BLUE PRESSURE User’s Manual 2.15.2017

BLUE PRESSURE is a suite of Matlab (version 2014b) programs that calculates the extreme positive and negative (suction) wind pressure, using Lieblein’s (1974) BLUE (Best Linear Unbiased Estimate) method. The suite consists of:

* Function blue4pressure, which estimates the extreme pressures and calls function bluecoeff.
* Function bluecoeff, which provides the coefficients of BLUE for 4 ≤ n epochs ≤ 100 (n is an integer).

**Function [p1\_max, p2\_max, p1\_rmax, p2\_rmax, u\_max, b\_max, cp\_max, p1\_min, p2\_min, p1\_rmin, p2\_rmin, u\_min, b\_min, cp\_min] = blue4pressure (cp, n, P1, P2, dur)**

From a time series of pressure coefficients, blue4pressure estimates extremes of positive and negative pressures based on Lieblein's BLUE method applied to n epochs. Extremes are estimated for one and dur epochs for probabilities of non-exceedance P1 and P2 of the Gumbel distribution fitted to the epochal peaks.

INPUT

cp = vector of time history of pressure coefficients.

n = number of epochs (integer) of cp data, 4 ≤ n ≤ 100, for fitting Gumbel distribution.

dur ≥ 1 is the number of epochs for estimating extremes, and does not need to be an integer. It defaults to dur = n if user sets it to 0.

P1, P2 = probabilities of non-exceedance of extremes in EV1 (Gumbel). P1 defaults to 0.80 (ISO)and P2 to 0.5704 (mean) if user sets them to 0.

OUTPUT

suffix max for + peaks, min for - peaks of pressure coefficients.

p1\_max (p1\_min) = extreme value of positive (negative) peaks with probability of non-exceedance P1 for one epoch.

p2\_max (p2\_min) = extreme value of positive (negative) peaks with probability of exceedance P2 for one epoch.

p1\_rmax (p1\_rmin) = extreme value of positive (negative) peaks with probability of non-exceedance P1 for dur epochs.

p2\_rmax (p2\_rmin) = extreme value of positive (negative) peaks with probability of non-exceedance P2 for dur epochs.

cp\_max (cp\_min) = vector of n positive (negative) epochal peaks.

u\_max, b\_max (u\_min, b\_min) = location and scale parameters of EV1 (Gumbel) distribution for positive (negative) peaks.

**function [ai,bi] = bluecoeff (n)**

Function bluecoeff provides the coefficients of BLUE for 4 ≤ n epochs ≤ 100 (n is an integer).

INPUT:

n = number of epochs to divide data, 4 ≤ n ≤ 100.

OUTPUT:

ai, bi = coefficients for BLUE, calculated according to formulas by Lieblein (who provided coefficients for 4 ≤ n ≤ 16).

**Here is how to run the program:**

1. Select a time series of wind pressure and write a script to read it;
2. Place data, read-script, function blue4pressure (p. 7) and function bluecoeff in one directory (or use addpath command to fetch data);
3. Run the programs. An example run is shown on p. 3.

**Results**

Extreme wind pressures corresponding to the probabilities of non-exceedance P1 (default = 0.80 [ISO 4354]) and P2 (default = 0.5704 [mean]) of the Gumbel distribution fitted to the data by Lieblein’s BLUE method for one and for dur epochs, where dur > 1 does not have to be an integer. The default value is dur = n, i.e., same duration as data. The program also outputs the location (mu) and scale (sigma) parameters of the Gumbel distribution.

**References**

1) Lieblein, J. (1974) “Efficient Methods of Extreme Value Methodology,” NBSIR 74-602, National Bureau of Standards, Washington, DC 20234

2) Cook, N.J. (1985) "The designer's guide to wind loading of building structures" part 1, British Research Establishment, Table C3, pp. 321-323 for n = 17:24.

3) International Standard, ISO 4354 (2009-06-01), 2nd edition, “Wind actions on structures,” Annex D (informative) “Aerodynamic pressure and force coefficients,” Geneva, Switzerland, p. 22

**Examples:** in following pages, % indicates a Matlab non executable comment, in green. Results are in blue.

This example script reads pressure data file jp1tap708wind360 from the NIST database, [http://www.itl.nist.gov/div898/winds/datasets.htm](http://www.itl.nist.gov/div898/winds/datasets.htmUWO_BLUE4PRESSURE). For verification, it also plots the data, and finds its length, the max and the min and their indexes. Finally, it calls function blue4pressure to estimate extreme wind pressures using five sets of epochs, from 13 to 17 epochs, and write the results in file Blue4PressureResults.csv.

Example script UWO\_BLUE4PRESSURE

clear all;

% addpath('C:\windPRESSURE');

% addpath('C:\windPRESSURE\jp1tap708wind360blue'); % locate data

filename = '\_jp1tap708wind360.csv';

xlRange = 'A1:A49792';

cp = xlsread(filename,xlRange); % read data

plot(cp)

t = length(cp) % Size of data

% check time series is the correct one

[M,I] = max(cp) % maximum and its index

[m,i] = min(cp) % minimum and its index

Blueresults = zeros(5,13); % preset size of results

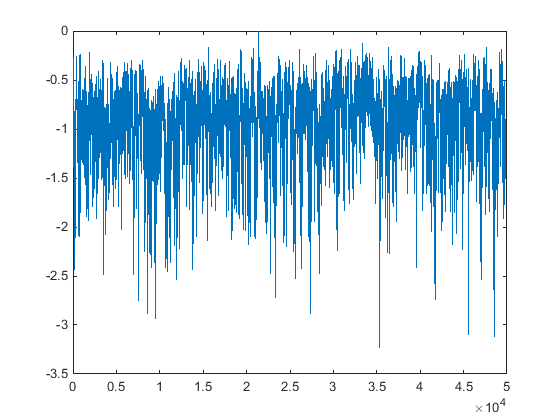
for n = 13:17 % try 5 sets of epochs from 13 to 17

[p1\_max, p2\_max, p1\_rmax, p2\_rmax, u\_max, b\_max, cp\_max, p1\_min, p2\_min, p1\_rmin, p2\_rmin, u\_min, b\_min, cp\_min] = blue4pressure(cp, n, 0, 0, 0) % call function

Blueresults(n,:) = [n;p1\_max;p2\_max;p1\_rmax;p2\_rmax;u\_max;b\_max; p1\_min;p2\_min;p1\_rmin;p2\_rmin;u\_min;b\_min]; % extract results

end

csvwrite(['\_Blue4PressureResults.csv'],Blueresults); % write results to table



Length t = 49792 data; Max = 0 at time I = 21379; min = -3.2387 at time i = 35359.

n = 13 epochs produce the following + and – peaks:

cp\_max = -0.2155 -0.3035 -0.2965 -0.2803 -0.1714 0 -0.1853 -0.1923 -0.1321 -0.1598 -0.2085 -0.2317 -0.1668

cp\_min = -2.4835 -2.7522 -2.9399 -2.5391 -2.1360 -2.4719 -2.7244 -2.8866 -1.8371 -3.2387 -2.7383 -3.1021 -3.1229

…

n = 17 epochs produce the following + and – peaks:

cp\_max = -0.2155 -0.3035 -0.2965 -0.4332 -0.2803 -0.1714 -0.1923 0 -0.1853 -0.2294 -0.1830 -0.1321 -0.1598 -0.2085 -0.2432 -0.2317 -0.1668

cp\_min = -2.4325 -2.4835 -2.8866 -2.9399 -2.5391 -2.1360 -2.1846 -2.7244 -2.5298 -2.8866 -2.2379 -1.7630 -3.2387 -2.4140 -2.7383 -3.1021 -3.1229

Table 1 Extreme + and – pressures for P1 = 0.80 and P2 = 0.57 for 1 epoch and n epochs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | max | | | | | |
|  | 1 epoch | | n epochs | | Gumbel | |
| n | 80% | 57% | 80% | 57% | location | scale |
| 13 | -0.13351 | -0.19471 | 0.036612 | -0.024584 | -0.233 | 0.066326 |
| 14 | -0.13087 | -0.19142 | 0.042335 | -0.018218 | -0.22931 | 0.06563 |
| 15 | -0.14496 | -0.19999 | 0.016559 | -0.038471 | -0.23442 | 0.059643 |
| 16 | -0.15019 | -0.2092 | 0.027154 | -0.03186 | -0.24612 | 0.063961 |
| 17 | -0.12567 | -0.20683 | 0.12355 | 0.042387 | -0.25761 | 0.087963 |
|  |  |  |  |  |  |  |
|  | min | | | | | |
|  | 1 epoch | | n epochs | | Gumbel | |
| n | 80% | 57% | 80% | 57% | location | scale |
| 13 | -3.1307 | -2.7302 | -4.244 | -3.8435 | -2.4797 | 0.43404 |
| 14 | -2.9572 | -2.6546 | -3.8226 | -3.52 | -2.4653 | 0.32793 |
| 15 | -2.9416 | -2.6656 | -3.7519 | -3.4759 | -2.4928 | 0.29922 |
| 16 | -3.0209 | -2.6535 | -4.1251 | -3.7577 | -2.4235 | 0.39826 |
| 17 | -2.993 | -2.6306 | -4.106 | -3.7436 | -2.4038 | 0.39284 |

Rerun with reversed signs.

n = 13 epochs produce the following + and – peaks:

cp\_max = 2.4835 2.7522 2.9399 2.5391 2.1360 2.4719 2.7244 2.8866 1.8371 3.2387 2.7383 3.1021 3.1229

cp\_min = 0.2155 0.3035 0.2965 0.2803 0.1714 0 0.1853 0.1923 0.1321 0.1598 0.2085 0.2317 0.1668

…

n = 17 epochs produce the following + and – peaks:

cp\_max = 2.4325 2.4835 2.8866 2.9399 2.5391 2.1360 2.1846 2.7244 2.5298 2.8866 2.2379 1.7630 3.2387 2.4140 2.7383 3.1021 3.1229

cp\_min = 0.2155 0.3035 0.2965 0.4332 0.2803 0.1714 0.1923 0 0.1853 0.2294 0.1830 0.1321 0.1598 0.2085 0.2432 0.2317 0.1668

Table 2 Extreme + and – pressures for P1 = 0.80 and P2 = 0.57 for 1 epoch and n epochs, reversed sign time series

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | max | | | | | |
|  | 1 epoch | | n epochs | | Gumbel | |
| n | 80% | 57% | 80% | 57% | location | scale |
| 13 | 3.1307 | 2.7302 | 4.244 | 3.8435 | 2.4797 | 0.43404 |
| 14 | 2.9572 | 2.6546 | 3.8226 | 3.52 | 2.4653 | 0.32793 |
| 15 | 2.9416 | 2.6656 | 3.7519 | 3.4759 | 2.4928 | 0.29922 |
| 16 | 3.0209 | 2.6535 | 4.1251 | 3.7577 | 2.4235 | 0.39826 |
| 17 | 2.993 | 2.6306 | 4.106 | 3.7436 | 2.4038 | 0.39284 |
|  |  |  |  |  |  |  |
|  | min | | | | | |
|  | 1 epoch | | n epochs | | Gumbel | |
| n | 80% | 57% | 80% | 57% | location | scale |
| 13 | 0.13351 | 0.19471 | -0.036612 | 0.024584 | 0.233 | 0.066326 |
| 14 | 0.13087 | 0.19142 | -0.042335 | 0.018218 | 0.22931 | 0.06563 |
| 15 | 0.14496 | 0.19999 | -0.016559 | 0.038471 | 0.23442 | 0.059643 |
| 16 | 0.15019 | 0.2092 | -0.027154 | 0.03186 | 0.24612 | 0.063961 |
| 17 | 0.12567 | 0.20683 | -0.12355 | -0.042387 | 0.25761 | 0.087963 |

Function blue4pressure

function [p1\_max, p2\_max, p1\_rmax, p2\_rmax, u\_max, b\_max, cp\_max, p1\_min, p2\_min, p1\_rmin, p2\_rmin, u\_min, b\_min, cp\_min] = blue4pressure(cp, n, P1, P2, dur)

% From a time series of pressure coefficients, blue4pressure

% estimates extremes of positive and negative pressures based on

% Lieblein's BLUE (Best Linear Unbiased Estimate) method applied

% to n epochs. Extremes are estimated for 1 and dur epochs for

% probabilities of non- exceedance. P1 and P2 of the Gumbel % distribution fitted to the epochal peaks.

% n = integer, dur need not be an integer.

% Written by Dat Duthinh 8\_25\_2015, 2\_2\_2016, 2\_6\_2017

% Reference: 1) Julius Lieblein "Efficient Methods of Extreme-

% Value Methodology" NBSIR 74-602 OCT 1974 for n = 4:16

% 2) Nicholas John Cook "The designer's guide to wind loading of

% building structures" part 1, British Research Establishment

% 1985 Table C3 pp. 321-323 for n = 17:24. Extension to n=100 by % Adam Pintar Feb 12 2016.

% 3) INTERNATIONAL STANDARD, ISO 4354 (2009-06-01), 2nd edition,

% “Wind actions on structures,” Annex D (informative)

% “Aerodynamic pressure and force coefficients,” Geneva, % Switzerland, p. 22

% INPUT

% cp = vector of time history of pressure coefficients

% n = number of epochs(integer)of cp data, 4 <= n <= 100

% dur = number of epochs for estimation of extremes.

% Default dur = n

% dur needs not be an integer

% P1, P2 = probabilities of non-exceedance of extremes in EV1 % (Gumbel)

% P1 defaults to 0.80 (ISO)and P2 to 0.5704 (mean).

% OUTPUT

% suffix max for + peaks, min for - peaks of pressure coeff.

% p1\_max (p1\_min)= extreme value of positive (negative) peaks

% with probability of non-exceedance P1 for 1 epoch

% p2\_max (p2\_min)= extreme value of positive (negative) peaks

% with probability of exceedance P2 for 1 epoch

% p1\_rmax (p1\_rmin)= extreme value of positive (negative) peaks

% with probability of non-exceedance P1 for dur epochs

% p2\_rmax (p2\_rmin)= extreme value of positive (negative) peaks

% with probability of non-exceedance P2 for for dur epochs

% cp\_max (cp\_min)= vector of n positive (negative) epochal peaks

% u\_max, b\_max (u\_min, b\_min) = mean and location parameters of

% EV1 (Gumbel) for positive (negative) peaks

% Size of cp array

t = length(cp);

% Initialize variables

cp\_max = zeros(1,n);

cp\_min = zeros(1,n);

% Find the peaks for each of the n user-defined epochs

% and store in cpmax and cpmin arrays

% Separate cases if n evenly divides t or not

r = rem(t,n);

if r == 0

for i = 1:n

a = cp(1+(i-1)\*t/n:i\*t/n);

cp\_max(i) = max(a);

cp\_min(i) = min(a);

end

elseif r > n/2

q = fix(t/n)+1;

for i = 1:n-1

a = cp(1+(i-1)\*q:i\*q);

cp\_max(i) = max(a);

cp\_min(i) = min(a);

end

a = cp(1+(n-1)\*q:t);

cp\_max(n) = max(a);

cp\_min(n) = min(a);

else

q = fix(t/n);

for i = 1:n-1

a = cp(1+(i-1)\*q:i\*q);

cp\_max(i) = max(a);

cp\_min(i) = min(a);

end

a = cp(1+(n-1)\*q:t);

cp\_max(n) = max(a);

cp\_min(n) = min(a);

end

% Coefficients for all n

[ai,bi]= bluecoeff(n);

% Organize values in ascending or descending order

x\_max = sort(cp\_max);

x\_min = sort(cp\_min,'descend');

if (P1==0)

P1 = 0.80;

end

if (P2==0)

P2 = 0.5704;

end

if (dur==0)

dur = n;

end

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MAX CASE PEAK \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

u = 0; % location parameter

b = 0; % scale parameter

% Calculate parameters of location and scale

for j = 1:n

u = u + ai(j)\*x\_max(j);

b = b + bi(j)\*x\_max(j);

end

p1\_max = u - b\*log(-log(P1)); % for 1 epoch

p1\_rmax = p1\_max + b\*log(dur); % for longer duration

p2\_max = u - b\*log(-log(P2)); % for 1 epoch

p2\_rmax = p2\_max + b\*log(dur); % for longer duration

u\_max = u;

b\_max = b;

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MIN CASE PEAK \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

u = 0;

b = 0;

% Calculate parameters of location and scale

for j = 1:n

u = u + ai(j)\*x\_min(j);

b = b + bi(j)\*x\_min(j);

end

p1\_min = u - b\*log(-log(P1)); % for 1 epoch

p1\_rmin = p1\_min + b\*log(dur); % for longer duration

p2\_min = u - b\*log(-log(P2)); % for 1 epoch

p2\_rmin = p2\_min + b\*log(dur); % for longer duration

u\_min = u;

b\_min = abs(b);

end