shear stresses between the water and channel bed and banks:

$$h_f = L\overline{S_f} \qquad \text{(Eq. 3-8)}$$

Where: h_f = friction head loss (m)

L = distance between adjacent cross-sections (m)

Contraction/expansion head losses can occur due to the formation of eddies wherever there is a contraction or expansion of the channel (Prasuhn, 1992):

$$h_o = C\left|\frac{\alpha_2 V_2^2}{2g} - \frac{\alpha_1 V_1^2}{2g}\right| \qquad \text{(Eq. 3-9)}$$

Where: h_o = contraction or expansion head loss (m)

C = contraction or expansion coefficient

Typical values for contraction and expansion coefficients for subcritical flow are shown in Table 3-3 (HEC, 1997).

Table 3-3. Subcritical Flow Contraction and Expansion Coefficients

Type of Channel Transition	**Contraction**	**Expansion**
None	0.0	0.0
Gradual	0.1	0.3
Typical bridge sections	0.3	0.5
Abrupt	0.6	0.8

3.2 HEC-RAS

HEC-RAS is a hydraulic model developed by the Hydrologic Engineering Center (HEC) of the U.S. Army Corps of Engineers. In 1964, HEC released the HEC-2 computer model to aid hydraulic engineers in stream channel analysis and floodplain determination. HEC-2 quickly became the standard stream hydraulic analysis program, and its capabilities were expanded in the ensuing years to provide for, among other things, bridge, weir, and culvert analyses. Although HEC-2 was originally developed for mainframe computer use, it can currently operate on personal computers (in DOS mode) and workstations (Beavers, 1994).

Due to the increased use of Windows-based personal computing software, in the early 1990's HEC released a Windows-compatible counterpart to HEC-2 called the River Analysis System (RAS). HEC-RAS has a graphical user interface programmed in Visual Basic, to which are attached flow computation algorithms programmed in FORTRAN, many of which were derived from the HEC-2 model. HEC-RAS is a one-dimensional steady flow model, intended for computation of water surface profile computations. Modules for unsteady flow simulation and movable-boundary sediment transport calculations are scheduled to be included. The system is capable of modeling subcritical, supercritical, and mixed-flow regimes for streams consisting of a full network of channels, a dendritic system, or a single river reach. The model results are typically applied in floodplain management and flood insurance studies in order to evaluate the effects of floodway encroachments (HEC, 1997).

3.2.1 HEC-RAS Parameters

HEC-RAS uses a number of input parameters for hydraulic analysis of the stream channel geometry and water flow. These parameters are used to establish a series of cross-sections along the stream. In each cross-section, the locations of the stream banks are identified and used to divide into segments of left floodway, main channel, and right floodway (Figure 3-3). HEC-RAS subdivides the cross-sections in this manner, because of differences in hydraulic parameters. For example, the wetted perimeter in the floodway is much higher than in the main channel. Thus, friction forces between the water and channel bed have a greater influence in flow resistance in the floodway, leading to lower values of the

Manning coefficient. As a result, the flow velocity and conveyance are substantially higher in the main channel than in the floodway.

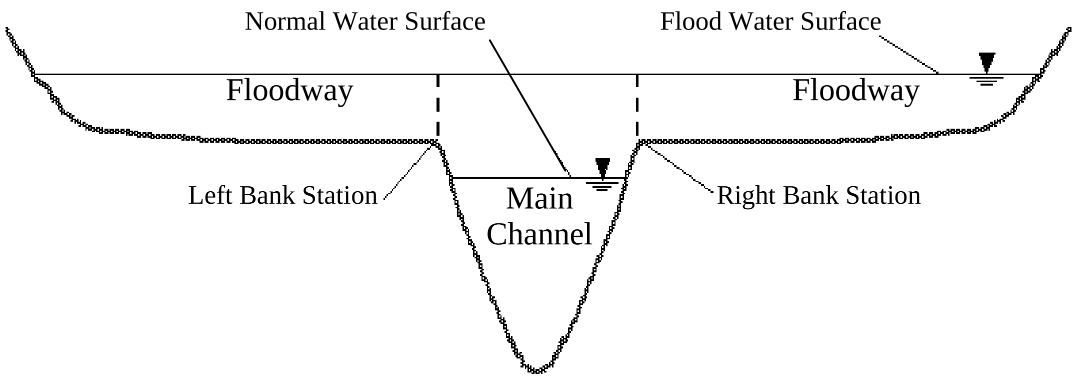


Figure 3-3. Stream cross-section schematic.

At each cross-section, HEC-RAS uses several input parameters to describe shape, elevation, and relative location along the stream (Figure 3-4):

- River station (cross-section) number
- Lateral and elevation coordinates for each (dry, unflooded) terrain point
- Left and right bank station locations
- Reach lengths between the left floodway, stream centerline, and right floodway of adjacent cross-sections (The three reach lengths represent the average flow path through each segment of the cross-section pair. As such, the three reach lengths between adjacent cross-sections may differ in magnitude due to bends in the stream.)
- Manning's roughness coefficients

- Channel contraction and expansion coefficients
- Geometric description of any hydraulic structures, such as bridges, culverts, and weirs.

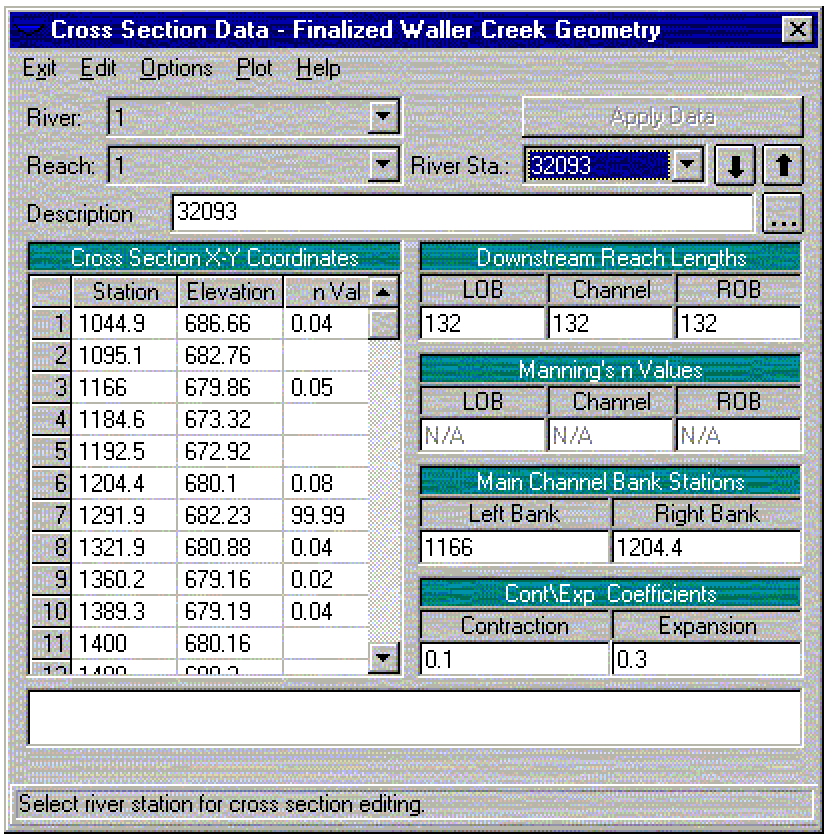


Figure 3-4. HEC-RAS cross-section input parameters.

HEC-RAS assumes that the energy head is constant across the cross-section and the velocity vector is perpendicular to the cross-section (i.e., quasi one-dimensional flow on a two-dimensional domain). As such, care should be taken that the flow through each selected cross-section meets these criteria. After defining the stream geometry, flow values for each reach within the river system are entered. The channel geometric description and flow rate values are the primary model inputs for the hydraulic computations.

3.2.2 Water Surface Profile Computation

For steady, gradually varied flow, the primary procedure for computing water surface profiles between cross-sections is called the direct step method (HEC-RAS also supports the momentum, WSPRO bridge, and Yarnell methods). The basic computational procedure is based on the iterative solution of the energy equation. Given the flow and water surface elevation at one cross-section, the goal of the standard step method is to compute the water surface elevation at the adjacent cross-section. For subcritical flow, the computations begin at the downstream boundary and proceed upstream; for supercritical flow, the computations begin at the upstream boundary and proceed downstream. At the boundary, the flow and water surface elevation must be known. The procedure is summarized below (assuming subcritical flow).

1. Assume a water surface elevation at cross-section 1.
2. Determine the area, hydraulic radius, and velocity (Eq. 3-2) of cross-section 1 (Figure 3-2) based on the cross-section profile.
3. Compute the associated conveyance (Eq. 3-4) and velocity head values.

4. Calculate friction slope (Eq. 3-5), friction loss (Eq. 3-8), and contraction/expansion loss (Eq. 3-9).
5. Solve the energy equation (Eq. 3-6) for the water surface elevation at the adjacent cross-section.
6. Compare the computed water surface elevation with the value assumed in step 1.
7. Repeat steps 1 through 6 until the assumed and computed water surface elevations are within a predetermined tolerance.

The discussion thus far in this chapter has centered on hydraulic analysis with HEC-RAS. To automate the floodplain mapping process, GIS is required to assign map coordinates to the output data. In the following subsection, some basic GIS concepts are introduced.

3.3 GEOGRAPHIC INFORMATION SYSTEMS

GISs are defined as computer systems capable of assembling, storing, manipulating, and displaying geographically referenced information (USGS, 1998). Originally developed as a tool for cartographers, GIS has recently gained widespread use in engineering design and analysis, especially in the fields of water quality, hydrology, and hydraulics. GIS provides a setting in which to overlay data layers and perform spatial queries, and thus create new spatial data. The results can be digitally mapped and tabulated, facilitating efficient analysis and decision making. Structurally, GIS consists of a computer environment that joins graphical elements (points, lines, polygons) with associated tabular attribute descriptions. This characteristic sets GIS apart from both computer-aided design software (geographic representation) and databases (tabular descriptive data). For example, in a GIS view of a river network, the graphical elements represent the location and shape of the rivers, whereas the attributes might describe the stream name, length, and flow rate. This one-to-one relationship between each feature and its associated attributes makes the GIS environment unique. In order to provide a conceptual framework, it is necessary to first define some basic GIS constructs.

3.3.1 Data Models

Geographic elements in a GIS are typically described by one of three data models: vector, raster, or triangular irregular network. Each of these is described below.

3.3.1.1 Vector

Vector objects include three types of elements: points, lines, and polygons (Figure 3-5). A point is defined by a single set of Cartesian coordinates [easting(x),northing(y)]. A line is defined by a string of points in which the beginning and end points are called nodes, and intermediate points are called vertices (Smith, 1995). A straight line consists of two nodes and no vertices whereas a curved line consists of two nodes and a varying number of vertices. Three or more lines that connect to form an enclosed area define a polygon.

Vector feature representation is typically used for linear feature modeling (roads, lakes, etc.), cartographic base maps, and time-varying process modeling. In Figure 3-5, points represent flow gages, lines represent streams, and polygons represent watershed boundaries.

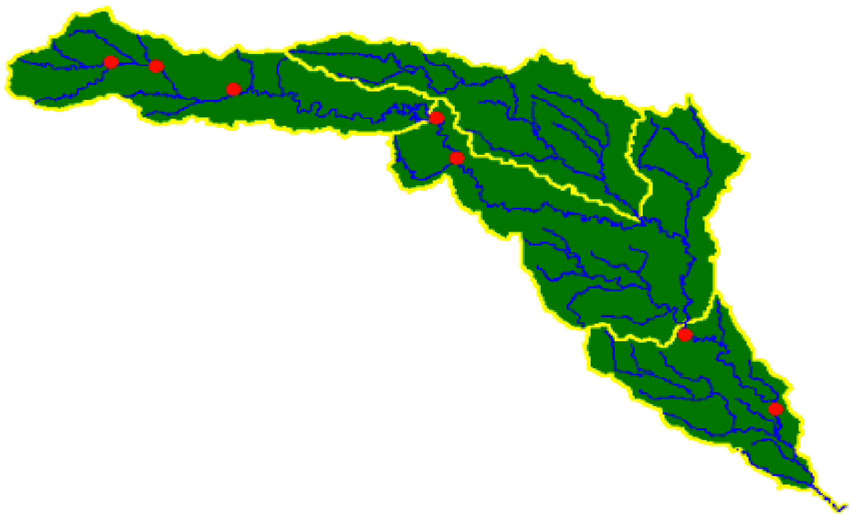


Figure 3-5. Vector feature representation.

3.3.1.2 Raster

The raster data structure consists of a rectangular mesh of points joined with lines, creating a grid of uniformly sized square cells (Figure 3-6). Each cell is assigned a numerical value that defines the condition of any desired spatially varied quantity (Smith, 1995). Grids are the basis of analysis in raster GIS, and are typically used for steady-state spatial modeling and two-dimensional surface representation. A land surface representation in the raster domain is called a digital elevation model (DEM).

78	72	69	71	58	49
74	67	56	49	46	50
69	53	44	37	38	48
64	58	55	22	31	24
68	61	47	21	16	19
74	53	34	12	11	12

Figure 3-6. Raster DEM

3.3.1.3 Triangular Irregular Network (TIN)

A TIN is a triangulated mesh constructed on the (x,y) locations of a set of data points. To form the TIN, a perimeter around the data points is first established, called the convex hull. To connect the interior points, triangles are created with all internal angles as nearly equiangular as possible. This procedure is called Delaunay triangulation. By including the dimension of height (z) for each triangle vertex, the triangles can be raised and tilted to form a plane. The collection of all such triangular planes forms a representation of the land surface terrain in a considerable degree of detail (Figure 3-7). The TIN triangles are small where the land surface is complex and detailed, such as river channels, and larger in flat or gently sloping areas.

Additional elevation data, such as spot elevations at summits and depressions and break lines, can also be included in the TIN model. Break lines represent significant terrain features like a streams or roads that are indicative of a

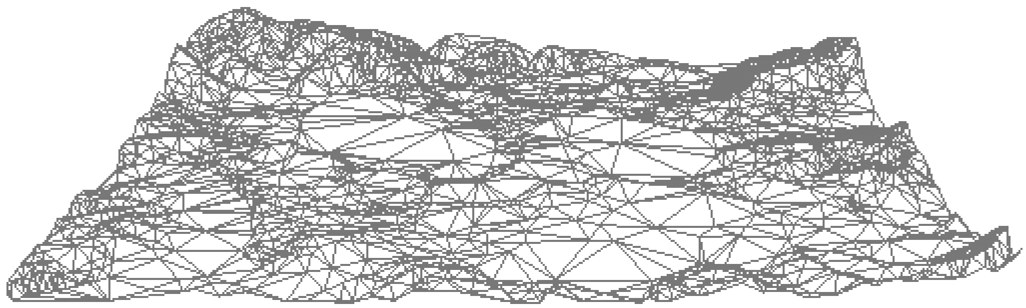


Figure 3-7. TIN land surface representation

change in slope; TIN triangles do not cross break lines. In three-dimensional surface representation and modeling, the TIN is generally the preferred GIS data model. Some reasons for the TIN model preference include the following:

- requires a much smaller number of points than does a grid in order to represent the surface terrain with equal accuracy
- can be readily adapted to variable complexity of terrain
- supports point, line, and polygon features
- original input data is maintained in the model and honored in analysis

3.3.2 ArcView GIS

The ArcView GIS software package, developed by the Environmental Systems Research Institute, was used as the computer development environment for this research. In the past several years, ArcView has emerged as the industry leader in desktop GIS software. All activities within ArcView are organized with a project, which may consist of a number of views, tables, charts, layouts and scripts (Maidment, 1998). Files created in ArcView are called projects and are denoted by an ".apr" file extension. Vector data files in ArcView are called shapefiles. The functions of ArcView include: displaying shapefiles in a view, viewing and editing the related attribute tables of this view, plotting charts to display spatial information, and creating layouts of the view and related tables and charts. Specialized ArcView software, called extensions, are required to manipulate and analyze raster and TIN data. The ArcView Spatial Analyst extension is designed for creating, querying, mapping, and analyzing raster data, whereas the 3D Analyst extension is intended for creating, analyzing, and visualizing TINs and three-dimensional vector data (ESRI, 1999). GIS data in ArcView can be manipulated using Avenue, a customization and development programming language embedded in the software package. Numerous Avenue scripts were written for this research and they are provided in Appendix C.

3.4 AERIAL PHOTOGRAMMETRY

Photogrammetry is the science, art, and technology of obtaining reliable measurements from maps, digital elevation models, and other derived products from photographs (Lillesand and Kiefer, 1994). One of the most common uses of photogrammetry is the analysis of aerial photography to extract ground elevations for the production of topographic maps. However, photogrammetric techniques are also used to produce digital terrain models and digital orthophotographs, which were used in this research. Photogrammetric techniques relevant to the development of digital orthophotographs and digital terrain models are described in the following subsections.

3.4.1 Image Acquisition

The first step in the production of digital orthophotographs and terrain models is capturing aerial photographs of the land surface. To acquire the images, a plane travels over a study area in a straight flight line, and photographs are taken such that every ground point appears in at least two successive photographs. The resulting photos are called a stereo pair, and the area of common coverage is called overlap. If the study area of interest requires more than one flight line for complete coverage, additional flight lines are flown parallel to the original line. Overlap between adjacent flight lines is called sidelap, and is used to prevent gaps in coverage. Standard overlap is 60% forward and 15% side (Figure 3-8).

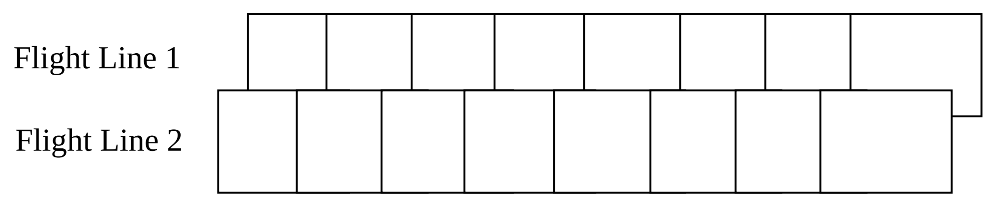


Figure 3-8. Image acquisition using overlapping photographs

Typically, aerial photographs designed for digital orthophotograph production are taken with a camera with a 6-inch focal length lens, at an altitude of 15,000 feet; this generates photographs with a scale of 1:30,000. The film diapositives (transparencies) are later scanned with a precision image scanner to create a digital raster image file. Each raster cell, or pixel, is assigned a gray scale value that corresponds to the average intensity of the ground area covered by the pixel (Lillesand and Kiefer, 1994).

3.4.2 Digital Terrain Model Development

The digital terrain model can come from an existing source, but it is often developed from the aerial photography. The aerial photos are taken using a stereoscopic camera, with which two pictures of a particular area are simultaneously taken from slightly different angles. The overlapping area of the two resulting photos is called a stereo pair. Using digital image processing software or an analog instrument called a stereoplotter, the stereo pair can be viewed as a single image with the appearance of depth or relief. Ground control points are established based on ground surveys or aerial triangulation, and are viewed in the stereoplotter in conjunction with the stereo pair. In this setting, the image coordinates of (x,y,z) points in the stereo pair are digitized. These points, in conjunction with the control points, comprise the data points used to construct a digital model of the terrain.

3.4.3 Digital Orthophotographs

Digital orthophotos are scale-correct aerial photographs. Conventional aerial photographs have limited use in GIS because they are not true to scale. When you look at the center of an aerial photograph, the view is the same as if looking straight down from the aircraft. However, the view of the ground toward the edges of the photograph is from an angle. This is called a central perspective projection; scale is true at the very center of the aerial photograph, but not elsewhere. In order to create a scale correct photograph that can be accurately measured, an orthographic projection is necessary, in which the view is straight down over every point in the photograph.

The procedure used to create digital orthophotos, called ortho-rectification, requires aerial photographs and a TIN digital terrain model as inputs. The TIN surface is used to orthogonally rectify the scanned raster image file. By combining the TIN and raster image, each image pixel is attributed with a known location and intensity value. In the rectification process, the intensity value for each pixel is re-sampled using a space resection equation, removing image displacements caused by central perspective projection, camera tilt, and terrain relief. The individual photographs are then clipped and seamlessly joined together over the entire study area. The result is a digital image that combines the image characteristics of a photograph with the geometric qualities of a map--a true to scale photographic map. The resulting ground/pixel resolutions can be as fine as 1 meter. However, the accuracy of the final digital orthophoto depends in large part on the point density of the TIN terrain model. Typically, digital orthophotographs are distributed as images covering one-fourth of a U.S. Geological Survey (USGS) 7.5-minute topographic map. As such, these images are referred to as Digital Ortho Quarter Quads (DOQQs) (Figure 3-9).

Figure 3-9. DOQQ (2.5-meter resolution).

Digital orthophotos have many uses in the GIS environment. Some orthophoto benefits include the following:

- May be used as a GIS base map for a variety of uses, including urban and regional planning, revision of digital line graphs and topographic maps, creation of soil maps, and drainage studies
- More cost effective and display more surface features than conventional maps
- Easily available over the internet from USGS or the Texas Natural Resources Information System (TNRIS).

3.5 DIGITAL DATA SOURCES

Although a great deal of GIS data were created as the result of this research, there were relatively few input data. These consist of the following:

- HEC-RAS flow and geometry files provided by the City of Austin
- Ten- and thirty-meter resolution DEMs from TNRIS
- One-meter resolution digital orthophotography purchased from TNRIS
- Vector shapefile of Austin roads provided by the City of Austin

A data dictionary describing these input data and any other GIS data created during the course of this research is provided in Appendix B.

The input data sources possess varying datums, map projections, and units. ArcView allows vector, raster, and TIN data to be viewed together provided that they have a common datum, map projection and coordinate system. As such, a standard map projection is required. Because specialized software is required to reproject images, the projection of the digital orthophoto was chosen as the standard map projection.

DOQQs available from TNRIS are cast in the Universal Transverse Mercator (UTM) zone 14 projection, based on the 1983 North American Datum (NAD83). In the UTM coordinate system, the earth is divided into 60 zones, each 6° of longitude in width; the UTM projection is applied to each zone using its centerline as the principle meridian (Strange, 1998). For UTM zone 14, the principal meridian is 99 degrees west longitude.

Vector data can be reprojected in ArcView using the Avenue script called **Projector.ave**. This script was applied to convert the roads coverage from the State Plane Texas Central, NAD27 projection to UTM zone 14, NAD83. Unfortunately, ArcView does not have the capability to reproject raster data. However, ESRI's Arc/Info GIS software can be used to reproject a DEM. The Arc/Info command used for grid projection is named **project**, and has the following syntax:

Arc: *project grid <input filename> <output filename> <projection file> <interpolation method> <output cell size>*

The projection file is a text file that specifies the input and output map projection parameters. Like the digital orthophoto, the TNRIS DEM also has a UTM projection, but it is based on NAD27. To project a grid using UTM to UTM, while only changing the datum, the projection file takes the following form:

input
projection
zone
datum
units
spheroid
parameters
output
projection
zone
datum
units
spheroid
parameters
end

As shown, the projection file for both the input and output data require description of the following values:

- the type of map projection
- UTM zone
- horizontal datum
- units of the coordinate system
- spheroid used to approximate Earth's geometry
- parameters which specifies the spatial projection parameters

For the projection of the TNRIS DEM from UTM NAD27 to UTM NAD83, the following projection file is used:

input
projection UTM
zone 14
datum NAD27
units meters
spheroid Clarke 1866
parameter
output
projection UTM
zone 14
datum NAD83
units meters
spheroid GRS80
parameter
end

Chapter 4: Procedure of Application

This chapter details the procedure developed to process HEC-RAS output for terrain modeling and floodplain delineation in the ArcView GIS. Application of the methodology reduces the analysis time and improves accuracy by integrating spatial stream geometry with hydraulic analysis. The approach is based on assigning map coordinates to stream cross-sections and computed water surface profile data stored in HEC-RAS model coordinates. The procedure consists of five primary steps:

1. Data import from HEC-RAS
2. Stream centerline definition
3. Cross-section georeferencing
4. Terrain modeling
5. Floodplain mapping

These steps are discussed in greater detail in the following subsections. HEC-RAS and GIS data for the Waller Creek study area are used to illustrate the procedures.

4.1 DATA IMPORT FROM HEC-RAS

In order to move into the GIS environment, the HEC-RAS output data must be extracted. Because the approach presented herein assumes that input terrain model is not the source of the cross-section descriptions, the GIS data export option in HEC-RAS is not employed. Instead an output report using the **File/Generate Report** menu option from the HEC-RAS main project window is used. In the resulting report generator window, "Plan Data" and "Geometric Data" should be checked as the general input data, "Reach Lengths" under the summary column, and "Cross Section Table" under the specific tables output option (Figure 4-1).

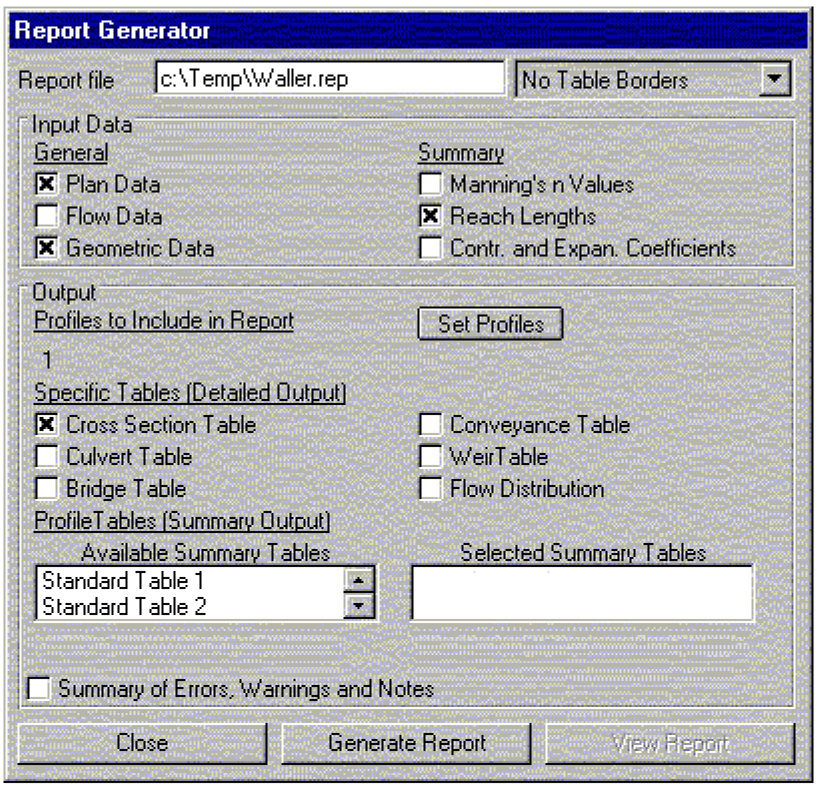


Figure 4-1. HEC-RAS output report generator.

It is important that only one profile is selected for the output report and the modeled stream has only one branch (the approach cannot currently operate on multiple flow profiles or stream networks). If model includes more than one flow profile, the specific profile used for the output report can be selected using the "Set Profiles" button. After clicking the "Generate Report" button, the output report is created. The report is a text file that contains input data describing cross-sectional geometries and stream flow rates, and output data describing computed water surface profiles (Figure 4-2).

```
CROSS SECTION          RIVER: 1
REACH: 1                  RS: 32093

INPUT
Description: 32093
Station Elevation Data    num=      12
     Sta    Elev     Sta    Elev     Sta    Elev     Sta    El
  1044.9  686.66  1095.1  682.76    1166  679.86  1184.6  673.
  1204.4   680.1  1291.9  682.23  1321.9  680.88  1360.2  679.
    1400  680.16    1400   690.2

Manning's n Values        num=       7
     Sta   n Val     Sta   n Val     Sta   n Val     Sta   n V
  1044.9     .04    1166     .05  1204.4     .08  1291.9   99.
  1360.2     .02  1389.3     .04

Bank Sta: Left   Right    Lengths: Left Channel   Right     Co
          1166  1204.4              132     132     132

CROSS SECTION OUTPUT    Profile #PF 4

  E.G. Elev (ft)              681.60   Element
  Vel Head (ft)                 0.95   Wt. n-Val.
  W.S. Elev (ft)              680.65   Reach Len. (ft)
  lines omitted....
  Crit W.S. (ft)              680.65   Flow Area (sq ft)
```

Figure 4-2. Example HEC-RAS output report.

An Avenue script named **RAS-Read.ave** was developed to read the RAS output text file and write key stream parameters to ArcView. The .ave suffix denotes that the script is written in the Avenue programming language. All Avenue scripts developed for this project are provided in Appendix C. The parameters processed at each cross-section include the following:

- River station (cross-section) number
- Coordinates of the stream center, located at the point of minimum channel elevation
- Floodplain boundary locations, as measured from the stream center
- Bank station locations, as measured from the stream center
- Reach lengths
- Water surface elevation

Figure 4-3 shows an HEC-RAS cross-section plot, in which these parameters can be seen. In the figure legend, EG stands for energy grade line, and WS refers to the water surface.

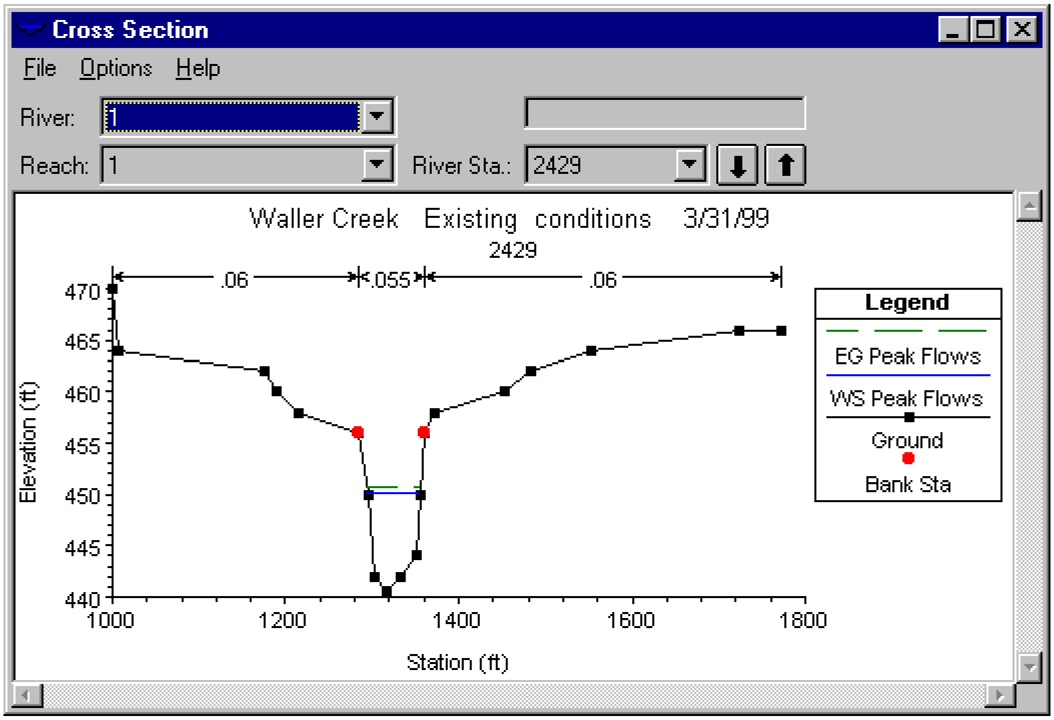


Figure 4-3. HEC-RAS cross-section plot.

For each cross-section, the lateral and elevation coordinates of all points (black squares in Figure 4-3) are read and stored in an ArcView global variable. Using these points, the coordinates of the point possessing the minimum channel elevation are determined. If there are multiple points possessing the same minimum channel elevation, the lateral coordinate of the channel center is calculated by averaging the lateral coordinates of all points possessing the same minimum elevation.

In order to determine the lateral coordinates of the floodplain boundaries, the computed water surface elevation is used. The cross-section coordinates are read from the left end of the cross-section to the right end. When the computed water surface elevation falls between the elevation coordinates of two adjacent points, the coordinates of these bounding points are noted. The lateral coordinates of the floodplain boundaries are subsequently calculated:

$$x3 = [(z3-z1)*(x2-x1)/(z2-z1)] + x1 \qquad \text{(Eq. 4-1)}$$

Where:
x1 = left bounding point lateral coordinate
z1 = left bounding point elevation coordinate
x2 = right bounding point lateral coordinate
z2 = right bounding point elevation coordinate
x3 = lateral coordinate of floodplain boundary
z3 = computed water surface elevation

With the lateral coordinates of the floodplain boundaries known, the lateral distance from the stream center is calculated and stored in an ArcView table. In the same manner, the lateral distance from the bank stations (red circles in Figure 4-3) to the stream center is also calculated and stored in the table along with the elevation coordinates. The remaining cross-section parameters written to the ArcView table, are the river station number, any text description of the cross-section, reach lengths, and the computed water surface elevation.

One potential problem is the question of units. Typically, hydraulic modeling is performed using U.S. units (feet). However, many GIS data sets, including DEMs and digital orthophotographs use SI units (meters). As such, the **RAS-Read.ave** script assumes the output report file has coordinates measure in feet, and prompts the user for the units to be used in ArcView (feet or meters). If "meters" is selected, all coordinates written to the ArcView table are converted to meters by multiplying by a factor 0.3048.

Figure 4-4 shows an example of the ArcView cross-section parameter table. Descriptions of the data stored in each of the columns are provided in Table 4-1.

HEC-RAS Cross-Section Parameter Table

Station	Descriptio	Type	FloodElev	LFloodX	LBankX	LBankZ	ChannelX	ChannelZ	RBankX	RBankZ	RFloodX
28490	55 1/2 St	Culver			11.2	197.8		195.8	4.1	197.7	
28465			198.7	75.1	11.2	197.8	1117.1	195.8	4.1	197.7	56.3
28428			198.5	42.2	7.0	197.6	1153.7	195.8	3.7	197.7	53.6
28308			197.9	29.0	7.0	197.2	1211.6	195.4	3.7	197.3	45.1
28118			197.6	66.1	10.0	197.1	1219.5	195.3	8.5	197.1	36.1
28092			197.6	71.9	9.4	197.0	1235.0	195.1	9.1	197.1	42.7
28066	55th Stre	Culver			9.4	197.0		195.1	9.1	197.1	
28041			197.6	69.5	9.4	197.0	1247.2	195.1	9.1	197.1	40.6
28001			197.4	61.3	3.4	196.8	1277.7	194.9	7.6	196.9	34.4
27901			197.1	55.5	3.4	196.5	1309.1	194.6	7.6	196.6	32.9
27798			197.2	74.5	8.7	196.2	1316.1	194.2	6.5	196.4	40.8
27775			197.2	72.3	7.0	196.2	1330.1	194.2	8.3	196.4	42.3
27752	Nelray St	Culver			7.0	196.2		194.2	8.3	196.4	
27729			197.2	77.0	7.0	196.2	1339.3	194.2	8.3	196.4	45.3

Figure 4-4. ArcView cross-section parameter table.

The purpose of the data import step is to transform HEC-RAS output from text file format into a tabular format readable by ArcView. However, the cross-section coordinates are still tied to the HEC-RAS coordinate system. In order to map the floodplain, the cross-sections must be assigned map coordinates. This requires associating the HEC-RAS stream cross-sections with a geographically referenced digital representation of the stream.

Table 4-1. Cross-Section Parameter Table Data Descriptions

Column Title	Data Description
Station	River station number
Description	Short text description of the cross-section location (if included in the HEC-RAS geometry file)
Type	Hydraulic structure (bridge or culvert) at the cross-section
FloodElev	Computed water surface elevation
LFloodX	Lateral distance from the stream center to the left floodplain boundary
LBankX	Lateral distance from the stream center to the left bank station
LBankZ	Left bank station elevation
ChannelY	Cumulative reach length, beginning at the upstream end
ChannelZ	Channel center elevation
RightBankX	Lateral distance from the stream center to the right bank station
RightBankZ	Right bank station elevation
RightFloodX	Lateral distance from the stream center to the right floodplain boundary

4.2 STREAM CENTERLINE DEFINITION

After importing the HEC-RAS output data into ArcView, it is necessary to link the HEC-RAS stream representation to the digital representation of the stream in ArcView. There are four primary ways to obtain a digital representation of the stream centerline:

4.2.1 Reach Files

Reach files are a series of national hydrologic databases that uniquely identify and interconnect the stream segments or "reaches" that comprise the nation's surface water drainage system. The databases include such information as unique reach codes for each stream segment, upstream/downstream relationships, and stream names (where possible). The latest release, reach file 3 (RF3), consists of attributed 1:100,000 scale digital line graph hydrography. The data can be downloaded from the U.S. Environmental Protection Agency (EPA) BASINS website at http://www.epa.gov/OST/BASINS/gisdata.html.

4.2.2 DEM-Based Delineation

Using the capabilities of the ArcView Spatial Analyst extension, a vector stream network can be derived using a DEM as the sole input. An example of a tool for developing such a vector stream network is CRWR Pre-Pro. CRWR Pre-Pro is a system of ArcView scripts and associated controls developed by the Center for Research in Water Resources (CRWR), to extract hydrologic, topographic, and topologic information from digital spatial data of a hydrologic system, and prepare an input file for hydrologic modeling. CRWR-PrePro can be obtained over the internet from the University of Texas at http://www.ce.utexas.edu/prof/olivera/prepro/prepro.htm.

4.2.3 Digitize the Stream

Using either orthophotography or a digital raster graphic (DRG) as base map, the stream centerline can be digitized using tools in ArcView. DRGs are digitized and geographically referenced topographic maps. DRGs and DOQQs for the state of Texas can be obtained from the TNRIS website at http://www.tnris.state.tx.us/digital.htm.

4.2.4 Land Surveys

Data representing the stream centerline may be available from land surveys. If this data is tied to a global coordinate system, it can be used as a vector representation of the stream.

Of the four methods listed above, the first three were evaluated for this project. Attempts to use RF3 were discontinued early on because its stream centerline representation was often inconsistent with the representation derived from a 30-meter DEM and digitizing from a DOQQ (Figure 4-5). DEM-based delineation and digitizing from a DOQQ base map were both found to be acceptable methods for stream delineation. However, the accuracy of the stream delineation by each method depends on the resolution of the DEM and the orthophotography, respectively. For this research, 1-meter resolution orthophotography of the Austin East 7.5-minute quadrangle was obtained from TNRIS. In ArcView, the digital imagery was used as a base map upon which to digitize Waller Creek. A data dictionary for the digital orthophotography is provided in Appendix B.

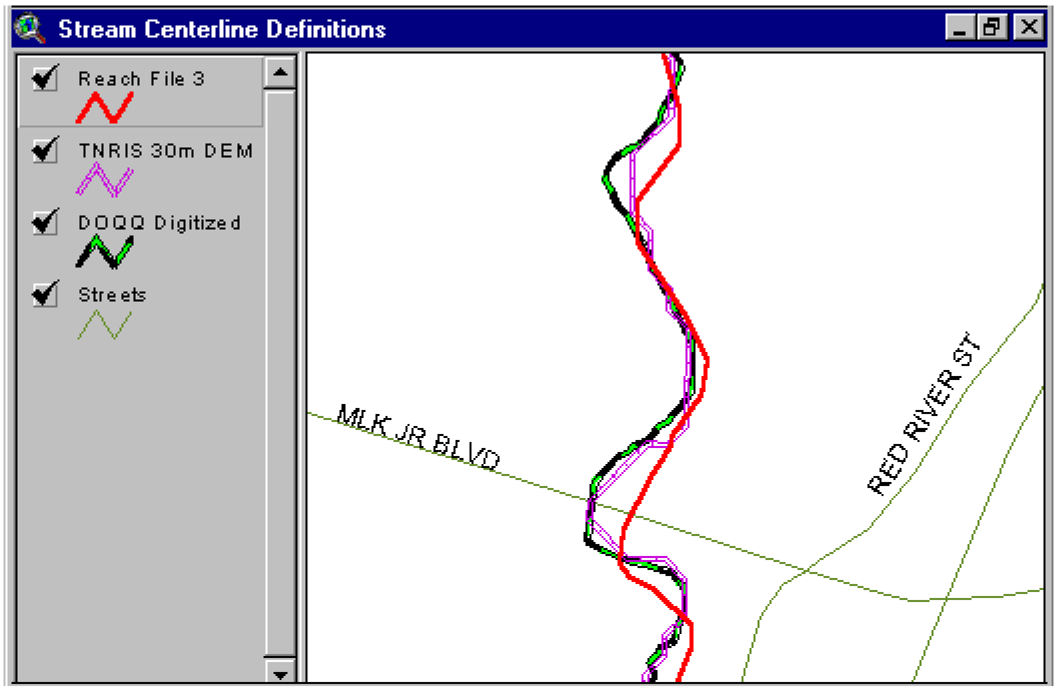


Figure 4-5. Stream centerline representations.

4.3 CROSS-SECTION GEOREFERENCING

The first step in geographically referencing the cross-sections is to compare the definitions of the RAS stream and their digital counterpart. It is possible for example, that the digital stream centerline is defined to a point farther upstream than the RAS stream, or vice versa. Hence, it is necessary to define the upstream and downstream boundaries of the RAS stream on the digital stream. To this end, the Avenue script **Addpnt.ave** was developed, with which the upstream and downstream boundaries can be established with a click of the mouse. Intermediate stream definition points corresponding to known RAS cross-sections such as bridges or culverts can also be defined. This process takes advantage of HEC-RAS models in which descriptions of cross-section locations are included. These descriptions are brought into ArcView during the data import step, and can be used to help define intermediate stream definition points.

In order to use the script **Addpnt.ave**, a stream centerline shapefile is required. When the user clicks on a point, the script determines the nearest point along the stream centerline and snaps the point onto the digital stream. The output of the script is a point shapefile. Often, the definition points are more easily pinpointed by comparison to the location of an existing structure (e.g., road, bridge, etc). As such, a theme of roads can be used in addition to the DOQQ to assist in the point selection process. As the number of defined points increases, so does the accuracy of the resulting floodplain. In Figure 4-6, seven stream definition points are shown. Of these, two represent the upstream and downstream boundaries, and five are intermediate stream definition points.

Once the stream definition points are established, the next step is to add the cross-sections between them. To do this, two attributes must be known for

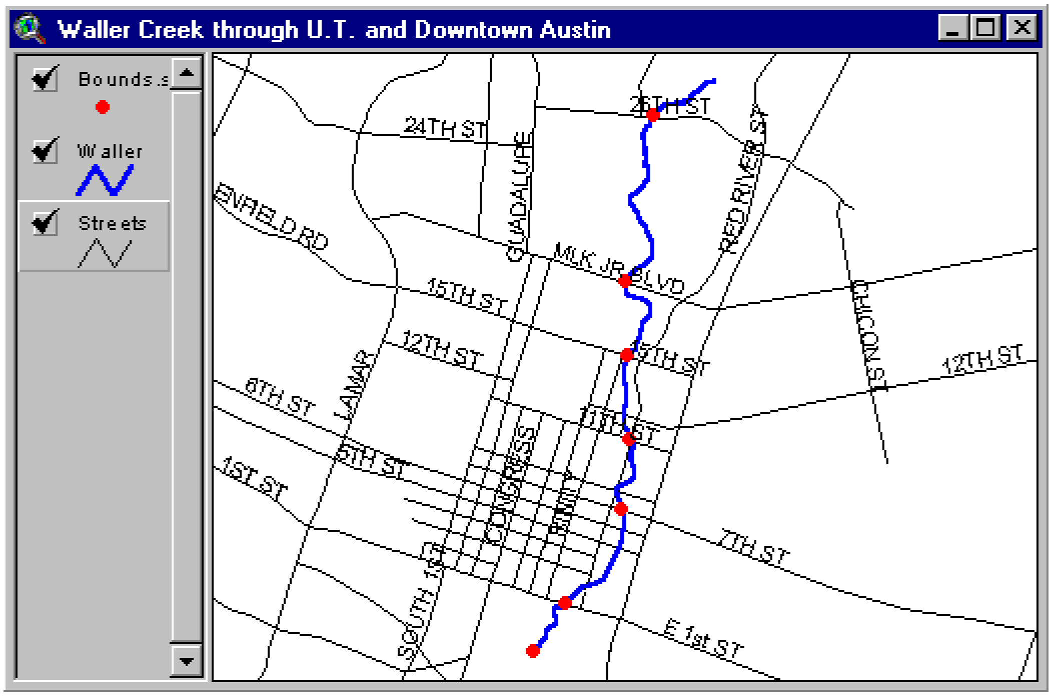


Figure 4-6. Stream Definition Points

each cross-section: location along the stream, and orientation; the Avenue script **Terrain3D.ave** aids in the determination of these attributes. As input, the script requires the stream centerline theme and the stream definition point theme. In order to determine the location of each cross-section along the stream centerline, a one-to-one relationship is established between each stream definition point and its associated cross-section record in the cross-section parameter table (Figure 4-7).

The HEC-RAS stream centerline definition is based on land surveys and topographic maps, whereas the basis for the GIS stream centerline is aerial

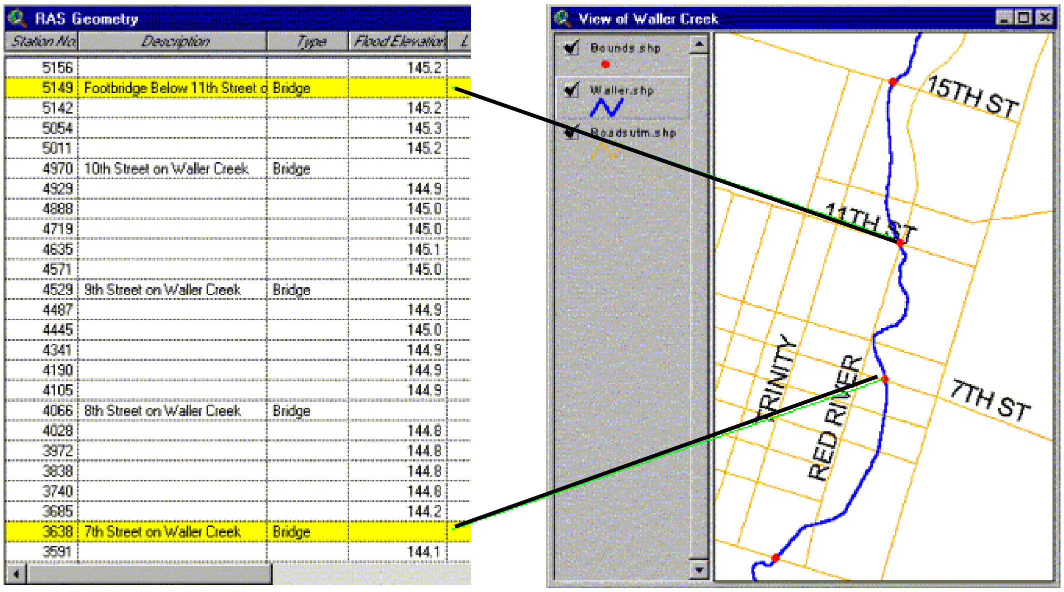


Figure 4-7. One-to-one relationship between table records and definition points.

photogrammetry and digitizing. Because of these differences, the length along the stream between any two points likely varies to some degree. To evaluate the difference in lengths, between each set of adjacent definition points, **Terrain3d.ave** calculates the ratio of the length of the RAS-modeled stream to that of the digital stream. The length of the RAS stream segment is determined as the difference in cumulative reach lengths stored in the cross-section parameter table. To determine the length of the digital stream segment, the position of each definition point is calculated as a percentage along the total digital line length. The percentage difference is then multiplied by the total line length:

$$SL = (P1-P2)*(L*100) \quad \text{(Eq. 4-2)}$$

Where: SL = length of digital centerline segment

P1 = point position (%) of stream definition point 1

P2 = point position (5) of stream definition point 2

L = total length of digital centerline

If the RAS reach lengths and digital stream representation are accurate, and the streams definition points were precisely placed, then the ratio of segment lengths should be nearly one. If the length of the RAS stream segment exceeds that of the digital stream, the reach lengths between cross-sections are compressed by the ratio. However, if the RAS stream length is less than the digital stream length, the reach lengths between cross-sections are expanded by the same ratio. In this manner, the proportionality of RAS reach lengths is preserved. In the Waller Creek data sets, the ratio generally ranged between 0.98 and 1.02.

At this point, the cross-section locations along the stream centerline are known, but not their orientation. The HEC-RAS model requires cross-sections to be defined such that they are perpendicular to flow lines in the floodways and main channel. Within relatively straight portions of the channel this means straight-line cross-sections. Near bends in the stream, the cross-sections sometimes are doglegged in the floodways to ensure perpendicularity to flow. Unfortunately, information concerning the orientation of each cross-section is typically indicated on survey maps, but is not stored by HEC-RAS. Therefore, an assumption of orientation is required in order to map the cross-sections in GIS. In this case, all cross-sections are assumed to occur in straight lines, perpendicular to the stream centerline.

In order to determine the direction of a perpendicular line to the stream centerline, the bearing of the centerline must first be known. The centerline direction can be determined by calculating the bearing between two points located immediately upstream and downstream of the cross-section location along the stream:

$$m1 = (y1-y2)/(x1-x2) \tag{Eq. 4-3}$$

Where: $m1$ = bearing of stream centerline

$x1,y1$ = easting and northing of upstream point

$x2,y2$ = easting and northing of downstream point

The bearing of the cross-section ($m2$) is subsequently calculated as the negative inverse of the centerline bearing:

$$m2 = (-1)/m1 \tag{Eq. 4-4}$$

However, assuming absolute perpendicularity could result in intersecting cross-sections at bends in the stream, a condition that is unrealistic. As such, the user is prompted to assign the distance between the cross-section location and the points used for determining centerline bearing, a number termed the "distance value." The distance value is input as a percentage of the total stream length (Figure 4-8).

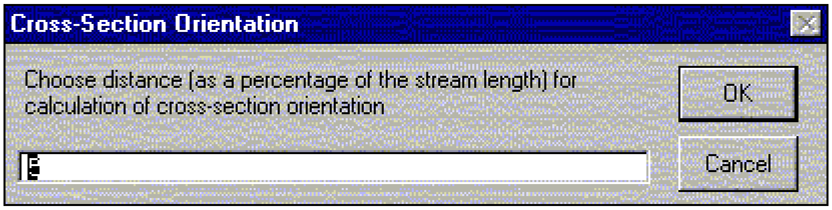


Figure 4-8. User prompt for assignment of cross-section orientation

With the cross-section locations and orientations known, a line representing each cross-section can be mapped. If the resulting cross-sections intersect near bends in the stream (Figure 4-9), **Terrain3D.ave** can be re-run using a higher distance value. As the distance value increases, so does the departure from true perpendicular cross-sections (Figure 4-10).

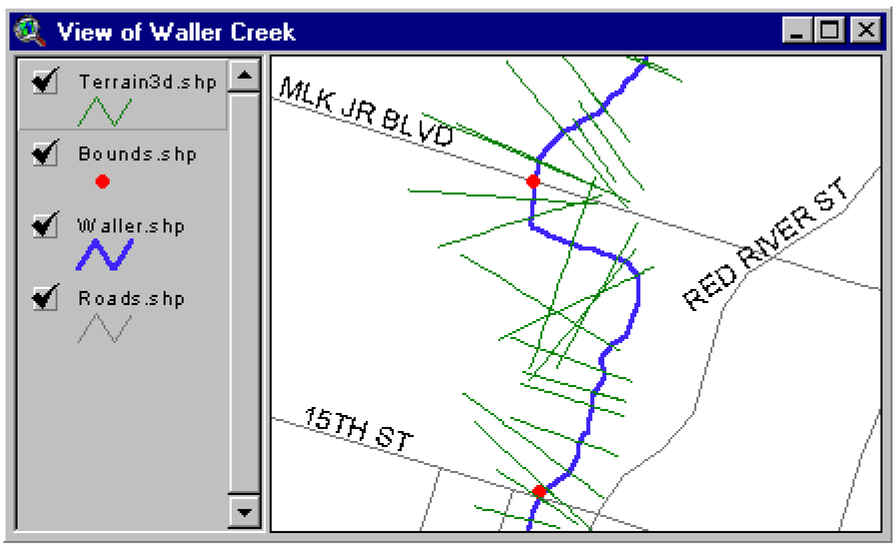


Figure 4-9. Cross-section mapping using a distance value of 1.

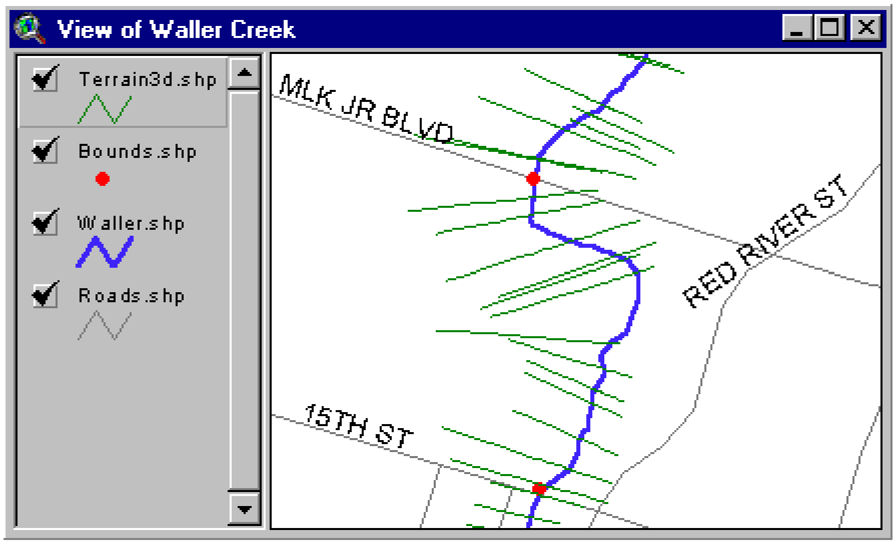


Figure 4-10. Cross-section mapping using a distance value of 5.

For this project, a distance value of 5 was used to map the cross-sections, which corresponds to a distance of approximately 200 meters. Each cross-section is attributed with river station number, cross-section length, and the location of the stream center and bank stations.

4.4 FORMATION OF AN INTEGRATED TERRAIN MODEL

In order to produce a floodplain map, accurate topographic information is required. In GIS, the TIN model is the best data model for large-scale terrain representation. The result of the cross-section georeferencing described in the previous subsection is that every vertex of every cross-section is assigned three-dimensional map coordinates. The easting and northing coordinates come from the mapping process, and the elevation coordinate from the global variable created in the data import step. Using these three-dimensional cross-section points, a TIN model of the stream channel and floodplain can be constructed. Unfortunately, a TIN created solely from this vector data would include the stream channel and floodplain, but not the surrounding landscape. To best represent the terrain, the TIN should include areas both inside and outside the floodplain in order to give a sense of context. The grid (DEM) is the standard data model used for small-scale representation of the general land surface. So in order to create a comprehensive TIN, a method to integrate relatively low-resolution digital elevation model (DEM) data with comparatively higher resolution vector floodplain data is required. By combining the vector and raster data to form a TIN, the intended result is a continuous three-dimensional landscape surface that contains additional detail in stream channels. This approach was employed to form the TIN terrain model (Figure 4-11). Application of the approach consists of the following steps:

1. Clip the 30m DEM to a manageable size. The TNRIS DEM has the same areal extent as a USGS 7.5-minute quadrangle map, an area of approximately 17,000 square kilometers. Using the script **Gridclip.ave,** the DEM can be clipped to the extent of any given polygon theme. The script requires a DEM and clipping polygon theme as inputs. The output is a clipped DEM.
2. Perform a raster to vector conversion on the clipped DEM to create a point shapefile of terrain elevations. This is performed using the script **R2Vpoint.ave.** In the conversion process, the DEM cells are converted to a point shapefile, with each point attributed with the elevation of the cell.

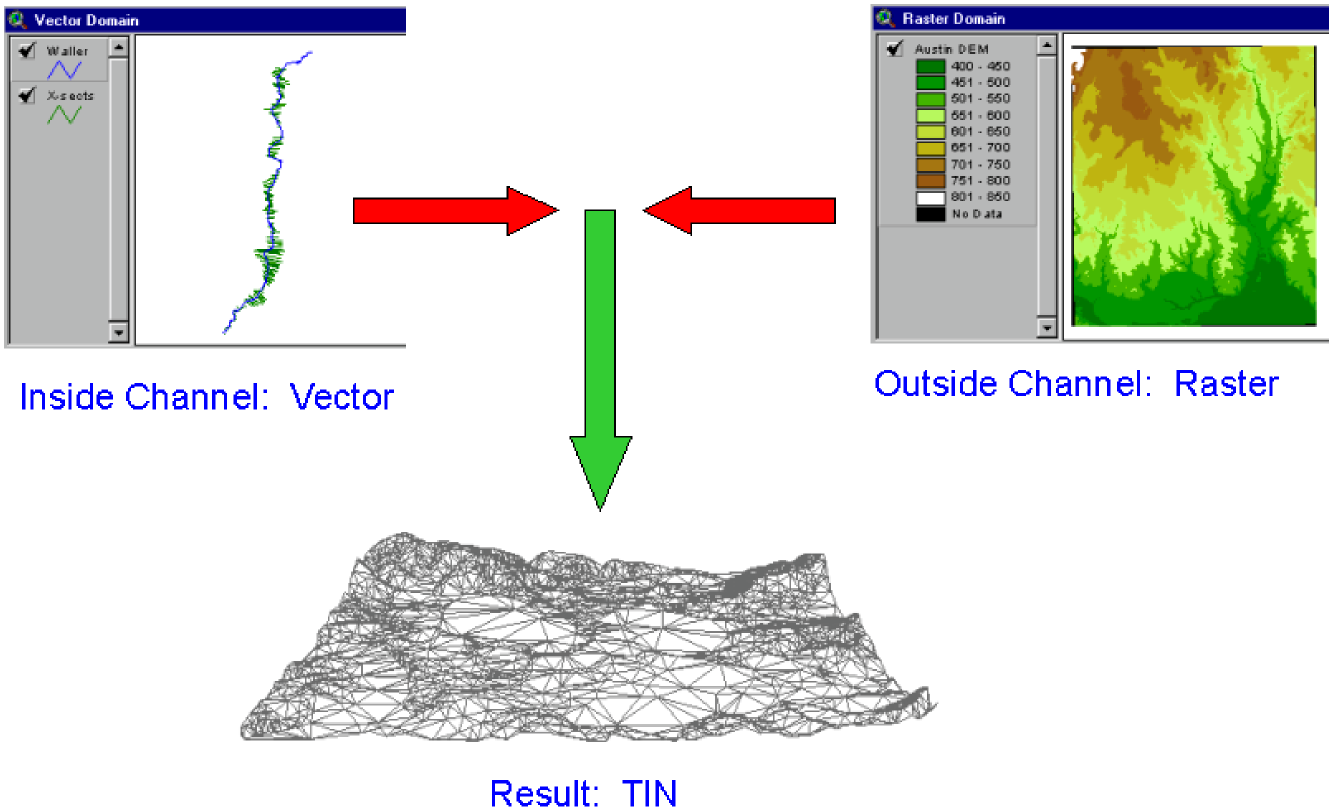


Figure 4-11. Three-dimensional TIN terrain modeling approach.

3. Construct a bounding polygon from the cross-section endpoints. The script **Boundary.ave** carries out this task. The script uses the cross-section theme as input to create the bounding polygon theme.
4. Eliminate any points from the DEM point theme that fall within the bounding polygon. First, the DEM point theme is intersected with the bounding polygon theme using the **Theme/Select by Theme** menu option in the ArcView view window. The selected points can then be deleted from the point shapefile using the **Table/Start Editing** menu option in the ArcView table window, followed by **Edit/Delete Records**.
5. From the cross-section theme, extract the center point and bank stations at each cross-section in order to create a three-dimensional line theme consisting of the stream centerline and bank lines. The Avenue script **Banklines.ave** performs this task.
6. Create a TIN using three input data sources: the DEM point theme, cross-section line theme, and the centerline and banks line theme. TIN nodes are formed from the DEM points and the vertices of the cross-section lines. The stream centerline and bank lines are enforced in the TIN as breaklines. The TIN is created using the **Surface/Create TIN from Features** menu item from the ArcView view window.

The TIN inputs are illustrated in Figure 4-12.

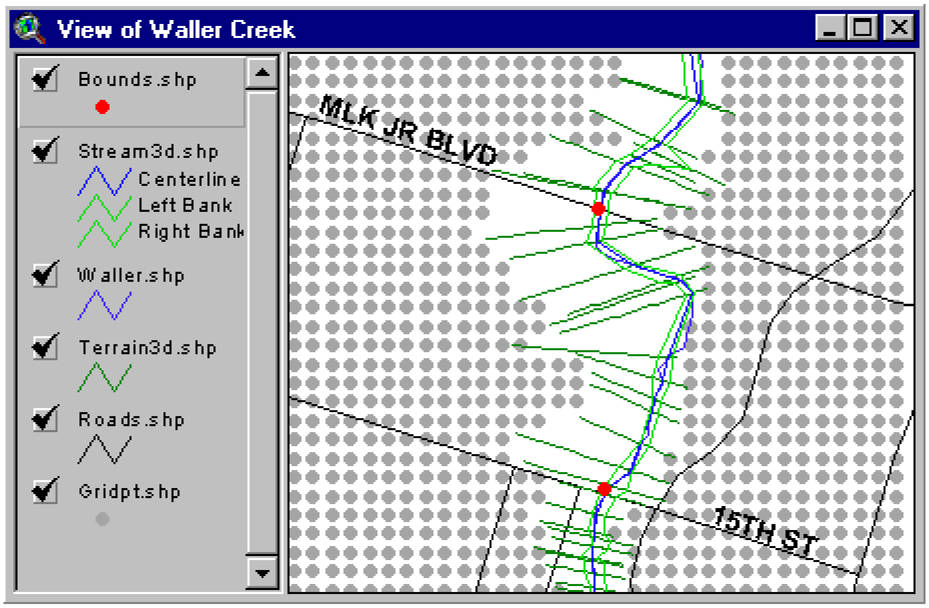


Figure 4-12. Terrain TIN data inputs.

A three-dimensional terrain TIN was subsequently constructed such that the stream channel data supersede the terrain data within the area for which they are defined and the DEM point data prevails elsewhere. The initial concern with this procedure was that it would not produce a smooth zone of transition between the vector and raster data points. A cursory examination of the resulting TIN (Figure 4-8) shows this to indeed be the case. The TIN has quite a rugged surface, especially in the zone of transition, that is not representative of actual terrain conditions.

Given that the RAS vector data and DEM raster data have different collection times, methods, and resolution, it is not surprising that they are

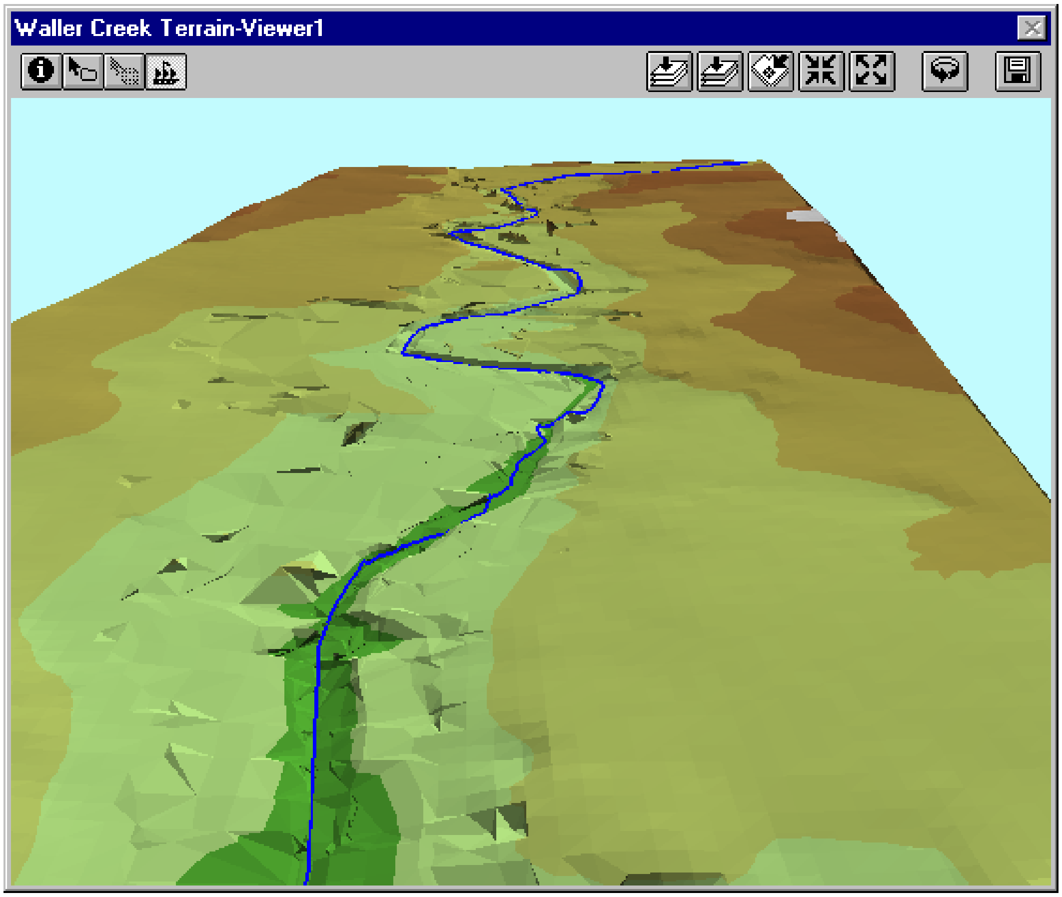


Figure 4-13. Waller Creek terrain TIN.

somewhat incompatible. The floodplain elevation data is an amalgam of land surveys and topographic map estimations over a period of 10-15 years in the 1970's and 1980's. In contrast, the DEM is a product of the process used to develop the TNRIS 1995 DOQQs. The zone of transition between the two data sets typically had an elevation gap ranging between 0 and 2 meters. These elevation differences may not be representative of actual terrain conditions. As such, the next task was to develop an approach with which to smooth the RAS-DEM data transition zone.

The cross-section point elevations for the Waller Creek HEC-RAS model were collected during various land surveys. However, elevations in the floodway were often estimated by City of Austin engineers, based on available topographic maps. Hence, the accuracy of the HEC-RAS data is likely greater within the channel than in the floodway. So in comparison to the HEC-RAS data, the DEM elevations in the floodway were considered the more accurate, albeit lower resolution data source. Also, the DEM represents the entire floodway terrain, not only the places where the cross-sections exist. In order to smooth the transition zone, elevations in the cross-sections and/or DEM data points need to be altered. Because the cross-section elevation data in the floodway are considered the least accurate, an interpolation approach is applied to these data.

Within each cross-section, the elevations of all points between the bank stations (within the channel) are left unchanged. At the end of each cross-section, the elevation is assumed equal to that of the DEM. The smoothing approach was applied to all cross-section points in the left and right floodways. The approach is as follows for each cross-section:

1. Calculate the lateral distance along the cross-section between the bank station and the cross-section end, on both the left and right hand side of the cross-section.
2. Identify the first cross-section point outside the bank station and note its elevation.
3. Query the DEM elevation at that location.
4. Determine the point's location in the floodway, as a percentage of the distance calculated in step 1. The new elevation of the point is calculated as a weighted average of the original elevation and the DEM elevation:

$$z2 = [z1*(1-Pct) + (DEMz*Pct)] \qquad \text{(Eq. 4-5)}$$

Where: z1 = original point elevation

DEMz = DEM elevation at the point

z2 = new point elevation

Pct = point location in the floodway, measured as a percentage of the lateral distance between the bank and cross-section end

For example, if the point is located 40% away from the bank station, the new elevation is the sum of 60% of the original elevation and 40% of the DEM elevation.

5. Repeat steps 1 through 6 for all floodway cross-section points.

The Avenue script **NewXSects.ave** carries out the approach described above. Akin to the original cross-section line theme (Figure 4-10), the resampled cross-section theme is attributed with river station number, cross-section length, and the locations of the center point and bank stations. By employing the transition zone smoothing approach, the point elevations in the floodway will gradually trend toward the DEM elevation, moving from the bank station to the end of the cross-section. To help illustrate the effect of the cross-section elevation resampling, the Avenue script **Compare.ave** is applied. The script determines the coordinates of points along a given cross-section as defined by three input data sets: the original HEC-RAS data, the DEM, and the resampled HEC-RAS data. The coordinates are written to a comma delimited text file, which can subsequently be imported into a spreadsheet and plotted (Figure 4-14).

As shown in the Figure 4-14, the original and resampled cross-sections are identical within the channel. But in the floodway, the elevation of the resampled cross-section falls somewhere between the original and DEM elevation. Using the resampled cross-section theme as input, the terrain TIN is reconstructed. In the reconstructed TIN, the land surface is more fluid and the zone of transition is difficult to discern. The figure also demonstrates how 30-meter DEMs do not provide sufficient detail within the channel for hydraulic modeling applications. However, recent advances in digital photogrammetry have allowed the production of higher resolution DEMs, such as a 10m DEM (Figure 4-15). The 10m DEM generates a somewhat better representation of the channel than the 30m DEM.

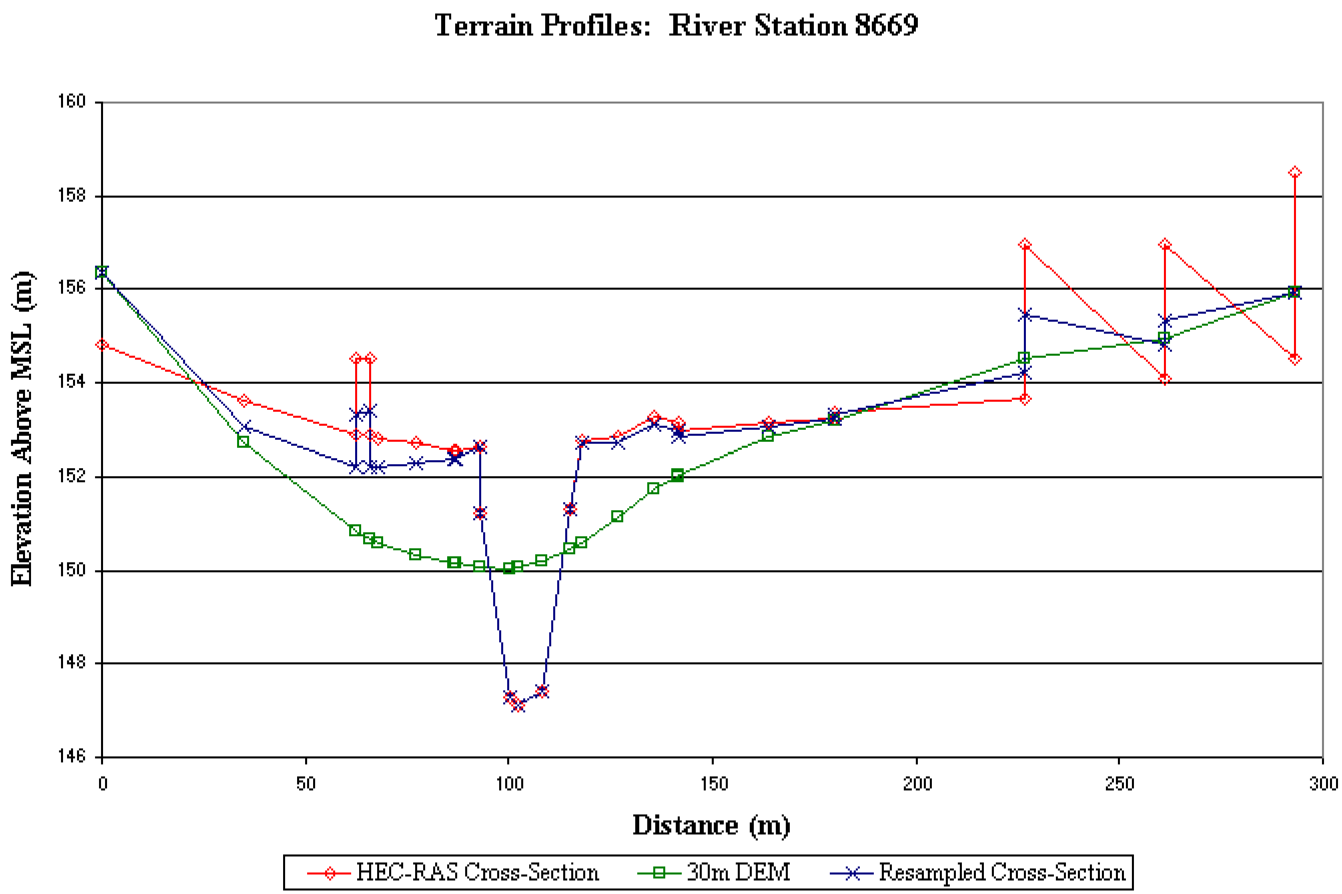


Figure 4-14. Terrain profiles based on vector cross-sections and a 30m DEM.

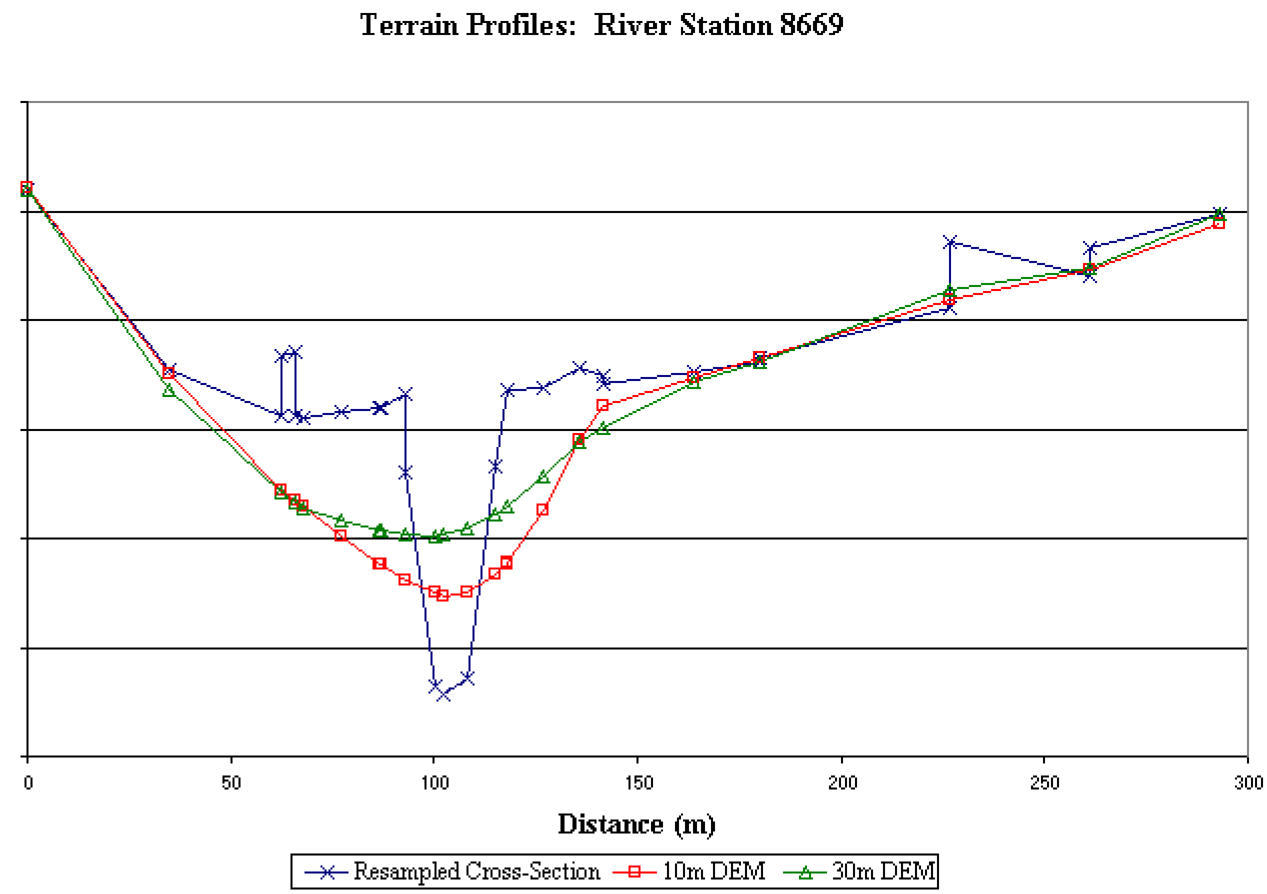


Figure 4-15. Terrain profile comparison between HEC-RAS and 10m and 30m DEMs.

4.5 FLOODPLAIN MAPPING

With the terrain model complete, the final step is to delineate the floodplain. HEC-RAS represents stream floodplains as a computed water surface elevation at each cross-section. During the data import step (Subsection 4.1), these elevations, along with the distance from the stream centerline to the left and right floodplain boundaries, are brought into ArcView and stored in the cross-section parameter table. Hence, two things are known about the floodplain at each cross-section: water surface elevation and extent. The script **Water3D.ave** was developed to map these attributes. On Figure 4-16, the water surface profiles (heavy blue lines) are mapped based on the cross-section line theme (green).

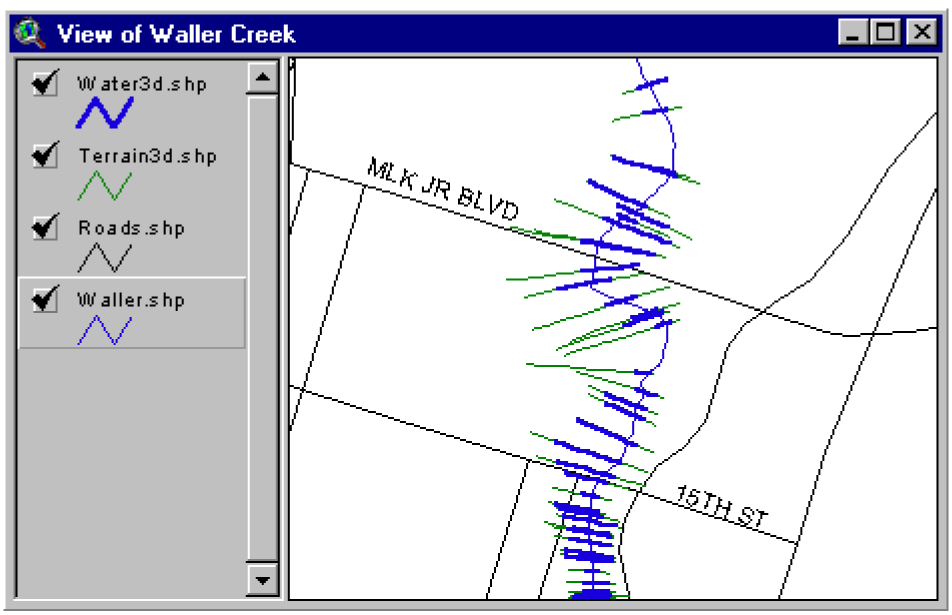


Figure 4-16. Water surface profiles.

As inputs, the script requires the cross-section line theme and the cross-section parameter table. The output is a line theme that is identical to the cross-section theme in location and orientation, but is not as wide. In a properly set up HEC-RAS model, the cross-sections should be designed wide enough such that the computed water surface elevations are contained within them.

Using the water surface profile at each cross-section, a TIN representing the entire floodwater surface can be constructed using the **Surface/Create TIN from Features** menu option in ArcView. The water surface lines are used as breaklines, and the cross-section bounding polygon is used to bound the aerial extent of the water surface. When viewed in conjunction with the terrain TIN, flooded areas can be seen (Figure 4-17).

The three-dimensional floodplain view is quite useful for floodplain visualization. But the view shown in Figure 4-17 doesn't appear much like the actual landscape. To remedy this, themes of roads, buildings, railroads, and other typical landscape features can be added to the view to more closely approximate reality. If building elevation information is not available, building elevations can be measured on a DOQQ based on shadow length. The calculation requires one known building height (Avery, 1992):

$$H_2 = (H_1 * L_2)/L_1 \qquad \text{(Eq. 4-6)}$$

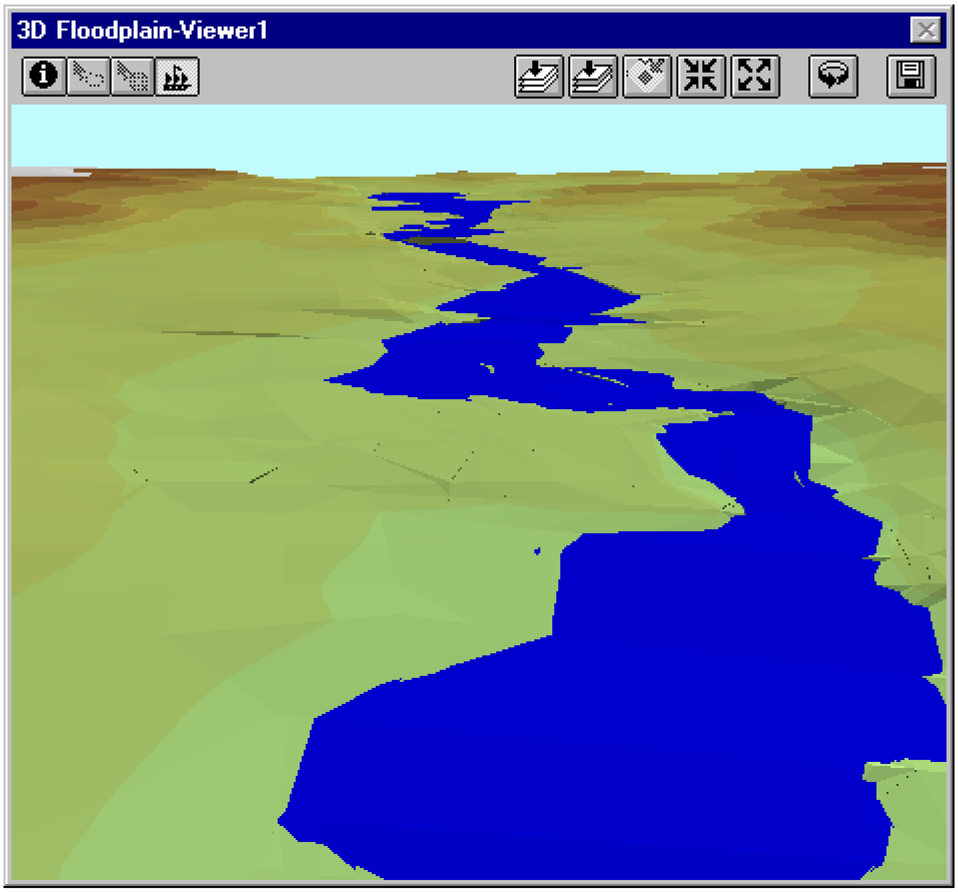


Figure 4-17. Three-dimensional floodplain rendering.

Where: H_n = height of building n

L_n = shadow length of building n

Figure 4-18 shows the level of detail that can be obtained by building a detailed TIN surface model that includes buildings and roads. The view shows the Colorado River south of downtown Austin.

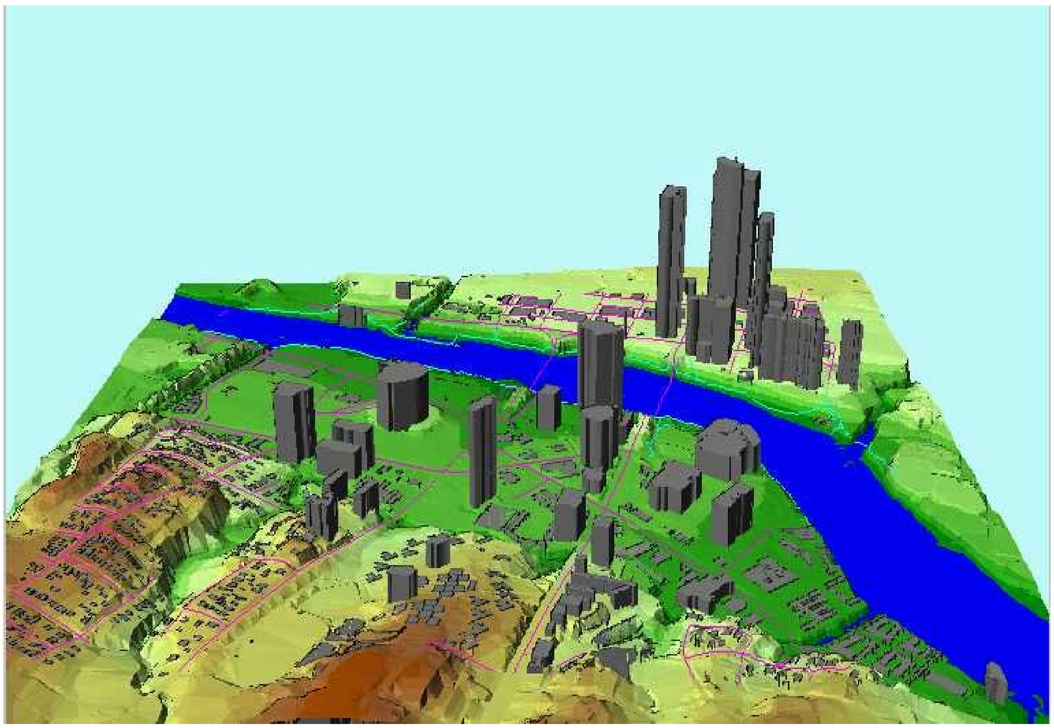


Figure 4-18. Potential level of TIN detail.

For detailed analysis, the floodplain can also be viewed with a planimetric perspective using a digital orthophotography as a base map. However, the method developed for planimetric floodplain visualization is different than that for three-dimensional floodplain visualization because it is based on the raster data model instead of the TIN. By definition, areas inundated by floodwaters occur wherever the elevation of the water exceeds that of the land surface. In terms of the data sets developed thus far, the floodplain exists wherever the elevation of the water surface TIN exceeds the terrain TIN. However, the delineation of these areas is more readily performed using the raster data model instead of TINs. In the raster domain, grid cells of water surface elevation and grid cells of land surface elevation can be easily compared.

But in order to be consistent, the inundated areas should not be delineated using the integrated (resampled) terrain TIN. This is because the HEC-RAS water surface profiles used to create the water surface TIN were computed using the original HEC-RAS cross-sections and not the resampled cross-sections. As such the cross-section elevation correction should be neglected. So the terrain TIN is reconstructed using the original cross-section line theme. As before, the TIN is created using three input data sources: the DEM point theme, original cross-section line theme, and the centerline and banks line theme. TIN nodes are formed from the DEM points and the vertices of the cross-section lines. The stream centerline and bank lines are enforced in the TIN as breaklines. The **Surface/Create TIN from Features** menu selection is used to build the TIN.

Using the **Theme/Convert to Grid** menu selection available in ArcView view window, the terrain and water surface TINs are subsequently each converted to grids, with the same analysis extent and resolution. An output grid resolution of 1 meter is used for the TIN to raster conversion. The Avenue script

Floodpln.ave is then used to compare the two grids and delineate areas inundated by floodwater. The script employs the following steps:

1. Subtract the terrain grid from the water surface grid to create a grid of flood depths. In the flood depth grid, cells with positive values represent areas inundated by floodwater; cells with negative values denote unflooded areas.

2. Query the flood depth grid for all cells with a value greater than zero. In the resulting query grid, flooded cells are assigned a value of one (true) and unflooded cells receive a value of zero (false).

3. Divide the flood depth grid by the query grid to create a revised flood depth grid. In this step, flooded cells are simply divided by one and thus retain their original value. However, unflooded cells are divided by zero, and are therefore assigned a value of "no data" by ArcView. So in the revised depth grid, only cells inundated by floodwater possess a numerical value.

The revised flood depth grid can be overlaid on the orthophoto to facilitate spatial analysis of both floodplain extent and depth (Figure 4-19).

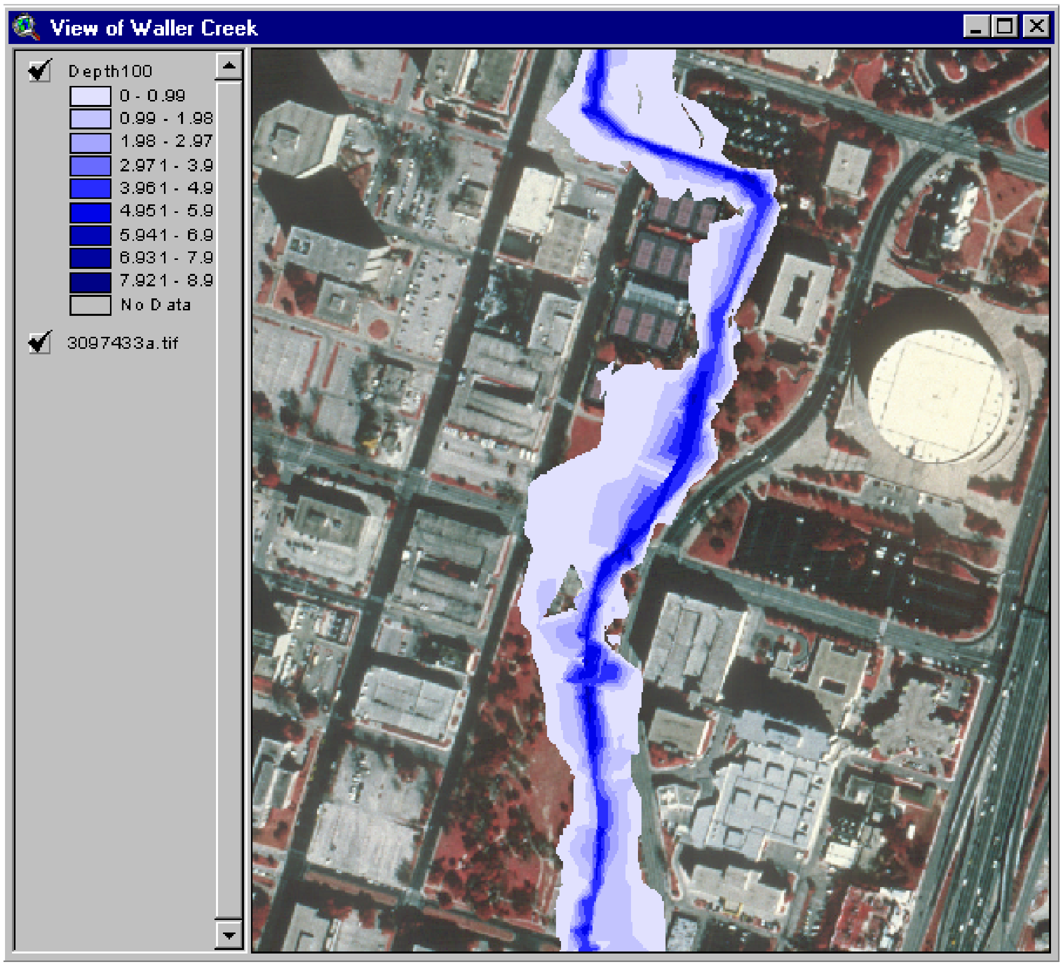


Figure 4-19. Planimetric view of grid-based floodplain delineation.

Chapter 5: Results and Discussion

Using the procedure detailed in Chapter 4, a TIN terrain model and raster floodplain map was developed. The discussion in this chapter includes validation of the research results and potential applications.

5.1 VALIDATION

In order to assess the accuracy of the terrain model and floodplain map, they must be validated against independent data sources.

5.1.1 TIN Terrain Model

For the TIN terrain model, a comparison is made with a TIN based on data from the Capital Area Planning Council (CAPCO). CAPCO is an agency tasked with obtaining and developing aerial photogrammetry data for the Austin area and nine surrounding counties. The photogrammetry data, which includes among other things, digital orthophotographs, DEMs, spot elevations, hydrography, and hypsography, were collected between 1996 and 1998.

To validate the terrain TIN, CAPCO spot elevation coordinates for the Waller Creek area were used to create a digital terrain model (DTM) of the land surface. The resulting DTM has a very high resolution, with a data point density of up to one per square foot in some areas. To generate cross-section profiiles of the integrated TIN and CAPCO DTM, the **Compare.ave** script was employed.

The script is designed to operate on grids, so the TIN and DTM were first converted to 1-meter resolution grids before the script was applied to them. The resulting profiles are shown on Figures 5-1 through 5-3.

The general shapes of the profiles are quite similar, but a horizontal offset ranging between 5 and 15 meters between the data sets is consistent. Given that the profile shapes are so similar, this offset could be indicative of differences between the CAPCO stream centerline and the stream centerline representation used in this research. Perhaps if the stream centerline for the integrated TIN were defined based on a DEM, the offset would be less. Neglecting the horizontal shift, the elevation offset is significantly less, typically ranging between 0 and 2 meters. Both the integrated TIN and the CAPCO DTM have a high density of points within the channel. As such, it appears that both terrain models could be used as a source of cross-sectional data for river hydraulic modeling. However, the availability of the input data distinguishes the two terrain models. Although more and more cities and agencies are contracting aerial photogrammetry projects, photogrammetry spot elevation data currently are not generally available. In contrast, in many areas, both developed HEC-RAS models and 30m DEMs are easily available. But it is significant that data of comparable quality can be obtained from standard aerial photogrammetric procedures.

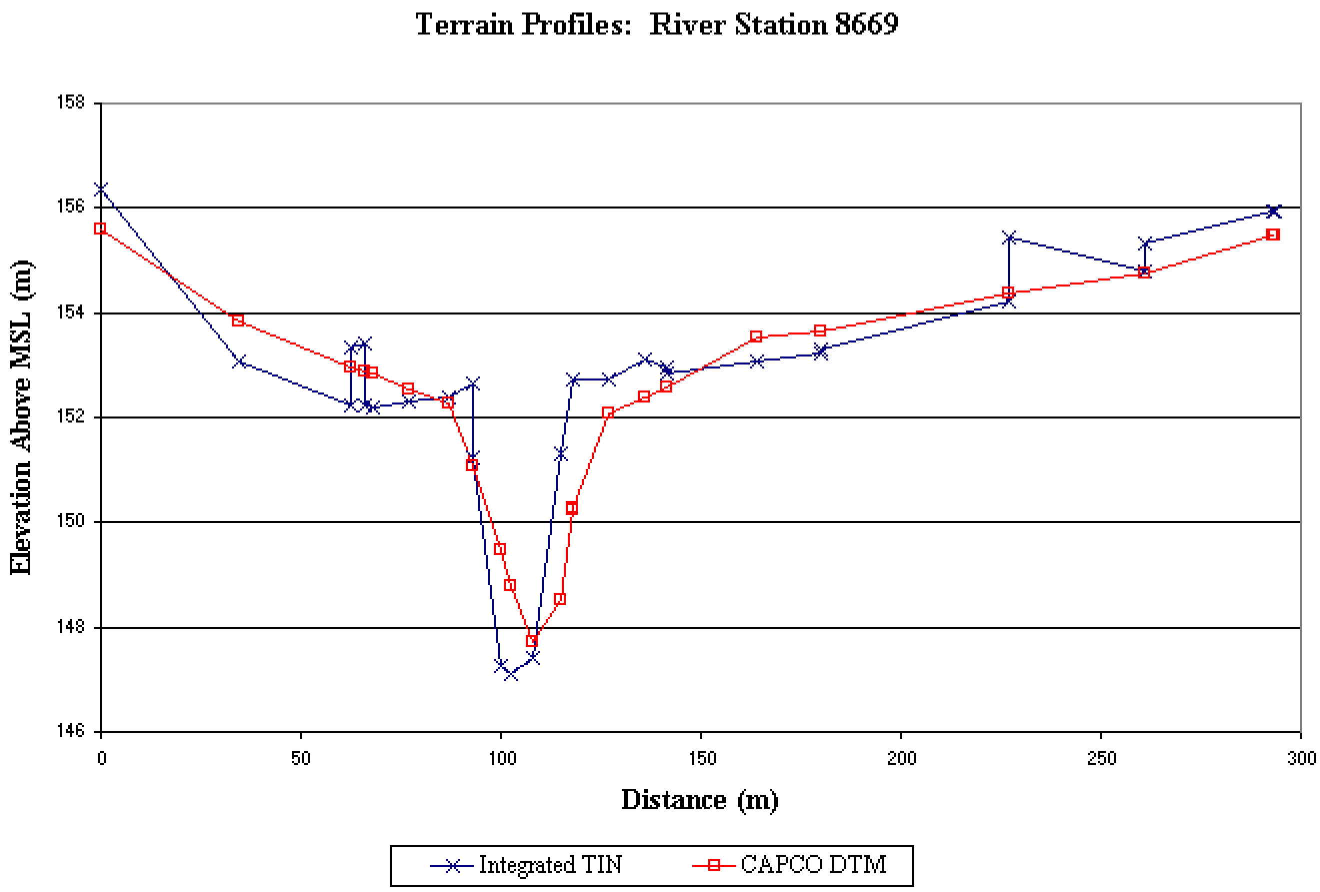


Figure 5-1. Integrated TIN and CAPCO DTM comparison, station 8669.

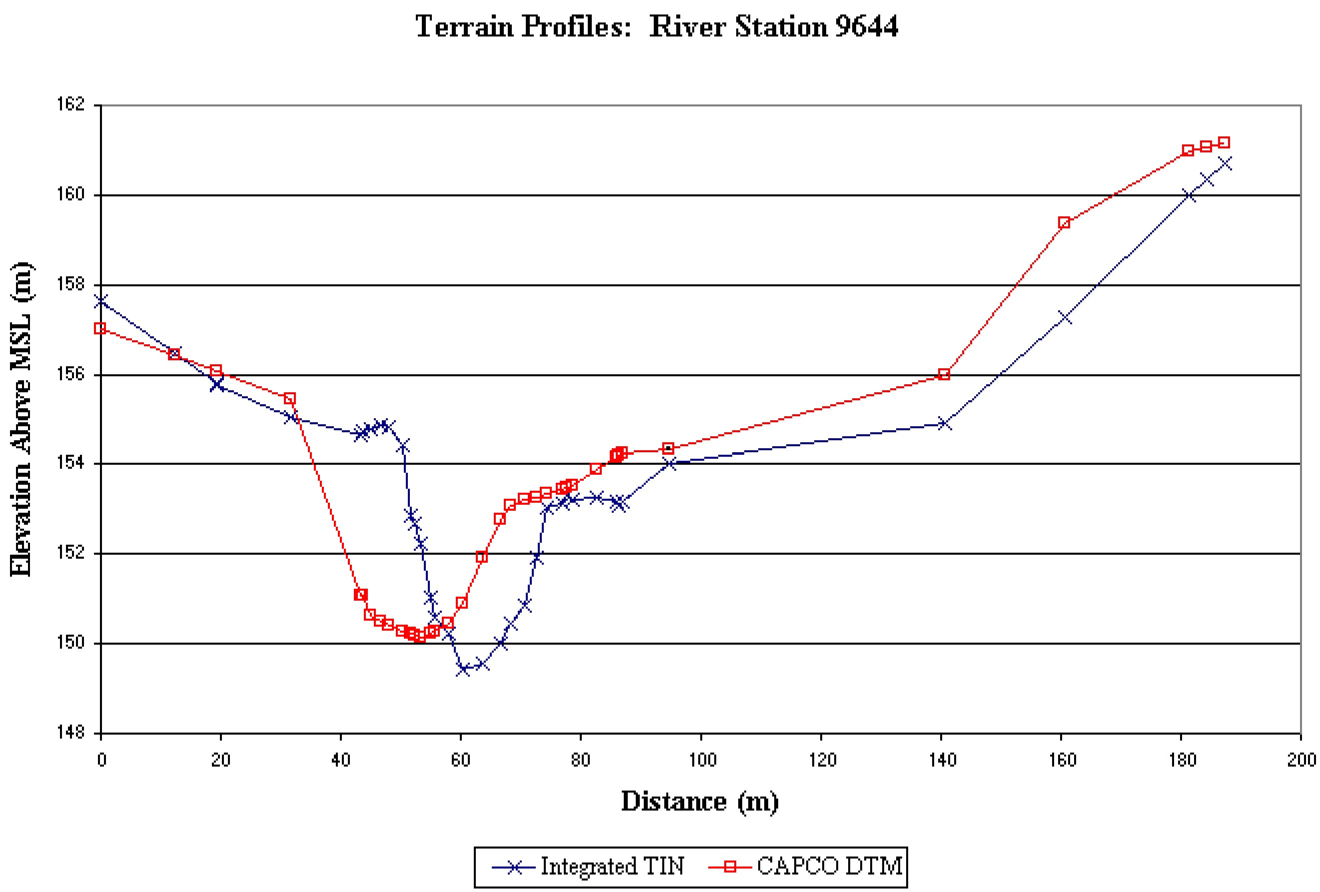


Figure 5-2. Integrated TIN and CAPCO DTM comparison, station 9644.

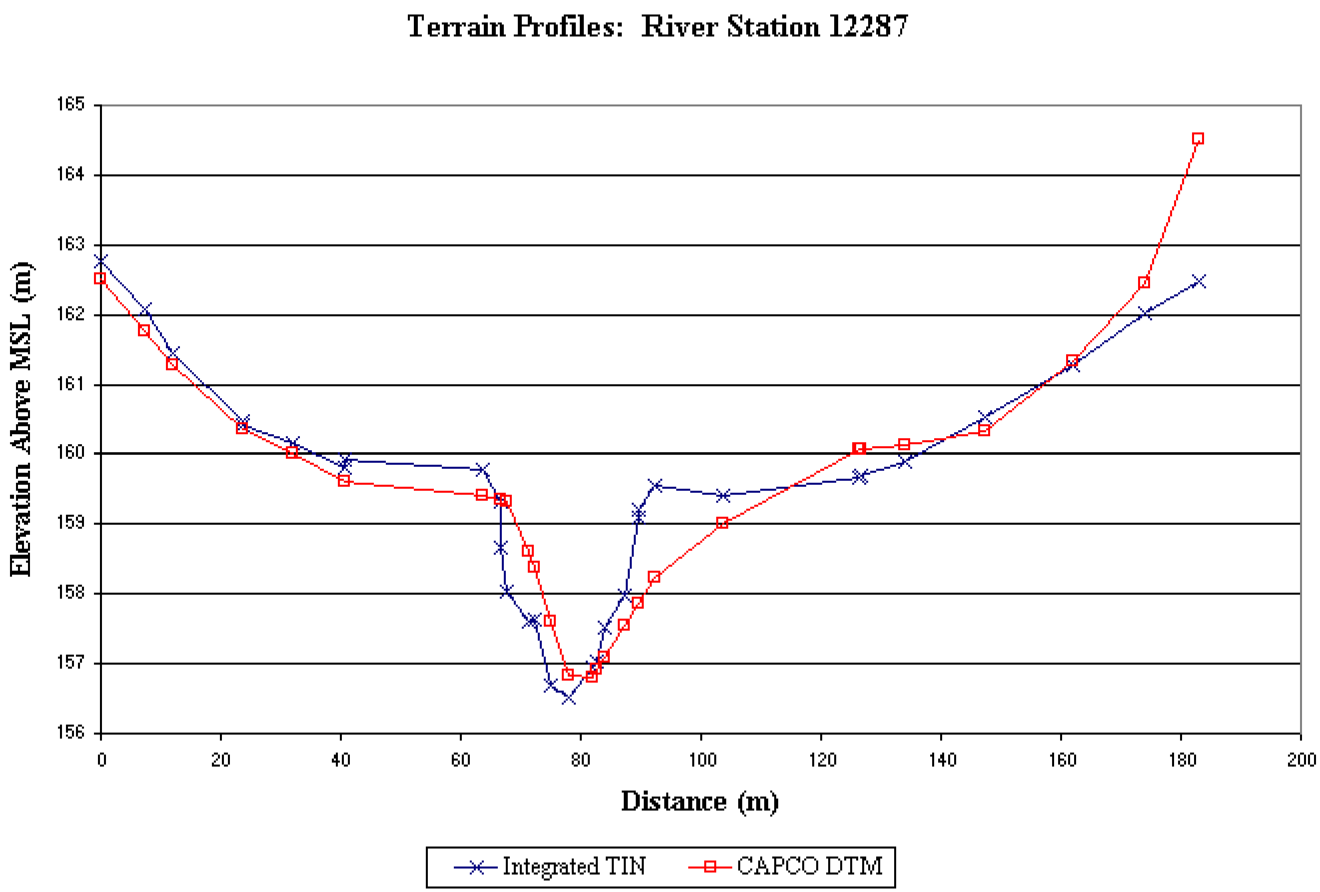


Figure 5-3. Integrated TIN and CAPCO DTM comparison, station 12287.

5.1.2 Floodplain Delineation

Digital Q3 flood data were used to validate the floodplain delineation. Digital Q3 flood data are developed by the Federal Emergency Management Agency (FEMA), by scanning hardcopy Federal Insurance Rate Maps (FIRM). FIRMs identify areas of 100-year flood hazard in a community based on detailed or approximate analyses (FEMA, 1999b). FIRMs are issued by FEMA and are the official standard for floodplain delineation. Digital Q3 data for certain counties are available over the internet from FEMA's website. The Q3 data for Travis County, Texas were downloaded, imported into ArcView, and projected to UTM zone 14, NAD 83. In ArcView, the Q3 data are represented as a polygon shapefile. But unlike the grid-based delineation, Q3 has no attribute of water depth. However, the two floodplains can be compared based on floodplain location/extent.

As shown in Figure 5-4, the general shapes of the floodplain representations are similar, but in several places, the floodplains cover different parts of the land surface. In addition, the raster floodplain representation is often narrower than the Q3 polygon. Unfortunately, a head to head comparison of the grid and Q3 floodplains is not necessarily valid due to fundamental differences with the input data used to develop each floodplain map. It is likely that FEMA's HEC-2 model and the City of Austin's HEC-RAS model use different stream cross-sectional and 100-year flow data. The Q3 data is based on a 1993 FIRM, while the HEC-RAS geometry and 100- year flows were last updated in 1998.

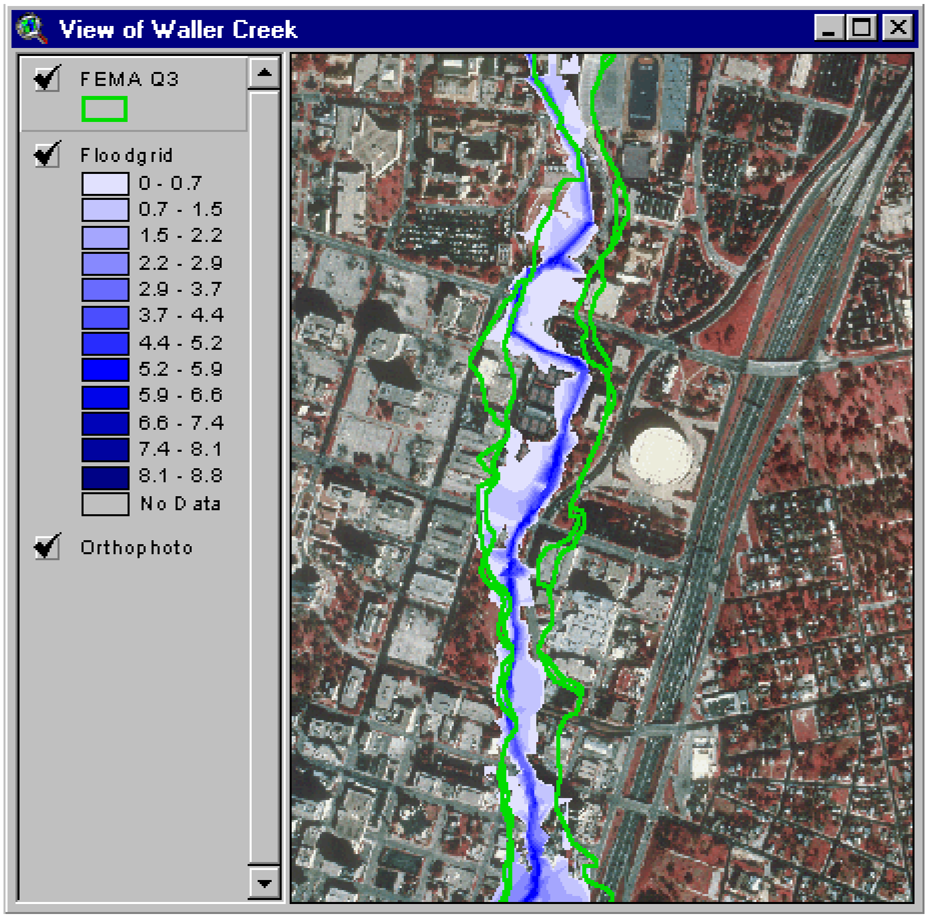


Figure 5-4. Comparison of floodplain with FEMA Q3

Based on these differences, the grid and Q3 floodplains cannot be closely compared. Unfortunately, FEMA's Q3 is the only widely accepted digital floodplain data available for Waller Creek. As such, the validation of the raster floodplain delineation cannot be completely confirmed.

5.2 APPLICATIONS

The research described in this report offers procedures for the automation of floodplain mapping based on hydraulic modeling output. The Avenue scripts are associated with menu items in ArcView, so the user can map floodplains primarily with only a few clicks of the mouse. The GIS based delineation should result in significant savings of time and resources when compared to manually plotting the HEC-RAS output. Some potential applications for this work include the following:

- **Hydraulic design**. An important design component of bridges, culverts, and other drainage control facilities involves hydraulic analyses to determine conveyance capacity. Using the grid-based floodplain delineation, a hydraulic engineer can zoom in on the area of a particular drainage control structure, and both view the floodplain extent and query the flood grid for water depth at various locations of interest.

- **Terrain modeling**. As more local and state governments invest in photogrammetry studies such as CAPCO, more detailed DTMs will be available. However, these studies often do not obtain elevation data for areas perennially inundated by water. Hence, a method to integrate the DTMs with surveyed channel elevation data obtained for hydraulic modeling is important. As stated previously (Subsection 1.3), the HEC-RAS model contains a user option to import three-dimensional river reach and cross-sectional data from a

GIS. The integrated TIN accurately describes both the floodway and channel morphology to a degree required for hydraulic modeling.

- **Flood warning systems**. Real-time analysis during an intense storm would involve using measured rainfall as input for hydrologic modeling, applying the output flow rates to hydraulic modeling, and finally mapping the output in a GIS. Theoretically, this information could then be used to coordinate flood warning activities such as road closures and evacuations. Unfortunately, the stream velocities that occur during a flood are generally too high to make on the fly floodplain mapping practical. Instead, the floodplain mapping procedures outlined in this report could be used to create a series of floodplain maps based on a variety of flow scenarios. During an intense storm event, the flood warning coordinator could select the specific digital floodplain map that corresponds most closely to the real-time stream flow measurements.

Chapter 6: Conclusions and Recommendations

An approach for automated floodplain mapping and terrain modeling is presented. The work provides a link between hydraulic modeling using HEC-RAS, and spatial display and analysis of floodplain data in ArcView GIS. As inputs, the approach requires a completed HEC-RAS model simulation and a GIS stream centerline representation. The procedure consists of several steps: data import from HEC-RAS, stream centerline representation, cross-section georeferencing, terrain modeling, and floodplain mapping. The outputs are a digital floodplain map that shows both extent and depth of inundation, and a TIN model of the terrain synthesized from high resolution HEC-RAS cross-section data and lower resolution digital elevation model (DEM) data describing the regional landscape. The discussion in this chapter includes benefits and limitations of the approach, recommendations, and suggestions for additional research.

6.1 BENEFITS

The process developed for automating terrain modeling and floodplain delineation has several noteworthy benefits:

- **User interface**. Through the use of menu items and sample exercises, floodplain mapping is automated and simplified. The user need not be a GIS expert to quickly and easily produce detailed floodplain maps. In addition to showing the aerial extent of flooding, the floodplain delineation includes flood depth information.

- **Digital output**. Rendering the floodplain in digital GIS format allows the floodplain data to be easily compared with other digital data, such as digital orthophotography and GIS coverages of infrastructure, buildings, and land parcels.

- **Integrated terrain model**. Currently, there are no available methods with which to combine hydraulic model data with a DEM. The terrain models produced by the procedure are detailed in the stream channel, but also represent the general landscape. The integrated terrain model is comparable in quality to terrain model data acquired solely through aerial photogrammetry. The integrated terrain modeling approach could be used to prepare input data for hydraulic modeling.

- **Resource savings**. Many floodplain maps need to be revised because they are outdated. The automated mapping approach developed for this research saves time and money versus conventional floodplain delineation on paper maps. Thus, floodplain maps can be updated more frequently, as changes in hydrologic and hydraulic conditions warrant.

6.2 LIMITATIONS

The main limitation of this approach is the assumption of straight-line cross-sections. The HEC-RAS model requires cross-sections to be defined such that they are perpendicular to the flow lines in both the floodways and main channel. As a result, land surveys of river cross-sections take the perpendicularity requirement into account. Within relatively straight portions of the channel this equates to straight-line cross-sections. Near bends in the stream, the cross-sections are surveyed perpendicular to the channel, but doglegged in the floodways to ensure perpendicularity to flow. Unfortunately, information concerning the orientation of each cross-section is indicated on survey maps, but is not routinely stored by HEC-RAS. In order to map the cross-sections, the approach in this study assumes that all cross-sections occur in straight lines.

The effect of the straight-line assumption on the accuracy of the resulting terrain models and floodplain maps varies with the distance from the stream channel. Within the channel and near the stream banks, the effects of the straight-line assumption are minor because surveyed cross-sections are not likely to be doglegged. Moreover, the distance from the stream centerline is small in these areas, so changes in cross-section orientation have little effect on the accuracy of the terrain model and floodplain map. If the intention of the mapping project is to evaluate terrain and floodwater conditions in the channel, this approach is effective.

As the distance along the cross-section away from the channel increases, so does the likelihood of incorrectly mapping the cross-section. This is especially true of cross-sections near bends in the stream, where the survey maps may indicate a dogleg. The negative effect on the accuracy of the terrain model will be muted because in the floodway, the terrain TIN includes data from both HEC-RAS and a DEM. For the floodplain map, the effects will be more profound. If a straight-line water surface profile is mapped where a doglegged profile should be, the resulting floodplain will err in location. In addition, the floodplain width may be overestimated because a straight-line water surface profile is mapped instead of a bent line.

6.3 RECOMMENDATIONS

During the course of this research, several important concepts were discovered and noted, that have an important effect on the ease of the hydraulic data mapping process and the quality of the resulting output. These concepts are shared in the following paragraphs.

1. **Alone, 30m and 10m DEMs do not provide sufficiently detailed channel representations for floodplain modeling**. The level of detail contained in the DEM channel representations is not adequate to be used as the source of cross-sectional data for hydraulic modeling.

2. **Datums, map projections, and units are very important**. For this project, the HEC-RAS data were described in feet. The orthophotography had units of meters, with a UTM map projection based on the NAD83 datum. The DEM had horizontal units of meters, elevation units of feet, and had a UTM map projection based on the NAD27 datum. Based on these significant differences between the three data sources, establishing a common datum, projection, and unit of measurement is critical. Otherwise, the resulting terrain model and floodplain delineation are incorrect.

3. **Use a DEM for stream centerline definition**. In Subsection 4.2, four methods were presented for definition of a stream centerline: land surveys, reach files, digitizing from a digital ortho quarter quad (DOQQ), and DEM-based delineation. Surveyed stream centerline data in GIS format is often not available, and RF3 was shown to be inaccurate in the Waller Creek study area. The stream centerline representation for this project was developed through digitizing on a DOQQ base map. This method is probably the best if the primary goal is floodplain mapping. Because the final floodplain map is displayed on the DOQQ base map, it makes sense to also develop the stream centerline based on the DOQQ. If the primary goal of the procedure is terrain modeling, perhaps a DEM-based approach for stream centerline delineation is preferred. As shown on Figures 5-1 through 5-3, the stream channel in the integrated TIN had a consistent horizontal offset from its counterpart in the CAPCO DTM. This offset is likely the result of differences in the stream

centerline location. Delineating the centerline based on a DEM may reduce the magnitude of this problem. In addition, the DEM-based approach is automated and the delineation is wholly reproducible, reducing the opportunities for human error.

4. **Make sure enough cross-sections are defined in HEC-RAS**. As part of the procedure detailed in Chapter 4, two representations of the stream centerline are developed. In the stream centerline definition step (Subsection 4.2), methods for creating a two-dimensional representation of the centerline are discussed. During the terrain modeling process (Subsection 4.4), a three-dimensional line theme of the stream centerline and bank stations is created. These three-dimensional lines define the channel of the TIN terrain model. However, vertices of the three-dimensional centerline and banklines only occur wherever a cross-section intersects the two-dimensional centerline representation. As such, a sufficient density of cross-sections is required to closely approximate the shape of the 2D centerline. Otherwise, bends and sharp elevation changes in the channel may not be well represented in the terrain model. The appropriate density of cross-sections should be determined based on the shape of the channel and requirements for hydraulic modeling. To increase the density of cross-sections in HEC-RAS, the cross-section interpolation menu option can be employed

5. **Limit HEC-RAS coordinate data to two decimal places**. When the HEC-RAS cross-section coordinate data is defined to three or more decimal places, a problem sometimes occurs in the data import step. This because the HEC-RAS output report only allows enough room for six digits. If more than six digits are used to define a particular coordinate, the value may be joined to the next coordinate in the output file, not space delimited as it should be. For example, a given location with the HEC-RAS coordinates 1144.5797, 650.84 might appear in the output text file as 1144.5797650.84. When presented with a value such as this, the ArcView import script will not work properly. To avoid this potential problem, it is recommended that the HEC-RAS geometry data be edited so that all coordinates possess no more than two decimal places.

6.4 ADDITIONAL RESEARCH

The procedures described herein can be improved upon to facilitate expanded analysis and visualization capabilities. Several modes of additional research are suggested:

- Further evaluate the terrain model. For a series of cross-sections, it would be interesting to observe and quantify the differences in representation provided by the 30-meter DEM, 10-meter DEM, CAPCO DTM, and the integrated TIN model developed by this research.

- Modify the Avenue scripts to allow development of terrain models and floodplain maps for stream networks. HEC-RAS has the capability to model stream networks, but the research approach can currently only be applied to a single, unbranched stream. The scripts could be modified to read output data based on networks and map the associated terrain and floodplain data.

- Further develop the terrain model to allow additional terrain features to be added to the surface model. These features, such as roads and buildings, could be used to produce a three-dimensional rendering of the terrain that closely approximates reality. Using the ArcView 3D Analyst extension, these data can be draped over the TIN surface to facilitate three-dimensional visualization.

Appendix A: Tutorial Exercises

Introduction to HEC-RAS

Prepared by Eric Tate
Center for Research in Water Resources
April 1999

Table of Contents

Goals of the Exercise

The primary goal of this exercise is to introduce you to channel flow modeling using the HEC River Analysis System (HEC-RAS). By the end of this exercise, you should be able to:

- Import and edit cross-sectional geometry data
- Import and edit flow data from HEC-HMS
- Perform a steady flow simulation

- View and analyze HEC-RAS output

Software and Data Requirements

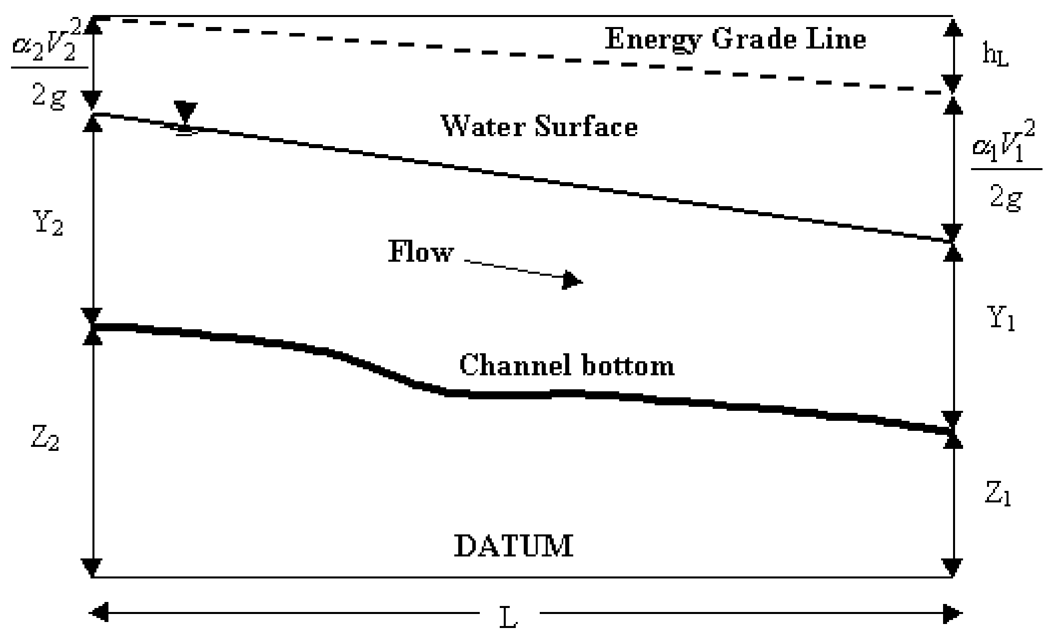
The HEC-RAS program can be downloaded free of charge from the Hydrologic Engineering Center's home page at: http://www.wrc-hec.usace.army.mil/. A user's manual is also available at this location. The program runs on Windows 95 & 98, NT, and Unix platforms. This tutorial requires that you have running HEC-RAS version 2.2. The data required for the tutorial consist of HEC-RAS input files. These data can be downloaded from this website as the file hecras.zip. The files included are:

- Waller.dss – HEC-HMS time series output in Data Storage System format
- Waller.g01 – Waller Creek HEC-RAS geometry file

Select a working directory on your computer and download these files into it.

HEC-RAS Hydraulics

HEC-RAS is a one-dimensional steady flow hydraulic model designed to aid hydraulic engineers in channel flow analysis and floodplain determination. The results of the model can be applied in floodplain management and flood insurance studies. If you recall from hydraulics, steady flow describes conditions in which depth and velocity at a given channel location does not change with time. Gradually varied flow is characterized by minor changes in water depth and velocity from cross-section to cross-section. The primary procedure used by HEC-RAS to compute water surface profiles assumes a steady, gradually varied flow scenario, and is called the direct step method. The basic computational procedure is based on an iterative solution of the energy equation: $H = Z + Y + \frac{\alpha V^2}{2g}$, which states that the total energy (H) at any given location along the stream is the sum of potential energy (Z + Y) and kinetic energy ($\alpha V^2/2g$). The change in energy between two cross-sections is called head loss (h_L). The energy equation parameters are illustrated in the following graphic:



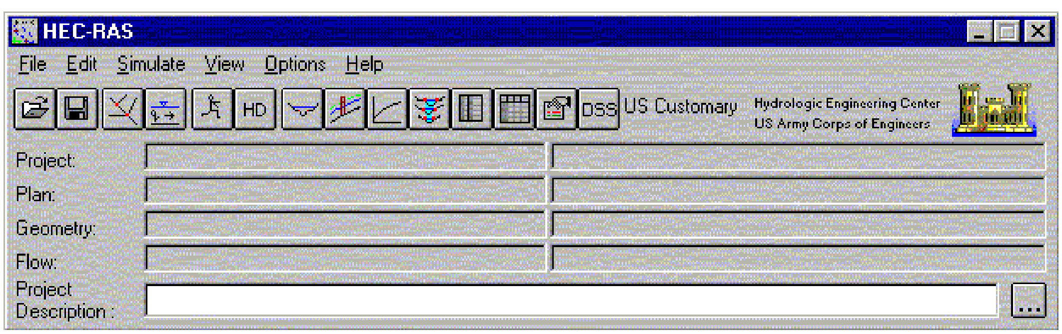
Given the flow and water surface elevation at one cross-section, the goal of the direct step method is to compute the water surface elevation at the adjacent cross-section. Whether the computations proceed from upstream to downstream or vice versa, depends on the flow regime. The dimensionless Froude number (Fr) is used to characterize flow regime, where:

- Fr < 1 denotes subcritical flow
- Fr > 1 denotes supercritical flow
- Fr = 1 denotes critical flow

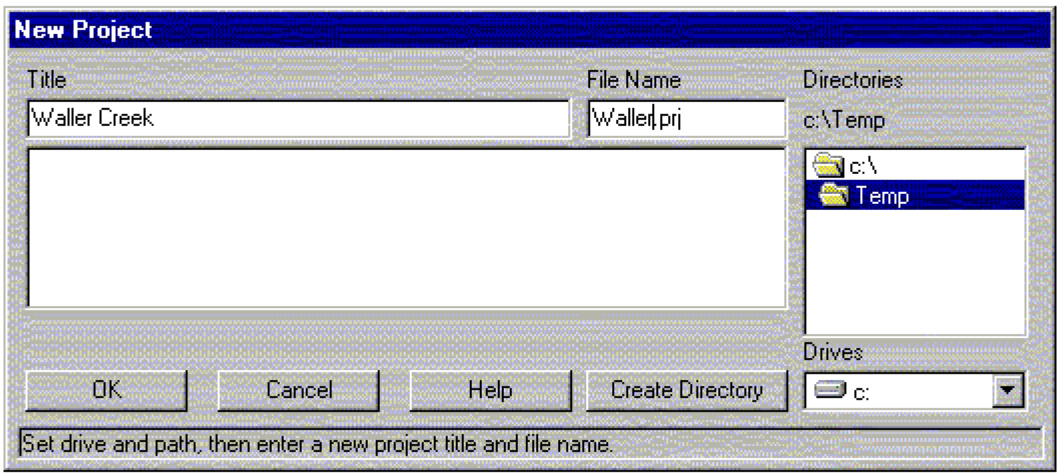
For a subcritical flow scenario, direct step computations would begin at the downstream end of the reach, and progress upstream between adjacent cross-sections. For supercritical flow, the computations would begin at the upstream end of the reach and proceed downstream.

Starting a Project

You may start the HEC-RAS program by clicking **Start/Programs/Hec/HEC-RAS 2.2**. The following window should subsequently appear:



Henceforth, this window will be referred to as the main project window. A **Project** in RAS refers to all of the data sets associated with a particular river system. To define a new project, select **File/New Project** to bring up the main project window:

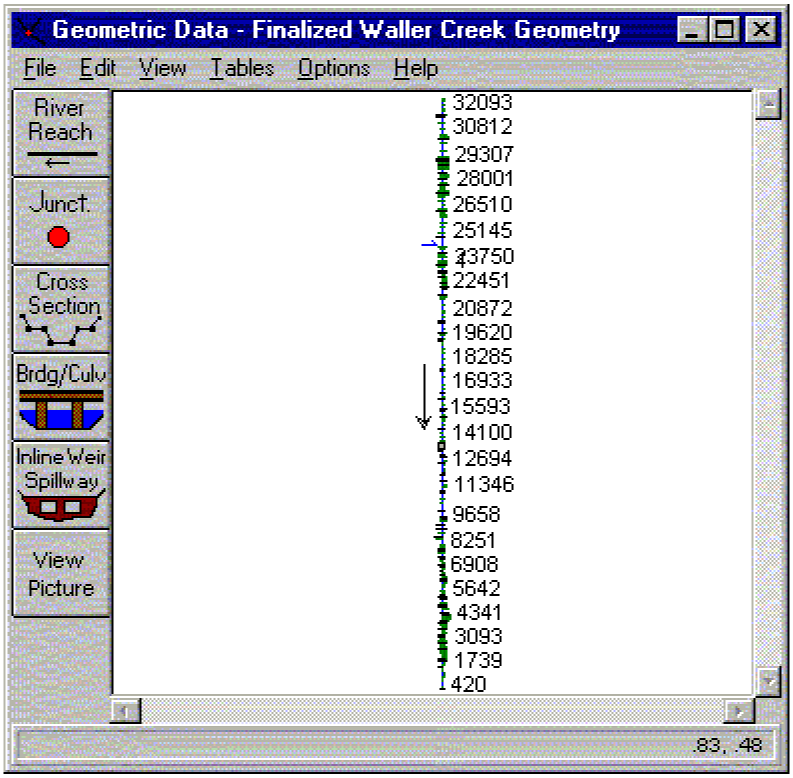


You will first need to select your working directory, and then a title (Waller Creek), and file name (Waller.prj). All project filenames for HEC-RAS are assigned the extension ".prj". Click on the OK button and a window will open confirming the information you just entered. Again click the OK button. The project line in the main project window should now be filled in. The **Project Description** line at the bottom of the project definition window allows you to type a detailed name for the actual short **Project** name. If desired, you may click on the ellipsis to the right of the **Description** bar, and additional space for you to type a lengthy **Description** will appear. Any time you see an ellipsis in a window in HEC-RAS, it means you may access additional space for writing descriptive text.

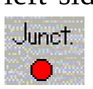
For each HEC-RAS project, there are three required components--**Geometry** data, **Flow** data, and **Plan** data. The **Geometry** data, for instance, consists of a description of the size, shape, and connectivity of stream cross-sections. Likewise, the **Flow** data contains discharge rates. Finally, **Plan** data contains information pertinent to the run specifications of the model, including a description of the flow regime. Each of these components is explored below individually.

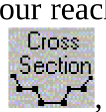
Importing and Editing Geometric Data

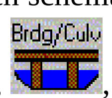
The first of the components we will consider is the channel geometry. To analyze stream flow, HEC-RAS represents a stream channel and floodplain as a series of cross-sections along the channel. To create our geometric model of Waller Creek for example, we need to import the geometry file that you just downloaded. In the main HEC-RAS project window, use **File/Import HEC-RAS Data** and choose the file Waller.g01. This HEC-RAS geometry file contains physical parameters describing Waller Creek cross-sections. To view the data, select **Edit/Geometric Data** from the project window.



The resulting view shows a schematic of Waller Creek from near Highland Mall to the Colorado River. This is the main geometric data-editing window. The tick marks and corresponding numbers denote individual cross-sections. Choices under the **View** menu provide for zoom and pan tools. The six buttons on the left side of the screen are used to input and edit geometric data. The 

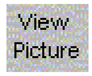
and 

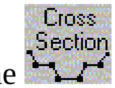
buttons are used to create the reach schematic. A reach is simply a subsection of a river, and a junction occurs at the confluence of two rivers. Since our reach schematic is already defined, we have no need to use these buttons. The 

, 

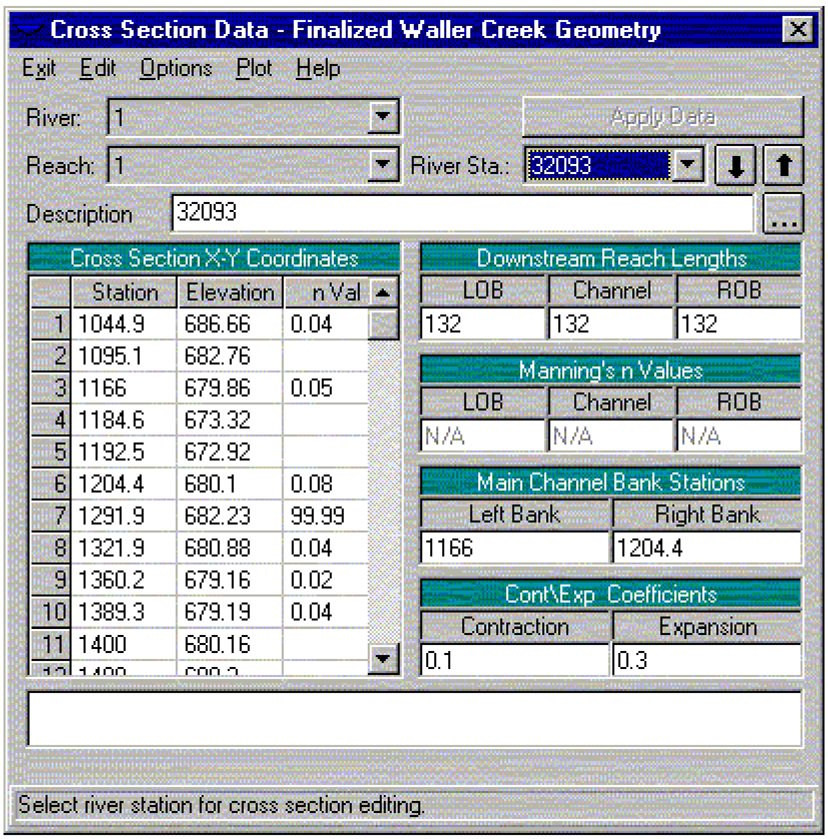
, and 

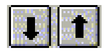
buttons are used to input and edit

geometric descriptions for cross-sections, and hydraulic structures such as bridges, culverts, and weirs. The 

allows you to associate an image file (photograph) with a particular cross-section. Click on the 

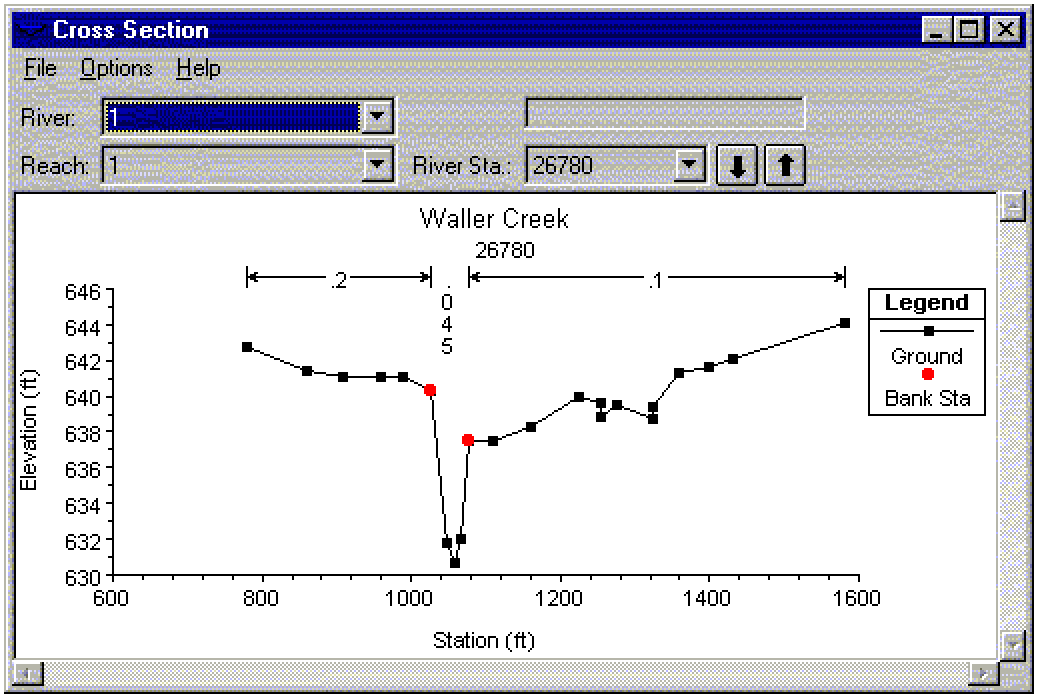
button to open the cross-section data window:

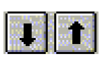


The data used to describe the cross-sections include the river station/cross-section number (32093 in the figure), lateral and elevation coordinates for each terrain point (station & elevation columns), Manning's roughness coefficients (nVal), reach lengths between adjacent cross-sections, left and right bank station, and channel contraction and expansion coefficients. These data are typically obtained by field surveys. The  buttons can be used to toggle between different cross-sections. Use them to scroll to cross-section 26780. To edit data, simply double-click on the field of interest. As an example, double-click on station 779, change the value to 778, and hit the enter key. You may notice that this action caused all of the data fields to turn red and it enabled the "Apply Data" button. Whenever you see input data colored red in HEC-RAS, it means that you are in edit mode. There are two ways to leave the edit mode (you can do whichever you like):

1. Click the "Apply Data" button. The data fields will turn black, indicating you're out of edit mode, and the data changes are applied.
2. Select **Edit/Undo Editing**. You'll leave the edit mode without changing any of the data.

To actually see what the cross-sections look like, select the **Plot/Plot Cross-Section** menu item.



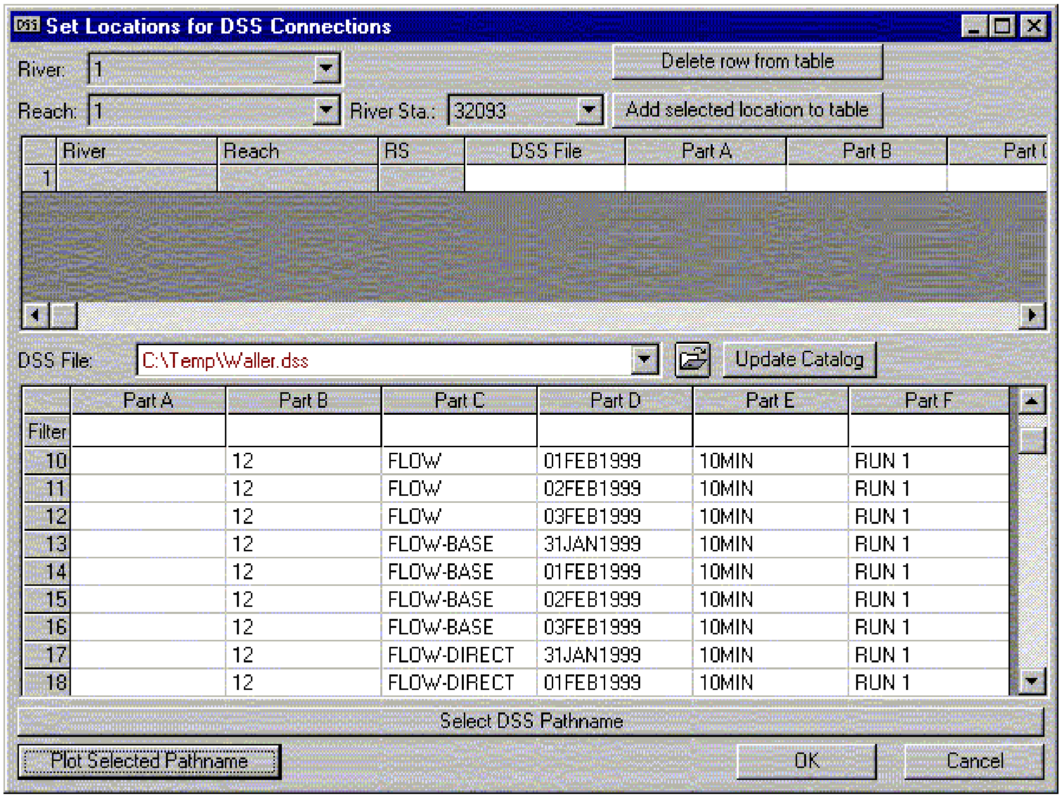
The cross-section points appear black and bank stations are denoted with red. Manning roughness coefficients appear across the top of the plot. Again, the  buttons can be used to maneuver between different cross-sections. Any solid black areas occurring in a cross-section represent blocked obstructions. These are areas in the cross-section through which no flow can occur. Some cross-sections contain green arrows and gray areas. This symbolism is indicative of the presence of a bridge or culvert. Input data and plots specifically associated with bridges and culverts can be accessed from the main geometric data editor window by clicking on the 

button. Take a little time to familiarize yourself with the geometric data by flipping through some different cross-sections and bridges/culverts. When you are finished, return to the geometric editor window and select **File/Save Geometric Data**. Return to the main project window using **File/Exit Geometry Data Editor**. At this point, save your HEC-RAS project just in case the program crashes for some reason or another.

Importing and Editing Flow Data

Enter the flow editor using **Edit/Steady Flow Data** from the main project window. Instead of importing an existing HEC-RAS flow file, which is also an option, we'll use stream flow output from an HEC-HMS model run similar to the one previously completed for the Introduction to HMS Exercise (http://www.ce.utexas.edu/prof/maidment/gradhydro99/hmwk1/hmsintro.htm). The resulting flows are based on the 100-year design storm on Waller Creek, between its junction with the Hemphill Branch, and the Colorado River.

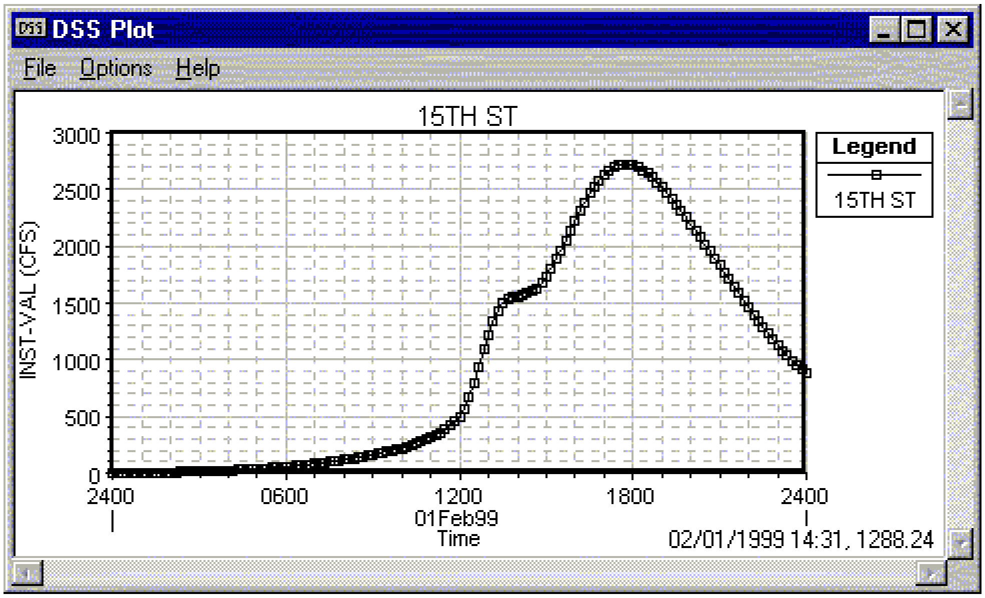
Output data from the HEC-HMS model are stored in files with a .dss extension. DSS stands for the HEC **D**ata **S**torage **S**ystem, which is essentially a database for storing time-series information. To use these data, select **File/Set Locations for DSS Connections** from the main flow data window. To open the DSS file, click on the  button and select the Waller.dss file from your working directory. The window should now look like this:



The DSS data are stored in table records, each one representing a 24-hour increment of time-series flow data. Each record is described by several parameters, some of which are shown in the columns titled Part A, Part B, etc., as follows:

Column	Description
A	Unknown
B	HMS hydrologic element (subbasin, junction, etc.) identifier
C	Flow type (baseflow, floodflow)
D	Date
E	Model Time Step
F	HMS Run ID

HEC-RAS allows you to view the hydrograph of any DSS record. Since the highest flows for our model run occur on February 1, we'll concentrate on the data from this day. Click on any record with Column C = FLOW and Column D = 01FEB1999 and then click on Plot Selected Pathname to see the associated hydrograph:



The coordinates of the cursor (time,flow rate) are displayed in the bottom right corner of the plot. Gridlines can be shown by invoking the **Options/Grid** menu item. Exit the plot window and return to the "Set Locations for DSS Connections" window. We're now going to link the HEC-RAS cross-sections with their calculated DSS flows from HEC-HMS. The following table shows the relationship between the junctions in the HEC-HMS basin model and cross-sections in the HEC-RAS geometry file:

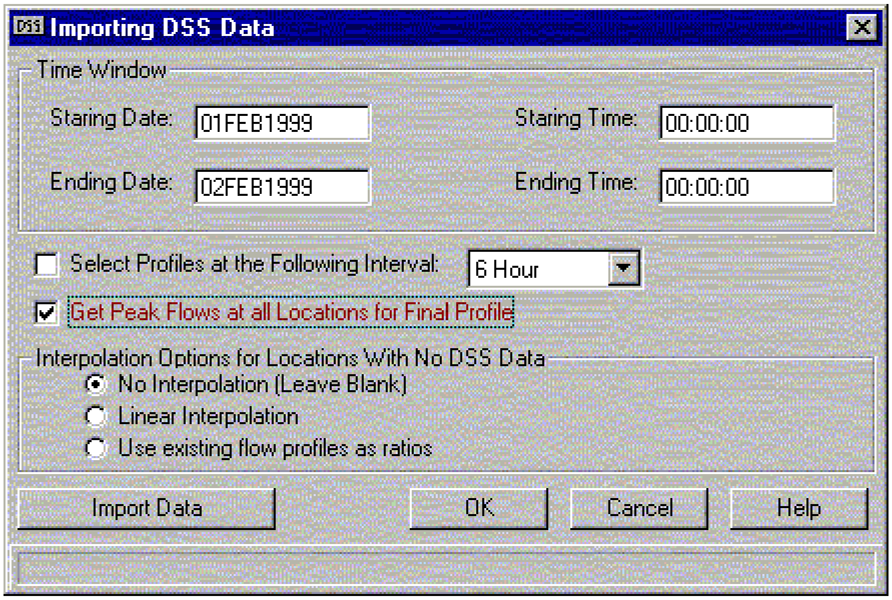
HEC-HMS Junction	HEC-RAS Cross-Section
Junction with Hemphill Branch	12609
MLK Blvd	8916
15th Street	7089
7th Street	3591

1st Street	1157
Colorado River	0

The procedure for linking the DSS records with their associated cross-sections is as follows:

1. Choose the river station from the drop-down list in the “Set Locations for DSS” window
2. Click on the button "Add selected location to table"
3. Click on the DSS record in which the Part B corresponds to the selected cross-section. Ensure that Part B column reads "FLOW" and Part C says "01FEB1999". Click on "Select DSS Pathname" to link the data.
4. Repeat for each of the junctions (click OK when finished with all junctions).

After the DSS records for the six junctions have been set, return to the main steady flow data window and select **File/DSS Import**. Fill in the fields as shown below:



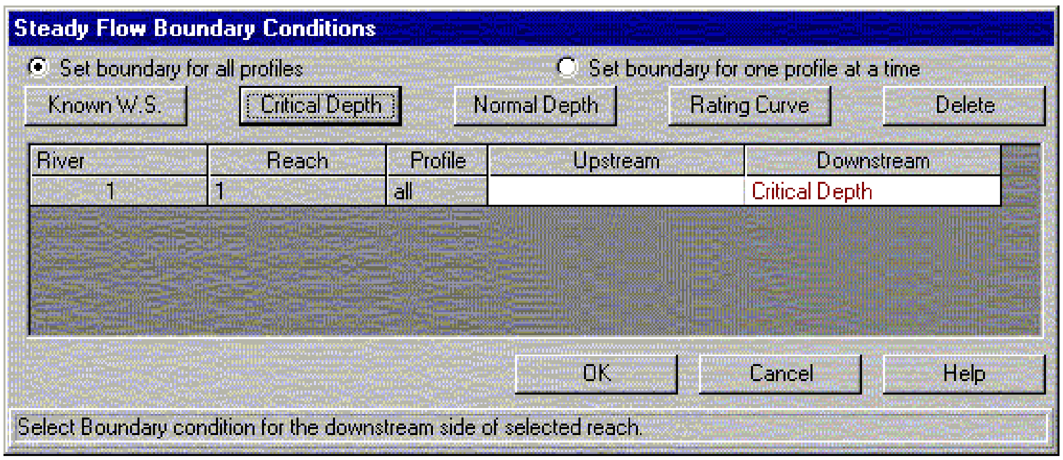
Click on the "Import Data" button, and the flows from HEC-HMS will be imported into your HEC-RAS model. As discussed earlier, the direct step method uses a known water surface elevation (and several hydraulic parameters) to calculate the water surface elevation at an adjacent cross-section. We'll assume a subcritical flow regime for our model, so the computations will begin at the downstream end. As such, the water surface elevation at the downstream boundary must be known. To establish this value, click on the 

button from the steady flow data window. HEC-RAS allows the user to set the water surface elevation boundary condition by four methods:

- Known water surface - based on observed data
- Critical depth - the program will calculate critical depth

- Normal depth - the program will calculate normal depth
- Rating curve - elevation determined from an existing stage-discharge relationship curve

For this tutorial, we'll use the critical depth option. Click in the box in the "Downstream" column and then click on the critical depth button.

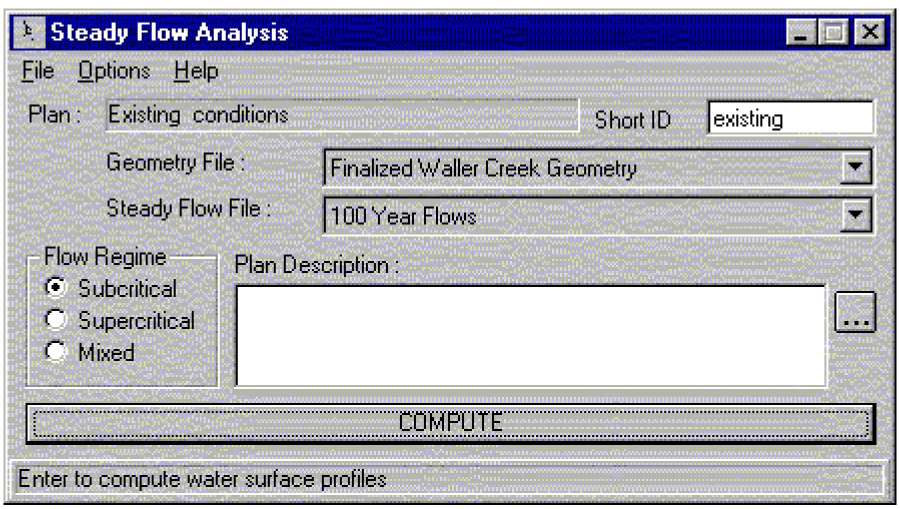


Click OK to return to the main steady flow window. You'll notice that each of the junctions have now been assigned peak flow values from the HMS DSS output. For cross-sections falling between HMS junctions, the flow value of the upstream junction is applied. However, the most upstream cross-section, number 32093, hasn't been assigned a flow value. You'll need to manually input a number here, but its magnitude is really inconsequential because the computations will proceed from downstream to upstream (subcritical flow). And for this tutorial, we are mainly interested in water surface profiles between U.T. and the

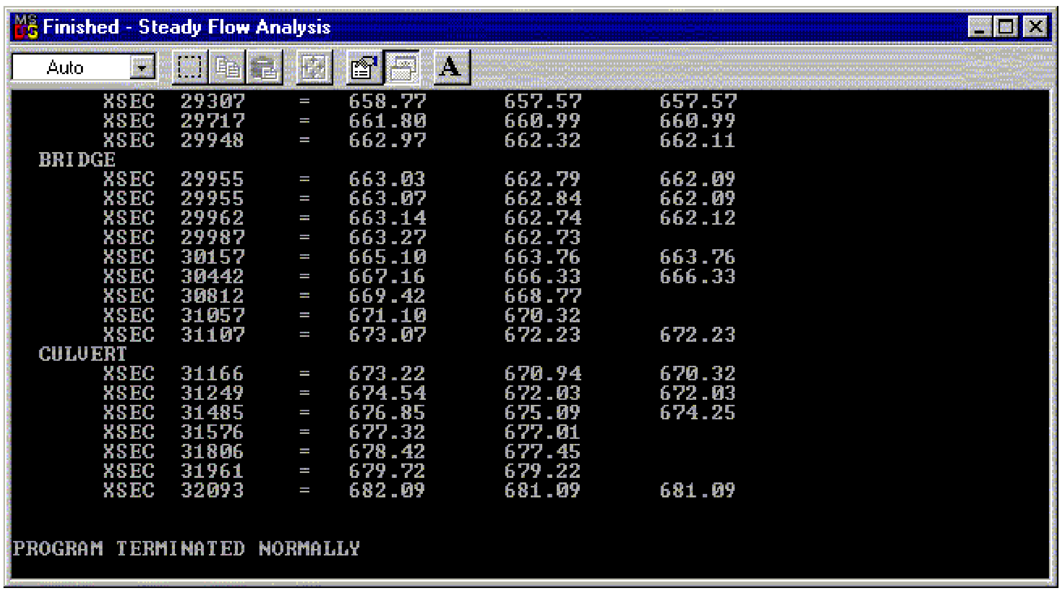
Colorado River. Input a value of 2700. All of the required flow parameters have now been entered into the model! From the file menu, select **Save Flow Data** and save the flow data under the name "100 year flows." To leave the flow data editor and return to the HEC-RAS project window, choose **File/Exit Flow Data Editor**.

Executing the Model

With the geometry and flow files established, the HEC-RAS model can be executed. Select **Simulate/ Steady Flow Analysis** from the project window. But before running the model, one final step is required: definition of a plan. The plan specifies the geometry and flow files to be used in the simulation. To define a plan, select **File/New Plan**. You'll be subsequently asked to provide a plan title and a 12 character short identifier.



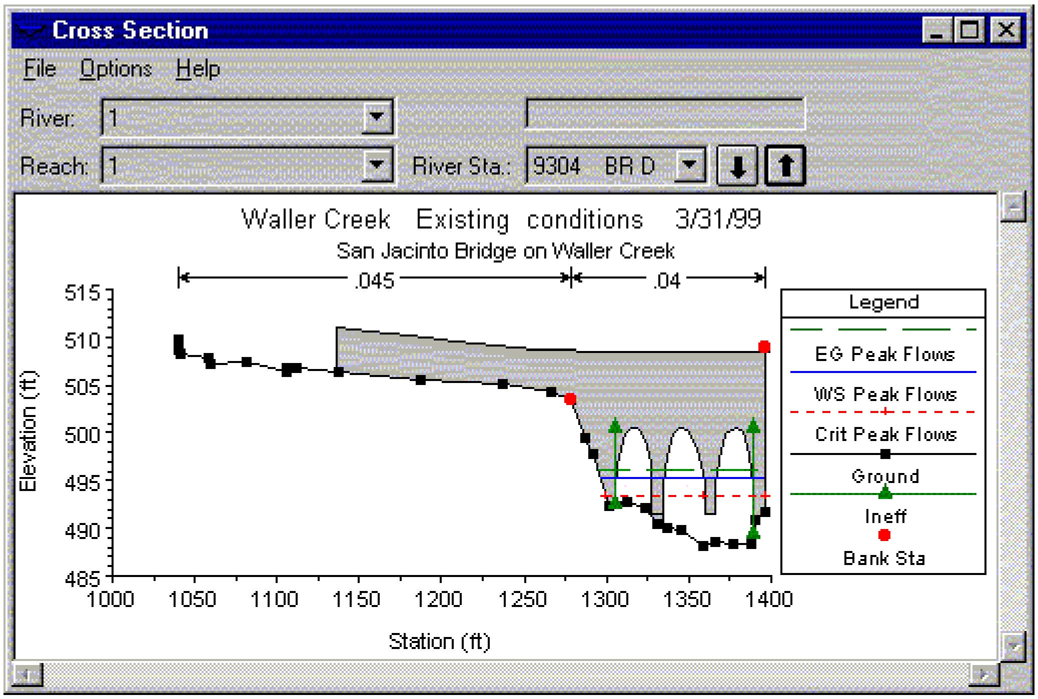
To execute the model, first ensure that the flow regime radio button is set to "Subcritical" and then click the compute button. All of the HEC-RAS windows you've used to this point are simply graphical user interfaces used to input data for the model. The computations are actually performed by a FORTRAN program named SNET. Clicking the compute button starts SNET and opens a DOS window that shows the progress of the simulation. When the computations are complete, the **PROGRAM TERMINATED NORMALLY** message should appear.

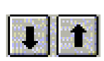


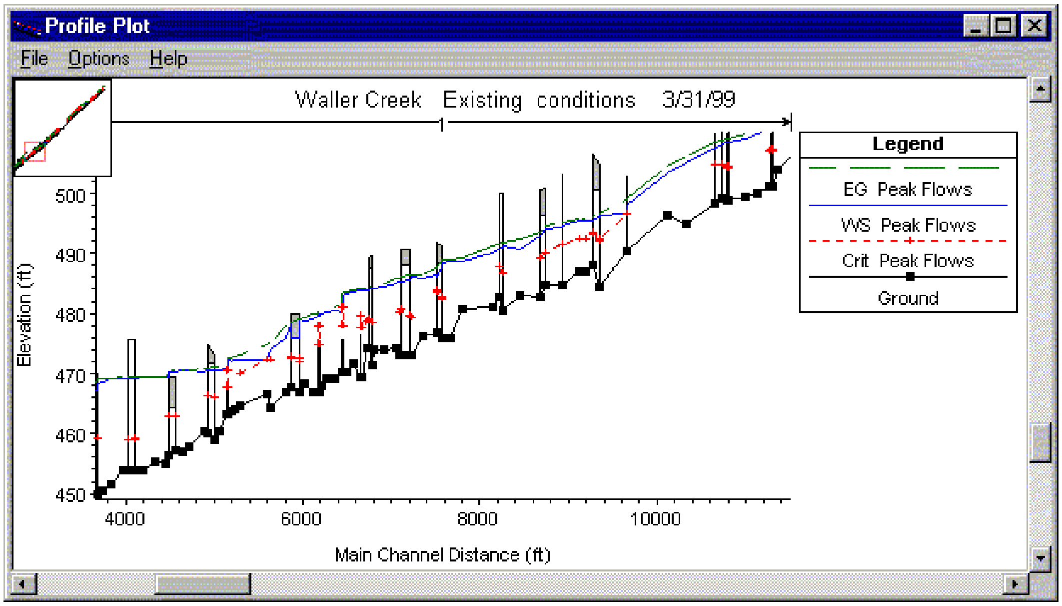
Dismiss the DOS window by clicking the X in the upper right corner.

Viewing the Results

There are several methods available with which to view HEC-RAS output, including cross-section profiles, perspective plots, and data tables. From the project window, select **View/Cross-Sections**.

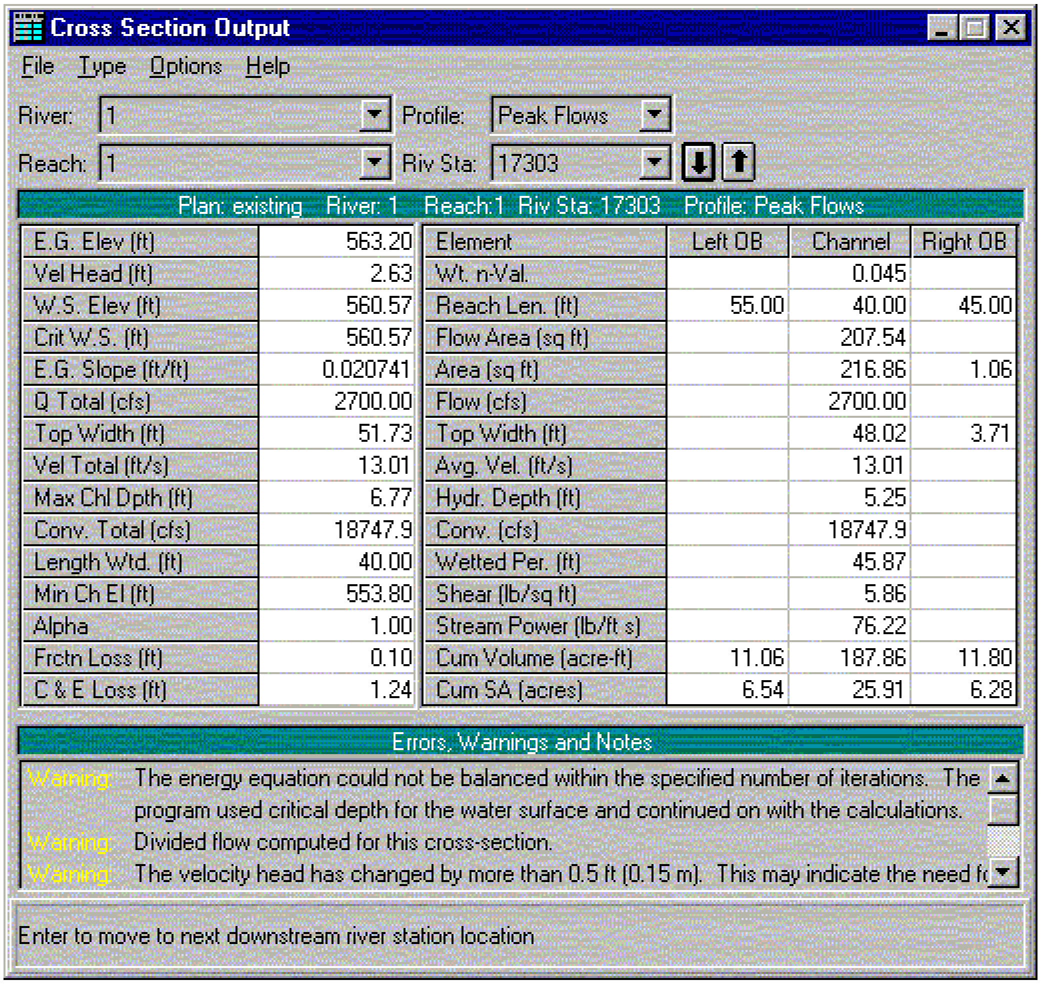


The cross-section view is similar to the one shown when we edited the cross-section data. However, the output view also shows the elevation of the total energy head line (shown in the legend as "EG Peak Flows"), the water surface ("WS Peak Flows"), and critical depth ("Crit Peak Flows"). As with the cross-section geometry editor, you can use the  to scroll to other cross-sections. For a profile of the entire reach, select **View/Water Surface Profiles** from the project window.



Using the **Options/Zoom In** menu option, you can focus on a particular stretch of reach to see how the water surface relates to structures in the channel such as bridges. Other available options for graphical display of output data include plots of velocity distribution (**View/Cross-Sections/Options/Velocity Distribution**) and pseudo 3D plots (**View/X-Y-Z Perspective Plots**). Spend a little time playing around with some of the display options.

For hydraulic design, it is often useful to know the calculated values of various hydraulic parameters. HEC-RAS offers numerous options for tabular output data display. From project window, choose **View/ Cross Section Table**.



The resulting table includes a number of hydraulic parameters, including water surface elevation, head losses, and cross-sectional area. At the bottom of the window, error and notes (if any) resulting from the steady flow computations are shown. As you scroll through the cross-sections, take a look at some of the error messages. For our model, it looks like the primary areas of concern is too few cross-sections. Additional tabular output data can be accessed from the invoking

View/Profile Table from the main project window. Numerous formats and data types can be viewed by selecting different tables from the **Std. Tables** menu.

Welcome to the end of what I hope gave you more of an insight to hydraulic river modeling. Go ahead and close HEC-RAS by selecting **Exit** from the **File** menu in the main project window.

Floodplain Mapping and Terrain Modeling
Using HEC-RAS and ArcView GIS

Prepared by Eric Tate
Center for Research in Water Resources
April 1999

Table of Contents

Introduction

For several years, the Hydrologic Engineering Center (HEC) has distributed the HEC-RAS hydraulic modeling package. HEC-RAS is designed to aid in the determination of water surface profiles associated based on inputs describing the channel shape, hydraulic parameters, and flow. However, a consistent deficiency of hydraulic models such as HEC-RAS has been their inability to connect the information concerning the water profiles with their physical locations on the land surface. Often the computed water surface elevations must be manually plotted in order to delineate floodplains. This is a task that if automated, would result in significant resource savings.

A few commercial software companies have capitalized on this opportunity. BOSS International currently distributes RiverCAD software that allows users to display HEC-RAS output in AutoCAD. However, it is our understanding that Geographic information systems (GIS) offer a superior environment for this type of work. Although CAD is a good environment for visualization, GIS provides tools for more complex queries, storage, mapping, analysis, and visualization of spatial data. Recognizing the advantages of GIS, Dodson & Associates, Inc. has developed GIS Stream Pro, with which the user can develop HEC-RAS cross-sections based on a terrain model. After running HEC-RAS, GIS Stream Pro can delineate the floodplain in ArcView GIS. GIS Stream Pro requires the user to input parameters in HEC-RAS, such as Manning coefficients, bridge and culvert descriptions, and channel contraction/expansion coefficients. For a large HEC-RAS model, this could require a large amount of time for model development. But more importantly, most terrain models do not represent streams very well, due to interference from trees and water in the photogrammetry process. So will cross-sections developed from a terrain model contain sufficient resolution for hydraulic modeling in HEC-RAS?

Many practicing engineers already have established HEC-RAS models for floodplain analysis. The tools described in this tutorial offer a method by which to quickly post-process HEC-RAS output, to allow floodplain visualization and analysis in ArcView GIS. The method can also be used develop a terrain model with a density of points in the channel sufficient for hydraulic modeling. This terrain model could then be used as an input for software such as GIS Stream Pro.

Goals of the Exercise

The primary goal of this exercise is to introduce you to automated floodplain mapping using HEC-RAS and ArcView GIS. By the end of this exercise, you should be able to:

- Import HEC-RAS output into ArcView
- Create a digital stream representation
- Combine HEC-RAS and DEM data to develop a three-dimensional terrain model
- Delineate and analyze the HEC-RAS floodplain

Software and Data Requirements

The HEC-RAS program can be downloaded free of charge from the Hydrologic Engineering Center's home page at: http://www.wrc-hec.usace.army.mil/. The program runs on Windows 95 & 98, NT, and Unix platforms. A user's manual is also available at this location. ArcView GIS is a commercial product, and hence, is not free. This exercise requires that you have running HEC-RAS version 2.0, or later, and ArcView version 3.0a or later, with the Spatial Analyst and 3D Analyst extensions. The ArcView Spatial Analyst extension is designed for creating, querying, mapping, and analyzing raster data. The ArcView 3D Analyst extension is intended for creating, analyzing, and visualizing surface data and other 3D data.

The data required for this exercise consist of an HEC-RAS output data file and ArcView shapefiles, which can be downloaded as a zip file: floodmap.zip. The data will be used in this tutorial to construct a digital floodplain map of Waller Creek, located in Austin, Texas. The zip file contains the following data:

- waller.rep - HEC-RAS output text file
- floodmap.apr - ArcView project containing the scripts and menus needed for the floodplain mapping
- roads.shp, roads.shx, roads.dbf - shapefile of Austin streets
- digitize.shp, digitize.shx, digitize.dbf - shapefile of Waller Creek centerline

- polyclip.shp, polyclip.shx, polyclip.dbf - shapefile of study area boundary
- orthophoto.tif, orthophoto.tfw - 1 meter resolution digital orthophotography of the Austin area
- auseast.e00 - digital elevation model in Arc/Info export format
- land.avl, water1.avl, water2.avl - theme color schemes

Select a working directory on your computer and download these files into it. Some notables:

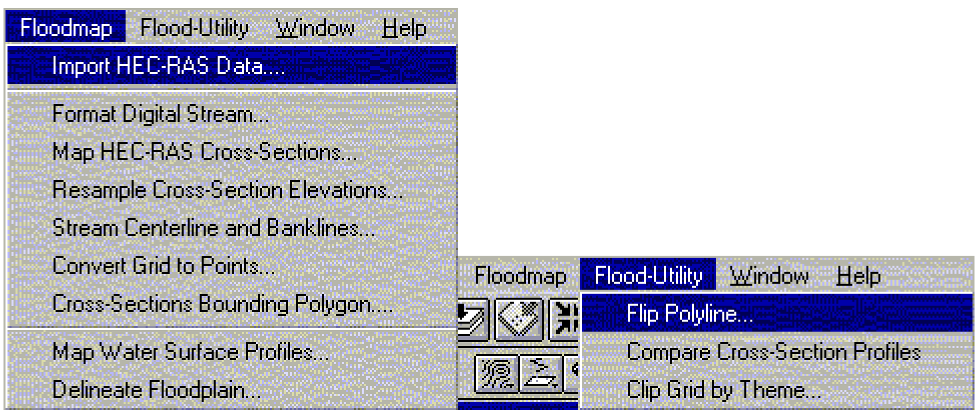
- During this tutorial, you’ll create GIS data that will consume approximately 120 MB of hard drive space. Make sure you have enough room before beginning.

- The procedures described within require use of the digital elevation model, auseast. In order to open in ArcView, you’ll need to convert it from the Arc/Info export format (.e00) to a grid. This can be accomplished using the Import71 function from the Start Menu/Programs/ESRI folder.

Procedure

1. Using the Floodmap Menus

Start ArcView and open the floodplain mapping project floodmap.apr by selecting File/ Open New Project from the main ArcView window. Notice that your working project is now called floodmap.apr. If you double-click on View1 in your project window, a new window, called View 1, will appear. It is in this window where most of the GIS work you are going to do will be accomplished. Before continuing, select the File/Set Working Directory menu option and choose a working directory. Some of the data created during this tutorial will be saved to the working directory. This project is customized with special menu options to help in the floodplain mapping process. The primary menus of interest are the Floodmap and Flood-Utility selections, located on the main-menu bar of ArcView. The following image shows your options when you access these menus:



Summarized descriptions of these menu items are provided as follows:

A. Data Import

- **Import HEC-RAS Data** – Data in the HEC-RAS output file is translated from textfile to dBASE format.

B. Terrain Modeling - These steps are used to produce a triangular irregular network (TIN) terrain model.

- **Format Digital Stream** - The GIS digital representation must be formatted in order to be used in the subsequent steps.

- **Map HEC-RAS Cross-Sections** - Using the HEC-RAS output data and the digital stream, map coordinates are assigned to the HEC-RAS cross-sections.
- **Resample Cross-Section Elevations –** Resamples cross-sections elevations to include DEM elevations.
- **Stream Centerline and Banklines** - Forms a three-dimensional line theme of the stream centerline, right banks, and left banks.
- **Convert Grid to Points** -Raster to vector conversion of a DEM.
- **Cross-Section Bounding Polygon** - Forms a polygon representing the outer boundary of the mapped cross-sections.

C. Floodplain Mapping

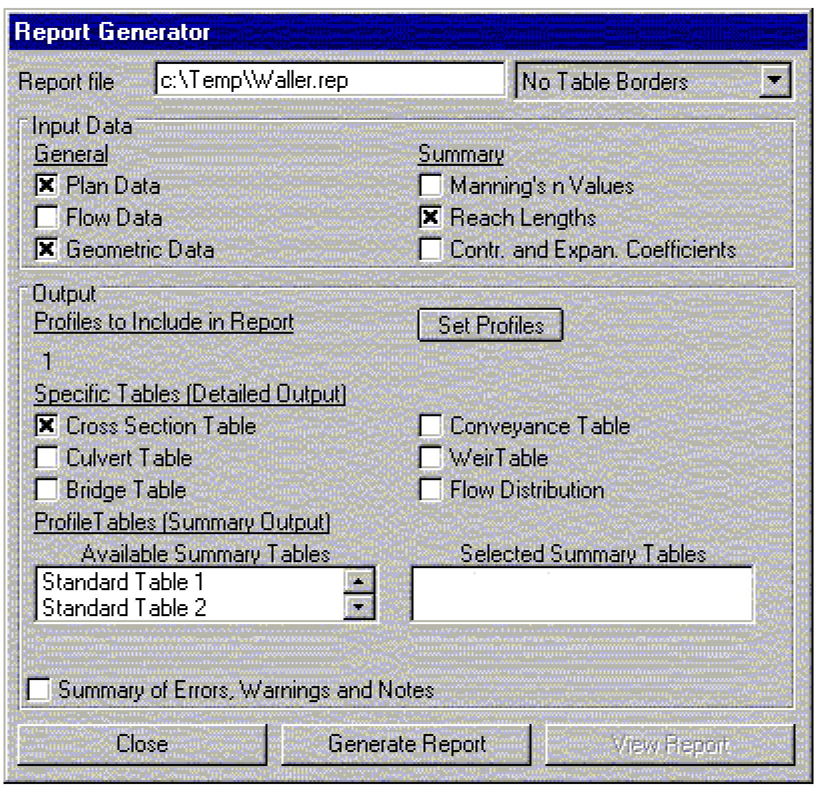
- **Map Water Surface Profiles** - Computed water surface profiles are mapped for each cross-section.
- **Delineate Floodplain** - Areas inundated by flooding are delineated.

D. Utilities

- **Flip Polyline** - Reverses the orientation of a line theme.
- **Compare Cross-Section Profiles –** Creates a text file containing profile data for cross-section visualization.
- **Clip Grid by Theme –** Clips a DEM based on a given polygon theme.

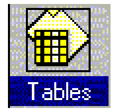
2. Importing HEC-RAS Output

In order to move into the GIS environment, the HEC-RAS output information must first be extracted. Open HEC-RAS. After executing the model, the output text file can be created by invoking the File/Generate Report menu item from the HEC-RAS main project window. When the selection window comes up choose plan and geometric as the general input data, reach lengths under the summary column, and cross-section table under the detailed output option. In addition, it is important that only one profile is selected for the report and the modeled stream has only one branch (this level of complexity cannot be currently handled by the ArcView scripts). If modeling more than one flow profile, the specific profile used for the output report can be selected using the "Set Profiles" button. The report generator window in HEC-RAS should look somewhat like the following graphic:



When you are finished setting up the window, click the "Generate Report" button. HEC-RAS will subsequently create an output file and save it in a location you designate. The resulting output file can be viewed in any text editor. Note the location where you saved the file, and exit HEC-RAS.

Next we will import the HEC-RAS data into our ArcView project. Go to the view window and select the Floodmap/Import HEC-RAS Data menu option. Before continuing, it is necessary to choose the units you'd like to work with. For this tutorial, the HEC-RAS model has units of feet. However, the GIS data (shapefiles, orthophotography, and DEM) are in units of meters. When queried by ArcView for the units, select meters:

Depending on the size of the output file and the speed of your computer, the import process could take anywhere between a few seconds to a couple of minutes. The progress is shown at the bottom of the view window. When the processing is complete, select a file name and location when prompted. Basically, this step transforms the output data from textfile format into dBASE format, which can be read by ArcView as a table. To view the data in ArcView, return to the main project window and click on the 

icon. The table selection window should now show "Table1." Open it and your window should appear something like this:

Table1

Station	Description	Type	FloodElev	LFloodX	LBankX	LBankZ	ChannelX	ChannelZ	RBankX	RBankZ	RFloodX
32093			207.5	14.0	8.1	207.2	40.2	205.1	3.6	207.3	10.5
31961			206.9	8.0	8.6	207.2	87.5	204.8	5.5	206.3	68.2
31806			206.4	4.5	4.9	206.6	157.6	204.2	10.4	206.1	72.2
31576			205.9	9.4	9.9	206.1	185.3	203.5	6.1	205.8	69.6
31485			205.4	9.4	11.0	206.1	257.3	203.1	3.4	205.6	34.7
31249			204.6	5.9	6.1	204.8	282.5	202.5	7.3	205.2	6.8
31166			204.6	3.9	3.9	204.8	300.5	202.6	7.7	204.8	7.7
31136	Denson Drive on Walle	Culvert			3.9	204.8		202.6	7.7	204.8	
31107			204.1	3.8	3.9	204.8	315.8	202.6	7.7	204.8	7.6
31057			204.3	6.5	8.2	205.2	390.4	202.0	5.8	204.3	5.8
30812			203.8	6.6	8.2	204.7	503.2	201.5	5.8	203.8	5.8
30442			202.9	2.3	2.4	203.0	590.1	200.9	10.1	202.5	78.5
30157			202.0	26.4	5.5	201.5	641.9	199.5	7.3	201.9	10.1
29987			201.9	48.9	7.0	201.3	649.5	199.6	6.4	201.3	71.3
29962			201.9	55.0	7.0	201.2	653.8	199.5	6.4	201.2	76.8
29955	Skyview Footbridge on	Bridge			7.0	201.2		199.5	6.4	201.2	
29948			201.8	45.8	7.0	201.2	724.2	199.5	6.4	201.2	70.7
29717			201.4	60.3	7.6	200.9	849.2	198.9	4.6	200.8	52.4
29307			200.2	79.5	7.9	199.8	969.6	198.0	4.0	200.0	9.2
28912			199.3	60.1	7.3	198.7	989.4	197.1	5.2	199.2	13.2
28847			199.3	82.1	7.6	198.8	1000.4	197.2	7.7	199.0	54.8

For each cross-section in the HEC-RAS output file, the following data has been retrieved and stored (the text in parenthesis refers to the table column name):

- cross-section number (Station)
- description of the cross-section, if provided in HEC-RAS (Description)
- type of hydraulic structure (Type)
- water elevation (FloodElev)

- the lateral and elevation coordinates of all cross-section points (stored in a variable, not in the table)
- the width of the left and right floodplain as measured from the stream centerline (LFloodX, RFloodX)
- the distance to the bank stations as measured from the stream centerline (LBankX, RBankX)
- the elevation of the left bank station, stream center, and right bank station (LBankZ, ChannelZ, RBankZ)
- the additive reach lengths between cross-sections starting from the upstream end (ChannelY)

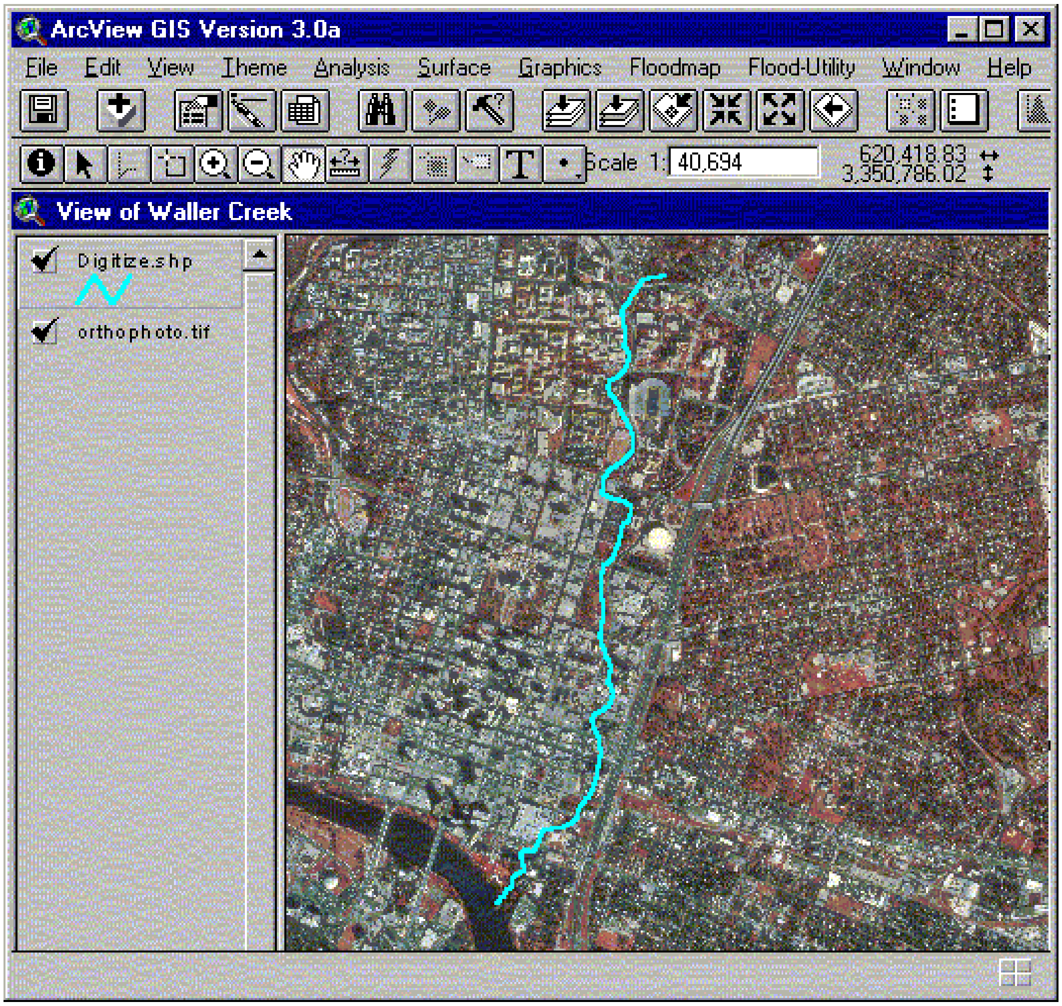
3. Digital Stream Representation

The overall goal of mapping HEC-RAS output is to take cross-sectional and water surface profile data from a one-dimensional model, and transform it into two-dimensional map coordinates. In the previous step, importing HEC-RAS output, the model data was brought into ArcView. What is needed now is the basis for which to assign map coordinates to these data in geographic space. This basis is a GIS representation of the stream centerline. There are four primary sources from which to obtain a digital representation of the HEC-RAS stream:

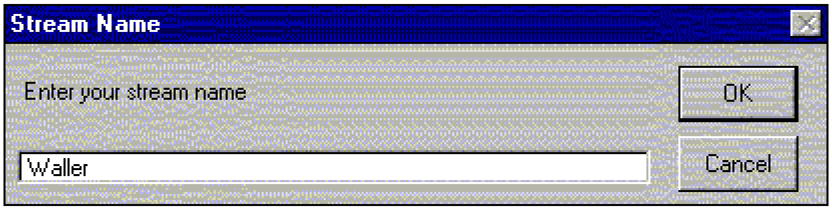
- Survey Data - Data representing the stream centerline may be available from field surveys. If so, this is perhaps the quickest way to generate a vector GIS representation of the stream.

- Reach Files - Reach files are a series of national hydrologic databases that uniquely identify and interconnect the stream segments or "reaches" that comprise the nation's surface water drainage system. The data can be downloaded from the U.S. Environmental Protection Agency (EPA) BASINS website (http://www.epa.gov/OST/BASINS/gisdata.html).
- DEM-Based –A program such as CRWR-PrePro can be used to derive a vector stream representation using a DEM as the sole input. CRWR-PrePro and user instructions can be obtained over the internet from the University of Texas at http://www.ce.utexas.edu/prof/olivera/prepro/prepro.htm.
- Digitize the Stream - Using either a digital orthophotograph (DOQQ) or digital raster graphic (DRG) as base map, the stream can be digitized using tools in ArcView. DRGs and DOQQs for the state of Texas can be obtained from the TNRIS website (http://www.tnris.state.tx.us/digital.htm).

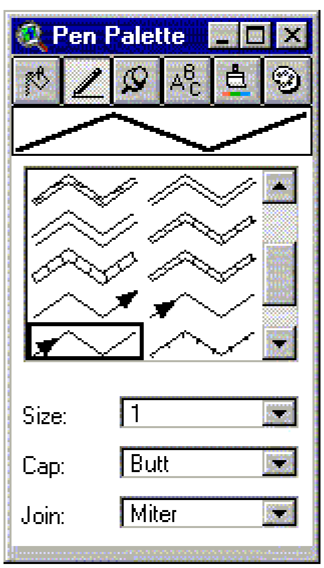
For this tutorial, the stream was digitized in ArcView using a digital orthophoto basemap. This process was easy and fast, but it is probably only feasible if 1-meter (or finer) resolution orthophotography is available. The resulting shapefile is named digitize.shp. To add the theme to the view, click on the , the add theme button. After making sure that the data source type in the add theme window is set to "Feature Data Source", choose digitize.shp. To provide some perspective, also use the  button to open the digital orthophoto, orthophoto.tif. To do so, make sure the data source type in the add theme window is set to "Image Data Source." The view window should now look like this:



Regardless of the source of the stream centerline, the attribute table of the shapefile will need to be modified to allow its use in subsequent steps of the floodplain mapping. To do so, make the digitized line theme active, and select Floodmap/Format Digital Stream. When prompted for the stream name, enter "Waller":



We no longer need the theme digitize.shp, so you may delete it from the view window. At this point, it is important to know the direction of the stream line theme. Although common sense says that it should point upstream to downstream, HEC-RAS defines streams in the opposite direction. To be consistent with the HEC-RAS data, the digital stream should point from downstream to upstream. To check the direction of the line, double click on the theme in the legend bar in order to bring up the legend editor. Click on the pen palette and choose an arrow representation.

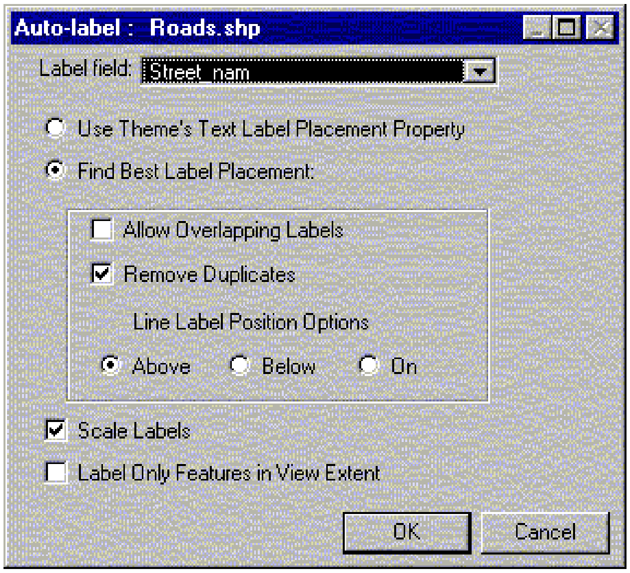


In actuality, Waller Creek flows from north to south. In this case the digital stream is oriented correctly, from downstream to upstream. If this wasn't the case, the direction could be reversed by making the theme active, and selecting Flood-Utility/Flip Polyline.

4. Cross-Section Mapping

The first step in assigning map coordinates to the cross-sections along the stream is to compare the reach lengths according to the RAS stream and their digital counterparts. It is possible, for example, that the digital stream is defined to a point farther upstream than the RAS stream, or vice versa. Hence, it's necessary to delineate the upstream and downstream boundaries of the RAS stream on the digital stream. Intermediate stream definition points corresponding to important RAS cross-sections such as bridges or culverts can also be defined. The importance of the intermediate definition points will be given soon.

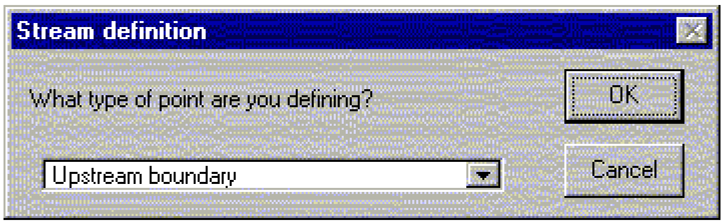
To help define the upstream, downstream, and intermediate points, we'll use a theme of Austin roads in the point delineation process. Click on the  button and add the theme roads.shp. In the legend bar, activate the roads theme and click on its checkbox to view it. Then select Theme/Auto-label. The following window should appear:



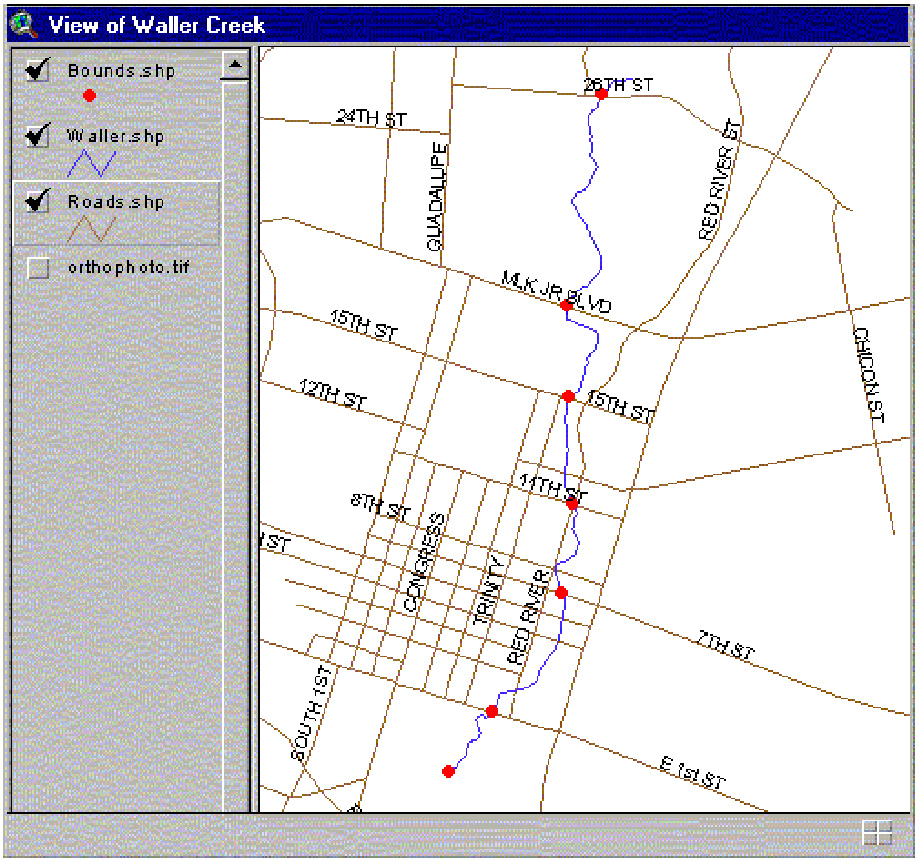
As the label field, select "Street_nam" and click OK. The roads theme will now be labeled with the individual street names. Next activate waller.shp in the legend bar, and click on the  tool. We're now ready to start defining points on the digital stream. The points we will define are as follows:

Location	**Type**
26th Street	Upstream Boundary
MLK Blvd.	Intermediate Point
15th Street	Intermediate Point
11th Street	Intermediate Point
7th Street	Intermediate Point
1st Street	Intermediate Point
End of Waller Creek	Downstream Boundary

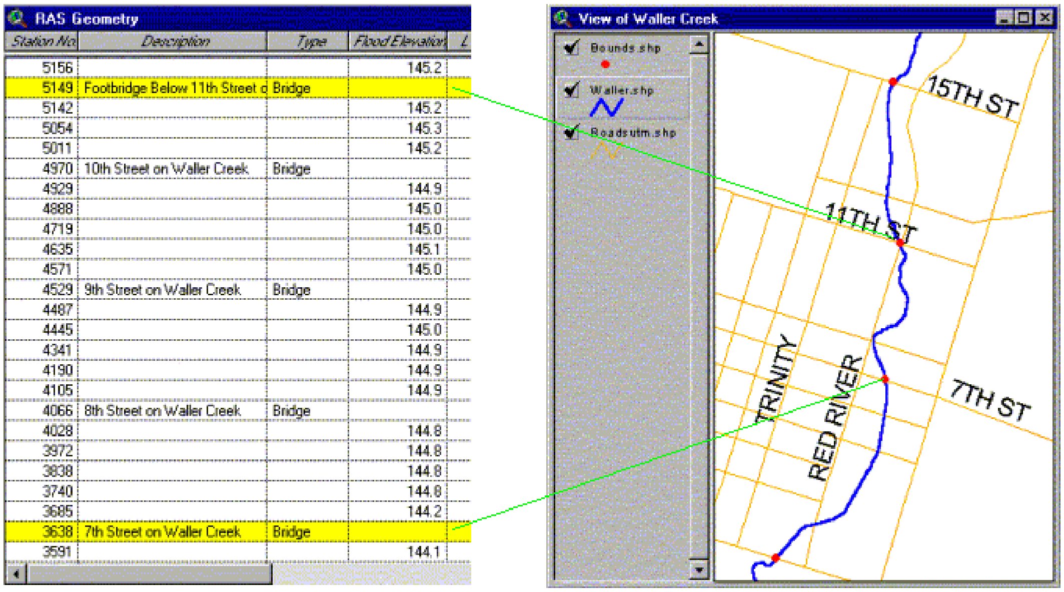
You can display the roads theme and/or the orthophoto to help locate the definition points. But for now, we'll just use the roads theme. Turn off the orthophoto theme in the legend bar. To begin, find the intersection of 26th Street and Waller Creek, and click the mouse at that spot. When queried for the type of boundary, select "Upstream boundary."



The click of the mouse causes the script attached to the  tool to determine the nearest point along the stream centerline and snap the point onto the digital stream. The point shapefile called "bounds.shp" will be added to the view. Next, find the intersection of MLK Blvd. and Waller Creek and define it as an intermediate point. Proceed to define the rest of the points, moving from upstream to downstream. The view window should now look like:



The stream definition points will form the backbone of the cross-section georeferencing process. Using the data table resulting from the data import step, the HEC-RAS data corresponding to the definition point locations will be associated with these map coordinates. For example, the table record for Waller Creek at 11th street will be connected with the definition point located at Waller Creek at 11th street. This concept is illustrated in the following graphic:

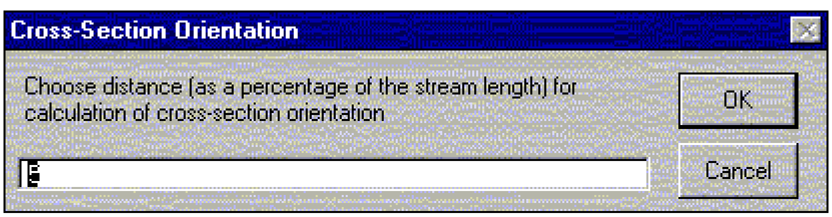


To select the relevant HEC-RAS data, return to Table1. Using the "Station" and "Description" columns as a guide, highlight the record for 26th Street at Waller Creek (Station 13167). Scroll down to select the remaining six records, holding down the shift key to select more than one record.

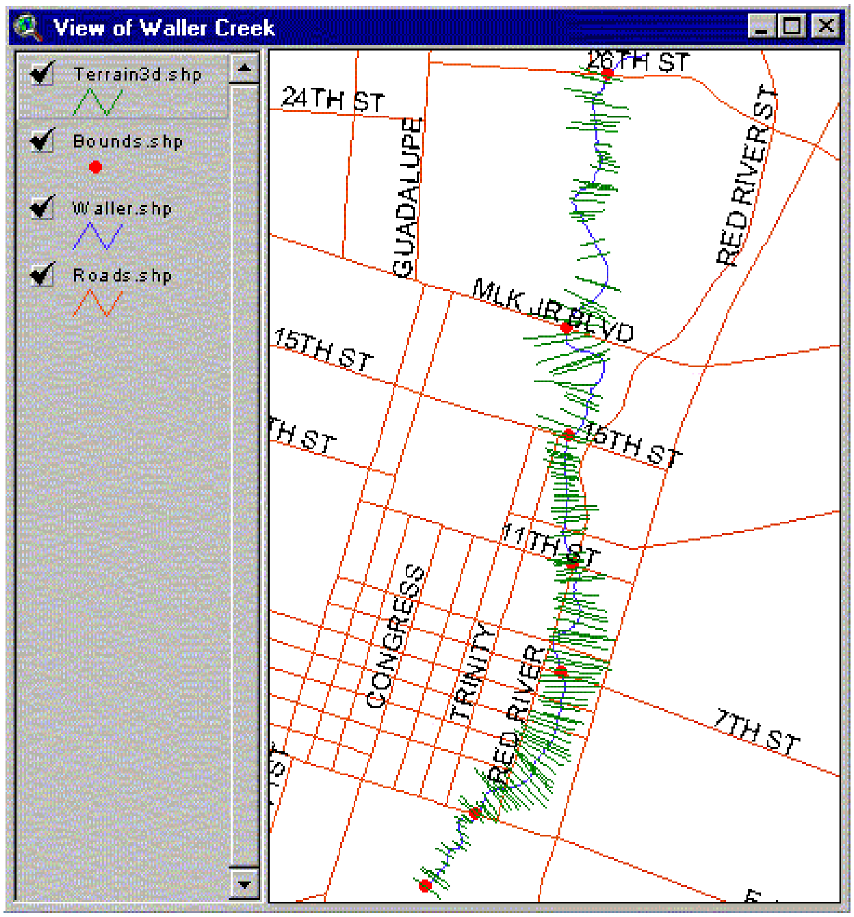
Definition Point	**Station**	**Description**
26th Street	13167	26th Street on Waller Creek
MLK Blvd.	8920	Footbridge Below MLK on Waller Creek
15th Street	7148	15th Street Bridge on Waller Creek
11th Street	5149	Footbridge Below 11th Street on Waller Creek
7th Street	3638	7th Street on Waller Creek
1st Street	1198	Cesar Chavez on Waller

End of Waller Creek	0	---

When finished, click on the tool to promote the highlighted records to the top. Check them to make sure you've selected the correct seven records. Now return to the view window and select Floodmap/Map HEC-RAS Cross-Sections. (*Note: It is important that this step is performed in the same sitting as the data import step. Otherwise, you'll receive the error message "A(n) nil object does not recognize the request get." This is because much of the cross-section data is stored in global variables, which are not saved with the project when exiting ArcView*.) When queried, choose "Bounds.shp" as the stream definition point theme, Waller.shp as the stream centerline theme, and Table1 as the HEC-RAS geometry table. One of the query windows will appear as follows:



This input window is for a parameter that will determine the orientation of the cross-section relative to the stream centerline. Using a value of 0 will cause each cross-section to be mapped perpendicular to the stream, at the precise cross-section location. Using a zero value could cause some cross-sections to intersect near bends in the stream. As the orientation parameter increases, perpendicularity is determined based on a longer segment of the stream. For this tutorial, use the default value of 5.



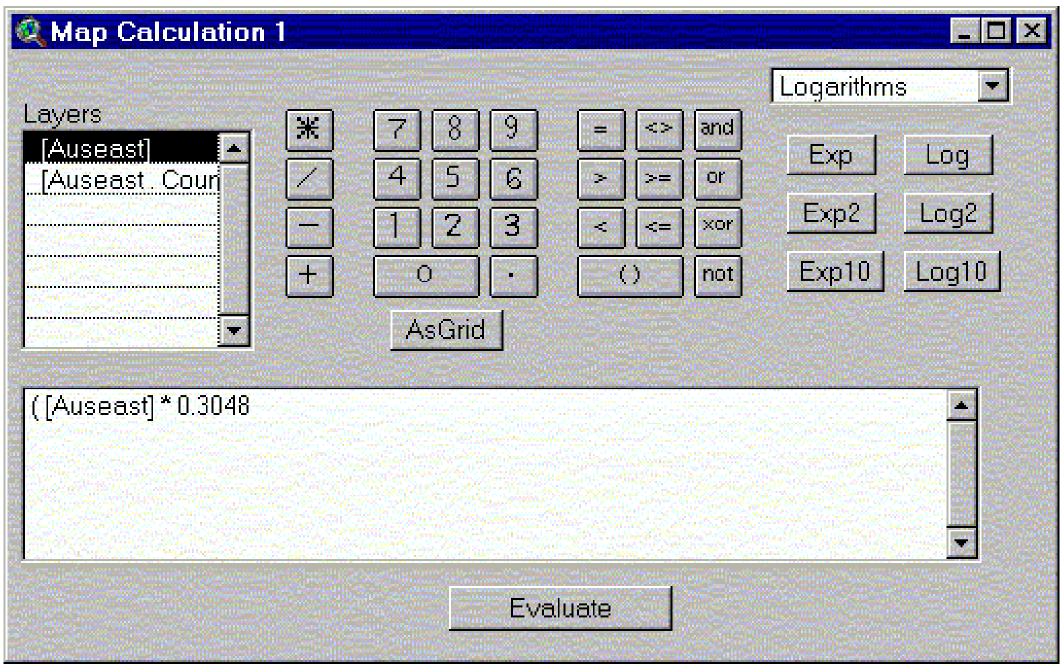
The resulting line theme of the HEC-RAS cross-sections ("terrain3d.shp") should appear like the graphic shown above. You'll notice that each of the cross-sections is represented as a straight line. In reality, some of the cross-sections may have been set up as doglegs. However, this information is not stored in HEC-

RAS, and as such, the straight line assumption was made. At this point, it would be a good idea to save the ArcView project.

5. Terrain Modeling

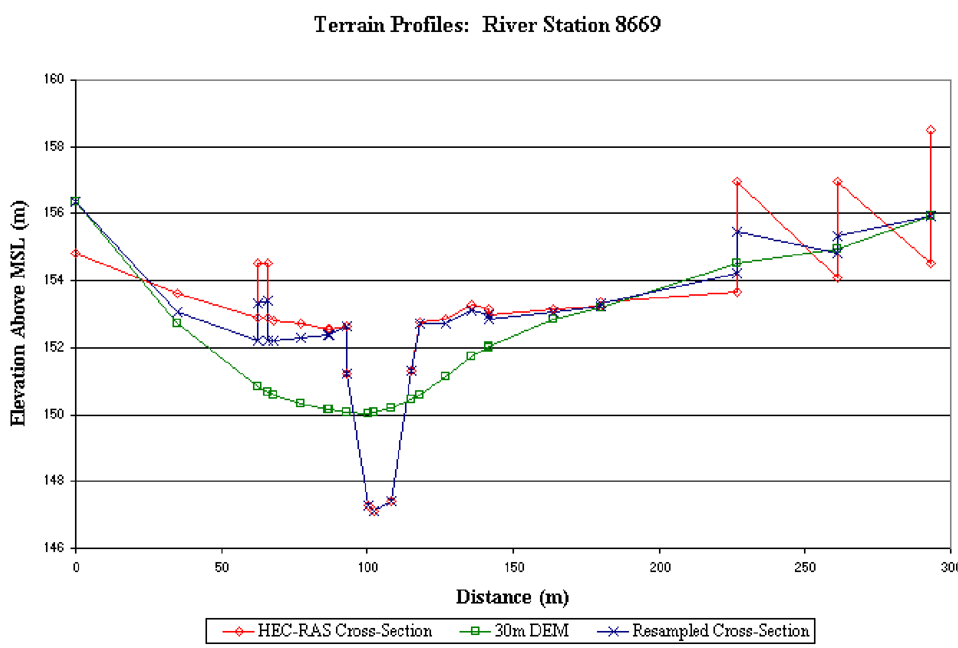
Most available digital elevation models, regardless of the resolution, do not achieve a high level of accuracy or resolution within stream banks. HEC-RAS is a good source of elevation data describing the channel, but typically these data do not possess map coordinates. If the DEM and HEC-RAS data could be combined to form an integrated terrain model, the result would contain elevation information describing the general landscape, with the advantage of extra detail within the stream channel. This integrated terrain model could be used for 3D terrain and floodplain visualization or as an input for software coupling terrain models and hydraulic modeling, such as GIS Stream Pro. The triangular irregular network (TIN) data model will be used to form the integrated terrain model.

Creating a TIN terrain model from two different data sources presents its challenges: the DEM and HEC-RAS data have different collection times, methods, and resolutions. At the point where the HEC-RAS data ends and the DEM data begins, what I call the transition zone, some elevation differences are expected. In order to smooth the transition zone, the elevations between the stream banks and the ends are resampled for each cross-section using elevation values from the DEM. To initiate this process, add the 30-meter DEM to the view. We'll need to do some work on the DEM before continuing. For some reason, many DEMs are constructed with latitude and longitude measured in meters and elevation measured in feet. This is the case with our data set. To remedy this, first select Analysis/Properties. Set both the Analysis Extent and Analysis Cell Size to "Same as Auseast." Click OK and select Analysis/Map Calculator and set up the window as shown below:



After clicking OK, a new grid will be created, this time with the elevations in meters. Highlight the new grid in the legend bar, and save the new grid under the name "auseast1" by selecting Theme/Save Data Set. Although you've given the new DEM a name, the legend bar will still show "Map Calculation 1." You can change the legend bar name by selecting Theme/Properties.

We're now ready to begin the terrain modeling. Select Floodmap/Re-Sample Cross-Section Elevations. When queried, choose "Terrain3d.shp" and "auseast1" as the inputs. This process could take a few minutes--the elevation of every cross-section point outside of the main channel is being recalculated. To help visualize the result of the cross-section resampling, a tool with which to create three profiles of any given cross-section has been developed. Select Flood-Utility/Compare Cross- Section Profiles. Select the appropriate themes and the cross-section (river station) to profile; use river station 8669. The coordinate data is written to an ASCII text file that can be plotted using Microsoft Excel. Save the text file and open it in Excel. The Excel text import wizard should engage. The text file is comma delimited, so click next and in the second window check the comma box, followed by the finish button. With the data now in a spreadsheet, they can be graphed on an x-y scatter plot:



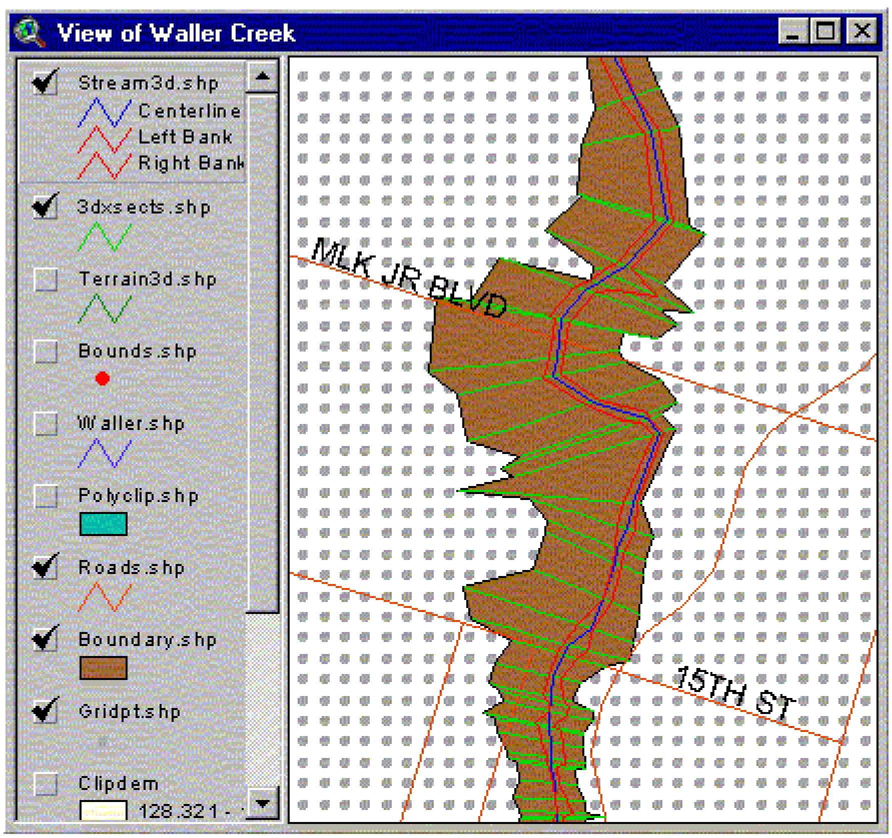
There are two things to be noted about this plot:

1. The resampled cross-section (in blue) is identical to the original cross-section between the stream banks. But in the floodway it begins to approximate the DEM as you move to the end of the cross-section. This ensures a smooth transition from the hydraulic data to the DEM data.
2. By looking at the figure, it is easy to see why 30-meter DEM data is not good enough to use as the basis for hydraulic modeling: the representation of the stream channel does not provide a sufficient level of detail.

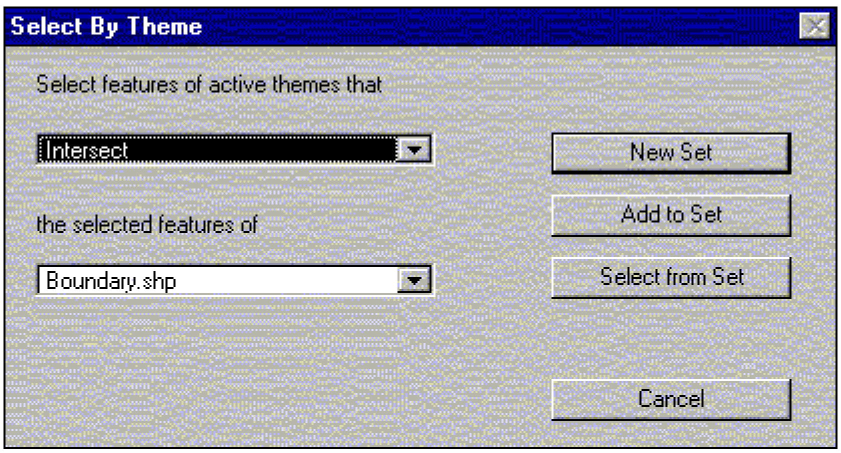
Next we will develop the data that will serve as TIN breaklines: the three-dimensional stream centerline and banklines. Breaklines indicate significant terrain features that represent a change in slope; TIN triangles do not cross breaklines. Select Floodmap/Stream Centerline and Banklines and choose "3dxsects.shp" as the cross-section line theme.

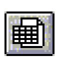
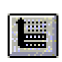
Now we will work on the DEM data. To create a TIN with ArcView 3D Analyst, the input data must be either raster or vector, but not both. So we'll perform a raster to vector conversion on the DEM. Because, the DEM is much larger than our study area of Waller Creek, it will be clipped down to a smaller size. Open the theme "polyclip.shp" from your working directory. The DEM will be clipped to the extent of this polygon theme. Now select the Flood-Utility/Clip DEM menu item, and use "30mDEM" as the grid and "polyclip.shp" as the clipping theme. Now we are ready for the raster to vector conversion. Select Floodmap/Convert Grid to Points to convert the clipped DEM to a point shapefile.

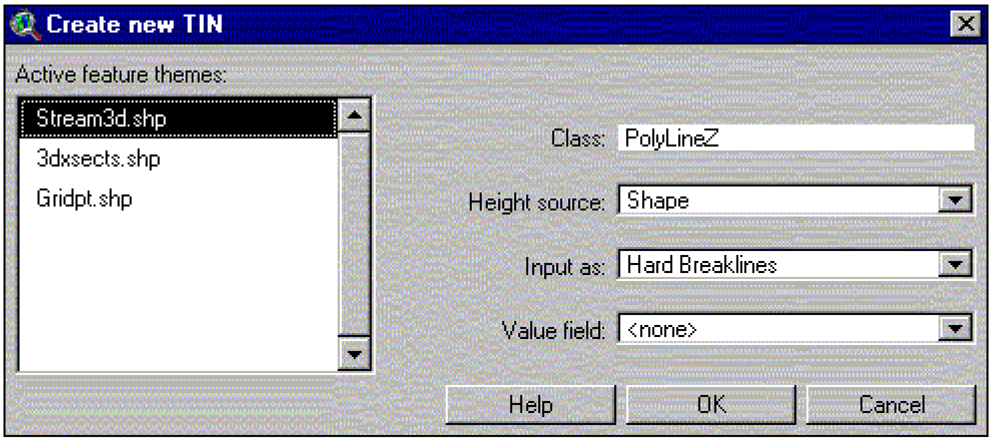
As input for our terrain TIN, the cross-sections will represent areas within the floodplain and the DEM points will provide elevations for the landscape outside the floodplain. A polygon bounding the cross-sections will be used to delineate the transition zone between the two data sources. To form the polygon, select Floodmap/Cross-Sections Bounding Polygon and choose "3dxsects.shp" as the line theme. At this point, the view window should look somewhat like the following graphic:



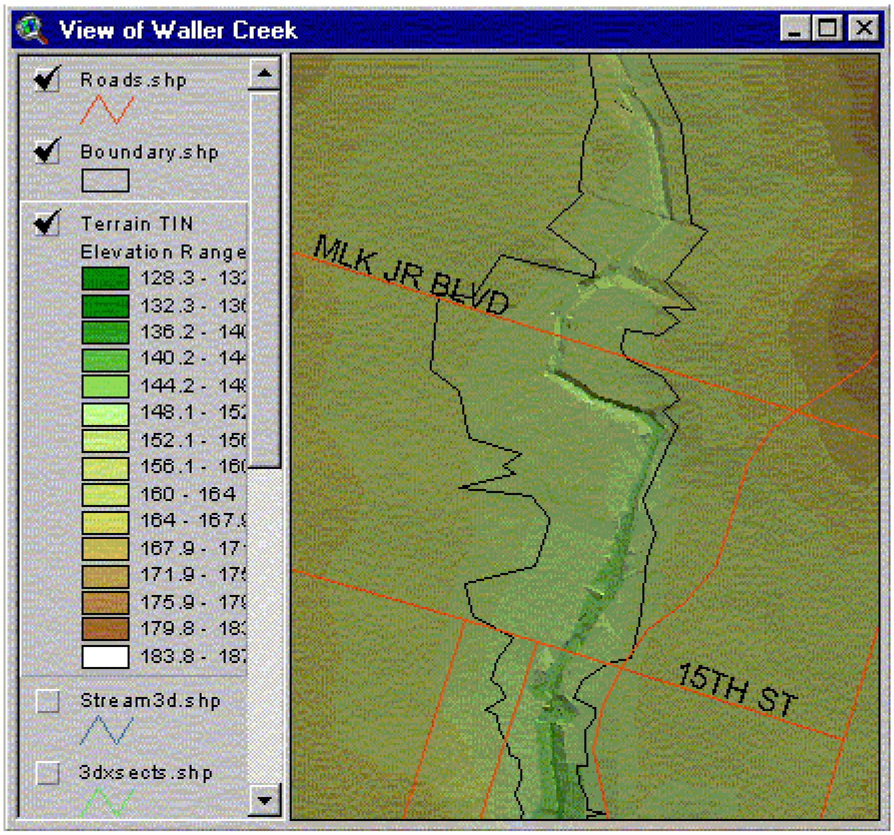
The final step before creating the TIN, is to eliminate any DEM points falling within the cross-section bounding polygon. We can identify these points by intersecting the DEM point theme with the bounding polygon. Make the bounding polygon theme ("Boundary.shp") active and highlight it using the  tool. Next make the DEM point theme ("Gridpt.shp") active and select Theme/Select by Theme. Using the drop-down lists, make the window look like the following graphic and click on the "New Set" button.



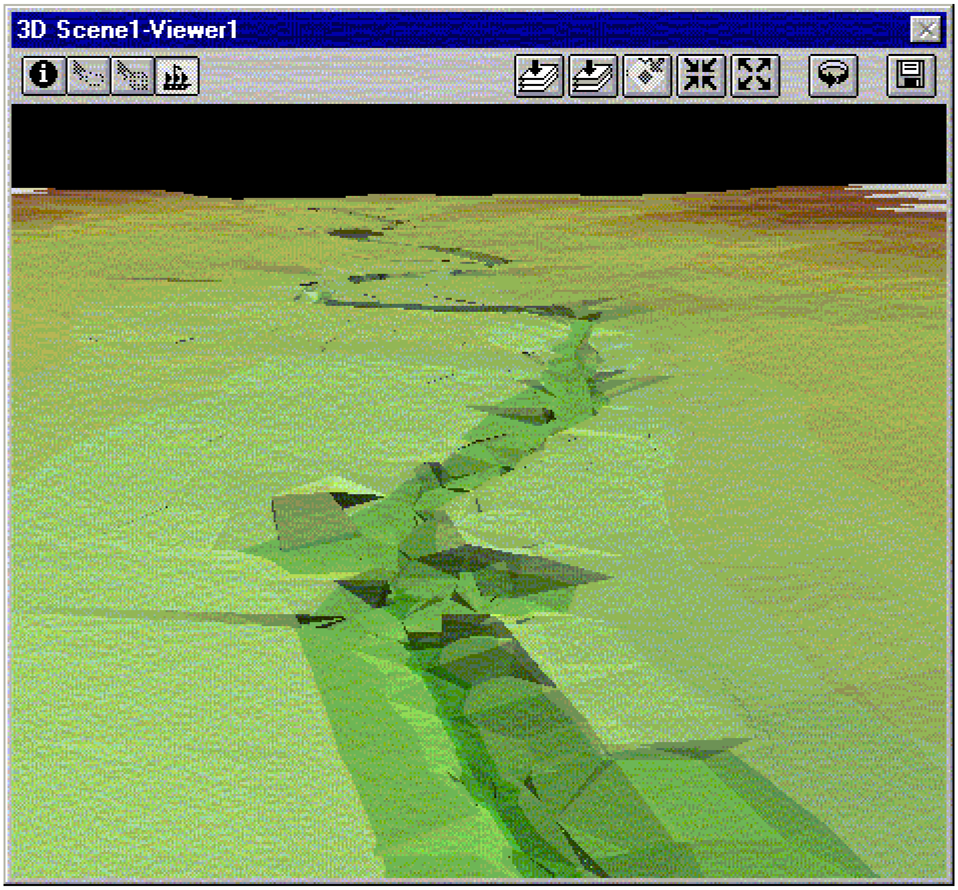
Click on the  button to go view the attribute table of the DEM point theme. In the table window click on the  and all of the selected point records to the top. We'll now edit these points out of the shapefile. Select Table/Start Editing and then Edit/Delete Records. Now select Table/Stop Editing and save the edits. If you now return to the view window, you'll notice that the DEM points only occur outside of the floodplain. We're now ready to create the terrain TIN. In the legend bar, make active the themes "Stream3D.shp", "3dxsects.shp", and "gridpt.shp". Then select Surface/Create TIN from Features and the following window should appear:



The theme "Stream3d.shp" is a three-dimensional line shapefile of the stream centerline and banklines. These data will be enforced in the TIN as breaklines. Fill in the window as shown in the figure. Next, click on the cross-section line theme "3dxsects.shp" on the left side of the window. In the "Input as:" drop down list, select "Mass Points." Finally, click on the DEM point theme, "Gridpt.shp." Select "Mass Points" from the "Input as:" window and "Grid_code" as the "Height Source." Now click the OK button to create the TIN. When ArcView has finished processing the data, the TIN theme will appear in the legend bar. Click the check box to display it. The default color scheme is not ideal for observing the TIN. Double click on the TIN's legend bar to open the TIN legend editor. First click off the check box next to "Lines" and then click the edit button in the "Faces" part of the window. This will open the regular legend editor. I have designed a legend that shows the TIN well. To open it, click the Load button and choose "land.avl." Click OK in the Load Legend" window and then apply in the legend editor. (The image captured from the screen is much grainier than the actual image in ArcView.)



Make the TIN theme active and use the  to query the elevations at different locations. Notice that there is a smooth transition where the hydraulic data (cross-sections) meets the DEM data. To view the TIN in 3D, select View/3D Scene and add the view as Themes when queried. To maneuver in the 3D scene, click and hold the left mouse button while moving the mouse. The time required to render the image will depend on your computer's processor speed, RAM, and video card. The right mouse button can be used to zoom in and out. By clicking and holding both mouse buttons, the pan tool is enabled.

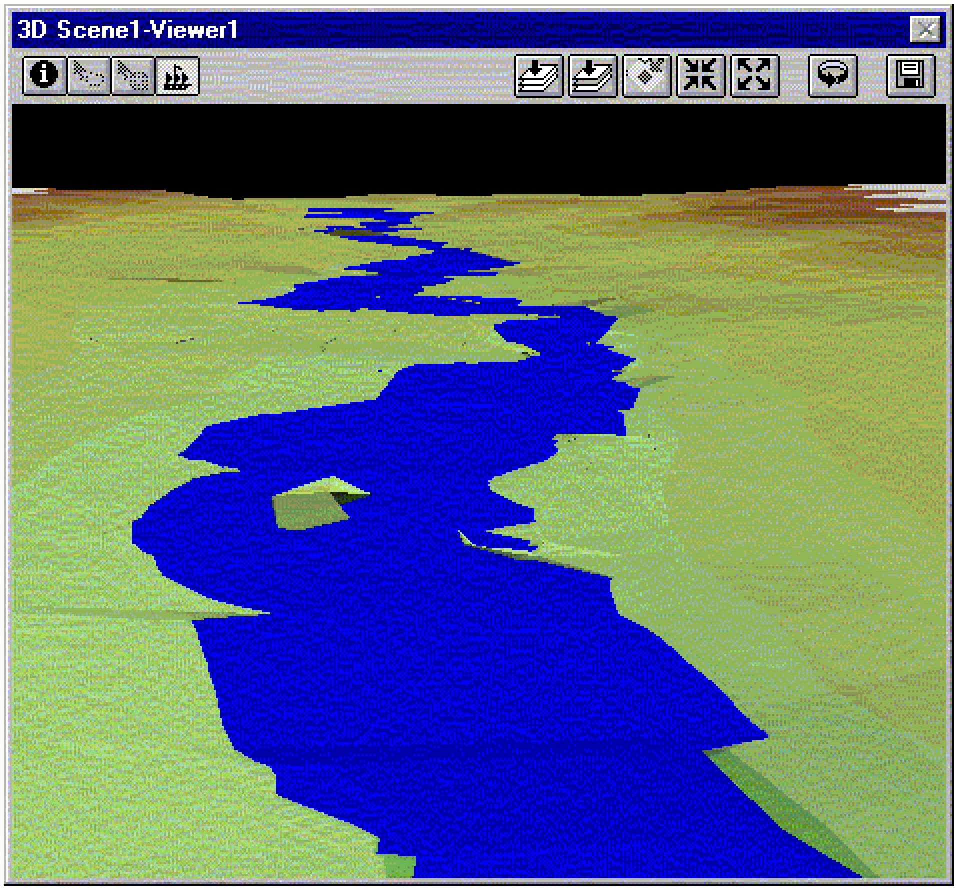


You now have a TIN that depicts the general landscape, but also has additional detail in the stream channel and floodplain. The density of points within the channel is sufficient for hydraulic modeling. A terrain model such as this could be used as an input for software such as GIS Stream Pro.

6. Floodplain Delineation

Areas inundated by flooding occur wherever the elevation of the floodwater exceeds that of the land. To delineate these areas, we'll create surface models of the floodwater and land surface, and then compare the elevations. Let's start with the floodwater model. HEC-RAS represents the floodplain as a computed water surface elevation at each cross-section. During the data import step, these elevations were brought into ArcView, along with the distance from the stream centerline to the left and right floodplain boundaries. Hence, two things are known about the floodplain at each cross-section: water surface elevation and width on each side of the centerline. Select Floodmap/Map Water Surface Profiles to map the water surface info. The data inputs are the original cross-section line theme ("Terrain3d.shp") and the HEC-RAS geometry table ("Table1").

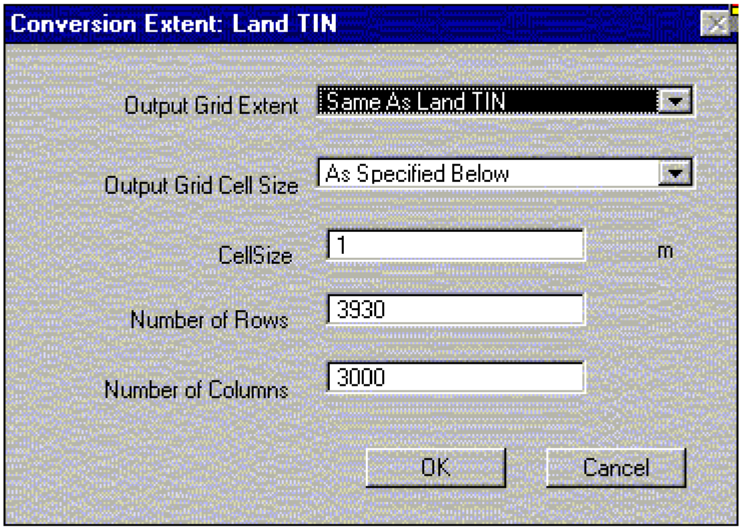
The water surface line theme can be used to create a TIN model of the floodwater surface. If the HEC-RAS model was set up correctly, the water surface extent will not exceed the cross-section extent. As such, the cross-section bounding polygon will serve as the outer boundary of our water surface TIN. Activate the water surface theme ("Water3d.shp") and the bounding polygon theme ("Boundary.shp") in the legend bar and select Surface/Create TIN from Features. In the TIN creation window, input the water surface theme into the TIN as hard breaklines, and the bounding polygon theme as a hard clip polygon and then click OK. The default color scheme really doesn't look much like floodwater, so if you like, you can load a legend named "water1.avl." When viewed in a 3D scene in conjunction with the terrain TIN, flooded areas can be seen:



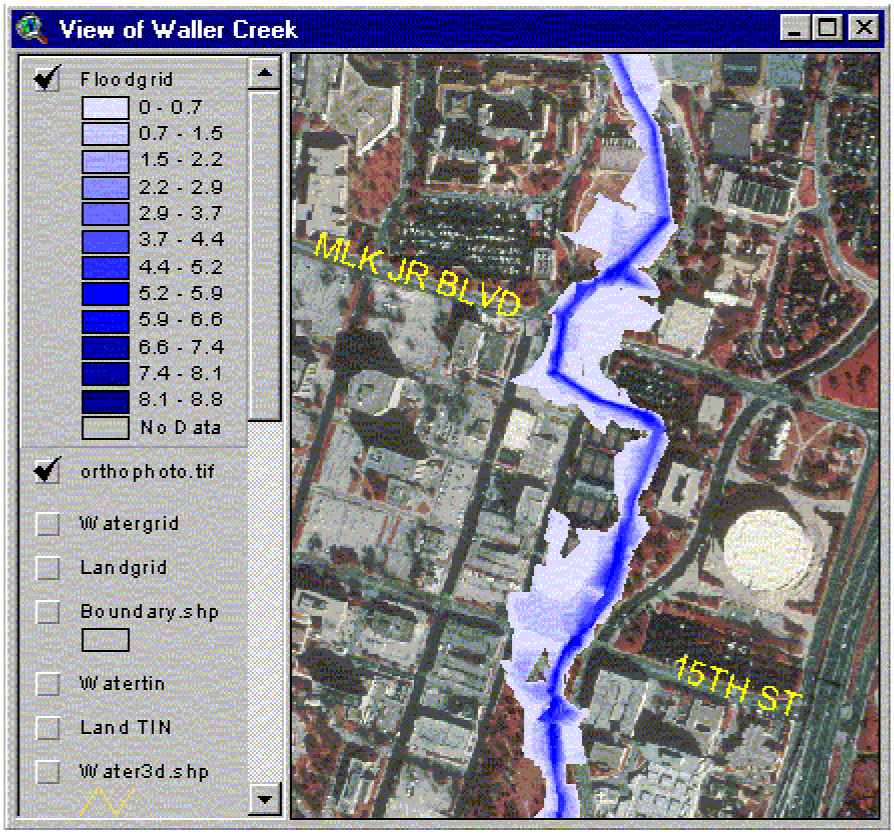
The three-dimensional floodplain view is quite useful for floodplain visualization. But the view shown in the graphic doesn't appear much like the actual landscape. To remedy this, 3D Analyst allows the addition of themes of roads, buildings, railroads, etc., which can be draped over the landscape to more closely approximate reality.

For detailed analysis, the floodplain can be viewed from a planimetric perspective, using a basemap such as a digital orthophotograph. But in order to be accurate, inundated areas should not be delineated using the terrain TIN. Why? The HEC-RAS water surface profiles used to create the water surface TIN were determined using the original cross-sections and not the resampled cross-sections. Hence, a new TIN of the surface terrain needs to be constructed. Activate the themes "Terrain3d.shp", "Stream3d.shp", and "Gridpt.shp" in the legend bar, and select Surface/Create TIN from Features. Specify the parameters just as before in the Terrain Modeling section and click OK. At this point, you can delete resampled cross-section line theme ("3dxsects.shp") and its associated terrain TIN from the view.

Now we have a TIN model for both the land surface and the floodwater. However, the floodplain delineation is more readily performed using the raster data model instead of TINs. In the raster domain, grid cells of water surface elevation and grid cells of land surface elevation can be easily compared using map algebra. As such, we need to convert both TINs to grids. Make the land TIN active and select Theme/Convert to Grid. Choose "landgrid" when prompted for a grid name. In the conversion extent window, set the Output Grid Extent to be the same as the land TIN, set the Output Grid Cell Size to "As Specified Below," and input the cell size as 1 meter:



After clicking OK, ArcView will convert the TIN to a 1-meter resolution grid. Do the same for the water surface TIN. But this time, name the grid watergrid and set the Output Grid Extent to be the same as the water surface TIN. Now both surfaces are represented as grids. Before comparing the two grids, it's necessary to define the analysis extent. Select Analysis/Properties. Set the Analysis Extent to Same as Watergrid, set the Analysis Cell Size "As Specified Below" and input it as 1 meter. Then select Floodmap/Delineate Floodplain. The script will compare the two grids and create an output grid consisting only of areas where the floodwater elevation exceeds the land surface elevation. For the floodplain grid, the color scheme, "water2.avl," can be loaded from the legend editor. Add the orthophoto to the view as a basemap:



You now have a floodplain map, in which the extent of flooding can easily be compared to structures of interest shown on the orthophoto, such as businesses, schools, and homes. In addition, the  tool can be used to query the depth of flooding at any point in the floodplain.

Conclusions

Welcome to the end of what I hope gave you more of an insight of how HEC-RAS and ArcView can be coupled to produce detailed terrain models and floodplain maps. If you use your own data for this process, there are some things you can do to increase the accuracy of your output:

- Decrease the spacing between cross-sections in HEC-RAS
- Increase the number of stream definition points
- Decrease the cell size when converting the land and floodwater TINs to grid

Appendix B: Data Dictionary

Data	Description	Class	Attribute	Units
3097433a.tif	One-meter resolution digital orthophotograph of the Austin East 7.5-minute USGS quadrangle	Image	----	Meters
3DXSects	Line theme of resampled cross-sections	PolylineZ	Station No., Length, Left Bank Pct, Center Pct, Right Bank Pct	Meters
Austineast83	10-meter resolution, drainage enforced DEM of the Austin East 7.5-minute USGS quadrangle	Grid	Elevation	Meters

Data	Description	Class	Attribute	Units
Austin East	30-meter DEM of the Austin East 7.5-minute USGS quadrangle	Grid	Elevation	Meters (northing, easting), feet (elevation)
Bounds	Point theme of stream definition points	Point	Definition type	----
Depth100	Grid of 100-year flood depths	Grid	Water Depth	Meters
Digitize	Line theme of Waller Creek	Polyline	----	Meters
Flood100TIN	TIN of 100-year flood water surface profiles	TIN	----	Meters
Floodgrid	Grid of 100-year storm water surface profiles	Grid	Elevation	Meters
Polyclip	Polygon theme used to clip DEM	Polygon	----	Meters

Data	Description	Class	Attribute	Units
Roads	Line theme of Austin roads	Polyline	Street name	Meters
Stream3D	Line theme of stream centerline and bank stations	PolylineZ	Line type	Meters
Terrain3D	Line theme of stream cross-sections	PolylineZ	Station No., Length, Left Bank Pct, Center Pct, Right Bank Pct	Meters
TerrainTIN	TIN of Waller Creek area terrain	TIN	----	Meters
Waller	Line theme of Waller Creek	Polyline	Name, Length	Meters
Waller.f10	HEC-RAS flow profiles for the 2, 10, 25, and 100 year design storms	Text file	----	Cubic feet per second
Waller.g05	HEC-RAS existing Waller Creek channel geometry	Text file	----	Feet
Waller10grd	Grid of Waller Creek area terrain	Grid	Elevation	Meters

Appendix C: Avenue Scripts

```
'*******************************************************
'Name: AddPnt
'Headline:
'Self:
'Returns: Point Shape file
'Description: Apply Script: Add a point to an existing shapefile or create a new point
                              shapefile defining stream boundaries.
'Topics:
'Search Keys:
'Requires:
'History: Created by Zichuan Ze and Eric Tate
'        Last modification:  3/5/99
'*******************************************************

pntThmName = "Bounds.shp"
pntFileName = FN.Merge(av.GetProject.GetWorkDir.asString,"bounds")

theView = av.GetActiveDoc
themesList = theView.Getactivethemes
themeCount = themesList.count
if (themeCount = 0) then
   msgbox.info("The stream centerline theme should be active","Error")
   exit
end
themeName = themesList.get(0)
rivFtab = themeName.getFtab
rivshpField = rivFtab.findfield("Shape")
theLine = rivFtab.returnvalue(rivshpField,0)
pt = theView.GetDisplay.ReturnUserPoint

boundtypelist = list.make
boundtypelist.add("Upstream boundary")
boundtypelist.add("Intermediate point")
boundtypelist.add("Downstream boundary")

boundtype=MsgBox.ChoiceAsString(boundtypelist,"What type of point are you
defining?","Stream definition")
if (boundtype = nil) then
  exit
end

'IF THIS IS THE FIRST POINT, ADD THE FTAB
if(theView.findTheme(pntThmName)=nil) then
   pntFTab = FTab.MakeNew(pntFileName,Point)
   pntTheme = Ftheme.make(pntFTab)
   pntFields = List.Make
   pntFields.Add(Field.Make("Type",#Field_Char,20,0))
   pntFields.Add(Field.Make("X-Coord",#Field_Decimal,12,6))
   pntFields.Add(Field.Make("Y-Coord",#Field_Decimal,12,6))
   pntFields.Add(Field.Make("AlongPct",#Field_Decimal,12,1))
   pntFieldsC = pntFields.DeepClone
```

```
  pntFTab.addFields(pntFieldsC)
  theView.addTheme(pntTheme)
  if(pntFTab.CanEdit) then
    pntFTab.SetEditable(True)
  else
    msgbox.info("Can't edit the output theme.","Error")
    exit
  end
  pnttheme.setvisible(true)

'IF THIS IS THE 2ND OR GREATER POINT, GET THE FTAB
else
  pntTheme = theView.FindTheme(pntThmName)
  pntFTab = pntTheme.getFTab
  if(pntFTab.CanEdit) then
   pntFTab.SetEditable(true)
  else
   msgbox.info("Can't edit point theme.","Error")
  exit
  end
end
addpnt=nil

shpField = pntFtab.FindField("Shape")
typeField = pntFtab.FindField("Type")
xfield = pntFtab.FindField("X-Coord")
yfield = pntFtab.FindField("Y-Coord")
alongField = pntFtab.FindField("AlongPct")

'SNAP THE POINT TO THE LINE
PntList = theLine.AsMultiPoint.AsList
ccx = pt.getX
ccy = pt.getY
Clength = 0.0

DistList = List.Make  'Dists between each potential snap point and original point.
NPntList = List.Make  'Potential snap-to points.
ClenList = List.Make  'Dists between from node to each potential snap-to points.

if(PntList.count=2)then
 aax = PntList.get(0).getX
 aay = PntList.get(0).getY
 bbx = PntList.get(1).getX
 bby = PntList.get(1).getY

 TLength = (((bbx-aax)^2) + ((bby-aay)^2))^(0.5)
 AC = ((bbx-aax)*(ccx-aax)) + ((bby-aay)*(ccy-aay))
 CLength = AC/TLength  'A.C=A.abs*C.abs*Cos(AC), Clength=C.abs*Cos(AC),
A.abs=TLength
 NewX = aax + ((bbx-aax)*(CLength/TLength))
 NewY = aay + ((bby-aay)*(CLength/TLength))
```

```
  NewPnt = Point.Make(NewX,NewY)
  PFound = true
else
  pcnt = PntList.Count-2
  PFound = False

  '--Start computation loop
   OldAC = nil
   TLength = 0.0
   for each idx in 0..pcnt
     aax = PntList.get(idx).getX
     aay = PntList.get(idx).getY
     bbx = PntList.get(idx+1).getX
     bby = PntList.get(idx+1).getY
     AC = ((bbx-aax)*(ccx-aax)) + ((bby-aay)*(ccy-aay))  'chkingDotProductOf A->C->
     BC = ((aax-bbx)*(ccx-bbx)) + ((aay-bby)*(ccy-bby))  'chkingDotProductOF B->C->
     LAC = ((ccx-aax)*(ccx-aax) + ((ccy-aay)*(ccy-aay))) 'chking Dist between A and C 8/1

     if (LAC < 5) then
       PFound = True
       DistList.Add(LAC.Clone)
       NPntList.Add(PntList.get(idx).Clone)
       CLenList.Add(Clength.Clone)
     end                                    '8/1

     ABL = (((bbx-aax)^2) + ((bby-aay)^2))
     TLength = TLength+(ABL.Sqrt)

     if((AC<0) or (BC<0))then  'segment AB doesn'tContain Point C
       if(OldAC = nil)then
         OldAC = AC
       else
         if((OldAC*AC) < 0)then  'Angle changes from <90 to >90, the end point contains thePntV
           NewX = aax
           NewY = aay
           NewPnt = Point.Make(NewX,NewY)
           ACLength = (((ccx-aax)^2)+((ccy-aay)^2)).sqrt

           if(Not (ACLength.IsNull))then
             DistList.Add(ACLength.Clone)
             NPntList.Add(NewPnt.Clone)
             ClenList.Add(Clength.Clone)
             PFound = True
           end
         else
           OldAC = AC
         end
       end  '(OldAC = nil)
         Clength = Clength+(ABL.sqrt)

     else  'Segment AB contains point C
```

```
      ACLength = (((ccx-aax)^2) + ((ccy-aay)^2)).sqrt
      NewX = aax + ((bbx-aax)*AC/ABL)
      NewY = aay + ((bby-aay)*AC/ABL)
      CLength = Clength+(AC/(ABL.sqrt))   'AC=A dot C
      CosA = AC/(ABL*ACLength)
      SinA = (1-(CosA^2)).sqrt
      Dist = ACLength*SinA

      if(Not (Dist.IsNull))then  'Zye 7/8/97
        DistList.Add(Dist.Clone)
        NewPnt = Point.Make(NewX,NewY)
        NPntList.Add(NewPnt.Clone)
        CLenList.Add(Clength.Clone)
        PFound = True
      end
    end
  end 'endfor each idx
 end 'endif(PntList.count <= 2)

if(PFound.Not)then
  return nil
else
  if(PntList.Count > 2)then
    Nmatch = NPntList.Count
    Ndx = 0
    MinDist = DistList.Get(0)
    NewPnt = NPntList.Get(0)
    CLength = ClenList.Get(0)

    if(Nmatch > 1)then
      for each i in 1..(Nmatch-1)
        tmpDist = DistList.Get(i)
        if(TmpDist < MinDist)then
          MinDist = tmpDist.Clone
          Clength = ClenList.Get(i).Clone
          NewPnt = NPntList.Get(i).Clone
        end
      end
    end
  end
end

pntx = NewPnt.getx
pnty = NewPnt.gety

'WRITE DATA TO THE POINT FTAB
if (boundtype = "Upstream boundary") then
    index = 0
    for each i in pntFtab
        text = pntFtab.returnvalue(typeField,i)
        if (text="Upstream boundary") then
```

```
        index = index+1
      end
    end
    if (index > 0) then
      msgbox.info("The upstream boundary has already been defined.","Error")
    else
      newrec = pntFTab.AddRecord
      pntFtab.SetValue(shpField,newrec,NewPnt)
      pntFtab.SetValue(typeField,newrec,"Upstream boundary")
      pntFtab.SetValue(xField,newrec,pntx)
      pntFtab.SetValue(yField,newrec,pnty)
      pntFtab.SetValue(alongField,newrec,theLine.PointPosition(NewPnt))
    end
  end

  if (boundtype = "Downstream boundary") then
    index = 0
    for each i in pntFtab
      text = pntFtab.returnvalue(typeField,i)
      if (text="Downstream boundary") then
        index = index+1
      end
    end
    if (index > 0) then
      msgbox.info("The downstream boundary has already been defined.","Error")
    else
      newrec = pntFTab.AddRecord
      pntFtab.SetValue(shpField,newrec,NewPnt)
      pntFtab.SetValue(typeField,newrec,"Downstream boundary")
      pntFtab.SetValue(xField,newrec,pntx)
      pntFtab.SetValue(yField,newrec,pnty)
      pntFtab.SetValue(alongField,newrec,theLine.PointPosition(NewPnt))
    end
  end

  if (boundtype = "Intermediate point") then
    exittest = false
    for each i in pntFtab
      xtest = pntFtab.ReturnValue(xField,i)
      ytest = pntFtab.ReturnValue(yField,i)
      if (xtest=pntx) then
        msgbox.info("This point has already been defined.","Error")
        exittest = true
      end
    end
    if (exittest= true) then
      exit
    end
    newrec = pntFTab.AddRecord
    pntFtab.SetValue(shpField,newrec,NewPnt)
    pntFtab.SetValue(typeField,newrec,"Intermediate point")
```

```
    pntFtab.SetValue(xField,newrec,pntx)
    pntFtab.SetValue(yField,newrec,pnty)
    pntFtab.SetValue(alongField,newrec,theLine.PointPosition(NewPnt))
end
pntFTab.SetEditable(false)
```

```
'-----------------------------------------------------------
'-----------------------------------------------------------
' Name: Banklines.ave
' Headline:
' Self:
' Returns:  Shape file
' Description: Takes channel cross-sections and creates a polylineZ
'            theme of the banklines and stream centerline.
' Topics:
' Search Keys:
' Requires: PolylineZ Ftheme of channel terrain.
' History:  Created by Eric Tate, 2/25/99
'          Last modification:  3/6/99
'
'-----------------------------------------------------------
'-----------------------------------------------------------

TheProject = av.GetProject
TheView = av.GetActiveDoc
TheDisplay = TheView.GetDisplay
ShapeLineList = List.Make
for each name in TheView.GetThemes
   if (name.is(Ftheme)) then
      if (name.GetFtab.FindField("shape").GetType=#FIELD_SHAPELINE) then
         ShapeLineList.add(name)
      end
   else
      continue
   end
end

XsecThemeName = Msgbox.ChoiceAsString
(ShapeLineList,"Choose the cross-section theme.","Line Theme")
if (XsecThemeName = nil) then
   exit
end
XsecFtab = XsecThemeName.GetFtab

'--- IDENTIFY INPUT FIELDS
ShpField = XsecFtab.FindField("Shape")
LBankField = XsecFtab.FindField("LBankPct")
CenterField = XsecFtab.FindField("CenterPct")
RBankField = XsecFtab.FindField("RBankPct")
LeftZField = XsecFtab.FindField("LBankZ")
CenterZField = XsecFtab.FindField("CenterZ")
RightZField = XsecFtab.FindField("RBankZ")

'RETRIEVE THE POINT LOCATIONS
leftPts = List.Make
centerPts = List.Make
rightPts = List.Make
```

```
for each rec in XsecFtab
   leftPct = XsecFtab.ReturnValue(LBankField,rec)
   centerPct = XsecFtab.ReturnValue(CenterField,rec)
   rightPct = XsecFtab.ReturnValue(RBankField,rec)
   thePolyLine = XsecFtab.ReturnValue(ShpField,rec)
   leftptx = thePolyLine.Along(leftPct).GetX
   leftpty = thePolyLine.Along(leftPct).GetY
   leftptz = XsecFtab.ReturnValue(LeftZField,rec)
   leftpt = PointZ.Make(leftptx,leftpty,leftptz)
   leftPts.add(leftpt)

   centerptx = thePolyLine.Along(centerPct).GetX
   centerpty = thePolyLine.Along(centerPct).GetY
   centerptz = XsecFtab.ReturnValue(CenterZField,rec)
   centerpt = PointZ.Make(centerptx,centerpty,centerptz)
   centerPts.add(centerpt)

   rightptx = thePolyLine.Along(rightPct).GetX
   rightpty = thePolyLine.Along(rightPct).GetY
   rightptz = XsecFtab.ReturnValue(RightZField,rec)
   rightpt = PointZ.Make(rightptx,rightpty,rightptz)
   rightPts.add(rightpt)
end

'CREATE THE CENTER AND BANK LINES
lBankline = polylineZ.make({leftPts})
centerline = polylineZ.make({centerPts})
rBankline = polylineZ.make({rightPts})

'--- READ AND PROCESS DATA FOR CENTER AND BANK LINES
'StreamLinesFileName = FN.Merge(av.GetProject.GetWorkDir.asString,"Stream3D")
StreamLinesFileName = FileDialog.Put
("Stream3D".asFilename,"*.shp","Stream Centerline and Banklines Shapefile")
if(StreamLinesFileName = Nil)then
   exit
end
StreamLinesFileName.SetExtension("shp")
StreamLinesFtab = Ftab.MakeNew(StreamLinesFileName,polylineZ)
StreamLinesTheme = Ftheme.make(StreamLinesFtab)
theView.addTheme(StreamLinesTheme)
StreamLinesTheme.SetVisible(true)

'CREATE FIELDS FOR THE 3D LINE ATTRIBUTE TABLE
StreamLinesFields = List.Make
StreamLinesFields.Add(Field.Make("Line Type",#FIELD_VCHAR,16,0))
StreamLinesFieldsc = StreamLinesFields.DeepClone
StreamLinesFtab.addFields(StreamLinesFieldsc)
if (StreamLinesFtab.CanEdit) then
   StreamLinesFtab.SetEditable(true)
else
```

```
    msgbox.info("Can’t edit the output theme.","Error")
    exit
end

’IDENTIFY FIELDS FOR WRITING
shpField = StreamLinesFtab.FindField("Shape")
typeField = StreamLinesFtab.FindField("Line Type")
StreamLinesFtab.AddRecord
StreamLinesFtab.SetValue(shpField,0,lBankline)
StreamLinesFtab.SetValue(typeField,0,"Left Bank")
StreamLinesFtab.AddRecord
StreamLinesFtab.SetValue(shpField,1,centerline)
StreamLinesFtab.SetValue(typeField,1,"Centerline")
StreamLinesFtab.AddRecord
StreamLinesFtab.SetValue(shpField,2,rBankline)
StreamLinesFtab.SetValue(typeField,2,"Right Bank")
StreamLinesFtab.SetEditable(false)
```

```
'------------------------------------------------------------
'------------------------------------------------------------
' Name: Boundary.ave
' Headline:
' Self:
' Returns:  Polygon
' Description: Takes 3D stream cross-sections from an polylineZ shapefile
'            and creates a 2D shapefile representing the boundary of
'            the cross-sections.
' Topics:
' Search Keys:
' Requires:  3D analyst extension, Ftheme of channel cross-sections.
' History:  Created by Eric Tate, 10/2/98
'          Last modification:  3/6/99
'
'------------------------------------------------------------
'------------------------------------------------------------

theProject = av.GetProject
theView = av.GetActiveDoc

ShapeLineList = List.Make
for each name in TheView.GetThemes
   if (name.is(Ftheme)) then
      if (name.GetFtab.FindField("shape").GetType=#FIELD_SHAPELINE) then
         ShapeLineList.add(name)
      end
   else
      continue
   end
end

xsectThemeName = msgbox.choiceasString(ShapeLineList,
            "Select the line theme.","Line Theme")
if (xsectThemename = nil) then
   exit
end
xsecFtab = xsectThemeName.getFtab

'--- IDENTIFY INPUT FIELDS
shpField = xsecFtab.FindField("Shape")

'--- RETRIEVE THE POINT PARAMETERS
lboundpts = list.make
rboundpts = list.make
for each rec in xsecFtab
   theLine = xsecFtab.returnValue(shpField,rec)
   thePts = theLine.asList
   ptList = thePts.get(0)
   ptcount = ptList.count-1
```

```
   firstpt = ptList.get(0)
   firstptx = firstpt.GetX
   firstpty = firstpt.GetY
   newFirstpt = Point.Make(firstptx,firstpty)
   lboundpts.add(newFirstpt)

   lastpt = ptList.get(ptcount)
   lastptx = lastpt.GetX
   lastpty = lastpt.GetY
   newLastpt = Point.Make(lastptx,lastpty)
   rboundpts.insert(newLastpt)
end
for each item in rboundpts
   lboundpts.add(item)
end
terrainPgon = polygon.make({lboundpts})

'--- READ AND PROCESS DATA FOR FLOODPLAIN POLYGON SHAPEFILE
'TerrainFileName = FN.Merge(av.GetProject.GetWorkDir.asString,"boundary")
TerrainFileName = FileDialog.Put("Boundary".asfilename,"*.shp","Bounding Polygon Shape
File")
if(TerrainFileName = Nil)then
   exit
end
TerrainFileName.SetExtension("shp")
TerrainFTab = FTab.MakeNew(TerrainFileName,polygon)

f2 = Field.Make("ThemeName",#FIELD_VCHAR,15,0)
TerrainFtab.AddFields({f2})
if(TerrainFtab.CanEdit)then
   TerrainFtab.SetEditable(true)
else
   msgbox.info("Can't edit the output theme.","Error")
   exit
end

f1 = TerrainFtab.FindField("Shape")
f2 = TerrainFtab.FindField("ThemeName")
newrec = TerrainFtab.AddRecord
TerrainFtab.SetValue(f1,newrec,terrainPgon)
TerrainFtab.SetValue(f2,newrec,xsectThemeName.asString)

TerrainTheme = Ftheme.make(TerrainFTab)
theView.addTheme(TerrainTheme)
TerrainTheme.setVisible(true)
TerrainFTab.setEditable(false)
```

```
'------------------------------------------------------------
'------------------------------------------------------------
' Name: Compare.ave
' Headline:
' Self:
' Returns:  ASCII Text File
' Description: Takes channel coordinates from three different themes
'            and writes the elevations to a text file
' Topics:
' Search Keys:
' Requires:  Two polylineZ Fthemes of cross-section coordinates and
'          a Gtheme DEM.
' History:  Created by Eric Tate, 12/29/98
'         Last modification:  3/8/99
'
'------------------------------------------------------------
'------------------------------------------------------------

'IDENTIFY THE THEMES
TheView = av.GetActiveDoc
FThemeList = List.Make
GThemeList = List.Make
for each name in TheView.GetThemes
   if (name.is(Ftheme)) then
      if (name.GetFtab.FindField("shape").GetType=#FIELD_SHAPELINE) then
         FThemeList.add(name)
      end
   elseif (name.is(Gtheme)) then
      GThemeList.Add(name)
   else
      continue
   end
end

NewTerrainTheme = Msgbox.ChoiceAsString(FThemeList,
"Choose the line theme of resampled 3D channel cross-sections.",
"Resampled Cross-Section Theme")
if (NewTerrainTheme = nil) then
   exit
end

OldTerrainTheme = Msgbox.ChoiceAsString(FThemeList,
"Choose the line theme of original 3D channel cross-sections.",
"Original Cross-Section Theme")
if (OldTerrainTheme = nil) then
   exit
end

DEMTheme = msgbox.ChoiceAsString(GThemeList,
"Choose the digital elevation model.","DEM Theme")
if (DEMTheme = nil) then
```

```
   exit
end

'DEMTheme2 = msgbox.ChoiceAsString(GThemeList,
'"Choose the digital elevation model.","DEM Theme")
'if (DEMTheme2 = nil) then
'    exit
'end

NewTerrainFtab = NewTerrainTheme.GetFtab
OldTerrainFtab = OldTerrainTheme.GetFtab
DEMGrid = DEMTheme.GetGrid
'DEMGrid2 = DEMTheme2.GetGrid


'IDENTIFY THE FIELDS
StationField = NewTerrainFtab.FindField("Station")
NewShpField = NewTerrainFtab.FindField("Shape")
OldShpField = OldTerrainFtab.FindField("Shape")
LengthField = OldTerrainFtab.FindField("Length")


'CHOOSE THE CROSS-SECTION (STATION) NUMBER
StationList = List.Make
for each rec in NewTerrainFtab
   station = NewTerrainFtab.ReturnValue(StationField,rec)
   StationList.add(station)
end
StationNo = Msgbox.ChoiceAsString(StationList,"Choose the river station number.","River
Station")
if (StationNo = nil) then
   exit
end

ListIndex = StationList.FindByValue(StationNo)
NewPolyline = NewTerrainFtab.ReturnValue(NewShpField,ListIndex)
OldPolyline = OldTerrainFtab.ReturnValue(OldShpField,ListIndex)
Length = OldTerrainFtab.ReturnValue(LengthField,ListIndex)
NewPolylineList = NewPolyline.AsList.Get(0)
OldPolylineList = OldPolyline.AsList.Get(0)
PolylineCount = NewPolylineList.Count - 1

AlongString = "X Distance"
NewString = NewTerrainTheme.AsString
OldString = OldTerrainTheme.AsString
DEMString = DEMTheme.AsString
'DEMString2 = DEMTheme2.AsString

for each pnt in 0..PolylineCount
   NewPointZ = NewPolylineList.Get(pnt)
   AlongPct = NewPolyline.PointPosition(NewPointZ)
```

```
   xLocation = ((AlongPct/100)*Length).AsString
   AlongString = AlongString+","+xLocation
   NewZ = NewPointZ.GetZ.AsString
   NewString = NewString+","+NewZ

   OldPointZ = OldPolylineList.Get(pnt)
   OldZ = OldPointZ.GetZ.AsString
   OldString = OldString+","+OldZ

   XCoord = NewPointZ.GetX
   YCoord = NewPointZ.GetY
   Point2D = Point.Make(XCoord,YCoord)
   DEMZ = DEMGrid.PointValue(Point2D,Prj.MakeNull).AsString
   DEMString = DEMString+","+DEMZ
'   DEMZ2 = DEMGrid2.PointValue(Point2D,Prj.MakeNull).AsString
'   DEMString2 = DEMString2+","+DEMZ2
end

'TheFileName = FN.Merge(av.GetProject.GetWorkDir.asString,StationNo.asString)
TheFileName = FileDialog.Put(StationNo.asString.asFilename,"*.txt","ASCII Text File")
if(TheFileName = Nil)then
   exit
end
TheFileName.SetExtension("txt")
TheLineFile = LineFile.Make(TheFileName, #FILE_PERM_WRITE)
TheLineFile.WriteElt(AlongString)
TheLineFile.WriteElt(NewString)
TheLineFile.WriteElt(OldString)
TheLineFile.WriteElt(DEMString)
'TheLineFile.WriteElt(DEMString2)
TheLineFile.Close
```

```
'------------------------------------------------------------
'------------------------------------------------------------
'Name: FlipLine.ave
'Headline:
'Self:
'Returns:  Shape file
'Description: Takes a polyline shapefile, and flips it to change
'            its direction.
'Topics:
'Search Keys:
'Requires:  Ftheme of digitized theme.
'History:  Created by Eric Tate, 3/20/99
'          Last modification:  3/21/99
'
'------------------------------------------------------------
'------------------------------------------------------------

theView = av.GetActiveDoc
FThemeList = List.Make
for each name in TheView.GetThemes
   if (name.is(Ftheme)) then
      if (name.GetFtab.FindField("shape").GetType=#FIELD_SHAPELINE) then
         FThemeList.add(name)
      end
   else
      continue
   end
end

RivThemeName = Msgbox.ChoiceAsString(FThemeList,
"Choose the stream centerline theme.",
"Stream Centerline Theme")
if (RivThemeName = nil) then
   exit
end

RivFtab = RivThemeName.GetFtab
RivFtab.SetEditable(true)

'ACCESS THE SELECTED RECORDS OF THE STREAM SHAPEFILE
shpField = RivFtab.FindField("Shape")
for each rec in RivFtab
   TheLine = RivFtab.ReturnValue(shpField,rec)
   FlipLine = TheLine.Flip
   RivFtab.SetValue(shpField,rec,FlipLine)
end
RivFtab.setEditable(false)
```

```
'------------------------------------------------------------
'------------------------------------------------------------
' Name: Floodplain.ave
' Headline:
' Self:
' Returns:  Grid
' Description: Takes grids representing the land surface and water
'             surface profiles and returns a new grid of flood
'             depths.
' Topics:
' Search Keys:
' Requires:  Gthemes of land surface and water surface profiles.
' History:  Created by Eric Tate, 3/5/99
'          Last modification:  3/5/99
'
'------------------------------------------------------------
'------------------------------------------------------------

'GET THE GRIDS
TheView = av.GetActiveDoc
GThemeList = List.Make
for each name in TheView.GetThemes
   if (name.is(Gtheme)) then
      GThemeList.Add(name)
   end
end

if (GThemeList.count = 0) then
   msgbox.error("No grid themes found", "Grid Themes")
   exit
end

LandGridName = Msgbox.ChoiceAsString(GThemeList,
"Choose the grid of land surface elevation.",
"Land Surface Grid")
if (LandGridName = nil) then
   exit
end
WaterGridName = msgbox.ChoiceAsString(GThemeList,
"Choose the grid of water surface profiles.",
"Water Surface Profile Grid")
if (WaterGridName = nil) then
   exit
end

LandGrid = LandGridName.GetGrid
WaterGrid = WaterGridName.GetGrid
DifferenceGrid = WaterGrid - LandGrid
QueryGrid = (DifferenceGrid > 0.asGrid).con(1.asGrid,0.asGrid)
FloodGrid = DifferenceGrid/QueryGrid
```

```
FloodGridFileName = FileDialog.Put("floodgrid".asFilename,"*.*","Floodplain Grid")
if(FloodGridFileName = Nil)then
   exit
end

'FloodGridFileName = FN.Merge(av.GetProject.GetWorkDir.asString,"floodgrid")
FloodGrid.SaveDataSet(FloodGridFileName)
FloodGTheme = Gtheme.Make(FloodGrid)
TheView.AddTheme(FloodGTheme)
FloodGTheme.SetVisible(false)
```

```
'-----------------------------------------------------------
'-----------------------------------------------------------
' Name: GetLength.ave
' Headline:
' Self:
' Returns:  Shape file
' Description: Takes a polyline coverage, determines the length
'            of each segment and edits the attribute table to include these lengths.
' Topics:
' Search Keys:
' Requires:  Ftheme of digitized theme.
' History:  Created by Eric Tate, 7/23/98
'         Last modification:  3/6/99
'-----------------------------------------------------------
'-----------------------------------------------------------

theView = av.GetActiveDoc
themesList = theView.GetActivethemes
themesCount = themesList.count
if (themescount <> 1) then
   msgbox.info("The stream shapefile theme should be active.","Error")
   exit
end
themeName = themesList.get(0)

DigiFtab = themeName.getFtab
DigiFtab.setEditable(true)

'ACCESS THE SELECTED RECORDS OF THE STREAM SHAPEFILE
   rivFields = List.Make
   rivFields.Add(Field.Make("Seg Length",#Field_Decimal,12,1))
'   rivFields.Add(Field.Make("Length",#Field_Decimal,12,1))
   rivFieldsC = rivFields.DeepClone
   DigiFTab.addFields(rivFieldsC)
   if(DigiFTab.CanEdit) then
      DigiFTab.SetEditable(True)
   else
      msgbox.info("Can't edit the output theme.","Error")
      exit
   end

shpField = DigiFtab.findfield("Shape")
lengthField = DigiFtab.findField("Seg Length")
totLengthField = DigiFtab.findField("Length")
totalLength = 0
allpoints = list.make
for each rec in DigiFtab
   streamSeg = DigiFtab.returnvalue(shpField,rec)
   digipts = streamSeg.asMultipoint.asList
   ptscount = digipts.count-1
   seglength = 0
```

```
    for each pt in 0..ptscount
        if (pt=ptscount) then
            break
        else
            allpoints.add(digipts.get(pt))
            nextpt = pt+1
            x1 = digipts.get(pt).getx
            y1 = digipts.get(pt).gety
            x2 = digipts.get(nextpt).getx
            y2 = digipts.get(nextpt).gety
            thislength = (((x1-x2)^2)+((y1-y2)^2))^0.5
            seglength = seglength + thislength
        end
    end
    digiFtab.setValue(lengthField,rec,seglength)
    totalLength = totalLength + seglength
end

rivpolyline = polyline.make({allpoints})
streamname = MsgBox.Input( "Enter your stream name", "Stream Name", "" )

'OutFileName = FN.Merge(av.GetProject.GetWorkDir.asString,streamname)
OutFileName = FileDialog.Put(streamname.asfilename,"*.shp","Output Shape File")
if(OutFileName = Nil)then
    exit
end
OutFileName.SetExtension("shp")
OutFtab = Ftab.MakeNew(OutFileName,polyline)


'CREATE FIELDS FOR THE NEW LINE ATTRIBUTE TABLE
outFields = List.Make
outFields.Add(Field.Make("Name",#FIELD_VCHAR,40,0))
outFields.Add(Field.Make("Length",#FIELD_DECIMAL,10,1))
outFieldsc = outFields.DeepClone
outFtab.addFields(outFieldsc)
if (outFtab.CanEdit) then
    outFtab.SetEditable(true)
else
    msgbox.info("Can't edit the output theme.","Error")
    exit
end


'IDENTIFY FIELDS FOR WRITING
newshpField = outFtab.FindField("Shape")
nameField = outFtab.FindField("Name")
newlengthField = outFtab.Findfield("Length")
newrec = outFtab.AddRecord
outFtab.setValue(newshpField,newrec,rivpolyline)
outFtab.setValue(nameField,newrec,streamname)
```

```
outFtab.setValue(newLengthField,newrec,totallength)

DigiFtab.setEditable(false)
outFtab.setEditable(false)

newRivTheme = Ftheme.make(outFtab)
theView.addTheme(newRivTheme)
newRivTheme.setVisible(true)
```

```
'***************************************
''
' Name: Gridclip.ave
' Headline:
' Self:
' Returns:
' Description: This program clip a grid Theme
' based on selected polygons in a polygon coverage.
'
' Topics:
' Search Keys:
' Requires:
' History: Modified from hydro.ExtByPly
'***************************************

theView = av.getactivedoc
theThemes = theView.getthemes
if (nil = theThemes) then exit end
if (theThemes.count < 2) then
  msgbox.error("At least 2 themes must be in the View","Error")
  exit
end
polythemes=list.make
gridthemes=list.make
for each t in thethemes
   if (t.getclass.getclassname = "GTheme") then
      gridthemes.add(t)
   elseif (t.getftab.findfield("shape").gettype=#field_shapepoly) then
      polythemes.add(t)
   end
end
GridTheme=Msgbox.choiceasstring(gridthemes,"Choose Grid to be clipped?","Clip Grid")
if (gridTheme = NIL) then exit end
clipTheme=Msgbox.Choiceasstring(polythemes,"Which polygon theme is the clipping
theme?","Clip Grid")
if (clipTheme = NIL) then exit end
'--- build a polygon which is the union of all selected polygons
'--- if no polygons are selected, then select all polygons in the
'--- clipTheme
clipftab=cliptheme.getftab
shapefield=clipftab.findfield("Shape")
clipselection = clipftab.getselection

if (clipselection.count=0) then
  clipftab.getselection.setall
  clipftab.updateselection
end

rec1=clipftab.getselection.getnextset(-1)
totalshape=clipftab.returnvalue(shapefield,rec1)
for each rec in clipftab.getselection
```

```
  totalshape=totalshape.returnunion(clipftab.returnvalue(shapefield, rec))
end

theclipshape = totalshape
sourceGrid=GridTheme.GetGrid
'--set the extent before extracting
ae = theView.GetExtension(AnalysisEnvironment)
ae.SetExtent(#ANALYSISENV_VALUE,theclipshape.ReturnExtent)
ae.Activate
CutGrid=SourceGrid.ExtractByPolygon(theclipshape,prj.makeNull,false)
cutFN=av.GetProject.GetWorkDir.MakeTmp(GridTheme.GetName,"")
CutGrid.ReName(cutFN)
outGTheme=GTheme.Make(CutGrid)
outGTheme.SetName(cutFN.getBaseName)
theView.AddTheme(outGTheme)
outGTheme.SetVisible(true)
```

```
'-------------------------------------------------------------
'-------------------------------------------------------------
' Name: NewXSects.ave
' Headline:
' Self:
' Returns:  Shape file
' Description: Takes a 3D shapefile of channel cross-sections, and
'             assigns new floodway elevations based on weighted
'             averaging with a raster digital elevation model.
' Topics:
' Search Keys:
' Requires:  Polyline Z Ftheme of channel cross-sections and Gtheme
'           of elevations.
' History:  Created by Eric Tate, 12/28/98
'          Last modification:  3/6/99
'
'-------------------------------------------------------------
'-------------------------------------------------------------

TheView = av.GetActiveDoc
FThemeList = List.Make
GThemeList = List.Make
for each name in TheView.GetThemes
   if (name.is(Ftheme)) then
      if (name.GetFtab.FindField("shape").GetType=#FIELD_SHAPELINE) then
         FThemeList.add(name)
      end
   elseif (name.is(Gtheme)) then
      GThemeList.Add(name)
   else
      continue
   end
end

OldXSecThemeName = Msgbox.ChoiceAsString(FThemeList,
"Choose the line theme of 3D channel cross-sections.",
"3D Cross-Section Theme")
if (OldXSecThemeName = nil) then
   exit
end
DEMThemeName = msgbox.ChoiceAsString(GThemeList,
"Choose the digital elevation model.","DEM Theme")
if (DEMThemeName = nil) then
   exit
end
OldFtab = OldXSecThemeName.GetFtab
NumRecords = OldFtab.GetNumRecords
DEMGrid = DEMThemeName.GetGrid


'--- READ AND PROCESS DATA FOR CENTER AND BANK LINES
```

```
'NewXSecsFileName = FN.Merge(av.GetProject.GetWorkDir.asString,"3DXSects")
NewXSecsFileName = FileDialog.Put("3DXSects".asFilename,"*.shp",
"New 3D Cross-Sections Shapefile")
if(NewXSecsFileName = Nil)then
  exit
end
NewXSecsFileName.SetExtension("shp")
NewFtab = Ftab.MakeNew(NewXSecsFileName,PolylineZ)
NewTheme = Ftheme.Make(NewFtab)
TheView.AddTheme(NewTheme)
NewTheme.SetVisible(true)

'CREATE FIELDS FOR THE 3D CROSS-SECTIONS ATTRIBUTE TABLE
NewFields = List.Make
NewFields.Add(Field.Make("Station",#FIELD_DECIMAL,12,0))
NewFields.Add(Field.Make("Length",#FIELD_DECIMAL,12,2))
NewFields.Add(Field.Make("LBankPct",#FIELD_DECIMAL,15,2))
NewFields.Add(Field.Make("CenterPct",#FIELD_DECIMAL,15,2))
NewFields.Add(Field.Make("RBankPct",#FIELD_DECIMAL,15,2))
NewFields.Add(Field.Make("LBankZ",#FIELD_DECIMAL,12,2))
NewFields.Add(Field.Make("CenterZ",#FIELD_DECIMAL,12,2))
NewFields.Add(Field.Make("RBankZ",#FIELD_DECIMAL,12,2))
NewFieldsc = NewFields.DeepClone
NewFtab.addFields(NewFieldsc)
if (NewFtab.CanEdit) then
  NewFtab.SetEditable(true)
else
  msgbox.info("Can't edit the output theme.","Error")
  exit
end

'IDENTIFY INPUT FIELDS
OldShpField = OldFtab.FindField("Shape")
OldStationField = OldFtab.FindField("Station")
LengthField = OldFtab.FindField("Length")
LeftPctField = OldFtab.FindField("LBankPct")
CenterPctField = OldFtab.FindField("CenterPct")
RightPctField = OldFtab.FindField("RBankPct")
LeftZField = OldFtab.FindField("LBankZ")
CenterZField = OldFtab.FindField("CenterZ")
RightZField = OldFtab.FindField("RBankZ")

'IDENTIFY FIELDS FOR WRITING
NewShpField = NewFtab.FindField("Shape")
NewStationField = NewFtab.FindField("Station")
NewLengthField = NewFtab.FindField("Length")
NewLeftPctField = NewFtab.FindField("LBankPct")
NewCenterPctField = NewFtab.FindField("CenterPct")
NewRightPctField = NewFtab.FindField("RBankPct")
NewLeftZField = NewFtab.FindField("LBankZ")
NewCenterZField = NewFtab.FindField("CenterZ")
```

```
NewRightZField = NewFtab.FindField("RBankZ")


'SHOW THE STATUS BAR
av.ShowMsg ("Recalculating cross-section elevations...")
cancelled = false
statusIndex = 0
count = 0
av.SetStatus(statusIndex)

'ACCESS THE CROSS-SECTION POINTS
for each rec in OldFtab
   OldPolylineZ = OldFtab.ReturnValue(OldShpField,rec)
   OldPointsList = OldPolylineZ.AsList
   OldPoints = OldPointsList.Get(0)
   OldPointsCount = OldPoints.Count - 1
   Length = OldFtab.ReturnValue(LengthField,rec)
   LeftPct = OldFtab.ReturnValue(LeftPctField,rec)
   LeftFloodwayLength = (LeftPct*Length)/100
   CenterPct = OldFtab.ReturnValue(CenterPctField,rec)
   RightPct = OldFtab.ReturnValue(RightPctField,rec)
   RightFloodwayLength = ((100-RightPct)*Length)/100
   CenterPoint = OldPolylineZ.Along(CenterPct)
   NewPoints = List.Make

   LeftFloodwayPts = List.Make
   LeftStartPoint = OldPolylineZ.Along(LeftPct)
   LeftFloodwayPts.Add(LeftStartPoint)
   LeftEndPoint = OldPolylineZ.Along(0)
   LeftFloodwayPts.Add(LeftEndPoint)
   LeftFloodwayLine = Polyline.Make({LeftFloodwayPts})

   RightFloodwayPts = List.Make
   RightStartPoint = OldPolylineZ.Along(RightPct)
   RightFloodwayPts.Add(RightStartPoint)
   RightEndPoint = OldPolylineZ.Along(100)
   RightFloodwayPts.Add(RightEndPoint)
   RightFloodwayLine = Polyline.Make({RightFloodwayPts})

   statusIndex = (count/NumRecords)*100
   cancelled = false
   av.ShowStopButton
   continued = av.SetStatus (statusIndex)
   if (not continued) then
      cancelled = true
      msgbox.info("Process Interrupted.","Stop")
      exit
   end

   for each pnt in 0..OldPointsCount
      OldPoint = OldPoints.Get(pnt)
```

```
      AlongPct = OldPolyLineZ.PointPosition(OldPoint)
      XCoord = OldPoint.GetX
      YCoord = OldPoint.GetY
      OldZCoord = OldPoint.GetZ
      DEMPoint = Point.Make(XCoord,YCoord)
      DEMZCoord = DEMGrid.PointValue(DEMPoint,Prj.MakeNull)

      if (AlongPct <= LeftPct) then
        FloodwayPct = (LeftFloodwayLine.PointPosition(DEMPoint))/100
        NewZCoord = (OldZCoord*(1-FloodwayPct)) + (DEMZCoord*FloodwayPct)
        NewPoint = PointZ.Make(XCoord,YCoord,NewZCoord)
        NewPoints.Add(NewPoint)
      elseif ((AlongPct > LeftPct) and (AlongPct < RightPct)) then
        NewPoints.Add(OldPoint)
      else
        FloodwayPct = (RightFloodwayLine.PointPosition(DEMPoint))/100
        NewZCoord = (OldZCoord*(1-FloodwayPct)) + (DEMZCoord*FloodwayPct)
        NewPoint = PointZ.Make(XCoord,YCoord,NewZCoord)
        NewPoints.Add(NewPoint)
      end
    end

    LeftBankZ = OldFtab.ReturnValue(LeftZField,rec)
    CenterZ = OldFtab.ReturnValue(CenterZField,rec)
    RightBankZ = OldFtab.ReturnValue(RightZField,rec)
    NewPolylineZ = PolylineZ.Make({NewPoints})
    count = count + 1

'WRITE THE CROSS-SECTION TO THE FTAB
    Station = OldFtab.ReturnValue(OldStationField,rec)
    newrec = NewFtab.AddRecord
    NewFtab.SetValue(NewShpField,newrec,NewPolylineZ)
    NewFtab.SetValue(NewStationField,newrec,Station)
    NewFtab.SetValue(NewLengthField,newrec,Length)
    NewFtab.SetValue(NewLeftPctField,newrec,NewPolylineZ.PointPosition(LeftStartPoint))
    NewFtab.SetValue(NewCenterPctField,newrec,NewPolylineZ.PointPosition(CenterPoint))
    NewFtab.SetValue(NewRightPctField,newrec,NewPolylineZ.PointPosition(RightStartPoint))
    NewFtab.SetValue(NewLeftZField,newrec,LeftBankZ)
    NewFtab.SetValue(NewCenterZField,newrec,CenterZ)
    NewFtab.SetValue(NewRightZField,newrec,RightBankZ)
end

NewFtab.SetEditable(false)
av.ClearStatus
av.ClearMsg
```

```
'Name: Projector!
'
'Headline: Allows user to project themes from one projection
'to another, using any projections ArcView supports.  User must
'know the input and output projection and units.
'Self:
'Returns:
'Description: Projects active fthemes in current view to
'new shape files in any projection that ArcView supports,
'projecting into units of feet, meters or decimal degrees.
'The user must know the units (and projection) of the input
'theme(s), and will be asked for the output units and projection.
'
'If neither input or output projection is geographic,
'then the user must have 2x the amount of space of the original
'shape file, as this script writes a temporary shapefile in geographic
'coordinates, projects that to the output projection and
'deletes the temporary files (much faster than stepping through records).
'
'Attach this as the click script to a control in a View GUI.
'Topics: Conversion, Themes
'Search Keys: Project, projections, theme, units, convert
'Requires:  Fthemes in a view with the map units set, and a
'knowledge of the projection of the themes.
'============================================================

'check for shift key, if shift key is down pop up instructions
'explain script, pops up a message with
'info about how to use

if (System.IsShiftKeyDown) then
  message =
    "To use the projection tool you must know the map units and projection"++
    "of the datasets being projected. You will be prompted for the output units and
projection."+nl+nl+
    "Brief Instructions:"+nl+nl+
    "1. Add some themes to the view."+nl+nl+
    "2. Set the map units appropriately in the View Properties window."+nl+nl+
    "3. Make the theme(s) you wish to project active."+nl+nl+
    "4. Press this button. You will be prompted for certain information"++
      "which may include the input projection, the output units and the"++
      "output projection."+nl+nl+
    "5. You will be asked if you want to recalculate area, perimeter and length fields."+nl+nl+
    "WARNING - If the field to be recalculated is not large enough to hold the new "+
    "(calculated) number, the value put in that field will be incorrect."+nl+nl+

    "6. You will be asked if you want to add the projected theme(s) to a view."+nl+nl+
    "7. You will be asked for output shapefile names for each theme to be projected."+nl

  msgbox.report(message,"Projector! Instructions")
  return nil
```

```
end

'get the active document, which should be a view
theView=av.GetActiveDoc

if (theView.Is(View).Not) then
  MsgBox.Error("This script must be run from a view. Exiting","Error")
  return nil
end

'Get the current source units and make sure they are not unknown
sourceunits = av.getactivedoc.getdisplay.getunits

if (sourceunits = #UNITS_LINEAR_UNKNOWN) then
  MsgBox.Error("View units must be set before projecting. Stopping.",
              "Error!")
  return nil
end

'build a list of fthemes from the active theme list, we will project these
thms=List.Make
for each t in theView.GetActiveThemes
  if (t.Is(Ftheme)) then
    thms.Add(t)
  end
end

'if no themes active let the user know and quit
if (thms.Count = 0) then
  System.Beep
  MsgBox.Error("Please make at least one feature theme active!","Error")
  return nil
end

'Get the projection of the view
myprj=av.getactivedoc.getprojection

'Try to figure out if the source data is geographic
' If our Prj.AsString <> "" then we have a projection and we are
' geographic, or if our sourceunits are set to decimal degrees without
' a projection we are too

inputgeographic =((myprj.AsString <> "") or
    ((sourceunits = #UNITS_LINEAR_DEGREES) and (myprj.AsString = "")))

'If not, then let user pick input and output
if (inputgeographic.Not) then

  Msgbox.Info("Please select the input projection"++
              "in the next dialog box","Projector!")
```

```
  'Pop up dialog box, to get input projection
  'and check for cancel button (nil)

  inputPrj = ProjectionDialog.Show(theView,sourceunits)
  if (inputPrj.IsNull) then
    return nil
  end

  'Pop up dialog box, to get output projection
  'and check for cancel button (nil)

else  'your input projection is geographic
  sourceunits = #UNITS_LINEAR_DEGREES
end

unitslist1 = {"meters","feet","decimal degrees","miles","kilometers","yards",
          "centimeters","inches","millimeters"}

unitslist2 = {#UNITS_LINEAR_METERS, #UNITS_LINEAR_FEET,
           #UNITS_LINEAR_DEGREES, #UNITS_LINEAR_MILES,
           #UNITS_LINEAR_KILOMETERS, #UNITS_LINEAR_YARDS,
           #UNITS_LINEAR_CENTIMETERS, #UNITS_LINEAR_INCHES,
           #UNITS_LINEAR_MILLIMETERS}

outputunits = MsgBox.ChoiceAsString(unitslist1,
               "Please pick output units","Projector!")

if (outputunits = nil) then
  return nil
else
  outunits = unitslist2.Get(unitslist1.FindByValue(outputunits))
end

'get the output projection, using the dialog box and check for cancel (nil)
if (outputunits = "decimal degrees") then
  outputPrj = prj.MakeNull
  outputPrj.SetDescription("Geographic")
  outputgeographic = true
else
  outputPrj = ProjectionDialog.Show(theView,outunits)
end

if (outputPrj = nil) then
  return nil
end

'check to see if they are the same. If so, exit

if ((inputgeographic.Not) and ((outputPrj = inputPrj) and (sourceunits = outunits))) then
  MsgBox.Error("Input projection same as output projection.","Error")
  return nil
```

```
end

'check to see if output projection is geographic.  If it is, we won't need
'to do as much work below
outputgeographic = outputPrj.ReturnDescription.Contains("Geographic")

'if both input and output are geographic, quit here
if (inputgeographic and outputgeographic) then
  MsgBox.Error("Both input and output are geographic. Stopping.","Error")
  return nil
end

'check to see if we want to recalculate area, perimeter, length fields
recalc = Msgbox.YesNo("Recalculate area, perimeter and length fields"++
         "(if present) using"++outputunits+"?","Projector!",true)

'check to see if we want to put results into a view
    if (MsgBox.YesNo("Add projected shapefile(s) as theme(s) to a view?",
               "Projector!",true)) then
  'make a list of views
  viewlist = List.Make
  for each d in av.GetProject.GetDocs
    if (d.Is(View)) then
      viewlist.Add(d)
    end
  end  'for each d

  'provide a choice for a new view
  viewlist.Add("<New View>")
      AddToView = MsgBox.ListAsString(viewlist,"Add Theme to:",
                  "Projector!")

  if (AddToView <> nil) then
    if (AddToView = "<New View>") then
      AddToView = View.Make
      AddToView.GetWin.Open
    end
  end
else   'don't add to view
  AddtoView = nil
end



'step through each active theme
For each thm in thms

  'get a filename for the new shapefile

  defname = Filename.GetCWD.MakeTmp("theme","shp")
  outputfile = FileDialog.Put(defname,"*.shp","Project "+thm.GetName)
```

```
'if Canceled, then skip this theme
 if (outputfile = nil) then
  continue
end

'now export the ftab (selected records only),
   thmftab = thm.GetFtab
shapetype = thmftab.FindField("Shape").GetType

if (outputgeographic) then
  newFtab = thmFtab.ExportUnprojected(outputfile,inputPrj,
                  thmFtab.GetSelection.Count >0)
elseif (inputgeographic) then
  newFtab = thmFtab.ExportProjected(outputfile,outputPrj,
                  thmFtab.GetSelection.Count > 0)
else 'need to go to geographic, then to something else
  'make a temporary shape file
  tempshape = Filename.GetCWD.MakeTmp("xxprj","shp")
  tempftab = thmFtab.ExportUnprojected(tempshape,
                    inputPrj,thmFtab.GetSelection.Count >0)
  newFtab = tempFtab.ExportProjected(outputfile,outputPrj,false)

  'now clean up
  tempftab.DeActivate
  tempftab = nil
  av.PurgeObjects
  tempshpname = tempshape.GetBaseName.AsTokens(".").Get(0)
  tempshpdir = tempshape.GetFullName.Clone.AsFilename
  tempshpdir.Stripfile
  filesToDelete = tempshpdir.Readfiles(tempshpname+".*")

  for each f in filesToDelete
    File.Delete(f)
  end
end

'recalculate area, perim, length fields if asked to
 if (recalc) then

  'find the fields we need to recalculate
  newareafield = newftab.Findfield("Area")
  newperimfield = newftab.Findfield("Perimeter")
  newlengthfield = newftab.Findfield("Length")

  newftab.SetEditable(True)
  if (shapetype = #FIELD_SHAPEPOLY) then
    if (newareafield <> nil) then
      newftab.Calculate("[Shape].ReturnArea",newareafield)
    end
    if (newperimfield <> nil) then
```

```
      newftab.Calculate("[Shape].ReturnLength",newperimfield)
    end
  elseif (shapetype = #FIELD_SHAPELINE) then
    if (newlengthfield <> nil) then
      newftab.Calculate("[Shape].ReturnLength",newlengthfield)
    end
  end
  newftab.SetEditable(false)
 end 'if recalc

 if (addtoView <> nil) then
   'create a theme and add it to the specifiedView
   fthm = FTheme.Make(newFTab)
   AddToView.AddTheme(fthm)
   'put this theme in the same order as it was in the original view
   newplace = AddtoView.GetThemes.Count
   AddToView.GetThemes.Shuffle(fthm,newplace)
 end

end 'for each thm

if (addtoView <> nil) then
 'if view units aren't set, then set them
 if (AddtoView.GetDisplay.GetUnits = #UNITS_LINEAR_UNKNOWN) then
   AddtoView.GetDisplay.SetUnits(outunits)
 end

 'bring the View to the front
 AddToView.InvalidateTOC(nil)
 AddToView.GetWin.Activate
end
```

```
'-------------------------------------------------------------
'-------------------------------------------------------------
' Name: R2Vpoint.ave
' Headline:
' Self:
' Returns:  Point Ftheme
' Description:  Takes an Grid theme and converts it to a point shapefile.
'           It makes use of the asPointFtab function request
' Topics:  Raster to Vector Conversion
' Search Keys:
' Requires:  A view with a grid theme.
' History:  Created: ESRI user script
'        Last modified by Eric Tate:  4/18/99
'-------------------------------------------------------------
'-------------------------------------------------------------

theView = av.GetActiveDoc
theproj = theView.getprojection
GThemeList = List.Make
for each name in TheView.GetThemes
   if (name.is(Gtheme)) then
      GThemeList.Add(name)
   end
end

DEMThemeName = msgbox.ChoiceAsString(GThemeList,
"Choose the digital elevation model.","DEM Theme")
if (DEMThemeName = nil) then
   exit
end

thefilename = FileDialog.Put("Gridpts.shp".asfilename,"*.shp","Enter the new output filename.")
If (thefilename = nil) then
   exit
end

anFN = thefilename.asstring
position = anFN.IndexOf(".")
if (position = -1) then
   thefilename = (anFN + ".shp").asfilename
else
   anFN = anFN.Left(position -1)
   thefilename = (anFN + ".shp").asfilename
end

if (theproj.isnull.not) then
   keepprj = msgbox.YesNo(
   "Do you want the new shape file to remain in the current map projection? If you choose yes
then it will be saved in the current projection. If you choose No then it will be converted to
decimal degrees",
   "Projection",true)
```

```
else
   keepprj = true
end


theGrid = DEMThemeName.GetGrid
   if (keepprj = true) then
      lineResult = theGrid.AsPointFTab(thefilename,prj.makenull)
     else
      lineResult = theGrid.AsPointFtab(thefilename,theproj)
   end

theFTheme = FTheme.Make(lineResult)
theView.AddTheme(theFtheme)
```

```
'-------------------------------------------------------------
'-------------------------------------------------------------
' Name: RAS-Read.ave
' Headline:
' Self:
' Returns:  Table of RAS Cross-Section Descriptions
' Description: Reads a selected HEC-RAS report file and creates an
'            virtual table (Vtab) to contain selected channel and
'            flooplain parameters (in units of feet or meters).
' Topics:
' Search Keys:
' Requires:  HEC-RAS report (text) file
' History:  Created by Eric Tate, 1/12/98
' Last modified:  3/6/99
'-------------------------------------------------------------
'-------------------------------------------------------------

'SELECT THE HEC-RAS REPORT FILE
theFile = FileDialog.Show("*.rep*","HEC-RAS Report (.rep)",
        "Choose the HEC-RAS Report File")
if (nil = theFile) then
   exit
end

'CREATE THE COORDINATE LISTS
StationList = list.make
_xStations = list.make
_zStations = list.make
zmin = list.make
LFxvalues = list.make
newLFxvalues = list.make
LBxvalues = list.make
MCxvalues = list.make
RBxvalues = list.make
RFxvalues = list.make
newRFxvalues = list.make
LBzvalues = list.make
MCzvalues = list.make
RBzvalues = list.make
wselev = list.make
newwselev = list.make
mreach = list.make
describe = list.make
types = list.make

choiceList = list.make
choiceList.add("feet")
choiceList.add("meters")
Unit = MsgBox.ChoiceAsString(choiceList,"What units do you wish to work with in ArcView?",
"ArcView Units")
if (unit = nil) then
```

```
    exit
end

'SHOW THE STATUS BAR
av.ShowMsg ("Retrieving the HEC-RAS channel geometry info...")
canceled = false
statusIndex = 0
statusNum = 0
count = 0
av.SetStatus (statusIndex)

aFileName = Linefile.Make(theFile,#FILE_PERM_READ)
while (true)
    readline = aFileName.ReadElt
    if (readline = nil) then
        break
    end
    left6 = readline.left(6)
    left12 = readline.left(12)
    left16 = readline.left(16)

'GET THE NUMBER OF STATIONS
    if (left12 = "Number of:  ") then
        line1 = readline.astokens
                ("Number of: Cross Sections=Mulitple Openings")
        statusNum = line1.get(0).asNumber + line1.get(1).asNumber
        nextline = aFileName.ReadElt
        line1 = nextline.astokens("Culverts =Inline Weirs")
        statusNum = line1.get(0).asNumber + line1.get(1).asNumber + statusNum
        nextline = aFileName.ReadElt
        line1 = nextline.astokens("Bridges =")
        statusNum = line1.get(0).asNumber + statusNum
    end

'RETRIEVE THE RIVER STATION NUMBERS
    if (left6 = "REACH:") then
        count = count+1
        thelist = readline.astokens("REACHRS: ")
        StationList.add(thelist.get(1))
        statusIndex = (count/statusNum)*100
        cancelled = false
        av.ShowStopButton
        continued = av.SetStatus (statusIndex)
        if (not continued) then
            cancelled = true
            msgbox.info("Process Interrupted.","Stop")
            exit
        end
    end
```

```
'GET THE STATION DESCRIPTION
  if (left12 = "Description:") then
    loccount = readline.count
    newloccount = loccount-1
      if (loccount < 20) then
        mystring = "  "
      else
        mystring = readline.middle(13,newloccount)
      end
    describe.add(mystring)
  end

  if (left16 = "Downstream Bridg") then
    for each r in 1..20
      nextline1 = aFileName.ReadElt
    end
    continue
  end

'RETRIEVE THE X (LATERAL) AND Z (ELEVATION) COORDINATES
  if (left16 = "     Sta    Elev") then
    xcoordinates = list.make
    zcoordinates = list.make
    stations = list.make
    MCdepth = 10000
    while (true)
      nextline = aFileName.ReadElt
      left7 = nextline.left(7)
      if (nextline = nil) then
        break
      elseif (left7 = "Manning") then
        break
      else
      coordinates = list.make
      nextlist = nextline.astokens(" ")
      for each a in nextlist
        if (unit = "meters") then
          coordinates.add(a.asNumber*0.3048)
        else
          coordinates.add(a.asNumber)
        end
      end
      listcount = coordinates.count-1
      for each b in 0..listcount by 2
        xvalue = coordinates.get(b)
        xcoordinates.add(xvalue)
      end
      xcount = xcoordinates.count
      newxcount = xcount-1
      for each c in 1..listcount by 2
        zvalue = coordinates.get(c)
```

```
          zcoordinates.add(zvalue)
        end
        zcount = zcoordinates.count
        newzcount = zcount-1
        end
      end

'DETERMINE THE CHANNEL MINIMUM ELEVATION
      for each d in 0..newzcount
        depth = zcoordinates.get(d)
        newdepth = depth
        if (newdepth < MCdepth) then
          MCdepth = newdepth
          MCindex = d
        end
      end
      MCzvalues.add(MCdepth)
    end

'RETRIEVE THE BANK STATIONS
      if (left6 = "Bank S") then
        nextline = aFileName.ReadElt
        nextlist = nextline.astokens(" ")
        if (unit = "meters") then
          LBxvalue = (nextlist.get(0).asNumber)*0.3048
          RBxvalue = (nextlist.get(1).asNumber)*0.3048
        else
          LBxvalue = nextlist.get(0).asNumber
          RBxvalue = nextlist.get(1).asNumber
        end
        LBxvalues.add(LBxvalue)
        RBxvalues.add(RBxvalue)

'RETRIEVE THE DEPTHS AT BANK STATIONS
      for each e in 0..newxcount
        bankx = xcoordinates.get(e)
        if (bankx = LBxvalue) then
          LBzvalue = zcoordinates.get(e)
          LBzvalues.add(LBZvalue)
        break
        end
      end

      midchannel = list.make
      for each f in 0..newxcount
        MCz = zcoordinates.get(f)
        if (MCz = MCdepth) then
          MCxvalue = xcoordinates.get(f)
          midchannel.add(MCxvalue)
        end
      end
```

```
      midChannelCount = midChannel.count
      newmid = midChannelCount -1
      if (midChannelCount > 1) then
         MCxsum = 0
         for each val in 0..newmid
            MCxsum = MCxsum + midchannel.get(val)
         end
         avgMCx = MCxsum/midChannelCount
         MCxvalues.add(avgMCx)
      elseif (midChannelCount = 1) then
         MCxvalues.add(midchannel.get(0))
      else
      end

      for each g in 0..newxcount
         bankx = xcoordinates.get(g)
         if (bankx = RBxvalue) then
            RBzvalue = zcoordinates.get(g)
            RBzvalues.add(RBzvalue)
         break
         end
      end

'DETERMINE THE CROSS-SECTION COORDINATES (LATERAL & ELEVATION)
      mainxs =list.make
      mainx = MCxvalues.get(count-1)
      for each xitem in xcoordinates
         newxCoord = mainx-xitem
         mainxs.add(newxCoord)
      end
      _xStations.add(mainxs)
      _zStations.add(zcoordinates)
   end

'RETRIEVE THE WATER SURFACE ELEVATIONS
   if (left6 = "  W.S.") then
      thewslist = readline.astokens
               ("WSElev(ft)ElementLeftOBChannelRightOB ")
      if (unit = "meters") then
         z3 = (thewslist.get(2).asNumber)*0.3048
      else
         z3 = thewslist.get(2).asNumber
      end
      wselev.add(z3)

'RETRIEVE THE LEFT AND RIGHT FLOODPLAIN X (LATERAL) COORDINATES
      if (z3 > zcoordinates.get(0)) then
         lfloodx = xcoordinates.get(0)
      else
         for each j in MCindex..0
            z2 = zcoordinates.get(j)
```

```
          if (z3 < z2) then
            j2 = j
            break
          end
        end
        j1 = j2+1
        z1 = zcoordinates.get(j1)
        x2 = xcoordinates.get(j2)
        x1 = xcoordinates.get(j1)
        lfloodx = ((z3-z1)*(x2-x1)/(z2-z1)) + x1
      end
      LFxvalues.add(lfloodx)
      if (z3 > zcoordinates.get(newzcount)) then
        rfloodx = xcoordinates.get(newxcount)
      else
        for each k in MCindex..newxcount
          z2 = zcoordinates.get(k)
          if (z3 < z2) then
            k2 = k
            break
          end
        end
        k1 = k2-1
        z1 = zcoordinates.get(k1)
        x2 = xcoordinates.get(k2)
        x1 = xcoordinates.get(k1)
        rfloodx = ((z3-z1)*(x2-x1)/(z2-z1)) + x1
      end
      RFxvalues.add(rfloodx)
    end

'RETRIEVE THE CROSS-SECTION Y (REACH) COORDINATES
    if (left16 = "      Reach     ") then
      nextline = aFileName.ReadElt
      MCyvalue = 0
      ycounter = 0
      wscounter = 0
      while (true)
        nextline = aFileName.ReadElt
        ycoords = nextline.astokens(" ")
        if (ycoords.get(2) = "Culvert") then
          newLFxvalues.add(nil)
          mreach.add(nil)
          newRFxvalues.add(nil)
          newwselev.add(nil)
          ycounter = ycounter + 1
          types.add(ycoords.get(2))
        elseif (ycoords.get(2) = "Bridge") then
          newLFxvalues.add(nil)
          mreach.add(nil)
          newRFxvalues.add(nil)
```

```
            newwselev.add(nil)
            ycounter = ycounter + 1
            types.add(ycoords.get(2))
          else
            if (unit = "meters") then
              ReachValue = (ycoords.get(3).asNumber)*0.3048
            else
              ReachValue = ycoords.get(3).asNumber
            end
            MCyvalue = MCyvalue + ReachValue
            mreach.add(MCyvalue)
            newwselev.add(wselev.get(wscounter))
            newLFxvalues.add(LFxvalues.get(wscounter))
            newRFxvalues.add(RFxvalues.get(wscounter))
            types.add(nil)
            ycounter = ycounter + 1
            wscounter = wscounter + 1
            if (ycounter = count) then
              break
            end
          end
        end
      end
    end
av.ClearStatus
av.ClearMsg

'CREATE A VIRTUAL TABLE
'myFile = FN.Merge(av.GetProject.GetWorkDir.asString,"RASdata")
myFile = FileDialog.Put(" ".AsFileName,"*.*",
        "Specify File for Table Output")
if (nil = myFile) then
  exit
end
myFile.SetExtension("dbf")
theVtab = Vtab.MakeNew(myFile,dbase)
myTable = Table.Make(theVtab)

'CREATE THE FIELDS
StationField = Field.Make("Station",#FIELD_SHORT,8,0)
DescField = Field.Make("Description",#FIELD_VCHAR,30,0)
FloodElevField = Field.Make("FloodElev",#FIELD_FLOAT,8,1)
TypeField = Field.Make("Type",#FIELD_VCHAR,10,0)
LFloodXField = Field.Make("LFloodX",#FIELD_FLOAT,10,1)
LBankXField = Field.Make("LBankX",#FIELD_FLOAT,10,1)
LBankZField = Field.Make("LBankZ",#FIELD_FLOAT,10,1)
ChannelYField = Field.Make("ChannelY",#FIELD_FLOAT,10,1)
ChannelZField = Field.Make("ChannelZ",#FIELD_FLOAT,10,1)
RBankXField = Field.Make("RBankX",#FIELD_FLOAT,10,1)
RBankZField = Field.Make("RBankZ",#FIELD_FLOAT,10,1)
RFloodXField = Field.Make("RFloodX",#FIELD_FLOAT,10,1)
```

```
'ADD THE FIELDS TO THE TABLE
theVtab.AddFields({StationField,DescField,TypeField,FloodElevField,
        LFloodXField,LBankXField,LBankZField,ChannelYField,
        ChannelZField,RBankXField,RBankZField,RFloodXField})
newcount = count-1

'ADD THE TABLE DATA
for each i in 0..newcount
  rec = theVtab.AddRecord
  theVtab.SetValue(StationField,rec,StationList.get(i))
  if (newLFxvalues.get(i) = nil) then
    theVtab.SetValue(LFloodXField,rec,nil)
  else
    LFx = (MCxvalues.get(i) - newLFxvalues.get(i))
    theVtab.SetValue(LFloodXField,rec,LFx)
  end

  LBx = (MCxvalues.get(i) - LBxvalues.get(i))
  theVtab.SetValue(LBankXField,rec,LBx)

  theVtab.SetValue(LBankZField,rec,LBzvalues.get(i))
  if (mreach.get(i) = nil) then
    theVtab.SetValue(ChannelYField,rec,nil)
  else
    theVtab.SetValue(ChannelYField,rec,mreach.get(i))
  end

  theVtab.SetValue(ChannelZField,rec,MCzvalues.get(i))
  RBx = (MCxvalues.get(i) - RBxvalues.get(i))
  theVtab.SetValue(RBankXField,rec,RBx.abs)
  theVtab.SetValue(RBankZField,rec,RBzvalues.get(i))

  if (newRFxvalues.get(i) = nil) then
    theVtab.SetValue(RFloodXField,rec,nil)
  else
    RFx = (MCxvalues.get(i) - newRFxvalues.get(i))
    theVtab.SetValue(RFloodXField,rec,RFx.abs)
  end

  if (newwselev.get(i) = nil) then
    theVtab.SetValue(FloodElevField,rec,nil)
  else
    theVtab.SetValue(FloodElevField,rec,newwselev.get(i))
  end

  theVtab.SetValue(DescField,rec,describe.get(i))
  theVtab.SetValue(TypeField,rec,types.get(i))
end
theVtab.setEditable(false)
av.GetProject.AddDoc(myTable)
```

```
'------------------------------------------------------------
'------------------------------------------------------------
' Name: Terain3D.ave
' Headline:
' Self:
' Returns:  Shape file
' Description: Takes channel geometry profiles and creates a
'            theme that places them along a selected stream.
' Topics:
' Search Keys:
' Requires:  Global variables containing HEC-RAS geometry data,
'          Fthemes of stream centerline and boundary points.
' History:  Created by Eric Tate, 10/2/98
'         Last modification:  4/18/99
'------------------------------------------------------------
'------------------------------------------------------------

TheProject = av.GetProject
TheView = av.GetActiveDoc
TheDisplay = TheView.GetDisplay
TheDocs = TheProject.GetDocs
TabList = List.Make
for each d in TheDocs
  if (d.Is(Table)) then
     TabList.Add(d.GetName)
  end
end

ShapeLineList = List.Make
ShapePointList = List.Make
for each name in TheView.GetThemes
   if (name.is(Ftheme)) then
      if (name.GetFtab.FindField("shape").GetType=#FIELD_SHAPELINE) then
         ShapeLineList.add(name)
      elseif (name.GetFtab.FindField("shape").GetType=#FIELD_SHAPEPOINT) then
         ShapePointList.add(name)
      else
      end
   else
      continue
   end
end


PnthemeName = Msgbox.ChoiceAsString
(ShapePointList,"Choose the stream definition point theme.","Point Theme")
if (PnthemeName = nil) then
   exit
end
RivThemeName = Msgbox.ChoiceAsString
```

```
(ShapeLineList,"Choose the stream centerline theme.","Line Theme")
if (RivThemeName = nil) then
   exit
end
PtFtab = PnthemeName.GetFtab
NumPtRecords = PtFtab.GetNumRecords
RivFtab = RivThemeName.GetFtab

'--- IDENTIFY INPUT FIELDS
RivShpField = RivFtab.findField("Shape")
RivNameField = RivFtab.findField("Name")
RivLengthField = RivFtab.findField("Length")
PtShpField = PtFtab.FindField("Shape")
PtTypeField = PtFtab.FindField("Type")

'--- IDENTIFY THE STREAM GEOMETRY TABLE
Intablename = Msgbox.ChoiceAsString(tabList,
"Choose HEC-RAS geometry table of the stream of interest.","Geometry Table")
if (Intablename = nil) then
   exit
end

myAnswer = MsgBox.YesNo("In order to continue, the records in"++intablename.asString++
"associated with the stream definition points should be highlighted.  Continue?", "Continue?",
FALSE)
if (myAnswer = false) then
   exit
end

intable = theProject.findDoc(intablename)
invtab = inTable.getVtab
infields = inVtab.getFields
tabStationField = inVtab.findField("Station")
tabDescriptionField = inVtab.findField("Description")
tabTypeField = inVtab.findField("Type")
watelevField = inVtab.findField("FloodElev")
tabLeftBankField = inVtab.findField("LBankX")
tabLeftBankZField = inVtab.findField("LBankZ")
tabRightBankField = inVtab.findField("RBankX")
tabRightBankZField = inVtab.findField("RBankZ")
tabChanYField = inVtab.findField("ChannelY")
tabChanZField = inVtab.findField("ChannelZ")
numVtabRecs = inVtab.getNumRecords - 1

'ADD RAS GEOMETRY TABLE DATA TO THE BOUNDARY POINT SHAPEFILE
selectedRecords = inVtab.getSelection
numXsecRecords = selectedRecords.count
if (numXsecRecords = 0) then
   msgbox.info("No cross-sections selected in the RAS geometry table.","Error")
   exit
end
```

```
if (numXsecRecords <> numPtRecords) then
   msgbox.info
   ("The number of stream boundary points and selected RAS cross-sections should be
equal.","Error")
   exit
end

stalist = list.make
chanyloclist = list.make
staloclist = list.make
recordslist = list.make
for each selrec in selectedRecords
   recnum = selectedRecords.GetNextSet(selrec-1)
   prevrecnum = recnum-1
   nextrecnum = recnum+1
   recordslist.add(recnum)
   stalist.add(inVtab.returnValue(tabStationField,selrec))
   staloclist.add(inVtab.returnValue(tabDescriptionField,selrec))
   if (inVtab.returnValue(tabChanYField,recnum).asString="Number null") then
      if (recnum = 0) then
         theYLoc = 0
      elseif (recnum = numVtabRecs) then
         theYLoc = inVtab.returnValue(tabChanYField,numVtabRecs-1)
      else
         theYLoc=((inVtab.returnValue(tabChanYField,prevrecnum)+
         inVtab.returnValue(tabChanYField,nextrecnum))/2)-
         inVtab.returnValue(tabChanYField,0)
      end
   else
      if (recnum = 0) then
         theYLoc = 0
      elseif (recnum = numVtabRecs) then
         theYLoc = inVtab.returnValue(tabChanYField,numVtabRecs-1)
      else
         theYLoc = inVtab.returnValue(tabChanYField,recnum)-
         inVtab.returnValue(tabChanYField,0)
      end
   end
   chanyloclist.add(theYLoc)
end
recordslistcount = recordslist.count - 2

if(ptFTab.CanEdit) then
   ptFTab.SetEditable(true)
else
   msgbox.info("Can't edit point theme.","Error")
exit
end
ptStaField = ptFtab.findField("Station")
ptAlongField = ptFtab.findField("AlongPct")
ptLocField = ptFtab.findField("StaLocation")
```

```
ptRASField = ptFtab.findField("RASChannelY")
ptTypeField = ptFtab.findField("Type")
if (ptStaField <> nil) then
   ptFtab.RemoveFields({ptStaField})
   ptFtab.RemoveFields({ptLocField})
   ptFtab.RemoveFields({ptRASField})
end

ptFtab.AddFields({Field.Make("Station",#FIELD_DECIMAL,12,0)})
ptFtab.AddFields({Field.Make("StaLocation",#FIELD_VCHAR,40,0)})
ptFtab.AddFields({Field.Make("RASChannelY",#FIELD_DECIMAL,14,1)})
ptStaField = ptFtab.findField("Station")
ptLocField = ptFtab.findField("StaLocation")
ptRASField = ptFtab.findField("RASChannelY")
listIndex = 0
for each Fpt in ptFtab
   thesta = stalist.get(listIndex)
   ptFtab.setValue(ptStaField,Fpt,thesta)
   thestaloc = staloclist.get(listIndex)
   ptFtab.setValue(ptLocField,Fpt,thestaloc)
   theyvalue = chanyloclist.get(listIndex)
   ptFtab.setValue(ptRASField,Fpt,theyvalue)
   listIndex = listIndex+1
end
ptFtab.setEditable(false)

'DETERMINE THE CROSS-SECTION LINE EQUATIONS
tolerance = MsgBox.Input
("Choose distance (as a percentage of the stream length) for calculation of cross-section
orientation",
"Cross-Section Orientation", "5")
if (tolerance = nil) then
   exit
end
tol = tolerance.asNumber

'--- READ AND PROCESS DATA
streamname = rivFtab.returnValue(rivNameField,0)
'TerrainFileName = FN.Merge(av.GetProject.GetWorkDir.asString,"Terain3D")
TerrainFileName = FileDialog.Put("Terrain3D".asFilename,"*.shp","RAS Channel Shape File")
if(TerrainFileName = Nil)then
   exit
end
TerrainFileName.SetExtension("shp")
TerrainFtab = Ftab.MakeNew(TerrainFileName,polylineZ)
TerrainTheme = Ftheme.make(TerrainFtab)
theView.addTheme(TerrainTheme)
TerrainTheme.setVisible(true)

'CREATE FIELDS FOR THE NEW LINE ATTRIBUTE TABLE
TerrainFields = List.Make
```

```
TerrainFields.Add(Field.Make("Station",#FIELD_DECIMAL,12,0))
TerrainFields.Add(Field.Make("Location",#FIELD_VCHAR,40,0))
TerrainFields.Add(Field.Make("Length",#FIELD_DECIMAL,12,2))
TerrainFields.Add(Field.Make("LBankPct",#FIELD_DECIMAL,15,3))
TerrainFields.Add(Field.Make("CenterPct",#FIELD_DECIMAL,15,3))
TerrainFields.Add(Field.Make("RBankPct",#FIELD_DECIMAL,15,3))
TerrainFields.Add(Field.Make("LBankZ",#FIELD_DECIMAL,12,2))
TerrainFields.Add(Field.Make("CenterZ",#FIELD_DECIMAL,12,2))
TerrainFields.Add(Field.Make("RBankZ",#FIELD_DECIMAL,12,2))

TerrainFieldsc = TerrainFields.DeepClone
TerrainFtab.addFields(TerrainFieldsc)
if (TerrainFtab.CanEdit) then
   TerrainFtab.SetEditable(true)
else
   msgbox.info("Can't edit the output theme.","Error")
   exit
end

'IDENTIFY FIELDS FOR WRITING
shpField = TerrainFtab.FindField("Shape")
idfield = TerrainFtab.FindField("Station")
locate = TerrainFtab.Findfield("Location")
LengthField = TerrainFtab.Findfield("Length")
LeftPctField = TerrainFtab.Findfield("LBankPct")
CenterPctField = TerrainFtab.Findfield("CenterPct")
RightPctField = TerrainFtab.Findfield("RBankPct")
LeftZField = TerrainFtab.Findfield("LBankZ")
CenterZField = TerrainFtab.Findfield("CenterZ")
RightZField = TerrainFtab.Findfield("RBankZ")

'CREATE THE BOUNDED RIVER POLYLINE
for each j in ptFtab
   boundtest = ptFtab.ReturnValue(ptTypeField,j)
   if (boundtest = "Upstream boundary") then
      uppoint = ptFtab.ReturnValue(ptShpField,j)
   elseif (boundtest = "Downstream boundary") then
      downpoint = ptFtab.ReturnValue(ptShpField,j)
   else
      continue
   end
end
rivpolyline = rivFtab.ReturnValue(rivshpField,0)
rivpts = rivPolyline.asMultipoint.asList

'ELIMINATE POINTS DOWNSTREAM OF THE DOWNSTREAM BOUNDARY
dbound = rivPolyline.PointPosition(downpoint)
lowpoint = rivPolyline.PointPosition(rivpts.get(0))
while (dbound > lowpoint)
   rivpts.remove(0)
   lowpoint = rivPolyline.PointPosition(rivpts.get(0))
```

```
end
rivpts.insert(downpoint)


'ELIMINATE POINTS UPSTREAM OF THE UPSTREAM BOUNDARY
upbound = rivPolyline.PointPosition(uppoint)
rivptscount = rivpts.count - 1
highpoint = rivPolyline.PointPosition(rivpts.get(rivptscount))
while (upbound < highpoint)
   rivpts.remove(rivptscount)
   rivptscount = rivpts.count - 1
   highpoint = rivPolyline.PointPosition(rivpts.get(rivptscount))
end
rivpts.add(uppoint)

newrivline = polyline.make({rivpts})
newrivline.flip
index = 0
totRASlength=ptFtab.returnvalue(ptRASField,recordslistcount+1)-
          ptFtab.returnvalue(ptRASField,0)
topptpos = ptFtab.returnValue(ptAlongField,0)
botptpos = ptFtab.returnValue(ptAlongField,recordslistcount+1)
totRealLength = rivFtab.returnValue(rivLengthField,0)*
            (topptpos-botptpos)/100
proportionList = list.make
oldpct = 0

lBankpts = list.make
centerpts = list.make
rBankpts = list.make
loclist = list.make

'DO A LOOP BETWEEN EACH SET OF STREAM BOUNDARY POINTS
for each element in 0..recordslistcount
   nextelement = element + 1
   upChanY = ptFtab.returnValue(ptRASField,element)
   downChanY = ptFtab.returnValue(ptRASField,nextelement)
   RASlength = downChanY-upChanY
   upptpos = ptFtab.returnValue(ptAlongField,element)
   downptpos = ptFtab.returnValue(ptAlongField,nextelement)
   realLength = rivFtab.returnValue(rivLengthField,0)*(upptpos-downptpos)/100
   proportion = realLength/RASlength
   proportionList.add(proportion)

   for each rec in recordslist.get(element)..recordslist.get(nextelement)
      prevrec = rec-1
      nextrec = rec+1
      RASstation = inVtab.returnValue(tabStationField,rec)
      LeftBank = inVtab.returnValue(tabLeftBankField,rec)
      RightBank = 0 - (inVtab.returnValue(tabRightBankField,rec))
```

```
'AT INTERMEDIATE POINTS, DON'T WRITE CROSS-SECTIONS TWICE

    if (index > 0) then
      if (RASstation = TerrainFtab.returnValue(idField,index-1)) then
        continue
      end
    end

'USE THE PREVIOUS PROPORTION IF YOU'RE AT THE RECORD RIGHT AFTER AN
INTERMEDIATE
'POINT THAT'S ALSO A BRIDGE OR CULVERT
    proportion = realLength/RASlength
    if (element>0) then
      if ((prevrec=recordslist.get(element)) and
        (inVtab.returnValue(tabChanYField,prevrec).asString="Number null")) then
          proportion = proportionList.get(element-1)
      end
    end

'DETERMINE THE RAS DISTANCE ALONG THE STREAM
    if (inVtab.returnValue(tabChanYField,rec).asString="Number null") then
      if (rec = 0) then
        YChanLoc = 0
        floodelev = nil
      elseif (recordslist.get(recordslistcount+1) = rec) then
        YChanLoc = inVtab.returnValue(tabChanYField,prevrec)+
              inVtab.returnValue(tabChanYField,0)
        floodelev = inVtab.returnvalue(watelevField,prevrec)
      else
        YChanLoc = ((inVtab.returnValue(tabChanYField,prevrec)+
               inVtab.returnValue(tabChanYField,nextrec))/2)-
               inVtab.returnValue(tabChanYField,0)
        floodelev = (inVtab.returnvalue(watelevField,prevrec)+
               inVtab.returnvalue(watelevField,nextrec))/2
      end
    else
      floodelev = inVtab.returnvalue(watelevField,rec)
      if (recordslist.get(recordslistcount+1) = rec) then
        YChanLoc = inVtab.returnValue(tabChanYField,rec)
      else
        YChanLoc = inVtab.returnValue(tabChanYField,rec)-
              inVtab.returnValue(tabChanYField,0)
      end
    end

'CALCULATE THE CROSS-SECTION LOCATION ALONG THE STREAM
    loclist.add(inVtab.returnValue(tabTypeField,rec))
    rivmile = YChanLoc - ptFtab.returnValue(ptRASField,element)
    pct = ((rivmile*proportion*100)/totrealLength) + oldpct
    pt1 = newrivline.along(pct)
```

```
        centerptx = pt1.getx
        centerpty = pt1.gety
        centerptz = inVtab.returnValue(tabChanZField,rec)
        pt2 = newrivline.along(pct+tol)
        if (pct = 0) then
            pt3 = pt1
        else
            pt3 = newrivline.along(pct-tol)
        end

'CALCULATE THE SLOPE
        if (pt2.getx-pt3.getx = 0) then
            m2 = 0
        else
            m1 = (pt2.gety-pt3.gety)/(pt2.getx-pt3.getx)
            m2 = (-1)/m1
        end
        theta = m2.atan

        newrec = TerrainFtab.AddRecord
        terrainPts = list.make
        zindex = 0
        xstationList = _xStations.get(rec)
        zstationList = _zStations.get(rec)
            for each xstation in xstationList
                terrainptx = centerptx + (xstation*theta.cos)
                terrainpty = centerpty + (xstation*theta.sin)
                terrainptz = zstationList.get(zindex)
                terrainpt = pointZ.make(terrainptx,terrainpty,terrainptz)
                terrainPts.add(terrainpt)
                zindex = zindex + 1
            end

        xsecline = polylineZ.make({terrainPts})
        loc = loclist.get(index)
        index = index + 1

'DETERMINE BANK STATION LOCATIONS
        LeftBankx = centerptx + (LeftBank*theta.cos)
        LeftBanky = centerpty + (LeftBank*theta.sin)
        LeftBankz = inVtab.ReturnValue(tabLeftBankZField,rec)
        LeftBank3DPoint = PointZ.make(LeftBankx,LeftBanky,LeftBankz)
        lBankpts.Add(LeftBank3DPoint)

        centerptz = inVtab.ReturnValue(tabChanZField,rec)
        Center3DPoint = PointZ.make(centerptx,centerpty,centerptz)
        centerpts.Add(Center3DPoint)
        RightBankx = centerptx + (RightBank*theta.cos)
        RightBanky = centerpty + (RightBank*theta.sin)
        RightBankz = inVtab.ReturnValue(tabRightBankZField,rec)
        RightBank3DPoint = PointZ.Make(RightBankx,RightBanky,RightBankz)
```

```
        rBankpts.Add(RightBank3DPoint)
        LeftBankPct = xsecline.PointPosition(LeftBank3DPoint)
        CenterBankPct = xsecline.PointPosition(Center3DPoint)
        RightBankPct = xsecline.PointPosition(RightBank3DPoint)

'DETERMINE THE CROSS-SECTION LENGTH
        startpoint = terrainPts.get(0)
        x1 = startpoint.getX
        y1 = startpoint.getY
        endpoint = terrainPts.get(zindex-1)
        x2 = endpoint.getX
        y2 = endpoint.getY
        XsecLength = (((x1-x2)^2)+((y1-y2)^2))^(0.5)

'WRITE THE CROSS-SECTION TO THE FTAB
        TerrainFtab.SetValue(shpField,newrec,xsecline)
        TerrainFtab.SetValue(idfield,newrec,RASstation)
        TerrainFtab.SetValue(locate,newrec,loc)
        TerrainFtab.SetValue(lengthField,newrec,XsecLength)
        TerrainFtab.SetValue(LeftpctField,newrec,LeftBankPct)
        TerrainFtab.SetValue(CenterpctField,newrec,CenterBankPct)
        TerrainFtab.SetValue(RightpctField,newrec,RightBankPct)
        TerrainFtab.SetValue(LeftZField,newrec,LeftBankz)
        TerrainFtab.SetValue(CenterZField,newrec,centerptz)
        TerrainFtab.SetValue(RightZField,newrec,RightBankz)
    end
    theView.Draw(theDisplay)
    oldpct = oldpct + ((rivmile*proportion*100)/totrealLength)
end

'HIGHLIGHT RECORDS FOR BRIDGES AND CULVERTS
theBitmap = TerrainFtab.GetSelection
theBitmap.SetAll
for each selrec in theBitmap
    description = TerrainFtab.ReturnValue(locate,selrec)
    selrecnum = theBitmap.GetNextSet(selrec-1)
    if (description.Count > 0) then
        theBitmap.Clear(selrecnum)
        TerrainFtab.RemoveRecord(selrecnum)
    end
end
theBitmap.ClearAll
RemoveFieldsList = List.Make
RemoveFieldsList.Add(locate)
TerrainFTab.RemoveFields(RemoveFieldsList)
TerrainFtab.SetEditable(false)

msgbox.info
("If any of the cross-sections intersect, re-run this script and choose a greater distance for orientation calculation.",
"Warning:  Check the cross-sections")
```

```
’----------------------------------------------------------------
’----------------------------------------------------------------
’ Name: Water3D.ave
’ Headline:
’ Self:
’ Returns:  Shape file
’ Description: Takes channel geometry profiles and creates a
’             water surface polylineZ theme.
’ Topics:
’ Search Keys:
’ Requires:  HEC-RAS geometry table water surface data,
’           PolylineZ Ftheme of channel terrain.
’ History:  Created by Eric Tate, 10/4/98
’          Last modification:  3/6/99
’----------------------------------------------------------------
’----------------------------------------------------------------

TheProject = av.GetProject
TheView = av.GetActiveDoc
TheDocs = TheProject.GetDocs
TabList = List.Make
for each d in theDocs
   if (d.Is(Table)) then
      TabList.add(d.getname)
   end
end

FthemeList = List.Make
for each name in TheView.GetThemes
   if (name.is(Ftheme)) then
      if (name.GetFtab.FindField("shape").GetType=#FIELD_SHAPELINE) then
         FThemeList.add(name)
      end
   else
      continue
   end
end
TerrainThemeName = Msgbox.ChoiceAsString(FthemeList,"Choose the cross-section line
theme.","Line Theme")
if (TerrainThemeName = nil) then
   exit
end
TerrainFtab = TerrainThemeName.GetFtab
TerrainshpField = TerrainFtab.FindField("Shape")
TerrainStaField = TerrainFtab.FindField("Station")
TerrainLengthField = TerrainFtab.FindField("Length")
TerrainCenterField = TerrainFtab.FindField("CenterPct")

’--- IDENTIFY THE STREAM GEOMETRY TABLE
intablename = Msgbox.ChoiceAsString(tabList,
```

```
"Choose HEC-RAS geometry table of the stream of interest.","Geometry Table")
if (intablename = nil) then
   exit
end

intable = TheProject.findDoc(intablename)
inVtab = inTable.getVtab
theBitmap = inVtab.getDefBitmap
infields = inVtab.getFields
tabStationField = inVtab.findField("Station")
tabLocationField = inVtab.findField("Location")
watelevField = inVtab.findField("FloodElev")
leftFloodField = inVtab.findField("LFloodX")
rightFloodField = inVtab.findField("RFloodX")

'FIND THE FIRST VTAB RECORD ASSOCIATED WITH THE FIRST CROSS-SECTION
sta1 = TerrainFtab.returnValue(TerrainStaField,0)
for each record in inVtab
   sta2 = inVtab.returnvalue(tabStationField,record)
   if (sta1 = sta2) then
      recnum = theBitmap.GetNextSet(record-1)
      break
   end
end

'--- READ AND PROCESS DATA
'WaterName = FN.Merge(av.GetProject.GetWorkDir.asString,"Water3D")
WaterName = FileDialog.Put("Water3D".asFilename,"*.shp","3D Water Surface Shape File")
if(WaterName = Nil)then
   exit
end
WaterName.SetExtension("shp")
WaterFtab = Ftab.MakeNew(WaterName,polylineZ)

'CREATE FIELDS FOR THE NEW POLYLINEZ ATTRIBUTE TABLE
outFields = List.Make
outFields.Add(Field.Make("Station",#FIELD_DECIMAL,10,0))
outFields.Add(Field.Make("Elevation",#FIELD_DECIMAL,10,2))
outFields.Add(Field.Make("Length",#FIELD_DECIMAL,10,2))
outFieldsc = outFields.DeepClone
WaterFtab.addFields(outFieldsc)
if (WaterFtab.CanEdit) then
   WaterFtab.SetEditable(true)
else
   msgbox.info("Can't edit the output theme.","Error")
   exit
end

'IDENTIFY FIELDS FOR WRITING
shpField = WaterFtab.FindField("Shape")
idField = WaterFtab.FindField("Station")
```

```
WaterElevField = WaterFtab.FindField("Elevation")
WaterLengthField = WaterFtab.FindField("Length")

'DETERMINE THE 3D WATER PROFILE
TerrainRecs = TerrainFtab.GetNumRecords - 1
for each rec in TerrainFtab
   WaterElev = inVtab.returnValue(watelevField,recnum)
   if (WaterElev.asString="Number Null") then
      recnum = recnum + 1
      WaterElev = inVtab.returnValue(watelevField,recnum)
   end
   Station = inVtab.returnValue(tabStationField,recnum)
   xsection = TerrainFtab.ReturnValue(TerrainshpField,rec)
   TerrainLength = TerrainFtab.ReturnValue(TerrainLengthField,rec)
   CenterPct = TerrainFtab.ReturnValue(TerrainCenterField,rec)
   CenterLength = (CenterPct/100)*TerrainLength
   LeftFloodBound = inVtab.ReturnValue(LeftFloodField,recnum)
   LeftLength = CenterLength - LeftFloodBound
   LeftPct = (LeftLength/TerrainLength)*100
   LeftPoint = xsection.Along(LeftPct)

   RightFloodBound = inVtab.ReturnValue(RightFloodField,recnum)
   RightLength = CenterLength + RightFloodBound
   RightPct = (RightLength/TerrainLength)*100
   RightPoint = xsection.Along(RightPct)
   WaterLength = ((RightPct-LeftPct)/100)*TerrainLength

   NewWaterElev = WaterElev
   ptList1 = xsection.asList
   ptList2 = ptList1.get(0)
   waterpts = list.make
   StartPoint = pointZ.make(LeftPoint.Getx,LeftPoint.Gety,NewWaterElev)
   waterpts.add(StartPoint)
   for each pnt in ptList2
      alongpct = xsection.PointPosition(pnt)
      if ((alongpct<LeftPct) or (alongpct>RightPct)) then
         continue
      else
         pnt.SetZ(NewWaterElev)
         waterpts.add(pnt)
      end
   end
   EndPoint = pointZ.make(RightPoint.Getx,RightPoint.Gety,NewWaterElev)
   waterpts.add(EndPoint)
   waterline = polylineZ.make({waterpts})

   newrec = WaterFtab.addRecord
   WaterFtab.SetValue(shpField,newrec,waterline)
   WaterFtab.SetValue(idField,newrec,Station)
   WaterFtab.SetValue(WaterElevField,newrec,NewWaterElev)
   WaterFtab.SetValue(WaterLengthField,newrec,WaterLength)
```

```
    recnum = recnum + 1
end

WaterFtab.setEditable(false)
WaterTheme = Ftheme.make(WaterFtab)
TheView.addTheme(WaterTheme)
WaterTheme.setVisible(true)
```

References

Avery, T.E. and G.L. Berlin. 1992. *Fundamentals of Remote Sensing and Airphoto Interpretation, 5th ed*. Macmillan Publishing Company, New York, NY.

Beavers, M.A. 1994. *Floodplain Determination Using HEC-2 and Geographic Information Systems*. Masters Thesis, Department of Civil Engineering, The University of Texas at Austin.

Campbell, J.B. *Introduction to Remote Sensing*, 2nd ed. The Guilford Press, New York, NY, 1996.

Carter, J.R. (1988). "Digital Representations of Topographic Surfaces." Photogrammetric Engineering and Remote Sensing, 54(11), 1577-1580.

CES (Continuing Engineering Studies). 1998. *HEC River Analysis System: River and Flood Plain Hydraulics*. The University of Texas at Austin, College of Engineering.

DHI (Danish Hydraulic Institute). 1999. *MIKE 11 GIS*. Internet site, http://www.dhi.dk/mike11/m11gis/M11GIS.htm.

Djokic, D. and Maidment, D.R. (1991). "Terrain Analysis for Urban Stormwater Modeling." Hydrological Processes, 5, 115-124.

EMRL (Environmental Modeling Research Laboratory). 1999. *SMS*. Internet site, http://www.emrl.byu.edu/sms.htm.

ESRI (Environmental Systems Research Institute). 1999. *ArcView GIS Extensions*. Internet site, http://www.esri.com/software/arcview/extensions/index.html.

FEMA (Federal Emergency Management Agency). 1999a. *National Flood Insurance Program*. Internet site, http://www.fema.gov/nfip/summary.htm.

FEMA. 1999b. *Map Service Center Products*. Internet site, http://www.fema.gov/msc/product.htm.

HEC (Hydrologic Engineering Center). 1997. *HEC-RAS River Analysis System: Hydraulic Reference Manual*. Hydrologic Engineering Center, Davis, CA.

Jenson, S.K. and Dominique, J.O. (1988). "Extracting Topographic Structure from Digital Elevation Data for Geographic Information Systems analysis." Photogrammetric Engineering and Remote Sensing, 54(11), 1593-1600.

Jones, J.L., T.L. Haluska, A.K. Williamson, and M.L. Erwin. 1998. Updating Flood Maps Efficiently: Building on Existing Hydraulic Information and Modern Elevation Data with a GIS. USGS Open-File Report 98-200.

Lillesand, T.M. and R.W. Kiefer. *Remote Sensing and Image Interpretation*. John Wiley & Sons, New York, NY, 1994.

Maidment, D.R. 1999. *Exercise 1: Introduction to ArcView*. Internet site, http://www.ce.utexas.edu/prof/maidment/giswr98/ex198/WEBFILES/Ex198.html

O'Callahan, J.F. and Mark, D.M. (1984). "The Extraction of Drainage Networks from Digital Elevation Data." Computer Vision, Graphics and Image Processing, 28, 323-344.

Prasuhn, A.L. *Fundamentals of Hydraulic Engineering*. Oxford University Press, New York, NY, 1992.

Smith, P. *Hydrologic Data Development System*. Department of Civil Engineering, The University of Texas at Austin. 1995. CRWR Online Report 95-1.

Strange, I.W. 1998. *Development of a GIS Database for Estimating Biogenic Hydrocarbon Emissions in North Central Texas*. Masters Thesis, Department of Civil Engineering, The University of Texas at Austin.

Streeter, V.L. and E.B. Wylie. *Fluid Mechanics*, 8^{th} ed. McGraw-Hill Publishing Company, New York, NY, 1985.

United States Geological Survey. 1998. *Geographic Information Systems*. Internet site, http://www.usgs.gov/research/gis/title.html.

Vincent, R.K. *Fundamentals of Geological and Environmental Remote Sensing*. Prentice-Hall Inc., Upper Saddle River, NJ, 1997.

Vita

Eric Christopher Tate was born in Cincinnati, Ohio on May 20, 1971, the son of Joanne LaVerne Tate and Charles Earl Tate. After graduating from MacArthur High School in San Antonio, Texas in 1989, he matriculated at Rice University in Houston, Texas. During his stay at Rice, he spent one semester studying at Flinders University in Adelaide, Australia through a study abroad program. He received a degree of Bachelor of Science in Civil Engineering from Rice University in May 1994. For the next three years, he was employed as an environmental engineer at the consulting firm Roy F. Weston, Inc. in Houston, Texas. In August 1997, he entered the Graduate School at the University of Texas at Austin.

Permanent address: 2370 Orange Avenue #3
Costa Mesa, CA 92627

This thesis was typed by the author.