# On the usage of the Gram matrix for the multivariate functional principal components analysis

Steven Golovkine[*] Edward Gunning[†] Andrew J. Simpkin[‡]
Norma Bargary[§]

March 21, 2023

**Abstract**

Dimension reduction has a major importance in functional data analysis (FDA). The key tool to reduce the dimension of the data is functional principal component analysis. Existing approaches for functional principal component analysis usually involve the diagonalisation of the covariance operator. Using the duality of the space of observations and the space of functional components, we propose to use the inner-product between the curves to estimate the eigenelements of, eventually multivariate and multidimensional, functional datasets. The relationship between the eigenelements of the covariance operator and the ones of the inner-product matrix is established. We explore the application of these methodologies in several FDA settings and provide general guidance on their usability.

***Keywords***— Dimension Reduction; Functional Data Analysis; Functional Principal Components; Multivariate Functional Data

## 1 Introduction

Functional data analysis (FDA) is a statistical methodology for analyzing data that can be represented as functions. These functions could represent measurements taken over time or space, such as temperature readings over a period of time or spatial patterns of disease occurrence. The goal of FDA is to extract meaningful information from these functions and to model their behavior. In this article, we will provide a general introduction to FDA, including its history, key concepts, and applications. FPCA is usally used as a preprocessing step to feed regression and classification models.

Neuroscience: MFPCA has been used to analyze brain imaging data to identify patterns of brain activity across multiple brain regions. This can help to better understand the neural mechanisms underlying various cognitive processes and disorders. Climate Science: MFPCA has been used to analyze spatiotemporal climate data to identify patterns of temperature and precipitation variability across multiple regions. This can help to better understand the drivers of climate change and variability. Finance: MFPCA has been used to analyze multivariate financial time series data to identify common patterns of market behavior across multiple stocks. This can help to better understand market trends and to develop more accurate financial forecasting models. Genetics: MFPCA has been used to analyze gene expression data to identify patterns of gene expression across multiple tissues or conditions. This can help to better understand the underlying biological mechanisms involved in various diseases and disorders. Environmental Monitoring: MFPCA has been used to analyze multivariate spatiotemporal environmental monitoring data, such as water quality or air pollution data, to identify patterns of environmental variability across multiple regions. This can help to identify areas where environmental regulations or remediation efforts may be necessary.

To start, we should first define what multivariate functional principal components analysis (MFPCA) is. MFPCA is a statistical technique that is used to analyze multivariate functional data, which are data that consist of multiple functions that are observed simultaneously. MFPCA is based on the idea of functional principal components analysis (FPCA), which is a technique for

[*] MACSI, Department of Mathematics and Statistics, University of Limerick, Ireland steven.golovkine@ul.ie
[†] MACSI, Department of Mathematics and Statistics, University of Limerick, Ireland edward.gunning@ul.ie
[‡] School of Mathematical and Statistical Sciences, University of Galway, Ireland andrew.simpkin@nuigalway.ie
[§] MACSI, Department of Mathematics and Statistics, University of Limerick, Ireland norma.bargary@ul.ie

analyzing single functional data sets. In MFPCA, we seek to decompose the covariance structure of the multivariate functional data into a set of orthogonal basis functions, or principal components, which capture the main sources of variation in the data. These basis functions are determined by eigenanalysis of the covariance matrix of the multivariate functional data. To conduct an MFPCA analysis, we first need to collect multivariate functional data. This data can come from a variety of sources, such as medical imaging, environmental monitoring, or financial data analysis. Once we have collected the data, we can then apply the MFPCA technique to the data to extract the principal components. One of the key benefits of MFPCA is that it allows us to identify and visualize the main sources of variation in the multivariate functional data. This can be useful in many different applications, such as identifying patterns of gene expression in genomics, analyzing changes in brain activity in neuroscience, or predicting financial market trends in economics. When writing an article on MFPCA, it is important to provide a clear explanation of the technique and its applications, as well as a detailed description of the data used in the analysis and the results obtained. It may also be useful to discuss any limitations or assumptions of the MFPCA technique, as well as any extensions or modifications that have been proposed in the literature. In addition to discussing the technical details of the MFPCA technique, it is important to write the article in a clear and concise manner, using appropriate statistical and mathematical notation where necessary. The article should be structured in a logical and organized manner, with clear headings and subheadings to guide the reader through the analysis.

General approaches for FPCA and MFPCA

- Most of the existing methods for FPCA are build upon Ramsay and Silverman (2005) paper.
- Stack the multivariate observation into one and perform usual FPCA Ramsay and Silverman (2005)
- Expand each curves into a basis of functions Jacques and Preda (2014)
- Normed FPCA Jacques and Preda (2014); Chiou et al. (2014), different type of normalization for both of them.
- PCA for each time point Berrendero et al. (2011)
- Develop general methodology for MFPCA Happ and Greven (2015), allows multidimensional data and basis expansion.

From ChatGPT Direct Extension of FPCA: The simplest approach to MFPCA is to perform functional principal components analysis (FPCA) on each of the functional variables separately and then combine the principal components across the variables to obtain a multivariate decomposition. This approach is sometimes referred to as "direct extension" of FPCA. While this approach is straightforward, it may not fully capture the dependence structure between the functional variables. Kronecker Product Approach: The Kronecker product approach involves taking the Kronecker product of the covariance functions of the individual functional variables to obtain a joint covariance function for the multivariate functional data. This joint covariance function can then be used to perform FPCA to obtain the multivariate principal components. Reduced Rank Regression Approach: The reduced rank regression approach involves modeling the multivariate functional data as a linear combination of a small number of basis functions. The coefficients of the linear combination are then modeled as multivariate Gaussian random variables, and the covariance structure of the coefficients is used to perform the MFPCA. Tensor Decomposition Approach: The tensor decomposition approach involves decomposing the multivariate functional data into a higher-order tensor, and then performing a tensor-based decomposition to obtain the multivariate principal components. This approach can be useful when dealing with very high-dimensional functional data. Bayesian Approaches: Bayesian approaches to MFPCA involve specifying a prior distribution on the principal components and the covariance matrix of the multivariate functional data, and then using Bayesian inference to estimate the posterior distribution of the parameters. Bayesian approaches can be useful when dealing with complex dependence structures and small sample sizes.

Key motivations of the paper – Duality between rows and colums of a matrix the duality between rows and columns of a data matrix is a fundamental concept in classical statistics that can be used to understand the relationship between different types of statistical analyses. In classical statistics, a data matrix typically represents a set of observations of multiple variables, where each row corresponds to an individual observation and each column corresponds to a variable. The duality between rows and columns refers to the fact that many statistical analyses can be conducted either on the rows or the columns of the data matrix, and the resulting analyses will be related to each other in a specific way. For example, consider a data matrix where the rows represent individual subjects and the columns represent different measurements taken on each subject, such as blood pressure, weight, and age. Conducting a principal component analysis (PCA) on the columns of this data matrix will result in a set of principal components that capture the variability in the different measurements across all subjects. These principal components can be interpreted as patterns of co-variation across the different measurements. Alternatively, conducting a PCA on the rows of the same data matrix will result in a set of principal components that capture the variability in the measurements across different subjects. These principal components can be interpreted as patterns of co-variation across the different subjects. The duality between rows and columns means that the principal components obtained from these two different analyses will be related in a specific way. Specifically, the principal components obtained from the column PCA can be used to predict the principal components obtained from the row PCA, and vice versa. This relationship can be expressed mathematically as a matrix transpose operation, where the columns of the original data matrix become the rows of the transposed data matrix, and vice versa. In the context of multivariate functional principal components analysis (MFPCA), the duality between rows and columns can be used to understand the relationship between the principal components obtained from the analysis of multiple functional variables observed on the same set of subjects or units. This duality is particularly useful when dealing with high-dimensional functional data where it may not be feasible to analyze all the functional variables at once. The key motivation of the paper is that in a large number of applications, the number of components of the functional datasets is very large, and estimating the eigencomponents of the covariance operator require the diagonalisation of each univariate component which can be computationaly extensive. Using the duality between rows and colums of the data matrix reduce the number of matrix disgonalisation to only one. Another arguments is that for data defined on multidimensional domains, the estimation of the covariance operator remains unclear (how to compute the covariance of 2-dimensional data?). The use of the inner-product matrix allows the eigencomponents of multidimensional data to be computed.

Duality of usual matrix Escofier (1979); Saporta (1990); Pagès (2004); Härdle and Simar (2019)

Duality in the functional data case For functional data, Ramsay (1982) explains the duality of the space of functions and the space of time points. Used in Benko et al. (2009) for univariate and unidimensional functional data. Chen et al. (2017)

Remainder of the paper. The remainder of the paper is organized as follows. In Section 2, we define multivariate functional data with the coordinates possibly having different definition domains. In Section 3, we develop the duality between the observations' space and the functional components space. The relationship between the eigencomponents of the covariance operator of the functional datasets and the eigencomponents of the inner-product matrix between the observations is derived in Section 4. Extensive simulations are given in Section 5. We also provide guidelines on the method to use in which case. The paper concludes with a discussion and an outlook in Section 6.

## 2 Model

The structure of the data we consider, referred to as *multivariate functional data*, is very similar to that presented in Happ and Greven (2015). The data consist of independent trajectories of a vector-valued stochastic process $X = (X^{(1)}, \dots, X^{(P)})^\top$, $P \geq 1$. (Here and in the following, for any matrix $A$, $A^\top$ denotes its transpose.) For each $1 \leq p \leq P$, let $\mathcal{T}_p$ be a rectangle in some Euclidean space $\mathbb{R}^{d_p}$ with $d_p \geq 1$, *e.g.*, $\mathcal{T}_p = [0,1]^{d_p}$. Each coordinate $X^{(p)} : \mathcal{T}_p \to \mathbb{R}$ is assumed to belong to $\mathcal{L}^2(\mathcal{T}_p)$, the Hilbert space of squared-integrable real-valued functions defined on $\mathcal{T}_p$, having the usual inner product that we denote by $\langle \cdot, \cdot \rangle$, and $\lVert \cdot \rVert$ the associated norm. Thus $X$ is a stochastic process indexed by $\mathbf{t} = (t_1, \dots, t_P)$ belonging to the $P-$fold Cartesian product $\mathcal{T} := \mathcal{T}_1 \times \cdots \times \mathcal{T}_P$ and taking values in the $P-$fold Cartesian product space $\mathcal{H} \coloneqq \mathcal{L}^2(\mathcal{T}_1) \times \cdots \times \mathcal{L}^2(\mathcal{T}_P)$.

We consider the function $\langle\!\langle \cdot, \cdot \rangle\!\rangle : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$,

$$\langle\!\langle f, g \rangle\!\rangle \coloneqq \sum_{p=1}^{P} \langle f^{(p)}, g^{(p)} \rangle = \sum_{p=1}^{P} \int_{\mathcal{T}_p} f^{(p)}(t_p) g^{(p)}(t_p) \mathrm{d}t_p, \quad f, g \in \mathcal{H}.$$

$\mathcal{H}$ is a Hilbert space with respect to the inner product $\langle\!\langle \cdot, \cdot \rangle\!\rangle$(Happ and Greven, 2015). We denote by $|\!|\!| \cdot |\!|\!|$, the norm induced by $\langle\!\langle \cdot, \cdot \rangle\!\rangle$. Let $\mu : \mathcal{T} \to \mathcal{H}$ denote the mean function of the process $X$, $\mu(\mathbf{t}) \coloneqq \mathbb{E}(X(\mathbf{t})), \mathbf{t} \in \mathcal{T}$. Let $C$ denotes the $P \times P$ matrix-valued covariance function which, for

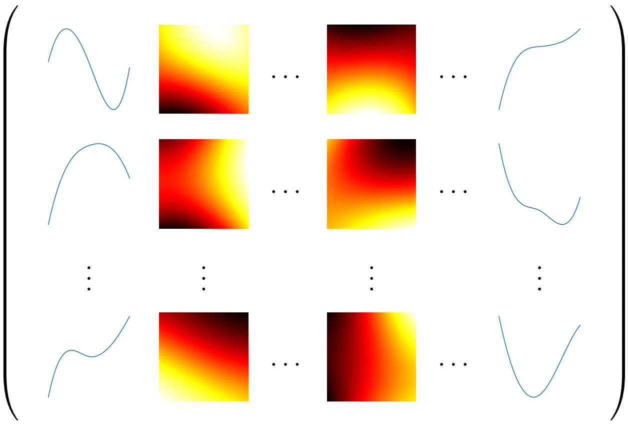

Figure 1: Data matrix, adapted from Berrendero et al. (2011).

$\mathbf{s}, \mathbf{t} \in \mathcal{T}$, is defined as

$$C(\mathbf{s}, \mathbf{t}) \coloneqq \mathbb{E}\left(\{X(\mathbf{s}) - \mu(\mathbf{s})\}\{X(\mathbf{t}) - \mu(\mathbf{t})\}^\top\right), \quad \mathbf{s}, \mathbf{t} \in \mathcal{T}.$$

More precisely, for $1 \leq p, q \leq P$, the $(p, q)$th entry of the matrix $C(\mathbf{s}, \mathbf{t})$ is the covariance function between the $p$th and the $q$th components of the process $X$:

$$C_{p,q}(s_p, t_q) \coloneqq \mathbb{E}\left(\{X^{(p)}(s_p) - \mu^{(p)}(s_p)\}\{X^{(q)}(t_q) - \mu^{(q)}(t_q)\}\right), \quad s_p \in \mathcal{T}_p, t_q \in \mathcal{T}_q.$$

Let $\Gamma : \mathcal{H} \rightarrow \mathcal{H}$ denotes the covariance operator of $X$, defined as an integral operator with kernel $C$. That is, for $f \in \mathcal{H}$ and $\mathbf{t} \in \mathcal{T}$, the $p$th component of $\Gamma f(\mathbf{t})$ is given by

$$(\Gamma f)^{(p)}(t_p) \coloneqq \langle\langle C_{p,\cdot}(t_p, \cdot), f(\cdot)\rangle\rangle = \langle\langle C_{\cdot,p}(\cdot, t_p), f(\cdot)\rangle\rangle, \quad t_p \in \mathcal{T}_p.$$

Let us consider the set of $N$ curves $\mathcal{X} = \{X_1, \dots, X_n, \dots, X_N\}$ generated as a random sample of the $P$-dimensional stochastic process $X$ with continuous trajectories. Unless otherwise stated, the data are assumed to be observed without error. The data can be viewed as a table with $N$ rows and $P$ columns where each entry is a curve, eventually multidimensional (see Figure 1). Each row of this matrix represents an observation; while each column represents a functional variable. At the intersection of row $n$ and column $p$, we thus have $X_n^{(p)}$ which is the curve that concerns the feature $p$ for the individual $n$.

For $n \in \{1, \dots, N\}$, each observation $n$ is attributed the weight $\pi_n$ such that $\sum_n \pi_n = 1$, e.g., $\pi_n = 1/N$. For a given $p \in \{1, \dots, P\}$, the mean curve of the $p$th component along the $N$ observations is denoted by $\mu^{(p)}$. This quantity can be computed using

$$\mu^{(p)}(t_p) = \sum_{n=1}^{N} \pi_n X_n^{(p)}(t_p), \quad t_p \in \mathcal{T}_p, \quad p \in \{1, \dots, P\}.$$

The covariance function of the $p$th component along the $N$ observations can be computed using

$$C_{p,p}(s_p, t_p) = \sum_{n=1}^{N} \pi_n X_n^{(p)}(s_p) X_n^{(p)}(t_p) - \mu^{(p)}(s_p)\mu^{(p)}(t_p), \quad s_p, t_p \in \mathcal{T}_p, \quad p \in \{1, \dots, P\}.$$

# 3 On the geometry of multivariate functional data

## 3.1 Cloud of individuals

Given $n \in \{1, \dots, N\}$, let $\{X_n^{(p)}(t_p),\, t_p \in \mathcal{T}_p,\, p = 1, \dots, P\}$ be the features set for a particular observation $n$. We identify this set as the point $\mathrm{M}_n$ in the space $\mathcal{H}$. The space $\mathcal{H}$ is refered as the observations' space. The cloud of points that represented the set of observations is denoted by $\mathcal{C}_N$. Let $\mathrm{G}_N$ be the centre of gravity of the cloud $\mathcal{C}_N$. In the space $\mathcal{H}$, its coordinates are given by $\{\mu^{(p)}(t_p),\, t_p \in \mathcal{T}_p,\, p = 1, \dots, P\}$. If the variables are centered, the origin $\mathrm{O}_\mathcal{H}$ of the axes in $\mathcal{H}$ coincides with $\mathrm{G}_N$.

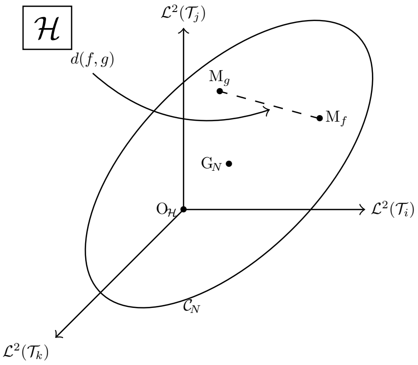

Figure 2: Cloud of observations. The observation $f$ (resp. $g$) is identified by the point $\mathrm{M}_f$ (resp. $\mathrm{M}_g$) in the cloud $\mathcal{C}_N$. The point $\mathrm{G}_N$ is the center of gravity of $\mathcal{C}_N$ and the point $\mathrm{O}_\mathcal{H}$ is the origin of the space $\mathcal{H}$.

Let $f$ and $g$ be two elements in $\mathcal{H}$ and denotes by $\mathrm{M}_f$ and $\mathrm{M}_g$ their associated points in $\mathcal{C}_N$ (see Figure 2). The distance between these observations is defined as

$$d^2(f,g) = \sum_{p=1}^{P} \left\| f^{(p)} - g^{(p)} \right\|^2 = \sum_{p=1}^{P} \int_{\mathcal{T}_p} \left\{ f^{(p)}(t_p) - g^{(p)}(t_p) \right\}^2 \mathrm{d}t_p.$$

This distance measures how different the observations are, and thus characterized the shape of the cloud $\mathcal{C}_N$. Another description of this shape is to consider the distance between each observation and $\mathrm{G}_N$, the center of the cloud. Let $f$ be an element of $\mathcal{H}$, associated to the point $\mathrm{M}_f$ and $\mu$ the element of $\mathcal{H}$ related to $\mathrm{G}_N$, the distance between $f$ and $\mu$ is given by

$$d^2(f,\mu) = \sum_{p=1}^{P} \left\| f^{(p)} - \mu^{(p)} \right\|^2 = \sum_{p=1}^{P} \int_{\mathcal{T}_p} \left\{ f^{(p)}(t_p) - \mu^{(p)}(t_p) \right\}^2 \mathrm{d}t_p.$$

Given the set $\mathcal{X}$, the total inertia of $\mathcal{C}_N$, with respect to $\mathrm{G}_N$, is given by

$$\sum_{n=1}^{N} \pi_n d^2(X_n, \mu) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \pi_i \pi_j d^2(X_i, X_j) = \sum_{p=1}^{P} \int_{\mathcal{T}_p} \operatorname{Var} X^{(p)}(t_p) \mathrm{d}t_p.$$

The derivation of these equalities are given in Appendix A.

**Remark 1.** *These results have the same interpretation as for multivariate scalar data. This is also the multivariate analogue of the relation between variance and sum of squared differences known for univariate functional data. If the variables are reduced beforehand, the total inertia of the cloud $\mathcal{C}_N$ is equal to the number of components $P$. We are, in general, not interested by the total inertia but mostly how this variance is spread among the variables.*

## 3.2 Cloud of variables

For now, we need that all the variables are defined on the same space $\mathcal{T}_0$.

Let $\{X_n^{(p)}(t), n = 1, \ldots, N\}$ be the observations set for a particular feature $p$. We identify this set as the point $\mathsf{M}_p$ in the space $\mathcal{G} := \mathcal{L}^2\left(\mathcal{T}_0\right)^N$. The set $\mathcal{G}$ is refered as the features' space, or variables' space. The cloud of points that represented the set of variables is denoted by $\mathcal{C}_P$. Let $\mathcal{O}_\mathcal{G}$ be the centre of this space. Its coordinates are given by a vector of functions of length $N$ where each entry is $f(t) = 0$ for all $t \in \mathcal{T}_0$.

We assume that the observations are centered. Consider $\mathsf{M}_h$ a point in $\mathcal{C}_P$ and $h$ the element of $\mathcal{G}$ representing by $\mathsf{M}_h$. Let $o$ be the element of $\mathcal{O}_\mathcal{G}$ representing by $\mathcal{O}_\mathcal{G}$. The distance between $\mathsf{M}_h$ and $\mathcal{O}_\mathcal{G}$ is defined as

$$d^2(h, o) = \sum_{n=1}^{N} p_n \left\| h_n - \mu_h \right\|^2 = \int_{\mathcal{T}_0} \operatorname{Var} h^{(p)}(t) dt.$$

For now, the cloud of variables is not clear!

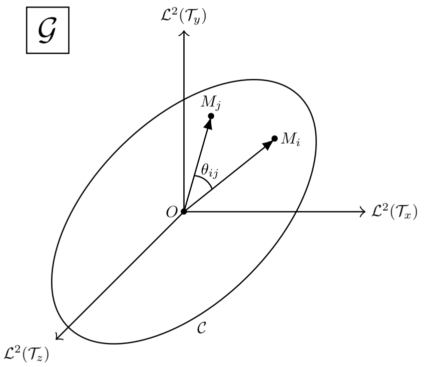
$\mathcal{G}$
$\mathcal{L}^2(\mathcal{T}_y)$
$M_j$
$M_i$
$\theta_{ij}$
$O$
$\mathcal{L}^2(\mathcal{T}_x)$
$\mathcal{C}$
$\mathcal{L}^2(\mathcal{T}_z)$

Figure 3: Cloud of features.

### 3.3 On centering and reducing

Take a look at Prothero et al. (2021)

In MFPCA, the components are usually assumed centred Happ and Greven (2015). If curves are not centered, we replace $X_n^{(p)}(t)$ by $X_n^{(p)}(t) - \mu^{(p)}(t)$. The geometric interpretation of the centering differs if we refer to $\mathcal{H}$ or $\mathcal{G}$. Within the space $\mathcal{H}$, centering is interpreted as translating the centre of gravity of the curves $\mathrm{G}_N$ to the the origin point $\mathrm{O}_{\mathcal{H}}$ of $\mathcal{H}$. This transformation, being a translation, does not change the shape of the cloud $\mathcal{C}_N$. Within the space $\mathcal{G}$, the centering is harder to interpret and has not the same meaning as in the multivariate case (projection on the subspace orthogonal to the constant vector).

**Remark 2.** *What happened if we project the multivariate curves onto the vector of constant functions? In the space $\mathcal{G}$, the inner product is given by*

$$\langle\!\langle f, g\rangle\!\rangle = \sum_{i=1}^{N} \int_{\mathcal{T}_k} f_i^{(k)}(t) g_i^{(k)}(t) dt, \quad f, g \in \mathcal{G}.$$

*Let $\mathbf{1}$ be the vector of constant function in $\mathcal{G}$ and $f$ an element of $\mathcal{G}$. Then, the projection of $f$ onto $\mathbf{1}$ is given by*

$$P_{\mathbf{1}} f = \frac{\langle\!\langle f, \mathbf{1}\rangle\!\rangle}{|\!|\!|\mathbf{1}|\!|\!|}\mathbf{1} = \frac{1}{N|\mathcal{T}_k|} \sum_{i=1}^{N} \int_{\mathcal{T}_k} f_i^{(k)}(t) dt \mathbf{1}$$

*In practice, this is equivalent to compute the mean value of the mean curve for each component.*

Concerning standardisation, there are different proposal in the literature. Happ and Greven (2015) propose to use $w_k = (\int_{\mathcal{T}_k} \operatorname{Var} X^{(k)}(t) dt)^{-1/2}$, while Chiou et al. (2014) consider the function $w_k(t) = (\operatorname{Var} X^{(k)}(t))^{-1/2}$.

## 4 Multivariate functional principal components analysis

Using the results in Happ and Greven (2015), and the theory of Hilbert-Schmidt operators, e.g. Reed and Simon (1980), there exists a complete orthonormal basis $\Phi = \{\phi_k\}_{k \geq 1} \subset \mathcal{H}$ associated to a set of real numbers $\{\lambda_k\}_{k \geq 1}$ such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$ that satisfy

$$\Gamma \phi_k = \lambda_k \phi_k, \quad \text{and} \quad \lambda_k \to 0 \quad \text{as} \quad k \to \infty. \tag{1}$$

The set $\{\lambda_k\}_{k \geq 1}$ contains the eigenvalues of the covariance operator $\Gamma$ and $\Phi$ contains the associated eigenfunctions. Using the multivariate Karhunen-Loève theorem (Happ and Greven, 2015), we obtain the decomposition

$$X(\mathbf{t}) = \mu(\mathbf{t}) + \sum_{k=1}^{\infty} \mathfrak{c}_k \phi_k(\mathbf{t}), \quad \mathbf{t} \in \mathcal{T}$$

where $\mathfrak{c}_k = \langle\!\langle X - \mu, \phi_k \rangle\!\rangle$ are the projection of the centered curve onto the eigenfunctions. We have that $\mathbb{E}(\mathfrak{c}_k) = 0$, $\mathbb{E}(\mathfrak{c}_k^2) = \lambda_k$ and $\mathbb{E}(\mathfrak{c}_k \mathfrak{c}_{k'}) = 0$ for $k \neq k'$. Note that the coefficients $\mathfrak{c}_k$ are scalar random variables while $\phi_k$ is a vector of functions. Let us call $\Phi$ the multivariate functional principal component analysis basis.

********

Assume that for each component $p = 1, \dots, P$, there exists a basis of functions $\{\psi_k^{(p)}\}_{1 \leq k \leq K_p}$ such that each component of the curves can be extended into this basis

$$X^{(p)}(t) = \sum_{k=1}^{K_p} c_{pk}\psi_k^{(p)}(t), \quad t \in \mathcal{T}_0.$$

The $p$th component of the mean function is given by

$$\mu^{(p)}(t) = \sum_{k=1}^{K_p} \mathbb{E}(c_{pk})\psi_k^{(p)}(t), \quad t \in \mathcal{T}_0.$$

The $(p, q)$th entry of the covariance function is given by

$$C_{pq}(s, t) = \sum_{k=1}^{K_p}\sum_{l=1}^{K_q} \left(\mathbb{E}(c_{pk}c_{ql}) - \mathbb{E}(c_{pk})\mathbb{E}(c_{ql})\right)\psi_k^{(p)}(s)\psi_l^{(q)}(t), \quad s, t \in \mathcal{T}_0.$$

Using $N$ realizations of the process, we have $X(t) = C\Psi(t)$ where

$$X(t) = \begin{pmatrix} X_1^{(1)}(t) & \cdots & X_1^{(P)}(t) \\ \vdots & \ddots & \vdots \\ X_N^{(1)}(t) & \cdots & X_N^{(P)}(t) \end{pmatrix},$$

$$C = \begin{pmatrix} C^{(1)} & \cdots & C^{(P)} \end{pmatrix}, \quad \text{with} \quad C^{(p)} = \begin{pmatrix} c_{1p1} & \cdots & c_{1pK_p} \\ \vdots & \ddots & \vdots \\ c_{Np1} & \cdots & c_{NpK_p} \end{pmatrix}$$

$$\Psi(t) = \begin{pmatrix} \psi^{(1)}(t) & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \psi^{(P)}(t) \end{pmatrix}, \quad \text{with} \quad \Psi^{(p)}(t) = \begin{pmatrix} \psi_1^{(p)}(t) \\ \vdots \\ \psi_{K_p}^{(p)}(t) \end{pmatrix}.$$

The estimation of the mean and covariance functions are given by

$$\widehat{\mu}(t) = \frac{1}{N}\mathbb{1}_N^\top C\Psi(t) \quad \text{and} \quad \widehat{C}(s, t) = \frac{1}{N}\Psi(s)^\top C^\top C\Psi(t).$$

We denote by $W$ the matrix of inner products of the basis $\Psi$. The entries of $W$ are given by

$$W_{k,l}^{(p)} = \langle \psi_k^{(p)}, \psi_l^{(p)} \rangle, \quad 1 \leq k, l \leq K_p, \quad 1 \leq p \leq P.$$

## 4.1 By diagonalization of the covariance operator

See Happ and Greven (2015).

## 4.2 By diagonalization of the inner product matrix

We can use the duality relation between row and column spaces of a data matrix to estimate the eigencomponents of the covariance operator. Consider the inner-product matrix $M$ with entries

$$M_{ij} = \langle\!\langle X_i - \mu, X_j - \mu \rangle\!\rangle, \quad i, j = 1, \dots, N.$$

Let $\{l_k\}_{1 \leq k \leq N}$ such that $l_1 \geq \cdots \geq l_N \geq 0$ be the set of eigenvalues and $\{v_k\}_{1 \leq k \leq N}$ be the set of eigenvectors of the matrix $M$. The relationship between all nonzero eigenvalues of the covariance operator and the eigenvalues of $M$ is given by

$$\widehat{\lambda}_k = \frac{l_k}{N}, \quad k = 1, 2, \dots, N,$$

while the relationship between the multivariate eigenfunctions of the covariance operator and the orthonormal eigenvectors of $M$ is given by

$$\widehat{\phi}_k(\mathbf{t}) = \sum_{i=1}^{N} v_{ik}\left\{X_i(\mathbf{t}) - \mu(\mathbf{t})\right\}, \quad \mathbf{t} \in \mathcal{T}, \quad k = 1, 2, \dots, N,$$

where $v_{ik}$ is the $i$th entry of the vector $v_k$. The scores are then computed as the inner-product between the curves and the eigenfunctions and are given by

$$\mathfrak{c}_{ik} = \sqrt{l_k} v_{ik}, \quad i = 1, 2, \dots, N, \quad k = 1, 2, \dots, N.$$

The derivation of these equalities are given in Appendix B.

add using the coefficient expansion

### 4.3 On the smoothing

Where should we do the smoothing in the case we estimate the eigencomponents using the inner product matrix?

Using the covariance operator, Ramsay and Silverman (2005) propose to smooth the eigenfunctions by penalizing their roughness by its integrated squared second derivative.

Using the inner product matrix, we may think about multiple way to do it. First, we smooth all the curves beforehand using kernel regression or local polynomial (see Golovkine et al. (2022) for estimating the optimal bandwidth). Under the twice differentiable curve assumption, we may use the optimal bandwidth as defined by Tsybakov (2009). Second, we could smooth the eigenfunctions directly, maybe using a smoothness penalty as in Ramsay and Silverman (2005) by cross-validation.

### 4.4 Sparse data

I guess we should leave this part for another article. What you should we do for sparse and irregularly sampled functional data? This question can be rewritten as how to compute the inner product between two curves while they are not sampled on the same grid and contains only few observations? One idea would be to consider the PACE algorithm Yao et al. (2005). We might also only need to smooth each curve in "a good way" to estimate the curves on a common grid and apply the usual algorithm Golovkine et al. (2022).

### 4.5 Computational complexity

We describe the time and space complexity for the computation of the MFPCA algorithm using the covariance operator and the inner product matrix. Considering the observation of $N$ curves with $P$ components, we assume that all observations of the component $p$ are sampled on a common grid of $M_p$ points. For $a \in \mathbb{N}$, let $M^a = \sum_p M_p^a$. Let $K$ be the number of multivariate eigenfunctions to estimate. For the estimation of the eigencomponents using the covariance operator, we need to compute the univariate eigencomponents for each component $p$. Let $K_p$ be the number of estimated eigencomponents for the component $p$. In this case, we note $K = \sum_p K_p$. While $K$ has the same interpretation for both the eigendecomposition of the covariance operator and the eigendecomposition of the inner product matrix, in the latter case, it is not computed as the summation over the univariate elements, but rather as the number of components needed to achieve a certain amount of variance explained (see Section 5.4). Here, we also assume that the curves are perfectly observed, and thus no smoothing step is included in the expression of the time complexity.

To estimate the time complexity of an algorithm, we count the number of elementary operations realized, considering a fixed execution time for each elementary operation. Worst-case time complexity is usually considered; and this is what we do in the following. We first give the time complexity for the estimation of the eigencomponents using the covariance operator by expliciting the time complexity of each individual step (see Happ and Greven (2015) and Section 4.1). For each component $p$, the time complexity of the estimation of the covariance matrix is $\mathcal{O}(NM_p^2)$, of the eigendecomposition of the matrix is $\mathcal{O}(M_p^3)$ and of the univariate score is $\mathcal{O}(NM_pK_p)$. Considering all the components, the time complexity is the sum over $p$ the univariate time complexity. The covariance matrix of the stacked univariate scores is then computed with a time complexity of $\mathcal{O}(NK^2)$, because the dimension of this matrix is $N \times K$. The eigendecomposition of this matrix has a time complexity of $\mathcal{O}(K^3)$. The final step is to compute the multivariate eigenfunctions and scores. For the estimation of the $K$ eigenfunctions, the time complexity is $\mathcal{O}(K \sum_p M_pK_p)$ and for the estimation of the scores, the time complexity is $\mathcal{O}(NK^2)$. Gathering all the results, the final complexity of the estimation of the eigencomponents using the eigendecomposition of the covariance operator is

$$\mathcal{O}\left(NM^2 + M^3 + N\sum_{p=1}^{P} M_pK_p + NK^2 + K^3 + K\sum_{p=1}^{P} M_pK_p + NK^2\right).$$

We now consider the time complexity of the estimation of the eigencomponents using the eigendecomposition of the inner product matrix (see Section 4.2). The inner product between two curves can be estimated in $\mathcal{O}(M^1)$. There are $N^2$ terms in the matrix. The time complexity for the computation of the inner product matrix is then $\mathcal{O}(N^2M^1)$. The eigendecomposition of this matrix has a time complexity of $\mathcal{O}(N^3)$. For the multivariate eigenfunctions, the time complexity is $\mathcal{O}(KNP)$ and is $\mathcal{O}(KN)$ for the multivariate scores. Gathering all the results, the final complexity of the estimation of eigencomponents using the eigendecomposition of the inner product matrix is

$$\mathcal{O}\left(N^2M^1 + N^3 + KNP + KN\right).$$

Develop for space complexity

Comment on the results of the time complexity, explain the trade-off between the two, the leading term, etc.

**Remark 3.** *We can use the SVD in both case to make the algorithm faster as it allows to compute only the first $k$ eigenfunctions.*

### 4.6 Percentage of variance explained

We argue that the percentage of variance explained in Happp and Greven (2015) is not the good one as they consider the variance explained by each of the components separately and not as a all. Using the inner product matrix however gives the right number of eigenfunctions for a given amount of variance explained.

See attached, a very brief simulation. It seems to show that choosing a univariate cut-off within each dimension (e.g., 95%), tends to overestimate the final amount of variance – the sum of the final eigenvalues is larger than the sum of the true eigenvalues. I did some reading – Happ and Greven considered the effect of pve only on eigenfunction estimation. I've attached the relevant pages from their supplementary material. I'm not sure they considered eigenvalues or the total variation after. Moreover, I think (but not sure) they assume in their simulation that $M$ is known where they say $\min\{M_1 + M_2, M\}$; whereas in practice, it is obviously unknown. But maybe they're just referring to M as the total number of sample eigenfunctions with non-zero eigenvalues (rather than the true number of multivariate eiegenfunctions).

Note that in Happ and Greven (2015), the percentage of variance explained in not specified in the MFPCA algorithm. The number of components is specified and is chosen to be equal to be the minimum between $\sum_{p=1}^{P} K_p$ and $K$.

## 5 Simulations

**Theorem 1** (Mercer's theorem)**.** *Let $p \in \{1, \dots, P\}$. Then*

$$\mathrm{Cov}(X^{(p)}(s), X^{(p)}(t)) = \sum_{k=1}^{\infty} \lambda_k \psi_k^{(p)}(s) \psi_k^{(p)}(t), \quad s, t \in \mathcal{T}_p,$$

*where the convergence is absolute and uniform.*

*Proof.* See Happ and Greven (2015) for a proof. □

**Theorem 2** (Multivariate Karhunen-Loève theorem)**.** *Under some assumption,*

$$X(t) = \sum_{k=1}^{\infty} \langle\langle X, \psi_k \rangle\rangle \psi_k(t), t \in \mathcal{T}.$$

*Proof.* See Happ and Greven (2015) for a proof. □

We run simulations to check the goodness of fit of the MFPCA using both methods and compare the time complexity of the two ways.

Some ideas:

- Varies number of curves in the sample $N = 50, 100, 200$.
- Varies number of points per curve $M = 20, 50, 100$.
- Varies number of components $P = 1, 5, 10, \dots$.

- Common grid / Irregularly sampled grid
- Denoising

For all settings, we consider that the curves are centered such that $\mu = 0$. We generate covariance matrices $\Gamma$ using Mercer's theorem. We generate samples of independent paths $X_i$ from the Gaussian process characterized by $\mu$ and $\Gamma$.

We consider some number of experiments, each of them replicated 500 times.

The results of the simulation are compared using computation times, the integrated squared error (ISE) risk for the multivariate eigenfunctions and the mean squared error (MSE) risk for the eigenvalues and the mean integreated squared error (MISE) risk for the reconstructed data.

Definition ISE for eigenfunctions Let $psi$ be the true eigenfunction and $\widehat{\psi}$ the estimated eigenfunction defined on $\mathcal{T}$. We then defined the ISE as

$$\text{ISE}(\psi, \widehat{\psi}) = \sum_{p=1}^{P} \int_{\mathcal{T}_p} \{\psi^{(p)}(t) - \widehat{\psi}^{(p)}(t)\}^2 \text{d}t.$$

Definition MSE for eigenvalues Let $\lambda = (\lambda_1, \dots, \lambda_K)$ be the vector of true eigenvalues and $\widehat{\lambda} = (\widehat{\lambda}_1, \dots, \widehat{\lambda}_K)$ be the vector of estimated eigenvalues. We then define the MSE as

$$\text{MSE}(\lambda, \widehat{\lambda}) = \sum_{k=1}^{K} \frac{\widehat{\lambda}_k}{\lambda_k}.$$

Definition MISE for reconstructed data Let $\mathcal{X}$ be the set of true data and $\widehat{\mathcal{X}}$ be the set of reconstructed data. We defined the MISE as

$$\text{MISE}(\mathcal{X}, \widehat{\mathcal{X}}) = \frac{1}{N} \sum_{n=1}^{N} \sum_{p=1}^{P} \int_{\mathcal{T}_p} \left\{ X_n^{(p)} - \widehat{X}_n^{(p)} \right\}^2 \text{d}t.$$

Consider the simulations in Happ and Greven (2015) as example.

We first consider curves observed without noise and a common and regularly spaced grid. For 2D data, we compare the decomposition of the Gram matrix with the FCP-TPA algorithm Allen (2013). We choose to compare with this algortihm because it is the one that is proposed in Happ and Greven (2015) to compute the MFPCA where some components are 2D. Note that we could also use a 2D basis expansion such as penalized tensor splines or discrete cosine transform.

## 5.1 Simulation settings

The simulation settings are based on the simulation in Happ and Greven (2015). For data on one-dimensional domains, the data generating process is based on a truncated version of the Karhunen-Loève decomposition. First, we generate a large orthonormal basis $\{\phi_k\}_{k \in \mathbb{N}}$ of $\mathcal{L}^2(\mathcal{T})$ on an interval $\mathcal{T} = [0, T] \subset \mathbb{R}$. We define $P$ cutting points $T_1, \dots, T_P$ in $\mathcal{T}$, $\eta_1, \dots, \eta_P \in \mathbb{R}$ and $\sigma_1, \dots, \sigma_P \in \{-1, 1\}$. The univariate components are defined as

$$\psi_k^{(p)}(t_p) = \sigma_p \phi_k\big|_{[T_p, T_{p+1}]}(t_p - \eta_p), \quad k = 1, \dots, K.$$

The set of functions $\{\psi_k, k = 1, \dots, K\}$ is an orthonormal system in $\mathcal{H} \coloneqq \mathcal{L}^2(\mathcal{T}_1) \times \cdots \times \mathcal{L}^2(\mathcal{T}_P)$ with $\mathcal{T}_p = [T_p + \eta_p, T_{p+1} + \eta_p]$.

For images data, the eigenfunctions are simulated using the truncated Karhunen-Loève decomposition. The images basis of functions is build on a tensor product of an orthonormal basis of functions.

The scores are generated as independent realisations of a Gaussian distribution with mean 0 and variance $\lambda_k$. We use an exponential decreasing of the eigenvalues.

## 5.2 Simulation results

## 5.3 Computation time

## 5.4 Percentage of variance explained

We show that choosing a univariate cut-off within each univariate dimension (e.g, 95%) does not lead to the same percentage of variance explained for the multivariate functions.

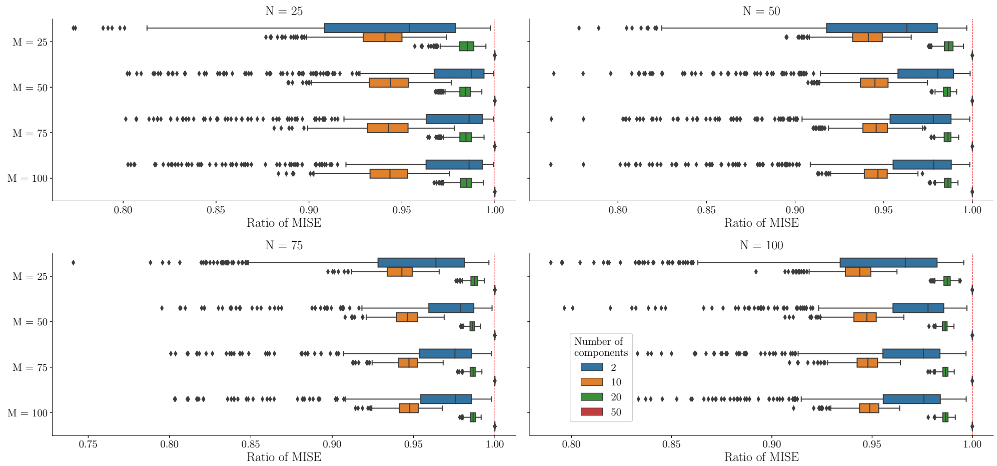

Figure 4: MISE for multivariate functional data. Each univariate component is defined on a one-dimensional domain.

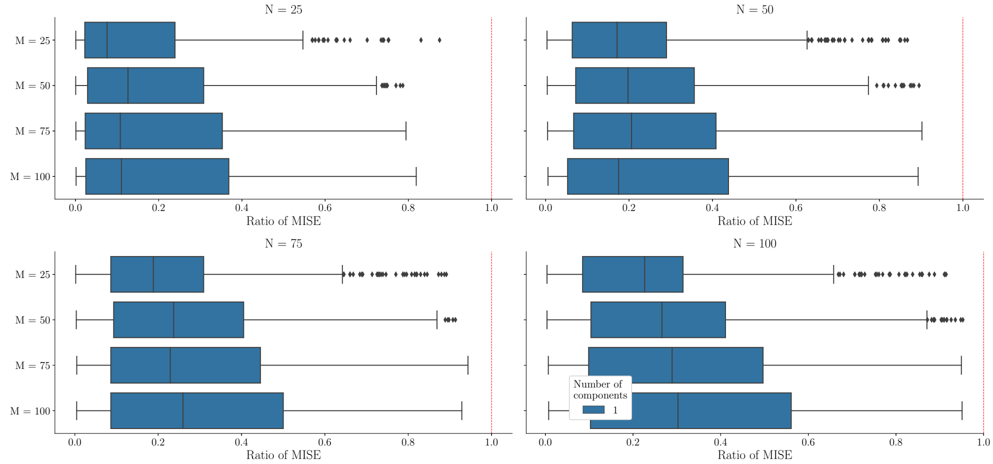

Figure 5: MISE for 2D data

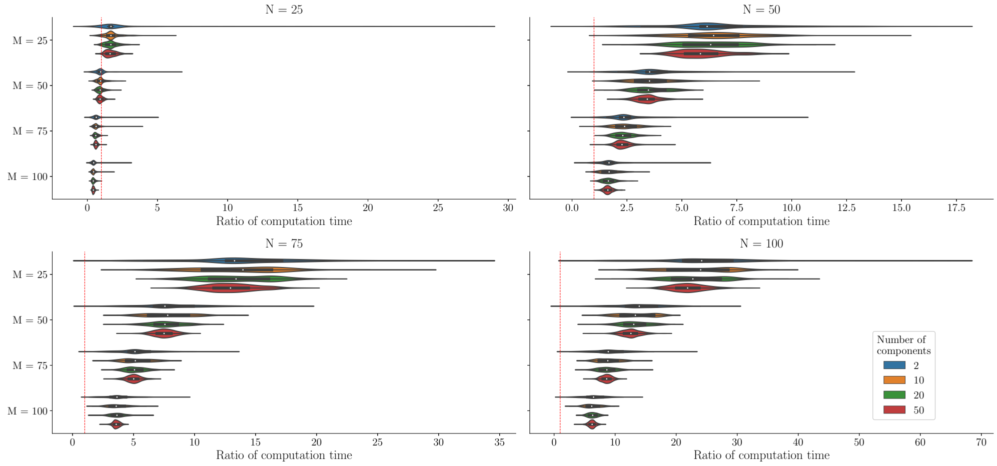

Figure 6: Computation time for multivariate functional data. Each univariate component is defined on a one-dimensional domain.

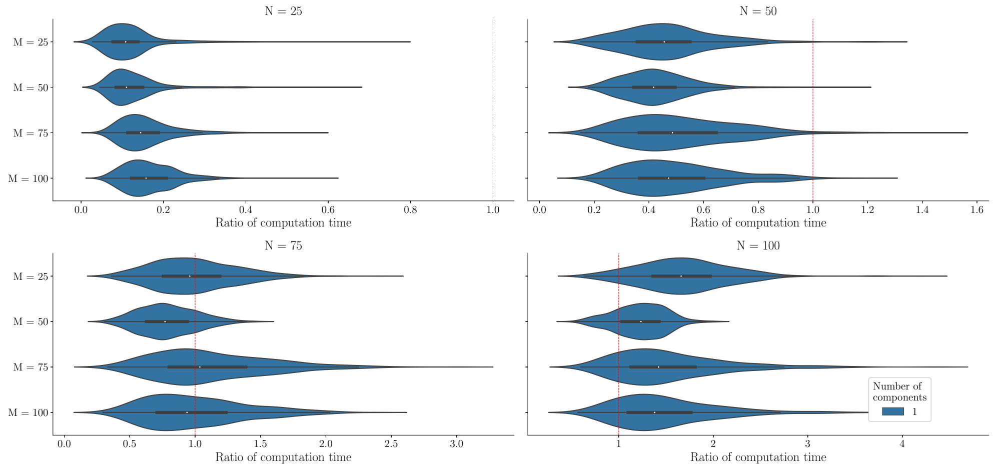

Figure 7: Computation time for univariate functional data of images data

We consider the following settings of simulation. Using the multivariate Karhunen-Loève theorem, we generate

$$\begin{pmatrix} X^{(1)} \\ X^{(2)} \end{pmatrix} = \sum_{k=1}^{20} \mathfrak{c}_k \begin{pmatrix} \psi_k^{(1)} \\ \psi_k^{(2)} \end{pmatrix},$$

where $\{\psi_k^{(1)}\}_{k=1,\dots,20}$ are the first 20 functions of the Fourier basis and $\{\psi_k^{(2)}\}_{k=1,\dots,20}$ are the first 20 functions of the Wiener basis, the coefficients $\mathfrak{c}_k$ are generated independently from a Gaussian distribution with mean 0 and variance $\lambda_k$. We consider linear and exponential decreasing of the eigenvalues.

The true percentage of variance explained by the $k$th component is given by $\lambda_k / \sum_{k'=1}^{20} \lambda_{k'}$. The true percentage of variance explained by the first $K$ components is given by $\sum_{k=1}^{K} \lambda_k / \sum_{k'=1}^{20} \lambda_{k'}$.

We fix a certain percentage of variance explained $\alpha$. To compare the estimation of the estimation of the number of components needed to reach $\alpha$, we first run FPCA on each univariate components to estimate the number of components needed to explained $\alpha\%$ of the variance for each components. Then, we run an MFPCA with the number of components equal to the sum of the univariate components. We finally compare the estimated number of components with the true one.

Using the Gram matrix, we directly estimated the number of components needed to explain $\alpha\%$ of the variance within the data.

### 5.5 Noisy data

We consider noisy dataset. We simulate datasets are the same as before except we add a random noise as an independent Gaussian variable with mean 0 and variance $\sigma^2$ to the curve.

$$Y(t) = X(t) + \varepsilon(t)$$

## 6 Discussion

Discussion about the methods and possible extension. What about the sparse/noisy data?

## A Derivation of the inertia of the clouds

In this Section, we derive the inertia of the cloud $\mathcal{C}_N$. Recall that

$$\mathrm{Var}\{X^{(p)}(t_p)\} = \sum_{n=1}^{N} \pi_n \{X_n^{(p)}(t_p)\}^2 - \{\mu^{(p)}(t_p)\}^2 \quad \text{where} \quad \mu^{(p)}(t_p) = \sum_{n=1}^{N} \pi_n X_n^{(p)}(t_p), \quad t_p \in \mathcal{T}_p.$$

$$
\begin{aligned}
\sum_{n=1}^{N} \pi_n d^2(X_n, \mu) &= \sum_{n=1}^{N} \pi_n \sum_{p=1}^{P} \left\| X_n^{(p)} - \mu^{(p)} \right\|^2 \\
&= \sum_{p=1}^{P} \left( \sum_{n=1}^{N} \pi_n \left\| X_n^{(p)} \right\|^2 - \left\| \mu^{(p)} \right\|^2 \right) \\
&= \sum_{p=1}^{P} \int_{\mathcal{T}_p} \operatorname{Var} X^{(p)}(t_p) \mathrm{d}t_p \\
\sum_{i=1}^{N} \sum_{j=1}^{N} \pi_i \pi_j d^2(X_i, X_j) &= \sum_{i=1}^{N} \sum_{j=1}^{N} \pi_i \pi_j \sum_{p=1}^{P} \left\| X_i^{(p)} - X_j^{(p)} \right\|^2 \\
&= \sum_{p=1}^{P} \left( 2 \sum_{i=1}^{N} \pi_i \left\| X_i^{(p)} \right\|^2 - 2 \sum_{i=1}^{N} \sum_{j=1}^{N} \langle X_i^{(p)}, X_j^{(p)} \rangle \right) \\
&= \sum_{p=1}^{P} \left( 2 \sum_{i=1}^{N} \pi_i \left\| X_i^{(p)} \right\|^2 - 2 \left\| \mu^{(p)} \right\|^2 - 2 \sum_{i=1}^{N} \sum_{j=1}^{N} \langle X_i^{(p)}, X_j^{(p)} \rangle + 2 \left\| \mu^{(p)} \right\|^2 \right) \\
&= 2 \sum_{p=1}^{P} \int_{\mathcal{T}_p} \operatorname{Var} X^{(p)}(t_p) \mathrm{d}t_p
\end{aligned}
$$

# B Derivation of the eigencomponents

Using the Hilbert-Schmidt theorem, there exists a complete orthonormal basis of eigenvectors $\{v_k\}_{1 \leq k \leq N}$ of the inner-product matrix $M$ such that

$$
M v_k = l_k v_k. \tag{2}
$$

Let $X = (X_1 - \mu, \dots, X_N - \mu)^\top$. Recall that, in the case of $P$-dimensional process, the realisations of the process $X_i,\ i = 1, \cdots, N$ and $\mu$ are vectors of functions of length $P$, and thus $X$ is a matrix of functions of size $N \times P$. By left multiplying Equation (2) by $X^\top$, we obtain

$$
X^\top M v_k = l_k X^\top v_k. \tag{3}
$$

Expanding Equation (3), for each component $p = 1, \dots, P$, we have

$$
\sum_{i=1}^{N} \sum_{j=1}^{N} v_{jk} \left\{ X_i^{(p)} - \mu^{(p)} \right\} \langle\langle X_i - \mu, X_j - \mu \rangle\rangle = l_k \sum_{i=1}^{N} v_{ik} \left\{ X_i^{(p)} - \mu^{(p)} \right\}. \tag{4}
$$

In the following, we note $[a]_p$ the $p$th entry of the vector $a$. Starting from the left side of Equation (4), we get

$$
\begin{aligned}
[X^\top M v_k]_p &= \sum_{i=1}^{N} \sum_{j=1}^{N} v_{jk} \left\{ X_i^{(p)} - \mu^{(p)} \right\} \langle\langle X_i - \mu, X_j - \mu \rangle\rangle \qquad (5) \\
&= \sum_{q=1}^{P} \int_{\mathcal{T}_q} \sum_{i=1}^{N} \left\{ X_i^{(p)} - \mu^{(p)} \right\} \left\{ X_i^{(q)}(s) - \mu^{(q)}(s) \right\} \sum_{j=1}^{N} v_{jk} \left\{ X_j^{(q)}(s) - \mu^{(q)}(s) \right\} \mathrm{d}s \\
&= \sum_{q=1}^{P} \int_{\mathcal{T}_q} N C_{pq}(\cdot, s) \sum_{j=1}^{N} v_{jk} \left\{ X_j^{(q)}(s) - \mu^{(q)}(s) \right\} \mathrm{d}s \\
&= N \sum_{j=1}^{N} \langle\langle C_{p\cdot}(\cdot, \cdot), v_{jk} \left\{ X_j - \mu \right\} \rangle\rangle \\
&= N \Gamma \left( \sum_{j=1}^{N} v_{jk} \left\{ X_j - \mu \right\} \right)^{(p)} (\cdot)
\end{aligned}
$$

and, starting from the right side of Equation (4),

$$[l_k X^\top v_k]_p = l_k \sum_{i=1}^N v_{ik} \left\{ X_i^{(p)} - \mu^{(p)} \right\}. \tag{6}$$

From Equation (5) and Equation (6), we obtain

$$\Gamma \left( \sum_{j=1}^N v_{jk} \left\{ X_j - \mu \right\} \right)^{(p)} (\cdot) = \frac{l_k}{N} \sum_{i=1}^N v_{ik} \left\{ X_i^{(p)} - \mu^{(p)} \right\}, \quad \text{for all } p = 1, \dots, P.$$

By identification in Equation (1), we find that, for each components $p$,

$$\lambda_k = \frac{l_k}{N} \quad \text{and} \quad \phi_k^{(p)} = \sum_{i=1}^N v_{ik} \left\{ X_i^{(p)} - \mu^{(p)} \right\}, \quad k \geq 1.$$

For $k \geq 1$, the norm of the eigenfunction is computed as the following:

$$\begin{aligned} \lVert\!| \phi_k |\!\rVert^2 &= \sum_{i=1}^N \sum_{j=1}^N v_{ik} v_{jk} \langle\!\langle X_i - \mu, X_j - \mu \rangle\!\rangle = \sum_{i=1}^N v_{ik} \sum_{j=1}^N M_{ij} v_{jk} \\ &= \sum_{i=1}^N v_{ik} l_k v_{ik} = l_k \lVert v_k \rVert^2 = l_k. \end{aligned}$$

In order to have an orthonormal basis of eigenfunction, we normalise the previously found $\phi_k$ by $1/\sqrt{l_k}$. Concerning the estimation of the scores, for $i = 1, \dots, N$, for $k \geq 1$, we have

$$\begin{aligned} \widehat{\mathfrak{c}}_{ik} = \langle\!\langle X_i - \mu, \phi_k \rangle\!\rangle &= \frac{1}{\sqrt{l_k}} \sum_{j=1}^N v_{jk} \langle\!\langle X_i - \mu, X_j - \mu \rangle\!\rangle \\ &= \frac{1}{\sqrt{l_k}} \sum_{j=1}^N v_{jk} M_{ij} = \sqrt{l_k} v_{ik}. \end{aligned}$$

***********

Concerning the expansion of the data into the basis of function $\Psi$, we write

$$M = \left( C W^{1/2} \right) \left( C W^{1/2} \right)^\top.$$

We also assume that $\widehat{\phi}_1, \widehat{\phi}_2, \dots$ the eigenfunctions of $\widehat{\Gamma}_N$ have a decomposition into the basis $\Psi$

$$\widehat{\phi}_k(t) = \begin{pmatrix} \widehat{\phi}_k^{(1)}(t) \\ \vdots \\ \widehat{\phi}_k^{(P)}(t) \end{pmatrix} = \begin{pmatrix} \psi^{(1)\top}(t) b_{1k} \\ \vdots \\ \psi^{(P)\top}(t) b_{Pk} \end{pmatrix}, \quad \text{where} \quad b_{pk} = \left( b_{pk1}, \dots, b_{pkK_p} \right)^\top.$$

We have, for all $t \in \mathcal{T}_0$, $p = 1, \dots, P$,

$$\begin{aligned} \left( \widehat{\Gamma}_N \phi_k \right)^{(p)} (t) &= \sum_{q=1}^P \int_{\mathcal{T}_0} \widehat{C}_{pq}(t, s) \phi_k^{(q)}(s) ds \\ &= \frac{1}{N} \sum_{q=1}^P \int_{\mathcal{T}_0} \Psi(t)^{(p)\top} C^{(p)\top} C^{(q)} \Psi^{(q)}(s) \Psi^{(q)}(s)^\top b_{qk} ds \\ &= \frac{1}{N} \Psi(t)^{(p)\top} C^{(p)\top} \sum_{q=1}^P C^{(q)} \int_{\mathcal{T}_0} \Psi^{(q)}(s) \Psi(s)^{(q)\top} ds b_{qk} \\ &= \frac{1}{N} \Psi(t)^{(p)\top} C^{(p)\top} \sum_{q=1}^P C^{(q)} W^{(q)} b_{qk}. \end{aligned}$$

This equation is true for all $p = 1, \cdots, P$, this can be rewritten with matrices as

$$\widehat{\Gamma}_N \phi_k(t) = \frac{1}{N} \Psi^\top(t) C^\top C W b_k.$$

From the eigenequation, we have that

$$\widehat{\Gamma}_N \phi_k(t) = \lambda_k \phi_k(t) \Longleftrightarrow \frac{1}{N} \Psi(t)^\top C^\top C W b_k = \lambda_k \Psi(t)^\top b_k, \quad t \in \mathcal{T}_0.$$

Since this equation must be true for all $t \in \mathcal{T}_0$, this imply the equation

$$C^\top C W b_k = N \lambda_k b_k. \tag{7}$$

As the eigenfunctions are assumed to be normalized, $\lVert\!| \phi_k \rVert\!|^2 = 1$. And so, $b_k^\top W b_k = 1$. Let $u_k = W^{1/2} b_k$. Then, from Equation (7), we obtain

$$W^{1/2} C^\top C W^{1/2} u_k = N \lambda_k u_k \Longleftrightarrow \left( C W^{1/2} \right)^\top \left( C W^{1/2} \right) u_k = N \lambda_k u_k. \tag{8}$$

From the eigendecomposition of the matrix $M$, we get

$$M v_k = l_k v_k \Longleftrightarrow \left( C W^{1/2} \right) \left( C W^{1/2} \right)^T v_k = l_k v_k. \tag{9}$$

The equations (8) and (9) are eigenequations in the classical PCA case, with the duality $X^\top X$ and $X X^\top$. Following Pagès (2014), we find that, for $1 \leq k \leq K$,

$$\lambda_k = \frac{l_k}{N}, \quad v_k = \frac{1}{\sqrt{l_k}} C W^{1/2} u_k \quad \text{and} \quad u_k = \frac{1}{\sqrt{l_k}} W^{1/2} C^\top v_k.$$

And finally, to get the coefficient of the eigenfunctions, for $1 \leq k \leq K$,

$$b_k = W^{-1/2} u_k = \frac{1}{\sqrt{l_k}} C^\top v_k.$$

# Acknowledgment

S. Golovkine was partially supported by Science Foundation Ireland under Grant No. 19/FFP/7002 and co-funded under the European Regional Development Fund.

# References

Allen, G. I. (2013). Multi-way functional principal components analysis. In *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 220–223.

Benko, M., Härdle, W., and Kneip, A. (2009). Common functional principal components. *The Annals of Statistics*, 37(1):1–34.

Berrendero, J., Justel, A., and Svarc, M. (2011). Principal components for multivariate functional data. *Computational Statistics & Data Analysis*, 55:2619–2634.

Chen, K., Zhang, X., Petersen, A., and Müller, H.-G. (2017). Quantifying Infinite-Dimensional Data: Functional Data Analysis in Action. *Statistics in Biosciences*, 9(2):582–604.

Chiou, J.-M., Chen, Y.-T., and Yang, Y.-F. (2014). Multivariate Functional Principal Component Analysis: A Normalization Approach. *Statistica Sinica*, 24(4):1571–1596.

Escofier, B. (1979). Traitement simultané de variables qualitatives et quantitatives en analyse factorielle. *Cahiers de l'analyse des données*, 4(2):137–146.

Golovkine, S., Klutchnikoff, N., and Patilea, V. (2022). Learning the smoothness of noisy curves with application to online curve estimation. *Electronic Journal of Statistics*, 16.

Happ, C. and Greven, S. (2015). Multivariate Functional Principal Component Analysis for Data Observed on Different (Dimensional) Domains. *Journal of the American Statistical Association*, 113.

Härdle, W. K. and Simar, L. (2019). *Applied Multivariate Statistical Analysis*. Springer Nature.

Jacques, J. and Preda, C. (2014). Model-based clustering for multivariate functional data. *Computational Statistics & Data Analysis*, 71:92–106.

Pagès, J. (2004). Analyse factorielle de données mixtes. *Revue de Statistique Appliquée*, 52(4):93–111.

Pagès, J. (2014). *Multiple Factor Analysis by Example Using R*. CRC Press.

Prothero, J. B., Hannig, J., and Marron, J. S. (2021). New Perspectives on Centering.

Ramsay, J. and Silverman, B. W. (2005). *Functional Data Analysis*. Springer Science & Business Media.

Ramsay, J. O. (1982). When the data are functions. *Psychometrika*, 47(4):379–396.

Reed, M. and Simon, B. (1980). *Methods of Modern Mathematical Physics: Functional Analysis*. Academic Press.

Saporta, G. (1990). Simultaneous Analysis of Qualitative and Quantitative Data. *Atti 35° Riunione Scientifica della Societa Italiana di Statistica*, pages pp. 63–72.

Tsybakov, A. B. (2009). *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer New York, New York, NY.

Yao, F., Müller, H.-G., and Wang, J.-L. (2005). Functional Data Analysis for Sparse Longitudinal Data. *Journal of the American Statistical Association*, 100(470):577–590.