Notes on Percolation Analysis of Sampled Scalar Fields

Wiebke Köpp*, Anke Friederici*, Marco Atzori, Ricardo Vinuesa, Philipp Schlatter, and Tino Weinkauf

Abstract Percolation analysis is used to explore the connectivity of randomly connected infinite graphs. In the finite case, a closely related percolation function captures the relative volume of the largest connected component in a scalar field's superlevel set. While prior work has shown that random scalar fields with little spatial correlation yield a sharp transition in this function, little is known about its behavior on real data. In this work, we explore how different characteristics of a scalar field – such as its histogram or degree of structure – influence the shape of the percolation function. We estimate the critical value and transition width of the percolation function, and propose a corresponding normalization scheme that relates these values to known results on infinite graphs. In our experiments, we find that percolation analysis can be used to analyze the degree of structure in Gaussian random fields. On a simulated turbulent duct flow data set we observe that the critical values are stable and consistent across time. Our normalization scheme indeed aids comparison between data sets and relation to infinite graphs.

1 Introduction

Percolation theory has been initiated in 1957 by Broadbent and Hammersley [3]. It is widely used today to characterize and model complex random systems by studying random connectivity in a graph or lattice using statistics. Examples can be found

Wiebke Köpp, Anke Friederici, Tino Weinkauf
Division of Computational Science and Technology, KTH Royal Institute of Technology, Stockholm, Sweden, e-mail: { wiebkek | ankef | weinkauf }@kth.se

Marco Atzori, Ricardo Vinuesa, Philipp Schlatter
Linné FLOW Centre, KTH Mechanics, Royal Institute of Technology, Stockholm, Sweden.
e-mail: { atzori | rvinuesa | pschlatt }@mech.kth.se

*Both authors contributed equally to this work.

in many domains such as fluid dynamics, cosmology, geology, material science, epidemiology, and others [15].

Percolation is largely a theoretical tool to understand the topology of infinite graphs. One of its central components is the *percolation probability*, which describes the likelihood of the existence of an infinite connected subgraph under certain threshold criteria on the vertices or edges of the original graph. Only few previous works apply percolation theory to real data such as fluid flow simulations [11].

This paper explores the computational aspects and potential pitfalls when computing and analyzing a *percolation function*, closely related to the percolation probability, for real data defined on finite lattices. We give the following contributions:

- We discuss the consequences of computing percolation functions on sampled data. This includes how the percolation function is influenced by the dimensions of the grid. We also describe a normalization to the input data that is crucial for comparing percolation functions between different data sets and to the theory (Section 3).
- We propose a method for analyzing the percolation function and its features, combined with a comprehensive visualization for parameter- and time-dependent data sets (Section 3.2).
- We research the sensitivity of the percolation function to the amount of structure in data by designing a family of Gaussian random fields with varying degree of structure (Section 5.1).
- Finally, we apply our framework to fully developed turbulent flows (Section 5.2).

2 Related Work and Background

Consider an infinite graph $\mathbf{L}$. We define for each of its vertices to be *open* with probability $0 \leq p \leq 1$, and *closed* otherwise. Based on this, we define the *open* subgraph $\mathbf{L}'$ using the *open* vertices and their adjacent edges only. Percolation theory studies the structure of $\mathbf{L}'$ depending on the value of p. To do so, we observe the connected components of $\mathbf{L}'$: for small values of p, this subgraph consists of many small connected components which are finite in size (remember that $\mathbf{L}$ and $\mathbf{L}'$ themselves are infinite). For a *critical value* p_c, large-scale structures form and the open subgraph contains an infinite connected component pervading the entire domain: the *percolating cluster*, see Figure 1.

Intriguingly, the transition from finite components to an infinite connected component is sharp: for $p < p_c$, the open subgraph $\mathbf{L}'$ contains finite connected components only, whereas the picture immediately changes for $p > p_c$, for which we see an infinite percolating cluster in $\mathbf{L}'$. The critical value p_c is often referred to as *percolation threshold*. In random media such as porous rocks, the percolation threshold denotes the point where global physical properties of the medium change qualitatively. For example, a porous rock is impermeable before p_c, but lets liquids through after p_c.

The percolation threshold p_c depends *solely* on the connectivity in the infinite graph $\mathbf{L}$. Different topologies have been researched in the mathematics community [21]

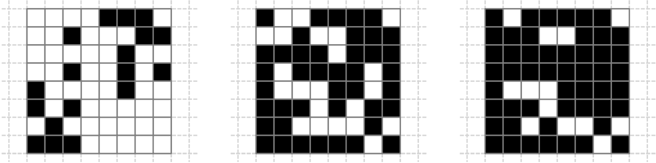


Fig. 1 Examples for site percolation on a 2D lattice with 4-connectivity for $p = 0.2, 0.6$ and 0.8. The percolating cluster of open sites (black) forms approximately at p_c^{2D} (middle).

such as 2D and 3D uniform lattices, triangle meshes, or bow-tie lattices. One also distinguishes between *site* and *bond* percolation, which refers to considering the vertices or edges of the $\mathbf{L}$ as open/closed, respectively. We are concerned with site percolation in this paper, but our work transfers to bond percolation straightforwardly.

The 2D lattice with 4-connectivity (infinite structured grid in 2D) has a site percolation threshold $p_c \approx 0.5927$ [5]. The 3D lattice with 6-connectivity (infinite structured grid in 3D) has a site percolation threshold $p_c \approx 0.3116$ [5]. These values are defined by considering the open subgraph and have been estimated through simulations on finite grids, see Figure 2a.

Levelset percolation [1, 14], which we are concerned with here, applies percolation theory to real data by considering (seemingly) random scalar data values at the vertices of a finite lattice. In this setup, we are looking at the superlevel set[1] of the scalar field $f(\mathbf{x})$ defined as the set of voxels fulfilling $f(\mathbf{x}) \geq p$. The superlevel set can be equated with the open subgraph $\mathbf{L}'$ from before. Again, we are interested in the threshold value p_c where the connected components of the superlevel set pervade the entire domain of the scalar field.

To determine the existence of the percolating cluster and the percolation threshold p_c in this scenario, we have to define a *percolation function*. Similar to Moisy and Jiménez [11], we choose a function based on the volumes of the connected components:

$$P_{\max}(p) = \frac{V_{\max}}{V_{\text{total}}} \tag{1}$$

where V_{total} denotes the total volume of the superlevel set for a given threshold p, and $V_{\max}$ is the volume of its largest connected component. Figure 2b plots $P_{\max}(p)$ for 2D and 3D random noise data sets.

The percolation function can be computed efficiently using an iterative Union-Find algorithm whose non-iterative version has first been suggested in the context of percolation by Hoshen and Kopelman [8]. The algorithm uses similar ingredients as a merge tree computation [4], has been adapted to work in a distributed setting for large-scale simulation data [7], and works as follows: We traverse all sample points $\mathbf{x}$ in decreasing order of their value $f(\mathbf{x})$. We set $p = f(\mathbf{x})$. Let $\mathcal{N}$ denote the set of components assigned to the neighbors of $\mathbf{x}$. We distinguish three cases:

[1] Levelset percolation traditionally refers to the study of the superlevel set, but all the analysis steps presented in this paper can be applied just as well to the sublevel set.

- $\mathcal{N} = \emptyset$: We *create* a new component containing only $\mathbf{x}$.
- $|\mathcal{N}| = 1$: We *extend* the single component $\mathcal{N}$ with $\mathbf{x}$.
- $|\mathcal{N}| > 1$: We *merge* all of the components in $\mathcal{N}$ and add $\mathbf{x}$ to the result.

V_{total} and $V_{\max}$ are recorded in equidistant intervals in order to graph $P_{\max}(p)$. We will show later in Section 3 how to normalize p with respect to the histogram of the data, which serves a number of purposes around the comparison of percolation functions. The computation of $P_{\max}$ essentially gathers statistics on the connected components of the superlevel set. Alternative statistics such as the number of components are not as expressive, as they are rather featureless. For further exploration, see [7].

It is of interest to automatically analyze the percolation function in order to obtain the critical value p_c and other characteristics such as the width Δ of the percolation transition. Stauffer and Aharony [16] suggest to set p_c simply where a percolation function ranging between 0 and 1 first assumes value 0.5. Similarly, they propose to define the transition's width Δ as the interval where the function ranges between either 0.1 and 0.9, or 0.2 and 0.8, both of which they found to yield empirically suitable estimations. Ziff [20] proposed more involved estimation methods. In contrast to our approach, the goal behind these estimations is to derive p_c for an infinite lattice from a number of simulations on finite lattices. We estimate p_c and Δ by fitting a suitable function parametrized with these values to the percolation function, see Section 3.2.

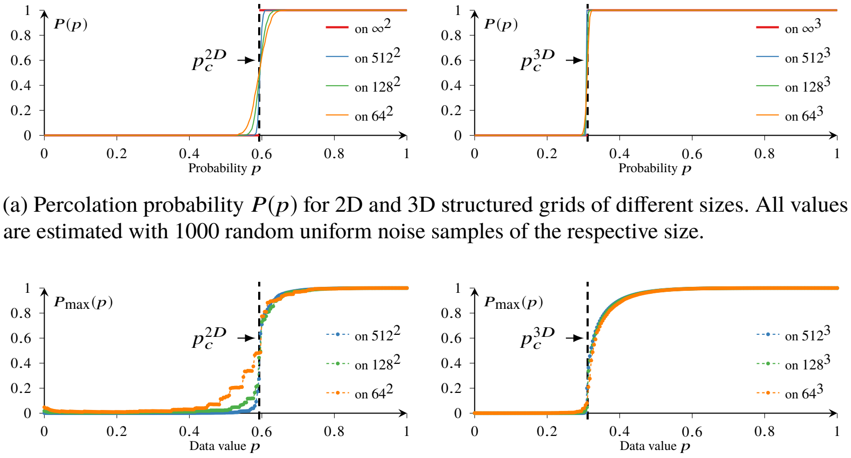
It is the purpose of this paper to provide methods and guidance for the computation, analysis, and visualization of $P_{\max}$, and to discuss the shape of this curve under varying conditions.

3 From Infinite to Finite

Percolation theory was conceived in the context of noise functions defined on infinite domains. When applying it to measured or simulated data, we have finite domains and not necessarily random data. These aspects affect the percolation function and some care has to be taken when computing it.

As described by Newman and Ziff [12], an approximation for the percolation threshold p_c on an infinite lattice can be obtained through determining the value at which the percolating cluster forms when sampling uniform random values on the vertices of a sufficiently large finite lattice. Repeating this procedure a large number of times and averaging over the function that is 1 where the percolating cluster exists and 0 elsewhere yields an estimation of the percolation probability $P(p)$, as shown in Figure 2a.

Based on these graphs, we can conclude that the size of the domain affects the transition width. The location of the percolation threshold on the other hand depends on the grids dimensionality. Similar characteristics can be observed in Figure 2b, where the percolation function for a single sample per grid size is shown. This is analogous to analyzing simulated or measured scalar data, as only a fixed set of data values is available then.



(a) Percolation probability $P(p)$ for 2D and 3D structured grids of different sizes. All values are estimated with 1000 random uniform noise samples of the respective size.

(b) Percolation function $P_{\max}$ for a single samples per 2D and 3D structured grid size.

Fig. 2 Percolation probability $P(p)$ and percolation function $P_{\max}$ for uniform random noise in 2D and 3D structured grids. The dashed lines denoted with p_c^{2D} and p_c^{3D} mark the theoretical critical values on infinite lattices. We can see that $P_{\max}$ features a sharp transition around these values even for finite examples.

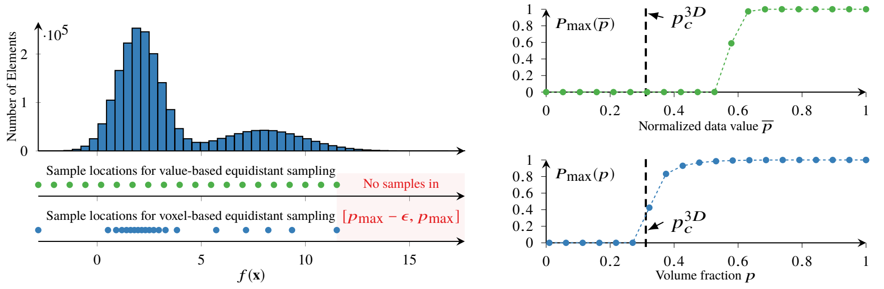
Note that in structured data, it is possible that the data intrinsically has different dimensionality than the underlying lattice. Consider, for example, a data set where the same 2D slice repeats over the third dimension. For uniform random data, we would expect p_c to be close to the theoretical two-dimensional value in this case.

As discussed next, the shape of the percolation function is further affected by value distribution and structure in the data.

3.1 The Extremes of the Value Range

We compute $P_{\max}(p)$ in the interval $[p_{\min}, p_{\max}]$ which corresponds to the data value range in the scalar field $f(\mathbf{x})$. Only few voxels will be part of the superlevel set around $p_{\max}$. The volume computations for Equation (1) can be rather erratic in this range, since they depend on the connectivity between a rather small number of data samples. For example, consider the very first voxel in our algorithm (global maximum): it is easy to see that $P_{\max}(p_{\max}) = 1$ in this case, since this single voxel is the only connected component in the superlevel set. Yet, the existence of an infinite cluster in an infinite domain would have probability 0.

Reliable values are obtained only after having iterated over a sufficiently large number of data points. We thus only start recording the percolation function after having reached a value $p_{\max} - \epsilon$. In this work, we set ϵ such $[p_{\max} - \epsilon, p_{\max}]$ amounts to the highest 1% of all data points.



(a) Histogram of the bimodal normal distribution $f(\mathbf{x})$ and corresponding sample locations.

(b) Value-based and voxel-based percolation functions for $f(\mathbf{x})$.

Fig. 3 Histogram, sampling schemes and percolation function for a scalar field $f(\mathbf{x})$ sampled randomly from a mixture of two Gaussians on a grid of size 128^3. Both value-based and voxel-based sampling yield a sharp transition in the percolation function. However, in case of value-based sampling, the transition is only sharp due to a large fraction of all voxels being processed around the transition. It is also not located near the theoretical threshold p_c^{3D}. This hinders the comparison between data sets and to known results on infinite graphs. Comparability can be achieved by computing the percolation function over the percentage of voxels in the superlevel set.

3.2 Histogram Distribution

Percolation thresholds in infinite domains are known for different random distributions such as uniform and Gaussian noise. See for example Figure 2. However, a measured or simulated scalar field will feature an arbitrary value distribution. This leads to different percolation functions shapes. Is it possible to align these cases such that we can utilize the theoretical knowledge? For example, it would be interesting to compare percolation thresholds as a way to judge the amount of randomness in data.

To this end, we propose a simple normalization scheme. Remember from Section 2 that the parameter p refers to the percentage of open sites/vertices in a lattice. This relates directly to the number of vertices in the superlevel set while computing the percolation function. Hence, instead of setting $p = f(\mathbf{x})$ as it is done in previous work such as by Moisy and Jiminéz [11], we set p in relation to the number of voxels in the superlevel set, which also corresponds to V_{total}. More precisely, from now on p will denote the percentage of voxels in the superlevel set. Computation-wise, this corresponds to the position of a given data value in the sorted data list. Meanwhile, the symbol $\overline{p}$ will refer to the scalar data value $f(\mathbf{x})$. We also normalize this value to the range [0, 1] to ease comparisons between different percolation functions.

Essentially, this procedure shifts the data values such that the histogram matches a uniform distribution. This allows us to relate the level of structure in a given data set to uniform random noise via their percolation functions. An example for how this affects sampling locations and percolation function in given in Figure 3.

4 Analysis and Visualization of Percolation Curve Ensembles

4.1 Analysis of a Single Percolation Curve

We are interested in analyzing the percolation function in Equation (1) to determine the critical value p_c and the width Δ of the percolation transition.

The percolation function in purely random data follows an S-shape. Suitable candidates for approximating curves of this shape come from the family of sigmoidal functions, which include the logistic function, the hyperbolic tangent and the error function. Due to its prevalence in the analysis of percolation curves from Monte-Carlo simulations [13, 19], we use an adapted version of the *error function*, defined as

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} \mathrm{d}t. \tag{2}$$

Note that the error function is related to the normal cumulative distribution function for mean μ and variance σ^2

$$\Phi(x, \mu, \sigma) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right). \tag{3}$$

As such, it ranges between -1 and 1, is monotonically increasing, symmetric with respect to the y-axis and has its point of maximal slope at $x = 0$. By inserting our two parameters p_c and Δ we get:

$$P_{\mathrm{erf}}(p, p_c, \Delta) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{p - p_c}{\Delta}\right)\right) \qquad \text{with } \Delta > 0. \tag{4}$$

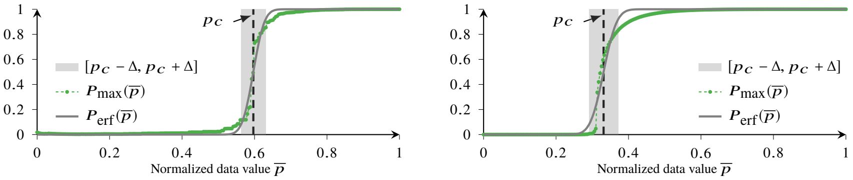
By design, P_{erf} has an inflection point and maximal absolute slope at p_c. Furthermore, the values obtained at p_c, $p_c - \Delta$ and $p_c + \Delta$ are close to the ones suggested by Stauffer and Aharony [16] for the analysis of percolation functions:

$$P_{\mathrm{erf}}(p_c, p_c, \Delta) = \frac{1}{2}\left(1 + \operatorname{erf}(0)\right) = \frac{1}{2} \tag{5}$$

$$P_{\mathrm{erf}}(p_c - \Delta, p_c, \Delta) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{-\Delta}{\Delta}\right)\right) \approx 0.07865 \tag{6}$$

$$P_{\mathrm{erf}}(p_c + \Delta, p_c, \Delta) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{\Delta}{\Delta}\right)\right) \approx 0.92135 \tag{7}$$

In order to fit the function P_{erf}, we use a non-linear least squares fitting algorithm available in SciPy [9]. We obtain an initial guess for p_c and Δ by estimating p_c as the point of maximal slope for a fitted polynomial and Δ as the half of the interval where that polynomial ranges between 0.1 and 0.9. While a polynomial of any degree is not able to capture the asymptotic behavior of the percolation function, estimates based on polynomials serve well as initialization to the actual curve fitting. Figure 4 shows



(a) 2D structured grid, 128^2.

(b) 3D structured grid, 128^3.

Fig. 4 A fit of function P_{erf} (gray) to the percolation functions $P_{\max}$ (green) for uniform random noise in 2D and 3D structured grids estimates percolation threshold p_c and transition width Δ.

the fitted function P_{erf} for the 2D and 3D examples from Figure 2. In both cases, the fitted curve nicely resembles the shape of the sampled percolation function.

Note that for the 3D example, the estimated $p_c = 0.33533$ does not quite reach $p_c^{3D} \approx 0.3116$. We attribute this to the approximate nature of the percolation function in Equation (1). To confirm this, we conducted another experiment in this data set testing for each threshold whether there exists an actual percolating cluster defined as a connected component of the superlevel set spanning the entire domain in any dimension. Indeed, we find the percolating cluster at $p = 0.311294$, which is much closer to the theoretical $p_c^{3D} \approx 0.3116$. We further observe that the percolation function is more asymmetric in the 3D case. While the current estimation scheme cannot capture the asymmetric nature of the curve, it yields sufficiently indicative values. An investigation of alternative estimation schemes is left for future work.

4.2 Analysis of Percolation Curve Ensembles

For the analysis of multiple percolation curves for a data set varying over a parameter t, such as time or dimension, the procedure in Section 4.1 is simply repeated for every sample of t. To then easily assess the development of p_c and Δ over parameter t while still being able to see the connection to the sampled curves, we visualize all three together in a heatmap. The heatmap consists of texels, one for each sample (p, t) with its color encoding the function value $P_{\max}(p)$. The x-axis of the heatmap corresponds to threshold p, and the y-axis varies over parameter t. Figure 5 shows an example: Random uniform noise is sampled on a structured grid of size $128 \times 128 \times L_z$. The row of colored squares for $L_z = 1$ at the very bottom of the plot corresponds to the curve in Figure 4a, whereas the top row with $L_z = 128$ is shown in Figure 4b. The estimated values for p_c and Δ are graphed on top of the heatmap.

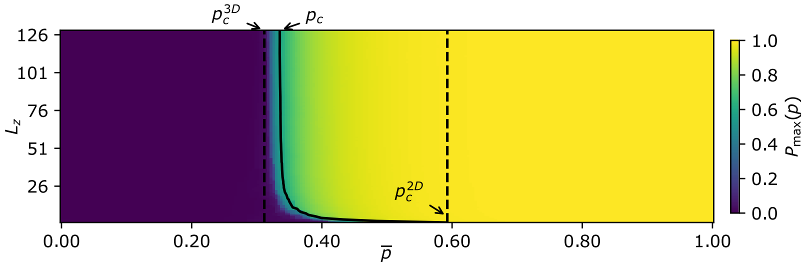


Fig. 5 Percolation function for random uniform noise of dimension 128 × 128 × L_z. The estimated value p_c for $L_z = 1$, i.e. the 2D case, is close to the site percolation threshold for the infinite square lattice $p_c^{2D} \approx 0.5927$. As we increase the dimensionality, p_c moves quickly towards the threshold for the infinite cubic lattice $p_c^{3D} \approx 0.3116$.

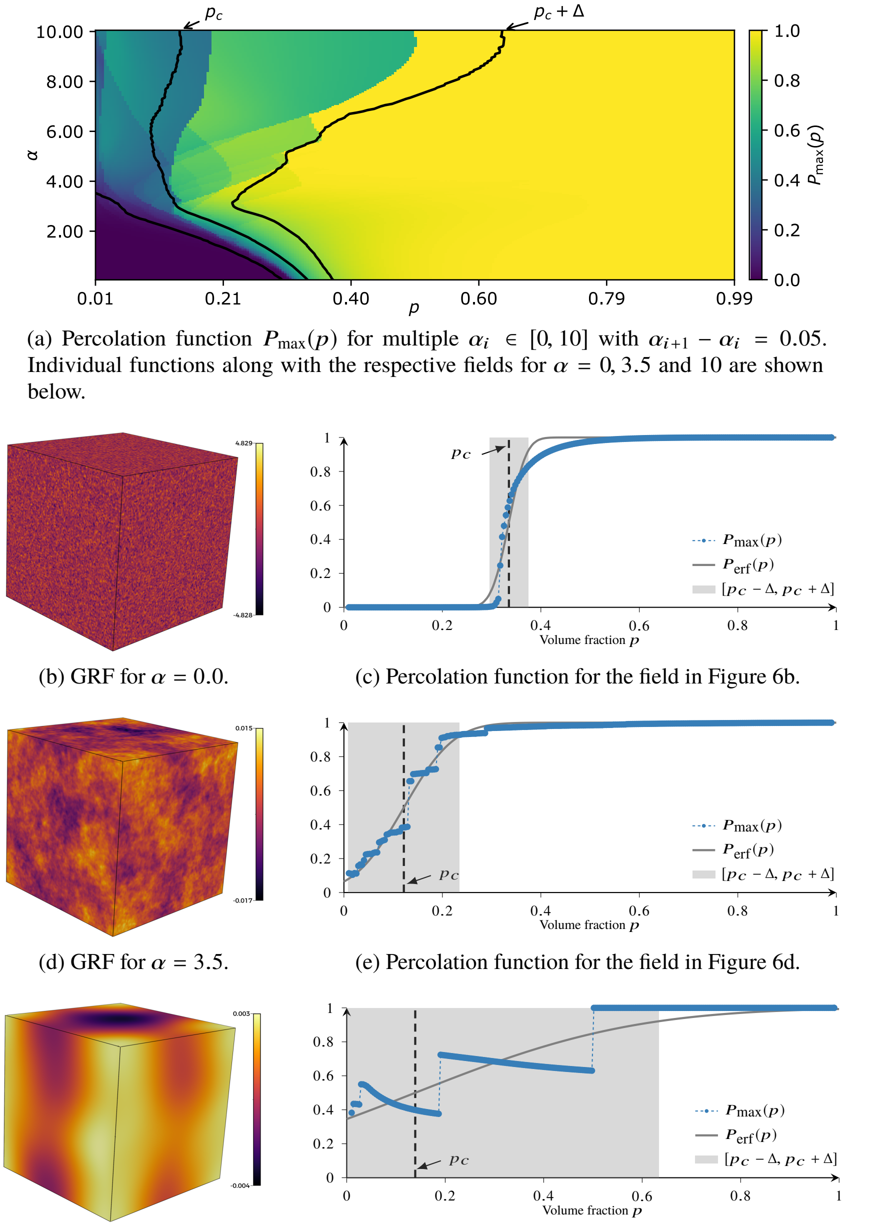
5 Experiments

In the following, we discuss the shape of the percolation curve under varying conditions. We examine a family of synthetic data sets with varying degree of randomness in Section 5.1 to understand how the interplay of structure and randomness caries over to the shape of the percolation function. A simulated flow data set is analyzed in Section 5.2, where we showcase the utility of percolation analysis and observe the effects of our algorithmic choices such as histogram normalization.

5.1 Randomness and Structure: Gaussian Random Fields

The original percolation theory [3] is built on homogeneous and isotropic randomness without correlation between data values. We want to explore the impact of structure in a data set on the resulting percolation function. To this end, we employ *Gaussian random fields* (GRF). which allow us to construct a family of fields with varying degrees of randomness. A GRF is stochastic process defined by a mean and a covariance function and can be understood as a probability distribution over functions, just like its one-dimensional version, the *Gaussian process*. The unique property of a GRF is that the values in every finite subset of sample locations have multivariate Gaussian distribution. There are many ways to sample from GRFs. We choose a rather efficient method using a fast Fourier transform, which computes a GRF in $O(n \log n)$ for a grid with n points. Gaussian white noise has a constant power spectrum. To introduce more structure into the field, lower frequencies need to be more pronounced. This is achieved by multiplying the power spectrum $P(k) = k^{-\alpha}$ for frequencies k with the power spectrum of generated Gaussian noise. Using an inverse Fourier transform, we get our desired scalar field with a level of structure depending on α.

We analyze the percolation for varying levels of structure α in Figure 6a, where the percolation function is shown over the threshold p for increasing structure parametrized by α. We focus on the percolation threshold p_c and the width of the



(a) Percolation function $P_{\max}(p)$ for multiple $\alpha_i \in [0, 10]$ with $\alpha_{i+1} - \alpha_i = 0.05$. Individual functions along with the respective fields for $\alpha = 0, 3.5$ and 10 are shown below.

(b) GRF for $\alpha = 0.0$.

(c) Percolation function for the field in Figure 6b.

(d) GRF for $\alpha = 3.5$.

(e) Percolation function for the field in Figure 6d.

(f) GRF for $\alpha = 10.0$.

(g) Percolation function for the field in Figure 6f.

Fig. 6 Percolation function for generated GRFs with power spectrum $P(k) = k^{-\alpha}$ of dimension 128^3. Three samples of $\alpha \in \{0, 3.5, 10\}$ are shown separately. We observe the transition until about $\alpha \approx 3$, at which point $P_{\max}(0)$ does not start from zero anymore, large discontinuities form and the width Δ of the fitted function increases considerably.

percolation transition Δ, which both change over α. The percolation function and a rendering of the scalar field are shown for three values of α in Figures 6b to 6g.

First, while all curves end with $P_{\max}(p) = 1$, not all of them start with the minimal value 0. When looking at the curve and scalar field at perfect white noise, $\alpha = 0$, we have several small connected components in the data for a low value of p. When increasing p, more of them will form and slowly grow larger, keeping the highest relative volume (cf. eq. (1)) close to 0. Only once a certain point is reached, we observe a sharp increase in $P_{\max}(p)$ as more and more of them merge, rapidly forming a percolating connected component at approximately p_c. For $\alpha > 0$, the non-zero value at $p = 0$ stems from the existence of at least one large connected component for the superlevel set of the largest percent of scalar values. This indicates a high level of structure. Indeed, in Figure 6a we can observe that the percolation function begins below 5% for all $\alpha \leq 3$, but not at any α above that. At approximately that point, no classic sharp percolation transition is visible anymore. Another interesting change is observed at around $\alpha \approx 3$: where the function was very smooth before, large jumps can be seen to appear in the individual percolation functions at higher levels of structure. They begin to form as several small discontinuities as can be seen in Figure 6e at $\alpha = 3.5$ and develop into large disconnected segments for $\alpha = 10$ in Figure 6g. Each such visible discontinuity marks the merging of the largest connected components with another large one. After a merge, the volume of that component decreases in relation to other structures growing, before merging again.

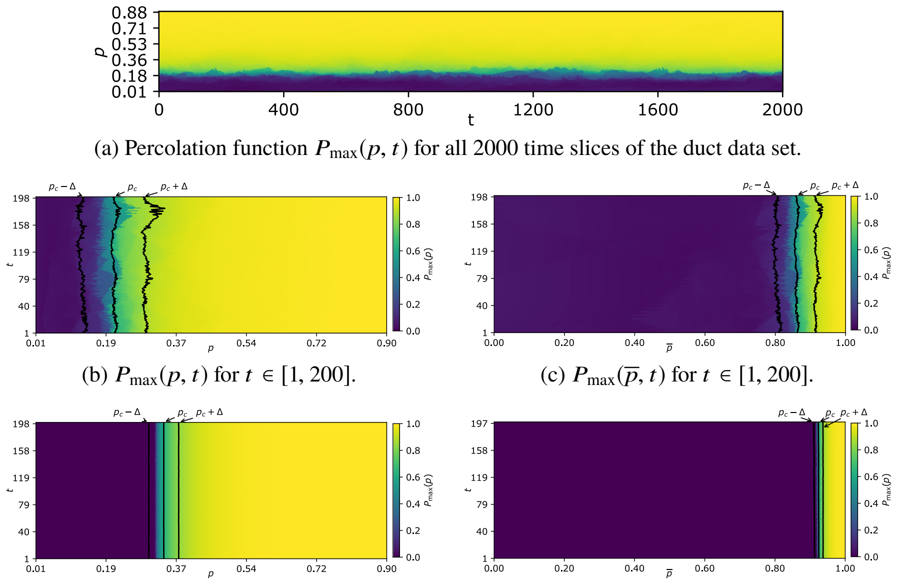
In the heatmap, we can see that the size of these segments grows with increasing α, as fewer and larger connected components form early on. At that point, with no values around zero and considerable jumps in the function, the percolation value p_c loses its meaning as the "point of steepest decrease" and the width Δ of the fitted function grows.

All these observations indicate that there is a strong correlation between the level of structure and the shape of the percolation function. While it is up to definition to pinpoint the exact value of α where no percolation transition occurs, several indicators point to the region of $\alpha \approx 3$.

5.2 Turbulent Flow: Duct Data Set

One common application for percolation analysis is the study of turbulence in fluid dynamics. It has, among other applications, been employed to study the impact of the Reynolds number on the transition from laminar to turbulent flow [10] and to find optimal thresholds to separate highly turbulent structures from the surrounding flow [11].

We analyze a duct flow simulated to investigate intense Reynolds-stress events in fully-developed turbulent flow [17]. It is sampled over 193 × 194 × 1000 data points at an approximately square cross-section and is periodic in the stream-wise direction. As an indicator of Reynolds stress, the scalar combination $\overline{uv}$ has been employed [2]. Here, $\overline{u} = \frac{u-u_{avg}}{u_{rms}}$ denotes the normalized stream-wise directional component of the



(a) Percolation function $P_{\max}(p, t)$ for all 2000 time slices of the duct data set.

(b) $P_{\max}(p, t)$ for $t \in [1, 200]$.

(c) $P_{\max}(\overline{p}, t)$ for $t \in [1, 200]$.

(d) $P_{\max}(p, t)$ after permutation

(e) $P_{\max}(\overline{p}, t)$ after permutation.

Fig. 7 Percolation function for the Reynolds-stress field of the duct data set. The first visualization gives an overview of all 2000 time slices, while the following plots are depicting the first 200 times. In the second row, the percolation function is displayed for both the volume-normalized (left) and the original scalar values (right). Below them, the same function is plotted after permuting the scalar field. We observe that the normalization of the original scalar values does have an impact on the percolation function, but mostly keeps the shape intact. Permuting, on the other hand, removes all variation between the individual time slices.

flow, perpendicular to the normalized cross-stream component $\overline{v}$. This average u_{avg} and the root mean squared deviation u_{rms} are accumulated over a large number of stream-wise slices and time samples. The data points close to the wall are disregarded, as turbulence does not show in that region. Percolation analysis aids in finding the exact wall distance in which to disregard values, see [6].

5.2.1 Stability of the Percolation Function

To find a sensible threshold for intense Reynolds stress events, it is common to compute the percolation function for n_t time slices and average all function values

$$P_{\max}(p) = \frac{1}{n_t} \sum_{t=1}^{n_t} P_{\max}(p, t). \tag{8}$$

The final analysis is applied to this rather smooth averaged curve. However, we are not aware of any work assessing the temporal evolution of the percolation function for real simulated data, which we will do in the following.

We compute the percolation function for a number of $n_t = 2000$ time slices and visualize them in Figure 7a. All time slices have highly similar statistics and are normalized with the same average and root square error. Note that the heatmap plot is rotated by 90 degrees as time t is plotted horizontally. This full range visualization gives an overview for a long time span, which shows us that the percolation function is rather stable.

For a more detailed view, Figure 7b shows the first $n_t = 200$ time slices. The approximated percolation threshold p_c and the transition width Δ are shown as well. The visualization reveals only minor variations in the percolation function, showing that the analysis is stable for temporal development.

5.2.2 Effect of Histogram Normalization

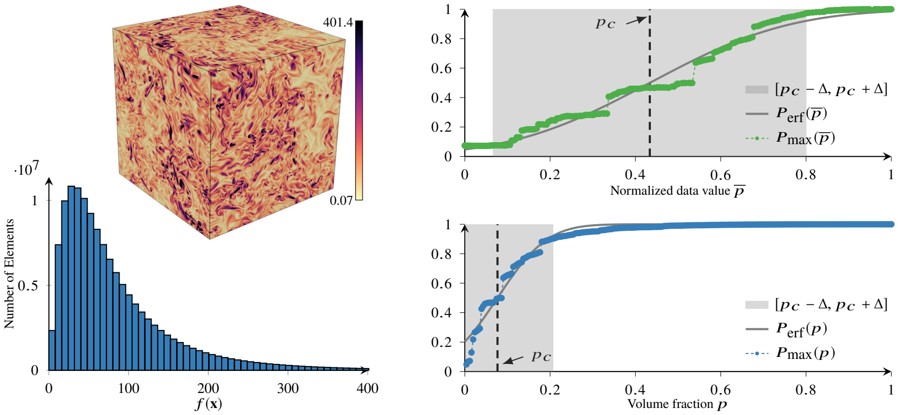
As discussed in Section 3, all visualizations of the duct shown so far are plotted for a normalized histogram of p, which is in practice the volume of the sublevel set. We compare this to working on the values of the scalar field itself. Especially in the case where percolation is used as an indicator for an optimal threshold, it becomes necessary to analyze the function by value. On the other hand, in order to connect these results to percolation theory, the comparison should always be voxel-based.

Figure 7c shows the visualization for $P_{\max}(\overline{p})$ with $t \in [1, 200]$. Figure 7b and 7c are overall very similar. Note that also in the value-based plot, the percolation transition p_c is very stable. However, the respective values of p_c and the width Δ are rather different. This shows that a percolation transition can appear in an irregularly distributed scalar field, but that normalization of these values has a huge impact on the actual function parameters.

We explore this further using an isotropic turbulent flow kindly provided by Yeung et al. [18]. This is a 512^3 sub-volume of a 4096^3 vorticity magnitude field. A visualization is provided in Figure 8a. The histogram reveals the non-uniform data value distribution in this data set. The effect on the value-based percolation function $P_{\max}(\overline{p})$ is drastic: contrary to previous examples, the typical sharp transition between 0 and 1 is lost (Figure 8b, top). However, we can recover the sharp transition with the voxel-based percolation function $P_{\max}(p)$ (Figure 8b, bottom). When comparing it to uniform random noise as shown in Figure 2b, we can see that the transition happens at much larger values of p. This indicates that this data is not random and has indeed some structure.

5.2.3 Effect of De-Correlation

To look deeper into the role that structure plays on the shape of the percolation function as opposed to the value distribution, we remove all structure from the field.



(a) Histogram and volume rendering of $f(\mathbf{x})$ in the sampled range

(b) Value-based and voxel-based percolation function for $f(\mathbf{x})$

Fig. 8 Percolation functions, rendering and histogram of the vorticity magnitude of an isotropic homogeneous turbulence data set provided by Yeung et al. [18]. Comparing this with Figure 2b, we can conclude that this data structurally differs from uniform noise, but is still random enough to not show any larger discontinuities.

By randomly permuting the input scalar field values, we keep the histogram intact, but remove any correlation.

The results for the duct data set are shown for both the volume-based sampling scheme in Figure 7d and the original data values in Figure 7e, which were permuted via the same random seed. As a histogram normalization for permuted values is the same as sampling a uniform random distribution for each point, the results are very close to the theoretical 3D percolation function shape shown in Figure 2a. Also, the variation between time slices is virtually zero. When permuting the original scalar values $\overline{p}$, the function is again stable over time slices, but the percolation transition p_c is shifted notably compared to the theoretical values. The latter confirms that the histogram normalization is effective in enabling the comparison of real data with the theoretical results.

6 Conclusions and Future Work

In this work, we presented a framework for analyzing the structure of existing scalar fields with the help of percolation theory. It was established that a normalization of the scalar values is needed in order to compare the translation and spread of the percolation function to the theoretical values of uniform white noise or other scalar fields. These parameters are obtained by fitting a Gauss error function. To visualize the results, we have shown curve plots of 1D percolation functions and a colormap-based representation of the evolution of percolation across a smoothly changing field.

The analysis was applied to two main data series, a Gaussian random field of increasing correlation and the time series of a fully turbulent duct flow. We could show that for an increase in structure and correlation, three main features point to the existence of a percolation transition: the percolation function assuming a value of almost zero for small p, the absence of large jumps in the curve and a low transition width Δ. For a time series of a statistically stable scalar field on the other hand, we observe a very consistent percolation function.

This work has made a first step in gaining insight into the underlying structure of scalar fields by means of their percolation function. However, more experiments need to be made to determine an arithmetic correlation between different kinds of structure and the parameters of the resulting percolation curve.

Acknowledgments

This work was supported through grants from the Swedish Foundation for Strategic Research (SSF, Project BD15-0082) and the Swedish e-Science Research Centre (SeRC). The presented concepts have been implemented in the Inviwo framework.

References

1. Alexander, K.S., Molchanov, S.A.: Percolation of level sets for two-dimensional random fields with lattice symmetry. Journal of Statistical Physics **77**(3), 627–643 (1994). DOI 10.1007/BF02179453. URL https://doi.org/10.1007/BF02179453
2. Atzori, M., Vinuesa, R., Lozano-Durán, A., Schlatter, P.: Characterization of turbulent coherent structures in square duct flow. Journal of Physics: Conference Series **1001**(1), 012008 (2018). DOI 10.1088/1742-6596/1001/1/012008. URL http://stacks.iop.org/1742-6596/1001/i=1/a=012008
3. Broadbent, S.R., Hammersley, J.M.: Percolation processes: I. Crystals and mazes. Mathematical Proceedings of the Cambridge Philosophical Society **53**(3), 629–641 (1957). DOI 10.1017/S0305004100032680
4. Carr, H.: Topological manipulation of isosurfaces. Ph.D. thesis, The University of British Columbia (2004)
5. Deng, Y., Blöte, H.W.J.: Monte Carlo study of the site-percolation model in two and three dimensions. Physical Review E **72**(1), 016126 (2005)
6. Friederici, A., Atzori, M., Vinuesa, R., Schlatter, P., Weinkauf, T.: An efficient algorithm for percolation analysis and its appliction to turbulent duct flow. In: Euromech Colloquium 598: Coherent structures in wall-bounded turbulence (2018)
7. Friederici, A., Köpp, W., Atzori, M., Vinuesa, R., Schlatter, P., Weinkauf, T.: Distributed Percolation Analysis for Turbulent Flows. In: 2019 IEEE 9th Symposium on Large Data Analysis and Visualization (LDAV) (2019)
8. Hoshen, J., Kopelman, R.: Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm. Phys. Rev. B **14**, 3438–3445 (1976). DOI 10.1103/PhysRevB.14.3438. URL https://link.aps.org/doi/10.1103/PhysRevB.14.3438

9. Jones, E., Oliphant, T., Peterson, P., et al.: SciPy: Open source scientific tools for Python (2001–). URL http://www.scipy.org/. [Online]
10. Lemoult, G., Shi, L., Avila, K., Jalikop, S.V., Avila, M., Hof, B.: Directed percolation phase transition to sustained turbulence in Couette flow. Nature Physics **12**, 254–258 (2016). URL https://doi.org/10.1038/nphys3675
11. Moisy, F., Jiménez, J.: Geometry and clustering of intense structures in isotropic turbulence. Journal of Fluid Mechanics **513**, 111–133 (2004). DOI 10.1017/S0022112004009802
12. Newman, M.E.J., Ziff, R.M.: Efficient monte carlo algorithm and high-precision results for percolation. Phys. Rev. Lett. **85**, 4104–4107 (2000). DOI 10.1103/PhysRevLett.85.4104. URL https://link.aps.org/doi/10.1103/PhysRevLett.85.4104
13. Rintoul, M.D., Torquato, S.: Precise determination of the critical threshold and exponents in a three-dimensional continuum percolation model. Journal of Physics A: Mathematical and General **30**(16), L585–L592 (1997). DOI 10.1088/0305-4470/30/16/005
14. Rodriguez, P.F., Sznitman, A.S.: Phase transition and level-set percolation for the Gaussian free field. Communications in Mathematical Physics **320**(2), 571–601 (2013). DOI 10.1007/s00220-012-1649-y. URL https://doi.org/10.1007/s00220-012-1649-y
15. Sahini, M., Sahimi, M.: Applications of percolation theory. CRC Press (2014)
16. Stauffer, D., Aharony, A.: Introduction To Percolation Theory. Taylor & Francis (1994)
17. Vinuesa, R., Schlatter, P., Nagib, H.M.: Secondary flow in turbulent ducts with increasing aspect ratio. Phys. Rev. Fluids **3**, 054606 (2018). DOI 10.1103/PhysRevFluids.3.054606. URL https://link.aps.org/doi/10.1103/PhysRevFluids.3.054606
18. Yeung, P.K., Donzis, D.A., Sreenivasan, K.R.: Dissipation, enstrophy and pressure statistics in turbulence simulations at high reynolds numbers. Journal of Fluid Mechanics **700**, 5–15 (2012). DOI 10.1017/jfm.2012.5
19. Yonezawa, F., Sakamoto, S., Hori, M.: Percolation in two-dimensional lattices. I. A technique for the estimation of thresholds. Phys. Rev. B **40**, 636–649 (1989). DOI 10.1103/PhysRevB.40.636. URL https://link.aps.org/doi/10.1103/PhysRevB.40.636
20. Ziff, R.M.: Results for a critical threshold, the correction-to-scaling exponent and susceptibility amplitude ratio for 2d percolation. Physics Procedia **15**, 106 – 112 (2011). DOI https://doi.org/10.1016/j.phpro.2011.06.009. URL http://www.sciencedirect.com/science/article/pii/S1875389211003403. Proceedings of the 24th Workshop on Computer Simulation Studies in Condensed Matter Physics (CSP2011)
21. Ziff, R.M., Scullard, C.R.: Exact bond percolation thresholds in two dimensions. Journal of Physics A: Mathematical and General **39**(49), 15083–15090 (2006). DOI 10.1088/0305-4470/39/49/003