



2010

# STREAMFLOW PREDICTION USING GIS FOR THE KENTUCKY RIVER BASIN

Bakkiyalakshmi Palanisamy

*University of Kentucky*, [bagya7780@hotmail.com](mailto:bagya7780@hotmail.com)

**[Click here to let us know how access to this document benefits you.](#)**

---

## Recommended Citation

Palanisamy, Bakkiyalakshmi, "STREAMFLOW PREDICTION USING GIS FOR THE KENTUCKY RIVER BASIN" (2010).

*University of Kentucky Doctoral Dissertations*. 53.

[https://uknowledge.uky.edu/gradschool\\_diss/53](https://uknowledge.uky.edu/gradschool_diss/53)

This Dissertation is brought to you for free and open access by the Graduate School at UKnowledge. It has been accepted for inclusion in University of Kentucky Doctoral Dissertations by an authorized administrator of UKnowledge. For more information, please contact [UKnowledge@lsv.uky.edu](mailto:UKnowledge@lsv.uky.edu).

ABSTRACT OF DISSERTATION

Bakkiyalakshmi Palanisamy

The Graduate School

University of Kentucky

2010

STREAMFLOW PREDICTION  
USING GIS FOR THE  
KENTUCKY RIVER BASIN

---

ABSTRACT OF DISSERTATION

---

A dissertation submitted in partial fulfillment of the  
requirements for the degree of Doctor of Philosophy in the  
College of Engineering  
at the University of Kentucky

By

Bakkiyalakshmi Palanisamy

Lexington, Kentucky

Director: Dr. Stephen R. Workman, Professor of Agricultural Engineering  
Lexington, Kentucky

2010

Copyright © Bakkiyalakshmi Palanisamy 2010

## ABSTRACT OF DISSERTATION

### STREAMFLOW PREDICTION USING GIS FOR THE KENTUCKY RIVER BASIN

The study was aimed at developing a simple methodology for flow prediction in ungauged basins using existing data resources. For this purpose, the streamflow measurements across the Kentucky River Basin located in Kentucky, USA were obtained from United States Geological Survey (USGS) archive. The flow transferring characteristics of the subbasins of the Kentucky River Basin were obtained by combining downstream and upstream stream gauges. The flow transferring function thus derived were related to watershed, channel and flow characteristics of the subbasins by multiple regression analysis. The gauge pairs were divided into two classes of subbasins representing Upper and Lower Kentucky, which were characterized mainly by the geology of the watersheds. The regression models corresponding to the two groups of subbasins were applied to example gauge pairs to evaluate the efficiency of the proposed model to predict streamflow in downstream channel. The estimated hydrographs agreed with the observed hydrographs with the performance efficiency of greater than 90%. The proposed method was tested for its applicability in first-order streams in the Goose Creek, a tributary to the Kentucky River. The overland flow component for the first-order streams was determined using TOPMODEL with topography, soil and climatic factors as inputs. The overland flow was routed to the Goose Creek outlet using the transfer function obtained from measured flow records. The simulated hydrographs were reproduced with 80% accuracy when compared with the observed hydrographs. The flow prediction of first-order ungauged streams was automated by the back-calibration algorithm. The algorithm is supported by the Shuffled Complex Evolution - University of Arizona algorithm for its optimization routine. The back-calibration procedure optimizes each first-order stream with the aid of the flow transferring function. The back-calibration procedure was imbedded in a Visual Basic.NET environment to automatically predict flow on a daily time scale and predicted was published on the internet using ESRI Arc Internet Mapping Server (ArcIMS). The project thus provides daily streamflow estimation for streams on a first-order level on every day basis, which will facilitate flow prediction of streams regardless of the size of the watersheds.

KEYWORDS: Flow transfer function, Observed hydrographs, Ungauged basins, GIS,  
ArcIMS

Bakkiyalakshmi Palanisamy

---

Student's Signature

06/30/2010

---

Date

STREAMFLOW PREDICITON  
USING GIS FOR THE  
KENTUCKY RIVER BASIN

By

Bakkiyalakshmi Palanisamy

Dr. Stephen R. Workman

---

Director of Dissertation

Dr. Dwayne R. Edwards

---

Director of Graduate Studies

06/30/2010

---

Date

## RULES FOR THE USE OF DISSERTATIONS

Unpublished dissertations submitted for the Doctor's degree and deposited in the University of Kentucky Library are as a rule open for inspection, but are to be used only with due regard to the rights of the authors. Bibliographical references may be noted, but quotations or summaries of parts may be published only with the permission of the author, and with the usual scholarly acknowledgments.

Extensive copying or publication of the dissertation in whole or in part also requires the consent of the Dean of the Graduate School of the University of Kentucky.

A library that borrows this dissertation for use by its patrons is expected to secure the signature of each user.

Name

Date

---

---

---

---

---

---

---

---

---

---

---

DISSERTATION

Bakkiyalakshmi Palanisamy

The Graduate School  
University of Kentucky

2010



STREAMFLOW PREDICITON  
USING GIS FOR THE  
KENTUCKY RIVER BASIN

---

DISSERTATION

---

A dissertation submitted in partial fulfillment of the  
requirements for the degree of Doctor of Philosophy in the  
College of Engineering  
at the University of Kentucky

By

Bakkiyalakshmi Palanisamy

Lexington, Kentucky

Director: Dr. Stephen R. Workman, Professor of Agricultural Engineering

Lexington, Kentucky

2010

Copyright © Bakkiyalakshmi Palanisamy 2010

Dedicated to my beloved husband, my mother and brothers

## Table of Contents

LIST OF FIGURES .....	V
LIST OF TABLES .....	VII
1. INTRODUCTION .....	1
1.1. Statement of the problem .....	1
1.2. Study Area .....	1
1.3. Scope of the study .....	2
1.4. Structure of the Dissertation .....	4
1.5. Literature Review .....	5
1.5.1. History of system transfer function identification .....	5
1.5.2. History of regionalization techniques .....	7
1.5.3. History of hydrologic models and parameterization .....	9
1.6. Summary of literature review .....	11
2. DEVELOPMENT OF FLOW TRANSFERRING CHARACTERISTICS FROM STREAMFLOW RECORDS .....	12
2.1. Abstract .....	12
2.2. Introduction .....	13
2.3. Literature Review .....	14
2.3.1. Non-Parametric methods in the Frequency domain .....	14
2.3.2. Parametric methods in the Time domain .....	15
2.3.3. Analytical procedures .....	15
2.3.4. Application of transfer function in flow estimation of ungauged catchments .....	16
2.4. Materials and Methods .....	19
2.4.1. Study area .....	19
2.4.2. Derivation of transfer function .....	19
2.4.3. Multiple regression analysis .....	21
2.5. Results and Discussion .....	23
2.5.1. Analysis of Regression1 watersheds .....	23
2.5.2. Analysis of Regression2 watersheds .....	23
2.5.3. Validation of the proposed model .....	24
2.5.3.1. Regression1 analysis .....	25
2.5.3.2. Regression2 analysis .....	27
3. FLOW ROUTING USING TRANSFER FUNCTION DERIVED FROM STREAMFLOW RECORDS .....	54
3.1. Abstract .....	54
3.2. Introduction .....	55
3.3. Objectives .....	57
3.4. Materials and Methods .....	57
3.4.1. Description of the study area .....	57
3.4.2. Dataset preparation .....	58

3.4.3. Channel routing method.....	60
3.5. Results and Discussion .....	62
3.6. Conclusions.....	64
4. A GIS FRAMEWORK FOR PUBLICATION OF STREAMFLOW FOR GOOSE CREEK IN KENTUCKY RIVER BASIN.....	77
4.1. Abstract.....	77
4.2. Introduction.....	78
4.3. Materials and Methods.....	80
4.3.1. Description of the study area .....	80
4.3.2 Dataset preparation .....	80
4.3.3. Back-calibration method .....	81
4.4. Results and Discussions.....	82
4.4.1. Back-calibration.....	82
4.4.2. Publication of streamflow value on the Internet using ArcIMS.....	82
4.5. Conclusions.....	82
5. GENERAL OBSERVATIONS AND CONCLUSIONS OF THE RESULTS OF THE STUDY .....	91
5.1. Derivation of the flow transferring function based on observed streamflow records .....	91
5.2. Development of flow routing procedure using Average Transfer Function derived from observed flow records .....	91
5.3. Development of back-calibration method for automated flow prediction algorithm.....	91
APPENDICES .....	93
Appendix A - Program to process NEXRAD dataset .....	93
Appendix B - Programs to process NCDC raingauge measured precipitation .....	110
Appendix C - Programs to process SSURGO soil database .....	122
Appendix D - Programs to create Topographic Wetness Index file .....	133
Appendix E - Programs to determine percent urban.....	147
Appendix F - Program to write upstream and downstream watersheds for flow routing .....	154
Appendix G - Programs to automate data retrieval.....	158
Appendix H - Program for ArcIMS flow value display.....	166
Appendix I - Program for routing method .....	169
REFERENCES .....	174
VITA.....	180

## List of Figures

Figure 1-1. Kentucky River basin and its major karst features and surface water monitoring stations .....	3
Figure 2-1. Kentucky River Basin: Elevation and surface water monitoring stations .....	31
Figure 2-2a. Evaluation of proposed Average Transfer Function model for Red river near Hazel Green for year 2001 from March through June .....	32
Figure 2-2b. Evaluation of proposed Average Transfer Function model for Red river near Hazel Green for year 2001 from July through September .....	33
Figure 2-2c. Evaluation of proposed Average Transfer Function model for Red river near Hazel Green for year 2001 from October through November .....	34
Figure 2-3. Delineated basins, Karst regions, and ecoregions of the Kentucky River Basin .....	35
Figure 2-4. Studentized residual plot of USGS gauge pairs after removing outliers.....	36
Figure 2-5. Residual plot of Upper Kentucky gauges.....	37
Figure 2-6. Residual plot of Lower Kentucky basins .....	38
Figure 2-7. Runoff event analysis at KY River at Lock 13 .....	39
Figure 2-8. Runoff event analysis at KY River at Lock 6 .....	40
Figure 2-9a. Seasonal hydrograph analysis of Red River near Hazel Green for year 2001 from March through June .....	41
Figure 2-9b. Seasonal hydrograph analysis of Red River near Hazel Green for year 2001 from July through September .....	42
Figure 2-9c. Seasonal hydrograph analysis of Red River near Hazel Green for year 2001 from October through November .....	43
Figure 2-10. Yearly hydrograph analysis of KY River at Lock 7.....	44
Figure 2-11. Inflow hydrograph analysis of KY River at Lock 8.....	45
Figure 2-12. Runoff event analysis of Town Branch Creek .....	46
Figure 2-13. Runoff event analysis of South Elkhorn Creek .....	47
Figure 2-14a. Seasonal hydrograph analysis of Elkhorn Creek for year 2007 from May through June .....	48
Figure 2-14b. Seasonal hydrograph analysis of Elkhorn Creek for year 2007 from July through September .....	49
Figure 2-14c. Seasonal hydrograph analysis of Elkhorn Creek for year 2007 from October through December.....	50
Figure 2-15. Yearly hydrograph analysis of KY River at Lock 2.....	51
Figure 2-16. Inflow hydrograph analysis of North Elkhorn Creek.....	52

Figure 3-1. Goose Creek basin and surface water monitoring stations.....	65
Figure 3-2. Measured and estimated Potential Evapotranspiration .....	66
Figure 3-3. Simulated hydrographs using proposed routing procedure without storage component .....	67
Figure 3-4. Precipitation distribution across the watershed for 2005 and 2006 runoff events.....	68
Figure 3-5. 2005 runoff events.....	69
Figure 3-6. 2006 runoff events.....	70
Figure 3-7. 2006 runoff events.....	71
Figure 3-8. 2006 runoff events.....	72
Figure 3-9. 2006 runoff events.....	73
Figure 3-10. Validation runoff event with NEXRAD and raingauge measured precipitation.....	74
Figure 4-1. Database structure for Flow Prediction algorithm .....	84
Figure 4-2. Flow chart of the back calibration method.....	85
Figure 4-3. Validation of back-calibration algorithm for subbasin1.....	86
Figure 4-4. Validation of back-calibration algorithm for subbasin2.....	87
Figure 4-5. Validation of back-calibration algorithm with calibrated and uncalibrated simulated flow - Event 1 .....	88
Figure 4-6. Validation of back-calibration algorithm with calibrated and uncalibrated simulated flow- Event 2 .....	89
Figure 4-7. Result from the automated flow prediction method.....	90

## List of Tables

Table 2.1. Example reaches used in validation of the proposed method.....	53
Table 3-1. Calibrated parameters.....	75
Table 3-2. Coefficient of Efficiency for selected runoff events.....	76

## 1. Introduction

### 1.1. Statement of the problem

Streamflow is important for understanding the processes occurring within a hydrologic boundary that varies in physical, climatological, and geological characteristics. Any change in watershed characteristics alters its flow generation behavior. The streamflow measurements impart knowledge about the hydrologic cycle of a watershed. For example, a streamflow time series with continuous surface water supply could be descriptive of a perennial stream with constant baseflow contribution. The constant baseflow is a typical characteristic of a forest or land uses with a well-aerated soil structure. On the other hand, the streamflow time series with flashy rising limb represents an urban watershed with impermeable landuse and land cover.

Estimation of a physically plausible streamflow for ungauged catchments has long been a challenge in the field of hydrology due to the lack of flow measurements for validation of a hydrologic model. The variability of flow increases in karst catchments for which the variables to define underground flow paths are hard to obtain and therefore the lack of stream gauges in these basins leave the modeler with limited information about the characteristics of flow processes. Thus, the uncertainty of a hydrologic model increases with the increasing number of unknown parameters for catchments dominated by karst geology.

Since measured streamflow gathers and secures information about catchment attributes, water balance components and transformation of flow to the watershed outlet, it provides valuable insight into water movement processes within catchments. The streamflow measured upstream and downstream of a river segment reflects the transformation of inflow to the catchment outlet. The purpose of the study is to develop flow transferring characteristics of watersheds using the measured streamflow by United States Geological Survey (USGS) and to develop methodologies to apply the flow transferring function to ungauged basins. The methods of flow transfer from upstream to downstream points of a basin are analyzed using example watersheds in the Kentucky River Basin in Kentucky, USA.

### 1.2. Study Area

The Kentucky River Basin (KRB), with an area totaling 18,000 km<sup>2</sup> (Figure 1), originates in Lee County in Kentucky. The Kentucky River flows 417 km with an average discharge of 285 m<sup>3</sup>/sec and the average rainfall for the basin is over 1000 mm. The Kentucky River Basin supplies water to approximately one-sixth of the Commonwealth of Kentucky. The KRB is divided into five major subbasins: Lower, Upper, South Fork, North Fork and Middle Fork Kentucky. Each of these subbasins has varying landuse and topography, with Lower Kentucky



characterized by the intense karst area of limestone substrate with springs concentrated on 70% of the watershed. The Lower Kentucky watersheds are separated from Upper watersheds by the Knobs, which alter the hydrology significantly. The hilly landscape of Upper subbasin is characterized by rapid surface runoff and slow drainage. The North Fork, Middle Fork, and South Fork subbasins are located within a mountainous terrain with rapid surface runoff. The predominant landuse is deciduous forest in Upper and North fork, South fork and Middle fork basins and pasture in Lower Kentucky.

### 1.3.Scope of the study

Forty active USGS gauges (as of 2007) were monitored in KRB, which provides one gauge per 450 km<sup>2</sup> of a 18,000 km<sup>2</sup> basin. The ungauged portions of the basin will require hydrologic modeling if one were interested in hydrologic behavior. A hydrologic transfer function was developed in this study for flow prediction of such ungauged basins. The function from gauged basins was correlated to watershed and stream characteristics. The availability of watershed and stream characteristics for any basin, gauged or ungauged, was facilitated by the Geographic Information System (GIS). Regardless of the size of the basin, the proposed methodology can be adapted to facilitate flow prediction in ungauged basins. The successful implementation of the research was completed by making it available on the internet to acquire flow estimates for ungauged streams in the KRB.

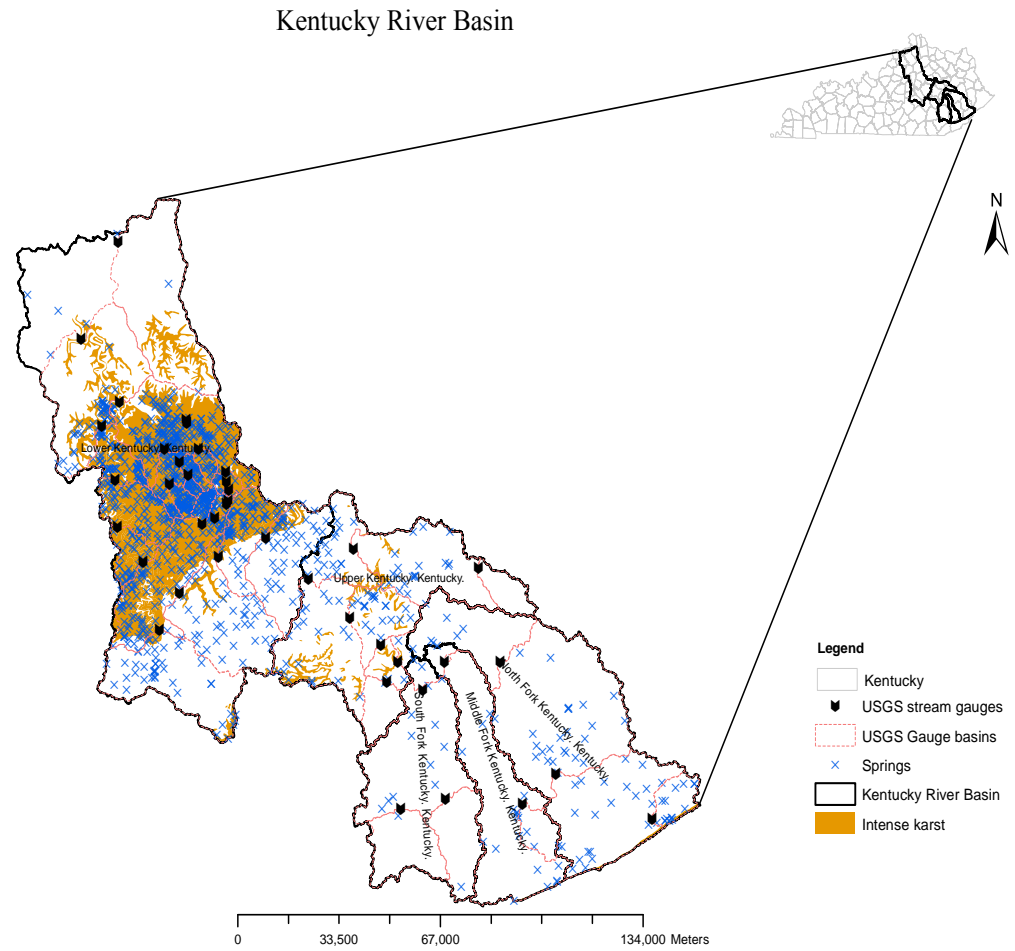


Figure 1-1. Kentucky River basin and its major karst features and surface water monitoring stations

#### 1.4. Structure of the Dissertation

The literature review section provides a summary of the past studies performed in the context of flow transfer functions, generalizing its functionality to ungauged basins, existing hydrologic models, and the calibration procedure for the purpose of selecting the best parameters.

Chapter 1 entitled “*Development of flow transfer characteristics from streamflow records*” discusses the overview of the developmental methodology to obtain a flow transfer function among USGS gauges of Kentucky River Basin. The chapter provides a comprehensive analysis carried out for identification of similar gauges by multivariate regression analysis, derivation of variables that contributed to the identification of the hydrologically similar drainage areas using GIS, the identification of similarity between frequency and time domain analysis in the context of transfer function derivation, derivation of average transfer function (ATF), and finally the application of developed methodology to example USGS-gauge pairs in the basin.

Chapter 2 entitled “*Flow routing using transfer function derived from streamflow records*” discusses the implementation of the ATF for flow routing of ungauged first-order stream networks in the KRB. The inflow was simulated using TOPMODEL and outflow at any first-order stream was determined using the ATF that was obtained in Chapter 1. The auto-calibration method was incorporated inside the TOPMODEL framework by adapting the Shuffling Complex Evolution-University of Arizona (SCE-UA) (Duan et al., 1992). This chapter also compares the effect of spatial resolution of rainfall data in hydrologic modeling by simulation with both raingauge measurements and NEXRAD rainfall estimates.

Chapter 3 entitled “*A GIS framework for publication of streamflow for the Kentucky River Basin*” discusses the GIS components used to publish streamflow values into the Internet using ArcIMS. The development of web pages, accessing USGS flow values on a daily temporal scale, the query structure for online retrieval of streamflow, and error statistics between the predicted and observed flow from USGS are also provided in this chapter.

Chapter 4 entitled “*General conclusions and observations of the results*” provides the overall assessment of the results of the transfer function derivation, using ATF for flow routing, effect of spatial resolution of rainfall in hydrologic modeling, effect of karst areas in implementing the proposed methodology, and structure of GIS bases queries and databases for real-time data acquisition.

The *appendix* section of the dissertation attaches the programs developed within ArcGIS suite for data acquiring and processing.

## 1.5. Literature Review

### 1.5.1. History of system transfer function identification

Measured streamflow values can help identify watershed properties that necessitate flow regulation within a catchment. These watershed properties that drive the flow transformation are related to inflow by the unit hydrograph, which is the transformation function of inflow into outflow. The Unit hydrograph introduced by Sherman (1932) operates on a linear input output relationship assuming uniform distribution of rainfall over a watershed area under consideration. The outflow is then determined by weighting the inflow with the unit hydrograph. Using this structure, given the inflow and outflow, the unit hydrograph can be derived by statistical and mathematical methods (Abraham, 1985; Duband et al., 1993; Lai, 1981).

For example, O'Donnell (1960) used Fourier transformation to obtain the unit hydrograph from observed runoff and rainfall excess. The unit hydrograph thus derived was convoluted with the rainfall excess to generate a hydrograph for Ash Brook catchment and it was found that the predicted runoff conformed well to observed runoff. The Fourier transformation was widely used for system characterization including catchment and aquifer (Long and Derickson, 1999; McGuire and McDonnell, 2006). However, Jain et al., (2005) found that the Fourier method provided negative transfer function coordinates and suggested using non-negative constraints in the transformation procedure. Long et al., (1999) overcame this disadvantage by logarithmically smoothing the transfer function and removing the negative coordinates.

Additionally, the transfer function can also be extracted using Laplace transformation and numerical procedures. Blank et al., (1971) evaluated the unit hydrograph using Laplace and Fourier transformations and concluded that both methods performed equally in terms of estimating runoff. Chapman (1996) developed a unit hydrograph using a numerical procedure in which the initial unit hydrograph was assumed and the measured runoff was used to obtain rainfall excess. A new unit hydrograph was derived using deconvolution of the computed rainfall excess and runoff coordinates. However, Laine (1970) mentioned that the unit hydrograph could be identified without the rainfall input assuming a linear response. In addition, Laine (1970) suggested that the magnitude of linearity of the catchment could also be found from their research. The unit hydrograph was derived by solving a set of polynomial equations obtained from the inflow and outflow series.

A method of discretization of the number of ordinates of the input and output series was used to obtain a unit hydrograph by Diskin and Boneh (1975) and the flow series was solved by

posing non-negative constraints to produce the positive unit hydrographs. Linear and non-linear programming was also used to extract flow-generating characteristics of a catchment from observed streamflow records (Mays and Taur, 1982; Deininger, 1969; Eagleson et al., 1966). Linear and non-linear programs help to obtain non-negative kernel functions that incorporate calculation of loss terms. Eagleson et al., (1966) proved that deconvolution of rainfall-runoff relationship could be represented by the least square approximation. Another study by Newman and Marsily (1976) shows that the parametric programming can impart knowledge about the shape of the response function provided the rainfall excess and observed runoff. These parametric programming allow criteria to be imposed during deconvolution of excess rainfall and runoff to make the unit hydrograph physically plausible.

More advanced unit hydrograph derivations have become available based on kinematic routing techniques. Solution to these techniques consists of hydraulic variables such as celerity and diffusivity of flood waves. Fernandez et al., (2006) coupled the hydrologic model - DRAINMOD, with the response function developed based on the Hayami kernel function, an analytical form of the diffusive equation (as cited by Moussa, 1996). Adequate surface flow prediction, with root mean square error (RMSE) of 0.8 mm for calibration period and a RMSE of 1.0 mm for validation period, was achieved from this coupled model. Chahinian et al., (2005) also used Hayami kernel function in which the kernel functions were obtained from measured hydrographs. Olivera and Maidment (1999) developed a grid based GIS model that used a flow path response function derived from an advection-dispersion equation. Flow responses from individual grids were convoluted to produce outflow at the watershed outlet. The surface runoff predicted from the model showed 4% volume difference with the observed runoff.

Although methods specified above worked well for synthesizing a hydrograph for ungauged catchments, application of these methods to karstic basins may introduce serious prediction errors. Karstic basins consist of highly varying water storage units, in which significant heterogeneities exist at different observation scales ranging from fine cracks to large holes and conduits. Therefore, application of linear and non-linear unit hydrograph transformations may not adequately represent the extreme dynamic flow process in karstic catchments. Labat et al., (2000) evaluated these linearity assumptions of unit hydrograph in karst basins. In their study, a statistical method was used that relates observed rainfall and runoff by auto- and cross-correlation functions. The unit hydrograph derived from this statistical method was compared against the unit hydrograph derived from ordinary least squares (OLS) method. The hydrograph simulated from the OLS derived unit hydrographs performed better than the statistically obtained unit hydrographs. Juki'c and Juki'c (2003) used a composite transfer function, a non-parametric

transformation of quick flow unit hydrograph and parametric transformation of subsurface flow unit hydrograph, for simulating hydrographs from karstic basins. The results from their study proved to be superior to non-parametric transformation in terms of accurately predicting low flow component of the spring hydrograph.

#### 1.5.2. History of regionalization techniques

Since flow-producing mechanisms must be known for hydrograph simulation of ungauged catchments, the properties extracted from measured streamflow have to be extrapolated. Their similarity to the ungauged catchments can be expressed as a weighting factor, where the weights represent the degree of similarity among watersheds (McIntyre et al., 2005). In a statistical sense, clustering of abstracted flow information from watersheds enables runoff prediction in ungauged watersheds. Catchments that show similar hydrogeologic characteristics produce hydrographs with, approximately, same shape and size. This similarity of catchments based on multivariate normal distributions was evaluated for catchments in Scotland by Acreman and Sinclair (1986). Acreman and Sinclair (1986) used basin characteristics such as drainage area, soil type, stream density, and rainfall data to delineate the basins. Results indicated that clusters based on watershed characteristics proved a useful measure in validating the flow behaviors. The clustering analysis was used to delineate homogeneous sections of streams in Tagus basin in Spain (Sanz and Jalon, 2005) based on 12 climatic and hydrologic parameters including flow properties such as high and low flow values and annual variations for wet and dry periods. Sanborn and Bledsoe (2006) also used cluster analysis to stratify the similar flow regimes. Principal component analysis was first used to filter correlated dependent variables. To decrease the uncertainty of model prediction, the cluster and principal component analysis was performed for low and high flows individually. The variables filtered out were applied to ungauged catchments based on a multiple regression model.

Hybrid cluster algorithms, which are the combination of hierarchical and partitional algorithms, operate on either agglomerative or divisive techniques. Starting with a single cluster, the agglomerative method combines the clusters together at each step of the process whereas the divisive method divides the single cluster into two at each step of the process (Lattin et al., 2003). Therefore, the hierarchical method is not influenced by the initialization procedures such as number of clusters and cluster centers. However, the partitional algorithms assigns points based on proximity to existing clusters and so is influenced by initialization of the clustering process. At each step of clustering, the clusters can move around the analysis space whereas the hierarchical procedure does not provide this functionality. Therefore, to use the advantages imparted by both methods, Rao and Srinivas (2006a) used hybrid of hierarchical and partitional algorithms to

define homogeneous watersheds. The clusters obtained by agglomerative hierarchical algorithms were refined with the help of partitional clustering and the study proved that the hybrid clustering is computationally effective in identifying catchments of similar hydrogeologic characteristics. In their study, the authors suggested increasing the number of variables that represent flood response characteristics of catchments could improve the performance of clustering algorithms. During any clustering procedure, the possibilities that stratified catchments might belong to more than one homogeneous region are large. In such cases, fuzzy cluster algorithms may prove useful to identify catchments in a homogeneous region sharing membership with other homogeneous regions. As suggested by its name, the crisp partition among the watershed clusters will be softened in order to share membership with other clusters (Rao and Srinivas, 2006b). The algorithms such as fuzzy c-means, Gustafson-Kessel, Gath-Geva (Hoppner et al., 1999) are available to be used in the fuzzy cluster analysis. The importance of quantity and quality of input was described by Ouarda et al., (2008). The authors compared four methods of regionalization methods: hierarchical, canonical cluster analysis (CCA), revised canonical cluster analysis and canonical kriging using flood quantile estimates. The results from the study showed that the hierarchical method provided better regional estimation and CCA provided better flood estimates with minimum relative mean square error.

In recent years, Artificial Neural Network (ANN) has become an attractive tool for regionalization techniques. Traditional regression analysis for regionalization was replaced by ANNs and they proved to be robust and effective in terms of accurate catchment parameter estimates (Heuvelmans et al., 2006). Performance of ANNs is increased by coupling them with traditional clustering algorithms such as CCA. Shu and Ouarda (2007) derived such a tool to take advantages of characteristics provided by both ANN and CCA. First CCA was used to delineate homogeneous catchment areas based on physiographic and climatic variables. Then the hydrologic behavior of the catchments within the clusters is identified with the help of ANN. The flow producing characteristics of the basins were then transferred to ungauged catchments based on the trained ANN. The authors found promising results from the coupled ANN and CCA models compared to the single ANN or original CCA models. Self Organizing Map (SOM) is an ANN method that is used as a regionalization technique. In a study to delineate homogeneous regions based on rainfall in Taiwan, SOM showed vastly improved performance over classic K-means and Ward's method (Lin and Chen, 2006). Even though ANN methods provide better estimation of a process, initialization of number of neurons and simulation of physical processes by black-box procedure pose uncertainties in the hydrologic modeling.

Efforts were made to combine SOM and fuzzy clustering by Srinivas et al., (2008). In ungauged watersheds, this combined method performed better than the traditional Canonical Correlation Analysis (CCA) and regression analysis. While clustering homogeneous watersheds, redundancy of streamflow gauges within a cluster could be removed (Burn and Goulter, 1991). The clustering procedure with hierarchical and partitional algorithms provides hard boundaries and thus does not allow distribution of membership among the clusters. Additionally, during the selection of regionalization method careful consideration should be made regarding the applicability of the procedure to regional and local estimation of the objective variables. For example, Ouarda et al., (2008) reported that compared to CCA clustering, the hierarchical method provided better estimation of flood quantile on a regional scale.

#### 1.5.3. History of hydrologic models and parameterization

The hydrologic models currently available range from field scale to basin scale and employ various methods for system transformation. The water balance components are estimated according to the inputs given to the modeling system. The overland flow is simulated based on either infiltration excess or saturation excess or subsurface return flow mechanism for a combination of soil and land use practices. The overland flow then becomes the channel flow along its length. The flow is transferred to the watershed outlet using instantaneous unit hydrograph, simple reservoir model, kinematic wave routing, or finite element routing models for system transformation depending on the structure of the hydrologic model. For example, Soil and Water Assessment Tool (SWAT), a semi distributed hydrologic model provides flexibility of using variable storage and Muskingum kinematic wave routing for transforming system inflow into outflow (Neitsch et al., 2005). The variable storage method works based on continuity equation and the Muskingum wave routing uses a combination of wedge and prism storages for modeling storage volume of flooding in a river (Chow et al., 1988). A field-scale DRAINMOD calculates hydrologic components based on the storage capacity of the soil profiles. The channel routing is performed using a kernel function that is based on the Hayami function. The Corp of Engineers' HEC-HMS model employs various routing models such as Muskingum-Kunge routing and kinematic routing (Feldman, 2000). Similar options are available in the spatially distributed TOPMODEL.

Fundamental concepts underlying TOPMODEL to generate runoff integrate the classic water balance components of runoff generating saturated areas and catchment surface storage (Kavetski et al., 2003). The principal component of TOPMODEL is the topographic index that delineates runoff generating saturated areas. The topographic index is a function of hydraulic gradient and soil transmissivity (Pan et al., 2004) and is obtained from Digital Elevation Models



(DEM) and soil maps. Therefore, the topographic index is considered to be proportional to the surface runoff volume. Subsurface flow is calculated as a function of storage deficit (Wolock et al., 1990). The impervious areas are modeled as a function of percentage of imperviousness and precipitation depth.

The efficiency of the model depends primarily on the parameterization that will reflect the processes of the hydrology in the watershed. In other words, the algorithms used in finding the best fitting hydrological parameters should also consider the naturally occurring flow processes. The search algorithms can be either local or global; the local search methods such as simplex method, Rosenbrocks method, Newton-Raphson method cannot validate the presence of multi-local optima (Duan et al., 1992) and they tend to be trapped in the cluster of local minima (Kuczera, 1997). Therefore, the global optimum search algorithms such as Genetic algorithm (GA), multistart simplex procedure, and Shuffled Complex Algorithm – University of Arizona (SCA-UA) were developed to eliminate the possibility of being caught up in the local minima. Additionally, the local search algorithms need intensive computer resources in terms of number of objective function (OF) evaluation and number of local optima.

The global search methods also show their own limitations by the computer resources required, size and shape of search space and number of OF evaluations needed. Franchini et al., (1998) showed that SCA-UA method proved its superiority by converging to a minimum number of OF evaluations within the given parameter space compared to the GA and Pattern Search methods. The multistart simplex procedure could produce a highly effective model calibration but will reduce the efficiency by requiring large number of OF evaluations for convergence. The SCA-UA needed only one-third of OF evaluations of the multistart simplex method in the study (Duan et al., 1992).

## 1.6. Summary of literature review

The literature review section explains past studies of flow transfer functions, methods to select watersheds that are similar hydrologically, the hydrologic model that satisfy the purpose of reduced number of variables and the calibration methods for models. To summarize, for the current study, a non-parametric method is adopted to obtain the transfer function. The physical description hidden in the inflow series will be represented by the data itself rather than imposing constraints to the transfer function model. To conclude the transfer function identification, the flow series from gauged catchments will be represented as Fourier series and the identification procedure will be performed in Fourier domain. Based on these observations of various clustering methods, the regression multivariate analysis is selected due to simplicity and strong statistical inferences that it can provide to analyze the significant variables.

Among the hydrologic models listed in the literature review, TOPMODEL was selected for simulation of overland flow because of its simplicity and fewer number of control parameters as compared to other models. From the evaluation of the local and global search methods, the SCA-UA method was chosen as an optimization procedure for this study. In the SCA, the parameter space is input with the lower and upper boundary around the true values. A sample space is generated from the population and developed into the user specified number of complexes and the OF is estimated for this sample space. However, to enable the communication in the sample parameter space, the complexes are shuffled and the corresponding OFs are calculated. The minimum OF is then found by searching in the improvement direction in which the convergence criterion is met.

## 2. Development of flow transferring characteristics from streamflow records

### 2.1. Abstract

This paper describes the extraction of the flow transferring characteristics of watersheds in the time domain using a novel approach. The daily streamflow data were processed to obtain the flow transfer function of Kentucky River Basin, Kentucky, USA. The stream gauges were ordered from upstream to downstream. A total of 194 such combinations of gauges were obtained from 40 active gauges monitored in the basin. The flow transfer function between gauge combinations were determined by dividing the outflow by the inflow and averaging over the analysis period. The average transfer function thus derived was related to the ratio of watershed and channel characteristics: drainage area, channel length, channel slope, mean annual flow, and mean annual minimum flow by multiple regression analysis. The average transfer function, determined from the regression analysis, was validated as a flow routing procedure for several gauge combinations located in Kentucky River Basin on event, seasonal and annual basis for multiple years. The proposed method was able to simulate the flow from basins of varying size; however, basins located in karst geology were overestimated. The coefficient of efficiency of the analyzed runoff events ranged up to 0.98. Therefore, it was concluded that the proposed methodology can be applied to flow prediction in ungauged basins using watershed and channel characteristics and using historic flow measurements.

Keywords: Historic measured streamflow, flow transfer, transfer function, channel and flow Characteristics, multiple regression analysis

## 2.2. Introduction

Flow estimation in ungauged basins is an important procedure in hydrologic and water quality modeling. Estimation of a physically plausible streamflow for ungauged catchments has been a challenge in the field of hydrology due to the lack of flow measurements for validation and the overwhelming number of variables and computational resources that are needed. The variability of flow increases in karst catchments for which the variables needed to define underground flow paths are hard to obtain. The lack of stream gauges in these basins leaves the modeler with limited information about the characteristics of flow processes. Therefore, the uncertainty of a hydrologic model increases with the increasing number of unknown parameters for catchments dominated by karst geology.

Studies concerning ungauged basins often discuss manipulating characteristics of gauged basins and using that information in hydrograph estimation. However, analyzing streamflow measured between the inlet and outlet of a watershed could prove useful for summarizing the processes occurring within a hydrologic boundary. Measured streamflow gathers and secures information about catchment attributes, water balance components, and transformation of these components to the watershed outlet. It has the advantage of providing valuable insight into water movement processes within catchments. Thus, the streamflow measured upstream and downstream of a river segment reflects the transformation of inflow to the catchment outlet. Therefore, comprehending the hydrograph should be a pivotal part of the hydrologic modeling and the water management practices of a catchment. Since the watersheds are physically connected with each other by the quantity of flow transferring between them. The watershed characteristics can be related with a flow transferring function that can be extrapolated from gauged basins to ungauged basins.

The flow transfer function in any hydrograph determination method is the function that relates the inflow and outflow. The impulse response function and pulse response function are examples of transfer function (TF) in linear system of watersheds. In a continuous time domain, the unit impulse response of the linear time invariant system can be obtained by the convolution of the impulse response function and inflow to the watershed as,

$$Q_{out}(t) = \int_0^t Q_{in}(\tau)u(t - \tau)d\tau \quad (1)$$

Where,

$Q_{out}(t)$  is outflow at time  $t$ ,  $L^3/T$

$Q_{in}(\tau)$  is inflow at time  $\tau$ ,  $L^3/T$

$u(t - \tau)$  is unit hydrograph / transfer function,  $1/T$

The convolution of the transfer function with any inflow will produce outflow at the basin outlet. Hydrologic models in current practice employ transfer functions in various forms to obtain flow at the watershed outlet given inflow. The routing / transferring function can be provided by either one of these methods: kinematic wave routing, instantaneous unit hydrograph, reservoir model, and finite element routing. Channel characteristics and hydraulics of wave transformation are the defining variables of routing. Hence, the strategy of this research is to relate the inflow and outflow to and from a channel segment with the aid of channel characteristics in such a manner that the wave transformation will be determined.

For this purpose, with historic flow measurements upstream and downstream of a watershed, the transfer function can be determined by deconvolution. The deconvolution methods are usually computationally intensive. There are three different methods involved in the identification of system transformation function: 1) non-parametric methods in frequency domain, 2) parametric methods in time domain and 3) analytic procedures. Non-parametric methods use fewer assumptions about the distribution of the data at hand than the parametric methods. In addition, the non-parametric methods can prove to be useful for their robustness and simplicity.

### 2.3. Literature Review

#### 2.3.1. Non-Parametric methods in the Frequency domain

O'Donnell (1960) used Fourier transformation to obtain the unit hydrograph from observed runoff and rainfall excess. The unit hydrograph was convoluted with the rainfall excess to generate a hydrograph for the Ash Brook catchment, and it was found that the predicted runoff conformed well to the observed runoff. The Fourier transformation was widely used for system characterization including catchments and aquifers (Long and Derickson, 1999); McGuire and McDonnell, 2006). Long and Derickson, (1999) reported that an oscillating transfer function obtained from Fourier analysis might not provide a physically valid representation of the system being analyzed. Therefore, the authors applied a filter to the dataset before bringing it into the frequency domain. The accuracy of the transfer function derived in the Fourier domain was found to be dependent on the number of data points used in the study. Jain et al., (2005) found that the Fourier method provides negative transfer function coordinates and suggested using non-negative constraints in the transformation procedure. Long et al., (1999) overcame this disadvantage by logarithmically smoothing the transfer function and deleting the negative coordinates.

Additionally, the transfer function can be extracted using a Laplace transformation and numerical procedures. Blank et al., (1971) evaluated the unit hydrograph using Laplace and

Fourier transformations and concluded that both methods performed equally in terms of estimating runoff. Chapman (1996) identified a unit hydrograph using a numerical procedure in which the initial unit hydrograph was assumed and the measured runoff was used to obtain rainfall excess. A new unit hydrograph was derived using deconvolution of the computed rainfall excess and runoff coordinates. However, Laine (1970) mentioned that the unit hydrograph could be identified without the rainfall input assuming a linear response. In addition, Laine (1970) suggested that the magnitude of linearity of the catchment could also be found. The unit hydrograph was derived by solving a set of polynomial equations obtained from the inflow and outflow series.

### 2.3.2. Parametric methods in the Time domain

A method of discretization of the number of ordinates of the input and output series was used to obtain a unit hydrograph by Diskin and Boneh (1975) and the flow series was solved by posing non-negative constraints to produce the positive unit hydrographs. Linear and non-linear programming was used to extract flow-generating characteristics of a catchment from observed streamflow records (Mays and Taur, 1982; Deininger, 1969; Eagleson et al., 1966). Linear and non-linear programs help to obtain non-negative kernel functions that incorporate calculation of loss terms. Eagleson et al., (1966) proved that deconvolution of the rainfall-runoff relationship could be represented by the least square approximation. Another study by Newman and Marsily (1976) showed that parametric programming can provide knowledge about the shape of the response function from the rainfall excess and observed runoff. These parametric programming methods facilitate the use of various criteria during deconvolution of excess rainfall and runoff to make the unit hydrograph physically plausible.

### 2.3.3. Analytical procedures

More advanced unit hydrograph derivations have become available based on kinematic routing techniques. Solution to these techniques consists of hydraulic variables such as celerity and diffusivity of flood waves. Fernandez et al., (2006) coupled the hydrologic model - DRAINMOD, with the response function developed based on the Hayami kernel function. Adequate surface flow prediction, with root mean square error (RMSE) of 0.8 mm for calibration period and a RMSE of 1.0 mm for validation period was achieved from this coupled model. Chahinian et al., (2005) also used the Hayami kernel function in which the kernel functions were obtained from measured hydrographs. Olivera and Maidment (1999) developed a grid based GIS model that uses flow path response function derived from an advection-dispersion equation. Flow responses from individual grids were convoluted to produce outflow at the watershed outlet. The

surface runoff predicted from the model showed 4% volume difference compared to observed runoff.

#### 2.3.4. Application of transfer function in flow estimation of ungauged catchments

The flow producing mechanisms of ungauged catchments are necessary for hydrologic modeling of such watersheds. Hence, if the transfer function is extrapolated to ungauged watersheds that are similar to the watersheds from which they are derived, the outflow can be determined by convolution. The extrapolation or the regionalization can be performed using techniques such as principal component analysis, cluster analysis, factorial analysis, least square approximations, and artificial neural networks.

Properties from measured streamflow have to be extrapolated and their similarity to the ungauged catchments can be expressed as a weighting factor, where the weights represent the degree of similarity among watersheds (McIntyre et al., 2005). In a statistical sense, clustering of abstracted flow information from watersheds enables runoff prediction in ungauged watersheds. Catchments that show similar hydrogeologic characteristics produce hydrographs with, approximately, the same shape and size. This similarity of catchments based on multivariate normal distributions was evaluated for catchments in Scotland by Acreman and Sinclair (1986). Acreman and Sinclair (1986) used basin characteristics such as catchment area, soil type, stream frequency and rainfall data to delineate the basins. Results indicated that clusters based on watershed characteristics proved a useful measure in validating the flow behaviors. The clustering analysis was used to delineate homogeneous sections of streams in Tagus basin in Spain (Sanz and Jalon, 2005) based on 12 climatic and hydrologic parameters including flow properties such as high and low flow values and annual variations for wet and dry periods. Sanborn and Bledsoe (2006) also used cluster analysis to stratify the similar flow regimes. Principal component analysis was first used to filter correlated dependent variables. In order to decrease the uncertainty of model prediction, the cluster and principal component analysis was performed for low and high flows individually. The variables filtered out were applied to ungauged catchments based on a multiple regression model.

Hybrid cluster algorithms, which are the combination of hierarchical and partitional algorithms operates on either agglomerative or divisive techniques. Starting with a single cluster, the agglomerative method combines the clusters together at each step of the process whereas the divisive method divides the single cluster into two at each step of the process (Lattin et al., 2003). Therefore, the hierarchical method is not influenced by the initialization procedures such as number of clusters and cluster centers. However, the partitional algorithms assigns points based on proximity to existing clusters and so is influenced by initialization of the clustering process. At

each step of clustering, the clusters can move around the analysis space whereas the hierarchical procedure does not provide this functionality. Therefore, to avail the advantages imparted by both methods, Rao and Srinivas (2006a) used a hybrid of hierarchical and partitional algorithms to define homogeneous watersheds. The clusters obtained by agglomerative hierarchical algorithms were refined with the help of partitional clustering and the study proved that the hybrid clustering is computationally effective in identifying catchments of similar hydrogeologic characteristics. In their study, increasing the number of variables that represent flood response characteristics of catchments could improve the performance of clustering algorithms. During any clustering procedure, the possibilities that stratified catchments might belong to more than one homogeneous region are large. In such cases, fuzzy cluster algorithms may prove useful to identify catchments in a homogeneous region sharing membership with other homogeneous regions. As suggested by its name, the crisp partition among the watershed clusters will be softened in order to share membership with other clusters (Rao and Srinivas, 2006b). The algorithms such as fuzzy c-means, Gustafson-Kessel, Gath-Geva (Hoppner et al., 1999) are available to be used in the fuzzy cluster analysis. Ouarda et al., (2008) compared four methods of regionalization methods: hierarchical, canonical cluster analysis (CCA), revised canonical cluster analysis and canonical kriging using flood quantile estimates. The results from the study showed that the hierarchical method provided better regional estimation and CCA provided better flood estimates with minimum relative mean square error.

In recent years, Artificial Neural Network (ANN) has become an attractive tool for regionalization techniques. Traditional regression analysis for regionalization was replaced by ANNs and they proved to be robust and effective in terms of accurate catchment parameter estimates (Heuvelmans et al., 2006). The performance of ANNs is increased by coupling them with traditional clustering algorithms such as CCA. Shu and Ouarda (2007) derived such a tool to take advantages of characteristics provided by both ANN and CCA. First CCA was used to delineate homogeneous catchment areas based on physiographic and climatic variables; then the hydrologic behavior of the catchments within the clusters are identified with the help of ANN. The flow producing characteristics of the basins were then transferred to ungauged catchments based on the trained ANN. The authors found promising results from the coupled ANN and CCA models compared to the single ANN or original CCA models. Self Organizing Map (SOM) is another ANN method that is also used as a regionalization technique. In a study to delineate homogeneous regions based on rainfall in Taiwan, SOM showed vastly improved performance over classic K-means and Ward's method (Lin and Chen, 2006). Even though ANN methods



provide better estimation of a process, initialization of the number of neurons and simulation of physical processes by black-box procedures pose uncertainties in the hydrologic modeling.

Efforts were made to combine SOM and fuzzy clustering by Srinivas et al., (2008). In ungauged watersheds, this combined method performed better than the traditional CCA and regression analysis. While clustering homogeneous watersheds, redundancy of streamflow gauges within a cluster could be removed in order to rationalize the type of users of streamflow data and nature of problem that needs to be addressed by a particular flow gauge (Burn and Goulter, 1991). The clustering procedure with hierarchical and partitional algorithms provides hard boundaries and thus does not allow distribution of membership among the clusters. Additionally, during the selection of regionalization method careful consideration should be made regarding the applicability of the procedure to regional and local estimation of the objective variables.

Based on the study of previous research described above, the frequency domain analysis and multiple regression analysis were selected. The above mentioned studies showed that given rainfall and outflow or input streamflow and output streamflow, the unit hydrograph can be derived. In this study, effort was made to identify a transfer function by dividing inflow and outflow of each time step (T). The transfer function was then averaged over the entire period of analysis, aided with the assumption of a time invariant linear hydrologic system. The time invariant is very suitably expressed as - "if the initial state and the input are the same, no matter at what time they are applied, the output waveform will always be the same". However, the output will be shifted from the initial time  $\{t_0\}$  to  $\{t_0 + T\}$  (Chen, 1999). This study aimed at deriving transfer function that averages over any number of periods given the outflow and inflow to and from a watershed.

The streamflow regimes with similar watershed characteristics produce similar hydrographs. However, transferring flow from one basin to another basin differs widely across a hydrologic boundary. Although having hydrologically similar basins help relate model parameters and watershed characteristics, the flow transferred from one set of hydrologically similar basins to another set of hydrologically similar basins require a transfer function that relates these basins. Therefore, the transfer function needs to be related to channel characteristics such as channel slope, length and drainage area from which it receives inflow. This relationship will guide the flow prediction procedure in ungauged basin if flow into the channel network is known. In addition, if outflow from one basin is known, the transfer function derived from the relationship can determine the inflow to the upstream basin.

Therefore, the objectives of the research can be described as follows:

- To derive a methodology for flow prediction in ungauged basins by
  - Determining a transfer function from observed streamflow records
  - Deriving a relationship between the transfer function and watershed characteristics using multiple regression analysis

## 2.4. Materials and Methods

### 2.4.1. Study area

The Kentucky River basin (KRB) with an area totaling 18,000 km<sup>2</sup>, originates in Lee County in Kentucky (Figure 2-1). The Kentucky River extends a length of 417 km with an average discharge of 285 m<sup>3</sup>/sec and the basin has an average annual rainfall of over 1000 mm. The river supplies water to approximately one-sixth of the state of Kentucky.

The KRB is divided into five major subbasins: Lower, Upper, South fork, North fork and Middle fork Kentucky. Each of these subbasins has varying landuse and topography, with Lower Kentucky characterized by the intense karst area of limestone substrate with springs concentrated around 70% of the watershed. The Lower Kentucky watersheds are separated from Upper subbasins by Knobs, which alters the hydrology significantly. The hilly landscape of Upper subbasin is characterized by rapid surface runoff and slow drainage. The North fork, Middle fork and South fork subbasins are located within a mountainous terrain with rapid surface runoff. The predominant land cover in these subbasins is deciduous forest in Upper and North fork, Middle fork and South fork basins and pasture in Lower Kentucky.

### 2.4.2. Derivation of transfer function

At steady state, the ratio of outflow and inflow can be defined as the transfer function for day ( $t$ ) in the frequency domain.

$$TF = u(t - \tau) = \frac{Q_{out}(t)}{Q_{in}(\tau)} \quad (2)$$

The watershed is assumed, at this point, to represent a first-order linear system. When a unit impulse is applied to the steady state system, the response can be determined as transfer function times the impulse response.

$$Q_{out} = TF \times Q_{in} \quad (3)$$

However, during estimation of transfer function using Fourier analysis, the data depicted a coincidence between frequency and time domain analysis. In the time domain, equation (2) will be a complex numerical procedure as provided in literature review. However, it is physically plausible to describe the transfer function in both time and frequency domain as “the ratio with which the inflow is transferred to outflow for basins at time T”. Therefore, instead of a Fourier analysis, in this study a simple ratio of outflow and inflow was used as the flow transfer function. The ratio can be averaged over any number of days in a strictly time invariant system. The average transfer function then represents for a given average amount of inflow, the average amount of outflow generated from the system. The theory was validated by generating outflow from an example pair of gauges located in the study area. First, the time-invariant characteristic of the hydrograph was analyzed. Being time-invariant, the inflow from time “T” will produce outflow at the same time proportional to the channel and flow characteristics of the basin.

When the transfer function is averaged, it provides a common response function that can be used in equation (3) as,

$$Q_{out} = ATF \times Q_{in} \quad (4)$$

The example basin, Red River near Hazel Green, used for validation of the method drains from an area of 170 km<sup>2</sup> into an area of 763 km<sup>2</sup> (Table 2-1). The daily measured flows were downloaded from USGS for year 2001, and the data were divided into three seasons: spring, summer and fall. For each season, the transfer function was calculated by equation (2) and the average of TF (ATF) was obtained over each season. Multiplication of inflow to the watershed with the ATF was used to calculate quantity of outflow. For validation, the ATF of each season was applied to estimate outflow of the other two seasons thus producing nine hydrographs for the year 2001 (Figure 2-2). The subbasin area drained by the USGS stream gauges were delineated using the ArcGIS software package (Figure 2-3). To understand the flow transfer characteristics of the whole basin, the stream gauges were arranged in upstream and downstream order such that 194 combinations of gauges were obtained. The resulting gauges provided upstream and downstream watersheds from the hydraulically remotest point of the KRB to its outlet. Channel and flow descriptors (CFD) were obtained between each of the 194 upstream and downstream gauges. The descriptors derived were:

- Area
- Channel slope
- Channel length

- Drainage density
- Annual mean flow
- Annual 7-day minimum flow
- Annual peak flow

Since the flow transfer actually happens in the channel, the upland parameters such as soil, landuse and climatic variables were not included in the regression analysis. To be in accordance with the ATF, the CFDs were determined as the ratio of upstream and downstream basins. Since the study area consists of karst features, application of transfer functions to karstic basins may introduce serious prediction errors as karst featured watersheds are complex water transport systems. Labat et al., (2000) evaluated the linearity assumptions of unit hydrographs in karst basins. In their study, a statistical method that relates observed rainfall and runoff by auto- and cross-correlation functions. The unit hydrograph derived from this statistical method was compared against the unit hydrograph derived from ordinary least squares (OLS) method. The hydrograph simulated from the OLS derived unit hydrographs performed better than the statistically obtained unit hydrographs. Juki'c and Juki'c (2003) used a composite transfer function, a non-parametric transformation of a quick flow unit hydrograph and a parametric transformation of the subsurface flow unit hydrograph, for simulating hydrographs from karstic basins. The results from their study proved to be superior to non-parametric transformation in terms of accurately predicting low flow component of the spring hydrograph. In the current research, a parametric regression method was used to relate CFD and transfer function in order to facilitate a simple yet effective methodology to reduce computational time.

#### 2.4.3. Multiple regression analysis

Multiple regression analysis between ATF and CFD was performed using the statistical package Statistical Analysis Software (SAS). The ATF was derived for each combination of gauge pair. At this point, it is important to note that not all gauge pairs would have the same number of days of recorded flow data. Therefore, in order to maintain the system assumption of time invariant, the ATF was calculated for the number of periods of data available for the gauge pairs. Fortunately, for most gauge pairs, at least three years of daily stream flow records were available.

The 194 gauge pairs were regressed with watershed drainage area, channel and flow characteristics. The Goodness of fit of the model was analyzed with the help of R-Square, mean square error and the parameter estimates were analyzed using t-statistics and its probability. Multicollinearity in the regressors was analyzed using Variation Inflation Factor (VIF) and variables showing a  $VIF \geq 10$  were assumed to have dependence among other variables and

therefore removed from the analysis. A parsimonious model was selected without sacrificing the accuracy using a Shapiro-Wilk test of normality. Outliers in the selected parsimonious model were detected and removed by referring to Hat Diag H, Difference in FITS (DFFITS) and Differences in Beta (DFBETAS). The selected outliers were removed and the model was checked for validation of assumptions using normal probability plots and fitness of the model was checked with R-square, F-statistics, t-statistics and their probabilities. Additionally, three different variable selection methods were used to identify the best regression model: Backward elimination, Stepwise procedure, and Maximum R-Square method. The results from these methods were used only as guidance to select the best variables. The best model was selected manually by considering the t-statistics, residual plots, and normality plots. The ATF obtained from the best regression model was applied to the example pairs of gauges (Table 2-1) to validate the proposed method.

The regression model with 194 gauge pairs provided a studentized residual plot with numerous outliers that affected the model in terms of mean square error, violation of normality assumption. The gauge pairs that were identified as outliers were removed carefully based on Hat Diag, DFFITS and DFBETAS statistics. It needs to be remembered that “numerous” gauge pairs in this study represent flow from a single gauge to several downstream basins. Therefore, the analysis did not suffer from removal of many numbers of basins as outliers; rather it was the removal of one upstream gauge that affected the modeling process downstream.

The number of gauge pairs that remained after outlier removal was 183; the studentized residual, residuals standardized with the standard deviation of the sample data, plot from the regression model of these 183 pairs (Figure 2-4) suggested that the Kentucky River Basin can be identified as two different hydrological regimes based on the values of the transfer function. Removing outliers, the observations with high residuals, from the model also suggested that the study basin should be treated as different entities. Therefore, the gauge pairs were identified as Upper (identified in this paper as Regression1) and Lower Kentucky watersheds (identified in this paper as Regression2). The Lower and Upper Kentucky watersheds are identified in Figure 2-3.

The Regression2 watersheds are located in the Outer and Inner Bluegrass regions. The Inner Bluegrass Region is dominated by numerous shallow sinkholes and very low relief with thin soils. The Outer Bluegrass Region, on the other hand, has a low to moderate relief with thick soils (McDowell, 1986). The geology and topography changes the path through which a stream flows thus causing the basins drained by this area different than the basins upstream. Additionally, the Regression2 watersheds can be seen to flow through various ecological regions, which are characterized by relief, geology, and landuse. For example, the basins that flow across

the Ohio/Kentucky Carboniferous Plateau show different flow transfer function than the other basins that share the same hydrologic divide. The Regression1 watersheds are located in the Dissected Appalachian Plateau, Northern forested plateau escarpment and Ohio/Kentucky carboniferous plateau. Basins located in a Dissected Appalachian Plateau shows varying transfer function than the other basins located in the same plateau. The analysis of characteristics of these basins showed that even though they share a boundary with the hydrologically similar watersheds within a same ecological region, they differ in transformation of flow based on the amount of precipitation that fell over the analysis period and terrain with rapid surface runoff flows to a level terrain of knobs with medium surface runoff.

## 2.5. Results and Discussion

### 2.5.1. Analysis of Regression1 watersheds

The studentized residuals of the upper Kentucky (Figure 2-5) shows no violation of assumptions of regression analysis that the errors are normally distributed and are statistically independent. Therefore, the parameters ratio of watershed area was selected as catchment descriptors that best describe the transfer function.

$$ATF = 1.06 * \left( \frac{\text{Downstream area}}{\text{Upstream area}} \right) \quad (5)$$

Where,

Downstream area is the watershed area drained by the downstream watershed

Upstream area is the watershed area drained by the upstream watershed

The minimum root mean square (0.078) value along with large coefficient of determination value (0.997) showed that the selected regression model can be used to explain the transfer function of the basins using the area of the upstream and downstream watersheds.

### 2.5.2. Analysis of Regression2 watersheds

Figure 2-6 shows the validity of the regression model selected for the Lower Kentucky watersheds. From the Studentized residual plot, the flow is transferred to outlet in direct proportion to the ratio of upstream and downstream area. Therefore, the equation to estimate ATF using drainage area ratio is given by,

$$ATF = 1.15 * \left( \frac{\text{Downstream area}}{\text{Upstream area}} \right) \quad (6)$$

Where,

Downstream area is the watershed area drained by the downstream watershed

Upstream area is the watershed area drained by the upstream watershed

The R-square value of the selected model was 0.989 with the root mean square error value of 0.500. In an effort to reduce the root mean square, the gauge pairs were analyzed with other explanatory variables. However, the root mean square error values increased significantly and the normality assumption of the model was violated. Therefore, equation (6) was selected to represent the Lower Kentucky watersheds based on the Goodness of fit, error statistics analyses.

The coefficients 1.06 and 1.15 increase the ratio of drainage areas thereby increasing the transfer function. Therefore, the significance of the coefficients 1.06 and 1.15 indicates that the inflow from an upstream basin increases proportional to its drainage area as it travels downstream by 6% and 15% for the study subbasins, respectively.

### 2.5.3. Validation of the proposed model

The South Elkhorn creek, North Elkhorn creek, Town Branch Creek and KY River at Lock 2 were selected for further analysis of the presented method for different drainage area ratio from Regression2. Similarly, Regression1 equation was also analyzed with Kentucky (KY) River at Locks 6, 7, 8 and 13. The draining areas for these creeks and analysis year can be seen in Table 2-1. Out of five gauge pairs selected, one gauge pair was analyzed seasonally, one pair was analyzed yearly, one pair of gauges was analyzed for inflow determination, and two pairs were analyzed on event basis to determine the ability of the proposed model to simulate peak flow and baseflow. The event based analysis includes a volumetric bias term and coefficient of efficiency to validate the performance of the proposed method, whereas, the seasonal and yearly analysis includes a volumetric bias to quantify the runoff volume. Each of example basins was discussed separately as below:

The Coefficient of Efficiency (COE) is defined as,

$$COE = 1 - \frac{(Q_{sim} - Q_{obs})^2}{(Q_{obs} - Q_{Avg_{obs}})^2} \quad (7)$$

The volumetric bias in this study was defined as,

$$Volumetric\ bias = \left( \frac{Q_{sim} - Q_{obs}}{Q_{obs}} \right) * 100 \quad (8)$$

Where,

$Q_{sim}$ - Simulated flow,  $m^3/sec$

$Q_{obs}$ - Observed flow,  $m^3/sec$

#### 2.5.3.1. Regression1 analysis

KY River at Lock 13 - Event based:

The inflow hydrograph used in this event analysis was 03282000, Kentucky River at Lock 14 and the outflow hydrograph to be analyzed against the estimated one was obtained from USGS gauge 03282060, Kentucky River at Lock 13. The watershed produced hydrograph as a direct proportion to the inflow from upstream (Figure 2-7) since the areas drained is in direct proportion with each other with a ratio of 1.05. Therefore, the model is simply a direct transformation of inflow into outflow with little attenuation. The volumetric basis of -0.9% shows overprediction of the model is negligible; the coefficient of efficiency for this event is 0.98 which shows the accuracy of the proposed method to model hydrograph given an inflow event.

KY River at Lock 6 - Event based

The inflow hydrograph used in this event analysis was 03284000, Kentucky River at Lock 10 and the outflow hydrograph to be analyzed against the estimated one was obtained from USGS gauge 03287000, Kentucky River at Lock 6. The hydrograph for this river branch shows that the volume of the hydrograph (Figure 2-8) is reproduced with 14% underprediction and with a coefficient of efficiency of 0.92. The underprediction is mainly due to the reduced baseflow values and rising hydrograph values in the simulated hydrographs. The discrepancy might have been introduced by the reduced average transfer function estimated by the proposed method, which is expected considering the number of days used in determining the transfer function for regression analysis and number of days used for validation purpose.

Red River near Hazel Green - Seasonal analysis:

The inflow hydrograph was obtained from streamgauge 03282500, Red River near Hazel Green and the outflow hydrograph to be analyzed against the estimated one was obtained from USGS gauge 03282060, Red River at Clay city. Figure 2-9 shows the seasonal hydrographs of Red River near Hazel Green; baseflow during the fall is underpredicted whereas baseflow during late winter is overpredicted (Figures 2- 9c and 2-9a). During this period, the streamflow is dominated by baseflow component of hydrologic cycle as the winter starts and increases during the spring. However, as the summer season approaches, the model was able to transfer an ample amount of inflow to the downstream basin. This particular basin is headwaters of Red River with topography ranging from mountainous topography with rapid surface runoff to plateau area as we proceed to the downstream region, where ground water drainage dominates. Since the proposed



method uses the inflow available from upstream and size of drainage areas, the baseflow from lower basin was damped in the averaging process.

The disagreement of simulated base flow with observed flow suggests that the average response function represents the characteristics of watershed better during rainfall events where intensity of precipitation is high causing the watersheds to have a uniform distribution across its geographical area. The percent of areas producing surface runoff is also uniform during this period, whereas when the rainfall is sparse and is of high intensity and short duration, the method suffers from capturing the non-uniformity of runoff producing rainfall quantity and runoff producing saturated areas. The process can also be validated from Figure 2-9a where we move towards summer. The effect of non-uniformity is further analyzed with other basins of KRB. KY River at Lock 7 - Yearly analysis:

Following the analysis of seasonal discharge, the annual hydrographs are simulated for the effect of averaged flow transfer along the channel length. The inflow hydrograph was obtained from streamgauge 03286500, Kentucky River at Lock 7 and the outflow hydrograph to be analyzed against the estimated one was obtained from USGS gauge 03287000, Kentucky River at Lock 6. Figure 2-10, the discharge from KY River at Lock 7 shows a constant underprediction of baseflow throughout the year, leading to an overall underprediction percent of 25%, whereas the peak discharge is estimated with good agreement. The discrepancy in low flows can be attributed to the reduction in transfer function and the baseflow is dampened by the averaged flow transfer. Since the observed flow includes storage in the channel from previous days whereas the proposed method averages the storage, input from overland flow and water losses, the discrepancy is expected.

KY River at Lock 8 - Inflow determination:

To exhibit the full functionality of the proposed method, the inflow was determined by dividing outflow by the average transfer function (Figure 2-11). The outflow hydrograph was obtained from streamgauge 03284500, Kentucky River at Lock 8 and the inflow hydrograph to be analyzed against the estimated one was obtained from USGS gauge 03282290, Kentucky River at Lock 11. The hydrograph was predicted with good accuracy with only 5% underprediction in baseflow during low or no rainfall periods. During no or low rainfall period, the major contributor in a channel segment is the volume of water stored from previous day and baseflow. The averaging property of the estimated ATF plays a vital role in flow reduction during no rainfall days. This limitation can be overcome by including baseflow and storage estimation in inflow estimation, which will be depicted in Chapter 3 of this dissertation.

### 2.5.3.2. Regression2 analysis

Town Branch Creek and South Elkhorn Creek - Event based:

The inflow hydrograph for this event was obtained from streamgauge 03289193, Wolf Run at Old Frankfort Pike and the outflow hydrograph to be analyzed against the estimated one was obtained from USGS gauge 03289200, Town Branch at Yarnallton road. The discharge plot of Town Branch Creek (Figure 2-12) shows peak flow is predicted accurately whereas the baseflow hydrograph is underpredicted making the volumetric bias 17%. The low flows during low intensity, short duration rainfall events agree with the effect of event specific response function compared to the average transfer function. Regardless of the underprediction in baseflow, the model performed well with a coefficient of efficiency of 0.89. The inflow hydrograph was obtained from streamgauge 03289193 Wolf Run at Old Frankfort Pike and the outflow hydrograph to be analyzed against the estimated one was obtained from USGS gauge 03289300, South Elkhorn Creek. Figure 2-13 of South Elkhorn creek follows the effect of using average response function in outflow estimation instead of event specific response function during low intensity short duration rainfall period. The coefficient of efficiency of the hydrograph is estimated to be 0.57 with an overprediction of 9%.

As depicted by the Regression1 hydrographs, the recession limb is underestimated in Town Branch and South Elkhorn Creeks. The Inner Bluegrass region has karst features that form limestone aquifers. The ground water drainage flows out through the karst features and thus a significant portion of streamflow is contributed by baseflow. The underprediction can be alleviated by including storage and baseflow in the stream inflow.

Elkhorn Creek - Seasonal analysis:

The inflow hydrograph was obtained from streamgauge 03289300, South Elkhorn Creek and the outflow hydrograph to be analyzed against the estimated one was obtained from USGS gauge 03289500, Elkhorn Creek. The hydrographs for Elkhorn creek (Figure 2-14) show constant over prediction of discharge. The summer season (Figure 2-14b) overpredicts stream discharge. Elkhorn creek is in the karst dominated Bluegrass area and accurate quantification of surface hydrology in karstic basins has always been a difficult task. The average response function for flow estimation introduces significant errors. Due to existence of springs and underground channels in Elkhorn Creek region, the discharge from upstream to downstream might not always follow the same path. When the rainfall is scarce across a watershed, the flow determination becomes complex, not to mention the importance of subsurface characteristics during low rainfall events. Therefore, the percent contribution of inflow should be considered in the proposed model.

In contrast to the summer event in Figure 2-14b, the spring and fall seasons (Figures 2-14a and 2-14c) over predict flow to a lesser degree than that of summer season, with an overprediction of 51% and 55%, respectively. The peaks of simulated flow are in good agreement with observed flow whereas the baseflow is overpredicted. The overprediction of baseflow when there are not adequate rainfall events suggests that the average response of the basin to outflow might have been overpredicted and the partitioning of precipitation that contributes to direct runoff on karst surfaces should be accounted for. If one were to analyze the recession and rising limb separately for karst basins, the transfer function would be able to provide insights for flow prediction in karstic ungauged basins.

KY River at Lock 9 - Annual analysis:

The inflow hydrograph was obtained from streamgauge 03284230, Kentucky River at Lock 9 and the outflow hydrograph to be analyzed against the estimated one was obtained from USGS gauge 03290500, Kentucky River at Lock 2. The inflow upstream gauge is located in intense karst area of KRB; the inflow and outflow hydrographs simulated using average transfer function is given Figure 2-16. The overall prediction is in accordance with observed hydrograph. However, few peak discharges during spring and early fall are underpredicted making the volumetric bias to be 5%.

North Elkhorn Creek - Inflow determination:

The outflow hydrograph was obtained from streamgauge 03287580, Elkhorn Creek near Frankfort and the inflow hydrograph to be analyzed against the estimated one was obtained from USGS gauge 03289500, North Elkhorn Creek. The inflow hydrograph flows from moderate karst area to intense karst area. The simulated discharge (Figure 2-15) shows overestimation of peak flows and a good estimation of baseflow hydrographs. The overall prediction of the method agrees well with the observed hydrograph.

## 2.6. Conclusions

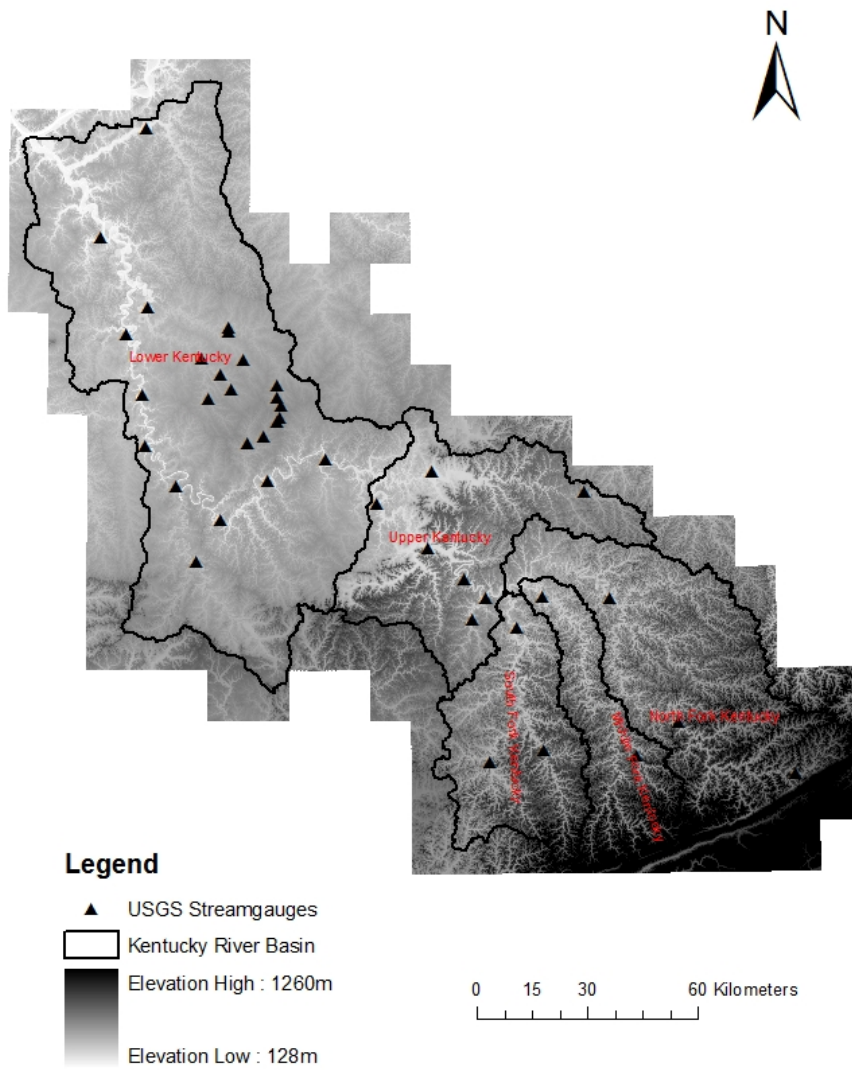
Flow prediction in ungauged basins is a challenging task for hydrologists. This paper presents a novel approach for flow prediction using simple watershed and flow characteristics such as drainage area. The Kentucky River Basin (KRB) was selected as a research area for this purpose. KRB is monitored by 40 active USGS stream flow gauges and areas drained by the gauges were delineated. The drainage areas were then combined into 194 gauge pairs and for each gauge pair, watershed and flow characteristics were obtained using GIS. The transfer function of each of these gauge pairs was estimated as a ratio of outflow and inflow, which were obtained from daily flow USGS database. The average of the transfer function (ATF) was estimated for the available duration of dataset assuming the watersheds are linear time invariant

systems. The ATF was then regressed against watershed and flow characteristics using multiple regression analysis. The studentized residual plots, root mean square and t-statistics were used to obtain the best regression model. The regression analysis showed that the gauge pairs can be divided into two different dataset as they represented two different hydrological regimes. The ATF was determined for two groups separately and used in estimation of streamflow.

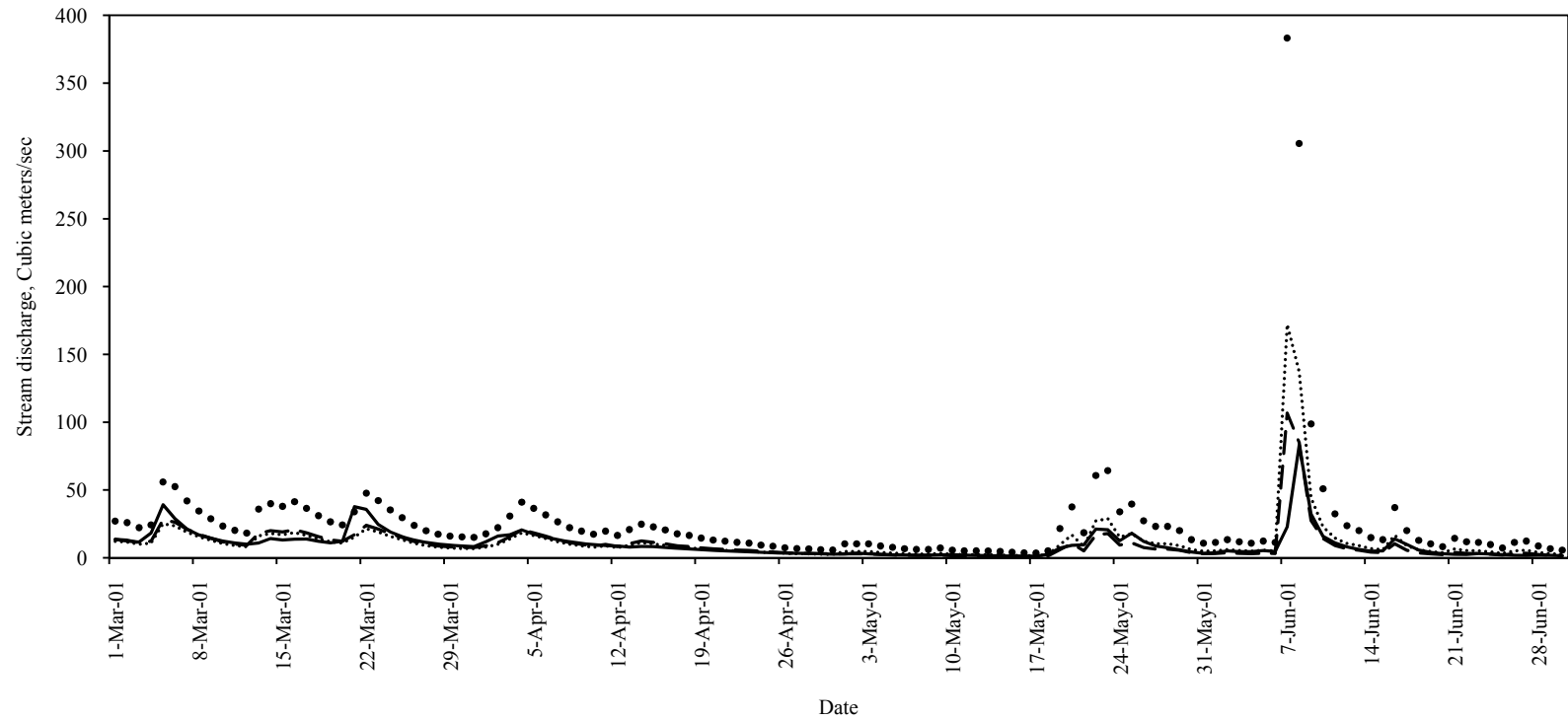
The proposed model relies on the assumption that the watershed is a linear time invariant system, the ATF is constant across any duration of period of observed flow, and the distribution of precipitation is uniform across both downstream and upstream watersheds. The assumption facilitates the use of ATF to determine the system response given an impulse input to the watershed. Additionally, the observed streamflow displayed that the flow transferring function across hydrologic boundaries are a simple ratio of outflow and inflow, which is commonly performed in the frequency domain. Estimation of the transfer function in frequency domain requires extensive computational time which can be avoided by a mere arithmetic procedure proposed in this article. Hence, the proposed method can easily be applied to any basin with any number of observed flow records to obtain the transfer function. The outflow from a watershed can be simulated by multiplying the transfer function with the inflow. Outflow hydrographs estimated by the inflow hydrograph from any tributary channel upstream was also shown to be correlated well with the observed outflow hydrographs. It is to be noted that the procedure can be used in estimation of inflow if the outflow from a basin is known by dividing it by the transfer function.

The model also showed underprediction of flows during dry periods due to the fact that storage and baseflow input were averaged by the estimated transfer function. Therefore, if a discrepancy exists in the model prediction, the inflow to the stream segment should be verified. The model, however, was unable to handle the non-uniform distribution of the precipitation across upstream and downstream watersheds. The karst watersheds also influenced hydrographs resulting from multiplying the inflow with the average transfer function due to the fact that the flow across the karst geology was not accounted for in the basins. The problem of non-uniform distribution could easily be solved by determining inflow from all downstream basins using distributed precipitation input. The karst watersheds could also be modeled with distributed precipitation data so as to account for the accurate quantity of inflow received by the upstream channel. In summary, the proposed method can prove to be a valuable source for flow prediction in ungauged basins, calibration of hydrologic models, and flow forecasting. The method is applicable to estimate outflow hydrographs from any channel located upstream. i.e., the inflow

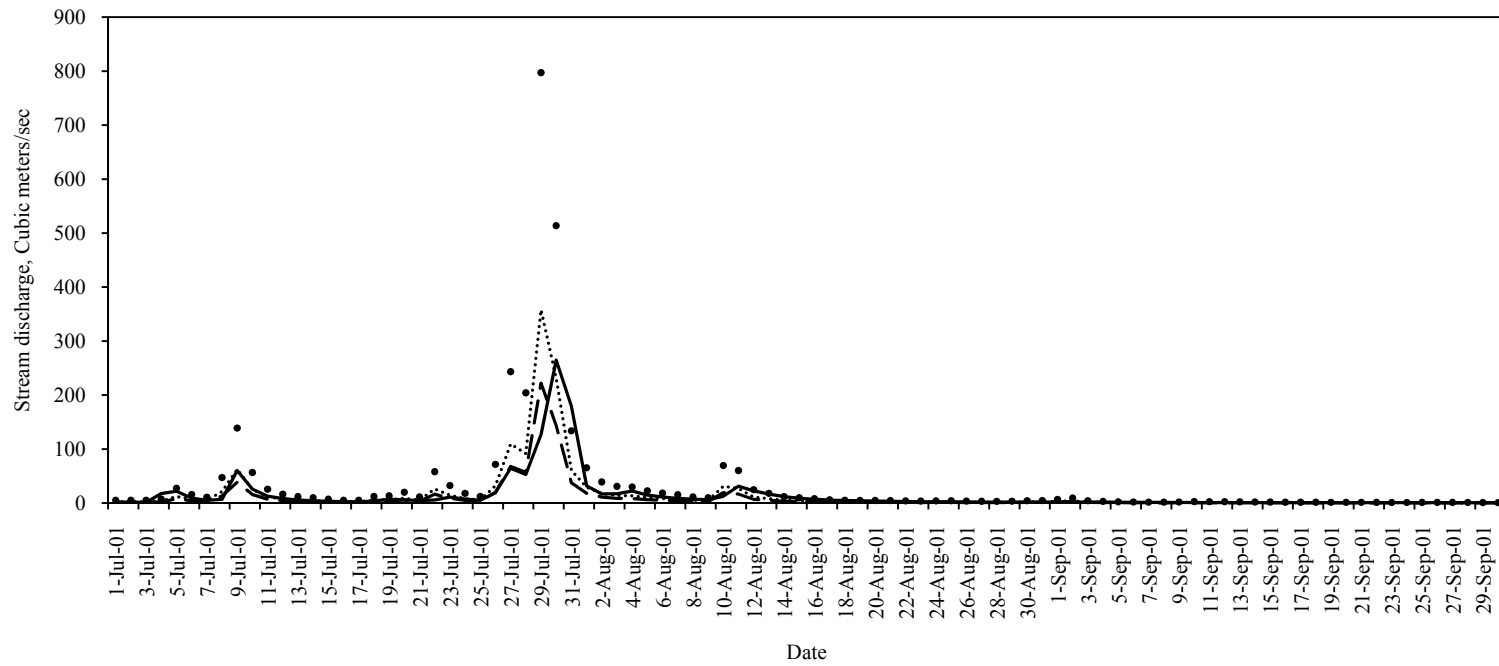
stream need not be in the same stem as the outflow stream. Besides, the average transfer function depends solely on the drainage area, the model can be applied to any size of watershed.



**Figure 2-1. Kentucky River Basin: Elevation and surface water monitoring stations**

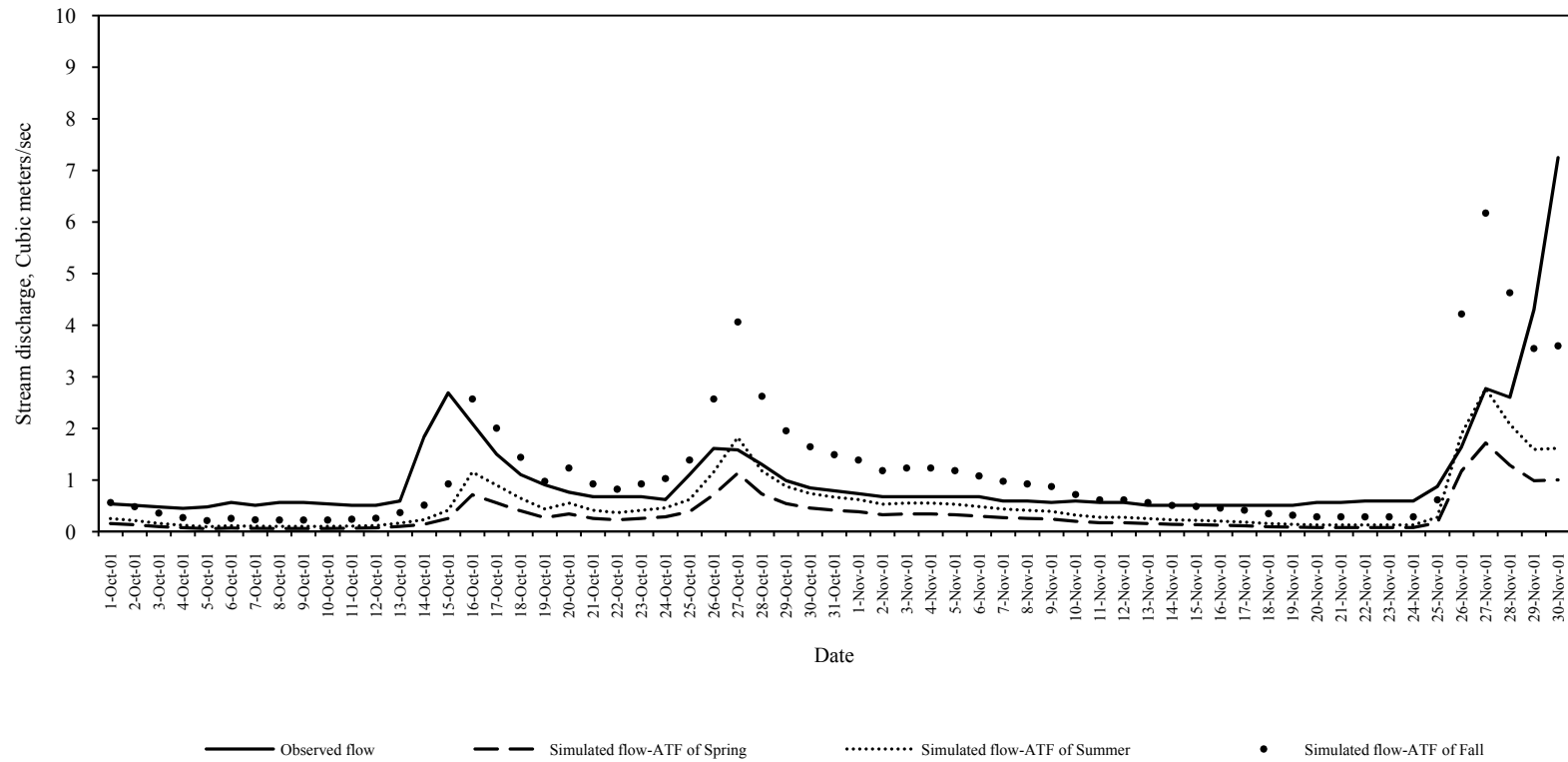


**Figure 2-2a. Evaluation of proposed Average Transfer Function model for Red river near Hazel Green for year 2001 from March through June**



**Figure 2-2b. Evaluation of proposed Average Transfer Function model for Red river near Hazel Green for year 2001 from July through September**





**Figure 2-2c. Evaluation of proposed Average Transfer Function model for Red river near Hazel Green for year 2001 from October through November**

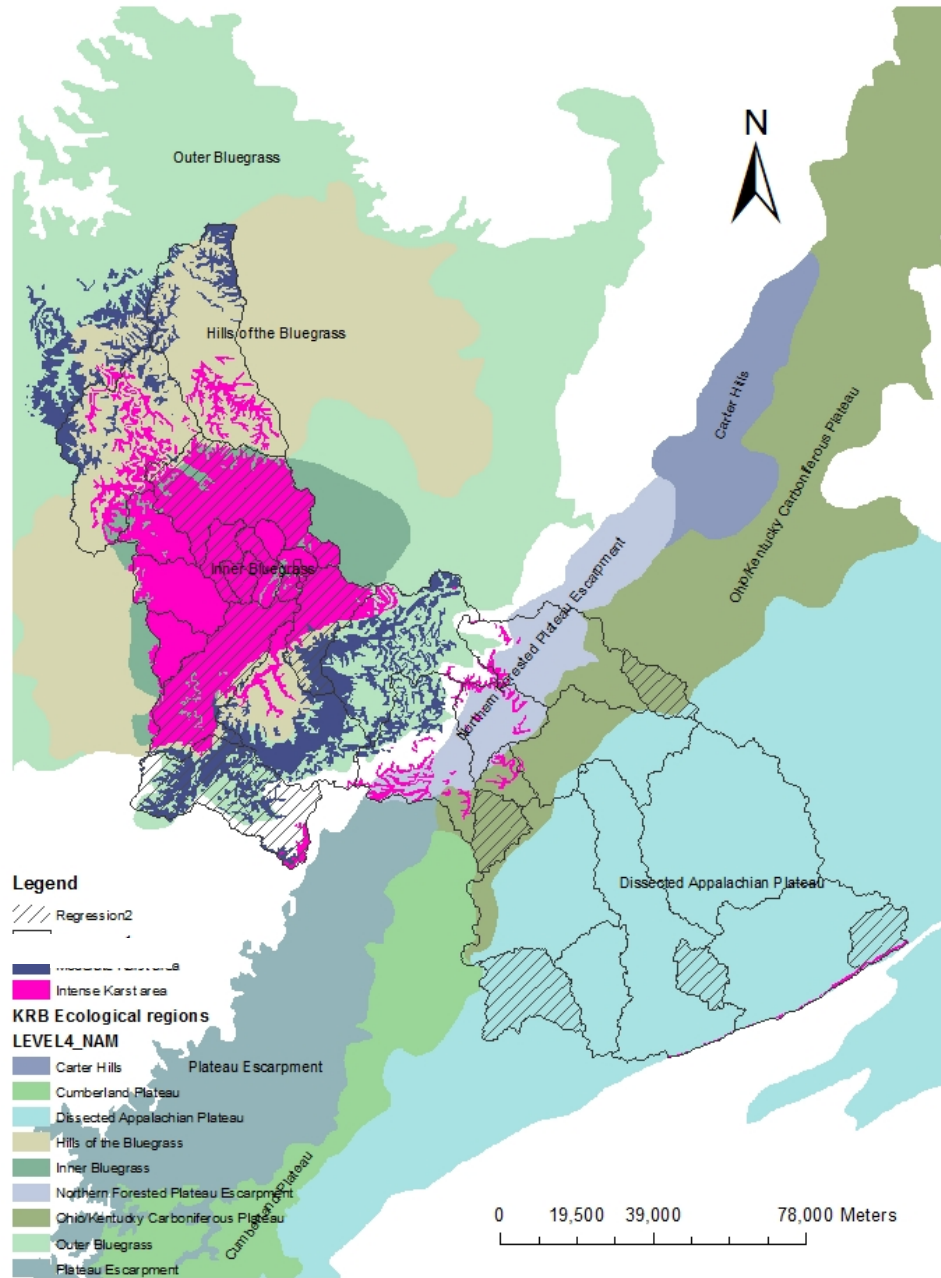
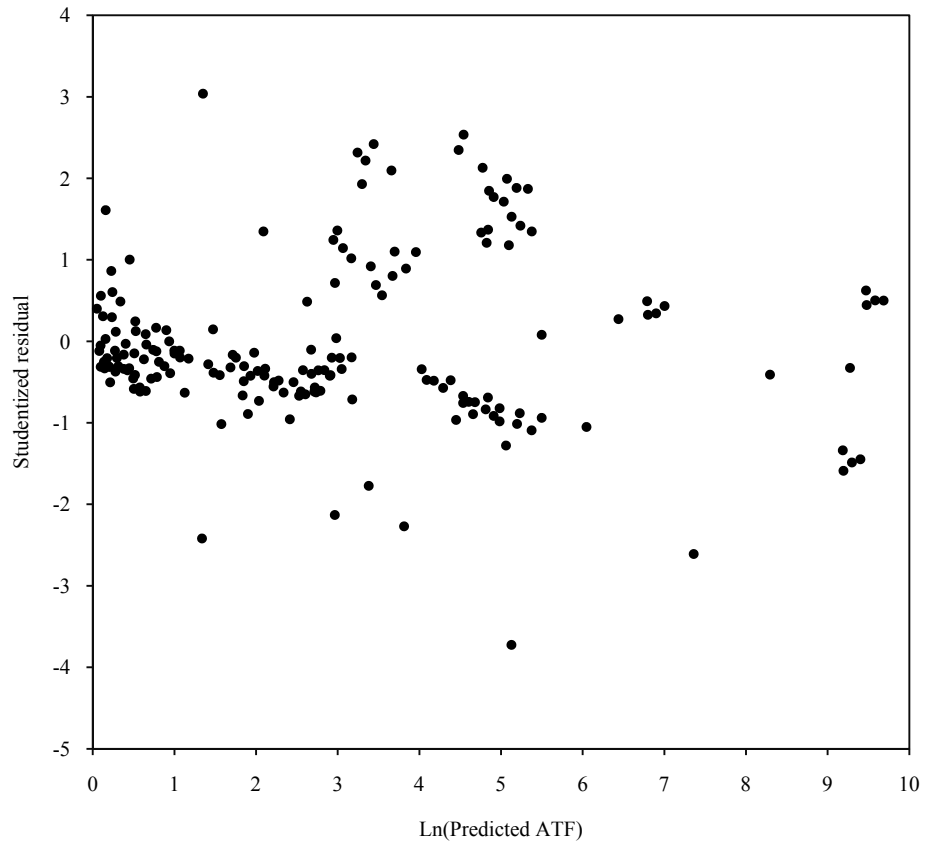
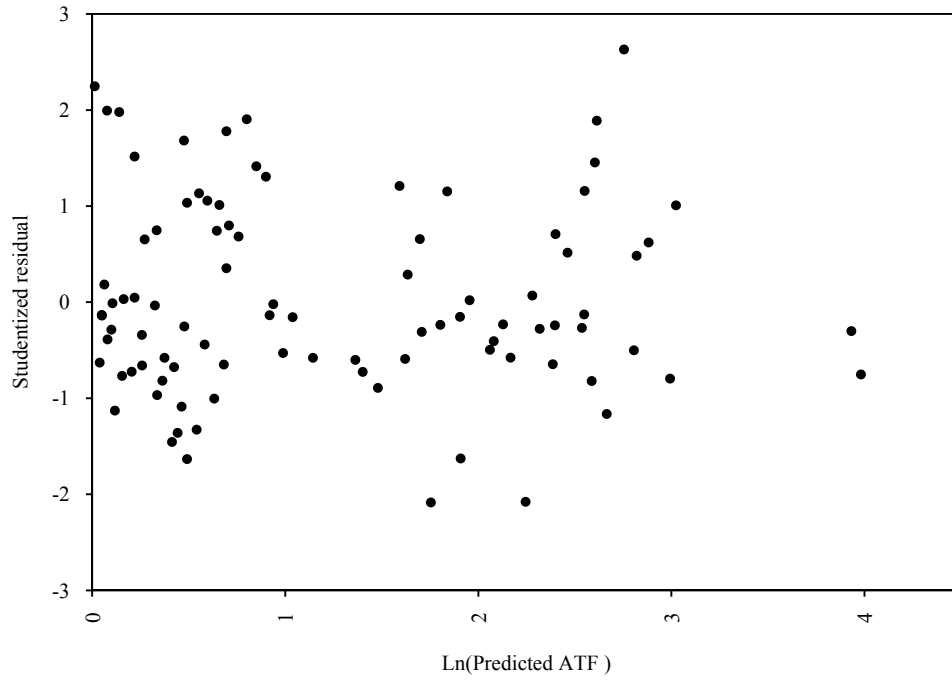
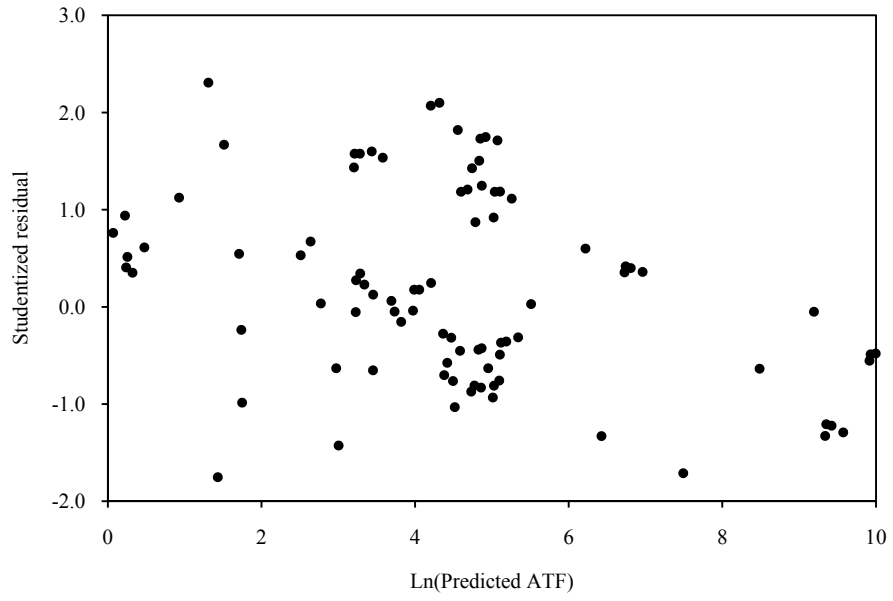


Figure 2-3. Delineated basins, Karst regions, and ecoregions of the Kentucky River Basin



**Figure 2-4. Studentized residual plot of USGS gauge pairs after removing outliers**





**Figure 2-6. Residual plot of Lower Kentucky basins**

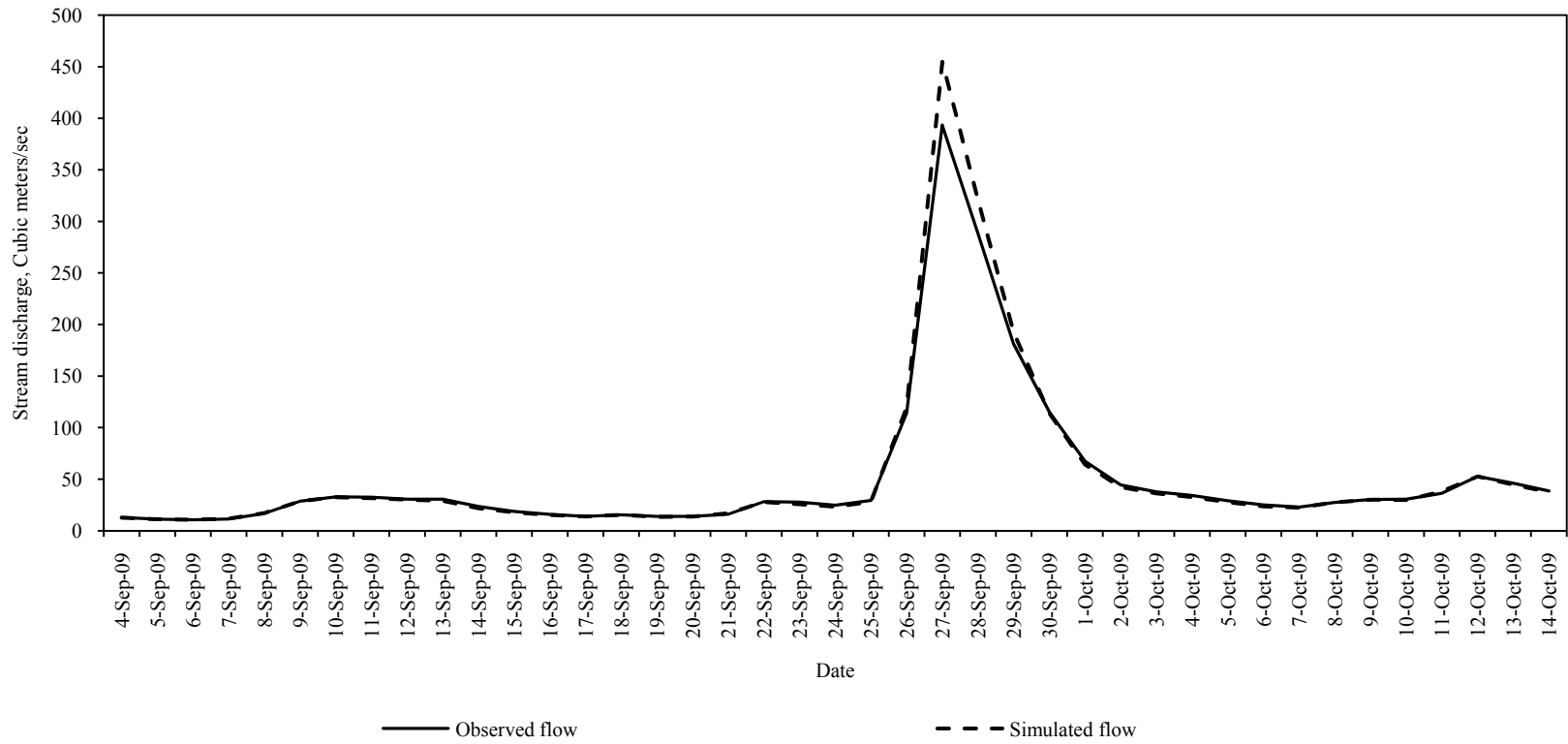


Figure 2-7. Runoff event analysis at KY River at Lock 13

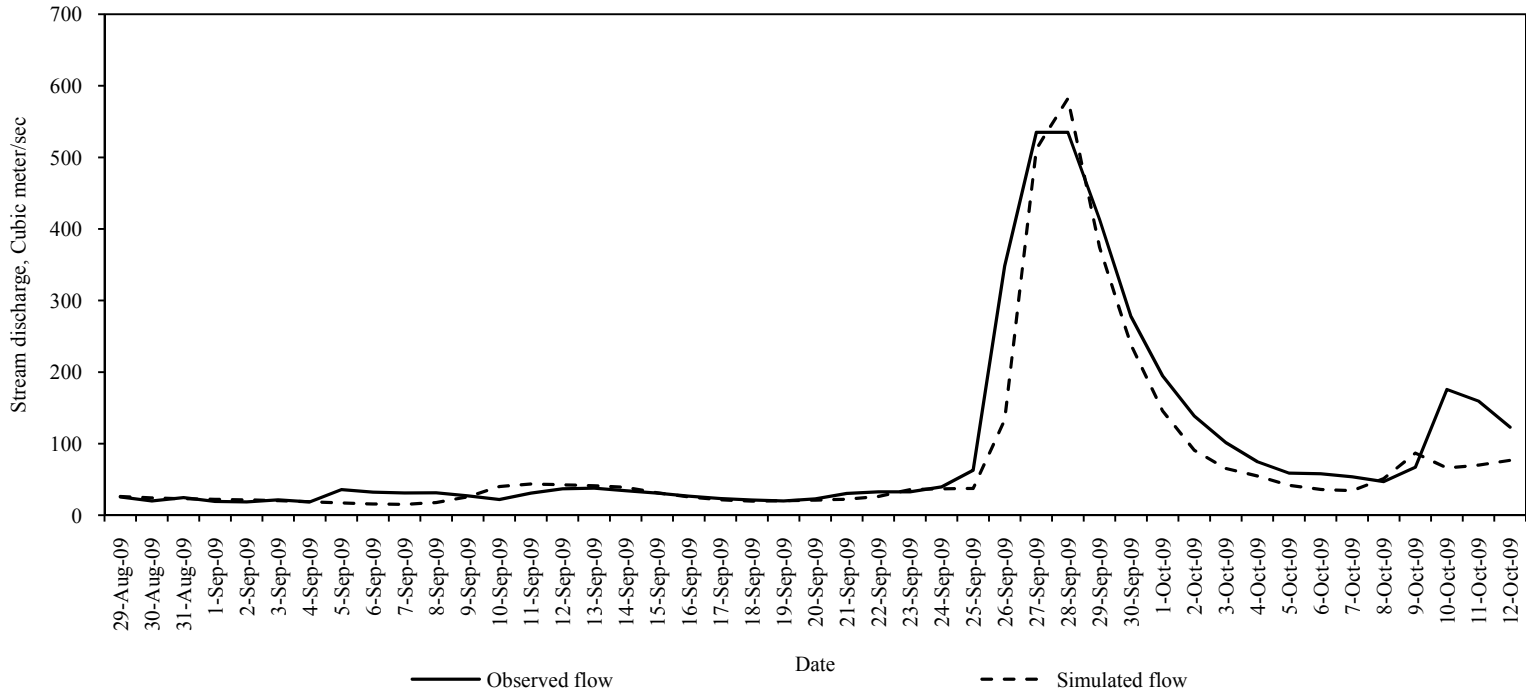


Figure 2-8. Runoff event analysis at KY River at Lock 6

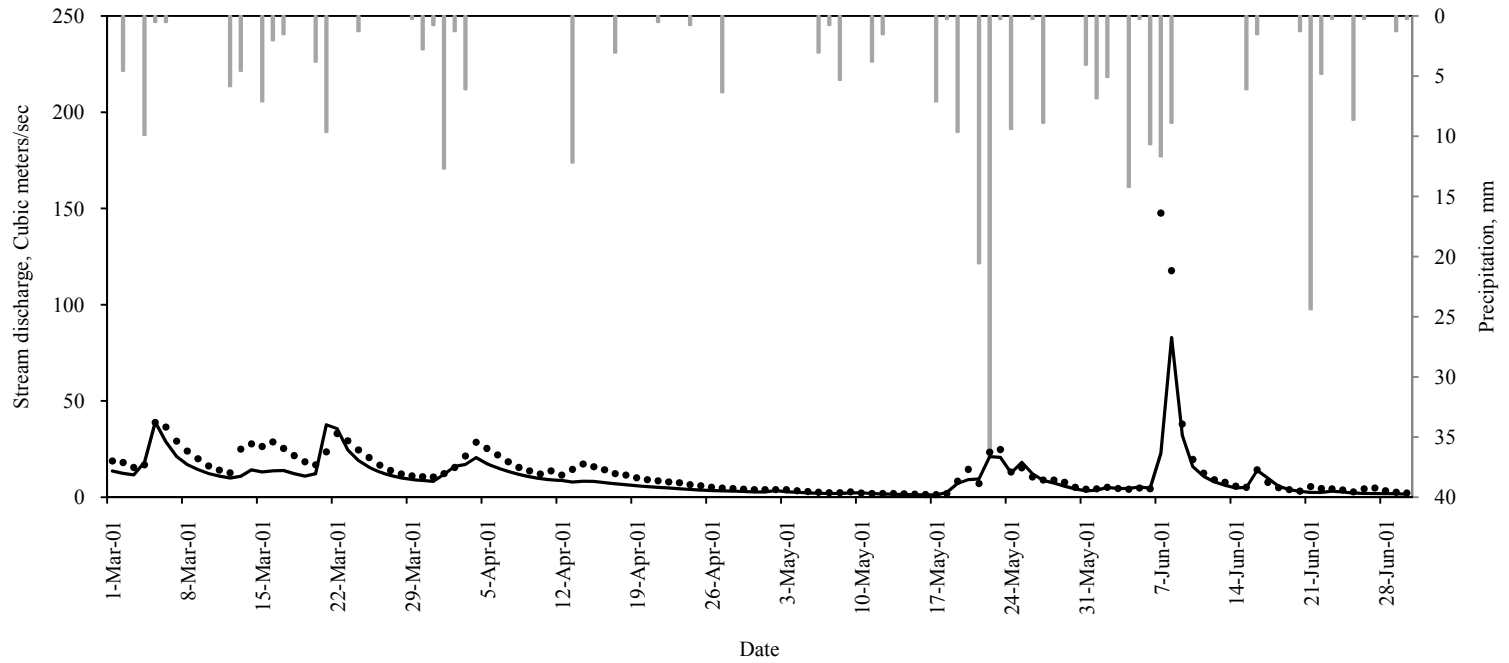


Figure 2-9a. Seasonal hydrograph analysis of Red River near Hazel Green for year 2001 from March through June





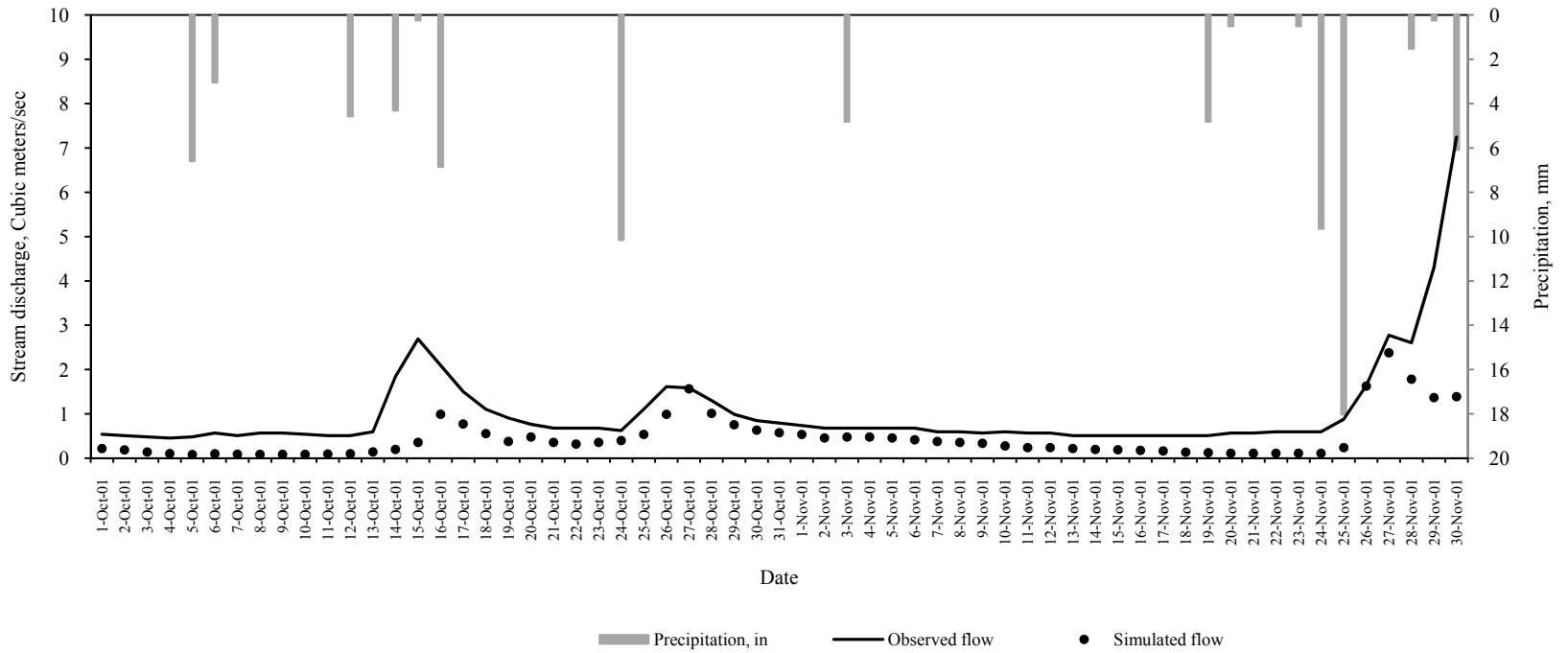


Figure 2-9c. Seasonal hydrograph analysis of Red River near Hazel Green for year 2001 from October through November

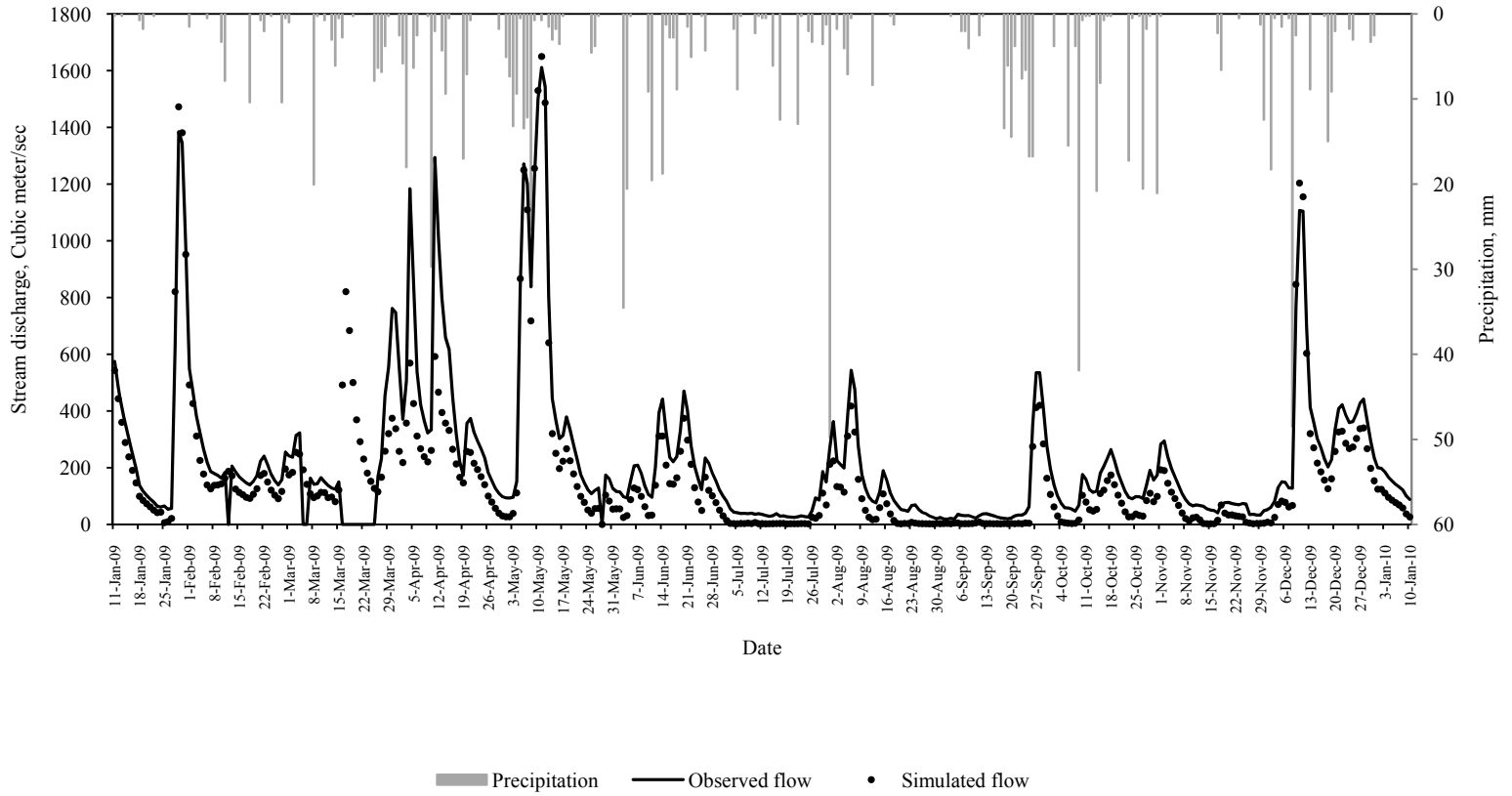


Figure 2-10. Yearly hydrograph analysis of KY River at Lock 7

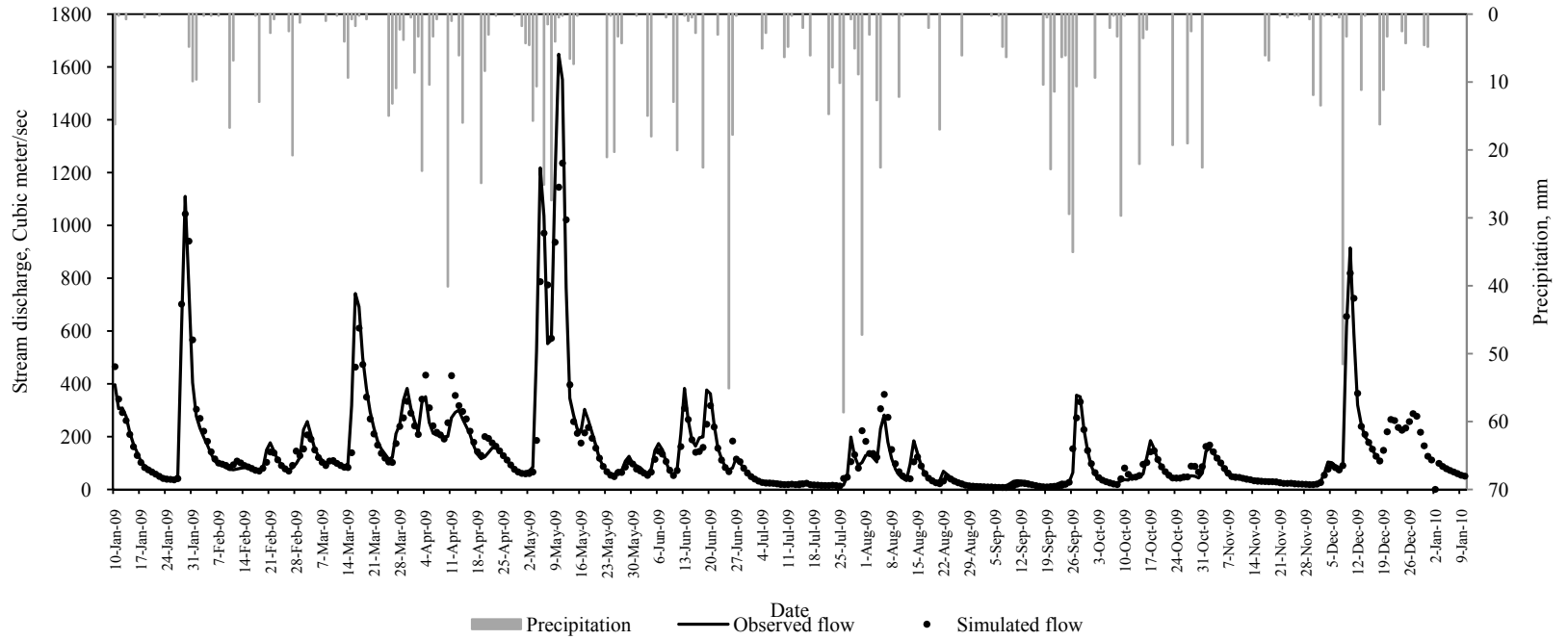


Figure 2-11. Inflow hydrograph analysis of KY River at Lock 8

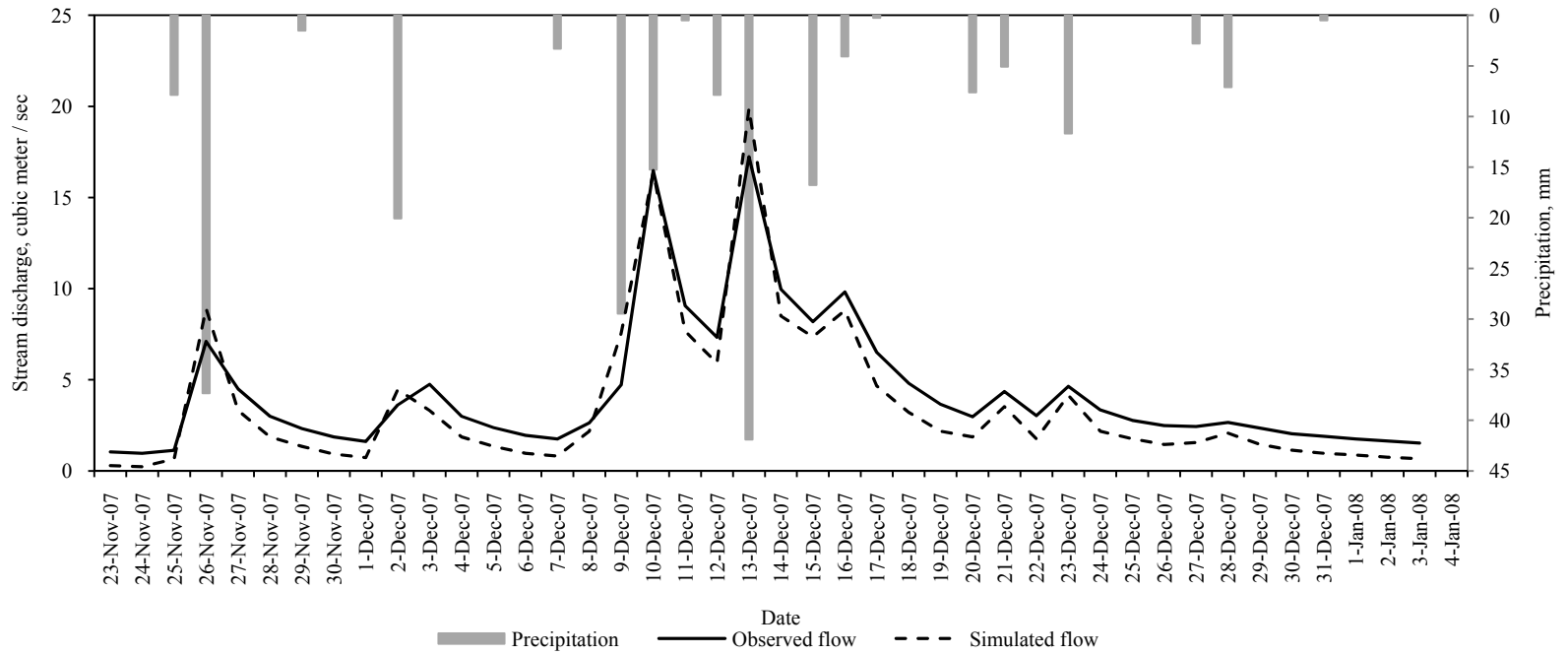
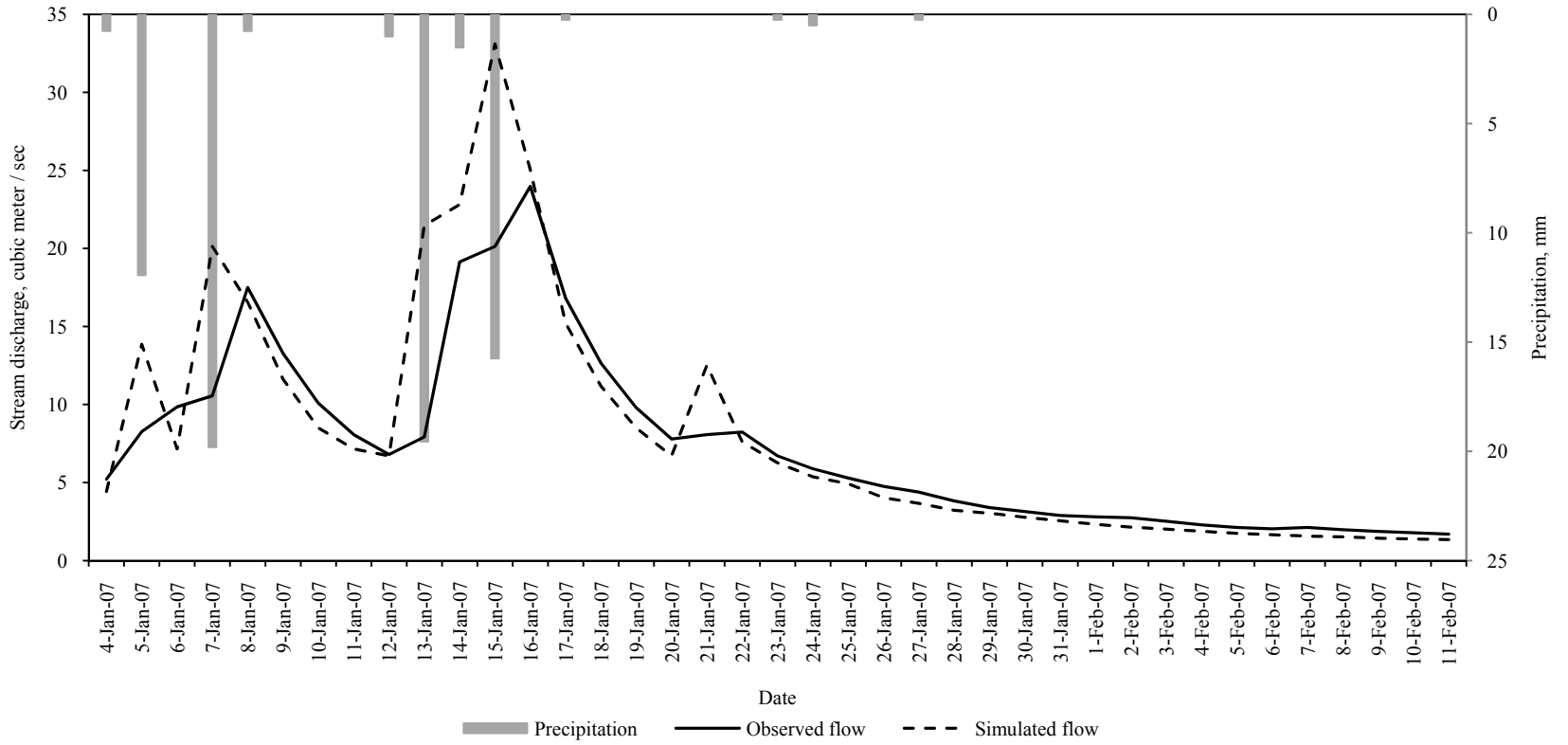


Figure 2-12. Runoff event analysis of Town Branch Creek



**Figure 2-13. Runoff event analysis of South Elkhorn Creek**

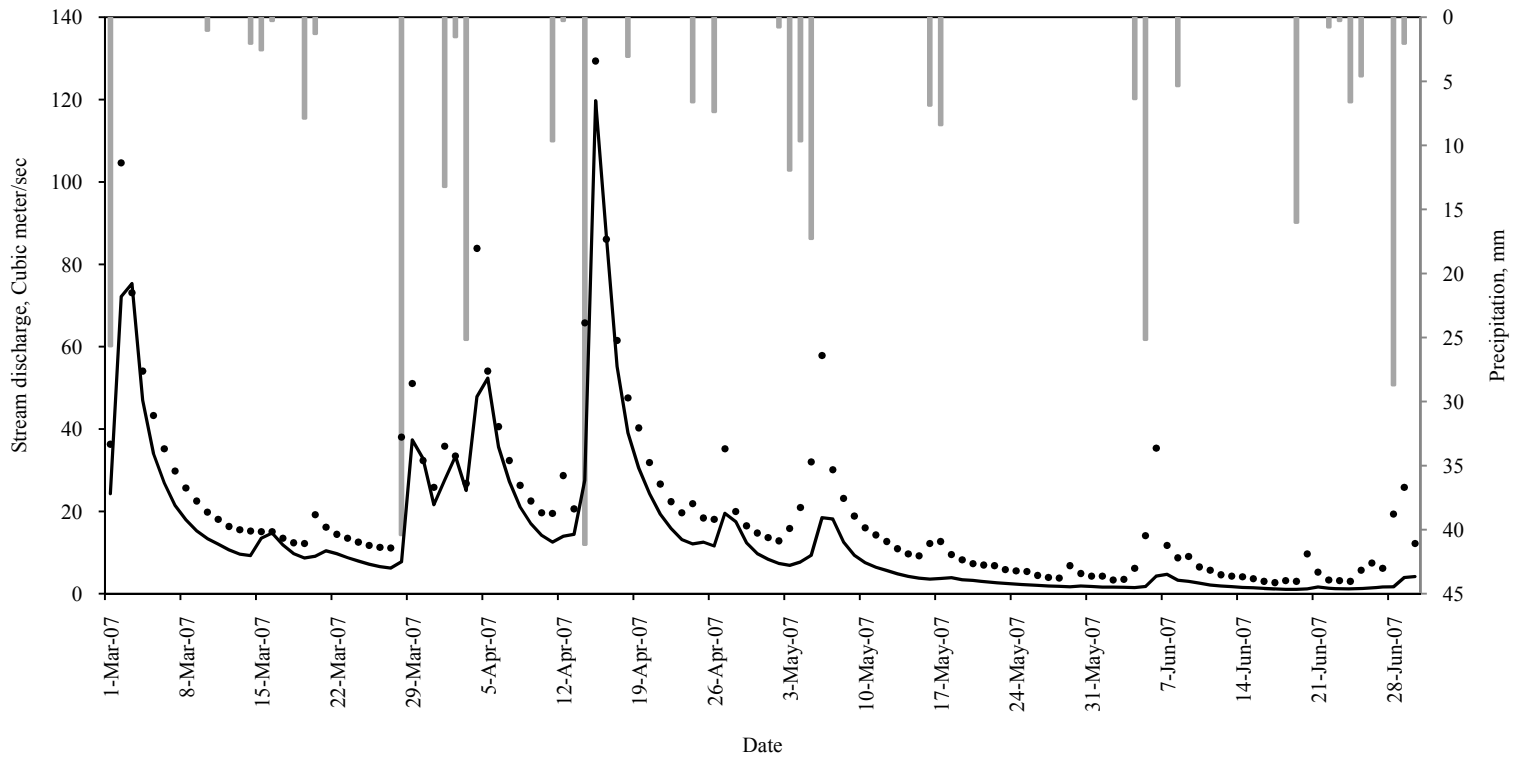
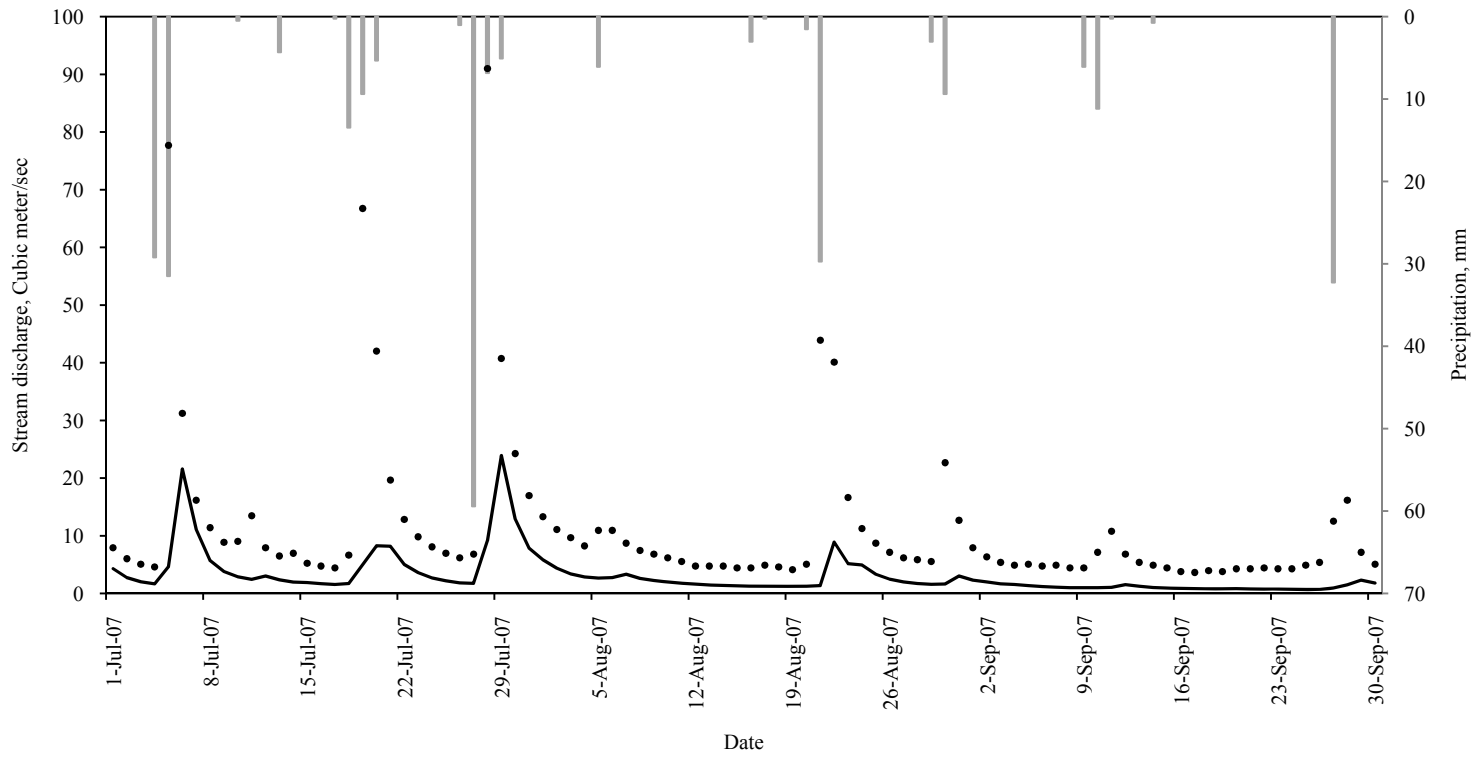


Figure 2-14a. Seasonal hydrograph analysis of Elkhorn Creek for year 2007 from May through June



**Figure 2-14b. Seasonal hydrograph analysis of Elkhorn Creek for year 2007 from July through September**



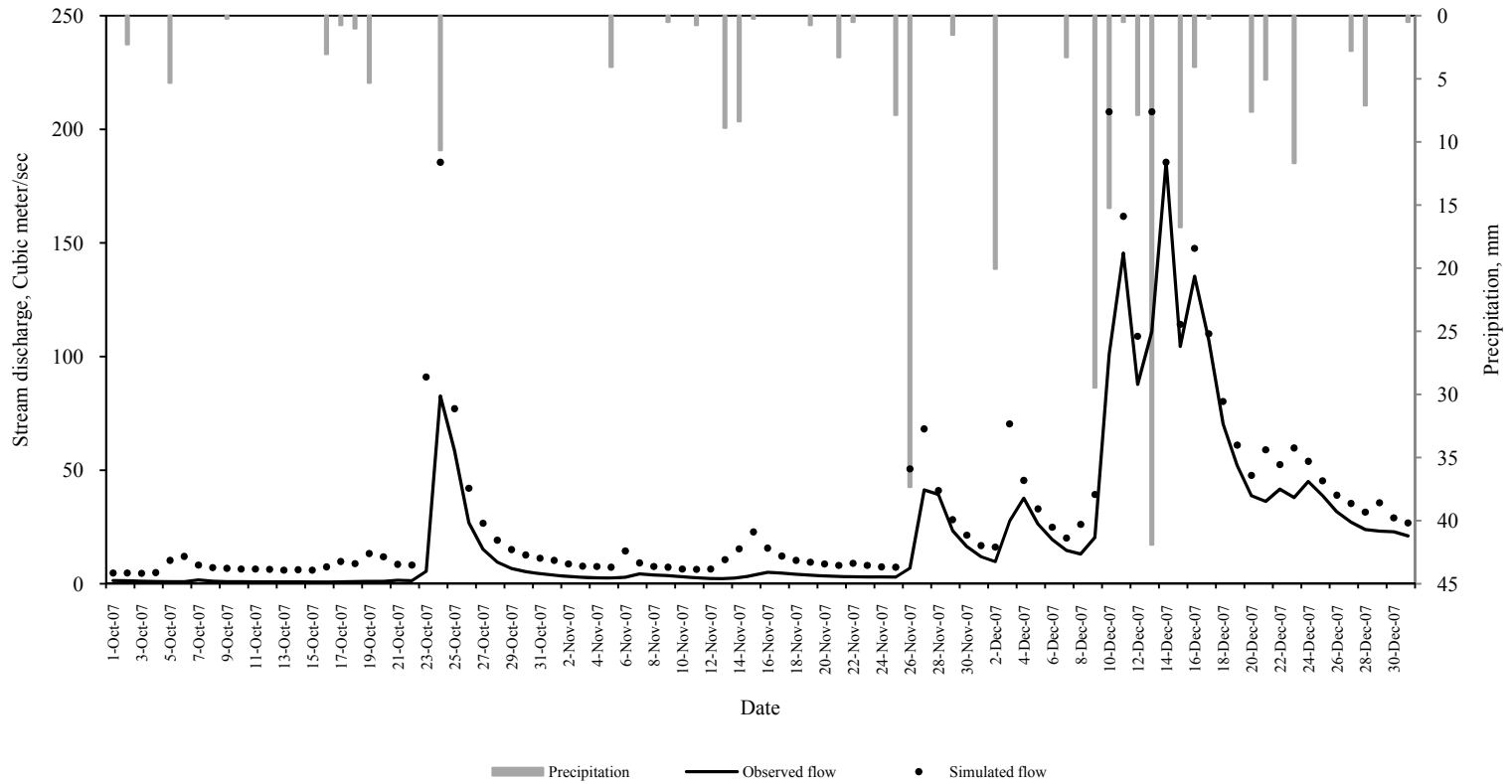


Figure 2-14c. Seasonal hydrograph analysis of Elkhorn Creek for year 2007 from October through December

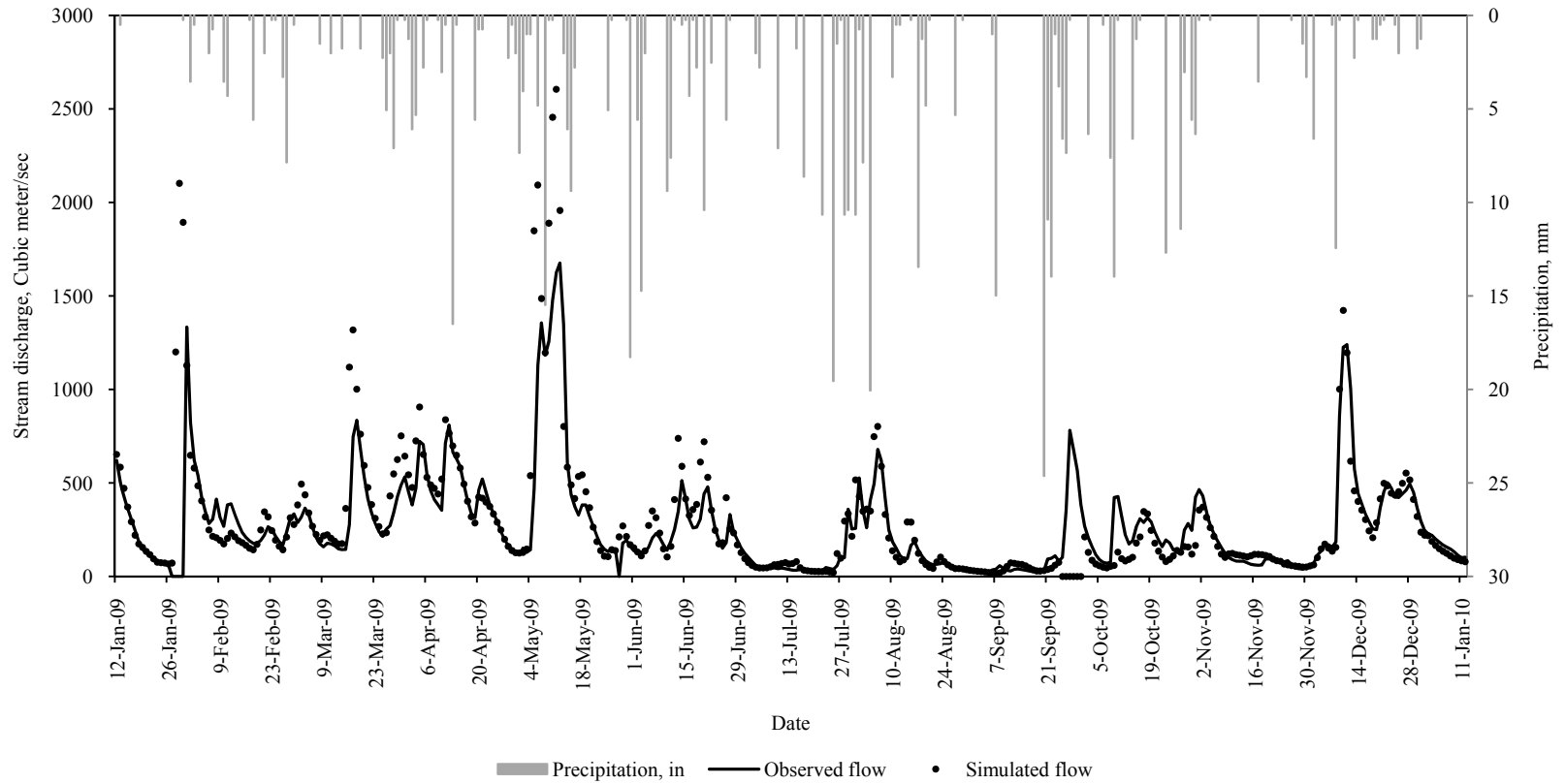


Figure 2-15. Yearly hydrograph analysis of KY River at Lock 2

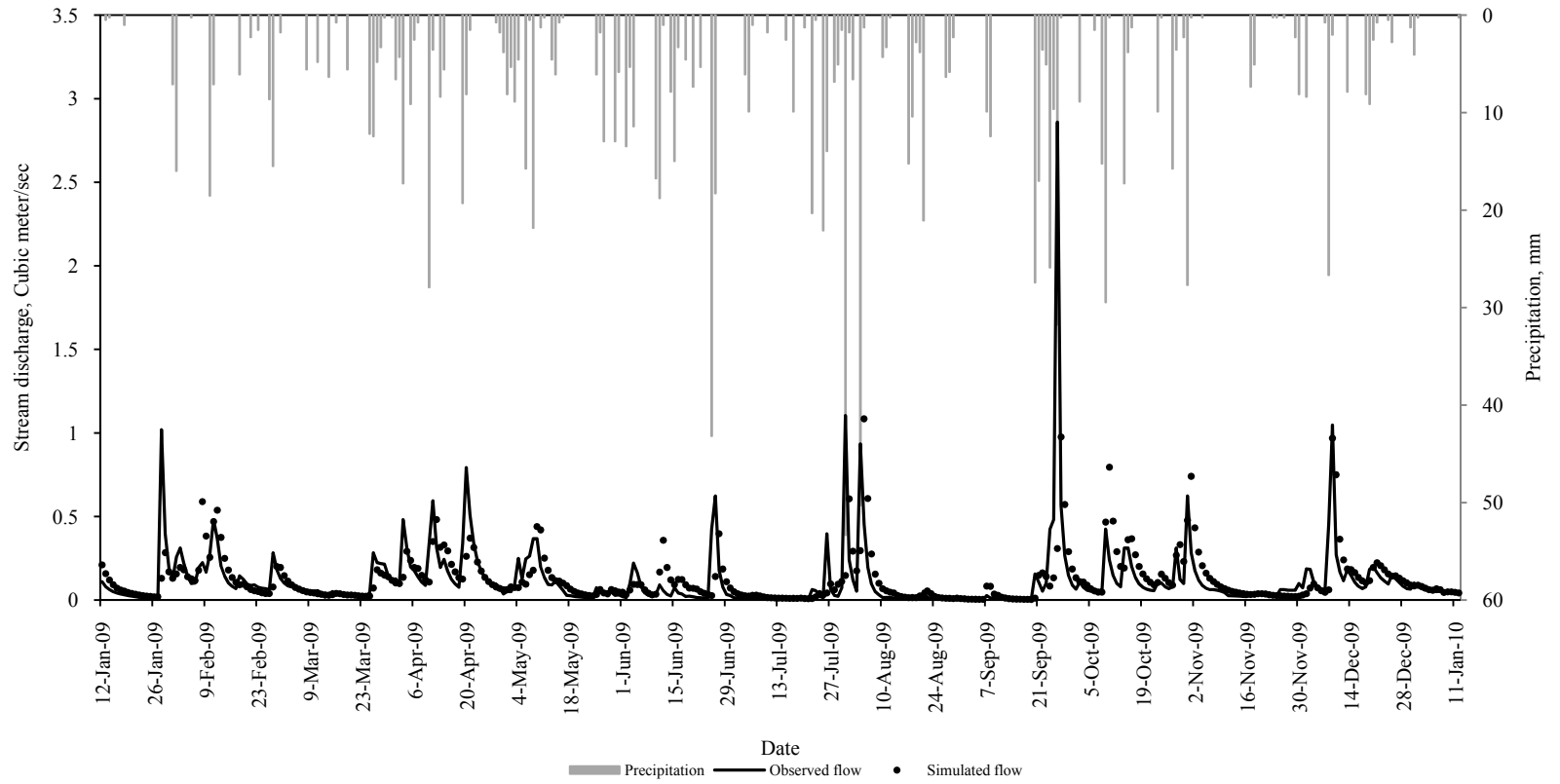


Figure 2-16. Inflow hydrograph analysis of North Elkhorn Creek

**Table 2.1. Example reaches used in validation of the proposed method**

<b>Regression</b>	<b>Gauge pair</b>	<b>Reach Name</b>	<b>Upstream area sq.km</b>	<b>Downstream area sq.km</b>	<b>Analysis Year</b>	<b>Type of analysis</b>
Regression1	03282000-03282060	KY River at Lock 13	6,882	7,210	2009	Event
	03284000-03287000	KY River at Lock 6	10,243	12,950	2009	Event
	03282500-03283500	Red River near Hazel Green	170	763	2001	Season
	03286500-03287000	KY River at Lock 7	13,043	13,214	2009	Year
	03282290-03284500	KY River at Lock 8	8,337	11,432	2009	Inflow
Regression2	03289193-03289200	Town Branch	25	77	2007	Event
	03289193-03299300	South Elkhorn creek	25	272	2007	Event
	03289300-03289500	Elkhorn creek	272	1,225	2007	Season
	03284230-03290500	KY River at Lock 2	10,621	15,498	2009	Year
	03287580-03289500	North Elkhorn Creek	5.7	1,225	2009	Inflow

### 3. Flow routing using transfer function derived from streamflow records

#### 3.1. Abstract

A flow routing function derived from observed streamflow records was tested for flow prediction in ungauged first-order streams in the Kentucky River Basin, Kentucky, USA. The routing function was used in place of the Unit hydrograph in determination of flow at the stream outlet. The flow transferring function was obtained from historic flow records. The function was averaged over the number of days of records to comply with the time-invariant property of the watershed system. The inflow to the streams was determined using TOPMODEL for the years 2005 through 2008 using raingauge measurements. The flow routing function was validated for the year 2009 with raingauge and radar rainfall measurements. The Shuffled-Complex Evolution from University of Arizona was incorporated into the model structure as a calibration algorithm. The model was auto-calibrated with an algorithm using soil properties and initial discharge as calibration parameters. The proposed model produced hydrographs with 80% of Coefficient of Efficiency. The volume of the hydrographs was reproduced well during peak events whereas baseflow events were overpredicted. The validation of the proposed model with radar rainfall estimates surpassed the modeled flow using raingauge measurements.

Keywords: Flow transfer, Average transfer function, TOPMODEL, auto-calibration, NEXRAD

### 3.2. Introduction

Flow generation characteristics of a watershed are exhibited by climate, topography and geology; however, the transfer function that carries the simulated flow from upstream to downstream of the watershed is affected mainly by channel characteristics. When attempting to predict flow from ungauged basins, the outflow can be estimated by multiplying the transfer function and inflow in the frequency domain or integrated in the time domain. However, the validation of the predicted outflow has always been a challenge, when one tries to analyze the performance of such a model. During the estimation of hydrologic components, the constants that drive the hydrology are the watershed characteristics. Therefore, using these watershed characteristics as a way to define transfer function has been widely discussed in the literature. In addition to watershed characteristics, historic flow measurements can also be used to define the transfer function, since the observed streamflow is the resultant of the process that happens in the watershed. As an example, a streamflow series with continuous surface water supply could be descriptive of a perennial stream with constant baseflow contribution. The baseflow is a typical behavior of a forest or land cover with a well-aerated soil structure. On the other hand, a streamflow series with flashy hydrograph represents an urban watershed with impermeable land cover. Therefore, studying any hydrograph will provides an understanding of the watershed and can be availed to determine how that streamflow will be transferred to the outlet. The secured information in observed records can be used for hydrograph identification in ungauged basins if it would be combined with hydrologic modeling for inflow determination.

Hydrologic modeling is performed to generate the water balance components such as overland flow, subsurface flow, and evapotranspiration. The verification of quantities of simulated flow is a significant step to assess the performance efficiency of the selected hydrologic model. The accuracy of the prediction depends on the resolution of the inputs, number of hydrologic processes involved in the model and thus the number of variables used to estimate the components. As the number of parameters increases, the calibration of the hydrologic model becomes strenuous and requires more computation time.

Since the current study focuses on providing a simple hydrologic model with few parameters to calibrate without compromising the efficiency of prediction, TOPMODEL was selected as a base to calculate the hydrologic components. Fundamental concepts underlying TOPMODEL to generate runoff were analyzed with the help of numerical schemes that integrate the classic water balance components of runoff generating saturated areas and catchment surface storage (Kavetski et al., 2003). The principal component of TOPMODEL is the topographic index that delineates runoff generating saturated areas. The topographic index is a function of hydraulic

gradient and soil transmissivity (Pan et al., 2004) and it is obtained from Digital Elevation Models (DEM) and soil maps. The surface runoff volume is proportional to topographic index in which, the subsurface flow is calculated as a function of watershed average storage deficit, change in hydraulic conductivity with depth and topography (Wolock et al., 1990). The storage deficit in TOPMODEL is the identifier of saturated surfaces in the watershed. The local storage deficit (LSD) value of zero or less than zero indicates a fully saturated surface and LSD with a value of greater than zero indicates that the deficit needs to be fulfilled before producing runoff (Kinner and Stallard, 2004). The evapotranspiration (ET) is estimated as a percent of potential ET, which can be derived from the root zone deficit. The unsaturated zone storage is updated as the root zone surpasses its field capacity. Finally when there is deficit in the soil profile, recharge to ground water is allowed as a function of amount of water stored in unsaturated zone.

During the developmental period of TOPMODEL, several routing algorithms were introduced into the model to increase the prediction efficiency (Gallart et al., 1994; Franchini et al., 1996; Shrestha et al., 2007; Andreassian et al., 2001). A non-linear convolution algorithm was used to route flow to the watershed outlet (Beven et al., 1984). The model was applied to three catchments dominated by impermeable soil profile and the simulated hydrographs suggested a more complex routing algorithm is needed to increase model prediction efficiency.

Efficiency of TOPMODEL was evaluated for submediterranean catchments by Durand et al., (1992) and the study concluded that the model might provide implausible runoff where Hortonian flow conditions exist. In their study, basic hydrologic processes were reproduced by TOPMODEL; however, modifications in infiltration module, and evapotranspiration module were suggested by the authors. Robson et al., (1993) used TOPMODEL for Balquhider catchments with saturated flow generating conditions. Although, TOPMODEL generates runoff from saturated conditions, it was recommended that the model might not work for catchments with a shallow water table due to quasi-steady state assumption in unsaturated zone. In addition to agricultural and Mediterranean regions, TOPURBAN - an urban version of TOPMODEL, was used for hydrograph generation of urban areas. Since a variable source area concept based runoff generation mechanism is used in TOPMODEL. TOPURBAN outperformed the comparative model QualHYMO that uses the infiltration excess runoff producing technique (Valeo et al., 2001). The TOPMODEL was used in the prediction of floods in mountainous terrains (Taschner et al, 2001). The authors successfully demonstrated that the ability of the model to produce actual scenarios depends highly on the inputs and initialization of model parameters. The raingauge measured data were found to outperform the radar data for flood prediction. In this context of input resolution, Pradhan et al., (2006) showed the effect of scale of DEM to obtain topographic

index in model simulations. Their article describes the differences in modeling results when the scale at which the model applied and scale at which the inputs are derived from, varies from each other. The TOPMODEL was analyzed for the purpose of prediction of soil erosion and sediment yield for individual rain storms (Wang et al., 2009). The erosion parameters such as the transport capacity of flow were included in the blockwise TOPMODEL. The model was able to simulate flow with greater than 90% Nash-Sutcliffe efficiency and the model was able to predict sediment yield within a 20% of acceptable error range.

An integrated three-layer TOPMODEL by Huang et al., (2008) provides a calibration procedure for identifying the most sensitive parameters that governs the prediction of shape and volume of hydrograph. Unit response function, solved by Manning's equation and energy dissipation theory, was used in their study for flow routing. Kennen et al., (2008) developed a hydro-ecological model that incorporates TOPMODEL for hydrologic components. A multiple linear regression analysis was used to select hydrologic and environment variables that are important for aquatic-invertebrate assemblages. To remove multicollinearity among the variables, principal component analysis was used in the study. Recently TOPMODEL processes of runoff generation were made available in a graphical user interface with automatic calibration technique (Wang et al., 2005).

### 3.3. Objectives

The study focuses on,

- Validation of the routing procedure developed using the Average Transfer Function obtained from observed streamflow records
- Validation of the proposed routing procedure to facilitate the use of TOPMODEL for flow prediction in ungauged basins

### 3.4. Materials and Methods

#### 3.4.1. Description of the study area

Goose Creek, a subbasin of South Fork Kentucky in Kentucky River Basin (Figure 3-1) with no karst was selected to evaluate the performance of the proposed method. Goose Creek drains portions of Clay and Knox counties. The creek drains 417 km<sup>2</sup> that runs for 38 km with an annual average discharge of 7 m<sup>3</sup>/sec and average annual peak of 872 m<sup>3</sup>/sec. The dominant landuse of the selected basin is forest. South Fork Kentucky watersheds are in hilly terrain with rapid surface runoff and moderate groundwater drainage with few or no sink holes and springs. The major portion of the area is underlain by shale and sandstones. (Kentucky Water Resources Research Institute, 2002).



TOPMODEL generates possible flow producing areas by taking into account the slope and area draining in a watershed, the Topographic Wetness Index (TWI). The watershed is divided into bands of TWI from upstream to the outlet. The areas closest to channel boundary and the outlet are assumed to represent probable areas of saturation excess. The foremost important variable in the model is the parameter “m” (VarM), which is defined as the ratio of drainable porosity and slope of surface hydraulic conductivity vs. depth (factor Ksat\_f). Drainable porosity is determined as the difference between the porosity and soil moisture at field capacity. Therefore, VarM can be thought to define the volume of drainable water at the rate given by the factor Ksat\_f.

To determine the saturation areas, the local storage deficit is determined as a function of VarM and maximum subsurface flow. The maximum subsurface flow is determined as a function of average areal topographical indices and spatially averaged surface hydraulic conductivity. When the local storage deficit is less than zero, then the area is assumed to produce overland flow; otherwise, the input is applied to the deficit. The Time-Area histogram is used to calculate overland flow that reaches the watershed outlet in the original TOPMODEL. The evapotranspiration is estimated as a function of potential evapotranspiration (PET) and root zone storage deficit.

#### 3.4.2. Dataset preparation

The elevation dataset was obtained from the USGS-National Elevation Dataset (NED) with 10 meters resolution for the KRB. The area draining ( $a_{drain}$ ,  $L^2 / L$ ) through the watershed was calculated as a function of Flow Accumulation. The TWI was then calculated as,

$$TWI = \ln \left( \frac{a_{drain}}{\tan(slope)} \right) \quad (1)$$

Where,

$a_{drain}$  – Area drained by the pixel

The USGS-National Hydrography Dataset (NHD) was used to obtain stream network and hydrologic divides were obtained from NHD-Basins. The 10 ordinate histogram of TWI for each NHD hydrologic divide was calculated using ArcGIS - Visual Studio .NET package (Appendix D). Since NHD-basins ranged from 0.001 km<sup>2</sup> to 9.5 km<sup>2</sup>, division of each of these basins into 10 bands of TWI did not affect the modeling process as opposed to the default 30 ordinate histogram for the TOPMODEL program.

The soil data required for modeling was extracted from the Soil SURvey Geographic (SSURGO) dataset (Appendix C). The soil parameters needed for deriving TOPMODEL components are:

- Soil moisture content at field capacity
- Wilting point
- Bulk density
- Available water content
- Saturated surface hydraulic conductivity

These parameters were extracted from the SSURGO dataset for each basin. Major soil components for a basin were obtained and the soil parameters were estimated on a component-average basis as,

$$\text{soil parameter, } L = \frac{\sum(\text{component percentage} * \text{depth})}{\sum \text{percent}} \quad (2)$$

The daily rainfall data for each basin was obtained from the National Climatic Data Center (NCDC) raingauges and NEXRAD radar inventory. The raingauges with at least five years of data and located in and around the selected basin were selected. The point measurements were interpolated using Inverse Distance Weighting (IDW) method in ArcInfo workstation. The mean precipitation in each of the NHDPlus basins was obtained using Zonal Mean function in ArcGIS (Appendix B).

NEXRAD data processing was performed in ArcGIS-Visual Studio.NET package. The NEXRAD files downloaded from NCDC archive are in XMRG binary format and are given in polar coordinates. In order to be used with the current project framework, the polar coordinates were converted to earth spherical coordinates. The reprojected dataset was then processed to get mean subbasin precipitation (Appendix A).

The potential evapotranspiration was obtained using the Blaney-Criddle method (Ponce, 1989):

$$PET = p \times (0.46T_{avg} + 8) \quad (3)$$

where,

$PET$  - Reference crop ET, mm/day

$p$  - Mean daily percentage of annual daytime hours

$T_{avg}$  - Average daily temperature, °C

It was studied to overestimate up to 40% in humid, clouded areas whereas in windy, dry, sunny areas the underestimation could be up to 60% (Irrigation water management: Irrigation water needs, 1986). In this research, the Blaney-Criddle method was analyzed by comparing the estimated reference ET with the reference PET calculated by multiplying pan evaporation measurements obtained from University of Kentucky Agricultural Weather Station with the pan coefficient of 0.85, assuming a Class A evaporation pan (Irrigation water management: Irrigation water needs, 1986). The PET estimates from Blaney-Criddle method were also compared against the PET estimates from Thornthwaite method (Ponce, 1989). As Figure 3-2 show, Blaney-Criddle produce PET values as close to the typical ET values for the study area (6 -7 mm/day).

The urban areas in the watershed were treated as proposed by Valeo and Moin (2000) (Appendix E). The storage ( $S_{Urb}$ ) and outflow from storage ( $Q_{Urb}$ ) from urban areas were calculated using the following equations:

$$S_{urb} = S_{urb}(t - 1) - Q_{Urb} + Flow_{Urban} \quad (4)$$

where,

$Q_{urb}$  -  $Q_0 e^{-1/T}$

$Flow_{urban}$  - Flow from urban area = (Imp x  $area_{urban}$ ) x precipitation

$imp$  - Percent Imperviousness coefficient

$area_{urban}$  - Percent of urban area in watershed

$T$  - Time delay

Macro flow was also allowed in the model (Wolock, 1993) as a fraction of precipitation that directly reaches the unsaturated zone storage, bypassing the root zone to allow flow through fractures in the soil profile. Approximated fractions of 0.5% was selected as the values below or above this percent were shown to affect the peak flows in the simulated hydrograph.

### 3.4.3. Channel routing method

The saturation excess overland flow was calculated using TOPMODEL for years 2005 to 2008 on a daily basis for NHD-Basins. The overland flow was added to the outlet instead of being routed to the channel because of the small subbasin areas. The flow was routed to the main watershed outlet using the regression analysis presented in Part I of this research article, i.e,

$$Q_{out} = ATF * Q_{in} \quad (5)$$

The ATF is the average transfer function obtained from streamflow measurements and  $Q_{in}$  is the overland flow generated from TOPMODEL and flow from upstream watershed. ATF is determined to use in Equation (5) as,

$$ATF = 1.15 \times \frac{(\text{Drainarea}_{downstream})}{(\text{Drainarea}_{upstream})} \quad (6)$$

where,

$\text{Drainarea}_{Downstream}$  is the watershed area drained by the downstream watershed

$\text{Drainarea}_{upstream}$  is the watershed area drained by the upstream watershed

Equation (6) is then multiplied with the overland inflow hydrograph produced by TOPMODEL to produce flow at the subbasin outlet. The preliminary analysis of simulation showed that the routing procedure adopted in the TOPMODEL using equation (5), affected the hydrograph such that the recession limb mirrored an urban hydrograph as shown in Figure 3-3 because the storage in the channel was not included. The hydrographs from Part 1 were predicted with good agreement with observed flow because the inflow used to simulate the flow included storage; whereas in this validation Part 2, the inflow is only the overland flow.

Therefore, the routing procedure was modified such that storage in current day was calculated as a proportion of total inflow to the stream. The proportion, by which the storage is calculated, was defined based on the fundamental principles of unit hydrograph i.e., the unit hydrograph values can only range between 0.0 and 1.1. Any watershed that has a unit hydrograph value, in this study referred to as ATF, that exceeds 1.0, will store water according to the explanation below:

*For a channel, if the ATF is estimated to be 1.1x, only 1x inflow is output to the next channel. The remaining 0.1x flow is designated as storage in the channel. This procedure was adopted based on the principle of conservation of mass i.e., the output equals to the input and change in storage (See Appendix I for the routing method in VB.NET)*

By this modified procedure, one can account for water stored in a channel segment as will be discussed from Figures 3-5 through 3-9.

The model efficiency was evaluated using the coefficient of efficiency. The proposed method was calibrated for years 2005 through 2008 using raingauge measurements. The

calibrated parameters were validated using both raingauge and radar rainfall values for the year 2009. Table 3-1 shows calibrated parameters derived from the SCE auto-calibration method. The raingauge measurements were complete for the years 2005 and 2006 and the average, of over 1000 mm, which was equal to the annual average of the study area. The missing data from 2007 through 2009 were replaced with zero precipitation values and the average rainfall for these years was only half of the annual average rainfall value. Therefore, for analysis of simulated hydrographs, the events from years 2007 and 2008 were excluded.

### 3.5. Results and Discussion

The calibrated hydrographs are given in Figures 3-5 through 3-9. The runoff events were selected randomly from years 2005 through 2006 and the estimated coefficient of efficiency for each of the events is given in Table 3-2. The 2005 event (Figure 3-5) shows a good agreement with the observed hydrograph with an efficiency of 0.66. The peak flow is underpredicted for an average rainfall event of 47 mm (Figure 3-4). The 2006 events (Figure 3-6) show better simulated hydrographs than the 2005 event with efficiency ranging from 0.67 to 0.80 and the overall efficiency for the multi-modal hydrograph is 0.71, which is mainly attributed to the distribution of rainfall in the vicinity of the watershed outlet (Figure 3-4). The analysis of precipitation for this runoff event showed that most of the precipitation occurs in the upstream areas of the watershed and therefore, the flow reaching the outlet is reduced greatly. Figures from 3-7 through 3-9 are the blow up of each mode of hydrographs in Figure 3-6.

The simulated hydrographs were also analyzed in the perspective of inputs and the routing procedure adopted in this research. The input category is mainly limited to precipitation and soil parameters such as surface hydraulic conductivity and transmissivity. The effect of distribution of precipitation was seen in the volume of flow that reaches the watershed outlet. The more the uniformity in distribution of precipitation across the subbasins, the better the rising and recession limbs are reproduced correctly such as in Figure 3-7. As rainfall non-uniformly fell across the watershed, the volume and shape of the hydrograph was significantly affected such as October and November outflows in year 2006 (Figures 3-8). Comparison of peak flows between Figure 3-5 and Figure 3-7 shows that April 30, 2005 peak event was underpredicted significantly whereas the September 23, 2006 runoff event was accurately reproduced. The only variable that affected the differences in peak was assumed to be precipitation, since the model applied the same soil properties for the years from 2005 through 2006. The proposed model also captures the fact that when the peak rainfall is received in the upstream of the outlet the consecutive rainy days near the outlet could not reproduce the peak flow (Figure 3-9). From Figure 3-4 it can be

seen that even with equal amount of rainfall intensity, the variation in distribution of the precipitation across the topography might have played a vital role in the simulation process.

Before adapting the auto-calibration procedure, manual calibration was performed in order to identify the minimum and maximum values of variable VarM. The initial root zone deficit, the available soil water capacity, soil transmissivity and initial discharge can directly be observed and validated according to the soil database. Initially the VarM was set to the estimated value as defined by the ratio of drainable porosity and  $K_{sat\_f}$ .

As the value of VarM decreased from the initial value, the simulated hydrograph showed flashy falling limbs similar to an urban hydrograph. An increase in VarM resulted in hydrographs that mimics the actual scenario. In addition, in order to be able produce the peak values, the VarM was increased to 0.3 meters. The range of VarM input to the auto-calibration method was 0.005 to 0.5. Besides VarM, soil transmissivity ( $T_0$ ) had a vital role in predicting the volume of hydrograph (Renata, 1997). Initially, the soil transmissivity was determined as a product of surface saturated conductivity and soil depth. However,  $T_0$  values determined as such underpredicted the hydrographs greatly. Therefore,  $T_0$  was calibrated manually to analyze the variation to reproduce observed peaks. The manual calibration showed that  $T_0$  should be  $0.3 \leq T_0 \leq 0.5 \text{ m}^2/\text{day}$ .

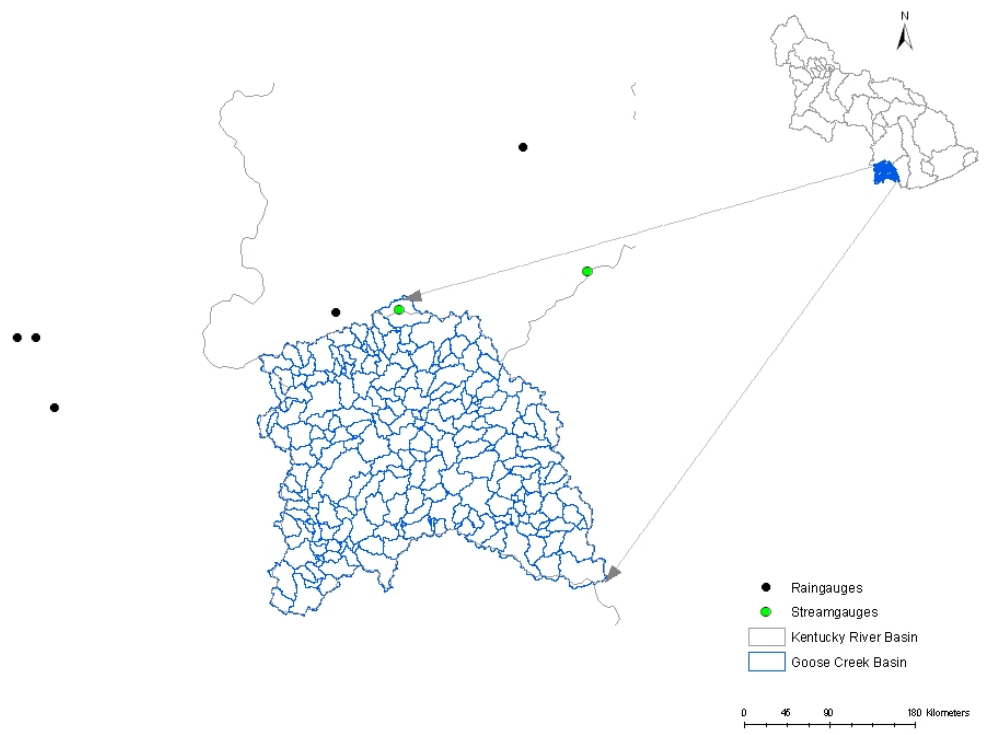
During dry periods, the model overestimated streamflow as can be seen in Figures 3-5 through 3-8. The TOPMODEL has been reported to perform well under wet conditions and with shallow soils (Renata, 1997). The topographic indices with large values are located near the stream banks and thus closer to the watershed outlet. Since TOPMODEL works under the principle that the overland flow is in direct proportion with the topographic index, during non-rainfall or low-rainfall periods, the topographic indices with large values always tend to produce some quantity of flow to the channel (Figures 3-5 and 3-6).

Validation of the routing function was performed for the year 2009 with raingauge and radar rainfall estimates. The simulated hydrographs are shown in Figure 3-10. The average rainfall measured by the gauge was one-fourth of the annual average rainfall whereas study area received half the annual average rainfall as estimated by the NEXRAD. The proposed routing function was able to reproduce observed hydrographs in direct proportion to the rainfall input to the model. The prediction of the model is in agreement with the observed streamflow record. However, Figure 3-10 reflects the effect of difference of rainfall in flow simulation. The peak flows are simulated with close agreement with observed streamflow with radar rainfall as opposed to that of raingauge measurements. The peak volume bias for the radar simulated flow ranged from 9% to 60% whereas the raingauge simulated flow ranged from 40% to 68%. The

volumetric bias determined for the validation dataset showed that for radar rainfall, the volume of the hydrograph is reproduced with better accuracy. The raingauge measurements significantly underpredicted volume of hydrograph for the selected runoff events.

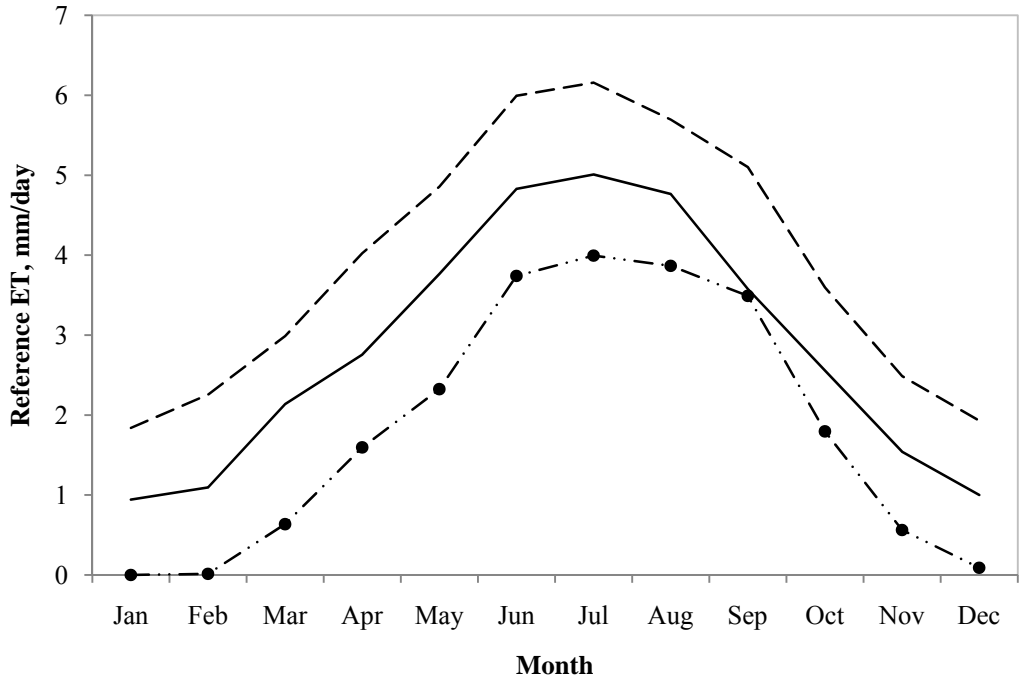
### 3.6. Conclusions

The applicability of the routing procedure developed based on streamflow measurements from gauged basins was analyzed in this paper. The routing procedure was incorporated in TOPMODEL and tested for a subwatershed in Kentucky River Basin, Kentucky, USA. Observed streamflow values upstream and downstream of a watershed were collected from United States Geological Survey (USGS) and processed to develop an Average Transfer Function (ATF). For any ungauged watershed, using only drainage areas, the ATF can be obtained. The inflow was then multiplied in time domain with the ATF to produce outflow at the watershed outlet. The ATF value greater than unity was portioned to account for storage in a channel on any given day. The proposed method performed with a high coefficient of efficiency. The hydrographs showed good agreement between observed and simulated flow. The discrepancies in simulated hydrographs are mainly affected by the distribution of rainfall input. Since the proposed method assumes, implicitly, uniformity in distribution of precipitation, when the rainfall occurs only in an upstream portion of the watershed the peak and baseflow were underpredicted significantly; whereas, when the precipitation occurs in close vicinity of watershed outlet, the peak flow is predicted accurately. When the precipitation is large enough to be distributed uniformly across the watershed, both peak and baseflow are predicted with high coefficient of efficiency. This study concludes that using observed streamflow downstream along with watershed characteristics can facilitate a physically reasonable flow prediction technique in ungauged basins. The proposed method can be used with reduced computation time and resources and the ATF can be updated on a real-time basis for hydrologic modeling of ungauged basins.



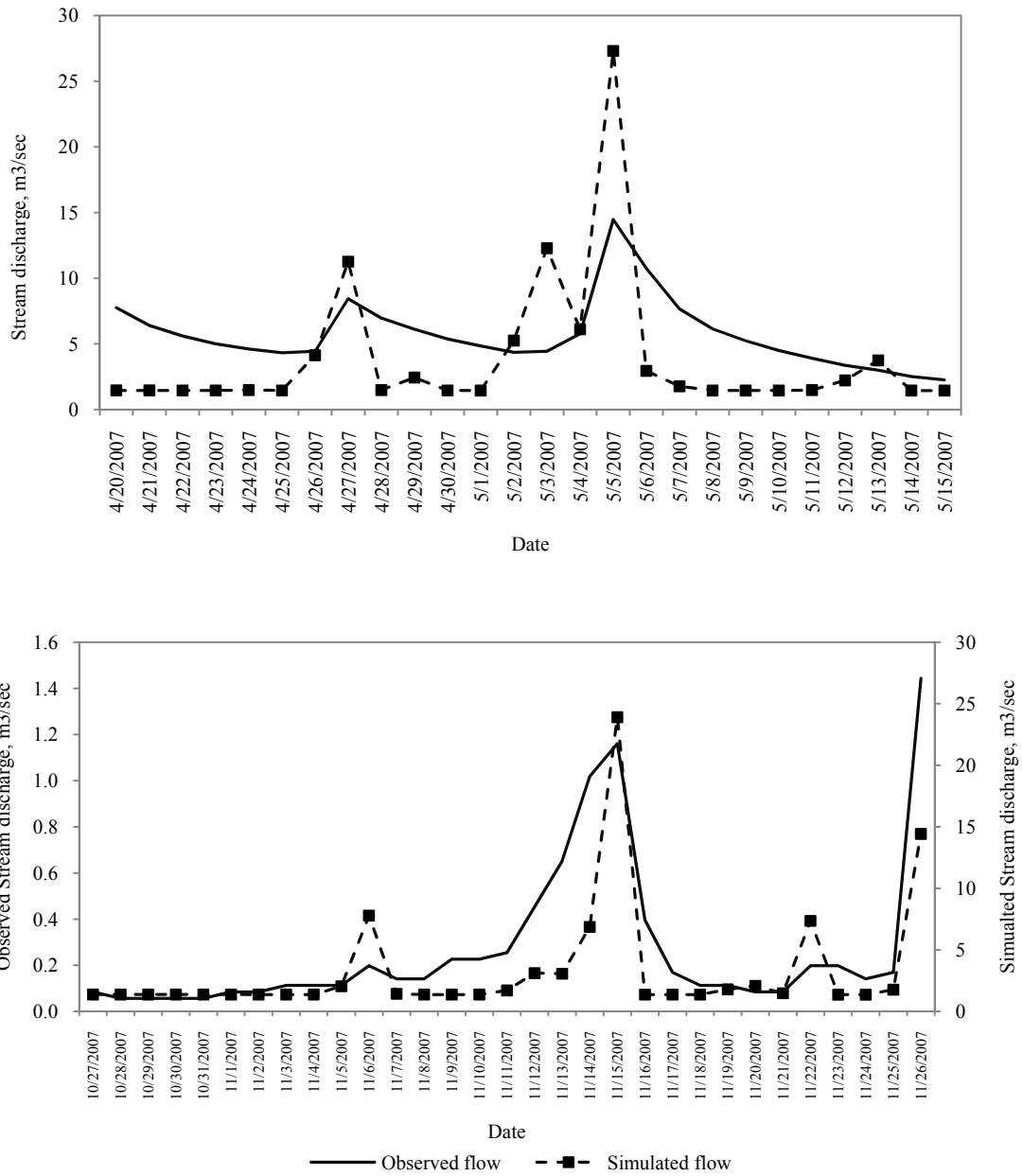
**Figure 3-1. Goose Creek basin and surface water monitoring stations**





— Measured Reference PET    - - - PET by Blaney-Criddle    -●- PET by Thornthwaite

**Figure 3-2. Measured and estimated Potential Evapotranspiration**



**Figure 3-3. Simulated hydrographs using proposed routing procedure without storage component**

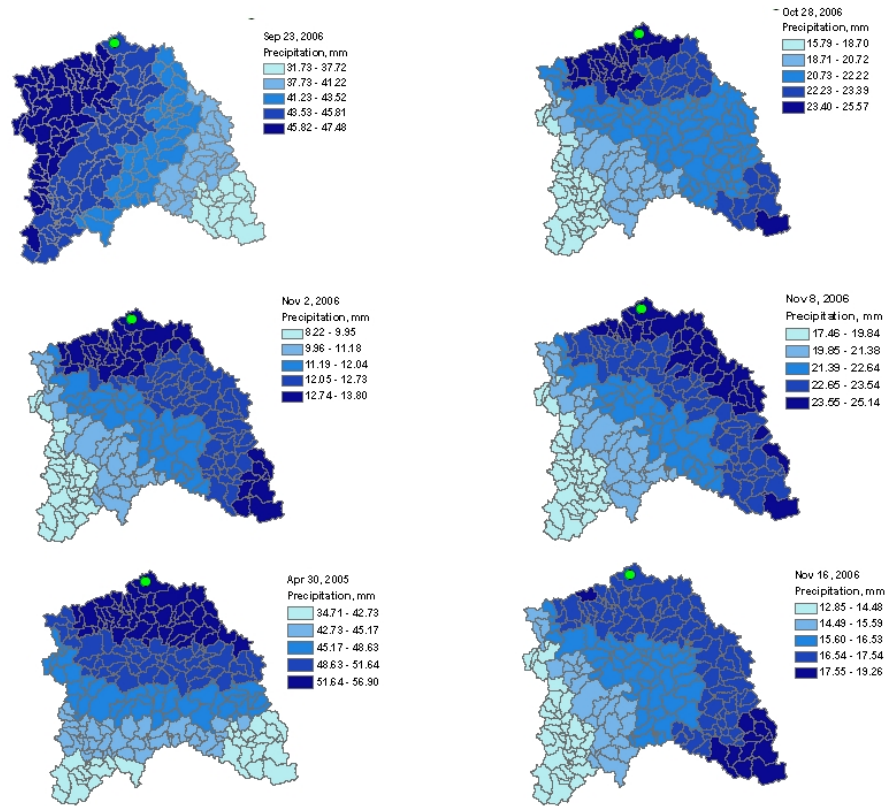


Figure 3-4. Precipitation distribution across the watershed for 2005 and 2006 runoff events

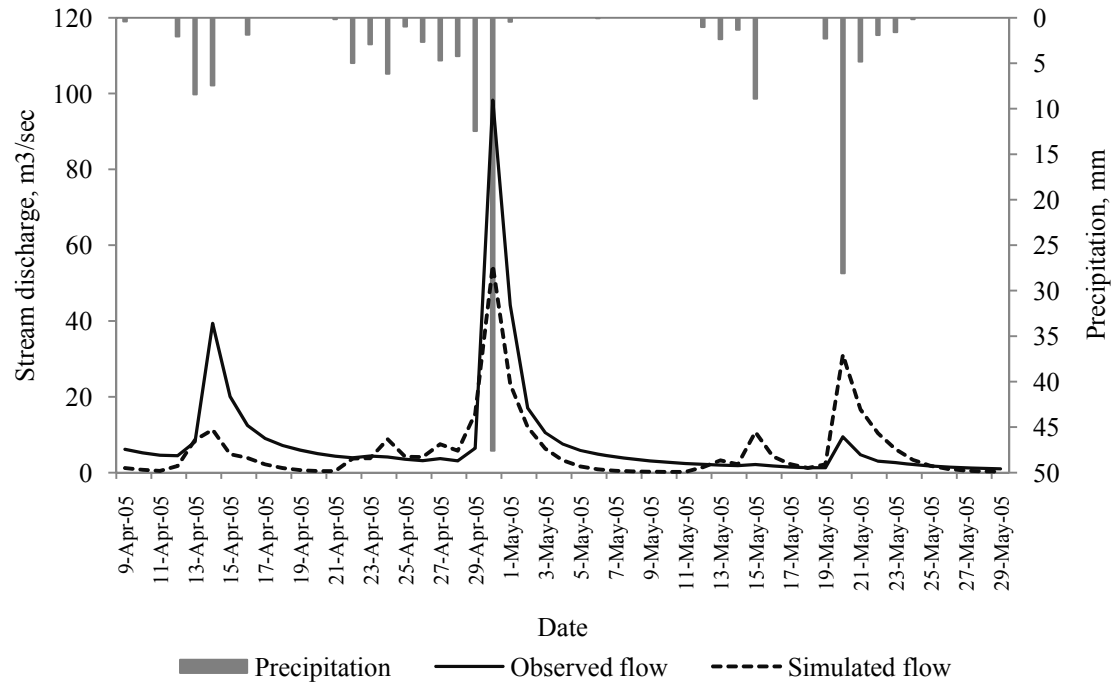


Figure 3-5. 2005 runoff events

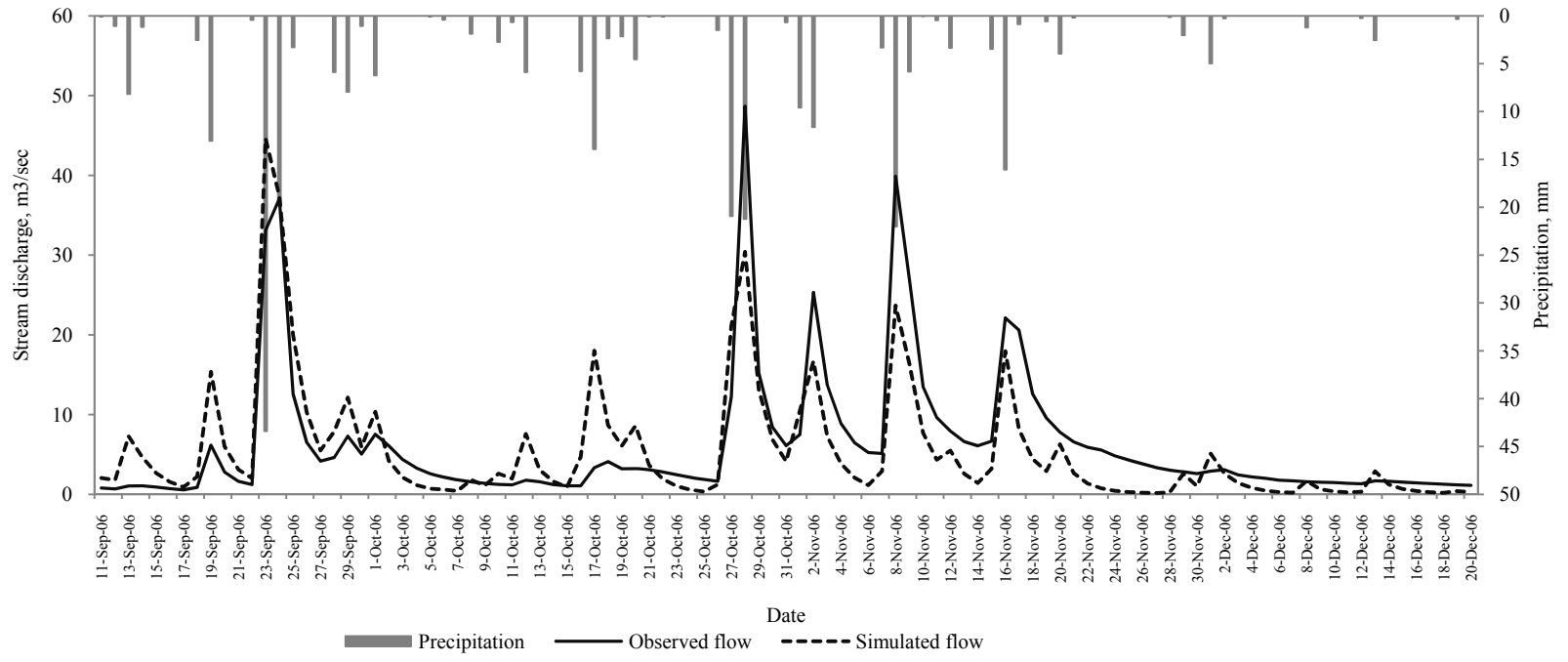


Figure 3-6. 2006 runoff events

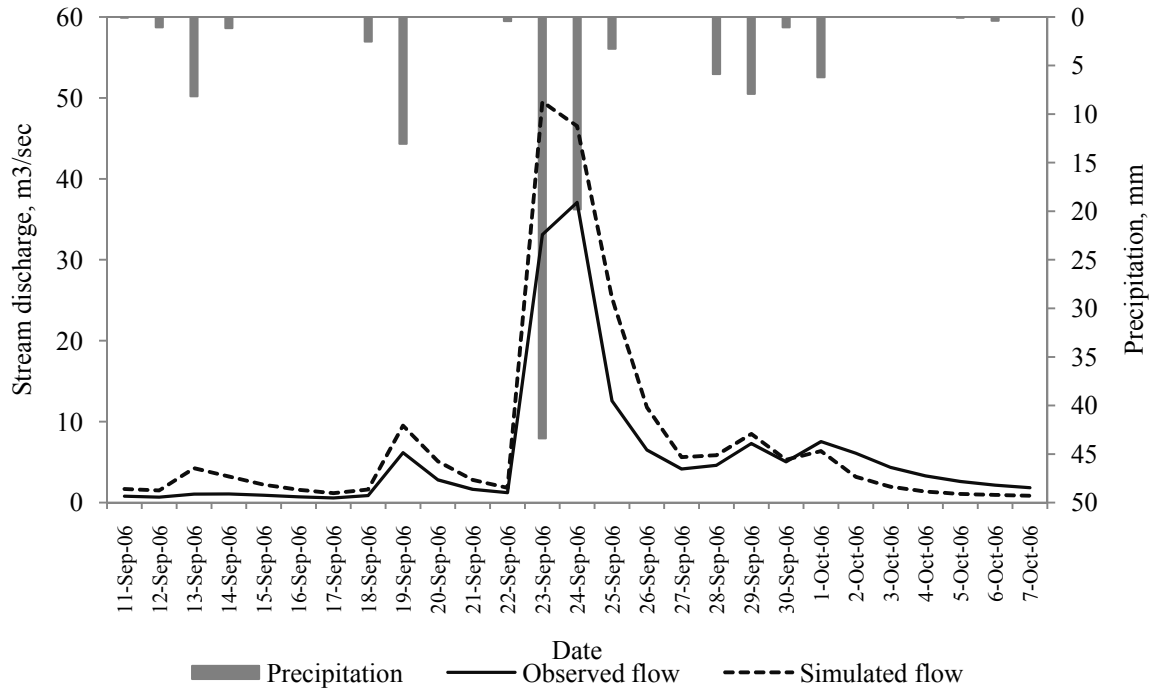
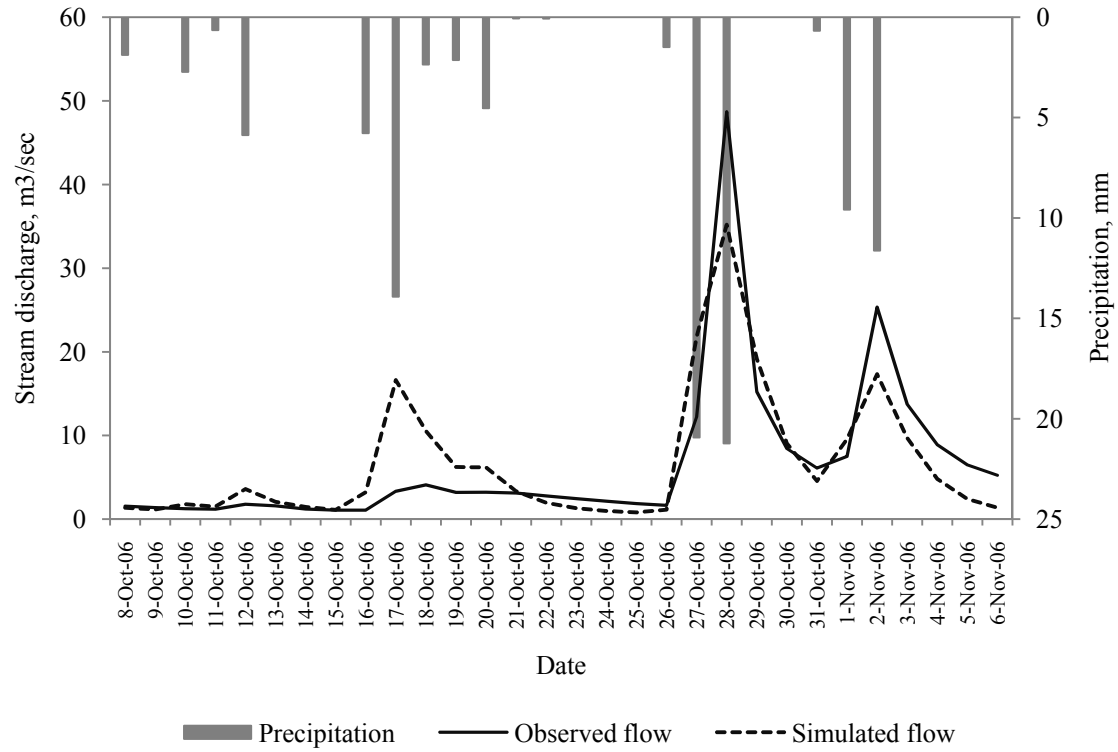


Figure 3-7. 2006 runoff events



**Figure 3-8. 2006 runoff events**

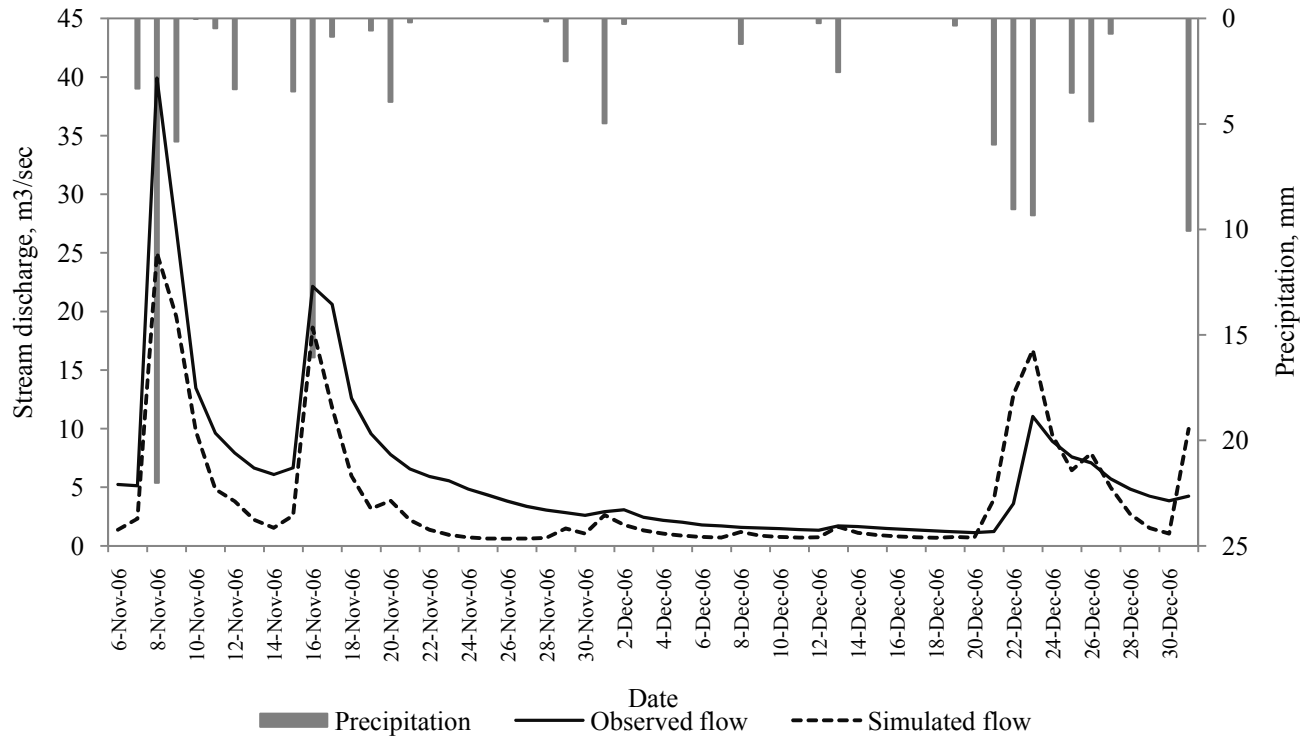
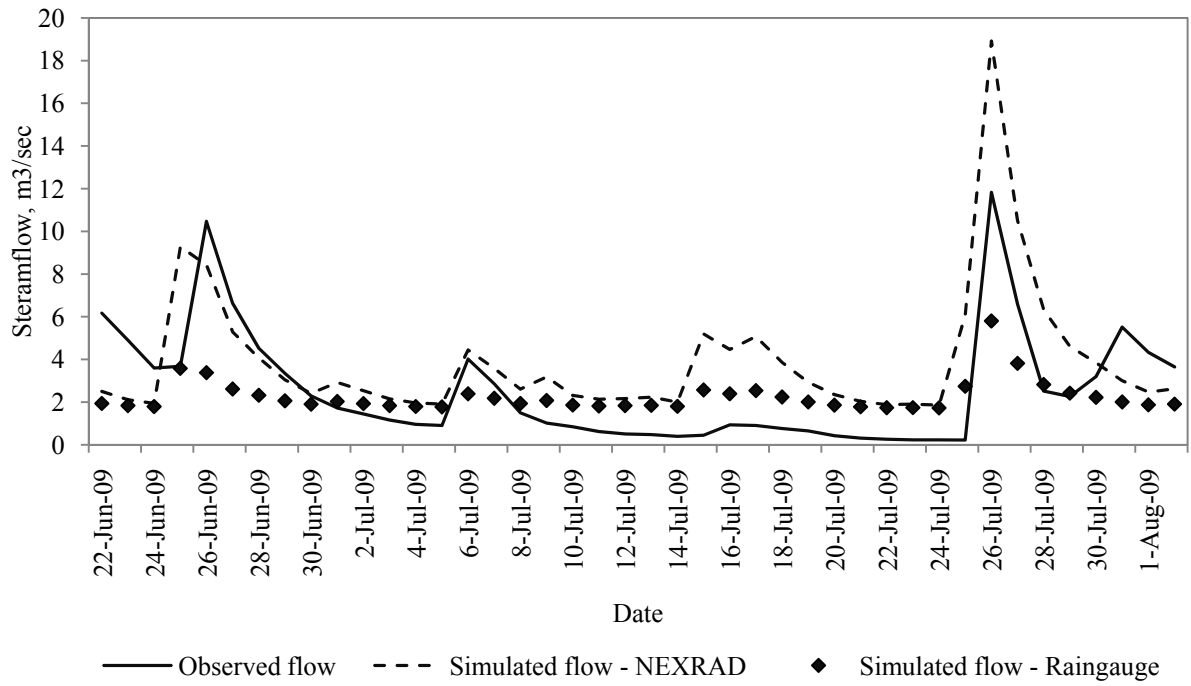


Figure 3-9. 2006 runoff events





**Figure 3-10. Validation runoff event with NEXRAD and raingauge measured precipitation**

**Table 3-1. Calibrated parameters**

<b>VarM</b>	<b>Ksat</b>	<b>Field Capacity</b>	<b>Initial Root Zone Deficit</b>	<b>Initial Discharge</b>
0.224	Reduced by 44%	0.02	0.00001	0.0003

**Table 3-2. Coefficient of Efficiency for selected runoff events**

<b>Event From</b>	<b>Event To</b>	<b>COE</b>
9-Apr-05	29-May-05	0.66
7-Jan-06	31-Dec-06	0.71
7-Jan-06	14-Feb-06	0.75
11-Sep-06	10-Oct-06	0.80
8-Oct-06	6-Nov-06	0.75
7-Nov-06	31-Dec-06	0.67

## 4. A GIS framework for publication of streamflow for Goose Creek in Kentucky River Basin

### 4.1. Abstract

The objective of this research paper is to evaluate a back-calibration method proposed for flow prediction in ungauged first-order streams, located in National Hydrography Dataset, in Goose Creek, Kentucky River Basin, USA. A flow routing function, called as average transfer function, derived from observed streamflow records was applied in the back-calibration procedure. Inside the back-calibration method, the inflow from the subbasins was calculated as a proportion of observed streamflow before being routed to the watershed outlet. The weighting parameter or the proportion was obtained based on the ratio of downstream and upstream observed streamflow records. TOPMODEL was used for simulation of hydrologic components of the subbasins. The inflow to a stream segment simulated by the TOPMODEL was calibrated against the pseudo observed flow, estimated by the back-calibration method, using the Shuffled Complex Evolution algorithm of University of Arizona. The outflow from the calibrated method was made available via ArcIMS website as an outcome of the project “Flow prediction in ungauged basins of Kentucky River Basin” conducted in University of Kentucky. The results from this study concluded that the proposed flow prediction method can be applied successfully where there is at least one stream segment is gauged and monitored.

Keywords: Flow prediction, Ungauged basins, back-calibration, TOPMODEL, ArcIMS

#### 4.2. Introduction

Flow prediction in ungauged basins can be achieved by taking advantage of the information that pertains to the estimation of hydrologic components. The “information” can range from the watershed or channel characteristics to the stream flow properties. The basin characteristics that are represented by soil type, stream density and rainfall data were used to identify watersheds similar to gauged basins, so that the flow generation and transforming properties of the ungauged basins can be determined (Acreman and Sinclair, 1986). However, Rao and Srinivas (2006b) in their study about the clustering of homogeneous watersheds said that the increasing flood response characteristics of the catchment could improve the identification of homogeneous watersheds for flow estimation of ungauged basins. The watershed and flow characteristics have been implemented in hydrologic modeling in such a way that the overland inflow to the main channel is estimated using these watershed and flow properties. To optimize the performance of the hydrologic model, the overland flow is, in practice, calibrated against the streamflow measurements at the watershed outlet.

The calibration algorithms can be either local or global. The local search methods such as simplex method, Rosenbrocks method, Newton-Raphson method cannot validate the presence of multi-local optima (Duan et al., 1992) and they tend to be trapped in the cluster of local minima (Kuczera, 1997). Therefore, the global optimum search algorithms such as Genetic algorithm (GA), multistart simplex procedure, shuffled complex algorithm (SCA-UA) were developed to eliminate the possibility of being caught in the local minima. Additionally, the local search algorithms need intensive computer resources in terms of number of objective function (OF) evaluation and number of local optima.

The global search methods also show their own limitations typically by the computer resources required, size and shape of search space, number of OF evaluations needed. Franchini et al., (1998) showed that SCA-UA method proved its superiority by converging with a minimum number of OF evaluations within the given parameter space compared to the GA and Pattern Search methods. The multistart simplex procedure could produce a highly effective model calibration but will reduce the efficiency from requiring large number of OF evaluations for convergence because of its inability to have more than one complex as compared to SCA-UA method. The SCA-UA needed only one-third of OF evaluations of the multistart simplex method in the study (Duan et al., 1992).

For calibration, the flow “generation” characteristics are calibrated with parameters that are pertaining to the watershed’s topography, soil physical properties, landuse coverage, and number of model coefficients. In terms of calibrating the flow “transferring” characteristic, which

mainly occurs along the channel, most generally the attenuation coefficients such as weighting factor and storage constant for travel time in Muskingum routing method are optimized. The limitation in this method of calibration is that a new parameter is introduced every time a new hydrologic component is included in the parameterization.

The user should have prior knowledge of the stream and watershed in order to initialize the range of parameters to be input to the calibration procedure. At this point, taking advantage of the information about the watershed, stored in the observed streamflow will be computationally time saving and efficient because measured streamflow gathers and secures information about catchment attributes, water balance components and transformation of flow to the watershed outlet. The streamflow measured upstream and downstream of a river segment reflect the transformation of inflow to the catchment outlet.

The transforming function is called the unit hydrograph, impulse response function, or step response function in hydrologic routing literature. The transfer function can be obtained from the flow measurements that are input to the outlet and released from that outlet. This function replaces the unit hydrograph in such a manner that it can be derived in time domain as a simple ratio of outflow to inflow. The transfer function can then be averaged over the number of days of flow records it was derived from. The averaging results in the prediction of average outflow given an inflow to the stream segment.

It can be concluded that if one were interested in knowing the streamflow conditions on a stream network, instead of performing simulation across the subbasins, it would be most efficient to simulate the flow using the gauged stream segment. By this method, the flow simulation from an upstream basin will be performed against the “pseudo” observed flow. The proportion of flow or the “pseudo” observed flow that is released from a subbasin is assumed to be in relationship with the transfer function that facilitates the travelling and attenuation of inflow along the channel to the watershed outlet.

The derivation of average transfer function is provided in the Part 1: Development of flow transferring characteristics from streamflow records. Implementation of average transfer function obtained from the observed flow is provided in Part 2: Flow routing using transfer function derived from streamflow records. The inflow simulated using the hydrologic model TOPMODEL and the flow simulated was routed to the watershed outlet using the transfer function. In this section of the research, the back-calibration procedure is proposed in order to estimate flow in any stream segment given at least one monitored stream. Therefore, the objectives of this paper are:

- Implementation of efficient calibration procedure, back-calibration, for flow routing along ungauged streams
- Development of a GIS system for validation of the back-calibration method to publish stream flow on the internet

#### 4.3. Materials and Methods

##### 4.3.1. Description of the study area

Goose Creek, a subbasin of South Fork Kentucky in Kentucky River Basin (Figure 3-1) with no karst was selected to evaluate the performance of the proposed method. Goose Creek drains portions of Clay and Knox counties. The creek drains 417 km<sup>2</sup> that runs for 38 km with an annual average discharge of 7 m<sup>3</sup>/sec and average annual peak of 872 m<sup>3</sup>/sec. The dominant landuse of the selected basin is forest. South Fork Kentucky watersheds are in hilly terrain with rapid surface runoff and moderate groundwater drainage with few or no sink holes and springs. The major portion of the area is underlain by shales and sandstones. (Kentucky Water Resources Research Institute, 2002).

##### 4.3.2 Dataset preparation

In order to enable the auto-back-calibration method for the stream segments in the study area, the input data that required are:

- Automatic retrieval of
  - Precipitation
  - Streamflow from a monitored reach segment
  - Monthly temperature

The precipitation and temperature data were downloaded from Agricultural Weather station of University of Kentucky (Agricultural Weather Center, 2010). The streamflow from a monitored reach segment in Goose Creek was selected and facilitated the automatic retrieval from the United States Geological Survey (2010) (Appendix G).

The inflow to the stream segment was estimated using the hydrologic model TOPMODEL. The inputs to the model are topographic information, soil properties, routing information of the first-order streams in the National Hydrography Dataset. The topographic information was derived from Digital Elevation Models that are processed inside ArcGIS-VB.NET application package. The soil properties were obtained from Soil Survey Geographic (SSURGO) Database from Natural Resources Conservation Services (NRCS, 2009). The processing of soil data was also performed in ArcGIS-VB.NET application. The database structure for the topographic, soil, climate and routing information is shown in Figure 4-2. The output from the model was written to a text file, from which it was input to the GIS environment.

#### 4.3.3. Back-calibration method

After the simulated flow on any given day is written to a text file for subbasins, the file was input into the calibration module where it was processed to initialize the object function criteria. The next step in this process was to calculate “pseudo” observed flow by the following equation:

$$Q_{in} = \frac{Q_{out}}{ATF} \quad (1)$$

Where,

$ATF$  - Average transfer function defined by,

$$ATF = K * \left\{ \frac{\text{Downstream area}}{\text{Upstream area}} \right\} \quad (2)$$

$K$ - Regression coefficient for the study basin

$Q_{in}$ - Pseudo observed streamflow for subbasin

$Q_{out}$ - Observed streamflow from USGS database

Once the “pseudo” observed flow was determined from equation (1), the auto-calibration method - SCE-UA, was initiated to optimize the simulated inflow for the current subbasin in loop against the calculated pseudo flow. The method was run for all subbasins in the study area and finally the flow was routed along the main channel to the watershed outlet using,

$$Q_{out} = ATF * Q_{in} \quad (3)$$

Since the equation does not account for storage, the routing procedure was modified such that storage in current day was calculated as a proportion of ATF to the stream. The proportion, by which the storage is calculated, was defined based on the fundamental principles of unit hydrograph i.e., the unit hydrograph values can only range between 0.0 and 1.1. Any watershed that has a unit hydrograph value, in this study referred to as ATF, that exceeds 1.0, will store water according to the explanation below:

*For a channel, if the ATF is estimated to be 1.1x, only 1x inflow is output to the next channel. The remaining 0.1x flow is designated as storage in the channel. This*



*procedure was adopted based on the principle of conservation of mass i.e., the output equals to the input and change in storage (See Appendix I for the routing method in VB.NET)*

The processes in back-calibration and estimation of flow at watershed outlet is described in the flow chart given in Figure 4-2.

#### 4.4. Results and Discussions

##### 4.4.1. Back-calibration

The back-calibration was performed to determine outflow at each tributary of Goose creek and the method was validated by analyzing two subbasins (subbasin1 and subbasin2). The resulting hydrographs (Figures 4-3 and 4-4) show that the outflow hydrographs are reproduced with good agreement with observed hydrographs. The outflow hydrographs are only a proportion of observed flow, in which the proportion is determined by the ratio of drainage areas (Equations (1) and (2)). The discrepancies in the low flow regions are forced to look larger than they are because of the range of scales between the X and Y axes. To avoid the discrepancy, the stream flow should be simulated at the subbasin where the USGS gauge is located.

The TOPMODEL generated overland flow was also analyzed against the “pseudo” observed flow as determined from Equation (1). Figures 4-5 and 4-6 show uncalibrated and calibrated hydrographs against the “pseudo” observed flow for a headwater stream in Goose creek for years 2005 and 2006. The resulting flow shows that the back-calibration procedure was able to reproduce both the peak and base flow with good accuracy whereas the flows were underpredicted before back-calibration. The processing time for the back-calibration was reduced compared to the processing time of ordinary calibration thereby increasing the efficacy of the proposed calibration method for an automated flow prediction technique.

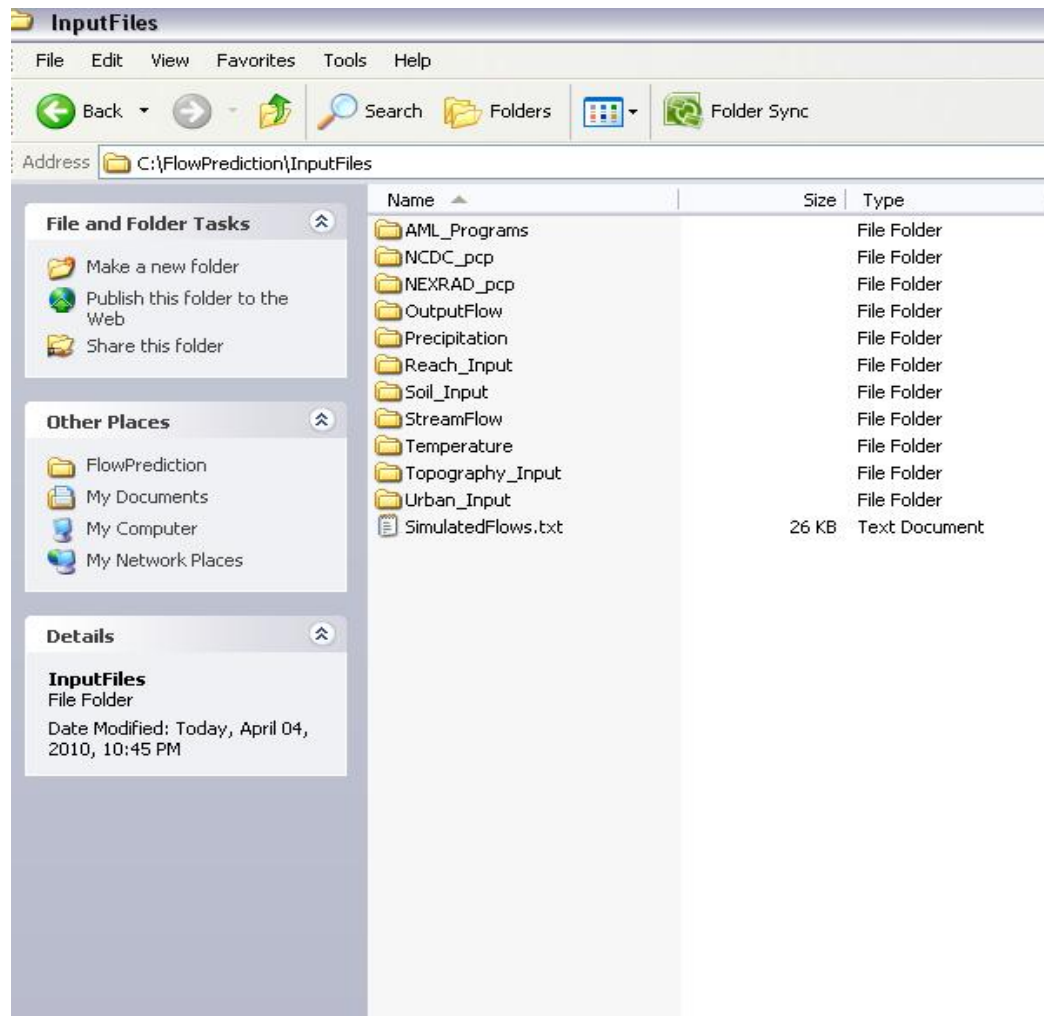
##### 4.4.2. Publication of streamflow value on the Internet using ArcIMS

The outflow from the back-calibration was written to a text file and the text file was exported to GIS environment as a database file format. To be updated in the ArcIMS website, the exported database file was joined to the subbasins with the NHD subbasin id as the common field (Appendix H). The resulting ArcIMS output can be seen in Figure 4-7. When the user clicks on a subbasin, the ArcIMS produces a table of subbasin id, the area, the date, and streamflow value (in Cubic ft/sec). The website will also be upgraded to produce historic hydrographs for each of the subbasins in Goose creek.

#### 4.5. Conclusions

An automated streamflow prediction in first-order streams, delineated by NHDPlus watersheds, in Goose Creek, Kentucky River Basin, was developed in this study using a method

called back-calibration. The overland flow from each of the subbasins were generated using TOPMODEL. The routing of the flow from the mouth of the subbasins to the outlet of the Goose Creek was carried out using the average transfer function (ATF) that was determined from the historic streamflow records. The TOPMODEL simulated flows were calibrated against the “pseudo” observed flows calculated at the subbasin outlets rather than the main channel outlet. The “pseudo” observed flows were derived as the ratio of measured streamflow records from a streamgauge and the ATF. The ATF was calculated as the ratio of drainage areas of subbasins and Goose Creek. Therefore, the “pseudo” observed flows can be considered as the portion of streamflow that contributes to the flow at the outlet of Goose Creek. The resulting hydrographs showed that the back-calibration method can be adapted to predict flow at the outlet of first-order streams. The results are published on the Internet using ArcGIS-ArcIMS to provide a ready-to-access streamflow database.



**Figure 4-1. Database structure for Flow Prediction algorithm**

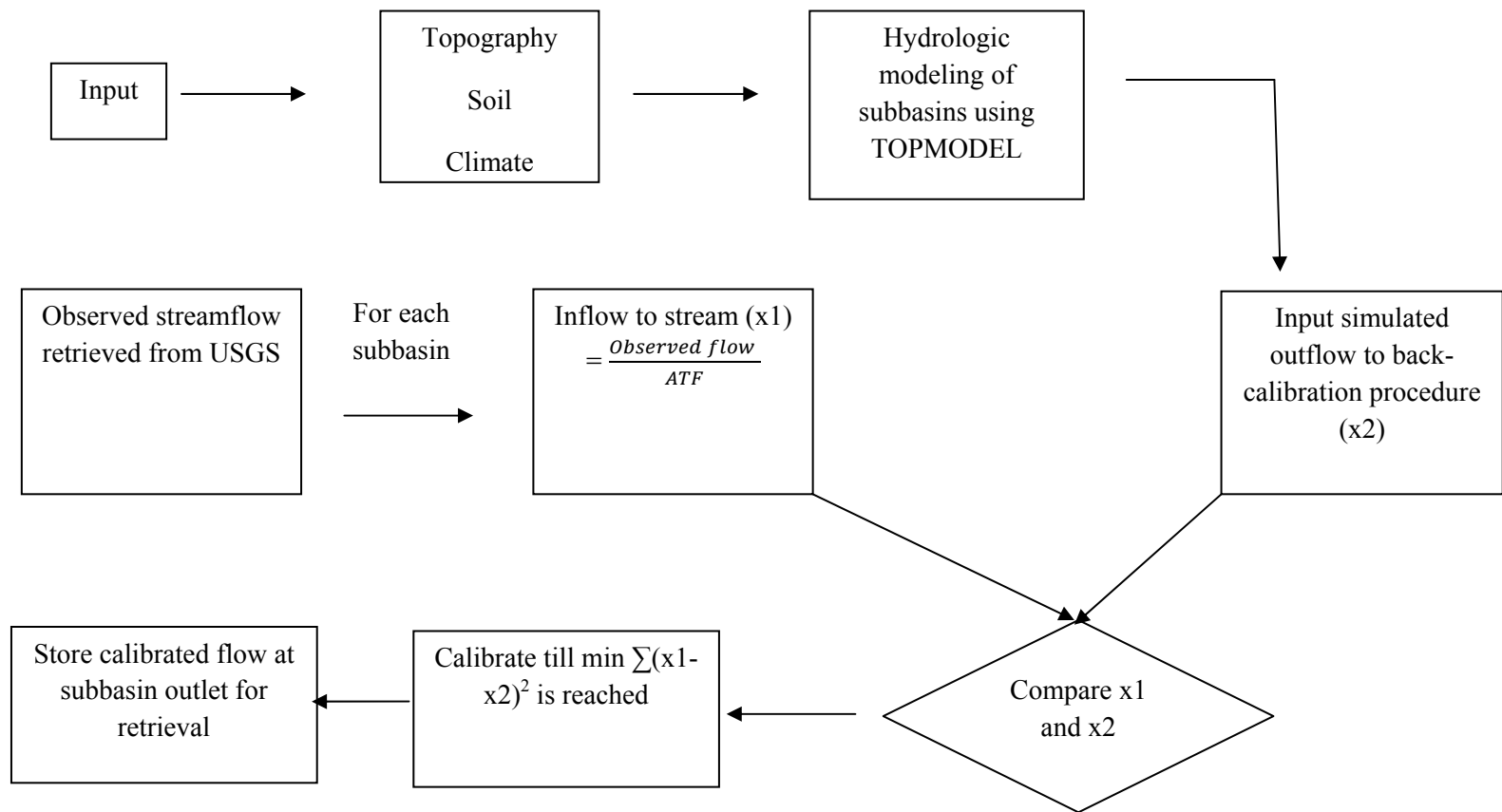
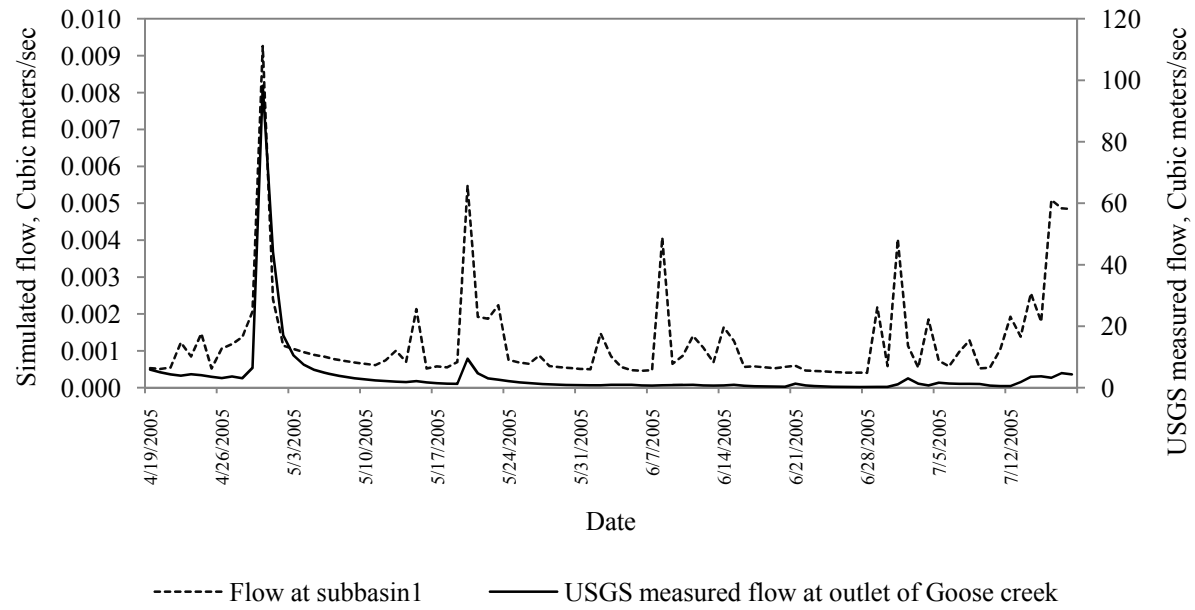
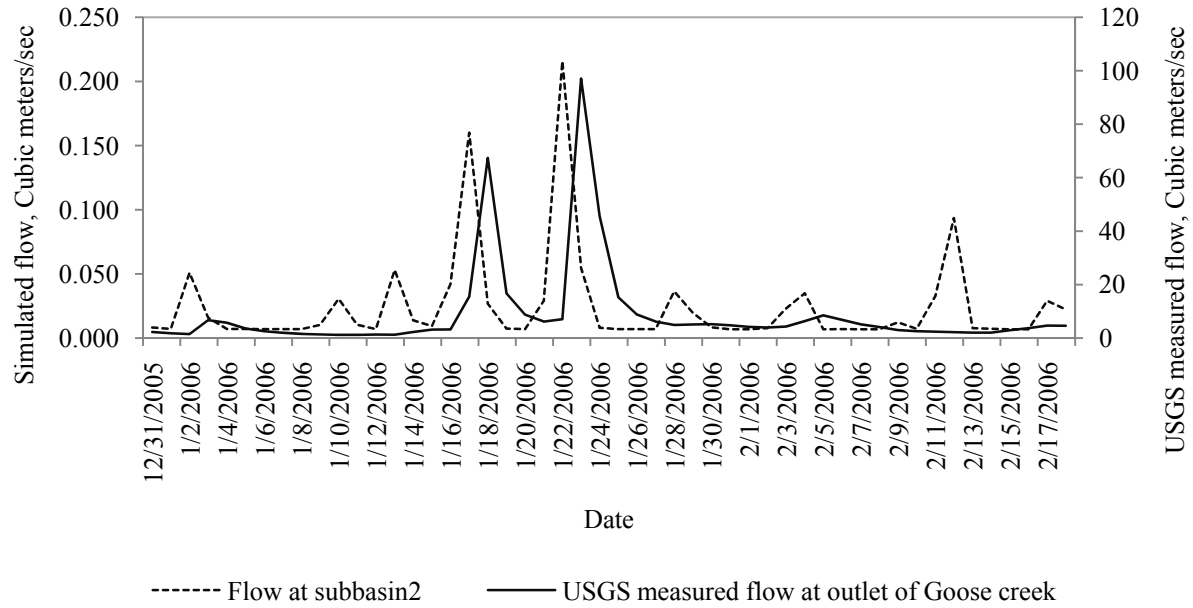


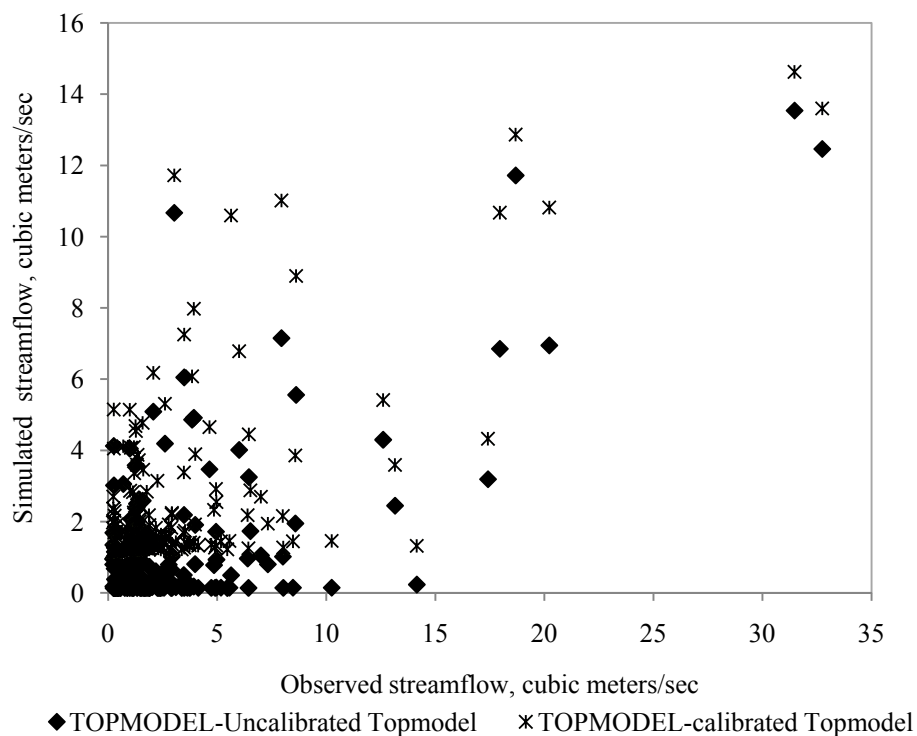
Figure 4-2. Flow chart of the back-calibration method



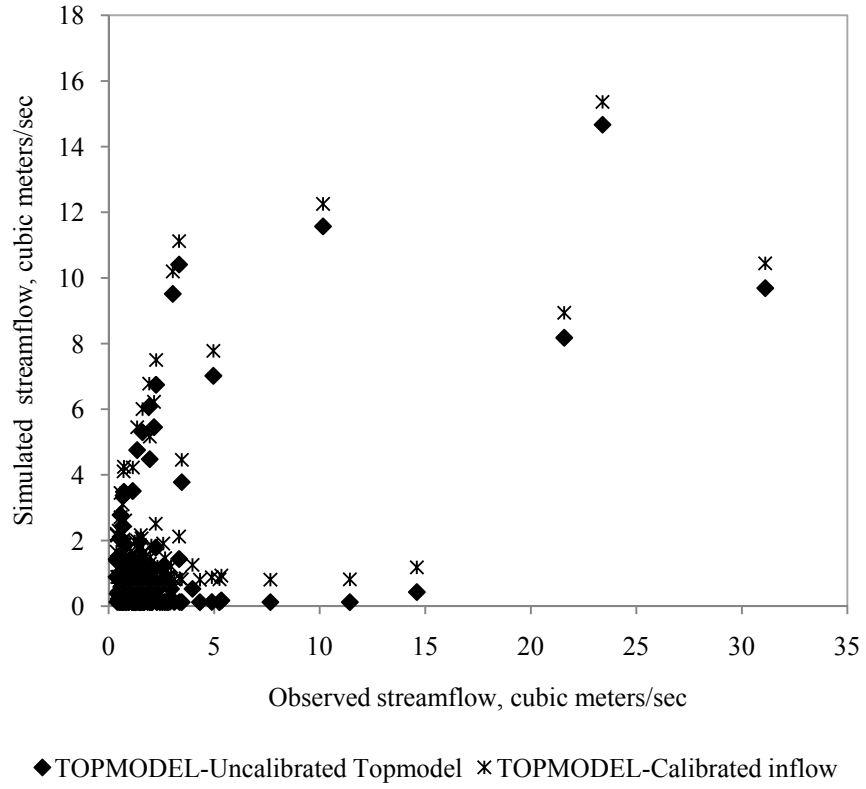
**Figure 4-3. Validation of back-calibration algorithm for subbasin1**



**Figure 4-4. Validation of back-calibration algorithm for subbasin2**



**Figure 4-5. Validation of back-calibration algorithm with calibrated and uncalibrated simulated flow - Event 1**



**Figure 4-6. Validation of back-calibration algorithm with calibrated and uncalibrated simulated flow- Event 2**



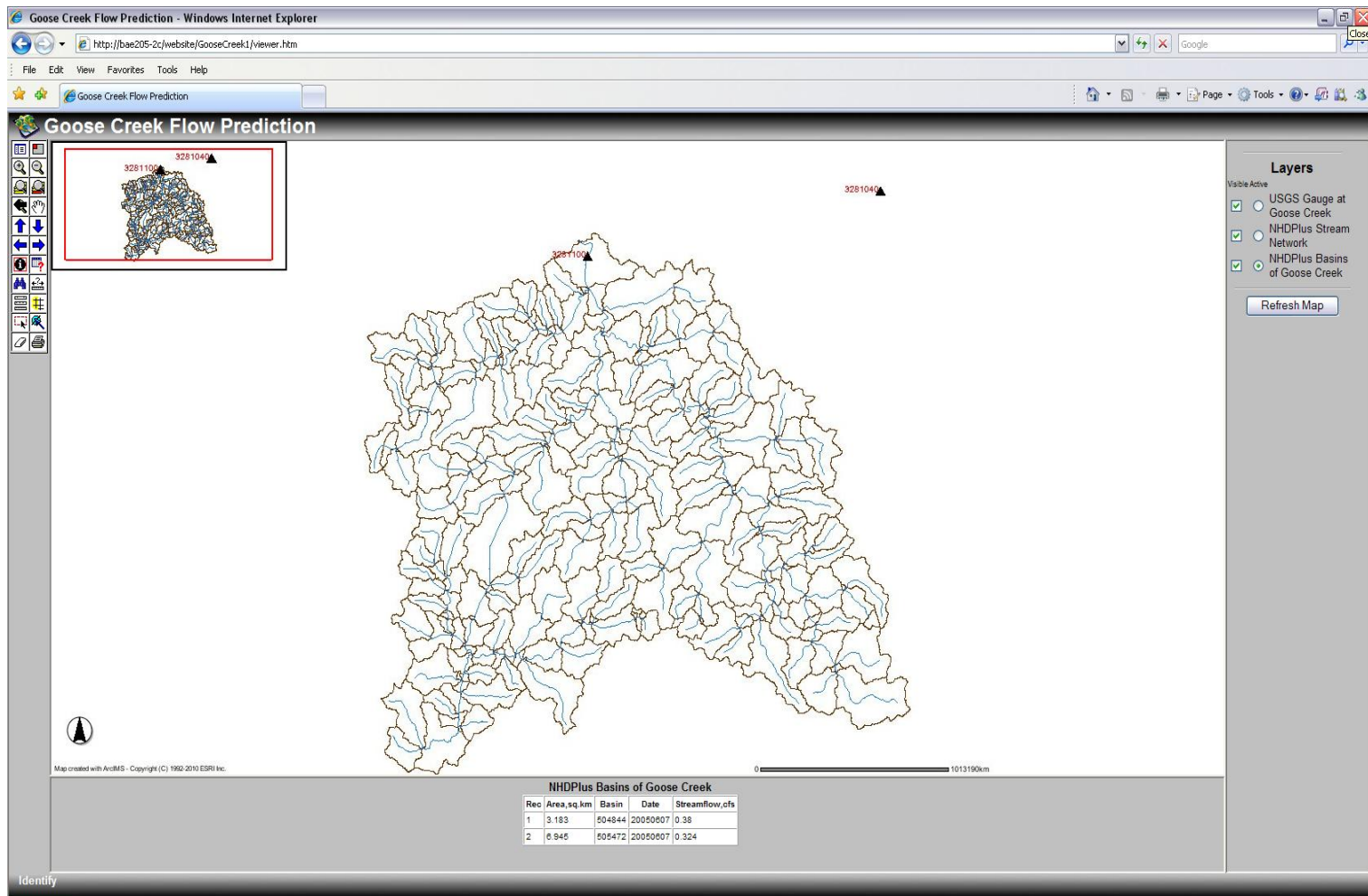


Figure 4-7. Result from the automated flow prediction method

## 5. General observations and conclusions of the results of the study

### 5.1. Derivation of the flow transferring function based on observed streamflow records

An average transfer function (ATF) was derived from historic flow measurements archived by the United States Geological Survey (USGS). The ATF was derived under the time-invariant, linear watershed systems so that it can be applied to determine outflow for any duration of rainfall event. The ATF was extrapolated to ungauged basins in Kentucky River Basin (KRB) using multiple regression model. The analysis of calculated streamflow across several sub watersheds in KRB showed that the ATF can be a useful tool in flow prediction in ungauged basins. The proposed transfer function performed better during wet periods than during dry or winter period. The karst hydrology was significantly overpredicted by the proposed transfer function due to its simplicity and the effects of karst features are averaged across the drainage areas.

### 5.2. Development of flow routing procedure using Average Transfer Function derived from observed flow records

The transfer function was used to determine flow at subbasin outlets of Goose Creek in Kentucky River Basin. A flow routing procedure was developed based on the ATF and was applied to overland hydrographs simulated using TOPMODEL. The results from this section of the research showed that the flow routing performed well to reproduce observed hydrographs. The TOPMODEL simulated flows agreed with the observed hydrographs for events with uniformly distributed rainfall. Therefore, more the uniformity in the distribution of rainfall, the higher the accuracy of the predicted flows. The auto-calibration method was adapted from the Shuffled Complex Evolution algorithm developed by University of Arizona (SCE-UA). The efficiency of the calibration method was validated with the large coefficient of efficiency values for selected runoff events.

### 5.3. Development of back-calibration method for automated flow prediction algorithm

A back-calibration procedure in place of the auto-calibration method was developed for flow prediction in ungauged basins. The back-calibration was developed with the help of observed streamflow records. In this method, each subbasin in the Goose Creek was estimated to have its own “pseudo” observed hydrographs. The “pseudo” observed flows were calculated as a proportion of USGS measured streamflow by weighting it with the average transfer function. The efficacy of the model to automate flow prediction was increased by back-calibration method since

only one subbasin was processed at each time the calibration was performed. The predicted streamflow were exported into ArcGIS-ArcIMS environment to be published on the Internet.

## Appendices

### Appendix A - Program to process NEXRAD dataset

Disclaimer: The basic structure of ArcGIS programs given in this section are obtained from ESRI support, ESRI forum and VB.NET forums and are specified in corresponding program snippets. The routing program was developed by the student and remaining programs are adopted to suit the data processing for this dissertation.

**Step 1:**

++++ Batch process to convert XMRG files to ASCII grid using "NCDC Java NEXRAD tool" ++++++

AscGrd.bat

@echo off

set start=1

set end=31

:loop

if %start% GTR %end% goto stop

cd D:\data\nexrad\_subhourly\wct-2.3.4\wct\

echo ----- Processing day %start% -----

IF %start% LSS 10 (

call mke\_fold9.bat

) ELSE (

call mke\_fold.bat

)

set /a start=%start%+1

goto loop

:stop

ECHO Finished writing files

:end

Mke\_fold9.bat

set pathssub=H:\nexrad\nex120%start%\

set pathout=D:\Data\nexrad\_subhourly\2009\nex120%start%\

call wct.bat %pathssub% %pathout% asc D:\data\nexrad\_subhourly\wct-2.3.4\wct\wctBatchconfig.xml

Mkd\_fold.bat

```
set pathssub=H:\nexrad\nex12%start%\
set pathout=D:\Data\nexrad_subhourly\2009\nex12%start%\
/* The following line of code obtained from NCDC NEXRAD website
call wct.bat %pathssub% %pathout% asc D:\data\nexrad_subhourly\wct-2.3.4\wct\wctBatchconfig.xml
Step 2:
```

```
+++++ Program to convert ASCII NEXRAD precipitation to ESRI GRID format ++++++
` Program written by Tony Soeller, NACS, U.C. Irvine
` Program modified by Bagya Palanisamy, BAE, University of Kentucky
```

```
Public Sub ascii2grid(ByVal dd As Integer)
```

```
intyear =2009
```

```
For mm = 1 to 12
```

```
    Select Case mm
        Case Is = 2
            endDay = 28
        Case Is = 4, 6, 9, 11
            endDay = 30
        Case Is = 1, 3, 5, 7, 8, 10, 12
            endDay = 31
    End Select
```

```
For dd = theDay To endDay
```

```
    If dd < 10 Then
        tmpDayz = "0" & dd.ToString
        intDayz = CInt(tmpDayz)
    Else
        tmpDayz = dd
        intDayz = dd
    End If
```

```

End If

If mm < 10 Then
    tmpMonthz = "0" & mm.ToString
    intMonthz = tmpMonthz
Else
    tmpMonthz = mm
    intMonthz = mm
End If

If IO.Directory.GetFiles(OutPath & "tmpRas\").Length > 1 Then
    Call EmptyGrdDir(OutPath & "tmpRas\")
End If

inPath = defPath & "nex" & tmpMonthz & tmpDayz & "\"

'Define the output raster workspace
pRasterWorkspaceOut = GetRasterWorkspace(OutPath & "tmpRas\")

' Get an enumeration of the input raster dataset names
pRasterWorkspace = GetRasterWorkspace(inPath)
pWorkspace = pRasterWorkspace 'QI
pEnumDSName = pWorkspace.DatasetNames(esriDatasetType.esriDTRasterDataset)

intDSCount = 0
pRasterDSName = pEnumDSName.Next

While Not pRasterDSName Is Nothing
    intDSCount = intDSCount + 1
    pRasterDSName = pEnumDSName.Next
End While

' Populate the string array with the names of the rasters in the dataset
ReDim aDSNames(intDSCount)
pEnumDSName.Reset()
intDSIndex = 0
pRasterDSName = pEnumDSName.Next

```

```

While Not pRasterDSName Is Nothing
    intDSIndex = intDSIndex + 1
    aDSNames(intDSIndex) = pRasterDSName.Name
    pRasterDSName = pEnumDSName.Next
End While

Dim jj As Integer
For jj = 1 To intDSIndex
    pRasterDataset = GetRasterDataset(inPath, aDSNames(jj))

    inputFile = inPath & aDSNames(jj)

    s = Split(aDSNames(jj), "_")
    third = s(3)
    Dim ss() As String
    ss = Split(third, ".")
    tstSec = Microsoft.VisualBasic.Mid(ss(0), 5, 4)
    tstThir = Microsoft.VisualBasic.Right(ss(0), 4)
    outGrdName = "p_" & tstSec & tstThir

    inFile = OutPath & "tmpRas\" & "p_" & tstSec & tstThir

    If (System.IO.Directory.Exists(OutPath & inFile) = True) Then
        If (System.IO.Directory.GetFiles(inFile).Length > 0) Then
            Continue For
        End If
    End If

    Dim pWSF As IWorkspaceFactory
    Dim pOutRaster As IRaster
    pWSF = New RasterWorkspaceFactory
    pWorkspace = pWSF.OpenFromFile(OutPath & "tmpRas\", 0)

    ' Create a Spatial operator
    Dim pAlgbOp As IMapAlgebraOp
    pAlgbOp = New RasterMapAlgebraOp

    ' Set output workspace

```



```

    pRasterAnalEnv = pAlgbOp
    pRasterAnalEnv.OutWorkspace = pWorkspace

    ' Perform ASCIIGRID operation
    pOutRaster = pAlgbOp.Execute(outGrdName & " = ASCIIGRID(" & inputFile & ", float)")

Next

Call Sum(OutPath, OutPath & "tmpRas\\"", tstSec)

Call EmptyGrdDir(OutPath & "tmpRas\")

Call EmptyGrdDir("C:\Documents and Settings\bagya\LocalSettings\Temp\")

Console.WriteLine("Emptied the temporary rasters")
Console.WriteLine("Completed nexrad processing for day " & tmpMonthz & tmpDayz)

Next ' End loop for day

Next ' End loop for month

    pOutputRaster = Nothing
    pRasterAnalEnv = Nothing
    pRasterBand = Nothing
    pRasterBandCollection = Nothing
    pRasterBandCollTemp = Nothing

Exit Sub

GC.Collect()

End Sub

Public Function GetRasterWorkspace(ByVal strPath As String) As IRasterWorkspace
    ' Get a raster workspace from the parent directory

    Dim pRasterWorkspaceFactory As IWorkspaceFactory

```

```
pRasterWorkspaceFactory = New RasterWorkspaceFactory
GetRasterWorkspace = pRasterWorkspaceFactory.OpenFromFile(strPath, 0)
pRasterWorkspaceFactory = Nothing
```

End Function

```
Public Function GetRasterDataset(ByVal strPath As String, ByVal pFileName As String) As
IRasterDataset
```

```
    ' Get a raster dataset
```

```
    Dim pRasterWorkspaceFactory As IWorkspaceFactory2
    Dim pRasterWorkspace As IRasterWorkspace
```

```
    pRasterWorkspaceFactory = New RasterWorkspaceFactory
    pRasterWorkspace = pRasterWorkspaceFactory.OpenFromFile(strPath, 0)
    GetRasterDataset = pRasterWorkspace.OpenRasterDataset(pFileName)
    pRasterWorkspaceFactory = Nothing
```

End Function

```
Sub Sum(ByVal strPath As String, ByVal strPathOut As String, ByVal mnt As String)
```

```
    'Create a Spatial operator
    Dim pMathOp As IMathOp
    pMathOp = New RasterMathOps
```

```
    pRasterWorkspaceOut = GetRasterWorkspace(strPath)
```

```
    ' Get an enumeration of the input raster dataset names
    pRasterWorkspace = GetRasterWorkspace(strPathOut)
    pWorkspace = pRasterWorkspace 'QI
    pEnumDSName = pWorkspace.DatasetNames(esriDatasetType.esriDTRasterDataset)
```

```
    intDSCount = 0
    pRasterDSName = pEnumDSName.Next
```

```
    While Not pRasterDSName Is Nothing
```

```

        If (Left(pRasterDSName.Name, 6) <> "p_" & mnt) Then
            Exit While
        End If
        intDSCount = intDSCount + 1
        pRasterDSName = pEnumDSName.Next
    End While

    ' Populate the string array with the names of the rasters in the dataset
    ReDim aDSNames(intDSCount)
    pEnumDSName.Reset()
    intDSIndex = 0
    pRasterDSName = pEnumDSName.Next
    While Not pRasterDSName Is Nothing
        If (Left(pRasterDSName.Name, 6) <> "p_" & mnt) Then
            Exit While
        End If
        intDSIndex = intDSIndex + 1
        aDSNames(intDSIndex) = pRasterDSName.Name
        pRasterDSName = pEnumDSName.Next
    End While

    Dim pAlgbOp As IMapAlgebraOp

    ' Perform processing on each set of groups
    pRasterBandCollection = New Raster

    For i = 1 To intDSIndex ' intDSStart To intDSEnd ' intDSIndex

        pRasterDataset = GetRasterDataset(strPathOut, aDSNames(i))

        ' Checking whether any of the radar grids are null
        pAlgbOp = New RasterMapAlgebraOp
        pAlgbOp.BindRaster(pRasterDataset, "LU1")
        pTmpRas = pAlgbOp.Execute("CON(ISNULL([LU1]), 0, [LU1])")
        pTmpRasterBandColl = pTmpRas ' pRasterDataset ' QI
        pAddRasDS = pTmpRasterBandColl.Item(0).RasterDataset

        pInputRasterBandColl = pAddRasDS
    
```

```

    pRasterBand = pInputRasterBandColl.Item(0)
    pRasterBandCollection.AppendBand(pRasterBand)

    pAlgbOp.UnbindRaster("LU1")
    pRasterDataset = Nothing
    pRasterBand = Nothing
    pInputRasterBandColl = Nothing
    pAlgbOp = Nothing
    pTmpRasterBandColl = Nothing
    pTmpRas = Nothing
    pAddRasDS = Nothing
Next

    pLocalOp = New RasterLocalOp
    pRasterAnalEnv = pLocalOp ' QI
        pRasterAnalEnv.SetExtent(esriRasterEnvSettingEnum.esriRasterEnvMinOf)
    pRasterAnalEnv.SetCellSize(esriRasterEnvSettingEnum.esriRasterEnvMinOf)
    ' Perform cell statistics on the combined raster layers and produce an output raster
    pOutputRaster = pLocalOp.LocalStatistics(pRasterBandCollection,
esriGeoAnalysisStatisticsEnum.esriGeoAnalysisStatsSum)
    pRaster = pOutputRaster
    pRasterBandCollTemp = pRaster

    ' Build the output raster name and check for existence
    strGridOut = "Nex" & mnt
    ' Save the new summation raster to disk
    pDatasetOut = pRasterBandCollTemp.SaveAs(strGridOut, pRasterWorkspaceOut, "GRID")
    ' Dismiss the temporary objects
    pOutputRaster = Nothing
    pRasterBandCollection = Nothing
    pLocalOp = Nothing
    pRasterAnalEnv = Nothing
    pRaster = Nothing
    pRasterBandCollTemp = Nothing
    pDatasetOut = Nothing

End Sub

```

```

Public Sub deleteRaster(ByVal folder As String, ByVal sName As String)
    On Error GoTo errhandler
    Dim pDS As IDataset
    Dim pRD As IRasterDataset
    pRD = GetRasterDataset(folder, sName)

    If (Not pRD Is Nothing) Then
        pDS = pRD
        pDS.Delete()
        Exit Sub
    Else
        Exit Sub
    End If
End Sub

errhandler:
    Debug.Print(Err.Number)
    Debug.Print(Err.Description)
End Sub

Public Sub EmptyGrdDir(ByVal tmpPath As String)
    ' Get an enumeration of the input raster dataset names
    pRasterWorkspace = GetRasterWorkspace(tmpPath)
    pWorkspace = pRasterWorkspace 'QI
    pEnumDSName = pWorkspace.DatasetNames(esriDatasetType.esriDTRasterDataset)

    pRasterDSName = pEnumDSName.Next

    While Not pRasterDSName Is Nothing
        deleteRaster(tmpPath, pRasterDSName.Name)
        pRasterDSName = pEnumDSName.Next
    End While

End Sub

Public Sub emptyAscFiles(ByVal TheMn As String, ByVal TheEnd As Integer)

    Dim ThePath As String

```

```

For dd = 1 To TheEnd

    If TheMn < 10 Then
        tmpMonthz = "0" & TheMn
    Else
        tmpMonthz = TheMn
    End If

    If dd < 10 Then
        tmpDayz = "0" & dd.ToString
        intDayz = CInt(tmpDayz)
    Else
        tmpDayz = dd
        intDayz = dd
    End If
    ThePath = "D:\Data\nexrad_subhourly\2009\nex" & tmpMonthz & tmpDayz & "\"
    Try
        My.Computer.FileSystem.DeleteDirectory(ThePath,
FileIO.DeleteDirectoryOption.DeleteAllContents)
    Catch ex As System.IO.IOException
        Continue For
    End Try
Next
End Sub

```

End Module

**Step 3:**

++++++ Program to reproject the NEXRAD data and to find the zonal mean of NEXRAD precipitation for the study basin ++++++

` Program written by Bagya Palanisamy, BAE, University of Kentucky

Public Sub prj\_nex()

```

For mm = 1 To 12
    If mm = 2 Then
        endDay = 28
    
```

```
ElseIf mm = 4 Then
    endDay = 30
Else
    endDay = 31
End If

For dd = 1 To endDay
    thePath = "d:\data\nexrad_subhourly\"
    If dd < 10 Then
        tmpDayz = "0" & dd.ToString
        intDayz = CInt(tmpDayz)
    Else
        intDayz = dd
    End If

    If mm < 10 Then
        tmpMonthz = "0" & mm.ToString
        intMonthz = tmpMonthz
    Else
        intMonthz = mm
    End If

    Dim theSlash As String
    theSlash = "\"

    rootPath = thePath & 2009 & tmpMonthz & theSlash ' D:\Data\nexrad_subhourly\200902
    basePath = "nex" & tmpMonthz & tmpDayz ' nex0206
    inPath = rootPath & basePath & theSlash ' D:\Data\nexrad_subhourly\200902
    OutPath = inPath & "tmpGrids" & theSlash
    addPath = inPath & "tmp" & tmpDayz & theSlash
    Dim theDate As String = "2009" & tmpMonthz & tmpDayz
    Dim outFPath As String = rootPath & "daily_nex" & theSlash
    Dim fName As String = "s_" & dd
    Dim out1 As String = "nexrad" & tmpMonthz & tmpDayz & ".txt"

    prjRaster(addPath, outFPath, fName, tmpDayz)

    'Get rasterdataset
```

```

Dim tmpHUC, tmpRas As IRasterDataset
Dim nexRas, huc As IGeoDataset
tmpRas = Module1.GetRasterDataset(outFPath, "nexradprj" & tmpDayz)

' Get zone dataset
tmpHUC = GetRasterDataset("D:\Data\InputFiles\gisdata\", "huc1")
huc = tmpHUC

' ===== Convert nexrad precipitation to millimeters

Dim pMapAlgeOp As IMapAlgebraOp
pMapAlgeOp = New RasterMapAlgebraOp
Dim pRasEnv As IRasterAnalysisEnvironment

pRasEnv = pMapAlgeOp
pRasEnv.SetCellSize(esriRasterEnvSettingEnum.esriRasterEnvValue, huc)
pRasEnv.SetExtent(esriRasterEnvSettingEnum.esriRasterEnvValue, huc, huc)

Dim tmp1 As IRaster
pMapAlgeOp.BindRaster(tmpRas, "tmm")
Dim strMath As String
strMath = "( [tmm] / " & 100.0 & " ) " & "*" & " ( " & 25.4 & " ) "
tmp1 = pMapAlgeOp.Execute(strMath)

Dim pRasBandCol As IRasterBandCollection
pRasBandCol = tmp1
Dim outWS As IRasterWorkspace = GetRasterWorkspace(outFPath) ' changed IWorkspace to
irasterworkspace
Dim mmNex As String = "nex" & tmpDayz
If Not outWS.OpenRasterDataset(mmNex) Is Nothing Then
    deleteRaster(outFPath, mmNex)
End If
Dim pRasNex As IGeoDataset = pRasBandCol.SaveAs(mmNex, outWS, "GRID")
nexRas = pRasNex

' =====

zonalPrecip(huc, nexRas, out1)

```



```
        tmp1 = Nothing
        pRasBandCol = Nothing
        outWS = Nothing
        pRasEnv = Nothing
        pMapAlgeOp = Nothing

    Next dd
Next mm
End Sub

Private Sub prjRaster(ByVal inPath As String, ByVal outFilePath As String, ByVal fileName As
String, ByVal dayId As String)
    '****This sub runs an AML script.
    Dim Arc1 As ESRI.Arc
    Dim Results As ESRIutil.Strings
    Results = New ESRIutil.Strings
    Arc1 = New ESRI.Arc
    Dim ArcStatus As Long
    Dim command1 As String
    Dim amlStr As String
    Dim pRasterSet As IRasterDataset
    Dim inGrd As String

    amlStr = "C:\bagya\Projects_NET\prjNexrad.aml"

    ' Get an enumeration of the input raster dataset names
    pRasterSet = Module1.GetRasterDataset(inPath, fileName)
    inGrd = pRasterSet.CompleteName

    command1 = "&run " & amlStr & " " & inGrd & " " & "tmpGrd" & " " & outFilePath & " " & dayId
    ArcStatus = Arc1.Command(command1, Results)

End Sub

++++ This program projects daily NEXRAD data that is generated from subhourly data ++++++
```

Program prjNexrad.aml

```
/* Inputs: Daily NEXRAD Data - The Name: inGrd
/* Output: Projected Nexrad Data - The Name: nexPrj
/* Intermediate output: Projection defined NEXRAD file - The Name: defPrj
```

```
&arg inGrd tmpGrd nexPath dayz
```

```
GRID
```

```
&if [exists %tmpGrd% -grid] &then
```

```
    kill %tmpGrd% all
```

```
setwindow %inGrd%
```

```
setcell %inGrd%
```

```
%tmpGrd% = project(%inGrd%, #, NEAREST, #, #)
```

```
INPUT
```

```
PROJECTION POLAR
```

```
UNITS METERS
```

```
PARAMETERS
```

```
-105 0 0
```

```
60 0 24.5304792
```

```
0 0
```

```
0 0
```

```
OUTPUT
```

```
PROJECTION GEOGRAPHIC
```

```
SPHEROID SPHERE
```

```
UNITS DD
```

```
PARAMETERS
```

```
END
```

```
&s nexPrj = %nexPath%nexradPrj%dayz%
```

```
&if [exists %nexPrj% -grid] &then
```

```
    kill %nexPrj% all
```

```
%nexPrj% = project(%tmpGrd%, #, NEAREST, 4000, #)
```

```
OUTPUT
```

```
PROJECTION STATEPLANE
```

```
FIPSZONE 1601
```

```
DATUM NAD83
```

```
UNITS METERS
```

PARAMETERS

END

Q

Q

exit

&return

**Step 4**

+++++ This program finds the zonal mean of the NEXRAD precipitation and writes it to a text file +++

Program ZonalNEXRAD.aml

/\* Inputs: Daily NEXRAD Data - The Name: inGrd

/\* Output: Text file - The Name: nexPrj

/\* Intermediate output: Projection defined NEXRAD file - The Name: defPrj

&arg tdate inGrid tmpGrid path\_name

&S mm = [substr %tdate% 5 2]

&S dd = [substr %tdate% 7 2]

/\* Nexrad precipitation output file

&if [EXISTS %path\_name%NEXRAD\_pcp\nex%mm%dd%.txt -file] &then

    &s outpcp = [delete %path\_name%NEXRAD\_pcp\nex%mm%dd%.txt -file]

&s outpcp = %path\_name%NEXRAD\_pcp\nex%mm%dd%.txt

&s huc = D:\Dissertation\InputFiles\gisdata\zonal\_huc1

&s avgtable = pcp.dat

&if [EXISTS pcp.dat -info] &then

    &s ncdcpcp = [delete pcp.dat -info]

&if [EXISTS %tmpGrid% -GRID] &then

    kill %tmpGrid% all

&if [EXISTS tmp.dat -info] &then

    &s ncdcpcp1 = [delete tmp.dat -info]

```
/* Converting NEXRAD precipitation from one-hundredth of mm to mm
```

```
GRID
```

```
setwindow %huc% %huc%
```

```
setcell %huc%
```

```
setmask %huc%
```

```
&type ===== Calculating zonal mean =====
```

```
%avgtable% = zonalstats(%huc%, %inGrid%, MEAN, DATA)
```

```
Q
```

```
copyinfo %huc%.vat tmp.dat
```

```
&type ===== Joining tables =====
```

```
joinitem tmp.dat pcp.dat tmp.dat VALUE
```

```
tables
```

```
select tmp.dat
```

```
unload %outpcp%, COMID, mean, DELIMITED INIT
```

```
Q
```

```
&s del = [delete pcp.dat -info]
```

```
&s dell = [delete tmp.dat -info]
```

```
&return
```

Appendix B - Programs to process NCDC raingauge measured precipitation

++++++ Processing NCDC measured rainfall values for basins in study area ++++++

Public Sub WriteNCDCData()

    ' Adding gauges

    ChkGauges.Add("156028")

    ChkGauges.Add("150700")

    ChkGauges.Add("154898")

    ChkGauges.Add("154893")

    ChkGauges.Add("154905")

    ChkGauges.Add("155111")

    pWorkspaceFactory = New ShapefileWorkspaceFactory

    pWorkspace = pWorkspaceFactory.OpenFromFile(fileIn & "TemporaryWorkingFiles\", 0)

    pFWorkspace = pWorkspace

    pTable = pFWorkspace.OpenTable("sel\_precip\_krb")

    For dd = 1 To 31

        If dd < 10 Then

            tmpDayz = "0" & dd.ToString

        Else

            tmpDayz = dd

        End If

        FldColl.Add("Day" & tmpDayz)

    Next

    For yrs = 2004 To 2005

        For mm = 1 To 12

            If mm < 10 Then

                tmpMonthz = "0" & mm.ToString

            Else

                tmpMonthz = mm

            End If

            monthColl.Add(yrs & tmpMonthz)

        Next

    Dim jj As Double

    Dim ss As Integer = 0

```

Dim kk As Integer

For Each EachMonth In monthColl
    Dim pcpColl As System.Array = System.Array.CreateInstance(GetType(Double), New
Integer() {16, 31}, New Integer() {0, 1})
    Double.TryParse(EachMonth, jj)
    pQFilter = New QueryFilter
    pQFilter.WhereClause = "YEARMO = " & jj '& "'
    pCursor = pTable.Search(pQFilter, False)
    pRow = pCursor.NextRow

    Do While Not pRow Is Nothing
        GaugeIndex = pRow.Fields.FindField("COOPID")
        pCOOPID = pRow.Value(GaugeIndex)
        GaugeColl.Add(pCOOPID)
        kk = 1
        For Each EachDay In FldColl
            pFldIndex = pRow.Fields.FindField(EachDay)
            pPrecip = pRow.Value(pFldIndex)
            Double.TryParse(pPrecip, ii)
            If ii > 0 Then
                tmpVal = (ii / 100.0) * 25.4
            Else
                tmpVal = (0.0)
            End If
            pcpColl(ss, kk) = Math.Truncate(tmpVal * 1000) / 1000
            kk = kk + 1
        Next
        ss = ss + 1
        pRow = pCursor.NextRow
    Loop

    checkingGauges(ss, pcpColl)

writePrecip(GaugeColl, pcpColl, EachMonth)
pcpColl = Nothing
GaugeColl.Clear()
pQFilter = Nothing

```

```

        pCursor = Nothing
        pRow = Nothing
        ss = 0
    Next
    monthColl.Clear()
Next

    pWorkspace = Nothing
    pFWorkspace = Nothing
    pWorkspaceFactory = Nothing
    pTable = Nothing
    ChkGauges.Clear()
End Sub

```

```

Public Sub checkingGauges(ByVal pp As Integer, ByVal pcpcoll As System.Array)

```

```

    For hh As Integer = 0 To ChkGauges.Count - 1
        If Not GaugeColl.Contains(ChkGauges(hh)) Then
            GaugeColl.Add(ChkGauges(hh))
            For qq As Integer = 1 To 31
                pcpcoll(pp, qq) = 0.0
            Next
            pp = pp + 1
        End If
    Next

```

```

End Sub

```

```

Public Sub writePrecip(ByVal output1 As ArrayList, ByVal output2 As System.Array, ByVal mnth As
String)

```

```

    Dim kk As Integer
    Dim tmpFName As String

```

```

    For aa = 1 To 31

```

```

        If aa < 10 Then
            tmpFName = "ncdc_" & mnth & "0" & aa & ".txt"

```



```

Else
    tmpFName = "ncdc_" & mnth & aa & ".txt"
End If

If File.Exists(fileOut & tmpFName) Then
    File.Delete(fileOut & tmpFName)
End If

Dim fs As New IO.FileStream(fileOut & tmpFName, FileMode.Create, FileAccess.Write)
Dim ss As New StreamWriter(fs)
ss.BaseStream.Seek(0, SeekOrigin.End)
For kk = 0 To output1.Count - 1
    ss.WriteLine(output1(kk) & "," & output2(kk, aa))
Next
ss.Flush()
ss.Close()
Next

```

End Sub

Public Sub Convert2Julian()

Dim enday As Integer

Dim JDay As String

For years = 2009 To 2009

For mnths = 1 To 12

If mnths < 10 Then

tmpMonthz = "0" & mnths.ToString

Else

tmpMonthz = mnths

End If

enday = SelLeapDays(years, mnths)

For dayz = 1 To enday

JDay = theday("#" & mnths & "/" & dayz & "/" & years & "#")

If dayz < 10 Then

tmpDayz = "0" & dayz.ToString

Else

```

        tmpDayz = dayz
    End If
    Try
        Microsoft.VisualBasic.Rename(fileIn & "nex" & tmpMonthz & tmpDayz & ".txt",
fileIn & years & "\" & "nex" & years & JDay & ".txt")
    Catch ex As FileNotFoundException
        Debug.Print("file not found for day " & tmpMonthz & tmpDayz)
    Continue For
    End Try
Next
Next
Next
End Sub

```

```

Public Function SelLeapDays(ByVal year As Integer, ByVal selMnth As Integer) As Integer

```

```

    Select Case selMnth
        Case Is = 1, 3, 5, 7, 8, 10, 12
            SelLeapDays = 31
        Case Is = 4, 6, 9, 11
            SelLeapDays = 30
        Case Else
            If julCal.IsLeapYear(year) Then
                SelLeapDays = 29
            Else
                SelLeapDays = 28
            End If
        End Select

```

```

End Function

```

```

Public Function theday(ByVal vDate As Date) As String
    Date2Julian = CLng(Format(Year(vDate), "0000") _
        + Format(DateDiff("d", CDate("01/01/" _
        + Format(Year(vDate), "0000")), vDate) _
        + 1, "000"))
    theday = Right(Date2Julian.ToString, 3)

```

End Function

Start\_IDW.aml

+++++ This section of the programs describe interpolation of point rainfall values using Inverse  
Distance Weighting method +++++

```
&s sdate = 20050101  
&s edate = 20091231  
&s tdate = %sdate%
```

```
&S yy [substr %tdate% 1 4]  
&S mm [substr %tdate% 5 2]  
&S dd [substr %tdate% 7 2]
```

```
&S Eyy [substr %edate% 1 4]  
&S Emm [substr %edate% 5 2]  
&S Edd [substr %edate% 7 2]
```

```
/* Set pathname  
&s path_name = C:\bagya\InputFiles\  

```

```
&do i = %yy% &to %Eyy%
```

```
  &do j = 1 &to 12
```

```
    &select %j%  
      &when 1, 3, 5, 7, 8, 10, 12  
        &s enday = 31  
      &when 4, 6, 9, 11  
        &s enday = 30  
      &otherwise  
        &if [mod %i% 4] = 0 and [mod %i% 100] <> 0 or [mod %i% 400] = 0 &then  
          &s enday = 29  
        &else  
          &s enday = 28  
    &end  
  
    &if %j% < 10 &then
```

```
        &s jj = 0%j%
    &else
        &s jj = %j%

    &do k = 1 &to %enday%

        &if %k% < 10 &then
            &s kk = 0%k%
        &else
            &s kk = %k%

    &r AML_Programs\IDW_NCDC.aml %i%%jj%%kk% %path_name%

    &end /* End of day loop

    &end /* End of month loop

    &end /* End of year loop
```

```

IDW_NCDC.aml
/* Written by Balaji Narashiman and modified by Bagya Palanisamy
&args tdate path_name

&S yy [substr %tdate% 1 4]

tables
dropitem ncdc_poly.pat ncdC_precip
q

/* NWS Precipitation file(Text file)
&s ptext = %path_name%NCDC_pcp\%yy%Ag\ag_%tdate%.txt

/* Output Table

&if [EXISTS %path_name%NCDC_pcp\%yy%IDW\ncdc_%tdate%.txt -file] &then
    &s outpcp = [delete %path_name%NCDC_pcp\%yy%IDW\ncdc_%tdate%.txt -file]

&s outtable = %path_name%NCDC_pcp\%yy%IDW\ncdc_%tdate%.txt
&s avgtable = pcp.dat
&s huc = D:\Dissertation\InputFiles\gisdata\zonal_huc1

&if [EXISTS ncdc_precip.dat -info] &then
    &s nwspcp = [delete ncdC_precip.dat -info]

&if [EXISTS pcp.dat -info] &then
    &s ncdcpcp = [delete pcp.dat -info]

&if [EXISTS tmp.dat -info] &then
    &s tmppcp = [delete tmp.dat -info]

/* Importing the precipitation file into a table
&type Creating Tables ncdc_precip.DAT

TABLES
DEFINE ncdc_precip.DAT
    COOPID,8,8,F,0
    ncdc_PRECIP,4,4,F,1

```

```
~
&type Adding precipitation data from ncdc_pcp file into the table ....
SELECT ncdc_precip.DAT
ADD FROM %ptext%
Q

/* Joining the table with the ncdc station point file
JOINITEM ncdc_poly.pat ncdc_precip.DAT ndcd_poly.pat COOPID

&s out_table = ncdc_pcp.dat

/* My temporary locations

&s inter_ncdc = temp\inter_ncdc

&if [EXISTS %inter_ncdc% -GRID] &then
  kill %inter_ncdc% all

/* Interpolating NCDC precipitation values using IDW method

GRID
%inter_ncdc% = IDW (ncdc_poly, ncdc_precip, #, #, #, #, #, 4000)
setwindow %huc% %huc%
setcell %huc% /* You need to set this to get mean value for all NHD hucs
setmask %huc%
%avgtable% = ZONALSTATS(%huc%, %inter_ncdc%, MEAN, DATA)
Q

copyinfo %huc%.vat tmp.dat

tables
select tmp.dat
dropitem tmp.dat count
dropitem tmp.dat area
dropitem tmp.dat mean
q
```

```
/* Changed huc_cov.pat to %avgtables%
JOINITEM tmp.dat pcp.dat tmp.dat value

tables
select tmp.dat
unload %outtable%, COMID, mean DELIMITED INIT
Q

&s del = [delete pcp.dat -info]
&s dell = [delete tmp.dat -info]

TABLES
SELECT ncdc_precip.DAT
ERASE ncdc_precip.DAT
Y
DROPITEM ncdc_poly.pat ncdc_precip
Q

&return
```



Appendix C - Programs to process SSURGO soil database

+++++++ This program reads ssurgo formatted text to database table ++++++

```
Public Sub readFile()  
    Dim f As Integer  
    Dim rgItems As String  
    Dim fStream As System.IO.StreamReader  
    Dim fName As String = "D:\Data\soils\" 'InputBox("Enter the Path name with backslash at the  
end")  
    Dim cntyName As String = "clay,knox" 'InputBox("Enter county names separated by comma")  
    Dim fNameArr As New List(Of String)  
    Dim fstCnty() As Object  
    Dim fNameEle, tmpTxt As String  
    Dim solColl As New Collection  
  
    solColl.Add("mapunit.txt")  
    solColl.Add("comp.txt")  
    solColl.Add("chorizon.txt")  
  
    For Each item In solColl  
  
        If cntyName.Contains(",") Then  
            fstCnty = cntyName.Split(",")  
            For ii = 0 To fstCnty.Length - 1  
                tmpTxt = (fstCnty.GetValue(ii))  
                tmpTxt = cntyFIPS.getFIPS(tmpTxt)  
                fNameEle = fName & "soil_" & tmpTxt & "tabular\" & item  
                fNameArr.Add(fNameEle)  
                fNameEle = Nothing  
            Next  
        Else  
            tmpTxt = cntyFIPS.getFIPS(cntyName)  
            fName = fName & tmpTxt & "tabular\" & item  
            fNameArr.Add(fName)  
        End If  
  
        f = FreeFile()  
  
        If IO.File.Exists("C:\bagya\ssurgo\xl_dBase\" & "tmp_" & Left(item, 4) & "Col.csv") Then
```

```

        IO.File.Delete("C:\bagya\ssurgo\xl_dBase\" & "tmp_" & Left(item, 4) & "Col.csv")
    End If

    IO.File.Copy("C:\bagya\ssurgo\xl_dBase\" & Left(item, 4) & "_col.csv",
"C:\bagya\ssurgo\xl_dBase\tmp_" & Left(item, 4) & "Col.csv")

    Dim fs As New IO.FileStream("C:\bagya\ssurgo\xl_dBase\tmp_" & Left(item, 4) & "Col.csv",
    FileMode.Append, FileAccess.Write)
    Dim ss As New StreamWriter(fs)

    For kk = 0 To fNameArr.Count - 1

        Try
            fStream = IO.File.OpenText(fNameArr(kk))
        Catch ex As DirectoryNotFoundException
            Exit For
        End Try

        Do While fStream.Peek >= 0
            rgItems = Regex.Replace(fStream.ReadLine, "\\|", ",")
            ss.WriteLine(rgItems)
        Loop

    Next

    ss.Flush()
    ss.Close()
    fNameArr.Clear()

    filePath = "C:\bagya\ssurgo\xl_dbase\"
    Select Case item
        Case Is = "mapunit.txt"
            outFile = "mapunit.dbf"
        Case Is = "comp.txt"
            outFile = "component.dbf"
        Case Is = "chorizon.txt"
            outFile = "horizon.dbf"
    End Select

```

```

        If IO.File.Exists(filePath & outFName) Then
            IO.File.Delete(filePath & outFName)
            IO.File.Delete(filePath & outFName & ".xml")
        End If

        convert2Table("tmp_" & Left(item, 4) & "Col.csv", outFName)
    Next

    joinSoil(filepath)

End Sub

Public Sub convert2Table(ByVal fname As String, ByVal outFname As String)
    Dim wsf As IWorkspaceFactory
    wsf = New TextFileWorkspaceFactory
    Dim wrkspc As IWorkspace
    wrkspc = wsf.OpenFromFile("C:\bagya\ssurgo\xl_dBase\", 0)
    Dim fws As IFeatureWorkspace
    fws = wrkspc
    Dim xlTable As ITable
    xlTable = fws.OpenTable(fname)
    Dim xl2Table As TableToTable
    xl2Table = New TableToTable
    xl2Table.in_rows = xlTable
    xl2Table.out_name = outFname
    xl2Table.out_path = "C:\bagya\ssurgo\xl_dBase\"
    Dim gp As Geoprocessor
    gp = New Geoprocessor
    gp.Execute(xl2Table, Nothing)
End Sub

```

++++ This program relates soil database files: mapunit, component and chorizon ++++++

```
Public Sub openXlFile()  
    Dim wsf As IWorkspaceFactory  
    wsf = New ExcelWorkspaceFactory  
    Dim wrkspace As IWorkspace  
    wrkspace = wsf.OpenFromFile("C:\bagya\ssurgo\xl_dbase\", 0)  
  
    Dim fws As IFeatureWorkspace  
    fws = wrkspace  
    Dim xlTable As ITable  
    xlTable = fws.OpenTable("chr")  
    Dim xl2Table As TableToTable  
    xl2Table = New TableToTable  
    xl2Table.in_rows = xlTable  
    xl2Table.out_name = "horizon.dbf"  
    xl2Table.out_path = "C:\bagya\ssurgo\xl_dbase\  
    Dim gp As Geoprocessor  
    gp = New Geoprocessor  
    gp.Execute(xl2Table, Nothing)  
End Sub  
  
Public Sub joinSoil(ByVal filePath As String)  
  
    pFact = New ShapefileWorkspaceFactory  
    pWorkspace = pFact.OpenFromFile(filePath, 0)  
    pFWorkspace = pWorkspace  
    pMapTable = pFWorkspace.OpenTable("mapunit")  
    pCompTable = pFWorkspace.OpenTable("component")  
    pHoriTable = pFWorkspace.OpenTable("horizon")  
  
    ' Remember to correctly identify the origin and target tables  
    pTableComp = TableJoin(pCompTable, pMapTable, "mukey", "mukey")  
    pTableHori = TableJoin(pTableComp, pHoriTable, "cokey", "cokey")  
    pDataset = pTableHori  
    pDSName = pDataset.FullName  
  
    ' Now for each COMID, grab the soil properties of major soil component
```

```

Dim zoneTabPath As String
zoneTabPath = "D:\Dissertation\Evaluation_Watershed\"

pSFact = New ShapefileWorkspaceFactory
pWs = pSFact.OpenFromFile(zoneTabPath, 0)
Dim pFWs As IFeatureWorkspace
pFWs = pWs
Dim pZoneTable As ITable
pZoneTable = pFWs.OpenTable("comid_soil")

'Get standalonetable
Dim pStTable As IStandaloneTable
pStTable = New StandaloneTable
pStTable.Table = pZoneTable

'Set cursor to increment through sorted table, one row at a time
pSortedCursor = pTableSort.Rows
pRowBuff = pSortedCursor.NextRow

'Loop through table and create a collection of values
Do While Not pRowBuff Is Nothing
    fldIndex = pRowBuff.Fields.FindField("MAJORITY")
    comIndex = pRowBuff.Fields.FindField("VALUE")
    comVal = pRowBuff.Value(comIndex)
    mukeyVal = pRowBuff.Value(fldIndex)
    pQueryFilter.WhereClause = "mukey = " & mukeyVal
    pSelSet = pTableHori.Select(pQueryFilter, esriSelectionType.esriSelectionTypeIDSet,
esriSelectionOption.esriSelectionOptionNormal, pWs)
    pRowBuff = pSortedCursor.NextRow
Loop

' -----

Dim pWkSpFactory As IWorkspaceFactory
pWkSpFactory = New ShapefileWorkspaceFactory
Dim outPath As String
outPath = "C:\bagya\Projects_NET\"
pWkSp = pWkSpFactory.OpenFromFile(outPath, 0)

```

```

pWkSpDS = pWkSp
pWkSpName = pWkSpDS.FullName
pOutDSName = New TableName
Dim pOut1 As String = "ssurgo051"
pOutDSName.WorkspaceName = pWkSpName
pOutDSName.Name = pOut1

If Dir$(outPath & "ssurgo051.dbf") <> "" Then
    Kill(outPath & "ssurgo051.dbf")
End If

pExpOp = New ExportOperation
pExpOp.ExportTable(pDSName, Nothing, Nothing, pOutDSName, 0)

End Sub

```

```

Private Function TableJoin(ByVal targetTable As ITable, ByVal joinTable As ITable, ByVal
fromField As String, ByVal toField As String) As ITable
    Dim mRCfactory As IMemoryRelationshipClassFactory
    mRCfactory = New MemoryRelationshipClassFactory
    Dim memRC As IRelationshipClass
    memRC = mRCfactory.Open("MuComp", targetTable, fromField, joinTable, toField, "forward",
"backward", esriRelCardinality.esriRelCardinalityOneToMany)
    Dim rqTable As IRelQueryTable
    Dim rqtfactory As IRelQueryTableFactory
    rqtfactory = New RelQueryTableFactory
    rqTable = rqtfactory.Open(memRC, True, Nothing, Nothing, "", True, False)
    TableJoin = rqTable
End Function

```

++++ This section of programs writes component averaged soil properties to soil input files +++++

```
Public Sub getTransmissivity()  
    pFact = New AccessWorkspaceFactory  
    pWorkspace = pFact.OpenFromFile("C:\bagya\ssurgo\ssurgoExt.mdb", 0)  
    pFWorkspace = pWorkspace  
    pSolTable = pFWorkspace.OpenTable("soils")  
    pResTable = pFWorkspace.OpenTable("restrictions")  
    pJoinedTable = TableJoin(pSolTable, pResTable, "CompHor_MapComp_cmp_cokey", "cokey")  
    export2Table(pJoinedTable, "restrictiveSoil", "C:\bagya\ssurgo\ssurgoExt.mdb")  
End Sub
```

```
Public Sub getTrans()  
    pFact = New AccessWorkspaceFactory  
    pWorkspace = pFact.OpenFromFile("D:\Dissertation\InputFiles\gisdata\ssurgoExt.mdb", 0)  
    pFWorkspace = pWorkspace  
    pSolTable = pFWorkspace.OpenTable("soils")  
    pSolCursor = pSolTable.Search(Nothing, False)  
    pRow = pSolCursor.NextRow  
    pComIdIndex = pRow.Fields.FindField("huc_ssurgo_COMID")  
  
    Do While Not pRow Is Nothing  
        pQueryfilter = New QueryFilter  
        pComId = pRow.Value(pComIdIndex)  
        If pComId = tmpCom Then  
            pRow = pSolCursor.NextRow  
            Continue Do  
        End If  
        pQueryfilter.WhereClause = "huc_ssurgo_COMID = " & pComId  
        pResDep = GetSummary(pSolTable, "srmax", pQueryfilter)  
        writeResDepth(pResDep, pComId)  
        tmpCom = pComId  
        pRow = pSolCursor.NextRow  
    Loop  
  
End Sub
```



```

Public Function GetSummary(ByVal pTable As ITable, ByVal sFieldName As String, ByVal queryFilt As
IQueryFilter) As Double
    Dim pCursor As ICursor
    pCursor = pTable.Search(queryFilt, False)
    Dim pDataStatistics As IDataStatistics
    pDataStatistics = New DataStatistics
    pDataStatistics.Field = sFieldName
    pDataStatistics.Cursor = pCursor
    Dim pStatistics As IStatisticsResults
    pStatistics = pDataStatistics.Statistics
    GetSummary = pStatistics.Mean
    pCursor = Nothing
    GC.Collect()
End Function

```

```

Public Sub getSurK()
    pFact = New AccessWorkspaceFactory
    pWorkspace = pFact.OpenFromFile("D:\Dissertation\InputFiles\gisdata\ssurgoExt.mdb", 0)
    pFWorkspace = pWorkspace
    pSolTable = pFWorkspace.OpenTable("soils")
    pSolCursor = pSolTable.Search(Nothing, False)
    pRow = pSolCursor.NextRow
    pComIdIndex = pRow.Fields.FindField("huc_ssurgo_COMID")
    pMukeyIndex = pRow.Fields.FindField("huc_ssurgo_mukey")
    Dim newPathFile As String = "C:\bagya\ssurgo\soil_analysis\properties\"

    Do While Not pRow Is Nothing
        pQueryfilter = New QueryFilter
        pComId = pRow.Value(pComIdIndex)
        pMukey = pRow.Value(pMukeyIndex)
        If pComId = tmpCom Then
            pRow = pSolCursor.NextRow
            Continue Do
        End If
        pQueryfilter.WhereClause = "huc_ssurgo_COMID = " & pComId
        writeResDepth(OpenSoilProps(newPathFile, pMukey), pComId)
        tmpCom = pComId
        pRow = pSolCursor.NextRow
    End While

```

Loop

End Sub

```
Public Function OpenSoilProps(ByVal newpathfile As String, ByVal SelMukey As String) As Double
    Dim solPath As String = newpathfile & SelMukey & ".txt"
    Dim srl As New StreamReader(solPath)
    Dim contents As String
    Dim contentsList() As String
    While srl.Peek <> -1
        contents = srl.ReadLine
    End While
    Double.TryParse(contents, OpenSoilProps)
    Return OpenSoilProps
End Function
```

```
Public Sub writeResDepth(ByVal resDepth As Double, ByVal comid As String)
```

```
    If IO.File.Exists("C:\bagya\InputFiles\solfiles\sol_" & comid & ".txt") Then
        Dim solPath As String = "C:\bagya\InputFiles\solfiles\sol_" & comid & ".txt"
        Dim srl As New StreamReader(solPath)
        Dim contents As String
        Dim contentsList As New List(Of String)
        Dim strItem, stritem1 As String
        While srl.Peek <> -1
            contents = srl.ReadLine
            contentsList.Add(contents)
        End While
        srl.Close()
        Dim sw1 As New StreamWriter(solPath, False)
        For ii = 0 To contentsList.Count - 1
            Select Case ii
                Case Is = 0
                    Dim strLength As Integer = contentsList.Item(0).Length
                    Dim additem1 As String = ",Ksat-m/day"
                    stritem1 = contentsList.Item(0).Insert(strLength, additem1)
                Case Is = 1
                    Dim additem2 As String = "," & resDepth
```

```
        Dim strLength1 As Integer = contentsList.Item(1).Length
        strItem = contentsList.Item(1).Insert(strLength1, additem2)
    End Select
Next
sw1.WriteLine(stritem1)
sw1.WriteLine(strItem)
Debug.Print("The written file is " & "sol_" & comid)
sw1.Flush()
sw1.Close()
Else
    Debug.Print("The file " & "sol_" & comid & " does not exist")
    Stop
End If

End Sub
```

Appendix D - Programs to create Topographic Wetness Index file

+++++ This program creates the histogram of topographic indices excluding urban cells +++++

Public Sub createHistogram()

```
    getFClass("D:\Data\InputFiles\gisdata\storeHUUCS\")
    pReclassOp = New RasterReclassOp
    pInRaster = twiGrid

    lulcData = GetRasterDataset("D:\Data\InputFiles\gisdata\temp\", "onlynonurb")
    Dim pRasLulc As IRaster
    Dim pExtOp As IExtractionOp
    Dim pUrbRasDescriptor As IRasterDescriptor
    Dim pUrbGrd As IRaster
    Dim pQFilter As IQueryFilter
    Dim pLulcGeoDS As IGeoDataset
    Dim pNonUrbanRas As IRaster
    pRasLulc = lulcData.CreateDefaultRaster
    pLulcGeoDS = lulcData

    For ii = 1 To intDSCount

        pFCMask = pFWorkspace.OpenFeatureClass(aDSNames(ii))

        pExtOp = New RasterExtractionOp
        ' Setting analysis environment and mask
        pRasEnv = pExtOp
        pRasEnv.Mask = pFCMask
        pRasEnv.SetExtent(esriRasterEnvSettingEnum.esriRasterEnvValue, pFCMask)

        Dim pWS1 As IWorkspace
        Dim pWSF1 As IWorkspaceFactory
        pWSF1 = New RasterWorkspaceFactory
        pWS1 = pWSF1.OpenFromFile("D:\Data\InputFiles\gisdata\temp\", 0)
        pRasEnv.OutWorkspace = pWS1

        ' Get urban cells
        ' ----- NOW DETERMINE HOW TO GET THE SECOND MASK -----
        pUrbRasDescriptor = New RasterDescriptor
```

```

pUrbRasDescriptor.Create(pRasLulc, Nothing, "value")
pUrbGrd = pExtOp.Attribute(pUrbRasDescriptor)

pRasEnv = Nothing
pRasEnv = New RasterAnalysis
pRasEnv.Mask = pUrbGrd
pRasEnv.SetExtent(esriRasterEnvSettingEnum.esriRasterEnvValue, pUrbGrd, pUrbGrd)

' Get the Map
Dim pRaster As IRaster
Dim vValues As Object = Nothing
Dim vFrequencies As Object = Nothing
Dim pTWI As IRaster

pRaster = twiGrid.CreateDefaultRaster
Dim pGeoTWI As IGeoDataset
pGeoTWI = twiGrid
pTWI = pExtOp.Raster(pGeoTWI, pUrbGrd)
Dim outName As String = "twi" & Mid(aDSNames(ii), 4, 6)
MakePermanent(pTWI, "D:\Data\InputFiles\gisdata\histogramTWI\", outName)

Next
End Sub

+++++ This program was extracted from ESRI Arcobjects model example ++++++
Public Sub MakePermanent(ByVal pResultOfSpatialOp As IRaster, ByVal outPath As String, ByVal
outFileName As String)
' Query the output (a Raster object) for IRasterBandCollection
Dim pRasBandC As IRasterBandCollection
pRasBandC = pResultOfSpatialOp

' Get the dataset from the first band
Dim pRasterDS As IRasterDataset
pRasterDS = pRasBandC.Item(0).RasterDataset

' Query the dataset for ITemporaryDataset
Dim pTemp As ITemporaryDataset

```

```

pTemp = pRasterDS

Dim pWSF As IWorkspaceFactory
Dim pWS As IWorkspace
pWSF = New RasterWorkspaceFactory
pWS = pWSF.OpenFromFile(outPath, 0)

' Deleting output raster if it exists already
Try
    Dim pOutputRas As IRasterDataset
    pOutputRas = GetRasterDataset(outPath, outFile_name)
    Dim pOutDS1 As IDataset
    pOutDS1 = pOutputRas
    If pOutDS1.CanDelete Then
        pOutDS1.Delete()
    End If
Catch ex As COMException
    Debug.Print("No raster named " & outFile_name & "found")
Exit Try
End Try
pTemp.MakePermanentAs(outFile_name, pWS, "GRID") ' Never use the open directory to store files
End Sub

```

```
+++++++ This section of programs classify and writes the classified topographic indices to text file
+++
+++++++ Programs downloaded from ESRI user forums, ESRI developer networks and modified to suit the
needs
        of the research +++++
```

```
Sub Main()
    Dim hucArea As Double
    getRasterEnum(pathName)
    For ii = 2 To intDSCount
        twiGrid = GetRasterDataset(pathName, aDSNames(ii))
        twiRas = twiGrid.CreateDefaultRaster
        hucArea = ReadArea.rchArea(Right(aDSNames(ii), 6))
        hucArea = hucArea * 1000000
        Reclassify_TWI.Classify_Click(aDSNames(ii), twiRas, hucArea)
    Next
End Sub
```

```
Public Sub Classify_Click(ByVal fileName As String, ByVal praster As IRaster, ByVal totArea As
Double)
    Dim vBreaks As Object
    vBreaks = GetClassificationBreaks(fileName, praster, totArea)
End Sub
```

```
Public Function GetClassificationBreaks(ByVal filename As String, ByVal inRaster As IRaster, ByVal
totarea As Double) As Object
    'Get unique values from the raster
    Dim pUV As IUniqueValues
    pUV = New UniqueValues
    Dim pRCUV As IRasterCalcUniqueValues
    pRCUV = New RasterCalcUniqueValues
    Try
        pRCUV.AddFromRaster(inRaster, 0, pUV)
    Catch ex As COMException
        MsgBox(ex.Message & "in raster " & filename)
        Return Nothing
    Exit Function
End Try
```



```
'Get the array of values and counts from the raster
Dim vVals, vCnts As Object
pUV.GetHistogram(vVals, vCnts)

'Sort the arrays using the QuickSort method
'This is quicker than sorting w/ the BubbleSort method
'when there are a large number of values.
Call dhQuickSort(vVals, vCnts)

'Define initial classification parameters
Dim pClassify As IClassifyGEN
Dim pReclassOp As IReclassOp
Dim classBreak As Object
classBreak = Nothing
pReclassOp = New RasterReclassOp
pClassify = New NaturalBreaks
pClassify.Classify(vVals, vCnts, 10)
classBreak = pClassify.ClassBreaks
Dim pRemap As IRemap
Dim pNumRemap As INumberRemap
Dim reclassTable As ITable
pNumRemap = New NumberRemap
Dim pOutputRaster As IRaster

'set the remap range
pNumRemap.MapRange(classBreak(0), classBreak(1), 1)
pNumRemap.MapRange(classBreak(1), classBreak(2), 2)
pNumRemap.MapRange(classBreak(2), classBreak(3), 3)
pNumRemap.MapRange(classBreak(3), classBreak(4), 4)
pNumRemap.MapRange(classBreak(4), classBreak(5), 5)
pNumRemap.MapRange(classBreak(5), classBreak(6), 6)
pNumRemap.MapRange(classBreak(6), classBreak(7), 7)
pNumRemap.MapRange(classBreak(7), classBreak(8), 8)
pNumRemap.MapRange(classBreak(8), classBreak(9), 9)
pNumRemap.MapRange(classBreak(9), classBreak(10), 10)

pRemap = pNumRemap
```

```

    reclassTable = pRemap.RepresentAsTable
    pOutputRaster = pReclassOp.ReclassByRemap(inRaster, pRemap, False)
    Call exportOut(filename, pOutputRaster, reclassTable, totarea)
    Return Nothing
End Function

```

```

Public Sub exportOut(ByVal filename As String, ByVal inGrd As IRaster, ByVal joinReclassTable As
ITable, ByVal totarea As Double) ' This is to export the output table

```

```

    Dim pTable As ITable
    Dim justTheCntTable As ITable
    Dim pRastercol As IRasterBandCollection
    Dim pBand As IRasterBand
    ' declare single field
    Dim pField As IField
    Dim pTableSort As ITableSort
    Dim intFieldIndex As Integer 'should go on top

    ' First getting raster attribute table
    pRastercol = inGrd
    pBand = pRastercol.Item(0)
    justTheCntTable = pBand.AttributeTable

    pTable = joinReclassTable

    Dim pMemClassFact As IMemoryRelationshipClassFactory
    pMemClassFact = New MemoryRelationshipClassFactory

    Dim pRelClass As IRelationshipClass
    pRelClass = pMemClassFact.Open("TWI_Class", pTable, "OUT", justTheCntTable, "VALUE",
"forward", "backwards", esriRelCardinality.esriRelCardinalityOneToOne)

    'Perform Join

    Dim pRQTFact As IRelQueryTableFactory
    Dim pRQTable As ITable

```

```
pRQTFact = New RelQueryTableFactory
pRQTable = pRQTFact.Open(pRelClass, True, Nothing, Nothing, "", True, True)
Dim pDSet As IDataset
pDSet = pRQTable
Dim pDSName As IDatasetName
pDSName = pDSet.FullName
' Export output to text file
Dim outPath As String
outPath = "D:\Data\InputFiles\gisdata\temp\"

' Get the output dataset name ready. In this
Dim pWkSpFactory1 As IWorkspaceFactory
Dim pWkSp1 As IWorkspace
Dim pFWS As IFeatureWorkspace
Dim pWkSpDS1 As IDataset
Dim pWkSpName1 As IWorkspaceName
Dim pOutDSName1 As IDatasetName
pWkSpFactory1 = New ShapefileWorkspaceFactory
pWkSp1 = pWkSpFactory1.OpenFromFile(outPath, 0)
pWkSpDS1 = pWkSp1
pWkSpName1 = pWkSpDS1.FullName
pOutDSName1 = New TableName
pOutDSName1.Name = "tmpTable"
pOutDSName1.WorkspaceName = pWkSpName1

' Opening temp table for deletion
Dim tmpWKFact As IWorkspaceFactory
Dim tmpFWS As IFeatureWorkspace
Dim tmpTabName As ITable
tmpWKFact = New ShapefileWorkspaceFactory
tmpFWS = tmpWKFact.OpenFromFile(outPath, 0)

Try
    tmpTabName = tmpFWS.OpenTable("tmpTable")
    ' Delete if the table already exists
    Dim pTmpDS2 As IDataset
    pTmpDS2 = tmpTabName
```

```

        If pTmpDS2.CanDelete Then
            pTmpDS2.Delete()
        End If
        tmpWKFact = Nothing
        tmpFWS = Nothing
        tmpTabName = Nothing
        pTmpDS2 = Nothing
    Catch ex As COMException
        Debug.Print("The tempTable does not exists")
        Exit Try
    End Try

    ' Export (Selection is ignored)
    Dim pExpOp As IExportOperation
    pExpOp = New ExportOperation
    pExpOp.ExportTable(pDSName, Nothing, Nothing, pOutDSName1, 0)

    pDSet = Nothing
    pDSName = Nothing
    pRelClass = Nothing
    pMemClassFact = Nothing

    pFWS = pWkSpFactory1.OpenFromFile(outPath, 0)
    pTable = pFWS.OpenTable("tmpTable")

    ' ----- NOW ADDING THE AREA FIELD -----

    Dim pFieldEdit As IFieldEdit
    Dim pFieldEdit2 As IFieldEdit2
    Dim pFldName(1) As String
    Dim pCalc As ICalculator
    Dim pCalcCursor As ICursor
    Dim expressionOne, expressionTwo As String
    expressionOne = "[COUNT]" & " * 9.144 * 9.144"
    expressionTwo = "[Area]" & " / " & totarea

    pCalc = New Calculator
    pFldName(0) = "Area"

```

```
pFldName(1) = "PerArea"  
pField = New Field  
pFieldEdit = pField  
Dim pField2 As IField2  
pField2 = New Field  
pFieldEdit2 = pField2  
  
pFieldEdit.Name_2 = "Area"  
pFieldEdit.Type_2 = esriFieldType.esriFieldTypeDouble  
pFieldEdit.Scale_2 = 4  
pFieldEdit.Precision_2 = 15  
pField = pFieldEdit  
pTable.AddField(pField)  
pFieldEdit2.Name_2 = "PerArea"  
pFieldEdit2.Type_2 = esriFieldType.esriFieldTypeDouble  
pFieldEdit2.Scale_2 = 4  
pFieldEdit2.Precision_2 = 15  
pField2 = pFieldEdit2  
pTable.AddField(pField2)  
  
' Calculate area and percentage area  
  
pCalcCursor = pTable.Update(Nothing, True)  
pCalc = New Calculator  
With pCalc  
    .Cursor = pCalcCursor  
    .Expression = expressionOne  
    .ShowErrorPrompt = True  
    .Field = "Area"  
    .Calculate()  
End With  
  
pCalc = Nothing  
pCalcCursor = Nothing  
  
pCalcCursor = pTable.Update(Nothing, True)  
pCalc = New Calculator  
With pCalc
```

```

        .Cursor = pCalcCursor
        .Expression = expressionTwo
        .ShowErrorPrompt = True
        .Field = "PerArea"
        .Calculate()
    End With

    ' Export output to text file
    Dim twiPath As String = "D:\Data\InputFiles\twifiles\"
    Dim pWkSpFactory As IWorkspaceFactory
    Dim pWkSp As IWorkspace
    Dim pWkSpDS As IDataset
    Dim pWkSpName As IWorkspaceName
    Dim pOutDSName As IDatasetName
    pDSet = pTable
    pDSName = pDSet.FullName
    pWkSpFactory = New ShapefileWorkspaceFactory
    pWkSp = pWkSpFactory.OpenFromFile(twiPath, 0)
    pWkSpDS = pWkSp
    pWkSpName = pWkSpDS.FullName
    pOutDSName = New TableName
    pOutDSName.Name = "twi_" & Right(filename, 6)
    pOutDSName.WorkspaceName = pWkSpName
    Dim pQFilter As IQueryFilter
    pQFilter = New QueryFilter
    pQFilter.WhereClause = ""
    pQFilter.SubFields = "TO,PerArea"

    If IO.File.Exists(twiPath & "twi_" & Right(filename, 6) & ".dbf") Then
        IO.File.Delete(twiPath & "twi_" & Right(filename, 6) & ".dbf")
    End If

    Dim pExpOp2 As IExportOperation
    pExpOp2 = New ExportOperation
    pExpOp2.ExportTable(pDSName, pQFilter, Nothing, pOutDSName, 0)
    pWkSpFactory = Nothing
    pWkSp = Nothing
    pWkSpDS = Nothing

```

```

pWkSpName = Nothing
pOutDSName = Nothing

export2TextFile(twiPath, "twi_" & Right(filename, 6))

pRelClass = Nothing
pMemClassFact = Nothing
pWkSpFactory = Nothing
pWkSpFactory1 = Nothing
pWkSp = Nothing
pWkSp1 = Nothing
pWkSpDS = Nothing
pWkSpDS1 = Nothing
pWkSpName = Nothing
pOutDSName = Nothing
pOutDSName1 = Nothing
pDSet = Nothing
pDSName = Nothing
pQFilter = Nothing

```

End Sub

```
Public Function export2TextFile(ByVal pathName As String, ByVal fileName As String)
```

```

' Export output to text file
Dim pOutTable As ITable
Dim pOutFWS As IFeatureWorkspace
Dim pDSet As IDataset
Dim pTableSort As ITableSort
Dim pwkspacfactory As IWorkspaceFactory
pwkspacfactory = New ShapefileWorkspaceFactory
pOutFWS = pwkspacfactory.OpenFromFile(pathName, 0)
pOutTable = pOutFWS.OpenTable(fileName)
pTableSort = New TableSort
With pTableSort
    .Fields = "TO"
    .Ascending("TO") = False

```

```

        .QueryFilter = Nothing
        .Table = pOutTable
        .Sort(Nothing)
    End With
    Dim pCursor As ICursor
    Dim pTWIList As Object
    Dim pAreaList As Object
    pCursor = pTableSort.Rows
    Dim pRow As IRow
    pRow = pCursor.NextRow
    ReDim pTWIList(0 To 9)
    ReDim pAreaList(0 To 9)
    Dim ii = 0
    Do While Not pRow Is Nothing
        pTWIList(ii) = pRow.Value(5)
        pAreaList(ii) = pRow.Value(9)
        pRow = pCursor.NextRow
        ii = ii + 1
    Loop

    writeData(pathName, fileName, pTWIList, pAreaList)
    pDSet = Nothing
    pDSet = pOutTable
    If pDSet.CanDelete Then
        pDSet.Delete()
    End If

    Return Nothing

End Function

Public Sub writeData(ByVal newPathFile As String, ByVal fileName As String, ByVal output() As Object,
ByVal output2() As Object)

    ' Copy header information to a new file for writing
    If IO.File.Exists(newPathFile & fileName & ".txt") Then
        IO.File.Delete(newPathFile & fileName & ".txt")
    End If

```



```
        Dim fs As New IO.FileStream(newpathFile & fileName & ".txt", FileMode.Create,  
FileAccess.Write)  
        Dim ss As New StreamWriter(fs)  
        Dim kk As Integer  
        kk = 0  
  
        For kk = 0 To output.GetUpperBound(0)  
            ss.Write(output(kk))  
            ss.Write(",")  
            ss.Write(output2(kk))  
            ss.WriteLine()  
        Next  
  
        ss.Flush()  
        ss.Close()
```

End Sub

Appendix E - Programs to determine percent urban

```
Public Sub lulc()  
    ' Open Shapefile  
    Dim pWF As IWorkspaceFactory  
    pWF = New ShapefileWorkspaceFactory  
    Dim pWs As IFeatureWorkspace  
    pWs = pWF.OpenFromFile(hucPath, 0)  
    Dim pInputFc As IFeatureClass  
    pInputFc = pWs.OpenFeatureClass(hucName)  
    tmpDS = pInputFc  
    tmpDSName = tmpDS.FullName  
  
    ' Getting the filter  
    pQFilter = New QueryFilter  
    pQFilter.WhereClause = "COMID > 0"  
    pHucCursor = pInputFc.Search(pQFilter, False)  
    pHucRow = pHucCursor.NextRow  
    pFldIndex = pHucRow.Fields.FindField("COMID")  
  
    Do While Not pHucRow Is Nothing  
        pQFilter = New QueryFilter  
        ' Get the COMID value from each row  
        selComId = pHucRow.Value(pFldIndex)  
        If selComId = 0 Then  
            pHucCursor.NextRow()  
            Continue Do  
        End If  
        ' Create query for each COMID  
        pQFilter.WhereClause = " COMID = " & selComId  
        ' Export the featureclass  
        ' Create a new feature class name  
        pOutFeatureClassName = New FeatureClassName  
        pOutDSName = pOutFeatureClassName  
        pOutDSName.Name = "tmp" & selComId  
  
        pOutWorkspaceName = New WorkspaceName  
        pOutWorkspaceName.PathName = sShpPath  
        pOutWorkspaceName.WorkspaceFactoryProgID = "esriCore.ShapefileWorkspaceFactory.1"  
        pOutDSName.WorkspaceName = pOutWorkspaceName
```

```

pOutFeatureClassName.FeatureType = esriFeatureType.esriFTSimple
pOutFeatureClassName.ShapeType = pInputFc.ShapeType
pOutFeatureClassName.ShapeFieldName = "Shape"

' Check if the output exists already
pDelFC(sShpPath, pOutDSName.Name)
pExpOp = New ExportOperation
pExpOp.ExportFeatureClass(tmpDSName, pQFilter, Nothing, Nothing, pOutDSName, 0)

' ----- Raster Analysis -----
'pMapAlgeOp = New RasterAnalysis
pRasEnv = New RasterAnalysis
' Get the mask
pWs = Nothing
pWkSpFactory = Nothing
pInputFc = Nothing
pWkSpFactory = New ShapefileWorkspaceFactory
pWs = pWkSpFactory.OpenFromFile(sShpPath, 0)
pInputFc = pWs.OpenFeatureClass(pOutDSName.Name)
' Get Extent from the pInputFC
pFCursor = pInputFc.Update(Nothing, False)
pFeature = pFCursor.NextFeature
pPolygon = pFeature.Shape
'pRasExp = New RasterExtractionOp
pComEnv = pPolygon.Envelope
' End Getting extent
pRasEnv.Mask = pInputFc
pRasEnv.SetExtent(esriRasterEnvSettingEnum.esriRasterEnvValue, pInputFc)
pRasEnv.SetAsNewDefaultEnvironment()
tmpLULC = GetRasterDataset(lulcPath, lulcName)
tmpRas = tmpLULC
MakePermanent(tmpRas, sShpPath)
pRasComid = tmpLULC.CreateDefaultRaster

ReDim outObj(0 To 5)

outObj(0) = getpixelcounts(pRasComid, 22)

```

```
If outObj(0) <> 0 Then
    outObj(0) = (outObj(0) * 9.14 * 9.14) / 10000000.0
    urbImp = 0.12
    outObj(1) = urbImp
Else
    outObj(1) = 0
End If
outObj(2) = getpixelcounts(pRasComid, 23)
If outObj(2) <> 0 Then
    outObj(2) = outObj(2) * 9.14 * 9.14
    urbImp = 0.38
    outObj(3) = urbImp
Else
    outObj(3) = 0
End If
outObj(4) = getpixelcounts(pRasComid, 24)
If (outObj(4) <> 0) Then
    outObj(4) = outObj(4) * 9.14 * 9.14
    urbImp = 0.6
    outObj(5) = urbImp
Else
    outObj(5) = 0
End If

writeData("D:\Data\InputFiles\urban\%", outObj)
pQFilter = Nothing
pRasterDataset = Nothing
pWs = Nothing
pWkSpFactory = Nothing
pInputFc = Nothing
pRasEnv = Nothing
Loop

End Sub
```

```

Public Function getpixelcounts(ByVal pcountraster As IRaster, ByVal val As Long) As Integer
    'Created by Gareth Mann for the Institute of Food and Agricultural Sciences, Fort Lauderdale,
    Florida, United States

    'This function gets the total habitat areas calculated from from a raster in cells

    Dim pRBC As IRasterBandCollection
    Dim pRB As IRasterBand
    Dim pTable As ITable
    Dim pRasProps As IRasterProps
    Dim pCursor As ICursor
    Dim pRow As IRow
    Dim pQueryFilter As IQueryFilter

    pRBC = pcountraster
    pRB = pRBC.Item(0)

    pTable = pRB.AttributeTable

    pQueryFilter = New QueryFilter

    pQueryFilter.WhereClause = "Value = " & val

    'now we retrieve the value from the specified field at the specified cursor:
    pCursor = pTable.Search(pQueryFilter, False)
    pRow = pCursor.NextRow

    'added this line just in case there are no counts
    If pRow Is Nothing Then
        getpixelcounts = 0
    Else
        getpixelcounts = pRow.Value(pTable.FindField("Count"))
    End If

    'and release.....
    pQueryFilter = Nothing
    pCursor = Nothing
    pRow = Nothing

```

```
pRasProps = Nothing
pRBC = Nothing
pRB = Nothing
pTable = Nothing
```

```
End Function
```

```
Public Sub writeData(ByVal pathFile As String, ByVal output() As Object)
```

```
    Dim searchResults As String() = Directory.GetFiles(pathFile, "sol_" & selComId & ".txt",
SearchOption.TopDirectoryOnly)
```

```
    If File.Exists("D:\Data\InputFiles\ssurgo\sol_" & selComId & ".txt") Then
```

```
        If IsDBNull(output) Then
```

```
            Dim fs1 As New IO.FileStream(pathFile, FileMode.Append, FileAccess.Write)
```

```
            Dim ss1 As New IO.StreamWriter(fs1)
```

```
            ss1.BaseStream.Seek(0, SeekOrigin.End)
```

```
            ss1.Write("Yes")
```

```
            ss1.Flush()
```

```
            ss1.Close()
```

```
        Else
```

```
            Dim fs1 As New IO.FileStream(pathFile, FileMode.Append, FileAccess.Write)
```

```
            Dim ss1 As New IO.StreamWriter(fs1)
```

```
            ss1.BaseStream.Seek(0, SeekOrigin.End)
```

```
            ss1.Write("No")
```

```
            ss1.Flush()
```

```
            ss1.Close()
```

```
        End If
```

```
    End If
```

```
    Dim tmpPath As String = pathFile & "urb_"
```

```
    Dim tmpFName As String = selComId & ".txt"
```

```
    Dim newPathFile As String = tmpPath & tmpFName
```

```
    ' Copy header information to a new file for writing
```

```
    If File.Exists(newPathFile) Then
```

```
        File.Delete(newPathFile)
```

```
    End If
```

```
IO.File.Copy(pathFile, newPathFile)

Dim fs As New IO.FileStream(pathFile, FileMode.Append, FileAccess.Write)
Dim ss As New IO.StreamWriter(fs)
Dim kk As Integer
kk = 0

ss.BaseStream.Seek(0, SeekOrigin.End)
For kk = 0 To output.GetUpperBound(0)
    ss.Write(output(kk))
    If (kk = output.GetUpperBound(0)) Then
        Exit For
    Else
        ss.Write(",")
    End If
Next

ss.Flush()
ss.Close()

End Sub
```



Appendix F - Program to write upstream and downstream watersheds for flow routing

+++++ This program was written by Kirthi, Department of Computer Science Engineering, University of Kentucky and modified by Bagya Palanisamy, BAE, University of Kentucky +++++

```
Public Sub initialize()  
    ' Retrieve the recordset from database  
    Dim com_id As String  
    Dim upws As String  
    Dim downws As String  
    Dim minseq As String  
    Dim pcomid As String  
    Dim pDown As String  
    Dim pUp As String  
    Dim pMinseq As String  
    Dim headwat As String  
    Dim headwatflag As String  
    Dim objWrite As Object  
    Dim i As Integer = 0  
    Dim idmap_list As New Dictionary(Of String, String)(0)  
    Dim sFieldName As String  
    Dim pTableSort As ITableSort  
    ReDim objWrite(0 To 5)  
    Dim kk As Integer  
  
    args = New Object(8) {}  
  
    pFact = New ShapefileWorkspaceFactory  
    pWorkspace = pFact.OpenFromFile(shpName, 0)  
    pFWorkspace = pWorkspace  
    pFClass = pFWorkspace.OpenFeatureClass("hucs_nhd")  
    pTable = pFWorkspace.OpenTable("hucs_nhd")  
  
    sFieldName = "downws"  
    pTableSort = New TableSort  
    With pTableSort  
        .Fields = sFieldName  
        .Ascending(sFieldName) = False  
        .Table = pTable  
    End With
```

```

'Sort the table
pTableSort.Sort(Nothing)
pFCursor = pTableSort.Rows
pRow = pFCursor.NextFeature

kk = 3
While (Not pRow Is Nothing)
  args = Nothing
  args = New Object(8) {}

  com_id = pRow.Fields.FindField("com_id")
  pcomid = pRow.Value(com_id)
  downws = pRow.Fields.FindField("downws")
  pDown = pRow.Value(downws)
  headwat = pRow.Fields.FindField("headwat")
  headwatflag = pRow.Value(headwat)

  If Not pDown.Equals("0") Then
    args(3) = pDown
    idmap_list.Add(pDown, pcomid)
    args(6) = pcomid
  Else
    pRow = pFCursor.NextFeature
    Continue While
  End If

  upws = pRow.Fields.FindField("upws")
  pUp = pRow.Value(upws)

  If Not pUp.Equals("0") Then
    args(1) = pUp
    args(2) = pUp
    If idmap_list.ContainsKey(pUp) Then
      args(0) = idmap_list.Item(pUp)
      args(5) = args(0)
    End If
  Else
    pRow = pFCursor.NextFeature

```

```
        Continue While
    End If

    minseq = pRow.Fields.FindField("minseq")
    pMinseq = pRow.Value(minseq)

    If Not pMinseq.Equals("0") Then
        args(4) = pMinseq
        If args(4).Equals(args(2)) Then
            args(4) = "0"
        End If
        If idmap_list.ContainsKey(pMinseq) Then
            args(7) = idmap_list.Item(pMinseq)
            If args(7).Equals(args(5)) Then
                args(7) = "0"
            End If
        End If
    Else
        pRow = pFCursor.NextFeature
        Continue While
    End If

    objWrite(0) = args(5)
    objWrite(1) = args(6)
    objWrite(2) = args(7)

    NHDLList.writeData(filePath & "ChannelNetworkHead.txt", objWrite)

    pRow = pFCursor.NextFeature

End While

End Sub
```

## Appendix G - Programs to automate data retrieval

```
Imports System.Net.WebRequestMethods
Imports System.Net
Imports System.Web
Imports System.IO
Imports System.Text.RegularExpressions
Imports System.Diagnostics.Process
Imports System.Text
Imports System.Convert
Imports Microsoft.VisualBasic
Imports ESRI.ArcGIS.Geodatabase
Imports ESRI.ArcGIS.DataSourcesFile
Imports ESRI.ArcGIS.Carto
Imports System.Math
Imports ESRI.ArcGIS.GeoDatabaseUI
Imports ESRI.ArcGIS.Geoprocessor
Imports ESRI.ArcGIS.ConversionTools
Imports ESRI.ArcGIS.DataSourcesOleDB
Imports ESRI.ArcGIS.DataSourcesGDB

Module Module1
    ' Define Variables
    Public theInPath As String = "C:\bagya\InputFiles\"
    Public request As WebRequest
    Public response As WebResponse
    Public str As String
    Public URL As String
    Public reader As StreamReader
    Public CurrDay, ModifiedDay As String
    Public tmp() As String
    Public tmp1, tmp2, tmp3 As String
    Public FlowToday As Single
    Public FlowStr() As String
    Public PrecipToday As Single
    Public PrecipStr() As String
    Public theMn As String
    Public PrecipDir, StreamFlowDir As String
    Public DateReFormatted As String
    Public InFile As String = "C:\Documents and Settings\bagya\Desktop\Book1.csv"
```

```
<STAThread()> _
Sub Main()
    JoinOutput("Flow.csv")
    Exit Sub
    CurrDay = Today.AddDays(-1) 'DateTime.Today - Get yesterday
    tmp = CurrDay.Split("/")
    theMn = GetMonthName(tmp(0))

    ' For day
    If tmp(1) < 10 Then
        tmp2 = "0" & tmp(1).ToString
    Else
        tmp2 = tmp(1)
    End If

    ' For month
    If tmp(0) < 10 Then
        tmp1 = "0" & tmp(0).ToString
    Else
        tmp1 = tmp(0)
    End If

    DateReFormatted = tmp(2) & tmp1 & tmp2

    PrecipDir = theInPath & "Precipitation\" & tmp(2)
    StreamFlowDir = theInPath & "StreamFlow\" & tmp(2)

    ' Create a year directory if it does not exist
    If Not Directory.Exists(PrecipDir) Then
        Directory.CreateDirectory(PrecipDir)
    End If

    If Not Directory.Exists(StreamFlowDir) Then
        Directory.CreateDirectory(StreamFlowDir)
    End If

    InitializeFPU(tmp(2) & "-" & tmp1 & "-" & tmp2)
```

```

GetPrecipAgWeather(tmp(2), theMn, tmp2)

' Write the current day value in a text file
WriteToPrecip(PrecipDir, StreamFlowDir, "London", "03281100")

' Process.Start("C:\bagya\Projects_NET\TopModel2\TopModel\bin\Debug\TopModel.exe")

Console.WriteLine("Streamflow at Goose Creek is " & FlowToday & "cfs on day " & CurrDay)
Console.WriteLine("Finished processing")

End Sub
Public Sub InitializeFPU(ByVal theDay As String)
    'Address of URL

    'URL = "http://nwis.waterdata.usgs.gov/ky/nwis/dv?cb_00060=on&format=rdb&begin_date=2010-03-
27&end_date=2010-03-27&site_no=03281100&referred_module=sw"
    URL = "http://nwis.waterdata.usgs.gov/ky/nwis/dv?cb_00060=on&format=rdb&begin_date=" & theDay
& "&end_date=" & theDay & "&site_no=03281100&referred_module=sw"
    request = WebRequest.Create(URL)
    response = request.GetResponse()
    reader = New StreamReader(response.GetResponseStream())
    str = reader.ReadLine()

    Do Until Left(str, 4) = "USGS"
        str = reader.ReadLine()
    Loop

    FlowStr = SplitWords(str)
    FlowToday = FlowStr(5)

End Sub
Public Function GetPrecipAgWeather(ByVal theYr As String, ByVal mn As String, ByVal theDay As
String) As Single
    URL = "http://www.gw.ca.uky.edu/cgi-bin/wxlist_int2?CityName=London&Begin_Month=" & mn &
"&Begin_Day=" & theDay & "&Begin_Year=" & theYr & "&End_Month=" & mn & "&End_Day=" & theDay &
"&End_Year=" & theYr & "&Preference=2&View_Save=View&Mailid="

```



```

        'URL = http://wwwagwx.ca.uky.edu/cgi-
bin/wxlist_int2?CityName=London&Begin_Month=Apr&Begin_Day=01&Begin_Year=2010&End_Month=Apr&End_Day=01
&End_Year=2010&Preference=2&View_Save=View&Mailid=
        request = WebRequest.Create(URL)
        response = request.GetResponse()
        reader = New StreamReader(response.GetResponseStream())
        str = reader.ReadLine()

        Do Until Left(str, 6) = "London"
            str = reader.ReadLine()
        Loop

        PrecipStr = SplitWords(str)
        PrecipToday = PrecipStr(4)
        Return PrecipToday
    End Function
    Public Function GetMonthName(ByVal monthNum As Integer) As String
        Dim strDate As New DateTime(1, monthNum, 1)
        Return strDate.ToString("MMM")
    End Function
    Public Function SplitWords(ByVal s As String) As String()
        Return Regex.Split(s, "\W+")
    End Function
    Public Sub WriteToPrecip(ByVal precipdir As String, ByVal streamflowdir As String, ByVal StnIdPcp
As String, ByVal StnIDFlow As String)

        Dim PrecipOutput As String = precipdir & "\Precip_" & DateReFormatted & ".txt"
        Dim StreamFlowOutput As String = streamflowdir & "\Flow_" & DateReFormatted & ".txt"
        Dim sw As StreamWriter

        ' Create precipitation output file if it doesn't exist

        If IO.File.Exists(PrecipOutput) Then
            IO.File.Delete(PrecipOutput)
        End If

        sw = IO.File.CreateText(PrecipOutput)
        sw.WriteLine("Station,Precipitation-mm")

```

```

sw.WriteLine(StnIdPcp & "," & PrecipToday * 25.4)
sw.Close()

' Create streamflow output file if it doesn't exist

If IO.File.Exists(StreamFlowOutput) Then
    IO.File.Delete(StreamFlowOutput)
End If

sw = IO.File.CreateText(StreamFlowOutput)
sw.WriteLine("Station,Flow-cfs")
sw.WriteLine(StnIDFlow & "," & FlowToday)
sw.Close()

```

End Sub

Public Sub JoinOutput(ByVal fName As String)

```

' Export output to text file
Dim twiPath As String = "C:\bagya\NHD_Downloaded\"
Dim wsf As IWorkspaceFactory
wsf = New TextFileWorkspaceFactory
Dim wrkspc As IWorkspace
wrkspc = wsf.OpenFromFile(twiPath, 0)
Dim fws As IFeatureWorkspace
fws = wrkspc
Dim xlTable As ITable
xlTable = fws.OpenTable(fName)
Dim xl2Table As TableToTable
xl2Table = New TableToTable
xl2Table.in_rows = xlTable
xl2Table.out_name = "flowTest"
xl2Table.out_path = twiPath
Dim gp As Geoprocessor
gp = New Geoprocessor
gp.Execute(xl2Table, Nothing) ' Add reference to esri.arcgis.system to avoid error in this

```

line

```

' Joining flow with the HUCs
Dim pSFact As ShapefileWorkspaceFactory

```

```

Dim pWs As IWorkspace
Dim pFClass As IFeatureClass
pSFact = New ShapefileWorkspaceFactory
pWs = pSFact.OpenFromFile(twiPath, 0)
Dim pFWs As IFeatureWorkspace
pFWs = pWs
Dim pZoneTable As ITable
pZoneTable = pFWs.OpenTable("flowTest")
pFClass = pFWs.OpenFeatureClass("hucs_nhd")
RelQryTabExample(pFClass, "com_id", pZoneTable, "id")
End Sub
Private Sub RelQryTabExample(ByVal pFCls As IFeatureClass, ByVal strFClsFld As String, ByVal
pTable As ITable, ByVal strTabFld As String)

' ++ Create the MemoryRelationshipClass that defines what is to be joined
Dim pMemRelClassFact As IMemoryRelationshipClassFactory
pMemRelClassFact = New MemoryRelationshipClassFactory
Dim pRelClass As IRelationshipClass
pRelClass = pMemRelClassFact.Open("JoinFlow", pFCls, _
strFClsFld, pTable, strTabFld, "forward", "backward",
esriRelCardinality.esriRelCardinalityOneToOne)

' ++ Perform the join
Dim pRelQueryTableFact As IRelQueryTableFactory
Dim pRelQueryTab As ITable
pRelQueryTableFact = New RelQueryTableFactory
pRelQueryTab = pRelQueryTableFact.Open(pRelClass, True, Nothing, Nothing, "", True, True)

' ++ Print the fields
Dim pCursor As ICursor
pCursor = pRelQueryTab.Search(Nothing, True)

Dim pField As IField
Dim pFields As IFields
Dim intI As Integer, intJ As Integer

pFields = pCursor.Fields
intI = pFields.FieldCount - 1

```

```
        For intJ = 0 To intI
            pField = pFields.Field(intJ)
            Debug.Print(pField.Name)
        Next intJ
    End Sub
End Module
```

Appendix H - Program for ArcIMS flow value display

++++ This program describes joining flow dataset with watersheds for internet publication ++++++

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<ARCXML version="1.1">
```

```
  <CONFIG>
```

```
    <ENVIRONMENT>
```

```
      <LOCALE country="US" language="en" variant="" />
```

```
      <UIFONT color="0,0,0" name="SansSerif" size="12" style="regular" />
```

```
      <SCREEN dpi="96" />
```

```
    </ENVIRONMENT>
```

```
    <MAP>
```

```
      <PROPERTIES>
```

```
        <ENVELOPE minx="532224.566168397" miny="-62669.53407239169" maxx="559867.0320038219" maxy="-37340.16427300125" name="Initial_Extent" />
```

```
        <MAPUNITS units="decimal_degrees" />
```

```
      </PROPERTIES>
```

```
      <WORKSPACES>
```

```
        <SHAPEWORKSPACE name="shp_ws-0" directory="C:\bagya\NHD_Downloaded" />
```

```
      </WORKSPACES>
```

```
      <LAYER type="featureclass" name="NHDPlus Basins of Goose Creek" visible="true" id="0">
```

```
        <DATASET name="hucs_nhd_Project" type="polygon" workspace="shp_ws-0" />
```

```
      <SPATIALQUERY jointables="HUC_20060519" joinexpression="To=[hucs_nhd_Project.com_id],  
From=[HUC_20060519.id], Type=[scan]"> </SPATIALQUERY>
```

```
      <SIMPLERENDERER>
```

```
        <SIMPLEPOLYGONSMBOL boundarytransparency="1.0" filltransparency="0.0"  
boundarycaptype="round" />
```

```
      </SIMPLERENDERER>
```

```
    </MAP>
```

```
  </CONFIG>
```

```
</ARCXML>
```







```

Public Function rchrouting (ByVal sumq As System.Array) As System.Array
    yrLoop = 1
    t = 0
    firstTerm = 1.15
    InReach = New List(Of String)
    OutReach = New List(Of String)
    ReDim rchstore(totDayz, UBound(fromReach))

    For iii As Integer = 0 To UBound(fromReach)
        InReach.Add(fromReach(iii))
        OutReach.Add(toReach(iii))
    Next

    For theYears = simStartYear To simEndYear
        simEndDay = EndDayz(yrLoop - 1)
        For dayLoop = simStartDay To simEndDay

            ' Emptying arrays
            ReDim flowIn(UBound(fromReach))
            ReDim flowOut(UBound(fromReach))
            ReDim flowAdd(UBound(fromReach))
            ReDim totalInflow(UBound(fromReach))
            GetReachIndex = New List(Of Integer)

            ' Start routing in the most upstream reaches

            flowIn(0) = sumq(yrLoop, dayLoop, FromReachIndex(0))
            flowOut(0) = sumq(yrLoop, dayLoop, ToReachIndex(0))
            If addReach(0) > 0 Then
                flowAdd(0) = sumq(yrLoop, dayLoop, AddReachIndex(0))
            Else
                flowAdd(0) = 0.0
            End If
            If t = 0 Then
                totalInflow(0) = flowIn(0) + flowAdd(0)
            End If
        Next
    Next
End Function

```

```

Else
    totalInflow(0) = flowIn(0) + flowAdd(0) + rchstore(t - 1, 0)
End If
Call CalOutFlow(0)

' Now proceed with the downstream reaches
For i = 1 To UBound(fromReach)

    flowIn(i) = sumq(yrLoop, dayLoop, FromReachIndex(i))
    flowOut(i) = sumq(yrLoop, dayLoop, ToReachIndex(i))

    totalInflow(i) = flowIn(i)

    If t = 0 Then
        'Determine whether there is any inflow from immediate upstream
        If fromReach(i) = toReach(i - 1) Then
            totalInflow(i) = totalInflow(i) + qreleased(yrLoop, dayLoop, i - 1)
        Else
            ' Determine if there is any other upstream mainchannel flow
            If OutReach.Contains(InReach(i)) Then
                GetReachIndex.Insert(0, OutReach.IndexOf(InReach(i)))
                totalInflow(i) = totalInflow(i) + qreleased(yrLoop, dayLoop,
GetReachIndex(0))

                End If
            End If

            ' Determine if there is any tributary joining the current main channel
            If addReach(i) > 0 Then
                flowAdd(i) = sumq(yrLoop, dayLoop, AddReachIndex(i))
                ' Checking if there is anyother upstream tributary
                If OutReach.Contains(addReach(i)) Then
                    GetReachIndex.Insert(0, OutReach.IndexOf(addReach(i)))
                    totalInflow(i) = totalInflow(i) + qreleased(yrLoop, dayLoop,
GetReachIndex(0))

                    End If
                totalInflow(i) = totalInflow(i) + flowAdd(i)
            End If
        End If
    End If
End For

```

```

Else
    'Determine whether there is any inflow from immediate upstream
    If fromReach(i) = toReach(i - 1) Then
        totalInflow(i) = qreleased(yrLoop, dayLoop, i - 1) + totalInflow(i)
    Else
        If OutReach.Contains(InReach(i)) Then
            GetReachIndex.Insert(0, OutReach.IndexOf(InReach(i)))
            totalInflow(i) = totalInflow(i) + qreleased(yrLoop, dayLoop,
GetReachIndex(0))
        End If
    End If

    ' Determine if there is any tributary joining the current main channel
    If addReach(i) > 0 Then
        flowAdd(i) = sumq(yrLoop, dayLoop, AddReachIndex(0))
        ' Checking if there is anyother upstream tributary
        If OutReach.Contains(addReach(i)) Then
            GetReachIndex.Insert(0, OutReach.IndexOf(addReach(i)))
            totalInflow(i) = totalInflow(i) + qreleased(yrLoop, dayLoop,
GetReachIndex(0))
        End If
        totalInflow(i) = totalInflow(i) + flowAdd(i)
    End If

    totalInflow(i) = totalInflow(i) + rchstore(t - 1, i)

End If

' Calculate outflow
CalOutFlow(i)

Next

t = t + 1

qOutDay.Add((qreleased(yrLoop, dayLoop, i - 1)) * 35.3)

```

```

        Next
        yrLoop = yrLoop + 1
    Next
    Return qreleased
End Function

```

```

Public Function CalOutFlow(ByVal k As Integer, ByVal iii As Integer, ByVal flwInTot As Single,
ByVal areaDown As Single, ByVal areaUp As Single) As Single

```

```

    secondTerm1 = Log(areaDown / areaUp)
    logATF1 = firstterm * secondTerm1
    atf1 = Exp(logATF1)

    If atf1 > 1 Then
        storeATF1 = atf1 - 1
        atf1 = 1
    End If

    rchstore(iii, k) = flwInTot * storeATF1
    If rchstore(iii, k) > flwInTot Then
        rchstore(iii, k) = flwInTot
    End If
    flwInTot = Max(0, flwInTot - rchstore(iii, k))

    ' Calculating outflow
    tmpVar = (atf1 * flwInTot)

    Return tmpVar
End Function

```

## References

- Abraham, B., 1985. Seasonal time series and transfer function modeling. *Journal of Business and Economic statistics* 3(4): 356-361.
- Acreman, M.C., and Sinclair, C.D., 1986. Classification of drainage basins according to their physical characteristics; An application for flood frequency analysis in Scotland. *Journal of Hydrology* 84(3-4): 365-380.
- Agricultural Weather Center, 2010. College of Agriculture, university of Kentucky. Electronic source: [http://www.wagwx.ca.uky.edu/cgi-bin/ky\\_clim\\_data\\_www.pl](http://www.wagwx.ca.uky.edu/cgi-bin/ky_clim_data_www.pl)
- Andreassian, V., Perrin, C., Michel, C., Usart-Sanchez, I.U., Lavabre, J., 2001. Impact of imperfect rainfall knowledge on the efficiency and the parameters of watershed models. *Journal of Hydrology* 250: 206-223.
- Feldman, A.D., 2000. Hydrologic Modeling System HEC-HMS Technical Reference Manual, U.S. Army Corps of Engineers. Electronic source: [http://www.hec.usace.army.mil/software/hec-hms/documentation/CPD-74B\\_2000Mar.pdf](http://www.hec.usace.army.mil/software/hec-hms/documentation/CPD-74B_2000Mar.pdf)
- Beven, K.J., Kirkby, M.J., Schofield, N., and Tagg, A.F., 1984. Testing a physically-based flood forecasting model (TOPMODEL) for three U.K. catchments. *Journal of Hydrology* 69: 119-143.
- Blank, D., Delleur, J.W., and Giorgini, A. 1971. Oscillatory kernel functions in linear hydrologic models. *Water Resources Research* 7(5): 1102-1117.
- Burn, D.H., and Goulter, I.C., 1991. An approach to the rationalization of streamflow data collection networks. *Journal of Hydrology* 122: 71-91.
- Chapman, T.G., 1996. Common unitgraphs for sets of runoff event. Part 1: Unitgraph identification from streamflow data. *Hydrological Processes* 10: 773-782.
- Chahinian, N., Moussa, R., Andrieux, P., and Voltz, M., 2005. comparison of infiltration models to simulate flood events at the field scale. *Journal of Hydrology* 306: 191-214.
- Chen, C., 1999. Linear system theory and design Third Edition. In Chapter 2. Mathematical descriptions of systems. New York Oxford: Oxford University Press, pp. 11.
- Chow V.T., Maidment, D.R., and Mays, L.W., 1988. Applied Hydrology. McGraw Hill: New York, NY, Pp. 257-260.
- Deininger, R.A., 1969. Linear programming for hydrologic analyses. *Water Resources Research* 5(5): 1105-1109.

- Diskin, M.H., and Boneh, A., 1975. Determination of an optimal IUH for linear, time invariant systems from multi-storm records. *Journal of Hydrology* 24: 57-76.
- Duan, Q., S. Sorooshian, and V.K. Gupta, 1992. Effective and Efficient Global Optimization for Conceptual Rainfall-runoff Models. *Water Resources Research*, 28(4): 1015-1031.
- Duan, Q., Sorooshian, S., and Gupta, V.K., 1994. Optimal use of the SCE-Ua global optimization method for calibrating watershed models. *Journal of Hydrology* 158(3-4): 265-284.
- Duband, D., Obled, C., rodriguex, J., 1993. Unit hydrograph revisited: an alternative approach to UH and effective precipitation identification. *Journal of Hydrology* 150(1): 115-150.
- Durand, P., Robson, A., and Neal, C., 1992. Modelling the hydrology of submediterranean montane catchments (Mont-Lozere, France) using TOPMODEL: initial results. *Journal of Hydrology* 139: 1-14.
- Eagleson, P.S., Mejia-R, R. and March, F., 1966. Computation of optimum realizable unit hydrographs. *Water Resources Research* 2(4): 755-764.
- Fernandez, G.P., Chescheir, G.M., Skaggs, R.W., and Amatya, D.M., 2006. DRAINMOD-GIS: A lumped parameter watershed scale drainage and water quality model. *Agricultural Water Management* 81(1-2): 77-97.
- Franchini, M., Galeati, G., and Berra, S., 1998. Global optimization techniques for the calibration of conceptual rainfall-runoff models. *Journal of Hydrologic Science*, 43(3), 443-458.
- Gallart, F., Llorens, P., and Latron, J., 1994. Studying the role of old agricultural terraces on runoff generation in a small Mediterranean mountainous basin. *Journal of Hydrology* 159: 291-303.
- Heuvelmans, g., Muys, B., and Feyen, J., 2006. Regionalization of the parameters of a hydrological model: Comparison of linear regression models with artificial neural nets. *Journal of Hydrology* 319: 245-265.
- Hoppner, F., Klawonn, F., Kruse, R., and Runkler, T., (1999). Fuzzy Cluster Analysis: Methods for Classification Data Analysis and Image Recognition. Wiley, New York
- Huang, J.-C., Lee, T.-Y. and Kao, S.-J., 2008. Simulating typhoon-induced storm hydrographs in subtropical mountainous watershed: an integrated 3-layer TOPMODEL. *Hydrology and Earth System Sciences Discussions*, 5: 1101-1135.
- Irrigation water management: Irrigation water needs, 1986. *In: Irrigation water management training manuals - 3*, S2022/E.
- Jain, a., Srinivasalu, S., and Bhattacharjya, R. K., 2005. Determination of an optimal unit pulse response function using real-coded genetic algorithm. *Journal of Hydrology*, 303: 199-214.

- Juki'c, D., and Juki'c, V.D. 2003. Composite transfer functions for karst aquifers. *Journal of Hydrology* 274: 80-94.
- Kavetski, D., Kuczera, G., and S.W. Franks., 2003. Semidistributed hydrological modeling: A “saturation path” perspective on TOPMODEL and VIC. *Water Resources Research* 39(9): 1246, doi:10.1029/2003WR002122, 2003.
- Kentucky Water Resources Research Institute, 2002. Kentucky River Basin management plans.
- Kennen, J.G., Kauffman, L.J., Ayers, M.A., Wolock, D.M., and Colarullo, S.J., 2008. Use of an integrated flow model to estimate ecologically relevant hydrologic characteristics at stream biomonitoring sites. *Ecological Modelling*, 211: 57-76
- Kinner, D.A., and Stallard, R.F., 2004. Identifying storm flow pathways in a rainforest catchment using hydrological and geochemical modeling. *Hydrological Processes*, 18: 2851-2875.
- Kuczera, G., 1997: Efficient subspace probabilistic parameter optimization for catchment models. *Water Resources Research* **33**, 177–185.
- Labat, D., Ababou, R., and Mangin, A., 2000. Rainfall-runoff relations for karstic springs. Part I: convolution and spectral analyses. *Journal of Hydrology* 238: 123-148.
- Lai, P., 1981, ‘Transfer Function Modeling: Relationship Between Time Series Variables’, in Concepts and Techniques in Modern Geography, Monograph No. 22, Springer-Verlag, New York, pp. 1–47.
- Laine, R.j., 1970. Deriving the unitgraph without using rainfall data. *Journal of Hydrology* 10: 379-390.
- Lattin, J., Carroll, J.D, Green, P.E., (2003). Discriminant Analysis: *In Analyzing Multivariate Data*. Pg: 426-466.
- Lin, G., and Chen, L., 2006. Identification of homogeneous regions for regional frequency analysis using the self-organizing map. *Journal of Hydrology* 324: 1-9.
- Long, A.J., and Derickson, R.G., 1999. Linear systems analysis in a karst aquifer. *Journal of Hydrology*, 219: 206-217.
- Mays, L.W., and Taur, C.K., 1982. Unit hydrographs via nonlinear programming. *Water Resources Research* 18(4): 744-752.
- McDowell, R.C., 1986. The geology of Kentucky - A text to accompany the geologic map of Kentucky. U.S. Geological survey professional paper 1151-H.
- McIntyre, N., Lee, H., and Wheeler, H., 2005. Ensemble predictions of runoff in ungauged catchments. *Water Resources research* 41, W12434, doi:10.1029/2005WR004289.
- McGuire, K.J., and McDonnell, J.J., 2006. A Review and evaluation of catchment transit time

- modeling. *Journal of Hydrology*, 330(3-4): 543-563.
- Moussa, R., 1996. Analytical Hayami solution for the diffusive wave flood routing problem with lateral inflow. *Hydrological Process*, 10: 1209-1227.
- Neitsch, S.L., Arnold, J.G., Kiniry, J.R., Williams, J.R., 2005. Soil water assessment tool theoretical document, version 2005. Grassland, Soil and Water Research Laboratory, Agricultural Research Service, 808 East Blackland Road, Temple, TX. <http://www.brc.tamus.edu/swat>
- Newman, S.P., and Marsily De, G., 1976. Identification of linear systems response by parametric programming. *Water Resources Research* 12(2): 253-262.
- NRCS (Natural Resources Conservation Services), 2009. Soil Survey Geographic (SSURGO) database. Electronic resource: <http://soils.usda.gov/survey/geography/ssurgo/>
- O'Donnell, T., 1960. Instantaneous unit hydrograph derivation by harmonic analysis. *International Association of Hydrological Sciences Publication* 51: 546-557.
- Olivera, F., and Maidment, M. 1999. Geographic Information Systems (GIS) – based spatially distributed model for runoff routing. *Water Resources Research* 35(4): 1155-1164.
- Ouarda, T.B.M.J., Ba, K.M., Diaz-Delgado, c., Carsteanu, A., Chokmani, K., Gingras, H., Quentin, E., Trujillo, E., Bobee, B., 2008. Intercomparison of regional flood frequency estimation methods at ungauged sites for a Mexico case study. *Journal of Hydrology* 348: 40-58.
- Pan, F., C.D.Peters-Lidard, M.J. Sale, and A.W. King, 2004. A comparison of geographical information systems-based algorithms for computing the TOPMODEL topographic index. *Water Resources Research* 40, W06303, doi:10.1029/2004WR003069.
- Ponce, V.M., 1989. Engineering Hydrology, Principles and Practices, Prentice Hall, pp: 48-51.
- Pradhan, N.R., Tachikawa, Y., and Takara, K., (2006). A downscaling method of topographic index distribution for matching the scales of model application and parameter identification. *Hydrological Processes*, 20: 1385-1405.
- Rao, A.R., and Srinivas, V.V., 2006a. Regionalization of watersheds by hybrid-cluster analysis. *Journal of Hydrology* 318: 37-56.
- Rao, A.R., and Srinivas, V.V., 2006b. Regionalization of watersheds by fuzzy cluster analysis. *Journal of Hydrology* 318: 57-79.
- Renata, R., 1997. A Matlab implementation of TOPMODEL. *Hydrological Processes*, 11: 1115-1129.
- Robson, A.J., Whitehead, P.G., and Johnson, R.C., 1993. An application of a physically based



- semi-distributed model to the Balquhider catchments. *Journal of Hydrology* 145: 357-370.
- Sanborn, S.C., and Bledsoe, B.P., 2006. Predicting streamflow regime metrics for ungauged streams in Colorado, Washington, and Oregon. *Journal of Hydrology* 325: 241-261.
- Sanz, D.B., and Jalon, D.G., 2005. Characterization of streamflow regimes in central Spain, based on relevant hydrobiological parameters. *Journal of Hydrology* 310: 266-279.
- Sherman, L.K., 1932. Streamflow from rainfall by the unit-graph methods. *Engineering News Record*, 108: 501-505.
- Shrestha, S., Bastola, S., Babel, M.S., Dulal, K.N., Magome, j., hapuarachchi, H.A.P., Kazama, F., Ishidaira, H., and Takeuchi, k., 2007. The assessment of spatial and temporal transferability of a physically based distributed hydrological model parameters in different physiographic regions of Nepal. *Journal of Hydrology* 347: 153-172.
- Shu, C., and Ouarda, T.B.M.J., 2007. Flood frequency analysis at ungauged sites using artificial neural networks in canonical correlation analysis physiographic space. *Water Resources Research* 43:
- Srinivas, V.V., Tripathi, S., Rao, A.R., and Govindaraju R.S., 2008. Regional flood frequency analysis by combining self-organizing feature map and fuzzy clustering. *Journal of Hydrology* 348: 148-166.
- Taschner, S., Ludwig, R., and Mauser, W., 2001. Multi-scenario flood modeling in a mountain watershed using data from a NWP model, rain radar and rain gauges. *Physics and Chemistry of the Earth (B)*, 26(7-8): 509-515.
- USGS (United States Geological Survey), 2010. Electronic source:  
[http://waterdata.usgs.gov/ky/nwis/dv/?site\\_no=03281100&agency\\_cd=USGS&referred\\_module=sw](http://waterdata.usgs.gov/ky/nwis/dv/?site_no=03281100&agency_cd=USGS&referred_module=sw)
- Valeo, C., and Moin, S.M.A., 2000. Variable source area modeling in urbanizing watersheds. *Journal of Hydrology*, 228: 68-81.
- Valeo, C., and Moin, Syed M.A., 2001. Hortonian and variable source area modeling in urbanizing basins. *Journal of Hydrologic Engineering* 6(4): 328-335.
- Wang, J., Endreny, T.A., and Hassett, J.M., 2005. A flexible modeling package for topographically based watershed hydrology. *Journal of Hydrology* 314: 78-91.
- Wang, G., Hapuarachchi, P., Ishidaira, H., Kiem, A.S., and Takeuchi, K., 2009. Estimation of soil erosion and sediment yield during individual rainstorms at catchment scale. *Water Resources Management*, 23: 1447-1465.
- Wolock, D.M., Hornberger, G.M., and Musgrove, T.J., 1990. Topographic effects on flow path

and surface water chemistry of the Llyn Brianne catchments in Wales. *Journal of Hydrology* 115: 243-259.

## VITA

**Name:** Bakkiyalakshmi Palanisamy  
**Date:** April 20, 1980  
**Place of Birth:** Paramakudi, India

### Education

TEXAS A&M UNIVERSITY – *College Station, Texas* 2006  
*Master of Science Degree in Agricultural Engineering*  
United States Geological Survey (USGS) Research Grant/Student Research Week Award

TAMIL NADU AGRICULTURAL UNIVERSITY – *Coimbatore, Tamil Nadu, India* 2001  
*Bachelor of Engineering Degree in Agricultural Engineering*

### Professional Experience & Research Accomplishments

**Engineering Associate**  
CITY OF AUSTIN – *Austin, Texas* 2005 – 2006

**Executive GIS Intern**  
SOUTH DAKOTA DEPARTMENT OF AGRICULTURE – *Pierre, South Dakota* 2005

**Junior Research Fellow**  
TAMIL NADU AGRICULTURAL UNIVERSITY – *Coimbatore, Tamil Nadu, India* 2002

### Publications & Presentations

#### Peer Reviewed Publications

Palanisamy, B., and Workman, S.R, (2010). A streamflow prediction technique in ungauged basins (To be submitted in Journal of Hydrologic Engineering)

Palanisamy, B., and Workman, S.R, (2010). Flow routing using transfer function derived from measured streamflow records (To be submitted in Journal of Spatial Hydrology)

Palanisamy, B., and Workman, S.R, (2010). Development of flow transferring characteristics using measured streamflow records (To be submitted in Journal of Hydrology)

Palanisamy, B., and Workman, S.R, (2009). Karst hydrology modeling using Soil and Water Assessment Tool (SWAT), Paper submitted to Transactions of the ASABE

#### Presentations

Palanisamy, B., and Workman, S.R., (2009). Regression Analysis of Channel Parameters for Runoff Calibration, Oral Presentation, American Society of Agricultural and Biological Engineers Annual Meeting

Palanisamy, B., and Workman, S.R., (2008). Time series analysis of Kentucky River Basin, Oral presentation, American Society of Agricultural and Biological Engineers Annual Meeting

Palanisamy, B., and Workman, S.R., (2008). Karst hydrology modeling using Soil and Water Assessment Tool (SWAT), Oral presentations, American and Society of Agricultural and Biological Engineers Annual Meeting

Palanisamy, B., and Workman, S.R., (2008). Karst hydrology modeling using Soil and Water Assessment Tool (SWAT), Oral presentations, American Water Resources Association-GIS Spring Specialty conference

Palanisamy, B., and Workman, S.R., (2007). Developmental strategies for a server based hydrologic model and runoff forecasting system for Kentucky, USA, Oral presentation, American Society of Agricultural and Biological Engineers Annual Meeting

Palanisamy, B., and Srinivasan, R., (2004). Mapping Potential Recharge Zones using NOAA-AVHRR Derived Vegetation Index and Spatially Distributed NEXRAD Rainfall in Texas, USA, Poster Presentation, American Water Resources Association-GIS Spring Specialty Conference

Palanisamy, B., and Srinivasan, R., (2004). Mapping Potential Recharge Zones using NOAA-AVHRR Derived Vegetation Index and Spatially Distributed NEXRAD Rainfall in Texas, USA, Poster Presentation, Texas Water Summit Poster Competition

Palanisamy, B., and Srinivasan, R., (2004). Mapping Potential Recharge Zones using NOAA-AVHRR Derived Vegetation Index and Spatially Distributed NEXRAD Rainfall in Texas, USA, Poster Presentation, Student Pathways Research Symposium

### **Affiliations**

Member, American Society of Agricultural and Biological Engineers (2003-Present); Member, American Geophysical Union (2007-Present); Member, American Society of Civil Engineers (2008-2009); Member, American Water Resources Association (2008-Present)