Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

A Solution to the Ill-Conditioning of Gradient-Enhanced Covariance Matrices for Gaussian Processes

André L. Marchildon | **David W. Zingg**

[1]Institute for Aerospace Studies, University of Toronto, Ontario, Canada

Correspondence
Corresponding author André L. Marchildon:
Email: andre.marchildon@mail.utoronto.ca

Funding Information
This research was supported by the Natural Sciences and Engineering Research Council of Canada and the Ontario Graduate Scholarship Program.

Abstract

Gaussian processes provide probabilistic surrogates for various applications including classification, uncertainty quantification, and optimization. Using a gradient-enhanced covariance matrix can be beneficial since it provides a more accurate surrogate relative to its gradient-free counterpart. An acute problem for Gaussian processes, particularly those that use gradients, is the ill-conditioning of their covariance matrices. Several methods have been developed to address this problem for gradient-enhanced Gaussian processes but they have various drawbacks such as limiting the data that can be used, imposing a minimum distance between evaluation points in the parameter space, or constraining the hyperparameters. In this paper a diagonal preconditioner is applied to the covariance matrix along with a modest nugget to ensure that the condition number of the covariance matrix is bounded, while avoiding the drawbacks listed above. The method can be applied with any twice-differentiable kernel and when there are noisy function and gradient evaluations. Optimization results for a gradient-enhanced Bayesian optimizer with the Gaussian kernel are compared with the use of the preconditioning method, a baseline method that constrains the hyperparameters, and a rescaling method that increases the distance between evaluation points. The Bayesian optimizer with the preconditioning method converges the optimality, i.e. the ℓ_2 norm of the gradient, an additional 5 to 9 orders of magnitude relative to when the baseline method is used and it does so in fewer iterations than with the rescaling method. The preconditioning method is available in the open source Python library GpGradPy, which can be found at https://github.com/marchildon/gpgradpy/tree/paper_precon.

KEYWORDS
Gaussian process, Covariance matrix, Condition number, Bayesian optimization

1 | INTRODUCTION

In diverse fields and for various applications, such as uncertainty quantification, classification, regression, and optimization, an expensive function of interest must be repeatedly evaluated [1,2,3,4]. To minimize the computational cost it is desirable to minimize the number of expensive function evaluations. One way to achieve this is by constructing a surrogate that approximates the function of interest and is inexpensive to evaluate. Various methods to construct surrogates are available, such as fixed basis functions (e.g. polynomials), splines, or Gaussian processes (GPs) [5,6]. GPs are popular since their posteriors are nonparametric probabilistic surrogates. The nonparametric component of the surrogate indicates that it does not depend on a parametric functional form, unlike a polynomial surrogate where the order of the basis function must be selected a priori. The probabilistic component enables the surrogate to provide an estimate for the function of interest and to quantify the uncertainty in its estimate [6]. A GP requires a mean and a covariance function [7,8]. A constant is often used for the former and its value is set by maximizing the marginal log-likelihood [9,10,11,12]. For the covariance function many kernels are available [13,6], the most popular of which is the Gaussian kernel, which is also known as the squared exponential kernel [6,2,14]. The desirable properties of this kernel include its hyperparameters that can be tuned, its simplicity, and its smoothness. This final property enables the surrogate to be constructed using gradient evaluations, which makes the surrogate more accurate [15,3,14].

Gradient-enhanced GPs use both the value and gradient of the function of interest to construct the probabilistic surrogate. By using gradients with the GP, a more accurate surrogate is constructed that matches both the value and gradient of the function of interest where it has been evaluated in the parameter space [16,17,18]. This is particularly useful in high-dimensional parameter spaces since a single gradient evaluation provides much more information than a single function evaluation. The gradient-enhanced covariance matrix can be constructed either with the direct method or the indirect method [19]. The former modifies the structure of the gradient-free covariance matrix while the latter does not. The direct method is much more common [20,15,14,21] and is used in this paper. A drawback of using gradient-enhanced GPs is that the covariance matrix is larger than its gradient-free counterpart and is thus more expensive to invert. Various strategies have been developed to mitigate this additional cost by using random Fourier features [22], or by exploiting the structure of the gradient-enhanced covariance matrix [23].

A ubiquitous problem in the use of GPs is the ill-conditioning of their covariance matrices [7,24,25]. This problem is present with the use of many kernels, including the Gaussian kernel. Various factors have been identified that exacerbate the ill-conditioning, such as having the data points too close together [13,26]. The ill-conditioning of the covariance matrix is problematic since it can cause the Cholesky factorization to fail [27], and it also increases the numerical error. Regularizing the gradient-free covariance matrix, i.e. adding a small positive nugget to the diagonal of the matrix, is sufficient to ensure that its condition number is below a user-set threshold [28].

The ill-conditioning of the gradient-enhanced covariance matrix is even more acute than the gradient-free case, and the addition of a nugget is insufficient on its own to alleviate this problem [29,30]. Various approaches have been attempted to mitigate the ill-conditioning problem, such as removing certain data points until the condition number is sufficiently low [31,15], or imposing a minimum distance constraint between data points in the parameter space [16]. Both methods have significant drawbacks since they restrict the data available to construct the surrogate. A recent method does ensure that the condition number of the gradient-enhanced covariance matrix remains below a user-set threshold when the Gaussian kernel is used [32]. This method uses non-isotropic rescaling of the data in order to have a set minimum distance between the data points. While data points cannot be collocated, they can get arbitrarily close in the parameter space, and the method allows all of the data points to be kept in the construction of the gradient-enhanced covariance matrix. However, the drawback of this method is that, in some cases, the rescaling needs to be done iteratively, which requires the hyperparameters to be optimized again. This adds additional complexity and computational cost.

The method presented in this paper expands on the secondary method presented in Dalbey [15]. This method involves two steps, preconditioning (also called equilibrating) the covariance matrix and then regularizing it with the addition of a nugget. The preconditioning method shares the same benefits as the rescaling method from Marchildon and Zingg [32], i.e. all of the data points can be used, there is no minimum distance constraint between the data points in the parameter space, and the condition number of the gradient-enhanced covariance matrix is bounded. The preconditioning method also has two additional benefits: it only requires a single optimization of the hyperparameters, i.e. it is not iterative, and there is no need for a constraint on the condition number for the optimization of the hyperparameters. This simplifies the implementation of the preconditioning method and reduces its computational cost.

The method presented in this paper provides several advantages relative to the one presented by Dalbey [15], which requires the condition number of the covariance matrix to be approximated. This adds computational cost and it does not guarantee that the selected nugget is sufficient to bound the condition number below a set threshold, as acknowledged by the author. Furthermore, the nugget varies as the hyperparameters are changed but its gradient cannot be accurately calculated since it depends on an approximation to the condition number of the covariance matrix. The method presented in this paper calculates a nugget value sufficient to bound the condition number of the covariance matrix with the use of the Gershgorin circle theorem. This nugget guarantees that the condition number of the covariance matrix is below a user-set threshold, it can be applied to cases with noisy objective and gradient evaluations, and accurate gradients can be calculated to perform gradient-based optimization of the hyperparameters.

The preconditioning and rescaling methods with the Gaussian, Matérn $\frac{5}{2}$, and rational quadratic kernels are available in the Python library GpGradPy, which can be accessed at https://github.com/marchildon/gpgradpy/tree/paper_precon.

The notation used in this paper is presented in Section 2. In Section 3 the GP is presented along with the Gaussian kernel and the covariance matrix. In Section 4 previous methods to bound the condition number of the gradient-enhanced covariance matrix are presented while the preconditioning method is introduced in Section 5. Numerical results are provided in Section 6, where it is demonstrated that the preconditioning method bounds the condition number of the covariance matrix for both noise-free and noisy test cases. Furthermore, test cases with a Bayesian optimizer are presented to showcase the practical advantage of using the preconditioning method over the methods from Section 4. Finally, the conclusions of the paper are in Section 7.

2 | NOTATION

Sans-serif capital letters are used for matrices. For example, I is the identity matrix, and X is an $n_x \times d$ matrix that holds the location of n_x evaluation points in a d dimensional parameter space. Vectors are denoted in lowercase bold font. For instance, $\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors of length d denoting arbitrary points in the parameter space. The i-th row of X is denoted as $\boldsymbol{x}_{i:}$ and its j-th column is indicated as $\boldsymbol{x}_{:j}$. Finally, scalars are denoted in lowercase letters such as x_{ij}, which is the entry at the i-th row and j-th column of X. The symbols $\mathbf{0}_d$ and $\mathbf{1}_d$ are vectors of length d with all of their entries equal to zero and one, respectively. In addition, $\mathbb{O}$ and $\mathbb{1}$ are square matrices with all of their entries equal to zero and one, respectively. The math operator $\text{diag}(\cdot)$ either converts a vector into a diagonal matrix or returns the diagonal entries of a matrix as a vector, depending on its input. Finally, matrices with exponents in brackets, e.g. $\mathsf{R}^{(2)}$, have the exponent applied elementwise to the entries of the matrix.

3 | GAUSSIAN PROCESS

3.1 | Overview of Gaussian processes

A GP is the generalization of a Gaussian distribution for a scalar random variable to a Gaussian distribution over functions[6]. Just like scalar values can be sampled from a Gaussian distribution, functions can be sampled from a GP. To fully define a Gaussian distribution we require a mean and standard deviation value. Similarly, to fully define a GP we require a mean function and a covariance function. The mean function is commonly selected to be the scalar β, which is a hyperparameter that is selected by maximizing the marginal log-likelihood function that will be presented in Section 3.4. The kernel, which describes the covariance for a random process, is introduced in Section 3.2.

The posterior of a GP is a probabilistic surrogate that approximates a function of interest $f(\boldsymbol{x})$. At each point in the parameter space the posterior of the GP is normally distributed and its mean and variance are straightforward to evaluate, as is demonstrated in Section 3.4. A GP can utilize noisy function and gradient evaluations of the function of interest. We assume that the noisy function and gradient evaluations are the result of additive noise that is independent, identically distributed, and drawn from a zero-mean Gaussian distribution. The noisy function and derivative evaluations are thus given by

$$\tilde{f}(\boldsymbol{x}) = f(\boldsymbol{x}) + \epsilon_f \tag{1}$$

$$\left.\widetilde{\frac{\partial f}{\partial x_i}}\right|_{\boldsymbol{x}} = \left.\frac{\partial f}{\partial x_i}\right|_{\boldsymbol{x}} + \epsilon_{\nabla f} \quad \forall\, i \in \{1, \ldots, d\}, \tag{2}$$

where $\epsilon_f \sim \mathcal{N}(0, \sigma_f^2)$ and $\epsilon_{\nabla f} \sim \mathcal{N}(0, \sigma_{\nabla f}^2)$ with σ_f and $\sigma_{\nabla f}$ being the standard deviation for the noise on the evaluation of the function $f(\cdot)$ and its gradient, respectively.

3.2 | Gradient-free covariance matrix

The method presented in this paper to bound the condition number of the covariance matrix can be applied with any twice-differentiable kernel, where this restriction is needed to form the gradient-enhanced kernel matrix, as will be seen in Section 3.3. We start by considering the Gaussian kernel[6], with the Matérn $\frac{5}{2}$ and rational quadratic kernels presented in Section 6.3:

$$k(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\gamma}) = e^{-\frac{1}{2}\sum_{i=1}^{d} \gamma_i^2 (x_i - y_i)^2}, \tag{3}$$

where $\boldsymbol{\gamma}$ are hyperparameters and $\gamma_i > 0 \,\forall\, i \in \{1, \ldots, d\}$, which will be indicated hereafter as $\boldsymbol{\gamma} > 0$ for conciseness. The Gaussian kernel is often presented with $\boldsymbol{\theta} = \boldsymbol{\gamma}^2/2$ or $\boldsymbol{\theta} = \boldsymbol{\gamma}^{-1}$ as its hyperparameters[6], but it is simpler in the later derivations to use $\boldsymbol{\gamma}$ instead. The Gaussian kernel is a stationary kernel since it depends only on $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{y}$, i.e. the relative location of $\boldsymbol{y}$ to $\boldsymbol{x}$.

The gradient-free Gaussian kernel matrix is

$$\mathsf{K} = \mathsf{K}(X;\gamma) = \begin{bmatrix} 1 & k(\boldsymbol{x}_{1:},\boldsymbol{x}_{2:};\gamma) & \dots & k(\boldsymbol{x}_{1:},\boldsymbol{x}_{n_x:};\gamma) \\ k(\boldsymbol{x}_{2:},\boldsymbol{x}_{1:};\gamma) & 1 & \dots & k(\boldsymbol{x}_{2:},\boldsymbol{x}_{n_x:};\gamma) \\ \vdots & \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_{n_x:},\boldsymbol{x}_{1:};\gamma) & k(\boldsymbol{x}_{n_x:},\boldsymbol{x}_{2:};\gamma) & \dots & 1 \end{bmatrix}, \tag{4}$$

where its diagonal entries are all unity and n_x represents the number of function evaluations. In general, the i-th diagonal entry of K is $k(\boldsymbol{x}_{i:},\boldsymbol{x}_{i:};\gamma)$. The gradient-free Gaussian kernel matrix K is positive semidefinite[6], and also a correlation matrix since it satisfies all of the properties of the following definition[33]. The two properties in Definition 1 ensure that all of the entries in a correlation matrix are between –1 and 1. From the Cauchy-Schwarz inequality for a symmetric positive definite matrix A we have $\left|\langle \boldsymbol{u},\boldsymbol{v}\rangle_{\mathsf{A}}\right|^2 \leq \langle \boldsymbol{u},\boldsymbol{u}\rangle_{\mathsf{A}}\langle \boldsymbol{v},\boldsymbol{v}\rangle_{\mathsf{A}}$, where $\langle \boldsymbol{u},\boldsymbol{v}\rangle_{\mathsf{A}} = \boldsymbol{u}^\top \mathsf{A}\boldsymbol{v}$. If A is a correlation matrix then all of its diagonal entries are unity, and if $\boldsymbol{u}$ and $\boldsymbol{v}$ are selected as the i-th and j -th columns of the identity matrix, respectively, then we have $\left|a_{ij}\right|^2 \leq a_{ii}\cdot a_{jj} = 1$.

Definition 1. A correlation matrix must satisfy the following conditions:

1. The diagonal entries of the matrix are all unity
2. The matrix is symmetric positive semidefinite

The gradient-free covariance matrix is commonly given by

$$\Sigma(\mathsf{X};\hat{\sigma}_{\mathsf{K}},\gamma,\hat{\sigma}_0) = \hat{\sigma}_{\mathsf{K}}^2\mathsf{K}(\mathsf{X};\gamma) + \hat{\sigma}_0^2\mathsf{I}, \tag{5}$$

where the hyperparameter $\hat{\sigma}_{\mathsf{K}}^2$ is the variance of the stationary residual error and $\hat{\sigma}_0^2$ is a hyperparameter that estimates σ_0^2, which is the true noise variance[34]. The hyperparameter $\hat{\sigma}_0^2$ is used when the function evaluations are noisy and, in practice, it also serves to regularize Σ in order to reduce its condition number[34]. To separate the need to regularize the covariance matrix from the estimation of the uncertainty of the function evaluations, we use the following notation

$$\Sigma(\mathsf{X};\hat{\sigma}_{\mathsf{K}},\gamma,\eta_{\mathsf{K}},\hat{\sigma}_f) = \hat{\sigma}_{\mathsf{K}}^2\left(\mathsf{K}(\mathsf{X};\gamma) + \eta_{\mathsf{K}}\mathsf{I}\right) + \mathsf{V}(\hat{\sigma}_f), \tag{6}$$

where the nugget $\eta_{\mathsf{K}} \geq 0$ is used to regularize Σ and the estimated variance of the uncertainty for the function evaluations is provided by

$$\mathsf{V}(\hat{\sigma}_f) = \hat{\sigma}_f^2\,\mathsf{I}_{n_x}, \tag{7}$$

where $\hat{\sigma}_f \geq 0$ is a hyperparameter that approximates the noise variance that exceeds what is provided by the nugget η_{K}. If the noise variance is known, then it can be used instead of selecting $\hat{\sigma}_f$ as a hyperparameter. The estimated noise variance $\hat{\sigma}_0^2$ from Eq. (5) can be related to the terms in Eq. (6) with

$$\hat{\sigma}_0^2 = \hat{\sigma}_f^2 + \hat{\sigma}_{\mathsf{K}}^2\eta_{\mathsf{K}}. \tag{8}$$

The nugget η_{K}, which is discussed in detail in Section 4.1, is used to ensure that $\kappa(\Sigma) \leq \kappa_{\max}$, where $\kappa(\cdot)$ is the condition number based on the ℓ_2 norm and $\kappa_{\max} > 1$ is the maximum allowed condition number, which is set by the user. The second term on the right-hand side of Eq. (8) is generally small, as will be demonstrated in Section 6, where the nugget η_{K} is on the order of 10^{-9} for $\kappa_{\max} = 10^{10}$. The numerical benefit of having a positive nugget even when there is no uncertainty in the function evaluations is also highlighted in Section 6.

3.3 | Gradient-enhanced covariance matrix

Constructing the gradient-enhanced kernel matrix requires the derivatives of the kernel with respect to its inputs:

$$\frac{\partial k(\boldsymbol{x},\boldsymbol{y})}{\partial x_i} = -\gamma_i^2\left(x_i - y_i\right)k(\boldsymbol{x},\boldsymbol{y}) \tag{9}$$

$$\frac{\partial k(\boldsymbol{x},\boldsymbol{y})}{\partial y_j} = \gamma_j^2\left(x_j - y_j\right)k(\boldsymbol{x},\boldsymbol{y}) \tag{10}$$

$$\frac{\partial^2 k(\boldsymbol{x},\boldsymbol{y})}{\partial x_i \partial y_j} = \left(\delta_{ij}\gamma_i^2 - \gamma_i^2\gamma_j^2\left(x_i - y_i\right)\left(x_j - y_j\right)\right)k(\boldsymbol{x},\boldsymbol{y}), \tag{11}$$

where δ_{ij} is the Kronecker delta. The gradient-enhanced kernel matrix is given by

$$\mathsf{K}_\nabla(\mathsf{X};\boldsymbol{\gamma}) = \begin{bmatrix} \mathsf{K} & \frac{\partial \mathsf{K}}{\partial y_1} & \cdots & \frac{\partial \mathsf{K}}{\partial y_d} \\ \frac{\partial \mathsf{K}}{\partial x_1} & \frac{\partial^2 \mathsf{K}}{\partial x_1 \partial y_1} & \cdots & \frac{\partial^2 \mathsf{K}}{\partial x_1 \partial y_d} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathsf{K}}{\partial x_d} & \frac{\partial^2 \mathsf{K}}{\partial x_d \partial y_1} & \cdots & \frac{\partial^2 \mathsf{K}}{\partial x_d \partial y_d} \end{bmatrix} \tag{12}$$

$$= \begin{bmatrix} \mathsf{K} & \gamma_1^2 \mathsf{R}_1 \odot \mathsf{K} & \cdots & \gamma_d^2 \mathsf{R}_d \odot \mathsf{K} \\ -\gamma_1^2 \mathsf{R}_1 \odot \mathsf{K} & \left(\gamma_1^2 \mathbb{1} - \gamma_1^4 \mathsf{R}_1^{(2)}\right) \odot \mathsf{K} & \cdots & -\gamma_1^2 \gamma_d^2 \mathsf{R}_1 \odot \mathsf{R}_d \odot \mathsf{K} \\ \vdots & \vdots & \ddots & \vdots \\ -\gamma_d^2 \mathsf{R}_d \odot \mathsf{K} & -\gamma_1^2 \gamma_d^2 \mathsf{R}_1 \odot \mathsf{R}_d \odot \mathsf{K} & \cdots & \left(\gamma_d^2 \mathbb{1} - \gamma_d^4 \mathsf{R}_d^{(2)}\right) \odot \mathsf{K} \end{bmatrix}, \tag{13}$$

where $\mathbb{1}$ is a matrix of ones of size $n_x \times n_x$, the operator $\odot$ is the Hadamard product for elementwise multiplication, and R_i is a skew-symmetric matrix given by

$$\mathsf{R}_i(\mathsf{X}) = \boldsymbol{x}_{:i}\mathbf{1}_{n_x}^\top - \mathbf{1}_{n_x}\boldsymbol{x}_{:i}^\top = \begin{bmatrix} 0 & x_{1i} - x_{2i} & \cdots & x_{1i} - x_{n_x i} \\ x_{2i} - x_{1i} & 0 & \cdots & x_{2i} - x_{n_x i} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n_x i} - x_{1i} & x_{n_x i} - x_{2i} & \cdots & 0 \end{bmatrix}. \tag{14}$$

Just like K, K_∇ is also symmetric positive semidefinite [15]. However, unlike K, K_∇ is not a correlation matrix since it does not satisfy the first condition in Definition 1. This is clear from checking the diagonal of K_∇:

$$\operatorname{diag}(\mathsf{K}_\nabla) = [\underbrace{1, \ldots, 1}_{n_x}, \underbrace{\gamma_1^2, \ldots \gamma_1^2}_{n_x}, \ldots, \underbrace{\gamma_d^2, \ldots \gamma_d^2}_{n_x}]. \tag{15}$$

The first condition of Definition 1 would only be satisfied for K_∇ if $\gamma_1 = \ldots = \gamma_d = 1$. However, the hyperparameters $\boldsymbol{\gamma}$ are set by maximizing the marginal log-likelihood, which is introduced in the following subsection. The gradient-enhanced covariance matrix is given in the same format as Eq. (6) by

$$\Sigma_\nabla(\mathsf{X}; \hat{\sigma}_\mathsf{K}, \boldsymbol{\gamma}, \eta_{\mathsf{K}_\nabla}, \mathsf{W}, \hat{\sigma}_f, \hat{\sigma}_{\nabla f}) = \hat{\sigma}_\mathsf{K}^2 \left(\mathsf{K}_\nabla + \eta_{\mathsf{K}_\nabla} \mathsf{W}\right) + \mathsf{V}_\nabla, \tag{16}$$

where $\eta_{\mathsf{K}_\nabla} \geq 0$ is a nugget, W is a nonnegative diagonal matrix that is traditionally set to the identity matrix, and V_∇ is given by

$$\mathsf{V}_\nabla(\hat{\sigma}_f, \hat{\sigma}_{\nabla f}) = \operatorname{diag}\left(\hat{\sigma}_f^2 \mathbf{1}_{n_x}^\top, \hat{\sigma}_{\nabla f}^2 \mathbf{1}_{n_x d}^\top\right), \tag{17}$$

where $\hat{\sigma}_{\nabla f}$ is a hyperparameter that estimates $\sigma_{\nabla f}$.

3.4 | Evaluating the Gaussian process's posterior

The mean and variance of the posterior of the gradient-enhanced GP are evaluated with [18]

$$\mu_{\mathrm{GP}}(\boldsymbol{x}) = \beta + \hat{\sigma}_\mathsf{K}^2 \boldsymbol{k}_\nabla^\top(\boldsymbol{x}) \Sigma_\nabla^{-1} \left(\boldsymbol{f}_\nabla - \beta \check{\mathbf{1}}\right) \tag{18}$$

$$\sigma_{\mathrm{GP}}^2(\boldsymbol{x}) = \hat{\sigma}_\mathsf{K}^2 \left(k(\boldsymbol{x}, \boldsymbol{x}) - \hat{\sigma}_\mathsf{K}^2 \boldsymbol{k}_\nabla^\top(\boldsymbol{x}) \Sigma_\nabla^{-1} \boldsymbol{k}_\nabla(\boldsymbol{x})\right), \tag{19}$$

where $\check{\mathbf{1}} = [\mathbf{1}_{n_x}^\top, \mathbf{0}_{n_x d}^\top]^\top$, and

$$\boldsymbol{k}_\nabla(\boldsymbol{x};\mathsf{X}) = \begin{bmatrix} \boldsymbol{k}(\mathsf{X},\boldsymbol{x}) \\ \frac{\partial \boldsymbol{k}(\mathsf{X},\boldsymbol{x})}{\partial x_1} \\ \vdots \\ \frac{\partial \boldsymbol{k}(\mathsf{X},\boldsymbol{x})}{\partial x_d} \end{bmatrix}, \quad \boldsymbol{f}_\nabla(\mathsf{X}) = \begin{bmatrix} \boldsymbol{f}(\mathsf{X}) \\ \frac{\partial \boldsymbol{f}(\mathsf{X})}{\partial x_1} \\ \vdots \\ \frac{\partial \boldsymbol{f}(\mathsf{X})}{\partial x_d} \end{bmatrix}, \tag{20}$$

where $\boldsymbol{f}(\mathsf{X})$ is the function of interest evaluated at all of the rows in X. In this paper the gradient of the function of interest is calculated analytically, but it could also be calculated with algorithmic differentiation or approximated with finite differences.

The hyperparameters of the GP are set by maximizing the marginal likelihood[9,10,11,12]

$$L(\boldsymbol{\gamma}, \beta, \hat{\sigma}_\mathsf{K}^2, \hat{\sigma}_f, \hat{\sigma}_{\nabla f}; \mathsf{X}, \boldsymbol{f}_\nabla, \eta_{\mathsf{K}_\nabla}) = \frac{e^{-\frac{(\boldsymbol{f}_\nabla - \beta \check{\mathbf{1}})^\top \Sigma_\nabla^{-1} (\boldsymbol{f}_\nabla - \beta \check{\mathbf{1}})}{2}}}{(2\pi)^{\frac{n_x(d+1)}{2}} \sqrt{\det(\Sigma_\nabla)}}. \tag{21}$$

It is straightforward to get a closed-form solution for β that maximizes the marginal likelihood[9]:

$$\beta(\boldsymbol{\gamma}, \hat{\sigma}_f, \hat{\sigma}_{\nabla f}; \mathsf{X}, \boldsymbol{f}_\nabla, \eta_{\mathsf{K}_\nabla}) = \frac{\check{\mathbf{1}}^\top \Sigma_\nabla^{-1} \boldsymbol{f}_\nabla}{\check{\mathbf{1}}^\top \Sigma_\nabla^{-1} \check{\mathbf{1}}}. \tag{22}$$

For the noise-free case, i.e. $\sigma_f = \sigma_{\nabla f} = 0$, we have $\mathsf{V}_\nabla = \mathbb{O}$, and we can derive the following closed form solution for $\hat{\sigma}_\mathsf{K}^2$ that maximizes the marginal likelihood[9]:

$$\hat{\sigma}_\mathsf{K}^2(\boldsymbol{\gamma}; \mathsf{X}, \boldsymbol{f}_\nabla, \eta_{\mathsf{K}_\nabla}, \beta) = \frac{\left(\boldsymbol{f}_\nabla - \beta \check{\mathbf{1}}\right)^\top \left(\mathsf{K}_\nabla + \eta_{\mathsf{K}_\nabla} \mathsf{I}\right)^{-1} \left(\boldsymbol{f}_\nabla - \beta \check{\mathbf{1}}\right)}{n_x(d+1)}. \tag{23}$$

Substituting this solution for $\hat{\sigma}_\mathsf{K}^2$ into $\ln(L)$ and dropping the constant terms gives

$$\ln(L(\boldsymbol{\gamma}; \mathsf{X}, \eta_{\mathsf{K}_\nabla}, \hat{\sigma}_\mathsf{K})) = -\frac{n_x(d+1)\ln(\hat{\sigma}_\mathsf{K}^2) + \ln(\det(\mathsf{K}_\nabla + \eta_{\mathsf{K}_\nabla} \mathsf{I}))}{2}. \tag{24}$$

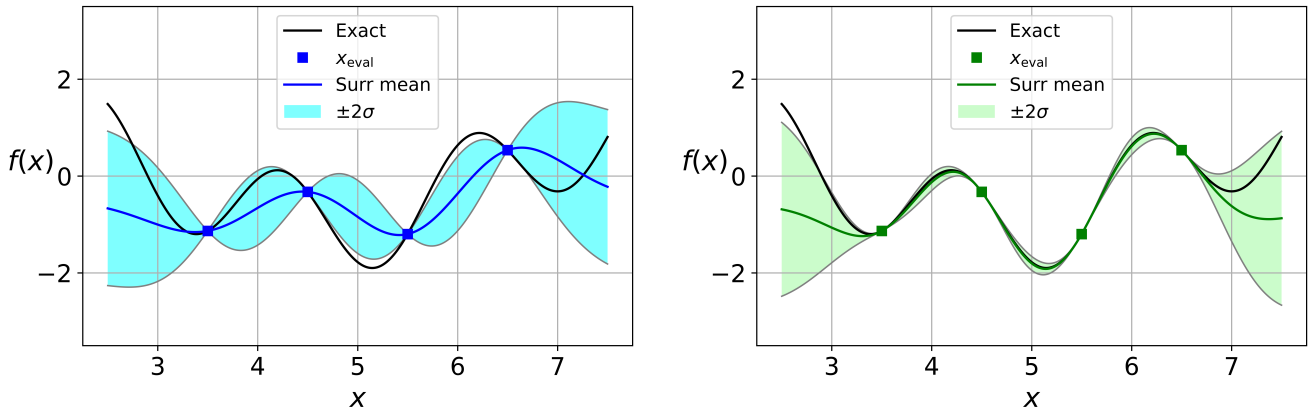
For the noise-free case the hyperparameters in the vector $\boldsymbol{\gamma}$ are selected by maximizing Eq. (24) numerically with the bound $\boldsymbol{\gamma} > 0$. When $\sigma_f > 0$ or $\sigma_{\nabla f} > 0$, it is not possible to get a closed-form solution for $\hat{\sigma}_\mathsf{K}^2$ and it must thus be optimized numerically along with $\hat{\sigma}_f$, $\hat{\sigma}_{\nabla f}$, and $\boldsymbol{\gamma}$. In contrast, the nugget η_{K_∇} is not a hyperparameter that is set by maximizing the marginal log-likelihood, as is discussed further in Section 4.1.

To demonstrate the benefit of using gradients we compare a gradient-free and a gradient-enhanced GP that approximate the following one-dimensional noise-free function:

$$f(x) = \sin(x) + \sin\left(\frac{10x}{3}\right), \tag{25}$$

which was evaluated at $\boldsymbol{x} = [3.5, 4.5, 5.5, 6.5]^\top$. The posterior of the gradient-free and gradient-enhanced GPs can be seen in Figs. 1a and 1b, respectively. For Fig. 1b the black line is Eq. (25), the solid green line is the mean of the surrogate from Eq. (18), and the light green area represents $\pm 2\sigma_{\text{GP}}(x)$, where σ_{GP} comes from Eq. (19). For Fig. 1a the mean and standard deviation of the posterior for the gradient-free GP are calculated with equations analogous to Eqs. (18) and (19) that omit the gradient evaluations and use the gradient-free kernel matrix. The hyperparameters $\beta = -0.62$, $\hat{\sigma}_\mathsf{K}^2 = 1.07$, and $\gamma = 1.7$ come from maximizing Eq. (24) numerically for the gradient-enhanced GP. The gradient-free GP uses the same hyperparameters since it was found that selecting its hyperparameters by maximizing the marginal log-likelihood with only the four evaluation points resulted in a significantly less accurate surrogate. It was found that at least 10 evaluation points were needed such that the hyperparameters for the gradient-free GP set by maximizing the marginal log-likelihood provided an accurate surrogate.

It is clear from Fig. 1 that the use of gradients to construct the GP significantly improves the accuracy of its posterior and also reduces its uncertainty, i.e. σ_{GP}. The benefit of using gradients to construct a GP is even greater for higher-dimensional parameter spaces since the gradient provides more information as the number of parameters increases. However, a significant problem for gradient-enhanced GPs is that their covariance matrices become extremely ill-conditioned, which is addressed in the two following sections.



(a) Gradient-free GP **(b)** Gradient-enhanced GP

FIGURE 1 The posterior of gradient-free and gradient-enhanced GPs approximating the function from Eq. (25) with the same hyperparameters $\beta = -0.62$, $\hat{\sigma}_{\mathsf{K}}^2 = 1.07$, and $\gamma = 1.7$.

4 | PREVIOUS METHODS TO MITIGATE ILL-CONDITIONING

4.1 | Regularization

A common approach to alleviate the ill-conditioning of a matrix is to regularize it, i.e. to add a positive nugget to its diagonal. For a GP, the addition of a nugget to the covariance matrix is analogous to having noisy data[6,34], as can be seen from Eqs. (6) and (16) for the gradient-free and gradient-enhanced cases, respectively. When the nugget is zero, the mean of the posterior for the GP will match the function of interest exactly at all points where it has been evaluated. The same applies to the evaluated gradients if a gradient-enhanced covariance matrix is used. However, if a positive nugget is used, the surrogate will generally not match the function of interest exactly at points where it has been sampled. It is therefore desirable to use the smallest nugget value required to ensure that the condition number of the covariance matrix is below a desired threshold.

For the gradient-free covariance matrix the addition of noise, i.e. $\sigma_f > 0$, helps reduce its condition number since σ_f, just like η_{K_∇}, is added to the entire diagonal of Σ from Eq. (6)[35]. For the gradient-enhanced case however, the addition of $\sigma_f > 0$, $\sigma_{\nabla f} > 0$, or both, may decrease or increase the condition number of Σ_∇. This is the result of σ_f and $\sigma_{\nabla f}$ only being added to part of the diagonal for Σ_∇, as seen in Eq. (16). In this subsection we consider the addition of a nugget to a noise-free covariance matrix, i.e. $\sigma_f = \sigma_{\nabla f} = 0$. Methods to bound the condition number of covariance matrices with noisy data will be considered in Section 5.

For the noise-free case the covariance matrices for the gradient-free and gradient-enhanced cases simply to $\Sigma = \hat{\sigma}_{\mathsf{K}}^2 (\mathsf{K} + \eta_{\mathsf{K}} \mathsf{I})$ and $\Sigma_\nabla = \hat{\sigma}_{\mathsf{K}}^2 \left(\mathsf{K}_\nabla + \eta_{\mathsf{K}_\nabla} \mathsf{I}\right)$, respectively. We omit the non-zero scalar $\hat{\sigma}_{\mathsf{K}}$ from the analysis since it does not affect the condition number. The eigenvalues of K and K_∇ are real since these are symmetric matrices. We derive the minimum nonnegative nugget value sufficient to have $\kappa(\mathsf{A} + \eta_{\min}\mathsf{I}) \leq \kappa_{\max}$, where A is an arbitrary symmetric semidefinite matrix, and the condition number is based on the ℓ_2 norm:

$$\begin{aligned} \kappa(\mathsf{A} + \eta_{\min}\mathsf{I}) &= \frac{\lambda_{\max} + \eta_{\min}}{\lambda_{\min} + \eta_{\min}} \leq \kappa_{\max} \\ \eta_{\min} &= \max\left(\frac{\lambda_{\max} - \lambda_{\min}\kappa_{\max}}{\kappa_{\max} - 1}, 0\right), \end{aligned} \tag{26}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the smallest and largest eigenvalues of A, respectively. For positive semidefinite matrices, such as K and K_∇, we have $\lambda_{\min} \geq 0$ and $\lambda_{\max}(\mathsf{K}) \leq \operatorname{tr}(\mathsf{K})$. From Eq. (26) it thus follows that nugget values sufficient to bound the condition numbers of the kernel matrices below $\kappa_{\max}$ are

$$\eta_{\mathsf{K}}(n_x; \kappa_{\max}) = \frac{n_x}{\kappa_{\max} - 1} \tag{27}$$

$$\eta_{\mathsf{K}_\nabla}(n_x, \gamma; \kappa_{\max}) = \frac{n_x(\mathbf{1}^\top \gamma^2 + 1)}{\kappa_{\max} - 1}. \tag{28}$$

These are sufficient but not necessary conditions to ensure that the condition numbers of the covariance matrices are smaller than $\kappa_{\max}$ since the bound $\lambda_{\max} \leq \mathrm{tr}(\mathsf{K})$ is not tight and Eqs. (27) and (28) were derived with the worst case $\lambda_{\min} = 0$.

Eq. (27) provides a small η_{K} that is sufficient to ensure that $\kappa(\Sigma) \leq \kappa_{\max}$. However, η_{K_∇} from Eq. (28) is undesirable since it depends on $\boldsymbol{\gamma}$. Consequently, as $\boldsymbol{\gamma}$ gets larger, η_{K_∇} must also increase to ensure that $\kappa(\Sigma_\nabla(\boldsymbol{\gamma})) \leq \kappa_{\max}$. In Sections 4.2 and 4.3 two methods are presented to ensure that $\kappa(\Sigma_\nabla(\boldsymbol{\gamma})) \leq \kappa_{\max}$ with a finite nugget value.

4.2 | Baseline method: constrained optimization of γ

An approach that has been used previously to ensure that $\kappa(\Sigma_\nabla) \leq \kappa_{\max}$ is to add a constraint to the maximization of the marginal log-likelihood from Eq. (24)[36,12]. The hyperparameters $\boldsymbol{\gamma}$ are thus selected by solving the following constrained optimization problem:

$$\boldsymbol{\gamma}^* = \underset{\boldsymbol{\gamma}>0}{\mathrm{argmax}}\, L(\boldsymbol{\gamma}) \quad \text{s.t.} \quad \kappa(\Sigma_\nabla(\boldsymbol{\gamma})) \leq \kappa_{\max}. \tag{29}$$

There will always be a feasible solution to Eq. (29) if $\eta_{\mathsf{K}_\nabla} \geq \frac{n_x}{\kappa_{\max}-1}$. This can be verified from Eq. (28) with $\boldsymbol{\gamma} \to \mathbf{0}_d$. Solving Eq. (29) to set the hyperparameters thus ensures that $\kappa(\Sigma_\nabla(\mathsf{X}, \boldsymbol{\gamma})) \leq \kappa_{\max} \forall\, \mathsf{X} \in \mathbb{R}^{n_x \times d}$. However, the constraint in Eq. (29) may result in selecting hyperparameters that provide a significantly lower marginal log-likelihood. This impacts the accuracy of the surrogate and is shown in Section 6.5 to be detrimental to the optimization case.

4.3 | Rescaling method

A short overview of the rescaling method from Marchildon and Zingg[32] is provided in this section. To ensure that $\kappa(\Sigma_\nabla(\mathsf{X}; \boldsymbol{\gamma}, \eta_{\mathsf{K}_\nabla})) \leq \kappa_{\max}$ when $\gamma_1 = \ldots = \gamma_d$ and Σ_∇ is not diagonally dominant, the parameter space is rescaled such that the minimum Euclidean distance between evaluation points is $v_{\min,\mathrm{set}}$, where

$$v_{\min,\mathrm{set}}(d, n_x) = \min\left(2\sqrt{d}, \frac{2 + \sqrt{4 + 2e^2 \ln\left(\frac{(n_x-1)(1+2\sqrt{d})}{2}\right)}}{e}\right). \tag{30}$$

The condition that Σ_∇ is not diagonally dominant is required since the condition number of Σ_∇ is unbounded as $\boldsymbol{\gamma}$ tends to infinity, regardless of the selected nugget, as seen in Eq. (15). However, since the condition on the diagonal dominance of Σ_∇ only applies for large values of $\boldsymbol{\gamma}$, it is not in practice limiting since there is little correlation between evaluation points and thus the marginal log-likelihood is unlikely to be maximized at these hyperparameter values[32].

To ensure that the minimum Euclidean distance between evaluation points is $v_{\min,\mathrm{set}}$ from Eq. (30), an isotropic scaling is performed

$$\begin{aligned} \mathsf{X} &= \tau \mathsf{X}_{\mathrm{initial}} \\ \frac{\partial f(\boldsymbol{x})}{\partial x_i} &= \frac{1}{\tau}\left(\frac{\partial f(\boldsymbol{x})}{\partial x_i}\right)_{\mathrm{initial}}, \end{aligned}$$

where $\tau = \frac{v_{\min,\mathrm{set}}}{v_{\min,\mathrm{initial}}}$ with $v_{\min,\mathrm{set}}$ coming from Eq. (30) and $v_{\min,\mathrm{initial}}$ being the initial minimum Euclidean distance between evaluation points prior to rescaling the data. The required nugget to bound the condition number of Σ_∇ with the rescaling method is

$$\eta_{\mathsf{K}_\nabla}(d, n_x; \kappa_{\max}) = \frac{1 + (n_x - 1)\frac{2\sqrt{d}}{v_{\min,\mathrm{set}}} e^{\frac{v_{\min,\mathrm{set}}}{2\sqrt{d}} - 1}}{\kappa_{\max} - 1}. \tag{31}$$

In order to have the optimized hyperparameters satisfy $\gamma_1 = \ldots = \gamma_d$, which is needed for this method to ensure that $\kappa(\Sigma_\nabla(\mathsf{X}; \boldsymbol{\gamma}, \eta_{\mathsf{K}_\nabla})) \leq \kappa_{\max}$, the data can be rescaled non-isotropically and iteratively[32]. Having to optimize the hyperparameters more than once increases the computational cost, but it ensures that the final optimized hyperparameters are unconstrained, unlike the method presented in Section 4.2. This enables a Bayesian optimizer using this rescaling method to achieve deeper convergence, i.e. to reduce the optimality several additional orders of magnitude, relative to the use of the baseline method from Section 4.2, as was shown in Marchildon and Zingg[32] and will be demonstrated in Section 6.5.

5 | PRECONDITIONING METHOD

5.1 | Preconditioned gradient-enhanced covariance matrix

In Section 3.3 it was indicated that the gradient-enhanced kernel matrix K_∇ is not a correlation matrix, unlike its gradient-free counterpart K. However, we can form a gradient-enhanced correlation matrix by preconditioning the unregularized, i.e. $\eta_{\mathsf{K}_\nabla} = 0$, gradient-enhanced covariance matrix Σ_∇ from Eq. (16) as follows:

$$\begin{aligned}\dot{\mathsf{K}}_\nabla(\mathsf{X}; \hat{\sigma}_\mathsf{K}, \boldsymbol{\gamma}, \hat{\sigma}_f, \hat{\sigma}_{\nabla f}) &= (\hat{\sigma}_\mathsf{K}\mathsf{P})^{-1}\, \Sigma_\nabla\, (\hat{\sigma}_\mathsf{K}\mathsf{P})^{-1} \\ &= \mathsf{P}^{-1}\left(\mathsf{K}_\nabla + \hat{\sigma}_\mathsf{K}^{-2}\mathsf{V}_\nabla\right)\mathsf{P}^{-1},\end{aligned} \tag{32}$$

where

$$\mathsf{P} = \operatorname{diag}\left(\sqrt{\operatorname{diag}\left(\hat{\sigma}_\mathsf{K}^{-2}\Sigma_\nabla\right)}\right) = \operatorname{diag}\left(\sqrt{\operatorname{diag}\left(\mathsf{K}_\nabla + \hat{\sigma}_\mathsf{K}^{-2}\mathsf{V}_\nabla\right)}\right). \tag{33}$$

For the noise-free case, i.e. $\hat{\sigma}_f = \hat{\sigma}_{\nabla f} = 0$, the preconditioning matrix for the gradient-enhanced Gaussian kernel from Eq. (3) simplifies to

$$\mathsf{P} = \operatorname{diag}(\underbrace{1, \ldots, 1}_{n_x}, \underbrace{\gamma_1, \ldots, \gamma_1}_{n_x}, \ldots, \underbrace{\gamma_d, \ldots, \gamma_d}_{n_x}). \tag{34}$$

More information on $\dot{\mathsf{K}}_\nabla$, which is a correlation matrix, is provided in Section 5.2. With the preconditioned gradient-enhanced kernel matrix we can form the preconditioned gradient-enhanced covariance matrix

$$\dot{\Sigma}_\nabla(\mathsf{X}; \hat{\sigma}_\mathsf{K}, \boldsymbol{\gamma}, \hat{\sigma}_f, \hat{\sigma}_{\nabla f}, \eta_{\dot{\mathsf{K}}_\nabla}) = \hat{\sigma}_\mathsf{K}^2\left(\dot{\mathsf{K}}_\nabla + \eta_{\dot{\mathsf{K}}_\nabla}\mathsf{I}\right). \tag{35}$$

In the following subsections different methods of regularizing $\dot{\Sigma}_\nabla$ to ensure that $\kappa(\dot{\Sigma}_\nabla(\boldsymbol{\gamma})) \leq \kappa_{\max}\ \forall\, \boldsymbol{\gamma} > 0$ are introduced. In Section 5.3 a constant nugget that scales as $\mathcal{O}(n_x d)$ is presented and compared to the secondary method from Dalbey [15], which uses the same preconditioning matrix P but only considers the noise-free case. In Section 5.4, a smaller constant nugget that scales as $\mathcal{O}\left(n_x\sqrt{d}\right)$ is derived but it only ensures that $\kappa(\dot{\Sigma}_\nabla(\boldsymbol{\gamma})) \leq \kappa_{\max}\ \forall\, \boldsymbol{\gamma} > 0$ if the Gaussian kernel is used. Finally, in Section 5.5 an even smaller nugget is presented that depends on $\boldsymbol{\gamma}$ and ensures that $\kappa(\dot{\Sigma}_\nabla(\boldsymbol{\gamma})) \leq \kappa_{\max}\ \forall\, \boldsymbol{\gamma} > 0$. These methods provide several advantages relative to the regularization approach from Dalbey [15], as detailed in the following subsections.

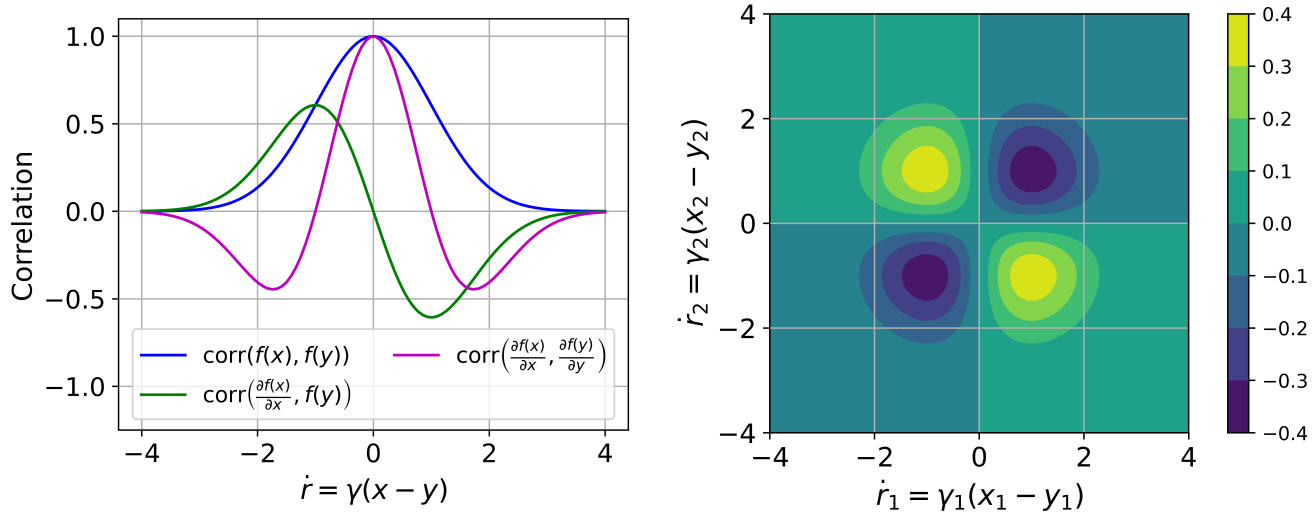
5.2 | Correlation matrix $\dot{\mathsf{K}}_\nabla$

The matrix $\dot{\mathsf{K}}_\nabla$ is a gradient-enhanced correlation matrix since it is formed by normalizing the gradient-enhanced covariance matrix Σ_∇ by its diagonal entries. For the noise-free case $\dot{\mathsf{K}}_\nabla$ from Eq. (32) can be calculated with

$$\begin{aligned}\dot{\mathsf{K}}_\nabla(\mathsf{X}; \boldsymbol{\gamma}) &= \mathsf{P}^{-1}\mathsf{K}_\nabla\mathsf{P}^{-1} \\ &= \begin{bmatrix} \mathsf{K} & \frac{\partial \mathsf{K}}{\partial \dot{y}_1} & \cdots & \frac{\partial \mathsf{K}}{\partial \dot{y}_d} \\ \frac{\partial \mathsf{K}}{\partial \dot{x}_1} & \frac{\partial^2 \mathsf{K}}{\partial \dot{x}_1 \partial \dot{y}_1} & \cdots & \frac{\partial^2 \mathsf{K}}{\partial \dot{x}_1 \partial \dot{y}_d} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathsf{K}}{\partial \dot{x}_d} & \frac{\partial^2 \mathsf{K}}{\partial \dot{x}_d \partial \dot{y}_1} & \cdots & \frac{\partial^2 \mathsf{K}}{\partial \dot{x}_d \partial \dot{y}_d} \end{bmatrix} \\ &= \begin{bmatrix} \mathsf{K} & \dot{\mathsf{R}}_1 \odot \mathsf{K} & \cdots & \dot{\mathsf{R}}_d \odot \mathsf{K} \\ -\dot{\mathsf{R}}_1 \odot \mathsf{K} & \left(\mathbb{1} - \dot{\mathsf{R}}_1 \odot \dot{\mathsf{R}}_1\right) \odot \mathsf{K} & \cdots & -\dot{\mathsf{R}}_1 \odot \dot{\mathsf{R}}_d \odot \mathsf{K} \\ \vdots & \vdots & \ddots & \vdots \\ -\dot{\mathsf{R}}_d \odot \mathsf{K} & -\dot{\mathsf{R}}_1 \odot \dot{\mathsf{R}}_d \odot \mathsf{K} & \cdots & \left(\mathbb{1} - \dot{\mathsf{R}}_d \odot \dot{\mathsf{R}}_d\right) \odot \mathsf{K} \end{bmatrix},\end{aligned} \tag{36}$$

where $\dot{\mathsf{R}}_i = \gamma_i \mathsf{R}_i(\mathsf{X}) = \mathsf{R}_i(\dot{X})$, and $\dot{X} = \mathsf{X}\mathsf{P}$.

The correlations for the entries in $\boldsymbol{f}_\nabla$ can be seen in Fig. 2. Having $\dot{\mathsf{K}}_\nabla$ as a correlation matrix makes the GP easier to interpret. Values in $\dot{\mathsf{K}}_\nabla$ close to –1 or 1 indicate near perfect inverse or direct correlations, respectively. On the other hand, values close to



(a) Correlations in one dimension

(b) Plot of corr $\left(\frac{\partial f(\boldsymbol{x})}{\partial x_1}, \frac{\partial f(\boldsymbol{y})}{\partial y_2}\right)$ for $d = 2$

FIGURE 2 Correlations for the Gaussian kernel $k(\boldsymbol{x},\boldsymbol{y})$ from Eq. (3) for the evaluation of a function of interest $f(\boldsymbol{x})$, such as Eq. (25), and its gradient. The correlations do not depend directly on the evaluation of the function $f(\boldsymbol{x})$, but rather on where it is evaluated, i.e. $\boldsymbol{x}$. However, the evaluation of $f(\boldsymbol{x})$ and of its gradient impacts the selection of the hyperparameters $\boldsymbol{\gamma}$, which thus impacts the correlations. The correlations are $\text{corr}(f(\boldsymbol{x}),f(\boldsymbol{y})) = k(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\gamma})$, $\text{corr}\left(\frac{\partial f(\boldsymbol{x})}{\partial x_i},f(\boldsymbol{y})\right) = \frac{\partial k(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\gamma})}{\partial \dot{x}_i}$, and $\text{corr}\left(\frac{\partial f(\boldsymbol{x})}{\partial x_i}, \frac{\partial f(\boldsymbol{y})}{\partial y_j}\right) = \frac{\partial^2 k(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\gamma})}{\partial \dot{x}_i \partial \dot{y}_j}$ for $i,j \in \{1,\dots,d\}$.

–1 and 1 in K_∇ indicate negative and positive relations, respectively, but provide little insight on the strength of the relations between the evaluation points.

5.3 | Regularizing $\dot{\mathsf{K}}_\nabla$ with $\eta_{\dot{\mathsf{K}}_\nabla} = \mathcal{O}(n_x d)$

To derive a nugget value that is sufficient to ensure that $\kappa(\dot{\Sigma}_\nabla(\boldsymbol{\gamma})) \leq \kappa_{\max}\ \forall\, \boldsymbol{\gamma} > 0$ we can follow the same approach taken in Section 4.1 to get

$$\eta_{\dot{\mathsf{K}}_\nabla}(n_x, d; \kappa_{\max}) = \frac{n_x(d+1)}{\kappa_{\max} - 1}. \tag{37}$$

Unlike η_{K_∇} from Eq. (28), $\eta_{\dot{\mathsf{K}}_\nabla}$ does not depend on the hyperparameters $\boldsymbol{\gamma}$. However, the nugget that Eq. (37) provides is much larger than required and it scales as $\eta_{\dot{\mathsf{K}}_\nabla} = \mathcal{O}(n_x d)$. Dalbey[15] took a slightly different approach by using the relation $\sum_{i=1} \lambda_i = \text{tr}\left(\dot{\mathsf{K}}_\nabla\right) = n_x(d+1)$ and considered the worst-case eigenvalue distribution that maximizes the ℓ_2 condition number:

$$\lambda_{\max} + (n_x(d+1) - 1)\lambda_{\min} = n_x(d+1). \tag{38}$$

Eq. (38) along with the relation $\lambda_{\min} = \frac{\lambda_{\max}}{\kappa_2}$, where κ_2 is the condition number of $\dot{\mathsf{K}}_\nabla$ based on the ℓ_2 norm, were then used to get the following relation:

$$\lambda_{\max} = \frac{n_x(d+1)\kappa_2}{\kappa_2 + n_x(d+1) - 1}. \tag{39}$$

Eq. (39) can be substituted into Eq. (26) to get

$$\begin{aligned}\eta_{\dot{\mathsf{K}}_\nabla} &= \frac{\lambda_{\max}}{\kappa_{\max} - 1}\left(1 - \frac{\kappa_{\max}}{\kappa_2}\right) \\ &= \frac{n_x(d+1)\kappa_2}{(\kappa_{\max} - 1)(\kappa_2 + n_x(d+1) - 1)}\left(1 - \frac{\kappa_{\max}}{\kappa_2}\right),\end{aligned} \tag{40}$$

which is sufficient to ensure that $\kappa(\dot{\mathsf{K}}_\nabla(\boldsymbol{\gamma}) + \eta_{\dot{\mathsf{K}}_\nabla}\mathsf{I}) \leq \kappa_{\max}\ \forall\, \boldsymbol{\gamma} > 0$. However, calculating κ_2 is expensive and Dalbey[15] avoids this by approximating the ℓ_1 condition number of $\dot{\mathsf{K}}_\nabla$, i.e. κ_1, and using the following relations $\frac{\kappa_1}{\sqrt{n_x(d+1)}} \leq \kappa_2 \leq \sqrt{n_x(d+1)}\,\kappa_1$.

Approximating κ_1 reduces the computational cost from calculating κ_2 but it makes the bound for $\eta_{\dot{\mathsf{K}}_\nabla}$ looser. Furthermore, as the author acknowledged, this method does not guarantee that $\kappa(\dot{\mathsf{K}}_\nabla + \eta_{\dot{\mathsf{K}}_\nabla}\mathsf{I}) \leq \kappa_{\max}$ since it uses an approximation to κ_1 rather than its exact value. Finally, when a gradient-based optimizer is used to select the hyperparameter $\boldsymbol{\gamma}$, the gradient of $\eta_{\dot{\mathsf{K}}_\nabla}$ is needed but it cannot be calculated if κ_1 is approximated, or it is very expensive to calculate if κ_2 is used since this would require the gradients of $\lambda_{\min}$ and $\lambda_{\max}$.

We can compare the nugget values obtained from Eqs. (40) and (37) by dividing the former by the latter:

$$\frac{\eta_{\dot{\mathsf{K}}_\nabla}}{\eta_{\mathrm{tr}\left(\dot{\mathsf{K}}_\nabla\right)}} = \frac{\kappa_2}{\kappa_2 + n_x(d+1) - 1}\left(1 - \frac{\kappa_{\max}}{\kappa_2}\right). \tag{41}$$

The first term on the right-hand side is always approximately unity since $\kappa_2 \gg n_x(d+1) - 1$ and the second term is also approximately unity when $\kappa_2 \gg \kappa_{\max}$, which is when $\eta_{\dot{\mathsf{K}}_\nabla}$ is needed to regularize $\dot{\mathsf{K}}_\nabla$. Therefore, when $\kappa_2 \gg \kappa_{\max}$, Eq. (40) and Eq. (28) provide roughly the same nugget value that scales as $\mathcal{O}(n_x d)$. This will be demonstrated in Section 6.2 when the different nugget values are compared in more detail.

A constant $\eta_{\dot{\mathsf{K}}_\nabla}$ that scales with $\mathcal{O}(n_x\sqrt{d})$ instead of $\mathcal{O}(n_x d)$ from Eq. (37) is derived in Section 5.4 for the Gaussian kernel. In Section 5.5 a smaller nugget than the one derived in Section 5.4 is provided which ensures that $\kappa(\dot{\mathsf{K}}_\nabla(\boldsymbol{\gamma}) + \eta_{\dot{\mathsf{K}}_\nabla}\mathsf{I}) \leq \kappa_{\max}\ \forall\, \boldsymbol{\gamma} > 0$ for any twice differentiable kernels. Another advantage of the nugget values derived in the following subsections is that they do not require the condition number of the $\dot{\mathsf{K}}_\nabla$ to be approximated, which makes them simpler and computationally cheaper to use than the nugget from Eq. (40).

5.4 | Regularizing $\dot{\mathsf{K}}_\nabla$ for the Gaussian kernel with $\eta_{\dot{\mathsf{K}}_\nabla} = \mathcal{O}(n_x\sqrt{d})$

In this section we derive a nugget for the Gaussian kernel that scales as $\eta_{\dot{\mathsf{K}}_\nabla} = \mathcal{O}(n_x\sqrt{d})$ instead of as $\eta_{\dot{\mathsf{K}}_\nabla} = \mathcal{O}(n_x d)$ if Eq. (37) is used while still ensuring $\kappa(\dot{\Sigma}_\nabla(\boldsymbol{\gamma})) \leq \kappa_{\max}\ \forall\, \boldsymbol{\gamma} > 0$. The derivation uses the Gershgorin circle theorem, which bounds the largest eigenvalue of a symmetric matrix A by

$$\lambda_{\max}(\mathsf{A}) \leq \max_i \left(a_{ii} + \sum_{j \neq i} \left|a_{ij}\right|\right). \tag{42}$$

The two following propositions provide an upper bound on the sum of the absolute values of the off-diagonal entries of $\dot{\mathsf{K}}_\nabla$ when it is constructed with the Gaussian kernel from Eq. (3).

Proposition 1. *For $n_x, d \in \mathbb{Z}^+$ the sum of the absolute values for the off-diagonal entries for any of the first n_x rows of the gradient-enhanced Gaussian kernel matrix $\dot{\mathsf{K}}_\nabla$ from Eq. (32) is bounded by u_G, where*

$$u_G(n_x, d) = (n_x - 1)\frac{1 + \sqrt{1+4d}}{2} e^{-\frac{1+2d-\sqrt{1+4d}}{4d}}. \tag{43}$$

Proof. We derive an upper bound for the sum of the absolute values for the off-diagonal entries for any of the first n_x rows of $\dot{\mathsf{K}}_\nabla$. We consider the noise-free case since the magnitude of the off-diagonal entries of $\dot{\mathsf{K}}_\nabla$ is largest when $\hat{\sigma}_f = \hat{\sigma}_{\nabla f} = 0$. For $\hat{\sigma}_f \geq 0$ and $\hat{\sigma}_{\nabla f} \geq 0$ the diagonal of $\dot{\mathsf{K}}_\nabla$ remains unity. However, the off-diagonal entries are smaller since the entries of P^{-1}, which is used to form $\dot{\mathsf{K}}_\nabla$ with Eq. (32), are inversely related to $\hat{\sigma}_f$ and $\hat{\sigma}_{\nabla f}$. The derivation is the same for any of the rows of $\dot{\mathsf{K}}_\nabla$ from Eq. (36) and we thus consider the a-th row, where $1 \leq a \leq n_x$ and

$$\begin{aligned}
\sum_{\substack{i=1\\ i\neq a}}^{n_x} \left|\dot{\mathsf{K}}_\nabla\right|_{ai} &= \sum_{\substack{i=1\\ i\neq a}}^{n_x} \left(1 + \sum_{j=1}^{d} \left|\dot{x}_{aj} - \dot{x}_{ij}\right|\right) e^{-\frac{\|\dot{x}_{a:} - \dot{x}_{i:}\|_2^2}{2}} \\
&\leq (n_x - 1)\max_i \left(1 + \sum_{j=1}^{d} \left|\dot{x}_{aj} - \dot{x}_{ij}\right|\right) e^{-\frac{\|\dot{x}_{a:} - \dot{x}_{i:}\|_2^2}{2}} \\
&\leq (n_x - 1)\max_{\boldsymbol{w} \geq 0} \left(\left(1 + \boldsymbol{w}^\top \mathbf{1}_d\right) e^{-\frac{\boldsymbol{w}^\top \boldsymbol{w}}{2}}\right),
\end{aligned}$$

where $w_j = |\dot{x}_{aj} - \dot{x}_{ij}|$ and we denote the expression inside the max function as $g(\boldsymbol{w})$. To identify the maximum of $g(\boldsymbol{w})$ we calculate its derivative and set it to zero:

$$\frac{\partial g(\boldsymbol{w})}{\partial w_i} = \left(1 - w_i(1 + \boldsymbol{w}^\top \mathbf{1}_d)\right) e^{-\frac{\boldsymbol{w}^\top \boldsymbol{w}}{2}} = 0$$
$$w_i = \frac{1}{1 + \boldsymbol{w}^\top \mathbf{1}} \; \forall\, i \in \{1, \ldots, d\}.$$

It is clear that the gradient of $g(\boldsymbol{w})$ is zero if and only if all of the entries in $\boldsymbol{w}$ are equal. We thus use $\boldsymbol{w} = \alpha \mathbf{1}_d$ and solve for the value of α that maximizes $g(\alpha \mathbf{1}_d)$:

$$\frac{\partial g(\alpha \mathbf{1}_d)}{\partial \alpha} = d\,(1 - \alpha(1 + d\alpha))\, e^{-\frac{d\alpha^2}{2}} = 0$$
$$\alpha^* = \frac{-1 + \sqrt{1 + 4d}}{2d},$$

where we only kept the positive root since $\boldsymbol{w} \geq 0$ and it is straightforward to verify that this provides the maximum of $g(\boldsymbol{w})$. Eq. (43) is recovered by evaluating $g(\alpha^* \mathbf{1}_d)$, which completes the proof. □

Proposition 2. *The sum of the absolute values for the off-diagonal entries for any of the last $n_x d$ rows of the preconditioned noise-free gradient-enhanced Gaussian kernel matrix $\dot{\mathsf{K}}_\nabla$ is smaller than $u_G(n_x, d)$ from Eq. (43) for $n_x, d \in \mathbb{Z}^+$.*

Proof. The proof can be found in Section A.1. □

As a result of Propositions 1 and 2 and the Gershgorin circle theorem, we have $\lambda_{\max}(\dot{\mathsf{K}}_\nabla) \leq 1 + u_G(n_x, d)$, where $u_G(n_x, d)$ is calculated from Eq. (43). Therefore, we can ensure that $\kappa(\dot{\mathsf{K}}_\nabla(\boldsymbol{\gamma}) + \eta_{\dot{\mathsf{K}}_\nabla} \mathsf{I}) \leq \kappa_{\max} \; \forall\, \boldsymbol{\gamma} > 0$ for the cases with and without noisy data by using Eqs. (26) and (43) to obtain

$$\eta_{\dot{\mathsf{K}}_\nabla}(n_x, d; \kappa_{\max}) = \frac{1 + (n_x - 1)\frac{1+\sqrt{1+4d}}{2} e^{-\frac{1+2d-\sqrt{1+4d}}{4d}}}{\kappa_{\max} - 1}. \tag{44}$$

Larger nugget values than the one provided by Eq. (44) would also ensure that $\kappa(\dot{\mathsf{K}}_\nabla(\boldsymbol{\gamma}) + \eta_{\dot{\mathsf{K}}_\nabla} \mathsf{I}) \leq \kappa_{\max} \; \forall\, \boldsymbol{\gamma} > 0$ but we are interested in using the smallest sufficient value. From Eq. (44) we have $\eta_{\dot{\mathsf{K}}_\nabla} = \mathcal{O}(n_x \sqrt{d})$, which is proven in the following lemma.

Lemma 1. *From Eq. (44) we have $\eta_{\dot{\mathsf{K}}_\nabla}(n_x, d) = \mathcal{O}(n_x \sqrt{d})$ for $n_x, d \in \mathbb{Z}^+$.*

Proof. From Eq. (44) we have

$$\eta_{\dot{\mathsf{K}}_\nabla}(n_x, d; \kappa_{\max}) = \frac{1 + (n_x - 1)\frac{1+\sqrt{1+4d}}{2} \left(e^{-\frac{1}{2}} e^{\frac{\sqrt{1+4d}-1}{4d}}\right)}{\kappa_{\max} - 1},$$

where the exponent for the final term goes to zero as $d \to \infty$. The only other term containing d scales as $\sqrt{d}$, which completes the proof. □

From Lemma 1 it is clear that the use of Eq. (44) to calculate $\eta_{\dot{\mathsf{K}}_\nabla}$ is advantageous for high-dimensional problems since it provides $\eta_{\dot{\mathsf{K}}_\nabla} = \mathcal{O}(n_x \sqrt{d})$ instead of $\eta_{\dot{\mathsf{K}}_\nabla} = \mathcal{O}(n_x d)$ from Eq. (37).

5.5 | Regularization with a variable nugget

Deriving the value of $\eta_{\dot{\mathsf{K}}_\nabla}$ in Section 5.4 to ensure that $\kappa(\dot{\Sigma}_\nabla(\boldsymbol{\gamma})) \leq \kappa_{\max} \; \forall\, \boldsymbol{\gamma} > 0$ when the Gaussian kernel is used required an extensive proof. This same process would need to be repeated to derive a nugget to ensure that $\kappa(\dot{\Sigma}_\nabla(\boldsymbol{\gamma})) \leq \kappa_{\max} \; \forall\, \boldsymbol{\gamma} > 0$ if other kernels were used. In this section this problem is avoided by having a nugget that depends on the hyperparameter $\boldsymbol{\gamma}$, and in the case with noisy data it also depends on $\hat{\sigma}_f$, $\hat{\sigma}_{\nabla f}$, and $\hat{\sigma}_{\mathsf{K}}$. From Eq. (26) with $\lambda_{\min} = 0$ we can ensure that $\kappa(\dot{\Sigma}_\nabla(\boldsymbol{\gamma})) \leq \kappa_{\max} \; \forall\, \boldsymbol{\gamma} > 0$ with

$$\eta_{\dot{\mathsf{K}}_\nabla}(\boldsymbol{\gamma}, \hat{\sigma}_{\mathsf{K}}, \hat{\sigma}_f, \hat{\sigma}_{\nabla f}; \mathsf{X}, \kappa_{\max}) = \frac{\max_i \sum_{j=1}^{n_x(d+1)} \left|\dot{\mathsf{K}}_\nabla\right|_{ij}}{\kappa_{\max} - 1}, \tag{45}$$

where the numerator is an upper bound on the maximum eigenvalue of $\dot{\mathsf{K}}_\nabla$ that comes from the application of the Gershgorin circle theorem. We now compare the nugget from this section to the one from Section 5.4:

$$\eta_{\text{G circle, vary}}(\mathsf{X}) \triangleq \frac{\max_i \sum_{j=1}^{n_x(d+1)} \left|\dot{\mathsf{K}}_\nabla(\mathsf{X})\right|_{ij}}{\kappa_{\max} - 1} \leq \frac{\max_{\mathsf{X}} \max_i \sum_{j=1}^{n_x(d+1)} \left|\dot{\mathsf{K}}_\nabla(\mathsf{X})\right|_{ij}}{\kappa_{\max} - 1} \leq \eta_{\text{G circle, const}}(n_x, d),$$

where $\eta_{\text{G circle, vary}}$ comes from Eq. (45) while $\eta_{\text{G circle, const}}$ come from Eq. (44) if the Gaussian kernel is used. In Section 6.2 $\eta_{\text{G circle, vary}}$ and $\eta_{\text{G circle, const}}$ will be compared for a a given X.

The nugget $\eta_{\dot{\mathsf{K}}_\nabla}$ from Eq. (45) depends on the hyperparameters $\boldsymbol{\gamma}$ and α, and this must be taken into consideration when optimizing the hyperparameters with a gradient-based optimizer. The derivative of $\eta_{\dot{\mathsf{K}}_\nabla}$ from Eq. (45) with respect to γ_ℓ for $\ell \in \{1, \ldots, d\}$ is

$$\begin{aligned}\frac{\partial \eta_{\dot{\mathsf{K}}_\nabla}}{\partial \gamma_\ell} &= \frac{\sum_{j=1}^{n_x(d+1)} \operatorname{sgn}\left(\left(\dot{\mathsf{K}}_\nabla\right)_{i^*j}\right)\left(\frac{\partial \dot{\mathsf{K}}_\nabla}{\partial \gamma_\ell}\right)_{i^*j}}{\kappa_{\max} - 1} \\ &= \frac{\sum_{j=1}^{n_x(d+1)} \operatorname{sgn}\left(\left(\dot{\mathsf{K}}_\nabla\right)_{i^*j}\right)\left[\mathsf{P}^{-1}\frac{\partial \mathsf{K}_\nabla}{\partial \gamma_\ell}\mathsf{P}^{-1} - \dot{\mathsf{K}}_\nabla\frac{\partial \mathsf{P}}{\partial \gamma_\ell}\mathsf{P}^{-1} - \mathsf{P}^{-1}\frac{\partial \mathsf{P}}{\partial \gamma_\ell}\dot{\mathsf{K}}_\nabla\right]_{i^*j}}{\kappa_{\max} - 1},\end{aligned} \tag{46}$$

where $\operatorname{sgn}(\cdot)$ returns +1 for positive numbers, –1 for negative numbers, and zero otherwise, and

$$i^* = \underset{i}{\operatorname{argmax}} \sum_{j=1}^{n_x(d+1)} \left|\dot{\mathsf{K}}_\nabla\right|_{ij}. \tag{47}$$

The derivatives of $\eta_{\dot{\mathsf{K}}_\nabla}$ from Eq. (45) with respect to $\hat{\sigma}_{\mathsf{K}}^2$, $\hat{\sigma}_f^2$, and $\hat{\sigma}_{\nabla f}^2$ are analogous to Eq. (46). When performing gradient-based optimization of the hyperparameters, the matrices $\frac{\partial \mathsf{K}_\nabla}{\partial \gamma_\ell}\ \forall\, \ell \in \{1, \ldots, d\}$ are already required whether $\eta_{\dot{\mathsf{K}}_\nabla}$ depends on the hyperparameters or not. Furthermore, calculating $\frac{\partial \mathsf{P}}{\partial \gamma_\ell}$ is inexpensive since it is a diagonal matrix. Therefore, calculating Eq. (46) does not significantly increase the computational cost of the hyperparameter optimization.

This same method can also be applied to the gradient-free kernel matrix in order to get a nugget that is smaller than the one provided by Eq. (27) while ensuring that $\kappa(\Sigma(\boldsymbol{\gamma})) \leq \kappa_{\max}\ \forall\, \boldsymbol{\gamma} > 0$. For the Gaussian kernel and others that provided non-negative gradient-free correlations, the gradient calculation of η_{K} is simpler since it does not require the preconditioning matrix P nor the operator $\operatorname{sgn}(\cdot)$:

$$\frac{\partial \eta_{\mathsf{K}}}{\partial \gamma_\ell} = \frac{\partial \sum_{j=1}^{n_x} \left(\frac{\partial \mathsf{K}}{\partial \gamma_\ell}\right)_{i^*j}}{\partial \kappa_{\max} - 1}, \tag{48}$$

where i^* for η_{K} is analogous to the one for $\eta_{\dot{\mathsf{K}}_\nabla}$ from Eq. (47). Similar equations to Eqs. (45) and (46) could also be applied to preconditioned Hessian-enhanced covariance matrices.

5.6 | Implementation

Since the inverse of the symmetric positive definite matrix Σ_∇^{-1} is needed to calculate μ_{GP}, σ_{GP}^2, and $\ln(L)$ along with their gradients, it is desirable to calculate its Cholesky decomposition. Once the Cholesky decomposition has been calculated, it becomes inexpensive to evaluate $\mu_{\text{GP}}(\boldsymbol{x})$ and $\sigma_{\text{GP}}^2(\boldsymbol{x})$ for various $\boldsymbol{x}$. However, doing so directly may cause the decomposition to fail since the condition number of Σ_∇ cannot always be bounded with a finite nugget, as explained in Section 4.1. Instead, the Cholesky decomposition of the preconditioned matrix $\dot{\Sigma}_\nabla$ from Eq. (35) is calculated in Algorithm 1, which is preferable since $\kappa(\dot{\Sigma}_\nabla(\boldsymbol{\gamma})) \leq \kappa_{\max}\ \forall\, \boldsymbol{\gamma} > 0$. For the Gaussian kernel the nugget $\eta_{\dot{\mathsf{K}}_\nabla}$ can be calculated with either Eq. (44) or Eq. (45). The former is simpler to implement since $\eta_{\dot{\mathsf{K}}_\nabla}$ does not depend on the hyperparameter while the latter provides a smaller nugget value. For kernels other than the Gaussian kernel only the use of Eq. (45) to calculate $\eta_{\dot{\mathsf{K}}_\nabla}$ ensures that $\kappa(\dot{\Sigma}_\nabla(\boldsymbol{\gamma})) \leq \kappa_{\max}\ \forall\, \boldsymbol{\gamma} > 0$.

Applying Algorithm 1 provides the following relation

$$\begin{aligned}\mathsf{L}\mathsf{L}^\top &= \mathsf{P}\dot{\mathsf{L}}\dot{\mathsf{L}}^\top\mathsf{P} \\ &= \hat{\sigma}_{\mathsf{K}}^2\left(\mathsf{K}_\nabla + \eta_{\dot{\mathsf{K}}_\nabla}\mathsf{P}\mathsf{P}\right) + \mathsf{V}_\nabla,\end{aligned}$$

Algorithm 1 Stable Cholesky decomposition for gradient-enhanced GP

Select evaluation points X and hyperparameters $\boldsymbol{\gamma}$, $\hat{\sigma}_\mathsf{K}$, $\hat{\sigma}_f$, and $\hat{\sigma}_{\nabla f}$
Calculate K_∇ and V_∇ with Eqs. (12) and (17), respectively
Calculate η_{K_∇} with Eq. (45), or Eq. (44) for the Gaussian kernel
From Eq. (33): $\mathsf{P} = \operatorname{diag}\left(\sqrt{\operatorname{diag}\left(\mathsf{K}_\nabla + \hat{\sigma}_\mathsf{K}^{-2}\mathsf{V}_\nabla\right)}\right)$
From Eq. (32): $\dot{\mathsf{K}}_\nabla = \mathsf{P}^{-1}\left(\mathsf{K}_\nabla + \hat{\sigma}_\mathsf{K}^{-2}\mathsf{V}_\nabla\right)\mathsf{P}^{-1}$
From Eq. (35): $\dot{\Sigma}_\nabla = \hat{\sigma}_\mathsf{K}^2\left(\dot{\mathsf{K}}_\nabla + \eta_{\dot{\mathsf{K}}_\nabla}\mathsf{I}\right)$
$\dot{\mathsf{L}}\dot{\mathsf{L}}^\top = \dot{\Sigma}_\nabla$
$\mathsf{L} = \mathsf{P}\dot{\mathsf{L}}$, where $\mathsf{L}\mathsf{L}^\top = \Sigma_\nabla$ from Eq. (16) with $\mathsf{W} = \mathsf{P}\mathsf{P}$

TABLE 1: Comparison of methods to address the ill-conditioning of the covariance matrix Σ_∇. The baseline and rescaling methods are summarized in Sections 4.2 and 4.3, respectively, and the implementation of the preconditioning method is provided in Section 5.6.

Method	Baseline	Rescale	Precondition
$\kappa(\Sigma_\nabla(\boldsymbol{\gamma})) \leq \kappa_{\max}, \forall \boldsymbol{\gamma} > 0$	✗	$\gamma_1 = \ldots = \gamma_d$	✓
Constraint free hyperparameter optz	✗	✗	✓
Nodes can be collocated	✓	✗	✓
Provides a correlation matrix	✗	✗	✓
Bounded $\kappa(\dot{\Sigma}_\nabla(\boldsymbol{\gamma}))$ for other kernels	✗	✗	✓
Works with Hessian-enhanced covariance matrix	✓	✗	✓

which is equal to Σ_∇ with $\mathsf{W} = \mathsf{P}\mathsf{P}$ from Eq. (16). The result of preconditioning the non-regularized covariance matrix Σ_∇ with P and then adding a constant nugget is thus equivalent to using a variable nugget. A varying nugget was also used by Chen *et al.*, but their method does not provide an upper bound on the condition number of the gradient-enhanced covariance matrix [30]. It is important to note that the condition number of $\dot{\Sigma}_\nabla$ can be several orders of magnitude smaller than the condition number for Σ_∇. As such, the Cholesky decomposition of the preconditioned matrix should be performed, as detailed in Algorithm 1.

5.7 | Summary of methods

In Sections 5.3, 5.4, and 5.5 variations of the preconditioning method are introduced. The variations from Sections 5.3 and 5.4 are the simplest to implement since their nuggets are independent of $\boldsymbol{\gamma}$. However, the nugget for the former scales as $\mathcal{O}(n_x d)$ and while the latter scales as $\mathcal{O}\left(n_x\sqrt{d}\right)$, it only ensures that $\kappa(\dot{\Sigma}_\nabla(\boldsymbol{\gamma})) \leq \kappa_{\max}\ \forall\, \boldsymbol{\gamma} > 0$ if the Gaussian kernel is used. The nugget from Section 5.5 depends on $\boldsymbol{\gamma}$, but it provides a nugget smaller than the one from Section 5.3 and, for the Gaussian kernel, it is also smaller than the one from Section 5.5.

Table 1 provides a comparison of the preconditioning method with the baseline method and rescaling methods from Sections 4.2 and 4.3, respectively. The greatest advantage of the preconditioning method is that it ensures that $\kappa(\dot{\Sigma}_\nabla(\boldsymbol{\gamma})) \leq \kappa_{\max}\ \forall\, \boldsymbol{\gamma} > 0$, which results in the Cholesky decomposition of a well conditioned matrix, so long as $\kappa_{\max}$ is not selected to be too large. The preconditioning method thus does not require a constraint on the condition number of the covariance matrix when optimizing $\boldsymbol{\gamma}$, unlike the baseline and rescaling methods. The preconditioning method also provides other benefits relative to the rescaling method such as allowing evaluation points to be collocated, enabling any twice differentiable kernel to be used, and it can also be applied to Hessian-enhanced covariance matrices. In the following section the practical benefits of using the preconditioning method for a Bayesian optimizer are demonstrated.

6 | RESULTS

6.1 | Condition numbers of the noise-free Gaussian kernel covariance matrices

The condition numbers of the gradient-free and gradient-enhanced covariance matrices for the noise-free case are compared for the baseline method presented in Section 4.2, the rescaling method from Marchildon and Zingg [32] that is summarized in

Section 4.3, and the preconditioning method introduced in this paper. The noise-free covariance matrices depend only on the evaluation points in X, the nugget, and the hyperparameters $\boldsymbol{\gamma}$, $\hat{\sigma}_f$, and $\hat{\sigma}_{\nabla f}$. However, the maximization of the marginal log-likelihood also depends on the function of interest and we use the Rosenbrock function:

$$f(\boldsymbol{x}) = \sum_{i=1}^{d-1} \left[10 \left(x_{i+1} - x_i^2 \right)^2 + (1 - x_i)^2 \right]. \tag{49}$$

The selection of the evaluation points is important since the ill-conditioning problem of the covariance matrix is made worse when the points are close together[13]. If the evaluation points are selected randomly, or come from an optimizer performing local optimization, some evaluation points will naturally be clustered close together, making the ill-conditioning problem more acute. Meanwhile, if the evaluation points are selected from a Latin hypercube sampling, they will inherently be spaced apart. The selection of the evaluation points is not problematic for the preconditioning method since it ensures that $\kappa(\dot{\Sigma}_\nabla) \leq \kappa_{\max} \; \forall \, \boldsymbol{\gamma} > 0, \mathsf{X} \in \mathbb{R}^{n_x \times d}$, including when some or all of the evaluation points are collocated. To demonstrate how the gradient-enhanced covariance matrix can become ill-conditioned even in the best case, i.e. when the evaluation points are evenly spaced apart, we select them with a $d = 2$ Latin hypercube sampling centred around $\boldsymbol{x} = [1, 1]^\top$, which is the minimum for the Rosenbrock function:

$$\mathsf{X} = 10^{-3} \times \begin{bmatrix} 1 & 9 & 7 & -9 & -5 & -7 & -3 & 5 & 3 & -1 \\ 1 & -3 & 7 & 3 & 5 & -9 & -7 & 9 & -1 & -5 \end{bmatrix}^\top + 1, \tag{50}$$

where $v_{\min} = \sqrt{2}/500 \approx 2.8 \times 10^{-3}$, which is the minimum Euclidean distance between evaluation points.

Fig. 3 plots the condition number of the noise-free covariance matrices as a function of $\boldsymbol{\gamma}$ for the evaluation points from Eq. (50). The star marker indicates where the marginal log-likelihood from Eq. (24) is maximized. The nugget value for all cases is $\eta = 1.5 \times 10^{-9}$, which comes from Eq. (44). Red regions in Fig. 3 indicate where the condition number is greater than $\kappa_{\max} = 10^{10}$. Selecting a different value for $\kappa_{\max}$ would not impact the results in Fig. 3 since $\eta_{\dot{\mathsf{K}}_\nabla}$ was calculated with Eq. (44), which takes into account $\kappa_{\max}$.

Figs. 3a and 3b plot the condition number of the noise-free gradient-free and gradient-enhanced covariance matrices, respectively, using the baseline method, which does not precondition the covariance matrix but adds the nugget to its diagonal. For the gradient-free case we have $\kappa(\Sigma(\boldsymbol{\gamma})) < \kappa_{\max} \; \forall \, \boldsymbol{\gamma} > 0$. However, for the baseline gradient-enhanced case $\kappa(\Sigma_\nabla(\boldsymbol{\gamma})) \geq \kappa_{\max}$ for most values of $\boldsymbol{\gamma}$, including where the marginal log-likelihood from Eq. (24) is maximized. Selecting the hyperparameters to satisfy the constraint $\kappa(\Sigma_\nabla(\boldsymbol{\gamma})) \leq \kappa_{\max}$ results in a lower marginal log-likelihood. This impacts the accuracy of the surrogate, which degrades the performance of the Bayesian optimizer, as will be shown in Section 6.5.

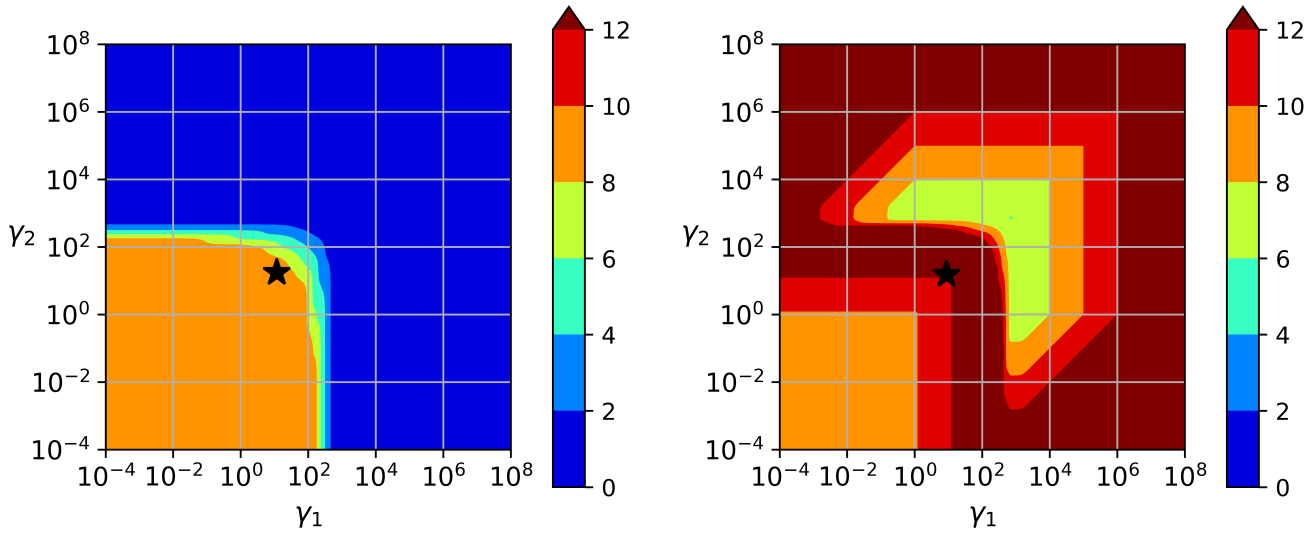
Fig. 3c plots $\log(\kappa(\Sigma_\nabla))$ with the use of the rescaling method. As stated in Section 4.3, the rescaling method only ensures that $\kappa(\Sigma_\nabla(\boldsymbol{\gamma})) \leq \kappa_{\max}$ when $\gamma_1 = \ldots = \gamma_d$ and Σ_∇ is not diagonally dominant. While there are several values of $\boldsymbol{\gamma}$ where $\kappa(\Sigma_\nabla(\boldsymbol{\gamma})) \geq \kappa_{\max}$, the condition number is below $\kappa_{\max}$ where the marginal log-likelihood is maximized.

From Fig. 3d we have $\kappa(\dot{\mathsf{K}}_\nabla) < \kappa_{\max} \; \forall \, \boldsymbol{\gamma} > 0$ as a result of the preconditioning method. Consequently, there is no need for a constraint on the condition number for the optimization of the hyperparameters. This ensures that the hyperparameters are never constrained by the need to bound the condition number of the covariance matrix. Furthermore, this also provides a small reduction in the cost of the hyperparameter optimization since the constraint and its gradient do not need to be calculated. The nugget $\eta_{\dot{\mathsf{K}}_\nabla}$ was calculated with Eq. (44). However, if Eq. (45) were used instead to calculate the nugget it would be 17% smaller at the point in the hyperparameter space where the marginal log-likelihood is maximized while still ensuring $\kappa(\dot{\Sigma}_\nabla) \leq \kappa_{\max}$.

The blue regions in Figs. 3a and 3d indicate where Σ and $\dot{\Sigma}_\nabla$ are nearly equal to the identity matrix. While the condition number is very small in these regions, the marginal log-likelihood is as well. This makes it undesirable to select those values of $\boldsymbol{\gamma}$.

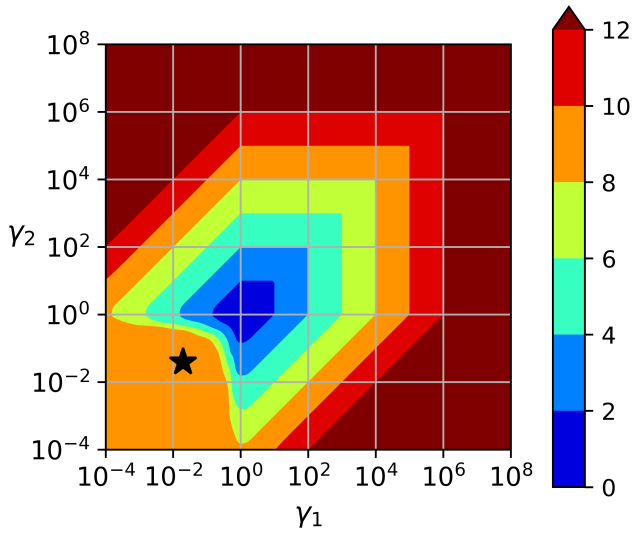
6.2 | Comparing different $\eta_{\dot{\mathsf{K}}_\nabla}$

In Fig. 4a the condition number of $\dot{\mathsf{K}}_\nabla(\mathsf{X}; \boldsymbol{\gamma})$ is plotted as a function of $\gamma_1 = \gamma_2$ and the set of evaluation X comes from Eq. (50). For $\gamma < 100$ we have $\kappa(\dot{\mathsf{K}}_\nabla) > \kappa_{\max} = 10^{10}$. With evaluations of the Rosenbrock function from Eq. (49), the marginal log likelihood from Eq. (24) is maximized at $\gamma = 18$, where $\kappa(\dot{\mathsf{K}}_\nabla) = 1.7 \times 10^{17}$. From Fig. 4b we can see that the nugget value from Eq. (40), which comes from Dalbey[15] and is normalized by $\eta_{\text{Tr}} = 3 \times 10^{-9}$, is approximately unity until $\gamma > 100$, which is where $\kappa(\dot{\mathsf{K}}_\nabla) < \kappa_{\max}$, as seen in Fig. 4a. For $\kappa(\dot{\mathsf{K}}_\nabla) > \kappa_{\max}$ the largest nugget values come from Eq. (40), the ones from Eq. (44) are half as large, and Eq. (45) provides nugget values that are even smaller while still ensuring that $\kappa(\dot{\mathsf{K}}_\nabla(\boldsymbol{\gamma}) + \eta_{\dot{\mathsf{K}}_\nabla} \mathsf{I}) \leq \kappa_{\max} \; \forall \, \boldsymbol{\gamma} > 0$.

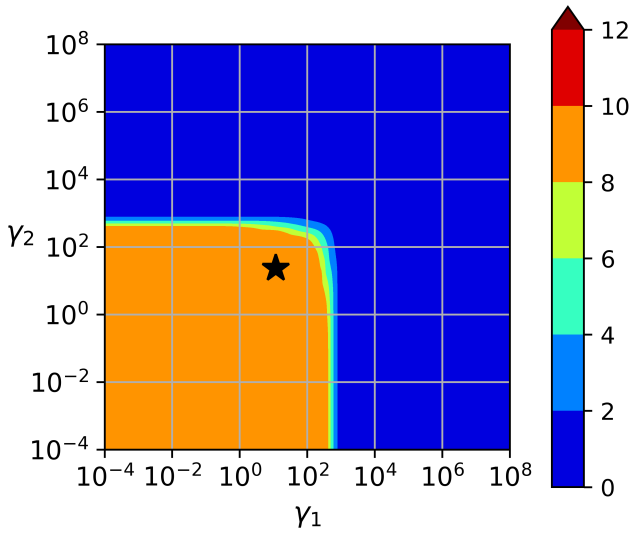


(a) Baseline gradient-free: $\log(\kappa(\Sigma))$

(b) Baseline gradient-enhanced: $\log(\kappa(\Sigma_\nabla))$



(c) Gradient-enhanced with the rescaling method: $\log(\kappa(\Sigma_\nabla))$



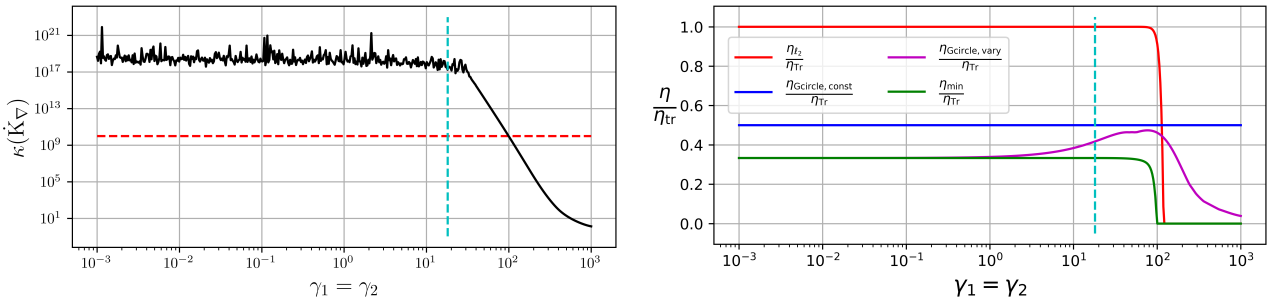
(d) Gradient-enhanced with the preconditioning method: $\log(\kappa(\dot{\Sigma}_\nabla))$

FIGURE 3 The condition number for the noise-free covariance matrices for the Gaussian kernel with the nugget $\eta_{\dot{\mathsf{K}}_\nabla} = 1.5 \times 10^{-9}$ from Eq. (44) with $\kappa_{\max} = 10^{10}$, and the set of evaluation points X from Eq. (50). The star markers indicate where the marginal log-likelihood function from Eq. (24) is maximized with the use of the Rosenbrock function from Eq. (49).

The relative advantage of using Eq. (44) to calculate $\eta_{\dot{\mathsf{K}}_\nabla}$ instead of Eq. (40) increases as the dimension increases since the former scales as $\eta_{\dot{\mathsf{K}}_\nabla} = \mathcal{O}(n_x\sqrt{d})$ while the latter scales as $\eta_{\dot{\mathsf{K}}_\nabla} = \mathcal{O}(n_x d)$ when $\kappa(\dot{\mathsf{K}}_\nabla) \gg \kappa_{\max}$. Eq. (45) can be used to calculate nugget values smaller than the ones provided by Eq. (44) and these ensure that $\kappa(\dot{\mathsf{K}}_\nabla(\gamma) + \eta_{\dot{\mathsf{K}}_\nabla}\mathsf{I}) \leq \kappa_{\max}\ \forall\, \gamma > 0$ for the use of any twice-differentiable kernels. Eq. (26) provides the smallest nonnegative nugget sufficient to ensure that $\kappa(\dot{\mathsf{K}}_\nabla(\gamma) + \eta_{\dot{\mathsf{K}}_\nabla}\mathsf{I}) \leq \kappa_{\max}\ \forall\, \gamma > 0$, but it requires $\lambda_{\min}$ and $\lambda_{\max}$ to be calculated exactly, which is computationally expensive.

6.3 | Application to other kernels

The preconditioning method can be applied to gradient-enhanced covariance matrices that utilize kernels other than the Gaussian kernel considered thus far. For example, the preconditioning method can be applied to the Matérn $\frac{5}{2}$ and rational quadratic



(a) The ℓ_2 condition number of $\dot{\mathsf{K}}_\nabla$ with $\kappa_{\max} = 10^{10}$ indicated by a dashed red line.

(b) The nuggets η_{ℓ_2}, $\eta_{\text{G circle, const}}$, $\eta_{\text{G circle, vary}}$, $\eta_{\min}$, and η_{Tr} can all be used in place of $\eta_{\dot{\mathsf{K}}_\nabla}$ and come from Eq. (40), (44), (45), (26), and (28), respectively.

FIGURE 4 Plots of $\kappa(\dot{\Sigma}_\nabla)$ with the set of evaluation points X from Eq. (50), $\hat{\sigma}_f = \hat{\sigma}_{\nabla f} = 0$, $\eta_{\text{Tr}} = 3 \times 10^{-9}$, and $\gamma_1 = \gamma_2$. The location where the marginal log likelihood from Eq. (24) is maximized with the condition $\gamma_1 = \gamma_2$ is indicated by the vertical dashed line.

kernels, which are both stationary like the Gaussian kernel[6]:

$$k_{\text{M}\frac{5}{2}}(\dot{\boldsymbol{r}}) = \left(1 + \sqrt{3}\|\dot{\boldsymbol{r}}\| + \|\dot{\boldsymbol{r}}\|^2\right) e^{-\sqrt{3}\|\dot{\boldsymbol{r}}\|} \tag{51}$$

$$k_{rq}(\dot{\boldsymbol{r}}) = \left(1 + \frac{\|\dot{\boldsymbol{r}}\|^2}{2\alpha}\right)^{-\alpha}, \tag{52}$$

where $\alpha > 0$ is a hyperparameter and $\dot{r}_i = \gamma_i(x_i - y_i)$. The hyperparameters for the Matérn $\frac{5}{2}$ and rational quadratic kernels from Eqs. (51) and (52), respectively, have been selected such that the preconditioning matrix P also comes from Eq. (34) for the noise-free case. The preconditioning method could also be applied to more general kernels that are, for example, non-stationary. The only practical limitation is that the kernel must be at least twice differentiable in order to construct the gradient-enhanced kernel matrix from Eq. (12).

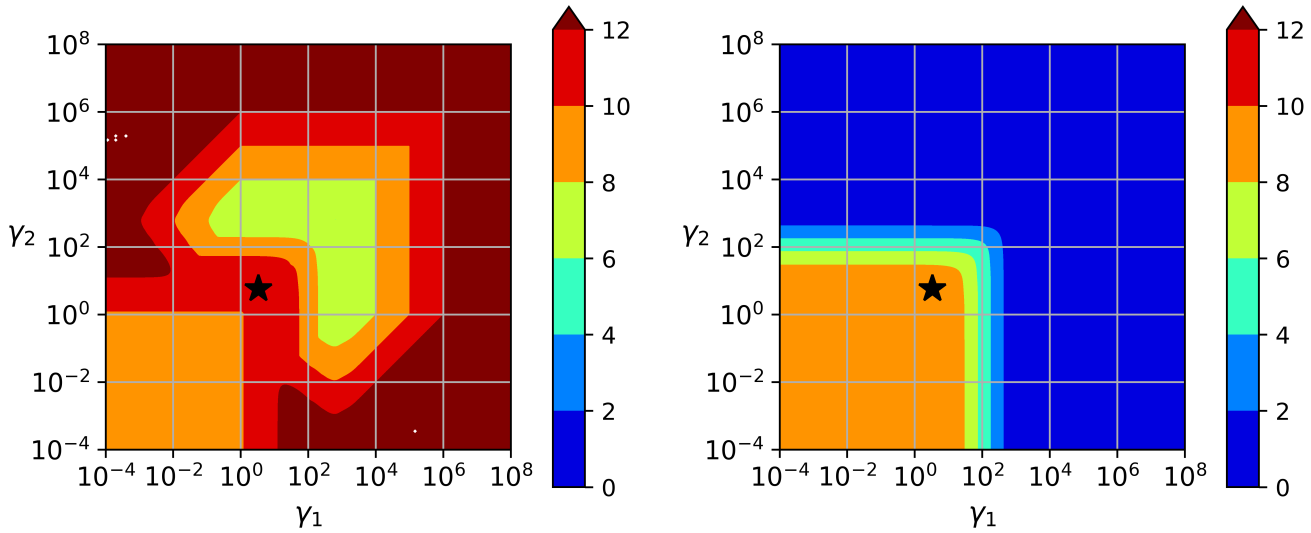
Fig. 5 plots the condition number of the baseline and preconditioned gradient-enhanced covariance matrices for the Matérn $\frac{5}{2}$ and rational quadratic kernels for the noise-free case with $\kappa_{\max} = 10^{10}$. The set of evaluation points in X comes from Eq. (50) and the nugget for the baseline method comes from Eq. (44) while Eq. (45) is used for the preconditioning method. It is clear from Figs. 5a and 5c that the condition number for the baseline method, i.e. the non-preconditioned gradient-enhanced covariance matrices, for both kernels is larger than $\kappa_{\max}$ for several values of $\boldsymbol{\gamma}$, including where the marginal log-likelihood is maximized at the star marker. However, with the preconditioning method we have $\kappa(\dot{\Sigma}_\nabla(\boldsymbol{\gamma})) \leq \kappa_{\max}\ \forall\, \boldsymbol{\gamma} > 0$ for both kernels as seen in Figs. 5b and 5d. This demonstrates that the gradient-enhanced covariance matrix constructed with various kernels suffers from severe ill-conditioning. Fortunately, the preconditioning method can be applied to bound the condition number of $\dot{\Sigma}_\nabla$ constructed with various kernels. Using Eq. (45) with the hyperparameters that maximize the marginal log-likelihood results in nugget values 29% and 24% smaller relative to the one from Eq. (44) for the Matérn $\frac{5}{2}$ and rational quadratic kernels, respectively.

6.4 | Noisy data

Fig. 6 plots the condition number of the preconditioned gradient-enhanced covariance matrix for the Gaussian, Matérn $\frac{5}{2}$, and rational quadratic kernels with $\hat{\sigma}_f = 10^{-6}$ and $\hat{\sigma}_{\nabla f} = 0.1$. The set of evaluation points X comes from Eq. (50); and the nugget $\eta_{\dot{\mathsf{K}}_\nabla}$ is calculated with Eq. (45) and $\kappa_{\max} = 10^{10}$. For all three kernels the preconditioning method ensures that $\kappa(\dot{\Sigma}_\nabla(\boldsymbol{\gamma}, \hat{\sigma}_f, \hat{\sigma}_{\nabla f}) \leq \kappa_{\max}\ \forall\, \boldsymbol{\gamma} > 0,\ \hat{\sigma}_f, \hat{\sigma}_{\nabla f} \geq 0$.

6.5 | Optimization

In this section, the baseline, rescaling, and preconditioning methods are compared when used with a Bayesian optimizer to minimize the Rosenbrock function from Eq. (49) with $d \in \{5, 10, 20\}$. For the preconditioning method the steps detailed in Algorith 1 are used with the nugget calculated with Eq. (45). The results demonstrate the same trends if Eq. (44) is used to



(a) Baseline Matérn $\frac{5}{2}$ kernel: $\log(\kappa(\Sigma_\nabla))$

(b) Preconditioned Matérn $\frac{5}{2}$ kernel: $\log(\kappa(\dot{\Sigma}_\nabla))$

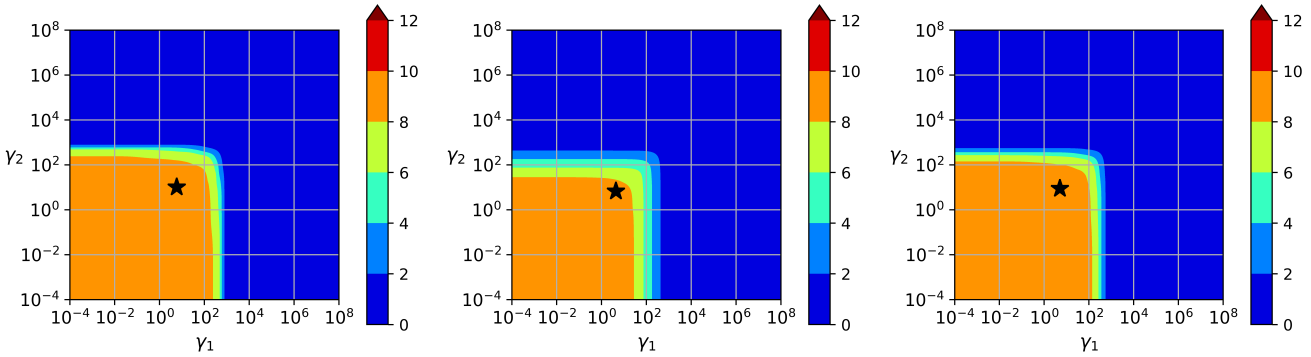
(c) Baseline rational quadratic kernel: $\log(\kappa(\Sigma_\nabla))$

(d) Preconditioned rational quadratic kernel: $\log(\kappa(\dot{\Sigma}_\nabla))$

FIGURE 5 The condition number of the noise-free gradient-enhanced covariance matrices with the baseline and preconditioned methods with the set of evaluation points X from Eq. (50). The baseline method uses Eq. (44) with $\kappa_{\max} = 10^{10}$ to calculate $\eta_{\mathsf{K}_\nabla} = 1.5 \times 10^{-9}$, while the preconditioned method uses Eq. (45) for $\eta_{\dot{\mathsf{K}}_\nabla}$. The value of the hyperparameters $\boldsymbol{\gamma}$ that maximizes the marginal log-likelihood function from Eq. (24) with the Rosenbrock function from Eq. (49) is indicated by a star marker.

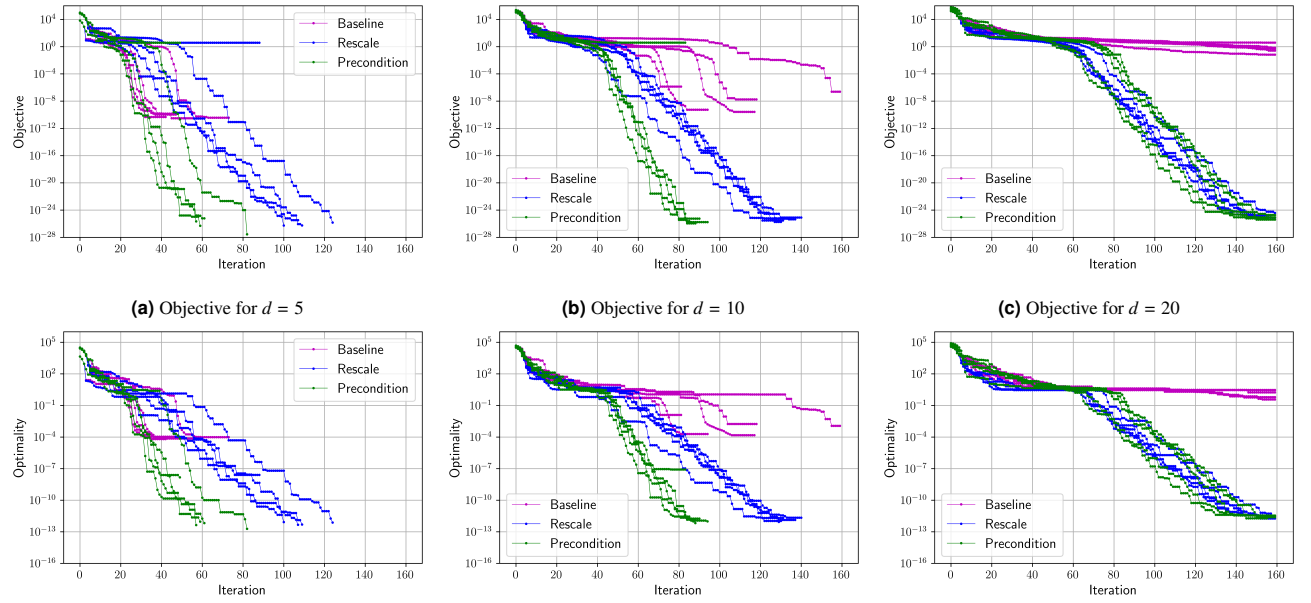
calculate the nugget instead. All of the different methods to address the ill-conditioning of the gradient-enhanced covariance matrix result in different linear systems being solved. As such, in the iterative process of minimizing the Rosenbrock function the Bayesian optimizer using the different methods will take different paths in the parameter space. The goal of these various methods is to alleviate the ill-conditioning problem of the Bayesian optimizer to enable it to find the minimum of the Rosenbrock function in the fewest number of iterations as possible, i.e. minimize the number of function and gradient evaluations of the Rosenbrock function.

Five separate runs of the Bayesian optimizers for each of the methods were performed, each initiated with one starting point. The starting points were the same for each of the methods and were selected from a Latin hypercube sampling from the open source Surrogate Modeling Toolbox[37] with lower and upper parameter bounds of –10 and 10, respectively. With the use of gradient-free GPs, a Bayesian optimizer would need to be initiated with several starting points in order for the posterior of the GP to be an accurate surrogate[2]. However, a single starting point is sufficient to initialize the gradient-enhanced Bayesian



(a) Gaussian kernel **(b)** Matérn $\frac{5}{2}$ kernel **(c)** Rational quadratic kernel

FIGURE 6 Plots of $\log(\kappa(\dot{\Sigma}_\nabla))$ with X from Eq. (50), $\hat{\sigma}_f = 10^{-6}$, $\hat{\sigma}_{\nabla f} = 0.1$, and the nugget is calculated with Eq. (45) with $\kappa_{\max} = 10^{10}$. The star marker indicates the location in the hyperparameter space where the marginal log-likelihood is maximized with function and gradient evaluations coming from the Rosenbrock function from Eq. (49).



(a) Objective for $d = 5$ **(b)** Objective for $d = 10$ **(c)** Objective for $d = 20$

(d) Optimality for $d = 5$ **(e)** Optimality for $d = 10$ **(f)** Optimality for $d = 20$

FIGURE 7 Bayesian optimization of the Rosenbrock function from Eq. (49) using the baseline, rescaling, and preconditioning methods for $d \in \{5, 10, 20\}$. The plots show the lowest evaluated objective or optimality, i.e. the ℓ_2 norm of the gradient of Eq. (49), for each optimization run at each iteration.

optimizer. The selected acquisition function is the upper-confidence function

$$q(\boldsymbol{x}) = \mu_{\text{GP}}(\boldsymbol{x}) - \omega\sigma_{\text{GP}}(\boldsymbol{x}), \tag{53}$$

where $\omega \geq 0$ promotes exploitation when it is small, and exploration when it is large. We use $\omega = 0$ since we are interested in local optimization for the unimodal Rosenbrock function. The gradient-based SLSQP optimizer from the Python library SciPy is used to select the hyperparameters by maximizing the marginal log-likelihood. The same optimizer is used to minimize the acquisition function to select the next point in the parameter space to evaluate the Rosenbrock function. A trust region is used in the minimization of the acquisition function, similar to the one used by Eriksson *et al.* [38], where a Bayesian optimizer was also used for local minimization. However, our trust region is set to be a hypersphere instead of a hyperrectangle.

In Fig. 7 the objective and optimality, which is the ℓ_2 norm of the gradient, are compared for the Bayesian optimizer using the baseline, rescaling, and preconditioning methods. The plots for the objective and optimality show similar trends and we thus focus on the latter. There are two important observations from the optimality plots: the depth and rate of convergence of the optimality. In all cases, the rescaling and preconditioning methods converge the optimality several orders of magnitude deeper than the baseline method. In fact, the rescaling and preconditioning methods converge the optimality to below 10^{-12} in all test cases. Meanwhile, the deepest optimality that the baseline method achieves is 10^{-4}. As the dimensionality increases, the optimizer with the baseline method is not able to converge the optimality as deeply and can only achieve an optimality of 10^{-1} for the $d = 20$ case. The optimizer with the rescaling and preconditioning methods is thus able to converge the optimality 5 to 9 additional orders of magnitude relative to the optimizer with the baseline method. For the baseline method, the hyperparameters γ are selected by solving Eq. (29), where the marginal log-likelihood is maximized with an upper bound on the condition number. As the optimality is converged, the evaluation points get closer together in the parameter space and this makes the ill-conditioning of the gradient-enhanced covariance matrix worse[32]. Consequently, solving Eq. (29) results in hyperparameters that provide a lower marginal log-likelihood since the upper bound on the condition number becomes a more onerous constraint. The rescaling and baseline methods do not suffer from this since, by construction, they guarantee that the selected hyperparameters maximize the marginal log-likelihood without being constrained by the condition number of the covariance matrix.

It is clear from Fig. 7 that the optimizer utilizing the preconditioning method achieves the fastest rate of convergence of the three methods for $d = 5$ and $d = 10$. Meanwhile, for the $d = 20$ test case the optimizer utilizing the rescaling and preconditioning methods achieve similar results. The slower convergence of the optimality for the optimizer using the rescaling method was also observed in Marchildon and Zingg[32] for test cases with the Rosenbrock function with $d = 2$ and $d = 5$. This was found to be a consequence of the rescaling method providing a surrogate with gradients that have larger errors relative to the baseline method.

In summary, the use of the preconditioning method with a gradient-enhanced Bayesian optimizer enables the optimality to be converged more deeply than with the use of the baseline method, and in fewer iterations than with the rescaling method.

7 | CONCLUSIONS

The posterior of a gradient-enhanced GP provides a more accurate probabilistic surrogate than its gradient-free counterpart but the ill-conditioning of its covariance matrix has been a hindrance to its use. The preconditioning method from this paper improves upon the secondary method from Dalbey[15]. The contributions of this paper are to derive a smaller nugget sufficient to bound the condition number of the preconditioned gradient-enhanced covariance matrix, to provide the gradients required to perform gradient-based optimization of the hyperparameters with this method, to handle cases with noisy function and gradient evaluations, and to do so with a method that is less computationally expensive than the one from Dalbey[15]. The method is simple to implement, as detailed in Algorithm 1 from Section 5.6. For the Gaussian kernel it was proven that a nugget value that is sufficient to bound the condition number of the preconditioned gradient-enhanced covariance matrix scales in the worst case as $\eta_{\dot{\mathsf{K}}_\nabla} = \mathcal{O}(n_x\sqrt{d})$.

The benefits of using the preconditioning method relative to the baseline and rescaling methods are summarized in Table 1. With the preconditioning method, all of the data points can be kept and there is no minimum distance requirement between evaluation points in the parameter space. Unlike the rescaling method, the points can even be collocated, which may be beneficial when the evaluations of the function of interest and of its gradient are noisy. Since the preconditioning method ensures that $\kappa(\dot{\Sigma}_\nabla(\gamma)) \leq \kappa_{\max} \; \forall \, \gamma > 0$, no constraint is required when maximizing the marginal log-likelihood. This simplifies the optimization and reduces its computational cost. The preconditioning method also provides a correlation matrix, which makes the GP easier to interpret. Eq. (45) can be used to provide a nugget value sufficient to ensure that $\kappa(\dot{\Sigma}_\nabla) \leq \kappa_{\max}$ for use with any kernel. Finally, the preconditioning method can be straightforwardly applied to Hessian-enhanced covariance matrices by preconditioning the matrix and then adding a nugget calculated with the same methodology as presented in Section 5.5. As described in Section 5.6, the preconditioning method with its preconditioning then regularization is equivalent to adding a non-constant nugget to the covariance matrix. Finally, the preconditioning method ensures that the Cholesky decomposition is always performed on a matrix with a condition number smaller than $\kappa_{\max}$.

In Section 6.5 the Rosenbrock function was optimized for $d \in \{5, 10, 20\}$ with a Bayesian optimizer using the baseline, rescaling and preconditioning methods. The Bayesian optimizer with the preconditioning method converged the optimality an additional 5-9 orders of magnitude relative to the optimizer with the baseline method. Furthermore, the preconditioning method enabled the Bayesian optimizer to converge the optimality more quickly than when the rescaling method is used, particularly for

the lower-dimensional problems. The slower convergence of a Bayesian optimizer using the rescaling method was previously identified to be the result of its surrogate having gradients with larger errors. In conclusion, the preconditioning method bounds the condition number of the preconditioned gradient-enhanced covariance matrix and it enables a Bayesian optimizer to achieve deeper and faster convergence relative to the use of either the baseline or rescaling methods.

The preconditioning method can also be used with gradient-enhanced GPs applied to various other applications such as uncertainty quantification, classification, and regression[2,3,4]. In all of these cases, using a gradient-enhanced GP provides a more accurate surrogate compared to the use of a gradient-free GP. The advantage of using a gradient-enhanced GP increases as the number of parameters increases.

ACKNOWLEDGMENTS

The authors would like to thank the Natural Sciences and Engineering Research Council of Canada and the Ontario Graduate Scholarship Program for their financial support. Furthermore, the authors are also thankful to the reviewers whose feedback helped improve this paper.

DATA AVAILABILITY STATEMENT

The baseline, rescaling, and preconditioning methods are all available in the open source Python library GpGradPy, which can be found at https://github.com/marchildon/gpgradpy/tree/paper_precon.

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

ORCID

André Marchildon: https://orcid.org/0000-0001-6407-3987

References

1. Zingg DW, Nemec M, Pulliam TH. A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization. *European Journal of Computational Mechanics.* 2008;17(1-2):103–126. doi: 10.3166/remn.17.103-126

2. Shahriari B, Swersky K, Wang Z, Adams RP, Freitas dN. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE.* 2016;104(1):148–175. doi: 10.1109/JPROC.2015.2494218

3. Eriksson D, Dong K, Lee E, Bindel D, Wilson AG. Scaling Gaussian Process Regression with Derivatives. In: Advances in Neural Information Processing Systems 2018; Montreal, Canada.

4. Schulz E, Speekenbrink M, Krause A. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of Mathematical Psychology.* 2018;85:1–16. doi: 10.1016/j.jmp.2018.03.001

5. Jones DR. A Taxonomy of Global Optimization Methods Based on Response Surfaces. *Journal of global optimization.* 2001;21:345–383. doi: 10.1023/A:1012771025575

6. Rasmussen CE, Williams CKI. *Gaussian processes for machine learning*. Adaptive computation and machine learningCambridge, Mass: MIT Press, 2006. OCLC: ocm61285753.

7. Ababou R, Bagtzoglou AC, Wood EF. On the condition number of covariance matrices in kriging, estimation, and simulation of random fields. *Mathematical Geology.* 1994;26(1):99–133. doi: 10.1007/BF02065878

8. Williams CKI. Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond. tech. rep., Springer Netherlands; Dordrecht: 1998

9. Toal DJJ, Bressloff NW, Keane AJ. Kriging Hyperparameter Tuning Strategies. *AIAA Journal.* 2008;46(5):1240–1252. doi: 10.2514/1.34822

10. Toal DJJ, Forrester AIJ, Bressloff NW, Keane AJ, Holden C. An adjoint for likelihood maximization. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences.* 2009;465(2111):3267–3287. doi: 10.1098/rspa.2009.0096

11. Toal DJ, Bressloff NW, Keane AJ, Holden CM. The development of a hybridized particle swarm for kriging hyperparameter tuning. *Engineering Optimization.* 2011;43(6):675–699. doi: 10.1080/0305215X.2010.508524

12. Ollar J, Mortished C, Jones R, Sienz J, Toropov V. Gradient based hyper-parameter optimisation for well conditioned kriging metamodels. *Structural and Multidisciplinary Optimization.* 2017;55(6):2029–2044. doi: 10.1007/s00158-016-1626-8

13. Davis GJ, Morris MD. Six Factors Which Affect the Condition Number of Matrices Associated with Kriging. *Mathematical Geology.* 1997;29(5):669–683. doi: 10.1007/BF02769650

14. Wu A, Aoi MC, Pillow JW. Exploiting gradients and Hessians in Bayesian optimization and Bayesian quadrature. *arXiv:1704.00060 [stat].* 2018. arXiv: 1704.00060.

15. Dalbey K. Efficient and robust gradient enhanced Kriging emulators.. Tech. Rep. SAND2013-7022, 1096451, Sandia National Laboratories; 2013

16. Osborne MA, Garnett R, Roberts SJ. Gaussian Processes for Global Optimization. In: Learning and Intelligent Optimization (LION) 2009; Trento, Italy.

17. Ulaganathan S, Couckuyt I, Dhaene T, Degroote J, Laermans E. Performance study of gradient-enhanced Kriging. *Engineering with Computers.* 2016;32(1):15–34. doi: 10.1007/s00366-015-0397-y

18. Wu J, Poloczek M, Wilson AG, Frazier P. Bayesian Optimization with Gradients. In: 31st Conference on Neural Information Processing Systems 2017; Long Beach, CA, USA:5273–5284.

19. Zimmermann R. On the Maximum Likelihood Training of Gradient-Enhanced Spatial Gaussian Processes. *SIAM Journal on Scientific Computing.* 2013;35(6):A2554–A2574. doi: 10.1137/13092229X

20. Han ZH, Görtz S, Zimmermann R. Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function. *Aerospace Science and Technology.* 2013;25(1):177–189. doi: 10.1016/j.ast.2012.01.006

21. Laurent L, Le Riche R, Soulier B, Boucard PA. An Overview of Gradient-Enhanced Metamodels with Applications. *Archives of Computational Methods in Engineering.* 2019;26(1):61–106. doi: 10.1007/s11831-017-9226-3

22. Hung TH, Chien P. A Random Fourier Feature Method for Emulating Computer Models With Gradient Information. *Technometrics.* 2021;63(4):500–509. doi: 10.1080/00401706.2020.1852973

23. De Roos F, Gessner A, Hennig P. High-Dimensional Gaussian Process Inference with Derivatives. In: International Conference on Machine Learning 2021:2535–2545.

24. Kostinski AB, Koivunen AC. On the condition number of Gaussian sample-covariance matrices. *IEEE Transactions on Geoscience and Remote Sensing.* 2000;38(1):329–332. doi: 10.1109/36.823928

25. Zimmermann R. On the condition number anomaly of Gaussian correlation matrices. *Linear Algebra and its Applications.* 2015;466:512–526. doi: 10.1016/j.laa.2014.10.038

26. Wendland H. *Scattered Data Approximation*. Cambridge University Press. 1 ed., 2004

27. Higham NJ. Cholesky factorization. *Wiley Interdisciplinary Reviews: Computational Statistics.* 2009;1(2):251–254. doi: 10.1002/wics.18

28. Mohammadi H, Le Riche R, Durrande N, Touboul E, Bay X. An analytic comparison of regularization methods for Gaussian Processes. 2017. arXiv:1602.00853 [math, stat].

29. He X, Chien P. On the Instability Issue of Gradient-Enhanced Gaussian Process Emulators for Computer Experiments. *SIAM/ASA Journal on Uncertainty Quantification.* 2018;6(2):627–644. doi: 10.1137/16M1088247

30. Chen Y, Hosseini B, Owhadi H, Stuart AM. Solving and learning nonlinear PDEs with Gaussian processes. *Journal of Computational Physics.* 2021;447:110668. doi: 10.1016/j.jcp.2021.110668

31. March A, Willcox K, Wang Q. Gradient-based multifidelity optimisation for aircraft design using Bayesian model calibration. *The Aeronautical Journal.* 2011;115(1174):729–738. doi: 10.1017/S0001924000006473

32. Marchildon AL, Zingg DW. A Non-intrusive Solution to the Ill-Conditioning Problem of the Gradient-Enhanced Gaussian Covariance Matrix for Gaussian Processes. *Journal of Scientific Computing.* 2023;95(3):65. doi: 10.1007/s10915-023-02190-w

33. Hadd A, Rodgers JL. *Understanding Correlation Matrices*. 186 of *Quantitative Applications in the Social Sciences*. SAGE Publications, Inc, 2021.

34. Ameli S, Shadden SC. Noise Estimation in Gaussian Process Regression. *arXiv.* 2022:41.

35. Gramacy RB, Lee HKH. Cases for the nugget in modeling computer experiments. *Statistics and Computing.* 2012;22(3):713–722. doi: 10.1007/s11222-010-9224-x

36. Won JH, Kim SJ. Maximum Likelihood Covariance Estimation with a Condition Number Constraint. In: IEEE 2006; Grove, CA, USA:1445–1449

37. Bouhlel MA, Hwang JT, Bartoli N, Lafage R, Morlier J, Martins JR. A Python surrogate modeling framework with derivatives. *Advances in Engineering Software.* 2019;135:102662. doi: 10.1016/j.advengsoft.2019.03.005

38. Eriksson D, Pearce M, Gardner J, Turner RD, Poloczek M. Scalable Global Optimization via Local Bayesian Optimization. In: Advances in Neural Information Processing Systems 2019; Vancouver, Canada:12.

□

APPENDIX

A PROOFS

A.1 Proof for Proposition 2

We consider the noise-free preconditioned matrix $\dot{\mathsf{K}}_\nabla$ since the magnitude of its entries are largest in this case, as explained in the proof for Proposition 1. The derivation of the upper bound for the sum of the absolute value of the off-diagonal entries is the same for any of the last $n_x d$ rows of $\dot{\mathsf{K}}_\nabla$, which comes from Eq. (32). Without loss of generality, we consider the b-th row of $\dot{\mathsf{K}}_\nabla$, where $b = pn_x + m$, and p and m can take any integer values that satisfy $1 \leq p \leq d$ and $1 \leq m \leq n_x$:

$$
\begin{aligned}
\sum_{\substack{i=1\\ i\neq b}}^{n_x} \left|\dot{\mathsf{K}}_\nabla\right|_{bi} &= \sum_{\substack{i=1\\ i\neq m}}^{n_x} \left(\left|\dot{x}_{mp} - \dot{x}_{ip}\right| + \sum_{j=1}^{d} \left|\delta_{jp} - \left(\dot{x}_{mp} - \dot{x}_{ip}\right)\left(\dot{x}_{mj} - \dot{x}_{ij}\right)\right| \right) \exp\left(-\frac{\|\dot{\boldsymbol{x}}_{m:} - \dot{\boldsymbol{x}}_{i:}\|_2^2}{2}\right) \\
&\leq \sum_{\substack{i=1\\ i\neq m}}^{n_x} \left(\left|\dot{x}_{mp} - \dot{x}_{ip}\right| + 1 + \sum_{\substack{j=1\\ j\neq p}}^{d} \left|\left(\dot{x}_{mp} - \dot{x}_{ip}\right)\left(\dot{x}_{mj} - \dot{x}_{ij}\right)\right| \right) \exp\left(-\frac{\|\dot{\boldsymbol{x}}_{m:} - \dot{\boldsymbol{x}}_{i:}\|_2^2}{2}\right) \\
&= \sum_{\substack{i=1\\ i\neq m}}^{n_x} \left(1 + \left|\dot{x}_{mp} - \dot{x}_{ip}\right| \left(1 + \sum_{\substack{j=1\\ j\neq p}}^{n_x} \left|\dot{x}_{mj} - \dot{x}_{ij}\right| \right) \right) \exp\left(-\frac{\|\dot{\boldsymbol{x}}_{m:} - \dot{\boldsymbol{x}}_{i:}\|_2^2}{2}\right) \\
&\leq (n_x - 1) \max_{\nu\geq 0, \check{\boldsymbol{w}}\geq 0} \left(1 + \nu\left(1 + \check{\mathbf{1}}_p^\top \check{\boldsymbol{w}}\right)\right) \exp\left(-\frac{1}{2}\left(\nu^2 + \check{\boldsymbol{w}}^\top \check{\boldsymbol{w}}\right)\right),
\end{aligned}
\tag{A1}
$$

where $\nu = |\dot{x}_{mp} - \dot{x}_{ip}|$ and $\check{w}_j = |\dot{x}_{mj} - \dot{x}_{ij}|$, except for $\check{w}_p = 0$. Similarly $\check{\mathbf{1}}_p$ is a vector of ones of length d with a zero at its p-th entry. The first inequality is a result of $\dot{\mathsf{K}}_\nabla$ being a correlation matrix as explained in Section 5.1.

An analogous approach to the one taken in Proposition 1 can be used to show that the maximization of Eq. (A1) requires $\check{\boldsymbol{w}} = \alpha\check{\mathbf{1}}_p$, i.e. that all but the p-th entries in $\check{\boldsymbol{w}}$ are equal. Using $\check{\boldsymbol{w}} = \alpha\check{\mathbf{1}}_p$ with Eq. (A1) gives

$$
g_1(\nu, \alpha; d) = (\nu + 1 + (d-1)\alpha\nu)\, e^{-\frac{\nu^2 + (d-1)\alpha^2}{2}}. \tag{A2}
$$

We thus need to prove that $(n_x - 1)g_1(\nu, \alpha; d) < u_{\mathrm{G}}(n_x, d)$ for $\nu, \alpha \geq 0$ and $d \in \mathbb{Z}^+$. The following lemma considers the case for $d = 1$.

Lemma 2. *For $d = 1$ we have*

$$
(n_x - 1) \max_{\nu\geq 0, \alpha\geq 0} g_1(\nu, \alpha; d = 1) = u_G(n_x, d = 1) = (n_x - 1)\frac{1 + \sqrt{5}}{2} e^{-\frac{3-\sqrt{5}}{4}}, \tag{A3}
$$

where $g_1(\nu, \alpha; d)$ comes from Eq. (A2) and $u_G(n_x, d)$ comes from Eq. (43).

Proof. For $d = 1$ the parameter α cancels out and we thus have a scalar function that we seek to maximize, giving

$$
\begin{aligned}
\frac{\partial g_1(\nu; d = 1)}{\partial \nu} &= \frac{\partial\left((\nu + 1)\, e^{-\frac{\nu^2}{2}}\right)}{\partial \nu} \\
&= -\left(\nu^2 + \nu - 1\right) e^{-\frac{\nu^2}{2}} = 0 \\
\nu^*_{d=1} &= \frac{-1 + \sqrt{5}}{2},
\end{aligned}
$$

where only the positive root was kept since $\nu \geq 0$ and it is straightforward to verify that this critical point maximizes $g_1(\nu, \alpha; d = 1)$. Eq. (A3) is recovered by evaluating g_1 with $\nu = \nu^*_{d=1}$ and $d = 1$, which completes the proof. □

To consider the cases for $d \geq 2$ we need to find the values of α and ν that maximize $g_1(\nu, \alpha; d)$ from Eq. (A2). The following lemma considers the maximization of g_1 with respect to α.

Lemma 3. *For $\nu, \alpha \geq 0$ and $d \geq 2$ we have $g_1(\nu, \alpha; d) \leq g_2(\nu; d)$, where g_1 comes from Eq. (A2) and*

$$g_2(\nu; d) = \left(\frac{\nu + 1 + \sqrt{h_1(\nu; d)}}{2} \right) e^{-\frac{\nu^2}{2} + h_2(\nu; d)}, \tag{A4}$$

where

$$h_1(\nu; d) = (\nu + 1)^2 + 4\nu^2(d - 1) \tag{A5}$$

$$h_2(\nu; d) = \frac{(\nu + 1)\sqrt{h_1(\nu; d)} - (\nu + 1)^2}{4\nu^2(d - 1)} - \frac{1}{2}. \tag{A6}$$

Proof. The maximum of $g_1(\nu, \alpha; d)$ with respect to α is identified by calculating its derivative, setting it to zero, and solving for α:

$$\frac{\partial g_1(\nu, \alpha; d)}{\partial \alpha} = -(d - 1)\left((d - 1)\nu\alpha^2 + (\nu + 1)\alpha - \nu\right) e^{-\frac{\nu^2 + (d-1)\alpha^2}{2}} = 0$$

$$\alpha^* = \frac{-(\nu + 1) + \sqrt{(\nu + 1)^2 + 4\nu^2(d - 1)}}{2\nu(d - 1)},$$

where only the positive root of the quadratic equation is kept since α must be positive, and it is straightforward to verify that this provides the maximum of $g_1(\nu, \alpha; d)$. The function $g_2(\nu; d)$ from Eq. (A4) is recovered by evaluating $g_1(\nu, \alpha^*; d)$, which completes the proof.

□

Both $\sqrt{h_1(\nu; d)}$ and $h_2(\nu; d)$ from Eqs. (A5) and (A6), respectively, are non-polynomial functions that make it impractical to find a closed-form maximum solution for $g_2(\nu; d)$. The following two lemmas provide upper bounds for these non-polynomial functions.

Lemma 4. *For $\nu \geq 0$ and $d \geq 2$, we have the bound $\sqrt{h_1(\nu; d)} \leq h_3(\nu; d)$, where $h_1(\nu; d)$ comes from Eq. (A5) and $h_3(\nu; d)$ is the following $\mathbb{C}^0$ continuous piecewise polynomial:*

$$h_3(\nu; d) = \begin{cases} (2\sqrt{d} - 1)\nu + 1 & \text{if } 0 \leq \nu \leq 1 \\ 2\sqrt{d}\nu & \text{if } \nu > 1. \end{cases} \tag{A7}$$

Proof. For $0 \leq \nu \leq 1$ and $d \geq 2$ we start by showing that $h_3^2 \geq h_1$:

$$\left((2\sqrt{d} - 1)\nu + 1\right)^2 \geq (\nu + 1)^2 + 4\nu^2(d - 1)$$

$$4\nu(1 - \nu)(\sqrt{d} - 1) \geq 0.$$

Next we demonstrate that $h_3^2 \geq h_1(\nu; d)$ for $\nu \geq 1$:

$$\left(2\sqrt{d}\nu\right)^2 \geq (\nu + 1)^2 + 4\nu^2(d - 1)$$

$$\left(\nu + \frac{1}{3}\right)(\nu - 1) \geq 0.$$

Finally, it is straightforward to verify that $h_3(\nu; d)$ is $\mathbb{C}^0$ continuous:

$$\lim_{\nu \to 1^-} h_3(\nu; d) = \lim_{\nu \to 1^+} h_3(\nu; d) = 2\sqrt{d}, \tag{A8}$$

which completes the proof. □

Lemma 5. *The maximum value for $h_2(\nu; d)$ from Eq. (A6) for $\nu \geq 0$ and $d \in \mathbb{Z}^+$ is*

$$\max_{\nu \geq 0} h_2(\nu; d) = \lim_{\nu \to 0} h_2(\nu, d) = 0. \tag{A9}$$

Proof. We start by proving that $h_2(\nu; d)$ is monotonically decreasing with respect to ν by showing that its derivative with respect to ν is nonpositive for $\nu \geq 0$ and $d \in \mathbb{Z}^+$

$$\frac{\partial h_2(\nu; d)}{\partial \nu} = -\left(\frac{h_1 - (\nu+1)\sqrt{h_1} - 2\nu^2(d-1)}{2\nu^3\sqrt{h_1}(d-1)}\right).$$

Since the denominator of $\frac{\partial h_2(\nu;d)}{\partial \nu}$ is always nonnegative for $\nu \geq 0$ and $d \in \mathbb{Z}^+$, we only need to show that its numerator is also nonnegative for the same range of parameters:

$$\begin{aligned} h_1 - (\nu+1)\sqrt{h_1} - 2\nu^2(d-1) &\geq 0 \\ \left[h_1 - 2\nu^2(d-1)\right]^2 &\geq \left[(\nu+1)\sqrt{h_1}\right]^2 \\ h_1\left(h_1 - \left[(\nu+1)^2 + 4\nu^2(d-1)\right]\right) + 4\nu^4(d-1)^2 &\geq 0 \\ 4\nu^4(d-1)^2 &\geq 0, \end{aligned}$$

where h_1 comes from Eq. (A5) and it straightforward to verify that $h_1 - 2\nu^2(d-1) > 0$ for $\nu \geq 0$ and $d \in \mathbb{Z}^+$ and thus, squaring this term on the second line does not change its sign. Since $h_2(\nu; d)$ is monotonically decreasing with respect to ν for $\nu \geq 0$ and $d \in \mathbb{Z}^+$, its maximum is at $\nu = 0$. To evaluate $h_2(\nu; d)$ we use a limit and apply l'Hôpital's rule twice:

$$\begin{aligned} \lim_{\nu\to 0} h_2(\nu; d) &= \lim_{\nu\to 0}\left[\frac{(\nu+1)\sqrt{h_1} - (\nu+1)^2}{4(d-1)\nu^2}\right] - \frac{1}{2} \\ &= \lim_{\nu\to 0}\left[\frac{2\sqrt{h_1} + \frac{4\nu(d-1)}{\sqrt{h_1}} - 2(\nu+1)}{8\nu(d-1)}\right] - \frac{1}{2} \\ &= \lim_{\nu\to 0}\left[\frac{\frac{4(\nu+1)(d-1)}{h_1^{3/2}} + \frac{(8d-6)\nu+2}{\sqrt{h_1}} - 2}{8(d-1)}\right] - \frac{1}{2} \\ &= 0, \end{aligned}$$

where $h_1(\nu; d)$ comes from Eq. (A5) and this completes the proof. □

Thanks to Lemmas 4 and 5 it is now possible to derive a closed-form solution for an upper bound of $g_2(\nu; d)$ from Eq. (A4) for $\nu \geq 0$ and $d \geq 2$. This is considered in the following two lemmas that consider the case for $0 \leq \nu \leq 1$ and $\nu > 1$, respectively.

Lemma 6. *For $0 \leq \nu \leq 1$ and $d \geq 2$ we have $(n_x - 1)g_2(\nu; d) < u_G(n_x, d)$, where $g_2(\nu; d)$ and $u_G(n_x, d)$ come from Eqs. (A4) and (43), respectively.*

Proof. The function $g_2(\nu; d)$ from Eq. (A4) contains the nonlinear functions $h_1(\nu; d)$ and $h_2(\nu; d)$ from Eqs. (A5) and (A6), respectively. We use the upper bounds provided by Lemmas 4 and 5 for these functions and $0 \leq \nu \leq 1$ to get $g_2(\nu; d) < g_3(\nu; d)$, where

$$\begin{aligned} g_3(\nu; d) &= \frac{\nu + 1 + \left[(2\sqrt{d} - 1)\nu + 1\right]}{2} e^{-\frac{\nu^2}{2}} \\ &= \left(\sqrt{d}\nu + 1\right) e^{-\frac{\nu^2}{2}}. \end{aligned} \tag{A10}$$

We now find the value of ν that maximizes $g_3(\nu; d)$

$$\begin{aligned} \frac{\partial g_3}{\partial \nu} &= \left(\sqrt{d} - \nu\left(\sqrt{d}\nu + 1\right)\right) e^{-\frac{\nu^2}{2}} = 0 \\ \nu_3^* &= \frac{-1 + \sqrt{1 + 4d}}{2\sqrt{d}}, \end{aligned} \tag{A11}$$

where only the positive root was kept and it is straightforward to verify that $0 < \nu^* < 1$ for $d \geq 2$, and that this is the maximum for the function g_3. Using ν_3^* from Eq. (A11) gives $(n_x - 1)g_3(\nu_3^*; d) = u_G(n_x, d)$, where $u_G(n_x, d)$ comes from Eq. (43). Therefore,

we have for $d \geq 2$:

$$(n_x - 1) \max_{0 \leq \nu \leq 1} g_2(\nu; d) < (n_x - 1) \max_{0 \leq \nu \leq 1} g_3(\nu; d) = (n_x - 1) g_3(\nu_3^*; d) = u_G(n_x, d), \tag{A12}$$

which completes the proof.

□

Lemma 7. *We have $(n_x - 1)g_2(\nu; d) < u_G(n_x, d)$ for $\nu \geq 1$ and $d \geq 2$, where $g_2(\nu; d)$ and $u_G(n_x, d)$ come from Eqs. (A4) and (43), respectively.*

Proof. We now consider the case for $\nu > 1$ by substituting $h_3(\nu; d)$ from Eq. (A7) for $\nu > 1$ into $g_2(\nu; d)$ for $\sqrt{h_1(\nu; d)}$ and using the results from Lemma 5 for an upper bound on $h_2(\nu; d)$. We get the bound $g_2(\nu; d) \leq g_4(\nu; d)$, where

$$g_4(\nu; d) = \frac{\nu + 1 + \left\lfloor 2\sqrt{d}\nu \right\rfloor}{2} e^{-\frac{\nu^2}{2}}. \tag{A13}$$

We now find the value of $\nu \geq 1$ that maximizes $g_4(\nu; d)$:

$$\frac{\partial g_4}{\partial \nu} = \frac{\left(2\sqrt{d} + 1\right) - \nu\left(\left(2\sqrt{d} + 1\right)\nu + 1\right)}{2} e^{-\frac{\nu^2}{2}} = 0$$

$$(2\sqrt{d} + 1)\nu^2 + \nu - (2\sqrt{d} + 1) = 0$$

$$\nu_4^* = \frac{-1 + \sqrt{1 + 4\left(2\sqrt{d} + 1\right)^2}}{2(2\sqrt{d} + 1)},$$

where only the positive root was kept and it is straightforward to show that this provides the maximum for $g_4(\nu; d)$. However, we now demonstrate that this root does not satisfy the constraint $\nu \geq 1$:

$$\nu_4^* < \frac{-1 + \left\lfloor 1 + 2\left(2\sqrt{d} + 1\right)\right\rfloor}{2(2\sqrt{d} + 1)} = 1,$$

where we used the inequality $\sqrt{b_1 + b_2} < \sqrt{b_1} + \sqrt{b_2}$ for $b_1, b_2 > 0$. Since there are no roots for $\nu \geq 1$ that maximize $g_4(\nu; d)$ for $d \geq 2$, it is either maximized at $\nu = 1$ or $\nu \to \infty$. For $\lim \nu \to \infty$ we have $g_4(\nu; d) = 0$ and for $\nu = 1$ we have

$$(n_x - 1)g_4(\nu = 1, d) = (n_x - 1)g_3(\nu = 1, d) < (n_x - 1)g_3(\nu_3^*, d) = u_G(n_x, d), \tag{A14}$$

where $g_3 = g_4$ for $\nu = 1$ since both functions used the relation $\sqrt{h_1(\nu; d)} \leq h_3(\nu; d)$ and it was shown in Lemma 4 that $h_3(\nu)$ from Eq. (A7) is $\mathbb{C}^0$ continuous. We thus have $(n_x - 1)g_2(\nu; d) < u_G(n_x, d)$ for $\nu \geq 1$ and $d \geq 2$, which completes the proof. □

It has been proven that the function g_1 from Eq. (A2), which provides an upper bound for the sum of absolute values for the off-diagonal entries for any of the last $n_x d$ rows of $\dot{\mathsf{K}}_\nabla$, is smaller than $u_G(n_x, d)$ for $n_x, d \in \mathbb{Z}^+$, which completes the proof.