A Multifidelity Function-on-Function Model Applied to an Abdominal Aortic Aneurysm

Christoph Striegel, Jonas Biehler, Wolfgang A. Wall & Göran Kauermann

TECHNOMETRICS
2022, VOL. 64, NO. 3, 279–290
https://doi.org/10.1080/00401706.2021.2024453

Taylor & Francis
Taylor & Francis Group

A Multifidelity Function-on-Function Model Applied to an Abdominal Aortic Aneurysm

Christoph Striegel[a], Jonas Biehler[b], Wolfgang A. Wall[b], and Göran Kauermann[a]

[a]Department of Statistics, Ludwig-Maximilians-Universität München, München, Germany; [b]Department of Mechanical Engineering, Technische Universität München, Garching bei München, Germany

ABSTRACT
In this work, we predict the outcomes of high fidelity multivariate computer simulations from low fidelity counterparts using function-to-function regression. The high fidelity simulation takes place on a high definition mesh, while its low fidelity counterpart takes place on a coarsened and truncated mesh. We showcase our approach by applying it to a complex finite element simulation of an abdominal aortic aneurysm which provides the displacement field of a blood vessel under pressure. In order to link the two multidimensional outcomes we compress them and then fit a function-to-function regression model. The data are high dimensional but of low sample size, meaning that only a few simulations are available, while the output of both low and high fidelity simulations is in the order of several thousands. To match this specific condition our compression method assumes a Gaussian Markov random field that takes the finite element geometry into account and only needs little data. In order to solve the function-to-function regression model we construct an appropriate prior with a shrinkage parameter which follows naturally from a Bayesian view of the Karhunen–Loève decomposition. Our model enables real multivariate predictions on the complete grid instead of resorting to the outcome of specific points.

ARTICLE HISTORY
Received August 2019
Accepted December 2021

KEYWORDS
Computer experiments; Functional data analysis; Gaussian Markov random fields; Multifidelity simulations; Multivariate data; Spatial Statistics

1. Introduction

In this work, we deal with data gathered from a high fidelity computer simulation, which depends on a variable parameter vector $\boldsymbol{\xi}$. The simulation is complex but deterministic and computes scalar variables on a three-dimensional finite element structure $S_y = \{s_j, j : j = 1, \ldots, N_y\}$. The overall simulation outcomes are vector-valued and denoted with the vector $\boldsymbol{y} = \left(y(s_1), \ldots, y(s_{N_y})\right)^T, \boldsymbol{y} \in \mathbb{R}^{N_y}$, with $y(s_j)$ denoting the outcome at location $s_j \in S_y$. The parameter vector $\boldsymbol{\xi}$ is high dimensional and does not enter the finite element simulation directly, instead it is used to generate necessary wall parameters. We refer to the supplementary material for an in depth description of the separate steps. As the outcome is wholly dependent upon the parameter, we denote the outcome of the simulation as $\boldsymbol{y}(\boldsymbol{\xi})$. Our finite element simulation is computationally expensive and determines the deformations and stresses of an abdominal aortic aneurysm (AAA) in response to blood pressure, see Biehler, Gee, and Wall (2014). The outcome of a single simulation is shown as a part of Figure 1. A low fidelity simulation is also available as a computationally less intensive alternative. This simulation depends on the same $\boldsymbol{\xi}$ and gives outcome $\boldsymbol{x}(\boldsymbol{\xi})$ as points on a low-dimensional mesh S_x, with $\boldsymbol{x}(\boldsymbol{\xi})$ following the same notational convention from above. This means $\boldsymbol{x} = \left(x(\tilde{s}_1), \ldots, x(\tilde{s}_{N_x})\right)^t$ is the vector valued outcome on the finite element structure $S_x = \{\tilde{s}_j : k = 1, \ldots N_x\}$. This outcome is shown in Figure 1. Please notice that $\boldsymbol{y}$ is in the order of 80,000 while $\boldsymbol{x}$ is in the order of 3500 components. The intention is to quantify the relation between $\boldsymbol{x}$ and $\boldsymbol{y}$, or more specifically, to predict $\boldsymbol{y}(\boldsymbol{\xi})$ from $\boldsymbol{x}(\boldsymbol{\xi})$ even if the number of pairs of outcomes is only a few dozens: Using a two-step approach that involves a compression with a Gaussian Markov random field assumption and a Bayesian regression model to connect the compressed vectors we will derive real vector valued predictions for $\boldsymbol{y}(\boldsymbol{\xi})$.

This could be useful in uncertainty quantification and optimization, both of which require a high number of high fidelity evaluations, which can then be substituted with adjusted cheap low fidelity counterparts. In our concrete case the ultimate goal is to take the aneurysm geometry of an actual patient and estimate distributions for the location and magnitude of high stress to guide further medical treatment. Here it is crucial to understand that the necessary wall parameters cannot be measured for an actual living patient, but instead we rely on distributions estimated from collected tissue samples and want to propagate this information. The framework we develop in this work is an important step toward this application, which will then require additional care in regard to the sampling scheme of $\boldsymbol{\xi}$ and the treatment of the overall error among other things.

The remainder of the article is organized as follows. In Section 2 we give a short review of the related literature and lay out how we want to build upon these previous results. In Section 3 we provide technical details about the computer simulation and explain how the data are generated. In Section 4 we derive the model before fitting it for the case of our specific application in

CONTACT Christoph Striegel christoph.striegel@stat.uni-muenchen.de Department of Statistics, Ludwig-Maximilians-Universität München, 80539 München, Germany.

Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/TECH.

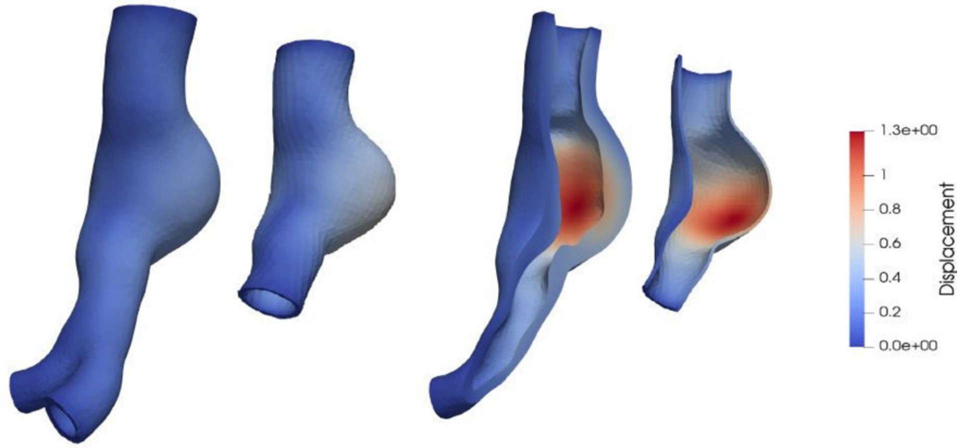

Figure 1. High (first and third image) and low (second and fourth image) fidelity output for the same set of parameters. The images show the artery from the outside (leftmost images) and cutaway (rightmost images), demonstrating the three dimensional structure of the data. In addition to other simplifications, the low fidelity model is also computed using a coarser mesh (not shown above) than the high fidelity model.

Section 5. Here we also give some brief notes on alternatives as well as numerical details and tuning parameters. The work finishes with a discussion in Section 6 that provides further context for the results.

2. Review of the Literature

The problem of relating high fidelity to low fidelity simulations is a frequently studied topic in both statistics and engineering. A good starting point is Kennedy and O'Hagan (2000). Using their notation, they propose the model

$$y(\boldsymbol{\xi}) = \rho \cdot x(\boldsymbol{\xi}) + \delta(\boldsymbol{\xi}), \tag{1}$$

where $\delta(\cdot)$ is of residual type and independent of $x(\boldsymbol{\xi})$. In this case, y and x are either univariate or have the same dimension. Taking a single location, for example s, which lies on both S_y and S_x, we can rewrite (1) as (see also Levy and Steinberg 2010)

$$y(\boldsymbol{\xi}, s) = \rho(s) \cdot x(\boldsymbol{\xi}, s) + \delta(\boldsymbol{\xi}, s). \tag{2}$$

For $y(\cdot)$ and $x(\cdot)$ on the same mesh, the authors assume that the δ term defines a Gaussian process and specify the parameters in a Bayesian fashion. Qian and Wu (2008) generalized this approach by extending (2) and Qian et al. (2006) to

$$y(\boldsymbol{\xi}, s) = \rho(\boldsymbol{\xi}, s) \cdot x(\boldsymbol{\xi}, s) + \delta(\boldsymbol{\xi}, s) + \epsilon(\boldsymbol{\xi}). \tag{3}$$

In case of deterministic simulations, the ϵ term is not of particular interest. The authors refer to ρ as the scale adjustment and δ as the location adjustment. Both ρ and δ are assumed to be Gaussian processes.

In a recent contribution, Le Gratiet and Garnier (2014) showed that the autoregressive scheme proposed by Kennedy and O'Hagan (2000) can be reformulated in a recursive fashion if the datasets are nested, that is, if the high-fidelity inputs are a subset of the low-fidelity inputs. This paved the way for a nonlinear autoregressive approach proposed by Perdikaris et al. (2017), which is more data-efficient.

Biehler, Gee, and Wall (2014) used the Bayesian framework proposed by Koutsourelakis (2009) and arrived at the following model for a fixed location s:

$$y(\boldsymbol{\xi}, s) = f(x(\boldsymbol{\xi}, s); \boldsymbol{\theta}) + \sigma Z, \quad Z \sim N(0, 1). \tag{4}$$

Here, $f(x, \boldsymbol{\theta})$ is defined by a first-order polynomial and a sum of Gaussian kernel functions. It is important to note that only the approach put forth by Koutsourelakis (2009) is suitable for high-dimensional data. All other multifidelity approaches do not address this fundamental need, as they require the training of Gaussian process models in the dimension of the parameter space, that is, in the dimension of $\boldsymbol{\xi}$. A general review of how to relate low fidelity with high fidelity outputs can be found in, for example, Levy and Steinberg (2010), Peherstorfer, Willcox, and Gunzburger (2016), Giselle Fernández-Godino, Park, Kim, and Haftka (2019), and Biehler et al. (2019).

The limitations of the methods mentioned above are 2-fold: First, they require that the relation between $\boldsymbol{y}$ and $\boldsymbol{x}$ is calculated for all locations s, which poses a significant computational burden and is often infeasible if we are interested in quantities that require the complete vector $\boldsymbol{y}$, rather than just a specific location. Second, incorporating the spatial structure of the problem may be beneficial for the overall performance of the model. Both of these points apply in particular if the meshes S_x and S_y differ not only in their resolution, but also in their size, as seen in Figure 1. This leaves certain points on the high fidelity mesh that have no low fidelity counterpart at all. The major objective of this work is to therefore, overcome the focus on individual points and to model the relationship between all components of $\boldsymbol{x}$ and $\boldsymbol{y}$ simultaneously.

An alternative perspective is that of functional data analysis (FDA). The term was introduced in Ramsay (1982) and we refer to Ramsay (2006) for a comprehensive introduction. Morris (2015) contains a more recent survey of developments in this field. From this perspective, we consider $y(s)$ as functional data observed on the discrete mesh S_y, whose points are considered fixed locations on the continuum $\{\mathcal{S}\}_y$. Equivalently, we consider the low fidelity outcome $x(t)$ as data on the space $\{\mathcal{S}\}_x$

observed at the discrete points $t \in S_x$. To relate x to y corresponds to function-to-function regression in the terminology of functional data analysis. We refer to Greven and Scheipl (2017) for a recent discussion or Ettinger, Perotto, and Sangalli (2016) for regression models on manifolds. The general idea is to relate $x(t)$ and $y(s)$ through the functional regression model

$$y(s) = \mu_0(s) + \int x(t)B(t,s)dt + \epsilon(s), \tag{5}$$

where $B(t,s)$ is a regression function that needs further constraints to be identifiable. A common strategy in functional data analysis is to reduce the dimensionality of the data, for example, by eigendecomposition. We pursue a similar approach, but instead of eigenfunctions we make use of an assumed Gaussian random field dependence structure to reduce the dimension. Gaussian Markov random fields have already been used for compression, for example, for textures in Chellappa, Chatterjee, and Bagdazian (1985), and are a great fit for grid data in general due to its intrinsic neighborhood structure. We refer to Rue and Held (2005) for an in depth discussion on the topic and to MacNab (2018) for an overview of its applications. We want to stress out that we do not model the data as proper Gaussian Markov random fields but solely use this assumption for the construction of effective compression methods.

The key approach for relating $y(\cdot)$ and $x(\cdot)$ is to reduce the dimension of both $y(\cdot)$ and $x(\cdot)$ and to relate the reduced outcomes via regression. While dimension reduction of $y(\cdot)$ and $x(\cdot)$ can rely on a Gaussian Markov random field structure, the regression of the reduced dimensional objects still has a large number of parameters. To address the fact that the number of simulation runs is usually small, that is, only few independent observations are available, we construct an appropriate prior with a shrinkage parameter in order to make the problem solvable.

The combination of functional data and computer experiments has already appeared multiple times in the literature, often in combination with basis reduction. Morris (2012) used a kriging model to approximate computer experiments where both the input and the output are time dependent functions. Muehlenstaedt, Fruth, and Roustant (2017) and more recently Wang and Xu (2019) discussed the use of kriging models for experiments which have—not necessarily time dependent—functional input and also functional output in the latter case. Finally, there is also work by Chen et al. (2020) which presents a functional kriging model that connects three-dimensional structures to response curves using a spectral distance correlation function to quantify the similarity between inputs. However, none of these articles uses high and low fidelity outcomes but instead mostly focus on emulating the overall experiment—which, as we will show in the following section, is not realistic in our case. Moreover, it is arguable whether the results readily extend to connecting high and low fidelity outcomes in high dimensions with small samples. Finally, the application of kriging itself is not straightforward if challenged with a heavily underdetermined setting.

3. Problem Formulation

Before discussing the problem in statistical terms, we provide a brief description of the data and explain why the relation of high to low fidelity outcomes is of particular interest in this case. In this article we consider a nonlinear, patient-specific finite element model of an abdominal aortic aneurysm (AAA). The aneurysm is described by means of nonlinear solid mechanics and a finite element model is used to compute the deformations and stresses as a result of the applied blood pressure. For a detailed description the reader is referred to Biehler, Gee, and Wall (2014). One crucial aspect of the simulation is the constitutive model which characterizes the tissue and describes the relationship between stresses and strains. Aside from the functional form of the constitutive model it also depends on a number of so-called constitutive parameters. In biomedical simulations calculating deterministic values for these parameters is difficult because experimental data are scarce and because they vary, not only from patient to patient, but also over the geometry of individual aneurysms. Thus, researchers have hitherto predominantly resorted to using unspecific population averaged values, despite the fact that these and several other parameters are uncertain. A more realistic approach is to employ a probabilistic model for these parameters, as proposed in Biehler, Gee, and Wall (2014). In this work we consider one constitutive parameter for the wall of the aneurysm called $\boldsymbol{\beta}$ as a random field, which depends on a high dimensional (tuning) parameter vector $\boldsymbol{\xi}$. For each setting of $\boldsymbol{\xi}$ we obtain a simulation on the high and low fidelity mesh, that is, corresponding high and low fidelity simulations depend on the same $\boldsymbol{\xi}$ and can be considered as connected and comparable. We refer to the supplementary material for more details and an in depth description of the modeling background.

We then take a Bayesian point of view and impose a prior distribution on parameter $\boldsymbol{\xi}$ such that for the high fidelity simulation we obtain the marginal distribution

$$f(\boldsymbol{y}) = \int f(\boldsymbol{y}|\boldsymbol{\xi})dF(\boldsymbol{\xi}). \tag{6}$$

This approach allows to quantify the uncertainty induced by variation of the input vector $\boldsymbol{\xi}$. In practice one might want to solve (6) via Monte Carlo simulations, that is, computing $\boldsymbol{y}$ for various draws of $\boldsymbol{\xi}$. However, because evaluation of the model is very costly and computationally demanding, a large number of simulation runs renders this approach infeasible. Additionally, the high dimension of $\boldsymbol{\xi}$ also precludes the efficient use of statistical surrogate models or emulators, such as Gaussian processes or Polynomial Chaos Expansions (please see Le Maître and Knio 2010; Xiu 2010; Ghanem and Spanos 2012). Instead, we follow a multifidelity approach which makes use of data from a simplified, low fidelity version of the simulation model. The low fidelity model is based on a coarser, truncated version of the grid, derived from a computer tomography (CT) scan of the aneurysm. The so-called intraluminal thrombus (coagulated blood accumulating inside the AAA) is also omitted, to further reduce the computational cost of the model. This can be seen in Figure 1. The low fidelity version consists only of the vessel wall and is therefore, significantly less complex. In the presented case, there are no common points in both meshes and we denote the outcome of the simplified model with $\boldsymbol{x}(\boldsymbol{\xi}) \in \mathbb{R}^{N_x}$. We draw a small sample of parameter $\boldsymbol{\xi}$ and get $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_n$. It is worth noting that the models are deterministic and the vector $\boldsymbol{\xi}$ is sampled iid (as opposed to following a sampling scheme). Note that n is small, in the order of dozens or hundreds, and certainly too small to properly assess $f(\boldsymbol{y})$ given in (6). For each of the $\boldsymbol{\xi}_k$ we compute both $\boldsymbol{y}_{<k>} := \boldsymbol{y}(\boldsymbol{\xi}_k)$ and $\boldsymbol{x}_{<k>} := \boldsymbol{x}(\boldsymbol{\xi}_k)$ (we refer to Biehler, Gee, and Wall 2014). The task of the current article is to make use of this small sample to construct a prediction model for the high fidelity outcome $\boldsymbol{y}$ using the low fidelity input $\boldsymbol{x}$.

The ultimate goal of this project is to estimate $f(\boldsymbol{y})$ based on a larger Monte Carlo simulation study as shown in (6). But instead of running high fidelity simulations $\boldsymbol{y}(\boldsymbol{\xi})$ one runs many (fast) low fidelity simulations $\boldsymbol{x}(\boldsymbol{\xi})$ and makes use of the prediction model to obtain a prediction $\hat{\boldsymbol{y}}(\boldsymbol{\xi})$ from $\boldsymbol{x}(\boldsymbol{\xi})$. To do so the main objective is an accurate prediction model, which we provide in this article.

4. Formulation of the Model

In spatial statistics, a frequently used tool for modeling data on a grid is a Gaussian Markov random field (GMRF). This approach is applied here, and we refer to Rue and Held (2005) for a general discussion of the methodology. We use a GMRF assumption to compress both high and low fidelity outcomes in combination with basis truncation, which is common in spatial statistics. GMRFs are based on a predefined neighbor structure, as opposed to Euclidean distances. This neighborhood structure is determined by the mesh S_x for the low fidelity and S_y for the high fidelity data, as can be seen in Figure 2. Let us first look at the low fidelity data. We represent S_x as an undirected graph, that is, we define a tuple $G_x := (V_x, E_x)$, where V_x is the set of nodes in the graph and E_x the set of edges $\{i,j\} \in E_x = \Big\{\{i,j\} : i,j \in V_x\Big\}$, if $i \neq j$ are neighbors. Moreover, we write $i \sim j$ if both points are neighbors and $N(i)$ for the neighborhood of i, that is, $\{j \in V_x : j \sim i\}$. We consider $\boldsymbol{x}(\cdot)$ as a random vector $\boldsymbol{x} = \left(x(s_1), \ldots, x(s_{N_x})\right)^T$ evaluated at each of the N_x points of the mesh S_x. We assume that $\boldsymbol{x}$ follows the GMRF defined by

$$\pi(\boldsymbol{x}) = (2\pi)^{-n/2}|\mathbf{Q}_x|^{1/2} exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\mathbf{Q}_x(\boldsymbol{x}-\boldsymbol{\mu})\right). \quad (7)$$

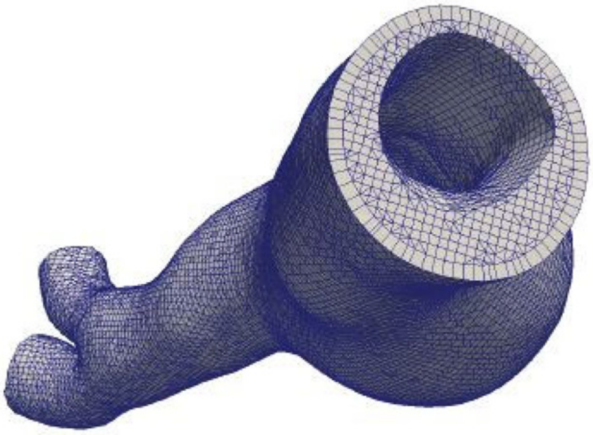

Figure 2. Zoomed in version of the high fidelity mesh.

The inverse variance matrix $\mathbf{Q}_x$ (also called the concentration matrix) has zero entries except for

$$\mathbf{Q}_{x,ij} = -1 \Leftrightarrow \{i,j\} \text{ in } E_x \text{ for } i,j \in V_x,$$
$$\mathbf{Q}_{x,ii} = |N(i)| \text{ for } i \text{ in } V_x.$$

In other words, $\boldsymbol{x}$ follows a multivariate Gaussian distribution parameterized by a thoughtfully chosen precision matrix $\mathbf{Q}_x = \boldsymbol{\Omega}_x^{-1}$.

It can be shown that for a single element $x_i = x(s_i)$ of $\boldsymbol{x}$, we obtain

$$E(x_i|\boldsymbol{x}_{-i}) = \boldsymbol{\mu}_i - \frac{1}{\mathbf{Q}_{x,ii}} \sum_{j \in N(i)} \mathbf{Q}_{x,ij}(x_j - \boldsymbol{\mu}_j), \quad (8)$$

where $\boldsymbol{x}_{-i}$ is the entire vector $\boldsymbol{x}$ except for the i-th entry. To be more specific, the ith value of $\boldsymbol{x}$ only depends on the neighbors of x_i in the mesh and is conditionally independent of the remaining values of $\boldsymbol{x}$. In the following we will compress $\boldsymbol{x}$ with the precision matrix. This is only possible if the mean vector $\boldsymbol{\mu}$ is equal to 0 which corresponds to using centered data. In practice, we therefore, center the data before the compression.

The idea is now to approximate $\boldsymbol{x}$ with the Karhunen–Loève theorem. To do so, we write

$$\boldsymbol{x} = \sum_{l=1}^{N_x} u_{x,l}\boldsymbol{\phi}_{x,l}, \quad (9)$$

where $u_{x,l}$ are uncorrelated random variables with zero mean and decreasing variance $\sigma^2_{x,l}$, that is, $u_{x,l} \sim N(0, \sigma^2_{x,l})$ and $\boldsymbol{\phi}_{x,l} = \left(\varphi_{x,l}(s_1), \varphi_{x,l}(s_2), \ldots, \varphi_{x,l}(s_{N_x})\right)^T$ are N_x dimensional vectors. We consider $\boldsymbol{\phi}_{x,l}$ as a basis on the mesh S_x. Note that we can rewrite the above formula pointwise such that

$$\boldsymbol{x}(s_i) = \sum_{l=1}^{N_x} u_{x,l}\boldsymbol{\phi}_{x,l}(s_i). \quad (10)$$

The structure of $\boldsymbol{\phi}_{x,l}$ can be approximated through singular value decomposition of the matrix $\mathbf{Q}_x$. Therefore, let $\boldsymbol{\phi}_{x,l}$ be defined as the eigenvector corresponding to the lth smallest eigenvalue of $\mathbf{Q}_x$ (or equivalently to the lth largest eigenvalue of $\mathbf{Q}_x^{-1}$).

We truncate the above expansion to the largest K_x eigenvectors of the variance matrix $\boldsymbol{\Omega}$ and set

$$\boldsymbol{x} = \sum_{l=1}^{K_x} u_{x,l}\boldsymbol{\phi}_{x,l} + \boldsymbol{\epsilon}_x, \quad (11)$$

where $\boldsymbol{\epsilon}_x \sim N(0, \sigma_x^2 \boldsymbol{I})$ is the white-noise component and $K_x < N_x$. This leads to the posterior distribution (see e.g., Harville 1976; Harville 1977; Harville and Zimmermann 1996)

$$\boldsymbol{u}_x|\boldsymbol{x} \sim N(\hat{\boldsymbol{u}}_x, \boldsymbol{\Sigma}_{u,x}), \quad (12)$$

with $\hat{\boldsymbol{u}}_x$ and $\boldsymbol{\Sigma}_{u,x}$ derived as follows. If we denote with $\boldsymbol{\phi}_x$ the truncated matrix $\boldsymbol{\phi}_x = \left(\boldsymbol{\phi}_{x,1}, \ldots, \boldsymbol{\phi}_{x,K_x}\right)$ and use the fact that $\boldsymbol{\phi}_{x,l}$ is orthonormal, we obtain

$$\hat{\boldsymbol{u}}_x = \left(\boldsymbol{I}_{K_x} + \sigma_x^{-2}\boldsymbol{\Lambda}_x^{-1}\right)^{-1} \boldsymbol{\phi}_x^T \boldsymbol{x}, \quad (13)$$

$$\begin{tikzcd}
x \arrow[r, "\phi_x^T"] & \hat{U}_x \arrow[r, "\text{regression}"] & \hat{U}_y \arrow[r, "\phi_y"] & \hat{y}
\end{tikzcd}$$

Figure 3. Complete Model Pipeline.

with $\boldsymbol{\Lambda}_x = \operatorname{diag}(\sigma^2_{x,1}, \ldots, \sigma^2_{x,K_x})$ denoting the diagonal matrix containing the K_x largest eigenvalues of $\boldsymbol{\Omega}$. Moreover, it holds that

$$\boldsymbol{\Sigma}_{u,x} = \left(\boldsymbol{\Lambda}_x^{-1} + \sigma_x^{-2}\boldsymbol{I}_{K_x}\right)^{-1}. \tag{14}$$

Note that with (13) we pursue a dimension reduction of the entire problem by compressing the N_x data points on the mesh $\boldsymbol{x}$ into K_x score values $\hat{\boldsymbol{u}}_x = (\hat{u}_{x,1}, \ldots, \hat{u}_{x,K_x})^T$.

This leads to a loss of information, which is expressed in σ_x^2 and can, for example, be estimated using restricted maximum likelihood (see Harville 1977, or Searle, Casella, and McCulloch 2006). The same formulas apply to an approximation of $\boldsymbol{y}$ and we obtain

$$\hat{\boldsymbol{u}}_y = \left(\boldsymbol{I}_{K_y} + \sigma_y^{-2}\boldsymbol{\Lambda}_y^{-1}\right)^{-1} \boldsymbol{\phi}_y^T \boldsymbol{y}. \tag{15}$$

Assume now that we have computed both the low and high fidelity simulations for a number of different input values $\boldsymbol{\xi}_k$, whose outcomes we denote with $\boldsymbol{y}_{<k>}$ and $\boldsymbol{x}_{<k>}$ for $k = 1, \ldots, n$. From both outcomes we calculate $\hat{\boldsymbol{u}}_{y<k>}$ and $\hat{\boldsymbol{u}}_{x<k>}$ using the formulas above. The intention is now to relate $\hat{\boldsymbol{u}}_{y<k>}$ to $\hat{\boldsymbol{u}}_{x<k>}$. To begin, we write $\hat{\boldsymbol{U}}_y = (\hat{\boldsymbol{u}}_{y<1>}, \ldots, \hat{\boldsymbol{u}}_{y<n>})^T$ and $\hat{\boldsymbol{U}}_x = (\hat{\boldsymbol{u}}_{x<1>}, \ldots, \hat{\boldsymbol{u}}_{x<n>})^T$. We may then relate the quantities through a linear model

$$\hat{\boldsymbol{U}}_y = \hat{\boldsymbol{U}}_x \boldsymbol{B} + \boldsymbol{\xi}, \tag{16}$$

where coefficient matrix $\boldsymbol{B}$ needs to be estimated from the n "data points." Using simple least squares we obtain:

$$\hat{\boldsymbol{B}} = \left(\hat{\boldsymbol{U}}_x^T \hat{\boldsymbol{U}}_x\right)^{-1} \hat{\boldsymbol{U}}_x^T \hat{\boldsymbol{U}}_y. \tag{17}$$

Note that matrix $\hat{\boldsymbol{U}}_x^T \hat{\boldsymbol{U}}_x$ is only of full rank if $K_x \le n$, which is not necessarily true if K_x is large. In fact, because high fidelity computation is time consuming, n is set to be small, which suggests that $n < K_x$ is actually more reasonable. This implies that we need to amend (17) to make it numerically solvable, which we achieve by imposing a Bayesian prior. Note that the rows of $\hat{\boldsymbol{U}}_x$ are $\hat{\boldsymbol{u}}_{x<k>}^T$, which is the posterior mean of $\boldsymbol{u}_x$ given $\boldsymbol{x}_{<k>}$. We assign a prior density for the coefficients in $\boldsymbol{B}$ which includes a shrinkage parameter g. In fact, for the lth column of $\boldsymbol{B}$ we assume

$$\boldsymbol{B}_l \sim N\left(0, g \cdot \boldsymbol{\Sigma}_{u,x}^{-1}\right), \tag{18}$$

where g is a weight and $\boldsymbol{\Sigma}_{u,x}$ is defined as in (14). The resulting posterior estimate for the lth column of $\boldsymbol{B}$ is then

$$\hat{\boldsymbol{B}}_l = \left(\hat{\boldsymbol{U}}_x^T \hat{\boldsymbol{U}}_x + g^{-1}\boldsymbol{\Sigma}_{u,x}\right)^{-1} \hat{\boldsymbol{U}}_x^T \hat{\boldsymbol{u}}_{y,l} \tag{19}$$

for $l = 1, \ldots, n$. Here $\hat{\boldsymbol{u}}_{y,l}$ denotes the lth column of $\hat{\boldsymbol{U}}_x$, that is, the coefficients of the lth eigenvector for all n low fidelity observations. The parameter g steers the stability, and its influence on the result tends to zero as it increases. It is chosen in a data-driven manner by minimizing the prediction error. Details will be provided in the next section. To sum up, we relate $\boldsymbol{x}$ to $\boldsymbol{y}$ with the steps shown in Figure 3.

Note that the calculation of both $\boldsymbol{\phi}_x$ and $\boldsymbol{\phi}_y$ does not require the full data. Instead, it uses the finite elements' neighborhood structure along with the Gaussian Markov random field assumption. Moreover, the data has to be centered, that is, strictly speaking we need estimates for the mean vectors of both high and low fidelity outcomes. The estimation of regression matrix $\boldsymbol{B}$, however, incorporates the information of the full data. As shown in the sketch above, it is important to keep in mind that the ultimate goal is the prediction of high fidelity from low fidelity output. This is indicated as $\hat{\boldsymbol{y}}$ and we aim to minimize the distance between $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$. In order to compare different models we split up our data into a train and test set.

A different modeling approach would be to apply a varying coefficient model, that is, adding a dependency of the regression coefficients $\boldsymbol{B}$ on the parameter vector $\boldsymbol{\xi}$. This dependency appears frequently in the literature—see for example Levy and Steinberg (2010)—and increases the flexibility of the model.

However, in our example we are very reluctant of pursuing this approach. First, $\boldsymbol{\xi}$, a vector of random angles, is used to sample the necessary parameters on the two grids with a complex algorithm. It does not enter the simulation itself, but really serves as a complex tuning parameter. Please see the supplementary material for a detailed description of this process. As a result an estimable and functional influence on $\boldsymbol{x}$ and $\boldsymbol{y}$ is unlikely. Second, the parameter $\boldsymbol{\xi}$ is high dimensional (order of several thousand) which makes the use of a varying coefficient model infeasible due to the dimensionality. One may then work with varying coefficient models combined with additive models in the style of Wood (2017), but even then we are still confronted with the dimensionality since in our case the number of additive components (dimension of $\boldsymbol{\xi}$) exceeds the number of observations (simulations) by far. All in all, small samples lead to numerical instability and large estimation variability for varying coefficient models and as a result we do not employ this approach in this work.

5. Application

5.1. Description of the Data

The data we consider in this work were generated using a patient-specific model of an abdominal aortic aneurysm. The specific model is described in Biehler, Gee, and Wall (2014). Further details regarding our methodology for the simulation of aneurysms using finite elements can be found in Reeps et al. (2010) and Maier et al. (2010). As described in Biehler, Gee, and Wall (2014), we created two versions of the aneurysm model, a high fidelity version and a computationally less expensive low fidelity version. The low fidelity model was created with a coarser discretization and with certain parts of the original model omitted. As a result, the mesh of the high fidelity model has $N_y = 84{,}889$ nodes, whereas the low fidelity mesh has $N_x = 3440$ nodes and as a consequence, the low fidelity model is almost 50 times faster to evaluate.

In total, both simulation models were run for $n = 900$ different samples of the parameter $\boldsymbol{\xi}$. First, a realization $\boldsymbol{\xi}_k$ was drawn from a proper prior model. Then both high and low fidelity outcomes $\boldsymbol{y}_{<k>}$ and $\boldsymbol{x}_{<k>}$ were computed using the in-house research code BACI (we refer to Wall et al. 2018). The result of the simulation are the displacements, strains and stresses for each point of the mesh under the prescribed load. In the ensuing sections, we consider the displacement magnitude for each point, as measured by the L_2 norm, to be the variable of interest.

Please note that we center the data by subtracting the means of a small training set of 10 additional pairs of high and low fidelity cases beforehand. The reason for this is that—as pointed out above—the GMRF compression assumes centered data.

5.2. Data Compression

The data compression process begins with the two precision matrices $\boldsymbol{Q}_x$ and $\boldsymbol{Q}_y$ that result from the assumed Gaussian Markov random field. Instead of first inverting the matrices and then computing their largest eigenvalues, we calculate the smallest eigenvalues and the corresponding eigenvectors of $\boldsymbol{Q}_x$ and $\boldsymbol{Q}_y$. This was done using the R package `RSpectra` by Xuan (2016) which is built upon the C++ package `Spectra`. Note that the eigenvectors, which can also be visualized, can be seen as the most important parts of the structure of the given geometry. The vectors corresponding to the four smallest eigenvalues of $\boldsymbol{Q}_x$ (which are also the largest ones of the variance matrix) are shown in Figure 4. As expected, the eigenvectors capture features of the geometry and also have a smoothing effect. Eigenvectors for $\boldsymbol{Q}_y$ show a similar picture and are therefore, not shown. However, it remains to determine the number of eigenvectors to include, that is, the dimensions of our bases $\boldsymbol{\phi}_x$ and $\boldsymbol{\phi}_y$. There are competing interests at play here. On the one hand, we want to keep the number of eigenvectors small in order to minimize the computational costs and maximize compression of the data. On the other hand, most of the information contained in the outcomes should be preserved in order to later construct accurate high fidelity outcomes. As a result, K_x and K_y will be tuning parameters for our proposed framework. We also briefly look at the compression accuracy and define the L_2 compression error of the low fidelity outcome (and analogously the high fidelity outcome) for a given number of eigenvectors as follows

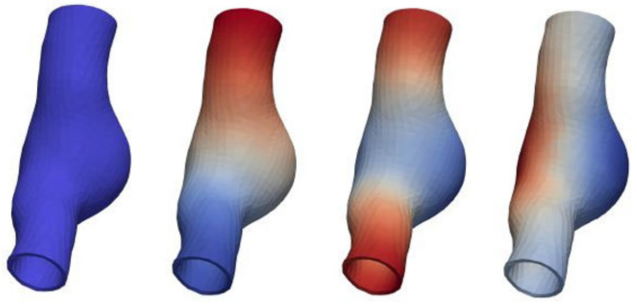

Figure 4. First four eigenvectors for the coarse grid.

$$e_{\text{comp}}(K_x) = \frac{\sum_{k=1}^{n} \left\| \boldsymbol{x}_{<k>} - \boldsymbol{\phi}_x \hat{\boldsymbol{u}}_{x<k>} \right\|}{\sum_{k=1}^{n} \|\boldsymbol{x}_{<k>}\|} = \frac{\sum_{k=1}^{n} \left\| \boldsymbol{x}_{<k>} - \boldsymbol{\phi}_x \left(\boldsymbol{I}_{K_x} + \sigma_x^{-2}\boldsymbol{\Lambda}_x^{-1}\right)^{-1} \boldsymbol{\phi}_x^T \boldsymbol{x}_{<k>} \right\|}{\sum_{k=1}^{n} \|\boldsymbol{x}_{<k>}\|}, \tag{20}$$

where basis $\boldsymbol{\phi}_x$ is K_x dimensional, which is omitted in the notation for simplicity, and $\|.\|$ denotes the Euclidean distance. Note that, in order to get a useful metric, we not only sum up the L_2 norm differences for our given sample but also normalize by the sum of the L_2 norms across all vectors in our sample, which gives the percentile of lost information.

Before we take a look at the results, let us briefly describe the estimation of σ_x^2. As outlined in the previous section, we want to estimate σ_x^2 in an iterative way. As a starting point, we use the unbiased standard estimate of σ_x^2 in an ordinary linear regression model, that is model (11) with a flat prior. We can then use this estimate to solve formula (13) and reestimate σ_x^2 with the result. Doing so shows that σ_x^2 hardly changes. This is to be expected, as we use the same eigenvectors in both models and should therefore, get similar errors. The error e_{comp} defined in (20) for the low fidelity outcomes decreases when we use more eigenvectors. While the first few eigenvectors account for most variability, later ones only slightly improve the approximation. In the end, we get a curve that asymptotically approaches an error equal to zero, which is attained with $K_x = N_x$. On the left of Figure 5, we plot the compression error and its counterpart, the preserved information, for the low fidelity simulations. Judging from the plot, $K_x = 500$ is a reasonable number of eigenvectors for the compression of the low fidelity outcomes. The eigenvectors of the high fidelity outcomes give similar results, where the error again decreases nearing $K_y = N_y$. In this case, $K_y = 2000$ seems reasonable.

5.3. Connecting High and Low Fidelity Outcomes

As outlined in Section 4, the Bayesian regression model in (16) and (18) will be used to connect the compressed outcomes $\hat{\boldsymbol{U}}_x$ and $\hat{\boldsymbol{U}}_y$. When combined with both projection steps, this gives us a consolidated model that connects the low and high fidelity results $\boldsymbol{x}$ and $\boldsymbol{y}$. To prepare the data we divide the $n = 900$ computed pairs into a training and test sample. The regression model is fitted with the training samples and a key question is then the influence of training sample size on prediction accuracy. Note that this number plays an interesting role for two different reasons. First, we showed that an ordinary linear model connecting $\hat{\boldsymbol{U}}_x$ and $\hat{\boldsymbol{U}}_y$ is in general not solvable and a g-prior is needed. Second, avoiding the high fidelity simulation process

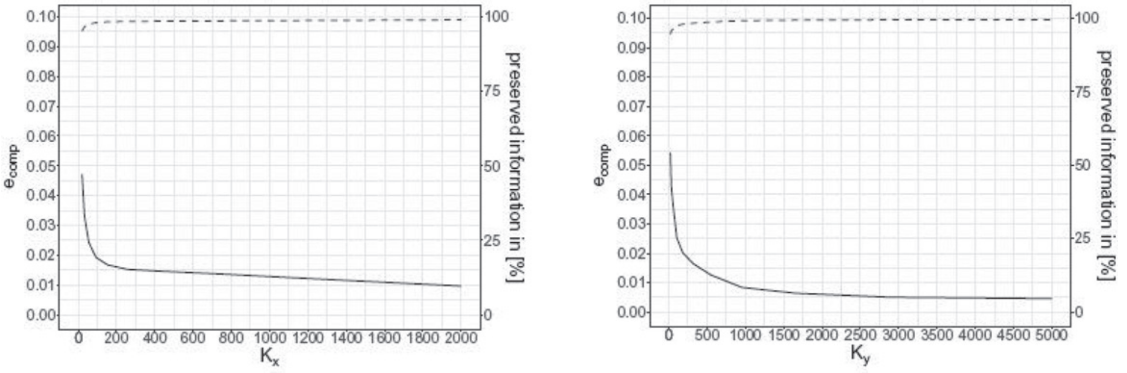

Figure 5. L_2 compression errors for the low fidelity outcome (left) and high fidelity outcome (right).

is the prime objective and reduction in sample size means a corresponding reduction in computation. In this case, a low fidelity simulation takes about 215 CPU seconds while a high fidelity simulation requires approximately 10,800. On the test sample we calculate the L_2 prediction error with

$$e_{\text{pred}}(K_x, K_y, g) = \frac{\sum_{k=1}^{n_{\text{test}}} \left\| \boldsymbol{y}_{<k>} - \hat{\boldsymbol{y}}_{<k>} \right\|}{\sum_{k=1}^{n_{\text{test}}} \left\| \boldsymbol{y}_{<k>} \right\|} = \frac{\sum_{k=1}^{n_{\text{test}}} \left\| \boldsymbol{y}_{<k>} - \boldsymbol{\phi}_y \left(\hat{\boldsymbol{u}}_{x<k>}^T \hat{\boldsymbol{B}} \right) \right\|}{\sum_{k=1}^{n_{\text{test}}} \left\| \boldsymbol{y}_{<k>} \right\|}$$
$$= \frac{\sum_{k=1}^{n_{\text{test}}} \left\| \boldsymbol{y}_{<k>} - \boldsymbol{\phi}_y \left(\left(\left(\boldsymbol{I}_{K_x} + \sigma_x^{-2} \boldsymbol{\Lambda}_{\boldsymbol{x}}^{-1} \right)^{-1} \boldsymbol{\phi}_x^T \boldsymbol{x}_{x<k>} \right)^T \hat{\boldsymbol{B}} \right) \right\|}{\sum_{k=1}^{n_{\text{test}}} \left\| \boldsymbol{y}_{<k>} \right\|}. \tag{21}$$

Different models arise for different tuning parameters, the most important of which are K_x and g. The number of eigenvectors used to compress $\boldsymbol{y}$, K_y, is kept to $K_y = 2000$, a sensible number as is shown on the right of Figure 5. This leaves us with three parameters. To explore this in a reasonable manner, we keep at least one parameter constant while investigating the different parameter combinations.

We start by exploring the error in relation to g and K_x. In order to do so, we fix the size of the training data to 50, that is, we use only 50 pairs of high and low fidelity outcomes to fit the model. For these 50 samples we vary K_x and g and compute the prediction error for the remaining 850 simulation results, which is shown in Figure 6. Since g is a multiplicative constant, we will plot it on a logarithmic scale. When observing Figure 6 the prediction error forms a band. It takes a minimum value at $g = 10^{-4}$ and increases for both smaller and larger values of g. This is unsurprising as the g-prior simply needs to be just large enough to make the model solvable. Even though we only used 50 cases to build the model there are no difficulties in estimating an accurate model with a substantially higher number of eigenvectors for the low fidelity outcome. For example for $K_x = 500$ the matrix $\boldsymbol{B}$ consists of one million coefficients. The error gets at first smaller as we use more and more eigenvectors. For $K_x \geq 100$ the benefit of adding more eigenvectors is close to zero. We repeat the above procedure for 25 and 100 cases and get the following results shown in Figure 7.

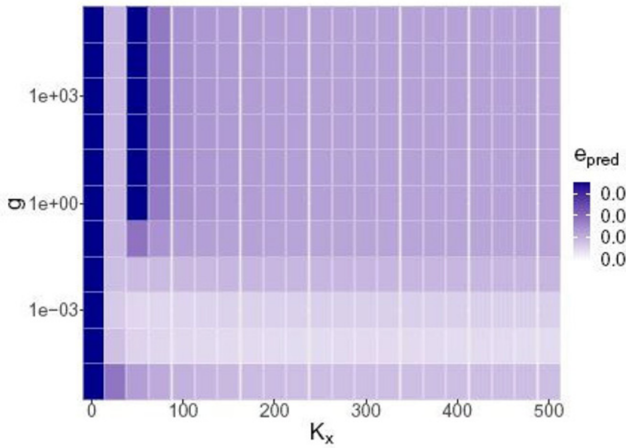

Figure 6. Average L_2 prediction error for different numbers of eigenvectors K_x and different values g for the g-prior specification using 50 training cases.

Both tile plots exhibit the same structure as the previous figure. The prediction error again takes a minimal value for $g = 10^{-4}$.

Next, in order to understand the influence of the size of the training data, we look at different values of K_x but set g to its optimal value. This gives the prediction error which is plotted in Figure 8 against K_x for different sizes of the training data. Our analysis shows that the information contained in the low fidelity outcome needs a reasonable number of eigenvectors and cases to allow for a good approximation of $\boldsymbol{y}$. Moreover, there is an asymptotic limit of the information that is contained in the sample, which gets approached for around 200 eigenvectors. Finally, we look at the observation $\boldsymbol{y}$ and the prediction $\hat{\boldsymbol{y}}$ for one randomly chosen example in the test data. This example was for $K_x = 200$, $K_y = 2000$, optimal g and based on 50 samples in the training data. The results are visualized in Figure 9. We also plot the pointwise error, that is the simple difference between the real and predicted value for each point on the grid to make any systematic problems clearer. This is particularly useful as we are dealing with high dimensional spatial data and the overall error defined in (21) therefore, incorporates a significant loss of information. It turns out that it is quite difficult to visually distinguish the real and the computed displacement. Nevertheless, we can see that the error is larger for the lower part of the geometry that is completely missing in the low fidelity

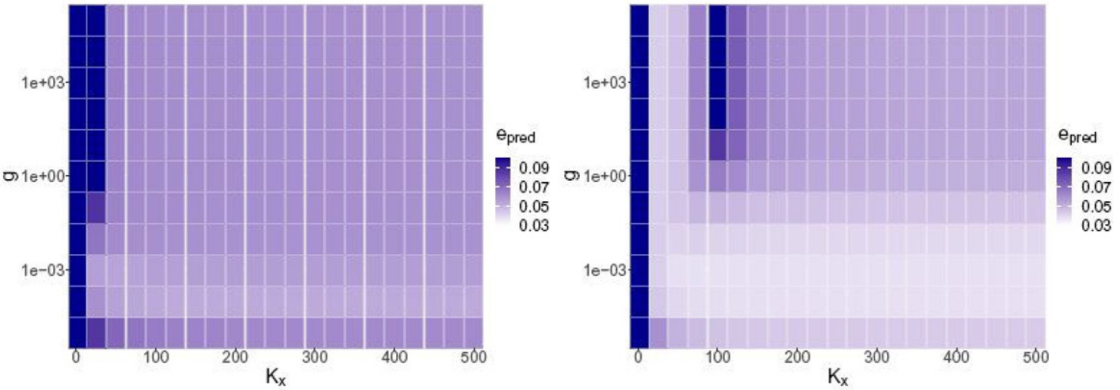

Figure 7. Average L_2 prediction error for 25 (left) and 100 training cases (right).

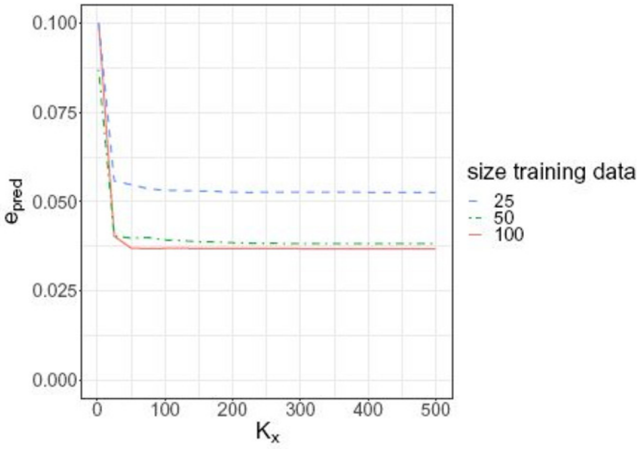

Figure 8. Average L_2 prediction error for different numbers of eigenvectors and different sample sizes.

model and at the back where the thrombus is missing. Overall the results of the method are satisfying and errors are well within reasonable bounds. Reproducing these plots for various other test cases shows the same result and is therefore, omitted.

We also plot in Figure 10 the mean of the pointwise absolute error in order to avoid selection bias. We can clearly see that the general spatial distribution of the error follows the same pattern as in the case of the single exemplary outcome.

5.4. Comparison to Alternatives

5.4.1. Unadjusted Low Fidelity Model

The question remains whether the predicted high fidelity outcomes are in fact a significant improvement over the raw low fidelity observations. In this context, it is crucial to stress that the low fidelity grid is not only thinner and truncated but is also discretized differently. In other words, no low fidelity points exist that directly correspond to a given high fidelity point. Additionally, our ultimate goal is not the accurate prediction of individual high fidelity realizations, but the accurate assessment of ensemble statistics of the high fidelity model output under uncertainty. However, we can still do a reasonable comparison for the small set of points that are, up to a very small error, part of both geometries. In order to do so, we first compute the relative distance between the coordinates of all points on both grids and select 80 pairs that have a value lower than 0.0002. These points are, as Figure 11 shows, evenly distributed over the entire geometry.

We use a tuned GMRF based model, as proposed in Section 4, with a training set of 50 pairs of observations and $K_x = 200$ for our comparison. The average absolute difference between the predicted high fidelity outcome and the real high fidelity outcome over all the points and test samples is 0.01 in contrast to 0.058 for the difference to the low fidelity outcome. In other words, the proposed model provides a great improvement over the raw low fidelity outcomes.

We can further plot the real high fidelity outcomes for the 80 selected points and the 850 test cases against both our predicted values and the low fidelity outcomes as show in Figure 12.

Again it appears that the predicted values are much closer to the real outcomes than the unadjusted low fidelity ones.

5.4.2. Principal Component Basis

A straightforward alternative to the proposed framework is to substitute the GMRF basis with principal component vectors. This approach has been used frequently, for example in Higdon et al. (2008), and has proven to be highly efficient. Moreover, Mak et al. (2018) showed that principal components can also capture physical features of the model. To be specific, in the following we examine a model similar to the one proposed in Figure 3, but with the important difference that we substitute the eigenvectors and eigenvalues of the GMRF covariance matrix with the principal components and eigenvalues of the training set, which was previously only used for regression. More formally, we substitute the compression matrices $\boldsymbol{\phi}_x$ and $\boldsymbol{\phi}_y$ with $\boldsymbol{\phi}_x^{\mathrm{PCA}}$ and $\boldsymbol{\phi}_y^{\mathrm{PCA}}$, the principal components of the low and high fidelity outcomes in the training set. Analogously, we replace the eigenvalue matrices $\boldsymbol{\Lambda}_x$ and $\boldsymbol{\Lambda}_y$ with $\boldsymbol{\Lambda}_x^{\mathrm{PCA}}$ and $\boldsymbol{\Lambda}_y^{\mathrm{PCA}}$, which consist of the eigenvalues corresponding to the principal components. It is important to stress that we now also use the training set for the compression process, while previously it only played a role in the overall regression of the compressed outcomes. We will omit the specific formulas for the model derivation, as they follow directly from Section 4 and will only state the more relevant error terms. First, we adapt the formula for the compression error in (20) by substituting the $\boldsymbol{\phi}$ and $\boldsymbol{\Lambda}$ matrices and only summing over the test set. This is necessary, because the compression now also depends on the

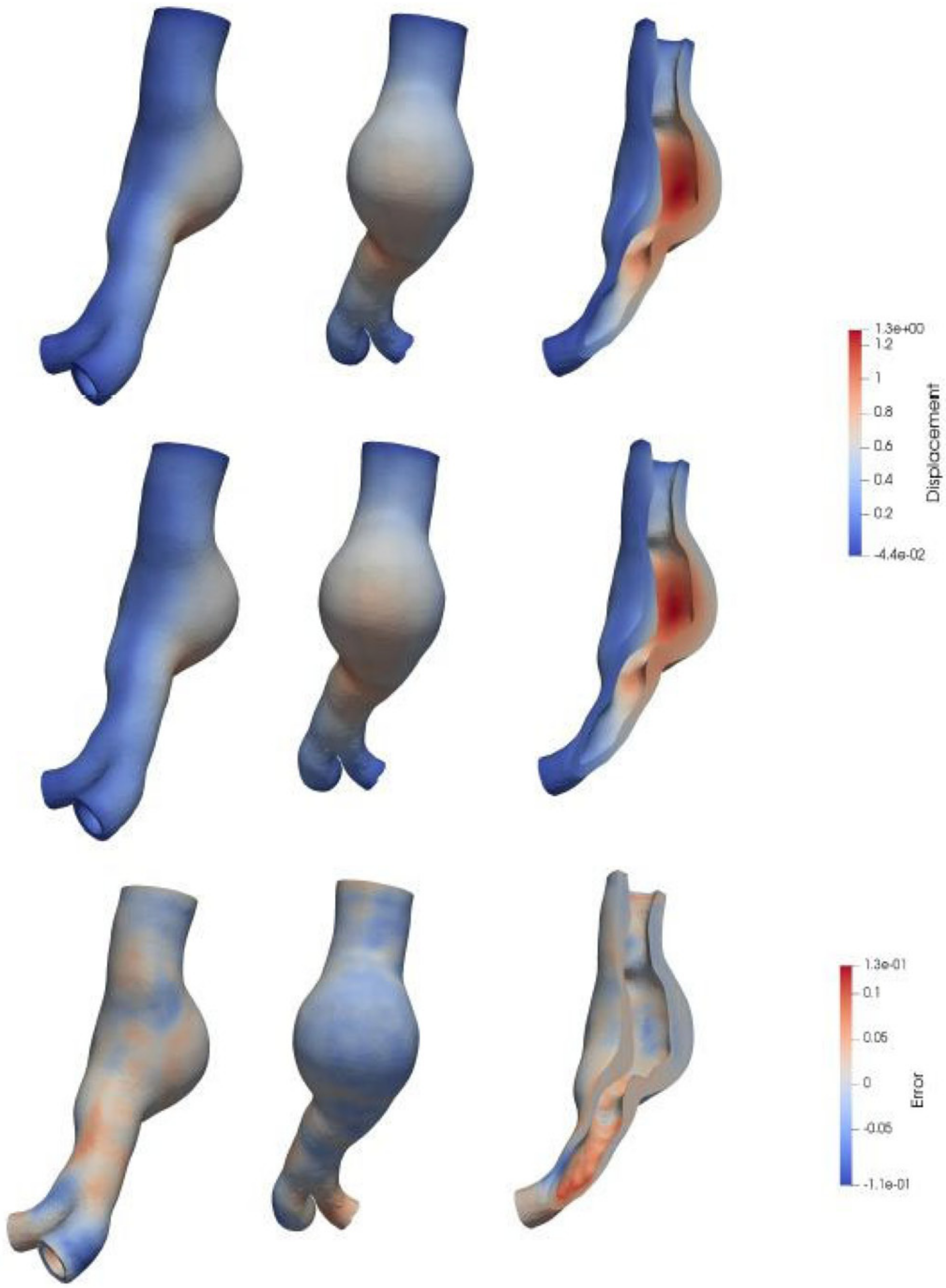

Figure 9. From top to bottom: Observation, prediction, and pointwise error (original, rotated, and opened) for a single test case.

training sample and should therefore, be evaluated exclusively on the test set. The size of the basis K_x is now limited form above by the size of the training set n_{train}:

$$e_{\text{comp}}(K_x) = \frac{\sum_{k=1}^{n_{\text{test}}} \left\| \boldsymbol{x}_{<k>} - \boldsymbol{\phi}_x^{\text{PCA}} \hat{\boldsymbol{u}}_{x<k>} \right\|}{\sum_{k=1}^{n_{\text{test}}} \|\boldsymbol{x}_{<k>}\|} = \frac{\sum_{k=1}^{n_{\text{test}}} \left\| \boldsymbol{x}_{<k>} - \boldsymbol{\phi}_x^{\text{PCA}} \left(\boldsymbol{I}_{K_x} + \sigma_x^{-2} \left(\boldsymbol{\Lambda}_x^{\text{PCA}}\right)^{-1} \right)^{-1} \left(\boldsymbol{\phi}_x^{\text{PCA}}\right)^T \boldsymbol{x}_{<k>} \right\|}{\sum_{k=1}^{n_{\text{test}}} \|\boldsymbol{x}_{<k>}\|}. \quad (22)$$

To derive the new prediction error from (21), it is sufficient to substitute the $\boldsymbol{\phi}$ and $\boldsymbol{\Lambda}$ matrices:

$$e_{\text{pred}}(K_x, K_y, g) = \frac{\sum_{k=1}^{n_{\text{test}}} \left\| \boldsymbol{y}_{<k>} - \boldsymbol{\phi}_y^{\text{PCA}} \left(\left(\left(\boldsymbol{I}_{K_x} + \sigma_x^{-2} \left(\boldsymbol{\Lambda}_x^{\text{PCA}}\right)^{-1} \right)^{-1} \boldsymbol{\phi}_x^T \boldsymbol{x}_{x<k>} \right)^T \hat{\boldsymbol{B}} \right) \right\|}{\sum_{k=1}^{n_{\text{test}}} \|\boldsymbol{y}_{<k>}\|}. \quad (23)$$

Using these formulas, we can compare the compression errors on the test set for PCA and GMRF based approaches. We are mostly interested in small sample sizes and restrict ourselves for brevity to the more difficult compression of high fidelity outcomes. The benchmark is the GMRF compression used for our regression models in the previous parts, that is, a model that uses $K_y = 2000$ eigenvectors. As shown in Figure 13 the compression error depends heavily on the number of cases used

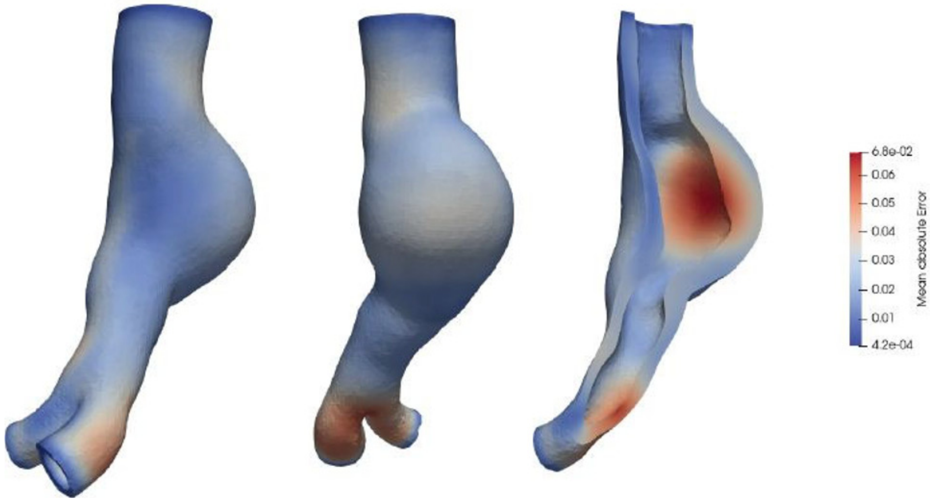

Figure 10. Mean absolute error for the predictions in the test set (original, rotated, and opened).

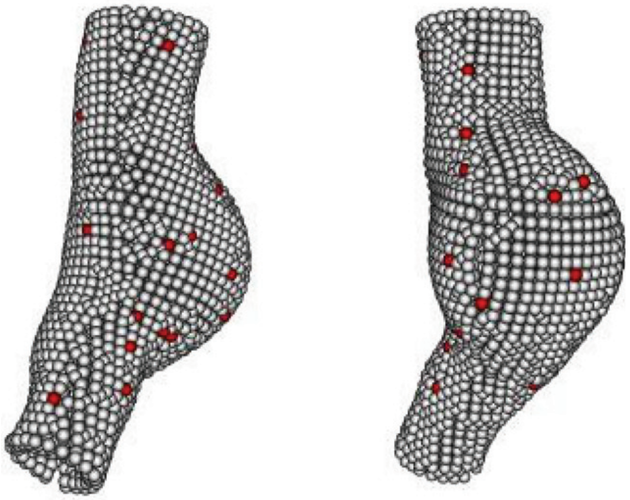

Figure 11. Location of the 80 points for the comparison on the low fidelity grid.

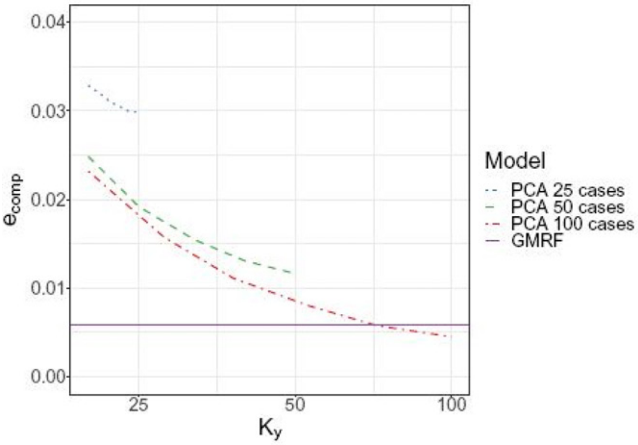

Figure 13. Compression error of the high fidelity outcomes for GMRF and PCA bases.

for the PCA, that is, the size of the training set. Moreover, the GMRF approach, where we can easily increase the number of vectors into the hundreds, has an edge over the PCA approach as long as the number of high and low fidelity pairs is below 100. If we fit an overall prediction model for high fidelity outcomes based on PCA compression and look at the error as defined in (23), we get a prediction error of 0.087 for 25 pairs of cases in the training set and 0.052 for 50 pairs of cases, both significantly higher than the average error for similar GMRF models. Even for 100 PCA vectors the error is slightly worse than the error of our GMRF approach. The PCA approach is probably superior

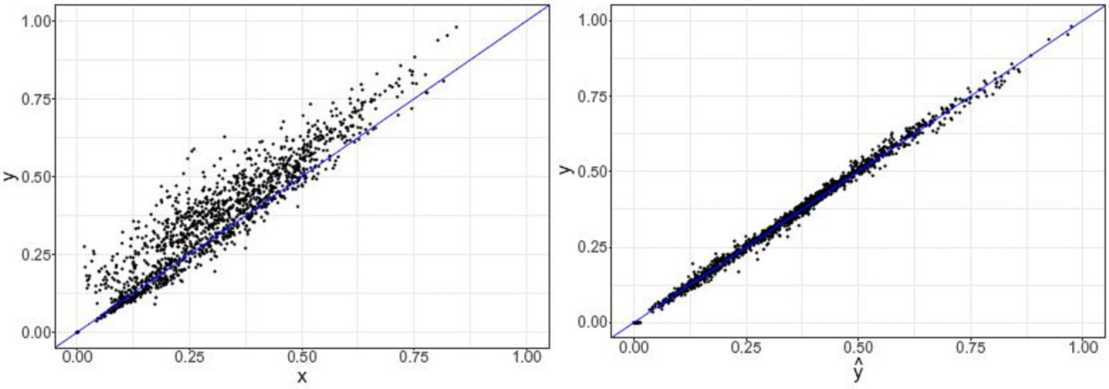

Figure 12. Low fidelity outcome and predicted high fidelity outcome versus real high fidelity outcome for 80 points and 850 samples in the test set.

Table 1. Comparison of the overall prediction error for GMRF and PCA based models.

n	GMRF	PCA
25	0.023	0.087
50	0.015	0.052
100	0.014	0.020

for even larger samples. However, this is problematic as we need to compute more high fidelity outputs and these are, as the next section shows, costly. We can therefore, conclude that the GRMF approach presented in this work has an edge for small samples, when only a few dozen model runs are available (Table 1).

5.5. Numerical Details

When we focus our attention on the numerical effort of our proposed function-to-function regression, the most demanding part is the computation of the eigenvectors. On our 32 GB memory workstation with 16 Intel Xeon 2.40GHz CPU cores, the computation of the 5000 eigenvectors for the high fidelity grid on one core took approximately 36 hr or 129,600 sec. The computation of the 2000 eigenvectors for the low fidelity grid took less than half an hour or 1800 sec. However, these computations need only be performed once and also only depend on the mesh structure. In other words, if the meshes for calculating the arterial wall pressure remain the same for two different patients, a recalculation of bases $\boldsymbol{\phi}_x$ and $\boldsymbol{\phi}_y$ is not required. The compression of the observations, that is, the estimation of $\boldsymbol{u}_x$ and $\boldsymbol{u}_y$ takes negligible time (about half a second per data entry in the training dataset). The same holds for the estimation of the Bayesian regression model (16). In fact, all 231 models which had to be generated for Figure 6 were computed in less than 30 min on 5 cores, that is, 9000 CPU seconds. The computation of a single high fidelity outcome takes approximately 10,800 CPU seconds, whereas the computation of a low fidelity one takes a mere 215 CPU seconds. The main cost factors are the evaluations of the high fidelity model, which in part determine the error as seen in Figure 8. The benefit of our framework therefore, depends on the acceptable error and the number of high fidelity outcomes that are needed. If the number of required outcomes is large we can replace the serial computation of high fidelity simulations with a smaller number for model fitting and scale down the computation time by orders of magnitude.

6. Discussion

The presented work provides a framework for computing a large number of high fidelity computer simulation outputs on a mesh if a low fidelity approximation is available. It is possible to compute approximations for all points of the high fidelity mesh, not only for a handful of selected locations, even if there is no corresponding location on the low fidelity grid. At the same time, the relative error we obtain with these predictions is reasonably small and can be adjusted for a specific application via the number of eigenvectors and the size of the training data, which have to be balanced against an increase in computation time. There are various possible extensions to the model, for example choosing a different neighborhood structure for the GMRF to better mimic mechanical forces or choosing a more involved regression model to connect the compressed outcomes.

Supplementary Materials

Code and Data: `repro_code.zip` is a compressed folder that contains code and data to reproduce the results on a small subset of the data used in the article. The full dataset is available at: *https://syncandshare.lrz.de/getlink/fi2soSUsR5PzuLnN6Rx8kZ6L/*.

Probabilistic Material Model: `probabilistic_material_model.pdf` contains further details on the finite element simulation with a focus on the modeling of wall parameters.

References

Biehler, J., Gee, M. W., and Wall, W. A. (2014), “Towards Efficient Uncertainty Quantification in Complex and Large-Scale Biomechanical Problems Based on a Bayesian Multi-Fidelity Scheme,” *Biomechanics and Modeling in Mechanobiology*, 14, 489–513. [279,280,281,282,283]

Biehler, J., Mäck, M., Nitzler, J., Hanss, M., Koutsourelakis, P.-S., and Wall, W. A. (2019), “Multifidelity Approaches for Uncertainty Quantification,” *GAMM-Mitteilungen*, 42, e201900008. [280]

Chellappa, R., Chatterjee, S., and Bagdazian, R. (1985), “Texture Synthesis and Compression Using Gaussian-Markov Random Field Models,” *IEEE Transactions on Systems, Man, and Cybernetics*, 15, 298–303. [281]

Chen, J., Mak, S., Joseph, V. R., and Zhang, C. (2020), “Function-on-Function Kriging, with Applications to Three-Dimensional Printing of Aortic Tissues,” *Technometrics*, 63, 384–395. [281]

Ettinger, B., Perotto, S., and Sangalli, L. M. (2016), “Spatial Regression Models Over Two-Dimensional Manifolds,” *Biometrika*, 103, 71–88. [281]

Ghanem, R., and Spanos, P. (2012), *Stochastic Finite Elements: A Spectral Approach*, New York: Springer. [281]

Giselle Fernández-Godino, M., Park, C., Kim, N. H., and Haftka, R. T. (2019), “Issues in Deciding Whether to Use Multifidelity Surrogates,” *AIAA Journal*, 57, 2039–2054. [280]

Greven, S., and Scheipl, F. (2017), “A General Framework for Functional Regression Modelling,” *Statistical Modelling: An International Journal*, 17, 1–35. [281]

Harville, D. (1976), “Extension of the Gauss-Markov Theorem to Include the Estimation of Random Effects,” *The Annals of Statistics*, 4, 384–395. [282]

———, D. A. (1977), “Maximum Likelihood Approaches to Variance Component Estimation and to Related Problems,” *Journal of the American Statistical Association*, 72, 320–338. [282,283]

Harville, D. A., and Zimmermann, A. G. (1996), “The Posterior Distribution of the Fixed and Random Effects in a Mixed-Effects Linear Model,” *Journal of Statistical Computation and Simulation*, 54, 211–229. [282]

Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008), “Computer Model Calibration Using High-Dimensional Output,” *Journal of the American Statistical Association*, 103, 570–583. [286]

Kennedy, M., and O’Hagan, A. (2000), “Predicting the Output from a Complex Computer Code When Fast Approximations are Available,” *Biometrika*, 87, 1–13. [280]

Koutsourelakis, P.-S. (2009), “Accurate Uncertainty Quantification Using Inaccurate Computational Models,” *SIAM Journal on Scientific Computing*, 31, 3274–3300. [280]

Le Gratiet, L., and Garnier, J. (2014), “Recursive Co-Kriging Model for Design of Computer Experiments with Multiple Levels of Fidelity,” *International Journal for Uncertainty Quantification*, 4, 365–386. [280]

Le Maître, O., and Knio, O. M. (2010), *Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics*, Luxemburg: Springer. [281]

Levy, S., and Steinberg, D. M. (2010), “Computer Experiments: A Review,” *AStA Advances in Statistical Analysis*, 94, 311–324. [280,283]

MacNab, Y. C. (2018), “Some Recent Work on Multivariate Gaussian Markov Random Fields,” *Test*, 27, 497–541. [281]

Maier, A., Gee, M. W., Reeps, C., Pongratz, J., Eckstein, H. H., and Wall, W. A. (2010), “A Comparison of Diameter, Wall Stress, and Rupture

Potential Index for Abdominal Aortic Aneurysm Rupture Risk Prediction," *Annals of Biomedical Engineering*, 38, 3124–3134. [283]

Mak, S., Sung, C.-L., Wang, X., Yeh, S.-T., Chang, Y.-H., Joseph, V. R., Yang, V., and Wu, C. F. J. (2018), "An Efficient Surrogate Model for Emulation and Physics Extraction of Large Eddy Simulations," *Journal of the American Statistical Association*, 113, 1443–1456. [286]

Morris, J. S. (2015), "Functional Regression," *Annual Review of Statistics and its Application*, 2, 321–359. [280]

Morris, M. D. (2012), "Gaussian Surrogates for Computer Models with Time-Varying Inputs and Outputs," *Technometrics*, 54, 42–50. [281]

Muehlenstaedt, T., Fruth, J., and Roustant, O. (2017), "Computer Experiments with Functional Inputs and Scalar Outputs by a Norm-based Approach," *Statistics and Computing*, 27, 1083–1097. [281]

Peherstorfer, B., Willcox, K., and Gunzburger, M. (2016), "Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization," Technical Report TR16-1, MIT. [280]

Perdikaris, P., Raissi, M., Damianou, A., Lawrence, N. D., and Karniadakis, G. E. (2017), "Nonlinear Information Fusion Algorithms for Data-Efficient Multi-Fidelity Modelling," *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 473, 20160751-17. [280]

Qian, P. Z., and Wu, C. J. (2008), "Bayesian Hierarchical Modeling for Integrating Low-Accuracy and High-Accuracy Experiments," *Technometrics*, 50, 192–204. [280]

Qian, Z., Seepersad, C. C., Joseph, V. R., Allen, J. K., and Wu, C. F. J. (2006), "Building Surrogate Models based on Detailed and Approximate Simulations," *Journal of Mechanical Design*, 128, 668–677. [280]

Ramsay, J. O. (1982), "When the Data are Functions," *Psychometrika*, 47, 379–396. [280]

——— (2006), *Functional Data Analysis*, New York: Springer. [280]

Reeps, C., Gee, M. W., Maier, A., Gurdan, M., Eckstein, H. H., and Wall, W. A. (2010), "The Impact of Model Assumptions on Results of Computational Mechanics in Abdominal Aortic Aneurysm," *Journal of Vascular Surgery*, 51, 679–688. [283]

Rue, H., and Held, L. (2005), *Gaussian Markov Random Fields: Theory and Applications*, London: Chapman & Hall/CRC. [281,282]

Searle, S. R., Casella, G., and McCulloch, C. E. (2006), *Variance Components*, New York: Wiley. [283]

Wall, W. A., Kronbichler, M., Ager, C., and Grill, M. (2018). "BACI: A Parallel Multiphysics Simulation Environment," Technical report, Institute for Computational Mechanics, Technische Universität München. [284]

Wang, B., and Xu, A. (2019), "Gaussian Process Methods for Nonparametric Functional Regression with Mixed Predictors," *Computational Statistics and Data Analysis*, 131, 80–90. [281]

Wood, S. (2017), *Generalized Additive Models: An Introduction with R*. Chapman & Hall / CRC Texts in Statistical Science, Boca Raton, FL: CRC Press. [283]

Xiu, D. (2010), *Numerical Methods for Stochastic Computations: A Spectral Method Approach*, Princeton, NJ: Princeton University Press. [281]

Xuan, Y. (2016), "Rspectra," available at *https://github.com/yixuan/RSpectra/*. [284]