Analyzing the Influence of Prior Covariances on a Bayesian Finite Element Method

by

Student: Uri Peker

Faculty of Civil Engineering and Geosciences
Delft University of Technology

Thesis committee:	Dr. F. P. van der Meer	TU Delft
	Dr. I. B. C. M. Rocha	TU Delft
	Dr. H. N. Kekkonen	TU Delft
	ir. A. Poot	TU Delft
Student Number:	5520649	

Abstract

Probabilistic numerics methods are a novel approach to quantifying the approximation errors in numerical computations as probabilistic uncertainties. A recent method that was developed is the Bayesian Finite Element Method, which aims to determine the discretization errors along a coarse mesh probabilistically. This work analyzes the use of priors in this method. It is shown that the priors in the right-hand side or forcing term of the partial differential equation are superior to applying them directly in the solution. It is demonstrated that the maximum log-likelihood is not an appropriate estimator of hyperparameters, and as solutions, hyperparameters are optimized using objective functions to capture the discretization error with the posterior deviation. The advantages of non-stationary priors are studied in order to have standard deviations that show the error along the mesh. Additionally, this work examines how the optimal hyperparameters change for mesh refinements and different arrangements of elements in the same problem. Finally, the research delves into determining the approximate number of samples necessary for ensembling covariance matrices and obtaining similar results.

Contents

List of Figures

List of Tables

Nomenclature

$\delta(x-x')$	Dirac function
L	Likelihood function
$\boldsymbol{\Delta}$	Vector of discretization errors
$\boldsymbol{\mathcal{D}}$	Domain
$\boldsymbol{\Gamma}$	Boundary of the domain
$\hat{\boldsymbol{u}},\ \hat{\boldsymbol{\Sigma}}$	Posterior mean and posterior covariance
$\boldsymbol{\mathcal{L}},\ \boldsymbol{\mathcal{B}}$	integro-differential operators
$\boldsymbol{\mu}(\boldsymbol{X})$	Mean vector
$\boldsymbol{\Phi}$	Mapping matrix of fine and coarse mesh
$\boldsymbol{\Sigma}(\boldsymbol{X},\boldsymbol{X})$	Covariance matrix
$\boldsymbol{\mathcal{U}},\ \boldsymbol{u}$	Space of trial functions and trial functions
$\boldsymbol{\mathcal{V}},\ \boldsymbol{v}$	Space of test or weight functions and test or weight functions
$\boldsymbol{b}$	Force vector of coarse space
$\boldsymbol{f}$	Force vector of fine space
$\boldsymbol{H}$	Stiffness matrix from Petrov-Galerkin
$\boldsymbol{K}$	Fine mesh stiffness matrix
$\boldsymbol{L}$	Cholesky decomposition of covariance matrix
$\boldsymbol{M}$	Fine mesh unit mass matrix
$\boldsymbol{N},\ \boldsymbol{Q}$	Shape functions of fine space and Shape functions of coarse space
$\boldsymbol{u_f},\ \boldsymbol{u_c}$	Fine and coarse mesh solutions
$\boldsymbol{u_s},\ \hat{\boldsymbol{u}}_{\boldsymbol{s}}$	Sample prior solution y sample posterior solution
$\boldsymbol{z}$	Vector of values sampled from a normal distribution
$k(x,x')$	Kernel function
$m(x)$	Mean function
n_c	Number of degrees of freedom of coarse mesh
n_f	Number of degrees of freedom of fine mesh

1

Introduction

1.1 Literature review

Mathematical modeling and computer simulation are suitable for civil and industrial applications. For instance, car crashes, fluid movement in petroleum reservoirs, and high-velocity airflow around space shuttles. Often, physical problems are idealized and represented with ordinary differential equations (ODEs) or partial differential equations (PDEs). Most of these equations do not have an analytical solution. Therefore, numerical analysis techniques have been developed to find approximate solutions for the equations that govern each problem. One of the most common is the finite element method (FEM). In this method, the weak form of the PDE is derived from the strong form, after which the domain is discretized into a mesh of finite elements. This reduces the infinite-dimensional problem into a finite-dimensional one. In the beginning, FEM was mainly used for linear solid mechanics. Nowadays, it is used for much more complex problems, for example, nonlinear geometrical and material analyses, multi-physics, and multiscale modeling.

The previously mentioned problems are known as *forward* problems because the input of the model and the parameters are known and a result or output is obtained. In contrast, *inverse* problems involve finding the parameters of the problem when the input and output are known. FEM is also used in inverse problems, which involve adjusting the parameters of the FEM to make the simulated results match the observed data. An example of an inverse problem in civil engineering where FEM is used is the pile integrity test (PIT) [1]. This method is popular as a quality check because of the possibility of finding cracks, soil inclusion, bulking, or stretching along the pile. To perform the test, the pile is hammered at the top (input), and the acceleration (output) is measured close to the hammered area. A 1D wave equation can be solved using FEM to obtain the expected solution of the top acceleration as a function of the hammered force and the assumed parameters, soil properties, pile area, and length. By comparing the expected acceleration on the top and with the measured, an engineer can easily detect the discontinuities due to the wave reflection on the discontinuity position. One approach to tackle this inverse problem involves utilizing FEM to approximate the solution of the PDE. The goal is to minimize the discrepancy between the measured signal and the numerical solution by optimizing the problem parameters. For instance, parameters like the area and length of a potential anomaly at a specific point in the pile need to be optimized. However, obtaining these optimal parameters requires significant computational effort since each solution involves solving a wave equation.

Even with the improvement of computers, solving some of the aforementioned forward and inverse problems can still take weeks or even months due to the large number of equations involved. To cope with this adversity, one solution is to use a coarse mesh to reduce the number of elements and thus the number of equations. Nevertheless, this solution may not be adequate since it leads to a loss of accuracy in the solution due to the increase of discretization errors compared with a fine mesh. Furthermore, discretization error can be a significant source of uncertainty in inverse Bayesian problems, as it can affect the accuracy and reliability of the forward model used to generate synthetic or simulated data. By obtaining a probabilistic discretization error along the mesh, it is possible to account for this source of uncertainty and include it in the overall uncertainty quantification for the inverse problem.

Several studies have been done to properly estimate the discretization errors in FEM. Based on the regularity of the exact solution, the polynomial order of interpolation, and the size of the mesh, an a priori estimate was developed to calculate the asymptotic rate of convergence [2]. This approach is not practical in real-world scenarios since the exact solution is typically known only in academic examples. In contrast, an a posteriori error estimate in the energy norm was elaborated [3]. In this method an approximation of the potential energy is estimated with the use of a sequence of mesh refinements; then the error can be calculated in the energy norm by comparing the potential energy of the mesh with the estimated one. These procedures have proved their utility in simple problems and have been extended to adaptive meshes [4]. However, they just calculate the global error without being able to show how the error is distributed along the domain. To address this limitation, many estimators have been developed. One example is the work by Zienkiewicz and Zhu, who developed an estimator that compares the gradients calculated by the FEM model with recovered gradients [5, 6]. These error estimators do not account for the uncertainty of the data and parameters, and they do not allow the incorporation of prior information or perform sensitivity analysis of the error.

Recently, a new numerical formulation has been developed in which the error is treated as a source of uncertainty, giving birth to *probabilistic numerics methods* (PNMs) [7]. In these methods the data can be treated stochastically and sensitivity analysis of the uncertainty can be performed. There are many examples where numerical analysis is combined with probability theory. For instance, O'Hagan created a Bayesian quadrature [8]. Instead of using low-order polynomials to approximate the function to integrate, as in the Newton-Cotes method, or using orthogonal Legendre polynomials, as Gaussian quadrature does, O'Hagan applied a Bayesian inference with a prior Gaussian process to approximate the function. Each integration point is seen as an observation used to improve the posterior that is integrated. Another example of a PNM is the work by Cockayne *et al.* [9]. In numerical analysis, solving large systems of equations using iterative solvers is a common task. Cockayne *et al.* developed a Bayesian conjugate gradient that obtains a probabilistic estimate of the solution, showing the error as an uncertainty.

Cockayne *et al.* [10] proposed a clear distinction between two setups of PNMs and Bayesian PNMs. In the first one numerical computations include sources of uncertainty in the ODEs or PDEs, and the problems are solved without using observed data. As a result, instead of obtaining one result as in a traditional numerical computation, a distribution of the solution is obtained. In the second one, observed data or evidence are included to improve the knowledge of the solution distribution. In this thesis, both sets of problems will be referred to as PNMs, whether or not they include evidence or observations to obtain a final solution.

Until now PNMs methods have been developed to solve different problems such as linear algebra, global and local optimization, numerical integration, and ODEs and PDEs [11]. In the case of the solutions for ODEs and PDEs, there have been different approaches developed [12–17]. The works [12–14] can be categorized as collocation problems. Chkrebtii *et al.* [12] applied the randomness in the ODE itself rather than in the solver. In this formalism, the solution and the derivatives that are involved in the ODE are assumed uncertain with a statistical distribution and update in each step of the process based on the previous values. Wang *et al.* [13] generalized the work of Chkrebtii for non-linear PDEs by locally linearizing the differential operations. Cockayne *et al.* [14] used a Bayesian approach to evaluate governing equations at finite locations of interest in a domain, which means that the PDE is strongly evaluated. In contrast to these methods, Conrad *et al.* [15] worked on randomized solutions of ODEs with Euler and Runge–Kutta methods, altering the numerical solution with a Gaussian noise that varies according to the size of the discretization.

In the context of FEM, Girolami *et al.* [16] developed a technique called *statFEM*, which treats the PDE as a stochastic problem by considering the diffusion coefficient and the forcing term as random sources modeled as a Gaussian process. Due to the linearity of the integro-differential operator in the PDE, the solution also follows a Gaussian process. By discretizing the domain of the SPDE into finite elements, a system of equations can be obtained, and the solution is represented as a multivariate Gaussian distribution with a mean vector and a covariance matrix. This process serves as the prior solution in the Bayesian inference framework, where the available observed data is used to obtain a posterior solution. One of the major drawbacks when using FEM to solve SPDEs is that it requires the inversion of stiffness matrices. In cases where the dimensionality of the problem is high, the computational cost can be prohibitive. Akyildiz *et al.* [17] use the unadjusted Langevin algorithm to obtain sample solutions of *statFEM*, without inverting the typical FEM stiffness matrices. This leads to a sampler that can be used to solve the problem of integrating over the uncertainties, and also call forward the problem of uncertainty quantification.

Recently, Poot *et al.* [18] developed a Bayesian finite element method (BFEM) to find a probabilistic distribution of the discretization errors along a mesh. In BFEM, a Petrov-Galerking scheme is applied [19], where the space of test functions is a subspace of the trial functions space, which leads to an underconstrained problem with an infinite set of solutions. The underconstrained system can be regularized by assuming a prior distribution over the solution or forcing the term of the SPDE. This prior should take into account the boundary conditions imposed on the system. The test function belongs to a space with a mesh fine enough to assume that the solution is approximately the true solution, while the test or weight function belongs to a fine mesh on which we want to obtain the discretization error. Within a Bayesian framework, the posterior distribution is derived after updating the prior distribution with the forcing term of the coarse mesh solution used as observational data. The posterior mean obtained should represent the solution in a fine mesh, and the deviation should represent the error due to discretization along the mesh. This method has shown suitable results for smooth problems with simple geometries, where the standard deviation of the solution qualitatively shows the discretization error.

1.2 Knowledge gap

The prior utilized in the work of Poot *et al.* [18] is found by assuming a random white noise source on the right-hand side (RHS) term of the PDE and integrating it after discretizing the domain. The same methodology is used by Owhadi [20], who developed a Bayesian framework to obtain the basis functions of a numerical homogenization problem. The difference between both approaches is that in Owhadi's work, the domain is not discretized, being a collocation problem. Poot *et al.* obtain a unit mass matrix as a prior covariance matrix after discretizing and integrating the random forcing term. Moreover, Roininen *et al.* [21] used a similar technique to obtain numerical approximations of Whittle-Matérn priors using finite-difference for structured meshes and FEM for unstructured meshes, reducing the computations needed to solve a large inverse problem since the covariance matrices become sparse. Lastly, Nguyen *et. al* [22] weakly treats the PDE and the knowledge of the physical system is improved by finding a distribution over the functionals. By this weak treatment of the PDE Nguyen *et. al* end up obtaining priors that are a combination of unit mass and unit stiffness matrices. In BFEM, no comparison has yet been made between priors in the left-hand side (LHS) or solution and the RHS or forcing term demonstrating the advantages of using the RHS.

The analytical derivation to obtain the hyperparameter of the white noise prior in Poot *et al.* work is done by using the maximum likelihood estimator (MLE). Simultaneously, they suggested treating Dirichlet and Neumann boundary conditions as random variables rather than deterministic values, thus enriching the prior distribution with noise. Nonetheless, it has not been shown if the posterior deviation obtained is quantitatively representative of the discretization error and if the MLE is a suitable estimate to find the optimal hyperparameter. In this thesis, we will demonstrate that the MLE does not provide representative results and that the white noise prior can be improved instead of adding noise by proposing non-stationarity functions.

Due to the significant computational cost associated with calculating prior and posterior covariances, Poot *et al.* utilized a Kalman ensemble [23, 24] to approximate these matrices through sampling. Furthermore, the required number of samples to accurately capture the posterior distribution in BFEM has not been investigated. Lastly, the impact of different mesh configurations on the ability of posterior deviations that represent discretization error has not been tested.

1.3 Thesis objectives and research questions

The main objective of this work is to study how covariance matrices and their hyperparameters influence BFEM to show discretization errors with posterior deviations. Therefore, the analyses conducted are focused on the prior matrices, hyperparameter estimation, the sensitivity of hyperparameters to modifications in the mesh, and how posterior deviation changes when the prior covariance matrix is an ensemble with samples. For each topic the research questions are:

- Optimal hyperparameters and error representation:
  - Is the maximum log-likelihood estimator a suitable way to obtain the optimal value of the hyperparameters to demonstrate the error as a source of uncertainty along the mesh?
  - What is the best approach to compare different kernel or covariance matrix priors quantitatively and qualitatively, to represent the error along the mesh effectively?
- Priors covariances:
  - Is placing a prior on the right-hand side or forcing term considered better than one on the solution or left-hand side?
  - Can the use of non-stationary covariance functions improve the distributions obtained by BFEM, particularly to show the discretization error?
- Mesh impact:
  - Do the optimal hyperparameters change between mesh refinements?
  - Does something similar happen if the arrangement of the mesh elements is also modified?
- Impact of the number of samples in ensemble covariance matrices:
  - In the case of using sampling to assemble covariance matrices, what is the range of samples that is necessary to obtain distributions showing an error similar to the one obtained by the matrix without sampling?

1.4 Structure of the thesis

The structure of the thesis is as follows, Chapter 2 will review the mathematical theory needed to understand the BFEM, the method itself, and the mathematical tools used during the thesis. Chapter 3 shows what are the common choices of prior, the differences between placing uncertainty directly in the solution or in the forcing term, whether the MLE is a good estimator, and how to compare different priors. In Chapter 4 a comparison between stationary and non-stationary priors is developed, in Chapter 5 we study how the results are modified with different mesh refinements, different element arrangements, and sampling. Finally, in Chapter 6 the conclusions of the work are made and directions for future work are discussed to answer new questions that arose throughout the thesis.

2

Theory

2.1 Gaussian Processes

Stochastic processes are used to represent systems that evolve randomly in time or space. Moreover, a stochastic process can be understood as a random distribution over functions. In particular, we are interested in Gaussian processes (GPs), which can be thought of as a generalization of the Gaussian probability density distribution [25]. A random variable is a scalar distribution, or in the case of a multivariate distribution, a vector. In contrast, the process is a distribution over a function. A function is represented by a mean function $m(x)$ and its covariance function $k(x, x')$.

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \tag{1}$$

Since a property of GPs is that any finite selection of points follows a multivariate normal distribution and there are an infinite number of locations to analyze in the process, a finite number of values are selected to represent the domain, denoted as $\boldsymbol{X}$. This transforms the problem into a multivariate Gaussian distribution, where each point in $\boldsymbol{X}$ represents a variable with a corresponding mean and covariance.

$$\boldsymbol{f}(\boldsymbol{X}) \sim \mathcal{N}\left(\boldsymbol{\mu}(\boldsymbol{X}), \boldsymbol{\Sigma}(\boldsymbol{X}, \boldsymbol{X})\right) \tag{2}$$

To obtain samples of a GP, the same method for any multivariate Gaussian is used. Firstly, a sample is obtained from a normal distribution with a mean vector of zero and an identity matrix as the covariance matrix. This can be expressed as

$$\boldsymbol{z} \sim N\left(\boldsymbol{0}, \boldsymbol{I}\right) \tag{3}$$

Secondly, the $\boldsymbol{z}$ vector is transformed into the vector $\bar{\boldsymbol{f}}$ by using the Cholesky decomposition of the covariance matrix $\boldsymbol{\Sigma}$, which can be expressed as:

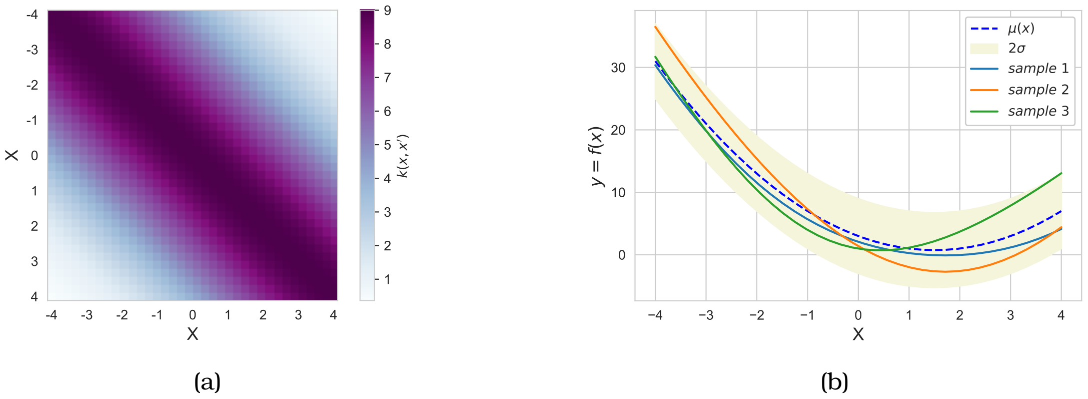
$$\bar{\boldsymbol{f}} = \boldsymbol{L}\boldsymbol{z} + \boldsymbol{m} \tag{4}$$

Here, $\boldsymbol{L}$ is the lower triangular matrix obtained from the Cholesky decomposition of $\boldsymbol{\Sigma}$, and $\boldsymbol{m}$ is the mean vector of the prior distribution.

An example of what a covariance matrix based on the square exponential kernel of Eq. (5) looks like is shown in Fig. 1a. To obtain this matrix, a vector $\boldsymbol{X}$ was generated with 64 equally spaced positions between $[-4, 4]$.

$$k(x, x') = \sigma^2 e^{-\frac{(x-x')^2}{2l^2}} \tag{5}$$

In addition, a polynomial mean function, the deviation, and three samples are plotted in Fig. (1b).



(a)

(b)

Figure 1: (a) Example of square exponential covariance matrix with $\sigma = 3$ and $l = 10$. (b) Example of a stationary Gaussian process with $\mu(x) = x^2 - x^3 + 3$, exponential covariance matrix with $\sigma = 3$ and $l = 10$, and domain $[-4, 4]$.

2.2 Kernel properties

A kernel function is called *stationary* as long as the relationship between the points $\boldsymbol{x}$ and $\boldsymbol{x'}$ depends only on their distance and not on a given position. This means that the difference between the positions of the points can be substituted by a variable that denotes the distance. For example, in eq. (5), the difference between the positions can be substituted by $d = x - x'$. Then, the square exponential kernel function will look like

$$k(d) = \sigma^2 e^{-\frac{d^2}{2l^2}} \tag{6}$$

In many cases, the stationary property is problematic because it is not able to show amplitude or length scale variations along the problem domain. In these cases, it is more useful to use *non-stationary* kernel functions. An example of this is the polynomial kernel function, which increases as the position moves away from $x = 0$:

$$k(x, x') = (\sigma^2 + xx')^p \tag{7}$$

If the covariance matrix in Fig. (1a) is compared with the covariance matrix in Fig. (2a), it can be observed that the former has no variation along the diagonal. In contrast, the values of the latter matrix change along the diagonal. Moreover, if the process shown in Fig. (1b) is repeated with the kernel function changed to a polynomial function, it is possible to notice how the variation increases at the extremes when moving away from $x = 0$, as shown in Fig. (2b).

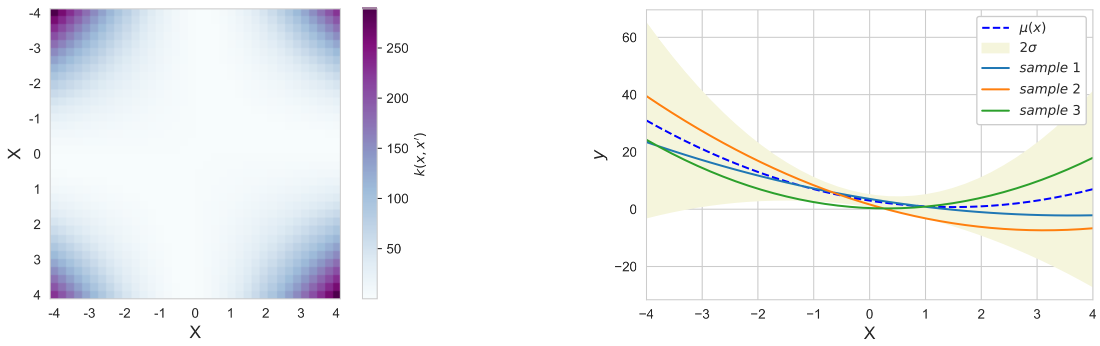
If the kernel has more than one dimension, then it will be *isotropic* if the level of correlation is the same in each direction.

$$k(x, x', y, y') = \sigma^2 e^{-\frac{(x-x')^2}{2l_1^2}} e^{-\frac{(y-y')^2}{2l_2^2}} \tag{8}$$

In contrast, it is called *anisotropic* if its behavior changes depending on the direction. For example, Equation (8) is an anisotropic kernel because it has different length scales in each direction ($l_1 \neq l_2$).

Based on input points $\boldsymbol{x} \in \mathrm{S}$, the kernels are evaluated to form a covariance matrix that has a symmetric and positive semi-definite form, meaning that $k(\boldsymbol{x}, \boldsymbol{x'}) = k(\boldsymbol{x'}, \boldsymbol{x})$ and $\sum_i^N \sum_j^N a_i a_j k(x_i, x'_j) \geq 0 \; \forall \; N, \boldsymbol{x} \in \mathbf{S}, \boldsymbol{a} \in \mathrm{IR}^N$ [26].

A GP can be subjected to integro-differential operators, allowing predictions of its derivatives. To do this, kernels have to be subjected to the linear operators twice if it has two inputs $\mathcal{L}_{\boldsymbol{x}} \mathcal{L}_{\boldsymbol{x'}} [k(\boldsymbol{x}, \boldsymbol{x'})]$.



(a)

(b)

Figure 2: (a) Example of polynomial covariance matrix with $\sigma = 1$ and $p = 2$. (b) Example of a non-stationary Gaussian process $\mu(x) = x^2 - x^3 + 3$ and polynomial covariance matrix with $\sigma = 1$, $p = 2$, and domain $[-4, 4]$.

2.3 Maximum log-likelihood estimator

The likelihood is a function that measures how probable a given data is to correspond to certain parameters. Therefore, the maximum likelihood will have the best parameters to make the distribution fit the data and is computed by multiplying the probability density function (PDF) across the data points. This approach assumes that the values used in the likelihood function are independent of each other. The likelihood function can be expressed as:

$$\mathrm{L}(\boldsymbol{X}|\boldsymbol{\theta}) = \prod_{i=1}^{n} p(X_i, \boldsymbol{\theta}) \tag{9}$$

Where X_i represents the observed data, $\boldsymbol{\theta}$ is the parameter vector, and $p(X_i, \boldsymbol{\theta})$ is the probability density function. Maximizing the log-likelihood is often preferred over maximizing the likelihood directly because it is simpler, and the obtained values are analogous. The Log-likelihood is

$$\ln\left(\mathrm{L}(\boldsymbol{X}|\boldsymbol{\theta})\right) = \sum_{i=1}^{n} \ln(p(X_i, \boldsymbol{\theta})) \tag{10}$$

and the partial derivatives are

$$\frac{\partial(\ln\left(\mathrm{L}(\boldsymbol{X}|\boldsymbol{\theta})\right))}{\partial\theta_k} = \sum_{i=1}^{n} \frac{\partial \ln(p(X_i, \boldsymbol{\theta}))}{\partial\theta_k} \tag{11}$$

In certain cases, the MLE can be found analytically by setting the partial derivatives of the log-likelihood function to zero, resulting in a system of equations to be solved. However, not all problems have straightforward analytical solutions. In such instances, is necessary to use mathematical optimization methods. These numerical techniques efficiently search for the maximum of the log-likelihood function by iteratively updating the parameter values until convergence is achieved.

An essential consideration is the convexity of the log-likelihood function. If the log-likelihood function is convex, it has a single global maximum, simplifying the optimization process. Standard optimization algorithms can efficiently find this global maximum. Nonetheless, if the log-likelihood function is not convex and exhibits multiple local maxima or points of inflection, finding the global maximum becomes more challenging.

2.4 Linear and Gaussian processes regression

A linear regression model is introduced, with the linear regression defined as:

$$y = \boldsymbol{\alpha}^T \boldsymbol{X} + \boldsymbol{\varepsilon} \tag{12}$$

where $\boldsymbol{X}$ is the input vector, y is the observed target value, $\boldsymbol{\alpha}$ is the weights vector, and ε is the additive noise, which can be assumed with a Gaussian distribution:

$$\varepsilon \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\right) \tag{13}$$

With n as the number of observed data points $(y_i, \boldsymbol{X}_i)$ the objective is to find the optimal values of $\boldsymbol{\alpha}$ that minimize the error function, usually measured using the mean square error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{\alpha}^T \boldsymbol{X}_i)^2 \tag{14}$$

The minimum MSE can be obtained by using an analytical ordinary least squares solution or numerical optimization algorithms such as gradient descent.

Another approach is the Bayesian formulation of linear regression, which allows us to assume a distribution for the weights and take into account any prior information we may have. For instance, it is common to assume a Gaussian distribution for the weights:

$$\boldsymbol{\alpha} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_p\right) \tag{15}$$

This prior distribution of the weights can be updated by using the Bayes' rule:

$$p(\boldsymbol{\alpha}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})}{\displaystyle\int p(\boldsymbol{y}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})d\boldsymbol{\alpha}} \tag{16}$$

The probability density distribution $p(\boldsymbol{\alpha})$ is the prior belief of the parameters, and the likelihood of the parameters is defined as $p(\boldsymbol{y}|\boldsymbol{\alpha})$ and the evidence or marginal likelihood is $\int p(\boldsymbol{y}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})d\boldsymbol{\alpha}$.

Unlike parametric models that assume a specific functional form with a fixed number of parameters, GPs learn the form by fitting to the data, which makes them more flexible than linear or polynomial regressions [27].

To make a GP regression function evaluated on the vector of points $\boldsymbol{X}_2$ based on the observed points $(\boldsymbol{X}_1, \boldsymbol{y}_1)$, we can assume a prior mean vector $\boldsymbol{\mu}(\boldsymbol{X}) = 0$ and a prior covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{X}, \boldsymbol{X})$. Then, the problem can be expressed as a multivariate Gaussian:

$$\begin{pmatrix}\boldsymbol{y}_1\\ \boldsymbol{y}_2\end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix}\mathbf{0}\\ \mathbf{0}\end{pmatrix}, \begin{pmatrix}\boldsymbol{\Sigma}(\boldsymbol{X}_1, \boldsymbol{X}_1) & \boldsymbol{\Sigma}(\boldsymbol{X}_1, \boldsymbol{X}_2)\\ \boldsymbol{\Sigma}(\boldsymbol{X}_2, \boldsymbol{X}_1) & \boldsymbol{\Sigma}(\boldsymbol{X}_2, \boldsymbol{X}_2)\end{pmatrix}\right) \tag{17}$$

A conditional distribution of $\boldsymbol{y}_2$ given $\boldsymbol{y}_1$ and the observed inputs $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ can be expressed as:

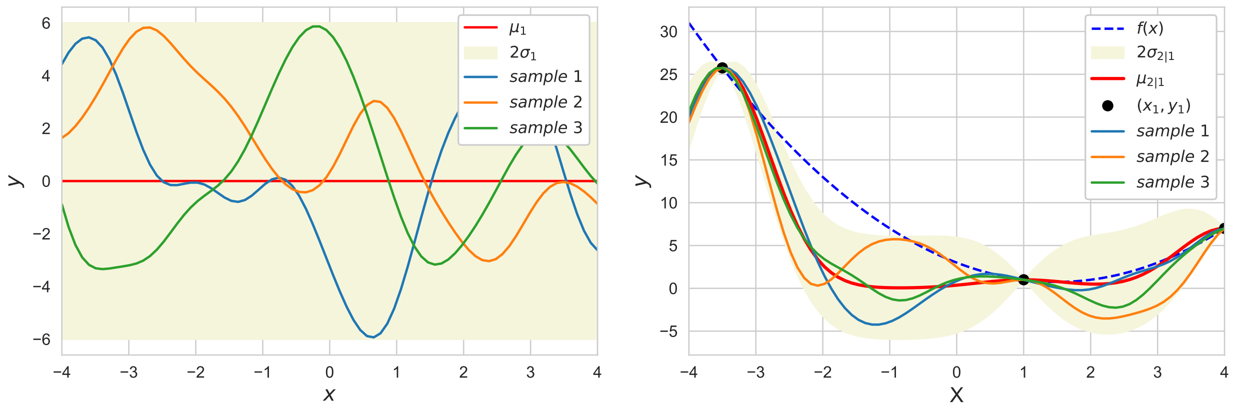
$$p(\boldsymbol{y}_2|\boldsymbol{y}_1, \boldsymbol{X}_1, \boldsymbol{X}_2) = \mathcal{N}\left(\boldsymbol{\mu}_{2|1}, \boldsymbol{\Sigma}_{2|1}\right) \tag{18}$$

where the posterior mean and posterior covariance are obtained as:

$$\boldsymbol{\mu}_{2|1} = \boldsymbol{\Sigma}(\boldsymbol{X}_2, \boldsymbol{X}_1)\boldsymbol{\Sigma}(\boldsymbol{X}_1, \boldsymbol{X}_1)^{-1}\boldsymbol{y}_1 \tag{19}$$

$$\boldsymbol{\Sigma}_{2|1} = \boldsymbol{\Sigma}(\boldsymbol{X}_2, \boldsymbol{X}_2) - \boldsymbol{\Sigma}(\boldsymbol{X}_2, \boldsymbol{X}_1)\boldsymbol{\Sigma}(\boldsymbol{X}_1, \boldsymbol{X}_1)^{-1}\boldsymbol{\Sigma}(\boldsymbol{X}_1, \boldsymbol{X}_2) \tag{20}$$

To clarify these concepts an example will be provided. Fig. (3a) shows a prior $f(x)$ with zero mean and a square exponential kernel. Moreover, this estimation is improved in Fig. (3b) by obtaining the posterior estimate based on 3 points that are known from $f(x)$. As can be seen, the posterior mean passes through the 3 points where the data are known and the deviation is zero. This deviation changes away from the known data.



(a) (b)

Figure 3: (a) Example of prior samples with $\mu(x) = 0$ and exponential covariance matrix with $\sigma = 3$ and $l = 0.5$, and domain $[-4, 4]$. (b) Posterior example where the real function is $f(x)$ and is estimated with a posterior mean $\boldsymbol{\mu}_{2|1}$ and posterior covariance $\boldsymbol{\Sigma}_{2|1}$.

2.5 Standard finite element method

The finite element method (FEM) is a numerical tool used to approximate the solution of continuous problems. Although the method can be used for all kinds of PDEs, the derivation of the method will be illustrated with a 1D boundary value problem as shown in Fig. (4), where the function to obtain is $u = u(x)$ in the one-dimensional domain $\mathcal{D} = [0, L]$:

$$-\frac{d}{dx}\left(EA\frac{du}{dx}\right) + ku = f \;\forall\; x \in \mathcal{D} \tag{21}$$

where $EA = EA(x)$, $k = k(x)$, and $f = f(x)$ are parameters of the problem.

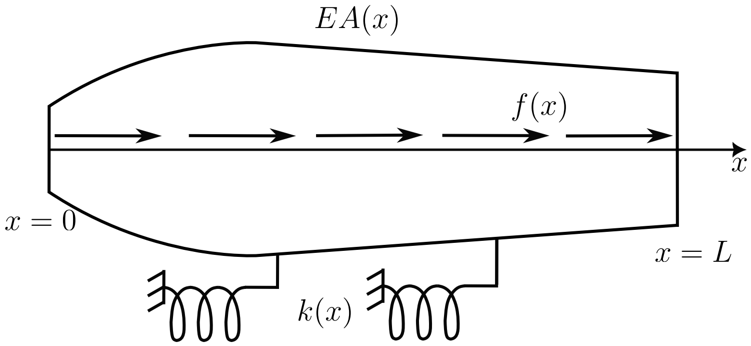


Figure 4: Example general 1D bar with distributed load.

2.5.1 Weak form

To set up the weak (or weighted-integral) formulation of any PDE, we have to multiply both sides by the test or weight function $v(x) = v$ and integrate over the domain $\mathcal{D}$. This way, we will obtain a solution that satisfies the equilibrium on average.

$$\int_0^L v\left(-\frac{d}{dx}\left(EA\frac{du}{dx}\right) + ku\right)dx = \int_0^L vf dx \quad \forall\; v \in \mathcal{V} \tag{22}$$

Now we do the integration by parts:

$$\int_0^L \frac{dv}{dx}EA\frac{du}{dx}dx + \int_0^L vku\ dx = \int_0^L vfdx + \underbrace{vEA\frac{du}{dx}\bigg|_0^L}_{\text{Neumann BCs}} \quad \forall\ v \in \mathcal{V} \tag{23}$$

In the weak formulation, the solution is more flexible and does not require a second derivative. This relaxed requirement leads to a trial function, which approximates the solution. Unlike the strong formulation, the weak formulation does not guarantee an exact solution, hence the term "*trial solution*". Overall, the weak formulation allows for broader function considerations and approximate solutions to differential equations.

2.5.2 Bubnov-Galerkin formulation

The solution of the Eq. (23) exists in the infinite-dimensional space $\mathcal{U}$. Therefore, it is impractical to search for solutions in this way. To solve this the domain $\mathcal{D}$ is partitioned in a number of subdomains or *finite elements* $\mathcal{D}^h \subseteq \mathcal{D}$. When relaxing the requirement for an infinite-dimensional space, it is sufficient for the functions to be continuous at the interfaces between elements, without requiring continuity of their derivatives in the interfaces.

$$\int_0^L \frac{dv^h}{dx}EA\frac{du^h}{dx}dx \int_0^L v^h k u^h\ dx = \int_0^L v^h f\ dx + v^h EA\frac{du^h}{dx}\bigg|_0^L \quad \forall\ v^h \in \mathcal{V}^h \tag{24}$$

In the Bubnov-Galerkin formulation, the trial and weight functions are assumed to belong to the same finite-dimensional space. Then $\mathcal{U}^h = \mathcal{V}^h$

$$\begin{aligned} u^h &= \sum_{i=1}^n u_i N_i(x),\ u^h \in \mathcal{U}^h, \\ v^h &= \sum_{j=1}^n v_j N_j(x),\ v^h \in \mathcal{V}^h \end{aligned} \tag{25}$$

Where $\boldsymbol{u}$ and $\boldsymbol{v}$ are vectors with the coefficients that act as amplitudes in each degree of freedom (*DOF*) and $\boldsymbol{N}$ is a vector with the shape functions. After replacing (25) in (24) we obtain

$$\int_0^L \sum_{j=1}^n v_j \frac{dN_j}{dx} EA \sum_{i=1}^n u_i \frac{dN_i}{dx} dx + \int_0^L \sum_{j=1}^n v_j N_j k \sum_{i=1}^n u_i N_i\ dx = \int_0^L \sum_{j=1}^n v_j N_j f\ dx + \sum_{j=1}^n v_j N_j EA \frac{du}{dx}\bigg|_0^L \quad \forall\ v^h \in \mathcal{V}^h \tag{26}$$

Since $\boldsymbol{u}$ and $\boldsymbol{v}$ are coefficients they can be taken out of the integrals

$$\sum_{j=1}^n v_j \left(\sum_{i=1}^n u_i \int_0^L \frac{dN_j}{dx} EA \frac{dN_i}{dx} + N_j k N_i\ dx \right) = \sum_{j=1}^n v_j \left(\int_0^L N_j f\ dx + N_j EA \frac{da}{dx}\bigg|_0^L \right) \quad \forall\ v^h \in \mathcal{V}^h \tag{27}$$

This relationship has to be fulfilled for each v_j this term can be canceled out

$$\sum_{i=1}^n \underbrace{\int_0^L \frac{dN_j}{dx} EA \frac{dN_i}{dx} + N_j k N_i\ dx}_{K_{ij}}\, u_i = \underbrace{\int_0^L N_j f\ dx + N_j EA \frac{du}{dx}\bigg|_0^L}_{f_j} \quad \forall\ v^h \in \mathcal{V}^h \tag{28}$$

the system of equations is

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f} \tag{29}$$

2.5.3 Boundary conditions

All elliptic differential equations have boundary conditions that can belong to the solution field or to the derivative of the solution, also called Dirichlet and Neumann boundary conditions. The stiffness matrix represents a semi-definite matrix since it has zero eigenvalues, which physically corresponds to rigid translations. After applying the Dirichlet boundary conditions, the stiffness matrix becomes positive definite due to the elimination of the rigid translations. Therefore, the matrix can be inverted. The way to impose them is to substitute the constraints in the system of equations obtained after assembling the system. For that, the system is divided in the following way.

$$\begin{pmatrix} \boldsymbol{K}_{11} & \boldsymbol{K}_{12} \\ \boldsymbol{K}_{21} & \boldsymbol{K}_{22} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \end{pmatrix} \tag{30}$$

Were $\boldsymbol{u}_1$ are the known Dirchlet boundary conditions, $\boldsymbol{u}_2$ the unknowns of the solution field, $\boldsymbol{f}_1$ are the unknowns of the forcing field, and $\boldsymbol{f}_2$ the Neumann boundary conditions and body force terms. Then is possible to rewrite (30) as

$$\begin{aligned} \boldsymbol{K}_{11}\boldsymbol{u}_1 + \boldsymbol{K}_{12}\boldsymbol{u}_2 = \boldsymbol{f}_1 \\ \boldsymbol{K}_{21}\boldsymbol{u}_1 + \boldsymbol{K}_{22}\boldsymbol{u}_2 = \boldsymbol{f}_2 \end{aligned} \tag{31}$$

therefore the unknown of the system can be calculated as

$$\boldsymbol{u}_2 = (\boldsymbol{K}_{22})^{-1} (\boldsymbol{f}_2 - \boldsymbol{K}_{21}\boldsymbol{u}_1) \tag{32}$$

and then by replacing $\boldsymbol{u}_2$ in the first equation of (31) it is possible to obtain the unknowns $\boldsymbol{f}_1$. Since the boundary conditions are going to be used in the BFEM prior covariance matrices with some differences, the system of equations of (30) will be re-arranged as follows

$$\begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_{22} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{f}_2 \end{pmatrix} + \begin{pmatrix} \boldsymbol{u}_1 \\ -\boldsymbol{K}_{21}\boldsymbol{u}_1 \end{pmatrix} \tag{33}$$

2.6 Bayesian finite element method

To introduce BFEM an illustrative example is done. Fig. (5) shows a 1D bar with uniformly distributed load and homogeneous Dirichlet boundary conditions. The ODE for this problem is shown in Eq. (34).

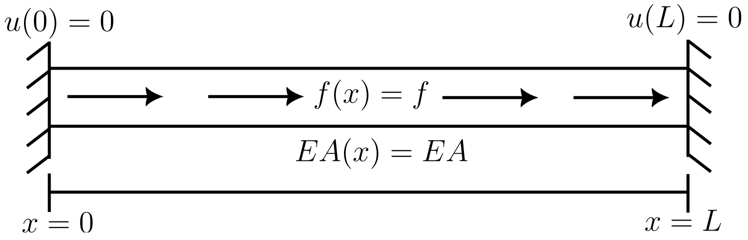
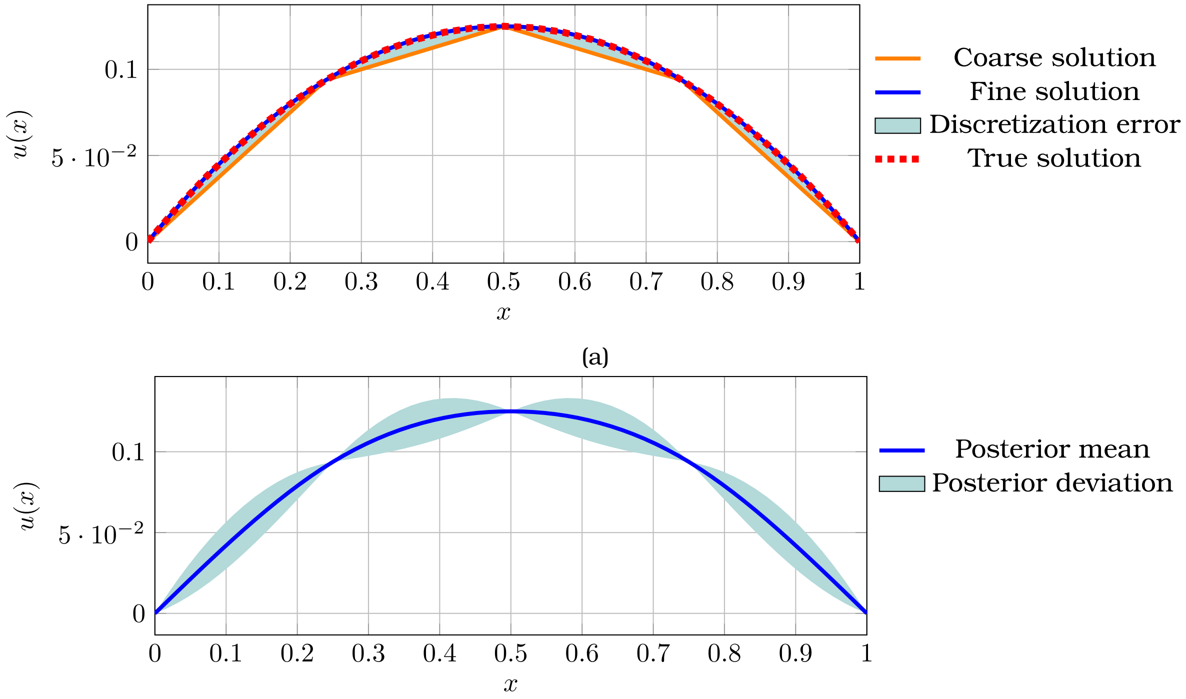


Figure 5: Example of 1D bar with the distributed load and homogenous boundary conditions. In this case, the distributed load is $f = 1$, the axial stiffness is $EA = 1$, and the length is $L = 1$.

$$-\frac{d^2u(x)}{dx^2}EA = f(x) \tag{34}$$

The true solution of the ODE and two numerical solutions obtained with standard FEM are plotted in Fig. (6a). The first numerical solution has a coarse of 4 elements and the second numerical solution has a fine mesh of 64 elements. Since the fine mesh contains a larger number of elements, the numerical solution closely resembles the true solution of the ODE. However, with fewer elements in the mesh, there is a noticeable difference between the numerical and true solutions. This difference is known as the discretization error, which arises because the number of elements used in the numerical method is insufficient to represent the solution accurately. To minimize this error and obtain precise results, a finer mesh with more elements is required.

In BFEM, the objective is to use a Gaussian regression as shown in Fig. (6b), where the posterior mean should be close to the fine solution that is assumed as the true solution, while the posterior standard deviation indicates the uncertainty associated with the discretization error of a coarse mesh.



(a)

(b)

Figure 6: Comparison between standard FEM solutions and BFEM solution.(a) Example of true and standard FEM solutions. (b) Example of statistical solution.

This method is based on the work of Poot *et al.* [18]. The force vector of a coarse mesh is used as data to perform a GP regression in a fine space. This means that in the fine space, a prior mean and prior covariance are proposed and the posterior mean and deviation are obtained after conditioning the distribution with the coarse mesh force vector. To formulate this method, the steps are the same as in the previous section until Eq. (23). After obtaining the weak formulation, the Petrov-Galerkin method is used, a prior is assumed to regularize the solution, and then make use of Bayesian inference to obtain the posterior distribution.

2.6.1 Petrov-Galerkin formulation

The Petrov-Galerkin formulation has the characteristic that the trial and weight or test functions belong to different spaces. In BFEM, this formulation is used to have a space of test functions, denoted as $\mathcal{V}$, which can be constructed as a subset of the space of trial functions, denoted as $\mathcal{U}$. Therefore, we have $\mathcal{V} \subseteq \mathcal{U}$.

The trial and test functions are defined in the same way as in Eq. (25), but in this case, the shape functions Q_j belong to the discretization of the coarse mesh, while the shape functions N_i belong to the fine mesh. This implies that the number of coarse shape functions denoted as n_c, is always less than the number of fine shape functions, denoted as n_f.

$$
\begin{aligned}
u^h &= \sum_{i=1}^{n_f} u_i N_i(x),\ u^h \in \mathcal{U}^h, \\
v^h &= \sum_{j=1}^{n_c} v_j Q_j(x),\ v^h \in \mathcal{V}^h
\end{aligned} \tag{35}
$$

If we substitute u^h and v^h from equation (35) into equation (24) and move the summation of v_j coefficients outside the integral, we obtain:

$$
\sum_{j=1}^{n_c} v_j \left(\sum_{i=1}^{n_f} \underbrace{\int_0^L \frac{dQ_j}{dx} EA \frac{dN_i}{dx} + Q_j k N_i \ dx}_{H_{ij}} u_i \right) = \sum_{j=1}^{n_c} v_j \left(\underbrace{\int_0^L Q_j f \ dx + Q_j EA \frac{du}{dx}\bigg|_0^L}_{g_j} \right) \quad \forall\ v^h \in \mathcal{V}^h \tag{36}
$$

Again, the v_j terms can be canceled out, and the system of equations is:

$$
\boldsymbol{H}\boldsymbol{u} = \boldsymbol{g} \tag{37}
$$

Since i is always bigger or equal to j, the $\boldsymbol{H}$ matrix is a rectangular matrix with i columns and j rows. To illustrate this, consider a 1D problem with a coarse mesh of 3 elements and a fine mesh of 6 elements, as shown in Fig. (7). Since the coarse mesh has 4 DOFs and the fine mesh has 7 DOFs, the $\boldsymbol{H}$ matrix will have 7 columns and 4 rows. The vector $\boldsymbol{u}$ will have 7 components, while $\boldsymbol{g}$ will have 4 components. Therefore, we have more unknowns to obtain than equations because the displacement belongs to the fine mesh, and the force vector belongs to the coarse mesh.

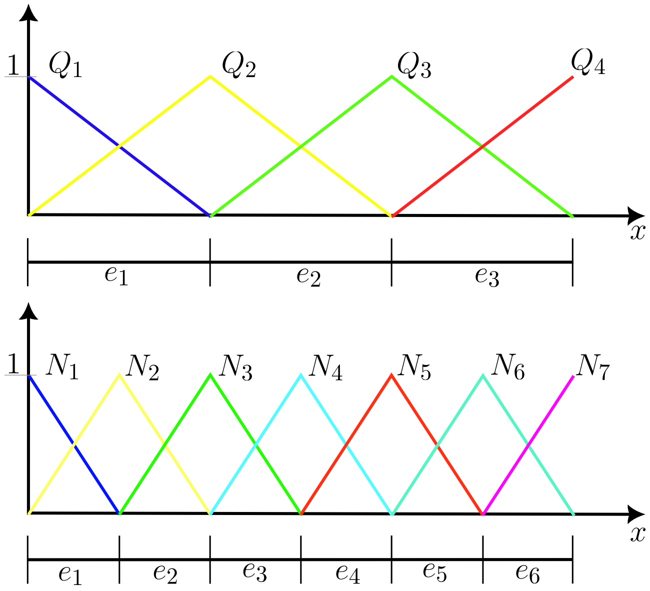


Figure 7: Example of a coarse and fine mesh used in BFEM.

As mentioned above, the coarse shape functions can be obtained as a linear combination of the fine shape functions. Eq. (38) shows how the combination of shape functions from the fine mesh can give a shape function from the coarse mesh:

$$
Q_j(x) = \sum_{k=1}^{n_f} \Phi_{kj} N_k(x) \tag{38}
$$

Here, $N_k(x)$ represents the shape functions of the fine mesh, Φ_{kj} are the components of a rectangular transformation matrix, and $Q_j(x)$ represents the coarse mesh shape functions. The index

j represents the index of the coarse DOFs, while the index k represents the indices from the fine mesh. If we replace the Q_j term on both sides of Eq. (36) with $\sum_{k=1}^{n_f}\Phi_{kj}N_k$, we obtain

$$\sum_{i=1}^{n_f}\int_0^L \frac{d\left(\sum_{k=1}^{n_f}\Phi_{kj}N_k\right)}{dx}EA\frac{dN_i}{dx}+\sum_{k=1}^{n_f}\Phi_{kj}N_k k N_i\;dx\;u_i = \int_0^L\sum_{k=1}^{n_f}\Phi_{kj}N_k f\;dx+\sum_{k=1}^{n_f}\Phi_{kj}N_k EA\left.\frac{du}{dx}\right|_0^L \quad \forall\; v^h\in\mathcal{V}^h \tag{39}$$

Now the Φ_{kj} components can be taken out of the derivative or integrals since these values do not depend on x.

$$\sum_{i=1}^{n_k}\sum_{k=1}^{n_f}\Phi_{kj}\underbrace{\int_0^L \frac{dN_k}{dx}EA\frac{dN_i}{dx}+N_k k N_i\;dx}_{K_{ki}}\,u_i = \sum_{k=1}^{n_f}\Phi_{kj}\underbrace{\int_0^L N_k f\;dx+N_k EA\left.\frac{du}{dx}\right|_0^L}_{f_k} \quad \forall\; v^h\in\mathcal{V}^h \tag{40}$$

What is proved with this final derivation is that $\sum_{k=1}^{n_f}\Phi_{kj}K_{ki} = H_{ji}$ and that $\sum_{k=1}^{n_f}\Phi_{kj}f_k = g_j$. Which in matrix form can be written as $\boldsymbol{H} = \boldsymbol{\Phi}^T\boldsymbol{K}$ and $\boldsymbol{g} = \boldsymbol{\Phi}^T\boldsymbol{f}$.

2.6.2 Bayesian inference

The current solution is underconstrained and requires regularization by introducing a prior solution. The chosen distribution has a zero prior mean and a prior covariance matrix.

$$\boldsymbol{u}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{\Sigma}) \tag{41}$$

Utilizing the connection between $\boldsymbol{u}$ and $\boldsymbol{g}$ as described in Eq. (37), it can be deduced that the joint distribution of both vectors is:

$$\begin{pmatrix}\boldsymbol{g}\\ \boldsymbol{u}\end{pmatrix}\sim\mathcal{N}\left(\begin{pmatrix}\boldsymbol{0}\\ \boldsymbol{0}\end{pmatrix},\begin{pmatrix}\boldsymbol{H\Sigma H}^T & \boldsymbol{H\Sigma}\\ \boldsymbol{\Sigma H}^T & \boldsymbol{\Sigma}\end{pmatrix}\right) \tag{42}$$

Since $\boldsymbol{g}$ is known we can condition $\boldsymbol{u}$ and obtain the posterior distribution

$$\boldsymbol{u}|\boldsymbol{g}\sim\mathcal{N}(\hat{\boldsymbol{u}},\hat{\boldsymbol{\Sigma}}) \tag{43}$$

The posterior mean and posterior covariance are

$$\hat{\boldsymbol{u}} = \boldsymbol{\Sigma H}^T\left(\boldsymbol{H\Sigma H}^T\right)^{-1}\boldsymbol{g} \tag{44}$$

$$\hat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}-\boldsymbol{\Sigma H}^T\left(\boldsymbol{H\Sigma H}^T\right)^{-1}\boldsymbol{H\Sigma} \tag{45}$$

2.6.3 Boundary conditions in BFEM

The treatment of Dirichlet or Neumann boundary conditions in a probabilistic manner is a possibility. However, in this particular study, these conditions are assumed to be deterministic and known, given the proposed prior distribution. For this reason, the conditions can be thought of as values changing the prior mean and with zero deviations in those DOFs. Then the system in Eq. (41) will be subdivided like in (30).

$$\begin{pmatrix}\boldsymbol{u_1}\\ \boldsymbol{u_2}\end{pmatrix}\sim\mathcal{N}\left(\begin{pmatrix}\boldsymbol{u_1}\\ \boldsymbol{K_{22}}^{-1}(\boldsymbol{f_2}-\boldsymbol{K_{21}u_1})\end{pmatrix},\begin{pmatrix}\boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{\Sigma}_{22}\end{pmatrix}\right) \tag{46}$$

Where $\boldsymbol{u_1}$ represents the vector of Dirichlet conditions, no matter if the condition is homogeneous or not, and $\boldsymbol{f_2}$ is the vector of known forces, i.e. the Neumann conditions. The DOFs where the solution is not known are represented by the subindex 2. In the DOFs of Dirichlet conditions, the deviation is zero. On the contrary in the DOFs where the solution is not known the components of the covariance prior matrix $\boldsymbol{\Sigma}$ remain the same. The problem with this is that then the matrix cannot be inverted since it is necessarily positive semidefinite. For this reason, a small noise $\sigma_\varepsilon \sim 10^{-7}$ will be added in the diagonal entries of the matrix to avoid numerical inconveniences.

As evident, unless the problem has homogeneous Dirichlet conditions, the prior mean vector will not be zero. From now on, in this thesis, we will refer to the mean vector in Eq. (46) as $\boldsymbol{u_m}$.

2.6.4 Sampling from prior and posterior

As mentioned in the introduction, the main idea of sampling is to use it to obtain covariance matrices more cost-effectively than analytically calculating the matrix. Now, if the prior samples are directly obtained from the solution $\boldsymbol{u}$, it would have to be performed as shown in Eq. (4) and Eq. (47).

$$\boldsymbol{u_s} = \boldsymbol{L}\boldsymbol{z} + \boldsymbol{u_m} \tag{47}$$

Where $\boldsymbol{L}$ is the lower triangular matrix obtained from the Cholesky decomposition of the prior covariance $\boldsymbol{\Sigma}$, $\boldsymbol{z}$ is a sample vector with zero mean, and an identity covariance matrix, and $\boldsymbol{u_s}$ is the sample prior. The disadvantage of this approach is that the matrix $\boldsymbol{L}$ is a decomposition of the prior covariance matrix which is dense.

To avoid this drawback, we can use a sparse covariance matrix in the RHS, as explained in section (3.1), to obtain less expensive $\boldsymbol{f_s}$ samples. Then prior samples of the force vector can be obtained by

$$\boldsymbol{f_s} = \boldsymbol{L_f}\boldsymbol{z} + \boldsymbol{f_m} \tag{48}$$

Where $\boldsymbol{L_f}$ is the Cholesky decomposition of a sparse matrix, $\boldsymbol{z}$ is a sample vector with zero mean, $\boldsymbol{f_m}$ is a vector that includes the Neumann boundary conditions, and an identity covariance matrix, and $\boldsymbol{f_s}$ is the sample prior of the force vector.

Once this is done, the system $\boldsymbol{K}\boldsymbol{u_s} = \boldsymbol{f_s}$ can be solved more cost-effectively than simply inverting $\boldsymbol{K}$, obtaining samples of the solution $\boldsymbol{u}$ without the need to decompose a dense prior matrix. In this thesis, sparse Cholesky factorization is used, but for large systems of degrees of freedom, an iterative solver such as the Gauss-Seidel method can be used.

Now the idea is to obtain posterior samples without the need to compute the posterior covariance matrix. If we take into account that the posterior mean and covariance are

$$\hat{\boldsymbol{u}} = \underbrace{\boldsymbol{\Sigma}\boldsymbol{H}^T\left(\boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{H}^T + \sigma_\varepsilon^2\boldsymbol{I}\right)^{-1}}_{\boldsymbol{G}}\boldsymbol{g} = \boldsymbol{G}\boldsymbol{g} \tag{49}$$

$$\hat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma} - \underbrace{\boldsymbol{\Sigma}\boldsymbol{H}^T\left(\boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{H}^T + \sigma_\varepsilon^2\boldsymbol{I}\right)^{-1}}_{\boldsymbol{G}}\boldsymbol{H}\boldsymbol{\Sigma} = \left(\boldsymbol{I} - \boldsymbol{G}\boldsymbol{H}\right)\boldsymbol{\Sigma} \tag{50}$$

where $\boldsymbol{G}$ is the Kalman gain matrix and σ_ε is the deviation of an observational noise distribution $\varepsilon = \mathcal{N}\left(\boldsymbol{0}, \sigma_\varepsilon^2\boldsymbol{I}\right)$, then we can rewrite the posterior covariance in the Joseph form:

$$\hat{\boldsymbol{\Sigma}} = \left(\boldsymbol{I} - \boldsymbol{G}\boldsymbol{H}\right)\boldsymbol{\Sigma}\left(\boldsymbol{I} - \boldsymbol{G}\boldsymbol{H}\right) + \sigma_\varepsilon\boldsymbol{G}\boldsymbol{G}^T \tag{51}$$

Now, posterior samples can be obtained by:

$$\hat{\boldsymbol{u}}_{\boldsymbol{s}} = \boldsymbol{u_s} + \boldsymbol{G}\left(\boldsymbol{g} - \boldsymbol{H}(\boldsymbol{u_s} - \boldsymbol{u}_m) + \hat{\varepsilon}\right) \tag{52}$$

where $\hat{\varepsilon}$ is a sample vector from the observational noise distribution. Finally, the term $\boldsymbol{G}$ depends on the prior covariance matrix, but this can be assembled with the prior samples.

2.7 Optimization methods

Throughout this thesis different optimization methods are used, therefore a brief summary of the most used tools will be made.

After obtaining the data and defining the objective function, the algorithm searches for the parameters of the function that correspond to a global minimum

$$\bar{\boldsymbol{x}} = arg\ \underset{\boldsymbol{x}}{min}\ f(\boldsymbol{x}) \tag{53}$$

Where $f(\mathbf{x})$ is the multivariate function to be minimized and $\bar{\mathbf{x}}$ is the value corresponding to the global minimum.

2.7.1 Gradient descent

This iterative method involves updating the parameters of the objective function by moving in a downward direction. To find the descent direction the gradient of the function is used [28].

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \alpha \nabla f(\boldsymbol{x}_i) \tag{54}$$

α is a positive real value representing the learning rate. It determines the step size for each model parameter update, and finding a suitable value can be challenging. Typically, determining the appropriate learning rate involves a trial-and-error approach, which can be highly impractical. On one hand, if the value is small it can converge slowly or get stuck in local minima. On the other hand, if the value is high it may result in an unstable convergence. In Fig. (8), it can be observed how the direction of the vectors varies based on the gradient at each point. The length of these vectors is determined by the gradient at each point and the learning rate.

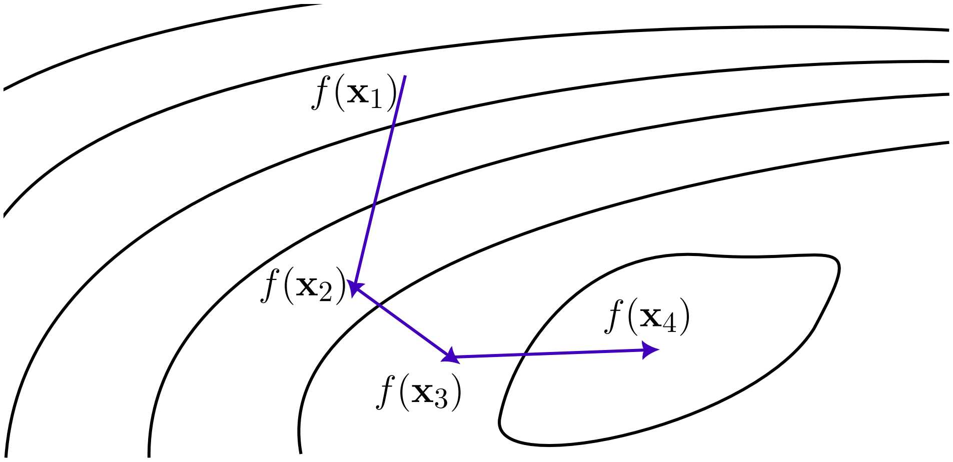


Figure 8: Plot of a gradient descent algorithm.

2.7.2 Newton methods for optimization

The gradient descent does not take into account higher-order terms, slowing down the convergence close to a minimum. To take these terms into account we will review Newton's method [28]. If we do a Taylor expansion of a one dimension problem, the second order of a function in the point $x_i + \Delta$ is

$$f(x_i + \Delta) = f(x_i) + f'(x_i)\Delta + \frac{1}{2}f''(x_i)\Delta^2 \tag{55}$$

To obtain the value of Delta to minimize this value we can derive in that direction and equal to zero.

$$\begin{aligned} f'(x_i) + f''(x_i)\Delta = 0 \\ \Delta = -\frac{f'(x_i)}{f''(x_i)} \end{aligned} \tag{56}$$

then the value of x_{i+1} can be obtained by

$$x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)} \tag{57}$$

As it is possible to observe this iterative method has a variable step size that is updated depending on the first and second derivative of the function. If we generalize it for multiple variables

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \boldsymbol{H}(\boldsymbol{x_i})^{-1} \nabla f(\boldsymbol{x}_i) \tag{58}$$

Where $\nabla f(\boldsymbol{x}_i)$ is the gradient vector and $\boldsymbol{H}(\boldsymbol{x_i})$ is the Hessian matrix, composed of the second derivatives. Eq. (58) is the form of the Newton method.

If this method is used for multiple variables, it becomes expensive compared to gradient descent. Computing an inverse Hessian matrix is $O(n^3)$, while the complexity of the gradient descent is $O(n)$.

Quasi-Newton methods are a family of optimization algorithms that approximate the Hessian matrix using gradient information, rather than computing it directly. This can significantly reduce the computational cost of the optimization algorithm, making it more efficient for large-scale problems. To explain it simply in a single-variable problem the way to obtain the points is

$$x_{i+1} = x_i - f'(x_i) \frac{x_i - x_{i-1}}{f'(x_i) - f'(x_{i-1})} \tag{59}$$

the second derivative inverse is approximated by

$$f''(x_i)^{-1} = \frac{x_i - x_{i-1}}{f'(x_i) - f'(x_{i-1})} \tag{60}$$

In multiple variables, this can be generalized as

$$\boldsymbol{B}_{i+1} (\boldsymbol{x}_{i+1} - \boldsymbol{x}_i) = \nabla f(\boldsymbol{x}_{i+1}) - \nabla f(\boldsymbol{x}_i) \tag{61}$$

Where $\boldsymbol{B}_{i+1}$ is the approximation of the Hessian matrix.

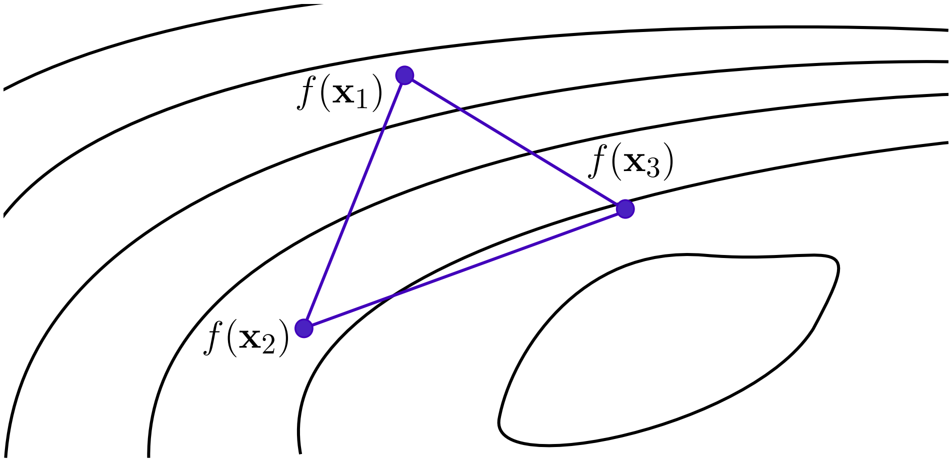


Figure 9: Example of triangle as simplex in a Nelder-Mead algorithm in a 2D objective function.

The most well-known quasi-Newton method and the one used in this thesis is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) [29] algorithm, which approximates the Hessian matrix using a rank-two update approach. Other quasi-Newton methods include the limited-memory BFGS (L-BFGS) algorithm [30].

2.7.3 Nelder–Mead

The Nelder-Mead method is a heuristic algorithm [31] since it solves a complex optimization problem by an intuitive approach for unconstrained multidimensional problems. Moreover, it does not use information from derivatives or curvatures but instead simply works as a point comparison method. It works by maintaining a simplex, which is a figure with one vertex more than the dimensions of the function. For example, a simplex of a 2-dimensional function is a triangle and a 3-dimensional function is a tetrahedron. At each iteration of the algorithm, the function values at the vertices of the simplex are computed and a series of transformations are performed to update the simplex until convergence is reached. In Fig. (9), it is illustrated how a simplex consisting of three points is constructed over the function $f(\boldsymbol{x})$. By comparing the function values at these points, a new triangle can be formed. The transformations include reflection, expansion, contraction, and shrinkage of the simplex. The Nelder-Mead method is easy to apply, requires only function evaluations, and can handle non-smooth or noisy functions. However, it can converge slowly or stagnate at local minima.

3

Model Selection in BFEM

In the previous chapter, BFEM was introduced as a method to obtain statistical solutions where the posterior deviation should show the discretization error as an uncertainty along the mesh. However, it did not explain the selection of a suitable prior covariance, the estimation of prior hyperparameters, and the appropriate criteria for assessing the quality of results obtained for a specific prior. In this chapter, we establish the fundamental concepts of model selection in BFEM, which involve exploring various covariance matrices to find those that are more appropriate. This includes techniques for estimating prior hyperparameters and defining measures to determine the adequacy of results achieved with a particular prior.

3.1 Choices of prior covariance

The prior distribution can be formulated as:

$$\boldsymbol{u} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}) \tag{62}$$

In this case, the prior covariance can be formed with a kernel, such as a square exponential kernel. One potential issue arises when using this approach: the resulting covariance matrix becomes dense, leading to computational inefficiencies, such as the need to decompose a dense matrix to perform sampling and the need to invert a prior dense matrix to obtain the posterior.

Alternatively, considering the nature of any PDE, there exists a fixed integro-differential relationship between the solution and the forcing term. Using this relationship allows us to assume white noise in the right-hand side (RHS) and derive an equivalent prior for the solution, as discussed in Owhadi [20].

To explain this in detail, let us consider a general example of a continuous problem where $\mathcal{L}$ and $\mathcal{B}$ are integro-differential operators defined on the domain $\mathcal{D}$ and the boundary Γ. We have the following PDE:

$$\begin{cases} \mathcal{L}u(\boldsymbol{x}) = f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{D} \\ \mathcal{B}u(\boldsymbol{x}) = 0, \quad \text{on } \Gamma \end{cases} \tag{63}$$

To give an example of what the linear operator would look like in a particular case we can take the left-hand side (LHS) of Eq. (21)

$$-\frac{d}{dx}\left(EA\frac{du(x)}{dx}\right) + ku(x) = \mathcal{L}u(x) \tag{64}$$

meaning that the operator $\mathcal{L}$ for this ODE is

$$\mathcal{L} = -\frac{d}{dx}\left(EA\frac{d}{dx}\right) + k \tag{65}$$

Now we make the assumption that there exists a Green's function, denoted as $\mathcal{G}(\boldsymbol{x}, \boldsymbol{w})$, corresponding to the operator $\mathcal{L}$. This Green's function is defined as follows:

$$\begin{cases} \mathcal{L}\mathcal{G}(\boldsymbol{x}, \boldsymbol{w}) = \delta(\boldsymbol{w} - \boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{D} \\ \mathcal{B}\mathcal{G}(\boldsymbol{x}, \boldsymbol{w}) = 0, \quad \text{on } \Gamma \end{cases} \tag{66}$$

The solution of the general PDE in Eq. (63) can then be expressed as:

$$u(\boldsymbol{x}) = \int_{\mathcal{D}} \mathcal{G}(\boldsymbol{x}, \boldsymbol{w}) f(\boldsymbol{w}) d\boldsymbol{w} \tag{67}$$

This solution is known as Green's solution for linear and inhomogeneous ODEs. Now, if a prior is assumed for the forcing term $f(\boldsymbol{x})$ as $\mathcal{GP}\left(\boldsymbol{0}, k(\boldsymbol{x}, \boldsymbol{x'})\right)$, the kernel of the solution is:

$$k_{\text{nat}}(\boldsymbol{x}, \boldsymbol{x'}) = \int_{\mathcal{D}}\int_{\mathcal{D}} \mathcal{G}(\boldsymbol{x}, \boldsymbol{w})\mathcal{G}(\boldsymbol{x'}, \boldsymbol{w'})k(\boldsymbol{x}, \boldsymbol{x'}) d\boldsymbol{w} d\boldsymbol{w'} \tag{68}$$

This approach is described as the "*natural kernel*" in Cockayne *et al.* [14], because it naturally captures part of the physics of the problem and its smoothness. For example, in an elastic problem with different stiffnesses in the domain, it reveals the spatial correlation change due to the variation in stiffness. In addition, the kernel in the forcing term $k(\boldsymbol{x}, \boldsymbol{x'})$ can be a white noise $\delta(\boldsymbol{x} - \boldsymbol{x'})$ and the result after evaluating the natural kernel at the $\boldsymbol{X}$ positions to assemble a covariance matrix would not result in a sparse matrix.

However, the previous example is applicable to collocation problems. In this work, the form of the stochastic partial differential equation (SPDE) is discrete, as in Eq. (29). Therefore, the distribution of the force vector $\boldsymbol{f}$ needs to be obtained based on a white noise forcing term $f(\boldsymbol{x}) \sim \mathcal{GP}\left(\boldsymbol{0}, \alpha_0^2 \delta(x - x')\right)$ integration, as shown in Eq. (28):

$$\boldsymbol{f} = \int_{\mathcal{D}} f(\boldsymbol{x})\boldsymbol{N}(\boldsymbol{x}) d\boldsymbol{x} \sim \mathcal{N}\left(\boldsymbol{0}, \int_{\mathcal{D}}\int_{\mathcal{D}} \alpha_0^2 \boldsymbol{N}(\boldsymbol{x})\delta(\boldsymbol{x} - \boldsymbol{x}')\boldsymbol{N}(\boldsymbol{x}') d\boldsymbol{x} d\boldsymbol{x}'\right) \tag{69}$$

where $\boldsymbol{N}(\boldsymbol{x})$ are the vector of shape functions of the fine mesh. After evaluating the integral, we obtain:

$$\boldsymbol{f} = \int_{\mathcal{D}} f(\boldsymbol{x})\boldsymbol{N}(\boldsymbol{x}) d\boldsymbol{x} \sim \mathcal{N}\left(\boldsymbol{0}, \alpha_0^2 \int_{\mathcal{D}} \boldsymbol{N}(\boldsymbol{x})\boldsymbol{N}(\boldsymbol{x}) d\boldsymbol{x}\right) \sim \mathcal{N}\left(\boldsymbol{0}, \alpha_0^2 \boldsymbol{M}\right) \tag{70}$$

A unit mass matrix $\boldsymbol{M}$ that depends only on the shape functions. Therefore, is obtained when assuming a white noise in the RHS term.

Similar to Eq. (67), the relationship between $\boldsymbol{u}$ and $\boldsymbol{f}$ is established through the inverse of the stiffness matrix $\boldsymbol{K}$.

$$\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{0}, \alpha_0^2 \boldsymbol{K^{-1}MK^{-1}}) \tag{71}$$

As can be observed, there exists an analogy between the utilization of Green's functions in a continuous problem and the utilization of the inverse of the stiffness matrix in a discrete problem, which incorporates structural information. It is important to note that inverting the stiffness matrix can be computationally expensive. However, due to its sparse structure, it is relatively easier to invert compared to a dense matrix formed with a kernel. In the work by Roininen *et al.* [21], it has been demonstrated that inverting the covariance matrix in Eq. (71) simplifies to Eq. (72), where the mass matrix is easy to invert due to its sparse structure and the stiffness matrix is known.

$$\boldsymbol{\Sigma}^{-1} = \frac{1}{\alpha_0^2}\boldsymbol{KM^{-1}K} \tag{72}$$

3.1.1 Example comparing LHS and RHS Priors

To compare priors in the LHS and RHS an illustrative example is made. The example is a 1D tapered bar subjected to a uniformly distributed load, as shown in Fig. (10). The kernel applied directly in the solution or LHS is a square exponential kernel and in the forcing term or RHS, a unit mass matrix is applied.

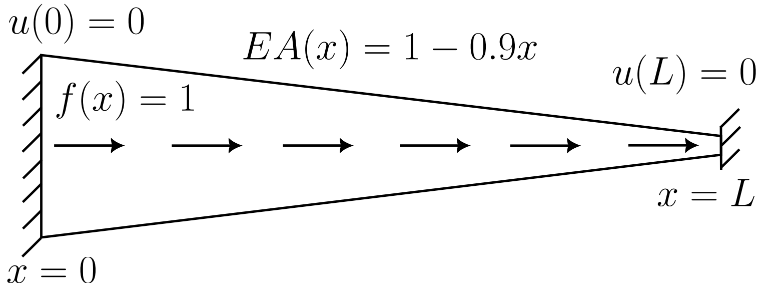
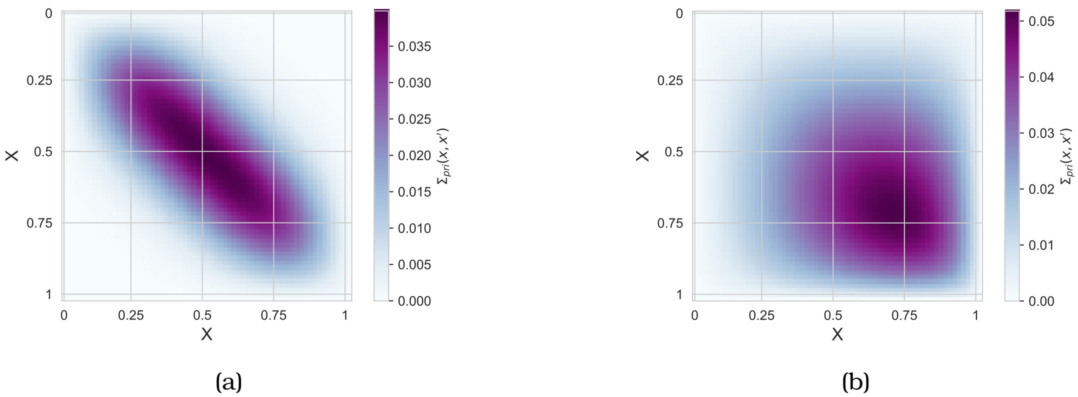


Figure 10: Example of a 1D tapered bar with smooth load. The body load is $f(x) = 1$, the boundary conditions are homogeneous Dirichlet conditions, the length is $L = 1$, and the material and geometrical properties have a linear variation $EA(x) = 1 - 0.9x$.

To solve this problem as posed, 2 meshes are used, a coarse line mesh of 4 elements and a fine line mesh of 64 elements. In both cases the progression of elements is uniform. Then, the LHS prior is

$$\boldsymbol{u} \sim \mathcal{N}(\mathbf{0}, \sigma^2 e^{-\frac{(x-x')^2}{2l^2}}) \tag{73}$$

and the RHS prior is shown in Eq. (71)



(a)

(b)

Figure 11: Comparison of prior covariance matrix in the LHS and the RHS. (a) Prior covariance matrix of 65×65, resulting from the use of a square exponential kernel with a length scale $l = 0.20$ and $\sigma = 0.65$. (b) Prior covariance matrix of 65×65, resulting from the use of a unit mass matrix with $\alpha_0 = 0.1$.

Figs. (11a) and (11b) show how the prior matrices look like. Both matrices show zero values at the boundaries due to the imposed Dirichlet conditions. Moreover, there are notable differences between the two. The first matrix displays a stationary behavior, influenced by the chosen kernel. On the other hand, the second matrix, resulting from the multiplication of stiffness matrices with the unit mass matrix, shows distinct characteristics due to the varying stiffness along the structure. Consequently, larger deviations occur in the direction of the region with lower stiffness values.

After applying the Bayesian inference with the coarse mesh force vector as the observed data, the obtained posterior covariance matrices are shown in Figs. (12a) and (12b). The results obtained with the prior placed on the LHS demonstrate similar deviation values along the length of

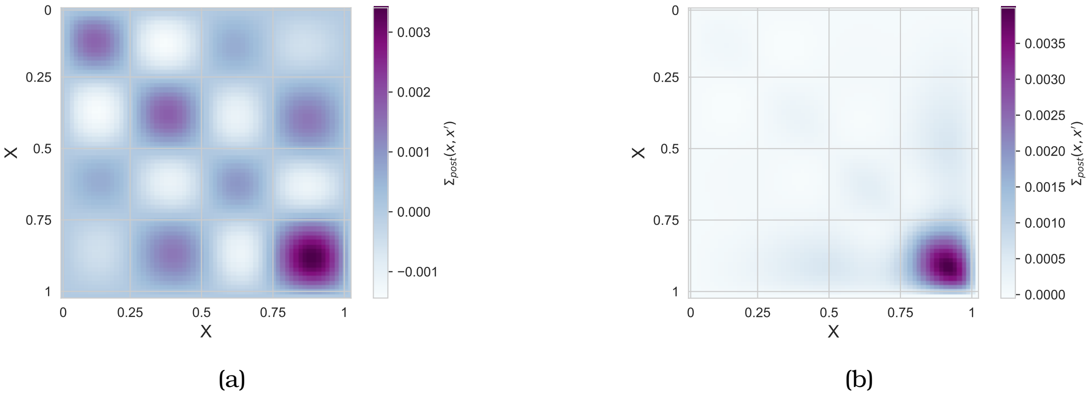


Figure 12: Comparison of posterior covariance matrix in the LHS and the RHS. (a) Posterior covariance matrix of 65×65, resulting from the use of a square exponential kernel with a length scale $l = 0.20$ and $\sigma = 0.65$. (b) Posterior covariance matrix of 65×65, resulting from the use of a unit mass matrix with $\alpha_0 = 0.1$.

the bar. In contrast, in the case of priors placed on the RHS, the standard deviation is predominantly concentrated in the region with lower stiffness.

By comparing these outcomes with the results depicted in Figs. (13a) and (13b), it can be affirmed that the prior placed on the RHS has superior performance. This conclusion arises from the fact that the higher discretization error is indeed localized in the area with lower stiffness values. Additionally, it is evident that the posterior mean aligns more closely with the solution obtained from the fine mesh.

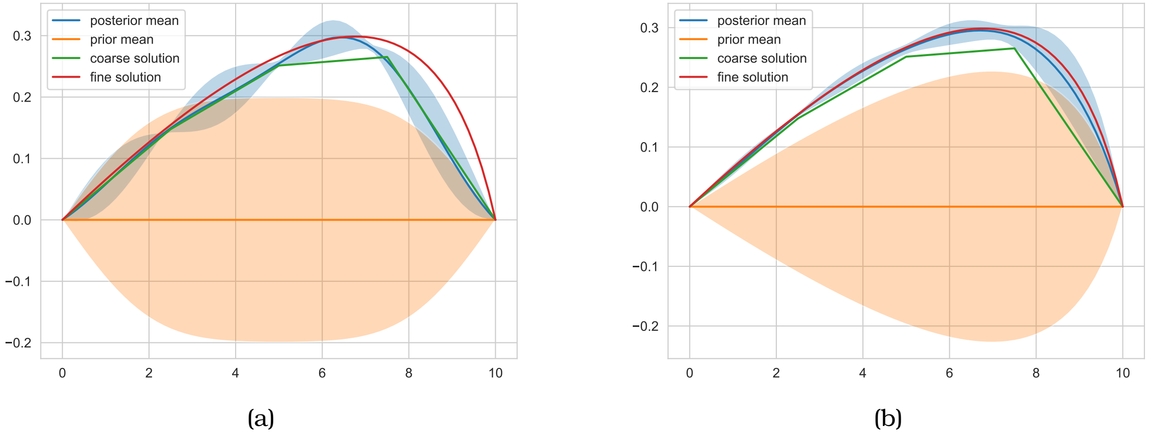


Figure 13: Comparison of solution with priors in the LHS and the RHS. (a) Solution obtained with a prior covariance matrix from the use of a square exponential kernel with a length scale $l = 0.20$ and $\sigma = 0.20$. (b) Solution obtained with a prior covariance matrix from the use of a unit mass matrix with $\alpha_0 = 0.65$.

This a simple example, but placing priors on the RHS generally leads to qualitatively superior distributions for the purpose of estimating the discretization error probabilistically. This becomes apparent when examining the prior and posterior standard deviation of both distributions. The RHS deviation more accurately represents the error and the posterior mean closely resembles the desired solution obtained from a fine mesh space. In this particular example with a limited number of coarse elements, the zone with lower stiffness has the highest error and the RHS prior takes this into account since the prior deviations are higher in this region. Additionally, in the case of performing sampling the RHS is also more beneficial because there is no need to invert a dense matrix at any time in the operations.

It is important to highlight that a white noise prior was employed in the RHS solution. In contrast, the same covariance matrix could not be used in the LHS approach. This is because, without being multiplied by the stiffness matrices, the resulting matrix would have a diagonal and sparse structure, which would introduce excessive noise throughout the bar and would not be able to visualize the discretization error.

3.1.2 Sparse RHS priors

As presented in the work of Poot *et al.* [18], another alternative is to directly assume a sparse covariance matrix in the prior distribution of the RHS and then obtain its equivalent in the solution. One option apart from the already mentioned unit mass matrix, is to assume the stiffness matrix as the prior covariance matrix of the RHS, which is

$$\boldsymbol{f} \sim \mathcal{N}\left(\mathbf{0}, \alpha_0^2 \boldsymbol{K}\right) \tag{74}$$

consequently, the resulting distribution in $\boldsymbol{u}$ can be expressed as:

$$\boldsymbol{u} \sim \mathcal{N}\left(\mathbf{0}, \alpha_0^2 \boldsymbol{K}^{-1}\right) \tag{75}$$

Furthermore, it can be demonstrated that utilizing the stiffness matrix as a prior in the RHS consistently leads to the coarse solution being the posterior mean.

In this thesis, the combination of the unit mass matrix and the stiffness matrix is analyzed:

$$\boldsymbol{f} \sim \mathcal{N}\left(\mathbf{0}, \alpha_0^2 \boldsymbol{M} + \beta_0^2 \boldsymbol{K}\right) \tag{76}$$

3.2 MLE as a hyperparameter estimator in BFEM

In BFEM, the MLE has been utilized to determine the optimal hyperparameter corresponding to a mass unit prior covariance matrix. Previous work by Poot *et al.* [18] has analytically demonstrated the methodology for obtaining this hyperparameter. However, it remains untested whether this approach leads to an accurate error estimate when considering the standard deviation.

To investigate this matter, Fig. (14) presents the distribution obtained for the problem depicted in Fig. (10) after obtaining the hyperparameter α_0 of a unit mass matrix in the RHS with MLE.

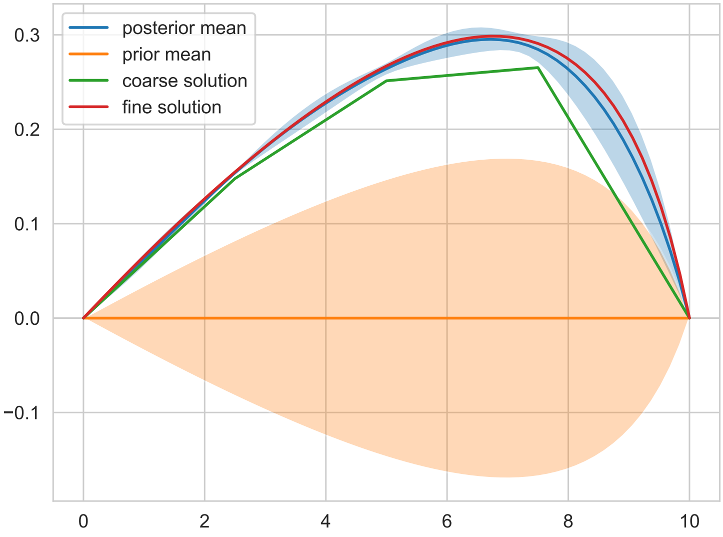


Figure 14: Distribution obtained in a 1D tapered bar with unit mass matrix in the RHS with a hyperparameter $\alpha_0 = 0.485$. The confidence interval shown is equal to one standard deviation.

In Fig. (14), it is evident that the estimated hyperparameter leads to a posterior standard deviation that appears smaller than the visually observed error between the coarse and fine mesh solutions. However, this happens because a confidence interval of only one standard deviation is shown. The main problem is that this deviation increases or decreases uniformly throughout the domain as the hyperparameter changes. Ideally, the standard deviation should accurately represent the error resulting from the mesh discretization in each DOF.

3.3 Performance assessment measures in BFEM

In this thesis, it is essential to compare different priors to see which one influences better the posterior distribution to represent the discretization error. At the same time, it is also necessary to be able to estimate the hyperparameters with the mentioned objective.

Therefore, two distinct measures are proposed. The first measure aims to compare the discretization error between the fine mesh and coarse mesh using the standard deviation. The second measure involves comparing the posterior mean with the solution on the fine mesh. In a perfect setting, the standard deviation should be equal to the error between the coarse and fine mesh, and the posterior mean should match the fine mesh solution.

The measure used in this thesis to compare the posterior deviation and discretization error is called $J_{\hat{\boldsymbol{\sigma}}}$ and is half the sum of the squared differences between the discretization error and posterior standard deviation, as shown in Eq. (77).

$$J_{\hat{\boldsymbol{\sigma}}} = \frac{1}{2}\sum_{i=1}^{n}\left(\Delta_i - \sqrt{\hat{\Sigma}_{ii}}\right)^2 \tag{77}$$

$$\boldsymbol{\Delta} = \boldsymbol{u_f} - \boldsymbol{\Phi}\boldsymbol{u_c} \tag{78}$$

where Δ_i represents the difference between the solution obtained at a DOF of the fine mesh and the projection of the coarse mesh in fine space, calculated using the matrix $\boldsymbol{\Phi}$ as shown in Eq. (78). The deviation in the fine space DOFs is denoted by $\sqrt{\hat{\Sigma}_{ii}} = \sigma_i$. Fig. (15a) illustrates how the DOFs of the fine mesh can be projected onto the coarse mesh to obtain the error between them. Additionally, Fig. (15b) presents the posterior distribution obtained with specific priors and hyperparameters.

The best values for the hyperparameters for a given prior are those that minimize $J_{\hat{\boldsymbol{\sigma}}}$. When comparing two different priors with optimized hyperparameters, the best choice is the one that leads to a lower value of $J_{\hat{\boldsymbol{\sigma}}}$.

Although arbitrary, we choose to set the standard deviation to a single standard deviation in $J_{\hat{\boldsymbol{\sigma}}}$, then the shape and scale of the distribution depend only on the hyperparameters.

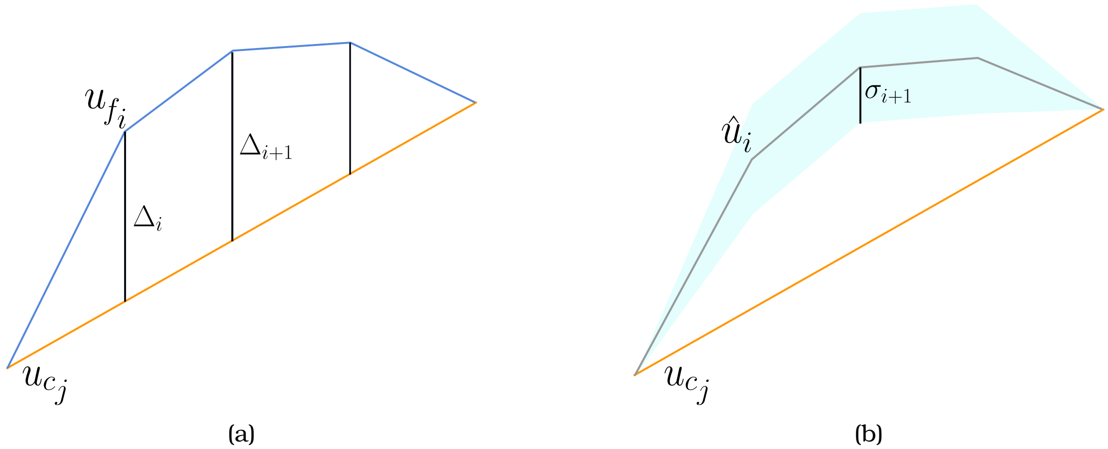


Figure 15: Example of discretization error between coarse and fine mesh, and standard deviation in a 1D problem. (a) Example of fine and coarse solutions and the discretization errors between both. (b) Example of posterior mean and posterior deviation in the fine mesh space.

The second measure is called $J_{\hat{\mathbf{u}}}$ and is half the sum of squared differences between the posterior mean and the fine mesh solution, as shown in Eq. (79).

$$J_{\hat{\mathbf{u}}} = \frac{1}{2}\sum_{i=1}^{n}\left(u_{f_i} - \hat{u}_i\right)^2 \tag{79}$$

Finally, since the objective is to minimize these two measures, they are used as objective functions, and the hyperparameters of each employed covariance matrix are the variables to optimize. To achieve this, an optimization algorithm that can use the Nelder Mead or L-BFGS-B algorithms was developed. The necessary gradients for the second method can be obtained following the procedure in Appendix (A.1).

To assess its effectiveness, a grid search is conducted over the hyperparameter α_0 of the prior unit mass matrix covariance. The problem used as an example is the one in Fig. (10). This grid search will cover the range from 10^{-3} to 10, with a grid size of $5x10^{-3}$ aiming to identify the α_0 corresponding to the MLE, the minimum $J_{\hat{\boldsymbol{\sigma}}}$, and the minimum $J_{\hat{\mathbf{u}}}$.

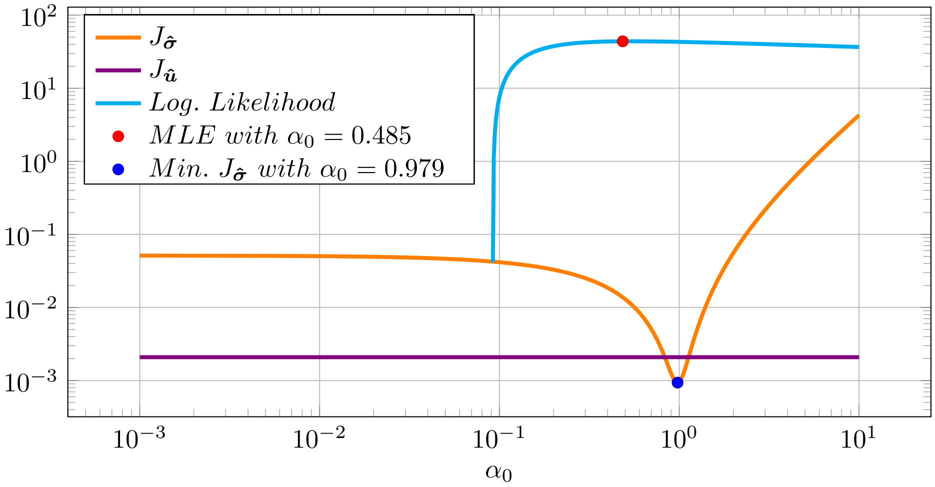


Figure 16: Plot showing the variation of $J_{\hat{\boldsymbol{\sigma}}}$, $J_{\hat{\mathbf{u}}}$, and Log-Likelihood for different values of α_0 of a unit mass matrix in the RHS.

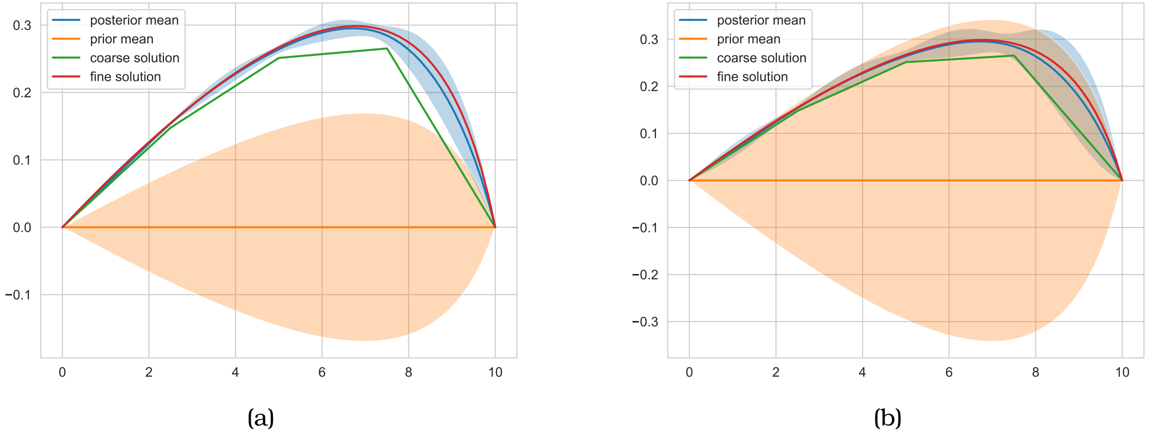


Figure 17: Comparision between the α_0 of the maximum MLE and the α_0 of the minimum $J_{\hat{\boldsymbol{\sigma}}}$. (a) distribution with the hyperparameter $\alpha_0 = 0.485$. (b) distribution with the hyperparameter $\alpha_0 = 0.979$.

As observed in Fig. (16), the differences between the α_0 of the maximum log-likelihood and the α_0 value that minimizes $J_{\hat{\boldsymbol{\sigma}}}$ is not negligible. To compare these two values, two distributions are shown in Figs. (17a) and (17b). It can be seen that when $J_{\hat{\boldsymbol{\sigma}}}$ is minimized, a better approximation of the posterior standard deviation, capturing the difference between the coarse and fine mesh solutions, is obtained.

When visualizing the results obtained by using the value of α_0 that maximizes the log-likelihood, it becomes apparent that the standard deviation fails to effectively capture the discretization error between the coarse and fine mesh with one deviation. A different outcome appears when employing the α_0 value that minimizes $J_{\hat{\boldsymbol{\sigma}}}$. A significantly clearer result is observed, where the posterior deviation shows the discretization error as a deviation.

A question that may arise is why not simply change the number of standard deviations shown so that the MLE correctly captures the discretization error? If it showed 2 deviations instead of 1, the MLE would appear notably accurate. The problem lies in that in this case both the number of standard deviations showed or the hyperparameter scales the posterior deviation. In problems with more hyperparameters, this becomes clearer because the main drawback with the MLE is that it makes the distribution fit the force vector data and does not necessarily match every standard deviation to each DOF discretization error.

As demonstrated in [18], changing the hyperparameter of a unit mass matrix does not alter the posterior mean. Considering the posterior mean in Eq. (80) for the case of an assumed uncertainty in the RHS derived in [18], it becomes clear that the hyperparameter has no influence on the posterior mean. This is because the hyperparameter cancels out, leaving only the matrix terms to determine the posterior mean. As $J_{\hat{\boldsymbol{u}}}$ is independent of the values of α_0 in Fig. (16), we have a constant function.

$$\hat{\boldsymbol{u}} = \boldsymbol{K}^{-1}\alpha^2\boldsymbol{A}\boldsymbol{\Phi}\left(\boldsymbol{\Phi}^T\alpha^2\boldsymbol{A}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{f} \tag{80}$$

So far, the unit mass prior presented has exhibited a constant standard deviation in the forcing term, with exceptions near the boundary conditions where deviations become zero. Because of this the hyperparameter scales the posterior distribution of the solution at each DOF, either amplifying or reducing their values. In the next chapter, we demonstrate that this is a drawback in other problems and also propose solutions for it.

4

Stationary and Non-Stationary Priors

This chapter analyzes the posterior distributions by optimizing the measures obtained in the previous chapter using the unit mass matrix prior covariance. With this prior the resulting distributions lack flexibility as they can be amplified or reduced by a single hyperparameter, making it challenging for the distribution to show variations of discretization errors. To address this complexity, an alternative approach to obtain flexible distributions is adopted.

4.1 Stationary priors

In this section, two distributions with stationary priors are obtained for the simply supported beam problem shown in Fig. (18), which is only subjected to gravity loads. The objective is to observe if they represent the discretization error of a 2D beam with the posterior standard deviation. In both cases, the hyperparameters are optimized in order to minimize $J_{\hat{\sigma}}$. The desired posterior standard deviation plot should be equal to Fig. (19). The optimization algorithm used to obtain the optimal hyperparameters by minimizing $J_{\hat{\sigma}}$ is Nelder–Mead.

The coarse and fine mesh used can be seen in Fig. (B1) in Annex (B.1). Fig. (19) shows the discretization error between a coarse mesh of 28 DOFs and a fine mesh of 1106 DOFs.

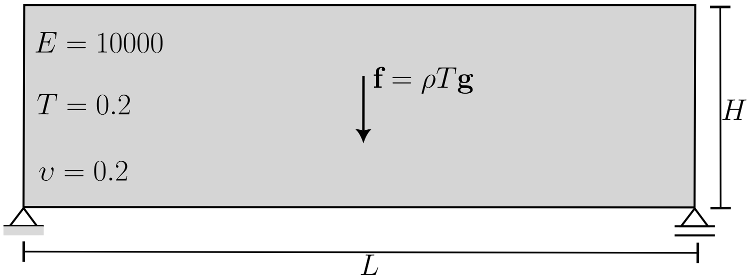


Figure 18: Simply supported 2D beam subjected to self-weight. Where $L = 10$, $H = 2$, the thickness is $T = 0.2$, the modulus of elasticity is $E = 10000$, the density $\rho = 1$, and the Poisson ratio is $\nu = 0.2$.

In the first example shown in Figs. (20a) and (20b) the prior is a unit mass matrix in the RHS. In Fig. (20a) the standard deviation is small in the center of the beam compared to the discretization error and the opposite happens near the supports. The discretization error between the coarse and fine mesh is not well captured with a prior that is only formed with the unit mass matrix.

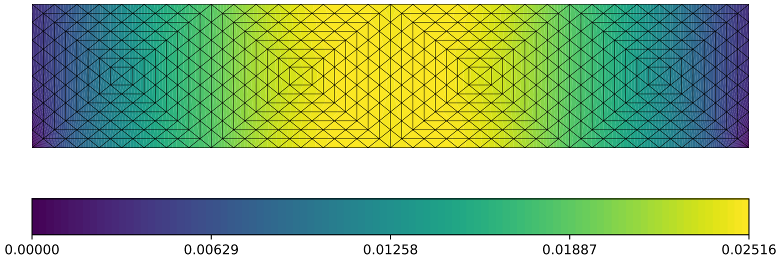


Figure 19: Discretization error between a coarse mesh of 28 DOFs and a fine mesh of 1106 DOFs.

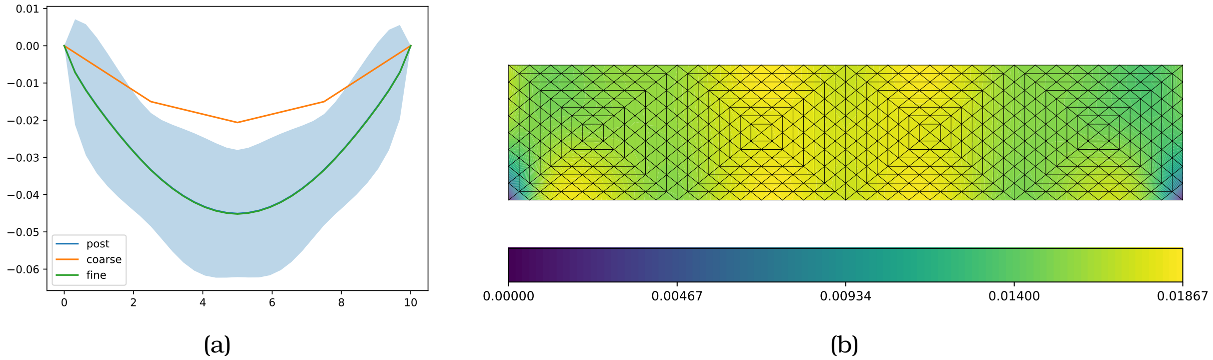


Figure 20: Probabilistic discretization error distribution based on the optimization of a unit mass matrix as a prior. (a) Posterior standard deviation in the lower edge of the 2D beam. (b) Posterior standard deviation in 2D beam based on a unit mass prior covariance matrix.

The second stationary example is a sum between a unit mass matrix and the stiffness matrix as the one in Eq. (76). The results are pictured in Figs. (21a) and (21b). This covariance is much better compared to only using a unit mass matrix since the standard deviation has bigger flexibility due to the combination of two different matrices. Moreover, the stiffness matrix term has the disadvantage that moving the posterior mean from the fine mesh solution. This happens because in the case of only using a stiffness matrix the posterior mean is the coarse solution.

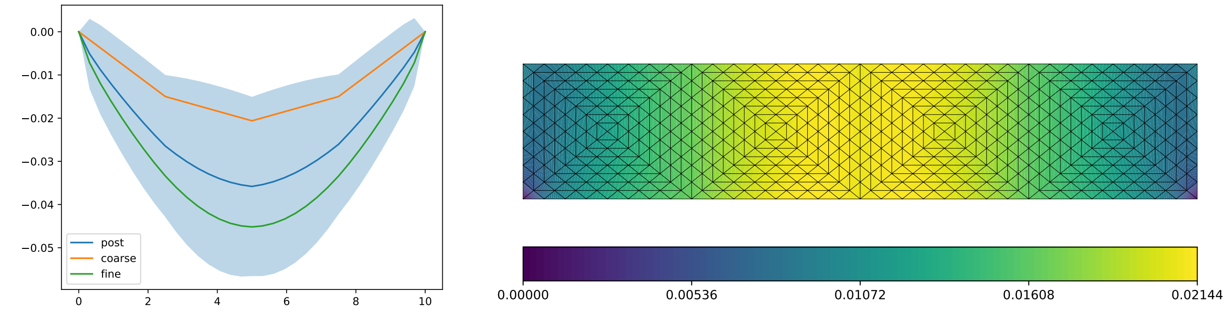
The results obtained in these two examples with stationary covariance matrices are not outstanding, this is because the discretization errors vary along the mesh and the posterior standard deviation distributions obtained do not have as much flexibility to capture that variation.

In the case of stationary sparse covariance matrices, where a single hyperparameter scales the matrix, the hyperparameter does not impact the posterior mean as it was explained in section (3.3). In this work, the sparse matrices utilized are the unit mass matrix and the stiffness matrix, the results with the mass matrix generate accurate outcomes, and the stiffness matrix results in a posterior mean matching the coarse solution. Consequently, when combining the unit mass matrix with the stiffness matrix in the $J_{\hat{\boldsymbol{u}}}$ optimization process, the hyperparameter associated with the stiffness matrix tends to approach zero, leaving only the mass matrix as a prior matrix.

4.2 Proposed non-stationary priors

Since usually, the discretization error varies along the mesh, we propose a non-stationary approach to improve the results of the sparse RHS matrices. By doing this, the prior will already include spatial variations. One way to include them involves introducing a function $\eta(\boldsymbol{x})$ that represents the variation of the error along the mesh. In this case, the prior unit mass covariance matrix can be formulated as:

$$\boldsymbol{f} \sim \mathcal{N}\left(\boldsymbol{0}, \left(\alpha_1^2 \eta(\boldsymbol{x_i})\eta(\boldsymbol{x_j}) + \alpha_0^2\right) M_{ij}\right) \tag{81}$$



(a)

(b)

Figure 21: Probabilistic discretization error distribution based on the optimization of the sum of a unit mass matrix and a stiffness matrix as a prior. (a) Posterior standard deviation in the lower edge of the 2D beam. (b) Posterior standard deviation in 2D beam with a unit mass and stiffness prior covariance matrix.

In Eq. (81), α_1 is a hyperparameter that amplifies the non-stationary effect, while α_0 accounts for the stationarity of the unit mass matrix. This prior covariance can be extended to the combination of unit mass matrices and stiffness matrices:

$$\boldsymbol{f} \sim \mathcal{N}\left(\mathbf{0}, \left(\alpha_1^2 \eta(\boldsymbol{x_i})\eta(\boldsymbol{x_j}) + \alpha_0^2\right) M_{ij} + \left(\beta_1^2 \eta(\boldsymbol{x_i})\eta(\boldsymbol{x_j}) + \beta_0^2\right) K_{ij}\right) \tag{82}$$

Where i and j are the DOFs where the matrix is evaluated. One challenge with the aforementioned approach is designing a suitable function $\eta(\boldsymbol{x_i})$ that is both general and flexible while keeping the number of hyperparameters small to avoid complicating the optimization process. As an alternative, the coarse mesh solution can be utilized:

$$\boldsymbol{f} \sim \mathcal{N}\left(\mathbf{0}, \left(\alpha_1^2 (u_i u_j)^2 + \alpha_0^2\right) M_{ij}\right) \tag{83}$$

$$\boldsymbol{f} \sim \mathcal{N}\left(\mathbf{0}, \left(\alpha_1^2 (u_i u_j)^2 + \alpha_0^2\right) M_{ij} + \left(\beta_1^2 (u_i u_j)^2 + \beta_0^2\right) K_{ij}\right) \tag{84}$$

Since the coarse mesh has fewer DOFs than the fine space, it is necessary to project the coarse solution onto the fine mesh space. Therefore, the values u_i and u_j are obtained by linearly interpolating at the positions of the fine mesh using the $\mathbf{\Phi}$ matrix. Therefore, $u_i = \sum_{k=0}^{m} \Phi_{ki} u_{c_k}$ or in matrix form $\boldsymbol{u} = \mathbf{\Phi}\boldsymbol{u_c}$.

By incorporating the coarse solution, the method can effectively capture regions with high values where the discretization error tends to be most significant. Additionally, if these functions have limited utility for a specific problem, the hyperparameters are allowed to go zero, and only the unit mass matrix or the stiffness matrix will remain, and the distribution can be similar to the stationary case.

4.3 Results of non-stationary priors

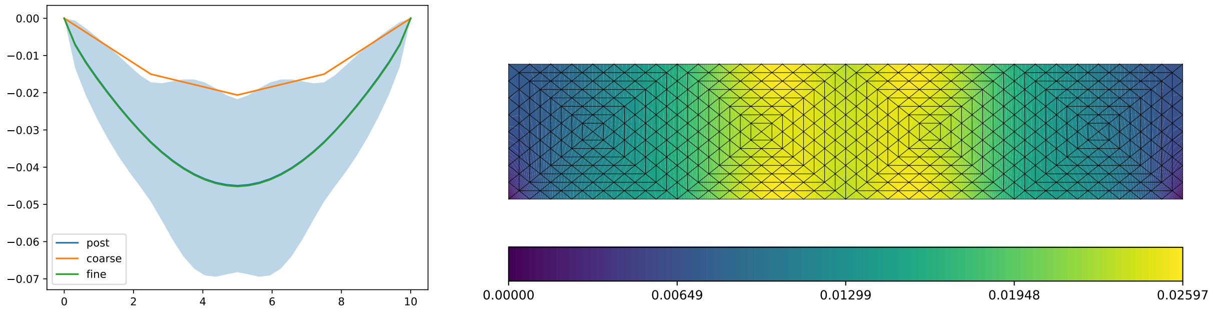
4.3.1 Posterior deviation

In this section, the proposed non-stationary covariance functions are applied to the example of Fig. (18), and at the end a comparison between the previous stationary matrices and the proposed ones is made. The optimization algorithm used to obtain the minimum values of the $J_{\hat{\boldsymbol{\sigma}}}$ in this case, is Nelder–Mead.

Knowing that the discretization error is higher at the center of the beam and almost zero near the supports as shown in Fig. (19) the function $\eta(\boldsymbol{x_i})$ from Eq. (82) is substituted with a sine function that only varies along the x axis, and it has its maximum value in the middle of the beam. Then, the prior covariance is

$$\boldsymbol{f} \sim \mathcal{N}\left(\mathbf{0}, \left(\alpha_1^2 \sin\left(\frac{x_i \pi}{10}\right) \sin\left(\frac{x_j \pi}{10}\right) + \alpha_0^2\right) M_{ij}\right) \tag{85}$$

As can be seen in Figs. (22a) and (22b) the results are favorable since after optimizing the hyperparameters the discretization error is well captured by the standard deviation.



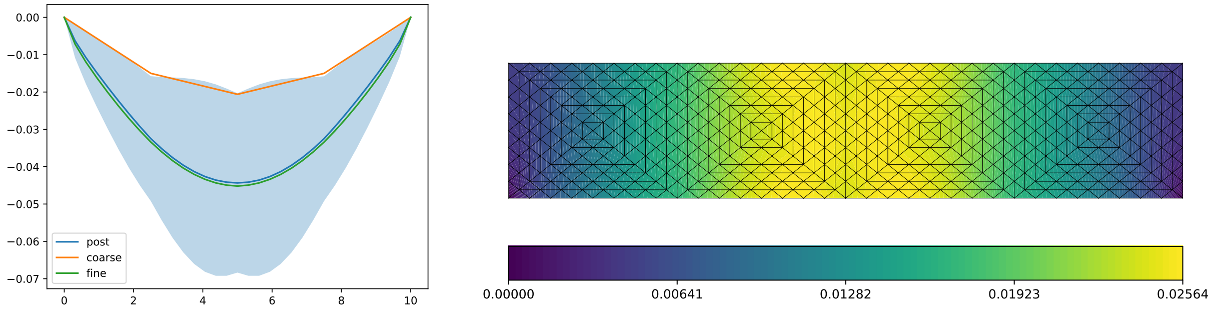
(a)

(b)

Figure 22: Probabilistic discretization error distribution based on the optimization of the product of a unit mass matrix and a non-stationary covariance sine function as a prior. (a) Posterior standard deviation in the lower edge of the 2D beam. (a) Posterior standard deviation of 2D.

By adding a stiffness matrix that is also multiplied by the same non-stationary covariance function, the results are further improved as shown in Figs. (23a) and (23b). Then, the covariance matrix is formed with the function:

$$\boldsymbol{f} \sim \mathcal{N}\left(\mathbf{0}, \left(\alpha_1^2 \sin\left(\frac{x_i \pi}{10}\right) \sin\left(\frac{x_j \pi}{10}\right) + \alpha_0^2\right) M_{ij} + \left(\beta_1^2 \sin\left(\frac{x_i \pi}{10}\right) \sin\left(\frac{x_j \pi}{10}\right) + \beta_0^2\right) K_{ij}\right) \tag{86}$$



(a)

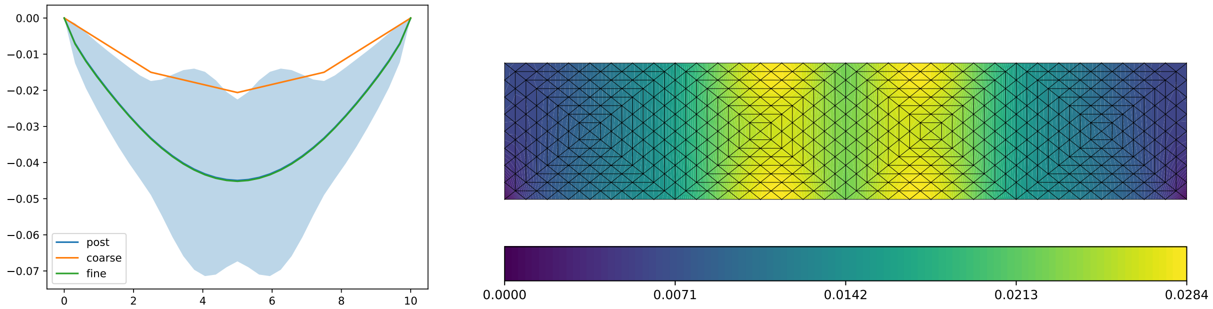
(b)

Figure 23: Probabilistic discretization error distribution based on the optimization of combining a unit mass matrix multiplied by a non-stationary covariance sine function with a stiffness matrix multiplied by a non-stationary covariance sine function as a prior. (a) Posterior standard deviation in the lower edge of the 2D beam. (b) Posterior standard deviation of the 2D beam.

The results of both covariance matrices are suitable. Nonetheless, it is easy to propose this non-stationary function since the discretization error, in this case, is known. In the next two examples, the covariance functions are formed with the use of the coarse mesh solution. The distribution obtained when using the prior of Eq. (83) is shown in Figs. (24a) and (24b). It can be observed, that the distribution with the prior formed by the mass matrix and the non-stationarity function produced by the coarse solution is suitable.

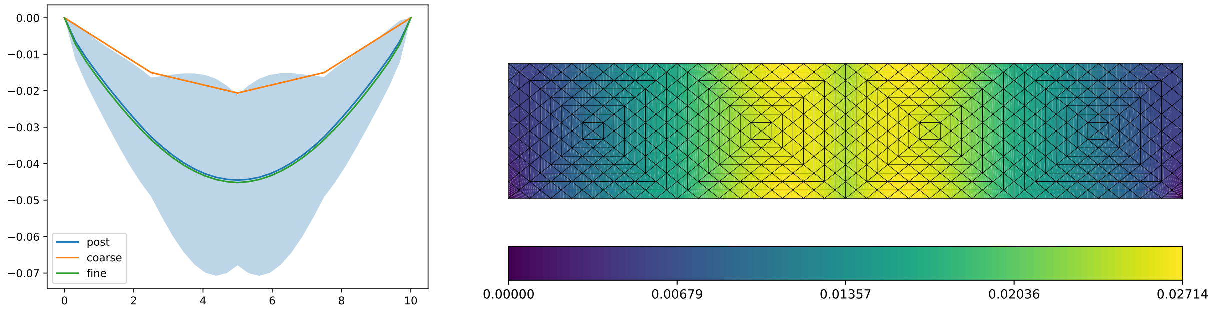
The last example shown in Figs. (25a) and (25b) uses the coarse mesh solution and incorporates the stiffness matrix, as shown in Eq. (84).

Table (1) presents the six optimized $J_{\hat{\sigma}}$ with the corresponding hyperparameters. The results demonstrate that incorporating a combination of mass and stiffness matrices leads to a reduction in the $J_{\hat{\sigma}}$ value. An even more significant reduction is observed when employing non-stationary covariance matrices, both with the proposed function and the coarse solution. On the whole, the results show a better capacity of non-stationary matrices in enhancing the distributions.



(a) (b)

Figure 24: Probabilistic discretization error distribution based on the optimization of the product of a unit mass matrix and a non-stationary covariance function form with coarse mesh solution as a prior. (a) Posterior standard deviation in the lower edge of 2D beam. (b) Posterior standard deviation of the 2D beam.



(a) (b)

Figure 25: Probabilistic discretization error distribution based on the optimization of adding the product of a unit mass matrix and a non-stationary covariance form of coarse mesh solution with the product of a stiffness matrix and a non-stationary covariance form of coarse mesh solution as a prior. (a) Posterior standard deviation in the lower edge of the 2D beam. (b) Posterior standard deviation of the 2D beam.

The non-stationary covariance functions in this problem tend to reduce $J_{\hat{\sigma}}$ by almost an order of magnitude. The hyperparameter values associated with the coarse solution tend to be high simply because the non-stationary function depends on the displacements that may be low. When the stiffness matrix is incorporated the posterior mean tends to separate from the fine solution because the prior incorporates the stiffness matrix.

4.3.2 Posterior mean

If a non-stationary sparse matrix that only has one hyperparameter multiplying the matrix is constructed, the hyperparameter can also be eliminated as in Eq. (80), resulting in the posterior mean depending solely on the unit terms of the matrix, independent of the amplifying hyperparameter. Therefore, when considering each case of a matrix multiplied by a hyperparameter separately, the $J_{\hat{u}}$ value remains constant. This measure, chosen in section (3.3) to assess the proximity of the posterior mean to a fine mesh solution, remains constant despite variations in the hyperparameter.

The objective of this section is to investigate whether the combination of a non-stationary covariance matrix with a stationary one can enhance the posterior mean solution or if the solution converges towards the minimum generated by either one matrix alone canceling the other matrix. To explore this, the combination of the two matrices presented in Eq. (87) is utilized. The beam in Fig. (18) is used again as an example and the Nelder-Mead method is utilized to optimize the

Prior covariance matrix	$J_{\hat{\boldsymbol{\sigma}}}$	α_0	α_1	β_0	β_1
$\alpha_0^2 \boldsymbol{M}$	2.089e-02	1.102e+01	-	-	
$\alpha_0^2 \boldsymbol{M} + \beta_0^2 \boldsymbol{K}$	5.482e-03	6.930e-01	-	1.590e-01	-
$\left(\alpha_1^2 \sin\left(\frac{x_i \pi}{10}\right) \sin\left(\frac{x_j \pi}{10}\right) + \alpha_0^2\right) M_{ij}$	1.485e-03	1.564e-05	2.490e+01	-	-
$\left(\alpha_1^2 \sin\left(\frac{x_i \pi}{10}\right) \sin\left(\frac{x_j \pi}{10}\right) + \alpha_0^2\right) M_{ij} +$ $\left(\beta_1^2 \sin\left(\frac{x_i \pi}{10}\right) \sin\left(\frac{x_j \pi}{10}\right) + \beta_0^2\right) K_{ij}$	1.058e-03	6.934e-06	1.478e+00	2.197e-07	1.930e-01
$\left(\alpha_1^2 (u_i u_j)^2 + \alpha_0^2\right) M_{ij}$	1.601e-03	3.744e+00	9.064e+04	-	-
$\left(\alpha_1^2 (u_i u_j)^2 + \alpha_0^2\right) M_{ij} +$ $\left(\beta_1^2 (u_i u_j)^2 + \beta_0^2\right) K_{ij}$	8.293e-04	7.516e-03	1.201e+04	4.582e-02	7.671e+02

Table 1: Optimized $J_{\hat{\boldsymbol{\sigma}}}$ for different prior covariance.

hyperparameters.

$$\hat{\boldsymbol{u}} = \boldsymbol{K}^{-1} \left(\alpha_0^2 \boldsymbol{M} + \alpha_1^2 \boldsymbol{M}_{NS}\right)\right) \boldsymbol{\Phi} \left(\boldsymbol{\Phi}^T (\alpha_0^2 \boldsymbol{M} + \alpha_1^2 \boldsymbol{M}_{NS}) \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^T \boldsymbol{f} \tag{87}$$

where $\boldsymbol{M}_{NS}$ is a matrix conformed with the combination of a mass matrix and the multiplication of the covariance function in each DOF. This function can be a predefined function like a sine or another approach like the coarse mesh solution. Table (2) presents the results of the hyperparameter values used for the mass matrix. Regardless of the value chosen, the same value of $J_{\hat{\boldsymbol{u}}}$ is obtained (that is why there is no value of the hyperparameter in α_0 of that row), as the posterior mean is independent of the hyperparameter. For both non-stationary cases, it can be observed that the hyperparameters multiplying the non-stationary matrix tend towards zero. This implies that, instead of improvement, the inclusion of the non-stationary matrix leads to a worsening of the $J_{\hat{\boldsymbol{u}}}$ result.

Prior covariance matrix	$J_{\hat{\boldsymbol{u}}}$	α_0	α_1
$\alpha_0^2 \boldsymbol{M}$	5.016e-06	-	-
$\left(\alpha_1^2 \sin\left(\frac{x_i \pi}{10}\right) \sin\left(\frac{x_j \pi}{10}\right) + \alpha_0^2\right) M_{ij}$	5.016e-06	1.214e+0	2.531e-04
$\left(\alpha_1^2 (u_i u_j)^2 + \alpha_0^2\right) M_{ij}$	5.016e-06	1.102e+01	9.647e-07

Table 2: Optimized $J_{\hat{\boldsymbol{u}}}$ for different prior covariance in the 2D beam.

To explain these observations, it is important to revisit [18], which mentions the existence of a projection matrix $\boldsymbol{P}$:

$$\hat{\boldsymbol{f}} = \underbrace{\alpha^2 \boldsymbol{A} \boldsymbol{\Phi} \left(\boldsymbol{\Phi}^T \alpha^2 \boldsymbol{A} \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^T}_{\boldsymbol{P}} \boldsymbol{f} = \boldsymbol{P} \boldsymbol{f} \tag{88}$$

The non-stationarity of the matrices used in this study worsens the projection of the original force vector $\boldsymbol{f}$ to the estimated force vector $\hat{\boldsymbol{f}}$. If the projection matrix $\boldsymbol{P}$ were to be equal to the identity matrix $\boldsymbol{I}$ (in an ideal scenario), $\hat{\boldsymbol{f}}$ would be equal to $\boldsymbol{f}$.

It is worth mentioning that since the posterior solution tends to converge to the coarse solution with the use of the stiffness matrix as the prior, adding the stationary matrix in an attempt to improve the posterior mean solution only cancels out the hyperparameters in the optimization process, resulting in convergence towards the unit mass matrix. Therefore, none of the covariance priors that have a stiffness matrix are included in the analysis.

To study how much the posterior mean varies with respect to the fine mesh solution after minimizing $J_{\hat{\boldsymbol{\sigma}}}$ we use the optimized hyperparameters of Table (1), and compute the corresponding $J_{\hat{\boldsymbol{u}}}$ values, we can summarize the results in Table (3).

Prior covariance matrix	$J_{\hat{\boldsymbol{\sigma}}}$	$J_{\hat{\boldsymbol{u}}}$
$\alpha_0^2\boldsymbol{M}$	2.089e-02	5.016e-06
$\alpha_0^2\boldsymbol{M} + \beta_0^2\boldsymbol{K}$	5.482e-03	1.619e-02
$\left(\alpha_1^2 \sin\left(\frac{x_i\pi}{10}\right)\sin\left(\frac{x_j\pi}{10}\right) + \alpha_0^2\right) M_{ij}$	1.485e-03	1.381e-05
$\left(\alpha_1^2 \sin\left(\frac{x_i\pi}{10}\right)\sin\left(\frac{x_j\pi}{10}\right) + \alpha_0^2\right) M_{ij} +$ $\left(\beta_1^2 \sin\left(\frac{x_i\pi}{10}\right)\sin\left(\frac{x_j\pi}{10}\right) + \beta_0^2\right) K_{ij}$	1.058e-03	3.781e-04
$\left(\alpha_1^2(u_iu_j)^2 + \alpha_0^2\right) M_{ij}$	1.601e-03	1.690e-05
$\left(\alpha_1^2(u_iu_j)^2 + \alpha_0^2\right) M_{ij} +$ $\left(\beta_1^2(u_iu_j)^2 + \beta_0^2\right) K_{ij}$	8.293e-04	2.598e-04

Table 3: $J_{\hat{\boldsymbol{u}}}$ values for optimized priors of $J_{\hat{\boldsymbol{\sigma}}}$ of Table (1)

Among the obtained values, the lowest $J_{\hat{\boldsymbol{u}}}$ correspond to cases where the stationary matrix is only the unit mass matrix, especially the lowest is the case of only the unit mass matrix. This indicates that whenever we introduce the stiffness matrix in the prior, it tends to cause a deviation between the posterior solution and the solution obtained using a finer mesh. In addition, by introducing a non-stationary term in the prior and seeking to minimize the $J_{\hat{\boldsymbol{\sigma}}}$, we obtain a sub-optimal result for $J_{\hat{\boldsymbol{u}}}$.

5

Impact of Meshing and Sampling

In this chapter, we extend our investigation by exploring mesh refinement, variation of the arrangement in the mesh, and sampling to challenge the findings from previous chapters. We aim to determine if the optimization results obtained in the previous chapters hold consistency even when we modify the mesh arrangement and refine it differently. By analyzing these variations, we can assess if the optimal hyperparameter for the first refinement closely aligns with that of a very fine mesh. In that case, a strategy could be to optimize it for a few degrees of freedom and then extrapolate this estimate to the finer mesh. If the optimal hyperparameters change significantly with different mesh arrangements, we will understand that these optimal values cannot be extrapolated to different meshes. Additionally, we delve into investigating whether the optimal hyperparameters remain consistent when covariance matrices are assembled with samples. In [18], ensemble posterior covariances are presented as an option to reduce the number of computations and storage. However, there has been no study on determining the number of samples needed to obtain covariance matrices with satisfactory performance. This analysis will provide valuable insights into the efficiency and accuracy of using ensemble posterior covariances, contributing to better computational strategies.

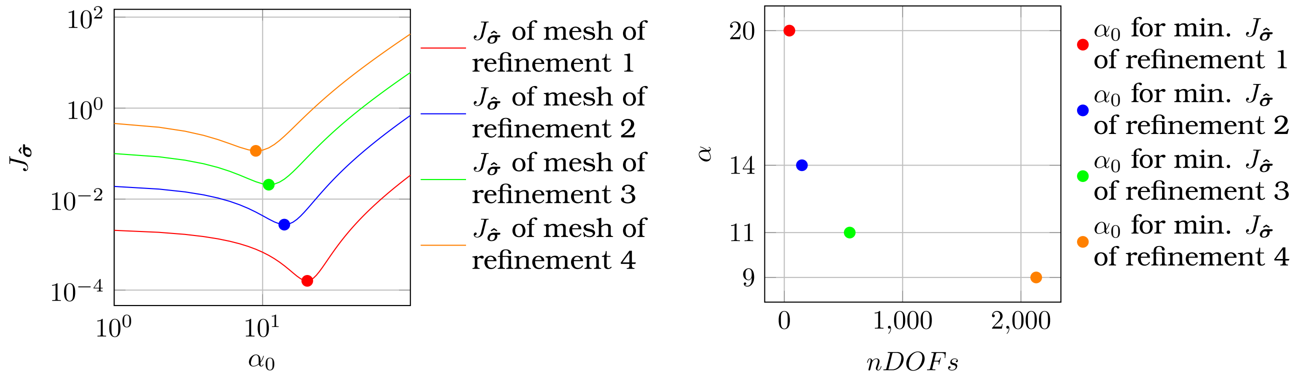
5.1 Meshing and its impact on error representation

In this section, we investigate the variation of the optimal hyperparameters and how the optimized posterior distribution can represent the error for different refinements of the same coarse mesh, as well as variations in the arrangement of finite elements for the same problem. In both cases, we use the problem shown in Fig. (18).

To examine the variation of the optimal hyperparameters after refining the mesh, we performed a grid search of the hyperparameter α_0, The range is from 1 to 100, with a step size of 1. We analyzed the variation of $J_{\hat{\sigma}}$ for a prior unit mass covariance matrix. The coarse mesh used is No. 1, as depicted in Appendix (B.1) in Fig. (B2) and the refinements of this mesh only reduce the size of the elements, keeping the triangular element type without changing the shape or interpolation functions. As an example of how the refinements look like, for the case of coarse mesh No. 1, they can be seen in Fig. (B1).

Fig. (26a) displays the four different grids along with the points representing the minimum values of $J_{\hat{\sigma}}$. On the other hand, Fig. (26b) illustrates the variation of the optimal α_0 as the number of DOFs increases in the fine mesh. As can be observed, there is a clear pattern in $J_{\hat{\sigma}}$ that remains consistent for each refinement, and both the minimum $J_{\hat{\sigma}}$ and the optimal hyperparameters vary in each case.

For the case of varying the arrangement of the elements, twelve distinct computational experiments are conducted on the beam problem illustrated in Fig. (18). The same grid search as in Fig.



(a)

(b)

Figure 26: $J_{\hat{\sigma}}$ grid of 2D beam for different mesh refinements and a prior unit mass matrix covariance in RHS. (b) Variation of the four $J_{\hat{\sigma}}$ functions for each α_0 grid search with their optimal α_0 and minimum $J_{\hat{\sigma}}$. (a) Minimum α_0 values for different DOFs of fine meshes.

(26) is performed, ranging the hyperparameter α_0 from 1 to 100 with a step of 1. That is why the minimum hyperparameter for each case in this study is an integer number. The coarse meshes used are shown in Fig. (B2). Each of these grids underwent four refinement iterations as in the example shown before in Fig. (26b). Therefore, a total of 48 $J_{\hat{\sigma}}$ functions are plotted, representing 4 refinements of the 12 meshes.

The grids of the twelve meshes are shown in Fig. (27) in which there are 4 boxes, one for each of the refinements, and in each one is shown how much the $J_{\hat{\sigma}}$ varies for each mesh. It can be seen how some grids are more capable than others in representing the error with the posterior standard deviation since they have smaller minima than the rest.

To summarize the results of these twelve analyses and their refinements, two tables are created. Table (4) presents the minimum α_0 values for each refinement of a specific coarse mesh, while Table (5) displays the minimum $J_{\hat{\sigma}}$. The minima of $J_{\hat{\sigma}}$ can be different for the same number of DOFs and different arrangements. Furthermore, in almost all cases, it happens that the optimal hyperparameters change in at least 10% when refining the mesh.

Mesh	Fine mesh 43 nodes	Fine mesh 149 nodes	Fine mesh 553 nodes	Fine mesh 2129 nodes
Coarse mesh 1	20	14	11	9
Coarse mesh 2	20	14	11	9
Coarse mesh 3	20	14	11	9
Coarse mesh 4	19	13	11	9
Coarse mesh 5	18	14	12	11
Coarse mesh 6	15	13	12	11
Coarse mesh 7	17	13	10	8
Coarse mesh 8	16	12	9	8
Coarse mesh 9	16	12	9	8
Coarse mesh 10	14	12	10	8
Coarse mesh 11	14	13	12	12
Coarse mesh 12	14	13	12	12

Table 4: Optimized α_0 of prior covariance unit mass matrix for different meshes and refinements of the 2D beam.

In order to understand graphically how different meshes facilitate the possibility of obtaining a posterior standard deviation that shows the discretization error with the unit mass matrix prior,

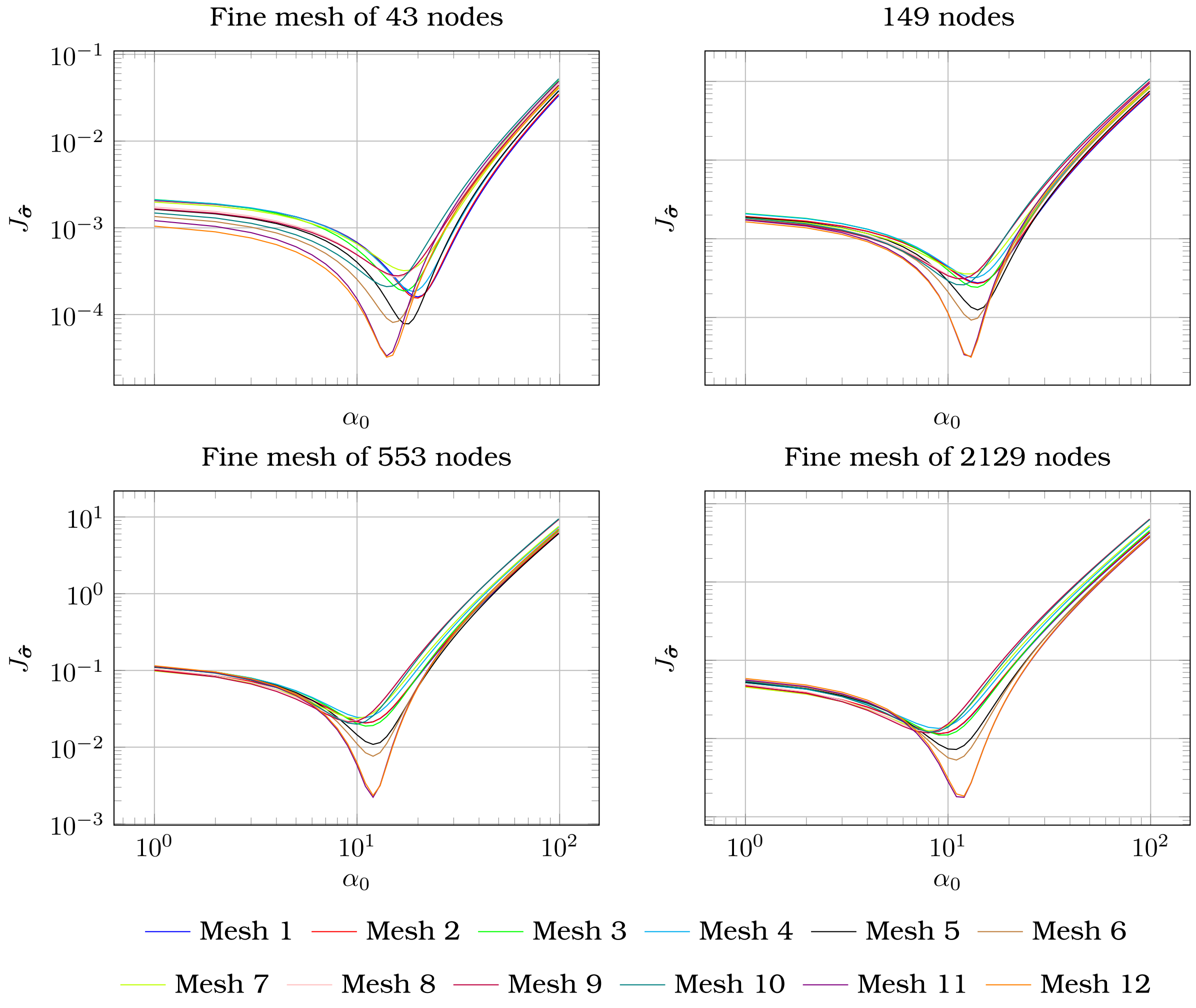


Figure 27: Summary of the grids search of $J_{\hat{\sigma}}$ performed on the 12 coarse meshes and their 4 refinements for a 2D beam and with a mass unit prior covariance.

the optimal result of the first refinement of mesh 7, which is the one with the lowest performance, and mesh 12, which is one of the best are shown in Fig. (28).

Mesh No. 7 exhibits inefficiency in capturing the discretization error of a fine mesh of 43 nodes. This can be seen both in the lower edge of Fig. (28a) as well as in the contour plot in Fig. (28b) where it can be seen how the standard deviations are not showing well the discretization error. Mesh No. 12 demonstrates remarkable proficiency in this regard. In the lower edge of Fig. (28c) the standard deviation almost perfectly captures the discretization error and the same happens in the contour plot of Fig. (28d) where this is also noticeable.

The reason why some meshes outperform others lies in the fact that in certain areas of the mesh, the discretization error is lower compared to others. By reducing the size of the elements in those areas, the values of the mass matrix are also reduced, and consequently, the subsequent standard deviations are as well. In other words, modifying the mesh transforms the discretization error of the problem as well as the posterior standard deviation.

As a result of this analysis, we can conclude that mesh variation changes the discretization error and the structure of the posterior covariance matrix. Therefore, the optimal hyperparameters may not be transferable between refinements of the same coarse mesh or changes in the mesh arrangement for the same problem.

Now that we already know that the optimal hyperaprameters change as the mesh is refined or have other values for different mesh arrangements of the same problem, another question that

Mesh	Fine mesh 43 nodes	Fine mesh 149 nodes	Fine mesh 553 nodes	Fine mesh 2129 nodes
Coarse mesh 1	1.597E-04	2.749E-03	2.075E-02	1.151E-01
Coarse mesh 2	1.550E-04	2.694E-03	2.060E-02	1.151E-01
Coarse mesh 3	1.870E-04	2.419E-03	1.884E-02	1.109E-01
Coarse mesh 4	1.847E-04	3.201E-03	2.416E-02	1.347E-01
Coarse mesh 5	7.820E-05	1.245E-03	1.087E-02	7.255E-02
Coarse mesh 6	8.110E-05	9.242E-04	7.615E-03	5.307E-02
Coarse mesh 7	3.207E-04	3.577E-03	2.367E-02	1.258E-01
Coarse mesh 8	2.700E-04	3.119E-03	2.189E-02	1.215E-01
Coarse mesh 9	2.798E-04	3.115E-03	2.127E-02	1.179E-01
Coarse mesh 10	2.098E-04	2.597E-03	1.993E-02	1.212E-01
Coarse mesh 11	3.330E-05	3.189E-04	2.223E-03	1.774E-02
Coarse mesh 12	3.220E-05	3.119E-04	2.350E-03	1.841E-02

Table 5: Minimal $J_{\hat{\sigma}}$ of prior covariance unit mass matrix for different meshes and refinements of the 2D beam.

arises is how much does the analysis change if a non-stationary prior is used? In this case, since the computational cost of performing the analysis with grids search is higher, the optimized hyperparameter pairs for each case of Eq. (83) are found. Basically, the combinations of the hyperparameter α_0 and α_1 are found for each mesh arrangement, but in this case, only three refinements of the same mesh are done.

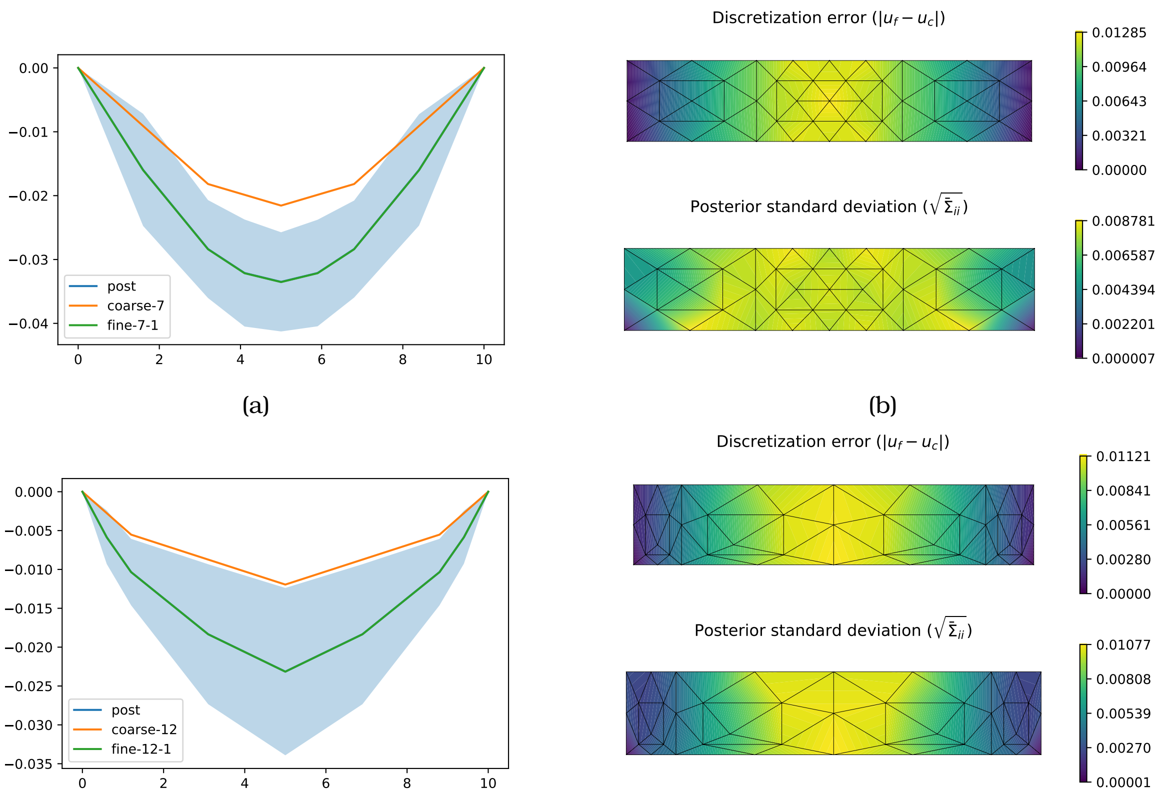
In the Table (6) and (7) the results for α_0 and α_1 are shown. What can be observed is that there are still non-negligible variations in the hyperparameters. However, the variations in percentage terms are minor compared to the case with only the unit mass matrix as prior.

Mesh	Fine mesh 43 nodes	Fine mesh 149 nodes	Fine mesh 553 nodes
Coarse mesh 1	4.35	4.45	3.74
Coarse mesh 2	4.27	4.38	3.65
Coarse mesh 3	2.23	3.64	3.30
Coarse mesh 4	3.95	3.85	3.24
Coarse mesh 5	4.81	5.52	5.25
Coarse mesh 6	4.91	5.65	5.55
Coarse mesh 7	1.19	0.00	0.00
Coarse mesh 8	0.00	0.00	0.00
Coarse mesh 9	0.00	0.00	0.00
Coarse mesh 10	3.87	4.30	3.83
Coarse mesh 11	8.16	8.61	8.21
Coarse mesh 12	8.91	8.59	8.16

Table 6: Optimized α_0 of prior covariance unit mass matrix for different meshes and refinements of the 2D beam.

Finally, when analyzing the results of the $J_{\hat{\sigma}}$ in Table (8), it can be seen that there are no meshes that are significantly superior to the others, as was the case when using a stationary matrix. This can be observed particularly when comparing the results of Mesh No. 7 with Mesh No. 12, in which in this case the difference is less than 10%, whereas, with only the stationary mesh, the difference is almost 8 times different.

It is evident that in Table (7) there are many hyperparameters α_0 that are zero, this is because, in the optimization process, the results were negligible. This result has logic because the



(a)

(b)

(c)

(d)

Figure 28: Comparison of discretization error and posterior standard deviation for the 14-node coarse mesh 7 and 12, and their 43-node fine mesh. (a) Posterior standard deviation in the lower edge of the 2D beam of mesh No. 7. (b) Posterior standard deviation and discretization error contour plot of mesh No. 7. (c) Posterior standard deviation in the lower edge of the 2D beam of mesh No. 12. (d) Posterior standard deviation and discretization error contour plot of mesh No. 12.

Mesh	Fine mesh 43 nodes	Fine mesh 149 nodes	Fine mesh 553 nodes
Coarse mesh 1	1.03E+05	9.14E+04	9.06E+04
Coarse mesh 2	1.05E+05	9.29E+04	9.21E+04
Coarse mesh 3	1.33E+05	1.14E+05	1.11E+05
Coarse mesh 4	1.24E+05	1.11E+05	1.10E+05
Coarse mesh 5	1.30E+05	1.08E+05	1.02E+05
Coarse mesh 6	1.54E+05	1.24E+05	1.16E+05
Coarse mesh 7	1.33E+05	1.20E+05	1.16E+05
Coarse mesh 8	1.71E+05	1.43E+05	1.26E+05
Coarse mesh 9	1.63E+05	1.34E+05	1.18E+05
Coarse mesh 10	1.23E+05	1.04E+05	1.00E+05
Coarse mesh 11	1.62E+05	1.20E+05	1.16E+05
Coarse mesh 12	2.02E+05	1.56E+05	1.49E+05

Table 7: Optimized α_1 of prior covariance unit mass matrix for different meshes and refinements of the 2D beam.

Mesh	Fine mesh 43 nodes	Fine mesh 149 nodes	Fine mesh 553 nodes
Coarse mesh 1	3.849E-05	2.987E-04	1.601E-03
Coarse mesh 2	3.678E-05	2.854E-04	1.526E-03
Coarse mesh 3	2.685E-05	2.126E-04	1.138E-03
Coarse mesh 4	3.525E-05	2.768E-04	1.517E-03
Coarse mesh 5	2.501E-05	2.326E-04	1.397E-03
Coarse mesh 6	2.105E-05	2.076E-04	1.294E-03
Coarse mesh 7	4.421E-05	2.289E-04	1.440E-03
Coarse mesh 8	7.159E-05	6.749E-04	4.870E-03
Coarse mesh 9	6.111E-05	6.258E-04	4.471E-03
Coarse mesh 10	4.308E-05	4.186E-04	2.550E-03
Coarse mesh 11	2.248E-05	2.156E-04	1.340E-03
Coarse mesh 12	2.278E-05	2.073E-04	1.344E-03

Table 8: Minimal $J_{\hat{\sigma}}$ of non-stationary prior covariance unit mass matrix for different meshes and refinements of 2D beam.

hyperparameter α_0 is associated with the effect of the stationary mass matrices that have a low performance when they are the only term of the prior covariance. Therefore, in the optimization, only the non-stationary term ends up remaining.

5.2 Minimum number of samples for ensemble

In the paper [18] the use of Kalman filters is proposed as an option to avoid computing complete covariance matrices of size $m \times m$ and instead utilize matrices obtained from a reduced number n of samples of size m. Therefore, the advantage lies in the fact that the required information for storage and operations is now of size $m \times n$. However, the investigation into the number of samples required to accurately represent the posterior distribution is still pending, whether to obtain a standard deviation similar to the discretization error or a mean value similar to the fine space solution.

To achieve this, the hyperparameters of ensemble prior covariance matrices for the RVE structure shown in Fig. (29) are optimized. Multiple cases are analyzed, each with a different number of samples. To conduct a comprehensive study, numerous initializations with the same number of samples are performed. As the number of samples increases, these initializations are reduced, as the results tend to improve and the computational cost of optimizing the ensemble covariance increases due to the number of samples.

The covariance matrices used to optimize the posterior standard deviation and posterior mean are the unit matrix as defined in Eq. (70) and the non-stationary matrix as described in Eq. (83).

The coarse and fine mesh used can be seen in Fig. (B3) in Annex (B.2). The coarse mesh has 44 nodes and the fine mesh has 147 nodes.

5.2.1 Solutions after optimizing the posterior standard deviation

In this subsection, the goal is to examine the convergence of the $J_{\hat{\sigma}}$ values obtained from sample ensemble matrices with those obtained analytically. As the number of samples increases the $J_{\hat{\sigma}}$ and hyperparameters should converge to those that would be obtained analytically. In this analysis, it is assumed that $J_{\hat{\sigma}}$ results above or below 10% of those obtained analytically are favorable.

Fig. (30a) shows the discretization error between the meshes used and Figs. (30b) and (30c) show the posterior standard deviations of the two matrices used after optimizing the $J_{\hat{\sigma}}$. The first image is the posterior deviation with a unit mass covariance matrix and the second is the solution with a non-stationary covariance function that combines the coarse solution with the

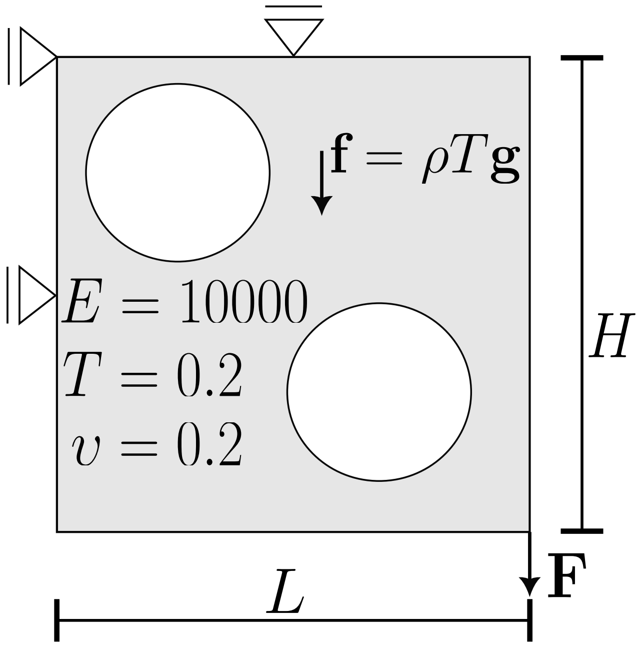


Figure 29: 2D plate with voids and complex boundary conditions.

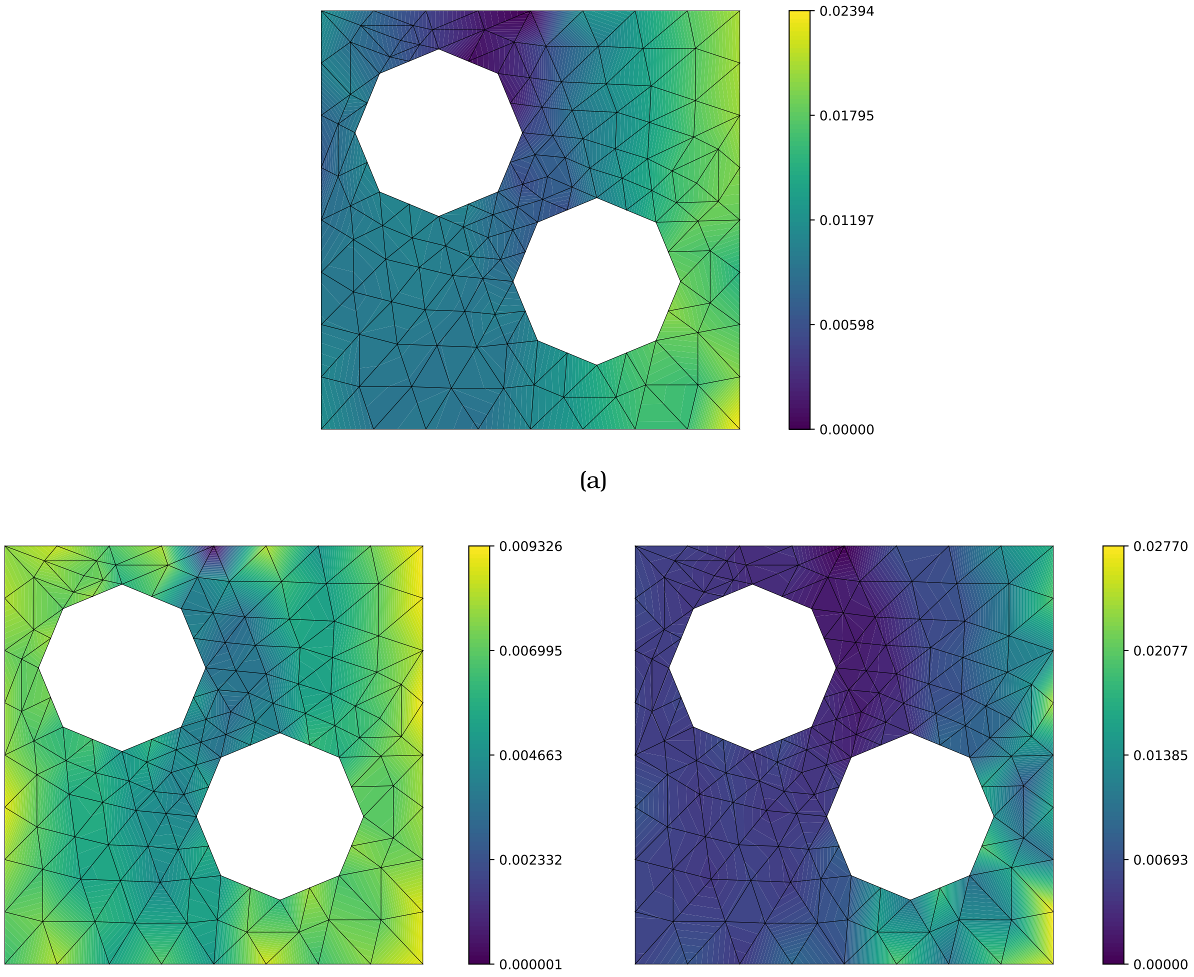
mass matrix.

Algorithm 1 Convergence analysis of ensemble covariances

```
procedure ConvergenceAnalysis
    Initialize hyperparameter $\alpha_0$
    for each desired number of samples do
        Sample vectors $\mathbf{z}_i$ from $N(\mathbf{0}, \boldsymbol{I})$
        Obtain initial samples $\boldsymbol{u}_{s_i}$ and posterior samples $\hat{\boldsymbol{u}}_{s_i}$
        Calculate posterior standard deviation $J_{\hat{\boldsymbol{\sigma}}}$
        while $J_{\hat{\boldsymbol{\sigma}}}$ has not converged do
            Update hyperparameter $\alpha_0$
            Obtain updated posterior $\hat{\boldsymbol{u}}_{s_i}$
            Calculate new $J_{\hat{\boldsymbol{\sigma}}}$
        end while
        Store final $J_{\hat{\boldsymbol{\sigma}}}$ and hyperparameter value
    end for
end procedure
```

To conduct this analysis, the procedure shown in Algorithm (1) was followed and the optimization algorithm used is L-BFGS-B. In the analysis, a sequence of integer numbers that exponentially increase from 2 to 10000 was generated. These values were used to determine the number of samples in each iteration. Fig. (31) illustrates the frequency at which the same analysis is repeated for a specific number of samples. For example, for 2 samples, the optimization was performed 20 times with 20 random initiations of the hyperparameters, while for 20 samples, the optimization was performed 8 times. This frequency progressively decreases as the sample count increases, as a higher number of samples leads to a smaller disparity compared to the analytically obtained values and the time of the optimization is higher because of the computational cost.

The same vector sampled from the Gaussian distribution of zero mean and unit deviation is used in each iteration of the optimization. It is only re-sampled at the initialization. This means that the posterior covariance matrices are optimized by changing the hyperparameters because the samples are the same.



(a)

(b)

(c)

Figure 30: Comparison between discretization error and posterior standard deviations of RVE structure. (a) Discretization error between a fine and coarse solution of RVE.(b) Posterior standard deviation with unit mass matrix prior. (c) Posterior standard deviation with non-stationary covariance function.

In this case, two distinct analyses were conducted. The first analysis utilized a prior covariance matrix with unit mass, while the second analysis incorporated non-stationarity using the coarse solution. The initial α_0 value for the case of the unit mass matrix is randomly initiated between 1 and 20.

In Fig. (32) the dots represent the value of $J_{\hat{\sigma}}$ minimized for that number of samples and a specific initialization. Then, the blue line represents the value obtained with an analytically assembled covariance matrix which is $J_{\hat{\sigma}} = 0.00428$, and finally the black lines are $10\%\pm$ with respect to this value. Similarly, Fig. (33) represents how the optimized hyperparameter α_0 varies for different numbers of samples, the blue line represents the value obtained from the analytical optimization, which is $\alpha_0 = 23.3673$, and the black lines are $10\%\pm$ of this value.

The first observation is that the sampled $J_{\hat{\sigma}}$ clearly converges to the analytically obtained value for the same problem. Similarly, in Fig. (33), we observe the convergence of the hyperparameter, and in both cases, there is a larger deviation from the analytical value with a low number of samples. Secondly, the randomly varied initial value of the hyperparameter ranged from 1 to 20. As seen in Fig. (33), there are instances where the approximated value surpasses the initial 20, indicating that even with a low number of samples, the hyperparameter does not remain stuck at

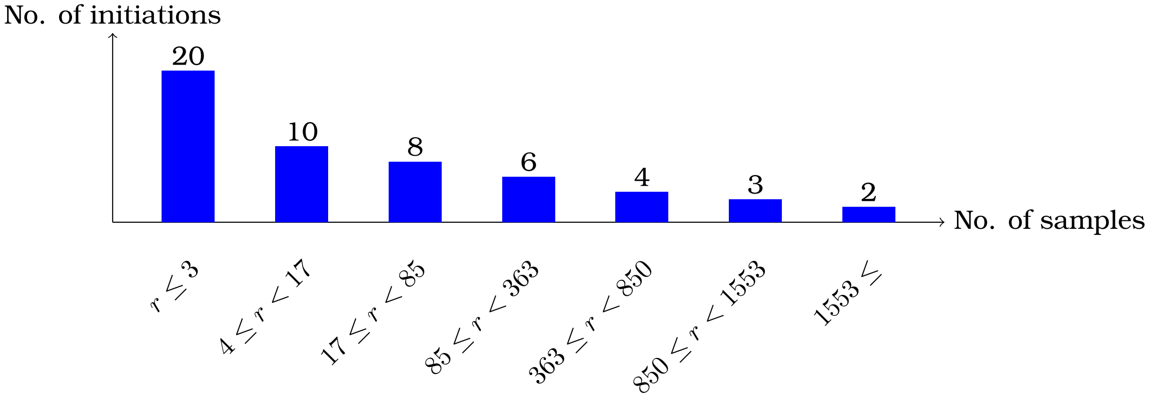


Figure 31: No. of initiations for each number of samples.

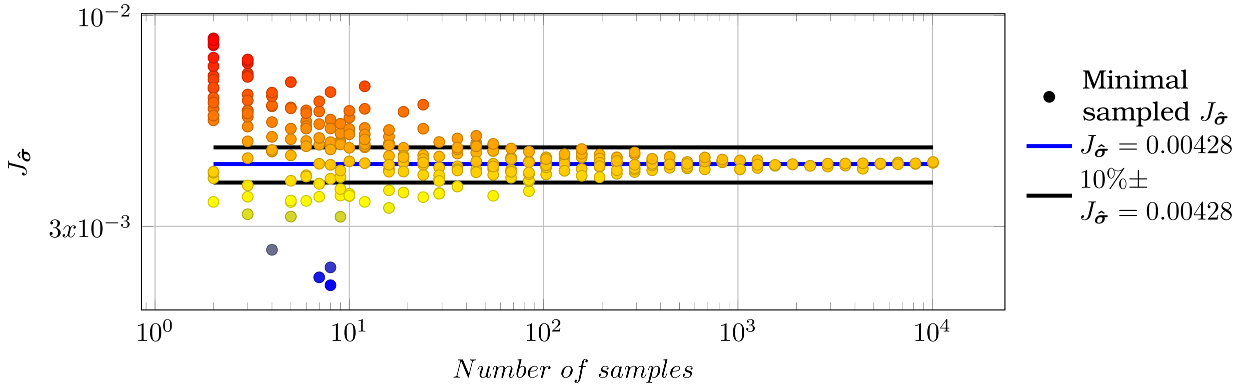


Figure 32: Plot comparing the optimized norms of posterior standard deviation for different numbers of samples. In this case, the prior is a unit mass matrix in the RHS.

the initial values.

Furthermore, the $J_{\hat{\sigma}}$ exhibits cases where optimized minimum values are lower than the analytical values. This phenomenon occurs primarily with a low number of samples, which is logical since with a smaller sample size, the randomness is higher.

In the case of the hyperparameter associated with the unit mass matrix, a similar behavior to $J_{\hat{\sigma}}$ is observed. The convergence is evident, and after approximately 100 samples, it becomes apparent that the value consistently deviates by around 10% from the analytical value, either above or below.

The second case involving the non-stationary matrix follows a similar procedure, but now there are two hyperparameters instead of just one. The blue lines represent the values obtained analytically, while the black lines represent a range of $\pm 10\%$ around the analytical values.

When examining the second case that incorporates non-stationarity, we can observe similar patterns to the previous case. Firstly, after optimizing the hyperparameters, the results converge towards the minimum of $J_{\hat{\sigma}}$, as shown in Fig. (34), and the hyperparameters displayed in Fig. (35) tend to align with those obtained using analytical covariances. Similarly to the case presented before, with approximately 100 samples, the solution tends to be around $\pm 10\%$ of the desired $J_{\hat{\sigma}}$. In other words, the achieved $J_{\hat{\sigma}}$ is a value lower due to the non-stationarity function, but it does not converge earlier because of it.

Fig. (35) illustrates the variation of the optimal α_0 and α_1 for different numbers of samples. It is important to mention that certain combinations of values have been excluded from the plots due to their significant influence on the scale, making most of the points difficult to differentiate. This occurred in a few combinations of hyperparameters for less than ten samples. In those cases

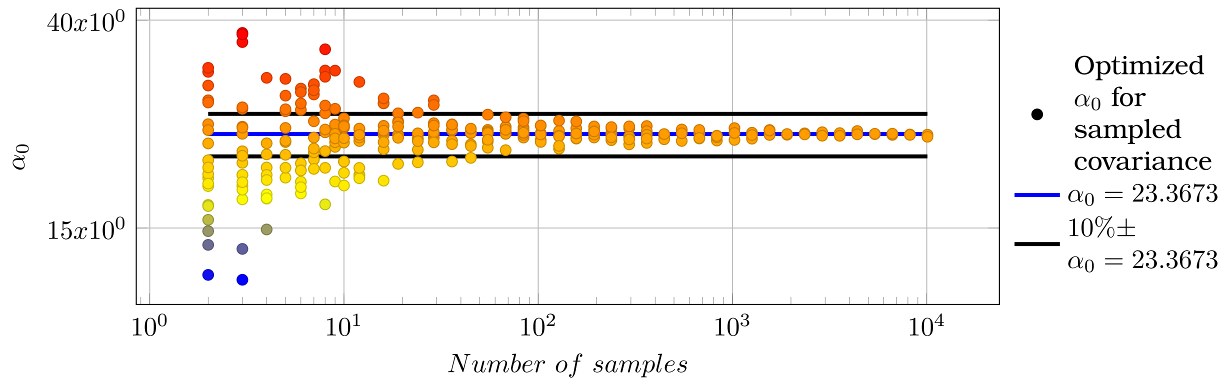


Figure 33: Plot comparing hyperparameters of the optimized norms of posterior standard deviation for different numbers of samples. In this case, the prior is a unit mass matrix in the RHS.

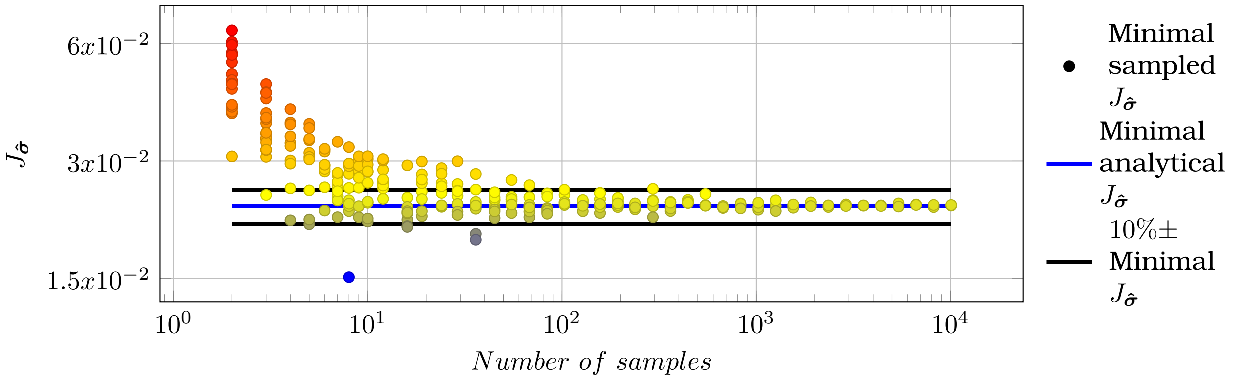


Figure 34: Plot comparing the optimized norms of posterior standard deviation for different numbers of samples. In this case, the prior is a non-stationary unit mass matrix in the RHS.

one of the hyperparameters approaches values close to zero, distorting the scale.

It can be seen that both hyperparameters also converge to the optimal combination of an analytically formed matrix and that after about 100 samples the values do not change so much.

Finally, upon comparing Fig. (32) and Fig. (34), it becomes apparent that the non-stationary case has less disparity. There are significantly more cases where the $J_{\hat{\sigma}}$ is less than -10% of the analytical value in the first case. In the second case, it can easily be counted the points that are below 10% less than the analytical minimum $J_{\hat{\sigma}}$.

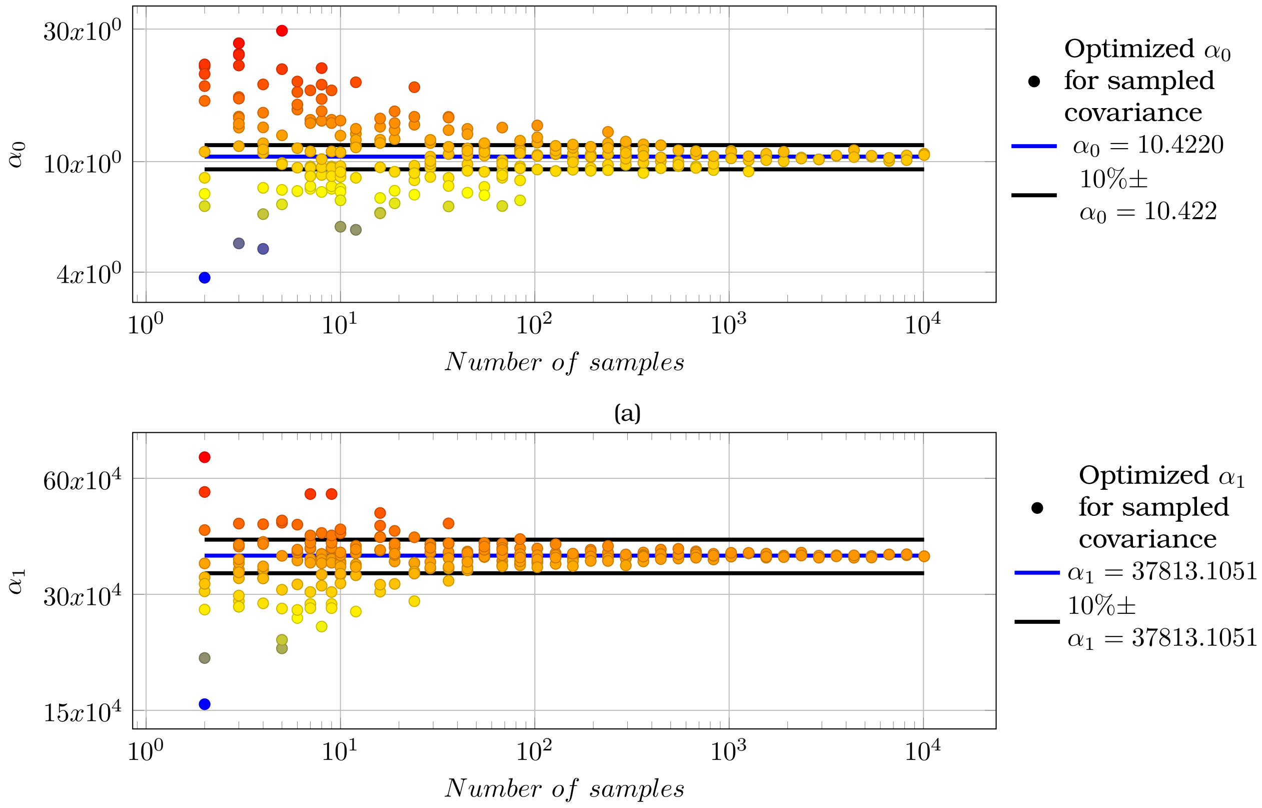
5.2.2 Solutions after optimizing the posterior mean

In this section the aim is to find the minimum values of the $J_{\hat{u}}$ for different numbers of samples, after optimizing the hyperparameters. The same procedure to the Algorithm (1) can be done but replacing the $J_{\hat{\sigma}}$ with the $J_{\hat{u}}$.

After performing this analysis the result was that in all sample cases the hyperparameters became close to zero and the optimization became unstable because the covariance matrix prior had very small values, the $J_{\hat{u}}$ tended to the analytical value when reducing the hyperparameters. This can be explained by rewritting the Eq. (4) applied to this case

$$\hat{\boldsymbol{u}}_s = \hat{\boldsymbol{u}} + \hat{\boldsymbol{L}}\boldsymbol{z} \tag{89}$$

where $\hat{\boldsymbol{u}}_s$ is the posterior sample vector, $\hat{\boldsymbol{u}}$ is the posterior mean, $\hat{\boldsymbol{L}}$ is the Cholensky decomposition of the posterior covariance, and $\boldsymbol{z}$ is a sample vector from a multivariate Gaussian distribution with zero mean and identity covariance matrix.



(a)

(b)

Figure 35: Plot comparing hyperparameters of the optimized norms of posterior standard deviation for different numbers of samples. In this case, the prior is a non stationary unit mass matrix in the RHS.

Although the procedure shown in section (2.6.4) is different, implicitly the same operation is performed, the difference is that the posterior covariance matrix is not computed analytically. Since the value $J_{\hat{\mathbf{u}}}$ measures the difference between the fine mesh solution and the posterior mean, the minimum obtained in this case would be for a value of the sampled posterior mean equal to the one obtained analytically.

Therefore, this operation becomes trivial, since the hyperparameters multiply the prior covariance, by simply replacing them with a zero value the complete matrix becomes a zero matrix and only the posterior mean in Eq. (89) remains.

6

Conclusions and Directions for Future Work

In the previous chapters, we analyzed model selection in BFEM, non-stationary priors, and the impact of meshing and sampling. In this chapter, we will recapitulate the questions presented in the first chapter and provide the corresponding answers. The chapter is divided into two parts: addressing the initial questions posed at the beginning of this thesis and discussing a few criticisms of BFEM and its potential as a basis for future work.

6.1 Answer to research questions

- Optimal hyperparameters and error representation:
  - *Is the maximum log-likelihood estimator a suitable way to obtain the optimal value of the hyperparameters to demonstrate the error as a source of uncertainty along the mesh?*

    It was shown that MLE does not generate proper results for estimating hyperparameters because it does not obtain minimum values of the difference between a posterior deviation and the discretization error between two meshes. This is because MLE adjusts the parameters of a distribution to fit the available data, and in this case, the available data is the forcing vector of a coarse mesh. However, the goal of BFEM is to quantify the discretization error with the standard deviation. Therefore, since the discretization error at each point of the coarse mesh is not used as data and is not part of the formulation that aims to quantify the discretization error through the posterior mean and deviation, the BFEM does not have to generate a good distribution.

  - *What is the best approach to compare different kernel or covariance matrix priors quantitatively and qualitatively, to represent the error along the mesh effectively?*

    In this thesis, two measures were used to facilitate the comparison of different priors to address the same problem. These measures were suitable for the objectives of the thesis, which simply measured which distribution better shows the discretization error or which posterior mean is closer to the fine mesh solution. The major drawback of these measures is that they are absolute, and the results obtained are not in percentages, so it is difficult to know how far a distribution is from resembling the discretization error. They are only useful to compare two priors applied to the same problem or to compare how the distribution changes for different mesh arrangements of the same number of

DOFs.

- Priors covariances:
  - *Is placing a prior on the right-hand side or forcing term considered better than one on the solution or left-hand side?*

    In this thesis, it was explained that incorporating a prior in the RHS is a way to consider the integro-differential relationships of the PDE. This is not achieved by directly placing a prior on the PDE solution itself. Additionally, it was clarified why the stiffness matrix is analogous to Green's function for solving PDEs. By utilizing sparse matrices on the right-hand side (RHS), we can achieve computationally more efficient covariance matrices compared to applying a kernel directly in the solution field. The inversion of the stiffness matrix is simpler due to its sparse structure, as opposed to a dense matrix. Additionally, obtaining samples in the forcing term and converting them into solution samples is easier when using a sparse covariance matrix. The samples are obtained through the decomposition of a sparse matrix like the mass matrix, followed by transforming the force samples into displacement samples. This transformation can be accomplished through the inversion of the sparse stiffness matrix or by utilizing approximate methods to solve the system of equations.

  - *Can the use of non-stationary covariance functions improve the distributions obtained by BFEM, particularly to show the discretization error?*

    It is necessary to mention that in this thesis the main focus was on obtaining appropriate distributions of discretization error with standard deviations and posterior means that show the fine mesh solution. However, the study did not aim to combine both aspects simultaneously; rather, they were investigated separately. Therefore, the use of stationary or non-stationary matrices varies between these two aspects.

    Stationary matrices have the disadvantage of not incorporating large variations in the posterior deviations after optimizing the hyperparameters. Consequently, they tend to generate a continuous distribution of error when deviating slightly from the observations used to generate the posterior. Therefore, they are incapable of generating suitable posterior standard deviations that accurately represent the error, unlike non-stationary matrices.

    Constructing non-stationary matrices is challenging since, a priori, the error is unknown. Therefore, finding a function to represent it and make the covariance non-stationary is challenging. However, using the coarse solution simplifies this process, obtaining not only qualitative representations but also, in simple cases such as the 2D beam example, suitable distributions of the error if the rights hyperparameters are chosen. In the case of the structure with voids, this was not the case as even with the incorporation of the coarse solution, the distribution did not improve significantly. This is due to the presence of numerous geometric singularities and boundary conditions in the problem.

    Concerning the posterior mean, it was demonstrated that it does not make sense, thus far, to use a prior matrix different from the unit mass matrix. The advantage of this approach is that there is no need to optimize or obtain the hyperparameter that amplifies the matrix since it does not change the solution. In all cases, with the use of this matrix, similar values of the posterior mean are observed compared to the fine mesh solution.

- Mesh impact:
  - *Do the optimal hyperparameters change between mesh refinements?*

- *Does something similar happen if the arrangement of the mesh elements is also modified?*

  It was demonstrated that refining the mesh leads to changes in the optimal values of the hyperparameters, as well as in a change of the mesh arrangement. In the case of using a sparse covariance matrix that depends on the shape functions of the mesh, better results can be obtained if the size of the elements is related to the error size in that area of the mesh.

  When utilizing a non-stationary covariance matrix, the hyperparameters exhibit relatively less variation with mesh refinements or between different arrangements of elements. However, these differences are still noticeable.

- Impact of the number of samples in ensemble covariance matrices:

  - *In the case of using sampling to assemble covariance matrices, what is the range of samples that is necessary to obtain distributions showing an error similar to the one obtained by the matrix without sampling?*

    It was proved that after obtaining approximately 100 samples, the hyperparameters of the prior can be optimized to obtain values that are similar to the posterior deviation of the analytically optimized distribution. Moreover, the distributions obtained with non-stationary priors do not converge faster but in general, have less disparity compared to those with stationary priors.

    Finally, in the case of optimizing the hyperparameters to obtain a posterior mean that is similar to the fine mesh solution, the optimization generates values of the hyperparameters that tend to be zero because implicitly the posterior mean is in the formula to obtain the posterior samples.

6.2 Directions for future work

One of the main drawbacks of BFEM is that it uses the stiffness matrix of the fine mesh to form the prior. This makes the method contradictory because if obtaining the most expensive part of the fine mesh is necessary, obtaining the discretization errors between the meshes could be directly solved. At the same time, the prior must take into account the boundary conditions, and in order to use Neumann conditions or non-homogeneous Dirichlet conditions, it is necessary to invert the fine mesh. It could be mentioned that these conditions could be studied probabilistically to avoid having to invert the matrix, but this is not intuitive, as the boundary conditions would become uncertain values when they are deterministic data.

Using sampling to assemble the covariance matrices has been proposed as a solution to reduce the computational cost of forming the matrices. However, a system of equations must still be solved or the stiffness matrix of the fine mesh must be inverted. Therefore, the cost of sampling remains high. In the future, a method such as the Langevin algorithm to generate posterior samples without the need to invert the stiffness matrix may be an option to reduce this cost.

Discretization errors have traditionally been studied to determine the extent to which the mesh size needs to be reduced to obtain consistent results of the solution, or, in the best-case scenario, optimize the mesh in areas of higher error and perform additional remeshing in those locations, which led to adaptivity techniques. These errors naturally arise from singularities where the slope of the primary field changes abruptly, such as point loads, sharp changes in the forcing load, or abrupt changes in geometry. Consequently, many error estimators have been formulated to identify changes in the derivatives of the primary field. However, in BFEM, the entire formulation is done in the primary field, meaning that singularities are not addressed at any point in the formulation. The proposed idea so far is to change or enrich the prior with noise in these degrees of freedom to account for these singularities. This introduces other additional challenges: how automatically can be determined where a singularity exists to avoid user intervention and how many DOFs have to be enriched with noise?

The result that the standard deviation can also represent the discretization error, as shown in this thesis, depends on many factors, including the chosen prior, the mesh itself, and the calibration of the hyperparameters of the prior. While it is possible to improve the prior by combining multiple stationary matrices and using non-stationary matrices to solve the problem, there is still the issue of obtaining the hyperparameters, and the more complex the prior becomes, the larger the amount of hyperparameters is likely to be, along with their estimation. Although a solution could involve training a neural network or some form of regression with multiple already optimized $J_{\hat{\sigma}}$, the number of inputs is enormous, and the fact that the optimal hyperparameters changes for each mesh represents an even greater challenge.

The cost of optimization in the sampling section was high due to the numerous optimizations performed for different sample sizes, and considering that the non-stationary prior case also requires additional computational time, it necessitated the use of a fine mesh that does not possess significantly more DOFs compared to the coarse mesh. Consequently, this raises another question: Can satisfactory results be achieved with only 100 samples if the DOFs of the fine mesh are several times greater than those of the coarse mesh?

Finally, if these questions are answered positively, it is likely that BFEM could be a useful method for inverse problems where an optimization algorithm is necessary to find the problem's parameters. In that case, BFEM could be employed in a fully probabilistic setting, where the mesh is coarsened to speed up each forward step in the optimization process, and where the error is propagated in the problem as another uncertainty.

A Gradients derivation

A.1 Gradients of $J_{\hat{\boldsymbol{\sigma}}}$ and $J_{\hat{\mathbf{u}}}$

The objective function to minimize the difference between the posterior standard deviation and the discretization error is given by:

$$J_{\hat{\boldsymbol{\sigma}}} = \frac{1}{2}\sum_{i=1}^{n}\left(\Delta_i - \sqrt{\hat{\Sigma}_{ii}(\boldsymbol{\xi})}\right)^2 \tag{90}$$

Δ_i represents the error between a coarse and a fine mesh in a DOF of the fine mesh, and $\hat{\Sigma}_{ii}$, is the variance in that degree of freedom. The objective is to minimize the difference between the error and the standard deviation by optimizing the hyperparameters that are represented by the vector $\boldsymbol{\xi}$.

To update the hyperparameters using the BFGS or L-BFGS algorithm, the gradients are calculated. Taking the partial derivative of the objective function with respect to the hyperparameters yields:

$$\frac{\partial J_{\hat{\boldsymbol{\sigma}}}}{\partial \xi_j} = \sum_{i=1}^{n}\left(\Delta_i - \sqrt{\hat{\Sigma}_{ii}(\boldsymbol{\xi})}\right)\left(-\frac{\partial\sqrt{\hat{\Sigma}_{ii}(\boldsymbol{\xi})}}{\partial \xi_j}\right) \tag{91}$$

This equation expresses the gradient of the objective function J_{STD} with respect to the hyperparameters ξ_j. It involves the difference between Δ_i and the square root of the posterior variance $\hat{\Sigma}ii$ evaluated at the hyperparameters $\boldsymbol{\xi}$. The partial derivative of $\sqrt{\hat{\Sigma}ii(\boldsymbol{\xi})}$ with respect to ξ_j is then multiplied by this difference.

By computing these gradients, is possible to update the hyperparameters iteratively in each step of the BFGS optimization algorithm. To proceed, the derivative of the posterior covariance has to be obtained. The posterior covariance is defined as:

$$\hat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma} - \boldsymbol{\Sigma}\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{H}^T)^{-1}\boldsymbol{H}\boldsymbol{\Sigma} \tag{92}$$

In order to obtain this derivative the inverse of a matrix is obtained:

$$\underbrace{\boldsymbol{I}'}_{=0} = \left(\boldsymbol{A}\boldsymbol{A}^{-1}\right)' = \boldsymbol{A}'\boldsymbol{A}^{-1} + \boldsymbol{A}\left(\boldsymbol{A}^{-1}\right)' \;\; \boldsymbol{A}\left(\boldsymbol{A}^{-1}\right)' = -\boldsymbol{A}'\boldsymbol{A}^{-1} \Rightarrow \left(\boldsymbol{A}^{-1}\right)' = -\boldsymbol{A}^{-1}\boldsymbol{A}'\boldsymbol{A}^{-1} \tag{93}$$

Now, applying this to our case, we obtain the derivative of the term $\left((\boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{H}^T)^{-1}\right)$:

$$\left((\boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{H}^T)^{-1}\right)' = -(\boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{H}^T)^{-1}\left(\boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{H}^T\right)'(\boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{H}^T)^{-1} \tag{94}$$

Next, the derivative of the term $\left(\sqrt{\hat{\Sigma}_{ii}}\right)$ is:

$$\begin{aligned}\frac{\partial\sqrt{\hat{\Sigma}(\boldsymbol{\xi})}_{ii}}{\partial\xi_j} = \frac{1}{2\sqrt{\hat{\Sigma}(\boldsymbol{\xi})}_{ii}}\Bigg\{\left(\frac{\partial\boldsymbol{\Sigma}(\boldsymbol{\xi})}{\partial\xi_j}\right) + \frac{\partial\boldsymbol{\Sigma}(\boldsymbol{\xi})}{\partial\xi_j}\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{\Sigma}(\boldsymbol{\xi})\boldsymbol{H}^T)^{-1}\boldsymbol{H}\boldsymbol{\Sigma}(\boldsymbol{\xi}) \\ -\boldsymbol{\Sigma}(\boldsymbol{\xi})\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{\Sigma}(\boldsymbol{\xi})\boldsymbol{H}^T)^{-1}\boldsymbol{H}\frac{\partial\boldsymbol{\Sigma}(\boldsymbol{\xi})}{\partial\xi_j}\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{\Sigma}(\boldsymbol{\xi})\boldsymbol{H}^T)^{-1}\boldsymbol{H}\boldsymbol{\Sigma}(\boldsymbol{\xi}) - \boldsymbol{\Sigma}\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{\Sigma}(\boldsymbol{\xi})\boldsymbol{H}^T)^{-1}\boldsymbol{H}\frac{\partial\boldsymbol{\Sigma}(\boldsymbol{\xi})}{\partial\xi_j}\Bigg\}_{ii}\end{aligned} \tag{95}$$

The second function to optimize is the difference between the posterior mean and the fine mesh solution.

$$J_{\hat{\boldsymbol{u}}} = \frac{1}{2}\sum_{i=1}^{n}(u_{f_i} - \hat{u}_i(\boldsymbol{\xi}))^2 \tag{96}$$

Again, the derivative of the function is:

$$\frac{\partial J_{\hat{\boldsymbol{u}}}}{\partial\xi_j} = \sum_{i=1}^{n}(u_{f_i} - \hat{u}_i(\boldsymbol{\xi}))\left(-\frac{\partial\hat{u}_i(\boldsymbol{\xi})}{\partial\xi_j}\right) \tag{97}$$

The posterior mean in the displacement field is:

$$\hat{\boldsymbol{u}} = \boldsymbol{\Sigma}\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{H}^T)^{-1}\boldsymbol{g} \tag{98}$$

Lastly, the gradient is

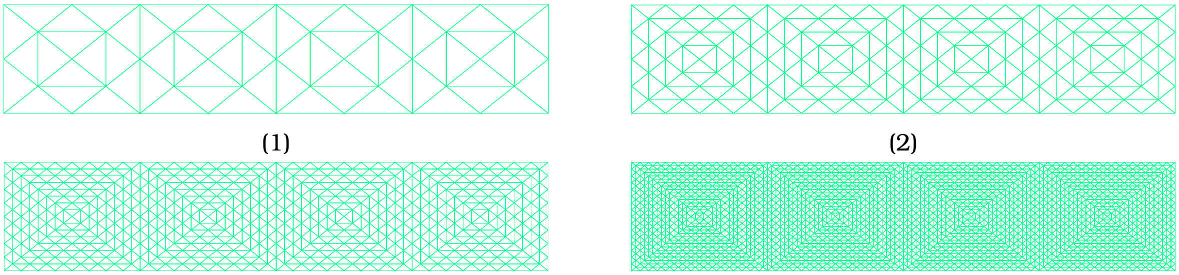
$$\frac{\partial\hat{u}_i(\boldsymbol{\xi})}{\partial\xi_j} = \frac{\partial\boldsymbol{\Sigma}(\boldsymbol{\xi})}{\partial\xi_j}\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{\Sigma}(\boldsymbol{\xi})\boldsymbol{H}^T)^{-1}\boldsymbol{g} + \boldsymbol{\Sigma}(\boldsymbol{\xi})\boldsymbol{H}^T\left(-(\boldsymbol{H}\boldsymbol{\Sigma}(\boldsymbol{\xi})\boldsymbol{H}^T)^{-1}\boldsymbol{H}\frac{\partial\boldsymbol{\Sigma}(\boldsymbol{\xi})}{\partial\xi_j}\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{\Sigma}(\boldsymbol{\xi})\boldsymbol{H}^T)^{-1}\right)\boldsymbol{g} \tag{99}$$

B

Meshes used

B.1 FEM Meshes

This appendix presents the coarse meshes utilized in section (5.1) to investigate the mesh dependence of the optimized hyperparameters. The corner nodes were preserved to maintain consistent dimensions across all meshes. However, the remaining nodes on the perimeter underwent modifications in nearly all instances, while the internal nodes were also adjusted to create scenarios where the finest elements are positioned near the supports or the center as the mesh is refined. A total of 12 cases were generated using the gmsh program, and the mesh refinement was achieved by utilizing the "refine by splitting" option provided by the program.



(1)

(2)

(3)

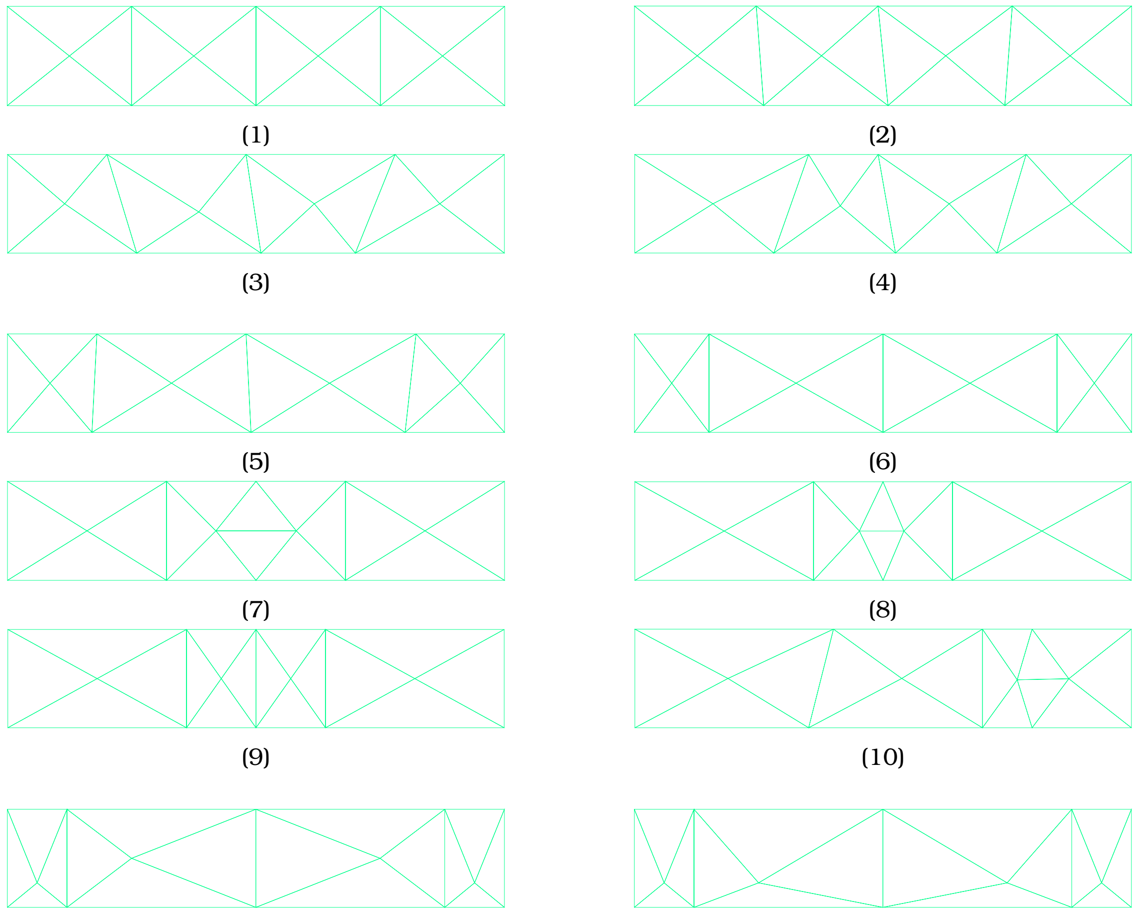
(4)

Figure B1: Four refinements of mesh No. 1

B.2 Meshes used for the RVE structure

In the images provided, you can observe the coarse and fine meshes used in the section (5.2). It is important to note that the Representative Volume Element (RVE) depicted in the coarse mesh contains two holes. However, it is not possible to refine these holes by adding more nodes along the circumference. The reason behind this limitation is that refining the holes in the coarse mesh would result in the loss of the projected fine DOFs onto the coarse structure.

To overcome this issue, an alternative approach is employed. Instead of refining the holes directly, additional nodes are introduced along the lines used to mesh the coarse mesh holes. This strategy allows for preserving the projected fine DOFs on the coarse structure while still introducing local refinements. By adding nodes on these lines, the desired mesh refinement can be achieved without compromising the accuracy of the projection process.



(1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12)

Figure B2: Meshes used to study the dependency of the optimized hyperparameters.

By adopting this method, it becomes possible to maintain the integrity of the fine-scale information while incorporating localized refinements in the mesh.

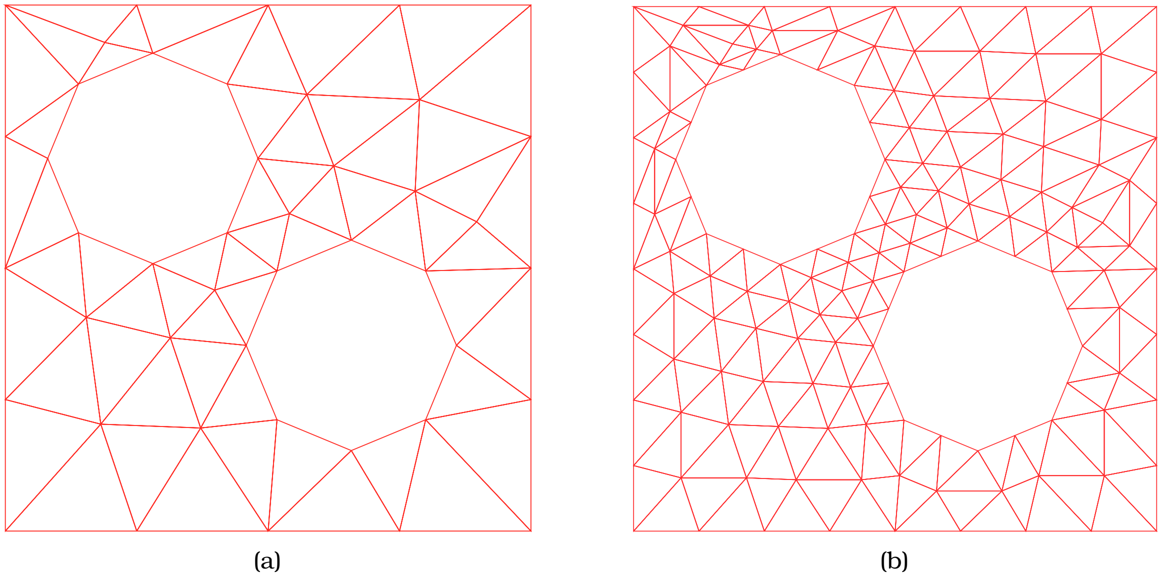


Figure B3: Meshes used in the RVE. (B3a) Coarse mesh of 44 nodes. (B3b) Fine mesh of 147 nodes.

References

[1] Y. D. Lugovtsova and A. I. Soldatov, “Comparison of Three Different Methods for Pile Integrity Testing on a Cylindrical Homogeneous Polyamide Specimen,” *Journal of Physics: Conference Series*, vol. 671, p. 012055, Jan. 2016. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/671/1/012055

[2] B. A. Szabo and I. Babuška, *Finite Element Analysis: Method, Verification and Validation*, second edition ed., ser. Wiley Series in Computational Mechanics. Wiley, 2021.

[3] I. Babuška and W. C. Rheinboldt, “A-posteriori error estimates for the finite element method,” vol. 12, no. 10, pp. 1597–1615. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/nme.1620121010

[4] ——, “Error estimates for adaptive finite element computations,” *SIAM Journal on Numerical Analysis*, vol. 15, no. 4, pp. 736–754, 1978. [Online]. Available: http://epubs.siam.org/doi/10.1137/0715049

[5] O. C. Zienkiewicz and J. Z. Zhu, “The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique,” *International Journal for Numerical Methods in Engineering*, vol. 33, no. 7, pp. 1331–1364, May 1992. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/nme.1620330702

[6] ——, “The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity,” *International Journal for Numerical Methods in Engineering*, vol. 33, no. 7, pp. 1365–1382, May 1992. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/nme.1620330703

[7] P. Hennig, M. A. Osborne, and M. Girolami, “Probabilistic numerics and uncertainty in computations,” vol. 471, no. 2179. [Online]. Available: https://royalsocietypublishing.org/doi/10.1098/rspa.2015.0142

[8] A. O’Hagan, “Bayes-hermite quadrature,” *Journal of Statistical Planning and Inference*, vol. 29, no. 3, pp. 245–260, 1991. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/037837589190002V

[9] J. Cockayne, C. J. Oates, I. C. Ipsen, and M. Girolami, “A bayesian conjugate gradient method (with discussion),” 2019.

[10] J. Cockayne, C. J. Oates, T. J. Sullivan, and M. Girolami, “Bayesian Probabilistic Numerical Methods,” *SIAM Review*, vol. 61, no. 3, pp. 756–789, Jan. 2019. [Online]. Available: https://epubs.siam.org/doi/10.1137/17M1139357

[11] P. Hennig, M. A. Osborne, and H. P. Kersting, *Probabilistic Numerics: Computation as Machine Learning*, 1st ed. Cambridge University Press. [Online]. Available: https://www.cambridge.org/core/product/identifier/9781316681411/type/book

[12] O. A. Chkrebtii, D. A. Campbell, B. Calderhead, and M. A. Girolami, “Bayesian solution uncertainty quantification for differential equations,” June 2013. [Online]. Available: https://arxiv.org/abs/1306.2365

[13] J. Wang, J. Cockayne, O. Chkrebtii, T. J. Sullivan, and C. J. Oates, “Bayesian numerical methods for nonlinear partial differential equations,” *Statistics and Computing*, vol. 31, no. 5, p. 55, 2021. [Online]. Available: https://link.springer.com/10.1007/s11222-021-10030-w

[14] J. Cockayne, C. Oates, T. Sullivan, and M. Girolami, “Probabilistic numerical methods for partial differential equations and bayesian inverse problems,” *arXiv*, 2016. [Online]. Available: https://arxiv.org/abs/1605.07811

[15] P. R. Conrad, M. Girolami, S. Särkkä, A. Stuart, and K. Zygalakis, "Statistical analysis of differential equations: introducing probability measures on numerical solutions," vol. 27, no. 4. [Online]. Available: https://link.springer.com/article/10.1007/s11222-017-9737-4

[16] M. Girolami, E. Febrianto, G. Yin, and F. Cirak, "The statistical finite element method (statFEM) for coherent synthesis of observation data and model predictions," *Computer Methods in Applied Mechanics and Engineering*, vol. 375, p. 113533, 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0045782520307180

[17] D. Akyildiz, C. Duffin, S. Sabanis, and M. Girolami, "Statistical finite elements via langevin dynamics," *SIAM/ASA Journal on Uncertainty Quantification*, vol. 10, no. 4, pp. 1560–1585, 2022. [Online]. Available: https://epubs.siam.org/doi/10.1137/21M1463094

[18] A. Poot, P. Kerfriden, I. Rocha, and F. van der Meer, "A bayesian approach to modeling finite element discretization error." [Online]. Available: http://arxiv.org/abs/2306.05993

[19] J. N. Reddy, *Introduction to the finite element method*, fourth edition ed., ser. Mechanical engineering. McGraw Hill Education, 2019, oCLC: on1065524414.

[20] H. Owhadi, "Bayesian numerical homogenization," *Multiscale Modeling & Simulation*, vol. 13, no. 3, pp. 812–828, 2015. [Online]. Available: https://doi.org/10.1137/140974596

[21] L. Roininen, J. M. Huttunen, and S. Lasanen, "Whittle-matérn priors for Bayesian statistical inversion with applications in electrical impedance tomography." *Inverse Problems & Imaging*, vol. 8, no. 2, 2014.

[22] N. C. Nguyen and J. Peraire, "Gaussian functional regression for linear partial differential equations," *Computer Methods in Applied Mechanics and Engineering*, vol. 287, pp. 69–89, 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0045782515000092

[23] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960. [Online]. Available: https://asmedigitalcollection.asme.org/fluidsengineering/article/82/1/35/397706/A-New-Approach-to-Linear-Filtering-and-Prediction

[24] M. Katzfuss, J. R. Stroud, and C. K. Wikle, "Understanding the Ensemble Kalman Filter," *The American Statistician*, vol. 70, no. 4, pp. 350–357, Oct. 2016. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/00031305.2016.1141709

[25] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2005. [Online]. Available: https://direct.mit.edu/books/book/2320/gaussian-processes-for-machine-learning

[26] M. M. Noack and J. A. Sethian, "Advanced Stationary and Non-Stationary Kernel Designs for Domain-Aware Gaussian Processes," *arXiv*, 2021. [Online]. Available: https://arxiv.org/abs/2102.03432

[27] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information science and statistics. New York: Springer, 2006.

[28] P. Y. Papalambros and D. J. Wilde, *Principles of Optimal Design Modeling and Computation*, 3rd ed. Cambridge University Press, 2017.

[29] C. G. Broyden, "The convergence of a class of double-rank minimization algorithms 1. general considerations," *IMA Journal of Applied Mathematics*, vol. 6, no. 1, pp. 76–90, 1970. [Online]. Available: https://academic.oup.com/imamat/article-lookup/doi/10.1093/imamat/6.1.76

[30] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on Scientific Computing*, vol. 16, no. 5, pp. 1190–1208, 1995. [Online]. Available: http://epubs.siam.org/doi/10.1137/0916069

[31] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965. [Online]. Available: https://academic.oup.com/comjnl/article-lookup/doi/10.1093/comjnl/7.4.308