THE UNIVERSITY OF WESTERN ONTARIO
DEPARTMENT OF CIVIL AND
ENVIRONMENTAL ENGINEERING

Water Resources Research Report

Flood frequency analysis using copula with mixed marginal distributions

By:
Subhankar Karmakar
and
Slobodan P. Simonovic

Report No: 055
Date: August 2007

ISSN: (print) 1913-3200; (online) 1913-3219;
ISBN: (print) 978-0-7714-2658-2; (online) 978-0-7714-2659-9;

Western

Flood frequency analysis using copula with mixed marginal distributions

Project Report

August 2007

Prepared by

Subhankar Karmakar

and

Slobodan P. Simonovic

Contents

List of Figures

List of Tables

Appendixes

Abstract

In flood frequency analysis, a flood event is mainly characterized by peak flow, volume and duration. These three variables or characteristics of flood are random in nature and mutually correlated. In this article, a methodology is developed to derive bivariate joint distributions of the flood characteristics using the concept of copula considering a set of parametric and nonparametric marginal distributions for peak flow, volume and duration to mathematically model the correlated nature among them. A set of parametric distribution functions, and nonparametric methods based on kernel density estimation and orthonormal series are used to determine the marginal distribution functions for peak flow, volume and duration. In conventional method of flood frequency analysis, the marginal distribution functions of peak flow, volume and duration are assumed to follow some specific parametric distribution function. The concept of copula relaxes the restriction of traditional flood frequency analysis by selecting marginals from different families of probability distribution functions for flood characteristics. The present work performs a better selection of marginal distribution functions for flood characteristics by parametric and nonparametric estimation procedures, and demonstrates how the concept of copula may be used for establishing joint distribution function with mixed marginal distributions. The methodology is demonstrated with seventy years streamflow data of Red River at Grand Forks of North Dakota, US. The research work reported here is already submitted by the authors as a manuscript for review to Water Resources Research, AGU.

I. INTRODUCTION

Most hydrologic design, planning and management problems require a detailed knowledge of flood event characteristics, i.e., flood peak flow, volume and duration. These are random in nature and mutually correlated. Flood frequency analysis defines the severity of a flood event by summarizing the characteristics of flood, and by finding out their mutual dependence structure. A number of methodologies have been developed to perform univariate (Kite, 1978; Cunnane, 1987; Rao and Hamed, 2000) and multivariate flood frequency analysis (Ashkar and Rousselle, 1982; Krstanovic and Singh, 1987; Sackl and Bergmann, 1987; Singh and Singh, 1991; Yue et al., 1999, 2001) but with many restrictive assumptions (Zhang and Singh, 2006). In hydrologic planning and design for flood management, it is not enough to know information about flood peak flow only, but it is also necessary to statistically value flood volume and duration. In order to obtain the information through the application of multivariate statistical analyses, joint Cumulative Distribution Function (CDF) and Probability Density Function (PDF) of flood characteristics are needed.

In conventional method of flood frequency analysis, the marginal distribution functions of peak flow, volume and duration are assumed to follow some specific family of parametric distribution function, for example - gamma, lognormal, exponential, extreme value distribution, etc. Ashkar and Rousselle (1982), Sackl and Bergmann (1987), Yue et al. (1999), Yue (2001) are some of the existing research studies on flood frequency analysis considering a specific family of parametric distribution (Gumbel and gamma distribution, respectively) for all the marginals. Krstanovic and Singh (1987) derived bivariate Gaussian and exponential distributions for frequency analysis of peak and volume. A drawback of this work is that, the same family of marginal distributions is assumed for all three flood characteristics. For example, in Yue et al. (1999), a bivariate Gumbel or Gumbel mixed distribution for combinations of peak flow - volume, and volume - duration, is established assuming Gumbel marginal distributions for peak flow, volume and duration. In some other literature bivariate normal (Sackl and Bergmann, 1987), bivariate exponential (Singh and Singh, 1991), bivariate gamma (Yue, 2001), bivariate lognormal (Yue, 2000) distribution functions are used for flood frequency analysis.

In most real case studies, the best fitted marginal distribution for peak flow, volume and duration need not be from the same family of probability distribution function. Following this observation, the concept of copula (Sklar, 1959; Nelsen, 1999) has been used recently (Favre et al., 2004; Zhang and Singh, 2006; Grimaldi and Serinaldi, 2006; Zhang and Singh, 2007) in flood frequency analysis to model the dependence structure among peak flow, volume and duration independent of the types of marginal distributions they follow. Modeling joint distribution using copula relaxes the restriction of traditional flood frequency analysis by selecting marginals from different families of probability distribution functions for flood characteristics. It is found from Zhang and Singh (2006), and Grimaldi and Serinaldi (2006) that copula based flood frequency analysis performs better than conventional flood frequency analysis as joint distribution based on copula fits the empirical joint distribution (i.e., from observed data using plotting position formula) better than that established by standard joint parametric distribution. Numerous successful applications of copula modeling have been made, most notably in survival analysis, actuarial science, and finance, but history of copula modeling in flood frequency analysis is short. Favre et al. (2004) use bivariate copulas to describe the dependence between peak flow and volume. De Michele et al. (2005) apply bivariate Archimedean copulas to simulate pairs of flood peak–volume to be used to build synthetic flood hydrographs. Zhang and Singh (2006) exploit Archimedean copulas to build bivariate joint distributions of peak flow and volume, and volume and duration. Genest and Favre (2007) present the successive steps required to build a copula model for hydrological purposes. They have demonstrated performance of extreme value copulas for modeling dependence structure of flood peak flow and volume. A limited number of studies on trivariate flood frequency analysis have been reported in the hydrologic literature. Grimaldi and Serinaldi (2006) build trivariate joint distribution of flood event variables using the fully nested or asymmetric Archimedean copula functions. Zhang and Singh (2007) use Gumbel-Hougaard family of Archimedean copula to determine trivariate distribution of peak flow, volume and duration, and conditional return periods. Serinaldi and Grimaldi (2007) describe the inference procedure to carry out a trivariate flood frequency analysis via asymmetric Archimedean copulas.

The frequency analysis is primarily based on the estimation of the PDF. The parametric approaches for estimating the PDF must assume that the data are drawn from a known parametric family of distributions. However, many studies on frequency analysis indicate that there is no universally accepted distribution for representing the hydrologic variables (Adamowski, 1985, 1996; Silverman, 1986; Yue et al., 1999; Smakhtin, 2001). It is evident that the parametric method, which depends on prior knowledge of the particular distribution function, has its limitations (Cunnane, 1985, Adamowski, 1989) and, as pointed out by Dooge (1986), “no amount of statistical refinement can overcome the disadvantage of not knowing the frequency distribution involved”. To overcome some of the limitations of parametric method, nonparametric density function estimations have been explored in hydrologic frequency estimation

(Lall, 1995). Nonparametric method does not require the assumption of any particular form of density function. As the choice of a distribution is bypassed altogether, the nonparametric method is uniform and its adoption would eliminate the need for any arbitrary imposition of uniformity in flood frequency analysis (Adamowski, 1989). Using the weighted moving averages of the records from a small neighborhood of the point of estimation, nonparametric function estimations have the advantage that they always reproduce the attributes represented by the sample (Lall, 1995; Sharma, 2000; Kim et al., 2003). Number of existing studies on nonparametric methods for hydrologic frequency analysis is not large. However, they show that nonparametric methods are accurate, uniform and particularly suitable for multimodal data. Adamowski (1985) proposed a nonparametric kernel estimation of flood frequencies. Lall et al. (1993) focused on the selection of the kernel function, representing the shape, and the bandwidth in nonparametric kernel estimation for flood frequencies. The techniques for selection of kernel function and band width are applied to three situations: Gaussian data, skewed data and mixed data. Adamowski (1989) performed a Monte Carlo simulation experiment to compare the nonparametric method with two parametric distributions, namely, log-Pearson type III and Weibull distributions. Results show that the nonparametric method gives more accurate results than parametric methods. Adamowski (1996) further proposed a nonparametric method for low-flow frequency analysis to find the conditions of drought. Kim et al. (2003, 2006) proposed a methodology for estimating bivariate drought return periods based on two different severity indices using nonparametric kernel estimator. A perusal of the statistical literature shows that nonparametric statistical estimation, using splines, kernel functions, nearest neighbor methods and orthonormal series methods (Efromovich, 1999; Bowman and Azzalini, 1997; Higgins 2004), is an active area of research, with major developments still unfolding (Silverman, 1986; Scott, 1992; Sharma et al., 1997).

It is already discussed that a significant effort have already been given towards modeling joint distribution using copula in flood frequency analysis, which relaxes the restriction of selecting marginals from same families of probability distribution functions. A major limitation in the analyses done earlier is that, the selection of marginal distributions for peak flow, volume and duration are confined within only parametric families of distribution functions. The objective of the present work is to perform a better selection of marginal distribution functions for flood characteristics by both parametric and nonparametric estimation procedures, and to demonstrate how the concept of copula may be used for establishing joint distribution function with mixed marginal distributions. Nonparametric methods based on kernel density estimation and orthonormal series (Schwartz, 1967; Bowman and Azzalini, 1997; Efromovich, 1999) are used to determine the nonparametric distribution functions for peak flow, volume and duration. A novel effort in flood frequency analysis has been given to define marginal distribution using orthonormal series method. It is found that nonparametric method based on orthonormal series is more appropriate than kernel estimation for assigning marginal distributions of flood characteristics as it can estimate the PDF over the whole range of possible values (Bowman and Azzalini, 1997). A set of parametric distribution functions is also tested to find out suitable marginals for flood characteristics. With the marginal distributions thus selected using parametric and nonparametric methods, a set of bivariate distributions for peak flow-volume, volume-duration and peak flow-duration are determined using the concept of bivariate copula. Conditional probabilities and corresponding return periods are determined for different combinations of peak flow, volume and duration. Seventy years of streamflow data for Red River at Grand Forks of North Dakota, US, is used for demonstration of the methodology.

As pre-requisites, a brief overview on estimating density function by nonparametric methods and a brief introduction to copula functions are provided in next two sections. Details of multivariate flood frequency analysis and determination of marginal distributions for flood characteristics, details of the case study, method to establish joint distribution function using copula, determination of conditional distributions, estimation of return periods, and conclusions are given in subsequent sections.

II. NONPARAMETRIC METHOD OF ESTIMATING MARGINAL DISTRIBUTION

Nonparametric methods based on kernel density estimation and orthonormal series (Schwartz, 1967; Bowman and Azzalini, 1997; Efromovich, 1999) are used to determine the nonparametric distribution functions for peak flow, volume and duration. The following two subsections present brief overview of these two methodologies. Mathematical development of the methodologies presented here is taken from Bowman and Azzalini (1997), Efromovich (1999), and Ghosh and Mujumdar (2007).

II.1 Univariate kernel density estimation

Most nonparametric density estimation methods can be expressed by a kernel density estimator, which entails a weighted moving average of the empirical frequency distribution of the sample (Scott, 1992; Sharma, 2000; Kim et al., 2003). Applications of kernel density estimation for determination of PDF for hydrologic variables may be found in Lall (1995), Lall et al. (1996), Sharma et al. (1997), Tarboton et al. (1998), Kim et al. (2003, 2006), and Ghosh and Mujumdar (2007). It involves the use of univariate kernel function [$K(x)$], defined by a function having following property:

$$\int_{-\infty}^{\infty} K(x)dx = 1 \tag{1}$$

A PDF can therefore be used as a kernel function. Table 1 shows examples of some univariate kernel functions typically used in hydrology and water resources engineering (Silverman 1986; Lall et al. 1996, Kim et al. 2003). For simplicity, a normal or Gaussian kernel function with zero mean and variance of one is applied in the present work.

Table 1. Some standard univariate kernel functions.

Sl. no.	Kernel	$K(x)$
1	Epanechnikov	$K(x) = 0.75(1-x^2), \lvert x \rvert \le 1$ $K(x) = 0, otherwise$
2	Gaussian or Normal	$K(x) = (2\pi)^{-1/2} e^{-(x^2)/2}$
3	Rectangular	$K(x) = 0.5, \lvert x \rvert \le 1$ $K(x) = 0, otherwise$
4	Triangular	$K(x) = 1 - \lvert x \rvert, \lvert x \rvert \le 1$ $K(x) = 0, otherwise$

A univariate kernel density estimator [$\hat{f}(x)$] of a PDF at x is defined by:

$$\hat{f}(x) = (nh)^{-1} \sum_{l=1}^{n} K\{(x - x_l)/h\} \tag{2}$$

where n is the number of observations; x_l is the l^{th} observation; and h is the smoothing parameter known as 'bandwidth', which is used for smoothing the shape of the estimated PDF. The selection of bandwidth is an important step in kernel estimation method. A change in bandwidth may dramatically change the shape of the kernel estimate (Adamowski, 1996; Efromovich, 1999). Plug-in estimates (Polansky and Baker, 2000) and least square cross validation (Scott, 1992; Tarboton et al., 1998) are some methods for estimating the bandwidth. Bandwidth for kernel estimation may be evaluated by minimizing the deviation of the estimated PDF from the actual one. When the actual PDF is unknown, the conventional method is to assume a normal distribution. For an asymptotically optimal choice for *h*, an overall measure of the effectiveness of

$\hat{f}(x)$ is provided by the Mean Integrated Squared Error (MISE), described by the following equation (Bowman and Azzalini, 1997, Kim et al. 2003):

$$MISE = E[\int \{\hat{f}(x) - f(x)\}^2 dx] \approx \{0.25h^4\sigma_K^4 \int f''(x)^2 dx\} + (1/nh)\alpha \quad (3)$$

where $\alpha = \int K^2(x)dx$; and σ_K^2 denotes the variance of the kernel function, $\int x^2 K(x)dx$. The bandwidth is estimated in such a way that it minimizes the MISE [equation (3)]. In the present study the bandwidth is estimated based on normal distribution for computational simplicity at each time step. Thus the optimal bandwidth (h_0) is given by (Azzalini, 1981; Silverman, 1986):

$$h_0 = (1.587)\sigma n^{-1/3} \quad (4)$$

where $\sigma = \min\{S, (IQR/1.349)\}$, S is the sample standard deviation and IQR is the interquartile range (i.e., a measure of statistical dispersion, is the range between third and first quartiles). The value of bandwidth thus evaluated is used to estimate the PDF of the data set using equation (2). The methodology of kernel estimation used in the present work is computationally simple, but with a drawback - a large sample can always give a better estimate of kernel density estimator. Therefore, when there is less available data the kernel estimation method may appear less efficient. The bandwidth is estimated by assuming that the actual density is normal, which may not be valid. To overcome the drawback of kernel density estimation, a methodology based on orthonormal series (Schwartz, 1967; Bowman and Azzalini, 1997; Efromovich, 1999) is used for estimation of nonparametric PDF. The next subsection presents the details of the methodology for estimation of PDF using orthonormal series.

II.2 Orthonormal series method

A PDF from a small sample can be estimated using orthonormal series method which is essentially a series of orthonormal functions, obtained from the sample. An orthonormal series is a series of orthonormal functions, $\Phi_s(x)$ and $\Phi_j(x)$ satisfying the following equations:

$$\int \Phi_s(x)\Phi_j(x)\,dx = 0 \quad \forall s \neq j \tag{5}$$

$$\int \{\Phi_j(x)\}^2\,dx = 1 \quad \forall j \tag{6}$$

Typically a univariate density function of a random variable X may be well approximated by an orthonormal series $\tilde{f}_J(x)$:

$$\tilde{f}_J(x) = \sum_{j=0}^{J} \theta_j \Phi_j(x) \tag{7}$$

where J is called the cut-off, $\Phi_j(x)$, $j = 0,1,2,\ldots$ are the functions of orthonormal system, and θ_j, $j = 0,1,2,\ldots$ are the coefficients corresponding to each function. In the present work, the subset of the Fourier series consisting of cosine functions is selected as orthonormal series:

$$\Phi_0(x) = 1 \tag{8}$$

$$\Phi_j(x) = \sqrt{2}\cos(\pi j x), \qquad j = 1, 2, 3, \ldots \tag{9}$$

The algorithm involved in estimating PDF based on orthonormal series is presented in the section IV.3.2. A brief introduction on copula function used for obtaining the joint distributions of different combinations of flood characteristics, considering the marginal distributions estimated by parametric and nonparametric methods is given in the next section.

III. THE COPULA FUNCTION

Although, the word 'copula' is first employed in a mathematical or statistical sense by Abe Sklar (Sklar, 1959), many of the basic results on copula are traced to the early work of Wassily Hoeffding (Hoeffding, 1940) (Nelsen, 1999). The existing mathematical theories developed on copula for multivariate dependence analysis is quite significant. The only definitions and concepts related to copulas, used in the present study, are discussed in this section. The interested reader is referred to Joe (1997) and Nelsen (1999) for further details.

A copula is a joint distribution function of standard uniform random variables. A bivariate copula can be represented as:

$$C : [0,1]^2 \rightarrow [0,1] \tag{10}$$

It has to fulfill the following conditions: (1) $C(1,u) = C(u,1) = u$ and $C(u,0) = C(0,u) = 0$, and (2) $C(u_1,u_2) + C(v_1,v_2) - C(u_1,v_2) - C(v_1,u_2) \geq 0$ if $u_1 \geq v_1$, $u_2 \geq v_2$ and $u_1,u_2,v_1,v_2 \in [0,1]$. The second condition ensures that the probability corresponding to any rectangle in the unit square is nonnegative. The Sklar's theorem (1959) is the foundation of the concept of copula. Sklar (1959) showed that every n-dimensional distribution function *F* can be written as:

$$F(x_1,......,x_n) = C(F_1(x_1),......,F_n(x_n)) \tag{11}$$

where $F_1, \ldots, F_n$ are marginal distribution functions. If $F_1, \ldots, F_n$ are continuous, then the copula function C is unique and has the following representation:

$$C(u_1,\ldots\ldots,u_n) = F(F_1^{-1}(u_1),\ldots\ldots,F_n^{-1}(u_n)), \quad 0 \le u_1,\ldots\ldots,u_n \le 1 \tag{12}$$

where the quantile F_k^{-1} is defined by $F_k^{-1}(u_k) = \inf\{x \in \Re \mid F_k(x) \ge u_k\}$. Conversely, it can be proven that if C is a copula function and $F_1, \ldots, F_n$ are distribution functions, then the function F defined by equation (11) is a n-dimensional distribution function with margins $F_1, \ldots, F_n$ (Nelsen, 1999).

The Archimedean, elliptical, extreme value copulas are some widely applied classes of copula functions. In the present study, Ali-Mikhail-Haq, Cook-Johnson and Gumbel-Hougaard bivariate copulas (n=2) are considered for the analysis, which belong to the class of Archimedean copula. These copulas find a wide range of applications for: (1) the ease with which they can be constructed, (2) the great variety of families of copulas which belong to Archimedean class and (3) the many nice properties possessed by the members of Archimedean class (Nelsen, 1999). In general, a bivariate Archimedean copula can be defined as (Nelsen, 1999):

$$C_\theta(u_1,u_2) = \phi^{-1}\{\phi(u_1)+\phi(u_2)\} \tag{13}$$

where subscript θ of copula C is the parameter hidden of the generating function ϕ. ϕ is a continuous function, called generator, strictly decreasing and convex from I = [0,1] to [0, ϕ(0)]. Nelsen (1999) provided some important single-parameter families of Archimedean copulas (Table 4.1, pages 116-118), along with their generators, the range of the parameter, and some special and limiting cases. The mathematical expressions of single-parameter bivariate Archimedean copulas and their fundamental properties applied in this study are listed in Table 2.

Table 2. Some single-parameter bivariate Archimedean copulas.

Sl. no.	Copula [$C_\theta(u_1,u_2)$]	$\theta \in$	Generating function[a] [$\phi(t)$]	$\tau = 1 + 4\int_0^1 \frac{\phi(t)}{\phi'(t)} dt$
1	Ali-Mikhail-Haq Family: $\frac{u_1 u_2}{[1-\theta(1-u_1)(1-u_2)]}$	[-1,1)	$\ln\{\frac{[1-\theta(1-t)]}{t}\}$	$[\frac{(3\theta-2)}{\theta}]-[\frac{2}{3}(1-\theta^{-1})^2\ln(1-\theta)]$
2	Cook-Johnson Family: $\{\max[(u_1)^{-\theta}+(u_2)^{-\theta}-1,0]\}^{-1/\theta}$; $\theta \geq 0$	[-1,∞)\{0}	$\frac{[(t)^{-\theta}-1]}{\theta}$	$\frac{\theta}{(\theta+2)}$
3	Gumbel-Hougaard Family: $\exp\{-[(-\ln u_1)^\theta+(-\ln u_2)^\theta]^{1/\theta}\}$	[1,∞)	$(-\ln t)^\theta$	$(1-\theta^{-1})$

[a] $t = u_1 \ or \ u_2$.

In these copula functions, the parameter θ synthesizes the dependence strength among the dependent random variables. For each bivariate Archimedean copulas, value of θ can be obtained by considering mathematical relationship (Nelsen, 1999) between Kendall's coefficient of correlation (τ) and generating function $\phi(t)$, which is given by $\tau = 1 + 4\int_0^1 [\phi(t)/\phi'(t)]dt$, where $t = u_1 \ or \ u_2$ (as shown in the last column of Table 2). The Kendall's coefficient of correlation (τ) is a well known nonparametric measure of dependence or association in theories of copulas, defined in terms of concordance as follows (Nelsen, 1999): Let {(x_1, y_1), (x_2, y_2), … , (x_n, y_n)} denote a random sample of n observations from a vector (X, Y) of continuous random variables. There are $\binom{n}{2}$ distinct pairs (x_i, y_i) and (x_j, y_j) of observations in the sample, and each pair is either concordant (agreeing) or discordant – let '*c*' denote the number of concordant pairs and '*d*' the number of discordant pairs. Then Kendall's tau for the sample defined as the following expression:

$$\tau = [\frac{(c-d)}{(c+d)}] = \frac{(c-d)}{\binom{n}{2}} \tag{14}$$

where (x_i, y_i) and (x_j, y_j) are concordant if $[(x_i - x_j)(y_i - y_j)] > 0$, and discordant if $[(x_i - x_j)(y_i - y_j)] < 0$. The Kendall's tau from the observation is determined as (i.e., estimate of τ):

$$\tau_n = \binom{n}{2}^{-1} \sum_{i<j} sign[(x_i - x_j)(y_i - y_j)] \tag{15}$$

where $sign = 1$ if $[(x_i - x_j)(y_i - y_j)] > 0$, $sign = -1$ if $[(x_i - x_j)(y_i - y_j)] < 0$; $i, j = 1, 2, ..., n$.

IV. MULTIVARIATE FLOOD FREQUENCY ANALYSIS

Many hydrological studies, related to hydraulic design, planning and management, use univariate frequency analysis of peak flows to find the probability of occurrence of a given flood event. The univariate frequency analysis of peak flows provides a limited assessment of flood events, while many hydrological problems require knowledge of complete information concerning a flood event, e.g., flood peak flow, flood volume, flood duration, shape of the hydrograph, etc. The most significant characteristics of a flood event are peak flow (*P*), volume (*V*) and duration (*D*) (Yue et al., 1999; Yue et al., 2002), as illustrated in Figure 1.

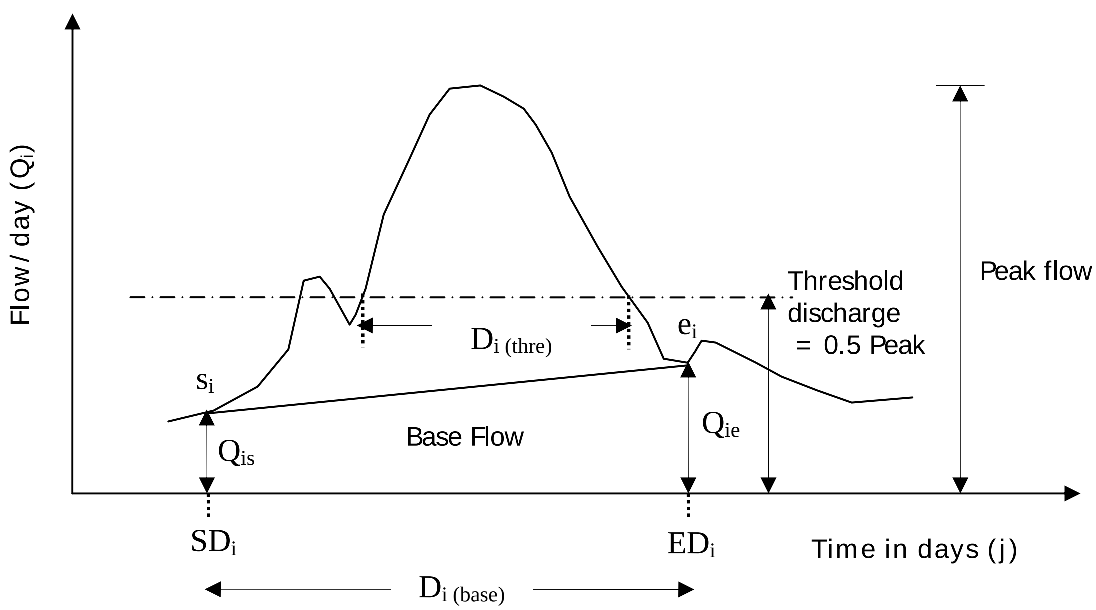

Figure 1. Characteristics of i^th flood event.

In multivariate flood frequency analysis, it is possible to evaluate joint distributions considering *P-V*, *V-D*, *P-D* and *P-V-D* combinations. The peak flow (*P*) of a river is determined by selecting maximum annual flow from streamflow data (*Q*). Theoretically determination of the flood duration involves the identification of the dates of start and end of flood runoff. Generally, start and end dates of a flood event is determined: (1) by fixing some threshold discharge and considering upper part of the hydrograph as a flood event, and corresponding duration value as flood duration, shown by $D_{i\ (thre)}$ in Figure 1 (Todorovic and Zelenhasic, 1970; Ashkar and Rousselle, 1983; Grimaldi and Serinaldi, 2006), and (2) by finding time boundaries marked by a rise in discharge from base flow, denoting start of a flood runoff, and a return to base flow, denoting end of flood runoff, as shown by $D_{i\ (base)}$ in Figure 1 (Linsley et al., 1975; Yue et al., 1999).

It is found from literature (Todorovic and Zelenhasic, 1970; Ashkar and Rousselle, 1983) that, fixing the threshold discharge ($D_{i\ (thre)}$) at a location for flood frequency analysis may not be straight forward as the mutual correlation among flood characteristics are very sensitive to the choice of the threshold value. Some investigation is performed in the present study to identify the pattern of mutual correlation between P, V and D for different threshold discharges, discussed in the section IV.2. In the present study, base flow is considered as a criterion to delineate flood hydrographs and hence define starting time and ending time of the flood event. In this process base flow can be determined for the historic hydrograph using methods that are largely graphical in nature and already discussed in Yue (2000). The starting of the surface runoff is usually marked by the abrupt rise of the rising limb of the hydrograph, and end of flood runoff can be identified by the flattening of the recession limb of the hydrograph. There exists a significant change in the slope of the hydrograph at the transition of surface runoff and base flow, as the characteristics of surface runoff are quite different from the characteristics of base flow. Based on this criterion, the staring date (SD_i, corresponding to starting point, s_i) and ending date (ED_i, corresponding to ending point e_i) of flood runoff for i^{th} year can be determined, as shown in Figure 1, and flood duration for i^{th}

year is determined as: D_i = $D_{i\ (base)}$ = (ED_i- SD_i). The flood volume for i[th] year is determined as (Yue and Rasmussen, 2002; Yue et al., 2002):

$$V_i = (V_i^{Total} - V_i^{Base}) = \{\sum_{j=SD_i}^{ED_i}[Q_{ij} - \frac{1}{2}(Q_{is} + Q_{ie})]\} - \frac{1}{2}D_i(Q_{is} + Q_{ie}) \qquad (16)$$

$$= \sum_{j=SD_i}^{ED_i} Q_{ij} - \frac{1}{2}(Q_{is} + Q_{ie})(1 + D_i) \qquad (17)$$

where Q_{ij} is the observed streamflow of j[th] day for i[th] year, Q_{is} and Q_{ie} are observed daily streamflows on the starting and ending dates of flood runoff for i[th] year, respectively. Finally, the annual flood peak flow (P), volume (V) and duration (D) series can be represented as P = {P_i}, V = {V_i} and D = {D_i}. P = {P_i}, respectively. The peak flow for i[th] year (P_i) is expressed as:

$$P_i = \max[(Q_{ij} - Q_{ij}^b),\ j = ED_i, 1 + ED_i, 2 + ED_i, \ldots\ldots., SD_i] \qquad (18)$$

where Q_{ij}^b is the jth day base flow value for i[th] year.

IV.1 Details of the case study

To illustrate the methodology developed in the present study for flood frequency analysis, 70 years (1936-2005) of daily streamflow data for Red River at Grand Forks in North Dakota, US, is used. The data is collected from U. S. Geological Survey (USGS) gauging station (05082500) located at latitude 47°55'37"N and longitude 97°01'44"W. It has a drainage area of 30,100 square miles, with contributing area of 26,300 square miles. The values of streamflow are relatively low in winter season, which lasts for about four months. The spring season represents high flow due to the contribution of snowmelt to the river runoff. The streamflow data and other details can be obtained from <http://waterdata.usgs.gov/nwis/nwisman/?site_no=05082500&amp>. On the basis of the procedures explained in Figure 1, the annual flood hydrograph series (70 hydrographs for the year 1936 to 2005), and corresponding annual flood peak flow, volume and duration series are determined. All the series for three flood characteristics are obtained: (1) by fixing the threshold discharge ($D_{i\,(thre)}$), taken as some fraction of peak flow (for example, $D_{i\,(thre)}$ = 0.5 * Peak flow), and (2) by deducting the base flow.

IV.2 Associations among flood characteristics

Since P, V and D are the random variables defined by the same physical phenomenon they should be mutually correlated. In fact, a study of the pair wise degree of association between P, V and D makes it evident how these variables are dependent on each other. An illustration of this observation is shown in Figures 2 and 3. The dependence of flood variables, *P*, *V* and *D* is ascertained using Pearson's linear correlation coefficient (ρ). Sample estimate of ρ_{XY} for random variables *X* and *Y* is expressed as:

$$\rho_{XY} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n\sqrt{S_X^2 S_Y^2}} \tag{19}$$

where n is the sample size; $\bar{x}$, S_X^2, $\bar{y}$ and S_Y^2 are sample estimates of mean and variance of random variables *X* and *Y*, respectively. Figure 2 shows the variation of the

Pearson's linear correlation coefficient for *P-V*, *V-D* and *P-D*, i.e., ρ_{PV}, ρ_{VD} and ρ_{PD}, with threshold discharge ($D_{i\,(thre)}$). It is found that, the values of correlation coefficients for peak flow-volume, volume-duration, and peak flow-duration are very sensitive with threshold discharge values. The correlation between peak flow and volume is highly significant at all levels of threshold discharge.

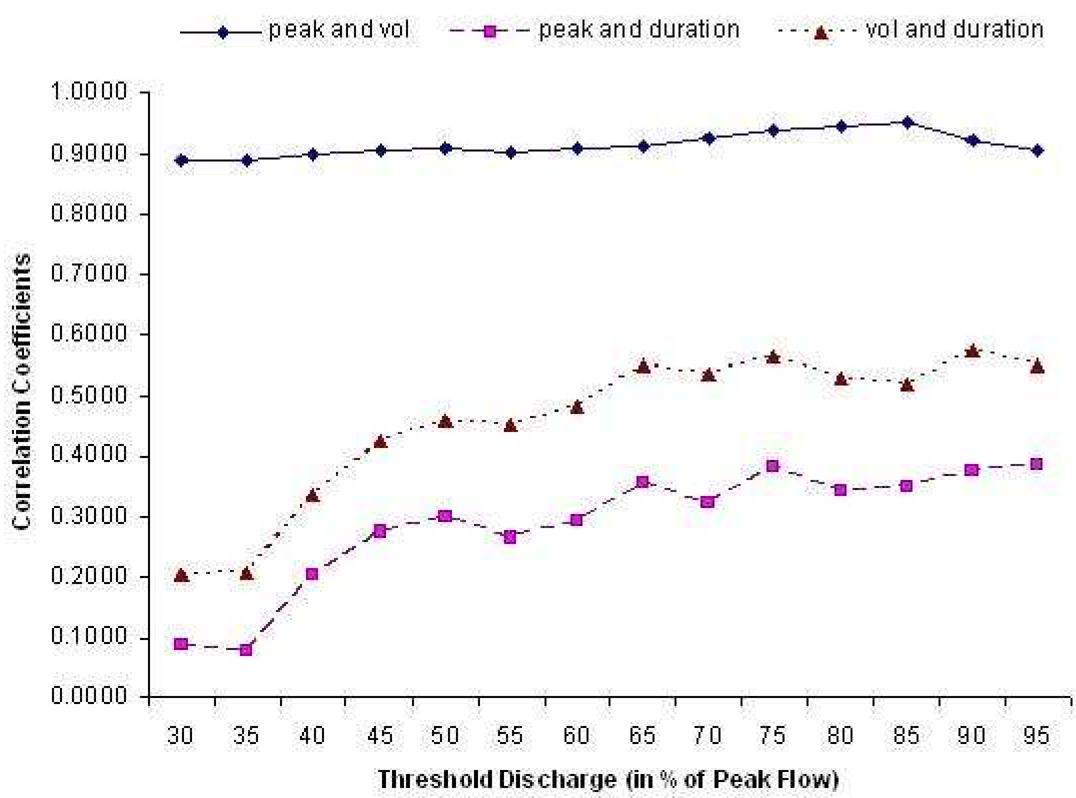

Figure 2. Variation of correlation among flood variables with threshold discharge.

In the present case study, the values obtained for ρ_{PV}, ρ_{VD} and ρ_{PD} at all threshold discharge levels are positive, which confirms the observations by Grimaldi and Serinaldi (2006). Yue et al. (1999) state that, "from the physical point of view, generally, there is no close correlation between peak flow (*P*) and duration (*D*), and there should not exist a negative or inverse correlation between peak flow and duration." Therefore, the present study partially confirms this assumption, as a positive correlation between peak flow and duration is obtained, but not insignificant (ρ_{PD} = 0.5476 for threshold discharge level at 95% of the peak flow value). Figure 3 shows the variation of average flood duration, i.e., average of 70 years duration values, with threshold discharge. It is shown that, the average flood duration decreases monotonically with increase in threshold discharge.

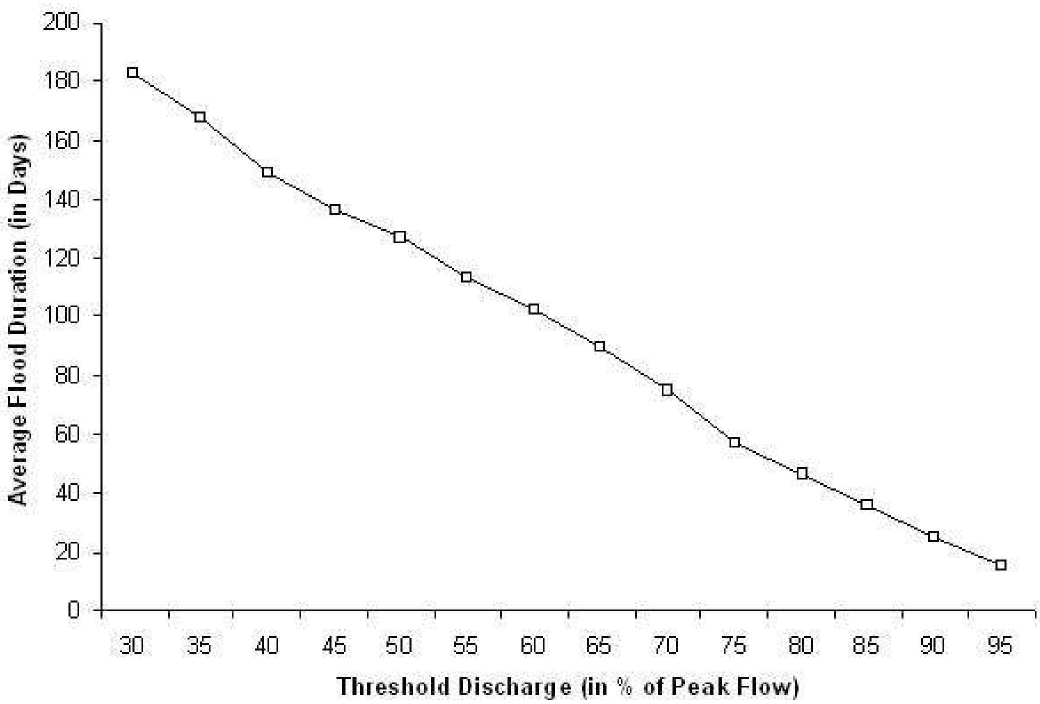

Figure 3. Variation of average flood duration with threshold discharge.

In the present study, base flow is considered as a criterion to delineate flood hydrographs using equations (16)-(18) and hence evaluate annual flood peak flow (P), volume (V) and duration (D) series for flood frequency analysis. The nature of mutual dependence among flood variables P, V and D are estimated using equations (15) and (19) to determine the values of Kendall's coefficient of correlation (τ) and Pearson's linear correlation coefficient (ρ), respectively. Kendall's coefficient of correlation is a more robust way to test the dependence as it is a rank-based procedure (Genest and Favre, 2007). Table 3 shows that, both measures of dependence exhibit high positive values, indicating high positive correlation among *P*, *V* and *D*. The pair *P*-*V* shows greater mutual degree of dependence than that one of *V*-*D* and *P*-*D*, which is similar to the observations by Grimaldi and Serinaldi (2006).

Table 3. Values of correlation coefficients for flood characteristics.

Sl. no.	Flood characteristics	Kendall's coefficient of correlation	Pearson's linear correlation coefficient
1	Peak Flow-Volume	0.7892	0.9359
2	Volume-Duration	0.5756	0.6934
3	Peak Flow- Duration	0.4033	0.5306

The high positive correlation among *P*, *V* and *D* indicates that a trivariate modeling of peak flow – volume – duration is required for the present study on flood frequency analysis. The objective of the present study as discussed earlier, is to perform an extensive task to identify better marginal distribution for *P*, *V* and *D* using both parametric and nonparametric methods, and to demonstrate how copula can be efficiently applied to model the dependence structures of *P*-*V*, *V*-*D* and *P*-*D*. Hence, the present flood frequency analysis is confined to an application of bivariate copula functions with mixed parametric and nonparametric marginal distributions. Next subsection describes the parametric and nonparametric methodologies to obtain the best fitted marginal distributions for flood variables.

IV.3 Marginal distributions of flood characteristics

Both parametric and nonparametric methods are applied to annual flood peak flow (*P*), volume (*V*) and duration (*D*) data to evaluate best fitted marginal probability distribution functions. The values of some useful statistical characteristics of flood variables *P*, *V* and *D* are in Table 4. Note that with copula approach, it is not necessary to have the same marginal distribution function for *P*, *V* and *D*, which imparts convenience to perform an extensive search for selecting best fitted margins.

IV.3.1 Parametric estimation

Many parametric distributions have been used to estimate flood frequencies from observed annual flood series (Kidson and Richards, 2005). Perhaps the most widely used is double-exponential and Extreme Value type-1 (EV1) or Gumbel distributions (Adamowski, 1989). The general extreme value distribution is recommended as a base method in the United Kingdom (Hall, 1984; Adamowski, 1989). The U. S. Water Resources Council (WRC), after appraising many distributions, issued a series of bulletins recommending the Log-Pearson type-III as a base method for use by all U. S. federal agencies (Adamowski, 1989). In hydrology, there is no evidence in favor of any particular parametric distribution or fitting procedure.

Table 4. Statistical characteristics of flood variables.

Sl. no.	Flood Characteristics				
			Peak flow, *P*, (MCum/day)	Volume, *V*, (MCum)	Duration, *D*, (Days)
1	Percentile value	Minimum	5.2601	46.1	16
		25%(Q1)	27.096	219.13	28.75
		50%(Median)	53.458	451	38
		75%(Q3)	83.245	1062.3	51.25
		Maximum	310.7151	4796.7	85
2	Range		305.45	4750.6	69
3	Coefficient of variation		0.79589	1.0588	0.37203
4	Arithmetic mean		63.0790	790.2257	40.8143
5	Standard deviation		50.2037	836.6556	15.1842
6	Coefficient of skewness		2.1838	2.1692	0.7333
7	Kurtosis		7.7762	6.7526	0.2584

Based on the goodness of fit tests (section IV.3.3), several distributions often would fit the data equally well, but each distribution would give different estimates of a given quantile, especially in the tails of the distribution. Based on the first four moments, as given in the last four rows of Table 4, and the histograms of the flood variables four most commonly used parametric distribution functions, i.e., exponential, gamma, Gumbel or EV1, lognormal are selected as possible margins. The parameters of each distribution are first estimated by the Maximum Likelihood Estimation (MLE) method. The values of the parameters are shown in Table 5.

Table 5. Parameters of marginal distribution functions of flood variables.

Sl. no.	PDF		Parameters		
			Peak flow (P)	Volume (V)	Duration (D)
1	Exponential: $f_X(x) = \frac{1}{\eta} e^{-x/\eta};\ x > 0$	η	0.0159	0.0013	0.0245
2	Gamma: $f_X(x) = \frac{\lambda^\beta x^{\beta-1} e^{-\lambda x}}{\Gamma(\beta)};\ x \ge 0, \lambda > 0, \beta > 0$ and $\Gamma(\beta) = \int_0^\infty u^{\beta-1} e^{-u} du$	λ	39.956	885.81	5.6490
		β	1.5787	0.8921	7.2251
3	Gumbel or EV1: $f_X(x) = \frac{1}{\alpha} \exp[-\frac{x-\varepsilon}{\alpha} - \exp(-\frac{x-\varepsilon}{\alpha})]$	ε	40.485	413.69	33.981
		α	39.144	652.34	11.839
4	Lognormal: $f(x) = \frac{1}{x\sqrt{2\pi}\sigma_y} \exp\left(-\frac{(y-\mu_y)^2}{2\sigma_y^2}\right)$, $y = \log x,\ x > 0, -\infty < \mu_y < \infty, \sigma_y > 0$	μ_y	3.8562	6.1470	3.6416
		σ_y	0.7971	1.0873	0.3697

To overcome some of the limitations of the parametric method, nonparametric methods based on kernel density estimation and orthonormal series are used to determine the marginal distribution functions for *P*, *V* and *D* data sets, following the procedures discussed in section II. These methods do not require the assumption of any particular form of density function.

IV.3.2 Nonparametric estimation

The marginal PDF for *P*, *V* and *D* data sets are estimated by nonparametric univariate kernel estimator as discussed in section II.1. The optimal bandwidths (h_0) of kernel function for three sets of data are calculated by using equation (4). The values of h_0 thus evaluated are used to estimate the PDFs of *P*, *V* and *D* data sets using equation (2). The numerical integration is performed to derive the CDFs of the flood variables. This numerical approximation of the CDF is an empirical way to generalize the CDF estimation since a nonparametric approach provides no closed forms of the PDF and CDF (Kim et al., 2006). The validation analyses in Kim et al. (2003) indicates that the kernel density estimator provides less biased and more stable results for estimating CDF than parametric models. However, some problems limit the application of kernel density estimator in flood frequency analysis (Adamowski, 1989). The most important are: (1) a constant value of h_0 for the distribution of a flood variable is unreasonable in skewed distribution; (2) the method implies a very small probability in extrapolation beyond the highest observed data in the sample. The extrapolation is based on the shape of the kernel density function (as different functions are shown in Table 1) assumed, and the value of h_0. Thus, only a few observations contained in the interval h_0 will influence the extrapolation.

To avoid such drawbacks, the methodology based on orthonormal series is applied, as discussed in section II.2, to estimate the marginal distributions of *P*, *V* and *D*. The algorithm for PDF estimation by orthonormal series method involves the following steps

(Schwartz, 1967; Bowman and Azzalini, 1997; Efromovich, 1999; Ghosh and Mujumdar, 2007):

Step 1 - Determination of Support and Scaling of Dataset:

The methodology presented for estimation of PDF using orthonormal systems is valid when the bound of random variable is [0, 1]. The random variable of interest may have different bounds (say, [a, b]) and thus need to be converted to a variable y having interval of [0, 1] by scaling the data. The scaled variable y may be given by: $y = [(x-a)/(b-a)]$. Considering the minimum and maximum values from the data set as the two bounds, [a, b], is not a realistic method as there is no guarantee that unobserved values will not cross these bounds. Methodology for determination of support from a data set may be found in Efromovich (1999). According to that methodology, if $y = x_1, x_2, \ldots\ldots, x_n$ are ordered observations ($x_1 \le x_2 \le \ldots. \le x_n$), then

$$a = (x_1 - d_1) \text{ and } b = (x_n + d_2) \tag{20}$$

where $d_1 = (x_{1+s} - x_1)/s$ and $d_1 = (x_n - x_{n-s})/s$; and s is a small positive integer assuming that the density is flat near the boundaries of its support. The default value of s is 1, which is considered in the present study.

Step 2 - Estimation of Orthonormal Series with Coefficients:

The functions involved in the orthonormal series can be obtained using equations (8)–(9). The coefficients $\theta_j,\ j = 0, 1, ...$ presented in equation (7) can be given by the following equation, where $f(y)$ is the PDF of the scaled random variable Y.

$$\theta_j = \int_{-\infty}^{\infty} f(y)\Phi_j(y)dy \tag{21}$$

It follows that θ_j is the expected value of $\Phi_j(y)$, i.e., $\theta_j = E[\Phi_j(y)]$; which in turn may be approximated from a finite sample of n observations (i.e., $y_j,\ j = 1, ..., n$) as:

$$\theta_j = \frac{1}{n}\sum_{l=1}^{n}\Phi_j(y_l) \tag{22}$$

Step 3 - Estimation of Cutoff J:

Determination of an appropriate cut-off J [equation (7)] is important in the method based on orthonormal series. The choice of J depends on the goodness of fit, which can be determined by Mean Integrated Square Error (MISE) using Parseval's inequality. Following Effromovich (1999) J can be computed as:

$$J = \underset{0 \le \tilde{J} \le J_n}{\arg\min} \sum_{j=0}^{\tilde{J}} (2d_j n^{-1} - \theta_j^2) \tag{23}$$

where $J_n = [C_{J0} + C_{J1} \ln(n)]$. The default values of C_{J0} and C_{J1} are 4 and 0.5 (Efromovich, 1999). Equation (23) denotes the value of $\tilde{J}$, for which the expression $\sum_{j=0}^{\tilde{J}} (2d_j n^{-1} - \theta_j^2)$ attends minimum value, which is considered as the cutoff J.

Step 4 - Smoothing of Estimated PDF:

In many cases it is worthwhile to smooth the Fourier coefficients by multiplying them with some constants that take values between 0 and 1. After smoothing, a modified PDF is given by:

$$\tilde{f}_J(y) = \sum_{j=0}^{J} w_j \theta_j \Phi_j(y); \; 0 \le y \le 1, 0 \le w_j \le 1 \tag{24}$$

The weights (w_j) used for smoothing Fourier coefficients may be given by:

$$w_0 = 1 \tag{25}$$

and $$w_j = [1 - \frac{d}{(n\theta_j^2)}]_+ \quad \forall j > 0 \tag{26}$$

Here $[1 - d/(n\theta_j^2)]_+ = \max\{0, [1 - d/(n\theta_j^2)]\}$, i.e. the positive part of $[1 - d/(n\theta_j^2)]$. Other than first J number of Fourier coefficients, a density function also requires a relatively large number of coefficients for a fair visualization. Thus high frequency terms are added, which are shrunk by a hard threshold procedure. After adding these extra terms equation (24) is modified to:

$$\tilde{f}_J(y) = \sum_{j=0}^{J} w_j \theta_j \Phi_j(y) + \sum_{j=J+1}^{C_{JM} \times J_n} I_{\{\theta_j^2 > C_T d \ln(n)/n\}} \theta_j \Phi_j(y);\ 0 \le y \le 1, 0 \le w_j \le 1 \tag{27}$$

where C_{JM} and C_T are parameters for hard threshold procedure that define the maximum number of elements included in the estimate of PDF. The default values are 6 and 4, respectively (Efromovich, 1999). I is an indicator variable such that $I_{\{A\}}$ has the value of 1 if A is true, zero otherwise. A high frequency term is included only if the corresponding Fourier coefficient is extremely large and thus the procedure does not reduce the smoothness of the estimate.

Step 5 - Modification of Area under PDF and Negative Values:

An improvement in the PDF, thus estimated is necessary when it takes negative values at some of the points/ regions. For such cases, the following steps (Ghosh and Mujumdar, 2007) are used in the present study, which ensures that the properties of PDF are satisfied by the estimated PDF:

I. If there exist negative values at some points, find the maximum negative value.
II. Add the magnitude of the maximum negative value to $\tilde{f}_J(y)$ to make the value of the function positive, everywhere.
III. Check the area under the curve. If it is less than 1, find out c by numerical methods, such that,

$$\int_{-\infty}^{\infty}[\tilde{f}_J(y)+c]dy=1 \tag{28}$$

$\tilde{f}_J(y)$ is now modified by adding c, as obtained from equation (28), to it.

$$\tilde{f}_J(y)=\tilde{f}_J(y)+c \tag{29}$$

IV. If the area under the curve is greater than 1, find out $c1$ in a similar procedure:

$$\int_{-\infty}^{\infty}[\tilde{f}_J(y)+c1]dy=1 \tag{30}$$

Subtracting $c1$ from $\tilde{f}_J(y)$ may lead to a negative value of PDF, which is not desirable. In such cases after the subtraction take only the positive part of $\tilde{f}_J(y)$.

$$\tilde{f}_J(y)=[\tilde{f}_J(y)-c1]_+ \tag{31}$$

where $[\tilde{f}_J(y)-c1]_+=\max\{0,[\tilde{f}_J(y)-c1]_+\}$. Check the area again and if it is not nearly equal to 1, go to III; else stop.

The other way of making adjustment in the estimated PDF to make the area under the curve equal to 1, is the use of multiplicative factor. In either case, the result will be almost same as the adjustment methodology is not very sensitive to the final PDF.

Step 6 - Estimation of PDF for Unscaled Dataset:

The scaled data/observations are distributed according to a density $f_Y(y)$, where $y \in [0,1]$. The PDF thus obtained corresponds to the scaled data set y over the interval of [0, 1]. The estimate of $f_X(x)$ of original dataset x over interval [a, b] may be given by:

$$f_X(x) = (b-a)^{-1} f_Y(y); \;\; x \in [a,b] \tag{32}$$

After estimating the PDF for peak flow, volume and duration by orthonormal series method, numerical integration is performed for evaluating the CDF for each flood variable.

One advantage of orthonormal series method over the kernel estimation, and others, is that it produces a 'proper' function. Once θ_j has been computed, they represent the estimated function over the whole range of possible values. A disadvantage of the method is that some people find its construction less intuitive and $\tilde{f}_J(y)$ can occasionally take negative values in the tails of the distribution (Bowman and Azzalini, 1997). The problem of negative values is tackled using equations (28)-(31).

Figure 4 shows the histogram of peak flow, volume and duration, with the parametric and nonparametric densities fitted to the data sets. It is seen that, the histogram of duration data has bimodal shape that cannot be reproduced by any parametric distributions commonly used. However, this bimodality is modeled effectively by the density function obtained from orthonormal series method. Note that, this is also effectively smoothing the histograms for other two flood variables, peak flow and volume.

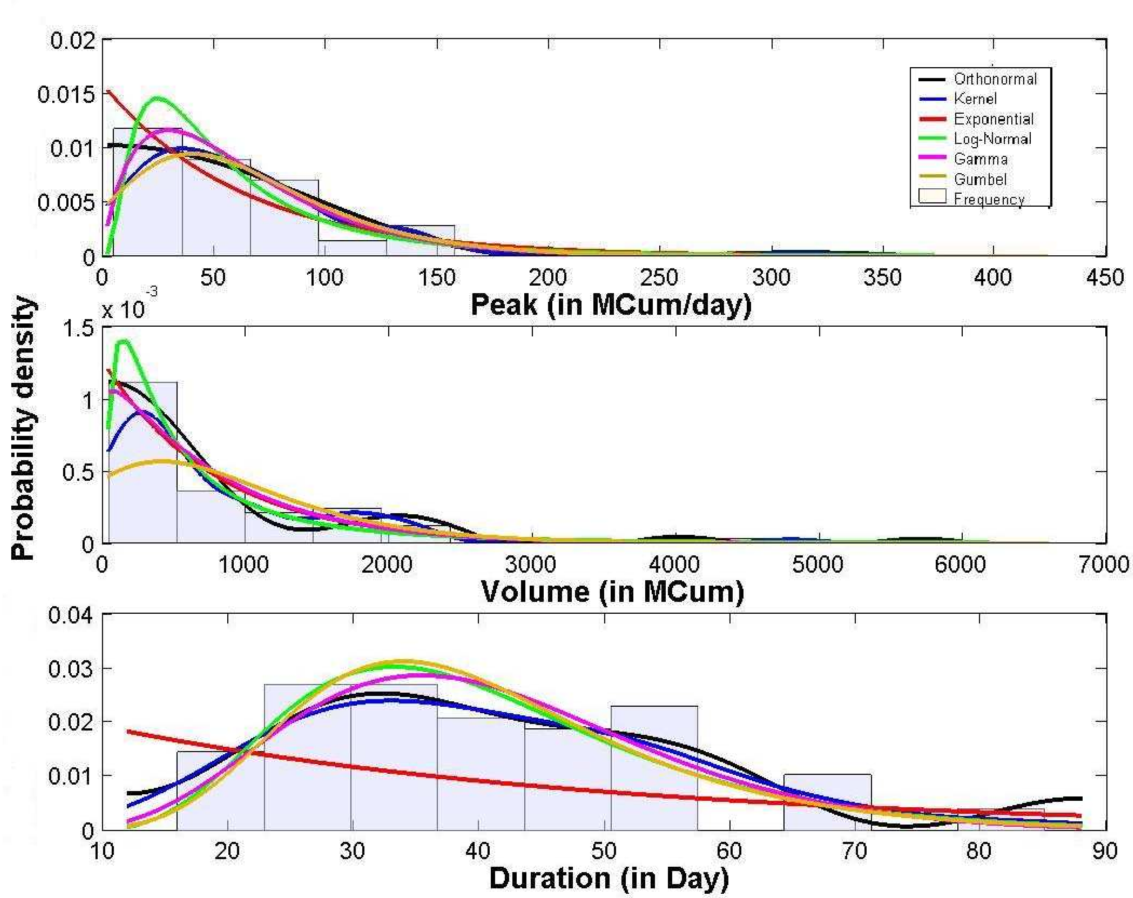

Figure 4. Comparison of different probability density estimates with observed frequency.

IV.3.3 Goodness of fit tests

The CDFs for flood variables estimated by parametric and nonparametric procedures are fitted to the data series of *P*, *V* and *D* and are compared with their empirical nonexceedance probabilities [$P[X \le x] = F(x)$], obtained from plotting position formula. An extended list of plotting position formula commonly used for flood frequency analysis is available in Cunnane (1989), and Rao and Hamed (2000). The

Gringorten plotting position formula (Gringorten, 1963) is one of the commonly used plotting position formula and is expressed as:

$$P(K \le k) = \frac{k - 0.44}{N + 0.12} \tag{33}$$

where k is the k^{th} smallest observation in the data set arranged in ascending order and N is the sample size.

In the present study, The Root Mean Square Error (RMSE) and Akaike Information Criterion (AIC) are used as goodness of fit statistics for the parametric and nonparametric probability distribution functions selected for possible margins of *P*, *V* and *D*. Further the chi-square test (Snedecor and Cochran, 1989) is performed to test if a sample data came from a population with a specific distribution. The RMSE is expressed as:

$$RMSE = \sqrt{E(x_c - x_o)^2} = \{\frac{1}{N-k}\sum_{i=1}^{N}[x_c(i) - x_o(i)]^2\}^{\frac{1}{2}} \tag{34}$$

where $E(*)$ is the expectation of $(*)$; $x_c(i)$ and $x_o(i)$ denote the i^{th} computed and observed values, respectively; k is the number of parameters used in obtaining the computed value; and N is the number of observations. The best fitted distribution function is the one that has the minimum RMSE value.

The AIC, developed by Akaike (1974), is used to identify most appropriate probability distribution. It includes (1) the lack of fit of the model, and (2) the unreliability of the model due to the number of model parameters (Zhang and Singh, 2007), and can be expressed as:

$$AIC = -2\log(\max imum\ likelihood\ \ for\ \mathrm{mod}\,el) + 2(no.of\ \ fitted\ \ parameters) \quad (35)$$

$$\text{or}\ \ AIC = N\log(MSE) + 2(no.of\ \ fitted\ \ parameters) \quad (36)$$

$$\text{where}\ \ MSE = E(x_c - x_o)^2 = \{\frac{1}{N-k}\sum_{i=1}^{N}[x_c(i) - x_o(i)]^2\} \quad (37)$$

The best model is the one which has the minimum AIC-value. Table 6 shows RMSE and AIC values for all parametric and nonparametric distribution functions used for peak, volume and duration.

Table 6. Comparison of RMSE and AIC values of flood variables for different statistical distributions.

Sl. no.	Distribution function	RMSE			AIC		
		Peak flow	Volume	Duration	Peak flow	Volume	Duration
1	Kernel	0.053852	0.103400	0.02595	-583.0276	-454.1767	-730.3393
2	Orthonormal	0.020117	0.020930	0.019447	-781.2385	-773.3609	-788.0164
3	Exponential	0.046904	0.044700	0.272600	-610.3089	-618.7217	-257.9715
4	Gamma	0.016802	0.038700	0.023446	-813.2527	-647.8939	-746.6113
5	Gumbel	0.065574	0.171200	0.026840	-540.4661	-349.0121	-719.5688
6	Lognormal	0.020666	0.025432	0.023486	-7718552	-730.3526	-746.2661

The peak flow data is best fit with the gamma distribution as it is seen in Table 6 that RMSE and AIC values are both at minimum for gamma distribution (RMSE = 0.016802 and AIC = -813.2527). The distribution functions obtained from orthonormal series method fit best the flood volume and duration data sets, as RMSE (0.020930 for volume and 0.019447 for duration) and AIC (-773.3609 for volume and -788.0164 for duration) values are at minimum. Therefore, the flood variable *P* seems to follow gamma distribution, and flood variables *V* and *D* seem to follow distribution function obtained from orthonormal series method.

A chi-square test is performed as a second step for evaluating goodness of fit. An attractive feature of the chi-square goodness of fit test is that it can be applied to any univariate distribution for which you can calculate the cumulative distribution function. The chi-square goodness of fit test is applied to data divided into classes. The values of the chi-square test statistics are dependent on how is the data divided in classes. One disadvantage of the chi-square test is that it requires a sufficient sample size in order for the chi-square approximation to be valid. In the present case study 70 years of data for flood variables appears sufficient for performing the test. For the chi-square goodness-of-fit computation, the data are divided into *k* classes and the test statistic is defined as:

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i} \tag{38}$$

where O_i is the observed frequency for class i and E_i is the expected frequency for class i. The expected frequency is calculated by: $E_i = N[F(Y_u) - F(Y_l)]$, where F is the CDF for the distribution being tested, Y_u is the upper limit for class i, Y_l is the lower limit for class i, and N is the sample size. The test statistic follows, approximately, a chi-square distribution with $(k-c)$ degrees of freedom where k is the number of non-empty classes and c = [the number of estimated parameters, including location-scale-shape parameters, for the distribution] + 1. For example, for a 2-parameter gamma distribution, c = 3. Therefore, the hypothesis that the data are from a population with the specified distribution is rejected if $\chi^2 > \chi^2_{(\alpha,k-c)}$, where $\chi^2_{(\alpha,k-c)}$ is the chi-square percent point function with $(k-c)$ degrees of freedom and a significance level of α. The Table 7 shows the results obtained from chi-square test to find out the acceptability of the distribution functions selected on the basis of RMSE and AIC criteria as presented in Table 6. The results show that the selected distributions (i.e., Gamma distribution for peak flow, and distributions from orthonormal series method for volume and duration) are accepted with significance level of 99.5%.

Table 7. Details from chi-square test on peak flow, volume and duration data.

Sl. no.	Marginal distribution	χ^2-value	Significance level, α	Cutoff obtained from chi-square probability table, $\chi^2_{(\alpha,k-c)}$	Conclusion
1	Peak flow fitted by gamma dist. (Parameter = 2)	0.2835	99.5%	0.989	Accepted
2	Volume fitted by orthonormal series function (Parameter = 0)	1.3790	99.5%	1.735	Accepted
3	Duration fitted by orthonormal series function (Parameter = 0)	0.0986	99.5%	1.735	Accepted

IV.4 Joint distributions for the flood characteristics

In the previous section, the best fitted marginal distributions for peak flow, volume and duration are selected, which are from different parametric and nonparametric families of distribution functions. In section III, it is already discussed, how copula function may be used for generating joint distribution function from different families of marginal distributions. Table 2 describes three widely used Archimedean copulas, namely Ali-Mikhail-Haq, Cook-Johnson and Gumbel-Hougaard family of copulas. The values of correlation coefficients for flood characteristics as shown in Table 3 are all positive and not wildly varying, which confirms the adequacy of modeling the dependence structure of flood variables using Archimedean copulas.

IV.4.1 Determination of generating function and resulting copula

The first step in determining a copula is to obtain its generating function from observed data. The procedure to obtain the generating function and the resulting copula is described by Genest and Rivest (1993). It assumes that for a random sample of bivariate observations $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ the underlying distribution function $H_{X,Y}(x, y)$ has an associated Archimedean copula C_θ which also can be regarded as an alternative expression of the joint CDF. The procedure involves the following steps:

1. Determine Kendall's τ from the observed data (i.e., estimate τ_n) using equation (15);
2. Determine the copula parameter, θ, from the above value of τ_n for each of the three copulas given in Table 2;
3. Obtain the generating function, ϕ, of each copula;
4. Obtain the copula from its generating function.

Figures 5, 6 and 7 show the joint distribution functions using three above mentioned copulas for *P-V*, *V-D* and *P-D*, respectively.

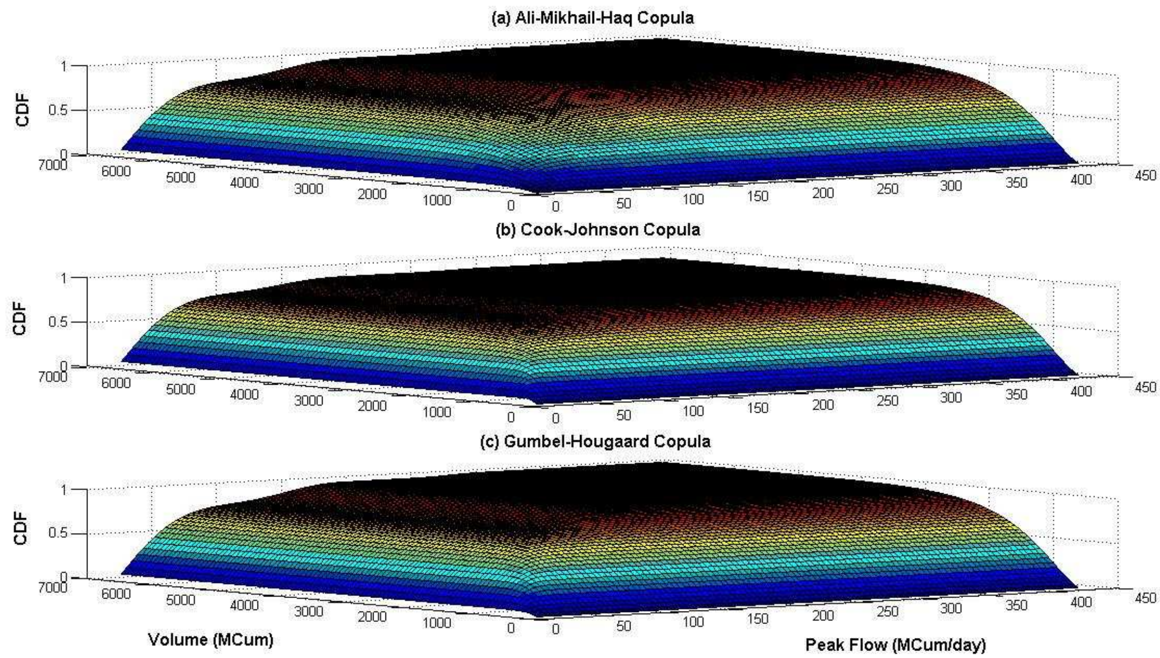

Figure 5. Joint probability distribution function for peak flow and volume.

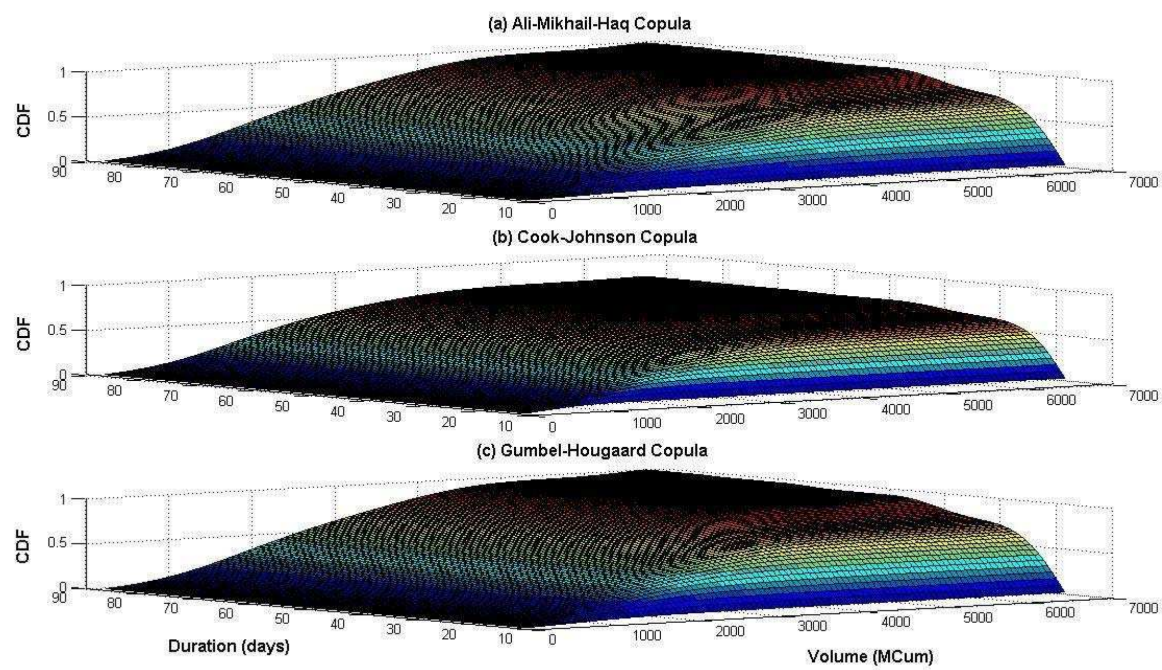

Figure 6. Joint probability distribution function for volume and duration.

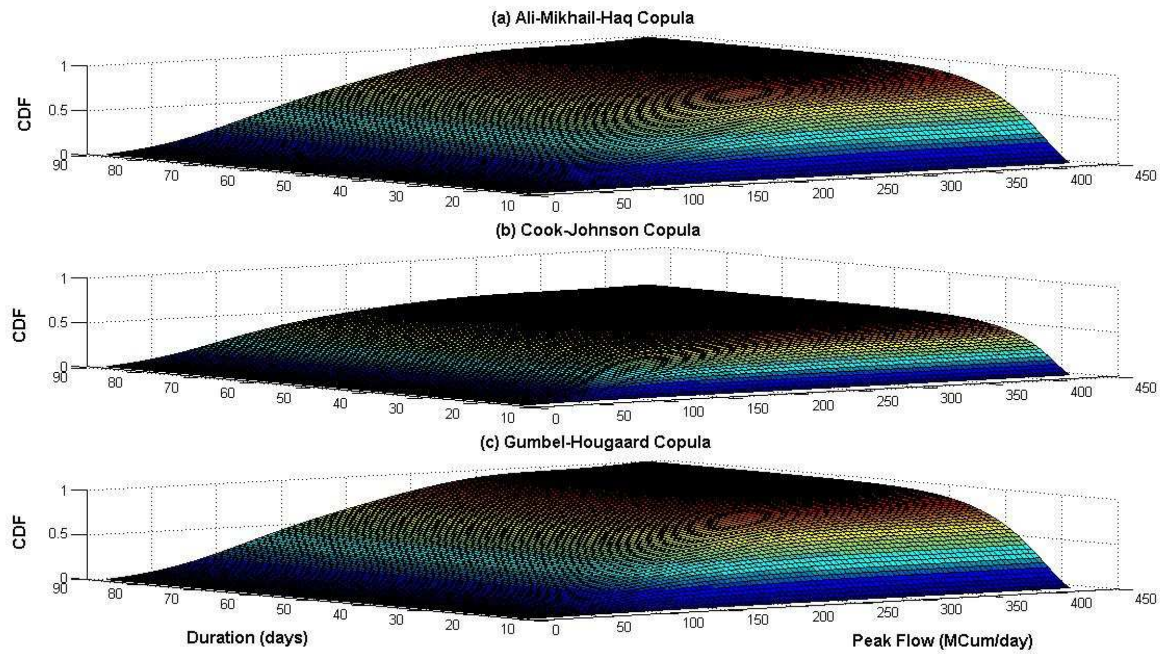

Figure 7. Joint probability distribution function for peak flow and duration.

IV.4.2 Identification of Archimedean copula

The next step is to identify an appropriate copula. Since there is a family of copulas, the question is: which copula should be used to obtain joint distributions of flood variables. This question was addressed by the procedure of Genest and Rivest (1993), who described identification of copulas involving the following steps (Zhang and Singh, 2006):

1. Define an intermediate random variable $Z=Z(x, y)$ which has a distribution function $K(z)=P(Z \le z)$, where z is a specific value of Z. This distribution function is related to the generating function of the Archimedean copula, determined earlier, as Genest and Rivest (1993).

$$K(z) = z - \frac{\phi(z)}{\phi'(z)} \tag{39}$$

where ϕ' is the derivative of ϕ' with respect to z.

2. Construct a nonparametric estimate of K as follows:

 - Obtain that z_i = {number of (x_j, y_j) such that $x_j < x_i$ and $y_j < y_i$}/(N-1) for i = 1, .., N.
 - Construct the estimate of K as $K_N(z)$ = the proportion of $z_i \le z$.

3. Construct a parametric estimate of K using equation (39) with z obtained from Step 2.
4. Plot nonparametrically estimated K_N(z) versus parametrically estimated K for each copula. The plot, called the *Q-Q* plot, indicates whether the quantiles of nonparametrically estimated K_N(z) and parametrically estimated $K(z)$ are in agreement. If the plot is in agreement with a straight line that passes through the origin at a 45° angle, then the generating function is satisfactory. The 45° line indicates that the quantiles are equal. Otherwise, the copula function needs to be reidentified.

The above steps show how to select the best fitted copula visually. Figures 8 – 10 show *Q-Q* plots and indicate whether the quantiles of nonparametrically estimated K_N(z) and parametrically estimated $K(z)$ are in agreement for three different copula models used in the present study.

The joint CDF for *P-V*, *V-D* and *P-D* evaluated from copula method may also be compared with their empirical nonexceedance probabilities, obtained from plotting position formula following the same procedure as discussed in section IV.4.3. Observed joint probabilities are computed based on the same principle as in the case of the single variable, as expressed in equation (33). The joint cumulative frequency (non-exceedance joint probability) is given as (Zhang and Singh, 2006, 2007):

$$F_{XY}(x_i, y_i) = P(X \le x_i, Y \le y_i) = \frac{\sum_{m=1}^{i}\sum_{l=1}^{i} n_{ml} - 0.44}{N + 0.12} \tag{40}$$

where n_{ml} is the number of pairs (x_j, y_j) counted as $x_j \le x_i$ and $y_j \le y_i$; $i, j = 1,......, N$; $1 \le j \le i$ and N is the sample size (=70 for this case study).

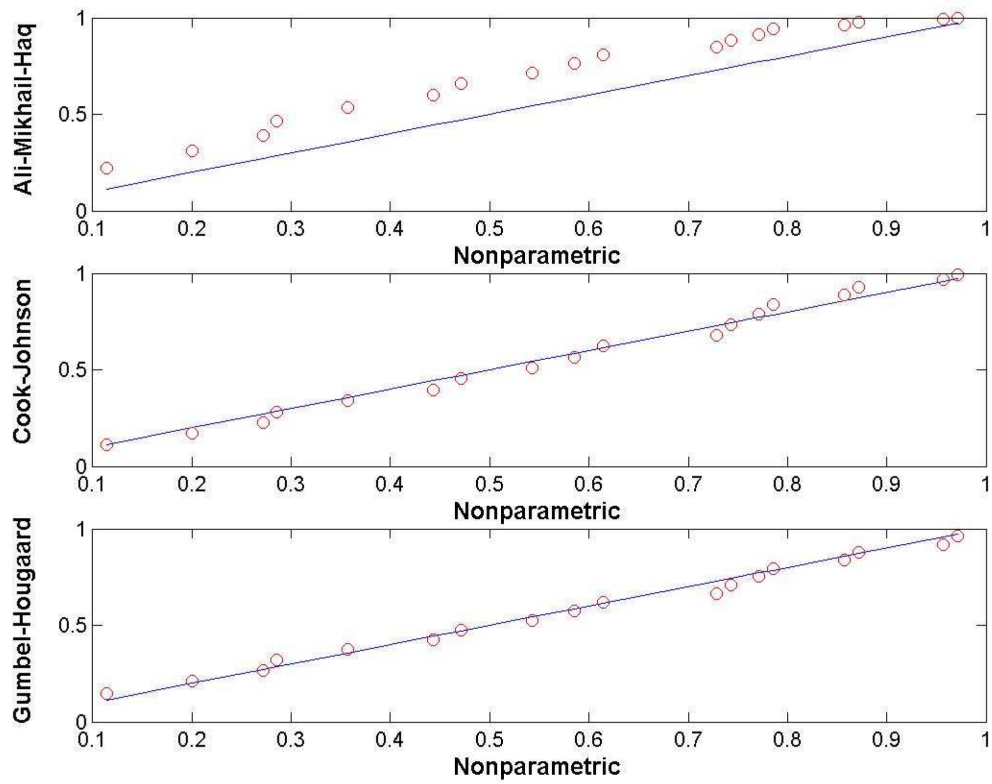

Figure 8. Comparison of parametric and nonparametric estimation of $K(z)$ of peak flow and volume.

The RMSE [equation (34)] and AIC statistics [equations (35)-(36)] are used to test the goodness of fit of sample data to the theoretical joint distribution obtained by using copula functions. Table 8 shows RMSE and AIC values for joint distributions obtained by using different copula function for *P-V*, *V-D* and *P-D*. It can be concluded from Table 8 that Gumbel-Hougaard copula is best model for the joint distributions of *P-V*, *V-D* and *P-D* combinations, as minimum value of RMSE and AIC are obtained for that type of copula (the last row of Table 8). In the next subsection conditional distributions and return periods for different combinations of flood characteristics are evaluated.

Table 8. Comparison of RMSE and AIC values for joint distributions of different combination of flood variables using copula models.

Sl. no.	Copula	RMSE			AIC		
		Peak Flow - Volume	Volume-Duration	Peak Flow-Duration	Peak Flow -Volume	Volume-Duration	Peak Flow-Duration
1	Ali-Mikhail-Haq	0.14140	0.090	0.0557	-68.4302	-84.6440	-101.7638
2	Cook-Johnson	0.03122	0.0583	0.0548	-122.7960	-100.5031	-102.3148
3	Gumbel-Hougaard	0.02519	0.0265	0.02028	-130.5276	-128.7351	-138.3348

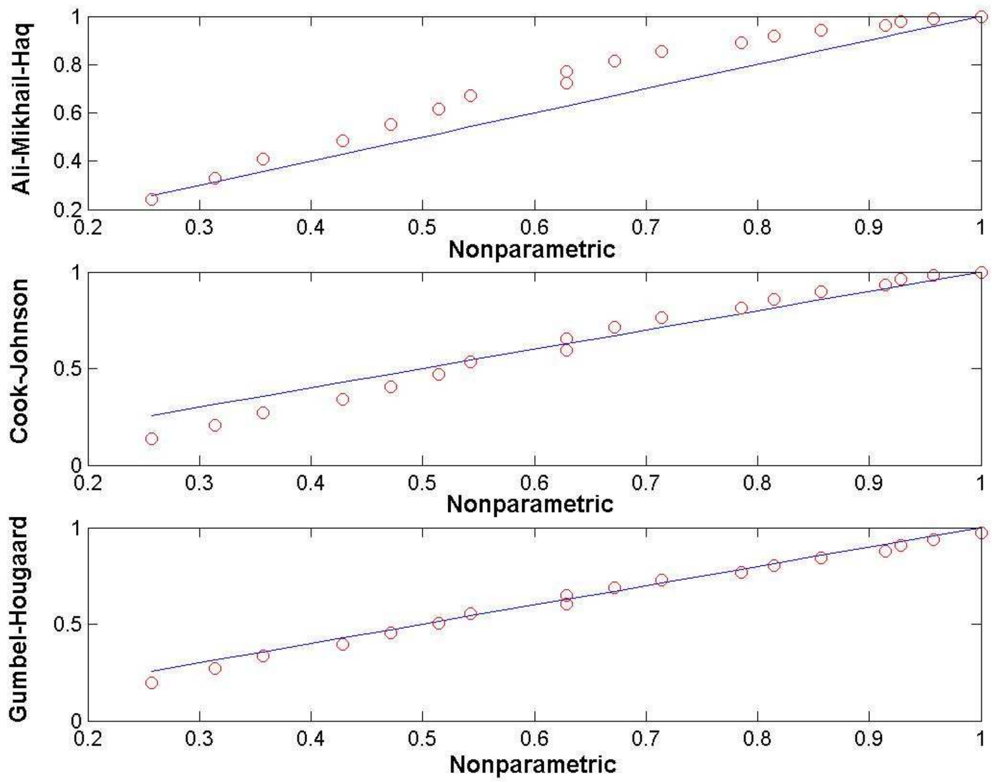

Figure 9. Comparison of parametric and nonparametric estimation of $K(z)$ of volume and duration.

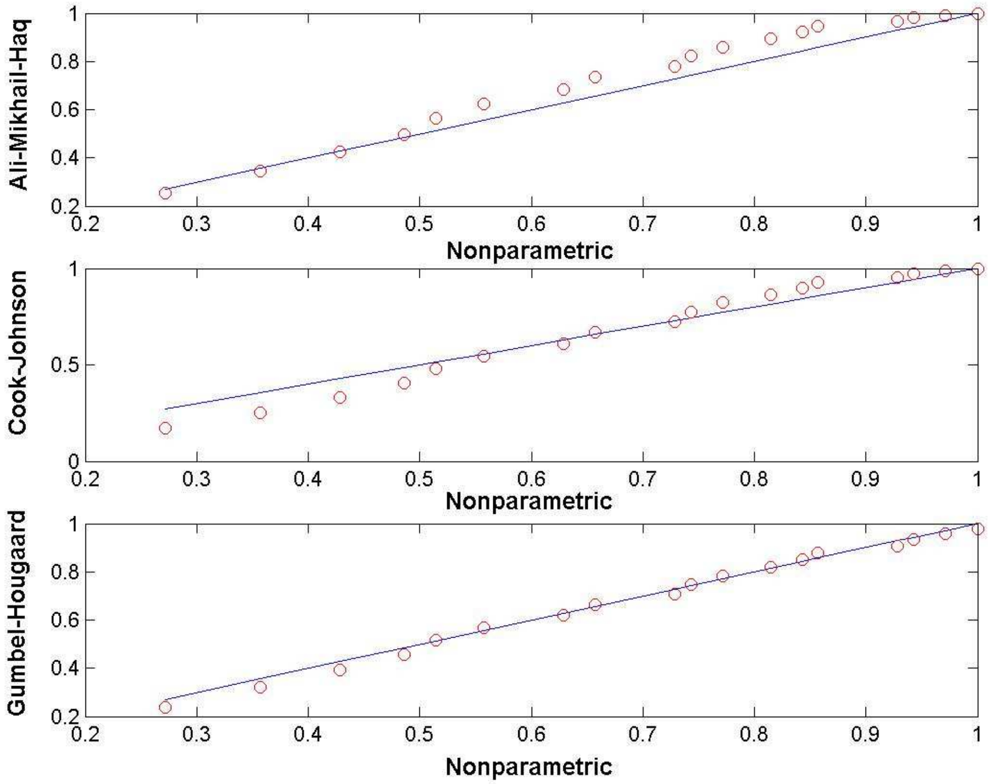

Figure 10. Comparison of parametric and nonparametric estimation of $K(z)$ of peak flow and duration.

IV.5 Conditional distributions and return periods for flood characteristics

The discussion on the copula-based joint distributions in section IV.4 shows that, the Gumbel-Hougaard copula is found to be the best fitted copula for all joint distributions, i.e., peak flow-volume, volume-duration and peak flow-duration. In the present analysis, few particular Conditional Cumulative Distribution Functions (Conditional CDFs) and return periods are evaluated for the Gumbel-Hougaard model. For hydrologic design and planning purposes, given a flood event return period, it is possible to obtain various occurrence combinations of flood peaks, volumes and durations, and vice versa. It is also desirable in flood frequency analysis to obtain information concerning the occurrence probabilities of flood volumes under the condition that a given flood peak or duration occurs, and vice versa. Conditional return period of $X \geq x$ given $Y \leq y$ may be expressed as:

$$T'_{X|Y}(x \mid y) = \frac{1}{1 - H'_{X|Y}(x \mid y)} \tag{41}$$

where $H'_{X|Y}(x \mid y) = H(x \mid Y \leq y) = \dfrac{H_{X,Y}(x, y)}{F_Y(y)}$ (42)

and $H_{X,Y}(x, y) = P(X \leq x, Y \leq y)$ is the joint CDF of random variable X and Y, P is nonexceedance probability. Likewise, an equivalent formula for Conditional return period of $Y \geq y$ given $X \leq x$ (i.e., $T'_{Y|X}(y \mid x)$) can be obtained. In the present study, the best fitted Gumbel-Hougaard copula model is used to obtain the joint CDF for *P-V*, *V-D* and *P-D* combinations, i.e., $H_{P,V}(p, v), H_{V,D}(v, d)$ and $H_{P,D}(p, d)$, respectively. Following the equation (42) the Conditional CDF, $H'_{P|V}(p \mid v)$, $H'_{V|P}(v \mid p)$, $H'_{V|D}(v \mid d)$, $H'_{D|V}(d \mid v)$, $H'_{P|D}(p \mid d)$ and $H'_{D|P}(d \mid p)$ of peak given volume, volume given peak, volume given duration, duration given volume, peak given duration and duration given peak can be computed. Figures 11 – 13 show the Conditional CDFs of peak flow-volume given flood volume, volume-duration given flood duration and peak-duration given flood duration, respectively, for the Red River at Grand Forks. Following equation (41) the Conditional return periods for different combinations and conditions can be evaluated. Figure 11 shows a decreasing trend of the value of Conditional CDF obtained for the some specified value of peak flow under different condition of volume, which indicates a positive correlation structure between peak flow and volume.

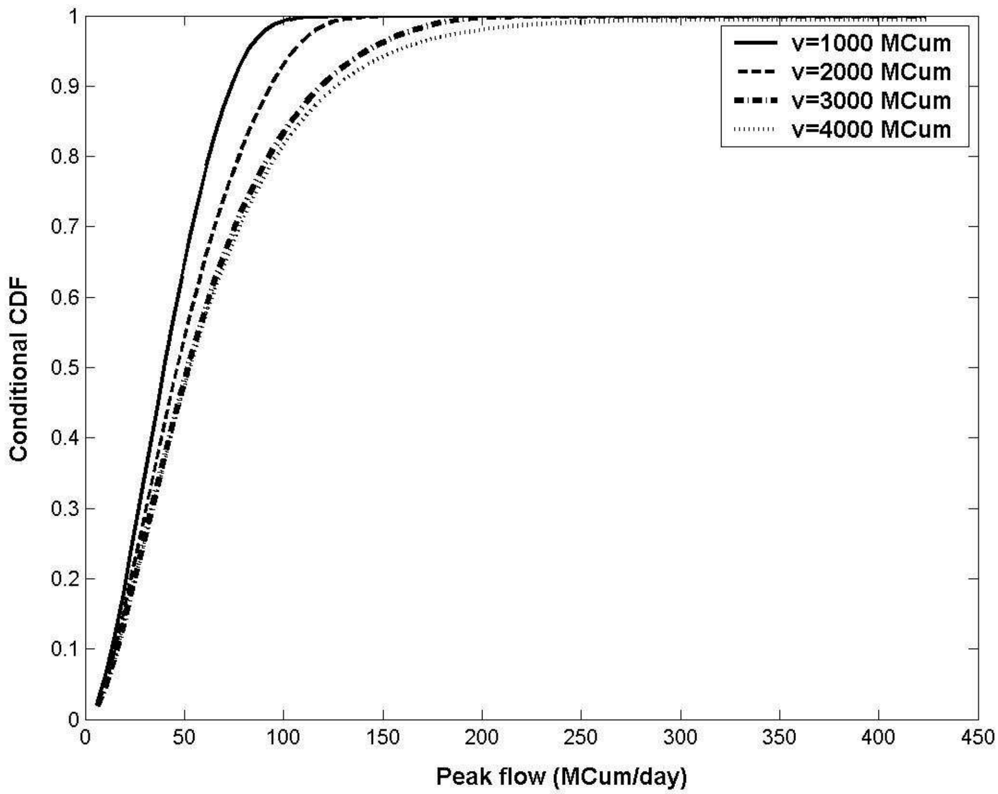

Figure 11. Conditional CDF of peak flow and volume for given flood volume.

Therefore, positive correlation between peak flow and volume results in less likely occurrence of specified peak flow for low conditioning volume than would be the case for the same specified peak flow under high conditioning volume. The observation supports the observation obtained by Zhang and Singh (2006). Similar results for the volume-duration and peak flow-duration combinations are obtained as shown in Figures 12 and 13, which indicate positive correlation between volume and duration, and peak flow and duration. Therefore, results shown in Figures 11-13 agree the positive correlation structure among as *P-V*, *V-D* and *P-D* as evaluated in Table 3.

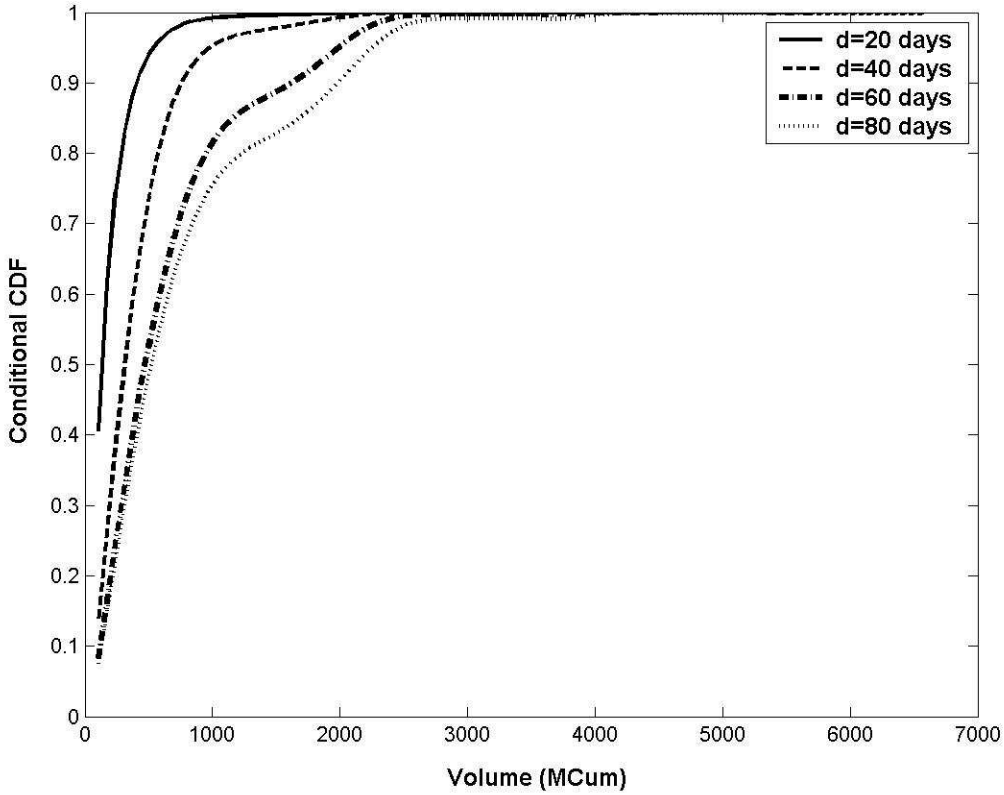

Figure 12. Conditional CDF of volume and duration for given flood duration.

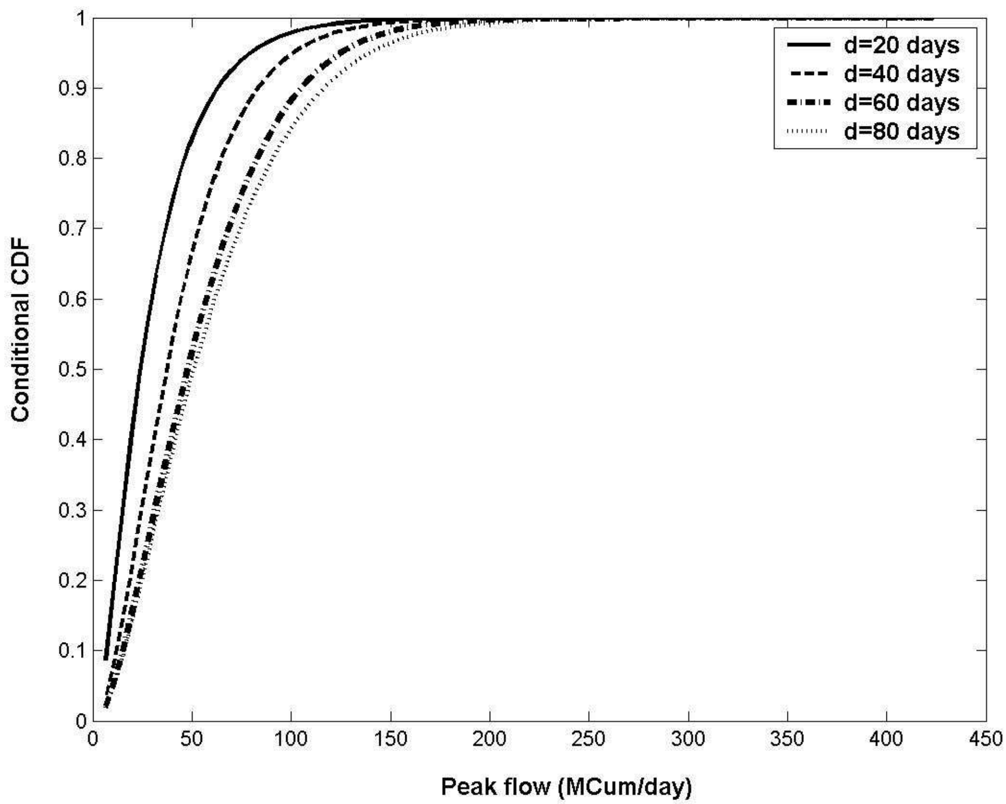

Figure 13. Conditional CDF of peak flow and duration for given flood duration.

Figure 14 shows the joint return period for peak flow and volume, i.e., $T_{P,V}(p,v) = 1/[1 - H_{P,V}(p,v)]$ where $H_{P,V}(p,v) = P(P \le p, V \le v)$ and $T_{P,V}(p,v)$ is the return period associated with the event $P > p\ and\ V > v, or\ P > p, or\ V > v$ that at least one of x and y is exceeded. Other combinations of joint return periods, i.e., $T_{V,D}(v,d)$ and $T_{P,D}(p,d)$, may be evaluated using the same concept.

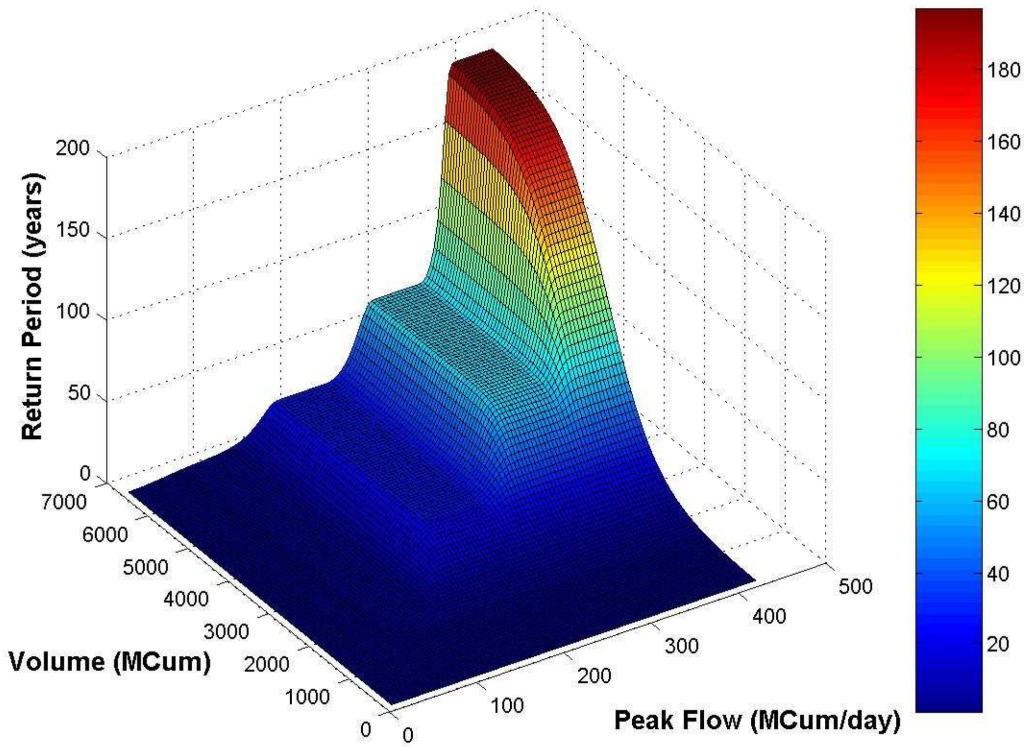

Figure 14. Joint return period for peak flow and volume.

V. CONCLUSIONS

In the present work an extensive selection of marginal distribution functions for flood variables is performed by parametric and nonparametric methods, and concept of copula is used for evaluating joint distribution function with mixed marginal distributions. Modeling joint distribution using copula relaxes the restriction of selecting marginals for flood variables from the same family of probability distribution functions. A major limitation in the analyses done earlier is that, the selection of marginal distributions for peak flow, volume and duration are confined within only parametric families of distribution functions. Nonparametric methods based on kernel density estimation and orthonormal series (Schwartz, 1967; Bowman and Azzalini, 1997; Efromovich, 1999) are used to determine the nonparametric distribution functions for peak flow, volume and duration. A novel contribution to flood frequency analysis has been provided by defining marginal distribution using orthonormal series method. It is found that nonparametric method based on orthonormal series is more appropriate than kernel estimation for determining marginal distributions of flood characteristics as it can estimate the PDF over the whole range of possible values (Bowman and Azzalini, 1997). A set of parametric distribution functions is also tested to find out suitable marginals for flood characteristics. With the marginal distributions thus selected using parametric and nonparametric methods, a set of bivariate distributions for peak flow-volume, volume-duration and peak flow-duration are determined using the concept of bivariate copula. Conditional probabilities and corresponding return periods are determined for different combinations of peak flow, volume and duration. Results indicate that the proposed approach of flood frequency analysis can be useful in solving several problems of hydrologic design and planning, for which single variable flood frequency analysis cannot provide answers. For example, given a flood event return period, it is possible to obtain various occurrence combinations of peak flows and volumes, and vice versa (Yue et al., 1999). These different scenarios can be useful for risk assessment associated with hydrologic problems, such as spillway design (De Michele, 2005), flood control, etc.

VI. REFERENCES

[1] Adamowski, K. (1985), Nonparametric kernel estimation of flood frequencies, *Water Resour. Res., 21*(11), 1885–1890.

[2] Adamowski, K. (1989), A Monte Carlo comparison of parametric and nonparametric estimation of flood frequencies, *J. Hydrol., 108*, 295-308.

[3] Adamowski, K. (1996), Nonparametric estimation of low-flow frequencies, *J. Hydraul. Eng., 122*(1), 46–49.

[4] Akaike, H. (1974), A new look at the statistical model identification, *IEEE Trans. Autom. Control*, *AC 19*(6), 716–722.

[5] Ashkar, F., and J. Rousselle (1982), A multivariate statistical analysis of flood magnitude, duration, and volume, in *Statistical analysis of rainfall and runoff*, edited by V. P. Singh, pp. 659–669, Water Resource Publication, Fort Collins, Colo.

[6] Ashkar, F., and J. Rousselle (1983), Some remarks on the truncation used in partial flood series models, *Water Resources Research, 19*(2), 477-480.

[7] Azzalini, A. (1981), A note on the estimation of a distribution function and quantiles by a kernel method, *Biometrika, 68*, 326-328.

[8] Bowman, A., and A. Azzalini (1997), *Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-plus Illustrations*, Oxford Univ. Press, New York.

[9] Cunnane, C. (1985), Factors affecting choice of distribution for flood series. *J. Hydrol. Sci., 30*(1, 3), 25-36.

[10] Cunnane, C. (1987), Review of statistical models for flood frequency estimation, in *Hydrologic Frequency Modeling*, edited by V. P. Singh, pp. 49–95, Reidel, Dordrecht, The Netherlands.

[11] Cunnane, C. (1989), Statistical Distributions for Flood Frequency Analysis, Word Meteorological Organization Operational Hydrology, *Report No. 33*, *WMO-No. 718,* Geneva, Switzerland.

[12] De Michele, C., G. Salvadori, M. Canossi, A. Petaccia, and R. Rosso (2005), Bivariate statistical approach to check adequacy of dam spillway, *Journal of Hydrologic Engineering, ASCE 10*(1), 50–57.

[13] Dooge, J. C. E. (1986), Looking for hydrologic laws, *Water Resour. Res., 22*(9), 465-485. Efromovich, S. (1999), *Nonparametric Curve Estimation: Methods, Theory and Applications*, Springer-Verlag, New York.

[14] Favre, A.-C., S. El Adlouni, L. Perreault, N. Thie´ monge, and B. Bobe´ e (2004), Multivariate hydrological frequency analysis using copulas, *Water. Resour. Res., 40*, W01101. doi:10.1029/2003WR002456.

[15] Genest, C., and A. –C. Favre (2007), Everything you always wanted to know about copula Modeling but were afraid to ask, *Journal of Hydrologic Engineering, ASCE, 12*(4), 347–368.

[16] Genest, C., and L.-P. Rivest (1993), Statistical inference procedure for bivariate Archimedean copulas, *J. Am. Stat. Assoc., 88*, 1034– 1043.

[17] Ghosh, S., and P. P. Mujumdar (2007), Nonparametric methods for modeling GCM and scenario uncertainty in drought assessment, *Water Resources Research, 43*, W07405, doi:10.1029/2006WR005351.

[18] Grimaldi, S., and F. Serinaldi (2006), Asymmetric copula in multivariate flood frequency analysis, *Adv. Water Resour., 29*(8), 1115-1167.

[19] Gringorten, I. I. (1963), A plotting rule of extreme probability paper, *J. Geophys. Res., 68*(3), 813–814.

[20] Hall, M. J. (1984), *Urban Hydrology*, 299 pp., Elsevier, Barking.

[21] Higgins, J. J. (2004), *Introduction to Modern Nonparametric Statistics*, 366 pp., Brooks/Cole Thomson Learning, CA.

[22] Hoeffding, W. (1940), Maszstabinvariante Korrelationstheorie, Univ. Berlin, *Schr. Math. Angew. Math, 5*(3), 181–233.

[23] Joe, H. (1997), *Multivariate Models and Dependence Concepts*, 399 pp., Chapman and Hall, New York.

[24] Kidson, R., and K. S. Richards (2005), Flood frequency analysis: assumptions and alternatives, *Progress in Physical Geography, 29*(3), 392-410.

[25] Kim, T. W., J. B. Valdés, and C. Yoo (2003), Nonparametric approach for estimating return periods of droughts in arid regions, *Journal of Hydrologic Engineering, ASCE, 8*(5), 237–246.

[26] Kim, T. W., J. B. Valdés, and C. Yoo (2006), Nonparametric approach for bivariate drought characterization using Palmer drought index, *Journal of Hydrologic Engineering, ASCE, 11*(2), 134–143.

[27] Kite, G. W. (1978), *Frequency and Risk Analysis in Hydrology*, Water resource publications, Fort Collins, Colo.

[28] Krstanovic, P. F., and V. P. Singh (1987), A multivariate stochastic flood analysis using entropy, in Hydrologic Frequency Modeling, edited by V. P. Singh, pp. 515–539, Reidel, Dordrecht, The Netherlands.

[29] Lall, U. (1995), Recent Advances in Nonparametric Function Estimation, Supplement, U.S. Natl. Rep. Int. Union Geod. Geophys. *19911994, Rev. Geophys., 33*, 1093-1102.

[30] Lall, U., B. Rajagopalan, D. G. Tarboton (1996), A nonparametric wet/dry spell model for resampling daily precipitation, *Water Resour. Res., 32*(9), 2803 2823.

[31] Lall, U., Y.-I. Moon, and K. Bosworth (1993), Kernel flood frequency estimators: Bandwidth selection and kernel choice, *Water Resour. Res., 29*(4), 1003–1015.

[32] Linsley, R. K., M. A. Kohler, J. L. H. Paulhus (1975), *Applied Hydrology*, McGraw-Hill.

[33] Nelsen, R. B. (1999), *An Introduction to Copulas*, Springer, New York.

[34] Polansky, A. M., and E. R. Baker (2000), Multistage plug-in bandwidth selection for kernel distribution function estimates, *J. Stat. Comput. Simul., 65*, 63–80.

[35] Rao, A. R., and K. H. Hamed (2000), *Flood Frequency Analysis*, CRC, Boca Raton, Fla.

[36] Sackl, B., and H. Bergmann (1987), A bivariate flood model and its application, *in Hydrologic Frequency Modeling*, edited by V. P. Singh, pp. 571–582, Dreidel, Dordrecht, The Netherlands.

[37] Schwartz, S. C. (1967), Estimation of probability density by an orthogonal series, *Ann. Math. Statist., 38*, 1261-1265.

[38] Scott, D. W. (1992), *Multivariate Density Estimation, Theory, Practice, and Visualization*, John Wiley, New York.

[39] Serinaldi, F., and S. Grimaldi (2007), Fully nested 3-copula: procedure and application on hydrological data, *Journal of Hydrologic Engineering, ASCE, 12*(4), 420-430.

[40] Sharma, A. (2000), Seasonal to interseasonal rainfall probabilistic forecasts for improved water supply management: Part 3 - A nonparametric probabilistic forecast model, *J. Hydrol., 239*, 249–258.

[41] Sharma, A., D. G. Tarboton, and U. Lall (1997), Streamflow simulation: A nonparametric approach, *Water Resour. Res.*, *33*(2), 291–308

[42] Silverman, B. W. (1986), *Density Estimation for Statistics and Data Analysis*, 1st edition, Chapman and Hall, London.

[43] Singh, K., and V. P. Singh (1991), Derivation of bivariate probability density functions with exponential marginals, *Stochastic Hydrol. Hydraul., 5*, 55– 68.

[44] Sklar A. (1959), Fonction de re´partition a` n dimensions et leurs marges. Publications de L´Institute de Statistique, Universite´ de Paris 8, 229–231.

[45] Smakhtin, V. U. (2001), Low flow hydrology: a review, *J. Hydrol., 240*, 147–186.

[46] Snedecor, G. W. and W. G. Cochran (1989), *Statistical Methods, Eighth Edition*, Iowa State University Press.

[47] Tarboton, D. G., A. Sharma, and U. Lall (1998), Disaggregation procedures for stochastic hydrology based on nonparametric density estimation, *Water Resour. Res.*, *34*(1), 107– 119.

[48] Todorovic, P., and E. Zelenhasic (1970), A stochastic model for flood analysis, *Water Resour. Res., 6*(6), 1641–1648.

[49] Yue, S. (2000), The bivariate lognormal distribution to model a multivariate flood episode, *Hydrol. Process., 14*(14), 2575–2588.

[50] Yue, S. (2001), A bivariate gamma distribution for use in multivariate flood frequency analysis. *Hydrol. Process., 15*(6), 1033–1045.

[51] Yue, S., and P. Rasmussen (2002), Bivariate frequency analysis: discussion of some useful concepts in hydrological application, *Hydrolog. Process., 16*(14), 2881–2898.

[52] Yue, S., T. B. M. J. Ouarda, and B. Bobe´e (2001), A review of bivariate gamma distribution for hydrological application, *J. Hydrol., 246*, 1 – 18.

[53] Yue, S., T. B. M. J. Ouarda, B. Bobe´e, P. Legendre, P. Bruneau (1999), The Gumbel mixed model for flood frequency analysis, *J. Hydrol., 226*(1–2), 88–100.

[54] Yue, S., T. B. M. J. Ouarda, B. Bobe´e, P. Legendre, P. Bruneau (2002), Approach for describing statistical properties of flood hydrograph. *J. Hydrol. Eng., 7*(2), 147–153.

[55] Zhang, L., and V. P. Singh (2006), Bivariate flood frequency analysis using the copula method, *Journal of Hydrologic Engineering, ASCE, 11*(2), 150–164.

[56] Zhang, L., and V. P. Singh (2007), Trivariate flood frequency analysis using the Gumbel–Hougaard copula, *Journal of Hydrologic Engineering, ASCE, 12*(4), 431–439.

APPENDIX I. PROGRAM LISTING

All the subroutines described below are written in MATLAB 6.1, release 12.1. Some MATLAB functions are directly used from Statistical Tool Box. The data file name for streamflow data of Red River at Grand Forks is <ftperday70_grandfork.xls>. The data are collected from - http://waterdata.usgs.gov/nwis/nwisman/?site_no=05082500&amp.

Ia. Time series plots for peak flow, volume and duration for different threshold discharges and Pearson's linear correlation coefficients

```
clear all;
data=xlsread('ftperday70_grandfork');
streamflow=data(:,1);
year=1936;
no_day=0;
for i=1:length(streamflow)
   if (rem(year,4)==0)
      tot_day=366;
   else
      tot_day=365;
   end
   no_day=no_day+1;
   if (no_day>tot_day)
      year=year+1;
      no_day=1;
   end
   stream(i,1)=year;
   stream(i,2)=streamflow(i);
end
start=1;
for y=1936:2005
   if (rem(y,4)==0)
      tot_day=366;
   else
      tot_day=365;
```

```
end
y
end_day=start+tot_day-1;
temp=stream(start:end_day,2);
%PERCENTAGE DISCHARGE=p, Change the value
p=0.95;
p1=(tot_day)*p;
if (rem(p1,1)<0.5)
   p1=p1-rem(p1,1);
else
   p1=p1+1-rem(p1,1);
end
temp_sort=sort(temp);
thre=temp_sort(p1);
start=end_day+1;
[peak,cp]=max(temp)
peak_flow(y-1935)=peak;
diff=100;
i=cp;
while (diff>0)&(i>0)
   diff=temp(i)-thre;
   i=i-1;
end
i+1
flood_start_day(y-1935)=i+1;
diff=100;
i=cp;
while (diff>0)&(i<=tot_day)
   diff=temp(i)-thre;
   i=i+1;
end
i-1
flood_end_day(y-1935)=i-1;
duration(y-1935)=flood_end_day(y-1935)-flood_start_day(y-1935)+1;
peak_day(y-1935)=cp;
n=y-1935;
qis(n)=temp(flood_start_day(n));
qie(n)=temp(flood_end_day(n));
floodflow=temp(flood_start_day(n):flood_end_day(n));
vol(n)=sum(floodflow)-0.5*duration(n)*(qis(n)+qie(n));
```

```
    clear floodflow
end
year=1936:2005;
corrcoef(peak_flow,vol)
corrcoef(peak_flow,duration)
corrcoef(vol,duration)
subplot(3,1,1),plot(year,peak_flow);
subplot(3,1,2),plot(year,vol,'k');
subplot(3,1,3),plot(year,duration,'m');
```

Ib. Joint CDF of peak flow and volume for three different Archimedean copulas and estimation of Kendall's τ

```
clear all;
data=xlsread('ftperday70_grandfork');
streamflow=data(:,1);
year=1936;
no_day=0;
% for i=1:length(streamflow)
%     if (rem(year,4)==0)
%         tot_day=366;
%     else
%         tot_day=365;
%     end
%     no_day=no_day+1;
%     if (no_day>tot_day)
%         year=year+1;
%         no_day=1;
%     end
%     stream(i,1)=year;
%     stream(i,2)=streamflow(i);
% end
stream(:,2)=streamflow(:)*(0.3048^3)/1e6;
start=1;
for y=1936:2005
    if (rem(y,4)==0)
        tot_day=366;
    else
        tot_day=365;
```

```
    end
    y
    end_day=start+tot_day-1;
    temp=stream(start:end_day,2);
    start=end_day+1;
    [peak,cp]=max(temp)
    peak_flow(y-1935)=peak;
    diff=100;
    i=cp;
    while (diff>0)
        diff=temp(i)-temp(i-1);
        if (temp(i)>0.5*max(temp))
            diff=100;
        end
        i=i-1;
    end
    i
    flood_start_day(y-1935)=i;
    diff=100;
    i=cp;
    while (diff>0)
        diff=temp(i)-temp(i+1);
        if (temp(i)>0.5*max(temp))
            diff=100;
        end
        i=i+1;
    end
    i
    flood_end_day(y-1935)=i;
    duration(y-1935)=flood_end_day(y-1935)-flood_start_day(y-1935)+1;
    peak_day(y-1935)=cp;
    n=y-1935;
    qis(n)=temp(flood_start_day(n));
    qie(n)=temp(flood_end_day(n));
    floodflow=temp(flood_start_day(n):flood_end_day(n));
    vol(n)=sum(floodflow)-0.5*duration(n)*(qis(n)+qie(n));
    clear floodflow
end
year=1936:2005;
corrcoef(peak_flow,vol)
```

```
corrcoef(peak_flow,duration)
corrcoef(vol,duration)
[sup1,sup2,ran_peak,pdf_peak]=nonparapdf(peak_flow);
meanx=mean(peak_flow);
for i=1:length(peak_flow)
   ln_data(i)=log(peak_flow(i));
end
A=log(meanx)-mean(ln_data);
alpha=(1/(4*A))*(1+sqrt(1+((4*A)/3)));
beta=meanx/alpha;
x_gamma=ran_peak;
for k=1:length(x_gamma)
   y_gamma(k)=(1/((beta^alpha)*gamma(alpha)))*(x_gamma(k)^(alpha-1))*exp(-x_gamma(k)/beta);
end
pdf_peak=y_gamma;
[sup1,sup2,ran_vol,pdf_vol]=nonparapdf(vol);
peak_flow_values=ran_peak(2:101);
vol_values=ran_vol(2:101);
for i=2:101
   x_trap=ran_peak(1:i);
   y_trap=pdf_peak(1:i);
   CDF_peak(i-1)=trapz(x_trap,y_trap);
   if CDF_peak(i-1)>1
      CDF_peak(i-1)=1;
   end
   clear x_trap;
   clear y_trap;
end
%plot(peak_flow_values,CDF_peak);
for i=2:101
   x_trap=ran_vol(1:i);
   y_trap=pdf_vol(1:i);
   CDF_vol(i-1)=trapz(x_trap,y_trap);
   if CDF_vol(i-1)>1
      CDF_vol(i-1)=1;
   end
   clear x_trap;
   clear y_trap;
end
%plot(vol_values,CDF_vol);
```

```
%Kendall's Tau
si=0;
for j=2:length(peak_flow)
   for i=1:j
      if (((peak_flow(i)-peak_flow(j))*(vol(i)-vol(j)))<0)
         si=si-1;
      else
         si=si+1;
      end
   end
end
tau=(1/((factorial(length(peak_flow)))/((factorial(length(peak_flow)-2))*2)))*si;
x_trial=[-1:0.05:0.99];
for i=1:length(x_trial)
   y(i)=abs(((3*x_trial(i)-2)/x_trial(i))-((2/3)*((1-(1/x_trial(i)))^2)*log(1-x_trial(i)))-tau);
end
[Y,I]=min(y);
XI=x_trial(I);
fun = inline('((3*x-2)/x)-((2/3)*((1-(1/x))^2)*log(1-x))-0.7892') % use the value of tau
[theta,FVAL]=fzero(fun,XI);
for i=1:length(peak_flow_values)
   for j=1:length(vol_values)
      u=CDF_peak(i);
      v=CDF_vol(j);
      C_1(i,j)=(u*v)/(1-theta*(1-u)*(1-v));
   end
end
subplot(3,1,1),surf(peak_flow_values,vol_values,C_1)
%Cook Jhonson
theta=(2*tau)/(1-tau);
for i=1:length(peak_flow_values)
   for j=1:length(vol_values)
      u=CDF_peak(i);
      v=CDF_vol(j);
      C_2(i,j)=((u^(-theta))+(v^(-theta)))^(-1/theta);
   end
end
subplot(3,1,2),surf(peak_flow_values,vol_values,C_2);
%Gumbel-Hougaard
theta=1/(1-tau);
```

```
for i=1:length(peak_flow_values)
  for j=1:length(vol_values)
    u=CDF_peak(i);
    v=CDF_vol(j);
    C_3(i,j)=exp(-((-log(u))^theta +(-log(v))^theta)^(1/theta));
  end
end
subplot(3,1,3),surf(peak_flow_values,vol_values,C_3);
for i=1:length(peak_flow)
    n(i)=0;
    for k=1:length(peak_flow)
      if (peak_flow(i)<peak_flow(k))&(vol(i)<vol(k))
          n(i)=n(i)+1;
      end
    end
    C_obs(i)=(n(i)-0.44)/(length(peak_flow)+0.12);
  end
end
C_1_check=interp2(peak_flow_values,vol_values,C_1,peak_flow,vol);
C_2_check=interp2(peak_flow_values,vol_values,C_2,peak_flow,vol);
C_3_check=interp2(peak_flow_values,vol_values,C_3,peak_flow,vol);
```

Ic. Joint CDF of volume and duration for three different Archimedean copulas and estimation of Kendall's τ

```
clear all;
data=xlsread('ftperday70_grandfork');
streamflow=data(:,1);
year=1936;
no_day=0;
% for i=1:length(streamflow)
%   if (rem(year,4)==0)
%     tot_day=366;
%   else
%     tot_day=365;
%   end
%   no_day=no_day+1;
%   if (no_day>tot_day)
%     year=year+1;
```

```
%        no_day=1;
%     end
%     stream(i,1)=year;
%     stream(i,2)=streamflow(i);
% end
stream(:,2)=streamflow(:)*(0.3048^3)/1e6;
start=1;
for y=1936:2005
   if (rem(y,4)==0)
      tot_day=366;
   else
      tot_day=365;
   end
   y
   end_day=start+tot_day-1;
   temp=stream(start:end_day,2);
   start=end_day+1;
   [peak,cp]=max(temp)
   peak_flow(y-1935)=peak;
   diff=100;
   i=cp;
   while (diff>0)
      diff=temp(i)-temp(i-1);
      if (temp(i)>0.5*max(temp))
         diff=100;
      end
      i=i-1;
   end
   i
   flood_start_day(y-1935)=i;
   diff=100;
   i=cp;
   while (diff>0)
      diff=temp(i)-temp(i+1);
      if (temp(i)>0.5*max(temp))
         diff=100;
      end
      i=i+1;
   end
   i
```

```
    flood_end_day(y-1935)=i;
    duration(y-1935)=flood_end_day(y-1935)-flood_start_day(y-1935)+1;
    peak_day(y-1935)=cp;
    n=y-1935;
    qis(n)=temp(flood_start_day(n));
    qie(n)=temp(flood_end_day(n));
    floodflow=temp(flood_start_day(n):flood_end_day(n));
    vol(n)=sum(floodflow)-0.5*duration(n)*(qis(n)+qie(n));
    clear floodflow
end
year=1936:2005;
corrcoef(peak_flow,vol)
corrcoef(peak_flow,duration)
corrcoef(vol,duration)
[sup1,sup2,ran_vol,pdf_vol]=nonparapdf(vol);
[sup1,sup2,ran_duration,pdf_duration]=nonparapdf(duration);
vol_values=ran_vol(2:101);
duration_values=ran_duration(2:101);
for i=2:101
    x_trap=ran_vol(1:i);
    y_trap=pdf_vol(1:i);
    CDF_vol(i-1)=trapz(x_trap,y_trap);
    if CDF_vol(i-1)>1
        CDF_vol(i-1)=1;
    end
    clear x_trap;
    clear y_trap;
end
%plot(vol_values,CDF_vol);
for i=2:101
    x_trap=ran_duration(1:i);
    y_trap=pdf_duration(1:i);
    CDF_duration(i-1)=trapz(x_trap,y_trap);
    if CDF_duration(i-1)>1
        CDF_duration(i-1)=1;
    end
    clear x_trap;
    clear y_trap;
end
%plot(duration_values,CDF_duration);
```

```
%Kendall's Tau
si=0;
for j=2:length(vol)
   for i=1:j
      if (((vol(i)-vol(j))*(duration(i)-duration(j)))<0)
         si=si-1;
      else
         si=si+1;
      end
   end
end
tau=(1/((factorial(length(vol)))/((factorial(length(vol)-2))*2)))*si;
x_trial=[-1:0.05:0.99];
for i=1:length(x_trial)
   y(i)=abs(((3*x_trial(i)-2)/x_trial(i))-((2/3)*((1-(1/x_trial(i)))^2)*log(1-x_trial(i)))-tau);
end
[Y,I]=min(y);
XI=x_trial(I);
fun = inline('((3*x-2)/x)-((2/3)*((1-(1/x))^2)*log(1-x))-0.5756') % use the value of tau
[theta,FVAL]=fzero(fun,XI);
for i=1:length(vol_values)
   for j=1:length(duration_values)
      u=CDF_vol(i);
      v=CDF_duration(j);
      C(i,j)=(u*v)/(1-theta*(1-u)*(1-v));
   end
end
subplot(3,1,1),surf(vol_values,duration_values,C)
%Cook Jhonson
theta=(2*tau)/(1-tau);
for i=1:length(vol_values)
   for j=1:length(duration_values)
      u=CDF_vol(i);
      v=CDF_duration(j);
      C(i,j)=((u^(-theta))+(v^(-theta)))^(-1/theta);
   end
end
subplot(3,1,2),surf(vol_values,duration_values,C);
%Gumbel-Hougaard
theta=1/(1-tau);
```

```
for i=1:length(vol_values)
   for j=1:length(duration_values)
      u=CDF_vol(i);
      v=CDF_duration(j);
      C(i,j)=exp(-((-log(u))^theta +(-log(v))^theta)^(1/theta));
   end
end
subplot(3,1,3),surf(vol_values,duration_values,C);
```

Id. Joint CDF of peak flow and duration for three different Archimedean copulas and estimation of Kendall's τ

```
clear all;
data=xlsread('ftperday70_grandfork');
streamflow=data(:,1);
year=1936;
no_day=0;
% for i=1:length(streamflow)
%    if (rem(year,4)==0)
%       tot_day=366;
%    else
%       tot_day=365;
%    end
%    no_day=no_day+1;
%    if (no_day>tot_day)
%       year=year+1;
%       no_day=1;
%    end
%    stream(i,1)=year;
%    stream(i,2)=streamflow(i);
% end
stream(:,2)=streamflow(:)*(0.3048^3)/1e6;
start=1;
for y=1936:2005
   if (rem(y,4)==0)
      tot_day=366;
   else
      tot_day=365;
   end
```

```
    y
    end_day=start+tot_day-1;
    temp=stream(start:end_day,2);
    start=end_day+1;
    [peak,cp]=max(temp)
    peak_flow(y-1935)=peak;
    diff=100;
    i=cp;
    while (diff>0)
        diff=temp(i)-temp(i-1);
        if (temp(i)>0.5*max(temp))
            diff=100;
        end
        i=i-1;
    end
    i
    flood_start_day(y-1935)=i;
    diff=100;
    i=cp;
    while (diff>0)
        diff=temp(i)-temp(i+1);
        if (temp(i)>0.5*max(temp))
            diff=100;
        end
        i=i+1;
    end
    i
    flood_end_day(y-1935)=i;
    duration(y-1935)=flood_end_day(y-1935)-flood_start_day(y-1935)+1;
    peak_day(y-1935)=cp;
    n=y-1935;
    qis(n)=temp(flood_start_day(n));
    qie(n)=temp(flood_end_day(n));
    floodflow=temp(flood_start_day(n):flood_end_day(n));
    vol(n)=sum(floodflow)-0.5*duration(n)*(qis(n)+qie(n));
    clear floodflow
end
year=1936:2005;
corrcoef(peak_flow,vol)
corrcoef(peak_flow,duration)
```

```
corrcoef(vol,duration)
[sup1,sup2,ran_peak,pdf_peak]=nonparapdf(peak_flow);
meanx=mean(peak_flow);
for i=1:length(peak_flow)
  ln_data(i)=log(peak_flow(i));
end
A=log(meanx)-mean(ln_data);
alpha=(1/(4*A))*(1+sqrt(1+((4*A)/3)));
beta=meanx/alpha;
x_gamma=ran_peak;
for k=1:length(x_gamma)
  y_gamma(k)=(1/((beta^alpha)*gamma(alpha)))*(x_gamma(k)^(alpha-1))*exp(-x_gamma(k)/beta);
end
pdf_peak=y_gamma;
[sup1,sup2,ran_duration,pdf_duration]=nonparapdf(duration);
peak_flow_values=ran_peak(2:101);
duration_values=ran_duration(2:101);
for i=2:101
  x_trap=ran_peak(1:i);
  y_trap=pdf_peak(1:i);
  CDF_peak(i-1)=trapz(x_trap,y_trap);
  if CDF_peak(i-1)>1
    CDF_peak(i-1)=1;
  end
  clear x_trap;
  clear y_trap;
end
%plot(peak_flow_values,CDF_peak);
for i=2:101
  x_trap=ran_duration(1:i);
  y_trap=pdf_duration(1:i);
  CDF_duration(i-1)=trapz(x_trap,y_trap);
  if CDF_duration(i-1)>1
    CDF_duration(i-1)=1;
  end
  clear x_trap;
  clear y_trap;
end
%plot(duration_values,CDF_duration);
%Kendall's Tau
```

```
si=0;
for j=2:length(peak_flow)
    for i=1:j
        if (((peak_flow(i)-peak_flow(j))*(duration(i)-duration(j)))<0)
            si=si-1;
        else
            si=si+1;
        end
    end
end
tau=(1/((factorial(length(peak_flow)))/((factorial(length(peak_flow)-2))*2)))*si;
x_trial=[-1:0.05:0.99];
for i=1:length(x_trial)
    y(i)=abs(((3*x_trial(i)-2)/x_trial(i))-((2/3)*((1-(1/x_trial(i)))^2)*log(1-x_trial(i)))-tau);
end
[Y,I]=min(y);
XI=x_trial(I);
fun = inline('((3*x-2)/x)-((2/3)*((1-(1/x))^2)*log(1-x))-0.4033') % use the value of tau
[theta,FVAL]=fzero(fun,XI);
for i=1:length(peak_flow_values)
    for j=1:length(duration_values)
        u=CDF_peak(i);
        v=CDF_duration(j);
        C(i,j)=(u*v)/(1-theta*(1-u)*(1-v));
    end
end
subplot(3,1,1),surf(peak_flow_values,duration_values,C)
%Cook Jhonson
theta=(2*tau)/(1-tau);
for i=1:length(peak_flow_values)
    for j=1:length(duration_values)
        u=CDF_peak(i);
        v=CDF_duration(j);
        C(i,j)=((u^(-theta))+(v^(-theta)))^(-1/theta);
    end
end
subplot(3,1,2),surf(peak_flow_values,duration_values,C);
%Gumbel-Hougaard
theta=1/(1-tau);
for i=1:length(peak_flow_values)
```

```
    for j=1:length(duration_values)
        u=CDF_peak(i);
        v=CDF_duration(j);
        C(i,j)=exp(-((-log(u))^theta +(-log(v))^theta)^(1/theta));
    end
end
subplot(3,1,3),surf(peak_flow_values,duration_values,C);
```

Ie. Details of <cdfestimate.m>

```
function [x1,y1]=cdfestimate(x,y)
k=length(y);
for i=2:length(y)
    x_trap=x(1:i);
    y_trap=y(1:i);
    CDF_vol(i-1)=trapz(x_trap,y_trap);
    if CDF_vol(i-1)>1
        CDF_vol(i-1)=1;
    end
    value(i-1)=x(i);
    clear x_trap;
    clear y_trap;
end
x1=value;
y1=CDF_vol;
```

If. Details of <hist1.m>

```
function [no,xo] = hist(y,x)
%HIST  Histogram.
%   N = HIST(Y) bins the elements of Y into 10 equally spaced containers
%   and returns the number of elements in each container.  If Y is a
%   matrix, HIST works down the columns.
%
%   N = HIST(Y,M), where M is a scalar, uses M bins.
%
%   N = HIST(Y,X), where X is a vector, returns the distribution of Y
%   among bins with centers specified by X.  Note: Use HISTC if it is
%   more natural to specify bin edges instead.
```

```
%
%   [N,X] = HIST(...) also returns the position of the bin centers in X.
%
%   HIST(...) without output arguments produces a histogram bar plot of
%   the results.
%
%   See also HISTC.

%   J.N. Little 2-06-86
%   Revised 10-29-87, 12-29-88 LS
%   Revised 8-13-91 by cmt, 2-3-92 by ls.
%   Copyright 1984-2002 The MathWorks, Inc.
%   $Revision: 5.20 $  $Date: 2002/06/05 17:06:38 $

if nargin == 0
   error('Requires one or two input arguments.')
end
if nargin == 1
   x = 10;
end
if min(size(y))==1, y = y(:); end
if isstr(x) | isstr(y)
   error('Input arguments must be numeric.')
end

[m,n] = size(y);
if isempty(y),
   if length(x) == 1,
      x = 1:x;
   end
   nn = zeros(size(x)); % No elements to count
else
   if length(x) == 1
      miny = min(min(y));
      maxy = max(max(y));
            if miny == maxy,
                    miny = miny - floor(x/2) - 0.5;
                    maxy = maxy + ceil(x/2) - 0.5;
            end
      binwidth = (maxy - miny) ./ x;
```

```
        xx = miny + binwidth*(0:x);
        xx(length(xx)) = maxy;
        x = xx(1:length(xx)-1) + binwidth/2;
    else
        xx = x(:)';
        miny = min(min(y));
        maxy = max(max(y));
        binwidth = [diff(xx) 0];
        xx = [xx(1)-binwidth(1)/2 xx+binwidth/2];
        xx(1) = min(xx(1),miny);
        xx(end) = max(xx(end),maxy);
    end
    nbin = length(xx);
    % Shift bins so the internal is ( ] instead of [ ).
    xx = full(real(xx)); y = full(real(y)); % For compatibility
    bins = xx + max(eps,eps*abs(xx));
    nn = histc(y,[-inf bins],1);
    % Combine first bin with 2nd bin and last bin with next to last bin
    nn(2,:) = nn(2,:)+nn(1,:);
    nn(end-1,:) = nn(end-1,:)+nn(end,:);
    nn = nn(2:end-1,:);
end
diff1=max(y)-min(y);
diff2=diff1/10;
for i=1:size(nn)
    nn(i)=nn(i)/(max(size(y))*diff2);
end
if nargout == 0
    bar(x,nn,'hist');
else
  if min(size(y))==1, % Return row vectors if possible.
    no = nn';
    xo = x;
  else
    no = nn;
    xo = x';
  end
end
```

Ig. Details of <li.m>

```
function y1=li(x,y,x1);
dim=max(size(x));
xmax=max(x);
xmin=min(x);
x1u=(x1-xmin)/(xmax-xmin);
int ival1;
ival1=(x1u-rem(x1u,0.01))*100+1;
ival1=ival1-rem(ival1,1);
ival2=ival1+1;
% if ival1==0
%     y1=0;
% else
x2=x(ival1);
x3=x(ival2);
y2=y(ival1);
y3=y(ival2);
y1=y2+((y3-y2)/(x3-x2))*(x1-x2);
% end
```

Ih. Details of <pos.m>

```
function y=pos(l,m,n)
si=size(m);
dim=max(si);
for i=1:dim
    m(i)=m(i)-n;
    if m(i)<0
        m(i)=0;
    end
end
y=trapz(l,m);
```

Ii. Details of <trapz0.m>

```
function z=trapz0(x,y);
[x1,k]=sort(x);
dim=max(size(x));
```

```
y2=y;
for i=1:dim
   n=k(i);
   y(i)=y2(n);
end
z=trapz(x1,y);
```

Ij. Details of <valestimate.m>

```
function [y2]=valestimate(x,y,x2)
k=0;
for i=1:length(x)
   if (i~=1)
      if (x(i)~=x(i-1))
         k=k+1;
         x_mod(k)=x(i);
         y_mod(k)=y(i);
      end
   else
      k=k+1;
      x_mod(k)=x(i);
      y_mod(k)=y(i);
   end
end
y2=interp1(x_mod,y_mod,x2);
```

Ik. Estimation of chi-value

```
function [chi]=chi_compute(y1,y2,para);
j=0;
for i=1:10:length(y1)
   j=j+1;
   y1_mod(j)=y1(i);
   y2_mod(j)=y2(i);
end
for i=2:length(y1_mod)
   O(i-1)=y1_mod(i)-y1_mod(i-1);
   E(i-1)=y2_mod(i)-y2_mod(i-1);
end
```

```
ch=0;
length(y1_mod)
for i=1:length(O)
  ch=ch+((O(i)-E(i))^2)/E(i);
end
chi=ch;
```

II. Parametric and nonparametric estimation of *K(z)* for peak flow and volume

```
clear all;
data=xlsread('ftperday70_grandfork');
streamflow=data(:,1);
year=1936;
no_day=0;
% for i=1:length(streamflow)
%    if (rem(year,4)==0)
%        tot_day=366;
%    else
%        tot_day=365;
%    end
%    no_day=no_day+1;
%    if (no_day>tot_day)
%        year=year+1;
%        no_day=1;
%    end
%    stream(i,1)=year;
%    stream(i,2)=streamflow(i);
% end
stream(:,2)=streamflow(:)*(0.3048^3)/1e6;
start=1;
for y=1936:2005
  if (rem(y,4)==0)
    tot_day=366;
  else
    tot_day=365;
  end
  y
  end_day=start+tot_day-1;
  temp=stream(start:end_day,2);
```

```
    start=end_day+1;
    [peak,cp]=max(temp)
    peak_flow(y-1935)=peak;
    diff=100;
    i=cp;
    while (diff>0)
        diff=temp(i)-temp(i-1);
        if (temp(i)>0.5*max(temp))
            diff=100;
        end
        i=i-1;
    end
    i
    flood_start_day(y-1935)=i;
    diff=100;
    i=cp;
    while (diff>0)
        diff=temp(i)-temp(i+1);
        if (temp(i)>0.5*max(temp))
            diff=100;
        end
        i=i+1;
    end
    i
    flood_end_day(y-1935)=i;
    duration(y-1935)=flood_end_day(y-1935)-flood_start_day(y-1935)+1;
    peak_day(y-1935)=cp;
    n=y-1935;
    qis(n)=temp(flood_start_day(n));
    qie(n)=temp(flood_end_day(n));
    floodflow=temp(flood_start_day(n):flood_end_day(n));
    vol(n)=sum(floodflow)-0.5*duration(n)*(qis(n)+qie(n));
    clear floodflow
end
year=1936:2005;
corrcoef(peak_flow,vol)
corrcoef(peak_flow,duration)
corrcoef(vol,duration)
[sup1,sup2,ran_peak,pdf_peak]=nonparapdf(peak_flow);
meanx=mean(peak_flow);
```

```
for i=1:length(peak_flow)
   ln_data(i)=log(peak_flow(i));
end
A=log(meanx)-mean(ln_data);
alpha=(1/(4*A))*(1+sqrt(1+((4*A)/3)));
beta=meanx/alpha;
x_gamma=ran_peak;
for k=1:length(x_gamma)
   y_gamma(k)=(1/((beta^alpha)*gamma(alpha)))*(x_gamma(k)^(alpha-1))*exp(-x_gamma(k)/beta);
end
pdf_peak=y_gamma;
[sup1,sup2,ran_vol,pdf_vol]=nonparapdf(vol);
peak_flow_values=ran_peak(2:101);
vol_values=ran_vol(2:101);
for i=2:101
   x_trap=ran_peak(1:i);
   y_trap=pdf_peak(1:i);
   CDF_peak(i-1)=trapz(x_trap,y_trap);
   if CDF_peak(i-1)>1
      CDF_peak(i-1)=1;
   end
   clear x_trap;
   clear y_trap;
end
%plot(peak_flow_values,CDF_peak);
for i=2:101
   x_trap=ran_vol(1:i);
   y_trap=pdf_vol(1:i);
   CDF_vol(i-1)=trapz(x_trap,y_trap);
   if CDF_vol(i-1)>1
      CDF_vol(i-1)=1;
   end
   clear x_trap;
   clear y_trap;
end
%plot(vol_values,CDF_vol);
%Kendall's Tau
si=0;
for j=2:length(peak_flow)
   for i=1:j
```

```
        if (((peak_flow(i)-peak_flow(j))*(vol(i)-vol(j)))<0)
            si=si-1;
        else
            si=si+1;
        end
    end
end
tau=(1/((factorial(length(peak_flow)))/((factorial(length(peak_flow)-2))*2)))*si;
i1=0;
for k=1:length(peak_flow_values)
    for l=1:length(vol_values)
        n1=0;
        for i=1:length(peak_flow)
            if (peak_flow(i)<=peak_flow_values(k))&(vol(i)<=vol_values(l))
                n1=n1+1;
            end
        end
        C_obs(k,l)=(n1-0.44)/(length(peak_flow)+0.12);
    end
end
N=0;
i1=0
for k=1:length(peak_flow)
        N=N+1;
        n1=0;
        for i=1:length(peak_flow)
            if (peak_flow(i)<peak_flow(k))&(vol(i)<vol(k))
                n1=n1+1;
            end
        end
        z(k)=n1/(length(peak_flow)-1);
    end
z1=0.1:0.05:1;
N2=0;
for i=1:length(z1)-1
    n2=0;
    N2=0;
    for j=1:length(z)
        N2=N2+1;
        if z(j)<=z1(i)
```

```
            n2=n2+1;
        end
    end
    Kn(i)=n2/N2;
end
x_trial=[-1:0.05:0.99];
for i=1:length(x_trial)
    y(i)=abs(((3*x_trial(i)-2)/x_trial(i))-((2/3)*((1-(1/x_trial(i)))^2)*log(1-x_trial(i)))-tau);
end
[Y,I]=min(y);
XI=x_trial(I);
fun = inline('((3*x-2)/x)-((2/3)*((1-(1/x))^2)*log(1-x))-0.7892') % use the value of tau
[theta,FVAL]=fzero(fun,XI);
for i=1:length(peak_flow_values)
    for j=1:length(vol_values)
        u=CDF_peak(i);
        v=CDF_vol(j);
        C_1(i,j)=(u*v)/(1-theta*(1-u)*(1-v));
    end
end
for i=1:length(z1)-1
    phi(i)=log((1-theta*(1-z1(i)))/z1(i));
    phi_1(i)=(theta-1)/(z1(i)*(1-theta*(1-z1(i))));
    K_ali(i)=z1(i)-phi(i)/phi_1(i);
end
%subplot(3,1,1),surf(peak_flow_values,vol_values,C_1)
%Cook Jhonson
theta=(2*tau)/(1-tau);
for i=1:length(peak_flow_values)
    for j=1:length(vol_values)
        u=CDF_peak(i);
        v=CDF_vol(j);
        C_2(i,j)=((u^(-theta))+(v^(-theta)))^(-1/theta);
    end
end
for i=1:length(z1)-1
    phi(i)=(z1(i)^-(theta))-1;
    phi_1(i)=(-theta)*z1(i)^(-theta-1);
    K_cook(i)=z1(i)-phi(i)/phi_1(i);
end
```

```
%subplot(3,1,2),surf(peak_flow_values,vol_values,C_2);
%Gumbel-Hougaard
theta=1/(1-tau);
for i=1:length(peak_flow_values)
    for j=1:length(vol_values)
        u=CDF_peak(i);
        v=CDF_vol(j);
        C_3(i,j)=exp(-((-log(u))^theta +(-log(v))^theta)^(1/theta));
    end
end
for i=1:length(z1)-1
    phi(i)=(-log(z1(i)))^(theta);
    phi_1(i)=(theta/(z1(i)*log(z1(i))))*((-log(z1(i)))^(theta));
    K_gumbel(i)=z1(i)-phi(i)/phi_1(i);
end
%subplot(3,1,3),surf(peak_flow_values,vol_values,C_3);
%Ali
subplot(3,1,1),plot(Kn,Kn);
hold on;
subplot(3,1,1),scatter(Kn,K_ali,'r');
subplot(3,1,2),plot(Kn,Kn);
hold on;
subplot(3,1,2),scatter(Kn,K_cook,'r');
subplot(3,1,3),plot(Kn,Kn);
hold on;
subplot(3,1,3),scatter(Kn,K_gumbel,'r');
mse_ali=(((Kn-K_ali)*(Kn-K_ali)')/(length(K_ali)))
mse_cook=(((Kn-K_cook)*(Kn-K_cook)')/(length(K_cook)))
mse_gumbel=(((Kn-K_gumbel)*(Kn-K_gumbel)')/(length(K_gumbel)))
aic_ali=length(K_ali)*log(mse_ali)+2*1
aic_cook=length(K_cook)*log(mse_cook)+2*1
aic_gumbel=length(K_gumbel)*log(mse_gumbel)+2*1
```

Im. Parametric and nonparametric estimation of *K(z)* for volume and duration

```
clear all;
data=xlsread('ftperday70_grandfork');
streamflow=data(:,1);
year=1936;
```

```
no_day=0;
% for i=1:length(streamflow)
%     if (rem(year,4)==0)
%         tot_day=366;
%     else
%         tot_day=365;
%     end
%     no_day=no_day+1;
%     if (no_day>tot_day)
%         year=year+1;
%         no_day=1;
%     end
%     stream(i,1)=year;
%     stream(i,2)=streamflow(i);
% end
stream(:,2)=streamflow(:)*(0.3048^3)/1e6;
start=1;
for y=1936:2005
    if (rem(y,4)==0)
        tot_day=366;
    else
        tot_day=365;
    end
    y
    end_day=start+tot_day-1;
    temp=stream(start:end_day,2);
    start=end_day+1;
    [peak,cp]=max(temp)
    peak_flow(y-1935)=peak;
    diff=100;
    i=cp;
    while (diff>0)
        diff=temp(i)-temp(i-1);
        if (temp(i)>0.5*max(temp))
            diff=100;
        end
        i=i-1;
    end
    i
    flood_start_day(y-1935)=i;
```

```
    diff=100;
    i=cp;
    while (diff>0)
        diff=temp(i)-temp(i+1);
        if (temp(i)>0.5*max(temp))
            diff=100;
        end
        i=i+1;
    end
    i
    flood_end_day(y-1935)=i;
    duration(y-1935)=flood_end_day(y-1935)-flood_start_day(y-1935)+1;
    peak_day(y-1935)=cp;
    n=y-1935;
    qis(n)=temp(flood_start_day(n));
    qie(n)=temp(flood_end_day(n));
    floodflow=temp(flood_start_day(n):flood_end_day(n));
    vol(n)=sum(floodflow)-0.5*duration(n)*(qis(n)+qie(n));
    clear floodflow
end
year=1936:2005;
corrcoef(peak_flow,vol)
corrcoef(peak_flow,duration)
corrcoef(vol,duration)
[sup1,sup2,ran_vol,pdf_vol]=nonparapdf(vol);
[sup1,sup2,ran_duration,pdf_duration]=nonparapdf(duration);
vol_values=ran_vol(2:101);
duration_values=ran_duration(2:101);
for i=2:101
    x_trap=ran_vol(1:i);
    y_trap=pdf_vol(1:i);
    CDF_vol(i-1)=trapz(x_trap,y_trap);
    if CDF_vol(i-1)>1
        CDF_vol(i-1)=1;
    end
    clear x_trap;
    clear y_trap;
end
%plot(vol_values,CDF_vol);
for i=2:101
```

```
    x_trap=ran_duration(1:i);
    y_trap=pdf_duration(1:i);
    CDF_duration(i-1)=trapz(x_trap,y_trap);
    if CDF_duration(i-1)>1
        CDF_duration(i-1)=1;
    end
    clear x_trap;
    clear y_trap;
end
%plot(duration_values,CDF_duration);
%Kendall's Tau
si=0;
for j=2:length(vol)
    for i=1:j
        if (((vol(i)-vol(j))*(duration(i)-duration(j)))<0)
            si=si-1;
        else
            si=si+1;
        end
    end
end
tau=(1/((factorial(length(vol)))/((factorial(length(vol)-2))*2)))*si;
i1=0;
for k=1:length(vol_values)
    for l=1:length(duration_values)
        n1=0;
        for i=1:length(vol)
            if (vol(i)<=vol_values(k))&(duration(i)<=duration_values(l))
                n1=n1+1;
            end
        end
        C_obs(k,l)=(n1-0.44)/(length(vol)+0.12);
    end
end
N=0;
i1=0
for k=1:length(vol)
        N=N+1;
        n1=0;
        for i=1:length(vol)
```

```
            if (vol(i)<vol(k))&(duration(i)<duration(k))
                n1=n1+1;
            end
        end
        z(k)=n1/(length(vol)-1);
    end
z1=0.1:0.05:1;
N2=0;
for i=1:length(z1)-1
    n2=0;
    N2=0;
    for j=1:length(z)
        N2=N2+1;
        if z(j)<=z1(i)
            n2=n2+1;
        end
    end
    Kn(i)=n2/N2;
end
x_trial=[-1:0.05:0.99];
for i=1:length(x_trial)
    y(i)=abs(((3*x_trial(i)-2)/x_trial(i))-((2/3)*((1-(1/x_trial(i)))^2)*log(1-x_trial(i)))-tau);
end
[Y,I]=min(y);
XI=x_trial(I);
fun = inline('((3*x-2)/x)-((2/3)*((1-(1/x))^2)*log(1-x))-0.5756') % use the value of tau
[theta,FVAL]=fzero(fun,XI);
for i=1:length(vol_values)
    for j=1:length(duration_values)
        u=CDF_vol(i);
        v=CDF_duration(j);
        C_1(i,j)=(u*v)/(1-theta*(1-u)*(1-v));
    end
end
for i=1:length(z1)-1
    phi(i)=log((1-theta*(1-z1(i)))/z1(i));
    phi_1(i)=(theta-1)/(z1(i)*(1-theta*(1-z1(i))));
    K_ali(i)=z1(i)-phi(i)/phi_1(i);
end
%subplot(3,1,1),surf(vol_values,duration_values,C_1)
```

```
%Cook Jhonson
theta=(2*tau)/(1-tau);
for i=1:length(vol_values)
   for j=1:length(duration_values)
      u=CDF_vol(i);
      v=CDF_duration(j);
      C_2(i,j)=((u^(-theta))+(v^(-theta)))^(-1/theta);
   end
end
for i=1:length(z1)-1
   phi(i)=(z1(i)^-(theta))-1;
   phi_1(i)=(-theta)*z1(i)^(-theta-1);
   K_cook(i)=z1(i)-phi(i)/phi_1(i);
end
%subplot(3,1,2),surf(vol_values,duration_values,C_2);
%Gumbel-Hougaard
theta=1/(1-tau);
for i=1:length(vol_values)
   for j=1:length(duration_values)
      u=CDF_vol(i);
      v=CDF_duration(j);
      C_3(i,j)=exp(-((-log(u))^theta +(-log(v))^theta)^(1/theta));
   end
end
for i=1:length(z1)-1
   phi(i)=(-log(z1(i)))^(theta);
   phi_1(i)=(theta/(z1(i)*log(z1(i))))*((-log(z1(i)))^(theta));
   K_gumbel(i)=z1(i)-phi(i)/phi_1(i);
end
%subplot(3,1,3),surf(vol_values,duration_values,C_3);
%Ali
subplot(3,1,1),plot(Kn,Kn);
hold on;
subplot(3,1,1),scatter(Kn,K_ali,'r');
subplot(3,1,2),plot(Kn,Kn);
hold on;
subplot(3,1,2),scatter(Kn,K_cook,'r');
subplot(3,1,3),plot(Kn,Kn);
hold on;
subplot(3,1,3),scatter(Kn,K_gumbel,'r');
```

```
mse_ali=(((Kn-K_ali)*(Kn-K_ali)')/(length(K_ali)))
mse_cook=(((Kn-K_cook)*(Kn-K_cook)')/(length(K_cook)))
mse_gumbel=(((Kn-K_gumbel)*(Kn-K_gumbel)')/(length(K_gumbel)))
aic_ali=length(K_ali)*log(mse_ali)+2*1
aic_cook=length(K_cook)*log(mse_cook)+2*1
aic_gumbel=length(K_gumbel)*log(mse_gumbel)+2*1
```

In. Parametric and nonparametric estimation of *K(z)* for peak flow and duration

```
clear all;
data=xlsread('ftperday70_grandfork');
streamflow=data(:,1);
year=1936;
no_day=0;
% for i=1:length(streamflow)
%     if (rem(year,4)==0)
%         tot_day=366;
%     else
%         tot_day=365;
%     end
%     no_day=no_day+1;
%     if (no_day>tot_day)
%         year=year+1;
%         no_day=1;
%     end
%     stream(i,1)=year;
%     stream(i,2)=streamflow(i);
% end
stream(:,2)=streamflow(:)*(0.3048^3)/1e6;
start=1;
for y=1936:2005
    if (rem(y,4)==0)
        tot_day=366;
    else
        tot_day=365;
    end
    y
    end_day=start+tot_day-1;
    temp=stream(start:end_day,2);
```

```
    start=end_day+1;
    [peak,cp]=max(temp)
    peak_flow(y-1935)=peak;
    diff=100;
    i=cp;
    while (diff>0)
        diff=temp(i)-temp(i-1);
        if (temp(i)>0.5*max(temp))
            diff=100;
        end
        i=i-1;
    end
    i
    flood_start_day(y-1935)=i;
    diff=100;
    i=cp;
    while (diff>0)
        diff=temp(i)-temp(i+1);
        if (temp(i)>0.5*max(temp))
            diff=100;
        end
        i=i+1;
    end
    i
    flood_end_day(y-1935)=i;
    duration(y-1935)=flood_end_day(y-1935)-flood_start_day(y-1935)+1;
    peak_day(y-1935)=cp;
    n=y-1935;
    qis(n)=temp(flood_start_day(n));
    qie(n)=temp(flood_end_day(n));
    floodflow=temp(flood_start_day(n):flood_end_day(n));
    vol(n)=sum(floodflow)-0.5*duration(n)*(qis(n)+qie(n));
    clear floodflow
end
year=1936:2005;
corrcoef(peak_flow,vol)
corrcoef(peak_flow,duration)
corrcoef(vol,duration)
[sup1,sup2,ran_peak,pdf_peak]=nonparapdf(peak_flow);
meanx=mean(peak_flow);
```

```
for i=1:length(peak_flow)
   ln_data(i)=log(peak_flow(i));
end
A=log(meanx)-mean(ln_data);
alpha=(1/(4*A))*(1+sqrt(1+((4*A)/3)));
beta=meanx/alpha;
x_gamma=ran_peak;
for k=1:length(x_gamma)
   y_gamma(k)=(1/((beta^alpha)*gamma(alpha)))*(x_gamma(k)^(alpha-1))*exp(-x_gamma(k)/beta);
end
pdf_peak=y_gamma;
[sup1,sup2,ran_duration,pdf_duration]=nonparapdf(duration);
peak_flow_values=ran_peak(2:101);
duration_values=ran_duration(2:101);
for i=2:101
   x_trap=ran_peak(1:i);
   y_trap=pdf_peak(1:i);
   CDF_peak(i-1)=trapz(x_trap,y_trap);
   if CDF_peak(i-1)>1
      CDF_peak(i-1)=1;
   end
   clear x_trap;
   clear y_trap;
end
%plot(peak_flow_values,CDF_peak);
for i=2:101
   x_trap=ran_duration(1:i);
   y_trap=pdf_duration(1:i);
   CDF_duration(i-1)=trapz(x_trap,y_trap);
   if CDF_duration(i-1)>1
      CDF_duration(i-1)=1;
   end
   clear x_trap;
   clear y_trap;
end
%plot(duration_values,CDF_duration);
%Kendall's Tau
si=0;
for j=2:length(peak_flow)
   for i=1:j
```

```
        if (((peak_flow(i)-peak_flow(j))*(duration(i)-duration(j)))<0)
            si=si-1;
        else
            si=si+1;
        end
    end
end
tau=(1/((factorial(length(peak_flow)))/((factorial(length(peak_flow)-2))*2)))*si;
i1=0;
for k=1:length(peak_flow_values)
    for l=1:length(duration_values)
        n1=0;
        for i=1:length(peak_flow)
            if (peak_flow(i)<=peak_flow_values(k))&(duration(i)<=duration_values(l))
                n1=n1+1;
            end
        end
        C_obs(k,l)=(n1-0.44)/(length(peak_flow)+0.12);
    end
end
N=0;
i1=0
for k=1:length(peak_flow)
        N=N+1;
        n1=0;
        for i=1:length(peak_flow)
            if (peak_flow(i)<peak_flow(k))&(duration(i)<duration(k))
                n1=n1+1;
            end
        end
        z(k)=n1/(length(peak_flow)-1);
    end
z1=0.1:0.05:1;
N2=0;
for i=1:length(z1)-1
    n2=0;
    N2=0;
    for j=1:length(z)
        N2=N2+1;
        if z(j)<=z1(i)
```

```
            n2=n2+1;
        end
    end
    Kn(i)=n2/N2;
end
x_trial=[-1:0.05:0.99];
for i=1:length(x_trial)
    y(i)=abs(((3*x_trial(i)-2)/x_trial(i))-((2/3)*((1-(1/x_trial(i)))^2)*log(1-x_trial(i)))-tau);
end
[Y,I]=min(y);
XI=x_trial(I);
fun = inline('((3*x-2)/x)-((2/3)*((1-(1/x))^2)*log(1-x))-0.4033') % use the value of tau
[theta,FVAL]=fzero(fun,XI);
for i=1:length(peak_flow_values)
    for j=1:length(duration_values)
        u=CDF_peak(i);
        v=CDF_duration(j);
        C_1(i,j)=(u*v)/(1-theta*(1-u)*(1-v));
    end
end
for i=1:length(z1)-1
    phi(i)=log((1-theta*(1-z1(i)))/z1(i));
    phi_1(i)=(theta-1)/(z1(i)*(1-theta*(1-z1(i))));
    K_ali(i)=z1(i)-phi(i)/phi_1(i);
end
%subplot(3,1,1),surf(peak_flow_values,duration_values,C_1)
%Cook Jhonson
theta=(2*tau)/(1-tau);
for i=1:length(peak_flow_values)
    for j=1:length(duration_values)
        u=CDF_peak(i);
        v=CDF_duration(j);
        C_2(i,j)=((u^(-theta))+(v^(-theta)))^(-1/theta);
    end
end
for i=1:length(z1)-1
    phi(i)=(z1(i)^-(theta))-1;
    phi_1(i)=(-theta)*z1(i)^(-theta-1);
    K_cook(i)=z1(i)-phi(i)/phi_1(i);
end
```

```
%subplot(3,1,2),surf(peak_flow_values,duration_values,C_2);
%Gumbel-Hougaard
theta=1/(1-tau);
for i=1:length(peak_flow_values)
    for j=1:length(duration_values)
        u=CDF_peak(i);
        v=CDF_duration(j);
        C_3(i,j)=exp(-((-log(u))^theta +(-log(v))^theta)^(1/theta));
    end
end
for i=1:length(z1)-1
    phi(i)=(-log(z1(i)))^(theta);
    phi_1(i)=(theta/(z1(i)*log(z1(i))))*((-log(z1(i)))^(theta));
    K_gumbel(i)=z1(i)-phi(i)/phi_1(i);
end
%subplot(3,1,3),surf(peak_flow_values,duration_values,C_3);
%Ali
subplot(3,1,1),plot(Kn,Kn);
hold on;
subplot(3,1,1),scatter(Kn,K_ali,'r');
subplot(3,1,2),plot(Kn,Kn);
hold on;
subplot(3,1,2),scatter(Kn,K_cook,'r');
subplot(3,1,3),plot(Kn,Kn);
hold on;
subplot(3,1,3),scatter(Kn,K_gumbel,'r');
mse_ali=(((Kn-K_ali)*(Kn-K_ali)')/(length(K_ali)))
mse_cook=(((Kn-K_cook)*(Kn-K_cook)')/(length(K_cook)))
mse_gumbel=(((Kn-K_gumbel)*(Kn-K_gumbel)')/(length(K_gumbel)))
aic_ali=length(K_ali)*log(mse_ali)+2*1
aic_cook=length(K_cook)*log(mse_cook)+2*1
aic_gumbel=length(K_gumbel)*log(mse_gumbel)+2*1
```

I o. Conditional distribution for peak flow and volume

```
clear all;
data=xlsread('ftperday70_grandfork');
streamflow=data(:,1);
year=1936;
```

```
no_day=0;
% for i=1:length(streamflow)
%     if (rem(year,4)==0)
%         tot_day=366;
%     else
%         tot_day=365;
%     end
%     no_day=no_day+1;
%     if (no_day>tot_day)
%         year=year+1;
%         no_day=1;
%     end
%     stream(i,1)=year;
%     stream(i,2)=streamflow(i);
% end
stream(:,2)=streamflow(:)*(0.3048^3)/1e6;
start=1;
for y=1936:2005
    if (rem(y,4)==0)
        tot_day=366;
    else
        tot_day=365;
    end
    y
    end_day=start+tot_day-1;
    temp=stream(start:end_day,2);
    start=end_day+1;
    [peak,cp]=max(temp)
    peak_flow(y-1935)=peak;
    diff=100;
    i=cp;
    while (diff>0)
        diff=temp(i)-temp(i-1);
        if (temp(i)>0.5*max(temp))
            diff=100;
        end
        i=i-1;
    end
    i
    flood_start_day(y-1935)=i;
```

```
    diff=100;
    i=cp;
    while (diff>0)
        diff=temp(i)-temp(i+1);
        if (temp(i)>0.5*max(temp))
            diff=100;
        end
        i=i+1;
    end
    i
    flood_end_day(y-1935)=i;
    duration(y-1935)=flood_end_day(y-1935)-flood_start_day(y-1935)+1;
    peak_day(y-1935)=cp;
    n=y-1935;
    qis(n)=temp(flood_start_day(n));
    qie(n)=temp(flood_end_day(n));
    floodflow=temp(flood_start_day(n):flood_end_day(n));
    vol(n)=sum(floodflow)-0.5*duration(n)*(qis(n)+qie(n));
    clear floodflow
end
year=1936:2005;
corrcoef(peak_flow,vol)
corrcoef(peak_flow,duration)
corrcoef(vol,duration)
[sup1,sup2,ran_peak,pdf_peak]=nonparapdf(peak_flow);
meanx=mean(peak_flow);
for i=1:length(peak_flow)
    ln_data(i)=log(peak_flow(i));
end
A=log(meanx)-mean(ln_data);
alpha=(1/(4*A))*(1+sqrt(1+((4*A)/3)));
beta=meanx/alpha;
x_gamma=ran_peak;
for k=1:length(x_gamma)
    y_gamma(k)=(1/((beta^alpha)*gamma(alpha)))*(x_gamma(k)^(alpha-1))*exp(-x_gamma(k)/beta);
end
pdf_peak=y_gamma;
[sup1,sup2,ran_vol,pdf_vol]=nonparapdf(vol);
peak_flow_values=ran_peak(2:101);
vol_values=ran_vol(2:101);
```

```
for i=2:101
    x_trap=ran_peak(1:i);
    y_trap=pdf_peak(1:i);
    CDF_peak(i-1)=trapz(x_trap,y_trap);
    if CDF_peak(i-1)>1
        CDF_peak(i-1)=1;
    end
    clear x_trap;
    clear y_trap;
end
%plot(peak_flow_values,CDF_peak);
for i=2:101
    x_trap=ran_vol(1:i);
    y_trap=pdf_vol(1:i);
    CDF_vol(i-1)=trapz(x_trap,y_trap);
    if CDF_vol(i-1)>1
        CDF_vol(i-1)=1;
    end
    clear x_trap;
    clear y_trap;
end
%plot(vol_values,CDF_vol);
%Kendall's Tau
si=0;
for j=2:length(peak_flow)
    for i=1:j
        if (((peak_flow(i)-peak_flow(j))*(vol(i)-vol(j)))<0)
            si=si-1;
        else
            si=si+1;
        end
    end
end
tau=(1/((factorial(length(peak_flow)))/((factorial(length(peak_flow)-2))*2)))*si;
i1=0;
for k=1:length(peak_flow_values)
    for l=1:length(vol_values)
        n1=0;
        for i=1:length(peak_flow)
            if (peak_flow(i)<=peak_flow_values(k))&(vol(i)<=vol_values(l))
```

```
            n1=n1+1;
        end
    end
    C_obs(k,l)=(n1-0.44)/(length(peak_flow)+0.12);
  end
end
N=0;
i1=0
for k=1:length(peak_flow)
    N=N+1;
    n1=0;
    for i=1:length(peak_flow)
        if (peak_flow(i)<peak_flow(k))&(vol(i)<vol(k))
            n1=n1+1;
        end
    end
    z(k)=n1/(length(peak_flow)-1);
  end
z1=0.1:0.05:1;
N2=0;
for i=1:length(z1)-1
  n2=0;
  N2=0;
  for j=1:length(z)
    N2=N2+1;
    if z(j)<=z1(i)
        n2=n2+1;
    end
  end
  Kn(i)=n2/N2;
end
x_trial=[-1:0.05:0.99];
for i=1:length(x_trial)
  y(i)=abs(((3*x_trial(i)-2)/x_trial(i))-((2/3)*((1-(1/x_trial(i)))^2)*log(1-x_trial(i)))-tau);
end
[Y,I]=min(y);
XI=x_trial(I);
fun = inline('((3*x-2)/x)-((2/3)*((1-(1/x))^2)*log(1-x))-0.7892') % use the value of tau
[theta,FVAL]=fzero(fun,XI);
for i=1:length(peak_flow_values)
```

```
    for j=1:length(vol_values)
        u=CDF_peak(i);
        v=CDF_vol(j);
        C_1(i,j)=(u*v)/(1-theta*(1-u)*(1-v));
    end
end
for i=1:length(z1)-1
    phi(i)=log((1-theta*(1-z1(i)))/z1(i));
    phi_1(i)=(theta-1)/(z1(i)*(1-theta*(1-z1(i))));
    K_ali(i)=z1(i)-phi(i)/phi_1(i);
end
%%subplot(3,1,1),surf(peak_flow_values,vol_values,C_1)
%Cook Jhonson
theta=(2*tau)/(1-tau);
for i=1:length(peak_flow_values)
    for j=1:length(vol_values)
        u=CDF_peak(i);
        v=CDF_vol(j);
        C_2(i,j)=((u^(-theta))+(v^(-theta)))^(-1/theta);
    end
end
for i=1:length(z1)-1
    phi(i)=(z1(i)^-(theta))-1;
    phi_1(i)=(-theta)*z1(i)^(-theta-1);
    K_cook(i)=z1(i)-phi(i)/phi_1(i);
end
%%subplot(3,1,2),surf(peak_flow_values,vol_values,C_2);
%Gumbel-Hougaard
theta=1/(1-tau);
for i=1:length(peak_flow_values)
    for j=1:length(vol_values)
        u=CDF_peak(i);
        v=CDF_vol(j);
        C_3(i,j)=exp(-((-log(u))^theta +(-log(v))^theta)^(1/theta));
    end
end
for i=1:length(z1)-1
    phi(i)=(-log(z1(i)))^(theta);
    phi_1(i)=(theta/(z1(i)*log(z1(i))))*((-log(z1(i)))^(theta));
    K_gumbel(i)=z1(i)-phi(i)/phi_1(i);
```

```
end
%%subplot(3,1,3),surf(peak_flow_values,vol_values,C_3);
%Ali
%subplot(3,1,1),plot(Kn,Kn);
hold on;
%subplot(3,1,1),scatter(Kn,K_ali,'r');
%subplot(3,1,2),plot(Kn,Kn);
hold on;
%subplot(3,1,2),scatter(Kn,K_cook,'r');
%subplot(3,1,3),plot(Kn,Kn);
hold on;
%subplot(3,1,3),scatter(Kn,K_gumbel,'r');
mse_ali=(((Kn-K_ali)*(Kn-K_ali)')/(length(K_ali)))
mse_cook=(((Kn-K_cook)*(Kn-K_cook)')/(length(K_cook)))
mse_gumbel=(((Kn-K_gumbel)*(Kn-K_gumbel)')/(length(K_gumbel)))
aic_ali=length(K_ali)*log(mse_ali)+2*1
aic_cook=length(K_cook)*log(mse_cook)+2*1
aic_gumbel=length(K_gumbel)*log(mse_gumbel)+2*1
%f_copula=gradient(C_3,(vol_values(2)-vol_values(1)),(peak_flow_values(2)-peak_flow_values(1)));
con_vol=[1000,2000,4000,6000];
con_vol_cdf=interp1(vol_values,CDF_vol,con_vol);
hold off;
theta=1/(1-tau);
for j=1:length(con_vol)
  v=con_vol_cdf(j);
  for i=1:length(peak_flow_values)
    u=CDF_peak(i);
    condi_C_3(i)=(1/v)*(exp((-(((-log(u))^(theta)+(-log(v))^(theta))^(1/theta))));
  end
  plot(peak_flow_values,condi_C_3);
  hold on;
end
```

Ip. Conditional distribution for volume and duration

```
clear all;
data=xlsread('ftperday70_grandfork');
streamflow=data(:,1);
year=1936;
```

```
no_day=0;
% for i=1:length(streamflow)
%     if (rem(year,4)==0)
%         tot_day=366;
%     else
%         tot_day=365;
%     end
%     no_day=no_day+1;
%     if (no_day>tot_day)
%         year=year+1;
%         no_day=1;
%     end
%     stream(i,1)=year;
%     stream(i,2)=streamflow(i);
% end
stream(:,2)=streamflow(:)*(0.3048^3)/1e6;
start=1;
for y=1936:2005
    if (rem(y,4)==0)
        tot_day=366;
    else
        tot_day=365;
    end
    y
    end_day=start+tot_day-1;
    temp=stream(start:end_day,2);
    start=end_day+1;
    [peak,cp]=max(temp)
    peak_flow(y-1935)=peak;
    diff=100;
    i=cp;
    while (diff>0)
        diff=temp(i)-temp(i-1);
        if (temp(i)>0.5*max(temp))
            diff=100;
        end
        i=i-1;
    end
    i
    flood_start_day(y-1935)=i;
```

```
    diff=100;
    i=cp;
    while (diff>0)
        diff=temp(i)-temp(i+1);
        if (temp(i)>0.5*max(temp))
            diff=100;
        end
        i=i+1;
    end
    i
    flood_end_day(y-1935)=i;
    duration(y-1935)=flood_end_day(y-1935)-flood_start_day(y-1935)+1;
    peak_day(y-1935)=cp;
    n=y-1935;
    qis(n)=temp(flood_start_day(n));
    qie(n)=temp(flood_end_day(n));
    floodflow=temp(flood_start_day(n):flood_end_day(n));
    vol(n)=sum(floodflow)-0.5*duration(n)*(qis(n)+qie(n));
    clear floodflow
end
year=1936:2005;
corrcoef(peak_flow,vol)
corrcoef(peak_flow,duration)
corrcoef(vol,duration)
[sup1,sup2,ran_vol,pdf_vol]=nonparapdf(vol);
[sup1,sup2,ran_duration,pdf_duration]=nonparapdf(duration);
vol_values=ran_vol(2:101);
duration_values=ran_duration(2:101);
for i=2:101
    x_trap=ran_vol(1:i);
    y_trap=pdf_vol(1:i);
    CDF_vol(i-1)=trapz(x_trap,y_trap);
    if CDF_vol(i-1)>1
        CDF_vol(i-1)=1;
    end
    clear x_trap;
    clear y_trap;
end
%plot(vol_values,CDF_vol);
for i=2:101
```

```
    x_trap=ran_duration(1:i);
    y_trap=pdf_duration(1:i);
    CDF_duration(i-1)=trapz(x_trap,y_trap);
    if CDF_duration(i-1)>1
        CDF_duration(i-1)=1;
    end
    clear x_trap;
    clear y_trap;
end
%plot(duration_values,CDF_duration);
%Kendall's Tau
si=0;
for j=2:length(vol)
    for i=1:j
        if (((vol(i)-vol(j))*(duration(i)-duration(j)))<0)
            si=si-1;
        else
            si=si+1;
        end
    end
end
tau=(1/((factorial(length(vol)))/((factorial(length(vol)-2))*2)))*si;
i1=0;
for k=1:length(vol_values)
    for l=1:length(duration_values)
        n1=0;
        for i=1:length(vol)
            if (vol(i)<=vol_values(k))&(duration(i)<=duration_values(l))
                n1=n1+1;
            end
        end
        C_obs(k,l)=(n1-0.44)/(length(vol)+0.12);
    end
end
N=0;
i1=0
for k=1:length(vol)
        N=N+1;
        n1=0;
        for i=1:length(vol)
```

```
            if (vol(i)<vol(k))&(duration(i)<duration(k))
                n1=n1+1;
            end
        end
        z(k)=n1/(length(vol)-1);
    end
z1=0.1:0.05:1;
N2=0;
for i=1:length(z1)-1
    n2=0;
    N2=0;
    for j=1:length(z)
        N2=N2+1;
        if z(j)<=z1(i)
            n2=n2+1;
        end
    end
    Kn(i)=n2/N2;
end
x_trial=[-1:0.05:0.99];
for i=1:length(x_trial)
    y(i)=abs(((3*x_trial(i)-2)/x_trial(i))-((2/3)*((1-(1/x_trial(i)))^2)*log(1-x_trial(i)))-tau);
end
[Y,I]=min(y);
XI=x_trial(I);
fun = inline('((3*x-2)/x)-((2/3)*((1-(1/x))^2)*log(1-x))-0.5756') % use the value of tau
[theta,FVAL]=fzero(fun,XI);
for i=1:length(vol_values)
    for j=1:length(duration_values)
        u=CDF_vol(i);
        v=CDF_duration(j);
        C_1(i,j)=(u*v)/(1-theta*(1-u)*(1-v));
    end
end
for i=1:length(z1)-1
    phi(i)=log((1-theta*(1-z1(i)))/z1(i));
    phi_1(i)=(theta-1)/(z1(i)*(1-theta*(1-z1(i))));
    K_ali(i)=z1(i)-phi(i)/phi_1(i);
end
%%subplot(3,1,1),surf(vol_values,duration_values,C_1)
```

```
%Cook Jhonson
theta=(2*tau)/(1-tau);
for i=1:length(vol_values)
  for j=1:length(duration_values)
    u=CDF_vol(i);
    v=CDF_duration(j);
    C_2(i,j)=((u^(-theta))+(v^(-theta)))^(-1/theta);
  end
end
for i=1:length(z1)-1
  phi(i)=(z1(i)^-(theta))-1;
  phi_1(i)=(-theta)*z1(i)^(-theta-1);
  K_cook(i)=z1(i)-phi(i)/phi_1(i);
end
%%subplot(3,1,2),surf(vol_values,duration_values,C_2);
%Gumbel-Hougaard
theta=1/(1-tau);
for i=1:length(vol_values)
  for j=1:length(duration_values)
    u=CDF_vol(i);
    v=CDF_duration(j);
    C_3(i,j)=exp(-((-log(u))^theta +(-log(v))^theta)^(1/theta));
  end
end
for i=1:length(z1)-1
  phi(i)=(-log(z1(i)))^(theta);
  phi_1(i)=(theta/(z1(i)*log(z1(i))))*((-log(z1(i)))^(theta));
  K_gumbel(i)=z1(i)-phi(i)/phi_1(i);
end
%%subplot(3,1,3),surf(vol_values,duration_values,C_3);
%Ali
%subplot(3,1,1),plot(Kn,Kn);
hold on;
%subplot(3,1,1),scatter(Kn,K_ali,'r');
%subplot(3,1,2),plot(Kn,Kn);
hold on;
%subplot(3,1,2),scatter(Kn,K_cook,'r');
%subplot(3,1,3),plot(Kn,Kn);
hold on;
%subplot(3,1,3),scatter(Kn,K_gumbel,'r');
```

```
mse_ali=(((Kn-K_ali)*(Kn-K_ali)')/(length(K_ali)))
mse_cook=(((Kn-K_cook)*(Kn-K_cook)')/(length(K_cook)))
mse_gumbel=(((Kn-K_gumbel)*(Kn-K_gumbel)')/(length(K_gumbel)))
aic_ali=length(K_ali)*log(mse_ali)+2*1
aic_cook=length(K_cook)*log(mse_cook)+2*1
aic_gumbel=length(K_gumbel)*log(mse_gumbel)+2*1
con_duration=[20,40,60,80];
con_duration_cdf=interp1(duration_values,CDF_duration,con_duration);
hold off;
theta=1/(1-tau);
for j=1:length(con_duration)
  v=con_duration_cdf(j);
  for i=1:length(vol_values)
    u=CDF_vol(i);
    condi_C_3(i)=(1/v)*(exp((-((-log(u))^(theta)+(-log(v))^(theta))^(1/theta))));
  end
  plot(vol_values,condi_C_3);
  hold on;
end
```

Iq. Conditional distribution for peak flow and duration

```
clear all;
data=xlsread('ftperday70_grandfork');
streamflow=data(:,1);
year=1936;
no_day=0;
% for i=1:length(streamflow)
%    if (rem(year,4)==0)
%       tot_day=366;
%    else
%       tot_day=365;
%    end
%    no_day=no_day+1;
%    if (no_day>tot_day)
%       year=year+1;
%       no_day=1;
%    end
%    stream(i,1)=year;
```

```
%     stream(i,2)=streamflow(i);
% end
stream(:,2)=streamflow(:)*(0.3048^3)/1e6;
start=1;
for y=1936:2005
   if (rem(y,4)==0)
      tot_day=366;
   else
      tot_day=365;
   end
   y
   end_day=start+tot_day-1;
   temp=stream(start:end_day,2);
   start=end_day+1;
   [peak,cp]=max(temp)
   peak_flow(y-1935)=peak;
   diff=100;
   i=cp;
   while (diff>0)
      diff=temp(i)-temp(i-1);
      if (temp(i)>0.5*max(temp))
         diff=100;
      end
      i=i-1;
   end
   i
   flood_start_day(y-1935)=i;
   diff=100;
   i=cp;
   while (diff>0)
      diff=temp(i)-temp(i+1);
      if (temp(i)>0.5*max(temp))
         diff=100;
      end
      i=i+1;
   end
   i
   flood_end_day(y-1935)=i;
   duration(y-1935)=flood_end_day(y-1935)-flood_start_day(y-1935)+1;
   peak_day(y-1935)=cp;
```

```
    n=y-1935;
    qis(n)=temp(flood_start_day(n));
    qie(n)=temp(flood_end_day(n));
    floodflow=temp(flood_start_day(n):flood_end_day(n));
    vol(n)=sum(floodflow)-0.5*duration(n)*(qis(n)+qie(n));
    clear floodflow
end
year=1936:2005;
corrcoef(peak_flow,vol)
corrcoef(peak_flow,duration)
corrcoef(vol,duration)
[sup1,sup2,ran_peak,pdf_peak]=nonparapdf(peak_flow);
meanx=mean(peak_flow);
for i=1:length(peak_flow)
    ln_data(i)=log(peak_flow(i));
end
A=log(meanx)-mean(ln_data);
alpha=(1/(4*A))*(1+sqrt(1+((4*A)/3)));
beta=meanx/alpha;
x_gamma=ran_peak;
for k=1:length(x_gamma)
    y_gamma(k)=(1/((beta^alpha)*gamma(alpha)))*(x_gamma(k)^(alpha-1))*exp(-x_gamma(k)/beta);
end
pdf_peak=y_gamma;
[sup1,sup2,ran_duration,pdf_duration]=nonparapdf(duration);
peak_flow_values=ran_peak(2:101);
duration_values=ran_duration(2:101);
for i=2:101
    x_trap=ran_peak(1:i);
    y_trap=pdf_peak(1:i);
    CDF_peak(i-1)=trapz(x_trap,y_trap);
    if CDF_peak(i-1)>1
        CDF_peak(i-1)=1;
    end
    clear x_trap;
    clear y_trap;
end
%plot(peak_flow_values,CDF_peak);
for i=2:101
    x_trap=ran_duration(1:i);
```

```
    y_trap=pdf_duration(1:i);
    CDF_duration(i-1)=trapz(x_trap,y_trap);
    if CDF_duration(i-1)>1
        CDF_duration(i-1)=1;
    end
    clear x_trap;
    clear y_trap;
end
%plot(duration_values,CDF_duration);
%Kendall's Tau
si=0;
for j=2:length(peak_flow)
    for i=1:j
        if (((peak_flow(i)-peak_flow(j))*(duration(i)-duration(j)))<0)
            si=si-1;
        else
            si=si+1;
        end
    end
end
tau=(1/((factorial(length(peak_flow)))/((factorial(length(peak_flow)-2))*2)))*si;
i1=0;
for k=1:length(peak_flow_values)
    for l=1:length(duration_values)
        n1=0;
        for i=1:length(peak_flow)
            if (peak_flow(i)<=peak_flow_values(k))&(duration(i)<=duration_values(l))
                n1=n1+1;
            end
        end
        C_obs(k,l)=(n1-0.44)/(length(peak_flow)+0.12);
    end
end
N=0;
i1=0
for k=1:length(peak_flow)
        N=N+1;
        n1=0;
        for i=1:length(peak_flow)
            if (peak_flow(i)<peak_flow(k))&(duration(i)<duration(k))
```

```
            n1=n1+1;
          end
        end
        z(k)=n1/(length(peak_flow)-1);
    end
z1=0.1:0.05:1;
N2=0;
for i=1:length(z1)-1
    n2=0;
    N2=0;
    for j=1:length(z)
        N2=N2+1;
        if z(j)<=z1(i)
            n2=n2+1;
        end
    end
    Kn(i)=n2/N2;
end
x_trial=[-1:0.05:0.99];
for i=1:length(x_trial)
    y(i)=abs(((3*x_trial(i)-2)/x_trial(i))-((2/3)*((1-(1/x_trial(i)))^2)*log(1-x_trial(i)))-tau);
end
[Y,I]=min(y);
XI=x_trial(I);
fun = inline('((3*x-2)/x)-((2/3)*((1-(1/x))^2)*log(1-x))-0.4033') % use the value of tau
[theta,FVAL]=fzero(fun,XI);
for i=1:length(peak_flow_values)
    for j=1:length(duration_values)
        u=CDF_peak(i);
        v=CDF_duration(j);
        C_1(i,j)=(u*v)/(1-theta*(1-u)*(1-v));
    end
end
for i=1:length(z1)-1
    phi(i)=log((1-theta*(1-z1(i)))/z1(i));
    phi_1(i)=(theta-1)/(z1(i)*(1-theta*(1-z1(i))));
    K_ali(i)=z1(i)-phi(i)/phi_1(i);
end
%%subplot(3,1,1),surf(peak_flow_values,duration_values,C_1)
%Cook Jhonson
```

```
theta=(2*tau)/(1-tau);
for i=1:length(peak_flow_values)
    for j=1:length(duration_values)
        u=CDF_peak(i);
        v=CDF_duration(j);
        C_2(i,j)=((u^(-theta))+(v^(-theta)))^(-1/theta);
    end
end
for i=1:length(z1)-1
    phi(i)=(z1(i)^-(theta))-1;
    phi_1(i)=(-theta)*z1(i)^(-theta-1);
    K_cook(i)=z1(i)-phi(i)/phi_1(i);
end
%%subplot(3,1,2),surf(peak_flow_values,duration_values,C_2);
%Gumbel-Hougaard
theta=1/(1-tau);
for i=1:length(peak_flow_values)
    for j=1:length(duration_values)
        u=CDF_peak(i);
        v=CDF_duration(j);
        C_3(i,j)=exp(-((-log(u))^theta +(-log(v))^theta)^(1/theta));
    end
end
for i=1:length(z1)-1
    phi(i)=(-log(z1(i)))^(theta);
    phi_1(i)=(theta/(z1(i)*log(z1(i))))*((-log(z1(i)))^(theta));
    K_gumbel(i)=z1(i)-phi(i)/phi_1(i);
end
%%subplot(3,1,3),surf(peak_flow_values,duration_values,C_3);
%Ali
%subplot(3,1,1),plot(Kn,Kn);
hold on;
%subplot(3,1,1),scatter(Kn,K_ali,'r');
%subplot(3,1,2),plot(Kn,Kn);
hold on;
%subplot(3,1,2),scatter(Kn,K_cook,'r');
%subplot(3,1,3),plot(Kn,Kn);
hold on;
%subplot(3,1,3),scatter(Kn,K_gumbel,'r');
mse_ali=(((Kn-K_ali)*(Kn-K_ali)')/(length(K_ali)))
```

```
mse_cook=(((Kn-K_cook)*(Kn-K_cook)')/(length(K_cook)))
mse_gumbel=(((Kn-K_gumbel)*(Kn-K_gumbel)')/(length(K_gumbel)))
aic_ali=length(K_ali)*log(mse_ali)+2*1
aic_cook=length(K_cook)*log(mse_cook)+2*1
aic_gumbel=length(K_gumbel)*log(mse_gumbel)+2*1
%f_copula=gradient(C_3,(peak_flow_values(2)-peak_flow_values(1)),(duration_values(2)-duration_values(1)));
con_duration=[20,40,60,80];
con_duration_cdf=interp1(duration_values,CDF_duration,con_duration);
hold off;
%Gumbel Hoggard Conditional
theta=1/(1-tau);
for j=1:length(con_duration)
  v=con_duration_cdf(j);
  for i=1:length(peak_flow_values)
    u=CDF_peak(i);
    condi_C_3(i)=(1/v)*(exp((-(((-log(u))^(theta)+(-log(v))^(theta))^(1/theta))));
  end
  plot(peak_flow_values,condi_C_3);
  hold on;
end
```

Ir. Joint return period of peak flow and volume

```
clear all;
data=xlsread('ftperday70_grandfork');
streamflow=data(:,1);
year=1936;
no_day=0;
% for i=1:length(streamflow)
%    if (rem(year,4)==0)
%       tot_day=366;
%    else
%       tot_day=365;
%    end
%    no_day=no_day+1;
%    if (no_day>tot_day)
%       year=year+1;
%       no_day=1;
%    end
```

```
%    stream(i,1)=year;
%    stream(i,2)=streamflow(i);
% end
stream(:,2)=streamflow(:)*(0.3048^3)/1e6;
start=1;
for y=1936:2005
   if (rem(y,4)==0)
      tot_day=366;
   else
      tot_day=365;
   end
   y
   end_day=start+tot_day-1;
   temp=stream(start:end_day,2);
   start=end_day+1;
   [peak,cp]=max(temp)
   peak_flow(y-1935)=peak;
   diff=100;
   i=cp;
   while (diff>0)
      diff=temp(i)-temp(i-1);
      if (temp(i)>0.5*max(temp))
         diff=100;
      end
      i=i-1;
   end
   i
   flood_start_day(y-1935)=i;
   diff=100;
   i=cp;
   while (diff>0)
      diff=temp(i)-temp(i+1);
      if (temp(i)>0.5*max(temp))
         diff=100;
      end
      i=i+1;
   end
   i
   flood_end_day(y-1935)=i;
   duration(y-1935)=flood_end_day(y-1935)-flood_start_day(y-1935)+1;
```

```
    peak_day(y-1935)=cp;
    n=y-1935;
    qis(n)=temp(flood_start_day(n));
    qie(n)=temp(flood_end_day(n));
    floodflow=temp(flood_start_day(n):flood_end_day(n));
    vol(n)=sum(floodflow)-0.5*duration(n)*(qis(n)+qie(n));
    clear floodflow
end
year=1936:2005;
corrcoef(peak_flow,vol)
corrcoef(peak_flow,duration)
corrcoef(vol,duration)
[sup1,sup2,ran_peak,pdf_peak]=nonparapdf(peak_flow);
meanx=mean(peak_flow);
for i=1:length(peak_flow)
    ln_data(i)=log(peak_flow(i));
end
A=log(meanx)-mean(ln_data);
alpha=(1/(4*A))*(1+sqrt(1+((4*A)/3)));
beta=meanx/alpha;
x_gamma=ran_peak;
for k=1:length(x_gamma)
    y_gamma(k)=(1/((beta^alpha)*gamma(alpha)))*(x_gamma(k)^(alpha-1))*exp(-x_gamma(k)/beta);
end
pdf_peak=y_gamma;
[sup1,sup2,ran_vol,pdf_vol]=nonparapdf(vol);
peak_flow_values=ran_peak(2:101);
vol_values=ran_vol(2:101);
for i=2:101
    x_trap=ran_peak(1:i);
    y_trap=pdf_peak(1:i);
    CDF_peak(i-1)=trapz(x_trap,y_trap);
    if CDF_peak(i-1)>1
        CDF_peak(i-1)=1;
    end
    clear x_trap;
    clear y_trap;
end
%plot(peak_flow_values,CDF_peak);
for i=2:101
```

```
    x_trap=ran_vol(1:i);
    y_trap=pdf_vol(1:i);
    CDF_vol(i-1)=trapz(x_trap,y_trap);
    if CDF_vol(i-1)>1
        CDF_vol(i-1)=1;
    end
    clear x_trap;
    clear y_trap;
end
%plot(vol_values,CDF_vol);
%Kendall's Tau
si=0;
for j=2:length(peak_flow)
    for i=1:j
        if (((peak_flow(i)-peak_flow(j))*(vol(i)-vol(j)))<0)
            si=si-1;
        else
            si=si+1;
        end
    end
end
tau=(1/((factorial(length(peak_flow)))/((factorial(length(peak_flow)-2))*2)))*si;
i1=0;
for k=1:length(peak_flow_values)
    for l=1:length(vol_values)
        n1=0;
        for i=1:length(peak_flow)
            if (peak_flow(i)<=peak_flow_values(k))&(vol(i)<=vol_values(l))
                n1=n1+1;
            end
        end
        C_obs(k,l)=(n1-0.44)/(length(peak_flow)+0.12);
    end
end
N=0;
i1=0
for k=1:length(peak_flow)
        N=N+1;
        n1=0;
        for i=1:length(peak_flow)
```

```
          if (peak_flow(i)<peak_flow(k))&(vol(i)<vol(k))
             n1=n1+1;
          end
       end
       z(k)=n1/(length(peak_flow)-1);
    end
z1=0.1:0.05:1;
N2=0;
for i=1:length(z1)-1
    n2=0;
    N2=0;
    for j=1:length(z)
       N2=N2+1;
       if z(j)<=z1(i)
          n2=n2+1;
       end
    end
    Kn(i)=n2/N2;
end
x_trial=[-1:0.05:0.99];
for i=1:length(x_trial)
    y(i)=abs(((3*x_trial(i)-2)/x_trial(i))-((2/3)*((1-(1/x_trial(i)))^2)*log(1-x_trial(i)))-tau);
end
[Y,I]=min(y);
XI=x_trial(I);
fun = inline('((3*x-2)/x)-((2/3)*((1-(1/x))^2)*log(1-x))-0.7892') % use the value of tau
[theta,FVAL]=fzero(fun,XI);
for i=1:length(peak_flow_values)
    for j=1:length(vol_values)
       u=CDF_peak(i);
       v=CDF_vol(j);
       C_1(i,j)=(u*v)/(1-theta*(1-u)*(1-v));
    end
end
for i=1:length(z1)-1
    phi(i)=log((1-theta*(1-z1(i)))/z1(i));
    phi_1(i)=(theta-1)/(z1(i)*(1-theta*(1-z1(i))));
    K_ali(i)=z1(i)-phi(i)/phi_1(i);
end
%%subplot(3,1,1),surf(peak_flow_values,vol_values,C_1)
```

```
%Cook Jhonson
theta=(2*tau)/(1-tau);
for i=1:length(peak_flow_values)
    for j=1:length(vol_values)
        u=CDF_peak(i);
        v=CDF_vol(j);
        C_2(i,j)=((u^(-theta))+(v^(-theta)))^(-1/theta);
    end
end
for i=1:length(z1)-1
    phi(i)=(z1(i)^-(theta))-1;
    phi_1(i)=(-theta)*z1(i)^(-theta-1);
    K_cook(i)=z1(i)-phi(i)/phi_1(i);
end
%%subplot(3,1,2),surf(peak_flow_values,vol_values,C_2);
%Gumbel-Hougaard
theta=1/(1-tau);
for i=1:length(peak_flow_values)
    for j=1:length(vol_values)
        u=CDF_peak(i);
        v=CDF_vol(j);
        C_3(i,j)=exp(-((-log(u))^theta +(-log(v))^theta)^(1/theta));
    end
end
for i=1:length(z1)-1
    phi(i)=(-log(z1(i)))^(theta);
    phi_1(i)=(theta/(z1(i)*log(z1(i))))*((-log(z1(i)))^(theta));
    K_gumbel(i)=z1(i)-phi(i)/phi_1(i);
end
%%subplot(3,1,3),surf(peak_flow_values,vol_values,C_3);
%Ali
%subplot(3,1,1),plot(Kn,Kn);
hold on;
%subplot(3,1,1),scatter(Kn,K_ali,'r');
%subplot(3,1,2),plot(Kn,Kn);
hold on;
%subplot(3,1,2),scatter(Kn,K_cook,'r');
%subplot(3,1,3),plot(Kn,Kn);
hold on;
%subplot(3,1,3),scatter(Kn,K_gumbel,'r');
```

```
mse_ali=(((Kn-K_ali)*(Kn-K_ali)')/(length(K_ali)))
mse_cook=(((Kn-K_cook)*(Kn-K_cook)')/(length(K_cook)))
mse_gumbel=(((Kn-K_gumbel)*(Kn-K_gumbel)')/(length(K_gumbel)))
aic_ali=length(K_ali)*log(mse_ali)+2*1
aic_cook=length(K_cook)*log(mse_cook)+2*1
aic_gumbel=length(K_gumbel)*log(mse_gumbel)+2*1
%f_copula=gradient(C_3,(vol_values(2)-vol_values(1)),(peak_flow_values(2)-peak_flow_values(1)));
theta=1/(1-tau);
for i=1:length(peak_flow_values)
  for j=1:length(vol_values)
    u=CDF_peak(i);
    v=CDF_vol(j);
    C_3(i,j)=exp(-((-log(u))^theta +(-log(v))^theta)^(1/theta));
    RP(i,j)=1/(1-C_3(i,j));
  end
end
surf(peak_flow_values,vol_values,RP);
```

Is. Fitting PDF for peak flow

```
clear all;
data=xlsread('ftperday70_grandfork');
streamflow=data(:,1);
year=1936;
no_day=0;
% for i=1:length(streamflow)
%    if (rem(year,4)==0)
%       tot_day=366;
%    else
%       tot_day=365;
%    end
%    no_day=no_day+1;
%    if (no_day>tot_day)
%       year=year+1;
%       no_day=1;
%    end
%    stream(i,1)=year;
%    stream(i,2)=streamflow(i);
% end
```

```
stream(:,2)=streamflow(:)*(0.3048^3)/1e6;
start=1;
for y=1936:2005
   if (rem(y,4)==0)
      tot_day=366;
   else
      tot_day=365;
   end
   y
   end_day=start+tot_day-1;
   temp=stream(start:end_day,2);
   start=end_day+1;
   [peak,cp]=max(temp)
   peak_flow(y-1935)=peak;
   diff=100;
   i=cp;
   while (diff>0)
      diff=temp(i)-temp(i-1);
      if (temp(i)>0.5*max(temp))
         diff=100;
      end
      i=i-1;
   end
   i
   flood_start_day(y-1935)=i;
   diff=100;
   i=cp;
   while (diff>0)
      diff=temp(i)-temp(i+1);
      if (temp(i)>0.5*max(temp))
         diff=100;
      end
      i=i+1;
   end
   i
   flood_end_day(y-1935)=i;
   duration(y-1935)=flood_end_day(y-1935)-flood_start_day(y-1935)+1;
   peak_day(y-1935)=cp;
   n=y-1935;
   qis(n)=temp(flood_start_day(n));
```

```
    qie(n)=temp(flood_end_day(n));
    floodflow=temp(flood_start_day(n):flood_end_day(n));
    vol(n)=sum(floodflow)-0.5*duration(n)*(qis(n)+qie(n));
    clear floodflow
end
year=1936:2005;
corrcoef(peak_flow,vol)
corrcoef(peak_flow,duration)
corrcoef(vol,duration)
[sup1,sup2,ran,pdf]=nonparapdf(peak_flow);
[val_ortho,cdf_ortho]=cdfestimate(ran,pdf);
hold on;
%plot(val_ortho,cdf_ortho,'m*');
%kernel
xi=ran;
[f]=ksdensity(peak_flow,xi);
f=f';
[val_kernel,cdf_kernel]=cdfestimate(xi,f);
hold on;
%plot(val_kernel,cdf_kernel,'y*');
subplot(3,1,1),hist1(peak_flow);
hold on;
subplot(3,1,1),plot(ran,pdf,'k');
subplot(3,1,1),plot(ran,f,'b');
%Weibull
peak_flow_sort1=sort(peak_flow);
val_weibull=val_ortho;
for k=1:length(val_weibull)
    j=0;
    for i=1:length(peak_flow)
        if (peak_flow(i)<=val_weibull(k))
            j=j+1;
        end
    end
    cdf_weibull(k)=j/(length(peak_flow)+1);
end
%plot(val_weibull,cdf_weibull);
%%subplot(3,1,1),plot(xi,f);
%%subplot(3,1,1),plot(ran,pdf,'m');
x=ran;
```

```
y=normpdf(x,mean(peak_flow),std(peak_flow));
lambda=1/(mean(peak_flow));
x_expo=ran;
for k=1:length(x_expo)
   y_expo(k)=lambda*exp((-lambda)*x_expo(k));
end
subplot(3,1,1),plot(x_expo,y_expo,'r');
[val_expo,cdf_expo]=cdfestimate(x_expo,y_expo);
hold on;
%plot(val_expo,cdf_expo,'k*');
for i=1:length(peak_flow)
   ln_data(i)=log(peak_flow(i));
end
for i=1:length(ran)
   log_ran(i)=log(ran(i));
end
meany=mean(ln_data);
stdy=std(ln_data);
for i=1:length(ran)
   pdf_log_normal(i)=normpdf(log_ran(i),meany,stdy)/ran(i);
end
x_ln_normal=ran;
y_ln_normal=pdf_log_normal;
[val_ln_normal,cdf_ln_normal]=cdfestimate(x_ln_normal,y_ln_normal);
subplot(3,1,1),plot(x_ln_normal,y_ln_normal,'g');
%%subplot(3,1,1),plot(x,y,'r');
%%subplot(3,1,1),plot(x_expo,y_expo,'c');
meanx=mean(peak_flow);
for i=1:length(peak_flow)
   ln_data(i)=log(peak_flow(i));
end
A=log(meanx)-mean(ln_data);
alpha=(1/(4*A))*(1+sqrt(1+((4*A)/3)));
beta=meanx/alpha;
x_gamma=ran;
for k=1:length(x_gamma)
   y_gamma(k)=(1/((beta^alpha)*gamma(alpha)))*(x_gamma(k)^(alpha-1))*exp(-x_gamma(k)/beta);
end
[val_gamma,cdf_gamma]=cdfestimate(x_gamma,y_gamma);
hold on;
```

```
subplot(3,1,1),plot(x_gamma,y_gamma,'m');
%plot(val_gamma,cdf_gamma,'r*');
%%subplot(3,1,1),plot(x_gamma,y_gamma,'g');
alpha=(sqrt(6)/pi)*std(peak_flow);
u=meanx-0.577*alpha;
x_gumbel=ran;
for k=1:length(x_gumbel)
   y_gumbel(k)=(1/alpha)*exp((-(x_gumbel(k)-u)/alpha)-exp(-(x_gumbel(k)-u)/alpha));
end
[val_gumbel,cdf_gumbel]=cdfestimate(x_gumbel,y_gumbel);
hold on;
subplot(3,1,1),plot(x_gumbel,y_gumbel,'y');
%plot(val_gumbel,cdf_gumbel,'g*');
%%subplot(3,1,1),plot(x_gumbel,y_gumbel,'k');
c_y=skewness(ln_data);
p=(2/c_y)^2;
beta=std(ln_data)/sqrt(p);
alpha=mean(ln_data)-p*beta;
x_lpt3=ran;
for k=1:length(x_lpt3)
   ln_trans(k)=log(x_lpt3(k));
   y_lpt3(k)=(1/(beta*gamma(p)))*(((ln_trans(k)-alpha)/beta)^(p-1))*exp(-(ln_trans(k)-alpha)/beta)/x_lpt3(k);
end
[val_lpt3,cdf_lpt3]=cdfestimate(x_lpt3,y_lpt3);
hold on;
%plot(val_lpt3,cdf_lpt3,'c*');
% q_weibull=val_weibull(:);
% q_weibull=q_weibull';
% q_kernel=valestimate(cdf_kernel,val_kernel,cdf_weibull);
% q_ortho=valestimate(cdf_ortho,val_ortho,cdf_weibull);
% q_expo=valestimate(cdf_expo,val_expo,cdf_weibull);
% q_gamma=valestimate(cdf_gamma,val_gamma,cdf_weibull);
% q_gumbel=valestimate(cdf_gumbel,val_gumbel,cdf_weibull);
% %q_lpt3=valestimate(cdf_lpt3,val_lpt3,cdf_weibull);
mse_kernel=(((cdf_kernel-cdf_weibull)*(cdf_kernel-cdf_weibull)')/(length(cdf_kernel)))
mse_ortho=(((cdf_ortho-cdf_weibull)*(cdf_ortho-cdf_weibull)')/(length(cdf_ortho)))
mse_expo=(((cdf_expo-cdf_weibull)*(cdf_expo-cdf_weibull)')/(length(cdf_expo)-1))
mse_gamma=(((cdf_gamma-cdf_weibull)*(cdf_gamma-cdf_weibull)')/(length(cdf_gamma)-2))
mse_gumbel=(((cdf_gumbel-cdf_weibull)*(cdf_gumbel-cdf_weibull)')/(length(cdf_gumbel)-2))
mse_ln_normal=(((cdf_ln_normal-cdf_weibull)*(cdf_ln_normal-cdf_weibull)')/(length(cdf_ln_normal)-2))
```

```
%mse_lpt3=(((cdf_lpt3-cdf_weibull)*(cdf_lpt3-cdf_weibull)')/(length(cdf_lpt3)-3))
aic_kernel=length(cdf_kernel)*log(mse_kernel)+2*0
aic_ortho=length(cdf_ortho)*log(mse_ortho)+2*0
aic_expo=length(cdf_expo)*log(mse_expo)+2*1
aic_gamma=length(cdf_gamma)*log(mse_gamma)+2*2
aic_gumbel=length(cdf_gumbel)*log(mse_gumbel)+2*2
aic_ln_normal=length(cdf_ln_normal)*log(mse_ln_normal)+2*2
%aic_lpt3=length(cdf_lpt3)*log(mse_lpt3)+2*3
para=2;
[chi_gamma]=chi_compute(cdf_weibull,cdf_gamma,para);
chi_gamma
```

It. Fitting PDF for volume

```
clear all;
data=xlsread('ftperday70_grandfork');
streamflow=data(:,1);
year=1936;
no_day=0;
% for i=1:length(streamflow)
%     if (rem(year,4)==0)
%         tot_day=366;
%     else
%         tot_day=365;
%     end
%     no_day=no_day+1;
%     if (no_day>tot_day)
%         year=year+1;
%         no_day=1;
%     end
%     stream(i,1)=year;
%     stream(i,2)=streamflow(i);
% end
stream(:,2)=streamflow(:)*(0.3048^3)/1e6;
start=1;
for y=1936:2005
    if (rem(y,4)==0)
        tot_day=366;
    else
```

```
        tot_day=365;
    end
    y
    end_day=start+tot_day-1;
    temp=stream(start:end_day,2);
    start=end_day+1;
    [peak,cp]=max(temp)
    peak_flow(y-1935)=peak;
    diff=100;
    i=cp;
    while (diff>0)
        diff=temp(i)-temp(i-1);
        if (temp(i)>0.5*max(temp))
            diff=100;
        end
        i=i-1;
    end
    i
    flood_start_day(y-1935)=i;
    diff=100;
    i=cp;
    while (diff>0)
        diff=temp(i)-temp(i+1);
        if (temp(i)>0.5*max(temp))
            diff=100;
        end
        i=i+1;
    end
    i
    flood_end_day(y-1935)=i;
    duration(y-1935)=flood_end_day(y-1935)-flood_start_day(y-1935)+1;
    peak_day(y-1935)=cp;
    n=y-1935;
    qis(n)=temp(flood_start_day(n));
    qie(n)=temp(flood_end_day(n));
    floodflow=temp(flood_start_day(n):flood_end_day(n));
    vol(n)=sum(floodflow)-0.5*duration(n)*(qis(n)+qie(n));
    clear floodflow
end
year=1936:2005;
```

```
corrcoef(peak_flow,vol)
corrcoef(peak_flow,duration)
corrcoef(vol,duration)
[sup1,sup2,ran,pdf]=nonparapdf(vol);
[val_ortho,cdf_ortho]=cdfestimate(ran,pdf);
hold on;
%plot(val_ortho,cdf_ortho,'m*');
%kernel
xi=ran;
[f]=ksdensity(vol,xi);
f=f';
[val_kernel,cdf_kernel]=cdfestimate(xi,f);
hold on;
%plot(val_kernel,cdf_kernel,'y*');
subplot(3,1,2),hist1(vol);
hold on;
subplot(3,1,2),plot(ran,pdf,'k');
subplot(3,1,2),plot(ran,f,'b');
%Weibull
vol_sort1=sort(vol);
val_weibull=val_ortho;
for k=1:length(val_weibull)
  j=0;
  for i=1:length(vol)
    if (vol(i)<=val_weibull(k))
      j=j+1;
    end
  end
  cdf_weibull(k)=j/(length(vol)+1);
end
%plot(val_weibull,cdf_weibull);
%%subplot(3,1,2),plot(xi,f);
%%subplot(3,1,2),plot(ran,pdf,'m');
x=ran;
y=normpdf(x,mean(vol),std(vol));
lambda=1/(mean(vol));
x_expo=ran;
for k=1:length(x_expo)
  y_expo(k)=lambda*exp((-lambda)*x_expo(k));
end
```

```
subplot(3,1,2),plot(x_expo,y_expo,'r');
[val_expo,cdf_expo]=cdfestimate(x_expo,y_expo);
hold on;
%plot(val_expo,cdf_expo,'k*');
for i=1:length(vol)
   ln_data(i)=log(vol(i));
end
for i=1:length(ran)
   log_ran(i)=log(ran(i));
end
meany=mean(ln_data);
stdy=std(ln_data);
for i=1:length(ran)
   pdf_log_normal(i)=normpdf(log_ran(i),meany,stdy)/ran(i);
end
x_ln_normal=ran;
y_ln_normal=pdf_log_normal;
[val_ln_normal,cdf_ln_normal]=cdfestimate(x_ln_normal,y_ln_normal);
subplot(3,1,2),plot(x_ln_normal,y_ln_normal,'g');
%%subplot(3,1,2),plot(x,y,'r');
%%subplot(3,1,2),plot(x_expo,y_expo,'c');
meanx=mean(vol);
for i=1:length(vol)
   ln_data(i)=log(vol(i));
end
A=log(meanx)-mean(ln_data);
alpha=(1/(4*A))*(1+sqrt(1+((4*A)/3)));
beta=meanx/alpha;
x_gamma=ran;
for k=1:length(x_gamma)
   y_gamma(k)=(1/((beta^alpha)*gamma(alpha)))*(x_gamma(k)^(alpha-1))*exp(-x_gamma(k)/beta);
end
[val_gamma,cdf_gamma]=cdfestimate(x_gamma,y_gamma);
hold on;
subplot(3,1,2),plot(x_gamma,y_gamma,'m');
%plot(val_gamma,cdf_gamma,'r*');
%%subplot(3,1,2),plot(x_gamma,y_gamma,'g');
alpha=(sqrt(6)/pi)*std(vol);
u=meanx-0.577*alpha;
x_gumbel=ran;
```

```
for k=1:length(x_gumbel)
    y_gumbel(k)=(1/alpha)*exp((-(x_gumbel(k)-u)/alpha)-exp(-(x_gumbel(k)-u)/alpha));
end
[val_gumbel,cdf_gumbel]=cdfestimate(x_gumbel,y_gumbel);
hold on;
subplot(3,1,2),plot(x_gumbel,y_gumbel,'y');
%plot(val_gumbel,cdf_gumbel,'g*');
%%subplot(3,1,2),plot(x_gumbel,y_gumbel,'k');
c_y=skewness(ln_data);
p=(2/c_y)^2;
beta=std(ln_data)/sqrt(p);
alpha=mean(ln_data)-p*beta;
x_lpt3=ran;
for k=1:length(x_lpt3)
    ln_trans(k)=log(x_lpt3(k));
    y_lpt3(k)=(1/(beta*gamma(p)))*(((ln_trans(k)-alpha)/beta)^(p-1))*exp(-(ln_trans(k)-alpha)/beta)/x_lpt3(k);
end
[val_lpt3,cdf_lpt3]=cdfestimate(x_lpt3,y_lpt3);
hold on;
%plot(val_lpt3,cdf_lpt3,'c*');
% q_weibull=val_weibull(:);
% q_weibull=q_weibull';
% q_kernel=valestimate(cdf_kernel,val_kernel,cdf_weibull);
% q_ortho=valestimate(cdf_ortho,val_ortho,cdf_weibull);
% q_expo=valestimate(cdf_expo,val_expo,cdf_weibull);
% q_gamma=valestimate(cdf_gamma,val_gamma,cdf_weibull);
% q_gumbel=valestimate(cdf_gumbel,val_gumbel,cdf_weibull);
% %q_lpt3=valestimate(cdf_lpt3,val_lpt3,cdf_weibull);
mse_kernel=(((cdf_kernel-cdf_weibull)*(cdf_kernel-cdf_weibull)')/(length(cdf_kernel)))
mse_ortho=(((cdf_ortho-cdf_weibull)*(cdf_ortho-cdf_weibull)')/(length(cdf_ortho)))
mse_expo=(((cdf_expo-cdf_weibull)*(cdf_expo-cdf_weibull)')/(length(cdf_expo)-1))
mse_gamma=(((cdf_gamma-cdf_weibull)*(cdf_gamma-cdf_weibull)')/(length(cdf_gamma)-2))
mse_gumbel=(((cdf_gumbel-cdf_weibull)*(cdf_gumbel-cdf_weibull)')/(length(cdf_gumbel)-2))
mse_ln_normal=(((cdf_ln_normal-cdf_weibull)*(cdf_ln_normal-cdf_weibull)')/(length(cdf_ln_normal)-2))
%mse_lpt3=(((cdf_lpt3-cdf_weibull)*(cdf_lpt3-cdf_weibull)')/(length(cdf_lpt3)-3))
aic_kernel=length(cdf_kernel)*log(mse_kernel)+2*0
aic_ortho=length(cdf_ortho)*log(mse_ortho)+2*0
aic_expo=length(cdf_expo)*log(mse_expo)+2*1
aic_gamma=length(cdf_gamma)*log(mse_gamma)+2*2
aic_gumbel=length(cdf_gumbel)*log(mse_gumbel)+2*2
```

```
aic_ln_normal=length(cdf_ln_normal)*log(mse_ln_normal)+2*2
%aic_lpt3=length(cdf_lpt3)*log(mse_lpt3)+2*3
para=2;
[chi_ortho]=chi_compute(cdf_weibull,cdf_ortho,para);
chi_ortho
```

Iu. Fitting PDF for duration

```
clear all;
data=xlsread('ftperday70_grandfork');
streamflow=data(:,1);
year=1936;
no_day=0;
% for i=1:length(streamflow)
%    if (rem(year,4)==0)
%       tot_day=366;
%    else
%       tot_day=365;
%    end
%    no_day=no_day+1;
%    if (no_day>tot_day)
%       year=year+1;
%       no_day=1;
%    end
%    stream(i,1)=year;
%    stream(i,2)=streamflow(i);
% end
stream(:,2)=streamflow(:)*(0.3048^3)/1e6;
start=1;
for y=1936:2005
   if (rem(y,4)==0)
      tot_day=366;
   else
      tot_day=365;
   end
   y
   end_day=start+tot_day-1;
   temp=stream(start:end_day,2);
   start=end_day+1;
```

```
    [peak,cp]=max(temp)
    peak_flow(y-1935)=peak;
    diff=100;
    i=cp;
    while (diff>0)
        diff=temp(i)-temp(i-1);
        if (temp(i)>0.5*max(temp))
            diff=100;
        end
        i=i-1;
    end
    i
    flood_start_day(y-1935)=i;
    diff=100;
    i=cp;
    while (diff>0)
        diff=temp(i)-temp(i+1);
        if (temp(i)>0.5*max(temp))
            diff=100;
        end
        i=i+1;
    end
    i
    flood_end_day(y-1935)=i;
    duration(y-1935)=flood_end_day(y-1935)-flood_start_day(y-1935)+1;
    peak_day(y-1935)=cp;
    n=y-1935;
    qis(n)=temp(flood_start_day(n));
    qie(n)=temp(flood_end_day(n));
    floodflow=temp(flood_start_day(n):flood_end_day(n));
    vol(n)=sum(floodflow)-0.5*duration(n)*(qis(n)+qie(n));
    clear floodflow
end
year=1936:2005;
corrcoef(peak_flow,vol)
corrcoef(peak_flow,duration)
corrcoef(vol,duration)
[sup1,sup2,ran,pdf]=nonparapdf(duration);
[val_ortho,cdf_ortho]=cdfestimate(ran,pdf);
hold on;
```

```
%plot(val_ortho,cdf_ortho,'m*');
%kernel
xi=ran;
[f]=ksdensity(duration,xi);
f=f';
[val_kernel,cdf_kernel]=cdfestimate(xi,f);
hold on;
%plot(val_kernel,cdf_kernel,'y*');
subplot(3,1,3),hist1(duration);
hold on;
subplot(3,1,3),plot(ran,pdf,'k');
subplot(3,1,3),plot(ran,f,'b');
%Weibull
duration_sort1=sort(duration);
val_weibull=val_ortho;
for k=1:length(val_weibull)
    j=0;
    for i=1:length(duration)
        if (duration(i)<=val_weibull(k))
            j=j+1;
        end
    end
    cdf_weibull(k)=j/(length(duration)+1);
end
%plot(val_weibull,cdf_weibull);
%%subplot(3,1,3),plot(xi,f);
%%subplot(3,1,3),plot(ran,pdf,'m');
x=ran;
y=normpdf(x,mean(duration),std(duration));
lambda=1/(mean(duration));
x_expo=ran;
for k=1:length(x_expo)
    y_expo(k)=lambda*exp((-lambda)*x_expo(k));
end
subplot(3,1,3),plot(x_expo,y_expo,'r');
[val_expo,cdf_expo]=cdfestimate(x_expo,y_expo);
hold on;
%plot(val_expo,cdf_expo,'k*');
for i=1:length(duration)
    ln_data(i)=log(duration(i));
```

```
end
for i=1:length(ran)
    log_ran(i)=log(ran(i));
end
meany=mean(ln_data);
stdy=std(ln_data);
for i=1:length(ran)
    pdf_log_normal(i)=normpdf(log_ran(i),meany,stdy)/ran(i);
end
x_ln_normal=ran;
y_ln_normal=pdf_log_normal;
[val_ln_normal,cdf_ln_normal]=cdfestimate(x_ln_normal,y_ln_normal);
subplot(3,1,3),plot(x_ln_normal,y_ln_normal,'g');
%%subplot(3,1,3),plot(x,y,'r');
%%subplot(3,1,3),plot(x_expo,y_expo,'c');
meanx=mean(duration);
for i=1:length(duration)
    ln_data(i)=log(duration(i));
end
A=log(meanx)-mean(ln_data);
alpha=(1/(4*A))*(1+sqrt(1+((4*A)/3)));
beta=meanx/alpha;
x_gamma=ran;
for k=1:length(x_gamma)
    y_gamma(k)=(1/((beta^alpha)*gamma(alpha)))*(x_gamma(k)^(alpha-1))*exp(-x_gamma(k)/beta);
end
[val_gamma,cdf_gamma]=cdfestimate(x_gamma,y_gamma);
hold on;
subplot(3,1,3),plot(x_gamma,y_gamma,'m');
%plot(val_gamma,cdf_gamma,'r*');
%%subplot(3,1,3),plot(x_gamma,y_gamma,'g');
alpha=(sqrt(6)/pi)*std(duration);
u=meanx-0.577*alpha;
x_gumbel=ran;
for k=1:length(x_gumbel)
    y_gumbel(k)=(1/alpha)*exp((-(x_gumbel(k)-u)/alpha)-exp(-(x_gumbel(k)-u)/alpha));
end
[val_gumbel,cdf_gumbel]=cdfestimate(x_gumbel,y_gumbel);
hold on;
subplot(3,1,3),plot(x_gumbel,y_gumbel,'y');
```

```
%plot(val_gumbel,cdf_gumbel,'g*');
%%subplot(3,1,3),plot(x_gumbel,y_gumbel,'k');
c_y=skewness(ln_data);
p=(2/c_y)^2;
beta=std(ln_data)/sqrt(p);
alpha=mean(ln_data)-p*beta;
x_lpt3=ran;
for k=1:length(x_lpt3)
  ln_trans(k)=log(x_lpt3(k));
  y_lpt3(k)=(1/(beta*gamma(p)))*(((ln_trans(k)-alpha)/beta)^(p-1))*exp(-(ln_trans(k)-alpha)/beta)/x_lpt3(k);
end
[val_lpt3,cdf_lpt3]=cdfestimate(x_lpt3,y_lpt3);
hold on;
%plot(val_lpt3,cdf_lpt3,'c*');
% q_weibull=val_weibull(:);
% q_weibull=q_weibull';
% q_kernel=valestimate(cdf_kernel,val_kernel,cdf_weibull);
% q_ortho=valestimate(cdf_ortho,val_ortho,cdf_weibull);
% q_expo=valestimate(cdf_expo,val_expo,cdf_weibull);
% q_gamma=valestimate(cdf_gamma,val_gamma,cdf_weibull);
% q_gumbel=valestimate(cdf_gumbel,val_gumbel,cdf_weibull);
% %q_lpt3=valestimate(cdf_lpt3,val_lpt3,cdf_weibull);
mse_kernel=(((cdf_kernel-cdf_weibull)*(cdf_kernel-cdf_weibull)')/(length(cdf_kernel)))
mse_ortho=(((cdf_ortho-cdf_weibull)*(cdf_ortho-cdf_weibull)')/(length(cdf_ortho)))
mse_expo=(((cdf_expo-cdf_weibull)*(cdf_expo-cdf_weibull)')/(length(cdf_expo)-1))
mse_gamma=(((cdf_gamma-cdf_weibull)*(cdf_gamma-cdf_weibull)')/(length(cdf_gamma)-2))
mse_gumbel=(((cdf_gumbel-cdf_weibull)*(cdf_gumbel-cdf_weibull)')/(length(cdf_gumbel)-2))
mse_ln_normal=(((cdf_ln_normal-cdf_weibull)*(cdf_ln_normal-cdf_weibull)')/(length(cdf_ln_normal)-2))
%mse_lpt3=(((cdf_lpt3-cdf_weibull)*(cdf_lpt3-cdf_weibull)')/(length(cdf_lpt3)-3))
aic_kernel=length(cdf_kernel)*log(mse_kernel)+2*0
aic_ortho=length(cdf_ortho)*log(mse_ortho)+2*0
aic_expo=length(cdf_expo)*log(mse_expo)+2*1
aic_gamma=length(cdf_gamma)*log(mse_gamma)+2*2
aic_gumbel=length(cdf_gumbel)*log(mse_gumbel)+2*2
aic_ln_normal=length(cdf_ln_normal)*log(mse_ln_normal)+2*2
%aic_lpt3=length(cdf_lpt3)*log(mse_lpt3)+2*3
para=2;
[chi_ortho]=chi_compute(cdf_weibull,cdf_ortho,para);
chi_ortho
```

APPENDIX II. PREVIOUS REPORTS IN THE SERIES

ISSN: (print) 1913-3200; (online) 1913-3219

[1] Slobodan P. Simonovic (2001). Assessment of the impact of climate variability and change on the reliability, resiliency and vulnerability of complex flood protection systems, Water Resources Research Report no. 038, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 91 pages. ISBN: (print) 978-0-7714-2606-3; (online) 978-0-7714-2607-0.

[2] Predrag Prodanovic (2001). Fuzzy set ranking methods and multiple expert decision making, Water Resources Research Report no. 039, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 68 pages. ISBN: (print) 978-0-7714-2608-7; (online) 978-0-7714-2609-4.

[3] Nirupama and Slobodan P. Simonovic (2002). Role of remote sensing in disaster management, Water Resources Research Report no. 040, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 107 pages. ISBN: (print) 978-0-7714-2610-0; (online) 978-0-7714-2611-7.

[4] Taslima Akter and Slobodan P. Simonovic (2002). A general overview of multi-objective multiple-participant decision making for flood management, Water Resources Research Report no. 041, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 65 pages. ISBN: (print) 978-0-7714-2612-4; (online) 978-0-7714-2613-1.

[5] Nirupama and Slobodan P. Simonovic (2002). A spatial fuzzy compromise approach for flood disaster management, Water Resources Research Report no. 042, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 138 pages. ISBN: (print) 978-0-7714-2614-8; (online) 978-0-7714-2615-5.

[6] K. D. W. Nandalal and Slobodan P. Simonovic (2002). State-of-the-Art Report on Systems Analysis Methods for Resolution of Conflicts in Water Resources Management, Water Resources Research Report no. 043, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 216 pages. ISBN: (print) 978-0-7714-2616-2; (online) 978-0-7714-2617-9.

[7] K. D. W. Nandalal and Slobodan P. Simonovic (2003). Conflict Resolution Support System - A Software for the Resolution of Conflicts in Water Resource Management, Water Resources Research Report no. 044, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 144 pages. ISBN: (print) 978-0-7714-2618-6; (online) 978-0-7714-2619-3.

[8] Ibrahim El-Baroudy and Slobodan P. Simonovic (2003). New Fuzzy Performance Indices for Reliability Analysis of Water Supply Systems, Water Resources Research Report no. 045, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 90 pages. ISBN: (print) 978-0-7714-2620-9; (online) 978-0-7714-2621-6.

[9] Juraj Cunderlik (2003). Hydrologic Model Selection for the CFCAS Project: Assessment of Water Resources Risk and Vulnerability to Changing Climatic

Conditions. Water Resources Research Report no. 046, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 40 pages. ISBN: (print) 978-0-7714-2622-3; (online) 978-0-7714-2623-0.

[10] Juraj Cunderlik and Slobodan P. Simonovic (2004). Selection of Calibration and Verification Data for the HEC-HMS Hydrologic Model. Water Resources Research Report no. 047, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 29 pages. ISBN: (print) 978-0-7714-2624-7; (online) 978-0-7714-2625-4.

[11] Juraj Cunderlik and Slobodan P. Simonovic (2004). Calibration, verification and sensitivity analysis of the HEC-HMS hydrologic model. Water Resources Research Report no. 048, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 113 pages. ISBN: (print) 978-0-7714-2626-1; (online) 978-0-7714-2627-8.

[12] Predrag Prodanovic and Slobodan P. Simonovic (2004). Generation of synthetic design storms for the Upper Thames River basin. Water Resources Research Report no. 049, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 20 pages. ISBN: (print) 978-0-7714-2628-5; (online) 978-0-7714-2629-2.

[13] Ibrahim El-Baroudy and Slobodan P. Simonovic (2005). Application of the Fuzzy Performance Indices to the City of London Water Supply system, Water Resources Research Report no. 050, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 137 pages. ISBN: (print) 978-0-7714-2630-8; (online) 978-0-7714-2631-5.

[14] Ibrahim El-Baroudy and Slobodan P. Simonovic (2006). A Decision Support System for Integrated Risk Management, Water Resources Research Report no. 051, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 146 pages. ISBN: (print) 978-0-7714-2632-2; (online) 978-0-7714-2633-9.

[15] Predrag Prodanovic and Slobodan P. Simonovic (2006). Inverse Flood Risk Modelling of The Upper Thames River Basin. Water Resources Research Report no. 052, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 163 pages. ISBN: (print) 978-0-7714-2634-6; (online) 978-0-7714-2635-3.

[16] Predrag Prodanovic and Slobodan P. Simonovic (2006). Inverse Drought Risk Modelling of The Upper Thames River Basin. Water Resources Research Report no. 053, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 252 pages. ISBN: (print) 978-0-7714-2636-0; (online) 978-0-7714-2637-7.

[17] Predrag Prodanovic and Slobodan P. Simonovic (2007). Dynamic Feedback Coupling of Continuous Hydrologic and Socio-Economic Model Components of the Upper Thames River Basin. Water Resources Research Report no. 054, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 437 pages. ISBN: (print) 978-0-7714-2638-4; (online) 978-0-7714-2639-1.