Computational Geometry for planar random fields

Zinedine Bounnah

Supervisor: Elena Di Bernardino

Université Côte d'Azur

June 2023

Contents

1 Introduction

Random fields have emerged as a valuable framework for analyzing spatially dependent data in various fields. Within the topic of random fields, excursion sets play a crucial role. Excursion sets are subsets of the spatial domain where the field values exceed a certain threshold, allowing researchers to focus on regions of interest. They are crucial in understanding anomalies and specific features within the field. Examples of applications in neuroscience could be found in Keith J Worsley et al. 1992, Keith J. Worsley 1996 and Shafie et al. 2003. For application in astrophysics one can look at I. Gott J. R. et al. 1990, J. R. Gott et al. 2007, Torres 1994, Vogeley et al. 1994 or Schmalzing and Górski 1998. In oceanography M. S. Longuet-Higgins 1952 and Michael Selwyn Longuet-Higgins 1957 provide some applications.

In this project, we deal with the observation and analysis of specific geometrical features, called Lipschitz-Killing curvatures, associated with the excursion sets of Gaussian, Chi-squared, and Student t random fields using the `R` programming language. We explore these curvatures for fields observed in a bounded rectangle T, as T grows to $\mathbb{R}^2$ for various threshold levels.

More precisely, the Lipschitz-Killing curvatures provide valuable insights into the geometrical features of the excursion sets. In the context of $\mathbb{R}^2$, these curvatures correspond to the Euler characteristic, half the perimeter and the Lebesgue measure of the excursion sets. By examining these curvatures, we gain a deeper understanding of the shape, connectivity, and size of the regions where the random field exceeds the specified threshold.

The organization of the project will be as follows. In Section 2, we provide definitions and examples to establish the probabilistic framework of the present manuscript before discussing the problem of the bias of the Lipschitz-Killing curvatures and we present a possible correction for a certain class of random fields (called standard, see Defintion 12). Then we illustrate the performance of these estimators of the Lipschitz-Killing curvatures for excursion sets of particular random fields. Finally in Section 3, we will delve into the computational aspects of the project.

2 Geometry of random fields

2.1 Random fields and associated excursion sets

We first start by properly defining what a random field defined on a subset of $\mathbb{R}^2$ is.

Definition 1 (Random field (see Definition 1.1.1 from Adler and Taylor 2007)). *Given a probability space $(\Omega, \mathcal{F}, P)$ and a topological space $T \subset \mathbb{R}^2$, a measurable mapping $f : \Omega \mapsto \mathbb{R}^T$ is called a random field.*

Definition 2 (Stationary and isotropic random field). *We say that a random field is strongly stationary if its law does not change under translation. If it does not change under rotation, the random field is said to be isotropic.*

We now define one of the most important type of random fields, the Gaussian one.

Definition 3 (Gaussian random field (see section 1.2 from Adler and Taylor 2007)). *Let f be a random field on $T \subset \mathbb{R}^2$. If the distribution of $(f_{t_1}, .., f_{t_n})$ is multivariate Gaussian for $(t_1, ..., t_n) \in T^n$ then f is said to be a Gaussian random field.*

A Gaussian random field is determined by its mean and covariance functions :

$m(t) = \mathbb{E}[f(t)]$, for $t \in T$

$\rho(s,t) = \mathbb{E}[(f(s) - m(s))(f(t) - m(t))]$, for $s, t \in T$

In Figure 1 we display a Gaussian random field with zero mean and covariance function $\rho(r) = \exp(-\kappa^2 r^2)$, $\kappa = \frac{100}{2^8}$ with r the Euclidean distance. Note that the covariance function here only depends on the Euclidean distance due to stationary and isotropic hypothesis.

We also introduce the notion of second spectral moment of a random field :

$\lambda = -\rho''(0)$, which can be understood as a measure of its spatial variability.

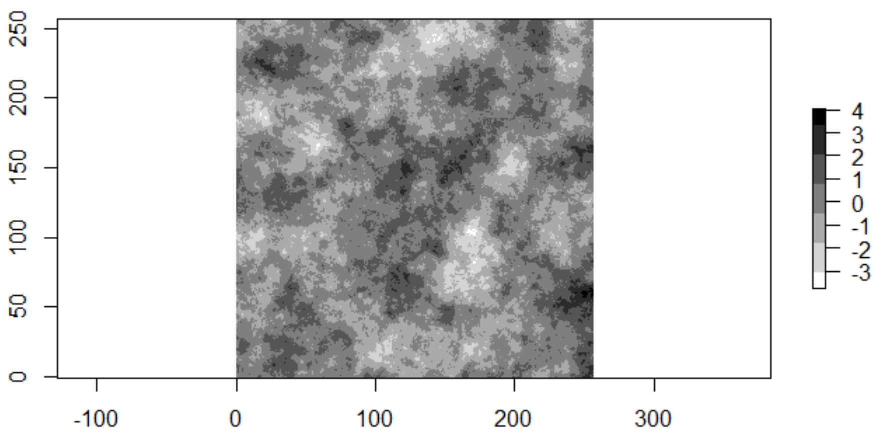

Figure 1: Gaussian field with zero mean, covariance function $\rho(r) = \exp(-\kappa^2 r^2)$, $\kappa = \frac{100}{2^8}$, second spectral moment $\lambda = 2\kappa^2$ with $T = [0; 2^8] \times [0; 2^8]$ using a discretization interval of 1 i.e. $2^8 \times 2^8$ pixels are being displayed. This random field is obtained using R and the function `RFsimulate()` from the `RandomFields` package Schlather et al. 2021 with the parameter `RMexp(scale=sqrt(2/(100/2**8)**2))`. See A.1 for more details.

Some random fields can be constructed as a function of a collection of Gaussian random fields, these fields are what we call fields of Gaussian type and we will define them properly.

Definition 4 (Fields of Gaussian type(see Definition 2.2 from Biermé et al. 2019))**.** *We call field of Gaussian type any random field $X = F(G)$, where for some $k \in \mathbb{N}^*$, $F : \mathbb{R}^k \mapsto \mathbb{R}$ is a C^2 function and $G = (G1, ..., G_k)$ is a family of i.i.d. Gaussian random fields defined on $\mathbb{R}^2$ that are C^3, stationary, isotropic, centered, with unit variance, and such that Var $G_i^{'}(0) = \lambda I_2$ for some $\lambda \geq 0$, with $G_i^{'}$ denoting the gradient of $G_i^{'}$ for all $i \in 1, ..., k$ and I_2 the 2×2 identity matrix.*

We will now provide two examples of such fields with a numerical illustration.

Example 2.1 (Chi-squared field (see 2.2.2 from Biermé et al. 2019))**.** Let $G = (G_1, ..., G_k)$ be a collection of i.i.d Gaussian random fields and $F : \mathbb{R}^k \mapsto \mathbb{R}$, $x \mapsto ||x||^2$ then F(G) is a Chi-squared field with k degrees of freedom.

A representation of a χ^2 field with zero mean and unit variance, 2 degrees of freedom and covariance function $\rho(r) = \exp(-\kappa^2 r^2)$, $\kappa = \frac{100}{2^8}$ can be found in Figure 2.

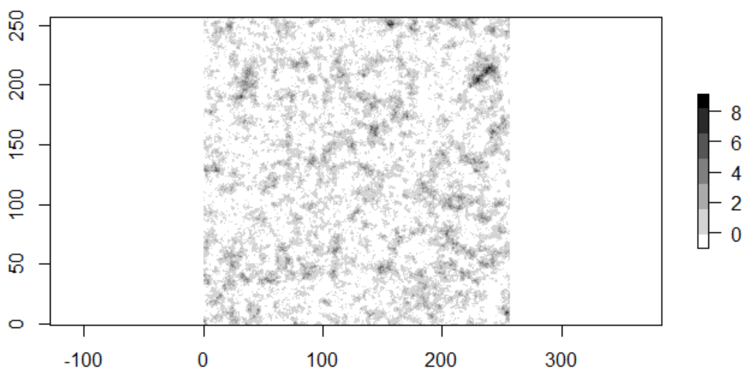

Figure 2: χ^2 field with 2 degrees of freedom, zero mean, unit variance, covariance function $\rho(r) = \exp(-\kappa^2 r^2)$, $\kappa = \frac{100}{2^8}$, second spectral moment $\lambda = 2\kappa^2$ with $T = [0; 2^8] \times [0; 2^8]$ using a discretization interval of 1 i.e. $2^8 \times 2^8$ pixels are being displayed. This random field is obtained by simulating two Gaussian fields as previously mentioned and adding the square of their values before centering and normalizing it. A.1 for more details

Example 2.2 (Student t field (see 2.2.3 from Biermé et al. 2019))**.** Let $G = (G_1, ..., G_{k+1})$ be a collection of i.i.d Gaussian random fields and $F : \mathbb{R}^{k+1} \mapsto \mathbb{R}$, $x = (x, y) \in \mathbb{R} \times \mathbb{R}^k \mapsto \frac{x}{\sqrt{||y||^2/k}}$ then $F(G)$ is a t field with k degrees of freedom.

Figure 3 displays a numerical illustration of such a field.

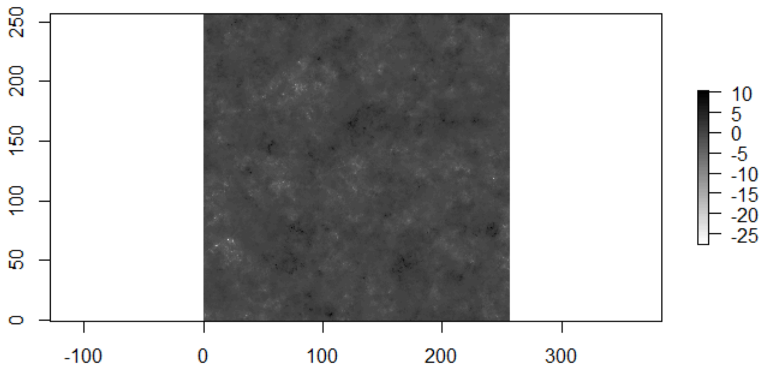

Figure 3: Student t field with 4 degrees of freedom, zero mean, unit variance, covariance function $\rho(r) = \exp(-\kappa^2 r^2)$, $\kappa = \frac{100}{2^8}$, second spectral moment $\lambda = 2\kappa^2$ with $T = [0; 2^8] \times [0; 2^8]$ using a discretization interval of 1 i.e. $2^8 \times 2^8$ pixels are being displayed. This random field is obtained by simulating 5 Gaussian fields as previously mentioned and creating a Student random variable at each point. A.1 for more details.

Having now defined the random fields we are interested in we also need to give a proper definition and numerical illustration of the excursion set of a random field within a bounded rectangle T in $\mathbb{R}^2$.

The following definition is based on the one from Biermé et al. 2019.

Definition 5 (Excursion set)**.** *Let T be a bounded rectangle in $\mathbb{R}^2$ with non-empty interior, $u \in \mathbb{R}$ and X be a random field defined on $\mathbb{R}^2$. The excursion set within T above level u can be defined as follows :*

$$\{t \in T : X(t) \geq u\} = T \cap E_X(u), \textit{where } E_X(u) := X^{-1}([u, +\infty]).$$

In Figure 4, examples of excursion set at different levels u are provided for the Gaussian field presented in Figure 1.

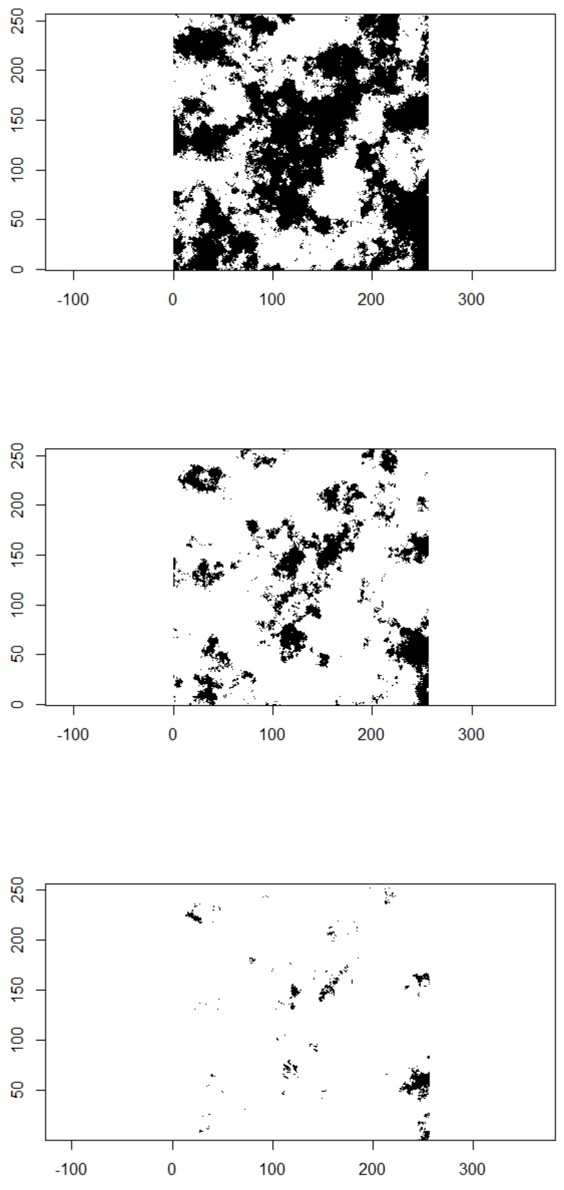

Figure 4: Excursion sets of a Gaussian field as in Figure 1 at level $u = 0$(top), $u = 1$(center), $u = 2$(bottom). More details on the code in A.1.

2.2 Lipschitz-Killing curvatures for excursion sets

Before giving a formal definition to the Lipschitz-Killing curvatures we need to define what a positive reach set is to then define Curvature Measures.

Definition 6 (Positive reach set (see Definition 1.1 from Biermé et al. 2019))**.** *For a set $A \subset \mathbb{R}^2$ and r a positive real number, let $A_r = \{x \in \mathbb{R}^2 : dist(A, x) \leq r\}$, with dist the Euclidean distance.*

Then , the reach of A is defined as :

$$reach(A) := \sup\{r \geq 0 : \forall y \in A_r, \exists a\ unique\ x \in A\ nearest\ to\ y\}.$$

The set A is said to have positive reach if $reach(A) > 0$.

A simple example of such a set could be provided by any circle; $\{x, y \in \mathbb{R}^2, (x - s)^2 + (y - t)^2 = h^2\}$, with h the radius of the circle and (s, t) the coordinates of the center of the circle.

Definition 7 (UPR (see Definition 1.3 from Biermé et al. 2019))**.** *Let UPR be the class of locally finite unions of positive reach sets.*

Definition 8 (One-dimensional Hausdorff measure $\mathcal{H}^1$, $|\ |_1$)**.** *Let (X, ρ) be a metric space, $A \subset X, \delta \in \mathbb{R}^*_+$.*

$diam(A) := \sup\{\ \rho(x, y) \mid x, y \in A\}$

Define the outer measure as follows :

$\mathcal{H}^1_\delta(A) := \inf\{\sum_{i=1}^{\infty} diam(A_i) \mid A \subset \bigcup_i A_i,\ diam(A_i) < \delta\}.$

The one dimensional Hausdorff measure of A is then :

$\mathcal{H}^1(A) = \lim_{\delta \to 0} \mathcal{H}^1_\delta(A).$

Definition 9 (Curvature measures (see Definition 1.2 from Biermé et al. 2019))**.** *For a positive reach set A and a Borel set $U \in \mathbb{R}^2$, the curvature measures $\Phi_i(A, U)$, $i = 0, 1, 2$ are :*

$\Phi_0(A, U) = \frac{TC(\partial A, U)}{2\pi}$,

$\Phi_1(A, U) = \frac{|\partial A \cap U|_1}{2}$,

$\Phi_2(A, U) = |A \cap U|$,

where $TC(\partial A, U)$ denotes the integral over U of the curvature along the positively oriented curve ∂A, $|.|$ the Lebesgue measure and $|.|_1$ the one dimensional Hausdorff measure (8).

In other words, $\Phi_0(A, U)$ denotes the Euler-Poincare characteristic, $\Phi_1(A, U)$ half the boundary length and $\Phi_2(A, U)$ the area of A.

An illustration for a simple object is provided in Figure 5.

Definition 10 (Euler-Poincare characteristic)**.** *It is a topological invariant describing the shape of a topological space.*

For a random field X defined on $T \subset \mathbb{R}^2$,

$$\Phi_0(X) = number\ of\ connected\ component - number\ of\ holes.$$

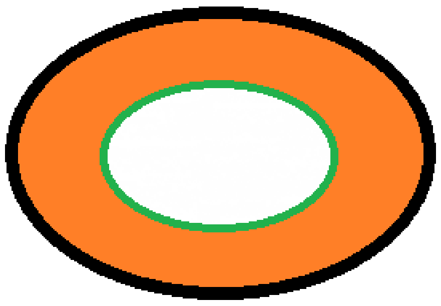

Figure 5: 2D shape. $\Phi_0 = 0$ since the shape has one connected component and one hole. $\Phi_1 =$ (length of the black curve + length of the green curve)/2 and $\Phi_2 = |orange|$, the area covered by the orange part.

Definition 11 (Lipschitz-Killing (LK) curvatures of the excursion set $E_X(u)$ within $T \subset \mathbb{R}^2$ (see Definition 1.4 from Biermé et al. 2019))**.** *Let X be a stationary random field on $\mathbb{R}^2$, T be a bounded rectangle in $\mathbb{R}^2$ with non empty interior and $T \cap E_X(u)$ be an UPR set. The LK curvatures of the excursion set $E_X(u)$ within T are then :*

$$C_i(X, u, T) := \Phi_i(T \cap E_X(u), T), \text{ for } i \ = \ 0, 1, 2.$$

The normalized LK curvatures are :

$$C_i^{/T}(X, u) := \frac{C_i(X, u, T)}{|T|}, \text{ for } i \ = \ 0, 1, 2.$$

Assuming the limits exist, the associated LK densities are :

$$C_i^*(X, u) := \lim_{T \to \mathbb{R}^2} \mathbb{E}[C_i^{/T}(X, u)], \text{ for } i \ = \ 0, 1, 2.$$

In particular, we remark that if i=2 we have :

$$C_2^*(X, u) = \mathbb{P}(X(0) \geq u).$$

This comes from the fact that we are looking here at the proportion of points being greater or equal to the level u; $E\left[|T \cap E_X(u)|\right] = |T|\mathbb{P}(X(0) \geq u)$ using Fubini-Tonelli theorem.

2.3 Statistical estimators for the Lipschitz-Killing curvatures of excursion sets

Definition 12 (Standard random field. Definition 2.1 from Biermé et al. 2019)**.** *Let X be a stationary isotropic random field defined on $\mathbb{R}^2$. We say that X is standard at level $u \in \mathbb{R}$ if $T \cap E_X(u)$ is UPR for any rectangle T in $\mathbb{R}^2$, if $C_i^*(X,u)$, for $i = 0, 1$ exist, and if*

$\mathbb{E}[C_0^{/T}(X,u)] = C_0^*(X,u) + \frac{1}{\pi} C_1^*(X,u) \frac{|\partial T|_1}{|T|} + \frac{C_2^*(X,u)}{|T|}$

$\mathbb{E}[C_1^{/T}(X,u)] = C_1^*(X,u) + \frac{1}{2} C_2^*(X,u) \frac{|\partial T|_1}{|T|}$

Some examples of such fields are the fields we are interested in for this project, *i.e.* fields of Gaussian type. (see Proposition 2.3 and its proof from Biermé et al. 2019)

We notice a bias in the expected values and therefore cannot simply use the estimated $C_i^{/T}(X,u)$. This bias comes from the fact that the random fields are observed in a bounded rectangle and therefore needs to be removed.

Following Proposition 2.5 from Biermé et al. 2019, we can compute unbiased estimators for the Lipschitz Killing densities of standard random fields as follows :

$$\hat{C}_{0,T}(X,u) = C_0^{/T}(X,u) - \frac{|\partial T|_1}{\pi |T|} C_1^{/T}(X,u) + \left(\frac{1}{2\pi} \left(\frac{|\partial T|_1}{|T|} \right)^2 - \frac{1}{|T|} \right) C_2^{/T}(X,u)$$

$$\hat{C}_{1,T}(X,u) = C_1^{/T}(X,u) - \frac{|\partial T|_1}{2|T|} C_2^{/T}(X,u))$$

$$\hat{C}_{2,T}(X,u) = C_2^{/T}(X,u)$$

3 Computation

3.1 Computation of the approximated Lipschitz-Killing curvatures

To compute the Euler Characteristic, we look at a binary raster of the excursion set of a random field and count the number of connected component and the number of holes using the `clump` function from the `raster` package (Hijmans 2023) with a connectivity of 4. A connectivity of 4 means that two pixels are considered adjacent if they are connected through edges only (Rook's case) and a connectivity of 8 allows them to be connected through their corners as well(Queen's case). To get the number of connected components we look directly at the binary raster and to get the number of holes we look at the complementary of the binary raster. Figure 6 provides an example of how the connectivity works.

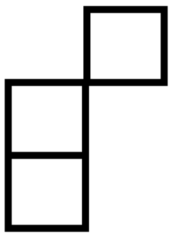

Figure 6: Example of connectivity. If a connectivity of 4 is used, this image would have 2 connected components whereas if a connectivity of 8 is used the image would only have 1 connected component since this type of connectivity allows components to be connected through their corners.

We now need to compute the perimeter of an excursion set to obtain the second Lipschitz-Killing curvatures (C_1). To do so, we first merge connected pixels together and transform them into polygons before computing the length of the boundary lines of each polygon using the `st_length` function from the `stars` package Pebesma and Bivand 2023.

Example of the process is shown in Figure 8 for the excursion set at level $u = 1$ of a Gaussian random field with covariance function $\rho(r) = \exp(-\kappa^2 r^2)$, $\kappa = \frac{100}{2^8}$ as displayed in Figure 7 .

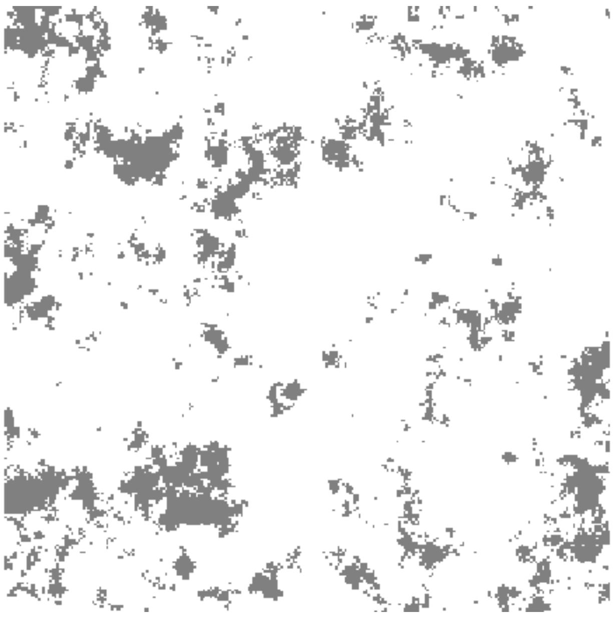

Figure 7: Excursion set at level $u = 1$ for a Gaussian random field with zero mean, covariance function $\rho(r) = \exp(-\kappa^2 r^2)$, $\kappa = \frac{100}{2^8}$ simulated in `R` on $T = [0; 2^8] \times [0; 2^8]$ using a discretization interval of 1 and using the `Excursion_set` function(see A.1).

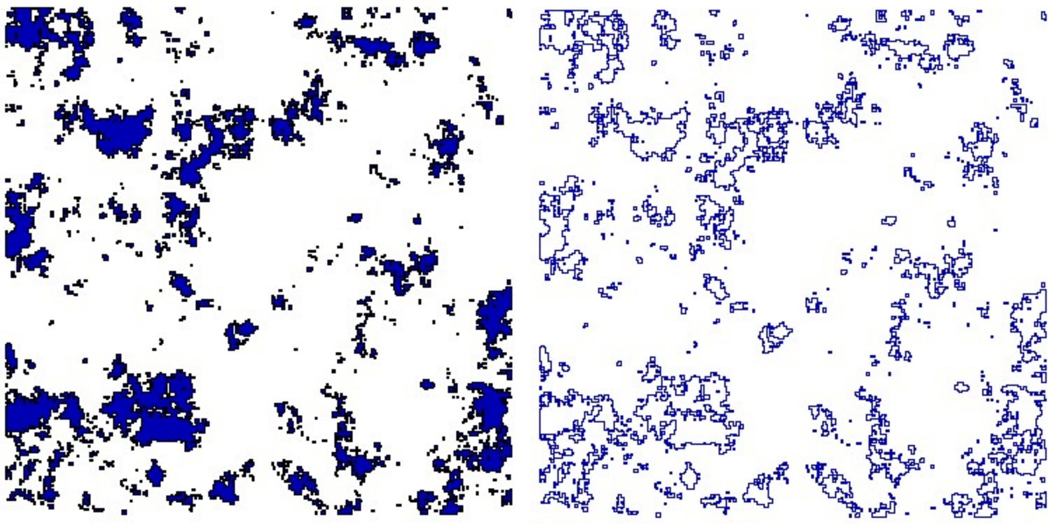

Figure 8: On the left image the transformation into polygons with the `st_as _sf` function and the parameter `merge=TRUE` is shown. On the right image only the boundary lines of each polygon is left by converting the polygons to `"MULTILINESTRING"` with the `st _as _cast` function. These functions come from the `stars` package (Pebesma and Bivand 2023)

Finally, the computation of the area of an excursion set at a given level u. This one is quite simple to obtain, as we only need to compute the proportion of black pixels multiplied by the area of the rectangle T. We first get the proportion and multiply it by the area of T to allow a discretization of the interval that can be different from $x_i = 1, 2...n$. For example if we were to look at the proportion of black pixels on a rectangle of size T and a discretization interval of 0.5 we would not be able to get the area by simply looking at the number of black pixels since we would have 4 times the amount of pixels in the image.

3.2 Applications

We will now compute the Lipschitz-Killing densities for Gaussian, Chi-squared and Student t field with Bergmann-Fock covariance function *i.e.* $\rho(r) = \exp(-\kappa^2 r^2)$. For each field we also provide the expected Lipschitz-Killing densities based on the equations provided in Biermé et al. 2019. For more details on how they are derived, see proof of Proposition 2.3 from Biermé et al. 2019 and Theorem 15.9.4 as well as 15.9.5 from Adler and Taylor 2007.

For a Gaussian random field X with zero mean and covariance function $\rho(r) = \exp(-\kappa^2 r^2)$, the expected Lipschitz-Killing curvatures of the excursion set at level $u \in \mathbb{R}$ are :

$C_0^*(X, u) = (2\pi)^{-3/2} 2\kappa^2 u e^{-u^2/2}$,

$C_1^*(X, u) = \frac{1}{4}\sqrt{2\kappa^2} e^{-u^2/2}$,

$C_2^*(X, u) = \psi(u)$ with $\psi()$ the gaussian tail distribution.

In Figure 9 an example of such a field for $\kappa = \frac{100}{2^8}$ is provided and Figure 10 displays the Lipschitz-Killing densities obtained using Monte-Carlo methods over 100 simulation for a field of this type (A.1).

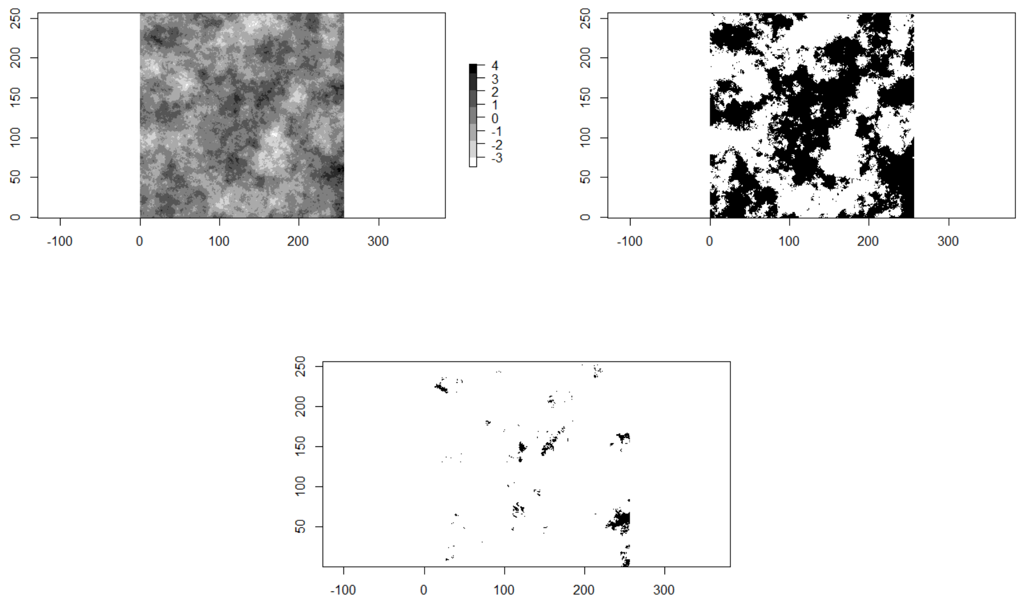

Figure 9: In the top left, a Gaussian field with zero mean, covariance function $\rho(r) = \exp\left(-\kappa^2 r^2\right)$, $\kappa = \frac{100}{2^8}$, second spectral moment $\lambda = 2\kappa^2$ with $T = [0; 2^8] \times [0; 2^8]$ using a discretization interval of 1. On the top right, the excursion set of this field at level $u = 0$. The image in the bottom displays the excursion set at level $u = 2$.

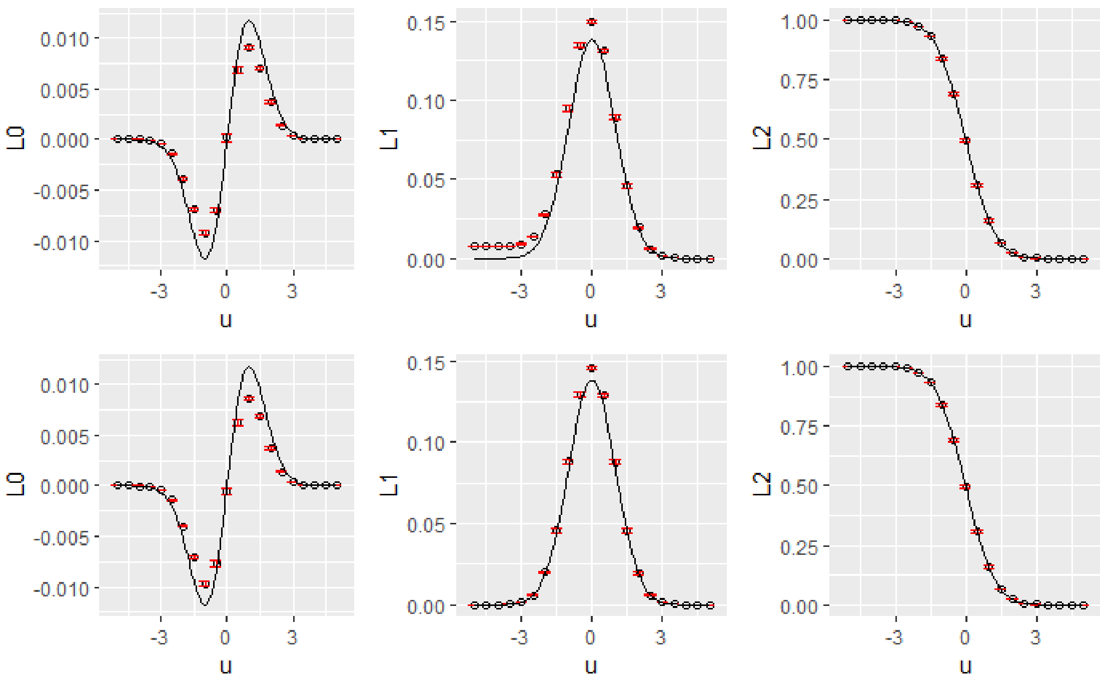

Figure 10: Lipschitz-Killing densities of the excursion set of a Gaussian field X simulated in R with zero mean, covariance function $\rho(r) = \exp(-\kappa^2 r^2)$ with $\kappa = \frac{100}{2^8}$, $T = [0; 2^8] \times [0; 2^8]$ using a discretization interval of 1 for different levels u over 100 simulations. On the first row, $C_i^{/\mathrm{T}}(X, u)$ for $i = 0, 1, 2$ are displayed while on the second row the unbiased estimators $\hat{C}_{i,T}(X, u)$ for $i = 0, 1, 2$ are displayed. The black curves are the expected Lipschitz-Killing curvatures $C_0^*(X, u)$ and in red are the confidence intervals.

By considering a random field as in Example 2.1 with $k = 2$ degrees of freedom and $\rho(r) = \exp(-\kappa^2 r^2)$, $\kappa = \frac{100}{2^8}$ we construct $Z = F(G)$, a χ^2 field with 2 degrees of freedom and we take $Z' = \frac{1}{\sqrt{2k}}(Z(t) - k), t \in \mathbb{R}^2$ to get a random field with zero mean and unit variance.

The expected Lipschitz-Killing curvatures for the excursion set at level $u \in \mathbb{R}$ of this χ^2 field are then:

$C_0^*(Z', u) = \frac{2\kappa^2(2u+1)exp(-\frac{2+2u}{2})}{2\pi\Gamma(1)}$,

$C_1^*(Z', u) = \frac{\sqrt{2\kappa^2\pi}}{2^{3/2}\Gamma(1)}(2+2u)^{1/2}exp(-\frac{2+2u}{2})$,

$C_2^*(Z', u) = \mathbb{P}(\chi^2 \geq 2 + u\sqrt{2})$.

A representation of this field as well as two excursion sets at different levels u can be seen in Figure 11 while the Lipschitz-Killing densities computed over 100 simulations can be found in Figure 12.

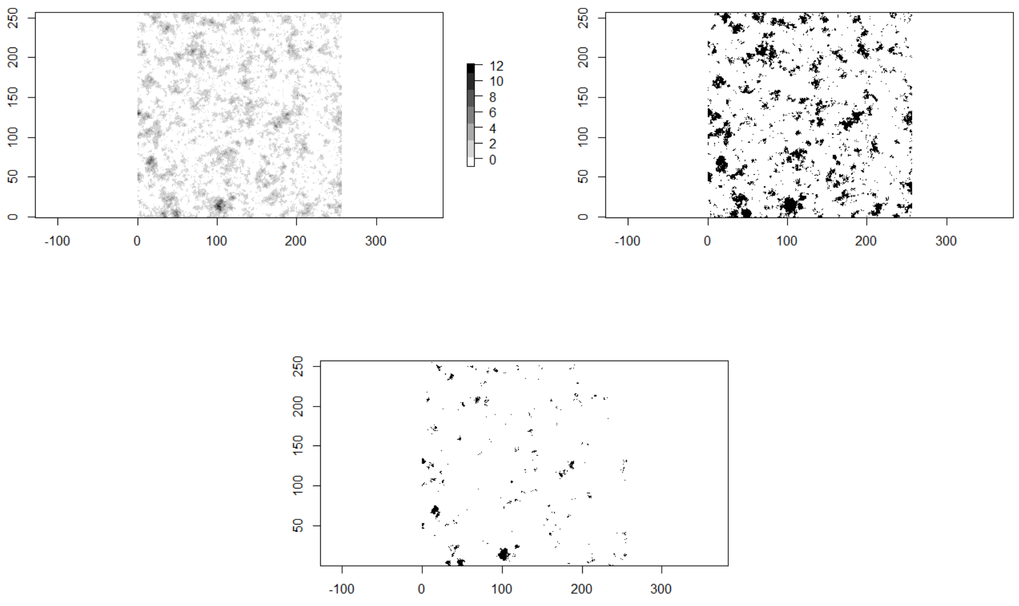

Figure 11: In the top left, a χ^2 field with 2 degrees of freedom, zero mean, unit variance, covariance function $\rho(r) = \exp(-\kappa^2 r^2)$, $\kappa = \frac{100}{2^8}$, second spectral moment $\lambda = 2\kappa^2$ with $T = [0; 2^8] \times [0; 2^8]$ using a discretization interval of 1. In the top right, the excursion set of this field at level $u = 1$. The bottom image displays the excursion set at level $u = 3$.

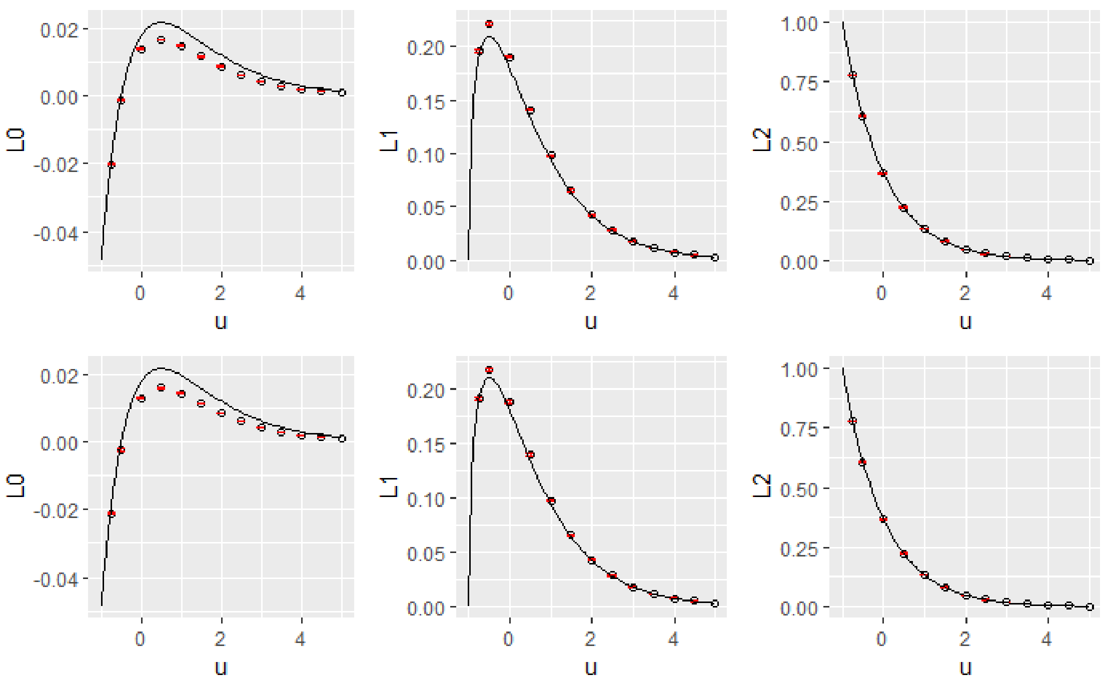

Figure 12: Lipschitz-Killing densities of the excursion set of a χ^2 field Z with 2 degrees of freedom simulated in `R` with zero mean, unit variance, covariance function $\rho(r) = \exp(-\kappa^2 r^2)$ with $\kappa = \frac{100}{2^8}$, $T = [0; 2^8] \times [0; 2^8]$ using a discretization interval of 1 for different levels u over 100 simulations. On the first row, $C_i^{/\mathrm{T}}(Z, u)$ for $i = 0, 1, 2$ are displayed while on the second row the unbiased estimators $\hat{C}_{i,T}(Z, u)$ for $i = 0, 1, 2$ are displayed. The black curves are the expected Lipschitz-Killing curvatures $C_0^*(Z, u)$ and in red are the confidence intervals.

Considering now a random field as in Example 2.2 with $k = 4$ degrees of freedom, $\rho(r) = \exp(-\kappa^2 r^2)$, $\kappa = \frac{100}{2^8}$ we construct a Student t field $S = F(G)$ with 4 degrees of freedom. Taking $S' = \sqrt{(k-2)/k}S(t)$, $t \in \mathbb{R}^2$ allows us to deal with a field with zero mean and unit variance.

The expected Lipschitz-Killing curvatures for the excursion set at level $u \in \mathbb{R}$ of a Student t field of this type are then:

$C_0^*(S'_k, u) = \frac{6\kappa^2}{4\pi^{3/2}} \frac{u}{\sqrt{2}} \frac{\Gamma(1)}{\Gamma(\frac{3}{2})} (1 + u^2)^{-1}$,

$C_1^*(S'_k, u) = \frac{\sqrt{2}\kappa^2}{4} (1 + \frac{u^2}{2})^{-3/2}$,

$C_2^*(S'_k, u) = \mathbb{P}(Student(4) \geq u\sqrt{2})$.

A representation of this type of Student t field alongside two excursion sets are displayed in Figure 13 while the Lipschitz-Killing densities computed over 100 simulations can be found in Figure 14.

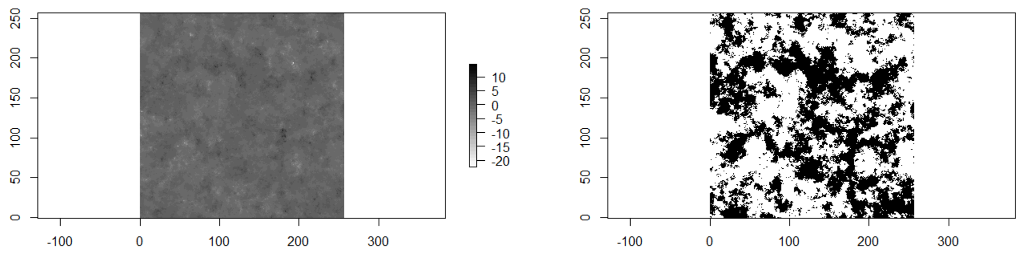
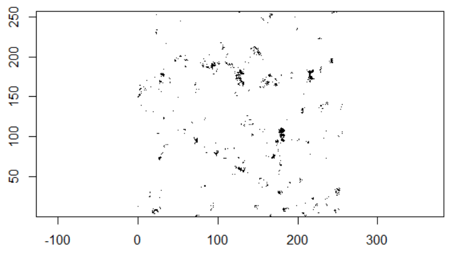
Figure 13: In the top left, a Student t field with 4 degrees of freedom, zero mean, unit variance, covariance function $\rho(r) = \exp(-\kappa^2 r^2)$, $\kappa = \frac{100}{2^8}$, second spectral moment $\lambda = 2\kappa^2$ with $T = [0; 2^8] \times [0; 2^8]$ using a discretization interval of 1. In the top right, the excursion set of this field at level $u = 0$ while the image in the bottom displays the excursion set at level $u = 3$.

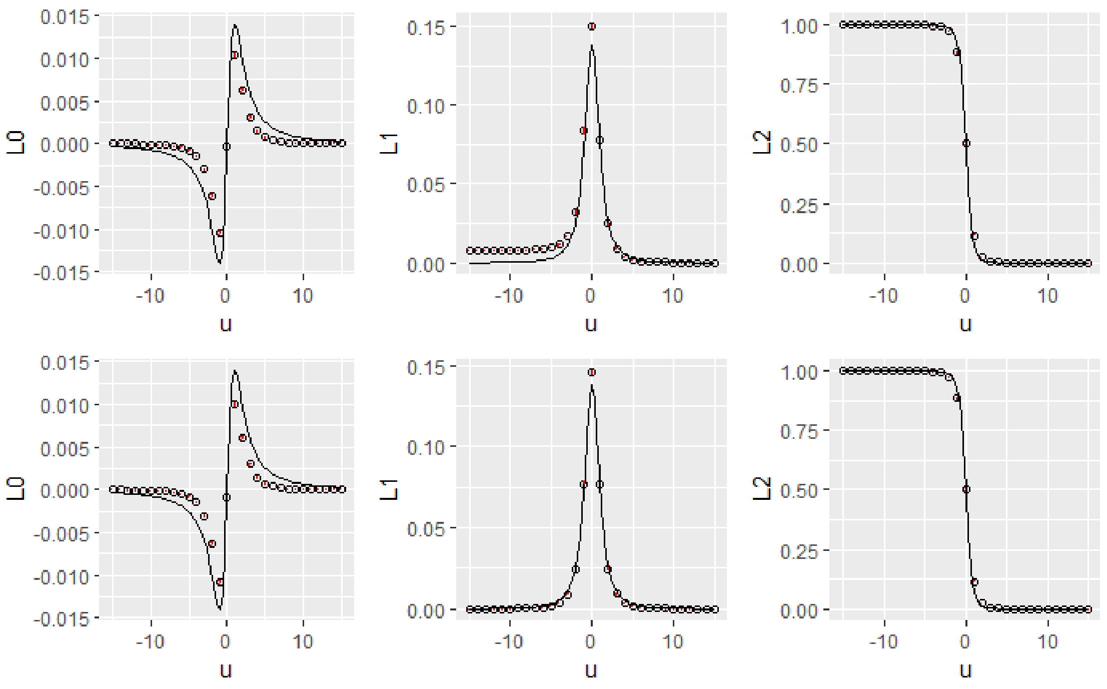
Figure 14: Lipschitz-Killing densities of the excursion set of a Student t field S with 4 degrees of freedom simulated in `R` with zero mean, unit variance, covariance function $\rho(r) = \exp(-\kappa^2 r^2)$ with $\kappa = \frac{100}{2^8}$, $T = [0; 2^8] \times [0; 2^8]$ using a discretization interval of 1 for different levels u over 100 simulations. On the first row, $C_i^{/\mathrm{T}}(S, u)$ for $i = 0, 1, 2$ are displayed while on the second row the unbiased estimators $\hat{C}_{i,T}(S, u)$ for $i = 0, 1, 2$ are displayed. The black curves are the expected Lipschitz-Killing curvatures $C_0^*(S, u)$ and in red are the confidence intervals.

As one could have noticed looking at these illustrations, while the perimeter and area seem to be well computed, the Euler Characteristic exhibits a small deviation from theoretical values. This deviation might be due to the discretization of T as using a smaller discretization interval yields better results regarding the Euler Characteristic but with a trade-off in terms of the half perimeter. Figure 16 shows one such example using a discretization interval of 0.75 over 100 simulations of Gaussian fields, as shown in Figure 15.

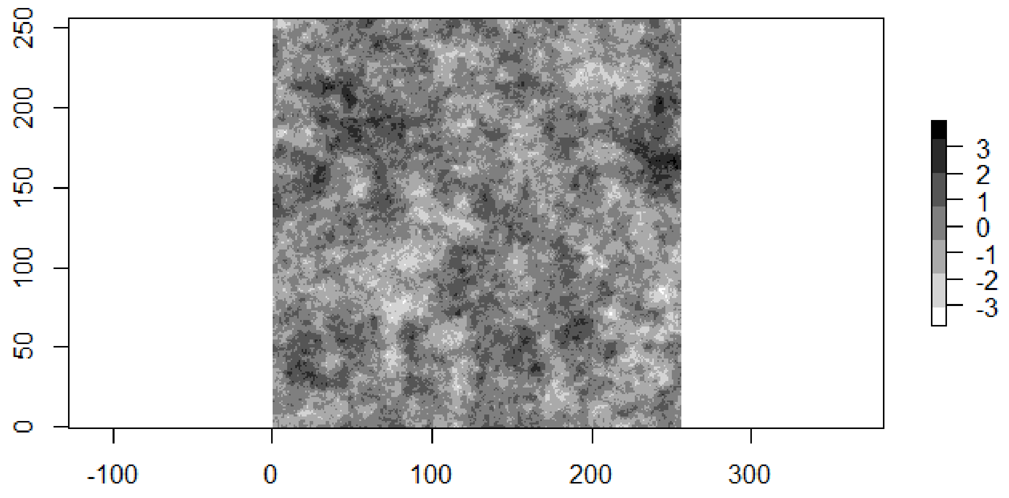

Figure 15: Gaussian field with zero mean, covariance function $\rho(r) = \exp(-\kappa^2 r^2)$, $\kappa = \frac{100}{2^8}$, second spectral moment $\lambda = 2\kappa^2$ with $T = [0; 2^8] \times [0; 2^8]$ using a discretization interval of 0.75. This random field is obtained using R and the function `RFsimulate()` from the `RandomFields` package Schlather et al. 2021 with the parameter `RMexp(scale=sqrt(2(100/2**8)**2))`.

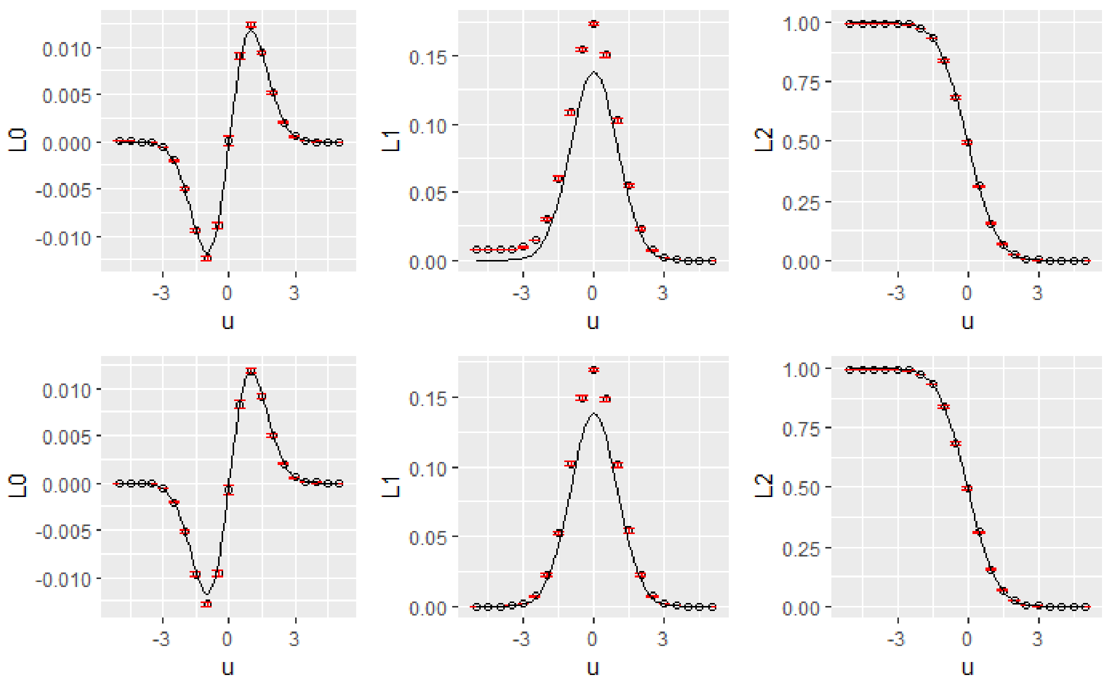

Figure 16: Lipschitz-Killing densities of the excursion set of a Student t field S with 4 degrees of freedom simulated in `R` with zero mean, unit variance, covariance function $\rho(r) = \exp(-\kappa^2 r^2)$ with $\kappa = \frac{100}{2^8}$, $T = [0; 2^8] \times [0; 2^8]$ using a discretization interval of 0.75 for different levels u over 100 simulations. On the first row, $C_i^{/\mathrm{T}}(S, u)$ for $i = 0, 1, 2$ are displayed while on the second row the unbiased estimators $\hat{C}_{i,T}(S, u)$ for $i = 0, 1, 2$ are displayed. The black curves are the expected Lipschitz-Killing curvatures $C_0^*(S, u)$ and in red are the confidence intervals.

References

[1] Robert J. Adler and Jonathan E. Taylor. *Random Fields and Geometry.* Springer, 2007.

[2] Hermine Biermé et al. “Lipschitz-Killing curvatures of excursion sets for two-dimensional random fields”. In: *Electronic Journal of Statistics* 13 (2019). hal-03080411, pp. 536–581. DOI: 10.1214/19-ejs1530.

[3] III Gott J. R. et al. “Topology of microwave background fluctuations - Theory”. In: *Astrophysical Journal, Part 1* 352 (1990). Research supported by New Jersey High Technology Grant. DOI: 10.1086/168511.

[4] J. Richard Gott et al. “Genus topology of the cosmic microwave background from the WMAP 3-year data”. In: *Monthly Notices of the Royal Astronomical Society* 377.4 (June 2007), pp. 1668–1678. DOI: 10.1111/j.1365-2966.2007.11730.x.

[5] Robert J. Hijmans. *raster: Geographic Data Analysis and Modeling.* R package version 3.6-20. 2023. URL: https://CRAN.R-project.org/package=raster.

[6] M. S. Longuet-Higgins. “On the Statistical Distribution of the Heights of Sea Waves”. In: *Journal of Marine Research* 11.3 (1952).

[7] Michael Selwyn Longuet-Higgins. “The statistical analysis of a random, moving surface”. In: *Philosophical Transactions of the Royal Society A* 249.966 (Feb. 1957), pp. 321–387. ISSN: 0080-4614. DOI: 10.1098/rsta.1957.0002. URL: https://doi.org/10.1098/rsta.1957.0002.

[8] Edzer Pebesma and Roger Bivand. *Spatial Data Science: With applications in R.* London: Chapman and Hall/CRC, 2023, p. 352. DOI: 10.1201/9780429459016. URL: https://r-spatial.org/book/.

[9] Martin Schlather et al. *RandomFields: Simulation and Analysis of Random Fields.* R package version 3.3.13. 2021. URL: https://cran.r-project.org/package=RandomFields.

[10] Jens Schmalzing and Krzysztof M. Górski. “Minkowski functionals used in the morphological analysis of cosmic microwave background anisotropy maps”. In: *Monthly Notices of the Royal Astronomical Society* 297.2 (June 1998), pp. 355–365. DOI: 10.1046/j.1365-8711.1998.01467.x.

[11] K. Shafie et al. “Rotation space random fields with an application to fMRI data”. In: *Annals of Statistics* 31.6 (Dec. 2003), pp. 1732–1771. DOI: 10.1214/aos/1074290326.

[12] Sergio Torres. “Topological Analysis of COBE-DMR Cosmic Microwave Background Maps”. In: *Astrophysical Journal Letters* 423 (Mar. 1994), p. L9. DOI: 10.1086/187223. URL: https://doi.org/10.1086/187223.

[13] Michael S. Vogeley et al. “Topological Analysis of the CfA Redshift Survey”. In: *The Astrophysical Journal* 420.2 (1994), p. 525. DOI: 10.1086/173583. URL: https://doi.org/10.1086/173583.

[14] Keith J Worsley et al. “A three-dimensional statistical analysis for CBF activation studies in human brain”. In: *Journal of Cerebral Blood Flow & Metabolism* 12.6 (Nov. 1992). DOI: 10.1038/jcbfm.1992.127.

[15] Keith J. Worsley. “The Geometry of Random Images”. In: *CHANCE* 9.1 (1996).

A Appendix

A.1 Code R

```r
# Load the required package
library(RandomFields)
library(raster)
library(RColorBrewer)
library(plotly)
library(stars)
library(ggplot2)
library(cowplot)

## GAUSSIAN

## define the locations:
from <- 0
to <- 2^8
x.seq <- seq(from, to, length=to)
y.seq <- seq(from, to, length=to)

ka=(100/2^8)^2

# Simulate the random field
rf=RFsimulate(RMexp(scale=sqrt(2/ka)), x=x.seq,y=y.seq)

# white/black color palette
pal <- colorRampPalette(c("white","black"))
plot(raster::raster(rf),col=pal(7))




##CHI 2 with 2 degrees of freedom
ka=(100/2^8)^2
# Simulate 2 gaussian random fields
rf1=RFsimulate(RMexp(scale=sqrt(2/ka)), x=x.seq,y=y.seq)
rf2=RFsimulate(RMexp(scale=sqrt(2/ka)), x=x.seq,y=y.seq)

# sum the squared values of these 2 random fields to get a Chi 2 random field
rf1@data[["variable1"]]=rf1@data[["variable1"]]^2+rf2@data[["variable1"]]^2

#unit variance
rf1@data[["variable1"]]=0.5*(rf1@data[["variable1"]]-2)

# Define color palette for plotting
pal <- colorRampPalette(c("white", "black"))

# Convert simulated random field to a raster object and plot it
r <- raster(rf1)

plot(r, col = pal(7))
```

```
k=4
ka=(100/2^8)^2
# Simulate the Gaussian random fields
rf=RFsimulate(RMexp(scale=sqrt(2/ka)), x=x.seq,y=y.seq)
rf1=RFsimulate(RMexp(scale=sqrt(2/ka)), x=x.seq,y=y.seq)
rf2=RFsimulate(RMexp(scale=sqrt(2/ka)), x=x.seq,y=y.seq)
rf3=RFsimulate(RMexp(scale=sqrt(2/ka)), x=x.seq,y=y.seq)
rf4=RFsimulate(RMexp(scale=sqrt(2/ka)), x=x.seq,y=y.seq)

# Create a chi2 one
chi=rf1@data[["variable1"]]^2 + rf2@data[["variable1"]]^2 + rf3@data[["variable1"]]^2

# Construct the t field
rf@data[["variable1"]]=rf@data[["variable1"]]/sqrt(chi/k)

# Unit variance
rf@data[["variable1"]]=rf@data[["variable1"]]*sqrt((k-2)/k)

# Define color palette for plotting
pal <- colorRampPalette(c("white", "black"))

# Convert simulated random field to a raster object and plot it
r <- raster(rf)
plot(r, col = pal(50))



Excursion_set=function(randomfield,level){
  # Set the threshold
  excursion_mask <- randomfield@data >= level

  # Create a copy of randomfield
  excursionset <- randomfield

  # Set values outside the excursion set to NA
  excursionset@data[!excursion_mask] <- NA

  return(excursionset)
}


## define the locations:
from <- 0
to <- 2^8
x.seq <- seq(from, to, length=to)
y.seq <- seq(from, to, length=to)


# Simulate the random field
rf=RFsimulate(RMexp(scale=sqrt(2)/ka), x=x.seq,y=y.seq)

plot(raster(rf),col=pal(7))

##PLOT WITHIN T
```

```
level=0
# Convert it to a raster and plot
excursionset=Excursion_set(rf,level)
exc_raster=raster(excursionset)
plot(exc_raster>=level,col=pal(2),legend=FALSE)


##PLOT WITHIN T
level=1
# Convert it to a raster and plot
excursionset=Excursion_set(rf,level)
exc_raster=raster(excursionset)
plot(exc_raster>=level,col=pal(2),legend=FALSE)



##PLOT WITHIN T
level=2
# Convert it to a raster and plot
excursionset=Excursion_set(rf,level)
exc_raster=raster(excursionset)
plot(exc_raster>=level,col=pal(2),ylim=c(0,2^8),legend=FALSE)


Area_exc2=function(excursionset,level){

  # Extract window size
  ##length of the axis
  maxvalues=excursionset@grid@cellsize * (excursionset@grid@cells.dim-1)
  minvalues=excursionset@grid@cellcentre.offset ##where the window start

  # Total area
  totarea=prod(maxvalues-minvalues)


  # Convert excursionset to a binary raster
  excursionset=raster(excursionset)
  excursionset=excursionset >= level

  # Convert logical to 0s and 1s
  binary_image <- as.integer(excursionset[] == TRUE)

  # Get the area by looking at the percentage of colored pixels in the image
  proportion=sum(binary_image == 1, na.rm = TRUE) / length(binary_image)

  area=proportion*totarea

  return(area)
}


Perim_exc=function(excursionset,level){
  ## Convert excursionset to a binary stars
  excursionsett=st_as_stars(excursionset)>=level

  ## Now convert to sf class
  exc_sf=st_as_sf(excursionsett,merge=TRUE)

  ## Exctract boundary lines and compute length
  perimeter=sum(st_length(st_cast(exc_sf,"MULTILINESTRING")))


  return(perimeter)
}




## define the locations:
from <- 0
to <- 2^8
ka=(100/2^8)^2
x.seq <- seq(from, to, by=1)
y.seq <- seq(from, to, by=1)
rf=RFsimulate(RMexp(scale=sqrt(2)/ka), x=x.seq,y=y.seq)
par(mfrow=c(1,3))

level=1
excursionset=Excursion_set(rf,level)
# Convert to a stars object
excursionsett=st_as_stars(excursionset)>=level


## Now convert to sf class (polygons)
exc_sf=st_as_sf(excursionsett,merge=TRUE)



merged=st_cast(exc_sf,"MULTILINESTRING")
par(mfrow=c(1,3))
plot(excursionsett,main=NULL)
plot(exc_sf,main=NULL)
plot(merged,main=NULL)

perimeter=sum(st_length(st_cast(exc_sf,"MULTILINESTRING")))

Euler_charac=function(randomfield,level){

  # Compute the excursion set
  excursionset <- Excursion_set(randomfield,level)

  # Set the threshold
  inv_mask <- randomfield@data < level

  # Create copy of the randomfield
  inv_exc <- randomfield
  # Set values outside the complementary of the excursion set to NA
  inv_exc@data[!inv_mask] <- NA

  # Convert them to binary rasters
  excursionset=raster(excursionset)
  inv_exc=raster(inv_exc)
  excursionset=excursionset >= level
  inv_exc=inv_exc < level


  # Compute the number of connected_comp with the clump function
  connected_comp=maxValue(clump(excursionset,gaps=FALSE,directions=4))
  if(is.na(connected_comp)){connected_comp=0}
  # Compute the number of holes, we use the clump function on the complementary of
  ##the excursion set and we substract 1 (the background).
  ##Connectivity of 4 to get an accurate value.
  holes=maxValue(clump(inv_exc,gaps=FALSE,directions=4))-1
  if(is.na(holes)){holes=0}

  euler_characteristic2=connected_comp - holes


  return(list(eul=euler_characteristic2, exc=excursionset,invexc=inv_exc))
}


LipschitzKilling=function(randomfield,level){
  excursionset=Excursion_set(randomfield,level)
  L0=Euler_charac(randomfield,level)
  L1=Perim_exc(excursionset,level)/2
  L2=Area_exc2(excursionset,level)
  return(list(L0=L0$eul, L1=L1,L2=L2))
}

MCLK=function(n,from,to,level,model,ka=1) {


L0s=numeric(n)
L1s=numeric(n)
L2s=numeric(n)

for(i in 1:n){
    ## define the locations:



# Simulate the random field
if(model=="Gaussian"){
x.seq <- seq(from, to, by=1)
y.seq <- seq(from, to, by=1)
  rf=RFsimulate(RMexp(scale=sqrt(2)/ka), x=x.seq,y=y.seq)
}
if(model=="chi"){
```

```
x.seq <- seq(from, to, by=1)
y.seq <- seq(from, to, by=1)


# Simulate the random field
rf=RFsimulate(RMexp(scale=sqrt(2)/ka), x=x.seq,y=y.seq)
rf2=RFsimulate(RMexp(scale=sqrt(2)/ka), x=x.seq,y=y.seq)

rf@data[["variable1"]]=rf@data[["variable1"]]^2+rf2@data[["variable1"]]^2
rf@data[["variable1"]]=0.5*(rf@data[["variable1"]]-2)
}
if(model=='t'){

x.seq <- seq(from, to, by=1)
y.seq <- seq(from, to, by=1)


rf=RFsimulate(RMexp(scale=sqrt(2)/ka), x=x.seq,y=y.seq)
rf1=RFsimulate(RMexp(scale=sqrt(2)/ka), x=x.seq,y=y.seq)
rf2=RFsimulate(RMexp(scale=sqrt(2)/ka), x=x.seq,y=y.seq)
rf3=RFsimulate(RMexp(scale=sqrt(2)/ka), x=x.seq,y=y.seq)
rf4=RFsimulate(RMexp(scale=sqrt(2)/ka), x=x.seq,y=y.seq)

chi=rf1@data[["variable1"]]^2+rf2@data[["variable1"]]^2+rf3@data[["variable1"]]^2
+rf4@data[["variable1"]]^2
rf@data[["variable1"]]=(rf@data[["variable1"]])/sqrt(chi/4)
rf@data[["variable1"]]=rf@data[["variable1"]]*sqrt(1/2)
}

LK=LipschitzKilling(rf,level)
L0s[i]=LK$L0[1]
L1s[i]=LK$L1[1]
L2s[i]=LK$L2[1]
}

L0=mean(L0s)
L0CI=list(-1.96*sd(L0s)/sqrt(n),1.96*sd(L0s)/sqrt(n))
L1=mean(L1s)
L1CI=list(-1.96*sd(L1s)/sqrt(n),1.96*sd(L1s)/sqrt(n))
L2=mean(L2s)
L2CI=list(-1.96*sd(L2s)/sqrt(n),1.96*sd(L2s)/sqrt(n))
return(list(L0,L1,L2,L0CI,L1CI,L2CI))
}

n=100
ka=(100/2^8)^2
from=0
to=2^8
levels=seq(-5,5,by=0.5)
MC=apply(as.matrix(levels),1,MCLK,from=from,to=to,n=n,model="Gaussian",ka=ka)

#biased
L0S=numeric(length(MC))
L1S=numeric(length(MC))
```

```
L2S=numeric(length(MC))
lowery=numeric(length(MC))
uppery=numeric(length(MC))
lowerz=numeric(length(MC))
upperz=numeric(length(MC))
lowert=numeric(length(MC))
uppert=numeric(length(MC))
for(i in 1:length(MC)){
  L0S[i]=MC[[i]][[1]]/(to*to)
  L1S[i]=MC[[i]][[2]]/(to*to)
  L2S[i]=MC[[i]][[3]]/(to*to)
  lowery[i]=MC[[i]][[4]][[1]]/(to*to)
  uppery[i]=MC[[i]][[4]][[2]]/(to*to)
  lowerz[i]=MC[[i]][[5]][[1]]/(to*to)
  upperz[i]=MC[[i]][[5]][[2]]/(to*to)
  lowert[i]=MC[[i]][[6]][[1]]/(to*to)
  uppert[i]=MC[[i]][[6]][[2]]/(to*to)
}

data=data.frame(u=levels,L0=L0S,L1=L1S,L2=L2S,
lowery=lowery,uppery=uppery,lowerz=lowerz,upperz=upperz,lowert=lowert,uppert=uppert)

par(mfrow=c(2,3), oma=c(0,0,2,0));

p1=ggplot(data, aes(u, L0)) +
 geom_point(shape=1) +
  geom_errorbar(
    aes(ymin = L0S+lowery, ymax = L0S+uppery),
    width = 0.5,colour="red") +
    geom_function(fun = function(u) (2*pi)^(-3/2) * 2 * ka * u * exp(-u^2/2))

p2=ggplot(data, aes(u, L1)) +
 geom_point(shape=1) +
  geom_errorbar(
    aes(ymin = L1S+lowerz, ymax = L1S+upperz),
    width = 0.5,colour="red") +
    geom_function(fun = function(u) (0.25*sqrt(2 * ka)*exp(-u^2/2)))

p3=ggplot(data, aes(u, L2)) +
 geom_point(shape=1) +
  geom_errorbar(
    aes(ymin = L2S+lowert, ymax = L2S+uppert),
    width = 0.5,colour="red") +
    geom_function(fun = function(u) (1-pnorm(u)))

##unbiased

L0s=numeric(length(MC))
L1s=numeric(length(MC))
L2s=numeric(length(MC))
lowery=numeric(length(MC))
uppery=numeric(length(MC))
lowerz=numeric(length(MC))
```

```
upperz=numeric(length(MC))
lowert=numeric(length(MC))
uppert=numeric(length(MC))
for(i in 1:length(MC)){
  L0s[i]=MC[[i]][[1]]/(to*to)-(to*4)*(MC[[i]][[2]]/(to*to))/(pi*to*to)
  +((to*4)^2/(2*pi*(to*to)^2) -1/(to*to) )*(MC[[i]][[3]]/(to*to))
  L1s[i]=MC[[i]][[2]]/(to*to)-(to*4)*(MC[[i]][[3]]/(to*to))/(2*to*to)
  L2s[i]=MC[[i]][[3]]/(to*to)
  lowery[i]=MC[[i]][[4]][[1]]/(to*to)
  uppery[i]=MC[[i]][[4]][[2]]/(to*to)
  lowerz[i]=MC[[i]][[5]][[1]]/(to*to)
  upperz[i]=MC[[i]][[5]][[2]]/(to*to)
  lowert[i]=MC[[i]][[6]][[1]]/(to*to)
  uppert[i]=MC[[i]][[6]][[2]]/(to*to)

}
data=data.frame(u=levels,L0=L0s,L1=L1s,L2=L2s,
lowery=lowery,uppery=uppery,lowerz=lowerz,upperz=upperz,lowert=lowert,uppert=uppert)

par(mfrow=c(2,3), oma=c(0,0,2,0));

p4=ggplot(data, aes(u, L0)) +
 geom_point(shape=1) +
  geom_errorbar(
    aes(ymin = L0s+lowery, ymax = L0s+uppery),
    width = 0.5,
    position=position_dodge(width=.2),colour="red") +
    geom_function(fun = function(u) (2*pi)^(-3/2) * 2 * ka * u * exp(-u^2/2))

p5=ggplot(data, aes(u, L1)) +
 geom_point(shape=1) +
  geom_errorbar(
    aes(ymin = L1s+lowerz, ymax = L1s+upperz),
    width = 0.5,
    position=position_dodge(width=.2),colour="red") +
    geom_function(fun = function(u) (0.25*sqrt(2*ka)*exp(-u^2/2)))

p6=ggplot(data, aes(u, L2)) +
 geom_point(shape=1) +
  geom_errorbar(
    aes(ymin = L2s+lowert, ymax = L2s+uppert),
    width = 0.5,
    position=position_dodge(width=.2),colour="red") +
    geom_function(fun = function(u) 1-pnorm(u))


plot_grid(p1,p2,p3,p4,p5,p6)
```