POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Nonparametric estimation of spatial covariance functions for spatial prediction of multi-temporal DInSAR data

Tesi di Laurea Magistrale in
Mathematical Engineering - Ingegneria Matematica

Roberta Troilo, 10659317

Advisor:
Prof.
Alessandra Menafoglio

Co-advisors:
Prof. Simone Vantini
Teresa Bortolotti

Academic year:
2022-2023

Abstract: Sentinel-1 satellites furnish vast Synthetic Aperture Radar (SAR) data globally, revisiting points of interest every six days. Exploiting this data, recent advancements in differential interferometric processing produce high-resolution ground displacement images, precise to centimeter or millimeter accuracy. These observations of evolving ground conditions enable thorough monitoring of large regions prone to environmental hazards. Nevertheless, challenges emerge when spatial elements (e.g., water or vegetated zones) exhibit inconsistent behavior across successive timeframes, resulting in missing data in individual pixels or entire regions, persistently absent over time. While statistical data reconstruction techniques are feasible, they typically rely on the unknown covariance operator of the functional data in the target area, often characterized by pronounced non-stationarities. In this investigation, we tackle the challenge of estimating the spatial covariance operator from evolving DInSAR images by introducing a novel non-parametric methodology rooted in the principles of functional data analysis. While the non-parametric approach offers flexibility in addressing field non-stationarities, a Laplacian regularization ensures smoothness in the reconstructed non-uniform spatial covariance operator. The methodology is demonstrated using multi-temporal DInSAR data collected to monitor the Phlegraean Fields, Italy, a region susceptible to seismic and bradisismic events.

Key-words: covariance kernel reconstruction, low-rank minimization, Laplacian regularization, functional completion, Differential Interferograms SAR

1. Introduction

As the amount of available data increases, so does the related complexity. This is why new statistical methods need to be developed in order to face the new challenges that come from this.
In particular, it often happens that data are incomplete, and there are several reasons that may cause this: sensor failures, data collection issues or simply the nature of the data. In the context of Functional Data Analysis (FDA) (Ramsay and Silvermann [2005] [15]), which is a statistical approach that focuses on analyzing data where the observations are functions rather than traditional scalar values, the issue of reconstructing missing data is crucial. Indeed, in FDA, data are treated as curves, surfaces, or more generally, as functions evolving over a continuum (e.g., time, space). This methodology is particularly useful in fields such as biology, finance, and signal processing, where understanding the entire function, rather than specific data points, is essential.

Each functional datum X_i, with $i = 1, ..., n$, represents one observation - among n realizations - of a random function $X(t) : D \to \mathbb{R}$, being D its domain of definition. When some of the realizations are only observed on a subset O_i of the domain, the available data are incomplete and most statistical methods designed for analysing functional data cannot be applied. Moreover, incomplete curves prevent from visualizing clear patterns and trends, so they can lead to inaccurate decision-making or introduce biases into statistical analyses. This is the reason why it is of particular interest to reconstruct partially observed functional data, by providing an accurate and reliable reconstruction of each functional datum over the part of the domain where it is unobserved.
In this framework, the literature offers many works that treat this issue in a specific way, based on the manner in which data are missing.
For example, in Gromenko, Kokoszka, Sojkain [2017] [6] a functional regression approach is developed to estimate the temporal cooling trend in the ionosphere based on partially observed temporal curves of the covariates.
In Stefanucci, Sangalli, Brutti [2018] [16], instead, several functional completion approaches based on principal component analysis are compared for classification purposes in the study of the pathology of cerebral aneurysms.
Instead, in the work of Kraus [2015] [10], a non-parametric method for functional completion is proposed by estimating mean and covariance operators, assuming that data are missing at random and that each point of the domain presents realizations in a sufficient number of curves, in order to provide a trustworthy reconstruction of at missing locations. In the same work, another method of reconstruction is also proposed, based on principal component analysis. Kneip and Liebl [2020] [9] argued against Kraus's method, believing the assumption of an Hilbert-Schmidt operator for the covariance operator to be too restrictive, so they proposed to use a wider class of operators, called reconstruction operators. Kraus and Stefanucci [2020] [11] demonstrate that the Hilbert-Schmidt assumption was actually not needed to achieve asymptotic optimality, so they showed it was possible to remove the restrictive assumption.
These methods are solid and efficient, as long as the estimated mean and covariance operators are consistent. This condition is not satisfied anymore when, for instance, data are not observed on the whole domain of definition of the random function $X(t)$. Indeed, in this case, there is not enough information to provide complete mean and covariance estimators, so a robust method for their estimate in the missing locations is needed.
Seeking to address this issue, Descary and Panaretos [2018] [5] propose a non-parametric reconstruction of the covariance function, considering the domain $D = [0, 1]$ and assuming that each functional sample X_i is only observed on a generic subinterval of the domain $O_i \subset [0, 1]$, of the same length δ, i.e. $\delta = |O_i| \ \forall i = 1, ..., n$, with $0 < \delta < 1$. Differently from the work of Kraus [2015], where O_i is a union of subintervals, in this case each functional datum is a continuous curve over O_i. This regime is defined as *banded* in Descary and Panaretos [2018], since the covariance function $r(s, t)$ - defined on the unit square $[0, 1]^2$ - is incomplete outside the band $\mathscr{B}_\delta = \{(s, t) \in [0, 1]^2 : |s - t| \leq \delta\}$ (see Figure 2 on the left), being each curve observed over an interval of maximum length δ, and is differenciated from the *blanket* regime in Kraus [2015], where the covariance function is complete, since it is assumed that, for each couple of locations in the domain, there is a sufficient number of curves observed in both locations. The covariance reconstruction method proposed by Descary and Panaretos [2018] consists in a low-rank matrix completion problem, which achieves good perfomance especially in the banded regime.
Aiming at extending even further the applicability of the method of Kraus [2015] for functional completion, this work confronts the case in which data do not miss at random, as Kraus [2015] assumes, and, in particular, each data sample is only registered on the same subset $O \subset D$ of the domain, differently from the banded regime of Descary and Panaretos [2018]. Indeed, in our problem setting $O_i = O$, $\forall i = 1, ..., n$, where O is not assumed to be connected, which implies the presence of completely unobserved rows and columns in the estimated covariance kernel (see Figure 2 on the right). For this reason, we decide to define our regime as *fragmented*.
The method of covariance reconstruction developed in this thesis is based on the combination of the low-rank matrix completion algorithm proposed by Descary and Panaretos [2018] and a Laplacian regularization, which encourages smoothness in the reconstruction by penalizing deviations from a smooth distribution across the elements. The main reason why this regularization is included is that it helps preserving the local structure of the covariance function, that is particularly beneficial when dealing with data exhibiting local dependencies, which is often the case with spatial data. A Laplacian regularized matrix completion problem is faced in the work of Tang et al. [2019] [17] where, to discover microRNA-disease associations, a dual Laplacian regularization is applied, one to smooth based on microRNA functional similarity (along rows) and one to smooth based on disease semantic similarity (along columns). In our case, the Laplacian regularization is not dual since the covariance is symmetric. Furthermore, the innovation brought by our method is the introduction of a weight in the Laplacian term, such that different weight is given to the Laplacian evaluated in the observed and missing parts of the covariance function. The sum of the two weights is one and their value is chosen through hyperparameter tuning. The selection criteria for the hyperparameters are widely discussed in the simulation study in Section 4 and in the case study in Section 5.
The methodological proposal is strongly motivated by the will of reconstructing Differential Interferometric

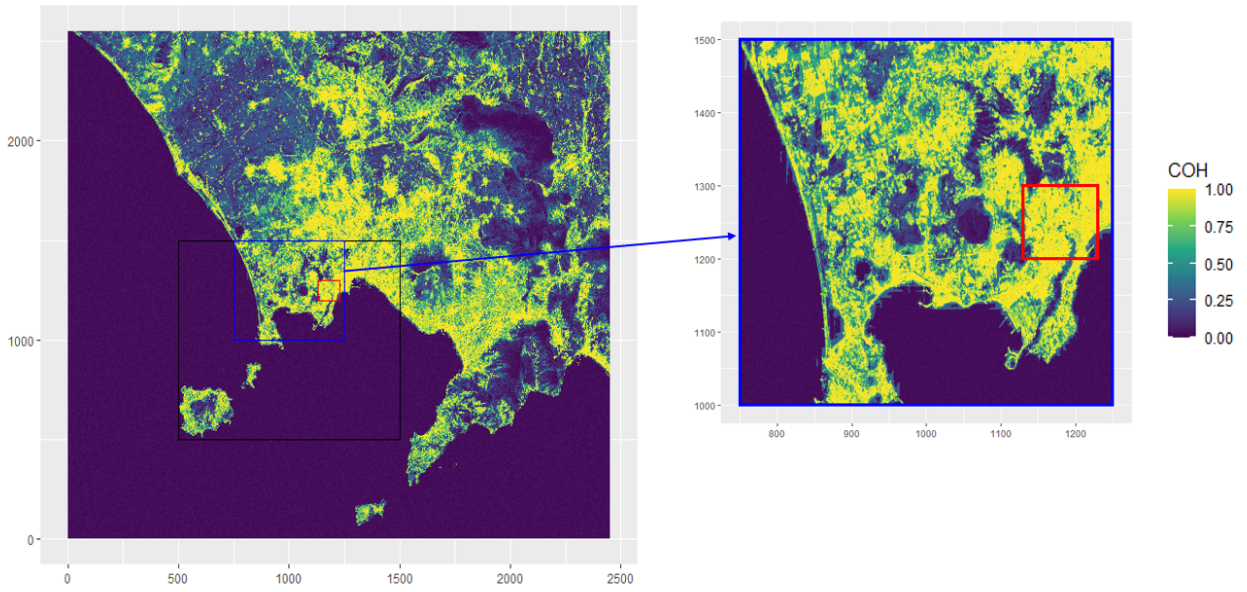

Figure 1: Coherence value per pixel in Phlegraean Fields, Italy

Synthetic Aperture Radar (DInSAR) images, reporting the temporal series of ground surface deformation. The presence of missing values in these images is justified by the scattering, absorption or reflection away from the sensor of radar signals, happening when specific spatial entities are radiated, such as water, vegetation, or rocks. In order to address this issue, the Small BAseline Subsets (SBAS) technique (Berardino et al. [2002] [1], Casu, Manzo, Lanari [2006] [3]) comes into effect, an Advanced DInSAR technique that preprocesses the temporal series of ground displacement with the purpose of removing noise. The main issue related to the SBAS-preprocessed SAR data is that a value of coherence is associated to each pixel (see Figure 1), denoting how reliable is the information inside the pixel. At locations where the coherence is below a certain threshold, the single datum in the pixel is considered as not trustworthy and discarded. This is the reason why our method intents of reconstructing pixels with low coherence, by exploiting the information carried by pixels with higher coherence.

In Figure 1, it is evident that coherence can attain exceptionally low values within very extensive regions, generating incomplete datasets and invalidating many statistical techniques of spatial analysis. The reconstruction of regions associated with low coherence helps preserve the spatial and temporal context of the data. This is particularly important in remote sensing applications like DInSAR, where spatial and temporal continuity is crucial for understanding dynamic processes, such as land surface deformation, vegetation growth or urban development. In this sense, the work of Bernardi et al. [2021] [2] provides a well-detailed summary of the literature concerning applications of satellite data in the study of natural hazards.

In this framework, our study endeavors to expand upon existing methodologies of reconstruction of functional data, and to integrate into the range of techniques that, over the last decades, have been developed to aid remote sensing and mitigate natural disasters.

The rest of this work is structured as follows. Section 2 formally outlines the problem, by summarizing results already documented in literature. Section 3 presents the innovation brought by our method and the complete procedure applied to solve the problem. Section 4 presents a simulation study with the aim of testing the performance of the method, as much as its limits. Section 5 presents the application on real data, considering the SBAS-preprocessed SAR images reconstruction problem previously mentioned. Section 6 ends the analysis drawing conclusions and giving insights for future research directions. Additional details about the data preprocessing and analysis can be found in Appendix A, while Appendix B contains the tables of results for hyperparameters tuning conducted in the simuation study.

2. Preliminaries

2.1. Problem statement and notation

The main aim of traditional functional data analysis is to infer characteristics of the law of a random function $X(t) : [0,1] \to \mathbb{R}$, being its realizations $X_1, X_2, ..., X_n$ completely observed on the domain $[0,1]$. All the observations X_i, with $i = 1, ..., n$, are considered as the realization of independent and identically distributed random variables. These variables are defined in the separable Hilbert space of square integrable functions on a bounded domain. Without loss of generality, this space is set as $L^2([0,1])$, with inner product $\langle f, g \rangle = \int_0^1 f(t)g(t)dt$, $f, g \in L^2([0,1])$ and $\|f\| = \langle f, g \rangle^{\frac{1}{2}}$.
In the case of partially observed functional data, instead, the realizations are not observed in the whole domain, but in a subinterval of it, such that $t \in O_i \subseteq [0,1]$, for all $X_i(t) \in \mathbb{R}$. As a consequence, $X_i(t)$ can be written as $X_i(t) = X_{iO_i}(t)\mathbf{1}_{O_i} + X_{iM_i}(t)\mathbf{1}_{M_i}$.
Differently from previous works, in this thesis we consider the case with a single subinterval of observation common to all the data points, such that $O_i = O, \forall i = 1, ..., n$. Moreover, we do not assume O to be a continuous interval, so that it can be considered as the union of several continuous intervals.
The elements that mainly characterize the distribution generating functional data are the mean function and the covariance operator. Being $X(t) \in L^2([0,1])$, and being X_i iid for all $i = 1, ..., n$, the mean function is $\mu : [0,1] \to \mathbb{R}$, where $\mu(t) = \mathbb{E}[X_1(t)]$ a.e. for $t \in [0,1]$ and the associated covariance operator is $\mathscr{R} : L^2([0,1]) \to L^2([0,1])$ defined as:

$$\mathscr{R}f = \mathbb{E}[\langle f, X_1 - \mu \rangle (X_1 - \mu)] = \int_0^1 r(\cdot, t)f(t)dt \tag{1}$$

for any $f \in L^2([0,1])$, with $r(s,t)$ being the covariance kernel of the random function X_1, such that $r(s,t) = Cov(X_1(s), X_1(t))$ a.e. for $s, t \in [0,1]$, and for any $g, f \in L^2([0,1])$.
The covariance operator is self-adjoint, i.e. $\langle g, \mathscr{R}f \rangle = \langle \mathscr{R}g, f \rangle$ for any $g, f \in L^2([0,1])$, which implies that there exists an orthonormal basis $\{v_k\}_{k=1}^{\infty}$ such that $\mathscr{R}v_k = \lambda_k v_k$, where $\{\lambda_k\}_{k=1}^{\infty}$ is the sequence of eigenvalues for $\mathscr{R}$. Additionally, the covariance operator is trace-class, so its trace is finite, i.e. $\mathrm{Tr}(\mathscr{R}) = \sum_{k=1}^{\infty} \langle \mathscr{R}e_k, e_k \rangle < +\infty$, where $\{e_k\}_{k=1}^{\infty}$ is any orthonormal basis, and it is compact, since it is a linear operator that maps bounded subsets of $L^2([0,1])$ into relatively compact subsets of $L^2([0,1])$. In Horváth and Kokoszka [2012][7], more insights on the Hilbert space approach to functional data can be found. We also refer to A. B. Kashlak et al. [2019] [8] and Pigoli et al. [2014] [14], for inference analysis on covariance operators.
In order to obtain an estimate of the covariance operator, it is necessary to find an estimator for its covariance kernel. For our purposes, the mean function and the covariance kernel need to be estimated from functional data.

2.1.1 Mean estimator

Being $X_1, ..., X_n$ the available functional data, the intuitive estimator for the mean is

$$\mu_n(t) = \frac{1}{n} \sum_{i=1}^{n} X_i(t) \tag{2}$$

for all $t \in [0,1]$. This formulation is well-suited for Kraus [2015] and Descary and Panaretos [2018], since one assumption is that each point in the domain is observed in a sufficient number of incomplete functional curves. This is not the case for the setting considered in our work, since all $X_i(t)$ are unobserved for $t \in [0,1] \backslash O$. As a consequence, the mean estimation as described in Equation (2) is well-suited for any $t \in O$, but remains undefined for $t \in [0.1] \backslash O$, so the mean estimator is incomplete and needs to be reconstructed in its missing part to accomplish the functional completion. In the specific case of our case study, which manages georeferenced data, we consider a smoothing technique to estimate the mean for the unosbserved pixels of the area of interest. In particular, we apply a Nadaraya–Watson kernel regression, that is a type of kernel regression that considers a locally weighted average of the observed curves using as a weighting function a kernel function, which is the absolute exponential kernel in our case. We later define its expression in the discretized setting.

2.1.2 Covariance kernel estimator

The empirical equivalent of the covariance kernel estimator proposed by Kraus [2015] is

$$r_n(s,t) = \frac{1}{n} \sum_{i=1}^{n} (X_i(s) - \mu_n(s))(X_i(t) - \mu_n(t)) \tag{3}$$

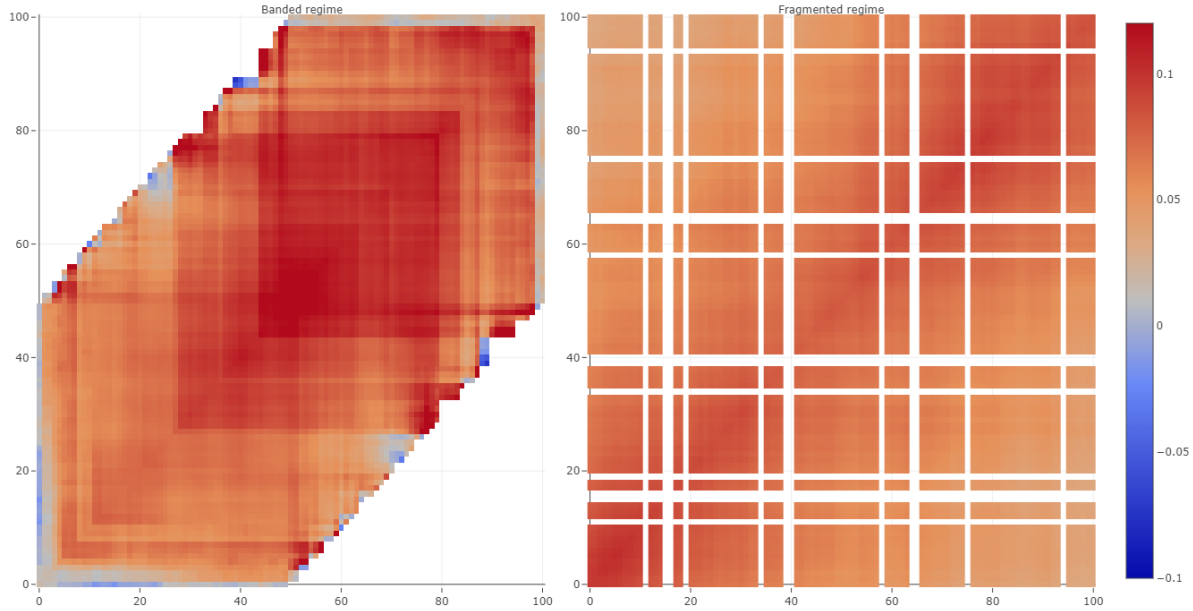

Figure 2: Graphical representation of two covariance kernels of the covariance operator of randomly generated functional samples of length 100, empirically estimated according to the method of Kraus [2015]. On the left, curves are assumed to be unobserved on a continuous subinterval of the domain ($\delta = 0.5$). On the right, the assumption of continuous interval does not hold anymore and, moreover, all curves are observed on the same subinterval.

where $s, t \in [0, 1]$ and the mean estimator μ_n is the one defined in Equation (2). In our case, as in Descary and Panaretos [2018], there can be couples of elements (s, t) of the domain such that no curve X_i is observed both at s and at t. This is the reason why a more correct formulation of the covariance kernel is the following for Descary and Panaretos [2018]:

$$r_n(s,t) = \frac{I(s,t)}{\sum_{i=1}^{n} U_i(s,t)} \sum_{i=1}^{n} U_i(s,t)[(X_i(s) - \mu_{n,st}(s))(X_i(t) - \mu_{n,st}(t))] \tag{4}$$

where $U_i(s,t) = \mathbb{1}_{O_i}(s)\mathbb{1}_{O_i}(t)$, $I(s,t) = \sum_{i=1}^{n} U_i(s,t)$ and $\mu_{n,st}(t) = \frac{I(s,t)}{\sum_{i=1}^{n} U_i(s,t)} \sum_{i=1}^{n} U_i(s,t)X_i(t)$. Given the setting of the case treated in this thesis, the previous formulation becomes

$$r_n(s,t) = \frac{I(s,t)}{n}[\sum_{i=1}^{n}(X_i(s) - \mu_n(s))(X_i(t) - \mu_n(t))] \tag{5}$$

It is evident that, with this estimator for the covariance kernel, $r_n(s,t) = 0$ if $s, t \in M = [0,1]\backslash O$, which is surely not a good estimate of the function, above all when the subinterval of observation is small with respect to $[0, 1]$. This implies that all the couples (s, t) such that $r(s,t) = 0$ are points of the unit square $[0,1]^2$ where the covariance kernel is initialized as zero and has to be reconstructed.

2.1.3 From continuous to discrete

We consider the case in which functional data are only measured on a finite grid of points. Even if the data in our case study are georeferenced, we still present a one-dimensional formulation, since each two-dimensional functional datum can be converted into a one-dimensional object by maintaining continuity across columns. This flattening procedure is explained in Section 5.
Each curve is represented by K evaluations at specific locations $(t_1, ..., t_K) \in T \subset O$, each corresponding to one successive point of the domain. More specifically, given the equally spaced partition $\{I_j\}_{j=1}^{K}$ of the domain $[0, 1]$, the curve X_i is explained by $X_i(t_1), X_i(t_2), ..., X_i(t_3)$, where $X_i(t_j) \in I_j$ for all $j \in 1, 2, ..., K$. This entails the following K-resolution representation of X:

$$X_i^K(t) = \sum_{j=1}^{K} X_i(t_j)\mathbb{1}(t \in I_j). \tag{6}$$

As a consequence, the discretized mean function estimator is defined as follows. If $t \in O$, then

$$\mu_n^K(t) = \frac{1}{n}\sum_{i=1}^{n} X_i^K(t) \tag{7}$$

otherwise, if $t \in M$, then

$$\mu_n^K(t) = \frac{\sum_{j=1}^{K} k(t,t_j)X_i^K(t_j)}{\sum_{j=1}^{K} k(t,t_j)} \tag{8}$$

where the kernel absolute exponential function is $k(t,s) = \sigma^2 \exp\left(\frac{-\|t-s\|}{\ell}\right)$, with σ being the scale factor and ℓ the length scale.
Nevertheless, this is just one specific way of estimating the mean function at missing locations; indeed many other smoothing techniques can be applied to solve this issue, which is not the main interest of this thesis.
Transitioning to the definition of the covariance kernel on the grid, this is defined as:

$$r^K(s,t) = Cov(X_i^K(s), X_i^K(t)) = \sum_{j,l=1}^{K} r(t_j,t_l)\mathbb{1}((s,t) \in I_j \times I_l). \tag{9}$$

In the previous formula, if we substitute $r(t_j,t_l)$ with the covariance kernel estimator previously defined $r_n(t_j,t_j)$, we obtain the $K-$ resolution version of the estimator, which can be synthesized by the $K \times K$ covariance matrix $R_n^K = \{r_n(t_j,t_l)\}_{j,l=1}^K$, as shown in the heatmaps in Figure 2. This matrix is incomplete, since any evaluation $X_i^K(t_j)$ is unobserved if $t_j \in M$. In the next subsection, we explain the method of Descary and Panaretos [2018] developed to estimate the values of the missing cells in the banded regime.

2.2. Low-rank matrix completion for covariance reconstruction

Starting from the approach of Descary and Panaretos [2018], we initially consider a matrix completion problem, which essentially aims to recover a low-rank matrix from a partially observed matrix.
In this subsection, we describe the optimization problem of the method of Descary and Panaretos [2018] and consider its resolution in our problem setting. To ensure the identifiability of the problem of their work, two main assumptions are made: the first one consists in assuring that $r(s,t)$ admits a Mercer decomposition with finite rank q, i.e. $r(s,t) = \sum_{j=1}^{q} \lambda_j \psi_j(s)\psi_j(t)$, and orthogonal eigenfunctions being real and analytic on $(0,1)$; the second one requires $K > \frac{2q+1}{\delta}$ and $\delta \in (0,1)$, being $\delta = |O_i|$, $\forall i = 1,...,n$.
Considering our context, the first assumption is satisfied by Mercer's theorem, since $r(s,t)$ is the covariance kernel of a covariance operator defined on $L^2([0,1])$, so it admits a Mercer decomposition and its eigenfunctions are real and continuous on $[0,1]$. Analyticity of the eigenfunctions and a finite rank of the Mercer decomposition are additional assumptions that are crucial to achieve a unique solution, so we also incorporate them in our problem setting. On the other hand, the second assumption is easily satified, taking into account that $\delta = \frac{n_{obs}}{K}$, where $n_{obs} = |\{t_j : t_j \in O, j = 1,...,K\}|$.
Being these conditions satisfied, the following matrix completion problem admits the unique solution

$$\hat{R}^K = \{\hat{r}_{j,l}\}_{j,l=1}^K = \{\hat{r}(t_j,t_l)\}_{j,l=1}^K = \underset{\theta \in \mathbb{R}^{K\times K}}{\operatorname{argmin}} \left\{ \frac{\|P^K \circ (R_n^K - \theta)\|_F^2}{K^2} + \tau \operatorname{rank}(\theta) \right\} \tag{10}$$

where $\tau > 0$ is a sufficiently small tuning parameter and P^K is a matrix with only 0 and 1 entries, having value 1 in the cells of the matrix corresponding to observed locations, 0 otherwise. By performing the element-by-element multiplication of P^K with $(R_n^K - \theta)$, one obtains the same $(R_n^K - \theta)$ matrix, filtered on the only observed part of the matrix, so assigning zero values to the missing cells. Hence, P^K can be regarded as a filtering mask for the observed cells.
As a consequence, the first term of the objective function preserves the information of the cells of R_n^K having non-zero value, where the covariance is estimated empirically just following the method of Kraus [2015], without influencing the missing cells. The second term instead, minimizing the rank of the whole matrix, acts on the estimate of the missing cells.
The practical problem to be solved consists in finding the best rank to approximate the previously estimated (incomplete) covariance matrix. To achieve this, a series of rank-constrained minimization problems are solved, with the addition of a hyperparameter $\tau > 0$ that prevents us from the trivial and not always optimal choice of selecting the highest rank as the one minimizing the objective function. As a consequence, by removing the rank constraint from the objective function, the problem achieves convergence in a much faster and simpler way.

Algorithm 1 Best rank estimation algorithm

1: **for** $i = 1, ..., \lceil K\delta \rceil$ **do**
2: solve the minimization problem

$$\min_{0 \leq \theta \in \mathbb{R}^{K \times K}} \left\{ \frac{\|P^K \circ (R_n^K - \theta)\|_F^2}{K^2} \right\}$$

$$\text{subject to } \text{rank}(\theta) \leq i$$

3: given the result $\hat{\theta}_i$ of the previous step, compute $f(i) = \frac{\|P^K \circ (R_n^K - \hat{\theta}_i)\|_F^2}{K^2}$
4: **end for**
5: choose the best rank i^* as the one minimizing $f(i) + \tau i$, for a fixed choice of $\tau > 0$.

In order to achieve a more rapid resolution, the problem at step 2 of Algorithm 1 can be re-elaborated by exploiting the positive-semidefinitedness of covariance matrices. Indeed, any positive-semidefinite matrix of rank i can be decomposed as $\theta = \gamma\gamma^T$, with $\gamma \in \mathbb{R}^{i \times K}$. In this case, the problem becomes unconstrained and is formulated as follows:

$$\min_{\gamma \in \mathbb{R}^{K \times i}} \|P^K \circ (R_n^K - \gamma\gamma^T)\|_F^2. \tag{11}$$

The initialization chosen by Descary and Panaretos [2018] for γ is $\gamma_0 = U_i \Lambda_i^{\frac{1}{2}}$, with $U\Lambda U^T$ being the singular value decomposition (SVD) of R_n^K, U_i being the matrix corresponding to the first i columns of U and Λ_i being the matrix generated by the first i rows and columns of Λ. Indeed, as stated by Ekart-Young theorem, the best rank-i approximation of a matrix, in terms of the Frobenius norm, is given by the truncated SVD of the matrix, which explains the choice in setting this value as starting point.
In the work of Descary and Panaretos [2018], a Quasi-Newton method is used to solve the optimization problem, in particular the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm, which iteratively builds up an approximation of the inverse Hessian matrix only using the gradient of the objective function. It is a method that achieves global optimality when the problem is convex, and the objective function in Equation (11) is convex in θ. Nevertheless, it is not convex in γ, which is necessarily introduced to remove the rank constraint and the symmetric positive definite condition for θ. This means that it is not guaranteed that the problem converges to a unique solution. However, the method is proven to be stable by Descary and Panaretos [2018], and often leads to unique and fast solutions.
Finally, the choice of the value for the hyperparameter τ is finalized by plotting, for each possible value of τ, the solutions $f(i)$ of step 2 of the Algorithm 1 over the range of the rank values i. A non-increasing behaviour is expected in each plot, especially for low values of τ, and Descary and Panaretos [2018] select τ by observing an elbow in the plot and by demanding that $f(i_\tau)$ is lower than a certain threshold ϵ, where i_τ is the rank minimizing the objective function for that choice of τ.
Details regarding the utilization of the low-rank matrix completion algorithm to reconstruct the covariance kernel in our problem setting are provided in Section 3.3.

2.3. Data reconstruction

Once the covariance matrix R^K is fully estimated, the covariance kernel estimator can be deduced from it, to directly form the covariance operator needed for the reconstruction of partially observed functional data in the method of Kraus [2015].
Back to the population problem, the issue consists in finding the best estimator of X_M given X_O, which is $\mathbf{E}[X_M|X_O]$. To have a simpler problem to solve, its best linear predictor is considered, which is (assuming null mean without loss of generality)

$$\hat{X}_M = \mathscr{A} X_O \tag{12}$$

with $\mathscr{A}$ being a continuous (bounded) linear operator from $L^2(O)$ to $L^2(M)$, chosen minimizing the expected square error. The result of the minimization problem is $\mathscr{A} = \mathscr{R}_{MO}\mathscr{R}_{OO}^{-1}$, with $\mathscr{R}_{MO}$ being the covariance operator restricted to $M \times O$ and, similarly, $\mathscr{R}_{OO}$ the one restricted to $O \times O$. To provide stability, the actual solution is $\mathscr{A}^{(\beta)} = \mathscr{R}_{MO}{\mathscr{R}_{OO}^{(\beta)}}^{-1}$, where $\mathscr{R}_{OO}^{(\beta)} = \mathscr{R}_{OO} + \beta\mathscr{I}_O$ is the Ridge regularized version of $\mathscr{R}_{OO}$, to prevent the unboundedness of its inverse. The regularization parameter β is chosen through generalized cross-validation. As a result, the final reconstruction of each fragmented curve X_i is obtained by computing

$$X_{iM_i} = \mu_{nM_i} + \mathscr{R}_{M_iO_i}\mathscr{R}_{O_iO_i}^{(\beta)\ -1}(X_{iO_i} - \mu_{nO_i}). \tag{13}$$

3. Covariance reconstruction procedure

In this section, we report the considerations, in each step of the problem resolution, that compose our method and bring us to the proposed reconstruction procedure. In particular, we describe how we estend the method of Descary and Panaretos [2018] to our problem setting, by modifyng the objective function of the optimization problem adding Laplacian regularization.

3.1. Covariance estimation and diagonal completion

The first step of the procedure consists in constructing the covariance matrix with all the available information that our data provide. This is achieved applying Eq.(5) in the discrete formulation of Eq.(9).
As regards the application of the low-rank matrix completion procedure (Section 2.2) to our fragmented regime, although the same assumptions of the work of Descary and Panaretos [2018] can be made, the covariance kernel structure of the fragmented regime is not well-suited for the specific optimization problem. Indeed, we are not assuming the set of observation O to be continuous in the K-grid, i.e. considering the discrete formulation, it may happen that $X_i(t_j)$ and $X_i(t_{j+2})$ are observed evaluations of X_i, but $X_i(t_{j+1})$ is not, concurrently for all $i = 1, ..., n$ since $O_i = O$ for all i. This means that R_n^K has entire rows and columns missing, generating discontinuity along the diagonal and making it difficult to apply the algorithm of Descary and Panaretos [2018], designed for the fragmented regime. This is clearly observable in Figure 2, where, on the left, the covariance kernel estimated for the covariance operator in a banded regime is shown, presenting a full diagonal (which is always the case when $\delta \geq 0.5$), since functional fragments are continuous. On the right, instead, an estimated covariance kernel for the fragmented regime is shown, displayng missing values over the diagonal.
Indeed, the method of Descary and Panaretos [2018] relies on analyticity, to achieve that, at each iteration i of the Algorithm 1, all minors of rank i are non-vanishing. Then, by combining analyticity with vanishing minors of order $i+1$, the matrix is gradually filled by solving determinental equations. By construction, this procedure is very efficient when the objective is to estimate the off-diagonal having a fully observed diagonal (banded regime). It encounters difficulties when, instead, the diagonal of the matrix presents missing values that require estimation.
To solve this issue, we suggest to fill the diagonal of the covariance matrix with appropriate non-zero values. Just like in the case of the mean estimator, several techniques can be applied to fill the diagonal of the matrix and still achieve good results. For instance, in the simulation study, which considers a stationary covariance matrix, the diagonal is filled by setting all the missing values of the diagonal equal to the mean of the diagonal, computed considering its observed cells. In the case study, instead, Nadaraya–Watson smoothing technique is applied once again, this time to estimate the variance of the missing pixels, since it directly corresponds to the diagonal of the covariance matrix. In this case, the choice of a smoothing technique is strongly motivated by the two-dimensionality of data, where a certain level of continuity between close pixels is presumed.
Once the diagonal of the covariance kernel in the fragmented regime is filled, the low-rank matrix completion proposed by Descary and Panaretos [2018] can be directly applied. However, the good performance of this method is not assured, since the setting we are considering does not entirely belong to the banded regime, even with a full diagonal for the covariance matrix.

3.2. Laplacian regularization

The application of the method of Descary and Panaretos [2018] to partially observed functional data in the fragmented regime results in an irregular reconstruction of the covariance kernel. In fact, while the values of the observed cells are accurately estimated by well-preserving the information carried by the observed parts of the functional curves, the values of the missing cells are barely optimized in the minimization problem.
As a result, we obtain an unlikely covariance kernel estimate, which is strongly discontinuous and very sensible to the initial condition of optimization problem, leading to unstable and unreliable solutions.
Therefore, the only usage of these two terms in the objective function of Equation (10) is not capable of capturing the true nature of the covariance operator that we want to estimate, when functional data are missing as in the fragmented regime.
That is the reason why we propose to add a Laplacian regularization term in the objective function to be minimized, to generate a much smoother reconstructed matrix. In fact, Laplacian regularization encourages smoothness and consistency in the solutions by penalizing abrupt changes or oscillations. This is particularly

beneficial in problems involving data with spatial or relational dependencies, where smoothness behaviour is desired. The introduction of this term might assist in maintaining a local structure in the covariance function, and, by this means, tackle the issue of irregularity.
Additionally, we introduce weights in the Laplacian regularization term, by assigning varying weights to the Laplacian computed in the observed or in the missing part of the covariance function, assuming that different parts of the initial estimate of R^K might be weighted in different ways. In the simulation study in Section 4 and in the case study in Section 5, we experiment with various weight values, to determine their applicability and utility within the context of our work.
The result is the following regularization term:

$$\mathrm{Tr}((P^m \circ (L^{\frac{1}{2}}\theta))^T(P^m \circ (L^{\frac{1}{2}}\theta))) \tag{14}$$

which mainly involves three elements: θ, P^m and L. $\theta \in \mathbb{R}^{K\times K}$ is the discretized covariance kernel that is being reconstructed through the optimization algorithm. $P^m \in \mathbb{R}^{K\times K}$ is the weight matrix, such that $P^m = (1-m)P^K + mP^{-K}$, where $P^{-K} \in \{0,1\}^{K\times K}$ is the complementary mask, i.e. $P^{-K} = 1 - P^K$, having value 1 for the missing cells and 0 for the remaining ones; $m \in [0,1]$ is the weight parameter, whose value is discussed and fixed through parameter selection in the simulation study (Section 4) and through cross-validation in the case study (Section 5). $L \in \mathbb{R}^{K\times K}$ is the Laplacian matrix (Maunu [2023] [12] and Pang [2017] [13]), which is symmetric and positive semi-definite. It holds that $L = D - A$, where A is the adjacency matrix and D the degree matrix. $A \in \{0,1\}^{K\times K}$ is a symmetric matrix indicating whether or not two points in the domain - indexed from 1 to K - are connected. If j and l are connected, then $a_{jl} = 1$, otherwise $a_{jl} = 0$. With the purpose of setting as adjacent successive points in the domain, i.e. we want j to be connected to $j-1$ and $j+1$ for all $j = 2, ..., K-1$, we assume that $a_{jl} = 1$ if $|j-l| = 1$, and $a_{jl} = 0$ otherwise. $D \in \mathbb{R}^{K\times K}$ is a diagonal matrix which indicates how many indices $l = 1, ..., K$, with $l \neq j$, are connected to each index j, i.e. such that $d_{jj} = \sum_{l=1}^{K} a_{jl}$.
When a Laplacian regularization term is added to the objective function of the matrix completion problem, it usually has the shape $\mathrm{Tr}(M^T LM)$ (as in Tang et al. [2019]), with M being the matrix to be regularized. The regularization term is typically added to discourage sudden variations in the learned matrix across neighboring cells and the Laplacian matrix is the main component of the term that enables this penalization.

3.3. Optimization problem

In light of considerations articulated in the preceding two subsections, the complete optimization problem that we propose is

$$\min_{\theta\in\mathbb{R}^{K\times K}} \left\{ \|P^K \circ (R_n^K - \theta)\|_F^2 + \tau\mathrm{rank}(\theta) + \alpha\mathrm{Tr}((P^m \circ (L^{\frac{1}{2}}\theta))^T(P^m \circ (L^{\frac{1}{2}}\theta))) \right\} \tag{15}$$

and it is solved in three main steps:

1. estimate the empirical covariance R_n^K according to Equation (5) and the full diagonal of the estimated matrix;
2. solve the optimization problem in Equation (10) to find the best rank, by setting τ as fixed;
3. solve the optimization problem in Equation (15) at fixed best rank found at step 2, and fixing α and m, with $\alpha > 0$ being the penalization parameter for the Laplacian regularization term.

The second step of the procedure consists in the application of the method of Descary and Panaretos [2018] (Section 2.2) with the purpose of finding the best rank for the resulting covariance kernel. We select τ among the quantities 10^{-10}, 10^{-8}, 10^{-6}, 10^{-3}, 10^{-2}, 10^{-1}, so we repeat Algorithm 1 for each possible value of τ. Then, we plot $f(i) + \tau i$ as a function of the rank i for different choices of τ and highlight the minimizing rank for each case. Finally, we choose the smallest τ not selecting the maximum rank as best rank. Fig.(3) provides a practical example of rank choice.
The third and final step of our method consists in solving the problem in Equation (15) setting $\mathrm{rank}(\theta)$ equal to the best rank found at the previous step. The practical final optimization step to be solved is an unconstrained minimization problem:

$$\min_{\gamma\in\mathbb{R}^{K\times i}} \left\{ \|P^K \circ (R_n^K - \gamma\gamma^T)\|_F^2 + \alpha\mathrm{Tr}((P^m \circ (L^{\frac{1}{2}}\gamma\gamma^T))^T(P^m \circ (L^{\frac{1}{2}}\gamma\gamma^T))) \right\}. \tag{16}$$

Similarly to the resolution of the low-rank matrix completion problem of Descary and Panaretos [2018] outlined in Section 2.2, we apply the BFGS Quasi-Newton method to solve the problem in Equation (16), and the initialization for γ is once again the square root of the truncated SVD of θ. The main difference in this case is in the gradient function, which is intuitively different since the objective function is different.
Moreover, the values of the hyperparameters $\alpha > 0$ and $m \in [0,1]$ are fixed at this stage of the procedure. Their fixed values are discussed and chosen in the simulation study in Section 4 and in the case study in Section 5.

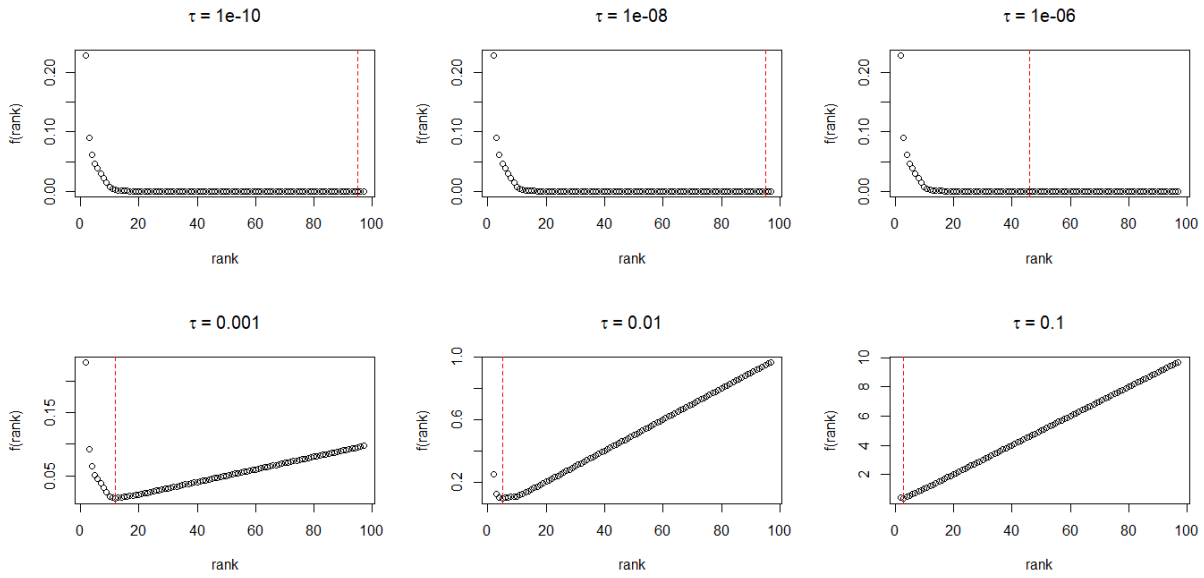

Figure 3: We report an example of best rank selection through τ selection. In each of the subplots, the minimizing rank is highlighted by the red dashed line. The best τ is 10^{-6}, since it is the lowest τ selecting as best rank (46) a value different from the maximum rank (97).

Differently from the previous step, where all possible ranks needed to be tested, in this last step the optimization problem is directly solved and quickly leads to a unique solution.
The resolution of this minimization problem results in finding a covariance matrix which effectively exploits the information carried by the observed cells, while guaranteeing continuity over the missing rows and columns of the matrix.

4. Simulation study

In this section, we conduct a simulation study to evaluate the performance of the method of covariance reconstruction outlined in Section 2 and 3. Specifically, our focus lies in evaluating the suitability of the optimization problem (16) for reconstructing a covariance kernel within our fragmented regime and identifying the selection criteria for the hyperparameters α and m.
In order to attain this objective, in the following subsections, we describe in detail how data are generated, how the performance is evaluated and, finally, how the hyperparameters α and m entering problem (16) can be selected.

4.1. Data simulation

Our simulated data consist of 100 functional samples of a Gaussian process having zero mean and Matérn covariance kernel, which is defined as:

$$r_M(s,t) = r_M(|s-t|) = r_M(d) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{d}{\phi}\right)^{\nu} K_\nu \left(\frac{d}{\phi}\right) \tag{17}$$

with $(s,t) \in [0,1]$, where $\Gamma(.)$ denotes the Gamma function, K_ν is the modified Bessel function of the second kind, d is the distance between the locations s and t, and ν and ϕ are non-negative parameters. Considering the discrete case, we generate each sample over 101 locations, i.e. each functional datum is a curve over 101 successive equidistant points of the domain, which implies that $(s,t) \in \{1,2,...,101\}$ and $d \in [0,100]$. For our study, we set the values of the parameters to $\sigma^2 = 0.1$, $\nu = 0.5$ and $\phi = 101$.
The Matérn covariance kernel is stationary, since the covariance of two points s and t only depends on the distance d between the two points. Similarly to Descary and Panaretos [2018], we opt to employ the Matérn covariance kernel in our simulated scenario to further assess the stability of our method in cases where some of the initial assumptions are not fulfilled. In fact, the Matérn covariance function is not of finite rank and does not have analytic eigenfunctions, which are significant conditions of the rank minimization problem proposed

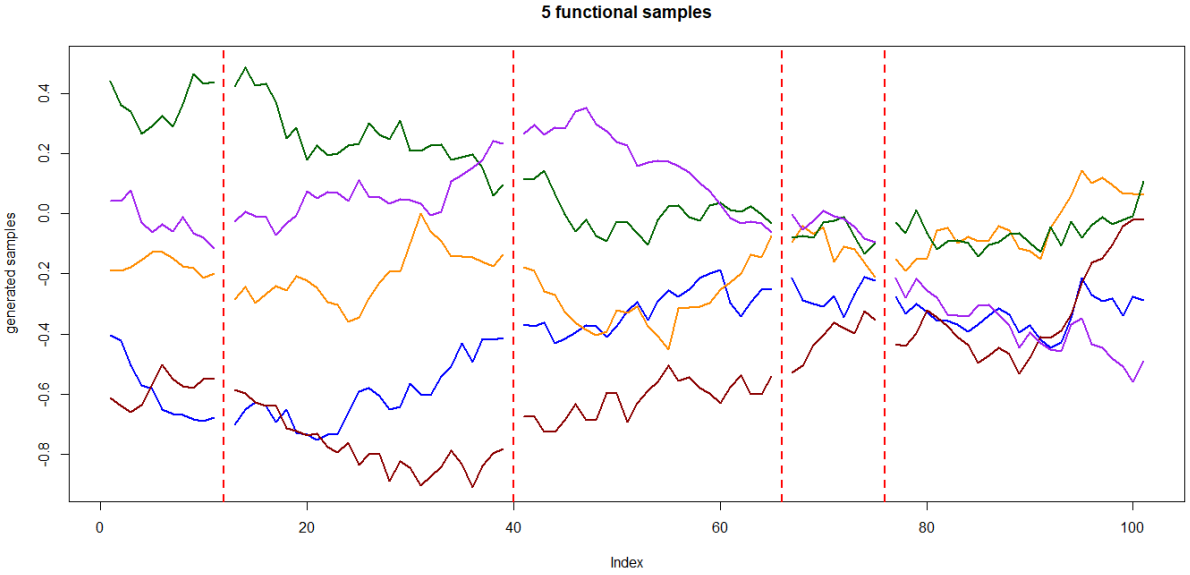

Figure 4: Functional data generated from a centered Gaussian process with Matérn covariance function, with missing values at indices 12, 40, 66 and 76. Here, we report 5 of the 100 samples generated for data simulation described in Section 4.1

by Descary and Panaretos [2018] and described in Section 2.2, that we also incorporate into our resolution procedure.
Once our data are generated, each realization of the process has 101 cells, and, to enter the fragmented regime, we set some of the cells as missing value, at the same locations for all the data. In particular, we assign missing values to each functional sample X_i at indices 12, 40, 66 and 76, since this is a specific allocation of missing values that we find in the case study. In Figure 4, we report 5 of 100 samples that we generate for this simulation study.
Being the partially observed functional data in the fragmented regime now available, we compute the covariance between each couple of locations (s, t). The resulting covariance matrix is empty at rows and columns corresponding to the missing indices, since these are locations that are never observed in any of the 100 samples. At this stage, what remains to be done is the estimate of the diagonal at the unobserved locations, since our method, combined with the one of Descary and Panaretos [2018], requires the diagonal to be fully observed. Taking advantage of the stationarity of the Matérn covariance kernel, we set the missing values of the diagonal equal to the mean of the observed elements of the diagonal. The resulting covariance to be reconstructed is shown on the top right panel of Figure 5.

4.2. Performance evaluation

To test the performance of our method, we conduct a Monte Carlo simulation over 50 covariance functions, such that each covariance is computed from 100 samples of a centred Gaussian process with Matérn covariance function, as specified in Section 4.1, and presents same missing cells at indices 12, 40, 66 and 76.
In order to compare the performance of different combinations of parameters α and m, we use the root mean squared error (RMSE) along all the cells of the matrix as a reconstruction index, considering both the observed cells and the missing cells.
Given the true covariance $R^K = \{r_{ij}^K\}_{i,j=1}^K$ and the reconstructed covariance $\hat{R}^K = \{\hat{r}_{ij}^K\}_{i,j=1}^K$, the root mean squared error is formulated as follows:

$$\text{RMSE}(\hat{R}^K) = \sqrt{\frac{\sum_{i=1}^K \sum_{j=1}^K (r_{ij}^K - \hat{r}_{ij}^K)^2}{K^2}}. \tag{18}$$

The RMSE, as seen Equation (18), is an index of the error per cell of the matrix after the reconstruction. In our study, we also evaluate separately the root mean squared error over the only observed part of the matrix and over the only missing part of the matrix. As a consequence, we also compute the following errors for each combination of parameters:

$$\text{RMSE}_O(\hat{R}^K) = \sqrt{\frac{\sum_{i=1}^{K}\sum_{j=1}^{K} P_{ij}^K (r_{ij}^K - \hat{r}_{ij}^K)^2}{N_{obs}}} \tag{19}$$

$$\text{RMSE}_M(\hat{R}^K) = \sqrt{\frac{\sum_{i=1}^{K}\sum_{j=1}^{K} P_{ij}^{-K} (r_{ij}^K - \hat{r}_{ij}^K)^2}{N_{miss}}} \tag{20}$$

where N_{obs} is the number of non-zero elements of the mask matrix P^K and N_{miss} is the number of non-zero elements of the complementary mask P^{-K}, which is equal to the number of zero elements of P^K.
More specifically, the denominator N_{miss} denotes the number of missing cells of the matrix, reminding that the incomplete covariance matrix entering our optimization algorithm is fully observed over the diagonal. N_{miss} can be defined as

$$N_{miss} = 2n_{miss}K - n_{miss}^2 - n_{miss} \tag{21}$$

where $n_{miss} = K - n_{obs}$ is the number of missing indices in each functional sample. For instance, in our study, $n_{miss} = 4$ and $K = 101$. The expression $2n_{miss}K$ represents the total count of elements found in n_{miss} rows and n_{miss} columns of length K. Because the preceding expression tallies duplicate values for the intersecting elements across rows and columns in the matrix, the term $-n_{miss}^2$ subtracts these redundant counts. Ultimately, given that our covariance matrix features a full diagonal, we exclude from consideration all counts on the diagonal corresponding to missing rows and columns. Hence, we subtract these values from the sum using the term $-n_{miss}$. Thus, the number of observed cells of the matrix is $N_{obs} = K^2 - (2n_{miss}K - n_{miss}^2 - n_{miss})$. With this formulation of the denominators for Equation (19) and (20), the root mean squared error related to the observed or missing part of the matrix are also errors per cell.

4.3. Parameters selection

In this section, we aim to provide a rule for selecting the best parameters α and m yielding the most optimal reconstruction of the incomplete covariance kernel. To achieve this, the rank of the resulting matrix is chosen - and fixed - according to the procedure described in Section 2.2. In particular, applying the rank minimization problem of Descary and Panaretos [2018] to one of the 50 covariances generated in the Monte Carlo simulation (with filled diagonal), the resulting best rank is $r_{opt} = 48$. Therefore, the optimization problem (15) is solved for a specific choice of parameters by searching for solutions in the space of $K \times K$ matrices of rank r_{opt}.
We perform tuning of the parameters through grid search, selecting from values within the interval $[0, 1]$ for m and within the interval $[0, 100]$ for α, where α values are powers of 10.
By considering these as candidates for the parameters, we also aim at testing the limit cases of our method. In fact, having as best solution $\alpha = 0$, would mean that there is no need to penalize for the roughness of the resulting reconstructed convariance. On the contrary, if $\alpha = 100$ is the best choice for α, then much more weight is given to the smoothing term, which may result in a reconstruction that assigns the same value to all cells of the matrix.
Similarly, if the best value for m is 0.5, it means that it is unnecessary to use a system of weights inside the Laplacian regularization term, since the same weight would be assigned to the observed and to the missing parts and, by the properties of trace, the value of the m parameter would be embedded in the Laplacian regularization parameter α. Instead, if $m = 0$ or $m = 1$ is the best solution, then the Laplacian should be evaluated respectively only on the observed part or only on the missing part of the matrix.
By looking at Figure 5, it is evident that none of these limit cases leads to a reliable reconstruction, at least from a visual perspective. In the remainder of this section, where the performance of the algorithm is evaluated in terms of reconstruction error, the performance of these limit cases will be expressed quantitatively and compared to other cases.
Thereafter, for each covariance function, the reconstruction algorithm is tested for each couple of parameters α and m, testing values for the parameters in the intervals specified earlier. Then, the mean and median of the root mean squared error over the 50 simulations are computed, as resulting in Tables 3 and 4 reported in Appendix B. In our simulation study, the best choice of parameters according to the total RMSE is $\alpha = 0.01$ and $m = 0.1$, since it generates the lowest mean and median error. In order to understand how this parameters selection is reached, we investigate the behaviour of the RMSE for different values of α and m, respectively, and we analyse how the three types of RSME, as formulated in Section 4.2, change based on the parameters.

4.3.1 Analysis per log(α)

We consider the behaviour of the error with respect to the values of the logarithm of α, keeping m fixed.

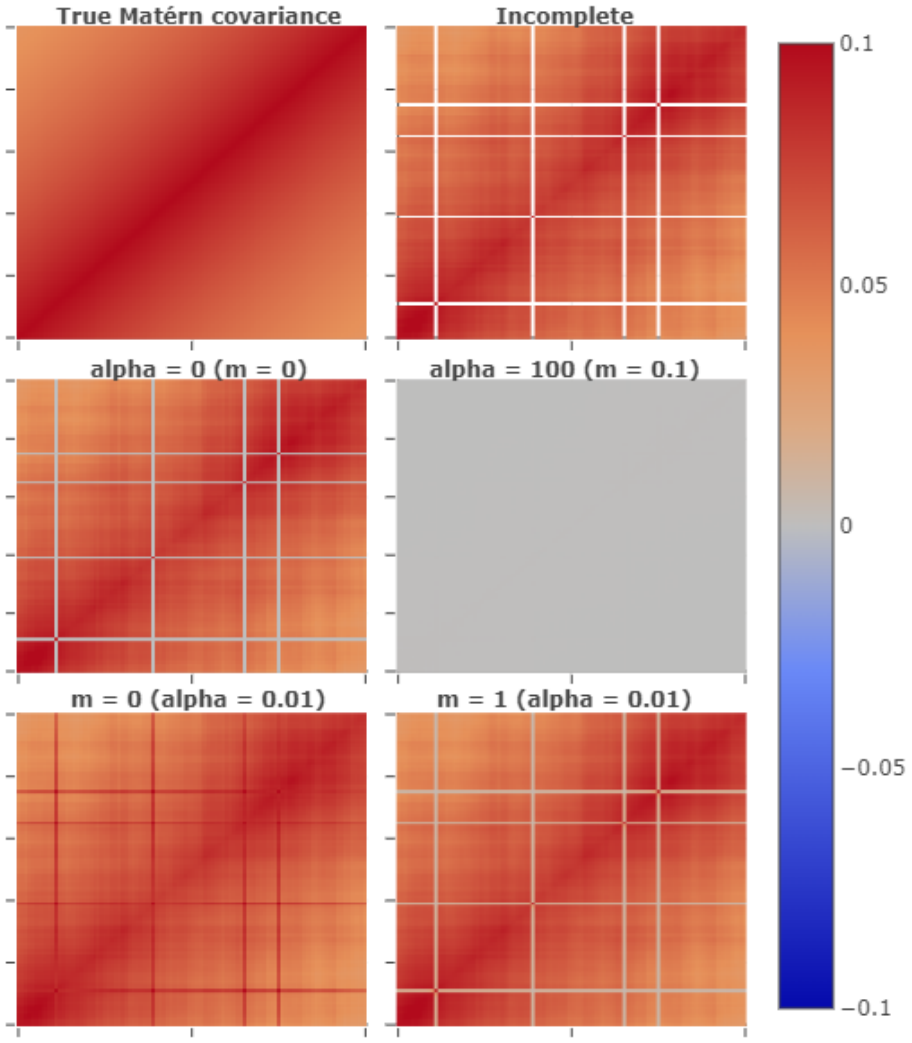

Figure 5: We here report the heatmap representing the true Matérn covariance function (on the top left panel) and the heatmap representing the incomplete covariance kernel that has to be reconstructed (on the top right panel). The remaining four heatmaps correspond to the reconstruction of the latter by means of the model described in Section 3.3, considering limit cases for the parameters α and m.

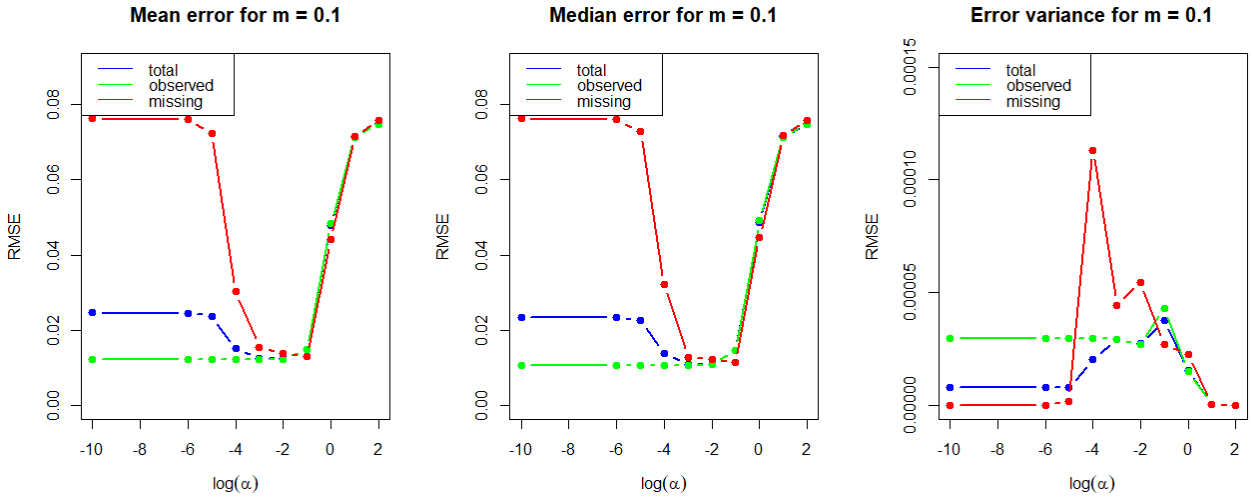

Figure 6: The mean, median and variance of RMSE over 50 simulations, for $m = 0.1$, are reported for each type of error: total RMSE (blue line), RMSE related to the reconstrucion of observed cells (green line), RMSE related to the reconstruction of missing cells (red line). The point $log(\alpha) = -10$ on the x-axis serves as a fictitious representation graphically introduced to illustrate the value of RMSE for $\alpha = 0$.

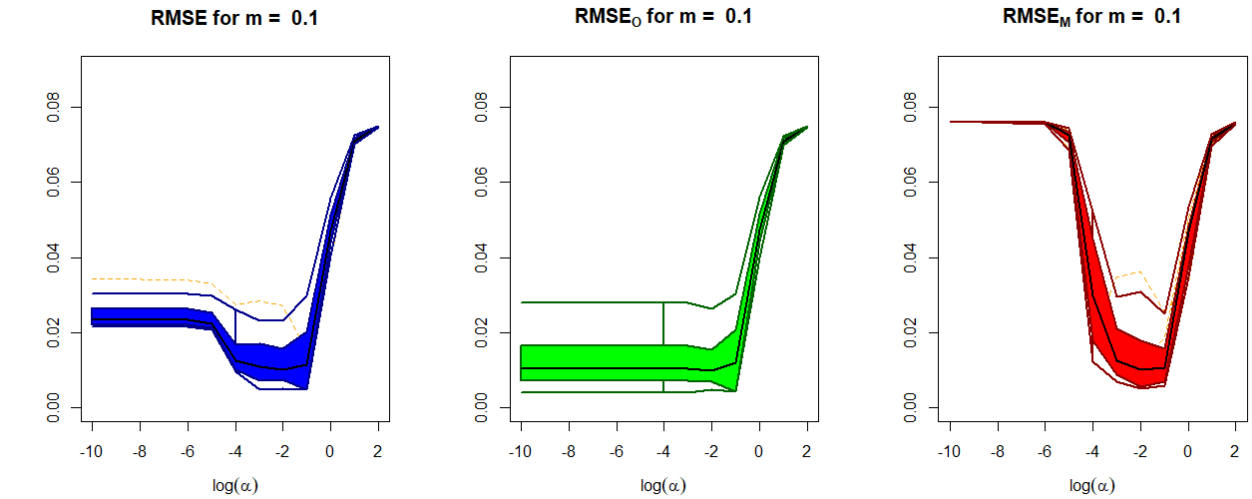

Figure 7: Functional boxplots of the curves of RMSE generated per each value of the logarithm of α over 50 simulations, for $m = 0.1$. Each functional boxplot is related to each type of error: total RMSE (blue), RMSE related to the reconstrucion of observed cells (green), RMSE related to the reconstruction of missing cells (red). The point $log(\alpha) = -10$ on the x-axis serves as a fictitious representation graphically introduced to illustrate the value of RMSE for $\alpha = 0$

The mean and the median root mean squared error have very similar patterns, so we here describe the behaviour of the mean, assuming that the median just has the same trend.
For any fixed value of m, the three curves corresponding to the three considered errors display a different behaviour for values of α smaller than 0.01, while they share a similar pattern for $\alpha > 0.01$.
For smaller values of α, the regularization parameter is insufficient to significantly impact the optimization problem. Consequently, the curve of mean RMSE$_M$ (Equation (20)) initially exhibits a higher value, gradually decreasing as the minimum $\alpha = 0.01$ approaches. On the other hand, the mean RMSE$_O$ (Equation (19)) initially remains constant, before beginning to increase once the minimum value - as regards the total error - is attained. This behavior is intuitive, as the cells associated with the observed portion of the matrix are effectively estimated by the initial term of the optimization problem in (16), rendering the Laplacian regularization term unnecessary for enhancing the reconstruction. This scenario corresponds to setting α to zero. Finally, the mean

RMSE appears to be a synthesis of the other two errors, with a notable influence from the curve associated with the missing cells. In fact, the minimum mean RMSE occurs when the mean RMSE_M reaches its minimum value.
For larger values of α, both the mean RMSE_O and mean RMSE_M curves and, consequently, the mean RMSE curve, exhibit very similar patterns. Indeed, it is expected that, if α is large, the second term of the optimization problem described in (16) - which is the smoothing term - predominates over the first term. This suggests that, when the model attempts to estimate the covariance, the significance of the information provided by the observed cells is reduced, resulting in an increase in the error associated with the observed part. As a consequence, since the missing cells are estimated based on a reliable knowledge of the observed cells, the mean RMSE_M likewise increases. The mean RMSE is a combination of the other two errors, so it also exhibits an increase.
Moreover, we can discern that, for big values of α, an asymptote is reached. This is justified by the fact that when α is too high, an excessive smoothing is enforced to the matrix. Consequently, with the increase in α, all cells of the matrix tend to converge towards a single value, reaching this value uniformly for α exceeding a certain threshold. This is also directly observable in Figure 5, where the heatmap of the covariance reconstruction for $\alpha = 100$ is reported.
The analysis of the error behaviour for several values of m demonstrates that it is possible to select the best value for α, even only having at our disposal the error related to the observed part, which is the exact situation when considering a real case study. This information will also be used in the covariance reconstruction performed in Section 5, where true data are utilized.

4.3.2 Analysis per m

Similarly to the previous analysis, we here examine how the reconstruction errors change based on different values for the parameter m, keeping α fixed.

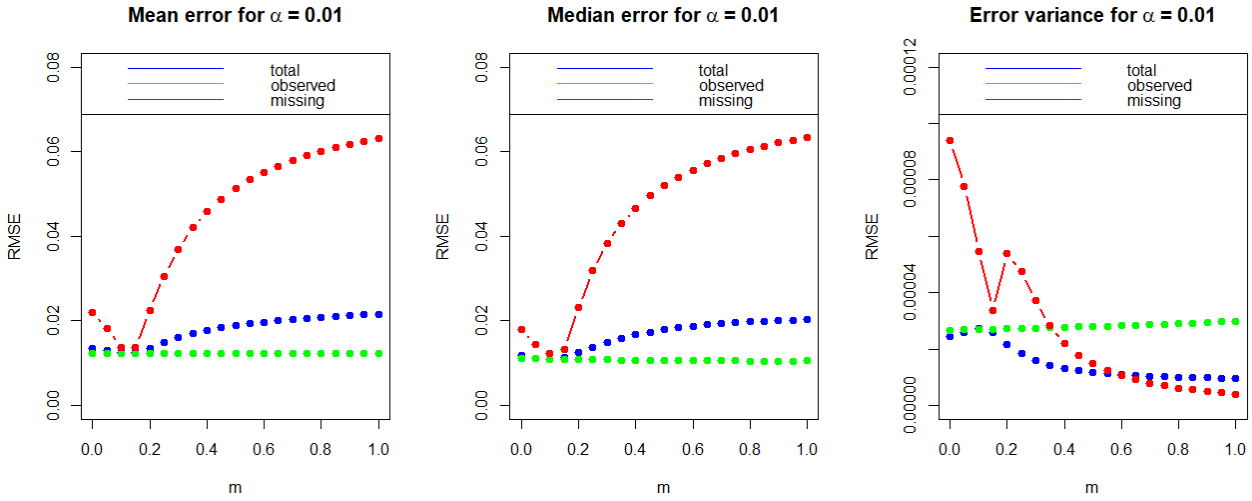

Figure 8: The mean and median RMSE and its variance over 50 simulations, for $\alpha = 0.01$, are reported for each type of error: total RMSE (blue line), RMSE related to the reconstrucion of observed cells (green line), RMSE related to the reconstruction of missing cells (red line).

As for the analysis of RMSE for m fixed, the patterns of the mean and median errors per different values of m are very similar, so we only provide commentary on the trend of the mean to avoid redundancy. The parameter m is incorporated into our problem formulation to assess whether a varied weighting of the observed and missing parts of the matrix could lead to a more effective reconstruction of the covariance function, particularly at the locations of missing cells. In fact, when α is fixed in correspondance of the optimal value 0.01, the curve of the mean RMSE_M clearly presents a global minimum in correspondance to the value of m providing the best reconstruction of the missing part of the matrix. Conversely, the curve of the mean RMSE_O is almost constant for small values of α and then starts decreasing when considering too large fixed values for α. This behaviour is due to the presence of the Laplacian regularization term which activates when $m < 1$, implying a greater regularity of the reconstructed matrix along with its lower adherence to the observed values. The behaviour of the total reconstruction error is influenced by both the errors related to the missing and to the observed cells, and, when α is optimal, the global minimum is achieved in correspondance of the reconstruction error on the missing part.

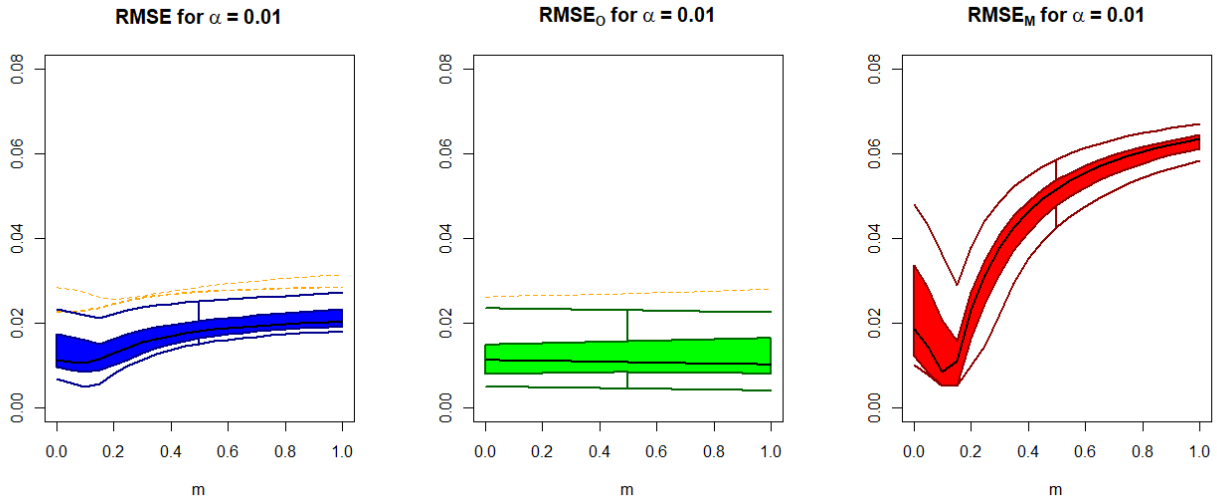

Figure 9: Functional boxplots of the curves of RMSE generated per each value of the logarithm of α over 50 simulations, for $\alpha = 0.01$. Each functional boxplot is related to each type of error: total RMSE (blue), RMSE related to the reconstrucion of observed cells (green), RMSE related to the reconstruction of missing cells (red).

It is relevant to note that, when a real case study is considered and the only error for the observed part of the matrix is available, it is not possible to set an optimal value for the m parameter just considering the charts in Figures 8 and 9. Moreover, choosing the best parameter m by only looking at the simulation study may not be a right choice. We refer to Section 5.3 for a thorough explanation on how cross-validation helps identify a criterion for choosing hyperparameter m in application contexts.
The examination of simulation results has disclosed strenghts and limits of our model. On one side, a right parameter choice can bring to very performing results, achieving a reliable reconstruction of the covariance kernel. On the other side, a correct parameter selection is hard to be made. Indeed, if the choice of α is straightforward looking at the curve of the reconstruction error related to the observed part of the matrix, the choice of m is much more challenging, mainly if the only available information is the one contained in the observed cells, that is always the case when solving real problems.
Furthermore, more covariance structures, different from the Matérn, could be involved in the simulation study, in order to test the robustness of the parameter choice. Results obtained with other combinations of parameters settings for the Matérn covariance function (not shown for brevity) confirm that the reconstruction procedure generally performs well; more extensive simulations will be the scope of future work.

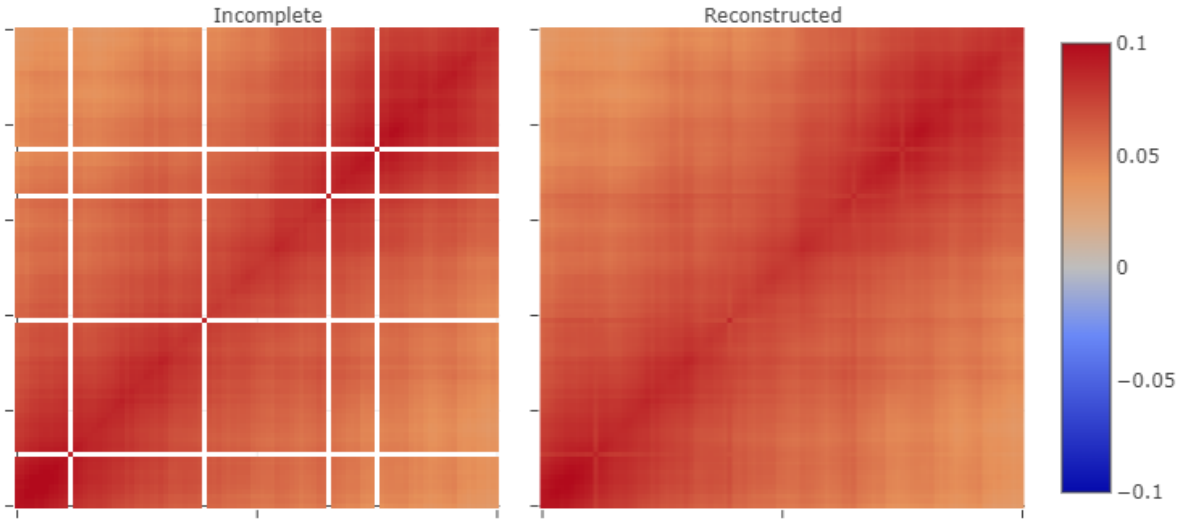

Figure 10: Reconstruction of the simulated covariance ($\alpha = 0.01$, $m = 0.13$).

5. Case study

Our focus now shifts to the primary motivation for commencing work on this problem, namely the application to the SBAS-processed images of the area of Phlegraen Fields, Italy.
In this section, we test on real data the procedure described in Section 2 and 3, drawing on the conclusions of the analysis carried out in the simulation study illustrated in Section 4. First, we describe the setting and motivation of the case study. Then, we reconstruct the covariance function of partially observed spatial functional data for a specific area of interest, choosing the best parameters through a targeted analysis. Eventually, we reconstruct each spatial functional observation.

5.1. Motivation

Satellites are equipped with (sensors) which measure the radiation that is reflected from the Earth. There are passive sensors, which use the Sun as a source of energy, or active sensors, which are a source of energy themselves. The former have some limitations since they only operate in the visible, infrared, thermal infrared and microwave portions of the electromagnetic spectrum. The latter, instead, which were only recently introduced in the field of remote sensing, operate within the microwave segment of the electromagnetic spectrum, allowing them to penetrate the atmosphere effectively even with adverse climatic conditions. Synthetic Aperture Radar (SAR) is an instance of active sensor technology. Thanks to its ability to collect images in any climatic condition, regardless of the presence of daylight, and to its high precision, this is a very powerful technology in monitoring and forecasting natural catastrophes, such as landslides, earthquakes, volcanic phenomena, and ground subsidence.
In practice, each SAR image associates a complex number to each pixel of the image representing the area of interest. By combining two SAR images, corresponding to acquisitions of the same area from different locations, multiplying the first image with the complex conjugate of the second image pixel by pixel, one obtains the SAR interferogram (InSAR), which contains the phase difference per pixel between the two SAR acquisitions, a crucial information that allows the detection of ground surface displacement. When the phase difference is explicitly accounted for, differential interferograms (DInSAR data) are derived. However, they are often impacted by ambiguities stemming from geometrical factors - such as changes in reflectivity due to variations in the incidence angle of radiation - or temporal factors - such as alterations in surface reflectivity over time, influenced by seasonal changes in vegetation cover - leading to decorrelation. In order to address this challenge, Advanced DInSAR techniques are employed to quantify the decorrelation and accurately estimate the phase difference. Small BAseline Subsets (SBAS) is a multi-temporal Advanced DInSAR technique which manipulates the temporal series of ground displacement to reduce decorrelation. Furthermore, it assigns a value between 0 and 1 to each pixel of the image representing the studied area, defined *temporal coherence*. If a pixel exhibits coherence equal to zero, it indicates that decorrelation has not been entirely mitigated and the information within the pixel is unreliable. Conversely, a coherence value of 1 indicates high trustworthiness.
The real data utilized for this study consist of differential interferograms of SAR data, where the SBAS technique has been applied.

5.2. Data

Our data involve a time series of $n = 391$ successive images, representing georeferenced ground displacement values, recorded over Phlegraen Fields (Italy), an area which is prone to seismic and bradisismic events. The peculiarity related to these recordings is a value of coherence associated to each pixel. Whenever the coherence is below a given threshold, the information is considered unreliable and hence discarded. In any further analysis, the pixel is associated to a missing value.
Looking at Figure 1, it is evident that coherence can be significantly low across many pixels. In particular, it can be observed that most of the areas with low coherence correspond to see-covered lands, where the reflectivity of water negatively influences the recording of data by satellites. This is the reason why, for our analysis, we focus on reconstructing a smaller region, which is the red square highlighted in Figure 1, that we will refer to as *red subwindow*, covering 101×101 pixels. Furthermore, we decide to set the values of ground displacement as missing if they correspond to pixels having coherence lower than 0.8, which is a threshold value for coherence commonly employed in this type of analyses. As a result, all images from 1 to n have missing values at the same locations.
Before computing the (incomplete) covariance kernel for the area of our interest, we carry out some data preparation aimed at removing noise and autocorrelation across the time series. Specifically, we perform a concurrent investigation across the time series for each pixel. We identify autocorrelation within individual time series and, after checking for stationarity, we eliminate autocorrelation by fitting a moving average process to

each pixel. Ultimately, we use the values of the residuals for the covariance kernel estimation. This preprocessing is described in Appendix A.
After obtaining the spatial maps of residuals for each time instant, we can treat the observations of the 101×101 area across all time instants as independent and incorporate them into our problem framework. Consequently, the covariance function can be computed for each couple of pixels from n independent realizations of ground displacement.

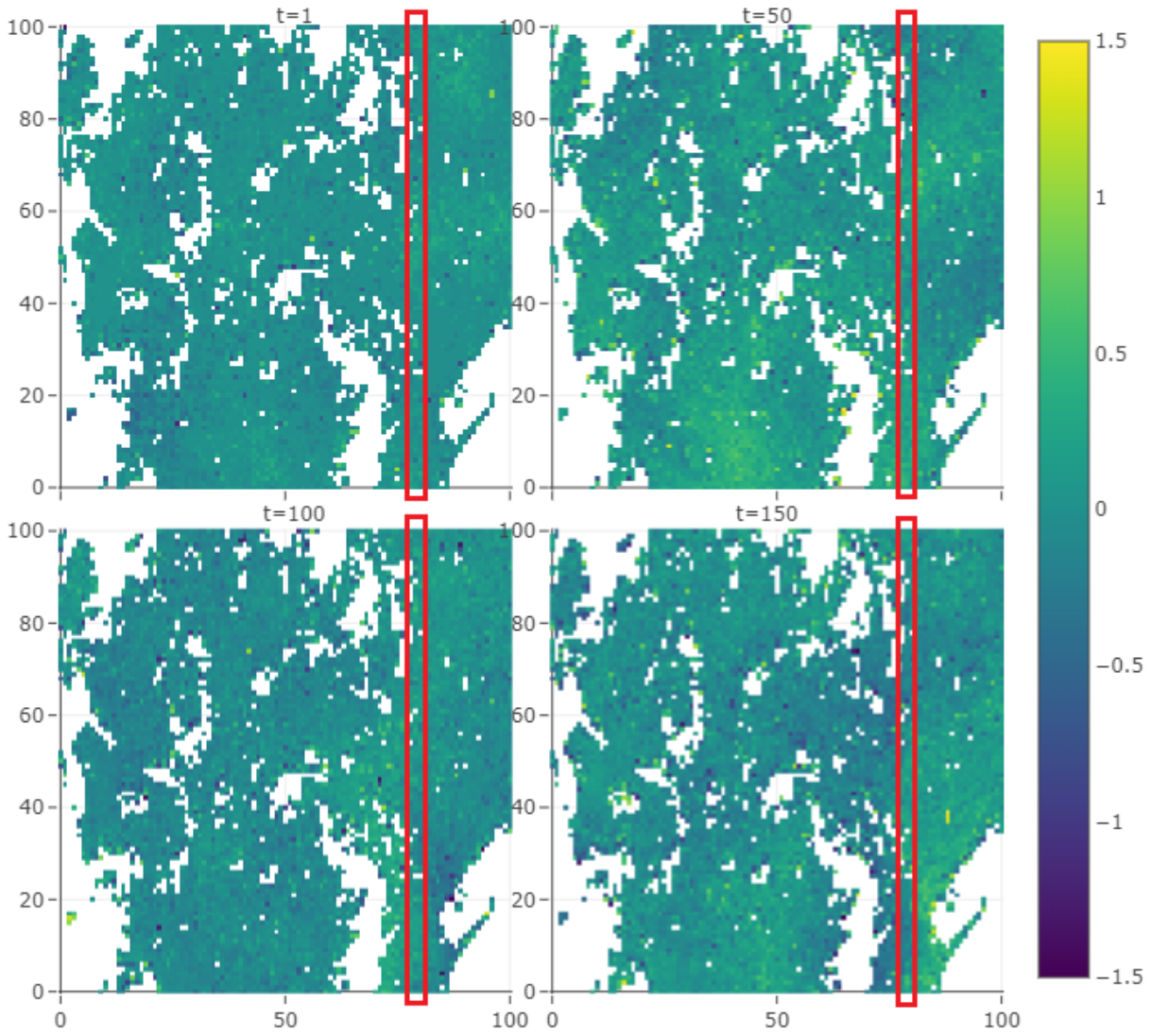

Figure 11: Ground displacement per pixel in the red subwindow, for time instants 1, 50, 100 and 150. We utilize the 80^{th} column of the spatial map for the analysis of Section 5.3 and the 79^{th}, the 80^{th} and the 81^{st} columns for the analyses in Section 5.4.

5.3. Reconstruction of one column

In order to lighten the computational load, we start testing our method on one single column of the red subwindow, the 80^{th} column, which has size of 101×1 pixels. The column has missing values at indices 12, 40, 66 and 76. As a result, the resolution of the discrete covariance kernel is $K \times K$, with $K = 101$, and it has missing values at rows and columns corresponding to 12, 40, 66 and 76.
Initially, we perform the rank selection, as described in Section 2.2, and find out that the best rank is 42; then, we fill the diagonal of the covariance kernel with the values of the 80^{th} column of the variance map after a Nadaraya–Watson smoothing is applied to it; subsequently, we look for solutions in the space of $K \times K$ matrices having rank equal to 42. Thereafter, in the interest of finding the best parameters for the reconstruction of the covariance kernel, we test the optimization problem in (16), over the same ranges of parameters choices

examined in the simulation study in Section 4.
For each combination of the two parameters' values, we compute RMSE_O, as defined in Equation (19). Clearly, this is the only one of the three types of errors considered in the simulation study that can be directly computed also in a real case study.

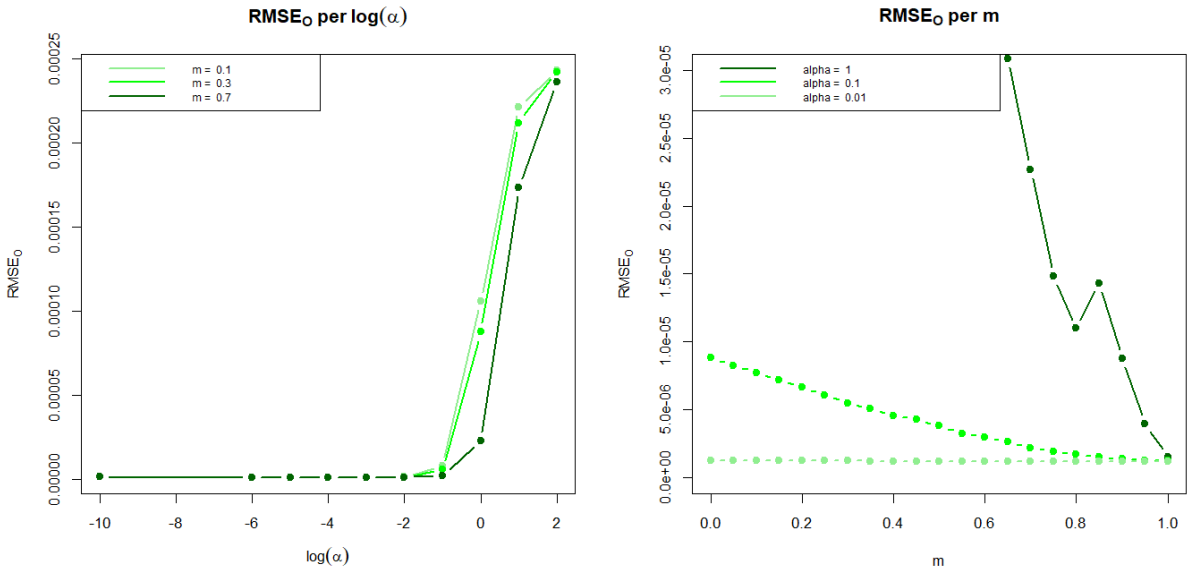

Figure 12: RMSE related to the observed part of the covariance kernel reconstruction for the 80^{th} column of the red subwindow.

Following the analyses conducted in the simulation study in Section 4, wherein the curves associated with RMSE and RMSE_M are accessible, we infer that, looking at the behaviour of RMSE_O across different values of the logarithm of α, the best value for the α parameter can be selected choosing the last value for which the error exhibits gradual or steady increase. As a consequence, observing the curves for several values of m (Figure 12 on the left), we set $\alpha = 0.01$.
On the other hand, the pattern of RMSE_O over the range of values for m does not provide insights for the selection of best parameter m. Indeed, for fixed values of α, each curve has a descending behaviour, which is understandable, as the observed cells do not need a Laplacian regularization term to achieve a satisfactory estimate, which would instead act as a form of penalty. Furthermore, this outcome is entirely anticipated and aligns seamlessly with the patterns observed in the simulated scenario in Section 4.
Consequently, unlike our approach to selecting a suitable value for α values based on the error across different m, we cannot discern an optimal value for m by examining the curve of RMSE_O over m values, which is the reason why we adopt a cross-validation approach, to determine the optimal method to select m in a covariance reconstruction problem.

5.3.1 Cross-validation for covariance reconstruction

Once the best α has been selected, it is necessary to find the best value for m.
Our aim is to investigate various spatial configurations of missing rows and columns within the matrix. Specifically, we seek to examine whether the distance between missing rows and columns in the matrix impacts its reconstruction and, more importantly, influences the selection of a global minimum for the parameter m of the Laplacian regularization term.
Indeed, we acknowledge the potential non-stationarity of the covariance matrix. Therefore, if we consider additional rows and columns as missing when they are proximate to existing missing rows and columns, the new spatial configuration is similar to the initial one. Neverthless, when evaluating the reconstruction error over several values for the parameter m, we take into consideration the fact that the Laplacian regularization incorporates information from neighboring cells to estimate the matrix. Therefore, removing a row or column adjacent to an existing missing one may potentially worsen the regularization by the Laplacian.
To perform the cross-validation in this setting, the idea is to subsequently consider the rows and columns near to the already missing rows and columns of the covariance kernel, and fictitiously treat them as missing. For each, we compute the error made in reconstructing the new missing rows and columns. The optimal m in cross-validation is the one corresponding to the lowest average reconstruction error.

Several configurations of additional missing values are considered, specifically, removing one index close to 12, 40, 66 or 76 at a time. In total, twelve configurations are concurrently considered as follows: for example, as for index 12, we consider the three subcases in which the covariance has missing rows and columns at indices

- 11,12,40,66,76;
- 10,12,40,66,76;
- 10,11,12,40,66,76.

We apply the same methodology for indices 40, 66 and 76, such that all the configurations in Table 1 are treated. The same configurations are represented in Figure 13.

Table 1: Removed indices for each case in cross-validation.

cases	**idx 12**	**idx 40**	**idx 66**	**idx 76**
-1	11	39	65	75
-2	10	38	64	74
-1,-2	10,11	38,39	64,65	74,75

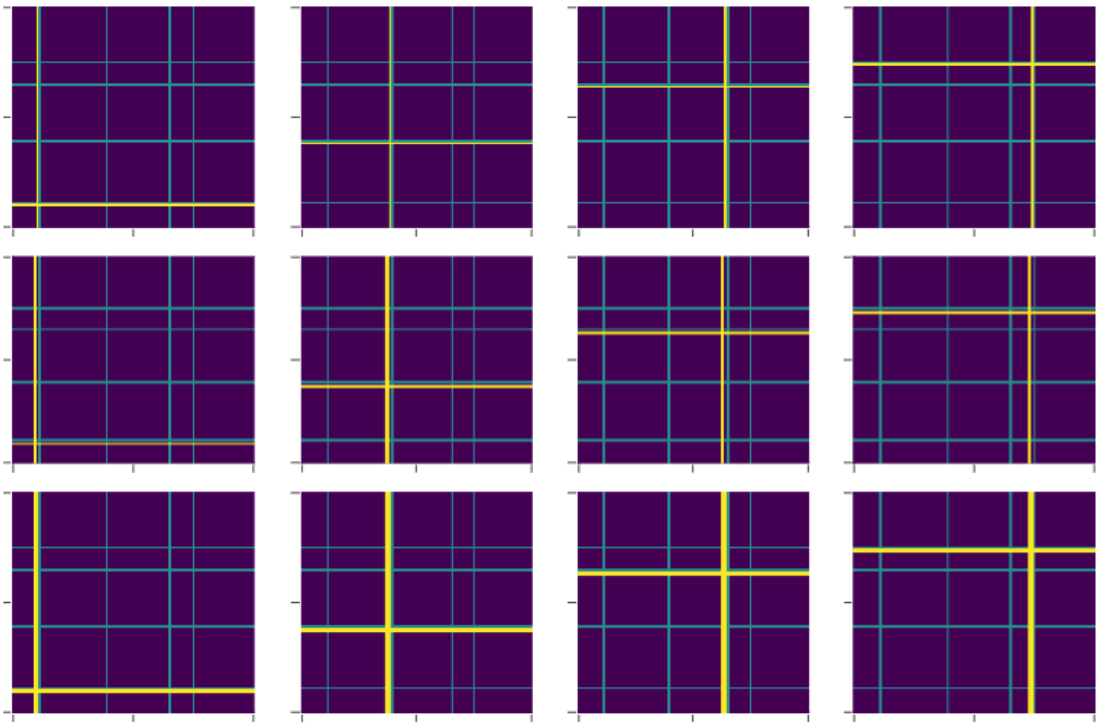

Figure 13: Cross-validation configurations are reported in the order of Table 1. The already missing indices are coloured in blue. The additional missing indices, used for computing RE_k, are coloured in yellow.

For each one of these configurations, we consider the reconstruction of the matrix - at fixed (best) rank equal to 42 - for all the possible values of m and for $\alpha = 0.01$. In turn, for each reconstruction, we compute the mean squared error related to the removed indices (except for the indices 12, 40, 66, 76 already missing).
The reconstruction error is formulated as follows:

$$\mathrm{RE}_k = \frac{\sum_{j\in I_k}\left[\sum_{i=1}^{101}(r_{ij}-\tilde{r}_{ij}^k)^2 + (r_{ji}-\tilde{r}_{ji}^k)^2\right] - (r_{jj}-\tilde{r}_{jj}^k)^2}{2|I_k|101 - |I_k|^2} \tag{22}$$

where I_k is the set of new missing indices related to the specific case (e.g. $I_1 = \{11\}$), $R = \{r_{ij}\}_{i,j=1}^{101}$ is the reconstructed matrix of the process having only 12, 40, 66 and 76 as missing indices, $\tilde{R}^k = \{\tilde{r}_{ij}^k\}_{i,j=1}^{101}$ is the reconstructed matrix for case k. The resulting curve of the errors is shown in Figure 14.
Examining the pattern of the reconstruction error for one index or two indices missing together, the values for an optimal m are 0.05, 0.1 or 0.15. As a result, we can observe that the cross-validation study leads to achieving

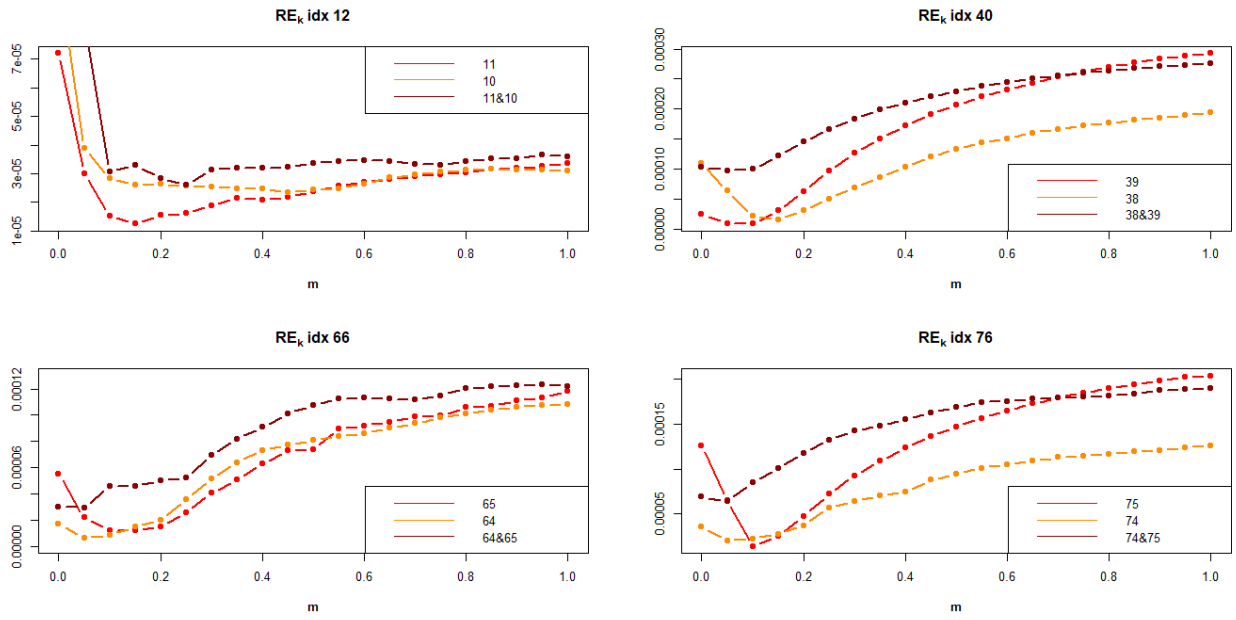

Figure 14: Cross-validation study for the covariance kernel reconstruction for the 80^{th} column of the red subwindow.

a global minimum for the reconstruction error across the range of values for m, similar to what occurs in the simulation study in Section 4 for the curve of the mean RMSE_O. This once again highlights the importance of introducing the parameter m in our problem resolution, while also demonstrating consistency with the outcomes in the simulation study, as concerns the behaviour of the RMSE_O curve.
Moreover, it is visible that each one of the four cases has different pattern. This is also a consequence of the fact that the covariance function related to the 80^{th} column clearly is not stationary (see Figure 15 on the left), hence the reconstruction of single rows/columns positioned at different locations in the matrix provides different reconstruction errors. However, upon individual examination, a clear minimum is attained in each case, enabling us to achieve our goal of finding a criterion for selecting the parameter m.
Another important insight is given by the position of the brown curve, corresponding to the error of two close indices, with respect to the red or the orange curves, which correspond to the error of single missing indices. In particular, the brown curve is usually above the other two, which is not surprising to us.
Indeed, the Laplacian regularization term of the optimization problem has smoothing power over rows and columns which are directly close. As a consequence, in general, it is expected that the error per cell of row and column 10 of the matrix, which are respectively positioned between two completely observed rows and columns, is lower than the error per cell for rows and columns 10 and 11 together. Hence, we believe that the cases with a single missing row/column are much more indicative of the behaviour of the reconstruction error of the missing rows and columns of the covariance matrix that we need to reconstruct.
In this sense, a different definition of the Laplacian matrix may help providing a better estimate, even in the setting where close indices are missing. A potential adjustment could be to define the adjacency matrix A as follows: $A = \{a_{ij}\}_{i,j=1}^{K}$, where $a_{ij} = 1$ if $1 \leq |i-j| \leq 2$. This implies that even indices which are not directly close (i.e. more than one index apart) would be considered as adjacent, meaning that, considering the previous example, the Laplacian regularization term would promote smoothness among pixels of the covariance matrix that are only one or two cells apart from each other.
Nevertheless, we have demonstrated that, performing a cross-validation study on the direct case study, one can make a thoughtful choice to set the parameter m in the optimization algorithm.

5.3.2 Covariance reconstruction

Once the parameters are chosen, we reconstruct the covariance with the best selection $\alpha = 0.01$ and $m = 0.1$ at fixed rank 42. According to this, the covariance reconstruction for the 80^{th} column is the one reported in Figure 15. The estimated rows and columns at indices 12, 40, 66 and 76 are not anymore identifiable in the new reconstructed matrix. This is a good sign from a visul point of view, since we expect that close pixels vary in similar ways across the matrix, so we would not expect to spot a substantial difference between rows - or columns - which are close in a covariance matrix.

Additionally, it seems like also the part of the covariance kernel related to observed cells is faithfully reconstructed. This means that our method is combining a good reconstruction of the missing cells with a loyal preservation of the observed cells.

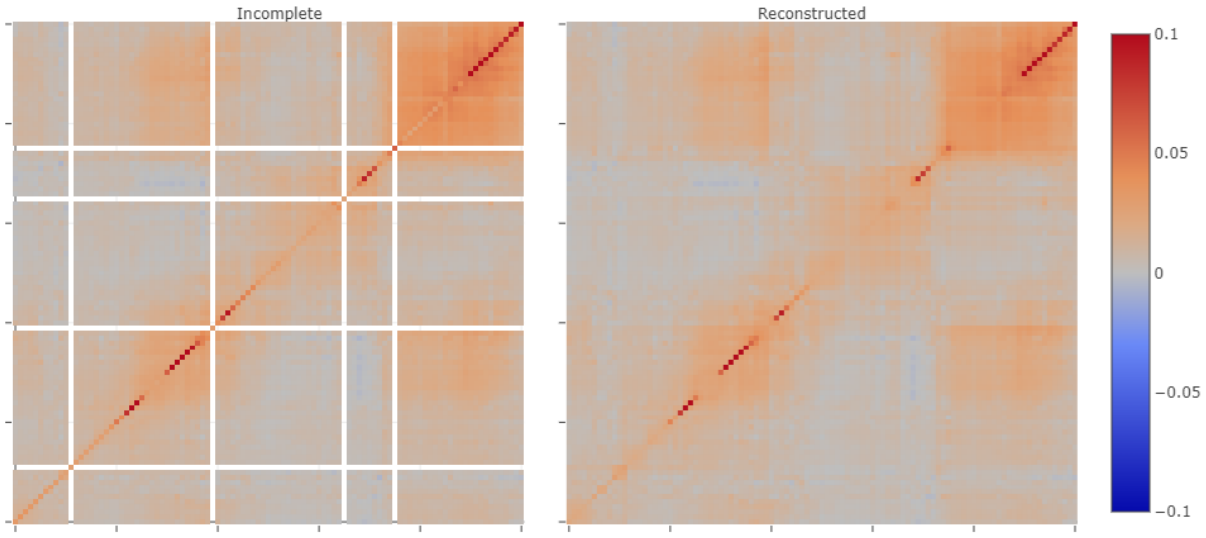

Figure 15: Reconstruction of the covariance kernel for the 80^{th} column of the red subwindow ($\alpha = 0.01$, $m = 0.1$).

Mean cell value (before reconstruction)	**Mean cell value (after reconstruction)**	**RMSE$_O$**
0.01163924	0.01152875	1.238235×10^{-6}

$\sqrt{(\mathbf{RE}_k)}$	**idx 12**	**idx 40**	**idx 66**	**idx 76**
-1	0.003907735	0.002932954	0.003521353	0.003706899
-2	0.005319602	0.004636186	0.00295375	0.004713608

Table 2: Quantitative performance of the covariance reconstruction algorithm over the 80^{th} row of the red subwindow (for $\alpha = 0.01$ and $m = 0.1$).

To also quantitatively asses the performance of the method, we examine the values of RMSE$_O$ and $\sqrt{RE_k}$ for $\alpha = 0.01$ and $m = 0.1$. In particular, we compute the square root of the reconstruction error to facilitate comparison with the order of magnitude of the mean value per cell of the covariance matrix, before and after the reconstruction. The results are reported in Table 2.
Specifically, we notice that the mean value per cell slightly changes after the reconstruction. Moreover, RMSE$_O$ is significantly lower than the mean value per cell, indicating that the estimation of the covariance matrix at the observed cells is not compromised by selecting as value for the m parameter the one that better estimates the missing part of the matrix.
As concerns the squared reconstruction error, we notice that the current reconstruction is satisfactory, yet there is room for improvement. Indeed, it is important to emphasize that, while matrix completion algorithms strive to minimize the error between observed and reconstructed matrix, achieving a recontruction error of exactly zero is typically impractical in real-world scenarios. The observed data provide only partial information about the complete matrix and this incomplete information inherently leads to some degree of uncertainty and error in the reconstructed matrix.

5.4. Proof of concept: reconstruction of a two-dimensional surface

At this stage of our analysis, we move our attention to two-dimensional functional data, in order to align with the specific requirements of our case study. To accomplish this, we consider the transformation of each two-dimensional datum into a one-dimensional form. This choice is strongly justified by the significant computational

load associated with handling two-dimensional data in this type of problem. Indeed, in this case, the covariance to be estimated to reconstruct data would be a four-dimensional tensor, given that each element of the covariance tensor denotes the covariance value between two pixels, and each pixel is identified by two indices. Accordingly, the Laplacian regularization term in the optimization problem also would include a four-dimensional Laplacian tensor, accounting for the four-dimensional proximity. Therefore, the reconstruction proceduce would entail managing high-dimensional data, which would increase the computational burden and complicate the attainment of interpretable results in small time.
To move to the two-dimensional case, we consider the reconstruction of the covariance of three successive columns - $79^{th}, 80^{th}$ and 81^{th} - of the size of 101×3 pixels.
To maintain continuity across them, we "flatten" the three columns creating a one-dimensional vector of length 303×1, such that the 1^{st} cell of the vector corresponds to the 1^{st} cell of the first column, the 101^{st} cell to the 101^{st} cell of the first column, the 102^{nd} cell to the 101^{st} cell of the second column, the 103^{rd} to the 99^{th} of the second column, and so on, up to having the 101^{st} cell of the third column as 303^{rd} cell of the new vector. A more intuitive representation of the flattening procedure is reported in Figure 16.

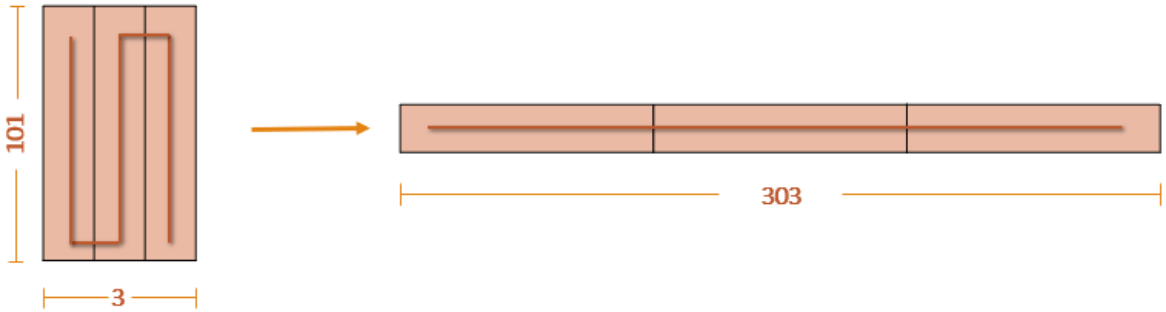

Figure 16: From 2D to 1D.

The main difference with respect to the previous case of covariance recontruction for one column is that in the upper (or lower) diagonal a transverse dependance pattern is visible, and it a direct consequence of the way in which we flattened the three columns.
Similarly to the scenario of reconstructing a single column, in this case as well, the diagonal is filled taking the values of the $79^{th}, 80^{th}$ and 81^{th} columns of the smoothed variance map. Then, we set $\alpha = 0.01$ and $m = 0.1$, then the rank is set to 130. The reconstruction is reported in Figure 17.
The relation between entire close rows and columns in the covariance is an information that would be lost by considering the reconstruction of one column at a time. Indeed, in that case, only the diagonal blocks would be available for data reconstruction.
Nevertheless, not all the dependance is captured. In fact, when stacking columns vertically, although continuity is maintained, we lose information regarding the dependence between adjacent row elements. This lack of consideration emerges because the Laplacian matrix only addresses two-dimensional proximity between cells in the covariance matrix. In this context, a potential expansion of our method would entail redefining the Laplacian term to take into consideration the four-dimensional proximity of elements within the covariance tensor.
Finally, we achieve our objective of estimating the value of ground displacement for missing pixels within the red subwindow (as identified in Figure 1), whose reconstruction is reported in Figure 18 for specific time instants. This is accomplished by considering five columns at a time and maintaining column continuity across them. In total, we estimate 20 covariances of size 505×505 and one additional covariance of size 101×101 to account for the last column, as the red window measures 101×101. The reconstruction is performed by first determining the optimal rank for each covariance kernel, selected through the rank minimization problem described in Section 2.2. Moreover, we set $\alpha = 0.01$ and $m = 0.1$. Afterwards, we apply the method of Kraus [2015] for functional completion, that we briefly describe in Section 2.3. For each group of columns within the red subwindow, we utilize the covariance kernel estimator of our method and the Nadaraya-Watson smoothing method to estimate the covariance operator and mean function, respectively.
The reconstruction is not entirely interpretable at this stage. While we can observe a certain level of smoothing between missing and observed pixels, we are unable to assess the performance comprehensively. However, it is evident that the reconstruction is heavily influenced by the estimated mean value at missing locations, which is still to be entirely estimated at missing pixels, since no observation is available. In particular, this task becomes challenging when there are many missing pixels in close proximity, as smoothing techniques for mean estimation become less efficient in such cases.

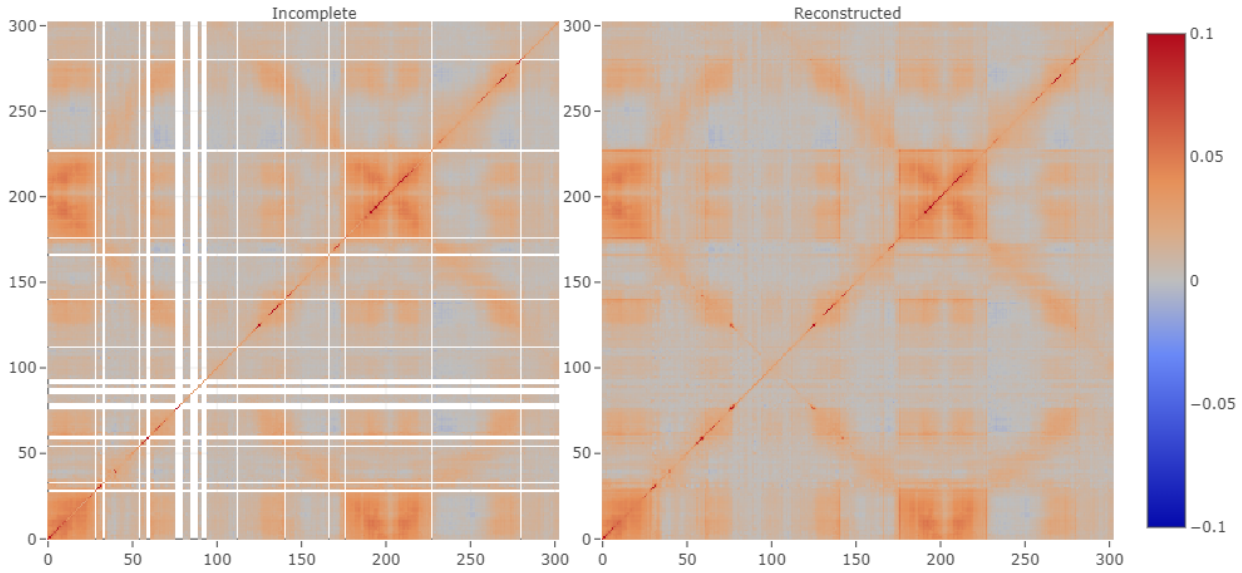

Figure 17: Reconstruction of the covariance kernel for 79^{th}, 80^{th} and 81^{st} column of the red subwindow ($\alpha = 0.01$, $m = 0.1$).

Furthermore, we observe that several potential outliers are located near the boundaries of the red subwindow. This observation suggests that the algorithm performs more effectively when the covariance information of most surrounding data points is available.
In this context, future applications of our method could explore the use of sliding windows for data reconstruction at missing pixels. This approach would involve estimating each pixel using a larger portion of the surrounding available information. However, it is essential to consider that this operation would significantly increase the computational load.

6. Conclusion and further developments

Having access to complete DInSAR Interferograms observations is essential for enhancing our understanding of natural disasters, assessing their risks and implementing effective measures to prevent and alleviate their effect on communities and infrastructures. In fact, for instance, the identification of patterns of deformation can allow the establishment of early warning systems to mitigate the impact of catastrophic events. Moreover, continuous monitoring with complete interferograms enables long-term assessment of hazard risks, by tracking changes in ground deformation over years or decades. In this regard, the significance of this thesis lies in its ability to effectively reconstruct incomplete data of ground displacement, presenting good performance in the estimation of the covariance kernel associated to functional spatial data in time.
The challenge of our problem arose from the consistent presence of missing values in DInSAR data across all time points, all at the same locations on the map. This led to a limitation in the application of many statistical methods already developed in FDA for partially observed functional data.
In order to estimate the ground displacement information at missing pixels, we developed a covariance reconstruction algorithm that aimed at estimating the covariance kernel empirical estimator at his missing locations. To accomplish this, firstly, we estimated the empirical covariance with the method of Kraus [2015], resulting in a covariance matrix where missing cells represented the covariance between pairs of missing pixels. Then, we applied the low-rank matrix completion problem defined in the work of Descary and Panaretos [2018], intended to assess the best rank for the resulting matrix. Ultimately, we focused on efficiently populating the fixed-rank covariance kernel at missing locations. To achieve this, we utilized a fixed-rank minimization approach. Its objective was twofold: to effectively preserve the information from observed cells while efficiently estimating values at missing locations by leveraging the information contained within the observed cells. This was accomplished by incorporating a Laplacian regularization term in the objective funtion of the optimization problem, promoting smoothness among neighboring cells. Moreover, a weighting system embedded within the Laplacian regularization term enabled a weighted utilization of the information carried by the observed and missing cells. The efficacy of our approach was validated through a simulation study and a case study application.
Specifically, the simulation study showed that even in a context where the true covariance function deviated from the assumptions of the formal problem definition, particularly in terms of analyticity and finite rank,

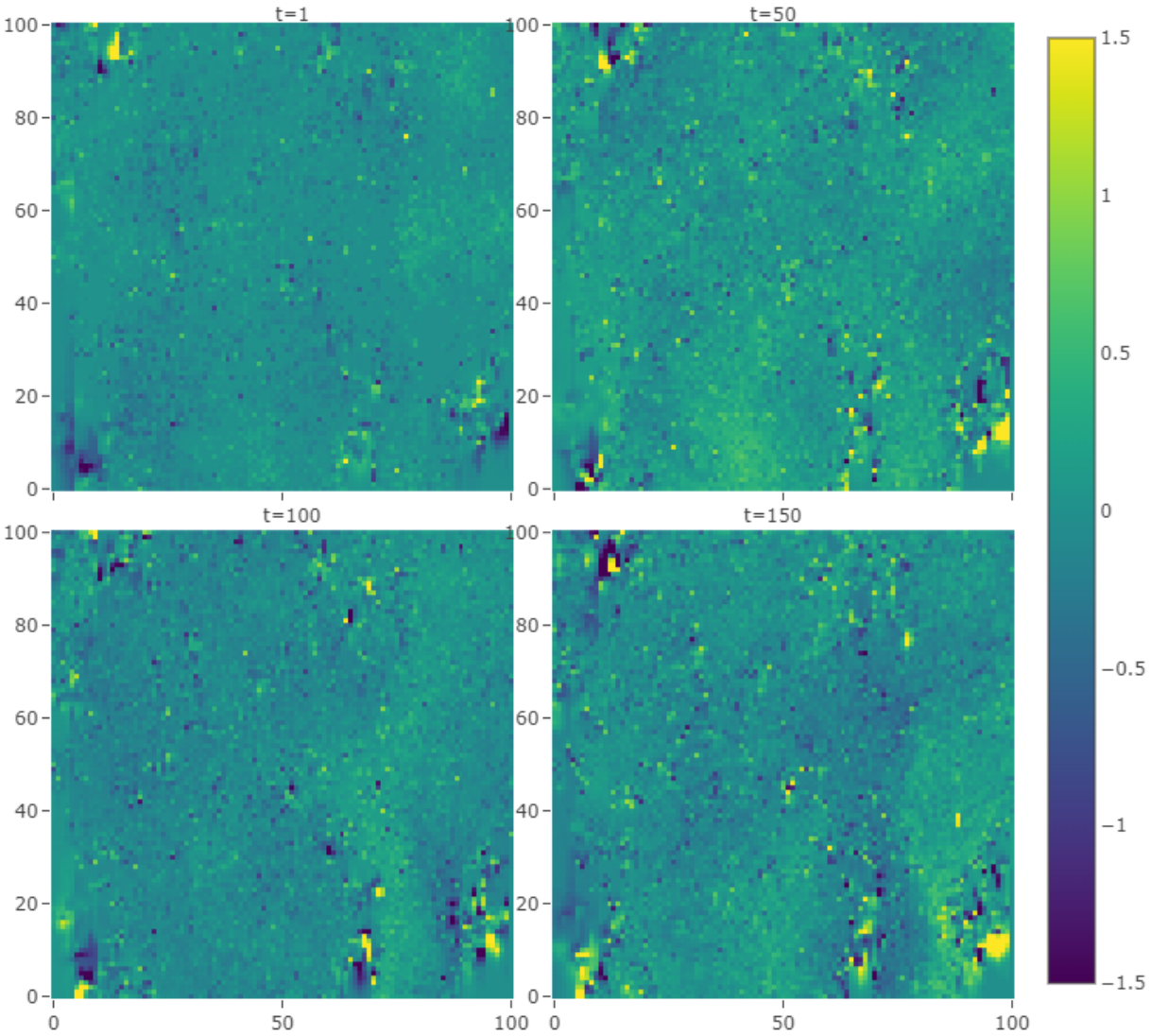

Figure 18: Reconstructed ground displacement per pixel in the red subwindow, for time instants 1, 50, 100 and 150.

good reconstructions were attainable. Concurrently, in the case study, where stationarity was not assumed, our method still exhibited strong performance, as confirmed by a cross-validation study. Optimal performance of our approach was achieved applying specific selection criteria of regularization parameter and weight parameter. Neverthless, we encountered some limitations. Due to computational constraints, the optimization problem for estimating the covariance matrix was addressed using a fixed rank approach. However, employing a combination of low-rank matrix completion and Laplacian regularization in the objective function, alongside the preservation of observed cells, might have yielded improved results in the optimization process. In fact, even when focusing only on small areas of interest, the low-rank matrix estimation algorithm proved to be highly time-intensive.
Furthermore, another constraint of the method for reconstructing two-dimensional surfaces was the transformation of our images into one-dimensional vectors by maintaining vertical continuity. This caused the preservation of continuity across columns, but the loss of continuity across rows, resulting in a significant loss of information regarding the dependence of pixels that are close in rows. As a potential future development, converting the Laplacian matrix into a four-dimensional tensor could lead to significantly more accurate and effective results, by consolidating additional information related to the two-dimensional continuity of a surface.
Another possible extension involves investigating among several definitions of the Laplacian matrix by testing different definitions of the adjacency matrix. In particular, it could be interesting to compare diverse configurations for the adjacency matrix, considering adjacency not only for cells that are one cell apart in the matrix but also for those further apart. Indeed, if two close rows are missing in the covariance matrix, it might be beneficial to gather information from more distant indices, although this may not always be the case. For this reason, testing various configurations for the Laplacian matrix and validating the results through a cross-validation

procedure could enhance the final reconstruction.
Additionally, further analyses could be employed to compare the outcomes of different data reconstruction methods, utilizing the covariance operator and the mean function estimated according to our study. However, we did not conduct cross-validation to assess the performance of our algorithm with the functional data from the area of interest. Therefore, the trustworthiness of our functional reconstruction relies on the demonstrated effectiveness of the method proposed by Kraus [2015]. Further extensions would involve, for instance, to test the principal scores reconstruction technique proposed in the same work by Kraus [2015] or to explore the functional data reconstruction based on reconstruction operators proposed by the work of Kneip and Liebl [2020]. Moreover, several alternative techniques could be employed to estimate the mean function at locations of missing pixels in the area, as this could also impact the outcome of the reconstruction.
Eventually, a critical decision to be made regards determining the optimal size of the area of interest, required to adequately capture all necessary dependencies and achieve efficient estimation at each location. If the area of interest is too small, there may be insufficient information available, while if it is too large, unnecessary information may be included. As a consequence, it is preferable to reduce the computational load by optimizing the size of the area, to strike a balance between information adequacy and computational efficiency.
The context in which this thesis was developed has presented numerous challenges, as well as interesting opportunities for reflection and improvement. DInSAR data represent valuable assets in forecasting catastrophic events, and their complete observation holds critical significance, especially given the progressively challenging task of predicting the behavior of the Earth. Therefore, it is increasingly important to develop methods that keep pace with the changing world and are capable of studying it more effectively. We hope that this work contributes to this endeavor and serves as a catalyst for further analyses of this kind.

References

[1] P. Berardino, G. Fornaro, R. Lanari, and E. Sansosti. A new algorithm for surface deformation monitoring based on small baseline differential sar interferograms. *IEEE Transactions on Geoscience and Remote Sensing*, 40(11):2375–2383, 2002.

[2] M. S. Bernardi, P. C. Africa, C. de Falco, and et al. On the use of interferometric synthetic aperture radar data for monitoring and forecasting natural hazards. *Mathematical Geosciences*, 53(5):1781–1812, 2021.

[3] Francesco Casu, Mariarosaria Manzo, and Riccardo Lanari. A quantitative assessment of the sbas algorithm performance for surface deformation retrieval from dinsar data. *Remote Sensing of Environment*, 102(3-4):195–210, 2006.

[4] J.J.F. Commandeur and S.J. Koopman. *Introduction to State Space Time Series Analysis*. Oxford University Press, 2007.

[5] Marie-Eve Descary and Victor M Panaretos. Recovering covariance from functional fragments. *Biometrika*, 105(4):883–896, 2018.

[6] Oleksandr Gromenko, Piotr Kokoszka, and Sheila Sojkain. Evaluation of the cooling trend in the ionosphere using functional regression with incomplete curves. *Annals of Statistics*, 11(2):898–918, 2017.

[7] Lajos Horváth and Piotr Kokoszka. *Inference for Functional Data with Applications*. Springer, 2012.

[8] A. B. Kashlak, J. A. D. Aston, and R. Nickl. Inference on covariance operators via concentration inequalities: k-sample tests, classification, and clustering via rademacher complexities. *Sankhya A*, 81(2):214–243, 2019.

[9] Alois Kneip and Dominik Liebl. On the optimal reconstruction of partially observed functional data. *Annals of Statistics*, 48(3):1692–1717, 2020.

[10] Daniel Kraus. Components and completion of partially observed functional data. *Journal of the Royal Statistical Society*, 77(4):777–801, 2015.

[11] Johannes Kraus and Andrea Stefanucci. Ridge reconstruction of partially observed functional data is asymptotically optimal. *Statistics & Probability Letters*, 165:108813, 2020.

[12] T. Maunu. First-order algorithms for optimization over graph laplacians. In *2023 International Conference on Sampling Theory and Applications (SampTA)*, pages 1–11, New Haven, CT, USA, 2023. IEEE.

[13] J. Pang and G. Cheung. Graph laplacian regularization for image denoising: Analysis in the continuous domain. *IEEE Transactions on Image Processing*, 26(4):1770–1785, April 2017.

[14] Davide Pigoli, John A. D. Aston, Ian L. Dryden, and Piercesare Secchi. Distances and inference for covariance operators. *Biometrika*, 101(2):409–422, 2014.

[15] James O Ramsay and Bernard W Silverman. *Functional Data Analysis*. Springer, 2005.

[16] Andrea Stefanucci, Laura Maria Sangalli, and Pierpaolo Brutti. Pca-based discrimination of partially observed functional data, with an application to aneurisk65 dataset. *Wiley Interdisciplinary Reviews: Computational Statistics*, 72(3):246–264, 2018.

[17] Chang Tang, Hua Zhou, Xiao Zheng, Yanming Zhang, and Xiaofeng Sha. Dual laplacian regularized matrix completion for microrna-disease associations prediction. *RNA Biology*, 16(5):601–611, 2019.

A. Appendix A - Data preprocessing

As mentioned in Section 5, the temporal series of ground displacement to be reconstructed necessitates a preprocessing phase before entering our problem setting. In particular, we must ensure that the temporal observations are independent, since their temporal nature may raise concerns.
Our time observations span from 27/03/2015 to 22/02/2023. We begin by examining the time intervals between each temporal map, and discover that the temporal interval varies across observations. In particular, we identify three distinct time ranges. In the first range, from 27/03/2015 to 27/09/2016, the observations are spaced 12 days apart. In the second range, from 27/09/2016 to 30/12/2021, there is a 12-day interval between observations. Finally, in the third range, from 30/12/2021 to 22/02/2023, the observations once again have a 12-day interval between them. Since we need equally spaced observations to account for temporal dependence, we begin our analysis by focusing on the second range of dates, as it comprises the largest number of observations, specifically 316 over 391.
First, we assess the presence of autocorrelation within the time series. To accomplish this, we employ the Durbin-Watson test, a statistical method utilized for identifying autocorrelation in the residual errors of a regression model. In our particular case, the following generic model is considered for the time series

$$y_t = \delta + e_t \tag{23}$$

for $t = 1, ..., T$, where $T = 316$, δ is the mean value of the time series, e_t for $t = 1, ...T$ are the residual errors. The null hypothesis suggests the existence of autocorrelation, whereas the alternative hypothesis suggests the absence of autocorrelation. The Durbin-Watson test statistic is formulated as follows

$$DW = \frac{\sum_{t=2}^{T}(e_t + e_{t-1})^2}{\sum_{t=1}^{T} e_t^2} \tag{24}$$

such that $DW \in (0, 4)$. If $0 < DW \ll 2$, then positive autocorrelation is detected, instead, if $2 \ll DW < 4$, the series is negatively autocorrelated. Finally, if $DW \approx 2$, then no autocorrelation is evident.
We perform the Durbin-Watson test for each observed pixel of the red subwindow. The outcome is that each test statistic (for every pixel) is very close to zero. This indicates that the time series exhibits negative autocorrelation, so it is necessaty to remove autocorrelation to attain independence among time instants. The result of Durbin-Watson test per pixel on the temporal series of ground displacement is shown in Figure 19.
To mitigate autocorrelation, one approach is to fit an ARMA (Autoregressive Moving Average) model to the time series. However, before proceeding with the model fitting, it is essential to check the stationarity of the time series for each pixel. This is accomplished using another statistical test, namely the Augmented Dickey-Fuller test. The test assesses the stationarity of the time series, with the null hypothesis being non-stationarity and the alternative hypothesis being stationarity. Upon conducting the test for each pixel, we generally obtain low p-values, leading us to reject with evidence the null hypothesis and assume stationarity for each time series.
At this stage, we must determine the appropriate order for fitting the ARMA model, reminding that our focus is still on observations from the second range of dates. To inform this decision, we examine the plots of the autocorrelation function (ACF) and partial autocorrelation function (PACF) for the time series of each pixel. Indeed, following the Box–Jenkins method (Commandeur and Koopman [2007] [4]), it is possible to select candidates for the order of ARIMA models by examining ACF and PACF plots. Specifically, if the ACF curve slowly decays while the PACF abruptly becomes zero after p lags, then an ARMA(p,0), i.e., AR(p), may be suitable for the problem. Conversely, if the PACF plot exponentially decays while the ACF cuts off after q lags, then an ARMA(0,q), i.e., MA(q), may be more appropriate.

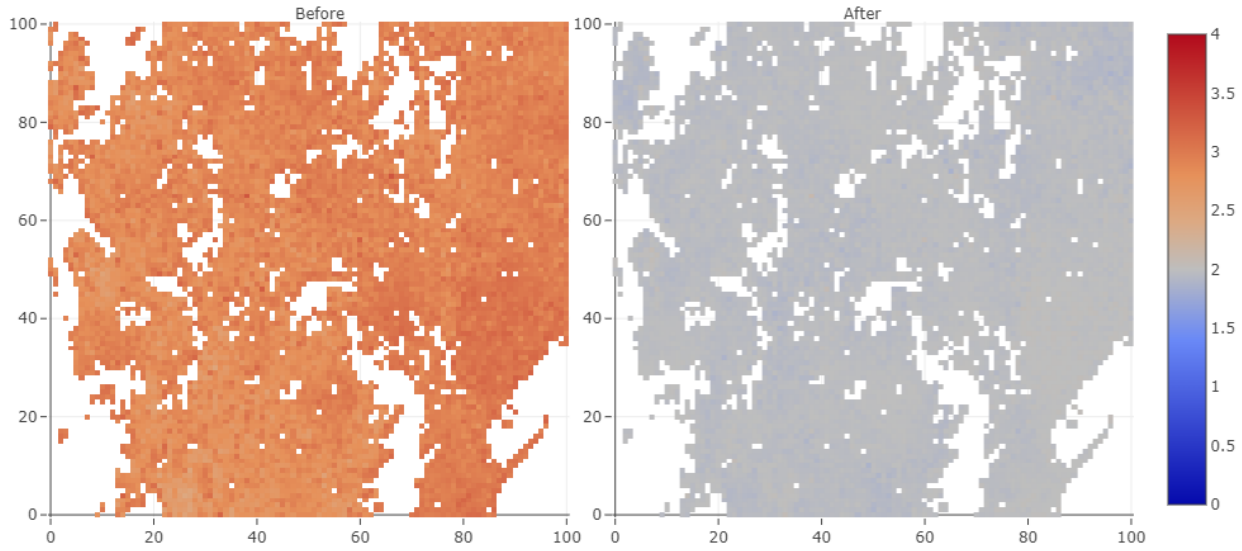

Figure 19: The Durbin-Watson test statistic is reported for each pixel considering the initial temporal series (on the left) and the residuals after fitting the MA(2) model (on the right).

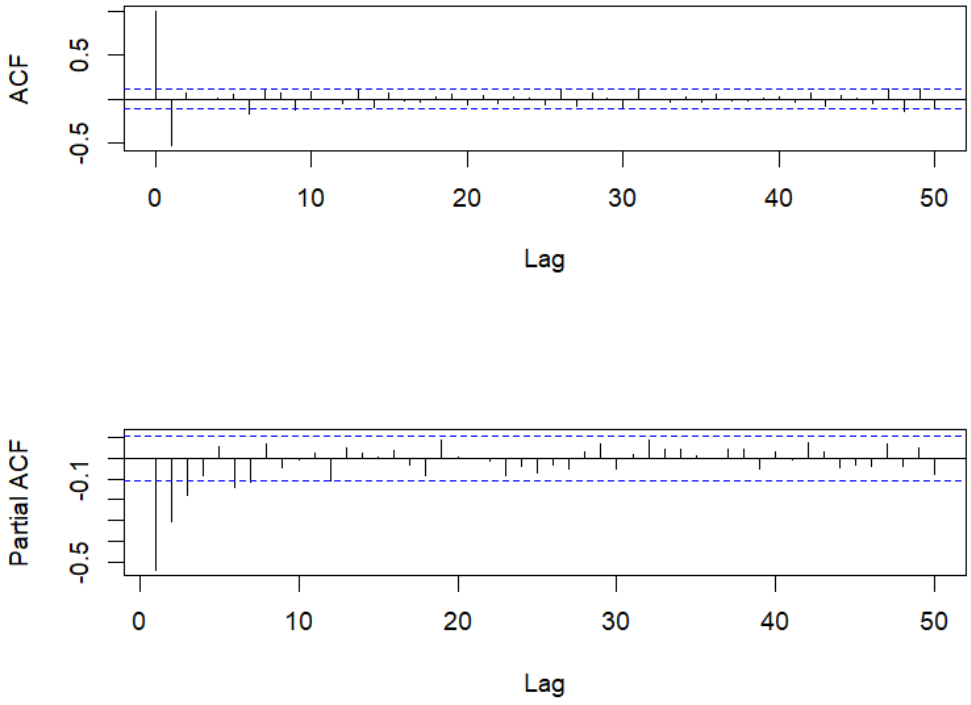

Figure 20: Autocorrelation and partial autocorrelation functions for pixel (50,1) of the red subwindow, considering time instants from the second range of dates.

The plots of ACF and PACF for one specific pixel are reported in Figure 20.
We observe that, based on the criterion outlined earlier, the optimal solution would be to fit an MA(2) model to the time series. However, we also compare the results of model fitting for AR(2), MA(3), and AR(3). Despite this, the MA(2) model emerges as the best option in terms of Akaike Information Criterion (AIC) prediction error. We illustrate the behavior of the time series compared with the fitted MA(2) process in Figure 21.
After fitting the MA(2) model to the time series, we conduct the Durbin-Watson test for autocorrelation once again, this time considering the residuals of the fitted MA(2) model. Encouragingly, we obtain satisfactory

results, as the Durbin-Watson statistic for each pixel is approximately 2, indicating that autocorrelation has been effectively mitigated.
We now aim to extend this finding to the other two date ranges available to us, which are spaced 12 days apart. As anticipated, autocorrelation is evident in these cases as well. Upon plotting the ACF and PACF and fitting an ARMA model, we find that an ARMA model of order 1 is the most suitable. Specifically, the AR(1) model outperforms the MA(1) model in this scenario. However, it is worth noting that this conclusion may be influenced by the limited number of time points available. Therefore, we place greater trust in the previous result obtained over the larger range of dates and decide to fit a MA(2) model for the range of dates spaced by 6 days and a MA(1) model for the ranges of dates spaced by 12 days.
As a result, we have successfully mitigated autocorrelation and achieved independence for the temporal time series by fitting a MA model to our data. Consequently, we can now solely focus on the residuals of the MA fitted values for our analysis. These residuals are independent and yet retain all the spatial dependence necessary for the covariance kernel reconstruction, which is the primary objective of this thesis.

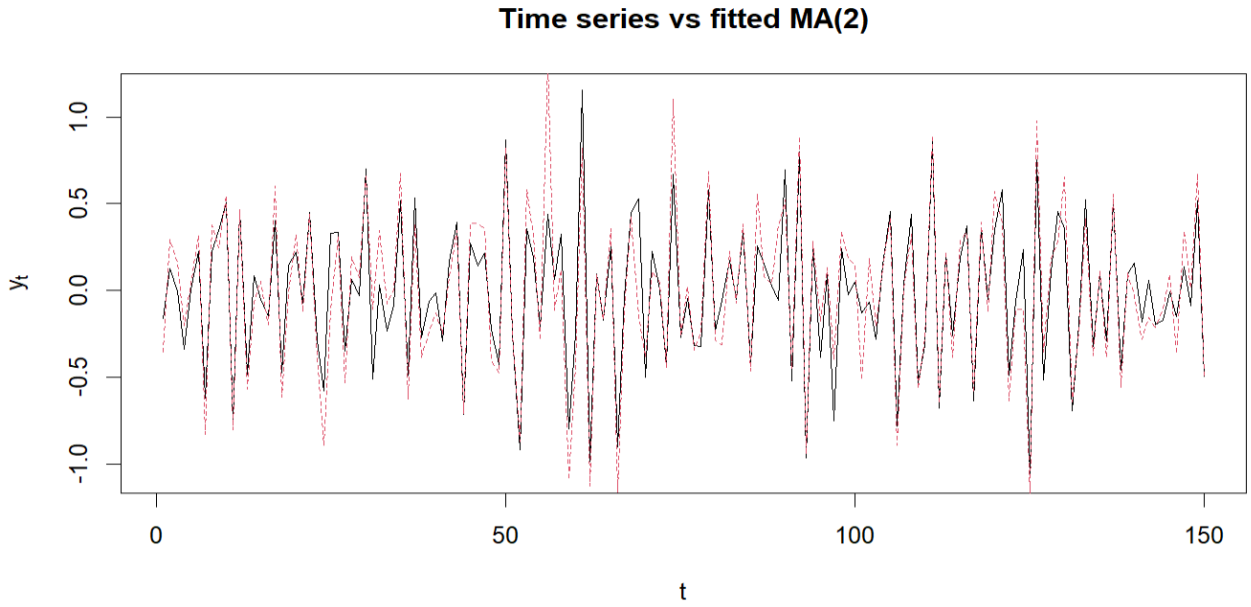

Figure 21: We report the curve of the temporal series of ground displacement for pixel (50,1) in black and its estimation through MA(2) model fitting in a dashed red line, evaluated ove a subinterval of time instants of the second range of dates.

B. Appendix B - Grid search results

We here report the results of the grid search for the hyperparameters α and m conducted in the simulation study in Section 4.

Table 3: Mean of total RMSE over 50 simulations.

$\mathbf{m \backslash \alpha}$	$\mathbf{0}$	$\mathbf{10^{-6}}$	$\mathbf{10^{-5}}$	$\mathbf{10^{-4}}$	$\mathbf{10^{-3}}$	$\mathbf{10^{-2}}$	$\mathbf{10^{-1}}$	$\mathbf{1}$	$\mathbf{10^{1}}$	$\mathbf{10^{2}}$
0	0.024676	0.024584	0.023735	0.014048	0.01305	0.013531	0.018152	0.048576	0.071348	0.074924
0.05	0.024676	0.024583	0.023741	0.014469	0.012874	0.012992	0.016028	0.048393	0.071352	0.074929
0.1	0.024676	0.024583	0.023746	0.015039	0.012634	**0.012431**	0.014837	0.047988	0.07129	0.074919
0.15	0.024676	0.024584	0.02375	0.015663	0.012715	0.012437	0.014948	0.047495	0.071127	0.074897
0.2	0.024676	0.024584	0.023758	0.01636	0.013716	0.013603	0.015793	0.046713	0.070917	0.074868
0.25	0.024676	0.024583	0.023763	0.016973	0.014864	0.014869	0.016819	0.045318	0.07065	0.074826
0.3	0.024676	0.024583	0.023769	0.017595	0.01605	0.016001	0.01763	0.043918	0.070306	0.074764
0.35	0.024676	0.024584	0.023776	0.018117	0.017011	0.016961	0.018253	0.042285	0.069818	0.074708
0.4	0.024676	0.024584	0.023779	0.018591	0.017743	0.017723	0.018876	0.040315	0.069242	0.074628
0.45	0.024676	0.024583	0.023787	0.018998	0.018356	0.018353	0.019326	0.038191	0.068362	0.074569
0.5	0.024676	0.024584	0.023793	0.019364	0.018876	0.018886	0.019746	0.036123	0.067486	0.074424
0.55	0.024676	0.024584	0.023799	0.019682	0.019326	0.019338	0.020062	0.033922	0.066495	0.074296
0.6	0.024676	0.024584	0.023801	0.019961	0.019718	0.019721	0.0203	0.032263	0.06586	0.074193
0.65	0.024676	0.024584	0.023807	0.020212	0.02006	0.020054	0.020531	0.030438	0.064669	0.073984
0.7	0.024676	0.024584	0.023812	0.020435	0.020357	0.020352	0.020625	0.029323	0.063107	0.073784
0.75	0.024676	0.024584	0.023817	0.020646	0.020619	0.020611	0.020787	0.027394	0.06084	0.073645
0.8	0.024676	0.024584	0.023822	0.020834	0.02085	0.020839	0.020938	0.025746	0.057907	0.073336
0.85	0.024676	0.024584	0.023827	0.021011	0.021054	0.021047	0.0211	0.024189	0.051965	0.072546
0.9	0.024676	0.024584	0.023831	0.021174	0.021239	0.021234	0.021236	0.02383	0.041907	0.071327
0.95	0.024676	0.024584	0.023836	0.021323	0.021405	0.021404	0.021403	0.022131	0.032075	0.065818
1	0.024676	0.024584	0.023843	0.021467	0.021557	0.021559	0.02157	0.021694	0.023374	0.029448

Table 4: Median of total RMSE over 50 simulations.

$\mathbf{m \backslash \alpha}$	$\mathbf{0}$	$\mathbf{10^{-6}}$	$\mathbf{10^{-5}}$	$\mathbf{10^{-4}}$	$\mathbf{10^{-3}}$	$\mathbf{10^{-2}}$	$\mathbf{10^{-1}}$	$\mathbf{1}$	$\mathbf{10^{1}}$	$\mathbf{10^{2}}$
0	0.023509	0.023412	0.022682	0.012892	0.011433	0.011844	0.017246	0.049266	0.071466	0.074938
0.05	0.023509	0.023412	0.022685	0.013383	0.011252	0.011205	0.015366	0.049196	0.071459	0.074944
0.1	0.023509	0.023412	0.022689	0.013877	0.011039	**0.010881**	0.014695	0.048802	0.071417	0.074925
0.15	0.023509	0.023412	0.022692	0.014453	0.011077	0.011362	0.014561	0.048608	0.071188	0.0749
0.2	0.023509	0.023412	0.022692	0.015173	0.012168	0.0126	0.015569	0.047741	0.071003	0.074887
0.25	0.023509	0.023407	0.022696	0.015763	0.013699	0.013801	0.016284	0.045766	0.070747	0.074827
0.3	0.023509	0.023407	0.022699	0.016228	0.014835	0.01499	0.016591	0.04431	0.070379	0.074751
0.35	0.023509	0.023407	0.022702	0.016878	0.015833	0.015941	0.017026	0.042538	0.069792	0.074706
0.4	0.023509	0.023407	0.022706	0.017521	0.016597	0.016711	0.017374	0.041027	0.069316	0.074638
0.45	0.023509	0.023407	0.022709	0.0179	0.01722	0.017382	0.018107	0.038415	0.068665	0.074568
0.5	0.023509	0.023407	0.022713	0.018311	0.017744	0.017928	0.018804	0.035505	0.067612	0.074452
0.55	0.023509	0.023407	0.022716	0.018687	0.018209	0.018397	0.018968	0.03392	0.066724	0.074346
0.6	0.023509	0.023407	0.022719	0.018997	0.01859	0.018765	0.019309	0.032602	0.065377	0.074258
0.65	0.023509	0.023408	0.022723	0.01927	0.018921	0.019055	0.019426	0.029824	0.065099	0.074025
0.7	0.023509	0.023408	0.022726	0.019483	0.019213	0.019306	0.019688	0.02924	0.063912	0.073858
0.75	0.023509	0.023408	0.022729	0.019643	0.019459	0.019576	0.019838	0.026905	0.062171	0.073666
0.8	0.023509	0.023408	0.022732	0.0198	0.019672	0.019775	0.020032	0.02474	0.059314	0.073385
0.85	0.023509	0.023408	0.022736	0.019947	0.019864	0.019938	0.020358	0.023123	0.053511	0.0727
0.9	0.023509	0.023408	0.022739	0.020084	0.020034	0.020086	0.020362	0.023054	0.040759	0.07152
0.95	0.023509	0.023408	0.022742	0.020206	0.020189	0.020216	0.020437	0.020974	0.031924	0.066191
1	0.023509	0.023408	0.022745	0.020328	0.020349	0.02035	0.02036	0.020564	0.022528	0.028518

Abstract in lingua italiana

I satelliti Sentinel-1 forniscono numerosi dati SAR (Synthetic Aperture Radar) a livello globale, rivisitando i punti di interesse ogni sei giorni. Sfruttando questi dati, i recenti progressi nell'elaborazione interferometrica differenziale (DInSAR) generano immagini dello spostamento del terreno ad alta risoluzione, con una precisione centimetrica o millimetrica. Queste osservazioni dell'evoluzione delle condizioni del terreno consentono un monitoraggio approfondito di vaste regioni soggette a rischi ambientali. Tuttavia, difficoltà emergono quando alcuni elementi spaziali (ad esempio, acqua o vegetazione) mostrano un comportamento incoerente in successivi intervalli di tempo, con conseguente mancanza di dati in singoli pixel o in intere regioni, persistentemente nel tempo. Le tecniche di ricostruzione statistica dei dati sono applicabili per la ricostruzionedel dato in queste aree, ma in genere fanno affidamento sull'operatore di covarianza, che è sconosciuto per dati funzionali nell'area di interesse, nonchè caratterizzato da pronunciate non stazionarietà. In questa tesi, affrontiamo la sfida di stimare l'operatore di covarianza spaziale da immagini DInSAR temporali introducendo una nuova metodologia non parametrica radicata nei principi dell'analisi funzionale dei dati. Mentre l'approccio non parametrico offre flessibilità nell'affrontare le non stazionarietà del campo, una regolarizzazione laplaciana assicura la continuità del non uniforme operatore di covarianza spaziale ricostruito. La metodologia è dimostrata utilizzando dati DInSAR multitemporali raccolti per monitorare i Campi Flegrei (Italia), una regione soggetta a eventi sismici e bradisismici.

Parole chiave: ricostruzione del nucleo di covarianza, minimizzazione a basso rango, regolarizzazione Laplaciana, completamento funzionale, interferogrammi differenziali SAR (DInSAR)

Acknowledgements

I would like to express my gratitude to Prof. Alessandra Menafoglio, Prof. Simone Vantini and Teresa Bortolotti. I am thankful not only for providing me with the opportunity to undertake this thesis work, but also for their constant support, invaluable guidance and encouragement throughout the process. Their contributions have made this experience not only instructive but also extremely rewarding.
This work has been partially supported by ACCORDO Quadro ASI-POLIMI "Attività di Ricerca e Innovazione" n. 2018-5-HH.0, collaboration agreement between the Italian Space Agency and Politecnico di Milano. The authors gratefully acknowledge the financial support of IREA-CNR (Istituto per il Rilevamento Elettromagnetico dell'Ambiente del Consiglio Nazionale delle Ricerche)