Computers & Geosciences 195 (2025) 105767

Contents lists available at ScienceDirect

Computers and Geosciences

journal homepage: www.elsevier.com/locate/cageo

Research paper

Functional multiple-point simulation

Oluwasegun Taiwo Ojo [a,*], Marc G. Genton [b]

[a] uc3m-Santander Big Data Institute, Universidad Carlos III de Madrid, Getafe, Madrid, Spain
[b] Statistics Program, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia

ARTICLE INFO

Keywords:
Functional data analysis
Functional kriging
Geostatistics
Multiple-point simulation
Spatial functional data
Wind profile

ABSTRACT

We present a new paradigm, called functional multiple-point simulation, in which multiple-point geostatistical simulation can be performed when functions or curves are observed at each location of a random field. Multiple-point simulation is a non-parametric method used for conditional geostatistical simulation of complex spatial patterns by inferring multiple-point statistics from a training image, rather than from a two-point variogram or covariance model. When the observable at each spatial location is a functional random variable, such multiple-point simulation must take into account not only the spatial correlation among locations but also the similarity of functions or curves observed at each location. The data events to be compared in this case are now functional, in the sense that they consist of spatial arrangements of functions. Consequently, we propose four distances, inspired by the functional data analysis literature, for measuring similarities between functional data events and use these to extend the direct sampling method to perform multiple-function geostatistical simulation with functional fields. We coin the new method Functional Direct Sampling and carry out extensive qualitative and quantitative performance comparison between the four proposed distances using simulation techniques on two well-known applications of multiple-point simulation: simulating copies of a functional random field and gap-filling of locations in a functional random field. We apply the proposed method to a gap-filling task of simulated wind profiles spatial functions over the Arabian Peninsula.

1. Introduction

Spatially correlated functional data are functional data with a spatial component, i.e., observations are curves $X(\mathbf{s}_i, t); t \in [0, T]$, observed at spatial locations $\{\mathbf{s}_i\}_{i=1}^n \subset D$, where D is the study area. Such data are ubiquitous since functional data are usually collected at a location, with spatial dependence expressed in the spatial arrangement of the functions in the study area. An example of spatial functional data (SFD) is wind profiles data containing wind speed at different pressure levels (or altitudes) for locations in the study area.

Statistical methods for analysing SFD in the literature include methods for dimension reduction (Liu et al., 2017; Kuenzer et al., 2021; Zhang and Li, 2022), mean estimation, hypothesis testing and modelling (Gromenko et al., 2012; Gromenko and Kokoszka, 2013; Arnone et al., 2019; Rachdi et al., 2021; Římalová et al., 2022; White et al., 2021; Liang et al., 2022; Hörmann et al., 2022), and clustering (Giraldo et al., 2012a; Romano and Verde, 2012; Romano et al., 2015; Abramowicz et al., 2017; Vandewalle et al., 2022). Attempts at extending geostatistical methods to SFD include kriging and regression methods for spatial prediction of curves (Nerini et al., 2010; Giraldo et al., 2011; Menafoglio et al., 2013; Caballero et al., 2013; Ignaccolo et al., 2014; Menafoglio et al., 2016; Aguilera-Morillo et al., 2017) and test for spatial autocorrelation (Giraldo et al., 2018). Recent surveys of the latest developments in inferential and geostatistical methods for spatial functional data include Menafoglio and Secchi (2017), Kokoszka and Reimherr (2019), Martínez-Hernández and Genton (2020), and Li et al. (2022, chap. 5). The compendium on geostatistical functional data analysis (Mateu and Giraldo, 2021) is also a comprehensive collection of the latest advances.

This work aims to extend multiple-point statistics (MPS) to SFD. MPS is a non-parametric method used for conditional geostatistical simulation of complex spatial structures (e.g., curve-linear geological features) by inferring multiple-point statistics from a training image (TI), rather than from a two-point variogram (Guardiano and Srivastava, 1993). Typically, the TI is a representative conceptual model from which spatial statistics are copied and used for simulation. MPS has found extensive applications in geology (Oriani et al., 2016), remote sensing (Vannametee et al., 2014), and imaging (Yin et al., 2017a,b; Pham, 2012).

MPS was introduced by Guardiano and Srivastava (1993), when they proposed the extended normal equations simulation (ENESIM) algorithm. ENESIM aims to estimate the conditional probability $\text{Prob}(Z(s) = c_k | d_n)$ (that a random field Z takes on a class c_k out

* Corresponding author.
E-mail address: seguntaiwoojo@gmail.com (O.T. Ojo).

https://doi.org/10.1016/j.cageo.2024.105767
Received 7 March 2024; Received in revised form 6 November 2024; Accepted 6 November 2024
Available online 14 November 2024
0098-3004/© 2024 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

of K possible classes $\{c_k|k = 1,\dots,K\}$ at a location $\boldsymbol{s}$, given the surrounding data event d_n) directly from a TI. The estimation of $\text{Prob}(Z(\boldsymbol{s}) = c_k|d_n)$ requires computing statistics of multiple points in the TI which can better reproduce complex features. This method had some drawbacks that limited its use including being computationally intensive and limited to categorical variables. Other MPS methods like snesim (Strebelle, 2002), filtersim (Zhang et al., 2006; Wu et al., 2008), direct sampling (Mariethoz et al., 2010) and quick sampling (Gravey and Mariethoz, 2020) were subsequently proposed.

We propose a new method for performing multiple-point geostatistical simulation with spatial functional data observed on a regular grid. We assume each observation is a realization of a functional random variable observed at a grid point in a study area which is partitioned into a finite number of regular grid points. Each simulated "point" at a location is a function conditioned on the data event made up of neighbouring functions. The TI is an SFD from which "multiple-point" statistics are computed (in reality, we are considering "multiple-function" statistics). The proposed method is useful for generating realizations of the unknown functional random field (FRF) underlying the TI, as well as for gap-filling applications.

In Section 2, we provide some background on MPS. We elaborate our proposal for functional MPS in Section 3 followed by some simulation studies in Section 4. We apply the proposed method to simulated wind profiles data over the Arabian Peninsula in Section 5 and conclude the article in Section 6.

2. Multiple-point geostatistical simulation

2.1. Multiple-point statistics

This section contains a brief background on geostatistical simulation using MPS (see Mariethoz and Caers, 2014 for a thorough coverage). MPS aims to use a TI to generate realizations that are conditioned on the features found in the TI. Usually, the TI shares some characteristics with the underlying phenomenon of interest although the TI itself is not the phenomenon (Guardiano and Srivastava, 1993). At each pixel location $\boldsymbol{s}$ in the TI, the pixel value, denoted by $Z(\boldsymbol{s})$, can only take on K different classes, $\{c_k|k = 1,\dots,K\}$, e.g., categorical outcomes. A simulation grid (SG) is created, and a few conditioning data (usually obtained by sampling a few pixel values from the TI) are assigned to the SG. The simulation then proceeds to fill up the remaining locations in the SG. For a location $\boldsymbol{s}$ to be simulated, the data event d_n around $\boldsymbol{s}$ is a set containing the values of its n−nearest neighbours:

$$d_n = \{Z(\boldsymbol{s}+\boldsymbol{h}_\alpha)|\ \alpha = 1,\dots,n\}, \tag{1}$$

where $\boldsymbol{h}_\alpha$ is the lag vector $\boldsymbol{s}_\alpha - \boldsymbol{s}$ indicating the displacement of location $\boldsymbol{s}_\alpha$ from $\boldsymbol{s}$. Let $\mathcal{I}_k(\boldsymbol{s}) := \mathbf{1}\{\text{if } Z(\boldsymbol{s}) = c_k\}$ indicate the occurrence of class c_k at $\boldsymbol{s}$ and let $D := \mathbf{1}\{\text{if } Z(\boldsymbol{s}+\boldsymbol{h}_\alpha) = c_{k_\alpha},\ \forall\ \alpha = 1,\dots,n\}$ indicate the occurrence of the data event centred on $\boldsymbol{s}$, where $\{c_{k_\alpha}, |\alpha = 1,\dots,n\}$, are the values of the neighbours of $\boldsymbol{s}$. ENESIM then uses an estimate (see Section S-I of the Supplementary Material) of the probability: $\text{Prob}(\mathcal{I}_k(\boldsymbol{s}) = 1|D = 1)$ to simulate a value for $\boldsymbol{s}$ from the K classes, which is then added to the grid and considered as part of the data event for subsequent simulations of the other locations.

ENESIM's support for only binary classes and its computational burden limits its application. Strebelle (2002) proposed the snesim algorithm which slightly reduces the computational burden, but it is still limited to categorical variables with a small number of classes.

2.2. Direct sampling method

The direct sampling technique (Mariethoz et al., 2010), unlike ENESIM and snesim, can deal with RFs taking continuous values. The multiple-point statistics are expressed as the cumulative distribution function (CDF) $F(z,\boldsymbol{s},d_n) = \text{Prob}(Z(\boldsymbol{s}) \le z|d_n)$ and the idea is that it is unnecessary to estimate $F(z,\boldsymbol{s},d_n)$ by counting data events found in the TI. Instead, values satisfying this CDF can be sampled from the TI and pasted into the SG. For each location $\boldsymbol{v}$, the data event d_n around $\boldsymbol{v}$ is compared to configurations of pixels (of the same size as d_n) in the TI using a distance metric. Once a pattern that matches (or is similar to) d_n is found in the TI, the central pixel of that pattern is directly pasted into the SG. The simulation process then moves on to the next location to be simulated. We provide an outline of the algorithm in Algorithm 1.

Algorithm 1 Direct Sampling Algorithm

Input: Training Image TI, number of neighbours n, distance acceptance threshold γ and a fraction f of the TI, empty Simulation Grid (SG)

1. Assign some conditioning data to the simulation grid (SG). The first conditioning datum can be drawn from the TG and then the simulation continues on from this datum;
2. Define a random or unilateral filling path through the SG;

for each location $\boldsymbol{v}$ to be simulated along the SG path **do**
3. Find the n−nearest neighbours of $\boldsymbol{v}$;
4. Compute the lag vectors $L = \{\boldsymbol{h}_1,\boldsymbol{h}_2,\dots,\boldsymbol{h}_n\} = \{\boldsymbol{v}_\alpha - \boldsymbol{v}|\alpha = 1,\dots,n\}$, and the neighbourhood set $N(\boldsymbol{v},L) = \{\boldsymbol{v}+\boldsymbol{h}_\alpha|\alpha = 1,\dots,n\}$;
5. Compute data event around $\boldsymbol{v}$: $d_n(\boldsymbol{v},L) = \{Z(\boldsymbol{v}+\boldsymbol{h}_\alpha)|\alpha = 1,\dots,n\}$.
6. Define the search window in the TI: $\{\boldsymbol{u} \in TI\ :\ N(\boldsymbol{u},L) \in TI\}$
for each location $\boldsymbol{u}$ in the search window: **do**
7. Compute the data event $d_n(\boldsymbol{u},L)$;
8. Compute the distance $D(d_n(\boldsymbol{u}),d_n(\boldsymbol{v}))$ between $d_n(\boldsymbol{u},L)$ and $d_n(\boldsymbol{v},L)$;
9. Store $\boldsymbol{u}$ and $Z(\boldsymbol{u})$ if distance $D(d_n(\boldsymbol{u}),d_n(\boldsymbol{v}))$ is the lowest distance obtained so far in the search window
10. If $D(d_n(\boldsymbol{u}),d_n(\boldsymbol{v})) < \gamma$, or the number of possible locations $\boldsymbol{u}$ already considered in the search window is more than the fraction f of the TI, set $Z(\boldsymbol{v}) := Z(\boldsymbol{u})$ and proceed to simulate the next location in the SG.
end for
end for

Output: Filled Simulation Grid SG

The distance measure used for comparing the data events of the location to be simulated ($d_n(\boldsymbol{v},L)$) and the candidate location ($d_n(\boldsymbol{u},L)$) depends on the nature of the variables considered. For categorical random fields, a distance measuring the proportion of mismatched pixels is appropriate. Thus $D(d_n(\boldsymbol{u}),d_n(\boldsymbol{v}))$ can be set to:

$$D_a(d_n(\boldsymbol{u}),d_n(\boldsymbol{v})) = \frac{1}{n}\sum_{\alpha=1}^{n}\psi_\alpha \quad \in [0,1], \tag{2}$$

where ψ_α is $\psi_\alpha := \mathbf{1}\{Z(\boldsymbol{v}_\alpha) \neq Z(\boldsymbol{u}_\alpha)\}$, for $Z(\boldsymbol{v}_\alpha) \in d_n(\boldsymbol{v},L)$ and $Z(\boldsymbol{u}_\alpha) \in d_n(\boldsymbol{u},L)$. A weighted version of $D_a(d_n(\boldsymbol{u}),d_n(\boldsymbol{v}))$ which assigns more weight to pixels nearer to the centre of the data events can be considered:

$$D_a(d_n(\boldsymbol{u}),d_n(\boldsymbol{v})) = \frac{\sum_{\alpha=1}^{n}\psi_\alpha\|\boldsymbol{h}_\alpha\|^{-\delta}}{\sum_{\alpha=1}^{n}\|\boldsymbol{h}_\alpha\|^{-\delta}} \quad \in [0,1], \tag{3}$$

where $\delta \ge 0$ is a weighting factor. For continuous variables, a weighted Euclidean distance works:

$$D_b(d_n(\boldsymbol{u}),d_n(\boldsymbol{v})) = \sqrt{\sum_{\alpha=1}^{n}\omega_\alpha\{Z(\boldsymbol{v}_\alpha) - Z(\boldsymbol{u}_\alpha)\}^2} \quad \in [0,1], \tag{4}$$

where

$$\omega_\alpha = \frac{\|\boldsymbol{h}_\alpha\|^{-\delta}}{d_{max}^2\sum_{j=1}^{n}\|\boldsymbol{h}_\alpha\|^{-\delta}}, \qquad d_{max} = \max_{\boldsymbol{u}\in TI}Z(\boldsymbol{u}) - \min_{\boldsymbol{u}\in TI}Z(\boldsymbol{u}), \qquad \delta \ge 0. \tag{5}$$

The distance measures D_a and D_b are all normalized to the interval $[0,1]$. This is helpful in selecting an acceptance threshold γ for the simulation. Other distance measures were proposed to deal with multivariate random fields and non-stationary random fields.

3. Functional multiple-point simulation

3.1. Functional direct sampling algorithm

In this section, we outline our proposal for conducting multiple-point simulations on SFD. It is assumed that there is a "training image" (hereafter referred to as the training grid TG), that is an SFD: $X(\mathbf{s}_i, t)$, observed on a grid of regularly spaced locations $\{\mathbf{s}_i\}_{i=1}^{r} \subset R \in \mathbb{R}^2$, the study area. Thus, each location in the TG is no longer a single value but a function. For simplicity, we assume that these functions are observed in the space of square-integrable functions on the interval $[0, 1]$, i.e., $X(\mathbf{s}_i, t) \in L^2[0, 1]$ (because we can define an inner product and norm on functions in this space).

The simulation grid (SG) is also assumed to be a regular grid, but not necessarily of the same size as the TG. Some initial conditioning data, which can also be an unconditional simulation from the TG (to avoid inconsistencies), are randomly assigned to the SG. The goal is to use MPS to fill up the locations of the SG with functions conditioned on the data events. We outline in Algorithm 2, a new algorithm, which we coin "Functional Direct Sampling" (FDS), to achieve this.

Algorithm 2 Functional Direct Sampling (FDS) Algorithm

Input: Training Grid TG, number of neighbours n, empty or partially filled Simulation Grid (SG)

1. Assign conditioning data to the simulation grid (SG). The first conditioning datum can be drawn from the TG and then the simulation continues on from this datum;
2. Define a random or unilateral filling path through the SG;

for each location $\boldsymbol{v}$ to be simulated along the SG path **do**
3. Find the $n-$nearest neighbours of $\boldsymbol{v}$ with curves/functions already assigned;
4. Compute the lag vectors $L = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \dots, \boldsymbol{h}_n\} = \{\boldsymbol{v}_\alpha - \boldsymbol{v} | \alpha = 1, \dots, n\}$, and the neighbourhood set $N(\boldsymbol{v}, L) = \{\boldsymbol{v} + \boldsymbol{h}_\alpha | \alpha = 1, \dots, n\}$;
5. Compute the "functional data event" around $\boldsymbol{v}$: $d_n(\boldsymbol{v}, L)(t) = \{Z(\boldsymbol{v} + \boldsymbol{h}_\alpha, t) | \alpha = 1, \dots, n\}$.
6. Define the search window in the TG: $\{\boldsymbol{u} \in TG | N(\boldsymbol{u}, L) \in TG\}$
for each location $\boldsymbol{u}$ in the search window: **do**
7. Compute the data event $d_n(\boldsymbol{u}, L)(t)$;
8. Compute the distance $D(d_n(\boldsymbol{u}, L)(t), d_n(\boldsymbol{v}, L)(t))$ between $d_n(\boldsymbol{u}, L)(t)$ and $d_n(\boldsymbol{v}, L)(t)$;
9. Store $\boldsymbol{u}$ and $Z(\boldsymbol{u}, t)$ if $D(d_n(\boldsymbol{u}, L)(t), d_n(\boldsymbol{v}, L)(t))$ is the lowest distance obtained so far in the search window;
end for
10. Copy the function of $\boldsymbol{u}$ with the lowest distance to location $\boldsymbol{v}$ in the SG, i.e., set $Z(\boldsymbol{v}, t) := Z(\boldsymbol{u}_{min}, t)$ where $\boldsymbol{u}_{min} = \min_{\boldsymbol{u} \in TG} D(d_n(\boldsymbol{u}, L)(t), d_n(\boldsymbol{v}, L)(t))$.
end for

Output: Filled Simulation Grid SG

Since functional observations can vary in different ways (e.g., functions can have different magnitudes but the same shape), we propose distances that can be used for comparing functional data events.

3.2. Distances based on norms and seminorms

The space $L^2[0, 1]$ is endowed with $\|f\| = \sqrt{\langle f, f \rangle} = \left(\int [f(t)]^2 \mathrm{d}t\right)^{1/2}$. For functions $f, g \in L^2$, the distance between f and g is: $\|f - g\|$. We can compare two data events $d_n(\boldsymbol{u}, L)(t)$ and $d_n(\boldsymbol{v}, L)(t)$ using the weighted sum of the distance between their constituent functions:

$$D_1(d_n(\boldsymbol{u})(t), d_n(\boldsymbol{v})(t)) = \left(\sum_{\alpha=1}^{n} \omega_\alpha \|Z(\boldsymbol{u} + \boldsymbol{h}_\alpha, t) - Z(\boldsymbol{v} + \boldsymbol{h}_\alpha, t)\|^2\right)^{1/2} \tag{6}$$

$$= \left(\sum_{\alpha=1}^{n} \omega_\alpha \int_0^1 \left[Z(\boldsymbol{u} + \boldsymbol{h}_\alpha, t) - Z(\boldsymbol{v} + \boldsymbol{h}_\alpha, t)\right]^2 \mathrm{d}t\right)^{1/2}, \tag{7}$$

where ω_α, $\alpha = 1, \dots, n$ are weights which can depend on $\boldsymbol{h}_\alpha$. We can also compare data events using the average normalized distance:

$$D_2(d_n(\boldsymbol{u})(t), d_n(\boldsymbol{v})(t)) = \frac{1}{n} \sum_{\alpha=1}^{n} \frac{\|Z(\boldsymbol{u} + \boldsymbol{h}_\alpha, t) - Z(\boldsymbol{v} + \boldsymbol{h}_\alpha, t)\|}{dmax}, \tag{8}$$

where $dmax = \max_{\boldsymbol{x}, \boldsymbol{y} \in \mathrm{TG}} \|Z(\boldsymbol{x}, t) - Z(\boldsymbol{y}, t)\|$.

Distances based on seminorms (Ferraty and Vieu, 2006) of derivatives of functions are useful for considering differences in velocity and acceleration of functions data events:

$$D_3^{(q)}(d_n(\boldsymbol{u})(t), d_n(\boldsymbol{v})(t)) = \left(\sum_{\alpha=1}^{n} \omega_\alpha \int_0^1 \left[Z(\boldsymbol{u} + \boldsymbol{h}_\alpha, t)^{(q)} - Z(\boldsymbol{v} + \boldsymbol{h}_\alpha, t)^{(q)}\right]^2 \mathrm{d}t\right)^{1/2}, \tag{9}$$

where $q \in \{1, 2\}$ and $Z(\boldsymbol{u} + \boldsymbol{h}_\alpha, t)^{(q)}$ indicate the qth derivative of the function $Z(\boldsymbol{u} + \boldsymbol{h}_\alpha, t)$.

3.3. Pseudo-distance based on FastMUOD indices

The fast massive unsupervised outlier detection (FastMUOD) indices proposed in Ojo et al. (2021) compute for each curve, a shape, amplitude, and magnitude index which measures the outlyingness of that curve in terms of shape, amplitude and magnitude. We adapt these indices to compare the shapes, amplitudes, and magnitudes of corresponding functions in the data events. For $\boldsymbol{v} \in$ SG and $\boldsymbol{u} \in$ TG, define the shape pseudo-distance between the data events of $\boldsymbol{u}$ and $\boldsymbol{v}$ as:

$$S_{\boldsymbol{u},\boldsymbol{v}} = S(d_n(\boldsymbol{u}, L)(t), d_n(\boldsymbol{v}, L)(t)) := \frac{1}{n} \sum_{\alpha=1}^{n} \hat{\rho}_\alpha, \tag{10}$$

where $\hat{\rho}_\alpha = 1 - \hat{\rho}(Z(\boldsymbol{v} + \boldsymbol{h}_\alpha, t), Z(\boldsymbol{u} + \boldsymbol{h}_\alpha, t))$ and $\hat{\rho}(Z(\boldsymbol{v} + \boldsymbol{h}_\alpha, t), Z(\boldsymbol{u} + \boldsymbol{h}_\alpha, t))$ is the estimated Pearson correlation coefficient between the observed points of $Z(\boldsymbol{v} + \boldsymbol{h}_\alpha, t)$ and the observed points of $Z(\boldsymbol{u} + \boldsymbol{h}_\alpha, t)$. Here, $S_{\boldsymbol{u},\boldsymbol{v}}$ measures the average similarity in terms of shape (quantified by the Pearson correlation) between $d_n(\boldsymbol{u}, L)(t)$ and $d_n(\boldsymbol{v}, L)(t)$. We define the amplitude pseudo-distance between $d_n(\boldsymbol{u}, L)(t)$ and $d_n(\boldsymbol{v}, L)(t)$ as:

$$A_{\boldsymbol{u},\boldsymbol{v}} = A(d_n(\boldsymbol{u}, L)(t), d_n(\boldsymbol{u}, L)(t)) := \frac{1}{n} \sum_{\alpha=1}^{n} |\beta_\alpha - 1|, \tag{11}$$

where $\beta_\alpha = \widehat{\mathrm{Cov}}(Z(\boldsymbol{v}+\boldsymbol{h}_\alpha, t), Z(\boldsymbol{u}+\boldsymbol{h}_\alpha, t))/\hat{\sigma}^2_{Z(\boldsymbol{v}+\boldsymbol{h}_\alpha, t)}$, $\widehat{\mathrm{Cov}}(Z(\boldsymbol{v}+\boldsymbol{h}_\alpha, t), Z(\boldsymbol{u}+\boldsymbol{h}_\alpha, t))$ is the sample covariance between the observed points of $Z(\boldsymbol{v} + \boldsymbol{h}_\alpha, t)$ and $Z(\boldsymbol{u} + \boldsymbol{h}_\alpha, t)$, and $\hat{\sigma}^2_{Z(\boldsymbol{v}+\boldsymbol{h}_\alpha, t)}$ is the sample variance of the observed points of $Z(\boldsymbol{v} + \boldsymbol{h}_\alpha, t)$. The term β_α is the estimated slope of a linear regression between the observed points of $Z(\boldsymbol{v} + \boldsymbol{h}_\alpha, t)$ and $Z(\boldsymbol{u}+\boldsymbol{h}_\alpha, t)$. If $Z(\boldsymbol{v}+\boldsymbol{h}_\alpha, t)$ and $Z(\boldsymbol{u}+\boldsymbol{h}_\alpha, t)$ have the same amplitude, then β_α will be close to 1, otherwise, β_α will have a value different from 1. Thus, lower values of $A_{\boldsymbol{u},\boldsymbol{v}}$ indicate that the functions in $d_n(\boldsymbol{u}, L)(t)$ and $d_n(\boldsymbol{v}, L)(t)$ have similar amplitude on average. We note that the pseudo-distance $A_{\boldsymbol{u},\boldsymbol{v}}$ also captures differences in phases between two functions, which is a desirable property as two functions with the same amplitude, but shifted in phase will have a high $A_{\boldsymbol{u},\boldsymbol{v}}$. Finally define the magnitude pseudo-distance between $d_n(\boldsymbol{u}, L)(t)$ and $d_n(\boldsymbol{v}, L)(t)$ as:

$$M_{\boldsymbol{u},\boldsymbol{v}} = M(d_n(\boldsymbol{u}, L)(t), d_n(\boldsymbol{v}, L)(t)) := \frac{1}{n} \sum_{\alpha=1}^{n} |\nu_\alpha|, \tag{12}$$

where $\nu_\alpha = \bar{Z}(\boldsymbol{u} + \boldsymbol{h}_\alpha, t) - \beta_\alpha \bar{Z}(\boldsymbol{v} + \boldsymbol{h}_\alpha, t)$ and $\bar{Z}(\boldsymbol{u} + \boldsymbol{h}_\alpha, t)$ is the mean of the observed points of $Z(\boldsymbol{u} + \boldsymbol{h}_\alpha, t)$. Again, ν_α is the estimated intercept of the linear regression between the observed points of $Z(\boldsymbol{v} + \boldsymbol{h}_\alpha, t)$ and $Z(\boldsymbol{u} + \boldsymbol{h}_\alpha, t)$. If $Z(\boldsymbol{v} + \boldsymbol{h}_\alpha, t)$ and $Z(\boldsymbol{u} + \boldsymbol{h}_\alpha, t)$ are similar in magnitude, then ν_α will be close to zero. Thus, a low value of $M_{\boldsymbol{u},\boldsymbol{v}}$ indicates that the functions of $d_n(\boldsymbol{u}, L)(t)$ and $d_n(\boldsymbol{v}, L)(t)$ have similar magnitude on average.

Finally, we define the FastMUOD pseudo-distance as the weighted sum of $S_{\boldsymbol{u},\boldsymbol{v}}$, $A_{\boldsymbol{u},\boldsymbol{v}}$, and $M_{\boldsymbol{u},\boldsymbol{v}}$ after normalization:

$$D_4(d_n(\boldsymbol{u}, L)(t), d_n(\boldsymbol{v}, L)(t)) := \eta_1 S_{\boldsymbol{u},\boldsymbol{v}} + \eta_2 A'_{\boldsymbol{u},\boldsymbol{v}} + \eta_3 M'_{\boldsymbol{u},\boldsymbol{v}}, \tag{13}$$

where $A'_{\boldsymbol{u},\boldsymbol{v}} = A_{\boldsymbol{u},\boldsymbol{v}} / \max_{\boldsymbol{u}\in \mathrm{TI}} A_{\boldsymbol{u},\boldsymbol{v}}$, and $M'_{\boldsymbol{u},\boldsymbol{v}} = M_{\boldsymbol{u},\boldsymbol{v}} / \max_{\boldsymbol{u}\in \mathrm{TI}} M_{\boldsymbol{u},\boldsymbol{v}}$, and $\sum_{i=1}^{3} \eta_i = 1$. The weights η_i allow to place more emphasis on any of the shape, amplitude and magnitude indices.

3.4. Differences between FDS and the classical direct sampling algorithm

The FDS algorithm outlined in Algorithm 2 slightly differs from the classical direct sampling algorithm in that the distance threshold γ and the fraction of TG f are not considered. While this simplifies the application of FDS (since there are less parameters to tune) and works for small SG and TG, there are possible disadvantages to this setup.

First, FDS always select the best possible candidate in the TI, which may lead to successive simulations being exactly the same (lack of randomness). Always selecting the best candidate has also been shown to produce simulations containing verbatim copies of parts of the TG, also called "patching" (Meerschman et al., 2013), especially in larger grids where the TG does not show enough pattern repeatability. Second, FDS scans all the candidate points in the search window, which might be computationally intensive for large TGs.

The distance threshold γ and fraction of TG to scan f can be incorporated into FDS (by imposing the sampling condition using γ and f after step 9 in Algorithm 2). To aid with selecting an appropriate threshold γ, the proposed distances can be scaled into $[0, 1]$. For example, distance D_1 can be scaled into $[0, 1]$ by dividing by the maximum squared distance in the TG:

$$D_1(d_n(\boldsymbol{u})(t), d_n(\boldsymbol{v})(t)) = \left(\sum_{\alpha=1}^{n} \frac{\omega_\alpha}{dmax^2} \| Z(\boldsymbol{u} + \boldsymbol{h}_\alpha, t) - Z(\boldsymbol{v} + \boldsymbol{h}_\alpha, t) \|^2 \right)^{1/2}, \tag{14}$$

where $dmax = \max_{\boldsymbol{x},\boldsymbol{y}\in \mathrm{TG}} \| Z(\boldsymbol{x}, t) - Z(\boldsymbol{y}, t) \|$. Similar to the classical direct sampling algorithm, our simulation experiments show that values of $\gamma \leq 0.1$ and $f \geq 0.5$ tend to produce good results (Meerschman et al., 2013). In Section 4.1.4, we show FDS simulations incorporating both γ and f.

4. Simulation study

We evaluate the performance of FDS using some simulation experiments, consisting of two application scenarios in which MPS is used: simulating new copies of an existing functional random field (FRF), using the random field as a training grid (TG); and filling gaps in an existing FRF using a TG.

4.1. Simulating copies of a functional random field

4.1.1. Simulation settings

To test the effectiveness of FDS using the four distances proposed in Section 3, we simulate two FRFs in a rectangular grid consisting of 51×51 locations with coordinates in $[0, 1] \times [0, 1]$. To generate the function at each location $\boldsymbol{s}_i \in [0, 1] \times [0, 1]$, we use the model:

$$X(\boldsymbol{s}_i, t) = \sum_{m=1}^{M} \phi_m(\boldsymbol{s}_i) \psi_m(t), \tag{15}$$

where ϕ_m, $m = 1, \ldots, M$, are realizations of a Gaussian random field using a Matérn covariance with parameters: $\sigma = 1$, $\nu = 1.5$, $\beta = 0.063$. The functions ψ_m are the first M B-spline basis functions, with $M = 10$. Figure S5 in the Supplementary Material shows realizations of the Gaussian random field used for the simulation of the two FRFs.

The first FRF simulated from Eq. (15) will serve as a TG used to simulate new copies of the FRF using the distance measures. Note that the aim of the simulation is not to reproduce the TG, but to create new FRFs that have similar features/characteristics as the TG. The initial conditioning data to assign to the (empty) SG are randomly sampled from the second FRF simulated from Eq. (15). The number of conditioning data assigned to the SG for each simulation is 20 and the number of neighbours used in each simulation is 10.

To visualize the simulated FRFs, we compute the norm of each function at each location in the SG. We then show the random field of the norms of these functions. The first two panels of Fig. 1 show the two FRFs simulated from Eq. (15). The second plot shows the spatial functional data (represented as a random field of the norms of the functions) used as the TG, while the first plot shows the spatial functional data from which the initial conditioning data is sampled at the beginning of each simulation.

4.1.2. Visual comparison of distances

The aim of the simulation study is then to compare the characteristics of the simulated FRFs to the characteristics of the TG. Fig. 1 shows the (random field of norms of the) four simulated FRFs, one RF for each distance, together with the RFs of the conditioning data and the TG. The simulated RFs show an overall good reproduction of the features of the TG, without creating an exact copy. There are few artefacts in the simulated FRFs, especially for that of D_3, with plenty of (norms of) functions seemingly out of place in their currently assigned locations, compared to their surrounding. This is because D_3 compares the data events of the training and simulation grid using the norms of the first derivative of neighbouring functions (i.e., $Z(\boldsymbol{u} + \boldsymbol{h}_\alpha, t)^{(1)}$ and $Z(\boldsymbol{v} + \boldsymbol{h}_\alpha, t)^{(1)}$) during the simulation process, while we are visualizing and comparing the simulated RFs using norms of functions. Since D_3 favours functions whose data event have similar total velocity over functions whose data event have similar total norm (like in D_1), and two functions with similar derivatives may have different norms, it is reasonable that the simulated RF based on D_3 showed more artefacts in Fig. 1. The simulated RFs for D_1 and D_2 show a very good reproduction of the features of the TG with minimal artefacts while the simulated RFs for D_4 also show a good reproduction of the features of the training grid, albeit with more artefacts than those of D_1 and D_2. The comparison shown in Fig. 1, based on norms of functions, is biased in favour of D_1 and D_2 since both are based on comparing norms of neighbouring functions in data events. If we compare the simulated RFs using derivatives of norms of functions, shown in Figure S6 of the Supplementary Material, we see a different perspective on the performance of the four distances. The simulated RFs by D_3 now show fewer artefacts compared to Fig. 1 while the simulated RF by D_1 shows a bit more artefact compared to its representation in Fig. 1. The simulated RF by distance D_4 maintains roughly the same level of artefacts in both Fig. 1 and S6 because D_4 compares shape, amplitude and magnitude of functions in the data events. Overall, distances D_2 and D_4 visually show a balanced performance across different representations of the simulated RFs. An overview of the functions in the FRFs whose norms are shown in Fig. 1 are visualized using the functional boxplot of Sun and Genton (2011) and presented in Fig. 2. They show an overall good match between the simulated and original functional data albeit with a slightly smaller variability (represented by the height of the magenta boundary) for the simulated functions. In Figures S7, S8, and S9 of the Supplementary Material, we show three random fields representing the values of the simulated functions at three different time points across the domain t, one close to the beginning ($t = 0.05$), at the middle ($t = 0.5$), and close to the end ($t = 0.95$).

We repeated the simulation with the number of conditioning data and neighbours set to 50 and 20, respectively. We observed similar results with distances D_2 and D_4 showing good performance out of the four distances (Figure S10 in the Supplementary Material). Section S-II of the Supplementary Material presents a quantitative comparison of the distances using 100 repetitions, where we compared distribution of the means, variances, and variograms of the FRFs of the 100 simulations (for each distance) to those of the training grid. The results show that the 100 simulations from distance D_2 show the most similar metrics to those of the TG.

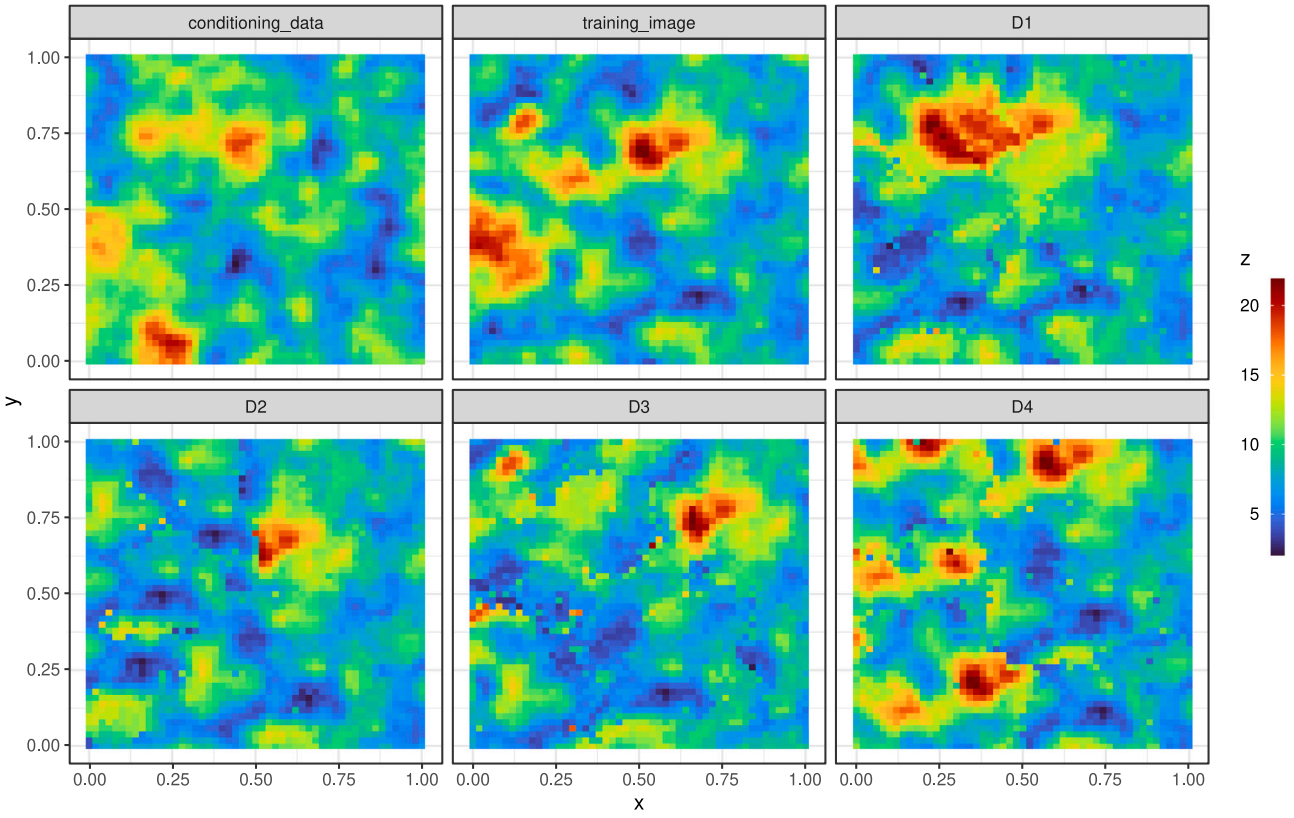

Fig. 1. Simulated functional random fields (shown as random fields of the norm of functions; 51 × 51 locations) used as the training grid (first row, second column) and conditioning data (first row, first column), followed by FDS simulations using distances D_1 to D_4. The number of conditioning data is 20 and the number of neighbours is 10.

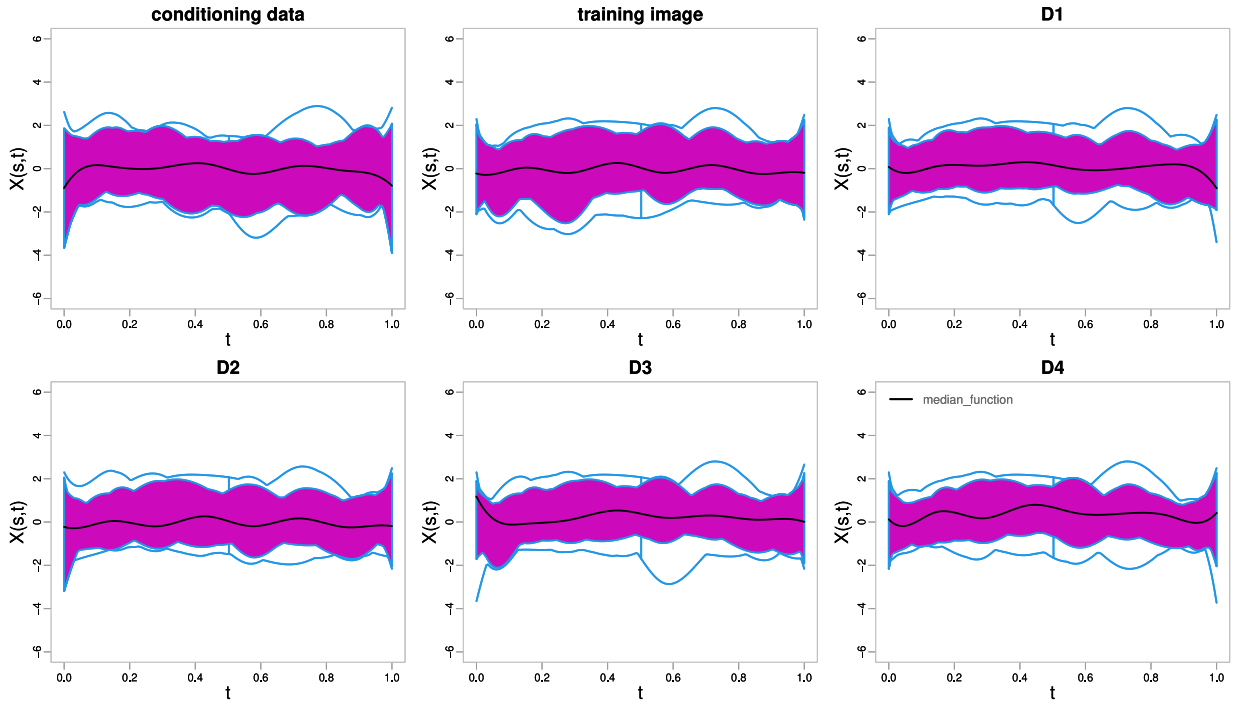

Fig. 2. The functional boxplots of the functions contained in each simulated FRF using distances D_1 to D_4 for FDS simulation. The number of conditioning data is 20 and the number of neighbours is 10.

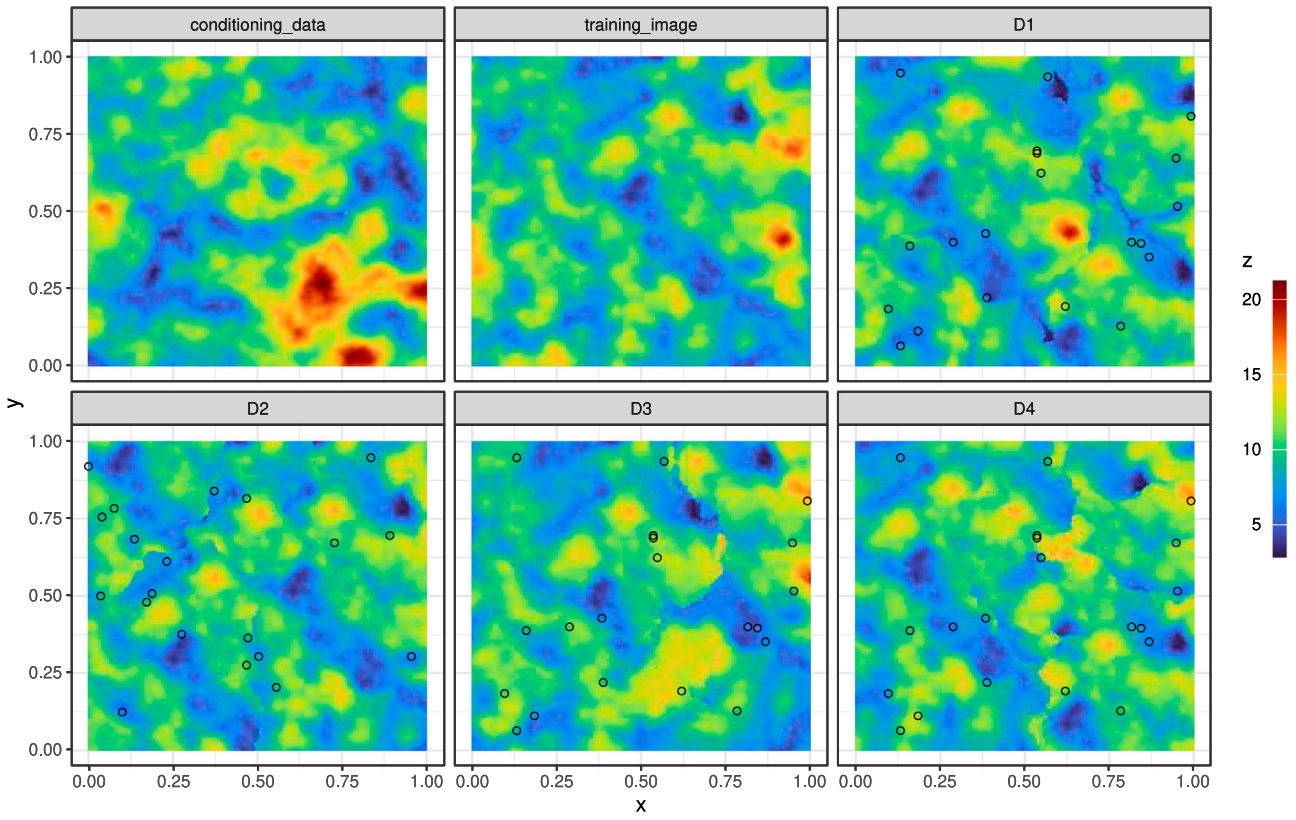

Fig. 3. Simulated functional random fields (shown as random fields of the norm of functions; 251 × 251 locations) used as the training grid (first row, second column) and conditioning data (first row, first column), followed by FDS simulations using distances D_1 to D_4. The number of conditioning data is 20 and the number of neighbours is 10. Conditioning points indicated with black circles.

4.1.3. Larger grids and number of neighbours

We tested the four distances using the same simulation procedure with larger training and simulation grids to assess their performance with larger grids of size 251 × 251.

Fig. 3 shows a single simulation for each distance. Interestingly, all the distances produced good simulations with less perceivable artefacts overall. None of the FDS simulations is an exact verbatim copy of the TG, although all the simulations produced contain verbatim copies of parts of the TG. This is due to the lack of pattern repeatability in the TG, leaving FDS with a limited choice of matching candidate functions during the simulation.

To test the effect of number of neighbours, we repeated the simulation with only distance D_2, while varying the number of neighbours ($n = \{20, 30, 40, 50\}$). Visual inspection of the simulations (see Figure S29 of the Supplementary Material) does not show perceivable significant improvement in simulation quality compared to that of D_2 in Fig. 3 produced with $n = 10$ neighbours.

4.1.4. Sampling using distance threshold and training grid scan fraction

In Section 3.4, we described how the distance threshold γ and fraction of TG to scan f can be incorporated into FDS. We tested the effect of the scan fraction using distance D_2 while varying f between 0.5 and 1. Values of f greater than 0.5 generally produce good results, with little improvements seen as f approaches 1 (see Figure S30 of the Supplementary Material).

Likewise, sensitivity analysis for γ shows that values of $\gamma \leq 0.1$ produce good simulations while $\gamma \leq 0.01$ produce simulations that are similar to using the best matching candidate. Fig. 4 shows a comparison between simulations produced with distance threshold values $\gamma \in \{0.01, 0.1, 0.2\}$ and scan fraction values $f \in \{0.5, 0.8\}$, using distance D_2. The simulations produced by $\gamma = 0.2$ contain a lot of noise and artefacts, compared to those produced by $\gamma = 0.1$ and $\gamma = 0.01$, while simulations produced by $f = 0.5$ and $f = 0.8$ show similar quality.

In general we suggest to always incorporate sampling using the distance threshold γ and TG scan fraction f in FDS, starting with values of $\gamma \leq 0.1$ and $f \geq 0.5$.

4.1.5. Conditioning data weighting

The simulations shown in Figs. 3 and 4 show many conditioning points misaligned with their surrounding pattern. To make FDS respect the conditioning data more, higher weights can be assigned to conditioning points while computing the distance between data events. This ensures that the conditioning points significantly influence the choice of patterns in their surroundings, while also helping reduce the generation of simulations with verbatim copy of parts of the TG. However, higher conditioning weights may introduce artefacts, which can be removed with a suitable post-processing method (see Section 4.3). In Section S-IV and Figure S4 of the Supplementary Material, we present results of simulations generated with higher conditioning weights.

4.2. Filling missing data in a functional random field

A common application of MPS is its use in simulating missing values, e.g., Oriani et al. (2016). We also test our proposal in this scenario. The aim is to fill in missing functions in an FRF using a TG. First we simulate the TG as described in Section 4.1.1. We then simulate another FRF, which we call the test image, and remove some of its functions in the lower left corner. Fig. 5 shows the RF (of the norm of functions) of the TG and the test image with the missing functions.

Since there are existing functions in the FRF of the test image, there is no need to assign any initial conditioning data. With the number of neighbours set to 10, we conduct four repetitions for each

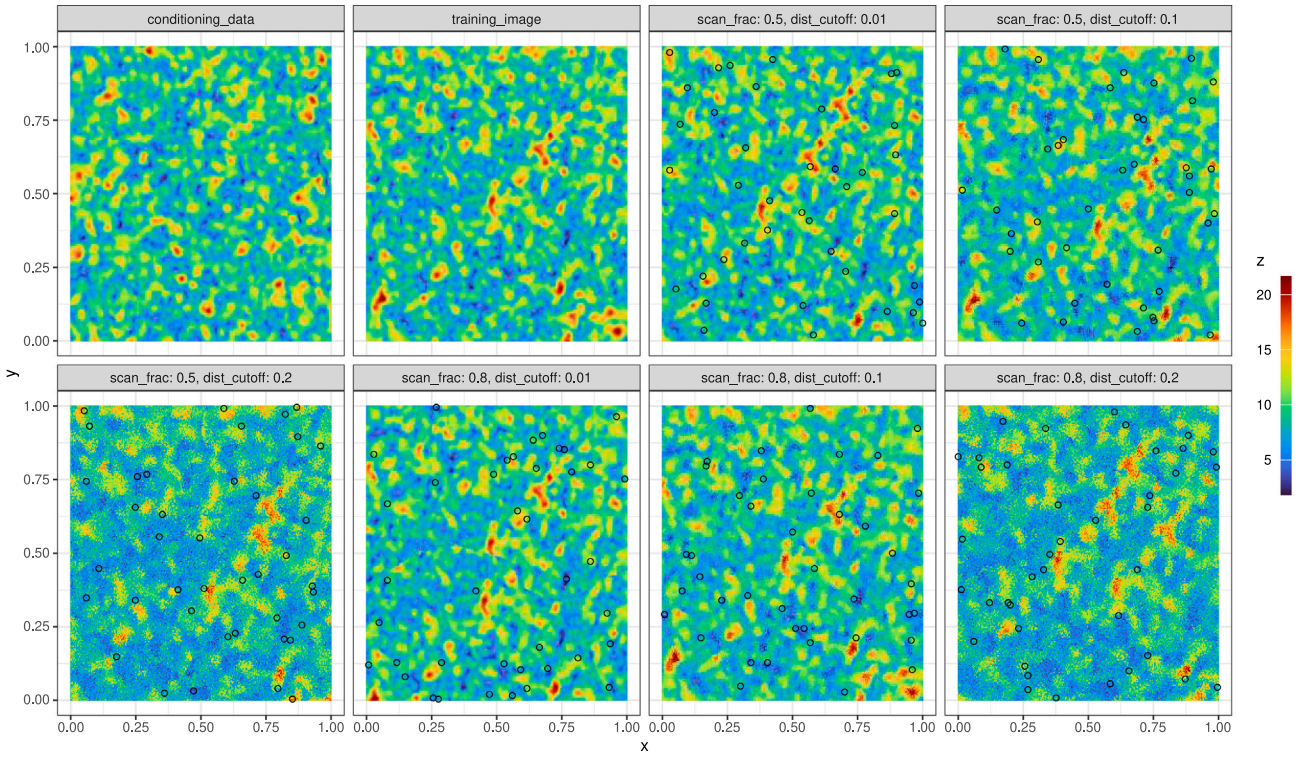

Fig. 4. Simulated functional random fields (shown as random fields of the norm of functions; 251 × 251 locations) used as the training grid (first row, second column) and conditioning data (first row, first column), followed by FDS simulations using distance D_2 with distance threshold $\gamma \in \{0.01, 0.1, 0.2\}$ and TG scan fraction $f \in \{0.5, 0.8\}$. The number of neighbours is 30. Location of conditioning points indicated with black circles.

distance. We show a single copy of these filled FRFs in Fig. 5. From Fig. 5, all the filled copies show some discontinuity, especially in the upper part of the missing area. Simulations from distances D_2 and D_4 showed the best results while those of distances D_1 and D_3 show more artefacts. In addition, the plots with facet labels "D2_Inner_Outward" and "D4_Inner_Outward" of Fig. 5 show simulations from D_2 and D_4 but with a centre-outward filling order compared to using a random order (as done for D_1 to D_4). Results from the centre-outward filling order show more artefacts, suggesting that FDS performs better with a randomly ordered filling process.

As the use of multiple-point simulation for predicting missing functions in an FRF is new in the literature, it is difficult to find a method to directly compare our proposal with. The most suitable method that we found for comparison is the ordinary kriging for functional data (OKFD) (Giraldo et al., 2011), implemented in the *geofd* R package (Giraldo et al., 2012b). However, the two methods are not directly comparable because our proposal requires the use of a dedicated TG while OKFD is based on an empirical variogram. The facet "OKFD" of Fig. 5 shows the norms of the predicted functions using OKFD. Although the norms of the predicted functions by OKFD show more continuity at the edges, they were unable to reproduce and capture the intricate features of the test image. This shows an advantage of using a TG with similar features for spatial functional prediction compared to using OKFD. Although a dedicated TG was used for FDS in this example, the non-missing part of the test image could as well have been used, eliminating the need for a TG. Figures S18 and S19 in the Supplementary Material show the same results as in Fig. 5, but with the number of neighbours set to 5 and 20, respectively. Also, Figure S17 in the Supplementary Material shows a functional boxplot visualizing the spread of the functions whose norms are represented in the RFs in Fig. 5. It is also useful to visualize the values of the simulated functions at some specific points of the domain to better assess their similarity to the neighbouring functions in the SG. Figures S20, S21, and S22 show three random fields representing the values of the simulated functions (and the non-missing part of the SG) at three different time points across the domain t ($t = 0.05$, $t = 0.5$ and $t = 0.95$), respectively.

FDS generates a single simulation while OKFD produces an estimation, which are results of different nature. To compare results of similar nature, we repeated the simulation 100 times and computed the functional mean of each filled location (over the 100 repetitions). Each of these means correspond to an expected function at each location and they are more comparable to the functions estimated by OKFD. Fig. 6 compares these functional means to the result of OKFD. We observe that the norms in the RFs of FDS have more variance compared to that of OKFD (where most of the estimated functions have small norms). Furthermore, the results of FDS are more similar to the non-missing part of the simulation grid, especially when distance D_4 is used.

In Section S-III of the Supplementary Material, we present a quantitative comparison, between the original test image and the filled FRFs (generated by the 100 repetitions of FDS) using mean, variance, and empirical variogram. The results show that FDS using the proposed distances is better at capturing intricate variations in the original test image, as indicated by the proximity of the distributions to the means, variances, and variograms of the FDS simulations to those of the original test image (Figure S3 of the Supplementary Material).

4.3. Post-processing

It is fairly common that a matching data event cannot be found in the TG during simulation. Such cases lead to the selection of a function that does not sufficiently match their neighbours in the simulation grid, leading to artefacts and reduced pattern reproduction. In such cases, post processing methods can be applied to re-simulate inconsistent locations.

Post processing methods can easily be applied on simulations from FDS to improve pattern reproduction and reduce artefact. For instance, the syn processing method (Mariethoz et al., 2010) or real-time post-processing method (Strebelle and Suzuki, 2007) can be adapted for use in FDS. We note that all simulations presented in this study did not use postprocessing.

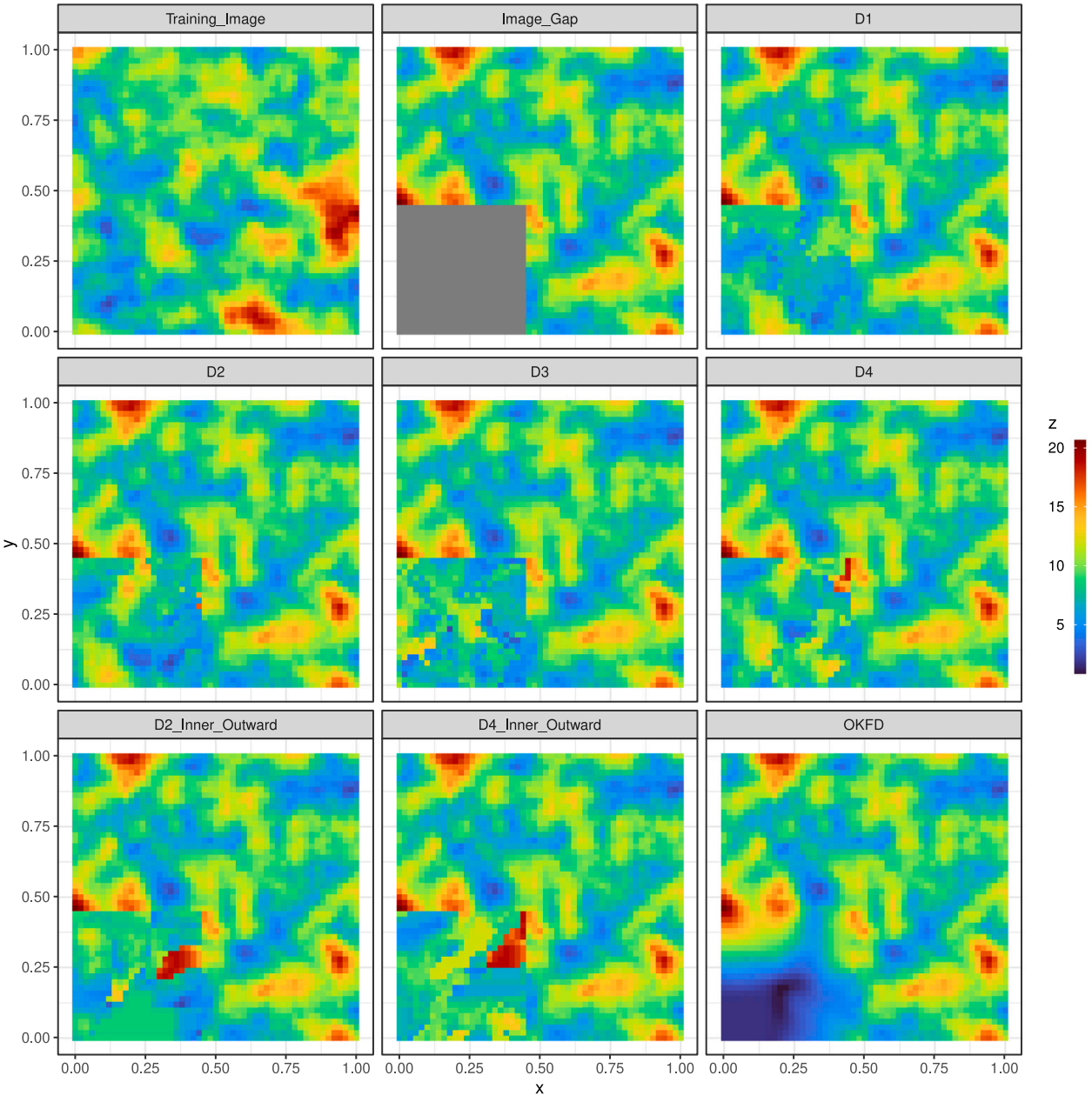

Fig. 5. Simulated functional random fields (shown as random fields of the norm of functions; 51 × 51 locations) of the test image with missing parts (top centre), using the training image (top left) and distances D_1 to D_4 in FDS simulation. Facet OKFD shows spatial prediction with Ordinary Kriging for Functional Data. Distances D_2 and D_4 were used in FDS for D2_Inner_Outward and D4_Inner_Outward (bottom left and centre), but with a centre-outward filling order.

5. Applications to wind profiles data

We apply the proposed method to wind profiles data over the Arabian Peninsula obtained from a year-long simulation of the Weather Research and Forecasting (WRF) model (Skamarock et al., 2008); (see run 4 from Giani et al. (2020)). The grid consists of $549 \times 499 = 273,951$ locations, each with a wind profile of average (yearly) wind speed at 40 vertical layers ranging from 9–11 m above sea level at layer 1 to 16–21 km above sea level at layer 40. By considering the wind speeds at each location as a function of the vertical layers (height), we then obtain an SFD where each location has a wind speed function. Figure S25 shows the norm of these functions observed over the Arabian Peninsula. Moreover, the left plot of Figure S15 shows these wind speed functions from 50 randomly selected locations in the covered grid while the right plot of Figure S16 shows the same for the first 50 locations in the grid. These two plots show that functions from nearby locations are similar in magnitude and shape, compared to functions from locations far apart.

For this application, we focus on the south-western part of Saudi Arabia which has the highest wind speed function norms in the country due to a contiguous mountain range lying along the south-western coast of the Red Sea (top-left panel of Fig. 7). We assume that some of the wind profiles functions along the southwestern coast are missing. These missing locations correspond to a part of the strip in the southwestern corner with lower wind speed norms, surrounded by areas of higher wind speed. This missing region is indicated inside the white boundary shown in the top-left panel of Fig. 7. For training, we use the locations in an area with similar characteristics in the southernmost part of the country, which corresponds to the area inside the black boundary in some panels of Fig. 7. The aim is to use FDS, to simulate wind profile functions for the missing locations, using functions obtained from the

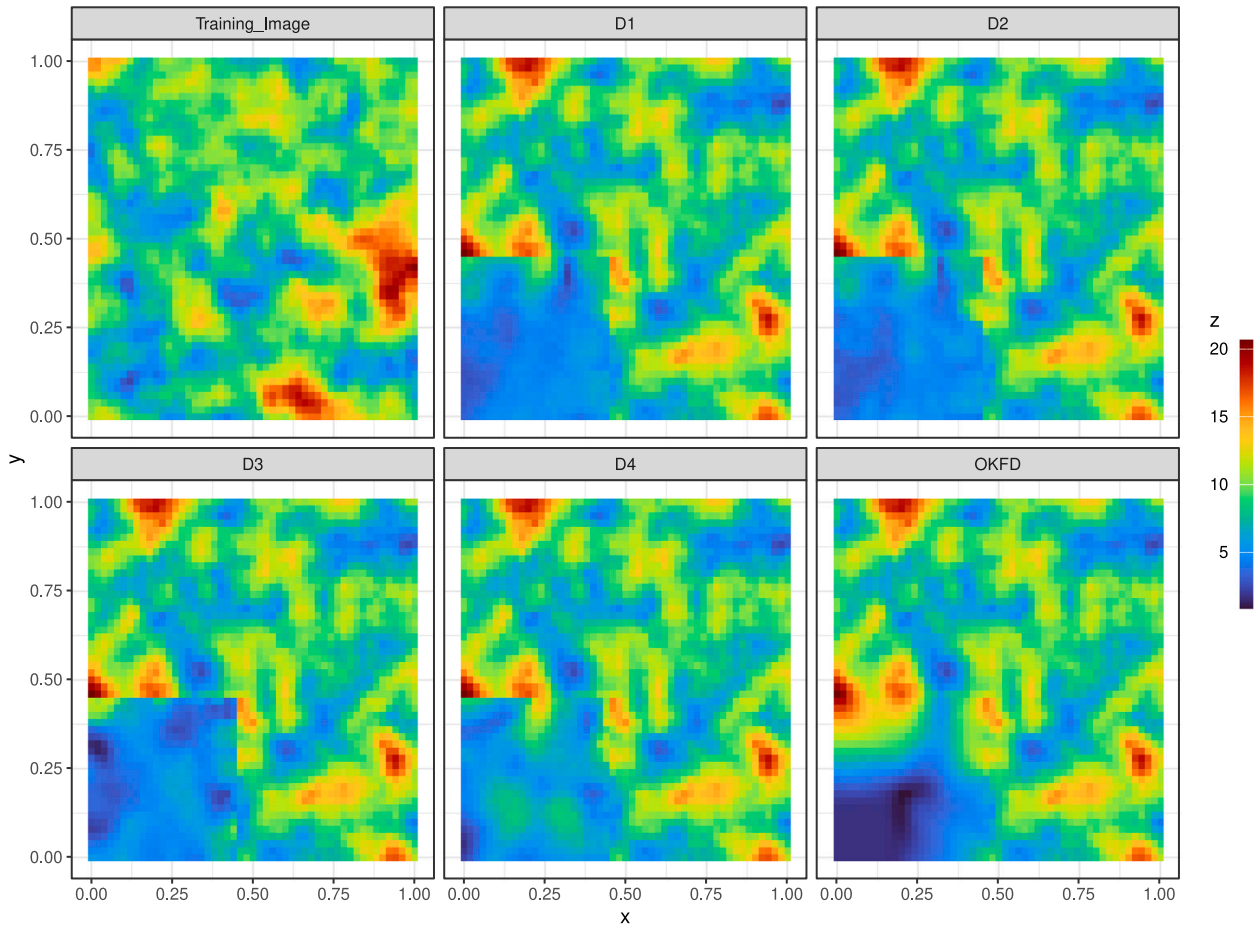

Fig. 6. Random fields of norm of functional means (over 100 FDS simulation repetitions; 51 × 51 locations) for each missing location using the training image (top left) and distances D_1 to D_4. Facet OKFD shows spatial prediction with Ordinary Kriging for Functional Data.

training area; with the number of neighbours set to 20. The remaining existing functions in the study area (locations outside the white and black rectangular boundaries) are used as the initial conditioning data for the FDS reconstruction. In panels D1 to D4 of Fig. 7, we show for each distance, the reconstructed area using FDS. The reconstructions for D_1, D_2, and D_4 were able to reasonably capture the variations inside the missing area while successfully reconstructing the strip of low wind speed norms, which continued almost seamlessly with the neighbouring region. However, the reconstructions do show some discontinuity in the top right corner of the reconstructed area, with very high norm functions placed in an area surrounded by lower norms. This discontinuity in the top right corner is less pronounced in the reconstruction for distance D_2, compared to those of distances D_1 and D_4, and it is due to the presence of a region with very high wind speed profiles to the right of the southernmost tip of the Saudi Arabian boundary in the training area. This region in the training area is bordered to the north-west by an area of relatively lower windspeed. This is especially obvious in the reproductions by distances D_1 and D_2 where the right area of their reproductions contains wind profile functions selected from this region and the area with low wind speed profile bordering it to the north-west. Imposing more number of neighbours does not seem to help with the observed discontinuity as shown in Figures S26 and S27. This observation demonstrates a key fact about MPS in general: the spatial predictions are as good as the TG from which they were taken.

The reconstruction from D_3 is poor with the resulting simulation unable to reconstruct the continuous strip of low wind speed norms in the missing area. This is because D_3 compares data events based on the first derivative of their constituent functions, disregarding other potentially important features, like the magnitude. Figure S28 shows the FDS reconstruction using inner-outward filling order. These reconstructions are worse compared to using a random filling order like in Fig. 7, in line with observations from the simulation experiments.

OKFD showed very good performance on this problem, with the missing area filled with spatial predictions that reasonably reproduce the strip of low wind speed norms while smoothing out the edges of this missing area with functions gradually increasing in norm (top-centre plot of Fig. 7). This is not surprising because the spatial pattern of functions contained in the missing area is simple, compared to those used in our gap-filling simulation tests in Section 4.2. Fig. 8 shows the functional boxplots of the functions represented with norms in the missing area of Fig. 7. The functional boxplot plot shows a clear difference in the overall trend of the functions simulated by FDS and the spatial predictions from OKFD. Distances, D_1, D_2 and D_4 on the average selected functions similar in magnitude and shape to the original functions in the missing area while the functions predicted by OKFD are smoother with less variation. We note again, that OKFD is not directly comparable to our proposal, as it is based on kriging, rather than on the use of a training grid.

Table 1 shows the median time taken by distances D_1 to D_4 to obtain the simulated FRFs shown in Fig. 7. The median was taken over 5 repetitions using the *tictoc* R package (Izrailev, 2023). Distance D_4 is the slowest (requiring a median of 31 min to fill the 946 locations contained in the missing area) as it is based on an average of three indices used in comparing the data events. Distances D_1 to D_3 show similar performance, requiring on average about 4 to 6 min to complete the gap filling process. OKFD took about 111.3 min to complete the process. It is worthy of note that while FDS is restricted to only the

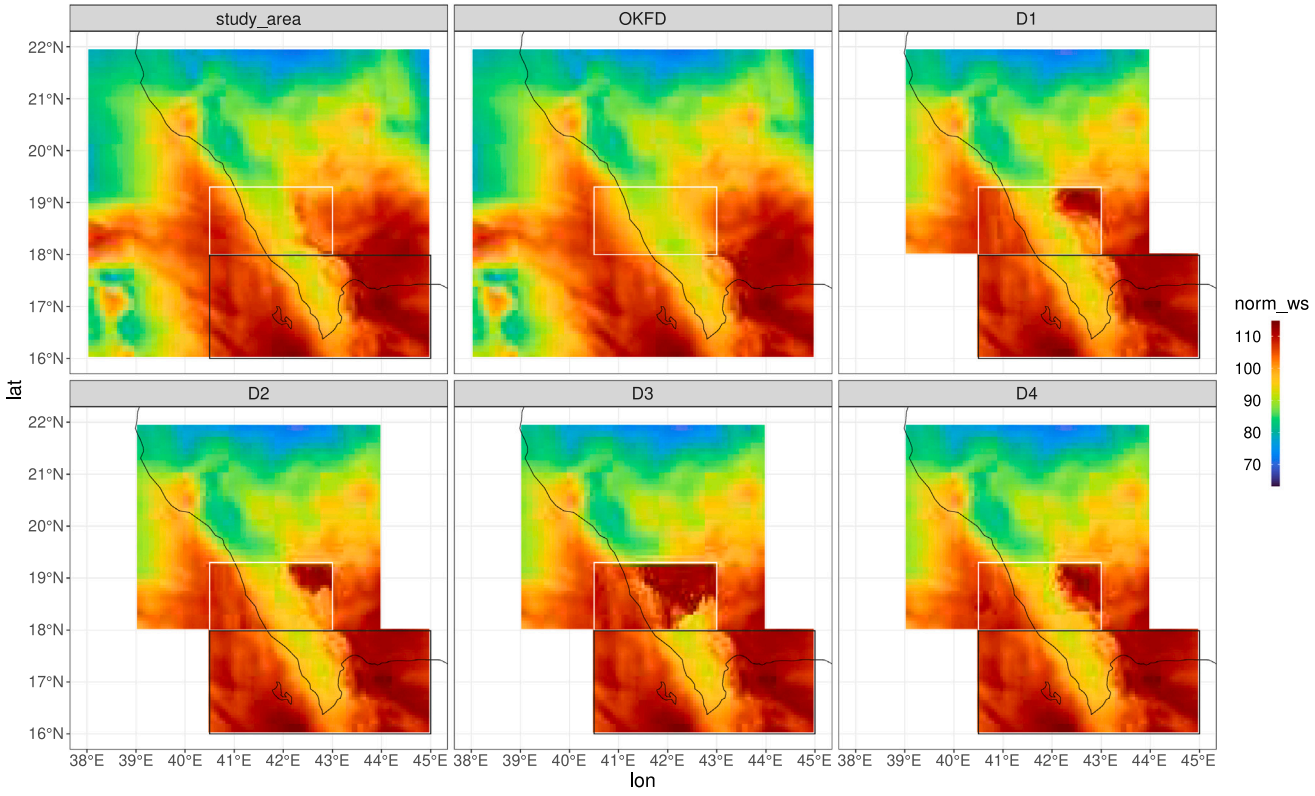

Fig. 7. Top-left: Norms of wind speed functions of study area of the Arabian Peninsula considered; assumed missing area in white rectangle and training area in black rectangle. Top-centre: Norms of wind speed functions with the missing area filled with spatial predictions from OKFD. Panels D1 to D4: Simulation results of the reconstruction of missing locations using the distances D_1 to D_4; a single repetition is shown for each distance. The number of neighbours used is set to 20.

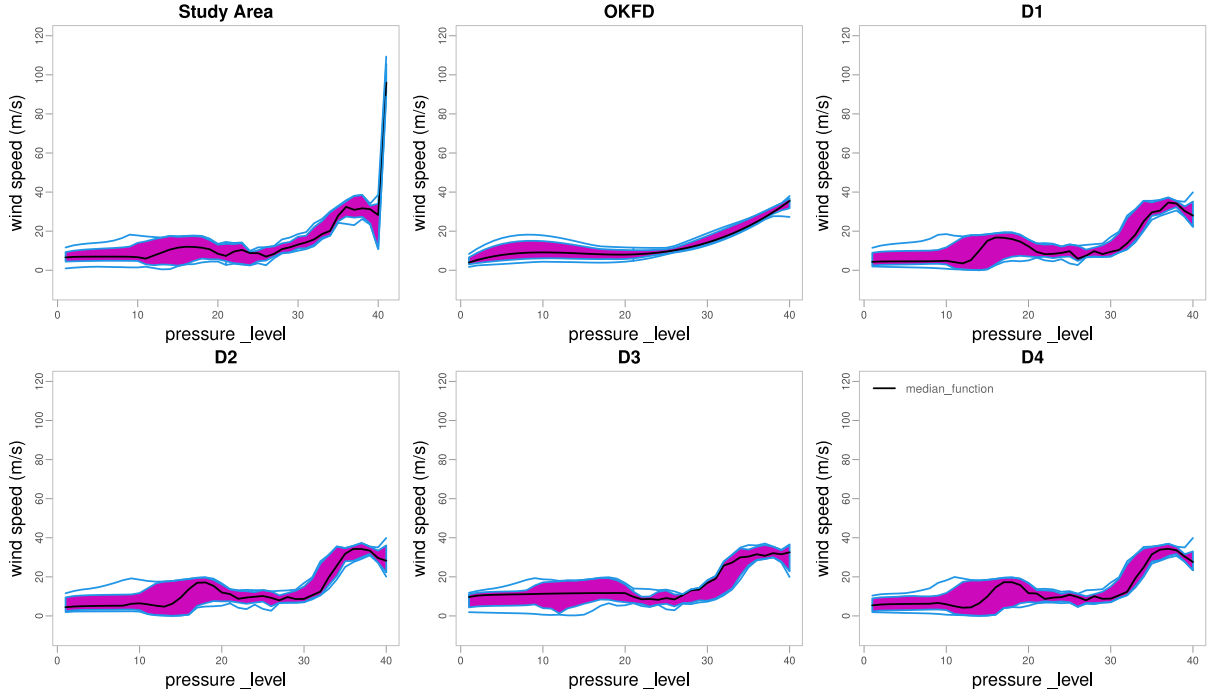

Fig. 8. Functional boxplots of the wind speed functions contained in the filled gap of the study area of the Arabian Peninsula by OKFD and FDS with distances D_1 to D_4.

TG (shown in the black rectangle in the panels of Fig. 7) in the filling process, the whole study area (except the missing area) was parsed to OKFD. Thus, its computational time is not directly comparable to those obtained by FDS. The test was run on a desktop computer with a 12-core Intel Core i7-8700 processor with 32 GB RAM running a Fedora operating system.

Table 1
Computational time in minutes for gap filling of wind profiles data.

Method	Median time (Minutes)
FDS with distance D_1	4.6
FDS with distance D_2	4.7
FDS with distance D_3	5.5
FDS with distance D_4	31.0
Spatial prediction with OKFD	111.3

6. Discussion

Multiple-point simulation has proven itself useful in various fields, and we extended its use to SFD in this work. We leveraged some tools for functional data to propose four different distances, useful for comparing a "functional data event". We then used these distances in the proposed FDS algorithm.

We tested the proposed distances using two simulation scenarios, one simulating an SFD, conditioned on the spatial features of another SFD (the TG). The second scenario attempted to fill gaps of missing data using another grid with similar characteristics as a TG. Distance D_2 showed the best results in both simulation scenarios with distances D_1 and D_4 also showing promising results. Distance D_3 performed poorly on most of the simulation tasks because it compares data events based on the derivatives of their constituent functions. However, D_3 may be useful in specialized cases where functions with similar velocity/acceleration are considered during simulation. We demonstrated the proposed method on simulated wind profiles obtained from WRF in a gap-filling task. Like in the simulation, distance D_2 showed the best result. In general, we recommend using the distance D_2 given its good performance across the different scenarios tested.

The proposed method creates an interesting merge between functional data analysis and multiple-point simulation, and provides a way to simulate functions with a spatial location, accounting for the spatial correlation and similarity of neighbouring functions. An alternative method to our proposal is ordinary kriging for functional data (OKFD) which tends to have worse performance when the spatial pattern is more complex and nuanced, as seen in the simulation results. Space–time covariance modelling could also be an alternative but its performance in the scenarios tested in this work is expected to be worse than that of OKFD because it does not consider the time points as a domain over which functions are defined. Our proposal has a similar advantage over multivariate multi-point simulation, in addition to being able to deal with very high dimensional data. Our proposal is also capable of measuring other types of similarities (e.g., shape and amplitude) between functions in a data event, thereby offering an additional case for its application.

Further areas of research include a detailed study on the use and effect of interesting features of MPS, like weighting, use of invariant distances, etc. It is also interesting to extend the FDS to multivariate functional data, in which case a vector of functions is observed at each location. Various similarity and distance measures already defined in the functional data analysis literature (Dai and Genton, 2018; Ojo et al., 2023) are good starting points towards such an extension.

CRediT authorship contribution statement

Oluwasegun Taiwo Ojo: Writing – original draft, Visualization, Software, Methodology, Conceptualization. **Marc G. Genton:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization.

Code availability section

Name of the code: fmps
Contact: oojo@inst.uc3m.es
Hardware requirements: Windows, Mac or Linux
Program language: R
Software required: R
The source codes are available for downloading at the link: https://github.com/otsegun/fmps

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Marc G. Genton and Oluwasegun Ojo's research was supported by the King Abdullah University of Science and Technology (KAUST), Saudi Arabia. This work has also been partially funded by the project AUDINT (Grant TED2021-132076B-I00) funded by MCIN/AEI/10.13039/501100011033 and the EU NextGeneration/PRTR funds. We thank Prof. Paola Crippa (pcrippa@nd.edu) for providing the wind profile data. She can be contacted for any information on the dataset or for data sharing.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cageo.2024.105767.

Data availability

Prof. Paola Crippa (pcrippa@nd.edu) can be contacted for any information on the wind profile dataset or for data sharing.

References

Abramowicz, K., Arnqvist, P., Secchi, P., De Luna, S.S., Vantini, S., Vitelli, V., 2017. Clustering misaligned dependent curves applied to varved lake sediment for climate reconstruction. Stoch. Environ. Res. Risk Assess. 31 (1), 71–85.
Aguilera-Morillo, M.C., Durbán, M., Aguilera, A.M., 2017. Prediction of functional data with spatial dependence: a penalized approach. Stoch. Environ. Res. Risk Assess. 31 (1), 7–22.
Arnone, E., Azzimonti, L., Nobile, F., Sangalli, L.M., 2019. Modeling spatially dependent functional data via regression with differential regularization. J. Multivariate Anal. 170, 275–295.
Caballero, W., Giraldo, R., Mateu, J., 2013. A universal kriging approach for spatial functional data. Stoch. Environ. Res. Risk Assess. 27 (7), 1553–1563.
Dai, W., Genton, M.G., 2018. Multivariate functional data visualization and outlier detection. J. Comput. Graph. Statist. 27 (4), 923–934. http://dx.doi.org/10.1080/10618600.2018.1473781, arXiv:https://doi.org/10.1080/10618600.2018.1473781.
Ferraty, F., Vieu, P., 2006. Nonparametric Functional Data Analysis: Theory and Practice. Springer, New York.
Giani, P., Tagle, F., Genton, M.G., Castruccio, S., Crippa, P., 2020. Closing the gap between wind energy targets and implementation for emerging countries. Appl. Energy 269, 115085.
Giraldo, R., Caballero, W., Camacho-Tamayo, J., 2018. Mantel test for spatial functional data. AStA Adv. Stat. Anal. 102 (1), 21–39.
Giraldo, R., Delicado, P., Mateu, J., 2011. Ordinary kriging for function-valued spatial data. Environ. Ecol. Stat. 18 (3), 411–426.
Giraldo, R., Delicado, P., Mateu, J., 2012a. Hierarchical clustering of spatially correlated functional data. Stat. Neerl. 66 (4), 403–421.
Giraldo, R., Mateu, J., Delicado, P., 2012b. Geofd: an R package for function-valued geostatistical prediction. Rev. Colombiana Estadíst. 35 (3), 385–407.
Gravey, M., Mariethoz, G., 2020. QuickSampling v1. 0: a robust and simplified pixel-based multiple-point simulation approach. Geosci. Model Dev. 13 (6), 2611–2630.

Gromenko, O., Kokoszka, P., 2013. Nonparametric inference in small data sets of spatially indexed curves with application to ionospheric trend determination. Comput. Statist. Data Anal. 59, 82–94.

Gromenko, O., Kokoszka, P., Zhu, L., Sojka, J., 2012. Estimation and testing for spatially indexed curves with application to ionospheric and magnetic field trends. Ann. Appl. Stat. 6 (2), 669–696.

Guardiano, F.B., Srivastava, R.M., 1993. Multivariate geostatistics: beyond bivariate moments. In: Geostatistics Troia'92. Springer, Dordrecht, pp. 133–144.

Hörmann, S., Kokoszka, P., Kuenzer, T., 2022. Testing normality of spatially indexed functional data. Canad. J. Statist. 50 (1), 304–326.

Ignaccolo, R., Mateu, J., Giraldo, R., 2014. Kriging with external drift for functional data for air quality monitoring. Stoch. Environ. Res. Risk Assess. 28 (5), 1171–1186.

Izrailev, S., 2023. Tictoc: Functions for timing r scripts, as well as implementations of "stack" and "StackList" structures. URL: https://CRAN.R-project.org/package=tictoc R package version 1.2.

Kokoszka, P., Reimherr, M., 2019. Some recent developments in inference for geostatistical functional data. Rev. Colombiana Estadíst. 42 (1), 101–122.

Kuenzer, T., Hörmann, S., Kokoszka, P., 2021. Principal component analysis of spatially indexed functions. J. Amer. Statist. Assoc. 116 (535), 1444–1456.

Li, Y., Qiu, Y., Xu, Y., 2022. From multivariate to functional data analysis: Fundamentals, recent developments, and emerging areas. J. Multivariate Anal. 188, 104806.

Liang, D., Huang, H., Guan, Y., Yao, F., 2022. Test of weak separability for spatially stationary functional field. J. Amer. Statist. Assoc. 1–14.

Liu, C., Ray, S., Hooker, G., 2017. Functional principal component analysis of spatially correlated data. Stat. Comput. 27 (6), 1639–1654.

Mariethoz, G., Caers, J., 2014. Multiple-Point Geostatistics: Stochastic Modeling with Training Images. Wiley-Blackwell, New Jersey.

Mariethoz, G., Renard, P., Straubhaar, J., 2010. The direct sampling method to perform multiple-point geostatistical simulations. Water Resour. Res. 46 (11), 1–14.

Martínez-Hernández, I., Genton, M.G., 2020. Recent developments in complex and spatially correlated functional data. Braz. J. Probab. Stat. 34 (2), 204–229.

Mateu, J., Giraldo, R., 2021. Geostatistical Functional Data Analysis. John Wiley & Sons, New York.

Meerschman, E., Pirot, G., Mariethoz, G., Straubhaar, J., Van Meirvenne, M., Renard, P., 2013. A practical guide to performing multiple-point statistical simulations with the direct sampling algorithm. Comput. Geosci. 52, 307–324. http://dx.doi.org/10.1016/j.cageo.2012.09.019, URL: https://www.sciencedirect.com/science/article/pii/S0098300412003299.

Menafoglio, A., Grujic, O., Caers, J., 2016. Universal kriging of functional data: Trace-variography vs cross-variography? Application to gas forecasting in unconventional shales. Spatial Stat. 15, 39–55.

Menafoglio, A., Secchi, P., 2017. Statistical analysis of complex and spatially dependent data: a review of object oriented spatial statistics. European J. Oper. Res. 258 (2), 401–410.

Menafoglio, A., Secchi, P., Dalla Rosa, M., 2013. A universal kriging predictor for spatially dependent functional data of a Hilbert space. Electron. J. Stat. 7, 2209–2240.

Nerini, D., Monestiez, P., Manté, C., 2010. Cokriging for spatial functional data. J. Multivariate Anal. 101 (2), 409–418.

Ojo, O.T., Anta, A.F., Lillo, R.E., Sguera, C., 2021. Detecting and classifying outliers in big functional data. Adv. Data Anal. Classif. 16 (3), 725–760. http://dx.doi.org/10.1007/s11634-021-00460-9.

Ojo, O.T., Fernández Anta, A., Genton, M.G., Lillo, R.E., 2023. Multivariate functional outlier detection using the fast massive unsupervised outlier detection indices. Stat. 12 (1), e567. http://dx.doi.org/10.1002/sta4.567, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/sta4.567, URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/sta4.567.

Oriani, F., Borghi, A., Straubhaar, J., Mariethoz, G., Renard, P., 2016. Missing data simulation inside flow rate time-series using multiple-point statistics. Environ. Model. Softw. 86, 264–276.

Pham, T.D., 2012. Supervised restoration of degraded medical images using multiple-point geostatistics. Comput. Methods Programs Biomed. 106 (3), 201–209.

Rachdi, M., Laksaci, A., Al-Awadhi, F.A., 2021. Parametric and nonparametric conditional quantile regression modeling for dependent spatial functional data. Spatial Stat. 43, 100498.

Římalová, V., Fišerová, E., Menafoglio, A., Pini, A., 2022. Inference for spatial regression models with functional response using a permutational approach. J. Multivariate Anal. 189, 104893. http://dx.doi.org/10.1016/j.jmva.2021.104893.

Romano, E., Mateu, J., Giraldo, R., 2015. On the performance of two clustering methods for spatial functional data. AStA Adv. Stat. Anal. 99 (4), 467–492.

Romano, E., Verde, R., 2012. Clustering geostatistical functional data. In: Di Ciaccio, A., Coli, M., Angulo Ibanez, J.M. (Eds.), Advanced Statistical Methods for the Analysis of Large Data-Sets. Springer, Berlin, Heidelberg, pp. 23–31.

Skamarock, W.C., Klemp, J.B., Dudhia, J., Gill, D.O., Barker, D.M., Wang, W., Powers, J.G., 2008. A description of the advanced research WRF version 3. In: University Corporation for Atmospheric Research (No. NCAR/TN-475+STR). http://dx.doi.org/10.5065/D68S4MVH.

Strebelle, S., 2002. Conditional simulation of complex geological structures using multiple-point statistics. Math. Geol. 34 (1), 1–21.

Strebelle, S., Suzuki, S., 2007. Real-time post-processing method to enhance multiple-point statistics simulation. In: EAGE Conference on Petroleum Geostatistics. European Association of Geoscientists & Engineers, pp. cp–32.

Sun, Y., Genton, M.G., 2011. Functional boxplots. J. Comput. Graph. Statist. 20 (2), 316–334. http://dx.doi.org/10.1198/jcgs.2011.09224, arXiv:https://doi.org/10.1198/jcgs.2011.09224.

Vandewalle, V., Preda, C., Dabo-Niang, S., 2022. Clustering spatial functional data. In: Geostatistical Functional Data Analysis. John Wiley & Sons, Ltd, https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119387916.ch7, pp. 155–174. http://dx.doi.org/10.1002/9781119387916.ch7, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119387916.ch7.

Vannametee, E., Babel, L., Hendriks, M., Schuur, J., De Jong, S., Bierkens, M., Karssenberg, D., 2014. Semi-automated mapping of landforms using multiple point geostatistics. Geomorphol. 221, 298–319.

White, P.A., Frye, H., Christensen, M.F., Gelfand, A.E., Silander Jr., J.A., 2021. Spatial functional data modeling of plant reflectances. arXiv preprint arXiv:2102.03249.

Wu, J., Zhang, T., Journel, A., 2008. Fast FILTERSIM simulation with score-based distance. Math. Geosci. 40 (7), 773–788.

Yin, G., Mariethoz, G., McCabe, M.F., 2017a. Gap-filling of landsat 7 imagery using the direct sampling method. Remote Sens. 9 (1), 12.

Yin, G., Mariethoz, G., Sun, Y., McCabe, M.F., 2017b. A comparison of gap-filling approaches for landsat-7 satellite data. Int. J. Remote Sens. 38 (23), 6653–6679.

Zhang, H., Li, Y., 2022. Unified principal component analysis for sparse and dense functional data under spatial dependency. J. Bus. Econom. Statist. 40 (4), 1523–1537.

Zhang, T., Switzer, P., Journel, A., 2006. Filter-based classification of training image patterns for spatial simulation. Math. Geol. 38 (1), 63–80.