

Efficient estimation for a smoothing thin plate spline in a two-dimensional space

Joaquin Cavieres¹ and Michael Karkulik²

¹Georg-August-Universität Göttingen, Germany

²Universidad Técnica Federico Santa María, Chile

April 3, 2024

Abstract

Using a deterministic framework allows us to estimate a function with the purpose of interpolating data in spatial statistics. Radial basis functions are commonly used for scattered data interpolation in a d -dimensional space, however, interpolation problems have to deal with dense matrices. For the case of smoothing thin plate splines, we propose an efficient way to address this problem by compressing the dense matrix by an hierarchical matrix (\mathcal{H} -matrix) and using the conjugate gradient method to solve the linear system of equations. A simulation study was conducted to assess the effectiveness of the spatial interpolation method. The results indicated that employing an \mathcal{H} -matrix along with the conjugate gradient method allows for efficient computations while maintaining a minimal error. We also provide a sensitivity analysis that covers a range of smoothing and compression parameter values, along with a Monte Carlo simulation aimed at quantifying uncertainty in the approximated function. Lastly, we present a comparative study between the proposed approach and thin plate regression using the “mgcv” package of the statistical software R. The comparison results demonstrate similar interpolation performance between the two methods.

Keywords: Smoothing thin plate spline, \mathcal{H} -matrix, spatial statistics, adaptive cross approximation.

1 Introduction

In the last years, the increased availability of data from sensors, satellites, or monitoring stations sparked special interest in scattered data interpolation methods (Xu et al. (2011)). In this context, the main objective is to approximate a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, given a set of sampled values $f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)$ at a set of spatial sites $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathbb{R}^d$.

A commonly employed category of interpolants to approximate f are radial basis functions (RBFs, Scheuerer et al. (2013)). This technique was initially applied in the 1980s to problems in machine learning, and has since been used in different contexts such as partial differential equations, computer graphics, learning theory, medical imaging, environmental variables, cartography, and surface reconstruction, among others (Wendland 2004; Chiles and Delfiner 2009). RBFs are a powerful tool for problems of multivariate data interpolation in arbitrary dimensions, such as required in spatial statistics (Wahba 1990; Lin and Chen 2004).

From a statistical perspective, several authors have introduced distinct methodologies involving RBFs. These encompass approaches like thin plate splines (TPS) and smoothing thin plate splines (STPS), also used for spatial interpolation. Among these authors, Wahba (1990), Green and Silverman (1993) and Wang (2011) offer an extensive read into smoothing splines. For example, Wahba (1975) shows how to choose the smoothing parameter for a periodic spline, Hutchinson and de Hoog (1985) presents a procedure to calculate the trace of the influence matrix for a polynomial smoothing spline of degree $2m-1$, and Craven and Wahba (1978) estimated the correct degree of smoothing using the Generalized Cross-Validation (GCV) method. An extension to the Bayesian context can be found in Wahba (1983) and Nychka (1988), who proposed an estimation of confident intervals for smoothing splines. Specifically, from the spatial interpolations applications, Wahba et al. (1995) proposed an ANOVA analysis for an epidemiological study, Trossman et al. (2011) applied a TPS for comparative purposes in an oceanographic study and Early and Sykulski (2020) interpolated noisy GPS data using smoothing splines. Another important contribution was made by Wood (2003), wherein the author introduced an approximation of the splines matrix through truncated spectral decomposition. This allows us to get an efficient

computation even when dealing with a large number of spatial observations (Wood (2017)). On the other hand, the Kriging method (Matheron (1963)), is mostly used for statistical interpolation in dimension $d = 2$. Both approximation methods, RBFs and Kriging, are similar and various authors have studied their connections (Scheuerer et al. 2013). The computational cost for both methods is cubic in the number of spatial observations, which is prohibitively high. To deal with this problem different authors have proposed alternatives (Lindgren et al. (2011); Bevilacqua and Morales-Onate (2018); Litvinenko et al. (2019), and Hristopulos et al. (2021)).

From a deterministic perspective, the theory of *Reproducing Kernel Hilbert Space* (RKHS) to interpolate spatial data (Wahba (1990)) is equivalent to the stochastic process assumed as random structure in the space (Berlinet and Thomas-Agnan 2011). Here, we consider the approximation of f by a function $g : \mathbb{R}^d \rightarrow \mathbb{R}$ of the form

$$g(\mathbf{x}) = \sum_{i=1}^n c_i \Phi(\mathbf{x}_i, \mathbf{x}) + \sum_{\ell=1}^q d_\ell p_\ell(\mathbf{x}), \quad (1)$$

where $\Phi(\mathbf{x}, \mathbf{y}) = \phi(\|\mathbf{x} - \mathbf{y}\|_2)$ with $\phi : [0, \infty) \rightarrow \mathbb{R}$ radially symmetric and $\|\cdot\|_2$ the Euclidean norm in \mathbb{R}^d . Common choices of ϕ include conditionally semidefinite functions (Wendland 2004, Chapter 8), i.e., functions generating a definite interpolation problem on the linear subspace given by $\sum_{i=1}^n c_i p(\mathbf{x}_i) = 0$ for all polynomials $p \in \mathcal{P}_m^d$ of degree less or equal to a certain number m (depending on ϕ) in dimension d . Hence, the functions $\{p_1, \dots, p_q\}$ in Equation (1) are chosen to form a basis of \mathcal{P}_m^d . In other words, the interpolation problem

$$\begin{aligned} \sum_{i=1}^n c_i \Phi(\mathbf{x}_i, \mathbf{x}_j) + \sum_{\ell=1}^q d_\ell p_\ell(\mathbf{x}_j) &= f(\mathbf{x}_j), \quad j = 1, \dots, n \\ \sum_{i=1}^n c_i p_k(\mathbf{x}_i) &= 0, \quad k = 1, \dots, q \end{aligned} \quad (2)$$

has a unique solution $\mathbf{c} = (c_1, \dots, c_n)^T \in \mathbb{R}^n$ and $\mathbf{d} = (d_1, \dots, d_q)^T \in \mathbb{R}^q$. Equations (2) can

be written as linear system

$$\begin{pmatrix} \mathbf{E} & \mathbf{P} \\ \mathbf{P}^\top & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{c} \\ \mathbf{d} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix}, \quad \text{where } \mathbf{E}_{j,i} = \Phi(\mathbf{x}_i, \mathbf{x}_j) \quad \text{and} \quad \mathbf{P}_{j,i} = p_i(\mathbf{x}_j). \quad (3)$$

If Φ has local support, then the matrix \mathbf{E} is sparsely populated and system (3) can be solved in linear complexity (Wendland 2006). Hence, there is no need for matrix compression techniques. If Φ has non-local support then the matrix \mathbf{E} is fully populated, and this entails a high computational cost $\mathcal{O}(n^3)$ for the solution of system (3). As the rise of new technologies enables us to gather vast amounts of data, we need to address this high cost to make computations viable (Iske et al. 2017). Compression of densely populated matrices by certain data-sparse formats is a necessity for computations in various other areas, for example for the numerical solution of integral equations or, more recently, fractional partial differential equations. In this context, different methods such as multipole expansions, panel clustering, wavelet compression, mosaic-skeleton methods, adaptive cross approximation, and hybrid cross-approximation have been presented (Lyche (2020), Bini et al. (2019)). In particular, hierarchical matrices (\mathcal{H} -matrices) are a convenient and general algebraic format for data-sparse compression, and in a nutshell they may be summarized as *blockwise low-rank matrices* (Hackbusch 2015; Bebendorf 2008). Although initially developed for application to integral equations, an \mathcal{H} -matrix can be used in a much wider context as long as the function generating the entries of the dense matrix (Φ in our case) is *asymptotically smooth* (Iske et al. (2017)). Hence, matrices in RBF interpolation and related problems can be approximated well by an \mathcal{H} -matrix, allowing for a matrix-vector multiplication of complexity $\mathcal{O}(n \log(n))$ and efficient iterative solution techniques, as witnessed in Iske et al. (2017) and Löhndorf and Melenk (2017).

In this work, we consider the special case of smoothing thin plate spline (STPS) interpolation, including a regularization term measuring the smoothness of the fitted curve. We can rewrite this problem in terms of a simple linear system of equations, where the dense splines matrix is compressed by an \mathcal{H} -matrix. To solve the linear system, we show how to rewrite it as a positive definite and symmetric problem, and present numerical experiments

using the Conjugate Gradient method.

The paper is structured as follows: In Section 2, we define the a thin plate spline (TPS) and a STPS. This section also describe general concepts related to the \mathcal{H} -matrix. In Section 3, we present the results of a simulated study focuses on solving a linear equation system using three different methods: a direct solver, the Conjugate Gradient method, and, lastly, the Conjugate Gradient method combined with the \mathcal{H} -matrix. In addition, we assessed the quality of the approximated function, computed the estimation time, conducted sensitivity analyses for the penalty parameter and the parameters to build the \mathcal{H} -matrix, performed Monte Carlo simulations, and compared our approach with the thin plate spline regression (TPSR) proposed by Wood (2003), also considering the associated uncertainty. In Section 4, we conclude by discussing the results and the potential incorporation of hierarchical matrices into the R software.

2 Methods

2.1 Thin plate spline

A thin plate spline (TPS) is defined in terms of a conditionally semidefinite RBF (Wendland (2004)) given by:

$$\Phi(\mathbf{x}) = \|\mathbf{x}\|^2 \log \|\mathbf{x}\|, \quad \mathbf{x} \in \mathbb{R}^d \quad (4)$$

where $\|\cdot\|$ is the Euclidean distance and the basis $\{p_1, p_2, p_3\}$ of

$$\mathcal{P}_1^2 = \{p : \mathbb{R}^2 \rightarrow \mathbb{R} \mid \exists a, b, c \in \mathbb{R} : p(\mathbf{x}) = a\mathbf{x}_1 + b\mathbf{x}_2 + c\}.$$

A function g is called a TPS on a set $\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathbb{R}^2$ if it has the form

$$g(\mathbf{x}) = \sum_{i=1}^n c_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) + \sum_{\ell=1}^3 d_\ell p_\ell(\mathbf{x}), \quad (5)$$

with coefficients $c_1, \dots, c_n, d_1, d_2, d_3$ (Green and Silverman 1993; Cavieres et al. 2022).

If the vector of coefficients $\mathbf{c} = (c_1, \dots, c_n)^\top$ satisfies $\sum_{i=1}^n c_i p(\mathbf{x}_i) = 0$ for all $p \in \mathcal{P}_1^2$, then g is called a natural TPS. Clearly, it is sufficient to impose the last identity only on a basis

of \mathcal{P}_1^2 , which leads to $\mathbf{P}^\top \mathbf{c} = \mathbf{0}$ for the matrix $\mathbf{P} \in \mathbb{R}^{n \times 3}$ given by $\mathbf{P}_{i,j} = p_j(\mathbf{x}_i)$. While we stick to the case $d = 2$ in order to simplify presentation and emphasize applications to spatial statistics, we stress that TPS and the corresponding compression by hierarchical matrices are not restricted to this case.

2.2 Smoothing thin plate spline (STPS)

A TPS provides an alternative option for scattered data approximation in the spatial domain, which does not consider a covariance function associated with spatial observations. In statistical modeling, considering the presence of inherent measurement uncertainty or “noise” in collected data, we choose to approximate the function using a smoothing thin plate spline (STPS) (Wahba 1990).

Let g be an approximating function for n observations y_i at sites \mathbf{x}_i . We can propose the following statistical model

$$y_i = g(\mathbf{x}_i) + \epsilon_i, \quad i = 1, \dots, n, \quad (6)$$

where ϵ_i is a random error assumed as $\mathcal{N}(0, 1)$. We can define the residual sum of penalized least squares terms by

$$S(g) = \sum_{i=1}^n |y_i - g(\mathbf{x}_i)|^2 + \lambda J(g). \quad (7)$$

The component $J(g)$ is a penalty function that measures the smoothness of g , and λ is a penalty parameter that controls the balance between the data fit and smoothness. For a TPS in \mathbb{R}^2 , the penalty function has the form

$$J(g) = \int_{\mathbb{R}^2} \left(\frac{\partial^2 g}{\partial \mathbf{x}_1^2} \right)^2(\mathbf{x}) + 2 \left(\frac{\partial^2 g}{\partial \mathbf{x}_1 \partial \mathbf{x}_2} \right)^2(\mathbf{x}) + \left(\frac{\partial^2 g}{\partial \mathbf{x}_2^2} \right)^2(\mathbf{x}) d\mathbf{x} \quad (8)$$

We aim to minimize $S(g)$ over the set of natural TPS, and according to (Green and Silverman 1993, Thm. 7.1), the analytical solution of Equation (8) is

$$J(g) = \mathbf{c}^\top \mathbf{E} \mathbf{c},$$

with \mathbf{E} from Equation (3). Hence, the minimization of $S(g)$ over all natural TPS can be stated as

$$\min_{\substack{\mathbf{c} \in \mathbb{R}^n, \mathbf{d} \in \mathbb{R}^3 \\ \mathbf{P}^\top \mathbf{c} = \mathbf{0}}} \|\mathbf{y} - \mathbf{E}\mathbf{c} - \mathbf{P}\mathbf{d}\|_2^2 + \lambda \mathbf{c}^\top \mathbf{E}\mathbf{c}, \quad (9)$$

It can be shown that if the sites $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ are not collinear, then Equation (9) has a unique solution given by the linear system

$$\begin{pmatrix} \mathbf{E} + \lambda \mathbf{I} & \mathbf{P} \\ \mathbf{P}^\top & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{c} \\ \mathbf{d} \end{pmatrix} = \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix}, \quad (10)$$

where $\mathbf{I} \in \mathbb{R}^{n \times n}$ is the identity matrix (Green and Silverman 1993, Sec. 7.1).

2.3 Hierarchical matrices (\mathcal{H} -matrix)

For a set of sites $\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathbb{R}^d$ we denote by $I = \{1, \dots, n\}$ the index set. A subset $\tau \subset I$ of indices is called a *cluster*. For the above we can define the *diameter* of a cluster τ as

$$\text{diam}(\tau) = \max_{j, i \in \tau} \|\mathbf{x}_j - \mathbf{x}_i\|_2,$$

and the distance of two clusters τ, σ as

$$\text{dist}(\tau, \sigma) = \min_{j \in \tau, i \in \sigma} \|\mathbf{x}_j - \mathbf{x}_i\|_2.$$

The quadratic cost in the number of indices of calculating diameter and distance can be avoided, e.g., by modified definitions of diameter and distance (Rjasanow and Steinbach 2007), or the use of so-called *bounding boxes* (Hackbusch 2015). For a parameter $\eta > 0$ we introduce a so-called admissibility condition

$$\text{adm} : \begin{cases} 2^I \times 2^I & \rightarrow \{\text{true}, \text{false}\} \\ (\tau, \sigma) & \mapsto \min(\text{diam}(\tau), \text{diam}(\sigma)) < \eta \text{dist}(\tau, \sigma). \end{cases}$$

A partition \mathcal{P} of $I \times I$ so that all elements $p \in \mathcal{P}$ are of the form $p = \tau \times \sigma$ with clusters $\tau, \sigma \subset I$ is called *block partition* and a p of the above form is called *block cluster*. For a constant $\sigma_{\text{small}} > 0$, a block partition is called σ_{small} -*admissible*, if it can be split into two parts $\mathcal{P} = \mathcal{P}_{\text{near}} \cup \mathcal{P}_{\text{far}}$ with $\text{adm}(\tau, \sigma) = \text{true}$ for all $\tau \times \sigma \in \mathcal{P}_{\text{far}}$ (the *far field*) and either $|\tau| \leq \sigma_{\text{small}}$ or $|\sigma| \leq \sigma_{\text{small}}$ for $\tau \times \sigma \in \mathcal{P}_{\text{near}}$ (the *near field*).

Definition 1 Let \mathcal{P} be an admissible block partition of $I \times I$. We define the space of \mathcal{H} -matrices of rank $r \in \mathbb{N}$ as

$$\mathcal{H}(\mathcal{P}, r) := \{B \in \mathbb{R}^{n \times n} \mid \forall \tau \times \sigma \in \mathcal{P}_{\text{far}} \exists X \in \mathbb{R}^{|\tau| \times r}, Y \in \mathbb{R}^{|\sigma| \times r} \text{ so that } B|_{\tau \times \sigma} = XY^\top\}$$

For uniformly distributed sites $\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathbb{R}^d$, σ_{small} -admissible block partitions can be created in complexity $\mathcal{O}(n \log(n))$, for example by geometry-based construction using hierarchical partitions of index set I , so-called *cluster trees* (Hackbusch 2015). Far-field blocks are admissible and big, while near-field blocks are small. It can be shown that in this case, the storage requirement of a matrix in $\mathbf{A}_{\mathcal{H}} \in \mathcal{H}(\mathcal{P}, r)$ is $\mathcal{O}(rn \log(n))$, while matrix-vector operations $\mathbf{x} \mapsto \mathbf{A}_{\mathcal{H}} \mathbf{x}$ can also be carried out in complexity $\mathcal{O}(rn \log(n))$.

Matrices X, Y associated to far-field blocks in \mathcal{P}_{far} can be calculated by interpolation of the kernel function, which needs to be known explicitly (Iske et al. 2017). This approach can be shown to approximate a given matrix $\mathbf{A}_{j,i} = \Phi(\mathbf{s}_j, \mathbf{s}_i)$ via a hierarchical matrix $\mathbf{A}_{\mathcal{H}}$ exponentially in the rank r , if the kernel function Φ is *asymptotically smooth* (Hackbusch 2015),

$$|\partial_{\mathbf{x}}^{\alpha} \partial_{\mathbf{y}}^{\beta} \Phi(\mathbf{x}, \mathbf{y})| \leq C(\alpha + \beta)! |\alpha + \beta| \gamma^{|\alpha + \beta|} |\mathbf{x} - \mathbf{y}|^{-|\alpha| - |\beta| - s},$$

for some independent constant $C > 0$, where $\alpha, \beta \in \mathbb{N}_0^d$ are multi-indices and $|\alpha| = \alpha_1 + \alpha_2$, $\alpha! = \alpha_1! \alpha_2!$ for $\alpha = (\alpha_1, \alpha_2)$. In particular, the TPS kernel Φ is asymptotically smooth (Iske et al. 2017; Löhndorf and Melenk 2017), and hence the matrix \mathbf{E} (or parts

of it) in the system (10) can be approximated exponentially in the rank r by an \mathcal{H} -matrix. Black-box methods which do not require the explicit knowledge of Φ also exist, the most obvious one being the Singular Value Decomposition (SVD). However, the SVD is not apt for large-scale computations due to its cubic complexity. We employed the computationally more convenient *Adaptive Cross Approximation* method (ACA, [Bebendorf \(2008\)](#)). Instead of a prescribed rank r , the ACA algorithm employs a prescribed error tolerance $\varepsilon > 0$ for the approximation of the far-field blocks, which are then approximated by different ranks. Definition 1 easily extends to this case.

2.4 Implementation

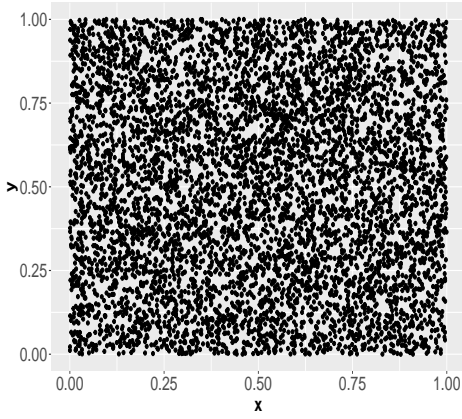
Hierarchical matrices are still not implemented in the statistical software R. As a remedy, we link the C++ library `Htool` ([Marchand 2020](#)) to R via `Rcpp` ([Eddelbuettel and François 2011](#)) and `RcppArmadillo` ([Eddelbuettel and Sanderson 2014](#)). With these latter two libraries we conducted all of the numerical experiments presented in Section (3) directly in R. To solve the linear system we use the Conjugate Gradient method ([Hestenes and Stiefel 1952](#)) along with Schur complement ([Zhang 2006](#)) (Appendix A).

3 Results

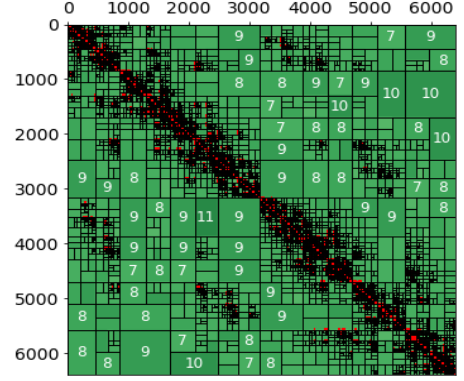
3.1 Matrix compression via an \mathcal{H} -matrix

First, we consider a grid of size 80^2 , $\lambda = 1$, $\varepsilon = 0.0001$, $\eta = 2$ as parameters for the \mathcal{H} -matrix compression. In Figure (1), we present the grid and the corresponding compressed \mathcal{H} -matrix indicating the local rank on admissible blocks as determined by the ACA algorithm according to the given tolerance ε . Here the number of points in the grid were chosen to have a better visualization of the \mathcal{H} -matrix and how the block matrices are distributed on it. Besides, this number of points will allow us to determine if, while the number of observations increases, then the computational cost has a linear growth for M3.

We studied the quality of the approximation (Subsection 3.2), the computational cost for three different approaches (Subsection 3.3), and the performance of the method under



(a) Unstructured points of size 80^2



(b) \mathcal{H} -matrix approximation of the dense matrix E

Figure 1: Unstructured points of size 80^2 (left side) and its respective \mathcal{H} -matrix (right side).

varying parameter settings (Subsection 3.4) considering the following:

- M1** : solving (10) with a direct solver provided by the `solver()` function in R,
- M2** : writing (10) equivalently as symmetric positive system (14) and using the Conjugate Gradient method,
- M3** : writing (10) equivalently as symmetric positive system (14) and using the Conjugate Gradient method, approximating the dense matrix E_{11} from (14) by an \mathcal{H} -matrix, given the tolerance ε for ACA and the admissibility coefficient η as compression parameters.

Experiments were done for a sequence of random grids generated by a two-dimensional uniform distribution on the square $(0,1) \times (0,1)$. The data $\mathbf{y}_i = f(\mathbf{x}_i)$, were generated by the Franke's function;

$$f(x, y) = \frac{3}{4}e^{(-\frac{1}{4}(9x-2)^2+(9y-2)^2)} + \frac{3}{4}e^{(-\frac{1}{49}(9x+1)^2-(\frac{1}{10})(9y+1)^2)} + \frac{1}{2}e^{(-\frac{1}{4}(9x-7)^2+(9y-3)^2)} - \frac{1}{2}e^{(-\frac{1}{5}(9x-4)^2+(9y-7)^2)},$$

which is a standard test function for two-dimensional spatial data fitting (Fasshauer 2007), and the spatial statistical model can be represented as

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \quad \text{for } i = 1, \dots, n, \quad \text{and } \varepsilon_i \sim N(0, \sigma^2 \mathbf{I}) \quad (11)$$

Naturally, we expect the computational time of the method M1 to scale cubically in the number of sites. For method M2 we expect mn^2 , m being the number of iterations of the conjugate gradient algorithm. However, the computational time of the method M3 is expected to scale like $mrn \log(n)$.

3.2 Quality of the approximation

We assume that method M1 is *computationally exact*, as it is carried out using a direct solver without matrix compression. Let f_j , $j = 1, 2, 3$ denote the solutions of the three different methods, and \mathbf{f} , respectively \mathbf{f}_j , be the vector given by the evaluations of f , respectively f_j , on a grid of size 80^2 . In Table 1, we compare the computational error of the three different methods by,

$$\text{comp-err}_j = \|\mathbf{f}_1 - \mathbf{f}_j\|_2,$$

and obtain the expected result that comp-err_3 is slightly bigger due to compression.

Table 1: Computational error of the methods M2 and M3 compared with M1 for a grid of size 80^2 .

Error	j=2	j=3
comp-err_j	2.6e-06	0.19

We also considered a sequence of grids with size ranging from 20^2 to 80^2 , generated by a

uniform distribution on the square $(0, 1) \times (0, 1)$ to compare the comp-err_j , on logarithmic scales (Figure 2).

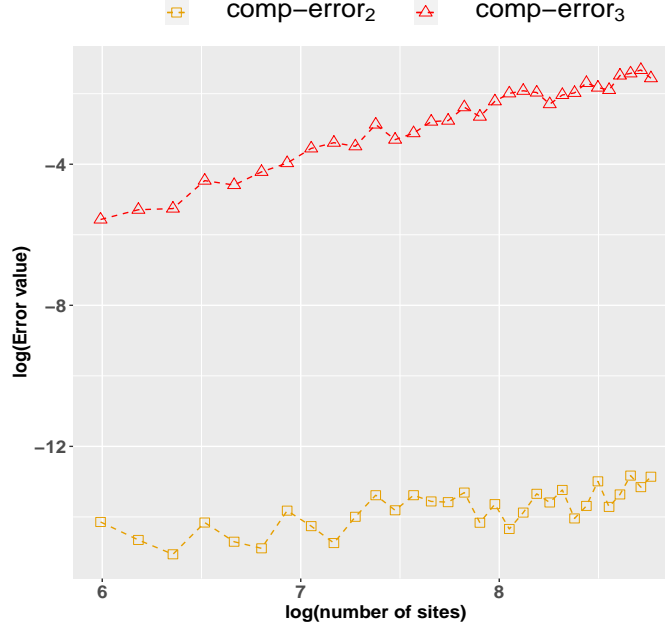


Figure 2: comp-error of estimation of the coefficients by the M1, M2 and M3 method for different sizes of the grid in logarithmic scale

Although the comp-err_3 increases as the number of sites is increasing as well, the scale of the error remains small.

To assess the interpolation performance of the three methods, we estimated the functions using M1, M2, and M3 on a grid of size 80^2 . Subsequently, we interpolated these functions onto a new grid of size 40^2 . For comparison purposes, we also applied the Franke’s function to the spatial coordinates to obtain the “real” values (\mathbf{f}_{real}), and then compare the interpolated function with those values using the Root Mean Square Error (RMSE) criterion,

$$\text{rmse}_j = \sqrt{\frac{1}{n} \sum_{i=1}^n (\mathbf{f}_{\text{real}} - \mathbf{f}_j)^2}$$

to measure the deviation of the STPS.

The three methods have similar structures, indicating that though the \mathcal{H} -matrix does not have the complete data, it gives us a good approximation of the interpolated function (Figure 3). Additionally, the rmse value is the same for all three methods (Table 2).

Table 2: rmse computed for M1, M2, and M3. The “real” values were obtained by applying the Franke’s function to a grid of size 40^2 .

	M1	M2	M3
rmse_j	0.01	0.01	0.01

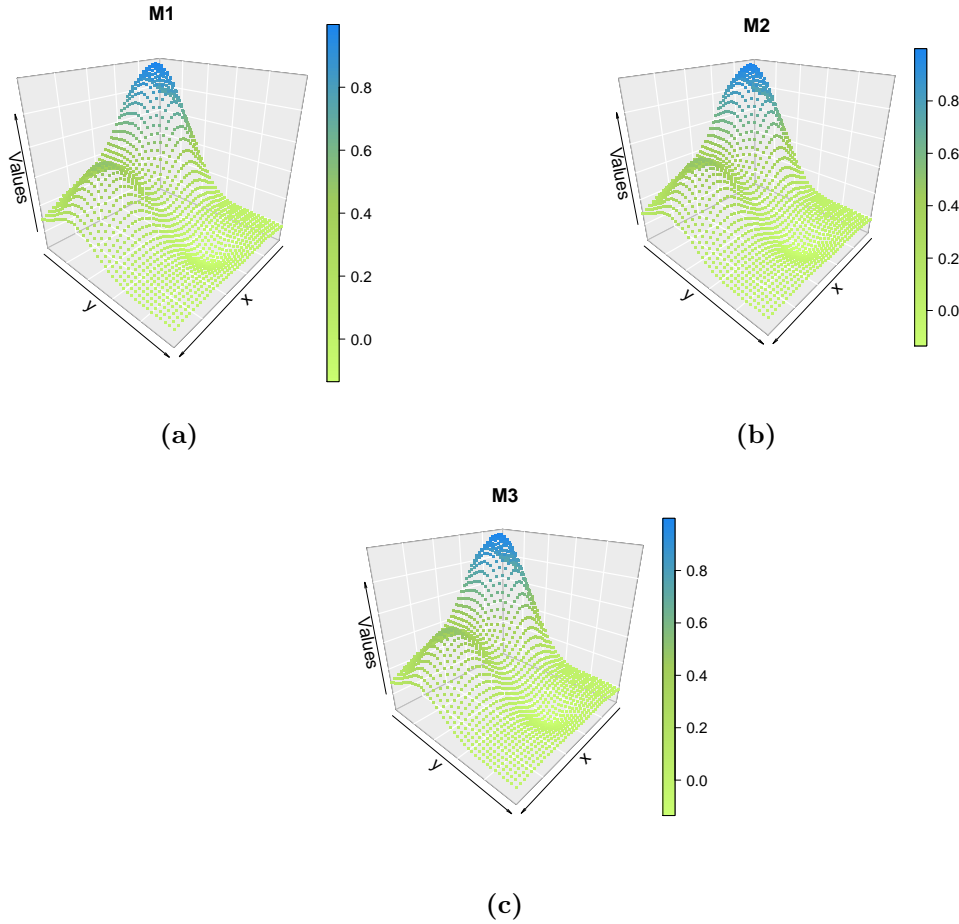


Figure 3: Spatial interpolation in a grid of 40^2 based on the functions calculated from the grid of size 80^2 by M1 (a), M2 (b) and M3 (c) respectively.

3.3 Computational efficiency

We plot the computational time for solving the linear system using the three methods mentioned before (also in logarithmic scales) for a sequence of grids with size ranging from 20^2 to 80^2 as well. As expected, the computational time of the method M1 scales cubically in the number of sites. We use a fixed tolerance for the conjugate gradient algorithm, and hence, as expected, the method M2 scales quadratically in the number of sites, while for

method M3 we observe practically a linear growth (Figure 4).

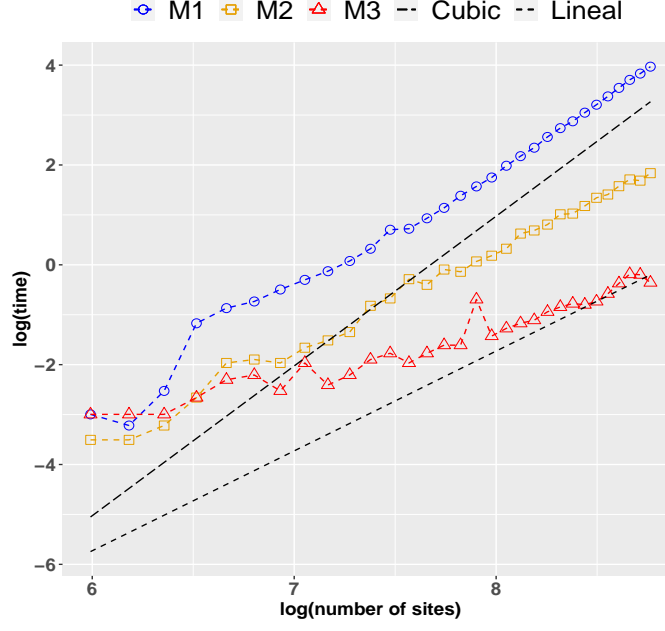


Figure 4: Time of estimation of the coefficients by the M1, M2 and M3 method for different sizes of the grid in logarithmic scale.

We also show a comparison in computational time and the `comp-error` for a small sequence of grids from 20^2 to 40^2 . This analysis considered the base case with $\varepsilon = 0.0001$ and $\eta = 2$. After increasing the number of observations, here also the \mathcal{H} -matrix (M3) allows us a fast computation compared with other methods (Figure 5), and the error in the last grid (40×40) remains relatively minimal in contrast to M1 (Table 3).

Table 3: Computational error of the methods M2 and M3 compared with M1 for a grid of 40×40 .

Error	j=2	j=3
<code>comp-err_j</code>	1.55e-06	0.05

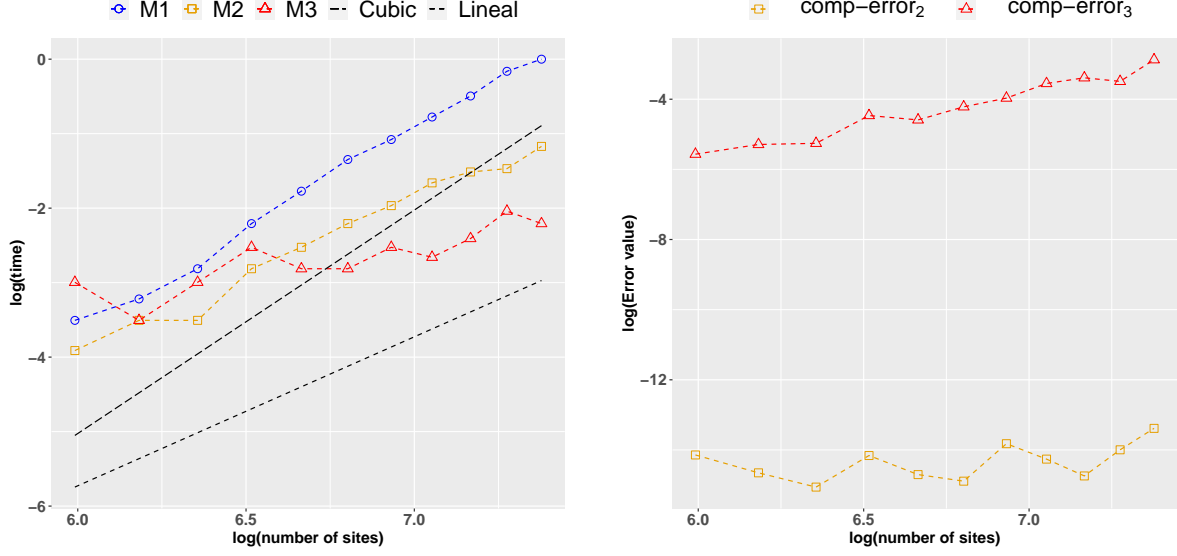


Figure 5: Time of estimation (left) and `comp-error` (right) for M1, M2 and M3 computed in a small sequences of grids, both are represented on logarithmic scales.

3.4 Sensitivity analysis

Various scenarios were assessed to determine the effects of different compression parameters in method M3 for spatial interpolation. The procedure outlined in subsection 3.2 was employed for these evaluations.

We prescribed different values of the compression parameters ε and η and calculated `rmse` (Table 4). We note that `rmsej` is almost constant if there is no compression present regarding the base case. However, the deviation grows with lower degree of compression, i.e., higher tolerance ε and/or admissibility parameter η . We also evaluated the computational performance (Figure 6) and errors `comp-errorj` (Figure 7) for all four cases given in Table 4.

Table 4: rmse for different compression parameters ε and η for interpolation purposes.

Cases	parameters \mathcal{H} -matrix	rmse
Case 1	$\varepsilon = 0.01, \eta = 5$	0.01
Case 2	$\varepsilon = 0.01, \eta = 10$	0.01
Case 3	$\varepsilon = 0.1, \eta = 5$	822
Case 4	$\varepsilon = 0.1, \eta = 10$	7526

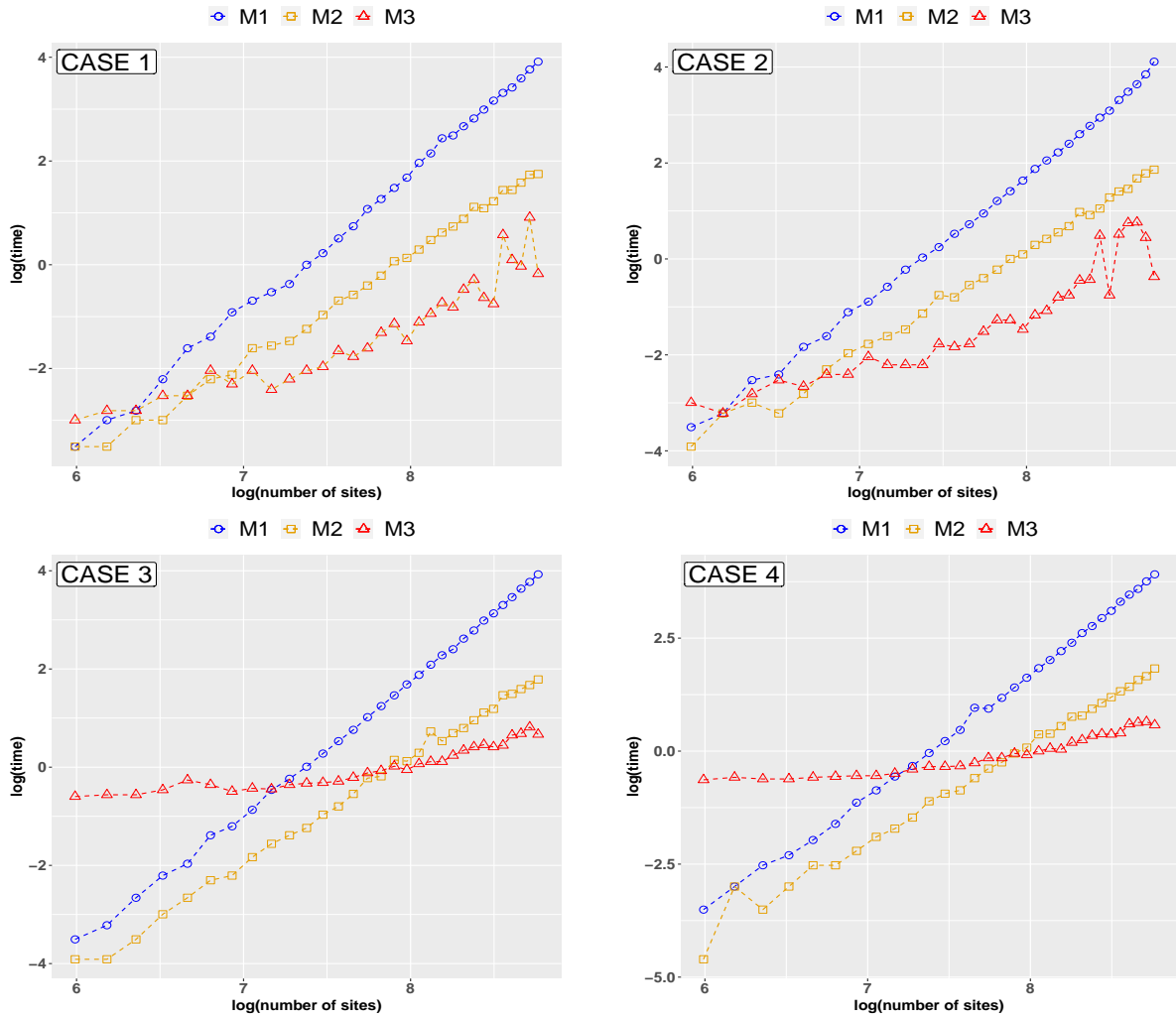


Figure 6: Times of estimation for M1, M2 and M3 using different parametrizations of the \mathcal{H} -matrix

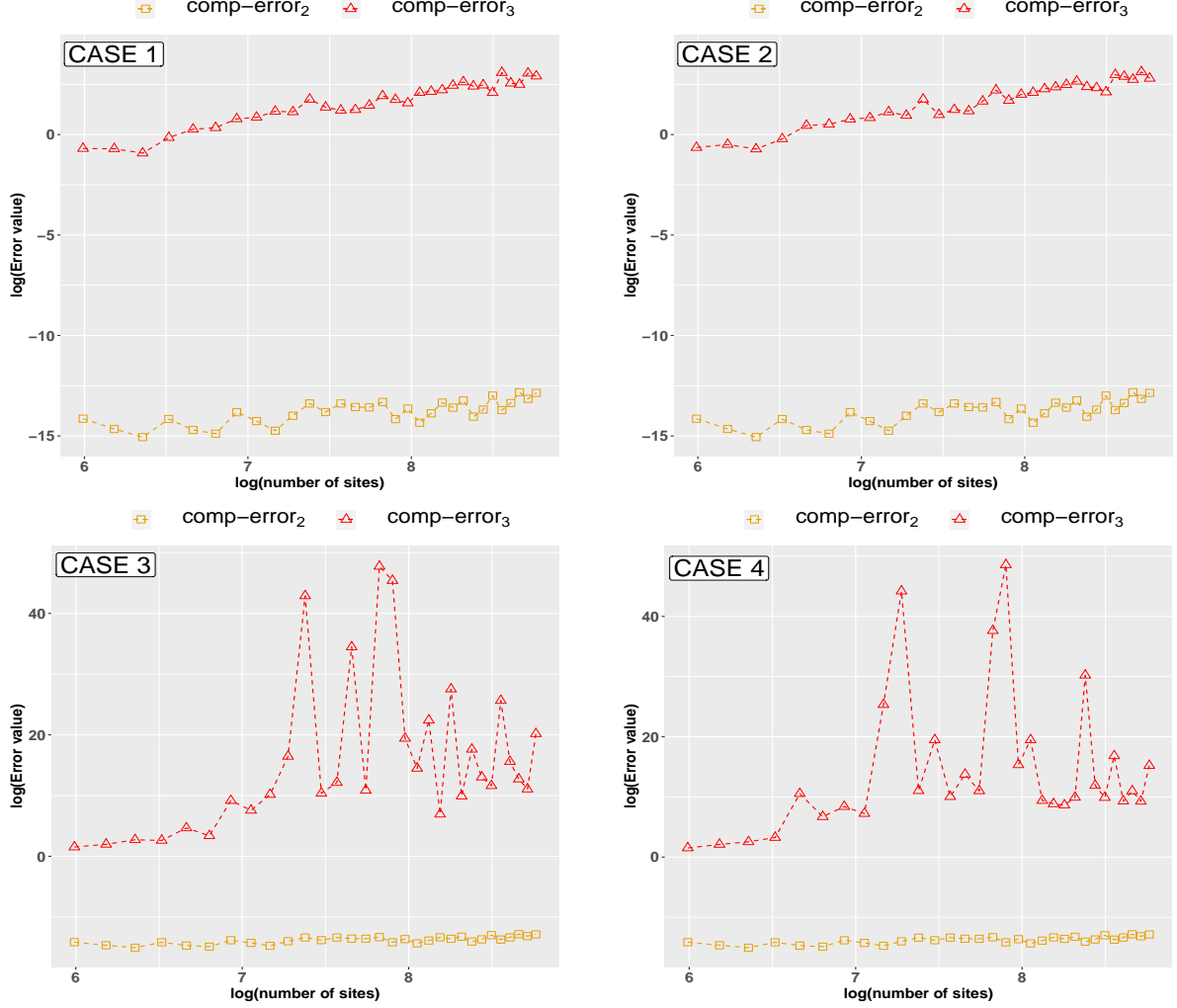


Figure 7: comp-error for M2 and M3 compared with M1 using different parametrizations to obtain the \mathcal{H} -matrix

For cases 1 and 2, when the value of ε is close to the value of case base ($\varepsilon = 0.0001$), the (**rmse**) is nearly the same. However, when this value is relaxed to expedite computation (cases 3 and 4), the **rmse** increases significantly, regardless of the value of η .

Since the smoothing parameter λ controls the smoothness of the estimated function, various scenarios for different values of this parameter are presented for method M3 evaluating its influence in the interpolation. To this end, we employ **rmse** as evaluation metric. As parameter ε controls the accuracy of the compressed matrix, we changed only the λ and η values (Table 5).

Table 5: rmse for different values of λ and η in M3

Smoothing parameter	ε	η	rmse
$\lambda = 1$	0.0001	2	0.01
$\lambda = 5$			0.04
$\lambda = 10$			0.05
$\lambda = 100$			0.10
$\lambda = 1$	0.0001	5	0.01
$\lambda = 5$			0.04
$\lambda = 10$			0.05
$\lambda = 100$			0.10
$\lambda = 1$	0.0001	10	0.01
$\lambda = 5$			0.04
$\lambda = 10$			0.05
$\lambda = 100$			0.10

3.5 Monte Carlo simulation

To evaluate the uncertainty in the approximation of the function $S(g)$, we perform a Monte Carlo simulation for spatial locations in a square grid of 20×20 distributed uniformly. Hence, the dense matrix is built in every iteration, and the Frank's function was perturbed adding an additional noise ($\varepsilon \sim N(0, 1)$) in every iteration as well. The grid size was selected to ascertain whether a limited number of spatial locations have implications in the estimated function of M3.

The Figure 8 shows the distribution of 1000 iterations using the Monte Carlo method for the sample median and sample interquartile range (IQR) for the function $S(g)$ from M1 and M3, along with their means (solid lines) and standard deviations (dashed lines).

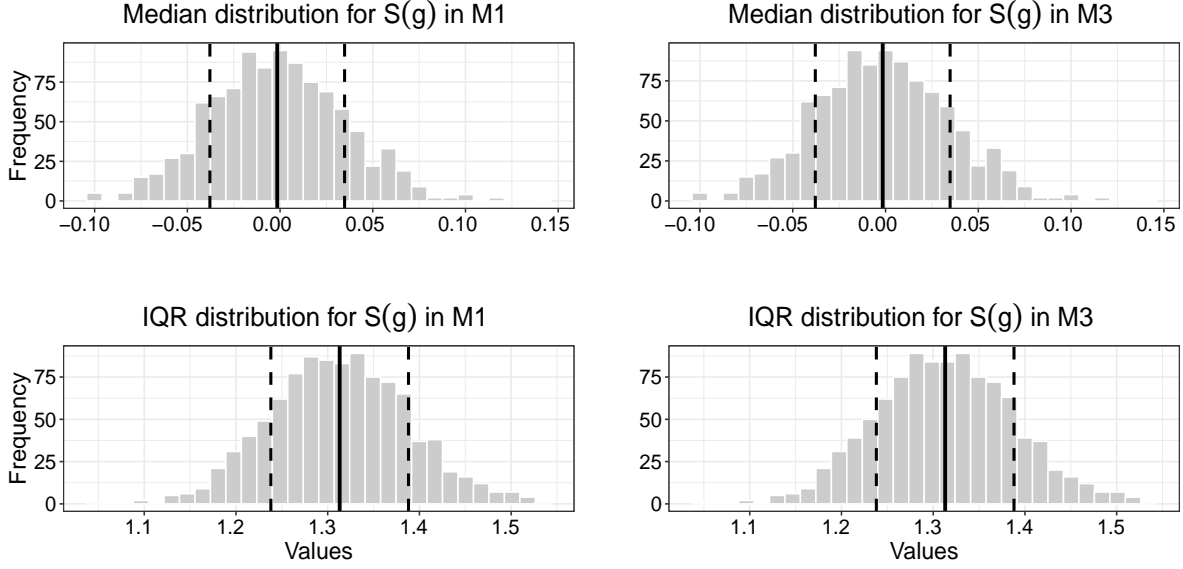


Figure 8: Median (top) and IQR (bottom) distribution from the Monte Carlo simulations for M1 and M3 respectively. The solid lines represents the mean of the distribution and the dashed lines represents the standard deviations.

The distribution of the summary statistics (median and IQR) are similar between M1 and M3, this implies that M3 effectively approximates the function $S(g)$ when contrasted with M1 (exact method). Both M1 and M3 exhibit the same mean and standard deviation values for their respective distributions of the median, namely -0.001 and 0.036. The corresponding confidence interval is $[-0.0038, 0.00072]$, lower and upper respectively.

Finally, as these statistics (median and IQR) have a probability distribution, we measure their spreads by its variances (Table 6),

Table 6: Variances estimated for M1 and M3 from the samples of the Monte Carlo simulations.

	Var(median)	Var(IQR)
M1	0.001	0.01
M3	0.001	0.01

showing that the variances in the distributions of the median and IQR remain consistent between M1 and M3.

3.6 Additional comparisons

We also compare M3 with a thin plate spline regression (TPSR), which is available in the R package “mgcv” (Wood (2011)). The methodology of TPSR is detailed in Wood (2003) and Wood (2017). For the TPSR, the *knots* were chosen considering the grid itself. That is, we use 20 *knots* for the grid of dimension 20×20 and so on to consider 80 *knots* in the grid of 80×80 .

The first comparison involves assessing the computational time required for the TPSR and M3 (Figure 9).

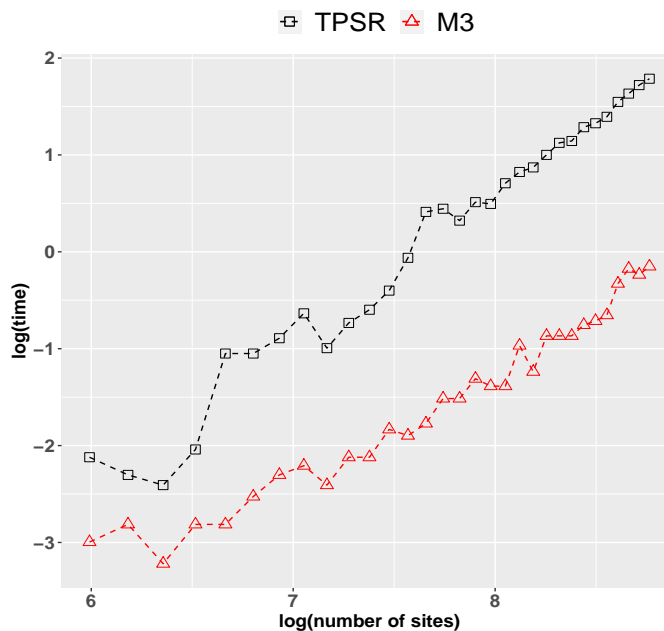


Figure 9: Time of estimation for the TPSR and M3 computed in a sequences of grids from 20^2 to 80^2 , represented on logarithmic scales.

For this particular case, the plot show us that M3 outperforms TPSR in terms of computational time for the selected grids. Finally, we interpolate a function on a new grid with dimensions 40^2 using method M3 and TPSR (Figure 10).

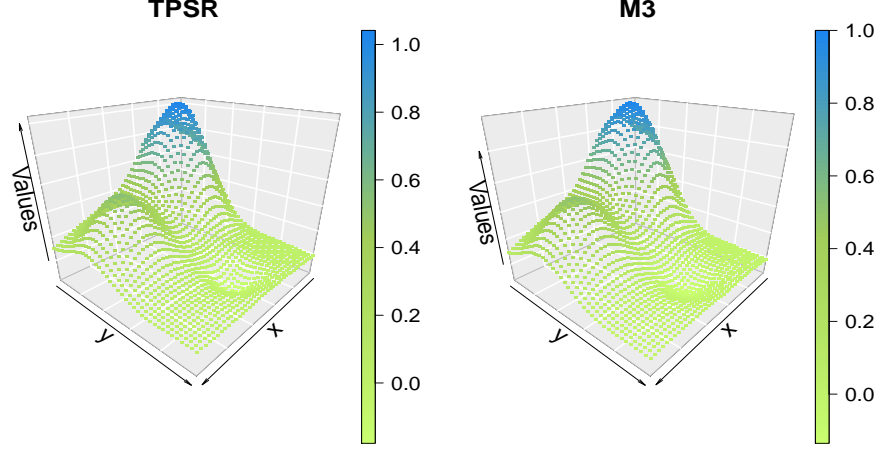


Figure 10: Spatial interpolation in a grid of size 40^2 from the TPSR and M3 respectively.

The function interpolated by TPSR and M3 are quite similar, with the only distinction being that in the former, values tend to be higher based on the structure of the function on the new grid. Figure 11 shows the interpolation for both methods, alongside their corresponding confidence intervals. We can see that TPSR interpolation displays heightened uncertainty towards the limits of the new grid, as it attempts to interpolate values right up to its spatial domain boundaries. Conversely, M3 demonstrates a smoother behavior, refraining from interpolating values at the extreme ends of the spatial domain.

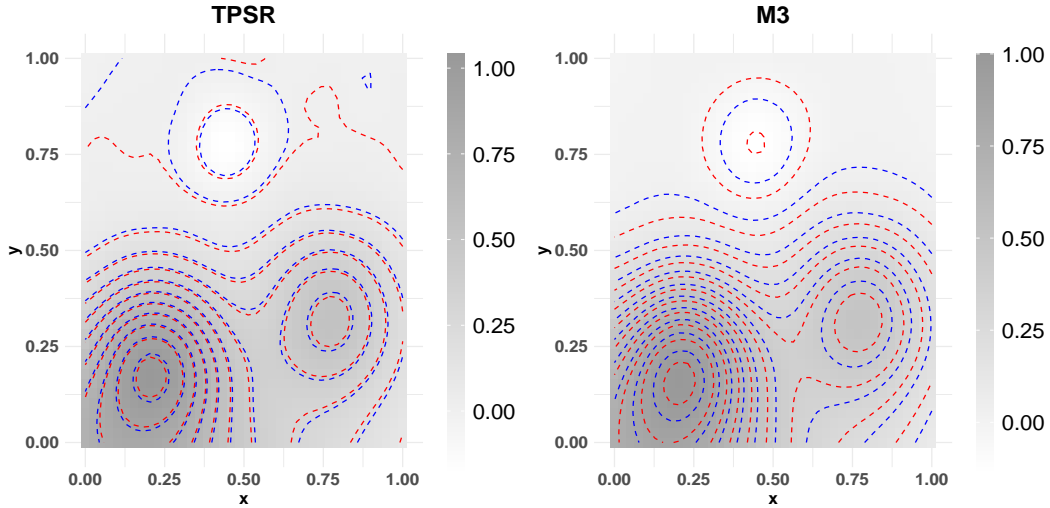


Figure 11: Spatial interpolation in a grid of size 40^2 from the TPSR and M3 respectively. In both cases, the dotted lines represents the lower confidence interval and the upper confidence interval.

For the above, and considering the **rmse**, M3 demonstrates a similar statistical performance than TPSR (Table 7).

Table 7: **rmse** for the interpolation of the TPSR and M3 in a grid of size 40^2 .

	TPSR	M3
rmse	0.004	0.010

4 Discussion

We presented a fast method for computing a STPS for spatial data, based on an \mathcal{H} -matrix. Different simulations on distinct grid sizes, comparing the computational cost (in time) and the error of estimation for three different methods were presented. In this comparative, our proposal (M3) was most efficient in computational time and with an acceptable error compared with the exact method (M1). Additional scenarios with a small number of observations were considered for M3, and in all of them the approximations were similar to the results obtained initially, that is, a fast computation and a small error. Furthermore, Monte Carlo simulations were conducted to assess the uncertainty of the approximated function when facing perturbations in the response variable and spatial locations. Finally, a comparison between M3 and the TPSR revealed that, in the specific case of simulation, data interpolation using M3 exhibits nearly equivalent statistical performance to TPSR.

While recent experiments have explored similar approaches (Litvinenko et al. (2019); Chen and Stein (2021)), it's important to note that these studies have primarily been conducted under the assumption of an underlying spatial stochastic process, which is itself determined by specific parameters defining its correlation structure. Furthermore, the estimation process was carried out utilizing the maximum likelihood method. Radial basis functions are not commonly employed directly in spatial or spatiotemporal modeling; however, they have been utilized to approximate covariance matrices within the probabilistic approach (Lindgren et al. (2011); Porcu et al. (2013); Nychka et al. (2015)).

One disadvantage of using SPTS for spatial interpolation is that it does not explicitly assume a spatial correlation between the sites (locations), in contrast to Kriging, which

utilizes variogram to describe the degree of spatial dependence of a spatial random field (Cressie (1989), Oliver et al. (2015)). For the above, it becomes challenging to ascertain the distance at which the decay of spatial correlation initiates. However, it’s worth noting that this aspect could also be considered a disadvantage for Kriging. This is because the effectiveness of spatial interpolation using Kriging relies on an underlying model (variogram) which, in turn, must be fitted using different correlation functions. On the other hand, STPS exhibit flexibility in accommodating spatial locations without necessitating the stringent assumption of stationarity, as the Kriging assumes (O’connell and Wolfinger (1997)). Furthermore, the inclusion of a “smooth” component in its structure enables us to assume the absence of singularities among the data points. This is a similar assumption made in Kriging by introducing the nugget effect (Hutchinson and Gessler (1994)). Regarding uncertainty in interpolation, Kriging provides a direct estimation of uncertainty through the variance, a quantity that STPS does not calculate. The variance in 11 can be estimated subsequent to fitting the spline model through Generalized Cross Validation (GCV) method. In contrast, when applying Kriging, this variance is estimated prior to the computation of the surface (Hutchinson and Gessler (1994)). Nevertheless, it is possible to derive this uncertainty using Monte Carlo simulations, as demonstrated in the study as well. In general, which method is more “appropriate” depends on the application and the objectives of the analysis (Handcock et al. (1994), Ren-ping et al. (2016)).

The current work also allows us to research in other mathematical methodologies to compress dense matrices, for example, using domain decomposition methods for scattered data interpolation (Le Borne and Wende (2019)) or using a preconditioner to solve an augmented linear system but viewed from a probabilistic perspective (Hennig et al. (2015); Oates and Sullivan (2019); Cockayne et al. (2019)). Also we think that this is the first step to use an \mathcal{H} -matrix in the R software since it is also available in Matlab (Ho (2020)), Python (Mikhalev and Oseledets (2016)) or Julia (Geoga et al. (2020)) as well. Today to use an R code with an \mathcal{H} -matrix is tricky, but all the machinery could be converted in a package and be available for the users in the future, since R is an essential software for statistical analysis.

Declaration of Competing Interest

The authors declare that he has no known competing financial interests or conflict of interest that could have appeared to influence the work reported in this paper.

CRedit authorship contribution statement

Joaquin Cavieres: Conceptualization, Formal analysis, Writing - original draft, Writing - review & editing, Visualization. **Michael Karkulik:** Conceptualization, Formal analysis, Writing - original draft, Supervision, Writing - review & editing.

References

- Mario Bebendorf. *Hierarchical matrices*. Springer, 2008. 4, 9
- Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011. 3
- M Bevilacqua and V Morales-Onate. Geomodels: a package for geostatistical gaussian and non gaussian data analysis. *R package version*, 1:3–4, 2018. 3
- Dario Andrea Bini, Fabio Di Benedetto, Eugene Tyrtysnikov, and Marc Van Barel. *Structured Matrices in Numerical Linear Algebra: Analysis, Algorithms and Applications*, volume 30. Springer, 2019. 4
- Joaquin Cavieres, German Ibacache-Pulgar, and Javier E Contreras-Reyes. Thin plate spline model under skew-normal random errors: estimation and diagnostic analysis for spatial data. *Journal of Statistical Computation and Simulation*, pages 1–21, 2022. 5
- Jie Chen and Michael L Stein. Linear-cost covariance functions for gaussian random fields. *Journal of the American Statistical Association*, pages 1–18, 2021. 22
- Jean-Paul Chiles and Pierre Delfiner. *Geostatistics: modeling spatial uncertainty*, volume 497. John Wiley & Sons, 2009. 2

- Jon Cockayne, Chris J Oates, Timothy John Sullivan, and Mark Girolami. Bayesian probabilistic numerical methods. *SIAM review*, 61(4):756–789, 2019. 23
- Peter Craven and Grace Wahba. Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische mathematik*, 31(4):377–403, 1978. 2
- Noel Cressie. Geostatistics. *The American Statistician*, 43(4):197–202, 1989. 23
- Jeffrey J Early and Adam M Sykulski. Smoothing and interpolating noisy gps data with smoothing splines. *Journal of Atmospheric and Oceanic Technology*, 37(3):449–465, 2020. 2
- Dirk Eddelbuettel and Romain François. Rcpp: Seamless r and c++ integration. *Journal of statistical software*, 40:1–18, 2011. 9
- Dirk Eddelbuettel and Conrad Sanderson. Rcpparmadillo: Accelerating r with high-performance c++ linear algebra. *Computational Statistics & Data Analysis*, 71:1054–1063, 2014. 9
- Gregory E Fasshauer. *Meshfree approximation methods with MATLAB*, volume 6. World Scientific, 2007. 11
- Christopher J Geoga, Mihai Anitescu, and Michael L Stein. Scalable gaussian process computations using hierarchical matrices. *Journal of Computational and Graphical Statistics*, 29(2):227–237, 2020. 23
- Peter J Green and Bernard W Silverman. *Nonparametric regression and generalized linear models: a roughness penalty approach*. Crc Press, 1993. 2, 5, 6, 7
- Wolfgang Hackbusch. *Hierarchical matrices: algorithms and analysis*, volume 49 of *Springer Series in Computational Mathematics*. Springer, Heidelberg, 2015. ISBN 978-3-662-47323-8; 978-3-662-47324-5. 10.1007/978-3-662-47324-5. URL <https://doi-org.usm.idm.oclc.org/10.1007/978-3-662-47324-5>. 4, 7, 8

- Mark S Handcock, Kristen Meier, and Douglas Nychka. Kriging and splines: An empirical comparison of their predictive performance in some applications: Comment. *Journal of the American Statistical Association*, 89(426):401–403, 1994. [23](#)
- Philipp Hennig, Michael A Osborne, and Mark Girolami. Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2179):20150142, 2015. [23](#)
- Magnus R Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving. *Journal of research of the National Bureau of Standards*, 49(6):409, 1952. [9](#)
- Kenneth L Ho. Flam: Fast linear algebra in matlab-algorithms for hierarchical matrices. *Journal of Open Source Software*, 5(51):1906, 2020. [23](#)
- Dionissios T Hristopulos, Andrew Pavlides, Vasiliki D Agou, and Panagiota Gkafa. Stochastic local interaction model: An alternative to kriging for massive datasets. *Mathematical Geosciences*, pages 1–43, 2021. [3](#)
- MF Hutchinson and PE Gessler. Splines—more than just a smooth interpolator. *Geoderma*, 62(1-3):45–67, 1994. [23](#)
- Michael F Hutchinson and Frank R de Hoog. Smoothing noisy data with spline functions. *Numerische Mathematik*, 47:99–106, 1985. [2](#)
- Armin Iske, Sabine Le Borne, and Michael Wende. Hierarchical matrix approximation for kernel-based scattered data interpolation. *SIAM Journal on Scientific Computing*, 39(5):A2287–A2316, 2017. [4](#), [8](#)
- Sabine Le Borne and Michael Wende. Domain decomposition methods in scattered data interpolation with conditionally positive definite radial basis functions. *Computers & Mathematics with Applications*, 77(4):1178–1196, 2019. [23](#)
- Gwo-Fong Lin and Lu-Hsien Chen. A spatial interpolation method based on radial basis function networks incorporating a semivariogram model. *Journal of Hydrology*, 288(3-4): 288–298, 2004. [2](#)

- Finn Lindgren, Håvard Rue, and Johan Lindström. An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011. 3, 22
- Alexander Litvinenko, Ying Sun, Marc G Genton, and David E Keyes. Likelihood approximation with hierarchical matrices for large spatial datasets. *Computational Statistics & Data Analysis*, 137:115–132, 2019. 3, 22
- M. Löhndorf and J. M. Melenk. On thin plate spline interpolation. In *Spectral and high order methods for partial differential equations—ICOSAHOM 2016*, volume 119 of *Lect. Notes Comput. Sci. Eng.*, pages 451–466. Springer, Cham, 2017. 10.1007/978-3-319-65870-4_3. URL https://doi-org.usm.idm.oclc.org/10.1007/978-3-319-65870-4_3. 4, 8
- Tom Lyche. *Numerical linear algebra and matrix factorizations*, volume 22. Springer Nature, 2020. 4
- Pierre Marchand. *Schwarz methods and boundary integral equations*. PhD thesis, Sorbonne Université, 2020. 9
- Georges Matheron. Principles of geostatistics. *Economic geology*, 58(8):1246–1266, 1963. 3
- A Yu Mikhalev and Ivan V Oseledets. Iterative representing set selection for nested cross approximation. *Numerical linear algebra with applications*, 23(2):230–248, 2016. 23
- Douglas Nychka. Bayesian confidence intervals for smoothing splines. *Journal of the American Statistical Association*, 83(404):1134–1143, 1988. 2
- Douglas Nychka, Soutir Bandyopadhyay, Dorit Hammerling, Finn Lindgren, and Stephan Sain. A multiresolution gaussian process model for the analysis of large spatial datasets. *Journal of Computational and Graphical Statistics*, 24(2):579–599, 2015. 22
- Chris J Oates and Timothy John Sullivan. A modern retrospective on probabilistic numerics. *Statistics and Computing*, 29(6):1335–1351, 2019. 23

- Michael A O’connell and Russell D Wolfinger. Spatial regression models, response surfaces, and process optimization. *Journal of Computational and Graphical Statistics*, 6(2):224–241, 1997. [23](#)
- Margaret A Oliver, Richard Webster, et al. Basic steps in geostatistics: the variogram and kriging. Technical report, Springer, 2015. [23](#)
- Emilio Porcu, Daryl J Daley, Martin Buhmann, and Moreno Bevilacqua. Radial basis functions with compact support for multivariate geostatistics. *Stochastic environmental research and risk assessment*, 27:909–922, 2013. [22](#)
- Zhang Ren-ping, Guo Jing, Liang Tian-gang, Feng Qi-sheng, and Yusupujiang Aimaiti. Comparing interpolation techniques for annual temperature mapping across xinjiang region. In *IOP Conference Series: Earth and Environmental Science*, volume 46, page 012028. IOP Publishing, 2016. [23](#)
- Sergej Rjasanow and Olaf Steinbach. *The fast solution of boundary integral equations. Mathematical and Analytical Techniques with Applications to Engineering*. Springer, New York, 2007. ISBN 978-0-387-34041-8; 0-387-34041-6. [7](#)
- Michael Scheuerer, Robert Schaback, and Martin Schlather. Interpolation of spatial data—a stochastic or a deterministic problem? *European Journal of Applied Mathematics*, 24(04):601–629, 2013. [2](#), [3](#)
- David S Trossman, LuAnne Thompson, and Susan L Hautala. Application of thin-plate splines in two dimensions to oceanographic tracer data. *Journal of Atmospheric and Oceanic Technology*, 28(11):1522–1538, 2011. [2](#)
- Grace Wahba. Smoothing noisy data with spline functions. *Numerische mathematik*, 24(5):383–393, 1975. [2](#)
- Grace Wahba. Bayesian “confidence intervals” for the cross-validated smoothing spline. *Journal of the Royal Statistical Society: Series B (Methodological)*, 45(1):133–150, 1983. [2](#)

- Grace Wahba. *Spline models for observational data*, volume 59. Siam, 1990. 2, 3, 6
- Grace Wahba, Yuedong Wang, Chong Gu, Ronald Klein, and Barbara Klein. Smoothing spline anova for exponential families, with application to the wisconsin epidemiological study of diabetic retinopathy: the 1994 neyman memorial lecture. *The Annals of Statistics*, 23(6):1865–1895, 1995. 2
- Yuedong Wang. *Smoothing splines: methods and applications*. CRC Press, 2011. 2
- Holger Wendland. *Scattered data approximation*, volume 17. Cambridge university press, 2004. 2, 3, 5
- Holger Wendland. Computational aspects of radial basis function approximation. In *Studies in Computational Mathematics*, volume 12, pages 231–256. Elsevier, 2006. 4
- Simon N Wood. Thin plate regression splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(1):95–114, 2003. 2, 5, 20
- Simon N Wood. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 73(1):3–36, 2011. 20
- Simon N Wood. *Generalized additive models: an introduction with R*. CRC press, 2017. 3, 20
- Yunfei Xu, Jongeun Choi, Sarat Dass, and Taps Maiti. Bayesian prediction and adaptive sampling algorithms for mobile sensor networks. In *Proceedings of the 2011 American Control Conference*, pages 4195–4200. IEEE, 2011. 2
- Fuzhen Zhang. *The Schur complement and its applications*, volume 4. Springer Science & Business Media, 2006. 9

A Appendix A

We aim to solve the augmented linear system expressed in Equation (10). The matrix on the left-hand side of this system is not positive definite, and neither is the sub-matrix $\mathbf{E} + \lambda \mathbf{I}$. However, $\mathbf{E} + \lambda \mathbf{I}$ is positive definite in the subspace $\{\mathbf{c} \in \mathbb{R}^n \mid \mathbf{P}^\top \mathbf{c} = \mathbf{0}\}$, since in this case

$$\mathbf{c}^\top (\mathbf{E} + \lambda \mathbf{I}) \mathbf{c} = \mathbf{c}^\top \mathbf{E} \mathbf{c} + \lambda \|\mathbf{c}\|_2^2 = J(g) + \lambda \|\mathbf{c}\|^2 \geq \lambda \|\mathbf{c}\|^2. \quad (12)$$

We separate the set of sites $S = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ into two sets $S_1 = \{\mathbf{x}_i, i = 1, \dots, n-3\}$ and $S_2 = \{\mathbf{x}_{n-2}, \mathbf{x}_{n-1}, \mathbf{x}_n\}$. Without loss of generality, we can assume that the points in S_2 are not collinear. We write accordingly $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$ and $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2)$ and the linear system in Equation (10) becomes

$$\begin{pmatrix} \mathbf{E}_{11} + \lambda \mathbf{I} & \mathbf{E}_{12} & \mathbf{P}_1 \\ \mathbf{E}_{21} & \mathbf{E}_{22} + \lambda \mathbf{I} & \mathbf{P}_2 \\ \mathbf{P}_1^\top & \mathbf{P}_2^\top & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{d} \end{pmatrix} = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{0} \end{pmatrix}, \quad (13)$$

where $\mathbf{E}_{11} + \lambda \mathbf{I} \in \mathbb{R}^{(n-3) \times (n-3)}$ and $\mathbf{E}_{22} + \lambda \mathbf{I} \in \mathbb{R}^{3 \times 3}$. Note that \mathbf{P}_2 is invertible because the points in S_2 are not collinear, and so

$$\begin{pmatrix} \mathbf{E}_{22} + \lambda \mathbf{I} & \mathbf{P}_2 \\ \mathbf{P}_2^\top & \mathbf{0} \end{pmatrix}$$

is invertible. Given the above, we can use the Schur complement to write

$$\begin{aligned}
(\mathbf{E}_{11} + \lambda \mathbf{I})\mathbf{c}_1 - \begin{pmatrix} \mathbf{E}_{12} & \mathbf{P}_1 \end{pmatrix} \begin{pmatrix} \mathbf{E}_{22} + \lambda \mathbf{I} & \mathbf{P}_2 \\ \mathbf{P}_2^\top & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{E}_{21} \\ \mathbf{P}_1^\top \end{pmatrix} \mathbf{c}_1 \\
= \mathbf{y}_1 - \begin{pmatrix} \mathbf{E}_{12} & \mathbf{P}_1 \end{pmatrix} \begin{pmatrix} \mathbf{E}_{22} + \lambda \mathbf{I} & \mathbf{P}_2 \\ \mathbf{P}_2^\top & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{y}_2 \\ \mathbf{0} \end{pmatrix}
\end{aligned} \tag{14}$$

The matrix of this linear system

$$\mathbf{M} = \mathbf{E}_{11} + \lambda \mathbf{I} - \begin{pmatrix} \mathbf{E}_{12} & \mathbf{P}_1 \end{pmatrix} \begin{pmatrix} \mathbf{E}_{22} + \lambda \mathbf{I} & \mathbf{P}_2 \\ \mathbf{P}_2^\top & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{E}_{21} \\ \mathbf{P}_1^\top \end{pmatrix}$$

is obviously symmetric. Furthermore, it is positive definite, since for $\mathbf{0} \neq \mathbf{c}_1 \in \mathbb{R}^{n-3}$ we have

$$\begin{aligned}
\mathbf{M}\mathbf{c}_1 &= (\mathbf{E}_{11} + \lambda \mathbf{I})\mathbf{c}_1 + \begin{pmatrix} \mathbf{E}_{12} & \mathbf{P}_1 \end{pmatrix} \begin{pmatrix} \mathbf{E}_{22} + \lambda \mathbf{I} & \mathbf{P}_2 \\ \mathbf{P}_2^\top & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} -\mathbf{E}_{21}\mathbf{c}_1 \\ -\mathbf{P}_1^\top \mathbf{c}_1 \end{pmatrix} \\
&= (\mathbf{E}_{11} + \lambda \mathbf{I})\mathbf{c}_1 + \begin{pmatrix} \mathbf{E}_{12} & \mathbf{P}_1 \end{pmatrix} \begin{pmatrix} -\mathbf{P}_2^{-\top} \mathbf{P}_1^\top \mathbf{c}_1 \\ -\mathbf{P}_2^{-1} \mathbf{E}_{21} \mathbf{c}_1 - \mathbf{P}_2^{-1} (\mathbf{E}_{22} + \lambda \mathbf{I}) (-\mathbf{P}_2^{-\top} \mathbf{P}_1^\top) \mathbf{c}_1 \end{pmatrix} \\
&= (\mathbf{E}_{11} + \lambda \mathbf{I})\mathbf{c}_1 + \mathbf{E}_{12} (-\mathbf{P}_2^{-\top} \mathbf{P}_1^\top) \mathbf{c}_1 + (-\mathbf{P}_1 \mathbf{P}_2^{-1}) \mathbf{E}_{21} \mathbf{c}_1 + (-\mathbf{P}_1 \mathbf{P}_2^{-1}) (\mathbf{E}_{22} + \lambda \mathbf{I}) \mathbf{P}_2^{-\top} \mathbf{P}_1^\top \mathbf{c}_1
\end{aligned}$$

We conclude that

$$\mathbf{c}_1^\top \mathbf{M} \mathbf{c}_1 = \begin{pmatrix} \mathbf{c}_1 \\ -\mathbf{P}_2^{-\top} \mathbf{P}_1^\top \mathbf{c}_1 \end{pmatrix}^\top (\mathbf{E} + \lambda \mathbf{I}) \begin{pmatrix} \mathbf{c}_1 \\ -\mathbf{P}_2^{-\top} \mathbf{P}_1^\top \mathbf{c}_1 \end{pmatrix},$$

and hence \mathbf{M} is positive definite since Equation (12),

$$\boldsymbol{P}^\top \begin{pmatrix} \boldsymbol{c}_1 \\ -\boldsymbol{P}_2^{-\top} \boldsymbol{P}_1^\top \boldsymbol{c}_1 \end{pmatrix} = \mathbf{0}.$$