Spatial Conditional Extremes MATLAB Code

Emma Ross*, Rob Shooter, Philip Jonathan

November 27, 2020

Abstract

This report describes MATLAB software which implements a Spatial Conditional Extremes (SCE) model for the estimation of extremal spatial dependence of significant wave height along satellite transects, using the conditional extremes model of Heffernan and Tawn [2004]. The software implements the methodology described in Shooter et al. [2020b], while this The proposed approach to Spatial Conditional Extremes involves (a) registering individual satellite passes onto a template transect, (b) marginal extreme value analysis at a set of locations on the template transect and transformation from physical to standard Laplace scale, (c) estimation of the conditional spatial extremes model for a set of locations on a template transect, and (d) comparison of extreme spatial dependence for different template transects. The MATLAB software which this user-guide describes covers steps c) and d); with a data-set preprocessed to address steps a) and b) provided along with the code. Preprocessed JASON satellite altimeter observations for the period 2002-2018 provide two illustrative case studies of spatial dependence of significant wave height in the North East Atlantic.

Major Updates Since Previous Release

- None: first release.

If you are using or are interested in using this software and have questions or feedback, including bug reports: please contact `e.ross@shell.com` or `philip.jonathan@shell.com`.
The MATLAB repository can be downloaded from
`https://github.com/ygraigarw/SpatialConditionalExtremesSatellite`.

*Corresponding author. Email: `e.ross@shell.com`

Contents

0 Introduction

This user guide and associated MATLAB software are provided as supplementary material to Shooter et al. 2020b, presented at the Waves SIG conference in September 2020.
The conditional extremes model of Heffernan and Tawn [2004], and extensions such as Jonathan et al. [2014], Keef et al. [2013] provide a framework to estimate multivariate extremal dependence in the presence of covariates, and hence to estimate design contours and other statistics of interest in metocean design. The approach is motivated by an asymptotic form for the limiting conditional distribution of one or more conditioned random variables given a large value of a conditioning variable. An outline of the approach is given by Jonathan et al. [2010]. Conditions for the asymptotic argument to hold have been explored by Heffernan and Resnick [2007].
List Of Symbols

- p = total number of registration locations which make up the transect of interest
- q = total number of remote locations (= set of registration locations minus the conditioning location)
- j = index on remote location
- X_j = random variable representing response data at remote location $r_R(j)$ on standard Laplace scale
- X_0 = random variable representing response data at conditioning location $r_R(0)$ on standard Laplace scale
- $r_R(0)$ = conditioning location
- $r_R(j)$ = remote location (indexed by j)
- $d_j = \text{dist}(r_R(j), r_R(0))$ = distance between remote location $r_R(j)$ and conditioning location $r_R(0)$
- $\alpha_j = \alpha(d_j)$ (and similarly β_j) = Heffernan & Tawn parameters, functions of the distance between remote location $r_R(j)$ and conditioning location $r_R(0)$
- μ_j, σ_j, δ_j = parameters characterising the residual Delta-Laplace (DL) distribution within the Heffernan & Tawn model), similarly these are functions of the distance between remote location $r_R(j)$ and the conditioning location $r_R(0)$
- ρ_1, ρ_2 = parameters characterising the residual correlation matrix, independent of the distance between remote and conditioning location
- n_d = number of nodes making up piece-wise linear form for the variation of parameters $\alpha, \beta, \mu, \sigma, \delta$ with distance

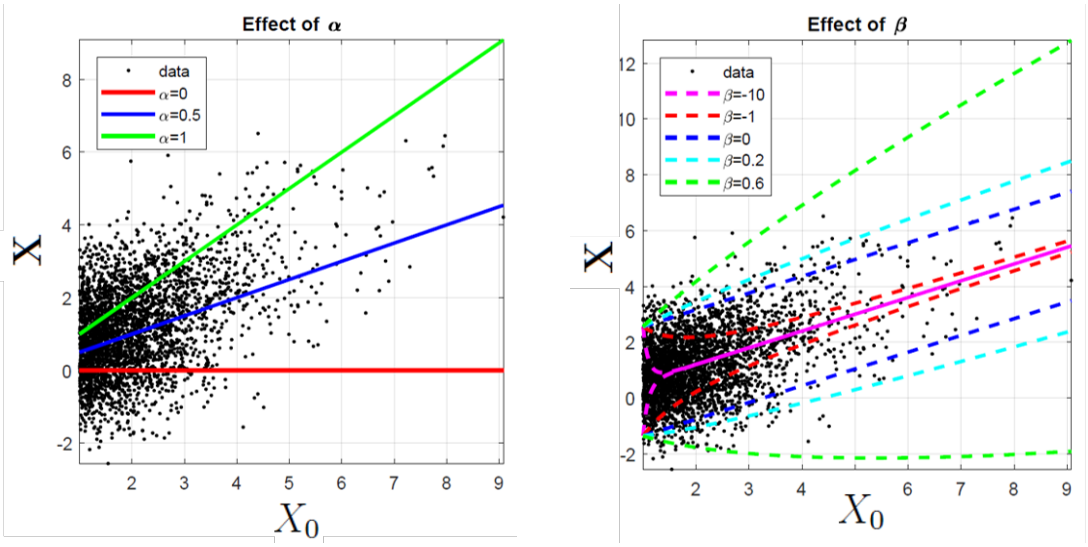
0.1 Model

Consider the vector $\mathbf{X} = (X_1, \ldots, X_q)$ corresponding to $p = q + 1$ registration locations with standard Laplace marginal distributions $X_j \sim \text{DL}(0, 2, 1)$ for $j = 0, \ldots, q$, where DL refers to the Delta Laplace distribution (a generalisation which leads to the Laplace distribution for the parameter-choice $(0, 2, 1)$). We assume, conditional on $X_0 = x_0$, for x_0, above some threshold u that

$$(X_1, \ldots, X_q)|\{X_0 = x_0\} = \boldsymbol{\alpha} x_0 + x_0^{\boldsymbol{\beta}} \mathbf{Z}$$

where $\mathbf{Z} \sim \text{DL}_q(\boldsymbol{\mu}, \boldsymbol{\sigma}^2, \boldsymbol{\delta}; \boldsymbol{\Sigma})$. $\boldsymbol{\Sigma}$ is the $q \times q$ correlation matrix for a Gaussian dependence structure between residual components. The j, j' element $\Sigma_{jj'}$ of residual correlation matrix $\boldsymbol{\Sigma}$, $j, j' = 1, 2, \ldots, q$ quantifies the dependence between SCE residuals (on standard Gaussian-scale) at registration locations $r_R(j)$ and $r_R(j')$ given conditioning on location $r_R(0)$. Elements of the correlation matrix $\boldsymbol{\Sigma}$ are characterised by parameters ρ_1 and ρ_2.
See Figure 1 for a bivariate illustration of the interpretation of two of the Heffernan and Tawn parameters (α and β). Different values for (α, β) indicate different classes of extremal dependence as follows: (α, β)

$= (1, 0)$ corresponds to Asymptotic Dependence (AD), $\alpha = 0$ to perfect independence, and intermediate values of α to Asymptotic Independence (AI). Thus, on Laplace scale, for positive dependence: (a) AD corresponds to X and X_0 for large X_0 growing at the same rate with conditional extremes slope parameter $\alpha = 1$, (b) AI corresponds to X growing more slowly than X_0, with $\alpha \in (0, 1)$, and (c) perfect independence corresponds to X not growing with X_0, and $\alpha = 0$.



(a) α (b) β

Figure 1: Heffernan & Tawn Model Bivariate Illustration

Bayesian inference is used to estimate the joint posterior distribution of the spatial conditional extremes model parameters $\Omega = \{\{\alpha_j, \beta_j, \mu_j, \sigma_j, \delta_j\}_{j=1}^{q}, \rho_1, \rho_2\}$. An Adaptive Metropolis algorithm of Roberts and Rosenthal [2009] is used (after an initial starting-parameter search and Metropolis-within-Gibbs procedure for 250 iterations) to jointly update the full set Ω of parameters. The user can adjust the number of iterations in the Adaptive Metropolis chain via input `C.nI`. Note that 50,000 iterations were used for the transect-analysis in the paper. A key calculation step in the MCMC iteration involves the negative log-likelihood function, given in Equation 6 of Shooter et al. 2020b, and implemented in the MATLAB function `SCENll.m`.

The variation of each of $\alpha, \beta, \mu, \sigma, \delta$ by distance is characterised using a piece-wise linear form consisting of n_d equally spaced nodes over appropriate physical domains for the transect of interest. Note that n_d is an input which the user can alter, via input `C.nH`. Sensitivity of inference to the choice of n_d near 10 has been explored for the transects covered in the paper and was found to be small. This means that a total of $n_d \times 5 + 2$ parameters need to be estimated.

The software provides the option to restrict the space of feasible combinations $\{\alpha_j, \beta_j\}$, $j = 1, 2, ..., q$ for each conditioning location to ensure that conditional quantiles from Asymptotic Independent models do not exceed those from Asymptotic Dependent models, as proposed by Keef et al. [2013] (see e.g. Shooter et al. 2019 for further details), via input `C.IsCQC`.

1 Data preparation

A data file called `Satellite_Preprocessed_NS_SWNE.mat` is supplied with this code for convenience and repeatability of the results presented in Shooter et al. 2020b.
The file contains H_S data (transformed to standard scale*, a mapped to a set of registration locations) measured by the JASON-1, JASON-2 and JASON-3 satellites, calibrated and quality controlled as described by Ribal and Young 2019. The original-scale H_S can be downloaded from an online portal, a link to which is provided in their paper.
We transform the original-scale H_S data to standard Laplace scale, because the Conditional Extremes model of Heffernan and Tawn [2004], the basis of the Spatial Conditional Extremes model, requires H_S data to be on standard Laplace or Gumbel scale. The following steps were taken to prepare the data for Spatial Conditional Extremes modeling:

- A discrete set of registration locations are defined over which the spacial conditional extremes model will be applied. Registration locations are chosen such that they are approximately equidistant between adjacent JASON-3 transects and approximately equally spaced in the transect direction.
- For each available satellite pass in a given time period over the region of interest, we find the nearest point on the satellite transect to each of the registration locations, and allocate the value of H_S to that registration location. If the nearest such point is > 50km away then the pass is not registered to any reference location.
- This results in a sample of H_S observations associated with each registration location. Marginal extreme value analysis is performed - fitting a generalised Pareto (GP) distribution independently for the sample data at each registration location. Using this GP fit, samples are thereby transformed (by the Probability Integral Transform) to Laplace scale, ready for input to the Heffernan & Tawn-inspired Spatial Conditional Extremes model.

Details of this data pre-processing step can be found in Section 2.2 of Shooter et al. 2020b.
The provided input data file's key field `Pk` has the structure shown in Figure 2, where

- `HsUnf` : each column corresponds to altimeter H_S measurements (on uniform scale) for a given registration location
- `HsLpl` : each column corresponds to altimeter H_S measurements (on Laplace scale) for a given registration location
- `Xi` : fitted generalised Pareto shape parameter for each registration location
- `Sgm` : fitted generalised Pareto scale parameter for each registration location
- `Psi` : threshold used for each registration location (where the generalised Pareto distribution is fitted onto to exceedances of that threshold)
- `Lct` : an array with 19 rows (one for each registration location) and 2 columns containing the longitude and latitude (respectively) for each location

For readers who wish to work with their own H_S (or other response) data; this procedure of registering their data at discrete set of locations and performing marginal extreme value analysis would need to be applied prior to using this SCE software - to obtain the Laplace-distributed data at a set of registration locations which it requires as input. For MATLAB software which performs such marginal extreme value analysis, piece-wise constant with respect to covariates (like direction, season) - the reader is directed to `https://github.com/ECSADES/ecsades-matlab`.
The user can then update the data file assigned to `X.DatFil` in line 27 of `RunSemiParametric`:
If the new input data file has been formatted in a manner similar to the file `Satellite_Preprocessed_NS_SWNE.mat` provided with this code, then no further edits to the run script are required. If, the new input data have a different format however, then the user should take care to pass the relevant fields (in the correct dimension/format) to following lines of `RunSemiParametric.m`.

```
Pk =

  struct with fields:

    HsUnf: [1325×19 double]
    HsLpl: [1325×19 double]
       Xi: [19×9 double]
      Sgm: [19×9 double]
      Psi: [19×9 double]
      Lct: [19×2 double]
```

Figure 2: Structure of key field 'Pk' in the provided (preprocessed) altimeter data file Satellite_Preprocessed_NS_SWNE.mat

```
100      %START OPTIONAL USER INPUT: INPUT PEAKS DATA >>
101 -    load(X.DatFil,'Pk'); %load data file
102 -    Dat.Unf=Pk.HsUnf; %peaks-data on uniform marginal scale
103 -    Dat.Lct=Pk.Lct; %Longitudes and latitudes of locations for analysis
104 -    clear Pk;
105      % << END OPTIONAL USER INPUT
```

Figure 3: Running the SCE code with an alternative data file - formatting inputs

2 Running the Code

`RunSemiParametric.m` is the main run/control file for SCE analysis. The `Code` folder then contains the key functions being called 'under the hood': including the likelihood function and methods to invoke the MCMC inference procedure. Key user-inputs are marked at the top of the file between `START USER INPUT` and `END USER INPUT`; though there are some further *optional* input fields a little further into the script in the 'Format Peaks Data' section.
From a clone of the GitHub folder, simply running this script unchanged will perform a first analysis - setting up the required sub-directories to store results and output figures. A new sub-directory will be created to store results every time you run a new analysis - by renaming the analysis sub-directory to reflect the key input-settings you have chosen.
For example, your first time running the code might create an analysis sub-directory called `Nep80nH10nA1Cl1Cqc1Rpt1`, indicating that:

- `Nep80` : non-exceedance probability (`X.Nep`) of 0.8 used
- `nH10` : number of distance nodes (`C.nH`) is 10
- `nA1` : number of directional nodes (`C.nA`) is 1
- `Cl1` : conditioning location index (`X.CndLct`) is 1, i.e. the first (most north-westerly) location in the transect
- `Cqc1` : conditional quantile constraints (`C.IsCQC`) are on
- `Rpt1` : analysis replication ID (`C.Rpl`) is 1

Note that to run repeats of a certain analysis set-up, you must update `C.Rpl` to a new number to avoid writing over previous runs with otherwise identical settings. Though the analysis reported in the paper is stationary (with respect to covariates like direction), the code can perform non-stationary analysis by setting the number of directional nodes `C.nA` to a number greater than 1. For more details on the non-stationary application of Spatial Conditional Extremes analysis, see Section 2.3 of Shooter et al. [2020a].
There is one preparatory step of note however: if you choose to run the code with Conditional Quantile Constraints switched on (so you have `C.IsCQC=1`), then make sure to check that you have a starting solution `C.Prm0` which satisfies these constraints. If your starting solution is infeasible, the MCMC procedure will not start. To check your starting solution is ok: set `StartingSolutionOk = false` on your first run of the script. This will switch on the 'review starting solution' section of the run-script which provides a useful illustration of your starting solution against the constraints. If you find that your starting solution is outside (upper-right of) the curved constraints as illustrated in Figure 4, then you can use this image to help toggle the starting values until feasible (to the lower-left of the constraints). Upon finding a feasible starting solution (as illustrated in Figure 5), revert `StartingSolutionOk` to `true` to start the analysis (and MCMC) proper. Note that the code will not continue to MCMC until you have confirmed your starting solution is ok by updating this input field.
The key input arguments which the user may want to play around with are listed in Figure 6. At least 5,000 MCMC iterations (`C.nI`) are recommended, this results in a run time of approximately 1 hour. The user should use the diagnostic output plots (see next Section) to discern whether the chain has converged or requires a higher number of iterations.

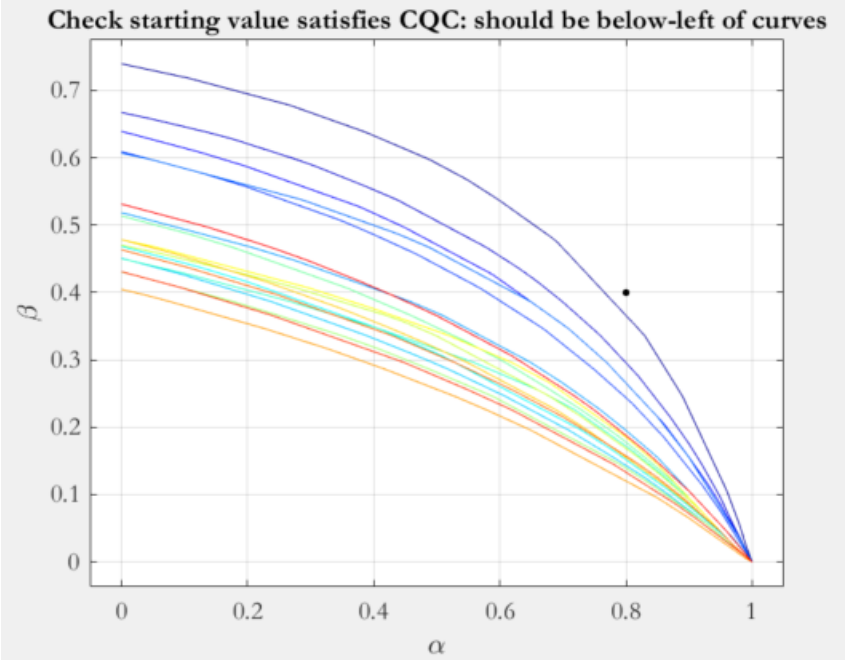


Figure 4: Figure produced when running code on 'check CQC constraints' mode - in this case the starting solution (black point) is infeasible (not within curved-boundaries associated with the CQC boundary for each remote location)

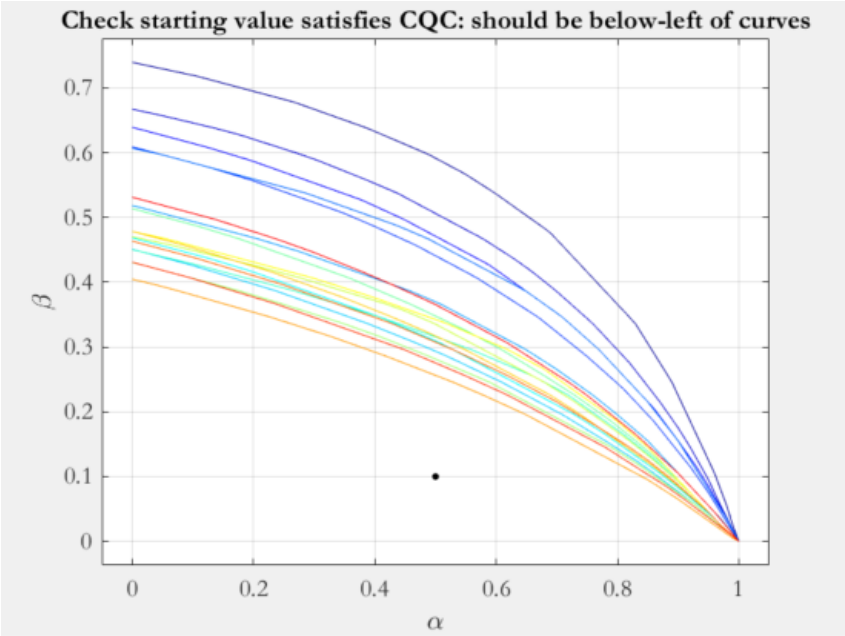


Figure 5: Figure produced when running code on 'check CQC constraints' mode - in this case the starting solution (black point) is feasible (within curved-boundaries associated with the CQC boundary for each remote location)

```
% X: peaks data-related
X.Nep=0.8; %non-exceedance probability for CSE model
X.CndLct=1; %index of conditioning location
% C: spatial conditional extremes model related
C.IsCQC=1; %=1 == Impose conditional quantile constraints; =0 == No constraints
C.StrAgn=0; %=1 == Start the analysis from scratch; =0 == Do not restart the analysis, i.e. use previously-saved progress
C.nI=5000; %number of MCMC iterations
C.Rpl=1; %replicate number (to distinguish replicate analyses)
C.nH=10; %number of distance nodes
C.nA=1; %number of directional nodes
C.HMxm=12; %maximum interlocation distance; in units of EarthRadius*pi/180; one unit is 111.2km
```

Figure 6: Key user-input settings captured in the sub-directory name for each analysis run

3 Interpreting the Output

Running the code produces two key data files as output:

- `<name_of_analysis_case>-CQC_Bnd.mat`
  - `CQCNepLst` : the scalar non-exceedances probability = `X.Nep`
  - `CQCBndL` : a cell array of length q (= the number of remote locations), containing 101×2 arrays defining the CQC boundary for each remote location, where the columns correspond to α and β respectively.
- `<name_of_analysis_case>-CQC_Prm.mat`
  - `C` : (key output) structure containing all data related to the analysis run. Key fields include:
    - `Alp` : `nI` $\times q$ (number of MCMC iterations by number of remote locations) array, containing full MCMC chain of fitted α parameters.
    - `Bet` : `nI` $\times q$ (number of MCMC iterations by number of remote locations) array, containing full MCMC chain of fitted β parameters.
    - `AccRat` : `nI` $\times$ (the total number of parameters being fitted), containing the acceptance rate for each iteration of the MCMC chain. The columns correspond to parameters in the same order as they appear in the MATLAB run script where `C.Prm0` is defined.
    - `Ngt` : `nI` $\times$ (the total number of parameters being fitted), adaptive MCMC proposal standard deviation for each iteration of the MCMC chain. The columns correspond to parameters in the same order as they appear in the MATLAB run script where `C.Prm0` is defined.
    - `Nll` : `nI` $\times$ 1 (number of MCMC iterations by 1) array, containing negative log-likelihood on each iteration of the MCMC chain.
    - `Prm` : `nI` $\times$ (the total number of parameters being fitted), containing the parameter estimates on each iteration of the MCMC chain. The columns correspond to parameters in the same order as they appear in the MATLAB run script where `C.Prm0` is defined.
  - `Nep` : non-exceedance probability for SCE model `X.Nep`
  - `mI` : the integer number of MCMC iterations `C.nI`
  - `nI` : the integer number of MCMC iterations `C.nI`
  - `NmbToPlt` : the integer number of MCMC iterations to plot, i.e. the last `NmbToPlt` iterations will be displayed in trace plots
  - `DatNam` : the string identifier for the analysis/run, i.e. `<name_of_analysis_case>`

In addition, some default diagnostic figures are generated automatically, examples of which are described below. The output data files can be used to create additional plots such as those in the full paper.

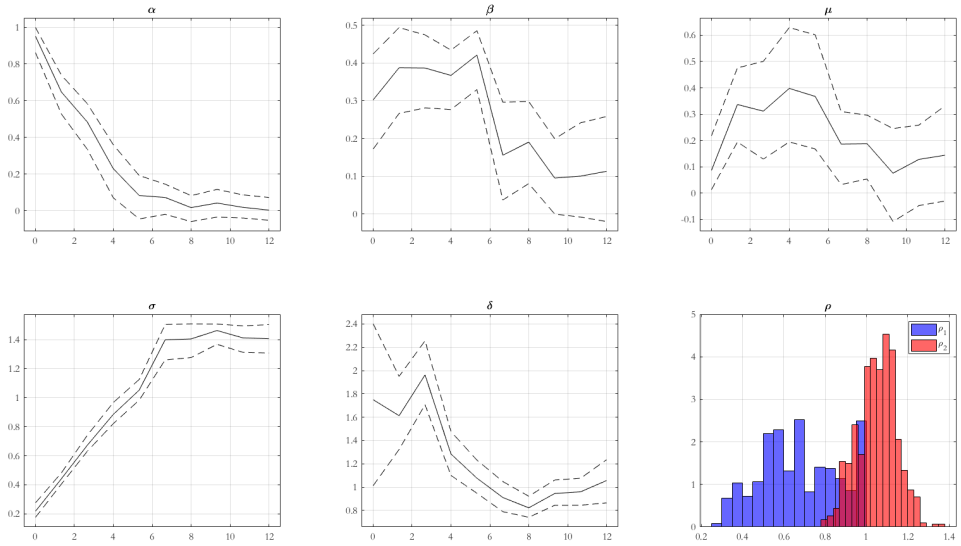


Figure 7: Posterior mean (solid line) parameter values plotted against distance (or lag) from conditioning location. Dashed lines show 95% posterior predictive interval. This plot can be used to observe the rate of decay of dependence as the distance between the (multiple) conditioned and single conditioning location increases. Since parameters ρ do not vary by distance from conditioning location, their posterior distributions are shown as histograms.

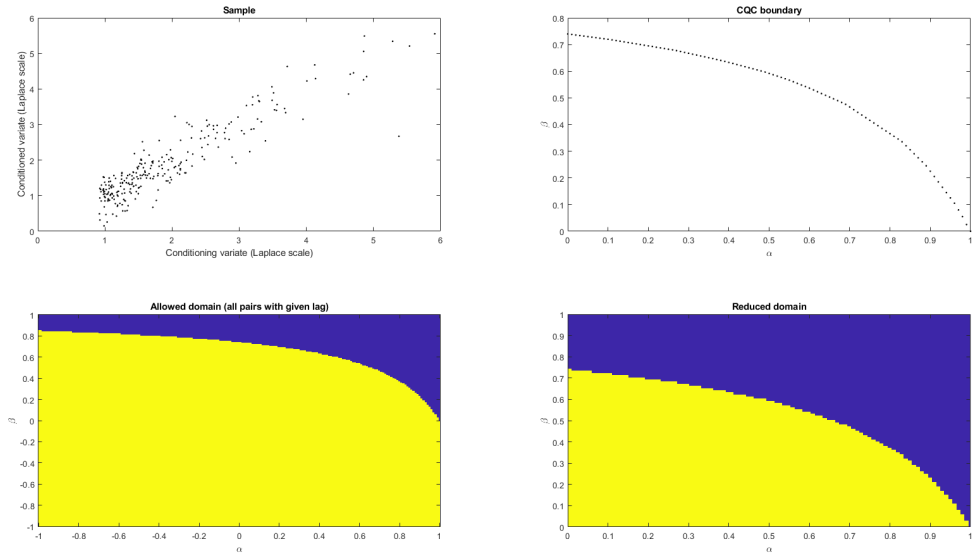


Figure 8: Example illustration (for a given remote location - here the lag 1 location, the closest to the conditioning location) of Conditional Quantile Constraint (CQC) "Keef" boundary. A different CQC boundary (and hence different version of this plot) exists for each [remote location, conditioning location] pair. Upper left: scatter plot of the Laplace-scale H_S observations at the remote (lag 1) location, plotted against Laplace-scale H_S at the conditioning location. Upper right: the CQC boundary for this (lag 1) remote location. Lower left: feasible domain (in yellow) of α, β, as defined by Keef constraints. Lower right: the 'allowed domain', equivalent to the feasible domain but α restricted to be non-negative, as we do not allow for negative dependence in this application. Feasible α, β combinations fall in the yellow region, and infeasible in the blue region.

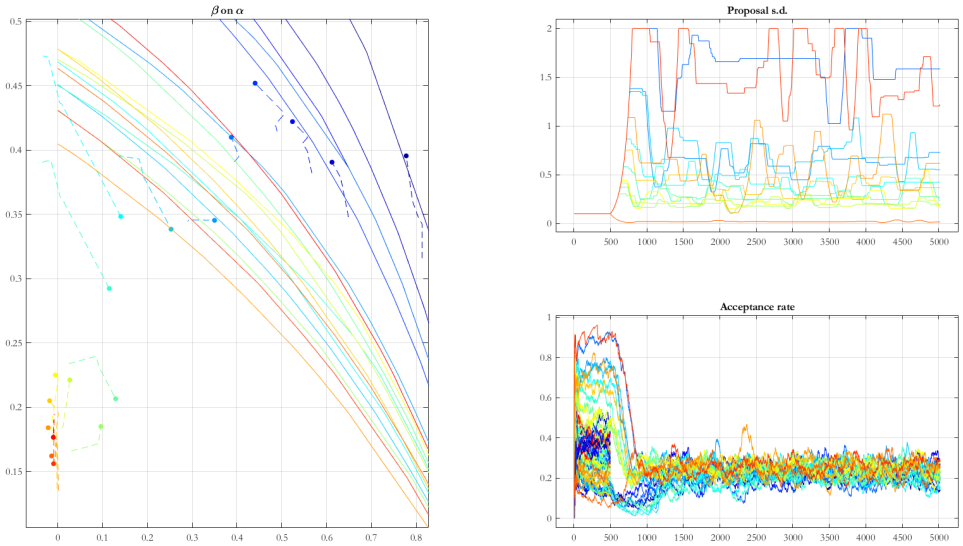


Figure 9: Illustration of CQC boundaries and MCMC trajectories for 10 iterations. Each colour corresponds to a [remote location, conditioning location] pair, and its associated CQC boundary. Coloured dots mark the most recent (last) iteration, dashed lines show path to reaching that last iteration. α on x-axis, β on y-axis. Top right: trace plot of proposal standard deviations, as move through MCMC iterations. Bottom right: trace plot of acceptance rate, as move through MCMC iterations. The user should be aiming for an acceptance rate around 23%, and a proposal standard deviation which has steadied out since the first few iterations (isn't still jumping around too much).

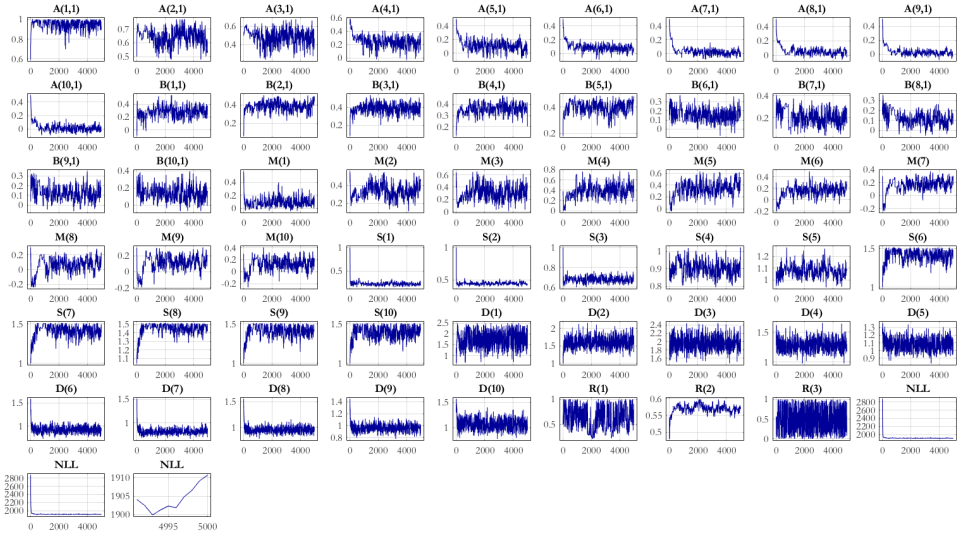


Figure 10: MCMC trace plots for all model parameters, and negative log posterior likelihood (NLL). Last 1000 iterations only are shown. When a sufficient number of MCMC iterations have been used, the user should be looking for a 'white noise'-type signal for all parameters and the NLL - convergence (with noise) around some static value i.e. not still heading up or down, and avoiding a 'Manhattan skyline' trace which is indicative of a low acceptance rate for new proposals (getting stuck at the same value for consecutive iterations).

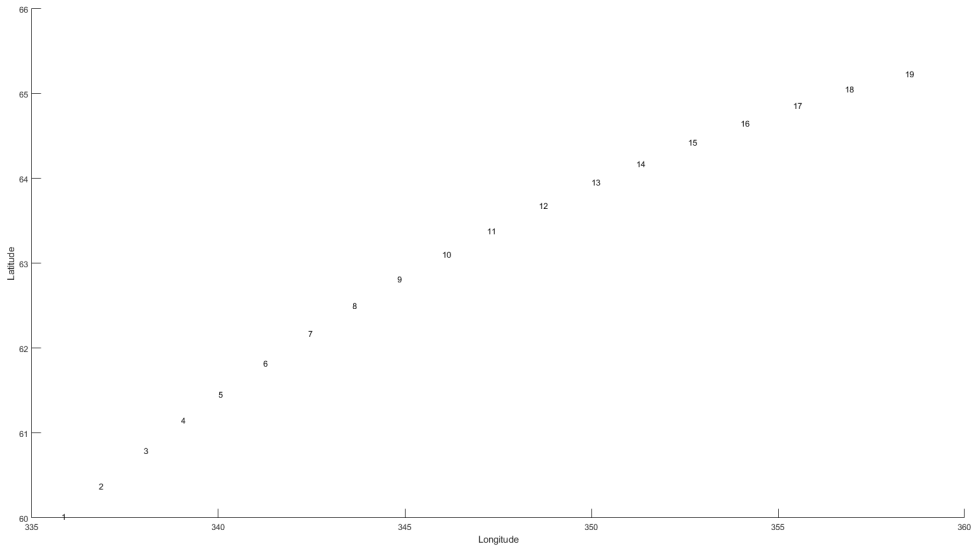


Figure 11: Illustration of the transect registration locations, with indices j for reference. This plot can be useful when choosing which location to use as your conditioning location with input setting X.CndLct.

4 Supporting Functions

- `MCMC.m` : Markov Chain Monte Carlo (MCMC) algorithm
- `SCENll.m` : SCE negative log-likelihood function calculation
- `CQC.m` : Calculates conditional quantile constraint boundaries and confirms compliance
- `ABMSDR.m` : Calculate α, β, μ, σ, δ and ρ for parameters P and distances H

References

J. E. Heffernan and S. I. Resnick. Limit laws for random vectors with an extreme component. *Ann. Appl. Probab.*, 17:537–571, 2007.

J. E. Heffernan and J. A. Tawn. A conditional approach for multivariate extreme values. *J. R. Statist. Soc. B*, 66:497–546, 2004.

P. Jonathan, J. Flynn, and K. C. Ewans. Joint modelling of wave spectral parameters for extreme sea states. *Ocean Eng.*, 37:1070–1080, 2010.

P. Jonathan, K. C. Ewans, and D. Randell. Non-stationary conditional extremes of northern North Sea storm characteristics. *Environmetrics*, 25:172–188, 2014.

C. Keef, I. Papastathopoulos, and J. A. Tawn. Estimation of the conditional distribution of a vector variable given that one of its components is large: additional constraints for the Heffernan and Tawn model. *J. Mult. Anal.*, 115:396–404, 2013.

A. Ribal and I. R. Young. 33 years of globally calibrated wave height and wind speed data based on altimeter observations. *Sci. Data*, 6:77, 2019.

G. O. Roberts and J. S. Rosenthal. Examples of adaptive MCMC. *J. Comp. Graph. Stat.*, 18:349–367, 2009.

R. Shooter, E. Ross, J. A. Tawn, and P. Jonathan. On spatial conditional extremes for ocean storm severity. *Environmetrics*, 30:e2562, 2019.

R Shooter, J A Tawn, E Ross, and P Jonathan. Basin-wide spatial conditional extremes for severe ocean storms. *Submitted to Extremes, May 2020*, 2020a.

R Shooter, E Ross A Ribal I R Young, and P Jonathan. Spatial dependence of extreme seas in the North East Atlantic from satellite altimeter measurements. *Submitted to Environmetrics, Jul 2020*, 2020b.