NIST GCR 06-905

A Model for Directional Hurricane Wind Speeds

Mircea Grigoriu
Cornell University

NIST
National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

NIST GCR 06-905

A Model for Directional Hurricane Wind Speeds

Prepared for
U.S. Department of Commerce
Building and Fire Research Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8611

By
Mircea Grigoriu
Cornell University
Ithaca, NY 14853

December 2006

U.S. Department of Commerce
Carlos M. Gutierrez, Secretary

Technology Administration
Robert Cresanti, Under Secretary of Commerce for Technology

National Institute of Standards and Technology
William Jeffrey, Director

Contents

A model for directional hurricane wind speeds

Mircea Grigoriu
Cornell University, Ithaca, NY 14853

1 Introduction

Let $\boldsymbol{X}$ be an $\mathbb{R}^d$-valued random variable whose coordinates $\{X_i\}$, $i = 1, \ldots, d$, denote hurricane wind speeds in d-directions at a site. Independent samples of $\boldsymbol{X}$ can be viewed as synthetic hurricane wind speeds occurring in different storms. The random vector $\boldsymbol{X}$ cannot be Gaussian since the sequence of wind speeds recorded in an arbitrary direction $i = 1, \ldots, d$ during different storm has 0's so that the marginal distribution of X_i has a finite mass at 0.

Our objectives are to develop (1) a probabilistic model for $\boldsymbol{X}$ describing hurricane wind speeds in 16 directions at angles $\theta_i = 22.5^o\, i$, $i = 1, \ldots, 16$, (2) a method for calibrating the model for $\boldsymbol{X}$ to records available at a site, and (3) a Monte Carlo algorithm for generating synthetic hurricane speeds over an arbitrary number of years a selected site.

2 Probability law of hurricane wind speed

Consider the special case in which the coordinates of $\boldsymbol{X}$ are Bernoulli random variables, that is,

$$X_i = \begin{cases} 0, & \text{probability } 1 - p_i \\ 1, & \text{probability } p_i, \end{cases} \tag{1}$$

where $p_i \in (0, 1)$ for $i = 1, \ldots, d$. The values 0 and 1 of a coordinate X_i of $\boldsymbol{X}$ correspond to 0 and non-zero hurricane wind speeds in direction $i = 1, \ldots, d$. The average number of 0's and 1's of X_i in n independent trials are $n\,(1 - p_i)$ and $n\,p_i$, respectively. We use the model in Eq. 1 to illustrated difficulties related to the complete probabilistic characterization of the hurricane wind vector $\boldsymbol{X}$.

If the coordinates of $\boldsymbol{X}$ are independent, Eq. 1 defines the probability law of $\boldsymbol{X}$. If the coordinates of $\boldsymbol{X}$ are dependent, additional information is needed to specify $\boldsymbol{X}$. Let $p_{k_1,\ldots,k_d} = P\big(\cap_{i=1}^d \{X_i = k_i\}\big)$ with $k_1, \ldots, k_d \in \{0, 1\}$ denote the probability that $\big(X_1, \ldots, X_d\big)$ is equal to a particular string $\big(k_1, \ldots, k_d\big)$ of 0's and 1's. We note that (1) the probabilities $\{p_{k_1,\ldots,k_d}\}$, $k_1, \ldots, k_d \in \{0, 1\}$, define uniquely the probability law of $\boldsymbol{X}$ and (2) $p_{k_1,\ldots,k_d} = \prod_{i=1}^d P\big(X_i = k_i\big)$ if $\boldsymbol{X}$ has independent coordinates.

The complete characterization of $\boldsymbol{X}$ involves two types of difficulties. First, the number of probabilities $\{p_{k_1,\ldots,k_d}\}$ defining the probability law of $\boldsymbol{X}$ increases rapidly with d. For example, suppose that $d = 3$. The probability law of $\boldsymbol{X}$ is completely defined by $2^d = 8$ probabilities $p_{k_1,k_2,k_3} = P\big(X_1 = k_1, X_2 = k_2, X_3 = k_3\big)$, $k_1, k_2, k_3 \in \{0, 1\}$, that the vector

$\big(X_1, X_2, X_3\big)$ is equal to $(0,0,0)$, $(1,0,0)$, $(0,1,0)$, $(0,0,1)$, $(1,1,0)$, $(1,0,1)$, $(0,0,1)$, and $(1,1,1)$. The number of probabilities $\{p_{k_1,\ldots,k_d}\}$ is 8; 32; 1,024; and 65,536 for $d = 3$; 5; 10; and 16, respectively. Numerical calculations involving 65,536 probabilities are not feasible. Second, the probabilities $\{p_{k_1,\ldots,k_d}\}$ need to be estimated from data. Estimates of these probabilities are likely to be unreliable or even impossible for vectors $\boldsymbol{X}$ with dimension $d = 8$ or larger if based on records of typical length. These considerations demonstrate the need for developing simplified models for $\boldsymbol{X}$ that are numerically tractable and their parameters can be estimated reliably from data.

3 Translation model for hurricane wind speeds

We propose a translation non-Gaussian model $\boldsymbol{X}_T$ for the wind speed vector $\boldsymbol{X}$, present a method for estimating the probability law of $\boldsymbol{X}_T$, and develop a Monte Carlo algorithm for generating samples of $\boldsymbol{X}_T$.

3.1 Model definition

Let p_i and F_i denote the probability that the coordinate X_i, $i = 1,\ldots,d$, of $\boldsymbol{X}$ is not 0 and the distribution of the non-zero values of this coordinate, so that

$$\tilde{F}_i(x) = (1 - p_i)\, 1\big(x \geq 0\big) + p_i\, F_i(x), \quad i = 1,\ldots,d, \tag{2}$$

is the distribution of X_i, where $1(A) = 1$ and 0 if statement A is valid and invalid, respectively. We can view X_i as a generalized Bernoulli variable that is 0 with probability $1 - p_i$ and is a random variable following the distribution F_i with probability p_i

Consider an $\mathbb{R}^d$-valued random variable $\boldsymbol{X}_T$ with coordinates $X_{T,i}$ defined by

$$X_{T,i} = \tilde{F}_i^{-1}\big(G_i\big), \quad i = 1,\ldots,d, \tag{3}$$

where $\boldsymbol{G} = (G_1,\ldots,G_d)$ is a standard $\mathbb{R}^d$-valued Gaussian variable, that is, $\text{Mean}[G_i] = 0$, $\text{Var}[G_i] = 1$, and $\text{Covariance}[G_i, G_j] = \rho_{ij}$, $i = 1,\ldots,d$. We refer to $\boldsymbol{X}_T$ as the translation model for $\boldsymbol{X}$. The model $\boldsymbol{X}_T$ has the same marginal distributions as $\boldsymbol{X}$ irrespective of the covariance matrix $\boldsymbol{\rho} = \{\rho_{ij}\}$ of $\boldsymbol{G}$ since $X_{T,i}$ is 0 with probability $P\big(\Phi(G_i) \leq 1 - p_i\big) = P\big(G_i \leq \Phi^{-1}(1-p_i)\big) = 1-p_i$ and has distribution F_i with the complement of this probability, that is, $P\big(X_i \neq 0\big) = p_i$ for all $i = 1,\ldots,d$. The dependence between the coordinates of $\boldsymbol{X}_{T,i}$ is defined by the covariance matrix $\boldsymbol{\rho}$ of $\boldsymbol{G}$ and the marginal distributions $\{F_i\}$ of $\boldsymbol{X}$. The relationship between the correlation structures of $\boldsymbol{G}$ and $\boldsymbol{X}_T$ is discussed in [1] (Section 3.1.1).

The translation model in Eq. 3 has two notable features. The model (1) has, as already stated, the same marginal distributions as $\boldsymbol{X}$ and (2) is sufficiently simple to be used in applications. A limitation of the model is that the complex dependence between the coordinates of $\boldsymbol{X}$ is represented approximately.

3.1.1 Parameter estimation

Let $\big(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\big)$ be n independent samples of $\boldsymbol{X}$, and let $\big(x_{i,1},\ldots,x_{i,n}\big)$ denote the corresponding n samples of coordinate X_i, $i = 1,\ldots,d$. Denote by $\big(y_{i,1},\ldots,y_{i,m_i}\big)$, $m_i \leq n$, the sequence of non-zero readings extracted from $\left(x_{i,1},\ldots,x_{i,n}\right)$. For example, $x_{i,1}$ is not included in $\left(y_{i,1},\ldots,y_{i,m_i}\right)$ if 0 and $y_{i,1}=x_{i,1}$ if $x_{i,1}\neq 0$.

The probabilities p_i and the marginal distributions F_i can be estimated by

$$p_i \simeq \hat{p}_i = \frac{m_i}{n}, \quad i=1,\ldots,d, \tag{4}$$

and

$$F_i(x) \simeq \hat{F}_i(x) = \frac{\sum_{j=1}^{m_i} 1\left(y_{i,j}\le x\right)}{m_i}, \quad i=1,\ldots,d. \tag{5}$$

Similarly, the mean μ_i and variance σ_i^2 of F_i can be estimated from

$$\mu_i \simeq \hat{\mu}_i = \frac{1}{m_i}\sum_{j=1}^{m_i} y_{i,j}$$
$$\sigma_i^2 \simeq \hat{\sigma^2}_i = \frac{1}{m_i}\sum_{j=1}^{m_i}\left(y_{i,j}-\hat{\mu}_i\right)^2. \tag{6}$$

The estimation of the correlation matrix $\boldsymbol{r}=\{r_{ij}\}$, $i,j=1,\ldots,d$, corresponding to non-zero values of $\boldsymbol{X}$ poses some difficulties since different coordinates of $\boldsymbol{X}$ may be non-zero in different storms. Two options have been considered. First, select from the available record $\left(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\right)$ only those storms in which all coordinates are non-zero. This option is not viable since data shows that the resulting sample can be so short that reliable estimates of $\boldsymbol{r}$ are not possible. Second, select from the available record $\left(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\right)$ all storms in which the entries of a particular pair (i,j) of coordinates are not zero and estimate r_{ij} from this record. The advantage of this approach is that allows more reliable estimates of $\boldsymbol{r}$. A potential problem is that the resulting estimate $\hat{\boldsymbol{r}}$ of $\boldsymbol{r}$ may not be positive definite. We present in the following section a procedure for handling this situation. Let $\hat{\boldsymbol{\zeta}}$ be the estimate of the matrix of correlation coefficients of the non-zero values of $\{X_i\}$ obtained from $\hat{\boldsymbol{r}}$ and Eq. 6. Since the differences between the correlation matrices $\boldsymbol{\rho}$ of the Gaussian image $\boldsymbol{G}$ of $\boldsymbol{X}_T$ and $\boldsymbol{\zeta}$ are not significant for positively correlated random variables ([1], Section 3.1.1), we approximate $\boldsymbol{\rho}$ by $\hat{\boldsymbol{\zeta}}$.

3.2 Monte Carlo algorithm

Suppose we need to generate n independent samples of $\boldsymbol{X}$. The proposed algorithm uses samples of $\boldsymbol{X}_T$ as a substitute for samples of $\boldsymbol{X}$, and involves the following two steps.

Step 1. Generate n independent samples $\left(\boldsymbol{g}_1,\ldots,\boldsymbol{g}_n\right)$ of $\boldsymbol{G}$ with mean $\boldsymbol{0}$ and covariance matrix $\hat{\boldsymbol{\zeta}}$.

Step 2. Calculate samples $\left(\boldsymbol{x}_{T,1},\ldots,\boldsymbol{x}_{T,n}\right)$ of $\boldsymbol{X}_T$ from $\left(\boldsymbol{g}_1,\ldots,\boldsymbol{g}_n\right)$ and Eq. 3, and plot the resulting samples. It is assumed that all F_i are reverse Weibull distributions.

As previously stated, the generation of samples of $\boldsymbol{G}$ may pose some difficulties since the estimate $\hat{\boldsymbol{r}}$ of the correlation matrix $\boldsymbol{r}$, and consequently the estimate $\hat{\boldsymbol{\zeta}}$ of $\boldsymbol{\zeta}$, may not be positive definite. The generation algorithm is based on the approximate representation

$$\boldsymbol{G} \simeq \tilde{\boldsymbol{G}} = \sum_{k=1}^{16} \nu_k^* \, V_k \, \boldsymbol{\phi}_k \tag{7}$$

of $\boldsymbol{G}$, where $\{V_k\}$ are independent Gaussian variables with mean 0 and variance 1, $\{\nu_k, \boldsymbol{\phi}_k\}$ denote the eigenvalues and the eigenvectors of $\hat{\boldsymbol{\zeta}}$, and $\nu_k^* = \nu_k$ if $\nu_k > 0$ and $\nu_k^* = 0$ otherwise. We use the approximation in Eq. 7 to generate samples of $\boldsymbol{G}$.

4 MATLAB functions

Two MATLAB functions have been developed,

hurricane_dir_est.m and
hurricane_dir_mc.m.

The first function estimates the parameters of the probability law of $\boldsymbol{X}_T$. The second function generate samples of $\boldsymbol{X}_T$. The dimension of $\boldsymbol{X}$ is $d = 16$.

4.1 MATLAB function hurricane_dir_est.m

The input consists of:

(1) A record at a specified milepost (see lines 23 to 27),

(2) A range $[\mathrm{cmin}, \mathrm{cmax}]$ of Weibull tail parameter c and the number nc of intervals in $[\mathrm{cmin}, \mathrm{cmax}]$. We note that cmax needs to be selected to avoid unrealistic tail parameters. It is suggested to set $\mathrm{cmax} = 10$, and

(3) A minimum number ncorr of non-zero pairs of non-zero readings needed to estimate entries of $\boldsymbol{\zeta}$. If ncorr is not reached for a pair (i, j), we set $\hat{\zeta}_{ij} = 0$. It is suggested to set $\mathrm{ncorr} = 10$.

The output consists of:

(1) Estimates of the probabilities $\mathrm{p(i)} = P\big(X_i = 0\big)$, $i = 1, \ldots, d$,

(2) Estimates of reverse Weibull parameters alpha1(i), c(i), and xi(i), $i = 1, \ldots, d$,

(3) Estimates $\mathrm{zeta1(i,j)}$ of the correlation coefficients ζ_{ij}, $i, j = 1, ..., d$, and

(4) Plots with estimates of the probabilities p_i; mean, standard deviation, skewness of non-zero values of X_i; estimates of the correlation coefficients of all data and of non-zero data; estimates of the parameters of the reverse Weibull distributions; and histograms of non-zero readings in all directions including Weibull densities fitted to these data.

The above output needs to be saved in a file for use in **hurricane_dir_mc.m**. The command **save estimates350 p zeta1 alpha1 c xi** may be used to store parameters needed for simulation. It is suggested that the file name be related to milepost number, for example, **estimates350** if dealing with milepost350.

4.2 MATLAB function hurricane_dir_emc.m

The input consists of:

(1) A file with estimates of the parameters needed to define the probability law of $\boldsymbol{X}_T$, for example, the file **estimates350** and

(2) The sample size ns and a seed nseed for sample generation.

The output consists of:

(1) Three dimensional plots of the generated samples of $\boldsymbol{G}$ and

(2) Three dimensional plots and contour lines of the generated samples of $\boldsymbol{X}_T$.

5 Conclusions

A non-Gaussian model has been developed for hurricane wind speeds recorded in 16 equally spaced directions based on the theory of translation variables. A method has been presented for calibrating the wind model to site records. The calibrated model has been used to generate synthetic hurricane wind speeds of arbitrary length at a selected site.

References

[1] M. Grigoriu. *Applied Non-Gaussian Processes: Examples, Theory, Simulation, Linear Random Vibration, and MATLAB Solutions.* Prentice Hall, Englewoods Cliffs, NJ, 1995.

Appendix A. MATLAB function hurricane_dir_est.m

```
function [p,mu,sig,gam3,zeta_t,zeta1,alpha1,c,xi] = ...
    hurricane_dir_est(cmin,cmax,nc,ncorr)
%
%   It estimates:
%       (1) The probability p(i)=P(X_i=0) that coordinate
%           i=1,...,16 of wind speed is 0
%       (2) The mean mu(i), standard deviation sig(i), and
%           skewness coefficient gam3(i) of the non-zero
%           values for each i=1,...,16
%       (3) The correlation coefficients {zeta_t(i,j)},
%           i,j=1,...,16, of the complete record,
%           i.e., including zero readings, and
%           {zeta1(i,j)}, i,j=1,...,16, of
%           non-zero readings
%--------------------------------------------------------------
%   INPUT:  (1) A record at a specified milepost
%               (see lines 23 to 27)
%           (2) Range [cmin,cmax] of Weibull tail
%               parameter c and nc = # of intervals
%               in [cmin,cmax]
%               NOTE:  cmax is also used to limit the value
%                      of the tail parameter, eg, cmax=10
%           (3) ncorr = the minimum number of non-zero
%               readings for which correlation is calculated
%               If ncorr is not reached, the correlation
%               coefficient is set 0
%               (Suggestion: Set ncorr=10)
%---------------------------------------------------------------
%   OUTPUT: (1) Estimates of {p(i)}, i=1,...,16
%           (2) Estimates of reverse Weibull parameters
%               {alpha1(i), c(i), xi(i)}, i=1,...,16
%           (3) Estimates of the correlation coefficients
%               {zeta1(i,j)}, i,j=1,...,16, corresponding
%               non-zero wind speeds
%=================================================================
%   Load record = a (999,17)-matrix for a Milepost
%           NOTE: THE FOLLOWING INSTRUCTION HAS TO BE MODIFIED
%                 TO SELECT A DIFFERENT MILEPOST #
%-----------------------------------------------------------
load  milepost350;
q=matrix;
nr=length(q(:,1));
nu=mean_rate;   %  nu = the average number of hurricane/year
%                        also in hppt://www.nist.gov/wind
%===========================================================
%   Estimates of probabilities p(i)
%   NOTE:  All readings are >=0
%-----------------------------------------------------------
for i=1:16,
    p(i)=sum(q(:,i)>0)/nr;
end,
figure
plot(1:16,p)
```

```
xlabel('Wind direction')
ylabel('Estimates of probabilities of non-zero values')
%-----------------------------------------------------------
%   Construct non-zero wind speed records in each
%   direction, estimate {mu(i), sig(i), gam3(i)}, and
%   calculte coeffiicents of variation vq(i)=sig(i)/mu(i)
%-----------------------------------------------------------
for i=1:16,
    nnz=0;
    for kr=1:nr,
        if q(kr,i)>0,
            nnz=nnz+1;
            xnz(nnz)=q(kr,i);
        end,
    end,
    xnzz=xnz(1:nnz);
    mu(i)=mean(xnzz);
    sig(i)=std(xnzz);
    vq(i)=sig(i)/mu(i);
    gam3(i)=mean(((xnzz-mu(i))/sig(i)).^3);
end,
figure
plot(1:16,mu,1:16,sig,':')
xlabel('Wind direction')
ylabel('Estimates of mean/std (solid/dotted lines) for non-zero values')
figure
plot(1:16,gam3)
xlabel('Wind direction')
ylabel('Estimates of skewness for non-zero values')
%------------------------------------------------------------
%   Estimates of correlation coefficients
%   {zeta_t(i,j)},  i,j=1,...,16
%------------------------------------------------------------
qq=q(:,1:16);
zeta_t=corrcoef(qq);
figure
mesh(1:16,1:16,zeta_t)
xlabel('Wind direction #')
ylabel('Wind direction #')
zlabel('Estimates of correlation coefficients \zeta_t')
%----------------------------------------------------------
%   Estimates of correlation coefficients
%   {zeta(i,j)},  i,j=1,...,16
%----------------------------------------------------------
for i=1:16,
    for j=1:16,
        q1=q(:,i);
        q2=q(:,j);
        nqq=0;
        for kr=1:nr,
            if q1(kr)>0 & q2(kr)>0,
                nqq=nqq+1;
                xqq(nqq,:)=[q1(kr) q2(kr)];
            end,
        end,
        if nqq<=01,
            zeta(i,j)=0;
```

```
        else,
            rr=corrcoef(xqq(1:nqq,1),xqq(1:nqq,2));
            rrr=rr(1,2);
            zeta(i,j)=rrr;
        end,
    end,
end,
figure
mesh(1:16,1:16,zeta)
xlabel('Wind direction #')
ylabel('Wind direction #')
zlabel('Estimates of correlation coefficients \zeta')
figure
contour(1:16,1:16,zeta)
xlabel('Wind direction #')
ylabel('Wind direction #')
title('Estimates of correlation coefficients \zeta')
%==================================================================
%   Estimates of the paramters of reverse Weibull distributions
%   fitted to non-zero wind speeds (Method of moments)
%   USE [- RECORD] in all directions
%------------------------------------------------------------------
%           Relationship between Weibull tail parameter
%           and skewness
%-------------------------------------------------------------------
dc=(cmax-cmin)/nc;
cc=cmin:dc:cmax;
lc=length(cc);
g1=gamma(1./cc+1);
g2=gamma(2./cc+1);
g3=gamma(3./cc+1);
skew=(g3-3*g1.*g2+2*g1.^3)./(g2-g1.^2).^(3/2);
% figure
% plot(cc,skew)
% xlabel('coefficient c')
% ylabel('skewness')
%---------------------------------------------
%           Calculation of skewness coefficients
%           for values of c>0 in [cmin,cmax]
%           and estimated tail parameters
%           {c(i)}, i=1,...,16
%------------------------------------------
for i=1:16,
    muw(i)=-mu(i);
    sigw(i)=sig(i);
    gamw3(i)=-gam3(i);
    c(i)=interp1(skew,cc,gamw3(i),'spline');
    %------------------------------------------------
    %   NOTE:  This condition is needed since
    %          c can take very large values
    %-----------------------------------------------
    if c(i)>cmax,
        c(i)=cmax;
    end,
end,
%-----------------------------------------------------
%           NOTE: If desired one or more or all c(i)'s
```

```
%                  can be assigned different values
%------------------------------------------------------
for i=1:16,
    ggw1(i)=gamma(1./c(i)+1);
    ggw2(i)=gamma(2./c(i)+1);
    ggw3(i)=gamma(3./c(i)+1);
    alpha(i)=sigw(i)/sqrt(ggw2(i)-ggw1(i)^2);
    xi(i)=muw(i)-alpha(i)*ggw1(i);
end,
figure
plot(1:16,alpha,1:16,c,':',1:16,xi,'--')
xlabel('Wind direction #')
ylabel('Reverse Weibull parameters for non-zero readings')
title('Estimates of \alpha, c, and \xi (solid, dotted, and dashed lines)')
%--------------------------------------------------------------------------
%   Plots of histograms and fitted reverse Weibull distributions
%   to non-zero wind speeds in all directions
%--------------------------------------------------------------------------
for i=1:16,
    nnz=0;
    for kr=1:nr,
        if q(kr,i)>0,
            nnz=nnz+1;
            xnz(nnz)=q(kr,i);
        end,
    end,
    xnzz=xnz(1:nnz);
    figure
    hist_est(xnzz',1,30)
    hold
    yxi=xi(i):.1:50;
    yw=(yxi-xi(i))/alpha(i);
    fw=(c(i)/alpha(i))*(yw.^(c(i)-1)).*exp(-yw.^c(i));
    plot(-yxi,fw)
    xlabel('Wind speed (mph)')
    ylabel(['Direction ' int2str(i)])
    %   print
end,
zeta1=zeta;
alpha1=alpha;
%===========================================================================
%   EXAMPLE:
%   [p,mu,sig,gam3,zeta_t,zeta1,alpha1,c,xi]=hurricane_dir_est(.1,10,1000,10);
%   NOTE: Save the output needed for Monte Carlo simulation, e.g., use
%         save estimates350 p zeta alpha c xi
%         (estimates350 = file name, 350 since mileplot350 is used)
```

Appendix B. MATLAB function hurricane_dir_mc.m

```
function [thurr,xrw_mc,xrw_mc_ind,xrws_mc,xrws_mc_ind] = ...
    hurricane_dir_mc(nyr,cws,nseed)
%
%   INPUT FROM hurricane_dir_est.m ---> estimates1450_cw10 (for milepost1450),
%         and consists of estimtes of the parameters:
%
%               * (alpha1, cw, xi) of reverse Weibull distributions
%                 fitted to non-zero wind speeds in 16 direction.
%               * (alphas, xis) of reverse Weibull distributions
%                 fitted to non-zero wind speeds in 16 direction
%                 with imposed tail parameter cws = 10 (c = - 0.1)
%                 in all directions.
%               * p = 16-dimensional vector with probabilities
%                     p(i)=P(X_i>0) of non-zero wind speeds.
%               * zeta1 = (16,16) matrix of correlation coefficients
%                         for non-zero wind speeds.
%-------------------------------------------------------------------
%   OTHER INPUT:
%
%               * nyr = # of years required for simulation.
%               * nseed = Monte Carlo simulation seed.
%-------------------------------------------------------------------
%
%   OUTPUT:
%               * thurr = times of thunderstorms in nyr years.
%               * xrw_mc = generated wind speeds in 16 directions/nyr years
%                 using estimates of (alpha1, cw, xi), p(i), and zeta1.
%               * xrw_mc_ind = generated wind speeds in 16 directions/nyr years
%                 using estimates of (alpha1, cw, xi) and p(i) under the
%                 assumption that wind speeds in different directions
%                 are mutually independent.
%               * xrws_mc = generated wind speeds in 16 directions/nyr years
%                 using estimates of (alphas, xis), p(i), and zeta1 for
%                 an imposed tail parameter cws = - 1/c.
%               * xrws_mc_ind = generated wind speeds in 16 directions/nyr years
%                 using estimates of (alphas, xis) and p(i) for an imposed
%                 tail parameter cws = - 1/c under the assumption that wind
%                 speeds in different directions are mutually independent.
%==========================================================================
%
%   REASONS FOR THE INDEPENDENCE ASSUMPTION AND THE RECOMMENDATION OF
%   USING xrw_mc_ind; xrws_mc_ind RATHER THAN xrw_mc; xrws_mc
%
%       (1) Correlation coefficients of all data (including 0's) are
%           relatively small (maximum values are of order 0.7).
%       (2) Correlation coefficients between random variables with
%           finite probability mass at 0 provide limited information
%           on the relationship between these random variables.
%       (3) Estimates of the correlation coefficients of non-zero
%           wind speeds can lead to inconsistencies, e.g., consider
%           wind speed readings in 3 directions x(i,j), j=1,2,3,
%           each of length n = 1000, and suppose the readings
%           x(600:1000,1), x(1:400,2), x(800:1000,2), and x(1:600,3)
```

```
%           are zero.  The estimates of the correlation coefficients
%           of these records are rho(1,2) not=0 (records x(:,1) & x(:,2)),
%           rho(2,3) not=0 (records x(:,2) & x(:,3)), but rho(1,3)=0
%           (records x(:,1) & x(:,3)).
%
%=================================================================
%   Load estimates delivered by hurricane_dir_est.m
%   for a selected milepost (here milepost1450)
%----------------------------------------------------------------
%   load estimates350
load milepost1450
nu=mean_rate;
load estimates1450_cw10
nd=length(p);
%--------------------------------------------------------
%   Total number of hurricanes in nyr years:
%       thurr = a vector with entries times at which
%               hurricanes occurr in nyr years
%       nhurr = # of hurricanes in nyr years
%-------------------------------------------------------
rand('seed',nseed)
time=0;
ktime=0;
while time<=nyr,
    ktime=ktime+1;
    time=time-log(rand(1,1))/nu;
    thr(ktime)=time;
end,
nhurr=ktime-1;
thurr=thr(1:nhurr);
%----------------------------------------------------------------
%   Set 0 the entries of the matrices in which generated wind
%   will be stores
%-----------------------------------------------------------------
xrw_mc=zeros(nhurr,16);
xrw_mc_ind=zeros(nhurr,16);
xrws_mc=zeros(nhurr,16);
xrws_mc_ind=zeros(nhurr,16);
%-----------------------------------------------------------------
%   Generation of nhurr independent samples of a 16-dimensional
%   standard Gaussian vector with covariance matrix zeta1
%-----------------------------------------------------------------
%           Construct an approximate spectral representation
%           for a correlated standard Gaussian vector with
%           covariance approximating zeta1
%-----------------------------------------------------------------
[vzeta,dzeta]=eig(zeta1);
ndd=0;
for kd=1:nd,
    if dzeta(kd,kd)>0,
        ndd=ndd+1;
        lamz(ndd)=dzeta(kd,kd);
        phiz(:,ndd)=vzeta(:,kd);
    end,
end,
%-----------------------------------------------------------------
%           Generate required Gaussian samples
```

```
%-----------------------------------------------------------------
randn('seed',nseed);
gg=zeros(nhurr,nd);
for ks=1:nhurr,
    rg=randn(1,ndd);
    for kdd=1:ndd,
        gg(ks,:)=gg(ks,:)+lamz(kdd)*rg(kdd)*phiz(:,kdd)';
    end,
end,
gg=cdf('normal',gg,0,1);
%-------------------------------------------------------------
% figure
% mesh(1:16,1:nhurr,gg)
% xlabel('Wind direction')
% ylabel('Sample number')
% zlabel('Gaussian image')
% xlim([1 16])
% set(gca,'xticklabel','')
% set(gca,'xtick',[1:16])
% set(gca,'xticklabel',[1:16])
% ylim([1 nhurr])
% set(gca,'yticklabel','')
% set(gca,'ytick',[1 10:10:nhurr])
% set(gca,'yticklabel',[1 10:10:nhurr])
% %   print
%============================================================
%   Translation from Gaussian to reverse Weibull space
%   CASE 1: Estimates of (alpha1, cw, xi), p(i), and zeta1
%------------------------------------------------------------
%       gg=cdf('normal',gg,0,1);
for ks=1:nhurr,
    for i=1:nd,
        if gg(ks,i)>=1-p(i),
            uu=(gg(ks,i)-(1-p(i)))/p(i);
            xrw_mc(ks,i)=-xi(i)-icdf('wbl',uu,alpha1(i),cw(i));
        end,
%         [ks i gg(ks,i) 1-p(i) xrw_mc(ks,i)]
%         pause
    end,
end,
%***********************************
%   UNDER INDEPENDENCE ASSUMPTION
%***********************************
for ks=1:nhurr,
    for i=1:nd,
        ur=rand(1,1);
        if ur>=1-p(i),
            uu=(ur-(1-p(i)))/p(i);
            xrw_mc_ind(ks,i)=-xi(i)-icdf('wbl',uu,alpha1(i),cw(i));
        end,
%         [ks i gg(ks,i) 1-p(i) xrw_mc_ind(ks,i)]
%         pause
    end,
end,
%-------------------------------------------------------------
%   Translation from Gaussian to reverse Weibull space
%   CASE 2: Estimates of (alphas, xis), p(i), and zeta1
```

```
%           for an imposed tail parameter cws = - 1/c
%------------------------------------------------------------
%    gg=cdf('normal',gg,0,1);
for ks=1:nhurr,
    for i=1:nd,
        if gg(ks,i)>=1-p(i),
            uu=(gg(ks,i)-(1-p(i)))/p(i);
            xrws_mc(ks,i)=-xis(i)-icdf('wbl',uu,alphas(i),cws);
        end,
%         [ks i gg(ks,i) 1-p(i) xrws_mc(ks,i)]
%         pause
    end,
end,
%**********************************
%   UNDER INDEPENDENCE ASSUMPTION
%**********************************
for ks=1:nhurr,
    for i=1:nd,
        ur=rand(1,1);
        if ur>=1-p(i),
            uu=(ur-(1-p(i)))/p(i);
            xrws_mc_ind(ks,i)=-xis(i)-icdf('wbl',uu,alphas(i),cws);
        end,
%         [ks i gg(ks,i) 1-p(i) xrws_mc_ind(ks,i)]
%         pause
    end,
end,
%-------------------------------------------------------------
% figure
% mesh(1:16,1:nhurr,xrw_mc)
% xlabel('Wind direction')
% ylabel('Sample number')
% zlabel('Simulated hurricane wind speeds')
% xlim([1 16])
% set(gca,'xticklabel','')
% set(gca,'xtick',[1:16])
% set(gca,'xticklabel',[1:16])
% ylim([1 nhurr])
% set(gca,'yticklabel','')
% set(gca,'ytick',[1 10:10:nhurr])
% set(gca,'yticklabel',[1 10:10:nhurr])
% %   print
%------------------------------------------------
% figure
% contour(1:16,1:ns,xrws_mc)
% xlabel('Wind direction')
% ylabel('Sample number')
% title('Simulated hurricane wind speeds')
% xlim([1 16])
% set(gca,'xticklabel','')
% set(gca,'xtick',[1:16])
% set(gca,'xticklabel',[1:16])
% ylim([1 nhurr])
% set(gca,'yticklabel','')
% set(gca,'ytick',[1 10:10:nhurr])
% set(gca,'yticklabel',[1 10:10:nhurr])
% %   print
```

```
% %-----------------------------------------------------
% figure
% contour(1:16,1:ns,xweib)
% xlabel('Wind direction')
% ylabel('Sample number')
% title('Simulated hurricane wind speeds')
% xlim([1 16])
% set(gca,'xticklabel','')
% set(gca,'xtick',[1:16])
% set(gca,'xticklabel',[1:16])
% ylim([1 ns])
% set(gca,'yticklabel','')
% set(gca,'ytick',[1 10:10:ns])
% set(gca,'yticklabel',[1 10:10:ns])
% %   print
%==================================================================
%   [thurr,xrw_mc,xrw_mc_ind,xrws_mc,xrws_mc_ind]=hurricane_dir_mc(200000,10,123);
```