the **SeaLev** package
user guide

Yves Deville

March 2, 2016, SeaLev version 0.4.2

Contents

1 Introduction **1**
- 1.1 Outlook . . . 1
  - 1.1.1 Goals . . . 1
  - 1.1.2 Limitations . . . 1
- 1.2 Context . . . 2
  - 1.2.1 Notations . . . 2
  - 1.2.2 Tidal X . . . 2
  - 1.2.3 Non-tidal component Y (surge) . . . 3
  - 1.2.4 Probability versus theoretical frequency . . . 3
- 1.3 Convolution and POT . . . 4
- 1.4 Return periods and return levels . . . 4
- 1.5 Inference based on the delta-method . . . 5
  - 1.5.1 Principle . . . 5
  - 1.5.2 Return levels . . . 5
- 1.6 Bayesian inference (Monte-Carlo) . . . 5
- 1.7 Expectation of the tide conditional on the Sea Level . . . 6
- 1.8 Return level plot . . . 6
- 1.9 Special case: exponential surges . . . 7

2 Using the `convSL`function **8**
- 2.1 Goals . . . 8
- 2.2 Non-parametric density of X . . . 8
- 2.3 Convolution . . . 9
  - 2.3.1 Specifying parameters for Y . . . 9
  - 2.3.2 Using a fitted POT model . . . 9
  - 2.3.3 Using a fitted non-POT model . . . 11
- 2.4 Predictions . . . 13
- 2.5 Adjusting the return level plot . . . 14

3 Frequently Asked Questions **16**
- 3.1 Calling `convSL` . . . 16
- 3.2 Inference . . . 16
- 3.3 Numerical precision . . . 17

A Numerical computation **18**
- A.1 Discrete convolution . . . 18
- A.2 Continuous convolution . . . 18
  - A.2.1 Grids . . . 18
  - A.2.2 Rectangles . . . 19

A.2.3 Trapezoidal and modification 20

B Validation and special cases **23**
B.1 Exponential surges 23
B.2 GPD surges 23

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 3.2.0 (2015-04-16). This is a DRAFT version. The functions calls may change in future versions.

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 book of David Pugh [Pug87], chap. 8, 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 independence assumption 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 μ_Y or σ_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

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

A similar relation can be given using the survival functions

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

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 is used 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.

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

Table 1.1: Distributions for surge POT. Some distributions 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 requirement will be that of 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 excesses $Y^\star = Y - u$ 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. We assume that **the rate λ 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 excesses 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 11 for an example using the Generalised Extreme Values (GEV) distribution. GPD and GEV distributions are provided in suitable form by the **evd** package [Ste02], available from the CRAN.

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 "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

[1]The nodal cycle, about 18.61 years

using an ergodicity condition involving an arbitrary two-variables test function $\phi(x,\, y)$

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

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 λ 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 λ 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 λ 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 $\lambda = 705.8\,\text{year}^{-1}$.

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 $T_Z(z_{\mathsf{max}}) = +\infty$.

We may alternatively be concerned with the return level $z(T)$ corresponding to a given return period, e.g. $T = 1000\,\text{years}$. 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 μ_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

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

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

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

The partial derivative in the integral is computed numerically using a finite difference. The partial derivative for $S_Z(z)$ is then obtained by convolution.

1.5.2 Return levels

For a fixed period T, the corresponding return level z depends on $\boldsymbol{\theta}_Y$ and λ, and therefore should be noted $z(T; \boldsymbol{\theta})$. Indeed in the equation (1.6) the left hand 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 λ 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

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

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 values 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** plots a distribution as a curve with points $[T,\, z(T)]$ where $z(T)$ is obtained by (1.7). It uses a logarithmic scale 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 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**.

When experimental points are to be added to the plot, the two formals `z` and `duration` are needed in the functions to compute the plotting positions. In the simplest case, `z` is a numeric vector and `duration` is a positive numeric value representing a duration in years. The

values in $\mathtt{z}$ are assumed to be the r-largest values Z_k of the sea level during a period having the specified duration. The underlying total number of seal levels n_{H} is the number of high tides corresponding to the yearly rate $\lambda_{\mathsf{H}} = 705.8\,\mathrm{year}^{-1}$. Assuming that the Z_k are in decreasing order, the return period $\widetilde{T}_k$ used for Z_k is such that $1/(\lambda_{\mathsf{H}}\widetilde{T}_k)$ is the estimated probability of exceedance of Z_k, i.e.

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

where w is the duration in years. For instance if $w = 10\,\mathrm{years}$, the largest experimental level Z_1 is considered as the largest value among $n_{\mathsf{H}} = 705.8 \times 10 = 7058$ 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. It is possible to specify several vectors using a list for $\mathtt{z}$, the duration being then a vector or a list with the same length as $\mathtt{z}$. See section 2.5 page 14 for an example.

1.9 Special case: exponential surges

A special case of interest is when Y has an exponential distribution with location μ_Y and scale σ_Y. It turns out then that the distribution of Z conditional on $Z > x_{\max} + \mu_Y$ is also exponential. More precisely for $z > x_{\max} + u$ 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. We also have $\mathbb{E}\,(X \mid Z = z) = \mu_X^\star$ for $Z^\star := \mu_X^\star + Y$. In other words, the return levels of Z are identical to those that would be obtained with a constant astronomical tide $X \equiv \mu_X^\star$. 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.

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

Remark. A comparable approximation is used in [CT90] for annual maxima of sea level considered as GEV. The impact of the tide on the distribution of annual maxima is a shift which is the mean value of $\sigma \exp(X_t/\sigma)$ where σ is the GEV scale parameter which plays the same role as the GPD scale for the tail. In view of (1.4) for above for $\phi := \exp$, the two shifts can be compared.

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 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 must be 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 determinated as the range of the `x` elements.

The dataset `Brest.tide` from `SeaLev` provides an example of estimated density for high-tide sea levels in Brest.

```
R> library(SeaLev)
R> convSL <- convSL2
R> data(Brest.tide)
R> class(Brest.tide)
[1] "list"
R> 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 ...
R> plot(Brest.tide, col = "SeaGreen", type = "l",
        main = "Density of high-tide sea level in Brest")
R> grid(); abline(h = 0)
```

The `plot` function call and subsequent graphics calls produce the plot on the left of the figure 2.1.

The level X is given in centimetres, the density values are accordingly in 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.

```
R> c(Brest.tide$y[1], Brest.tide$y[length(Brest.tide$y)])   ## values at end-points
[1] 0 0
R> h <- diff(Brest.tide$x)[1]                               ## grid step
R> h*sum(Brest.tide$y)                                      ## check integral
[1] 1
```

This 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 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`

```
R> u <- 50
R> theta.y <- list(scale = 10, shape = -0.01)
R> lambda <- 1.6
```

Note that we can use a numeric vector created with the `c` function rather than a `list`, but in both cases `names` must fit the parameters of the distribution

```
R> theta.y <- c(scale = 10, shape = -0.01)
R> names(theta.y)
[1] "scale" "shape"
```

Now we can use the created objects in the arguments `threshold.y`, `par.y` and `lambda` of the convolution function.

```
R> 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 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).

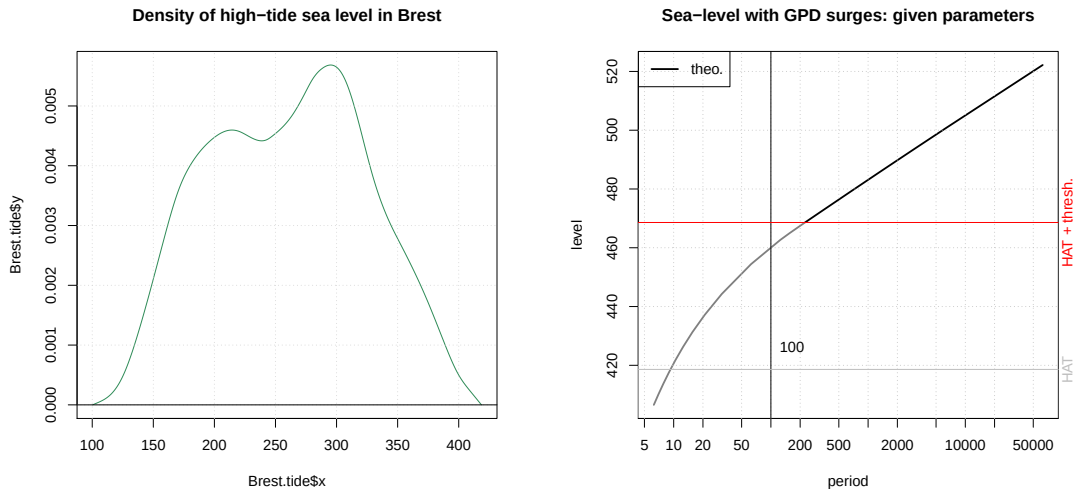

Figure 2.1: Left panel: density of the high-tide sea level X in Brest (France). Right panel: return level plot using convolution and a known GPD distribution for the surge Y. Only the part above the horizontal red line should be used, corresponding to return periods over about 200 years.

```
R> library(Renext); data(Brest)
R> fit.gpd1 <- Renouv(x = Brest$OTdata$Surge,
                      effDuration = as.numeric(Brest$OTinfo$effDuration),
                      threshold = 50, distname.y = "gpd",
                      main = "GPD surge")
R> coef(fit.gpd1)
      lambda        scale        shape
 1.612247663 10.667071264 -0.006423792
```

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.

```
R> cov1 <- vcov(fit.gpd1)
R> cov1
           lambda       scale        shape
lambda 0.01092161  0.00000000  0.000000000
scale  0.00000000  0.76269851 -0.026968031
shape  0.00000000 -0.02696803  0.002501267
```

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.

```
R> conv.gpd1 <- convSL(dens.x = Brest.tide,
                       threshold.y = 50,
                       distname.y = "gpd",
                       lambda = lambda, par.y = theta.y,
                       covpar.y = cov1,
                       main = "Sea-level for Brest with GPD surges")
```

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`.

```
R> 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 `lambda`. For instance, dropping the first row and the first column in `cov1`

```
R> cov1[-1, -1]
              scale        shape
scale  0.76269851 -0.026968031
shape -0.02696803  0.002501267
```

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.

```
R> 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: σ_Y takes the same values as in the GPD case, while μ_Y was chosen to give the same rate of exceedance over $u = 50$ cm.

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 $705.8\,\text{years}^{-1}$.

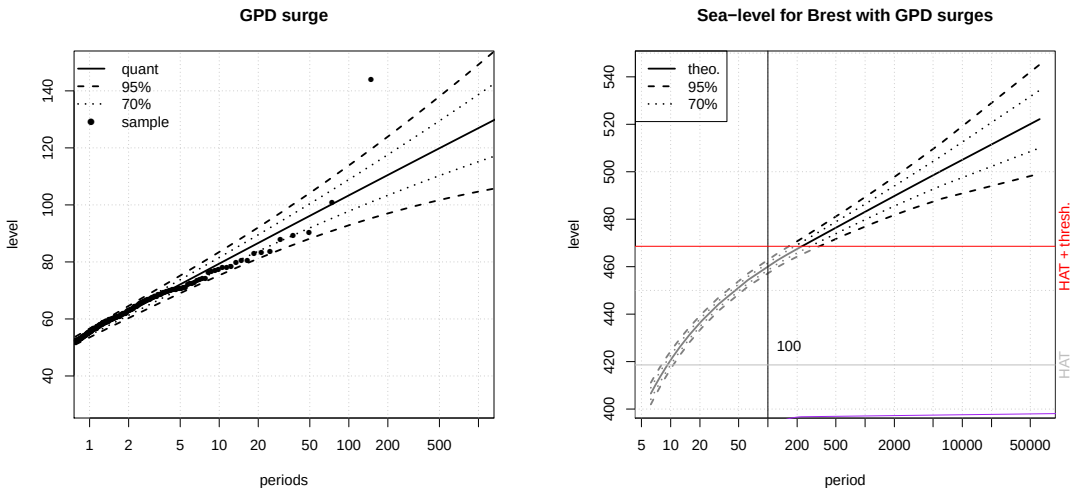

Figure 2.2: Fitting a POT model for Brest surge with **Renext** (left), and using the fitted distribution within a convolution.

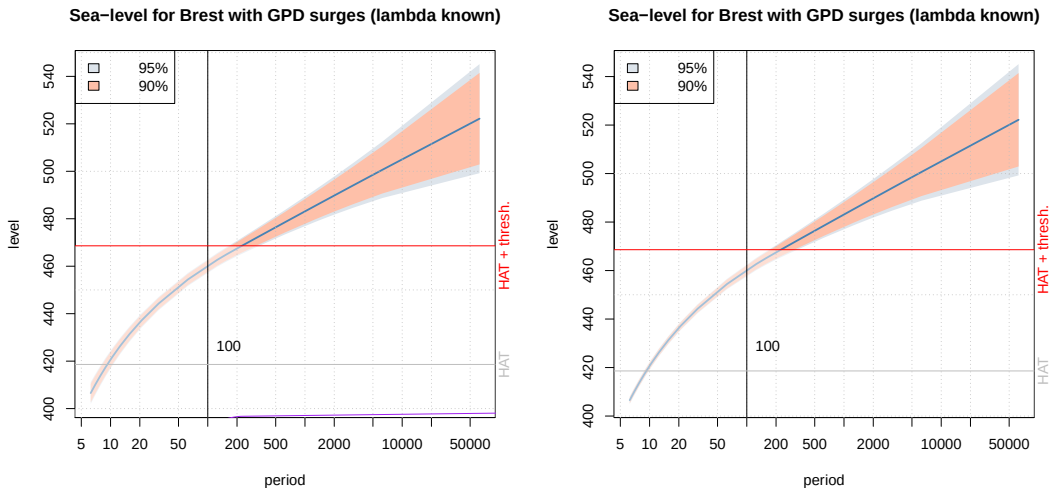

Figure 2.3: Comparison of two convolutions of the tide with the fitted GPD. On the left panel, the covariance concerns `lambda`. Right panel `use.covlambda = FALSE`.

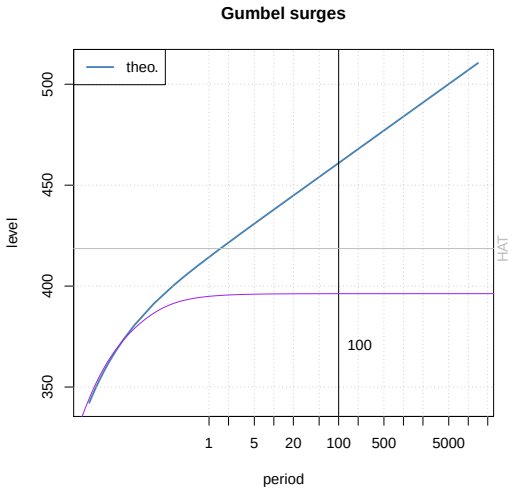

Figure 2.4: Using a Gumbel distribution (non-POT) for the surge.

```
R> par.y <- c(loc = -10.8, scale = 10)
R> 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.

2.4 Predictions

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

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

	prob	period	quant	L.95	U.95	L.90	U.90
100	0.993750	100	460.1913	457.7920	462.5905	458.1778	462.2048
200	0.996875	200	467.4752	464.5013	470.4491	464.9794	469.9710
500	0.998750	500	476.4099	471.8680	480.9519	472.5982	480.2216

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 desired periods or probabilities by using the `pred.period` and `pred.prob` formals 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 not exactly known and is affected by the uncertainty on the estimation of λ. 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 $z > u + x_{\max}$ 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 $\mathbb{E}(X \mid Z = z)$ an in

```
R> 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` or `problim`. In the first case, two limits are given in time (hence years). In the second case they are given in probability (of non-exceedance).

```
R> 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)")
```

Note that the results are recomputed but the parameters `Tlim`, `problim` are purely graphical ones and have no impact on the computed results.

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 arguments `z` and `duration` must be passed to the `RSLplot` . For instance, with meaningless points and in a purely illustrative purpose we get the plots of figure 2.6.

```
R> 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 = "Artifical empirical points (1 set)",
                    z = c(500, 490, 480, 460),
                    duration = 200)


R> 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))
```

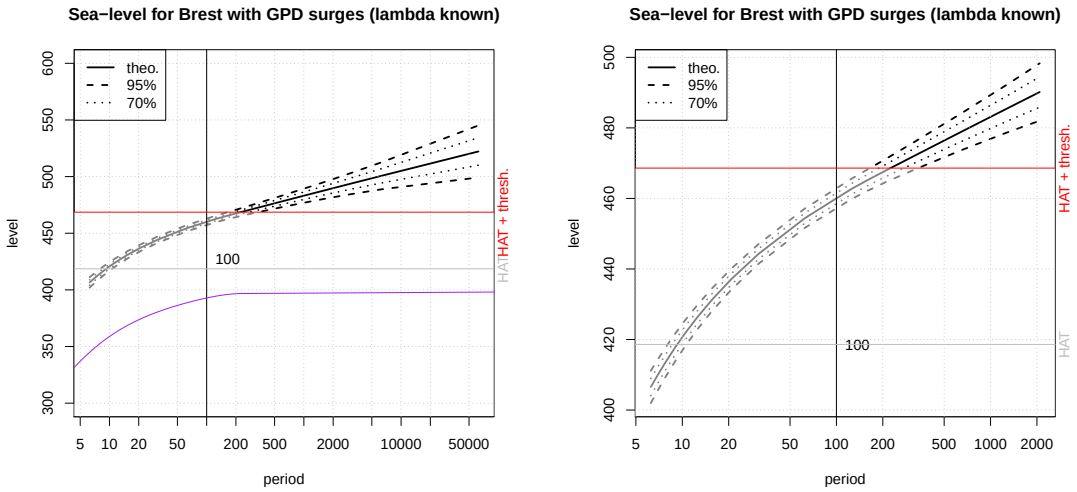

Figure 2.5: Changing the axes. Left: the `ylim` argument of `plot` was used. Right: the `Tlim` argument of `RSLplot` chooses the range of return periods, here from 100 to 5000.

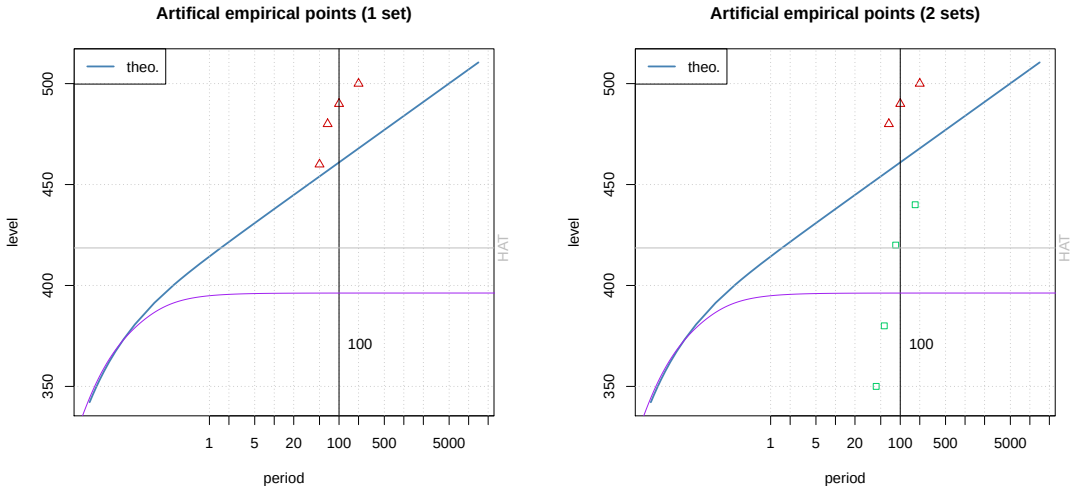

Figure 2.6: Adding empirical points to the return level plot using `z` and `duration`. When several sets are used, `z` must be a list of numeric vectors.

Chapter 3

Frequently Asked Questions

3.1 Calling `convSL`

Q. Calling `fRenouv`, 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. It 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

```
R> 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 11.

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**.

3.2 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 λ. In the estimation variance of a return level $z(T)$, the part that can be attributed to λ 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.

3.3 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 at the time. As an order of magnitude, the use of probability of exceedances about 10^{-5} seems viable for distributions of surges with exponential excesses (i.e. within the Gumbel domain of attraction).

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 λ^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)

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

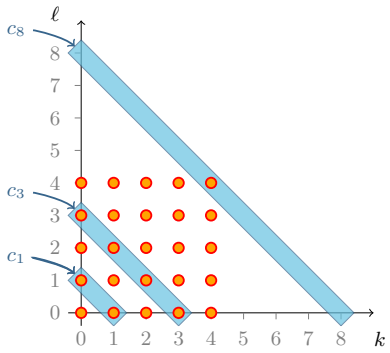

Figure A.1: Convolution of two vectors of length $N = 5$. If each point $[k, \ell]$ shows the product $a_k\, b_\ell$, then c_n comes by summation over a segment of the line $k + \ell = n$. For $n = 2N - 2$ (here $n = 8$), the sum boils down to $k = \ell = N - 1$.

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 ℓ. 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

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 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_ℓ 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.

A.2.3 Trapezoidal and modification

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}

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

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

$$f_Z(z_n) \approx \frac{1}{2h}\left\{2\sum_{k=0}^{N-1} a_k b_{n-k} - a_0 b_n + a_N b_{n-N}\right\} \tag{A.8}$$

for $0 \leqslant n \leqslant N-1$. Since $a_0 = 0$ and $a_N = 0$, the trapezoidal rule leads unsurprisingly to the same computation as the rectangles rule.

Rather than $b_k = f_Y(y_k)$, we can consider the mean value $b_k^\star$ of $f_Y(y)$ on the interval $(y_{k-1},\, y_k)$

$$b_k^\star = \frac{1}{h}\int_{y_{k-1}}^{y_k} f_Y(y)\,\mathrm{d}y = \frac{1}{h}\left\{S_Y(y_{k-1}) - S_Y(y_k)\right\} \tag{A.9}$$

for $0 \leqslant k \leqslant N-1$, with the convention $S_Y(y_k) = 1$ for $k=-1$. This choice uses the fact that $S_Y(y)$ is available for exact computation, and ensures that the vector $\mathbf{b}^\star$ has unit sum. The mean value theorem tells that

$$\int_{x_k}^{x_{k+1}} f_X(x)\, f_Y(z_n-x)\, \mathsf{d}x = f_X(\zeta_{n,k}) \times \int_{x_k}^{x_{k+1}} f_Y(z_n-x)\, \mathsf{d}x$$

for some $\zeta_{n,k}$ between x_k and x_{k+1}. Actually, $f_X(x)$ is continuous and $f_Y(z-x)$ does not change of sign on the interval. Thus

$$\int_{x_k}^{x_{k+1}} f_X(x)\, f_Y(z_n-x)\, \mathsf{d}x = f_X(\zeta_{n,k}) \times \int_{z_n-x_k}^{z_n-x_{k+1}} f_Y(y)\, \mathsf{d}y = h\, f_X(\zeta_{n,k})\, b_{n-k}^\star$$

A reasonable approximation for the unknown $f_X(\zeta_{n,k})$ is the average of the values of $f_X(x)$ at the two end-points x_k and x_{k+1}. This suggests the use of the following vectors

$$a_k^\star = \frac{1}{2}\left\{f_X(x_k) + f_X(x_{k+1})\right\}, \quad b_k^\star = \frac{1}{h}\left\{S_Y(y_{k-1}) - S_Y(y_k)\right\} \qquad 0 \leqslant k \leqslant N-1 \tag{A.10}$$

Then the approximation of the values $f_Z(z_n)$ is obtained by discrete convolution as in the rectangular case, but using $\mathbf{a}^\star$ and $\mathbf{b}^\star$ in place of $\mathbf{a}$ and $\mathbf{b}$.

Using some simple algebra, it can be shown that for $0 \leqslant n \leqslant N-1$

$$\sum_{k=1}^{N-1} a_k^\star\, b_{n-k}^\star = \frac{1}{2h}\left\{+\sum_{k=0}^{N-1} a_k\left[B_{n-k-1} - B_{n-k+1}\right] - a_0\left[B_n - B_{n+1}\right] + a_N\left[B_{n-N} - B_{n+1-N}\right]\right\} \tag{A.11}$$

where $B_k = S_Y(y_k)$ for $0 \leqslant k \leqslant N-1$ (the sequence B_k is decreasing). When the density $f_X(x)$ vanishes at its end-points, we have $a_0 = a_N = 0$ and the convolution of $\mathbf{a}^\star$ and $\mathbf{b}^\star$ can be computed as that of $\mathbf{a}$ and $\mathbf{b}^\dagger$ with

$$b_k^\dagger = \frac{1}{2h}\left\{S_Y(y_{k-1}) - S_Y(y_{k+1})\right\} \qquad 0 \leqslant k \leqslant N-1 \tag{A.12}$$

which is a centred difference approximation of $f_Y(y_k)$. Note that the right hand of the formula (A.11) is similar to that of (A.8), but each b_k is replaced by a difference approximation.

Choosing grid limits

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 $y_{\mathsf{max}} = +\infty$. Then we need to fix a suitable finite upper limit $y^\star_{\mathsf{max}}$ in the computations. The choice can be done using a small positive number $\varepsilon_{\mathsf{max}} > 0$ and solving $S_Y(y^\star_{\mathsf{max}}) = \varepsilon_{\mathsf{max}}$. Then $S_Z(z)$ can be computed for $z \leqslant x_{\max} + y^\star_{\mathsf{max}}$. Since then $S_Z(x_{\max} + y^\star_{\mathsf{max}}) \geqslant \varepsilon_{\mathsf{max}}$ the restriction on return periods will be $T \leqslant 1/(\lambda \times \varepsilon_{\mathsf{max}})$.

This strategy will be used even when the surge distribution has a finite upper end-point y_{max}, e.g. when Y is GPD with $\xi_Y < 0$. This allows the existence of an infinite density at the upper end-point, e.g. for a GPD with $\xi_Y < -1$.

One may be concerned by the case where the density $f_Y(y)$ is continuous but can be infinite at y_{min} as it is the case for the Weibull or gamma distributions with decreasing hazards. Then as before, we replace y_{min} by $y^\star_{\mathsf{min}}$ with $S_Y(y^\star_{\mathsf{min}}) = 1 - \varepsilon_{\mathsf{min}}$ and $\varepsilon_{\mathsf{min}} > 0$ is chosen small.

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.

Algorithm

$\varepsilon_{\mathsf{min}}$, $\varepsilon_{\mathsf{max}}$ are chosen small positive real numbers and N is the chosen grid length.

1. Let $H_X = x_{\max} - x_{\min}$.
2. Compute $y^\star_{\mathsf{min}} = q_Y(\varepsilon_{\mathsf{min}})$, $y^\star_{\mathsf{max}} = q_Y(1 - \varepsilon_{\mathsf{max}})$ and $H_Y = y^\star_{\mathsf{max}} - y^\star_{\mathsf{min}}$.
3. Let $H = \max(H_X, H_Y)$.
4. Fix the grid step h using $h = H/N$.
5. Let $x_0 = x_{\min}$, $y_0 = y^\star_{\mathsf{min}}$ and $z_0 = x_0 + y_0$.
6. Fill the two vectors $\mathbf{a}$ and $\mathbf{b}$ of length N with elements (A.4), using linear interpolation for $\mathbf{a}$.
7. Compute the convolution product vector $\mathbf{c}$.

The discrete convolution can rely on `convolve` from the `stats` package. This function uses the Fast Fourier Transform (FFT) and it is sound to choose N as a power of two i.e. $N = 2^L$ with L integer.

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 algorithm, the vectors $\mathbf{a}$ and $\mathbf{b}$ must be replaced by $\mathbf{a}$ and $\mathbf{b}^{\dagger}$ of (A.12) to obtain the modified trapezoidal rule.

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 $Z > x_{\max} + u$ the random variable Z then follows then an exponential distribution with shape $\sigma_Z = \sigma_Y$ and location μ_Z given by

$$\mu_Z = \mu_Y + \sigma_Y \log \mathbb{E}\left[e^{X/\sigma_Y}\right]$$

where μ_Y and σ_Y are the location and shape of Y.

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 $1/10.0\,\mathrm{cm}^{-1}$.

```
R> theta2.y <- c("rate" = 0.10)
R> 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, but with a *very slow* convergence. The return level curve should broadly behave as if Z was GPD with parameters $\mu_Z = \mathbb{E}(X) + \mu_Y$ (location), $\sigma_Z = \sigma_Y$ (scale), and $\xi_Z = \xi_Y$ (shape).

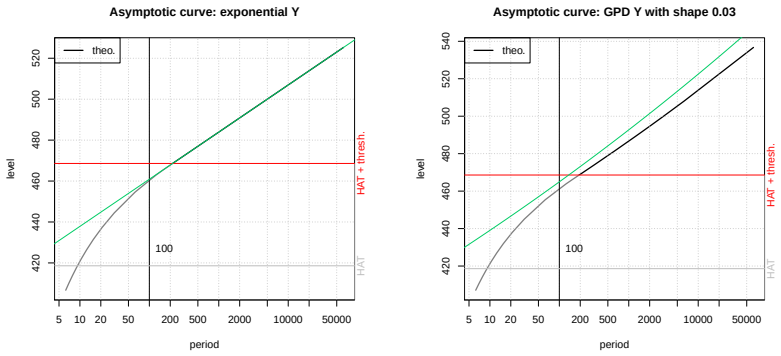

Figure B.1: Adding the asymptotic return level curve. Left panel: exponential, right panel GPD with shape $\xi_Y > 0$.

```
R> theta3.y <- c("scale" = 10, "shape" = 0.03)
R> 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. It suggests that for *very large return periods* the true curve could have a stronger convexity than the curve computed by numerical convolution.

Bibliography

[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.

[CT90] S. Coles and J.A. Tawn. Statistics of coastal flood prevention. *Philosphical Transactions of the Royal Society of London, series A*, 332(1627):457–476, 1990.

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

[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.

[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.

Index

axes, controlling the range, 14

confidence bands
    filled, 11
    percentage, 11
convolution
    algorithm, 21
    discrete, 18
    formula, 2

delta method, 5, 11

ergodicity, 3
experimental points, 6, 14

GEV (Generalised Extreme Values), 3, 7, 11
GPD (Generalised Pareto Distribution), 3
Gumbel distribution, 11

HAT (Highest Astronomical Tide), 2

non-parametric, 2

plotting positions, 6
POT (Peak Over the Threshold), 3, 9
prediction, 13

rate, sampling for high tides, 1, 4, 9
return level plot, 6, 9

skew surge, 1, 9