# the **SeaLev** package
## user guide

Yves Deville

March 10, 2016, SeaLev version 0.4.2

# Contents

**Abstract**

The `SeaLev` package has been specified by IRSN. The main goal is to implement the convolution-based method called *Joint Probability Method* as used in extreme Sea Level Analysis. The package allows approximate inference based on the "delta method".

# Chapter 1

# Introduction

This document is based on **SeaLev 0.4.2** using R version 3.2.3 (2015-12-10). This is a DRAFT version. The functions calls may change in future versions. Warnings are generally not shown in the code chunks of this document.

## 1.1 Outlook

### 1.1.1 Goals

**SeaLev** is an R package [R D10] dedicated to the probability analysis of high Sea Levels using the *Convolution Method* or *Joint Probability Method* (JPM) as described in the original articles of Pugh and Vassie [Pug79] and [Pug80]. More information on the context can be found in the books by David Pugh [Pug87, chap. 8], [Pug04, chap. 6], or that (in french) of Bernard Simon [Sim07, chap. VIII].

The method concerns a *still water* sea level, and relies on a decomposition of it as the sum of a *tide* part, and a *non-tidal* part – or *surge* part. The two components are considered as independent random variables, and the probability distribution of the sea level can then be computed by convolution. Note that the surge part can not be observed by itself and is obtained as the difference between the observed level and its tidal prediction. The surge is sometimes called the *tide residual*.

The assumption of independence between the tide and the surge is best supported when the modelled sea level is recorded at or near high tide [Col05]. The *skew surge* is computed as minus the difference of the predicted (astronomical) tide and the nearest experimented high water. Modelling high tide sea levels and skew surges (rather than, say, hourly levels and surges) is a valuable option as far as the interest is on extreme *high* sea levels. The time between two successive high tides is about 12 hours and 26 minutes for semi-diurnal tides, corresponding to a sampling rate of about 705.8 high tides by year. The method of convolution applied with Skew Surges is sometimes called the *Skew Surge Joint Probability Method* (SSJPM).

### 1.1.2 Limitations

The hypotheses retained for the convolution method are quite strong. Besides the independence it is assumed that no long-term trend exist in the tide or in the surge process, and therefore a possible change in climate or sea level can not be taken into account.

## 1.2 Context

### 1.2.1 Notations

Let $Z$ denote the sea level random variable, and let $X$ and $Y$ be the tide and the non-tidal or surge part

$$\underset{\text{sea level}}{Z} = \underset{\text{tide}}{X} + \underset{\text{surge}}{Y}$$

We will use the notation $f_Z(z)$, $F_Z(z)$ and $S_Z(z) = 1 - F_Z(z)$ to represent density, distribution and survival functions of $Z$, and similar notations for another random variable the symbol of which will appear as a subscript. Subscripting with a random variable symbol will also be used for parameters as in $\mu_Y$ or $\sigma_Y$.

Recall that when $X$ and $Y$ are independent and of continuous type with densities $f_X(x)$ and $f_Y(y)$, the random variable $Z$ has a density given by the convolution formula

$$f_Z(z) = \int_{-\infty}^{+\infty} f_X(x)\, f_Y(z-x)\, \mathrm{d}x \tag{1.1}$$

A similar relation can be given using the survival functions

$$S_Z(z) = \int_{-\infty}^{+\infty} f_X(x)\, S_Y(z-x)\, \mathrm{d}x \tag{1.2}$$

In both integrals, the variable of integration could be chosen to be a surge $y$, replacing then $x$ by $z - y$.

### 1.2.2 Tidal $X$

The distribution of the tidal component $X$ is assumed to be of continuous type with a bounded support

$$x_{\min} \leqslant X \leqslant x_{\max}$$

Therefore the bounds of the integrals in (1.1) and (1.2) can be replaced by $x_{\min}$ and $x_{\max}$. The density $f_X(x)$ is assumed to be available in a general non-parametric form. It is assumed in the current version of **SeaLev** that the density $f_X(x)$ is continuous and takes the value 0 at the end-points of $X$

$$f_X(x_{\min}) = 0 \qquad f_X(x_{\max}) = 0. \tag{1.3}$$

This condition can be useful in the numerical convolution.

**Remarks**

• In practice, $x_{\max}$ will be the *Highest Astronomical Tide* (HAT) which necessarily occurs at high tide. The minimum $x_{\min}$ will be for high tides.

• For semi-diurnal tides, the distribution $f_X(x)$ of astronomical high tides will often be bi-modal.

• The conditions (1.3) are not always fulfilled by an arbitrary periodic oscillation with range $(x_{\min}, x_{\max})$, see the example in [Pug80], p. 975.

• In practice, the distribution of the tide must be computed from "observations" $X_t$, that is *predictions* arising from a harmonic analysis of the sea level. The CRAN package **TideHarmonics** [Ste16] can be used to perform such an analysis.

| distribution | code | par. names | package |
|---|---|---|---|
| exponential | `exp` | `rate` | **stats** |
| generalised Pareto | `GPD` | `scale`, `shape` | **Renext** |
| | `gpd` | `scale`, `shape` | **evd** |
| gamma | `gamma` | `shape`, `scale` | **stats** |
| Weibull | `weibull` | `shape`, `scale` | **stats** |
| mixture of two exponentials | `mixexp2` | `prob1`, `rate1`, `delta` | **Renext** |

Table 1.1: Distributions for surge POT. Some other distributions may require the use of a specific (CRAN) package.

### 1.2.3 Non-tidal component $Y$ (surge)

For the non-tidal component or surge component $Y$, the required information will be the distribution of $Y$ conditional on $Y > u$, where $u$ is the threshold. Such a distribution typically results from a *Peak Over Threshold* (POT) modelling. The distribution will often be given for the excess $Y^\star = Y - u$ rather than for $Y$. It will be given as a specific element within a list of supported distributions, among which we find the Generalised Pareto used in traditional POT.

As in standard POT analysis, the distribution of the excess must come with a *rate* related to an underlying Homogeneous Poisson Process of threshold exceedances. We assume that *the rate $\lambda$ is expressed as a number of threshold exceedances by year.*

A desirable mathematical property for the density of $Y$ is continuity. For physical reasons, the density should be bounded near the threshold. This should put offside Weibull or gamma distributions with increasing hazards, that is with $0 <$ `shape` $< 1$ in both cases. However using such distributions is possible in **SeaLev**.

**SeaLev** contains a description of some "special" distributions for POT. See table 1.1. It is also possible to use other distributions for excess or even non-POT and hence an unconditional distribution for the surge. In this case, the distribution is for the the variable $Y$ and not for any kind of excess. See 2.3.3 page 12 for an example using the Generalised Extreme Value (GEV) distribution. GPD and GEV distributions are provided in suitable form by the **evd** package [Ste02], available from the CRAN, or by the **Renext** package.

**Remark**. The distributions provided by **evd** have suffix `"gpd"` and `"gev"`, while those by **Renext** have suffix `"GPD"` and `"GEV"`. In practice, using either package will make no difference here, because the difference between the two packages are in the treatment of invalid parameters, e.g. a negative scale. This would matter in unconstrained optimisation.

### 1.2.4 Probability versus theoretical frequency

The tidal part $X$ has a deterministic nature and is related to a deterministic cyclic process $X_t$ [Dix94]. Yet we may speak of "probability distribution" for observations $X_t$ provided that some points are well understood.

Let $X_t$ be the series of computed tidal sea levels at successive high tide times $t = 1, 2, \ldots$. This is a cyclic deterministic process with a fairly large period[1]. The density $f_X(x)$ is such that the probability that $X$ falls in some given interval should be equal to the correspondent frequency on large periods. This can be expressed as an *ergodicity condition*: for any arbitrary

[1]The nodal cycle, about 18.61 years

"test" function $\phi(x)$ defined on the support $(x_{\min}, x_{\max})$, the approximation

$$\frac{1}{T}\sum_{t=1}^{T}\phi(X_t) \approx \int \phi(x) f_X(x)\,\mathrm{d}x \tag{1.4}$$

must hold for large $T$ (relative to the period). The independence condition between $X$ and $Y$ can similarly be expressed using cross-frequencies over a large number of periods or equivalently using an ergodicity condition involving an arbitrary two-variables test function $\phi(x, y)$

$$\frac{1}{T}\sum_{t=1}^{T}\phi(X_t, Y_t) \approx \iint \phi(x, y) f_X(x) f_Y(y)\,\mathrm{d}x\,\mathrm{d}y \tag{1.5}$$

for large $T$.

## 1.3 Convolution and POT

The independence condition (1.5) implies that the distribution of $X$ conditional on $Y > u$ is identical to the unconditional distribution of $X$. Hence if a POT analysis is used for $Y$, we still can use the convolution distribution for $Z$ with some restrictions. Firstly, the return levels for $Z$ are computed by considering that $Z$ values are sampled at a rate $\lambda$ with $\lambda < 705.8$. Secondly, since the distribution of $Z$ is then only partially known, the formula (1.2) can only be used for $z > x_{\max} + u$. Actually, since $X \leqslant x_{\max}$ with probability one, we have for $z > x_{\max} + u$

$$S_Z(z) = \Pr(Z > z) = \Pr\left[Z > z \mid Y > u\right]$$

and the distribution of $Y$ conditional on $Y > u$ can be used in place of the unconditional one.

**Remarks**

• In the POT context, the needed independence between $X$ and $Y$ turns into a weaker condition of independence conditional on $Y > u$. There could be some dependence between $X$ and $Y$, but this must be limited to small $Y$.

• The GPD distribution of $Y$ can have a finite upper end-point (with negative shape parameter $\xi_Y < 0$).

## 1.4 Return periods and return levels

The rate $\lambda$ is used to compute the return period $T_Z(z)$ of a given level $z$ according to

$$T_Z(z) = \frac{1}{\lambda \times S_Z(z)}$$

This formula will be used for $z > x_{\max} + u$. An approximated value of $S_Z(z)$ will be computed with the convolution formula.

In most cases, the distribution of $Y$ is estimated within a parametric family. In the POT context, the rate $\lambda$ will be replaced by an estimation $\widehat{\lambda}$. When instead all high tide measurements are used, the rate must be considered as certain with value $\lambda = 705.8\,\mathrm{year}^{-1}$.

Note that when $Y$ has a finite upper end-point $y_{\max}$, the sea level $Z$ also has finite upper end-point $z_{\max}$. This finite level corresponds to an infinite return period $T_Z(z_{\max}) = +\infty$.

We may alternatively be concerned with the return level $z(T)$ corresponding to a given return period, e.g. $T = 1000$ years. This level is obtained as the solution $z$ of

$$S_Z(z) = \frac{1}{\lambda T} \tag{1.6}$$

The return level $z(T)$ can be expressed as

$$z(T) = q_Z(p), \qquad p := 1 - \frac{1}{\lambda T} \tag{1.7}$$

where $q_Z(p)$ is the standard quantile function defined for $0 < p < 1$.

## 1.5 Inference based on the delta-method

### 1.5.1 Principle

The distribution of the tidal part $X$ is assumed to be perfectly known. The (conditional) distribution of $Y$ depends on a parameter $\boldsymbol{\theta}_Y$ of length $p_Y$. The parameter vector is in the general case

$$\boldsymbol{\theta} = [\lambda, \boldsymbol{\theta}_Y^\top]^\top$$

The threshold $u$ for $Y$ is considered as fixed. For instance, when the GPD is used for $Y$ in a POT analysis, the two estimated parameters are the scale and shape $\boldsymbol{\theta}_Y = [\sigma_Y, \xi_Y]^\top$, while the location parameter $\mu_Y$ coincides with the threshold $u$, hence is fixed.

The *delta method* is a general framework for approximated inference, see [Col01]. It can be used in the convolution context, where the uncertainty on the distribution of $Y$ propagates on the distribution of $Z$. The survival $S_Z(z)$ depends on $\boldsymbol{\theta}_Y$ according to

$$S_Z(z; \boldsymbol{\theta}_Y) = \int_{x_{\min}}^{x_{\max}} f_X(x)\, S_Y(z - x; \boldsymbol{\theta}_Y)\, \mathrm{d}x$$

Under some mild assumptions, the derivative of the survival $S_Z(z; \boldsymbol{\theta}_Y)$ with respect to the parameter $\boldsymbol{\theta}_Y$ can be obtained by differentiating under the integral sign

$$\frac{\partial}{\partial \boldsymbol{\theta}_Y} S_Z(z; \boldsymbol{\theta}_Y) = \int_{x_{\min}}^{x_{\max}} f_X(x)\, \frac{\partial}{\partial \boldsymbol{\theta}_Y} S_Y(z - x; \boldsymbol{\theta}_Y)\, \mathrm{d}x.$$

The partial derivative for $S_Z(z)$ is thus given by a convolution. The partial derivative of $S_Y(z)$ in the integral can be replaced by a finite difference approximation.

### 1.5.2 Return levels

For a fixed period $T$, the corresponding return level $z$ depends on $\boldsymbol{\theta}_Y$ and $\lambda$, and therefore should be noted $z(T; \boldsymbol{\theta})$. Indeed in the equation (1.6) the left hand side should actually be written $S_Z(z; \boldsymbol{\theta}_Y)$ in place of $S_Z(z)$. The partial derivatives of $z$ with respect to $\boldsymbol{\theta}_Y$ and $\lambda$ can be obtained through the derivation of an implicit function. Using the fact that $\partial S_Z / \partial z$ is the opposite of the density $f_Z$, we get

$$\frac{\partial z}{\partial \boldsymbol{\theta}_Y} = \frac{1}{f_Z(z)} \times \frac{\partial S_Z(z)}{\partial \boldsymbol{\theta}_Y}, \qquad \frac{\partial z}{\partial \lambda} = \frac{1}{\lambda^2 T f_Z(z)} \tag{1.8}$$

where the dependence on $\boldsymbol{\theta}_Y$ has been omitted in the density $f_Z(z)$ and survival $S_Z(z)$.

## 1.6 Bayesian inference (Monte-Carlo)

In a Bayesian framework, one may have a posterior distribution for $\boldsymbol{\theta}$, say $p(\boldsymbol{\theta} \mid \mathbf{Y})$. A popular form for posterior distribution is a discrete approximation as a mixture of Dirac masses at $K$ outcomes

$$\boldsymbol{\theta}^{[1]},\ \boldsymbol{\theta}^{[2]},\ \ldots,\ \boldsymbol{\theta}^{[K]}$$

Such a distribution can be provided as a matrix with columns in correspondence with the parameters. Each row of the matrix contain a random drawing $\boldsymbol{\theta}^{[k]}$ from the posterior of $\boldsymbol{\theta}$. The random drawings are generally not independent Markov Chain Monte Carlo (MCMC). The posterior distribution of a return level $T_Z(z;\, \boldsymbol{\theta})$ for a fixed level $z$ has a straightforward discrete approximation. The posterior mean of the return period is estimated by the mean value

$$\mathbb{E}\left[T_Z(z) \mid \mathbf{Y}\right] \approx \frac{1}{K} \sum_{k=1}^{K} \frac{1}{\lambda^{[k]} \times S_Z(z;\, \boldsymbol{\theta}_Y^{[k]})}$$

and a similar formula will work for posterior moments or quantiles.

The Bayesian inference is not implemented yet.

## 1.7 Expectation of the tide conditional on the Sea Level

The importance of the tide in the formation of extreme sea level combinations can be investigated using the conditional expectation of the tide $X$ given the sea level $Z$, that is

$$\mathbb{E}\left[X \mid Z = z\right] = \frac{\int x\, f_X(x) f_Y(z-x)\, \mathrm{d}x}{\int f_X(x) f_Y(z-x)\, \mathrm{d}x} =: g(z). \tag{1.9}$$

The expectation provides an "inverse" prediction: for a given high sea level $z$, what tide $x$ should be expected on average? Note that conditional on $Z = z$ the distribution of $X$ will not in general be unimodal, and several scenarios of tide can occur.

The two integrals in the fraction of (1.9) are on the interval $(x_{\min},\, x_{\max})$ and can be computed numerically using a discrete convolution.

The behaviour of the function $g(z)$ for large $z$ mainly depends on the distribution of $Y$ and of some global features of the distribution of $X$. It can be shown that when $Y$ follows an exponential distribution $g(z)$ is constant for large $z$. Surprisingly enough, some distributions of $Y$ lead to a function $g(z)$ which is decreasing for $z$ large enough. Thus if $Y$ is $\texttt{GPD}(\mu_Y,\, \sigma_Y,\, \xi_Y)$ with $\xi_Y > 0$, it can be shown that $g(z)$ is decreasing for $z \geqslant x_{\max} + \mu_Y$ and tends to the unconditional expectation $\mathbb{E}[X]$ when $z$ tends to $+\infty$. This fact can be related to the asymptotic behaviour of the distribution of $Z$.

## 1.8 Return level plot

The return level plot is a general tool in extreme value analysis. It is often used to compare a fitted distribution for extreme values (e.g. POT) with experimental points. Usually the points are located at the largest order statistics of a sample.

The return level plot of **SeaLev** relies on the `RSLplot` function, which is called e.g. by `convSL`. It shows a distribution as a curve with points $[T,\, z(T)]$ where $z(T)$ is obtained by (1.7). A logarithmic scale is used for periods, and an ordinary scale for levels, thus the points actually plotted are couples $[\log(T),\, z(T)]$ where $T = [\lambda \times (1-p)]^{-1}$ and $z(T) = q_Z(p)$. When the

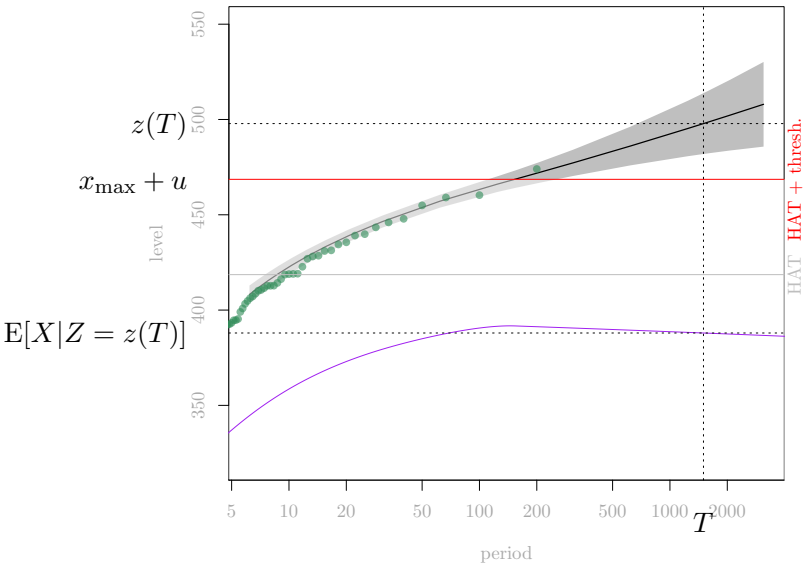


Figure 1.1: Return level plot. Empirical points can be shown. When a POT surge model with threshold $u$ is used, only levels $> x_{\max} + u$ must be considered.

distribution of $Z$ is close to the exponential, the theoretical curve is nearly a straight line. This will also be true for large return periods if the distribution of $Z$ falls in the Gumbel domain of attraction.

In the present context, the interest is focused on the sea level $Z$. Since the distribution of $Z$ results from the convolution and not from an estimation, there will generally be no use of experimental values for $Z$. In the POT context, the computation is exact only for $z > x_{\max} + u$, and thus *only the corresponding part of the return level curve must be used then.*

A curve showing the value of the conditional expectation $\mathbb{E}[X \mid Z]$ for $Z = z(T)$ is shown. In some cases, this curve may not be seen because of the limits of the axes. It is possible to change these limits, see section 2.5 later.

Experimental extreme values of the sea level can be displayed on the return level plot by using suitable plotting positions. Assume that we are given the $r$ largest values $Z_k$ of the sea level during a period with known duration $w$. The underlying total number of seal levels $n_{\mathtt{H}}$ is the number of high tides corresponding to the yearly rate $\lambda_{\mathtt{H}} = 705.8\,\mathrm{year}^{-1}$. Assuming that the $Z_k$ are in decreasing order, the return period $\widetilde{T}_k$ used for $Z_k$ is such that $1/(\lambda_{\mathtt{H}}\widetilde{T}_k)$ is the estimated probability of exceedance of $Z_k$, i.e.

$$\frac{1}{\lambda_{\mathtt{H}}\widetilde{T}_k} = \frac{k}{n_{\mathtt{H}}+1} = \frac{k}{\lambda_{\mathtt{H}} w + 1},$$

where the duration $w$ is assumed to be in years. For instance if $w = 10\,\mathrm{years}$, the largest experimental level $Z_1$ is considered as the largest value among $n_{\mathtt{H}} = 705.8 \times 10 = 7058$ levels, corresponding to a probability of exceedance of $1/7059$, and to a return period of $7059/705.8 \approx 10$ years. The rationale of the formula is that the tides $X_k$ corresponding to the $Z_k$ are assumed to occur at the same rate as randomly chosen tides. See section 2.5 page 15 for an example.

## 1.9 Special case: GPD surges

### 1.9.1 Exponential surges

A special case of interest is when $Y$ has an exponential distribution with location $\mu_Y$ and scale $\sigma_Y$. It turns out then that the distribution of $Z$ conditional on $Z > x_{\max} + \mu_Y$ is also exponential. More precisely for $z > x_{\max} + \mu_Y$ the value of the survival $S_Z(z)$ is identical to $S_{Z^\star}(z)$ with $Z^\star := \mu_X^\star + Y$ and

$$\mu_X^\star := \sigma_Y \log \mathbb{E}\left[e^{X/\sigma_Y}\right] = \sigma_Y K_X(1/\sigma_Y), \tag{1.10}$$

where $K_X(t) := \log \mathbb{E}[e^{tX}]$ is the generating function of the cumulants of $X$. In other words, except for small return periods, the return levels of $Z$ are identical to those that would be obtained with a constant astronomical tide $X \equiv \mu_X^\star$. We also have then

$$\mathbb{E}\left[X \mid Z = z\right] = \mu_X^\star \qquad \text{for } z > x_{\max} + \mu_Y.$$

Note that $\mathbb{E}[X] \leqslant \mu_X^\star \leqslant x_{\max}$, so the expected tide corresponding to large sea levels $Z$ falls somewhere between the unconditional mean tide and the maximal tide.

### 1.9.2 GPD surges

When the surge $Y$ has a GP distribution $\texttt{GPD}(\mu_Y, \sigma_Y, \xi_Y)$, the distribution of $Z$ is a continuous mixture of GPDs with scale $\sigma_Y$ and scale $\xi_Y$. The tail of $Z$ will resemble that of $Y$, with three possible behaviours.

- When $\xi_Y < 0$ the distribution of $Z$ has a finite upper end-point $z_{\max} = x_{\max} + y_{\max} < \infty$. The conditional expectation $\mathbb{E}\left[X \mid Z = z\right]$ tends to $x_{\max}$ for $z \to \infty$.
- When $\xi_Y = 0$, the tail distribution of $Z$ is exponential with scale $\sigma_Y$. The conditional expectation $\mathbb{E}\left[X \mid Z = z\right]$ is equal to $\mu_X^\star$ in (1.10) for $z > x_{\max} + \mu_Y$.
- When $\xi_Y > 0$ it can be shown that $Z$ is *tail-equivalent* to $Y$, i.e. $S_Z(z)/S_Y(z)$ tends to 1 when $z \to \infty$. So $Y$ is heavy-tailed. The conditional expectation $\mathbb{E}\left[X \mid Z = z\right]$ tends to $\mathbb{E}[X]$ for $z \to \infty$.

When the shape is small $\xi_Y \approx 0$, the distribution of $Z$ can be approximated as $\texttt{GPD}(\mu_X^\star + \mu_Y, \sigma_Y, \xi_Y)$ with $\mu_X^\star$ as above in (1.10).

**Remark**. A comparable approximation is used in [Col90] for annual maxima of sea level modelled with a GEV distribution. The impact of the tide on the distribution of annual maximal sea levels is simply to shift the distribution of annual maximal surges. The value of the shift is the average value of $\sigma \exp(X_t/\sigma)$, where $\sigma$ is the GEV scale parameter which plays the same role as the GPD scale for the tail distribution. In view of (1.4) above for the function $\phi := \exp$, the shift is a close approximation to $\mu_X^\star$.

# Chapter 2

# Using the convSL function

## 2.1 Goals

The `convSL` function computes return levels for the sea level $Z$ using the distributions for $X$ and $Y$ given on input. It returns a list with several objects, among which a "prediction" table associating return periods or probabilities to return levels. When possible, approximate confidence limits are computed using the delta method.

This function is not concerned with estimation tasks (e.g. POT), which should rely on other packages.

## 2.2 Non-parametric density of $X$

The density of $X$ can be provided as a list with elements `x` and `y`. It can be an R object of the (S3) class `density`, such as computed with the `density` function of the **stats** package. The range of $X$ is obtained as the range of the `x` element of the list.

The dataset `Brest.tide` from `SeaLev` provides an example of estimated density for high-tide sea levels in Brest. The `plot` function call and subsequent graphics calls produce the plot on the left of the figure 2.1.

```
library(SeaLev)
data(Brest.tide)
class(Brest.tide)

## [1] "list"

str(Brest.tide)

## List of 2
##  $ x: num [1:512] 100 101 102 102 103 ...
##  $ y: num [1:512] 0.00 5.84e-06 1.12e-05 1.66e-05 2.20e-05 ...

plot(Brest.tide, col = "SeaGreen", type = "l",
     main = "Density of high-tide sea level in Brest")
grid(); abline(h = 0)
```

The level $X$ is given here in centimetres, the density values are accordingly in $\mathrm{cm}^{-1}$. As implicitly admitted when plotting densities, it will be assumed that linear interpolation can be

used to evaluate $f_X(x)$ on another grid of values, usually a finer one. The required normalisation condition is that the trapezoidal rule for numerical integration should lead to an integral equal to 1.0. Provided that the density values are zero at end-points, the rectangles rule should also give the same value 1.0.

```
Brest.tide$y[c(1L, length(Brest.tide$y))]

## [1] 0 0

h <- diff(Brest.tide$x)[1]
h * sum(Brest.tide$y)

## [1] 1
```

These checks could be replaced in future versions by the registration of a formal class for discredited densities.

## 2.3 Convolution

### 2.3.1 Specifying parameters for $Y$

The parameter values (generally estimated) must be given as a named list or a numeric vector with named elements. For instance, consider the high-tide skew surges for Brest, and assume that in a POT analysis using a threshold $u = 50$ cm we got the estimated parameters $\sigma_Y = 10$ cm (scale) $\xi_Y = -0.01$ (shape), and that the exceedances occurred at a rate of `1.6` years$^{-1}$. We can store these informations as R objects say `u`, `theta.y` and `lambda`

```
u <- 50
theta.y <- c("scale" = 10, "shape" = -0.01)
lambda <- 1.6
```

Note that we can use a named numeric vector created with the `c` function or a `list`, but in both cases the element names must match the parameters names of the distribution. Now we can use the created objects as values for the formal arguments `threshold.y`, `par.y` and `lambda` of the convolution function.

```
conv.gpd0 <- convSL(dens.x = Brest.tide,
                    threshold.y = u, distname.y = "GPD",
                    lambda = lambda, par.y = theta.y,
                    main = "Sea-level with GPD surges: given parameters")
```

By default, a return level plot is produced as in figure 2.1. No "confidence band" can be plotted here since no information was given about estimation uncertainty.

### 2.3.2 Using a fitted POT model

The estimated values for the surge can be computed using **Renext** and its `Brest` dataset. The arguments to be passed to the `Renouv` function then include the vector of surges `x` and the effective duration (in years) in order to estimate the rate `lambda` (in inverse years).

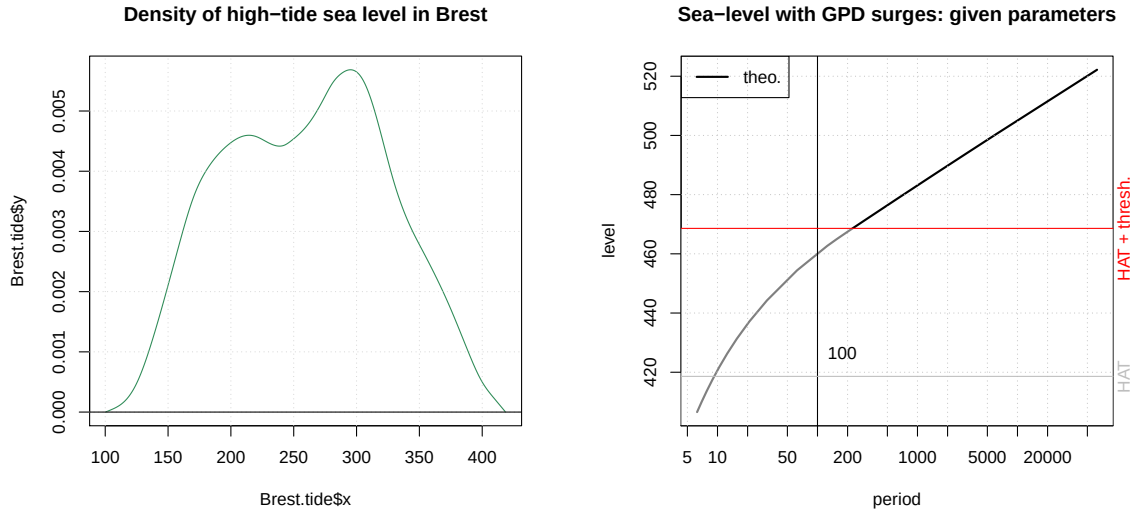


Figure 2.1: Left panel: density of the high-tide sea level $X$ in Brest (France). Right panel: return level plot using convolution and a known GPD distribution for the surge $Y$. Only the part above the horizontal red line should be used, corresponding to return periods over about 200 years.

```
library(Renext); data(Brest)
fit.gpd1 <- Renouv(x = Brest$OTdata$Surge,
                   effDuration = as.numeric(Brest$OTinfo$effDuration),
                   threshold = 50, distname.y = "GPD",
                   main = "GPD surge")
coef(fit.gpd1)

##       lambda        scale        shape
##  1.612247663 10.667124374 -0.006425839
```

The estimated parameters are very close to those used before. The fit produces the return level plot shown on the left of 2.2, with a 100-years return level of about 100 cm. The fitted object contains a covariance matrix of estimation.

```
cov1 <- vcov(fit.gpd1)
cov1

##              lambda       scale        shape
## lambda 0.01092161  0.00000000  0.000000000
## scale  0.00000000  0.95005346 -0.044531845
## shape  0.00000000 -0.04453185  0.004147856
```

This matrix can be used in the `covpar.y` formal argument of `convSL` function. As it is the case here, the matrix must have rownames and colnames, and these must agree with the parameter names of the distribution.
We get the return level at the right of figure 2.2, in which (pointwise) confidence bands are drawn for the return levels. These are obtained by "propagating the uncertainty" on the parameters

(as quantified by the covariance) to the return levels $z(T)$. This is done using the delta method and the partial derivatives (1.8).

The plot can be enhanced by filling the confidence region(s) and using colours. The confidence levels can be set using `pct.conf`.

```
conv.gpd2a <- convSL(dens.x = Brest.tide,
                     threshold.y = 50,
                     distname.y = "GPD",
                     lambda = lambda, par.y = theta.y,
                     pct.conf = c(95, 90),
                     filled.conf = TRUE, mono = FALSE,
                     covpar.y = cov1,
                     main = "Sea-level for Brest with GPD surges (lambda known)")
```

The plot is shown on the left panel of figure 2.3.

It is possible to use in `convSL` a covariance matrix without the elements related to the rate `"lambda"`. For instance, dropping the first row and the first column in `cov1`

```
cov1[-1, -1]

##              scale        shape
## scale  0.95005346 -0.044531845
## shape -0.04453185  0.004147856
```

leads to a matrix that can be used with `convSL`. The same effect can be obtained by specifying a `use.covlambda` argument with `FALSE` as its value.

```
conv.gpd2 <- convSL(dens.x = Brest.tide,
                    threshold.y = 50,
                    distname.y = "GPD",
                    lambda = lambda, par.y = theta.y,
                    use.covlambda = FALSE,
                    pct.conf = c(95, 90),
                    filled.conf = TRUE, mono = FALSE,
                    covpar.y = cov1,
                    main = "Sea-level for Brest with GPD surges (lambda known)")
```

The plot is shown on the right panel of figure 2.3. The effect of ignoring the uncertainty on `lambda` is to produce a narrower confidence band for small return periods. The effect for large periods is negligible.

### 2.3.3 Using a fitted non-POT model

Although a POT model will be used in most cases, it is yet possible to use a non-POT model, i.e. a non-conditional distribution for $Y$. For illustration purpose only, assume that the surge at Brest can be described by a Gumbel distribution with parameters $\mu_Y = -10.8$ cm (location) and $\sigma_Y = 10$ cm (scale). The Gumbel assumption for surges is very close to that of exponentially distributed excesses over a high enough threshold. Here the parameters were chosen in accordance with the POT estimation: $\sigma_Y$ takes the same values as in the GPD case, while $\mu_Y$ was chosen to give the same rate of exceedance over $u = 50$ cm.

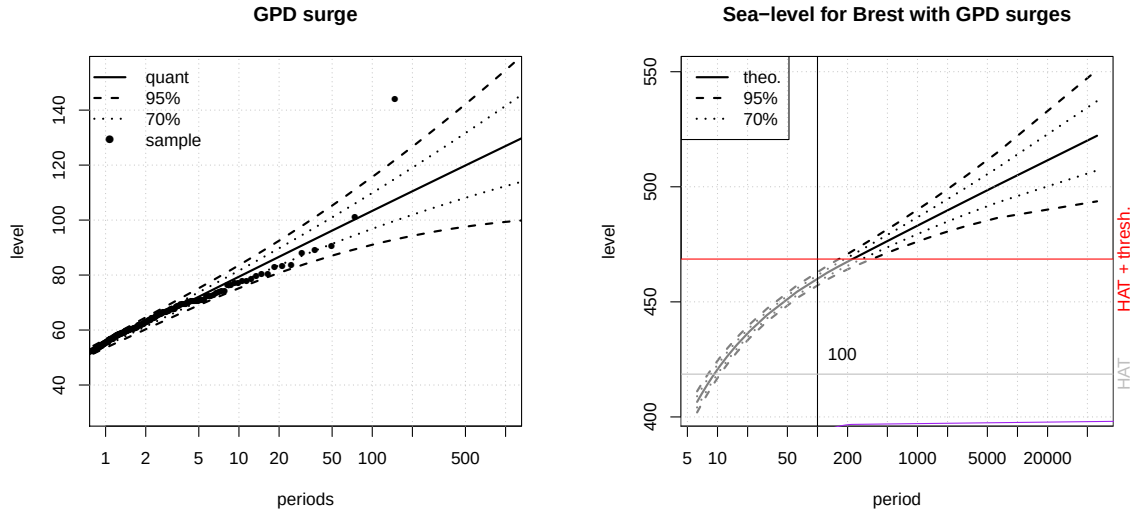


Figure 2.2: Fitting a POT model for Brest surge with **Renext** (left), and using the fitted distribution within a convolution.

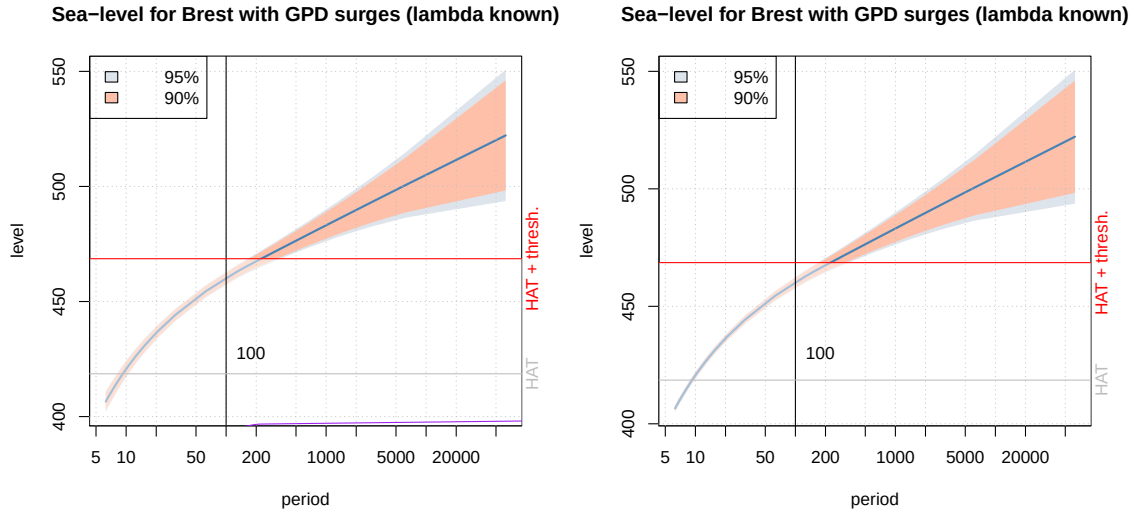


Figure 2.3: Comparison of two convolutions of the tide with the fitted GPD. On the left panel, the covariance concerns `lambda`. Right panel `use.covlambda = FALSE`.

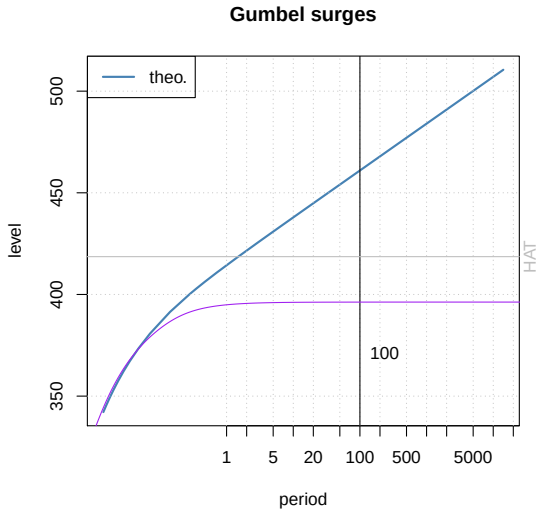


Figure 2.4: Using a Gumbel distribution (non-POT) for the surge. Note that the horizontal line $x_{\max} + u$ shown for POT surges no longer exists.

The arguments provided to `convSL` will be quite different than in the GPD case. We specify a non-POT distribution by using a `threshold.y` with value `NA`, and the rate `lambda` must now be 705.8 years$^{-1}$.

```
par.y <- c(loc = -10.8, scale = 10)
res.gumbel <- convSL(dens.x = Brest.tide,
                     threshold.y = NA,
                     distname.y = "gumbel",
                     lambda = 705.8,
                     par.y = par.y,
                     filled.conf = TRUE, mono = FALSE,
                     main = "Gumbel surges")
```

The return level is shown on figure 2.4. Note that `threshold.y` is equal to its default value `NA` and that we could have left `lambda` to its default value since this is 705.8 when `lambda` is `NA` (see the package manual). Thus these two arguments could have been omitted in the call.

**Remark**. By using `lambda = 705.8` we assume that the given distribution for $Y$ is for an arbitrary skew surge as in [Sim07, chap. VIII]. If instead we aim to use the a*nnual maximal surge*, then we must use `lambda = 1.0` with a suitable distribution.

## 2.4 Predictions

The computed return levels and confidence limits are returned within a data.frame `pred`. Here are the first rows.

```
head(conv.gpd2$pred, n = 3)

##         prob period    quant     L.95     U.95     L.90     U.90
```

```
## 100 0.993750    100 460.1913 457.7357 462.6469 458.1305 462.2521
## 200 0.996875    200 467.4752 464.3794 470.5711 464.8771 470.0733
## 500 0.998750    500 476.4099 471.5000 481.3198 472.2894 480.5304
```

Each row correspond to a given period $T$, e.g. $T = 100$ years, and give the corresponding probability of non-exceedance $p(T)$ (column `prob`), the corresponding return level $z(T)$ (column `quant`) as well as confidence limits for $z(T)$, here 70 pct and 95 pct. It is possible to specify the wanted periods by using the `pred.period` formal of `Renouv`.

Recall that $z(T)$ and $p(T)$ are connected to each other by $z = 1/[\widehat{\lambda} \times (1-p)]$, thus the relation between $T$ and $p$ is affected by the uncertainty on the estimation of $\lambda$. However, this uncertainty is small for large periods.

Also note that the term "prediction" can be misleading. The 100-years return level is the level that is exceeded on average once every 100 years. This level might occur twice or more in a given century.

When a POT model is used for the surges $Y$, only periods corresponding to levels $z > x_{\max} + u$ must be used, where $u$ is the threshold.

## 2.5 Adjusting the return level plot

The axis limits can be adjusted using the `ylim` parameters and the "dots" mechanism just like as for the `main` formal. It will generally be necessary to modify `ylim` in order to see the conditional expectation curve $\mathbb{E}(X \mid Z = z)$ as in

```
conv.gpd3 <- convSL(dens.x = Brest.tide,
                    threshold.y = 50, distname.y = "GPD",
                    lambda = lambda, par.y = theta.y, covpar.y = cov1,
                    ylim = c(300, 600),
                    main = "Sea-level for Brest with GPD surges (lambda known)")
```

leading to the plot on left of figure 2.5.

For the x-axis, which is in log-scale, it is preferable to work `Tlim` which allows to give the two limits in years.

```
conv.gpd3 <- convSL(dens.x = Brest.tide,
                    threshold.y = 50, distname.y = "GPD",
                    lambda = lambda, par.y = theta.y, covpar.y = cov1,
                    Tlim = c(100, 3000),
                    main = "Sea-level for Brest with GPD surges (lambda known)")
```

The plot can be annotated with the standard functions from the **graphics** package: `text`, `lines`, etc. Since the $x$-axis is in log-scale it will be simpler to use `par()usr` to get the world coordinates or to use the `locator` function.

In order to add experimental points to the plot, the two formal arguments `z` and `duration` must be passed to the `RSLplot`. In the simplest case, `z` is a numeric vector and `duration` is a positive numeric value representing a duration in years. For instance, with meaningless points and in a purely illustrative purpose we get the plots of figure 2.6.

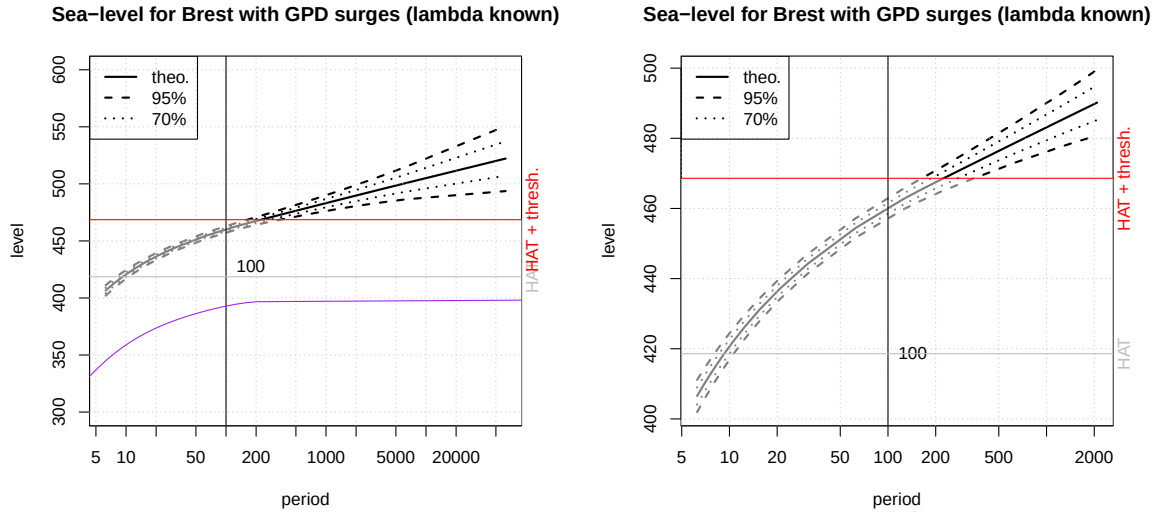


Figure 2.5: Changing the axes. Left: the `ylim` argument of `plot` was used. Right: the `Tlim` argument of `RSLplot` chooses the range of return periods, here from 100 to 5000.

```
res.g2 <- convSL(dens.x = Brest.tide,
                 threshold.y = NA, distname.y = "gumbel",
                 lambda = 705.8, par.y = par.y,
                 filled.conf = TRUE, mono = FALSE,
                 main = "Artificial empirical points (1 set)",
                 z = c(500, 490, 480, 460),
                 duration = 200)
```

It is possible to specify several vectors using a list for `z`, the duration being then a vector or a list with the same length as `z`.

```
res.g3 <- convSL(dens.x = Brest.tide,
                 threshold.y = NA, distname.y = "gumbel",
                 lambda = 705.8, par.y = par.y,
                 filled.conf = TRUE, mono = FALSE,
                 main = "Artificial empirical points (2 sets)",
                 z = list(c(500, 490, 480), c(440, 420, 380, 350)),
                 duration = c(200, 170))
```

Some properties of the points such as the colour can be changed by passing suitable arguments to the `RLSplot` function, with names `.points`. For instance, a vector of colours can be specified with `col.points`.

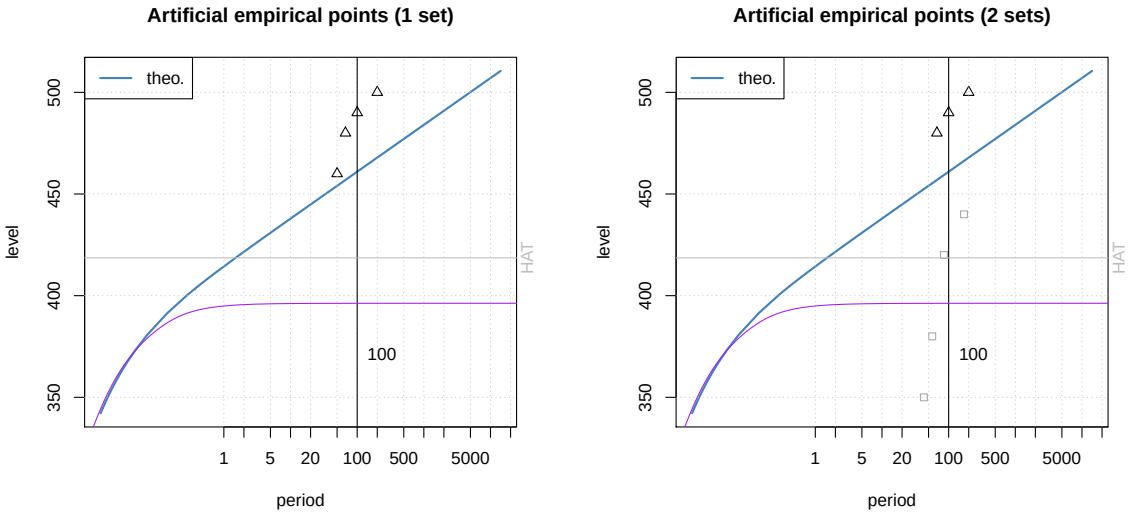


Figure 2.6: Adding empirical points to the return level plot using `z` and `duration`. When several sets are used, `z` must be a list of numeric vectors.

# Chapter 3

# Spline density for the tide

## 3.1 Motivation

From version >=0.4-0 **SeaLev** allows the use of a spline density for the tide. Splines can provide a fairly good representation of general smooth densities with bounded support. Moreover, using a spline density for $X$ has a special interest here because if $Y \sim$ `GPD` the value of the survival $S_Z(z)$ for $z$ large enough can be given in closed form. Some details are given in appendix A.3.

**SeaLev** embeds two main functions dedicated to spline densities.

- `SplineDensity` creates a spline density with bounded support $(x_{\min}, x_{\max})$. The created object has (S3) class `"SplineDensity"` for which a few methods such as `plot` or `predict` are provided.
- `GPtail` computes the convolution of a spline density with a GPD. The spline density is provided as an object with class `"SplineDensity"` while the GPD is given by its parameters as in the `convSL` function.

A few other functions are devoted to the `"SplineDensity"` class.

**Remark**. Log-splines are generally preferred to splines for probability densities because the positivity constraints are more easily coped with when the log-exponential transforms are used. However, the closed form of the convolution tail does not exist for a log-spline.

## 3.2 Example

Consider again the astronomical tide at Brest. From the original grid density, we can build a spline density and for example use the `plot` method to produce the two plots of figure 3.1.

```
SD <- SplineDensity(x = Brest.tide$x, f = Brest.tide$y)

## leftDeriv = 0 9.381932e-06 NA
## rightDeriv = 0 -2.479663e-05 NA

SD10 <- SplineDensity(x = Brest.tide$x, f = Brest.tide$y, nKnots = 10)

## leftDeriv = 0 9.381932e-06 NA
## rightDeriv = 0 -2.479663e-05 NA
```

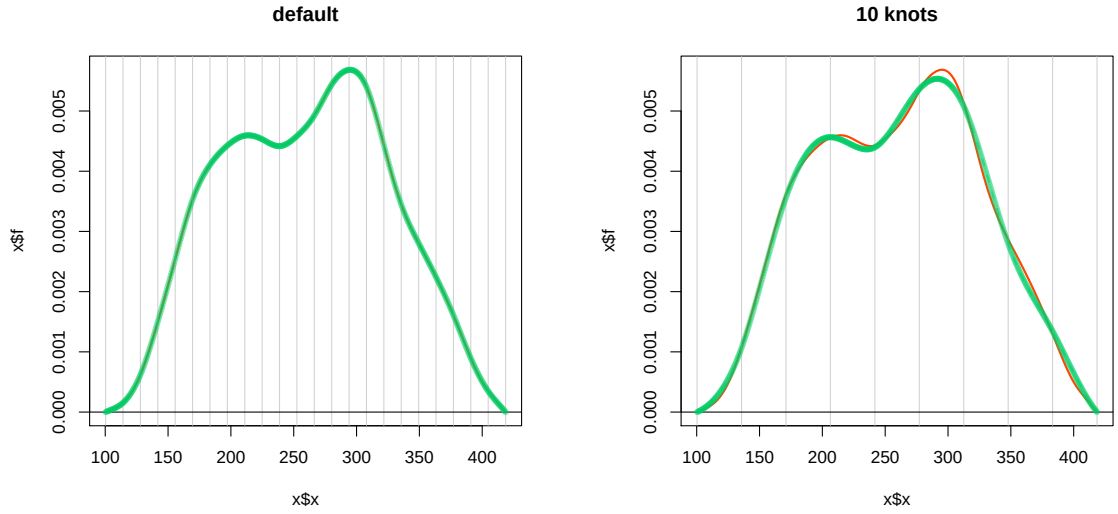


Figure 3.1: Spline density approximation for Brest. The original grid density (with 512 points) is shown as a red curve in both cases. With only the default 24 knots (left), the spline approximation is very close to the original density which is then hidden. With as few as 10 knots (right), the approximation is quite good.

```
plot(SD, main = "default")

## NULL

plot(SD10, main = "10 knots")

## NULL
```

Knots locations are chosen by default to be equispaced. The number of knots can be fixed with `nKnots` and the knots locations can be given as well, if needed, by using the `knots` formal argument. As it might be guessed from the generated message, the created spline uses the values of the derivatives at the two-end points $x_{\min}$ and $x_{\max}$. Here they are computed using a finite difference approximation, but they could have been specified. Using good boundary conditions is often an issue in density estimation. The derivatives of the spline at end-points can be prescribed thanks to a multiple knots strategy, see appendix A.3.

Note that the spline is fitted to the provided density by using a least-squares criterion with the constraint that the spline density integrates to 1 as wanted. Further equality constraints are used for boundary conditions if needed. However, positivity constraints are not used for now, which is an obvious limitation. As far as tide densities are used with a fine grid representation, positivity constraints will nearly hold in practice. The function `SplineDensity` may still warn about some slightly negative spline values, with no serious consequence.

**Remark**. Cubic splines with positivity constraints could be coped with using Second Order Constrained Programming (SOCP), [Pap14].

The S3 class `"GPDtail"` describes distributions which result from the convolution of spline density and a GPD. The `GPDtail` function is used to create an object of this class.

```r
set.seed(1234)
u <- 50
par.y <- c("scale" = rgamma(1, shape = 2, scale = 30),
           "shape" = 0.2 * runif(1))
res <- GPtail(x = SD, par.y = par.y, threshold.y = u, lambda = 1)

## o Period for return levels (pred.period)
##  [1] 1e+00 2e+00 5e+00 1e+01 2e+01 3e+01 4e+01 5e+01 6e+01 7e+01 8e+01 9e+01
## [13] 1e+02 2e+02 5e+02 1e+03 1e+04 1e+05
## ymax =    Inf
## E.y =   13.2 sd.y =   14.2
## o computing the expectation of the exponential tail: muStar.x = 354.625
## o Using closed form smax =      716

class(res)

## [1] "GPtail"

plot(res)
plot(res, which = 3)
```

The `plot` method for this class produces by default a plot showing the three densities as shown on figure 3.2. It has a `which` argument that can be used to produce a different plot, the return level plot as on figure 3.3 being obtained with `which = 3` (this value is consistent with the **evd** package).

The call to `GPtail` is not unlike a call to `convSL`. The probability distribution is not given here, because only the GPD can be used. Note that the name `"GPtail"` can be misleading because the distribution of $Z$ resulting from the convolution does not have a GPD tail but is tail-equivalent to a GPD for $\xi_Y > 0$.

## 3.3 Using `"SplineDensity"` objects

### 3.3.1 Evaluation

The `predict` method can be used to evaluate the density at some chosen points. The result is given as a list with two elements `x` and `y` and thus it can straightforwardly be plotted. For example, the spline density can be evaluated on a very fine grid if needed.

```r
pred <- predict(SD, newdata = seq(from = 200, to = 300, by = 0.1))
names(pred)

## [1] "x" "y"
```

### 3.3.2 Moments/cumulants generating function

The `momGen` function can be used to provide an accurate evaluation of the moment generating function $M_X(t) = \mathbb{E}[\exp(tX)]$ or of the cumulant generating function $K_X(t) = \log M_X(t)$. The later function is useful to assess the impact of a tide $X$ for an exponential or nearly exponential surge $Y$, see section 1.9.1 above.

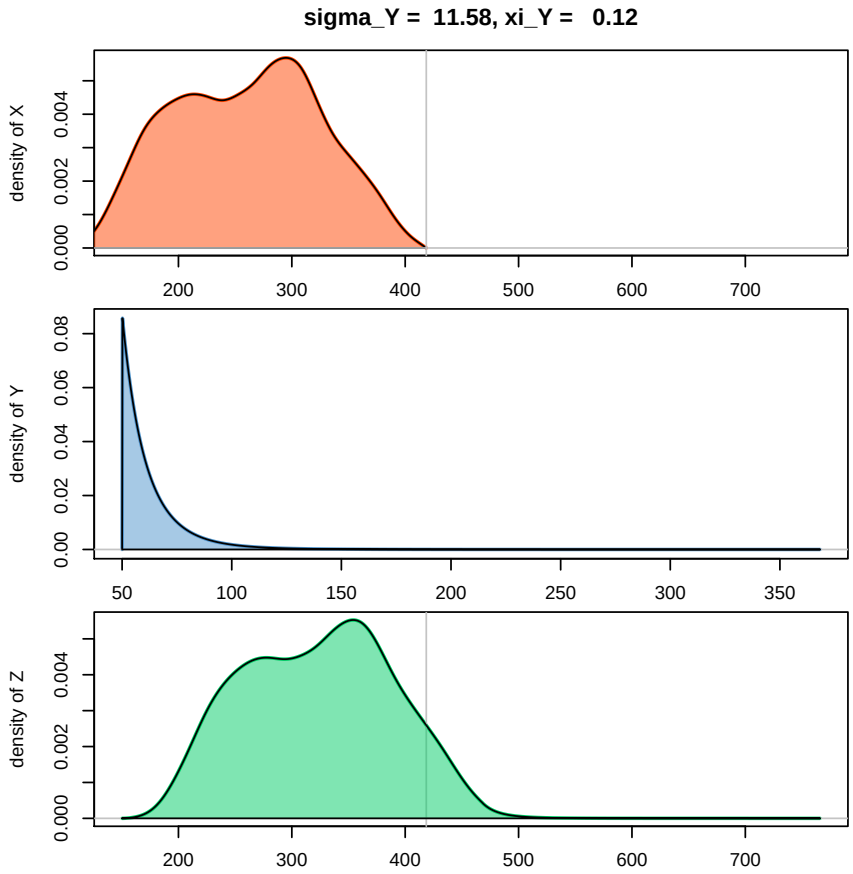


Figure 3.2: Convolution of a spline density and a GPD. The density of $Z$ seems similar to that of $X$ with a shift, but $Z$ has heavy tail, and is tail-equivalent to $Y$.

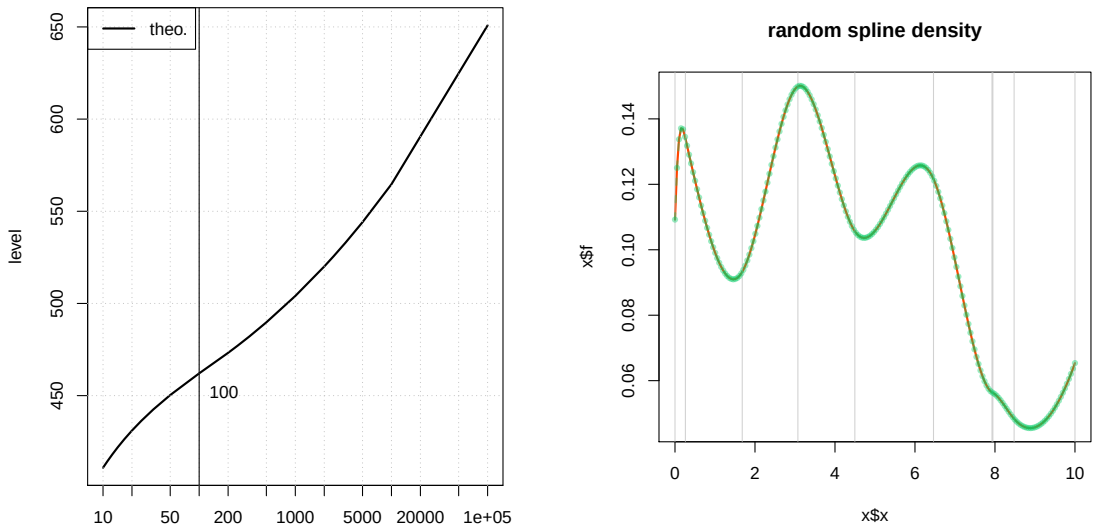


Figure 3.3: Left: Return level for the convolution of a spline density and a GPD for Brest data. Right: a random spline density.

### 3.3.3 Random `SplineDensity`

With the goal of testing the computations, a **`rSplineDensity`** function has been written to generate a random spline density with given support.

```
set.seed(1234)
SDrand <- rSplineDensity(order = 4, xmax = 10)
plot(SDrand, main = "random spline density")

## NULL
```

**Remark**. The simulated density is obtained using a B-spline basis and positive coefficients. We know that in this way we do not obtain an arbitrary positive spline because a spline with some negative coefficients still can take be a positive function. However, the densities generated in this way can take fairly different shapes as needed in checks.

# Chapter 4

# Frequently Asked Questions

## 4.1 Calling convSL

**Q.** Calling `convSL`, I get an error with a message concerning a function `qfun.y`.

**A.** A plausible explanation is that the distribution for the surge does not belong to the list of special distributions and that it does not meet the requirements about functions names. In the second case, it may be possible to redefine the probability functions by using a "wrapper". For instance, if the distribution depends on a parameter `bar` and has density `foodens`, a new density is defined

```
dMydist <- function(x, bar) foodens(x, bar = bar)
```

In order to use `"Mydist"` as a possible `distname.y` choice, the same thing must be done for the distribution and the quantile functions which must have name `pMydist` and `qMydist`.

**Q.** I do not want to use a threshold $u$ for $Y$ nor to describe excesses, but rather to use a known distribution for $Y$.

**A.** Make sure that the distribution meet the requirements on probability functions (see question above), and proceed as in the example of 2.3.3 page 12.

**Q.** I have a warning message when calling `convSL` mentioning that something `"is not a graphical parameter"`.

**A.** Due to the `dots` mechanism, no check is possible for the formal arguments of `convSL`. When a formal argument is not found in the list of arguments for `convSL` it is passed to `RSLplot` and possibly then to `plot`. When a formal is given which does not belong to any of the three argument lists, a message is addressed containing the above mentioned words. The formal will not be taken into consideration Most likely, it is a misuse, and **the message must be read carefully**.

## 4.2 Inference

**Q.** The confidence intervals for return levels are of decreasing width when the level $z$ increases, which seems unnatural.

**A.** Such a phenomenon can occur when the uncertainty about parameters is dominated by the uncertainty on the rate $\lambda$. In the estimation variance of a return level $z(T)$, the part that can be attributed to $\lambda$ is computed using the second partial derivative of (1.8) for $\lambda = \widehat{\lambda}$. This gives

$$\mathrm{Var}(\widehat{\lambda}) \times [\partial z/\partial \lambda]^2 = \mathrm{Var}(\widehat{\lambda}) \times \widehat{\lambda}^{-4} T^{-2} f_Z(z)^{-2}$$

It might be the case that $T^{-2} f_Z(z)^{-2}$ is decreasing with $T$. Note however that the uncertainty on the parameters of $Y$ is usually the main source of uncertainty on the return levels for large periods. Decreasing widths for the confidence intervals can also result from a misuse of a fitted model: wrong units, error in parameter names or order, etc.

## 4.3 Numerical precision

**Q.** What about the numerical error? What is the maximal period that can be trustfully be used?

**A.** The numerical precision depends on the distributions used for $X$ and $Y$, and no general indication can be given. As an order of magnitude, the use of probability of exceedances about $10^{-6}$ or even $10^{-7}$ seems viable in `convSL` for distributions of surges with a tail close to the exponential, as met in practice. The largest return levels will then be evaluated with a precision close to 1%. With the `GPtail` function, the precision should be better, but this still needs more investigations.

# Appendix A

# Numerical computation

## A.1 Discrete convolution

In this section we will consider vectors with 0 as starting index, e.g. a vector $\mathbf{a}$ of length $N$ writes

$$\mathbf{a} = [a_0,\, a_1,\, \ldots,\, a_{N-1}]^\top$$

Such a vector can be related to the polynomial

$$a(\lambda) = a_0 + a_1\,\lambda + a_1\,\lambda^2 + \cdots + a_{N-1}\,\lambda^{N-1}$$

Using two vectors $\mathbf{a}$ and $\mathbf{b}$ of length $N$ we can compute their convolution product which is the vector $\mathbf{c} = [c_n]_n$

$$c_n = \sum_{k=0}^{n} a_k\, b_{n-k} = \sum_{k,\,\ell\geqslant 0,\, k+\ell=n} a_k\, b_\ell \tag{A.1}$$

Note that $c_n$ is coefficient of $\lambda^n$ in the product of the two polynomials related to $\mathbf{a}$ and $\mathbf{b}$ i.e. $c(\lambda) = a(\lambda)\, b(\lambda)$. On a plane grid of points with integer coordinates $(k,\, \ell)$, the coefficient is obtained by summing products $a_k b_\ell$ on the line with equation $k + \ell = n$ with slope $-1$ (see figure A.1).

The product $c_n$ can be computed for any index $n \geqslant 0$ using the convention that $\mathbf{a}$ and $\mathbf{b}$ are completed by zeros e.g. $a_k = 0$ for $k \geqslant N$. Then $c_n$ can differ from zero for $n$ between 0 and $2N - 2$. Taking $n = 2N - 2$ the sum (A.1) reduces to one summand for $k = N - 1$, namely $a_{N-1}\, b_{N-1}$ (see figure A.1). In other words the convolution of two vectors of length $N$ has length $2N - 1$.

It can be remarked that

$$\sum_n c_n = \left[\sum_k a_k\right]\left[\sum_\ell b_\ell\right]$$

which is easily checked using $c(\lambda) = a(\lambda)b(\lambda)$ for $\lambda = 1$.

## A.2 Continuous convolution

### A.2.1 Grids

Consider the convolution integral of (1.1)

$$f_Z(z) = \int_{x_{\min}}^{x_{\max}} f_X(x)\, f_Y(z - x)\, \mathrm{d}x \tag{A.2}$$

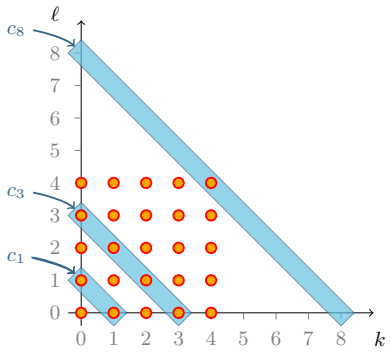


Figure A.1: Convolution of two vectors of length $N = 5$. If each point $[k, \ell]$ shows the product $a_k\, b_\ell$, then $c_n$ comes by summation over a segment of the line $k + \ell = n$. For $n = 2N - 2$ (here $n = 8$), the sum boils down to $k = \ell = N - 1$.

The densities $f_X(x)$ and $f_Y(y)$ will be used in relation with two discrete regular grids $x_k$ and $y_k$. The two grids are assumed to have the same step $h$ and the same number $N$ of intervals. For the $x$-grid, the intervals are $(x_k,\, x_{k+1})$ with

$$x_0 < x_1 < \cdots < x_N \qquad x_k = x_0 + k \times h \quad (k \text{ integer})$$

and similarly let $y_\ell = y_0 + \ell \times h$ for integer $\ell$. The grid $x_k$ is assumed to cover the support of $f_X(x)$, i.e. $x_0 \leqslant x_{\min}$ and $x_{\max} \leqslant x_N$. Then from (A.2)

$$f_Z(z) = \int_{x_{\min}}^{x_{\max}} = \int_{x_0}^{x_N} = \sum_{k=0}^{N-1} \int_{x_k}^{x_{k+1}} \tag{A.3}$$

where all integrals share the same integrand as (A.2). Consider the sequence $a_k$ and $b_k$ formed by the values of the densities $f_X(x)$ and $f_Y(y)$ at grid points

$$a_k = f_X(x_k), \quad b_k = f_Y(y_k) \qquad 0 \leqslant k \leqslant N - 1 \tag{A.4}$$

with the convention $a_k$ and $b_k$ are zero when $k < 0$ or $k > N - 1$. Let $z_0 = x_0 + y_0$ and $z_n = z_0 + n \times h$ for integer $n$. Then $z_n - x_k = y_{n-k}$ for all $n$, $k$.

### A.2.2 Rectangles or trapezes

The rectangles approximation for $f_Z(z)$ replaces each integral $\int_{x_k}^{x_{k+1}}$ in (A.3) by the product of the length $h = x_{k+1} - x_k$ and the integrand at $x_k$. This gives

$$\int_{x_k}^{x_{k+1}} f_X(x)\, f_Y(z - x)\, \mathrm{d}x \approx h f_X(x_k) f_Y(z - x_k) \tag{A.5}$$

Replacing $z$ by $z_n$ and summing for $k = 0$ to $k = N - 1$, we get

$$f_Z(z_n) \approx h \sum_{k=0}^{N-1} a_k b_{n-k}$$

The sum at right hand side has the same summand as the convolution product $c_n$, but the sum runs from $k = 0$ to $n$ for $c_n$, against $k = 0$ to $N - 1$ here. The two sums are identical for $0 \leqslant n \leqslant 2N - 1$, i.e.

$$\sum_{k=0}^{N-1} a_k b_{n-k} = \sum_{k=0}^{n} a_k b_{n-k} \qquad \text{for} \quad 0 \leqslant n \leqslant 2N - 1 \tag{A.6}$$

The reason is that $a_k$ and $b_\ell$ are zero when $[k,\, \ell]$ is outside the square $0 \leqslant k,\, \ell \leqslant N-1$ (see figure A.1). Note however that $f_Y(y_\ell)$ is only *approximately zero* and that the square side $Nh$ must be chosen with care, see A.2.3.

The rectangles rule is known to be less precise than the trapezoidal rule which has the same computational cost, and the later is always preferred. The integral $\int_{x_k}^{x_{k+1}}$ in (A.3) is then approximated by the product of the length $h$ and the mean value of the integrand at the two end-points $x_k$ and $x_{k+1}$

$$\int_{x_k}^{x_{k+1}} f_X(x)\, f_Y(z-x)\, \mathrm{d}x \approx \frac{h}{2}\left\{f_X(x_k) f_Y(z-x_k) + f_X(x_{k+1}) f_Y(z-x_{k+1})\right\} \tag{A.7}$$

When $a_0 = 0$ and $a_N = 0$, the trapezoidal rule leads unsurprisingly to the same computation as the rectangles rule.

### A.2.3 Moderate return levels: discrete convolution

While the support of $X$ is assumed to have finite bounds $x_{\min}$ and $x_{\max}$, the support of $Y$ will in most cases be infinite with $y_{\max} = +\infty$. Then we need to fix a suitable finite upper limit $y^\star_{\max}$ in the computations. Inasmuch as the density $f_Y(y)$ can be infinite at $y_{\min}$ for some distributions of interest[1], we will replace the lower end-point $y_{\min}$ by $y^\star_{\min}$ with $S_Y(y^\star_{\min}) = 1-\varepsilon$ and $\varepsilon > 0$ is chosen small. Then the value of $f_Z(z)$ and $F_Z(z)$ will be computed by discrete convolution for $y$ in the interval $(y^\star_{\min},\, y^\star_{\max})$ where $y^\star_{\max} \leqslant y_{\max}$ is chosen with the constraint that the computing range $y^\star_{\max} - y^\star_{\min}$ is not too large compared to $x_{\max} - x_{\min}$. So the density of $Z$ computed in this way will be only for values $z \leqslant x_{\max} + y^\star_{\max}$.

The discrete convolution can rely on the `convolve` functionfrom the `stats` package. This function uses the Fast Fourier Transform (FFT) hence is very fast.

**Remarks**

• The previous algorithm only describes the computation of $f_Z(z)$ at grid values $z_n$. Several results are obtained using a similar convolution: approximated confidence limits (delta method), conditional expectation $\mathbb{E}\,[X \mid Z = z]$. See the code of the function `convSL` for more details.

• In the cases where the surge distribution has unbounded density either at $y_{\min}$ or at $y_{\max}$, the results must be considered with care. A GPD density is always finite at $y_{\min}$; it infinite at $y_{\max}$ when $\xi < -1$ which is unlikely to occur in practical situations.

### A.2.4 Large return periods: quadratures

The discrete convolution works fine as far as the used range of $Y$ remains comparable to the range of $X$, for instance the discretised range $y^\star_{\max} - y^\star_{\min}$ remains $\leqslant 3\,[x_{\max} - x_{\min}]$. But if the upper end-point of $Y$ is $\infty$ and if very large return periods are needed, e.g. for $T > 10^4$ years, the discrete convolution might not be accurate enough. The reason is that the discrete grids for $X$ and $Y$ must have the same step $h$. But if $y^\star_{\max} - y^\star_{\min}$ is very large, then either the chosen step $h$ will be too large to provide a good description of the density of $X$, or the number $N$ of grid points will be very large. Though a large value of $N$ has a limited impact on the computation time, it still has a nasty effect on the accuracy of the results.

Fortunately enough, the function $S_Z(z)$ will in practice have only quite slow variations for large $z$. So the survival $S_Z(z)$ can be evaluated at some sparse large values rather than on a fine grid of large values. No more than two dozens of points are needed for, say, $10^3 \leqslant T \leqslant 10^7$. For

[1] Such as Weibull or gamma distributions with decreasing hazards.

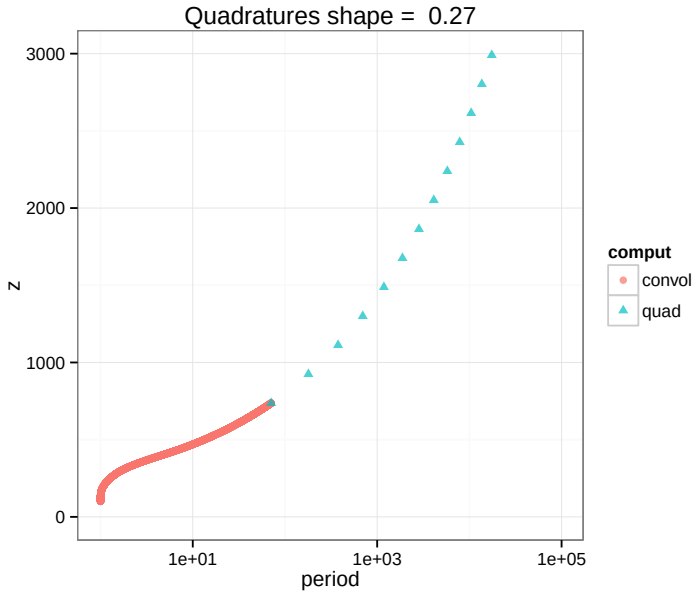


Figure A.2: A discrete convolution can be used to compute the return levels for moderately large periods. A few quadratures can then be used for larger periods. Note that the two parts of the curve are smoothly connected.

one given value of $z$, the value of the density $f_Z(z)$ as given by formula (A.2) can be evaluated accurately by using a quadrature with a few hundreds of points. So a discrete convolution can be used for moderately large periods, while a few quadratures can be used for larger periods, see figure A.2.

## A.3 Spline tide density and GPD surges

### A.3.1 Knots

In this section, the density $f_X(x)$ will be assumed to be a spline of order $k \geqslant 1$ with a sequence $\zeta_k$ of $\ell + 1$ knots

$$\zeta_0 < \zeta_1 < \cdots < \zeta_\ell$$

$\zeta_0 = x_{\min}$ and $\zeta_\ell = x_{\max}$. In an usual framework this means that $f_X(x)$ coincides with a polynomial of degree $\leqslant k-1$ on each of the $\ell$ intervals $(\zeta_i, \zeta_{i+1})$ and satisfies the following $m-1$ continuity conditions at each interior knot $\zeta_i$ for $2 \leqslant i \leqslant \ell - 1$

$$f_X^{[j]}(\zeta_i-) = f_X^{[j]}(\zeta_i+) \qquad (0 \leqslant j < k-1). \tag{A.8}$$

Thus if $k = 2$ we impose one continuity condition at each knot, while the cubic spline ($k = 4$) requires the continuity of the derivatives up to the second order. A spline of order $k = 1$ is simply a piecewise constant function with no conditions.

For the two boundary knots $\zeta_0$ and and $\zeta_\ell$, the conditions (A.8) are too strong for practical uses, because in practice the density does not even need to be continuous there. The classical solution is to think of these boundary knots $\zeta_i$ as *multiple knots* with multiplicity $\nu_i$. Then the $k - \nu_i$ following continuity conditions must hold for knot $\zeta_i$

$$f_X^{[j]}(\zeta_i-) = f_X^{[j]}(\zeta_i+) \qquad (0 \leqslant j < k-\nu_i). \tag{A.9}$$

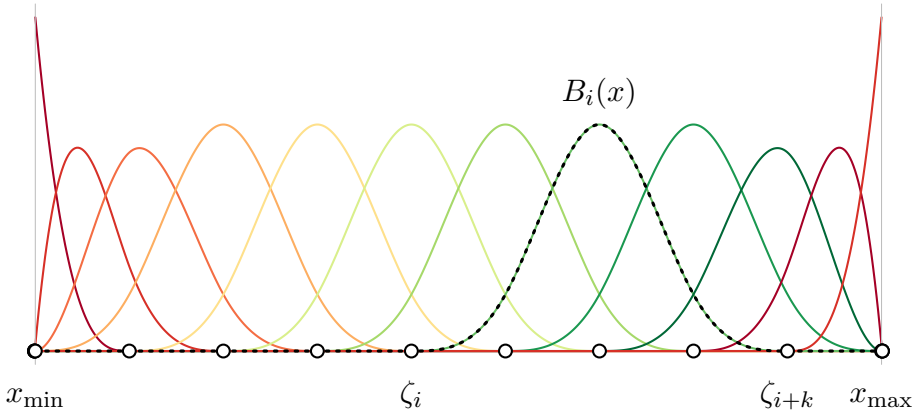


Figure A.3: B-spline basis functions for the 10-knots example, object `SD10`, which was shown on the right of figure 3.1. The locations of the knots $\zeta_i$ are shown by circles, with $\zeta_0 = x_{\min}$ and $\zeta_9 = x_{\max}$. The order is $k = 4$ (for a cubic spline). The support of a basis spline $B_i$ is the interval $(\zeta_i, \zeta_{i+k})$.

So if $\nu_i = k$ the density can be discontinuous at $\zeta_i$. At no cost in the discussion, we can assume that each of the $\ell + 1$ knots $\zeta_i$ is multiple with multiplicity $\nu_i \geqslant 1$ and continuity conditions (A.9). Thus

$$\underset{\text{number of continuity conditions}}{(k - \nu_i)} + \underset{\text{multiplicity}}{\nu_i} = \underset{\text{spline order}}{k}$$

see the book by Carl de Boor [de 01]. In the function `SplineDensity`, only the two end-knots are for now allowed to be multiple, but densities with a discontinuous derivative at an interior knot could be considered as well in the future.

### A.3.2 B-spline Basis

A spline density in **SeaLev** is coped with by using the B-spline basis related to the knots sequence. The spline density writes as

$$f_X(x) = \sum_i \alpha_i B_i(x) \tag{A.10}$$

where each $B_i(x)$ is a basis B-spline with support $(\zeta_i, \zeta_{i+k})$, see figure A.3. The basis is provided by the `splineDesign` function of the **splines** base package [R D10]. The unknown spline coefficients $\alpha_i$ of (A.10) are obtained by constrained least squares. Given a grid density, its spline approximation is found by solving a quadratic programming problem thanks to the **quadprog** package [Tur13]. The spline coefficients $\alpha_i$ are chosen in order to minimise the sum of squares of the differences between the grid and the spline densities at grid points. The constraints are equality constraints: one is for the normalisation of the density. Boundary conditions also translate into equality constraints. The normalising constraint is coped with by integrating the basis functions thanks to a recurrence relation, see [de 01, p. 128]. Of course, this method requires that a fine grid is used for the provided density.

### A.3.3 Some theoretical facts

The following results[2] are used in the `momGen` and `GPtail` functions.

[2]Which, to our best knowledge, seem new.

**Theorem 1.** *Assume that the density $f_X$ has support $[\zeta_0, \zeta_\ell]$ and has a sequence of $\ell+1$ knots $\zeta_i$ with multiplicities $\nu_i \geqslant 1$.*

- *Then the moment generating function of $X$, i.e. $M_X(t) := \mathbb{E}[e^{tX}]$ is given for $t \neq 0$ by*

$$M_X(t) = \sum_{i=0}^{\ell} \left\{ \sum_{j=k-\nu_i}^{k-1} \frac{(-1)^{j+1}}{t^{j+1}} \left[ f_X^{[j]}(\zeta_i+) - f_X^{[j]}(\zeta_i-) \right] \right\} e^{t\zeta_i}.$$

- *If $Y \sim \texttt{GPD}(\mu_Y, \sigma_Y, \xi_Y)$, and $1/\xi_Y$ is not equal to any of the integers $1, 2, \ldots, k$ then the survival function of $Z$ is given by*

$$S_Z(z) = \sum_{i=0}^{\ell} \sum_{j=k-\nu_i}^{k-1} a_j d_i^{[j]} \left[ 1 + \xi_Y \frac{z-\zeta_i}{\sigma_Y} \right]^{-1/\xi_Y + j + 1} \tag{A.11}$$

*for all real $z > \zeta_\ell$, where the coefficients $d_i^{[j]}$ are given by*

$$d_i^{[j]} := \left[ f_X^{[j]}(\zeta_i+) - f_X^{[j]}(\zeta_i-) \right]$$

*for $k - \nu_i \leqslant j \leqslant k-1$, and by $d_i^{[j]} := 0$ otherwise, while $a_j$ is given by*

$$a_j := \frac{(-\sigma_Y)^{j+1}}{(1-\xi_Y)\,(1-2\xi_Y)\,\ldots(1+(j+1)\xi_Y)}$$

*for $0 \leqslant j \leqslant k-1$.*

The proof of this theorem relies on recursive integrations by parts. Note that the expression for $S_Z(z)$ can be derived w.r.t. the parameters of the GPD, this allows the determination of confidence intervals for the return levels based on the delta method.

A remarkable point in the theorem is that the coefficients $a_j$ are not positive. If only simple knots are used, it can be shown that $\sum_i a_i \zeta_i^j = 0$ for $0 \leqslant j \leqslant k-1$ hence the coefficients $a_j$ sum to zero.

### A.3.4 Practical consequences

The formulas and the extension of the theorem above are used in the `GPDtail` and `momGen` functions.

The first statement of the theorem can be used for exponential surges, since the cumulant generating function for the tide gives the shift between the sea level tail and the surge tail, see section 1.9.1.

The second statement of the theorem is used for the general case $\xi_Y \neq 0$. It allows the accurate determination of the sea level tail distribution without using discrete convolution or quadratures. Since the formula for $S_Z(z)$ only works for $z > x_{\max} + \mu_Y$, a discrete convolution must be used to compute the bulk of the distribution.

# Appendix B

# Validation and special cases

## B.1 Exponential Surges

When a POT model with exponential distribution is used for the surge $Y$, the exact (conditional) distribution of $Z$ is known. More precisely, conditional on $Z > x_{\max} + \mu_Y$ the random variable $Z$ then follows an exponential distribution with shape $\sigma_Z = \sigma_Y$ and location $\mu_Z = \mu_X^\star + \mu_Y$ where $\mu_X^\star$ was given in (1.10) in section 1.9.1.

This result allows a simple check of the computation. The `show.asympt` argument of the `convSL` function allows us to add the theoretical return level curve to the one computed by convolution. As an example, we can use the computation for Brest. With the threshold $u = 50$ cm, the surge excesses can be considered as exponentially distributed with scale $\sigma_Y = 10$ cm, i.e. with rate $1/10.0\,\mathrm{cm}^{-1}$.

```
theta2.y <- c("rate" = 0.10)
conv.asympt <- convSL(dens.x = Brest.tide,
                      threshold.y = 50,
                      distname.y = "exponential",
                      lambda = lambda,
                      par.y = theta2.y,
                      show.asympt = TRUE,
                      Tlim = c(5, 1000000),
                      main = "Asymptotic curve: exponential Y")
```

It can bee seen on the left panel of figure B.1 that the return level curve computed by convolution nearly coincides with the exact result.

## B.2 GPD surges

When a POT model with GPD distribution is used for the surge $Y$, the exact (conditional) distribution of $Z$ is no longer known, but the asymptotic behaviour of the survival $S_Z(z)$ is known.

When $\xi_Y > 0$ it can be shown that $S_Z(z)/S_Y(z)$ tends to 1 when $z \to \infty$, but with a *very slow* convergence. The return level curve should broadly behave as if $Z$ was GPD with parameters $\sigma_Z = \sigma_Y$ (scale), and $\xi_Z = \xi_Y$ (shape). In view of the exponential case $\xi_Y = 0$, the location parameter $\mu_Z$ can be chosen as $\mu_X^\star + \mu_Y$ where $\mu_X^\star$ was defined in (1.10). Note that the tail equivalence does not imply that the difference between the return levels $z(T)$ and $y(T)$ tends to zero nor even to a finite limit.

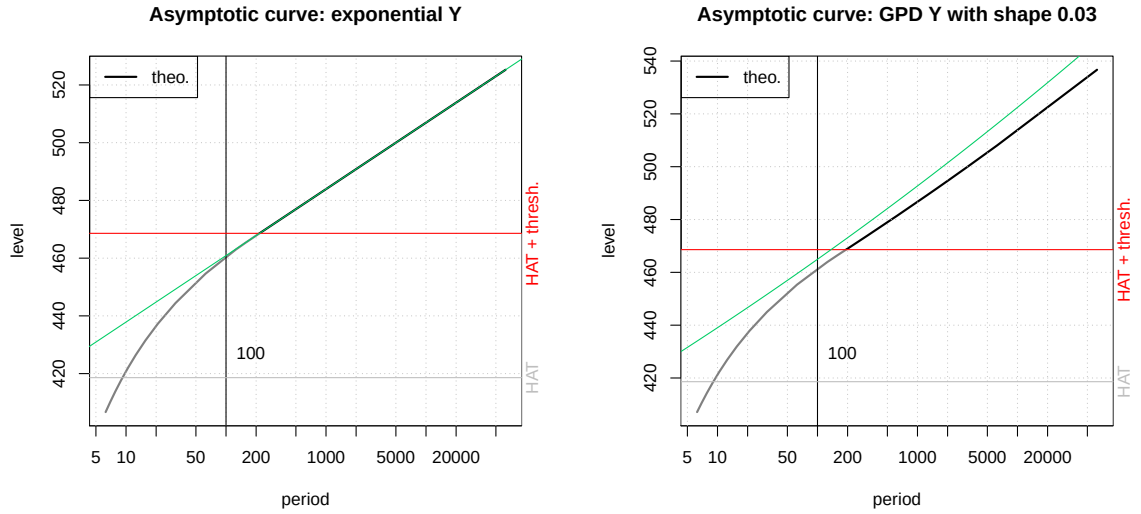


Figure B.1: Adding the asymptotic return level curve (in green). Left panel: exponential, right panel GPD with shape $\xi_Y > 0$.

```
theta3.y <- c("scale" = 10, "shape" = 0.03)
conv.asympt <- convSL(dens.x = Brest.tide,
                      threshold.y = 50,
                      distname.y = "GPD",
                      lambda = lambda,
                      par.y = theta3.y,
                      show.asympt = TRUE,
                      Tlim = c(5, 1000000),
                      main = sprintf("Asymptotic curve: GPD Y with shape %4.2f",
                          theta3.y["shape"]))
```

The plot is on the right panel of figure B.1.

## B.3 Comparing several computations

From **SeaLev >= 0.4-2** the function `convSL` function no longer uses a discrete convolution for the whole range of values of $z$. Rather, the discrete convolution is used for a limited range of $z$ values and quadratures are used for (sparse) large values of $z$. It is easily checked that the return periods computed by the two methods are – as expected – smoothly connected at the maximal value $z$ for which the discrete convolution is used (sse figure A.2). Moreover, the spline-based derivation of the tail when $Y$ has a GP distribution provides a new validation. Indeed, the computations based on the closed formula are completely independent of those arising from quadratures. It was checked that the return levels and the confidence limits are nearly identical in the two cases.

# Bibliography


[Col90] Coles S. and J.A. Tawn J.A. Statistics of coastal flood prevention. *Philosphical Transactions of the Royal Society of London, series A*, 332(1627):457–476, 1990.

[Col01] Coles, S. *An Introduction to Statistical Modelling of Extreme Values*. Springer, 2001.

[Col05] Coles, S. and Tawn, J.A. Bayesian modelling of extreme surges on the UK east coast. *Phil. Trans. R. Soc. London A*, 363:1387–1406, 2005.

[de 01] de Boor, C. *A Practical Guide to Splines, Revised Edition.* Springer-Verlag, 2001.

[Dix94] Dixon, M.J. and Tawn, J.A. Extreme Sea Levels at the UK A-Class Sites. Technical Report 65, Proudman Oceanographic Laboratory, 1994.

[Pap14] Papp, D. and Alizadeh, F. Shape-Constrained Estimation Using Nonnegative Splines. *J. Comput. Graph. Statist.*, 23(1):211–231, 2014.

[Pug79] Pugh, D.T and Vassie, J.M. Extreme sea levels from tide and surge probability. In *Proceedings of the 16th Coastal Engineering Conference, 1978*, volume 1, pages 911–930. American Society of Civil Engineers, 1979.

[Pug80] Pugh, D.T and Vassie, J.M. Applications of the joint probability method for extreme sea level computations. *Proc. Inst. Civil Eng. Part 2.*, 69:959–975, 1980.

[Pug87] Pugh, D.T. *Tides, Surges and Mean Sea-Level*. John Wiley, 1987.

[Pug04] Pugh, D.T. *Changing Sea Levels*. Cambridge University Press, 2004.

[R D10] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2010. ISBN 3-900051-07-0.

[Sim07] Simon, B. *La marée océanique et côtière*. Institut Océanographique, 2007.

[Ste02] Stephenson, A.G. evd: Extreme Value Distributions. *R News*, 2(2):0, June 2002.

[Ste16] Stephenson, A.G. *Harmonic Analysis of Tides Using TideHarmonics*, 2016. R package version 0.1.0.

[Tur13] Turlach A.B. and Weingessel A. *quadprog: Functions to solve Quadratic Programming Problems*, 2013. R package version 1.5-5.

# Index