ORCA: Spatial statistical analysis

Sub-project of Roadmap Loads and Strengths (9.1a, salt)

ORCA: Spatial statistical analysis

Sub-project of Roadmap Loads and Strengths (9.1a, salt)

Sofia Caires and Caroline Bos

1200266-003

Deltares

Title
ORCA: Spatial statistical analysis

Client	**Project**	**Reference**	**Pages**
SO 2009	1200266-003	1200266-003-HYE-0003-	29

Keywords
Spatial analysis, Error analysis, Extreme value analysis, ORCA, software development

Abstract

Deltares frequently carries out metocean studies. These studies involve quite a lot of statistical analyses, which are for a great part carried out using and in-house MATLAB tool called ***ORCA*** (metOcean data tRansformation, Classification and Analysis). At the moment, there are five functionalities available on the ***ORCA*** tool: Data validation, Normal conditions, Extreme conditions, Sea State analysis and Persistence statistics. Due to ***ORCA***'s flexibility and easiness of use, it is extensively used in Deltares projects, which leads to frequent requests and suggestions for the extension of the tool functionalities. In particular, the need has been found to add spatial error and extreme value analyses to the tool.

In the project reported here, the ***ORCA*** tool has been extended and now includes the possibility to carry out efficient spatial error analysis for different combinations of spatial data (sparse satellite observations, gridded data, etc). Furthermore, the particularities of linear and circular variables are properly considered when computing error statistics. In addition, ***ORCA*** can now also be used to carry out spatial extreme value analyses, in which uncertainties in the estimates can be decreased by means of a simple technique known as Regional Frequency Analysis.

References

Project plan for 2010 of the Roadmap "Loads and Strengths (9.1a, salt)" (Ap van Dongeren).

Version	Date	Authors	Initials	Review	Initials	Approval	Initials
1	dec. 2009	Sofia Caires and Caroline Bos	SC	Klaas Jan Bos	KJ	Klaas Jan Bos	KJ

State
final

Contents

Appendices

1 Introduction

1.1 Background

Deltares frequently carries out metocean studies. These studies involve statistical analyses, which are for a great part carried using a MATLAB tool called ***ORCA*** (metOcean data tRansformation, Classification and Analysis). At the moment, there are five functionalities available on the ***ORCA*** tool: Data validation, Normal conditions, Extreme conditions, Sea State analysis and Persistence statistics. Each functionality deals with certain aspects of data analysis, namely:

- Data validation - Procedures necessary for the definition of a "good data set". Consisting of, among others, procedures for reading of different data, analysis of gaps in timeseries, data visualization, data comparison and error statistics and plots, and data correction.
- Normal conditions - Procedures necessary to the definition of mean climates. Consisting of, among others, procedures for data classification, creation of wave roses, data transformation and definition of mean climate scenarios.
- Extreme conditions - Procedures for extreme value analysis. Consisting of, among others, procedures for the collection of peak-over-threshold data, the collection of annual maxima data, definition of optimal threshold for the estimation of the generalized Pareto distribution parameters, return value estimation, fitting of statistical distributions.
- Sea State analysis - Procedures for sea state analysis. Consisting of, among others, procedures for fitting of different spectral forms to discrete spectral data, estimation of sea state wave statistic (e.g: number of waves, H1%).
- Persistence statistics - Procedures for the computation of persistence statistics.

1.2 Motivation

ORCA was developed with the support of the software house of the Hydraulic Engineering (HYE) unit of Deltares. It was developed for internal use and eventual commercialization after consolidation by internal use. The tool is currently extensively used in HYE projects and, since it is available to all units in Deltares, it is also becoming popular in other units. The extensive use of ***ORCA***, its flexibility and easiness of use leads to frequent requestes and suggestions for the extension of the tool functionalities.

For instance, the need has been found to extend the extreme conditions module of ***ORCA*** to account for spatial correlations. Furthermore, as is the case for extreme value analysis, error analysis can only be carried out at a specific location and there is the whish to extend the tool in order to be possible to carry out spatial error analysis in an efficient way. There is currently an ad-hoc and not documented MATLAB script available in the ZKS unit of Deltares, which allows quantifying the performance of tidal models spatially by mean of root-mean-square errors and vector differences between model results and in-situ and remote measurements. It is desirable to have such error analyses available in ***ORCA***

1.3 Objectives

The goals of this project are two fold:

1. To extend the ***ORCA*** extreme conditions tool to allow for spatial extreme value analysis.
2. To extend the ***ORCA*** data validation tool to allow for spatial error analysis.

1.4 Approach

The following sequential approach was followed in this study:

- For each tool extension, the statistical analyses that needed to be carried out were studied and inventorized.
- The ***ORCA*** code was extended to include the needed functionalities (spatial analysis).
- Exemplary scripts with the added analyses were created and tested.
- Guidelines for the use and adaptation of the created exemplary scripts were written.

1.5 Outline of this report

The ***ORCA*** philosophy and setup is given in Chapter 2. In Chapter 3 the considered and implemented error statistics are described and guidelines are given on how to perform a spatial error analysis using ***ORCA***. In Chapter 4 the theoretical background of extreme value analyses and guidelines on how to perform a spatial extreme value analysis using ***ORCA*** are given. The report ends in Chapter 5 with a concluding summary. Appendix A lists the ***ORCA*** routines developed in this project

2 *ORCA* philosophy and setup

2.1 Introduction

The development of ***ORCA*** has followed a philosophy close to that used in Deltares McTools (see http://public.deltares.nl/display/MCTDOC/McTools+Website).

The ***ORCA*** setup aims at being as generic as possible so that it can be easily extended/improved, so that we can profit as much as possible from MATLAB script developments. In this way, specific project needs not yet available in ***ORCA*** during the execution of the project, can be fulfilled more easily.

The toolbox contains a range of practical analysis tools that are set up conform predescribed rules. An important rule for entering a new application into ***ORCA*** is that it should be structured according to a few relatively simple but strict rules:

- The MATLAB scripts are divided in 3 main types: *engines*, *applications* and *scripts*.
- General elements in routines should be separate functions and should be stored in the engine directory.
- Engines should be as generic as possible with general (non ***ORCA*** specific) inputs like x, y and z.
- Scripts provide examples of specific data analyses.
- The data communicated within applications and scripts has a predefined data structure; the OrcaData type (see below).
- Each routine should have a proper help description according to the ***ORCA*** standard help format (see below).

In the ***ORCA*** directory (w:\budata\hyedata\morelis\matlabTools\orca\) a small number of subdirectories exist:

- dataIO containing data input/output code
- DataValidation containing the Data Validation Tool code
- Extreme containing the Extreme Conditions Tool code
- General containing generic (not specific to a certain functionality) code
- Normal containing the Normal Conditions Tool code
- Persistence containing the Persistence Statistics Tool code
- Plot containing generic plotting code
- Seastate containing the Sea Stata Analysis Tool code

Each of these subdirectories contains the following three subdirectories:

- Engines low level/basic functions
- Applications functions using one or more engines
- Scripts scripts exemplifying how some analyses can be carried out.

2.2 Data structure

In ***ORCA*** the format of the data communicated between applications is the OrcaData structure. All the data input scripts transform the input data into an OrcaData structure.

OrcaData had, before the beginning of this project, four main structure types:

- 'timeseries' type Used for time series of one or more parameters (time is also a separate parameter) with given units.
- 'class' type Used for classified data, generally describes exceedances or occurrences in percentages or days of different combinations of parameters (e.g. Height/Period/Direction).
- 'conditions' type Used for different conditions with a specified duration. This type is similar to the time series type except for the 'duration' parameter.
- 'spectrum' type Used for wave spectra information.

The OrcaData structure also contains some common fields, such as name, type, location etc.

2.3 Help

Each ***ORCA*** function should have a proper help description. The style guidelines for the help blocks of ***ORCA*** code are as given below.

```
function varargout = FUNTION_NAME(varargin)
%FUNTION_NAME   Concise 1-line function description (will appear in list of functions)
%
% -------------------------------------------------------------
% Copyright (c) Deltares year FOR INTERNAL USE ONLY
% Version:      #th Beta Version #.#, <date>
% -------------------------------------------------------------
%
% Syntax:  varargout = FUNTION_NAME(varargin) Description of other combination.
%
% Input:  <variable description>
%
% Output:  <Output description>
%
% See also OTHER_FUNTION_NAME, …, YET_ANOTHER_FUNTION_NAME

%   Insert at least one a blank line after the "see also" block.
%   Only the first comment block is considered help documentation by MATLAB.
%   Put the relevant authorship and procedure description before the subsequent
%   MATLAB code, e.g.:
%0. Authors:
%   date: author, describe contribution
%   date: author, describe contribution
%
%1. Method:
%
% fill in relevant information
%
%2. Modules Used:
%
% fill in relevant information

%
```

2.4 How to get started

The ***ORCA*** guidelines are available in W:\bulletin\caires\ORCA\Guidelines\guidelines.pdf. The ***ORCA*** toolbox is available in the Deltares network and can be invoked in MATLAB as follows.

```
if isempty(which('createEmptyOrcaData.m'))
    addpath W:\budata\hyedata\morelis\matlabTools\;    addRMPaths;
    Orcainit;
end
```

2.5 How to contribute to ***ORCA***

It is very easy to add MATLAB programs developed using the above mentioned ***ORCA*** predescribed rules which would improve/extend the ***ORCA*** toolbox to it. Such programs are in fact highly desirable for the ***ORCA*** toolbox and the ***ORCA*** toolbox maintenance and support team appreciate any sort of code contributions.

The version management of the ***ORCA*** toolbox is done using Subversion (see http://public.deltares.nl/display/MCTDOC/Getting+started+with+TortoiseSVN) by the ***ORCA*** main developers. Somebody wishing to contribute to the ***ORCA*** toolbox is advised to send their ***ORCA*** compliant code to a member of the development, maintenance and support team. After approved, subversion is used to incorporate the new code in the ***ORCA*** toolbox.

Contributions to ***ORCA*** can also be made in the form of providing the development, maintenance and support team with comments about problems in the code. These can be reported in w:\budata\hyedata\caires\ORCA\aandachtspunten.xls.

3 Spatial error analysis

3.1 Introduction

In this chapter the implemented spatial error analysis is described. We start by describing which error statistics have been implemented, according to whether linear or circular data is being considered. We then describe how these statistics can be computed at several spatial locations (spatial analysis) efficiently. We end the chapter with guidelines on how to perform a spatial error analysis using ***ORCA***.

3.2 Error statistics

The error statistics that have been implemented in ***ORCA*** are described in this section.

A particularity of certain environmental data (e.g. wave data) is that they can be classified into *linear data* (*e.g.* mean wave period and significant wave height) and *circular data* (*e.g.* mean wave direction and directional spread), and this distinction has to be taken into consideration when carrying out error analysis. Indeed, the statistical techniques for dealing with these two types of data are different—circular (or directional) data require a special approach. Basic concepts of statistical analysis of circular data are given in the books of Mardia (1972) and Fisher (1993).

3.2.1 Linear variables

Differences between linear variables are often quantified using the following standard statistics:

- the bias: $\bar{y}-\bar{x}$; (3.1)
- the root-mean-square error: $RMSE=\sqrt{n^{-1}\sum(y_i-x_i)^2}$; (3.2)
- the scatter index: $SI=\sqrt{n^{-1}\sum\left[\left(y_i-\bar{y}\right)-\left(x_i-\bar{x}\right)\right]^2}\Big/\bar{x}$; (3.3)
- the correlation coefficient: $\rho=\sum\left[\left(x_i-\bar{x}\right)-\left(y_i-\bar{y}\right)\right]\Big/\sqrt{\sum\left(x_i-\bar{x}\right)^2\sum\left(y_i-\bar{y}\right)^2}$; and (3.4)
- the symmetric slope: $r=\sqrt{\sum x_i^2/\sum y_i^2}$. (3.5)

In all these formulae the x_i's usually represent observations (or the dataset which is considered less uncertain or baseline), the y_i's represent the model results (or the dataset which is considered more uncertain or with a certain deviation from the baseline results) and n the number of observations.

3.2.2 Circular variables

If we compute an average of angles as their arithmetic mean, we may find that the result is of little use as a statistical location measure. Consider for instance the case of two angles of 359º and 1º; their arithmetic mean is 180º, when in reality 359º is only two degrees away from

1º and the mid direction between the two is 0º. This phenomenon is peculiar to circular data, and illustrates the need for special definitions of statistical measures in general.

When dealing with circular data, each observation is considered as unit vector, and it is vector addition rather than ordinary (or scalar) addition that is used to compute the average of angles, the so-called mean direction.

Writing

$$C_n = \sum_{i=1}^{n} \cos x_i \quad \text{and} \quad S_n = \sum_{i=1}^{n} \sin x_i \,, \tag{3.6}$$

the *sample resultant vector* R_n of a sample $\mathbf{x} = \{x_i,\ i = 1,...,n\}$ is defined as

$$R_n = \sqrt{C_n^2 + S_n^2} \,, \tag{3.7}$$

and its *sample mean direction* $\bar{x} \equiv \bar{x}_n$ as the direction of R_n:

$$\bar{x} = TAN^{-1}(S_n / C_n) \tag{3.8}$$

where $TAN^{-1}(S_n / C_n)$ is the inverse of the tangent of (S_n / C_n) in the range $[0, 2\pi[$, *i.e.*,

$$TAN^{-1}(S_n/C_n) := \begin{cases} \tan^{-1}(S_n/C_n), & S_n > 0,\ C_n > 0 \\ \tan^{-1}(S_n/C_n) + \pi, & C_n < 0 \\ \tan^{-1}(S_n/C_n) + 2\pi, & S_n < 0,\ C \ > 0. \end{cases} . \tag{3.9}$$

The *sample mean resultant length* of $\mathbf{x} = \{x_i,\ i = 1,...,n\}$ is defined by

$$\bar{R}_n = R_n / n \,, \qquad 0 \le \bar{R} \le 1 \,. \tag{3.10}$$

If $\bar{R}_n = 1$, then all angles coincide.

Eq. (3.9) can be used to compute the bias between two circular variables by substituting x_i by $y_i - x_i$ in Eq. (3.6). In a similar way, the root-mean-square error and scatter-index between two circular variables can be computed.

Since circular data are concentrated on $[0^o, 360^o]$, and in spite of the analogies with the linear case, it makes no sense to consider a symmetric slope for circular data other than one.

There are several circular analogues of the correlation coefficient, but the most widely used is the one proposed by Fisher and Lee (1983), the so-called *T*-linear correlation coefficient. Given two sets $\mathbf{x} = \{x_i,\ i = 1,...,n\}$, $\mathbf{y} = \{y_i,\ i = 1,...,n\}$ of circular data, the *T-linear correlation coefficient* between **x** and **y** is defined by

$$\rho_T = \frac{\sum_{1\le i<j\le n} \sin(x_i - x_j)\sin(y_i - y_j)}{\sqrt{\sum_{1\le i<j\le n} \sin^2(x_i - x_j) \sum_{1\le i<j\le n} \sin^2(y_i - y_j)}} . \tag{3.11}$$

This statistic satisfies $-1 \le \rho_T \le 1$, and its population counterpart (which is not given here but can be seen in Fisher and Lee, 1983) satisfies properties analogous to those of the usual population correlation coefficient for linear data: that is, the population counterpart achieves the extreme values -1 and 1 if and only if the two population variables involved are exactly '*T*-linear associated', with the sign indicating discordant or concordant rotation, respectively (see Fisher (1993), p. 146, for these concepts).

For computational ease, we use an equivalent formula for ρ_T, given by Fisher (1993):

$$\rho_T = \frac{4(AB - CD)}{\sqrt{(n^2 - E^2 - F^2)}\sqrt{(n^2 - G^2 - H^2)}} , \tag{3.12}$$

where

$$\begin{aligned}
A &= \sum_{i=1}^{n} \cos x_i \cos y_i \ , \ B = \sum_{i=1}^{n} \sin x_i \sin y_i \ , \\
C &= \sum_{i=1}^{n} \cos x_i \sin y_i \ , \ D = \sum_{i=1}^{n} \sin x_i \cos y_i \ , \\
E &= \sum_{i=1}^{n} \cos(2x_i) , \quad F = \sum_{i=1}^{n} \sin(2x_i) , \\
G &= \sum_{i=1}^{n} \cos(2y_i) , \quad H = \sum_{i=1}^{n} \sin(2y_i) .
\end{aligned} \tag{3.13}$$

3.2.2.1 *Vector data*

For vector data (data with an x- and a y-component, for instance the u and v wind velocity components) an error statistic know as the *mean vector difference,*

$$VM = n^{-1} \sum_{i=1}^{n} \sqrt{(x_{1i} - x_{2i})^2 + (y_{1i} - y_{2i})^2} \quad , \tag{3.14}$$

is often considered.

3.3 Spatial error analysis

The above expressions of error statistics are for a given pair of single location ***x*** and ***y*** data. For instance, a timeseries of satellite measurements and model results at a given location or timeseries of two different model results at a given location. In MATLAB, when considering multiple locations in which the error analysis should be carried out, e.g. gridded model data or data at a set of nearby measuring locations, i.e. multidimensional data, computations can be carried out more efficiently using multidimensional operations, instead of computing statistics location per location (unidimensional). Therefore, in order to increase the computational efficiency, when the expressions allow, the above expressions have been implemented allowing the 'simultaneous' (single statement) computation of the required error statistics at all considered locations using multidimensional operations.

3.4 Guidelines on how to perform a spatial error analysis using *ORCA*

An exemplary ***ORCA*** script that could be easily adjusted to perform a spatial extreme value analysis is:

`w:\budata\hyedata\morelis\matlabTools\orca\toolbox\DataValidation\scripts\example_spatial_error_analysis.m`

Running this script will result in a spatial error analysis. All necessary input from the user is defined in this script. The steps of the analysis are explained below, on the basis of the MATLAB script. In each step, all applications and engines directly used are indicated below the text box.

```
% SPATIAL_ERROR_ANALYSIS - Batch script to perform multiple location error analysis
%
% -------------------------------------------------------------
% Copyright (c) Deltares 2009 FOR INTERNAL USE ONLY
% Version:      2st Beta Version 0.0, <2009 December 31st>
% -------------------------------------------------------------
%
% Syntax:   spatial_error_analysis
%
% This script represents an example for a spatial error analysis. Please apply the
% order of functions given below for analysis.
%
%0. Authors:
%   10/2009: CW Bos
%   12/2009: S. Caires, adjustments

% -------------------------------------------------------------------------
clear all;
close all;
% -------------------------------------------------------------------------
% ORCA settings
% -------------------------------------------------------------------------
if isempty(which('qpread.m'));
    wlsettings;
end;
if isempty(which('createEmptyOrcaData.m'));
    addpath W:\budata\hyedata\morelis\matlabTools\;
    addRMPaths;
    ORCAinit;
end;
```

Step 1 Initialization

In Step 1, the Initialization, a few settings are defined. The user can leave these as they are.

```
%-------------------------------
% DEFINE GENERAL PARAMETERS
%-------------------------------
dinput='W:\budata\hyedata\caires\ORCA\Guidelines\exampleData\DataValidation\'; % INPUT DIRECTORY
doutput='W:\budata\hyedata\caires\ORCA\Guidelines\exampleData\DataValidation\results\'; % OUTPUT DIRECTORY
```

Step 2 User defined general parameters

In Step 2, the user has to define a few input parameters.

Directory names:

- dinput = 'W:\budata\hyedata\caires\ORCA\Guidelines\exampleData\DataValidation\'
  Directory where the data are located.
- doutput = ‘W:\budata\hyedata\caires\ORCA\Guidelines\exampleData\ DataValidation \'
  Directory where the output is to be saved.

```
% ------------------------------------------------------------------------
% USER DEFINED INPUT DATA
% ------------------------------------------------------------------------
% TWO ORCA DATA STRUCTURES ARE NEEDED WHICH WILL BE COMPARED TO EACH OTHER.
% THE ORCA STRUCTURES MAY BE OF THE TYPE MAP, META OR TIMESERIES.
% POSSIBLE COMBINATIONS FOR THE TWO DATA STRUCTURES ARE:
% map - map --> finput = finput2 = 'map'
% meta - meta --> finput = finput2 = 'meta'
% timeseries - timeseries --> finput = finput2 = 'timeseries'
% timeseries - meta --> finput = 'timeseries'; finput2 = 'meta'
% timeseries - map --> finput = 'timeseries'; finput2 = 'map'

% Important: Please note that the *.mat files need to be available locally.
% MATLAB cannot open them from the 'W:budata' folder!

%finput='map';finput2='map'; %example 1
finput='timeseries';finput2='timeseries'; %example 2
%finput='meta';finput2='meta'; %example 3
%finput='timeseries';finput2='map'; %example 4
%finput='timeseries';finput2='meta'; %example 5

% ------------------------------------------------------------------------
param1='hsig wave vector (mean direction)';                  % NAME OF 1ST PARAMETER TO BE ANALYSED
param2='hsig wave height';                                   % OPTIONAL, NAME OF 2ND PARAMETER TO BE
ANALYSED

if strcmp(finput,'map')==1 % map - map
    load([ 's1-map.mat']);                                   % (if map data applied): LOAD MAP ORCA DATA
STRUCTURE
    load([ 's2-map.mat']);                                   % (if map data applied): LOAD SECOND MAP ORCA
DATA STRUCTURE
elseif strcmp(finput,'meta')==1 % meta - meta
    s1=createemptyorcadata('meta');
    s1.name='model data';                                    % (if meta data applied): DESCRIPTION OF DATA
    s1.location.coordinates=[];
    s1.location.system='';
    s1.data=[dinput 'wavm-kh1-all.dat'];                     % (if meta data applied): DATA FILE NAME
    s1.parameter.name=param1;
    s1.parameter(2).name=param2;
    s1.parameter(1).routine='vs_use';                        % (if meta data applied): ROUTINE TO BE
APPLIED FOR INITIATING DATA
    s1.parameter(1).read='vs_let';                           % (if meta data applied): ROUTINE TO BE
APPLIED FOR READING DATA
    s1.parameter(1).groupname='map-series';                  % (if meta data applied): GROUPNAME FOR
READING DATA (only by applying vs_let)
    s1.parameter(1).groupindex={1:114};                      % (if meta data applied): GROUPINDEX FOR
READING DATA (only by applying vs_let)
    s1.parameter(1).elementname='DIR';                       % (if meta data applied): ELEMENTNAME FOR
READING DATA (only by applying vs_let)
    s1.parameter(1).elementindex={1:365,1:271};              % (if meta data applied): ELEMENTINDEX FOR
READING DATA (only by applying vs_let)
    s1.parameter(2).routine='qpfopen';                       % (if meta data applied): ROUTINE TO BE
APPLIED FOR INITIATING DATA (2ND PARAMETER)
    s1.parameter(2).read='qpread';                           % (if meta data applied): ROUTINE TO BE
APPLIED FOR READING DATA (2ND PARAMETER)
    s2=s1;
    s2.data=[dinput 'wavm-kh2-all.dat'];                     % (if meta data applied): 2ND DATA FILE NAME
TO COMPARE WITH 1ST DATA STRUCTURE
elseif strcmp(finput,'timeseries')==1
    load([dinput 's1-timeseries.mat']);                      % (if timeseries data applied): LOAD
TIMESERIES DATA
    if strcmp(finput2,'timeseries')==1 % timeserie - timeseries
        load([dinput 's2-timeseries.mat']);                  % (if timeseries data applied): LOAD SECOND
TIMESERIES DATA
    elseif strcmp(finput2,'meta')==1 % timeserie - meta
        s2=createemptyorcadata('meta');
        s2.name='model data';                                % (if meta data applied for 2nd data): see
above for description meta structure data fields
        s2.data=[dinput 'wavm-maldives.dat'];
        s2.parameter.name=param1;
        s2.parameter(2).name=param2;
        s2.parameter(1).routine='vs_use';
        s2.parameter(1).read='vs_let';
        s2.parameter(1).groupname='map-series';
        s2.parameter(1).groupindex={1:30};
        s2.parameter(1).elementname='DIR';
        s2.parameter(1).elementindex={1:168,1:303};
        s2.parameter(2).routine='qpfopen';
        s2.parameter(2).read='qpread';
    elseif strcmp(finput2,'map')==1 % timeserie - map
        %        load([dinput 's2-map.mat']);                % (if map data applied for 2nd data): LOAD
MAP ORCA DATA STRUCTURE
        load(['s2-map.mat']);                                % (if map data applied for 2nd data): LOAD
MAP ORCA DATA STRUCTURE
    end
end
```

Step 3 Used defined data to be analysed. General engine used: <createemptyorcadata>.

In Step 3, the user should define the data to be analysed. Different data combinations are possible as indicated: map – map; meta – meta; timeseries – timeseries; timeseries – meta; and timeseries – map. Examples for all possible combinations are provided.

```
% User Data structure for plots: SIZE OF uDATA IS SAME AS NUMBER OF
% PARAMETERS TO BE ANALYSED
% -------------------------------------------------------------------------
% 1st parameter to analyse: wave direction (example)
% NOTE: WAVE DIRECTION, SO A '*' SHOULD BE WRITTEN BEFORE THE PARAMETER NAME
uData.name={'*hsig wave vector (mean direction)',...
    '*hsig wave vector (mean direction)'};                      % {1 X 2 CELL}: NAME OF PARAMETERS TO BE
ANALYSED FOR S1 AND S2 RESPECTIVELY)
uData.plot={'bias','rmse','correlation_coefficient'};           % NAME OF PARAMETERS IN STATS STRUCTRE TO BE
PLOTTED WITH PLOTSTATS
uData.ldb=[dinput 'ecm_WGS84-UTM40.ldb'];                       % NAME OF LANDBOUDNARY FILE
uData.doutput=doutput;
uData.xlim=[239000 285e3];                                      % XLIM
uData.ylim=[2727150 2775e3];                                    % YLIM
uData.xlabel='Easting [m]';                                     % XLABEL
uData.ylabel='Northing [m]';                                    % YLABEL
if strcmp(finput,'timeseries')==1
    uData.ldb=[dinput 'maldives.ldb'];
    uData.xlim=[72 74.5];
    uData.ylim=[1 5];
    uData.xlabel='Longitude [^{o}N]';                           % XLABEL
    uData.ylabel='Latitude [^{o}N]';                            % YLABEL
end
uData.figNr={'Fig. 1.1','Fig. 1.2','Fig. 1.3'};                 % FIGURE NUMBER
uData.filename={'Fig101_Bias','Fig102_RMSE','Fig103_CorrCoefficient'};   % FILENAME OF FIGURES
uData.cbarTitle={'Bias (^o)', 'RMSE (^o)','Correlation coefficient'};    % Colorbar titles for the figures (must be in
the same order as the parameters structure going into plotStats!)
uData.title=strvcat(['Results of example\_spatial\_error\_analysis'],...   % TITLE OF FIGURES (TEXT OF MDPAPER)
    ['Statistics MWD']);
uData.text1='ORCA';                                             % TEXT1 OF FIGURES (TEXT OF MDPAPER)
uData.text2='';                                                 % TEXT2 OF FIGURES (TEXT OF MDPAPER)
uData.text3='';                                                 % TEXT3 OF FIGURES (TEXT OF MDPAPER)
uData.projNr='1200266.003';                                     % PROJECT NUMBER OF FIGURES (TEXT OF MDPAPER)
% -------------------------------------------------------------------------
% 2nd parameter to analyse: wave height (example)
uData(2)=uData(1);                                              % FILL IN 2nd UDATA IF ANOTHER PARAMETER IS
ANALYSED
uData(2).name={'hsig wave height','hsig wave height'};
uData(2).plot={'bias','symmetric_slope','scatter_index','rmse'};
uData(2).filename={ 'Fig201_bias','Fig201_SS','Fig202_SI','Fig203_RMSE'};
uData(2).cbarTitle={'Bias (m)','Symmetric slope','S.I. (%)', 'RMSE (m)'};
uData(2).title=strvcat(['Results of example\_spatial\_error\_analysis'],...
    ['Statistics H_s']);
uData(2).figNr={'Fig. 2.1','Fig. 2.2','Fig. 2.3','Fig. 2.4','Fig. 2.5'};
% -------------------------------------------------------------------------
% END OF USER DEFINED INPUT DATA
% -------------------------------------------------------------------------
```

Step 4 User defined data structure for the analysis and plotting of the results.

In Step 4, the used defines the variables that should be analysed and the plotting settings for the application <plotStats>.

```
% -------------------------------------------------------------------------
% SPATIAL ERROR ANALYSIS STARTS BELOW
stats=spatErAnalysis(s1,s2,uData);
plotStats(stats,uData);
```

Step 5 Spatial error analysis and plotting of the results.. Applications used: <spatErAnalysis> and <plotStats>.

In Step 5, the spatial error analysis is carried out and the requested data plotted. Figures 3.1 to 3.3 show a few examples of plotted results. Plots are created already with the Deltares standard formatting so that they can be directly added to Deltares reports.

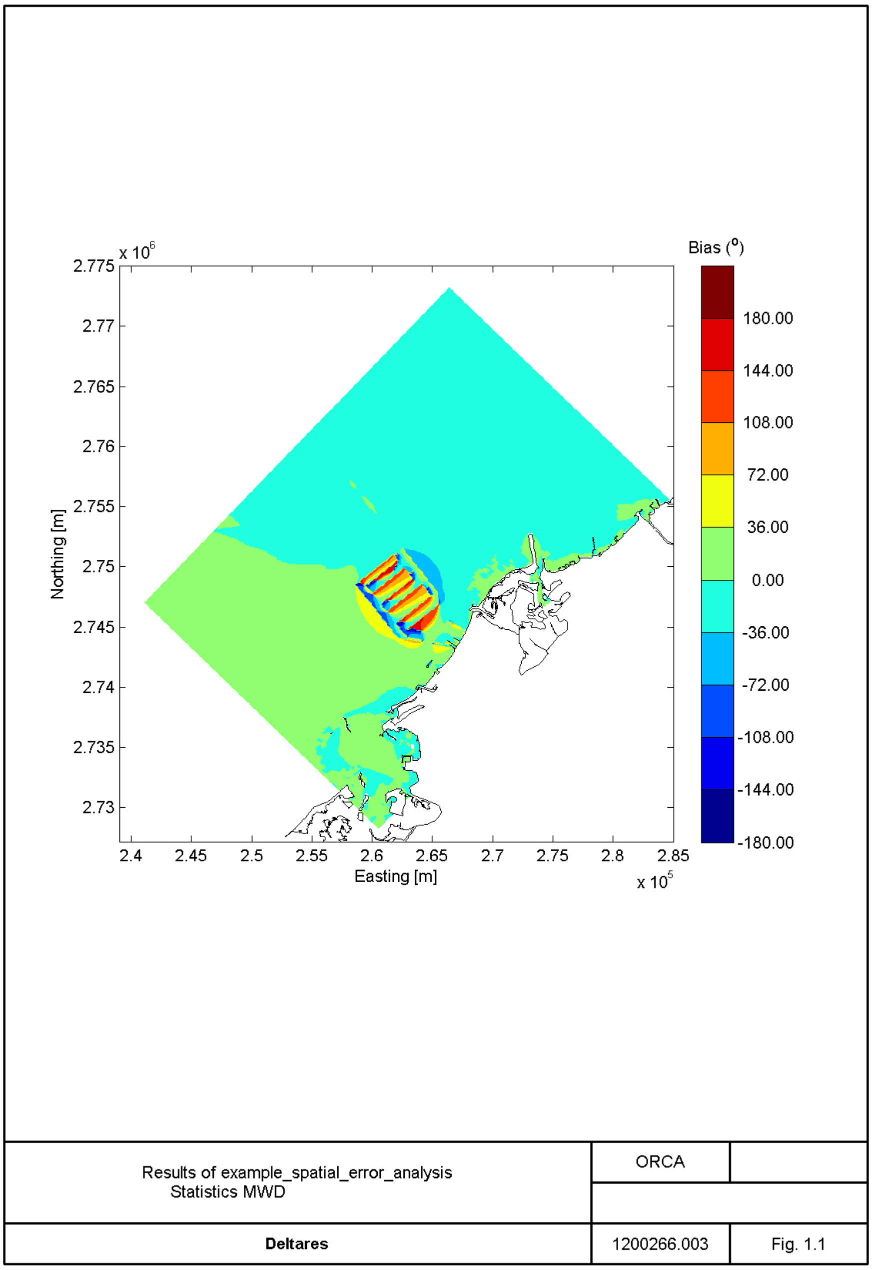


Figure 3.1 *Bias of two circular variables of the 'map' type.*

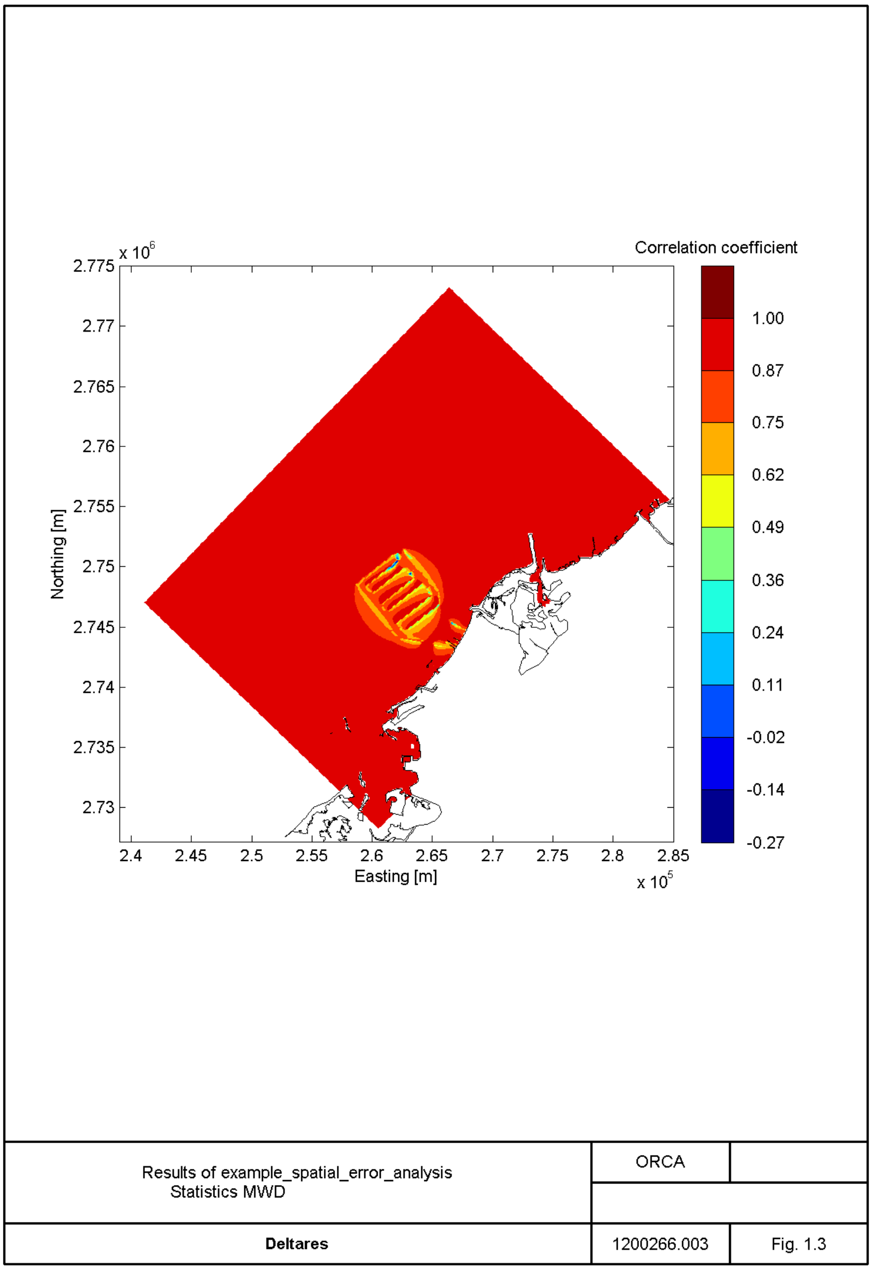


Figure 3.2 Correlation coefficient of two circular variables of the 'map' type.

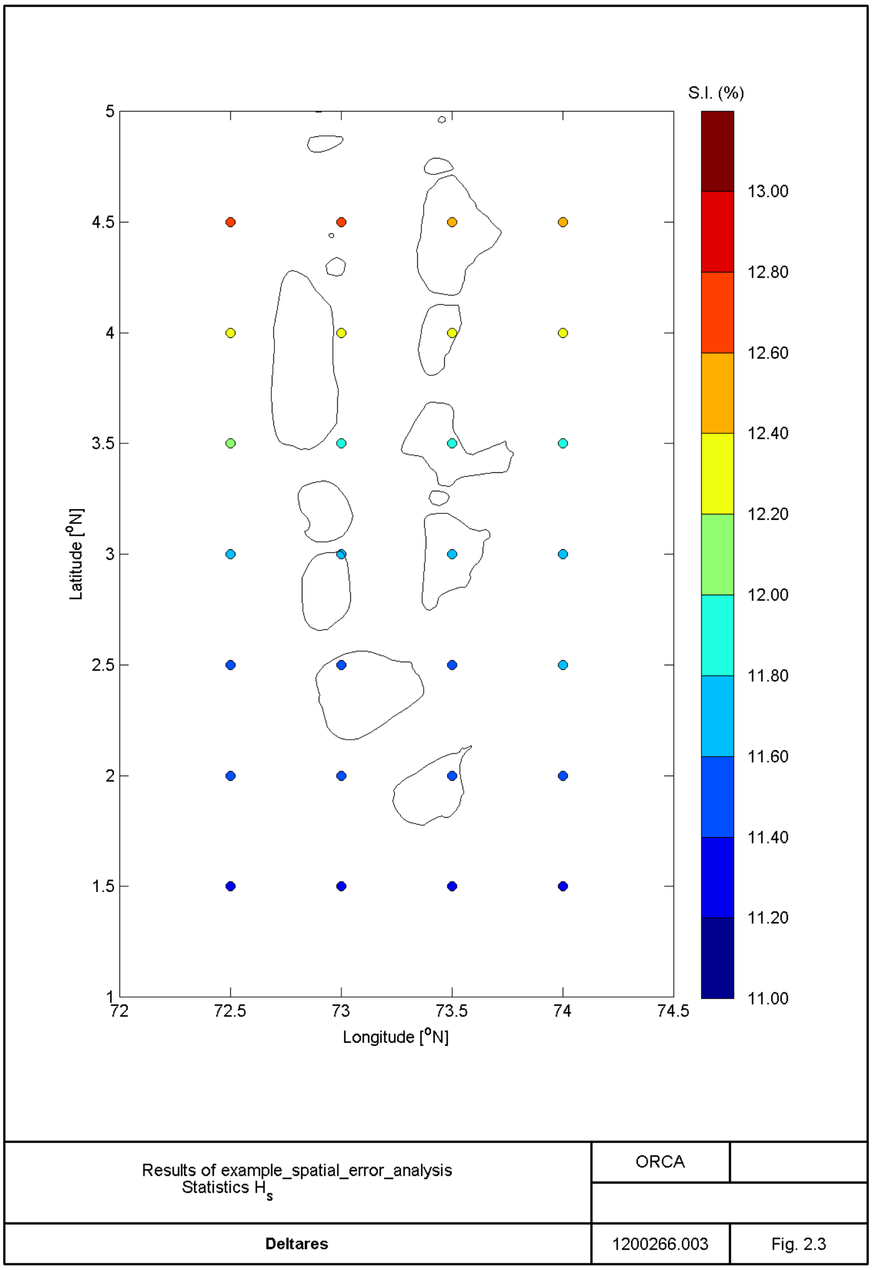


Figure 3.3 *Scatter index of two linear variables of the 'timeseries' type.*

In summary, after filling in the <example_spatial_error_analysis> script, three structures are created:

1. s1
2. s2
3. uData

The structures s1 and s2 are the data structures, containing the two data fields to compare with each other with the statistical analysis. They can be of the type:

- 'map',

- 'meta', or
- 'timeseries'

and different type combinations are possible as indicated in Table 3.1.

The time series of the two data structures do not need to have the same times nor the data structures need to have the same locations. The data is paired for the errors analysis based on the minimal spatial distance and the time instances considered are based on the minimal time difference.

For every parameter (e.g. wave height, period, direction) to be statistically analysed, a uData structure has to be filled in.

s1	s2
map	map
meta	meta
timeseries	timeseries
timeseries	map
timeseries	meta

*Table 3.1 Possible combinations of s1 and s2 **ORCA** structures for input in <spatErrorAnalysis>.*

Depending of the data being compared (cf. Table 3.1) the OrcaData structure of type 'map' containing the error statistics is one-dimensional (1D) or two-dimensional (2D) (cf. Table 3.2).

	1D	**1D – not time dependent**	**2D**	**2D – not time dependent**
s.parameter.data	t x m	m x 1	t x m x n	m x n
s.X	m x 1	m x 1	m x n	m x n
s.Y	m x 1	m x 1	m x n	m x n
s.time	t x 1	empty	t x 1	empty

*Table 3.2 **ORCA** data structure type 'map': 1D and 2D data.*

4 Spatial extreme value analysis

4.1 Theoretical background

Extreme value analysis is used to derive return values for certain parameters (e.g. for the significant wave height at a certain location) based on some limited amount of data, using extreme value theory. Extreme value theory provides analogues of the central limit theorem for the extreme values in a sample. According to the central limit theorem, the mean of a large number of random variables, irrespective of the distribution of each variable, is distributed approximately according to a Gaussian distribution. For example, the sea surface elevation is often modelled as a sum of several individual random waves and therefore its distribution can often be assumed to be Gaussian. According to extreme value theory, the extreme values in a large sample also have an approximate distribution that is independent of the distribution of each variable.

Some fundamental conditions have to be met in an extreme value analysis. The most important of these are that the values have to be statistically independent (i.e. sufficiently far separated in time) and have to be identically distributed (i.e. each value should be from the same population, meaning for example that sea and swell waves may have to be separated in the analysis).

4.1.1 Univariate analysis

Two approaches are implemented in ***ORCA*** for the estimation of univariate extreme values:

- the Annual Maxima / Generalized Extreme Value distribution, in short the AM/GEV approach, and
- the Peaks over Threshold / Generalized Pareto Distribution, in short the POT/GPD approach.

In this study only the POT/GPD approach was considered, since it is more efficient when sample sizes are small, which is generally the case.

The extreme value theory POT/GPD approach consists of fitting the GPD to the peaks of ‘clustered’ excesses over a threshold, the excesses being the observations in a cluster (of successive exceedances) minus the threshold, and calculating return values by taking into account the rate of occurrence of clusters (see Coles, 2001). Under very general conditions this procedure ensures that the data can have only three possible, albeit approximate, distributions (the three forms of the GPD) and, moreover, that observations belonging to different peak clusters are approximately independent. In the POT method, the peak excesses over a high threshold u of a time series are assumed to occur in time according to a Poisson process with rate λ_u and to be independently distributed with a GPD, whose distribution function is given by

$$F_u(y) = \begin{cases} 1 - \left(1 + \xi \dfrac{y}{\sigma}\right)^{-1/\xi}, \text{ for } \xi \neq 0 \\ 1 - \exp\left(-\dfrac{y}{\sigma}\right), \quad \text{for } \xi = 0, \end{cases} \tag{4.1}$$

where, y>0, $\sigma > 0$ and $(1+\xi(y/\sigma)) > 0$. The two parameters of the GPD are called the scale (σ) and shape (ξ) parameters. When $\xi = 0$ the GPD is said to have a type I tail and amounts to the exponential distribution with mean σ; when $\xi > 0$ it has a type II tail and it is the Pareto distribution; and when $\xi < 0$ it has a type III tail and it is a special case of the beta distribution. If $\xi < 0$ the support of the GPD has an upper bound, $-\sigma/\xi$, which is called the *upper end-point* of the GPD and is to be thought of as the upper-limit of the excesses, the upper limit of the variable of interest being then $u - \sigma/\xi$.

One of the main applications of extreme value theory is the estimation of the *once per m year (m-yr) return value*, the value which is exceeded on average once every *m* years. The *m*-yr return value based on a POT/GPD analysis, z_m, is given by

$$z_m = \begin{cases} u + \dfrac{\sigma}{\xi}\{(\lambda_\mathrm{u}\mathrm{m})^{\xi} - 1\}, \quad \text{for } \xi \neq 0 \\ u + \sigma \ln(\lambda_\mathrm{u}\mathrm{m}), \qquad \text{for } \xi = 0. \end{cases} \tag{4.2}$$

Note that this expression is obtained from Eq. (4.1) by solving $(1 - F_u(y)) = \dfrac{1}{\lambda_u m}$ for *y* and then adding the threshold *u* to the result.

4.1.2 Spatial analysis

100-yr to 10,000-yr return value estimates are usually needed for design and safety assessments. Since the available data records used to obtain such estimates are often short (between 10 and 50 years long), the statistical uncertainties associated with such estimates are inevitably large. A rather popular way to reduce the uncertainty in the estimation of return values at several nearby locations, is to profit from the spatial correlation of the data by means of a so-called "Regional Frequency analysis" (Hosking and Wallis, 1997). According to Hosking and Wallis "Regional Frequency Analysis (RFA) resolves this problem by 'trading space for time'; data from several sites are used in estimating event frequency at any one site". RFA virtually increases the length of the available data series and as such the uncertainties involved will decrease. In RFA data are assumed to come from homogeneous regions[1] in which the shape parameter of the data is assumed to be the same at all locations, or from regions in which the shape parameter varies according to other spatially varying variables (see e.g. Den Heijer et al., 2005 and Weerts and Diermanse, 2004).

The RFA consist of analysing the data of each location separately and then smoothing the shape parameter estimates by making it equal to the average of all the estimates or by using the estimates to fit a general relationship with respect to the physical phenomena which are assumed to influence the shape parameter. For instance, as in the presented example, to

1. *Homogeneous regions can be defined by means of objective and subjective techniques (see Hosking and Wallis, 1997).*

define the shape parameter of wave conditions at finite depth regions as a linear function of the water depth.

Another possible approach to account for spatial correlation of extremes is the one described in De Haan and Pereira (2006). Such approach, although considered, was not implemented because it is rather complex, involving a rather extensive analysis of correlations of the extremes at different locations, and has several parallels with bi/multi-variate extreme value analysis which is already available in ***ORCA***.

4.2 Guidelines on how to perform a spatial extreme value analysis using *ORCA*

An exemplary ***ORCA*** script that could be easily adjusted to perform a spatial extreme value analysis is:

`w:\budata\hyedata\morelis\matlabTools\orca\toolbox\Extreme\scripts\example_spatial_extreme_value_analysis.m`

Running this script will result in a spatial extreme value analysis. All necessary input from the user is defined in this script. The steps of the analysis are explained below, on the basis of the MATLAB script. In each step, all applications and engines directly used are indicated below the text box.

```
% SPATIAL_EXTREME_VALUE_ANALYSIS - Batch script to perform spatial extreme value analysis
%
% ------------------------------------------------------------
% Copyright (c) Deltares 2009 FOR INTERNAL USE ONLY
% Version:      2st Beta Version 0.0, <2009 December 31th>
% ------------------------------------------------------------
%
% Syntax:   example_spatial_extreme_value_analysis
%
% This script represents an example for a spatial extreme value analysis.
% Please apply the order of functions given below for analysis.
%
% In this example 3-hourly timeseries of North Sea mean wave period measurements
% are analysed and a regional frequency analysis (RFA) used to reduce the
% uncertainty in estimates. RFA, commonly know as "trading space for time",
% is applied here by making the shape parameter of the data dependent on
% the depth. The script can be easily adjusted to make the parameter dependent of
% other variables


%0. Authors:
%   12/2009: S. Caires, creation

% ------------------------------------------------------------------------
close all;
clear all;
% ------------------------------------------------------------------------
% ORCA settings
% ------------------------------------------------------------------------
if isempty(which('qpread.m'))
    wlsettings;
end
if isempty(which('createEmptyOrcaData.m'))
    addpath W:\budata\hyedata\morelis\matlabTools\;    addRMPaths;
    Orcainit;
end
% ------------------------------------------------------------------------
% Initiate random generator
% ------------------------------------------------------------------------
rand('state',0); % fix seed of the uniform random number generator.
rand('seed',0); % fix seed of the uniform random number generator.
```

Step 1   Initialization

In Step 1, the Initialization, a few settings are defined. The user can leave these as they are.

```
% SPATIAL_EXTREME_VALUE_ANALYSIS - Batch script to perform spatial extreme value analysis
%
% -------------------------------------------------------------
% Copyright (c) Deltares 2009 FOR INTERNAL USE ONLY
% Version:      2st Beta Version 0.0, <2009 December 31th>
%-------------------------------
% def. input and EVA parameters
%-------------------------------
dinput='W:\budata\hyedata\caires\ORCA\Guidelines\exampleData\Extreme\';
doutput='W:\budata\hyedata\caires\ORCA\Guidelines\exampleData\Extreme\results\';
fida=fopen([doutput 'example_spatial_extreme_value_analysis.txt'],'w');
minDurationEvent=48; %days
method='pwm';
rv_labels=[100 1000 10000]; %change here the rv that are to appear in the rv figure
R=[.3:.2:.9 1:1:10 20:10:100 200:100:1000 2000:1000:10e3]; %change here the range of rv to be plotted
dist='GPD';
cip.pc=95;   %change here the coverage of the confidence intervals
cip.nb=1000; %change here the number of bootstrap samples used in computing the c.i.
doplot=0;
```

Step 2 User defined parameters

In Step 2, the user has to define a few input parameters.

Directory and file names:

- dinput = 'W:\budata\hyedata\caires\ORCA\Guidelines\exampleData\Extreme\'
  Directory where the data (time series) are located. What the format of the time series is should be is explained in Step 4.
- doutput = ‘W:\budata\hyedata\caires\ORCA\Guidelines\exampleData\Extreme\results\'
  Directory where output is to be saved.
- fida=fopen([doutput 'example_spatial_extreme_value_analysis.txt'],'w');
  Name of file where output is to be written.

Parameters used in the determination of the POT data (for more information, see the application <find_peaks>):

- minDurationEvent = 48
  Minimum duration event. This is the minimum time distance between peaks, to ensure that selected peaks are statistically independent. Default value = 48 hours.

Parameters used in GPD fit and plots (for more information, see the application <compute_rv>):

- method = ‘pwm’
  Method used to fit the distribution to the extreme events
- dist='GPD';
  Distribution to be fitted to the extreme events.
- rv_labels = [100 1000 10000]
  Change here the return values that are to appear in the output file.
- R=[.3:.2:.9 1:1:10 20:10:100 200:100:1000 2000:1000:10e3];
  Change here the range of return values to computed.
- cip.pc = 95
  Change here the coverage of the confidence intervals. Default value = 95%.
- cip.nb = 1000

Change here the number of bootstrap samples used in computing the confidence intervals. Default value = 1000.

And a plot flag:

- doplot = 1
  Plot univariate analysis yes (1) or no (0).

```
%--------------------------------
% Extreme value analysis of each measurement timeseries
%--------------------------------
for buoy =1:9

    % Define the buoy characteristics (file name, location and threshold to be used)
    switch buoy;
        case 3;
            fname='gt3son_2008';c3=4.3; bname='SON'; lat(buoy)=53+(35+44/60)/60; lon(buoy)=06+10/60;
depth(buoy)=19;fignam='a';fx=8.66;
        case 7;
            fname='gt3eld_2008';c3=4.02;bname='ELD'; lat(buoy)=53+(16+37/60)/60; lon(buoy)=04+(39+42/60)/60;
depth(buoy)=26;fignam='b';fx=8.36;
        case 8;
            fname='gt3k13_2008';c3=3.9; bname='K13'; lat(buoy)=53+(13+04/60)/60; lon(buoy)=03+(13+13/60)/60;
depth(buoy)=30;fignam='c';fx=7.34;
        case 5;
            fname='gt3ym6_2008';c3=3.99;bname='YM6'; lat(buoy)=52+(33+00/60)/60; lon(buoy)=
04+(03+30/60)/60;depth(buoy)=21;fignam='d';fx=7.5;
        case 2;
            fname='gt3mpn_2008';c3=4.1; bname='MPN'; lat(buoy)=52+(16+26/60)/60; lon(buoy)=04+(17+46/60)/60;
depth(buoy)=18;fignam='e';fx=7;
        case 9;
            fname='gt3eur_2008';c3=3.79;bname='EUR'; lat(buoy)=51+(59+55/60)/60; lon(buoy)=
03+(16+35/60)/60;depth(buoy)=32;fignam='f';fx=7.18;
        case 6;
            fname='gt3leg_2008';c3=3.81;bname='LEG'; lat(buoy)=51+(55+33/60)/60; lon(buoy)=03+(40+11/60)/60;
depth(buoy)=21;fignam='g';fx=7.44;
        case 4;
            fname='gt3swb_2008';c3=3.97;bname='SWB'; lat(buoy)=51+(44+48/60)/60; lon(buoy)=03+(18+24/60)/60;
depth(buoy)=20;fignam='h';fx=7.1;
        case 1;
            fname='gt3scw_2008';c3=4.07;bname='SCW'; lat(buoy)=51+(23+32/60)/60; lon(buoy)=03+(02+57/60)/60;
depth(buoy)=15;fignam='i';fx=6.43;
    end;
```

Step 3 Used defined input data.

In Step 3, the name of the name of the data files and the location and identification of the time series being considered are defined. Furthermore, the thresholds to be used in each time series are also defined.

In this example 30 years of wave measurements at nine North Sea buoy locations are considered. The extreme value analyses will be carried out in the wind sea mean wave period data.

```
    % Read the buoy data and filter swell out
    data=read_gt(dinput,[fname '.vta'],'gt');
    datnums=data.time;
    vals=data.Tm_10_spectrum/100;  %consider the Tm-1,0 data
    hs=data.Hm0/100;
    vals(vals>(1.13*c3*(hs.^.5)+.15.*hs+.8))=-999; %remove the swell data
```

Step 4 Read data and filter swell out. Input/Output engine used: <read_gt>.

In Step 4, the data is read and swell wave conditions are filtered out as described in Weerts and Diermanse (2004). The filtering of the data is a particularity of this example and not generally applicable. Depending on the data being analysed, the user should decide and implement any filtering deemed necessary.

```
    % GPD fit using the chosen threshold
    fthresh=fx;
    [peak,idx2,lambda,lambda_v] = find_peaks(datnums,vals,fthresh,minDurationEvent,0);  % find peaks
    [rv(buoy)] = compute_rv(peak,lambda,fthresh,dist,method,R,cip,rv_labels,fida);      %compute return values
    rv(buoy).location.coordinates=[lon(buoy) lat(buoy) depth(buoy)];
    rv(buoy).name=bname;
    %--------------------------------

end
```

Step 5 Determination of the POT and computation of the return value estimates. Applications used: <find_peaks.m> and <compute_rv>.

In Step 5, the peaks that are higher than the threshold and that are statistically independent (defined by ‘minDurationEvent’) are determined in <find_peaks.m>. Furthermore, the GPD model parameters and the return value estimates are computed in <compute_rv>.

```
%--------------------------------
% Regional frequency analysis (RFA)
%--------------------------------

% smooth the shape parameter estimates using the linear relation between
% the estimates and the depth
for i=1:length(rv);     xi(i)=rv(i).model.par(1); end;
p=polyfit(depth,xi,1);
xi_rfa=p(1)*depth+p(2);

%do GPD fits using the smoothed shape parameter estimates
for i=1:length(rv)
    x0=rv(i).model.par(2) ; %first guess
    [Xgfx,Fmaxgfx,EXITFLAG,OUTPUT]=fminsearch('ml_gpdfx',x0,...
        optimset('disp','notify','TolX',1e-6,'MaxFunEvals',1e4),rv(i).sample,rv(i).thresh,xi_rfa(i));
    rv_rfa(i,:)=wgpdinv(1-1./(rv(i).lambda*R),-xi_rfa(i),Xgfx)'+rv(i).thresh;
end
```

Step 6 Regional frequency analysis. Engines used: <ml_gpdfx> and <wgpdinv>.

In Step 6, new shape parameter estimates are obtained by fitting a linear relation between the water depth at the buoy locations and the shape parameter estimates from each univariate analysis. These new, smoother and less uncertain shape parameter estimated are used to obtain new return value estimates from the data.

Note that the chosen physical relation to smooth the shape parameter is specific to this example and justified by the depth limit of waves in shallow regions. The physical relation to be used when adjusting this script to another application should be defined by the user, depending on the variable being considered and correlation between the shape parameter estimates and other variables being considered. Even if the user decides to just use the average of the shape parameter estimates, that should be done only if the analysis of the univariate shape parameter estimates justifies such assumption.

```
%--------------------------------
% Spatial plots with results
%--------------------------------

%--create structure with results
srv=createemptyorcadata('map');
srv.name='statistical data - results from EVA';
srv.parameter(1).name='depth';
srv.parameter(2).name='10,000-yr rv';
srv.parameter(3).name='10,000-yr rv RFA';
for i=1:length(rv)
    srv.X(i)=rv(i).location.coordinates(1);
    srv.Y(i)=rv(i).location.coordinates(2);
    srv.parameter(1).data(i)=depth(i);
    srv.parameter(2).data(i)=rv(i).parameter(1).data(end);
    srv.parameter(3).data(i)=rv_rfa(i,end);
end
srv.X=srv.X';
srv.Y=srv.Y';
srv.parameter(1).data=srv.parameter(1).data';
srv.parameter(2).data=srv.parameter(2).data';
srv.parameter(3).data=srv.parameter(3).data';

%-- User Data structure for plots
uData.plot={'depth','10,000-yr rv','10,000-yr rv RFA'};
uData.ldb=[dinput 'csm_wgs84.ldb'];
uData.doutput=doutput;
uData.xlim=[2.5 6.3];
uData.ylim=[51 54.];
uData.xlabel='Longitude [^{o}N]';
uData.ylabel='Latitude [^{o}N]';
uData.figNr={'Fig. 1','Fig. 2','Fig. 3'};
uData.filename={'Fig1_depth','Fig2_rv','Fig3_rvRFA'};
uData.cbarTitle={'depth (m)','10,000-yr rv (s)','RFA 10,000-yr rv (s)'};
uData.title=strvcat(['Results of example\_spatial\_extreme\_value\_analysis'],['']);
uData.text1='ORCA';
uData.text2='';
uData.text3='';
uData.projNr='1200266.003';

%--plot
plotStats(srv,uData);
%--------------------------------
fclose('all');
```

Step 7 Plotting of the spatial extreme value analysis results. Engines used: <createemptyorcadata>. Applications used: <plotStats>.

In Step 7, a 'map' type OrcaData structure is defined with the analysis results, plotting settings for the application <plotStats> are defined and the requested data plotted. Figures 4.1 to 4.3 show the plotted results. Plots are created already with the Deltares standard formatting so that they can be directly added to Deltares reports.

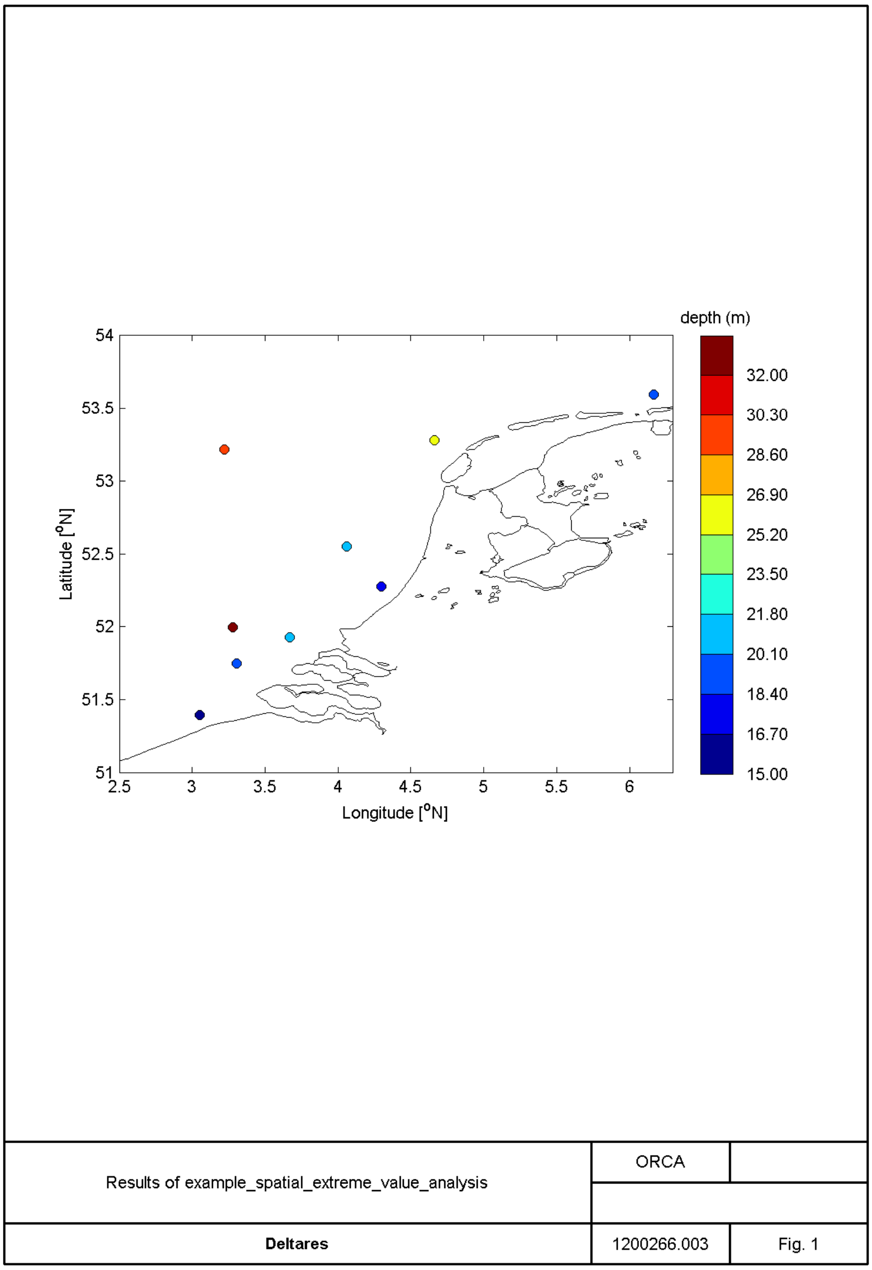
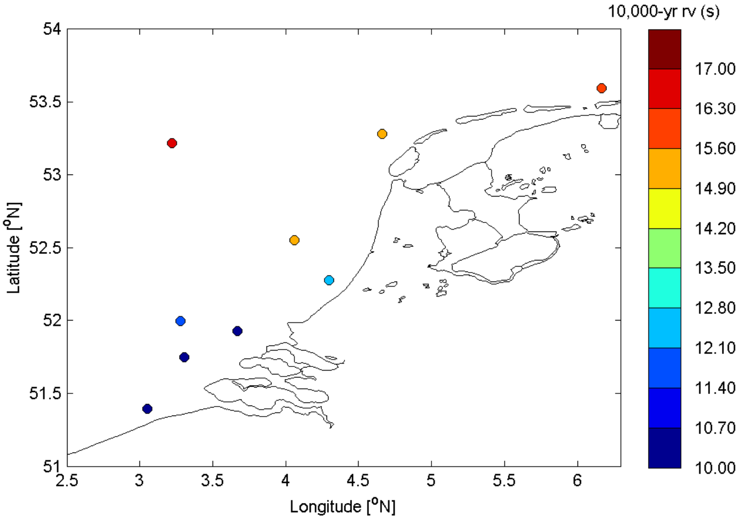
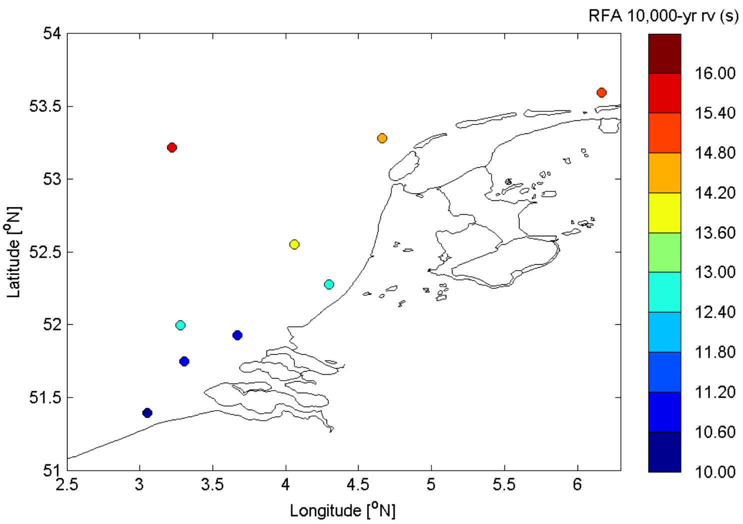


Figure 4.1 *Water depth at the considered buoy locations.*



Results of example_spatial_extreme_value_analysis	ORCA	
Deltares	1200266.003	Fig. 2

Figure 4.2 Univariate 10,0000-yr mean wave period return value estimates.



Results of example_spatial_extreme_value_analysis	ORCA	
Deltares	1200266.003	Fig. 3

Figure 4.3 *RFA 10,0000-yr mean wave period return value estimates.*

5 Conclusions

Deltares frequently carries out metocean studies. These studies involve quite a lot of statistical analyses, which are for the great part carried using and in-house MATLAB tool called ***ORCA*** (metOcean data tRansformation, Classification and Analysis). At the moment, there are five functionalities available on the ***ORCA*** tool: Data validation, Normal conditions, Extreme conditions, Sea State analysis and Persistence statistics. Due to ***ORCA***'s flexibility and easiness of use, it is extensively used in Deltares projects, which leads to frequent requests and suggestions for the extension of the tool functionalities. In particular, the need has been found to add spatial error and extreme value analyses to the tool.

The ***ORCA*** tool has been extended and now includes the possibility to carry out efficient spatial error analysis for different combinations of spatial data (sparse satellite observations, gridded data, etc). Furthermore, the particularities of linear and circular variables are properly considered when computing error statistics.

In addition, ***ORCA*** can now also be used to carry out spatial extreme value analyses, in which uncertainties in the estimates can be decreased by means of a simple technique known as Regional Frequency Analysis.

References

Coles, S., 2001: *An Introduction to the Statistical Modelling of Extreme Values.* Springer Texts in Statistics, Springer-Verlag: London.

Fisher, N.I., 1993. *Statistical analysis of circular data*. Cambridge Univ. Press, 277 pp.

Fisher, N.I. and A.J. Lee, 1983. A correlation coefficient for circular data. *Biometrika*, 70, pp. 327-32.

Heijer, F. den, F.L.M. Diermanse, and P.H.A.J.M. van Gelder, 2005: Extreme wave statistics using Regional Frequency Analysis. *Proc. ISSH – Stochastic Hydraulics 2005*, Nijmegen, The Netherlands.

Hosking, J. R. M. and J. R. Wallis, 1997: *Regional Frequency Analysis: An approach based on L-moments*. Cambridge University Press.

Mardia, K.V., 1972. *Statistics of directional data*. Academic Press, (London and New York).

Weerts, A.H. and F.L.M. Diermanse, 2004: *Golfstatistiek op relatief diep water 1979-2002* (In Dutch). WL | Delft Hydraulics Report Q3770.

A List of ORCA routines developed in this project

A number of new MATLAB routines were added to ORCA. These are listed in tables A.1 and A.2. Furthermore, in the General subdirectory the <createEmptyOrcaData>[2] engine was extended by adding the map and meta types to the OrcaData structure.

Script: *Example_spatial_error_analysis.m - Batch script to perform multiple location error analysis* *Applications:* *spatErAnalysis.m - Function to compute spatial error statistics* *errorStats.m - Function to carry out spatial error analysis* *plotStats.m - Plots spatial error statistics*

Table A.1 Routines added to the DataValidation subdirectory.

Script: *Example_spatial_extreme_value_analysis.m - Batch script to perform spatial extreme value analysis* *Engines* *ml_gpdfx.m - Computes the log-likelihood of the GPD with the shape parameter fixed*

Table A.2 Routines added to the Extreme subdirectory.

2. *createEmptyOrcaData - Creates empty OrcaData structure of given type*