

Bootstrapping Vector Fields

Paula Ceccon Ribeiro and Hélio Lopes

Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, Rio de Janeiro, Brazil

Keywords: Vector Field, Uncertainty Quantification, Helmholtz-Hodge Decomposition, Bootstrapping.

Abstract: Vector fields play an essential role in a large range of scientific applications. They are commonly generated through computer simulations. Such simulations may be a costly process since they usually require high computational time. When researchers want to quantify the uncertainty in such kind of applications, usually an ensemble of vector fields realizations are generated, making the process much more expensive. In this work, we propose the use of the Bootstrap technique jointly with the Helmholtz-Hodge Decomposition as a tool for stochastic generation of vector fields. Results show that this technique is capable of generating a variety of realizations that can be used to quantify the uncertainty in applications that use vector fields as an input.

1 INTRODUCTION

It is recognized in the literature that the task of modeling a physical spatial/temporal phenomenon is a very important for decision making applications (Beccali et al., 2003). When you have to deal specifically on the natural phenomena forecasting, it is mandatory to represent *uncertainty* (Mariethoz and Caers, 2014).

Several physical phenomena models that consider uncertainty have two categories: 1) deterministic models, which generates physically-based simulated outcomes; 2) stochastic models, which provides realizations that somehow cover the uncertainty space and at the same time mimic the physics (providing a certain level of realism) (Mariethoz and Caers, 2014).

The main objective of this paper is to present a new stochastic method to generate 2D *vector fields*, since they are very important in a variety set of decision making problems related to *Scientific Computing*. Applications that make use of vector fields include, for example: fluid flow simulation (Anderson and Wendt, 1995), analysis of MRI data for medical prognosis (Tong et al., 2003) and weather prediction (Luo et al., 2012), just to cite a few. The deterministic simulation of vector fields in such applications may require expensive numerical computations (Anderson and Wendt, 1995). The stochastic generation of physically realistic vector fields realizations is a challenging task. Many algorithms for multivariate stochastic simulation are based on very complex probabilistic models (Popescu et al., 1998; Xiu, 2009; Lall et al., 2016) and generally they are not adequate to mimic

physical phenomena such as wind, for example.

In this work, we propose an algorithm to stochastically simulate vector field realizations based on a given gridded 2D vector field \mathbf{V} , which will from now on be called the *training data*. Such algorithm is based on the Helmholtz-Hodge Decomposition (HHD) (Bhatia et al., 2013) and on the non-parametric Bootstrap method (Efron, 1979). The proposed algorithm aims to physically mimic \mathbf{V} and appropriately cover the space of uncertainty. More precisely, our algorithm first use the HHD of \mathbf{V} to obtain its rotational-free and divergence-free potentials components. With such potentials in hand, we perform a bootstrap-like approach to generate R other realizations of these potentials and differentiate them. Finally, we add the generated components to the original harmonic component to generate R vector field realizations. Through Multi-Dimension Scaling (MDS), we could verify that our results were capable to provide some variability. To exemplify its use, we apply our algorithm to the uncertainty quantification introduced by the use of the curl and the divergence finite-difference differential operators.

Paper Outline. The remainder of this paper is organized as follows: Section 2 presents some previous and related work. Section 3 and 4 describes the Bootstrap method and the Helmholtz-Hodge Decomposition, in that order. Section 6 presents an analyses of the method's capabilities, whilst Section 7 shows an application of the technique, followed by the perfor-

mance results in Section 8. Finally, Section 9 presents our conclusion as well as some final remarks and future studies.

2 RELATED WORK

This section has the objective to discuss the related work about the three main concepts used in this work: stochastic simulation, Helmholtz-Hodge Decomposition and the Bootstrap method.

Stochastic Simulation. As mentioned in the previous section, the stochastic generation of physically realistic vector fields realizations is a challenging task. In one side, many algorithms based on probabilistic models for multivariate stochastic simulation (Popescu et al., 1998; Xiu, 2009; Lall et al., 2016) are very complex mathematically speaking and generally they are not adequate to mimic physical phenomena such as wind, for example. In the other side, there are several geostatistical methods in the literature dedicated to the stochastic simulation of spatial physical phenomena (Lantuéjoul, 2013). Generally, they are applied to the generation of univariate continuous or categorical functions defined on a 2D or 3D grid. They usually propose a parametric model of uncertainty to formulate the lack of knowledge, and models based on variogram are the most traditional ones (Oliver and Webster, 2014). Alternatively, non-parametric approaches, such as the ones based on Multiple-Point Statistics (MPS), have received a lot of investigation in the last five years. These approaches generate realizations of a spatial phenomenon based on a training image, which implicitly describes the phenomenon's construction process (Mariethoz and Caers, 2014). These methods have a very strong connection with computer graphics' texture synthesis techniques (Mariethoz and Lefebvre, 2014), like Image Quilting (Efros and Freeman, 2001), for example. Similarly to MPS methods, this work proposes a new non-parametric method for the stochastic generation of 2D vector-fields that is also based on a training data. However, this new method uses the bootstrap technique instead of the MPS.

Helmholtz-Hodge Decomposition. A wide range of the applications of the Helmholtz-Hodge Decomposition can be found in the literature. These include the use of the HHD to detect singularities for fingerprint matching (Gao et al., 2010), its application in the field of complex ocean flow visualization and analysis for feature extraction (Wang and Deng, 2014), cardiac video analysis (Guo et al., 2006), hurricane

eye tracking (Palit, 2005) and the aerodynamic design of cars and aircrafts (Tong et al., 2003). Recently, Ribeiro and Lopes (Ribeiro et al., 2016) proposed the use of the HHD as a tool to analyze 2D vector field ensembles. This work will use the HHD to decompose the training data in order to obtain the rotational-free and the divergent-free potentials. With these two scalar fields in hands a bootstrap-based perturbation is performed and the resulted fields are then differentiated to construct a vector field realization by summing their perturbed components. Perturbing the scalar potentials independently is fundamental to achieve the objective of providing a certain level of realism of the generated vector fields.

Bootstrap. The *Bootstrap method* is a statistical method based on resampling with replacement. It is commonly applied to measure the accuracy of statistical estimators (Efron, 1979). In general, such accuracy could be defined in terms of bias, variance, confidence intervals, prediction error or some other dispersion measure. This technique has been applied to visual computing problems, such as: performance evaluation for computer vision systems (Cho et al., 1997), searching for radial basis function parameter (Liew et al., 2016), evaluation of the influence of hidden information on supervised learning problems (Wang et al., 2014) and edge detection (Fu et al., 2012), among others. This technique has in also very important in this paper. Not only because it performs the perturbation of the potential fields, but also because it is adopted to quantify the algorithm uncertainty introduced by the use of the curl and the divergence finite-difference differential operators.

3 THE BOOTSTRAP METHOD

The Bootstrap method is based on the notion of a *bootstrap sample* (Efron, 1979; Wasserman, 2004). To better understand it, let \hat{F} be an empirical distribution, with probability $1/n$ on each of the n observed values x_i , with $i \in \{1, 2, \dots, n\}$. Then, a bootstrap sample is defined as a random sample of size n drawn from \hat{F} with replacement, say $\mathbf{x}^* = (x_1^*, x_2^*, \dots, x_n^*)$. The star notation indicates that \mathbf{x}^* is not the actual data set \mathbf{x} , but a randomized, or *resampled*, version of \mathbf{x} . For more details about this technique, see (Wasserman, 2004).

With this concept in mind, assume that $T_n = g(x_1, x_2, \dots, x_n)$ is a statistic of the data set $\{x_1, \dots, x_n\}$. To compute the variance of T_n , denoted by $\mathbb{V}_F(T_n)$, it would be necessary to know the distribution F of the data. Often, however, this

is unknown. The Bootstrap technique estimates $\mathbb{V}_F(\mathbf{T}_n)$ by the use of stochastic simulations, where the unknown distribution F is approximated by a distribution named \hat{F} . Then, an approximation of $\mathbb{V}_F(\mathbf{T}_n)$ is computed as $\mathbb{V}_{\hat{F}}(\mathbf{T}_n)$. Generating B bootstrap samples, it is now possible to approximate the distribution of \mathbf{T}_n by evaluating $\mathbf{T}_n^* = g(x_1^*, \dots, x_n^*)$. Using this distribution, we can finally compute the variance $\mathbb{V}_{\hat{F}}(\mathbf{T}_n)$ according to the following formula:

$$\mathbb{V}_{\hat{F}}(\mathbf{T}_n) = \frac{1}{B} \sum_{i=1}^B \left(\mathbf{T}_i^* - \frac{1}{B} \sum_{b=1}^B \mathbf{T}_{n,b}^* \right)^2, \quad (1)$$

where \mathbf{T}_i^* , $i = 1, \dots, B$, represents the statistics computed at the i^{th} bootstrap sample.

4 HELMHOLTZ-HODGE DECOMPOSITION

The Helmholtz-Hodge Decomposition (Chorin and Marsden, 1993) states that a square-integrable vector field \mathbf{V} can be formulated as the sum of three orthogonal components:

$$\mathbf{V} = \nabla\phi + \nabla \times \boldsymbol{\psi} + \mathbf{h}, \quad (2)$$

where $\nabla\phi$ is the *rotational-free* term ($\nabla \times \nabla\phi = 0$), $\nabla \times \boldsymbol{\psi}$ is the *divergence-free* term ($\nabla \cdot (\nabla \times \boldsymbol{\psi}) = 0$) and \mathbf{h} is the *harmonic* term ($\nabla \times \mathbf{h} = 0$ and $\nabla \cdot \mathbf{h} = 0$). Figure 1 shows an example.

The scalar field ϕ is called the *potential* field of the curl-free term.

The curl of a 2D vector field \mathbf{V} is defined by

$$\nabla \times \mathbf{V} = \nabla \times (V_1, V_2) = \left(\frac{\partial V_2}{\partial x} - \frac{\partial V_1}{\partial y} \right),$$

Thus, one can write $\nabla \times \mathbf{V}$ as $(\nabla \cdot J)\mathbf{V}$, where J is an operator that rotates a vector by $\frac{\pi}{2}$ in a clockwise direction: $J(x, y) = (y, -x)$.

As a consequence, Equation 2 can be rewritten for a 2D vector field (Polthier and Preuß, 2003) as:

$$\mathbf{V} = \nabla\phi + J(\nabla\boldsymbol{\psi}) + \mathbf{h}, \quad (3)$$

where $\boldsymbol{\psi}$ is a scalar field that will be called the *potential field* of the divergent-free component.

To obtain the HHD of a given 2D vector field \mathbf{V} means to determine the scalar functions ϕ and $\boldsymbol{\psi}$ and the harmonic function \mathbf{h} that satisfies Equation 3. This leads to the following system of equations:

$$\begin{cases} \nabla \cdot \mathbf{V} = \Delta\phi \\ (\nabla \cdot J)\mathbf{V} = -\Delta\boldsymbol{\psi} \end{cases}, \quad (4)$$

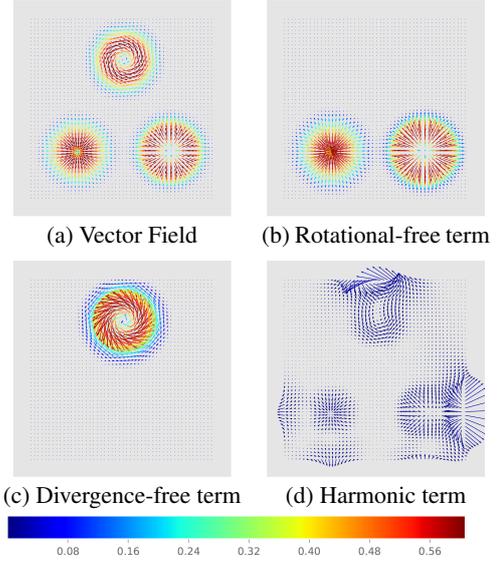


Figure 1: The HHD states that a vector field (a) is composed of a rotational-free (b), a divergence-free (c), and a harmonic component (d). The color bar represents the vector magnitudes.

where Δ is the the Laplacian operator.

An important fact is that the HHD is unique for vector fields vanishing at infinity on unbounded domains (Pascucci et al., 2014). However, to obtain an unique solution for closed domains, some boundary conditions should be established. The normal-parallel (NP) boundary condition is the most commonly used, which requires the divergence-free and the rotational-free components to be parallel and normal to the boundary, respectively:

$$\begin{cases} \nabla\phi \times \mathbf{n} = 0 \\ (\nabla \cdot J)\boldsymbol{\psi} \cdot \mathbf{n} = 0 \end{cases}, \quad (5)$$

where \mathbf{n} represents the outward normal to the boundary. Another possible boundary condition is to impose constant potentials on the boundary, which implies the rotational-free component normal to the boundary and the divergence-free tangent to it (Petronetto et al., 2010). However, these two types of boundary conditions may introduce artifacts that were not observed in the original field due to the imposed dependency between the vector field components and the shape and orientation of the boundary. To overcome this problem, Pascucci et al. (Pascucci et al., 2014) proposed the Natural HHD (NHHD), which decomposes \mathbf{V} by separating the components by its influences, which can be internal or external. Its formulation is written as follows:

$$\mathbf{V} = \nabla\phi^* + (\nabla \cdot J)\boldsymbol{\psi}^* + \mathbf{h}^*$$

where, $\nabla\phi^*$ is the natural divergence and $(\nabla \cdot J)\boldsymbol{\psi}^*$

is the natural rotational. They represent the components influenced by the divergence and rotational of \mathbf{V} inside the domain. Moreover \mathbf{h}^* is the natural harmonic, which is influenced only by the exterior of the domain.

In this work, we adopted the NHHD method to obtain the rotational-free, divergence-free and harmonic natural components of a given 2D vector field \mathbf{V} . More details for how to obtain this decomposition can be found in the original work of (Pascucci et al., 2014).

5 PROPOSED METHOD

This section presents a new stochastic method to generate 2D vector field realizations from a given training data, i.e. a gridded 2D vector field. This approach is based on the Bootstrap technique and uses the Helmholtz-Hodge Decomposition to consistently generate stochastic realizations of vector fields.

Consider a discrete sampling of a two-dimensional domain on a Cartesian grid structure $\mathbb{S}_{m,n} = \{\mathbf{x}_{i,j} \in \mathbb{R}^2 : 1 \leq i \leq m, 1 \leq j \leq n\}$. Also, suppose that a discrete 2D vector field \mathbf{V} is given, i.e., to each spatial point in $\mathbb{S}_{m,n}$ there is a 2D vector associated. This 2D vector field \mathbf{V} is the *training data*.

The main goal of this method is to randomly generate vector fields that have similar characteristics of the training one, i.e., that are structural perturbations of the original vector field.

Overview. The first step in our method is to compute the NHHD of the training data \mathbf{V} . So, at each point $\mathbf{x}_{i,j} \in \mathbb{S}_{m,n}$ we have the following equality:

$$\mathbf{V}^*(\mathbf{x}_{i,j}) = \nabla\phi^*(\mathbf{x}_{i,j}) + (\nabla \cdot \mathbf{J})\psi^*(\mathbf{x}_{i,j}) + \mathbf{h}^*(\mathbf{x}_{i,j}). \quad (6)$$

With the NHHD components of the given training data \mathbf{V} in hand, we stochastically generate other R 2D vector fields based on \mathbf{V} . To obtain each realization, we firstly perturb the divergence-free ϕ^* and rotational-free ψ^* scalar potentials around b points $\mathbf{x}_{i,j} \in \mathbb{S}_{m,n}$ using a Bootstrap-like technique. From these perturbed scalar potentials, we then compute the corresponding rotational-free and divergent-free terms from their partial derivatives. We add these two terms to the original harmonic term \mathbf{h}^* in order to finally create a vector field realization.

The number b of blocks in which to perform the Bootstrap is defined through a Poisson Distribution (Wasserman, 2004) with rate λ . This rate represents the mean number of blocks that are going to be perturbed. The greater the λ the higher the variability induced in the samples.

Given that we are dealing with vector fields, we adopted an strategy to preserve their structure during the resampling step. Such strategy is based on a kernel proposed by (Fu et al., 2012) and depicted in Figure 2. This kernel explores the directional coherence of the contours that pass through the central pixel. As can be seen, the kernel divides a $n \times n$ block into 8 subgroups. When performing the Bootstrap-based technique, each of these regions is resampled with replacement separately to obtain a Bootstrap sample around the central pixel. The size of the kernel presented in Figure 2 is 5×5 . The bigger the mask, the higher the variation of the Bootstrap samples in relation to the input sample.

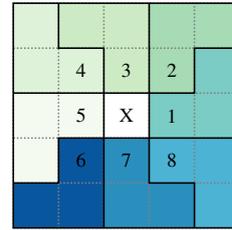


Figure 2: A kernel that divides a $n \times n$ block of the domain into 8 subgroups in order to preserve the vector field orientation after resampling with replacement the pixels in each subgroup separately.

Once again, taking as a realization the vector field depicted in Figure 1, one can perceive, through Figure 3, that the adopted kernel is capable of preserving the orientation of the vector field used as input for the Bootstrap method. More than that, in regions in which the potentials are practically constant, no noise is added to the vector samples.

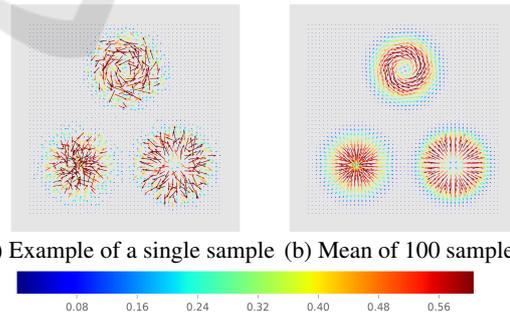


Figure 3: Example of vector fields obtained using a kernel divided in regions to preserve the vector field orientation. The color scale matches the one presented in Figure 1 for comparison purposes.

With this knowledge, we can now specify that, in this work, λ is defined as a percentage of the training data size divided by the kernel size.

At last, a smoothing step is performed through a *Gaussian Filter* (Gonzalez and Woods, 2006), which

standard deviation (σ) can be parameterized, for both x and y dimensions.

The Algorithm. We implemented the proposed method according to the pseudocode described in Algorithm 1. This pseudocode generates a stochastic realization \mathbf{R}^* based on the NHHD components of a training data \mathbf{V} .

The method has as input the following list of variables:

- the scalar potentials ϕ^* , ψ^* and the vector field \mathbf{h}^* obtained by the NHHD of the training data \mathbf{V} ;
- the kernel \mathbf{K} of size $l \times l$ used to perform the re-sampling with replacement on the potentials;
- the number b of blocks in which we will perform the Bootstrap.

Algorithm 1: Generation of a realization \mathbf{R}^* based on the NHHD components of a training 2D vector field \mathbf{V} .

```

input :  $\phi^*$ ,  $\psi^*$ ,  $\mathbf{h}^*$ ,  $\mathbf{K}$ ,  $b$ 
output:  $\mathbf{R}^*$ , a vector field realization
1  $\Phi_{boot}^* \leftarrow \phi^*$ ;
2  $\Psi_{boot}^* \leftarrow \psi^*$ ;
3  $x \leftarrow \text{randInt}(1, m, b)$ ;
4  $y \leftarrow \text{randInt}(1, n, b)$ ;
5 for  $k \leftarrow 1$  to  $b$  do
6    $i \leftarrow x[b]$ ;
7    $j \leftarrow y[b]$ ;
8    $boot\_indices \leftarrow \text{local\_bootstrap}(\mathbf{K})$ ;
9    $\Phi_{boot}^*(i, j) \leftarrow \hat{\mathbf{F}}\phi(boot\_indices)$ ;
10   $\Psi_{boot}^*(i, j) \leftarrow \hat{\mathbf{F}}\psi(boot\_indices)$ ;
11 end
12  $\Phi_{boot}^* \leftarrow \text{smooth}(\Phi_{boot}^*)$ ;
13  $\Psi_{boot}^* \leftarrow \text{smooth}(\Psi_{boot}^*)$ ;
14  $\nabla\phi_{\mathbf{R}}^* \leftarrow \text{divergent}(\Phi_{boot}^*)$ ;
15  $\nabla \times \psi_{\mathbf{R}}^* \leftarrow \text{curl}(\Psi_{boot}^*)$ ;
16  $\mathbf{R}^*(\mathbf{x}_{i,j}) \leftarrow \nabla\phi_{\mathbf{R}}^*(\mathbf{x}_{i,j}) + \nabla \times \psi_{\mathbf{R}}^*(\mathbf{x}_{i,j}) + \mathbf{h}^*$ ;
    
```

The input b defines the number of indexes that will be generated through an Uniform Distribution (Wasserman, 2004) (lines 3 and 4). These indexes represent central positions of regions in the scalar potentials of \mathbf{V} that are going to be perturbed using a Bootstrap-like approach.

Then, for each one of the b indexes pairs, say $\mathbf{x}_{i,j}$, we perform a local bootstrap (line 8) centered on $\mathbf{x}_{i,j}$ based on the input kernel \mathbf{K} , which results in a new organization of the $l \times l$ region around $\mathbf{x}_{i,j}$. In other words, the region around $\mathbf{x}_{i,j}$ will be perturbed and new values will be assigned to $\star_{boot}^*(i, j)$.

In the following, we perform a smoothing step on \star_{boot}^* , i.e., we obtain a smoothed version of $\hat{\mathbf{F}}_*$. The smoothing step is required because a small change in the potentials can lead to a significant change in the vector field, once this is obtained deriving these potentials. In this work, we used σ equal to 2 pixels in the smoothing step.

After these steps, we can now derive the new scalar potentials to obtain new realizations for the divergence-free (line 14) and rotational-free (line 15) components of \mathbf{V} . Finally, a new vector field realization is obtained summing these components with the original harmonic component of \mathbf{V} (line 16), following Equation 2.

Repeating this procedure R times, we will then have a set of R realizations of vector fields obtained through the original NHHD components of \mathbf{V} .

6 RESULTS AND DISCUSSION

To verify the results that the proposed method can achieve, we make use of a 2D vector field ensemble comprehended by seven multi-method wind forecast realizations \mathcal{E} , provided by the Brazilian Instituto Nacional de Pesquisas Espaciais (INPE). Each realization in \mathcal{E} represent a possible wind forecast for a region delimited by $35^\circ 48' S$ and $83^\circ W$ as the minimum latitude and longitude coordinates (DMS), respectively, and by $6^\circ 12' N$ and $25^\circ 48' W$ as the maximum latitude and longitude coordinates, in that order. The data is defined over a Cartesian grid structure with dimension of 144×106 .

As a first step, we apply the NHHD on each realization \mathbf{R} in \mathcal{E} to derive its divergence-free, rotational-free and harmonic components. Through this decomposition, we obtain the potentials of the rotational-free and divergence-free components. With those potentials in hand, we can then derive the rotational-free, divergence-free and harmonic components as stated in Equation 2. For each realization \mathbf{R} in \mathcal{E} , we apply Algorithm 1 to obtain 100 other new realizations.

Similarity Measure and MDS Projection. To provide a way of visually encode the similarity between the vector fields, we make use of the *MDS* (Kruskal, 1964) technique for dimensionality reduction to visualize high-dimensional data in a 2-dimensional space. The MDS method aims to provide insight in the underlying structure and relations between patterns by providing a geometrical representation of their similarities (Honarkhah and Caers, 2010). Mathematically speaking, MDS translates a dissimilarity ma-

trix into a configuration of points in a n -D Euclidean space.

For two vector fields \mathbf{A} and \mathbf{B} , we adopted the following similarity measure, known as the *Cosine Similarity*:

$$\text{similarity}_{\mathbf{A},\mathbf{B}} = \cos \theta = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \cdot \|\mathbf{B}\|} \quad (7)$$

Such a measure states how related two vector fields are given their angles. For similar vectors the similarity coefficient will be close to 1, for opposite vectors, such coefficient will be close to -1 . For unrelated vectors, on the other hand, this coefficient will be around 0.

To take into account both the magnitude and orientation of the vector fields \mathbf{A} and \mathbf{B} in the cosine similarity computation, we perform the following transformation.

Firstly, for a vector \mathbf{V} of dimensions $m \times n$, we unroll it from a 2-dimensional vector to a 1-dimensional vector. Then, we generate a new vector $\mathbf{V}^* = (\mathbf{v}_x^*, \mathbf{v}_y^*)$ based on \mathbf{V} such as:

$$\begin{aligned} \mathbf{v}_x^* &= \text{atan2}(\mathbf{v}_y, \mathbf{v}_x) \\ \mathbf{v}_y^* &= \|\mathbf{V}\| \\ \mathbf{V}^* &= (\mathbf{v}_x^*, \mathbf{v}_y^*) \end{aligned}$$

For an ensemble \mathcal{E} , after this step, we have a new set \mathcal{E}^* in hand. All vectors in \mathcal{E}^* are normalized as follows:

$$\begin{aligned} \mathbf{v}_x^* &= \frac{\mathbf{v}_x^*}{\pi/2} \\ \mathbf{v}_y^* &= \frac{\mathbf{v}_y^*}{\max(\mathbf{v}_y^* \in \mathcal{E}^*)} \end{aligned}$$

After this transformation, we apply the similarity measure for each pair of realizations in \mathcal{E}^* .

Figure 4 presents the MDS for the wind forecast ensemble and its mean vector, after applying the transformation described before.

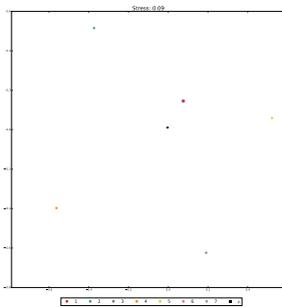


Figure 4: MDS visualization for the original ensemble \mathcal{E} . Colors represent each realization in \mathcal{E} . The black square represent the mean vector of \mathcal{E} .

Coverage Test. It is relevant to verify whether we can generate a set of realizations that covers the given ensemble set or not. This might state if, from a single realization, it is possible to obtain certain scenarios that could be derived through another simulation process (possible more costly). To do this, we first tried different values for the λ parameter given different bootstrap kernel sizes to generate 100 new samples from the mean vector field μ . They ranged from 30% to 90% and from 5×5 to 17×17 , respectively.

We achieved the best coverage using a λ value of 90% and a kernel size of 19×19 , as can be seen in Figure 5.

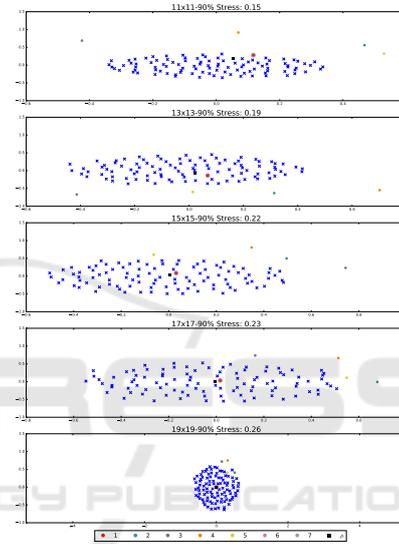


Figure 5: MDS visualization between \mathcal{E} and a new ensemble generated through the mean of \mathcal{E} , represented as a black square.

Given that, Figure 6 present the MDS for each vector field in the original set \mathcal{E} and a new set of realizations derived from it using a kernel size and λ as specified before. Markers of same color belongs to the same set, i.e., were generated based on a common realization of the set \mathcal{E} . Circle markers represent each realization of the set \mathcal{E} . Cross markers represent new realizations, and square markers show both the closest and farthest simulation given a base realization – Table 4 depict these simulations for each realization in \mathcal{E} . Through this image, we can see that, for each realization $s \in (1, \dots, 7)$, the resulting set of realizations present some variability in relation to the original vector field used as base for the stochastic simulation method.

Putting all these simulations together, we have the result presented in Figure 7. From this image, we can notice that the original set \mathcal{E} is completely surrounded by the new realizations.

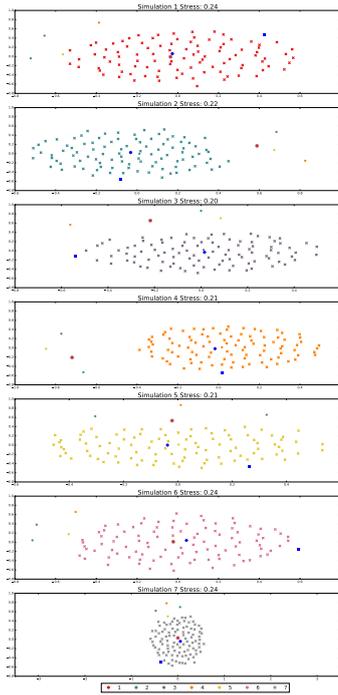


Figure 6: MDS visualization between each set of new realizations and the original ensemble \mathcal{E} . Colors represent each realization of the set \mathcal{E} . Circular markers represent each realization in \mathcal{E} . Cross markers represent, for each \mathbf{V} in \mathcal{E} , the new realizations derived from \mathbf{V} , both presenting the same color.

Evaluation. Willing to evaluate the quality of the results achieved with the proposed method, we conducted an informal study with 19 people with a varied age range as well as educational level (Figure 8). Here, we define quality as the capacity of a generated realization be as realistic as the input data set (1) and unique in comparison with its members (2).

To evaluate (1) we displayed 4 vector fields (2 of

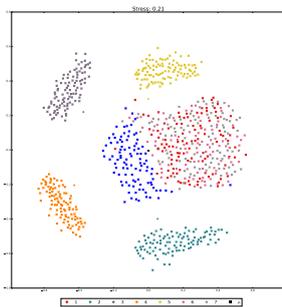


Figure 7: MDS visualization between each new realization and the original ensemble \mathcal{E} . Colors represent each realization of the set \mathcal{E} . Circular markers represent each realization in \mathcal{E} . Cross markers represent new realizations derived from the original one (presented with a circular marker of the same color).

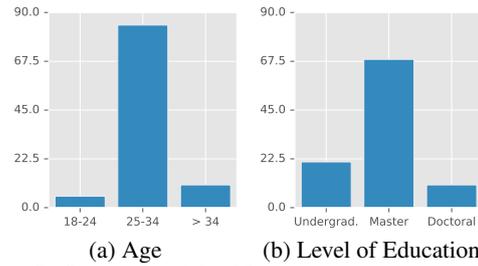


Figure 8: Summary of the 19 participants of our informal study.

them from the wind forecast data set – members 1 and 5¹ – and the other 2 generated through the proposed method – realizations 4f and 7c) and asked the participants to classify them as *training* or *stochastically generated* or choose the option *I don't know*. Most of the participants got the right answer, however, for a close call. For instance, training data 5 was correctly classified by 57.9% of the participants, while 21.1% classified it as stochastically generated and the remaining couldn't tell the difference. The same result was observed for the stochastically generated vector field 7c. On average, 60.55% of the participants chose the correct answer, 21% chose the wrong answer and 18.45% didn't know how to classify it.

For the evaluation of (2) we presented two sets of vector fields. The first one contained 3 members of the wind forecast data set (members 1, 2 and 3). The second one was composed by 3 vector fields (realizations 1c, 2c and 3c) generated using the first set members. We then asked the participants to indicate, for each member of the second set, which vector field in the first one was used to generate it, or the option *I don't know*. For all vector fields in the second set, the majority of the participants did not identify the correct training vector field. The percentage that did it was 10.5%, 21.1% and 10.5%, for each vector field in the second set. For the first vector field in this set, 1c, 31.1% of the participants couldn't chose the most similar vector field from the original data set. For the second and third vector fields, this percentage was 21.1%. Realization 3c was characterized as almost similar to two different vector fields (31.6% for 3 and 36.8% for 1), being considered more similar to a vector field different from its training one.

These results show that, despite being possible to identify the tested vector fields as training or stochas-

¹For all data used in this test and here presented, read Table 4 as:

1. member x : x -th vector field in column Realization;
2. xc : x -th vector field in column Closest Simulation;
3. xf : x -th vector field in column Farthest Simulation.

tically generated, our method was capable to generate realizations that mimics the physical simulation.

7 APPLICATIONS

As mentioned before, the presented approach may be useful in a varied range of applications. In this section, we present a quantification approach to the algorithm uncertainty related to different scenarios of the curl and divergence discrete differential operators.

Navier-Stokes. Consider the vector field presented in Figure 9. This field is defined over a grid of 64×64 , with its minimum and maximum as 0.007812 and 0.992188, respectively, in the x and y directions. This field is the result of a *Navier-Stokes* simulation (Chorin, 1968), which aims to describe the motion of viscous fluid flows. Such kind of simulation can be used to model a varied set of physics phenomena, ranging from waves simulation (Abadiea et al., 2010) to image and video inpainting (Bertalmio et al., 2001). As can be seen, the divergence-free component defines such field (we may consider the rotational-free and harmonic components as noise).

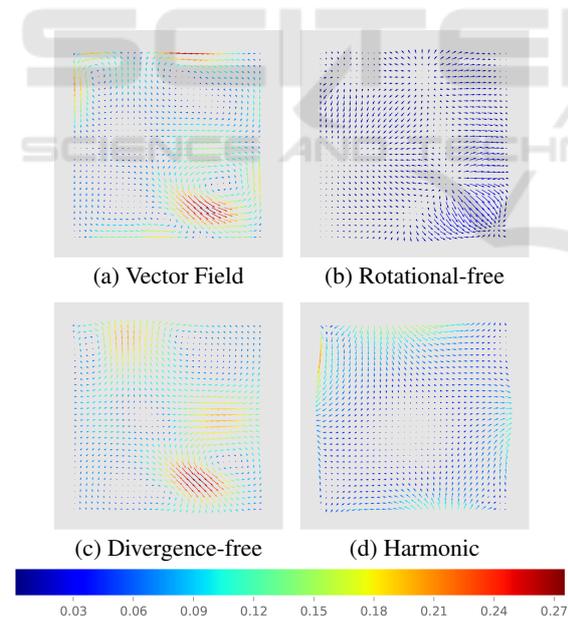


Figure 9: Navier-Stokes simulation and its NHH components.

After generating 100 new realizations through the procedure presented in Algorithm 1, using a kernel of 5×5 , we have a set of realizations \mathcal{E} . Figure 10 shows the MDS for this set. As can be seen, the original sample is surrounded by the new ones.

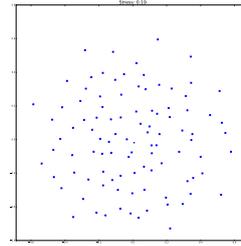


Figure 10: MDS visualization between each new realization and the original realization. Samples generated using a 5×5 kernel.

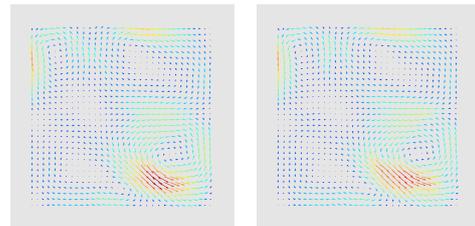


Figure 11: Closest and farthest simulation of the Navier-Stokes vector field.

Figure 11 shows the closest and farthest simulation derived from the original vector field. They are represented using the same magnitude scale as the original field (Figure 9).

With this set in hand, it is now possible to quantify the uncertainty related to the curl operator, which is obtained using partial derivatives. In other words, we can measure the uncertainty related to the kernel used to obtain such attribute. To do so, for each new realization $\mathbf{R} \in \mathcal{E}$, we obtain the curl of \mathbf{R} . We do the same for the original sample \mathbf{V} . To derive the uncertainty of the curl operator, we then compute its mean squared error (RMSE).

In statistics, the mean squared error, MSE, of an estimator is a way to measure the difference between values implied by an estimator and the true values of its target parameter (Wackerly et al., 2008).

For instance, being $\hat{\mathbf{T}}$ the curl of \mathbf{V} and \mathbf{T}_i^* , $i = 1, \dots, 100$ the curl of each one of the generated samples, the MSE of the predictor $\hat{\mathbf{T}}$ is defined as:

$$\text{MSE}(\hat{\mathbf{T}}) = \frac{1}{100} \sum_{i=1}^{100} (\hat{\mathbf{T}} - \mathbf{T}_i^*)^2 \quad (8)$$

The RMSE is given as the square root of the MSE, i.e., $\text{RMSE} = \sqrt{\text{MSE}}$.

Figure 12 presents the RMSE of the curl given the generated realizations.

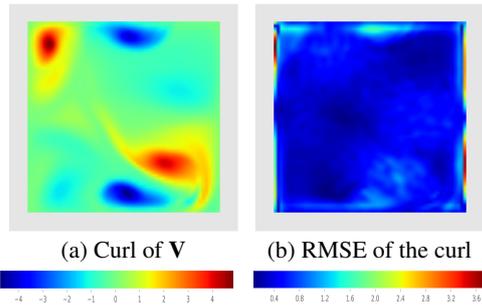


Figure 12: Curl of \mathbf{V} (a) and RMSE of the curl operator (b) between the set \mathcal{E} and the realization \mathbf{V} .

Particle-Image Velocimetry. Often, PIV applications aim to study the behavior of turbulent flows, analyzing the stability of features such as vortices. Besides providing means to perform this kind of study through the generation of different realizations, we can go further with the new samples generated using the proposed technique.

The following PIV simulation is defined over a grid of 124×126 . Its horizontal dimension ranges from 0.3824 to 47.4176. On the other hand, its vertical dimension ranges from 0.3824 to 48.1824. Figure 13 shows this vector field, as well as its NHHD components. This image corresponds to a velocity field of a gas flow that is continuously injected horizontally on the bottom left corner and that flows on the domain from left to right until it meets a wall (image's right edge). It is possible to observe that the divergence-free component seems to have a high magnitude and basically dominates the flow behavior; we can also notice that the rotational-free component presents some features that characterize it.

Figure 14 presents the MDS between the new realizations (generated using a kernel of size 19×19) and the original one. Once again, the training data is surrounded by the generated realizations.

Figure 15 shows the closest and farthest simulation derived from the original vector field. They are represented using the same magnitude scale as their original field (Figure 13).

From Figure 16, we can see that, for the curl operator, the RMSE is higher on regions with high magnitude. In such areas, the scalar field also presents high values. So, a small change in these regions is capable of generating a great change in the vector field. The same behavior happens with the divergence operator, i.e., we have a higher uncertainty in areas where the magnitude of the vector field is also higher.

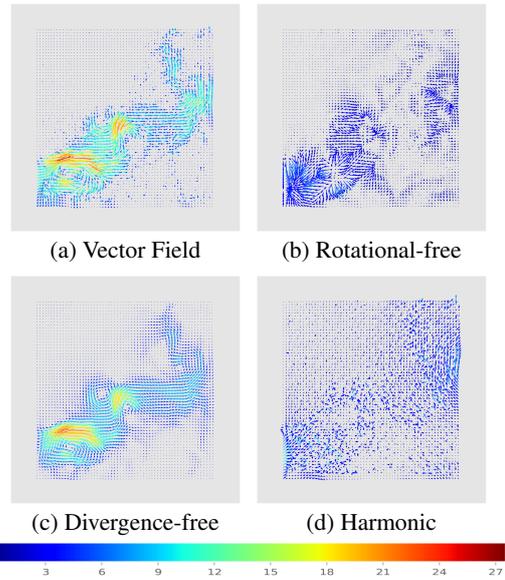


Figure 13: PIV simulation and its NHHD components.

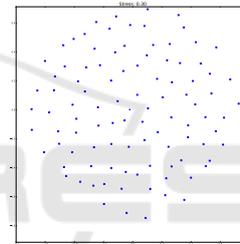
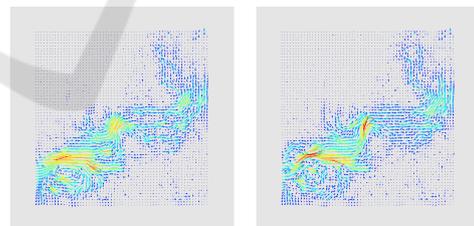


Figure 14: MDS visualization between each new realization and the original realization. Samples generated using a 19×19 kernel.



(a) Closest Simulation (b) Farthest Simulation

Figure 15: Closest and farthest simulation of the Navier-Stokes vector field.

8 PERFORMANCE

Here we present the performance of the proposed technique. Tests were performed using a machine running *ubuntu 16.04 LTS* with the configuration presented in Table 1.

For each data set presented in this paper, we measured the time necessary to compute the NHHD and to generate new realizations (as the mean of the time

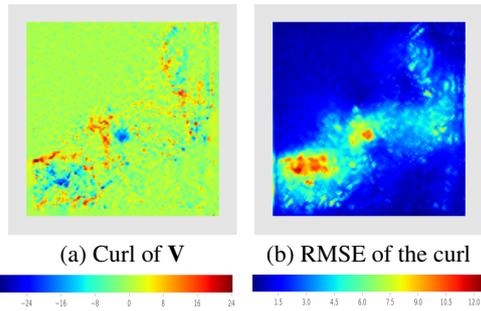


Figure 16: Curl of \mathbf{V} (a) and RMSE of the curl operator (b) between the set \mathcal{E} and the realization \mathbf{V} .

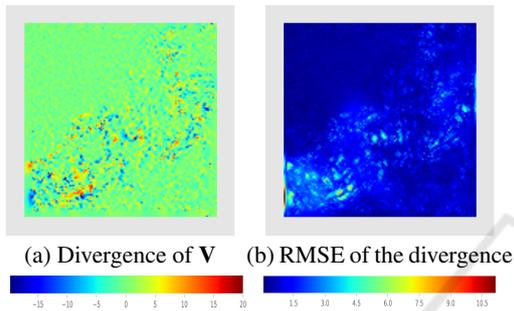


Figure 17: Divergence of \mathbf{V} (a) and RMSE of the divergence operator (b) between the set \mathcal{E} and the realization \mathbf{V} .

Table 1: Machine configuration.

Memory	62.8 GiB
Processor	Intel [®] Core [™] i7-5820K CPU @ 3.30 GHz × 12
Graphics	GeForce GTX 960/PCIe/SSE2
OS Type	64 bit
Disk	55 GB

spent to generate a set with 100 new samples). Those can be seen in Table 2. It is important to notice that all methods here presented, as well as the time measurement, were coded in *Python 2.7*, using the *Numpy* numerical library and the *SciPy* library of scientific tools.

Table 2: Performance of the proposed method per sample, in seconds. Tested using λ equal to 90% for all scenarios and a kernel of 15×15 .

	Forecast ²	Navier-Stokes ³	PIV ⁴
NHHD	1025.775	86.497	1200.545
Samples Gen.	0.584	0.209	0.992

As can be seen, the NHHD is the most time consuming step. For more details on the performance of the NHHD, see (Pascucci et al., 2014).

We also tested the effect of different kernel sizes on the samples generation step. This is shown in Ta-

² 19×19 kernel.

³ 5×5 kernel.

⁴ 19×19 kernel.

ble 3. As we can observe, the size of the kernel didn't cause a significant change in the algorithm performance. It is also interesting to note that, the bigger the size of the kernel the lesser the time consumption. This means that the bootstrap step performance is mostly affected by the number of blocks chosen, instead of the size of the chosen kernel.

Table 3: Performance of the sample generation step for different kernel sizes. Tested with the wind forecast ensemble mean and λ equal to 90%.

11×11	13×13	15×15	17×17
0.613	0.612	0.580	0.577

9 CONCLUSION

This paper proposed a technique to stochastic simulate vector fields given a single realization. Thanks to the Helmholtz-Hodge Decomposition method we could develop a method that provides a good level of realistic scenarios. To the best of our knowledge, this is the first approach that uses the Helmholtz-Hodge Decomposition to stochastic generate vector fields given a training data. Results were evaluated using a set of multi-method wind forecast realizations, as well as simulations from Navier-Stokes and PIV. For each data, 100 new scenarios were generated using the presented method. We applied the *MDS* technique to proper visualize the results; we could observe that the simulated scenarios were able to provide a great variability and that they mimic the training data. The applicability of this approach ranges from uncertainty quantification to data assimilation (Kalnay, 2003). Further studies includes expanding this method for 3-dimensional vector fields, as well as exploring other techniques for random vector field synthesis.

ACKNOWLEDGEMENTS

We would like to thank CAPES and CNPq for the financial support of this research. Additionally, we would like to thank INPE and Haroldo Fraga de Campos Velho for the wind forecast data models used in this work.

REFERENCES

- Abadia, S., Morichona, D., Grillib, S., and Glockner, S. (2010). Numerical simulation of waves generated by landslides using a multiple-fluid NavierStokes model. *Coastal Engineering*, 57(9):779–794.

- Anderson, J. D. and Wendt, J. (1995). *Computational fluid dynamics*, volume 206. Springer.
- Beccali, M., Cellura, M., and Mistretta, M. (2003). Decision-making in energy planning. application of the electre method at regional level for the diffusion of renewable energy technology. *Renewable Energy*, 28(13):2063–2087.
- Bertalmio, M., Bertozzi, A. L., and Sapiro, G. (2001). Navier-stokes, fluid dynamics, and image and video inpainting. In *Proceedings of the 2001 IEEE Computer Society Conference*, pages 355–362. IEEE.
- Bhatia, H., Norgard, G., Pascucci, V., and Bremer, P. (2013). The Helmholtz-Hodge Decomposition - A Survey. *IEEE Transactions on Visualization and Computer Graphics*, 19(8):1386–1404.
- Cho, K., Meer, P., and Cabrera, J. (1997). Performance assessment through bootstrap. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1185–1198.
- Chorin, A. J. (1968). Numerical solution of the Navier-Stokes equations. *Mathematics of Computation*, 22:745–762.
- Chorin, A. J. and Marsden, J. E. (1993). *A mathematical introduction to fluid mechanics*. Texts in Applied Mathematics. Springer-Verlag, 3rd edition.
- Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *Annals of Statistics*, 7:1–26.
- Efros, A. A. and Freeman, W. T. (2001). Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM.
- Fu, X., You, H., and Fu, K. (2012). A Statistical Approach to Detect Edges in SAR Images Based on Square Successive Difference of Averages. *IEEE Geoscience And Remote Sensing Letters*, 9(6):1094–1098.
- Gao, H., Mandal, M. K., Guo, G., and Wan, J. (2010). Singular point detection using Discrete Hodge Helmholtz Decomposition in fingerprint images. In *ICASSP'10*, pages 1094–1097. IEEE.
- Gonzalez, R. C. and Woods, R. E. (2006). *Digital Image Processing*. Prentice-Hall, Inc.
- Guo, Q., Mandal, M. K., Liu, G., and Kavanagh, K. M. (2006). Cardiac video analysis using Hodge-Helmholtz field decomposition. *Computers in Biology and Medicine*, 36(1):1–20.
- Honarkhah, M. and Caers, J. (2010). Stochastic Simulation of Patterns Using Distance-Based Pattern Modeling. *Mathematical Geosciences*, 42(5):487–517.
- Kalnay, E. (2003). *Atmospheric modeling, data assimilation and predictability*. Cambridge university press.
- Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):1–27.
- Lall, U., Devineni, N., and Kaheil, Y. (2016). An empirical, nonparametric simulator for multivariate random variables with differing marginal densities and nonlinear dependence with hydroclimatic applications. *Risk Analysis*, 36(1):57–73.
- Lantuéjoul, C. (2013). *Geostatistical simulation: models and algorithms*. Springer Science & Business Media.
- Liew, K. J., Ramli, A., and Majid, A. A. (2016). Searching for the optimum value of the smoothing parameter for a radial basis function surface with feature area by using the bootstrap method. *Computational and Applied Mathematics*, pages 1–16.
- Luo, C., Safa, I., and Wang, Y. (2012). Feature-aware streamline generation of planar vector fields via topological methods. *Computers & Graphics*, 36(6):754–766.
- Mariethoz, G. and Caers, J. (2014). *Multiple-point geostatistics: stochastic modeling with training images*. John Wiley & Sons.
- Mariethoz, G. and Lefebvre, S. (2014). Bridges between multiple-point geostatistics and texture synthesis: Review and guidelines for future research. *Computers & Geosciences*, 66:66–80.
- Oliver, M. and Webster, R. (2014). A tutorial guide to geostatistics: Computing and modelling variograms and kriging. *Catena*, 113:56–69.
- Palit, B. (2005). *Application of the Hodge Helmholtz Decomposition to Video and Image Processing*. Master's thesis, University of Alberta.
- Pascucci, V., Bremer, P.-T., and Bhatia, H. (2014). The Natural Helmholtz-Hodge Decomposition For Open-Boundary Flow Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 99(PrePrints):1–11.
- Petronetto, F., Paiva, A., Lage, M., Tavares, G., Lopes, H., and Lewiner, T. (2010). Meshless Helmholtz-Hodge Decomposition. *IEEE Transactions on Visualization and Computer Graphics*, 16(2):338–349.
- Polthier, K. and Preuß, E. (2003). Identifying Vector Field Singularities Using a Discrete Hodge Decomposition. In *Visualization and Mathematics III*, pages 113–134. Springer Berlin Heidelberg.
- Popescu, R., Deodatis, G., and Prevost, J. H. (1998). Simulation of homogeneous nongaussian stochastic vector fields. *Probabilistic Engineering Mechanics*, 13(1):1–13.
- Ribeiro, P. C., de Campos Velho, H. F., and Lopes, H. (2016). Helmholtz-hodge decomposition and the analysis of 2d vector field ensembles. *Computers & Graphics*, 55:80–96.
- Tong, Y., Lombeyda, S., Hirani, A. N., and Desbrun, M. (2003). Discrete Multiscale Vector Field Decomposition. In *ACM SIGGRAPH 2003 Papers*, pages 445–452. ACM.
- Wackerly, D. D., Mendenhall, W., and Scheaffer, R. L. (2008). *Mathematical Statistics with Applications*. Thomson.
- Wang, H. and Deng, J. (2014). Feature extraction of complex ocean flow field using the helmholtz-hodge decomposition. In *2013 IEEE International Conference on Multimedia and Expo Workshops*, pages 1–6. IEEE.
- Wang, Z., Wang, X., and Ji, Q. (2014). Learning with hidden information. In *ICPR*, pages 238–243.
- Wasserman, L. (2004). *All of Statistics: A Concise Course in Statistical Inference*. Springer.
- Xiu, D. (2009). Fast numerical methods for stochastic computations: a review. *Communications in computational physics*, 5(2-4):242–272.

Table 4: Original realization and its closest and farthest simulation.

Realization	Closest Simulation	Farthest Simulation
