Gradient-Enhanced Bayesian Optimization with Application to Aerodynamic Shape Optimization

André L. Marchildon [*] and David W. Zingg [†]
4925 Dufferin St, North York, ON M3H 5T6

Bayesian optimizers have several desirable properties that make them well suited for various aerodynamic shape optimization applications. For example, the design space can often be multimodal, and Bayesian optimizers are efficient global optimizers. Bayesian optimizers also enable the use of mixed-fidelity data, the use of inexact function and gradient evaluations, and uncertainty quantification thanks to their use of probabilistic surrogates. The challenges of applying a Bayesian optimizer to aerodynamic shape optimization problems include the high-dimensional design space, the nonlinear constraints, and their limited application to local optimization. A local optimization framework for a gradient-enhanced Bayesian optimizer is developed in this paper that is shown to be competitive with the popular quasi-Newton based optimizer SNOPT for the nonlinearly constrained aerodynamic shape optimization of a transonic airfoil. A recently developed preconditioning method is used to address the ill-conditioning of the gradient-enhanced covariance matrix, which enables the Bayesian optimizer to converge the optimality as deeply as SNOPT. With these developments, gradient-enhanced Bayesian optimization represents a versatile option for a wide range of challenging aerodynamic shape optimization problems, including unimodal and multimodal problems, and chaotic flows where calculating accurate gradients is challenging.

I. Introduction

Aerodynamic shape optimization enables the design of aircraft that generate reduced drag while satisfying geometric, trim, and other constraints [1]. Challenges of performing aerodynamic shape optimization include the significant computing resources required for the computational fluid dynamics (CFD) simulations, the presence of several nonlinear constraints, and a design space that is high-dimensional and potentially multimodal [2, 3]. There is also growing interest in using large-eddy simulation for aerodynamic shape optimization, which presents additional challenges, namely greater computational resources and memory requirements, and most challenging of all, chaotic flows [4, 5]. Conventional methods to calculate sensitivities, such as the adjoint method, break down for chaotic systems and the alternative methods that have been developed provide inexact gradients [6–8].

Gradient-based optimizers have been found to be more effective than gradient-free optimizers for aerodynamic shape optimization due to the high-dimensional design space [9]. The gradients for the objective and nonlinear constraints can be calculated efficiently using the adjoint method [10, 11]. Historically, quasi-Newton based optimizers have been used for aerodynamic shape optimization [12, 13]. They can handle linear and nonlinear constraints and are effective in a high-dimensional design space [14]. When the design space is multimodal, quasi-Newton based optimizers are at risk of getting stuck in a local minimum that results in a suboptimal design. To avoid this, gradient-based multistart has been used, which requires a large number of expensive flow solves to be performed [2, 3]. Another drawback of quasi-Newton optimizers is that they are deterministic and thus cannot quantify the error in the function or gradient evaluations that they are provided. An inexact gradient that does not point in a direction of descent could cause a quasi-Newton optimizer to stall since it cannot find a suitable step size to reduce the function of interest [14].

Bayesian optimization is well suited for global optimization and to handle inexact function and gradient evaluations [15, 16]. A Bayesian optimizer uses a Gaussian process (GP) to form a probabilistic surrogate to approximate the function of interest. The GP can be provided with inexact function and gradient evaluations and it will estimate their uncertainty. If the data is noise-free, the posterior of the GP can interpolate all previous function evaluations, and gradients if they are available, thus making it well suited for multimodal design spaces [15, 17]. Unlike gradient-based multistart, where each optimization is performed independently, all of the function and gradient evaluations can be used

[*] PhD Candidate, University of Toronto Institute for Aerospace Studies, AIAA student member

[†] Distinguished Professor of Computational Aerodynamics and Sustainable Aviation, University of Toronto Institute for Aerospace Studies, AIAA Associate Fellow

by the Bayesian optimizer to perform the global optimization efficiently, i.e. with as few expensive function evaluations as possible.

Other desirable properties of Bayesian optimizers that could prove useful for aerodynamic shape optimization include being able to handle mixed-fidelity data and inexact function and gradient evaluations. This latter advantage would be useful to perform aerodynamic shape optimization when the flow is chaotic since only inexact gradients can be calculated in such cases [6–8]. Chaotic flows arise for example with the use of large-eddy simulations, where there is growing interest in using them for optimization [4, 5]. Bayesian optimizers are able to efficiently use inexact function and gradient evaluations since they utilize a probabilistic surrogate, which is commonly a GP [17].

Bayesian optimization has previously been used with and without gradients to perform optimization for various aerospace applications involving structures, aerodynamics, and tuning turbulence model coefficients [18–20]. Some of the challenges faced in these and other applications of Bayesian optimizers include the ill-conditioning of the covariance matrix, particularly when gradients are used, the handling of nonlinear constraints, and the limited use of Bayesian optimizers for local optimization.

Using a gradient-enhanced Bayesian optimizer instead of a gradient-free variant helps alleviate the curse of dimensionality [21, 22]. Unfortunately, its use has been limited by the severe ill-conditioning of the gradient-enhanced covariance matrix [23]. The ill-conditioning problem is also present for gradient-enhanced Kriging [19, 24]. Various methods were developed to mitigate this problem, such as limiting how close the evaluation points can get to each other [25], constraining the hyperparameters [26, 27], or removing evaluation points until the condition number is sufficiently low [24, 28, 29]. All of these methods result in a less accurate surrogate and are undesirable to perform efficient local optimization. This paper uses a recently developed method that preconditions and regularizes the gradient-enhanced covariance matrix to ensure that its condition number is below a user-set threshold [30].

For the handful of methods that have been developed for Bayesian optimizers to perform constrained optimization, a separate GP is generally used to approximate each of the nonlinear constraints [31–33]. The most popular method takes the product of the expected improvement acquisition function with the probabilities that each of the nonlinear constraints are satisfied [31, 34]. Unfortunately, this method can only be applied to nonlinear inequality constraints, whereas nonlinear equality constraints are common for aerodynamic shape optimization, e.g. lift and trim constraints. The local optimization framework developed in this paper is able to handle nonlinear inequality and equality constraints. Instead of using the posteriors of the GPs approximating the nonlinear constraints to form an acquisition function, their means are provided to the acquisition function minimizer as nonlinear constraints.

The objective of this paper is to develop an efficient local optimization framework for nonlinearly constrained problems using gradient-enhanced Bayesian optimization. This will complement the advantageous properties that Bayesian optimizers have for solving problems that are multimodal or have inexact gradients. This expanded versatility will enable the application of Bayesian optimizers to a wider range of optimization problems that includes aerodynamic shape optimization.

The notation used in this paper is presented in Section II, gradient-enhanced GPs are introduced in Section III, and the preconditioning method to address the ill-conditioning of the covariance matrix is summarized in Section IV. Details on the Bayesian optimizer, such as its acquisition function and its trust region, can be found in Section V. A nonlinearly constrained aerodynamic shape optimization problem is presented in Section VI. Finally, the conclusions of this paper are in Section VII.

II. Notation

Scalars are denoted with lowercase non-bolded symbols, which are either Greek letters or sans serif Latin letters. For integers, the letter n is typically used along with a subscript, e.g. n_x and n_d are used to indicate the number of evaluation points and dimensions, respectively. Vectors are denoted with bold lowercase symbols and matrices with uppercase symbols. For example, the matrix of nodal locations is given by $\mathsf{X} \in \mathbb{R}^{n_x \times n_d}$, its i-th row is given by $\boldsymbol{x}_{i:}$, and its entry at the i-th row and j-th column is given by x_{ij}. Lower and upper bounds are denoted by, for example, $\underline{\boldsymbol{x}}$ and $\overline{\boldsymbol{x}}$, respectively.

III. Gradient-enhanced Gaussian processes

A GP is typically used as the probabilistic surrogate required for a Bayesian optimizer [25, 35]. Fully defining a Gaussian process requires a mean function, which is often taken to be a constant, and a covariance function. Two popular kernels that are used for the latter are the Gaussian and Matérn $\frac{5}{2}$ kernels [17]:

$$k_{\mathrm{G}}(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\gamma}) = k_{\mathrm{G}}(\dot{\boldsymbol{r}}) \quad = e^{-\frac{1}{2}\|\dot{\boldsymbol{r}}\|^2} \tag{1}$$

$$k_{\mathrm{M}\frac{5}{2}}(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\gamma}) = k_{\mathrm{M}\frac{5}{2}}(\dot{\boldsymbol{r}}) = \left(1 + \sqrt{3}\|\dot{\boldsymbol{r}}\| + \|\dot{\boldsymbol{r}}\|^2\right) e^{-\sqrt{3}\|\dot{\boldsymbol{r}}\|}, \tag{2}$$

where $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{n_d}$ are points in the parameter space, $\boldsymbol{\gamma} \in \mathbb{R}_+^{n_d}$ is a vector of hyperparameters, and $\dot{r}_i = \gamma_i(x_i - y_i)\ \forall\, i \in \{1, \dots, n_d\}$. The Gaussian and Matérn $\frac{5}{2}$ kernels are popular since they are simple, they contain hyperparameters that can be tuned, and they are at least twice continuously differentiable, which makes them suitable for use with a gradient-enhanced Gaussian process. The Gaussian kernel is infinitely continuously differentiable, while the Matérn $\frac{5}{2}$ kernel is only twice continuously differentiable. The gradient-free kernel matrix, which can be formed using either of the presented kernel functions, is given by

$$\mathsf{K} = \mathsf{K}(\mathsf{X};\boldsymbol{\gamma}) = \begin{bmatrix} 1 & k(\boldsymbol{x}_{1:}, \boldsymbol{x}_{2:}, ; \boldsymbol{\gamma}) & \dots & k(\boldsymbol{x}_{1:}, \boldsymbol{x}_{n_x:}; \boldsymbol{\gamma}) \\ k(\boldsymbol{x}_{2:}, \boldsymbol{x}_{1:}; \boldsymbol{\gamma}) & 1 & \dots & k(\boldsymbol{x}_{2:}, \boldsymbol{x}_{n_x:}; \boldsymbol{\gamma}) \\ \vdots & \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_{n_x:}, \boldsymbol{x}_{1:}; \boldsymbol{\gamma}) & k(\boldsymbol{x}_{n_x:}, \boldsymbol{x}_{2:}; \boldsymbol{\gamma}) & \dots & 1, \end{bmatrix}, \tag{3}$$

where $x_{i:}$ is the i-th row of X, which holds the n_x points in the design space where the function of interest has been evaluated. In general, the i-th diagonal entry of K is $k(\boldsymbol{x}_{i:}, \boldsymbol{x}_{i:}; \boldsymbol{\gamma})$, which is simply unity for the Gaussian and Matérn $\frac{5}{2}$ kernels, but not for all kernels [17]. For a gradient-enhanced Gaussian process we also require the gradient-enhanced kernel matrix:

$$\mathsf{K}_\nabla(\mathsf{X};\boldsymbol{\gamma}) = \begin{bmatrix} \mathsf{K} & \frac{\partial \mathsf{K}}{\partial y_1} & \dots & \frac{\partial \mathsf{K}}{\partial y_d} \\ \frac{\partial \mathsf{K}}{\partial x_1} & \frac{\partial^2 \mathsf{K}}{\partial x_1 \partial y_1} & \dots & \frac{\partial^2 \mathsf{K}}{\partial x_1 \partial y_d} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathsf{K}}{\partial x_d} & \frac{\partial^2 \mathsf{K}}{\partial x_d \partial y_1} & \dots & \frac{\partial^2 \mathsf{K}}{\partial x_d \partial y_d} \end{bmatrix}, \tag{4}$$

where the derivatives of K with respect to x_i and y_j indicate derivatives of the kernel function with respect to the i-th entry of its first and second arguments, respectively. The noise-free gradient-enhanced covariance matrix is given by

$$\Sigma_\nabla(\mathsf{X}; \hat{\sigma}_{\mathsf{K}}, \boldsymbol{\gamma}, \eta_{\mathsf{K}_\nabla}, \mathsf{W}) = \hat{\sigma}_{\mathsf{K}}^2 \left(\mathsf{K}_\nabla(\mathsf{X};\boldsymbol{\gamma}) + \eta_{\mathsf{K}_\nabla}\mathsf{W}\right), \tag{5}$$

where the hyperparameter $\hat{\sigma}_{\mathsf{K}}^2$ is the variance of the stationary residual error, W is a diagonal matrix with nonnegative entries, and $\eta_{\mathsf{K}_\nabla} \geq 0$ is a nugget that regularizes the covariance matrix to help alleviate its ill-conditioning.

The mean and variance of the gradient-enhanced Gaussian process are evaluated with [35]

$$\mu_{\mathrm{GP}}(\boldsymbol{x}) = \beta + \hat{\sigma}_{\mathsf{K}}^2\, \boldsymbol{k}_\nabla^\top(\boldsymbol{x}) \Sigma_\nabla^{-1} \left(\boldsymbol{f}_\nabla - \beta \check{\mathbf{1}}\right) \tag{6}$$

$$\sigma_{\mathrm{GP}}^2(\boldsymbol{x}) = \hat{\sigma}_{\mathsf{K}}^2 \left(k(\boldsymbol{x},\boldsymbol{x}) - \hat{\sigma}_{\mathsf{K}}^2\, \boldsymbol{k}_\nabla^\top(\boldsymbol{x}) \Sigma_\nabla^{-1} \boldsymbol{k}_\nabla(\boldsymbol{x})\right), \tag{7}$$

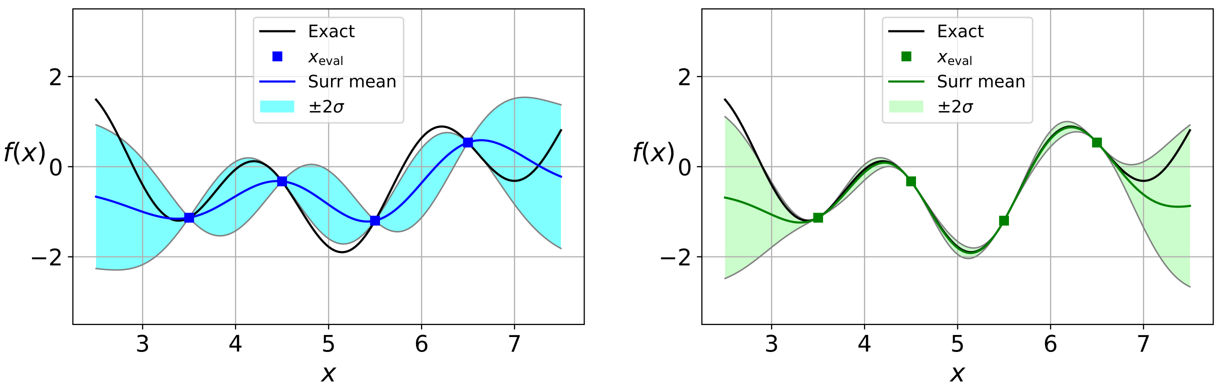
where β is the constant for the mean function of the Gaussian process, $\check{\mathbf{1}} = [\mathbf{1}_{n_x}^\top, \mathbf{0}_{n_x n_d}^\top]^\top$, and

$$\boldsymbol{k}_\nabla(\boldsymbol{x};\mathsf{X}) = \begin{bmatrix} \boldsymbol{k}(\mathsf{X},\boldsymbol{x}) \\ \frac{\partial \boldsymbol{k}(\mathsf{X},\boldsymbol{x})}{\partial x_1} \\ \vdots \\ \frac{\partial \boldsymbol{k}(\mathsf{X},\boldsymbol{x})}{\partial x_{n_d}} \end{bmatrix}, \quad \boldsymbol{f}_\nabla(\mathsf{X}) = \begin{bmatrix} \boldsymbol{f}(\mathsf{X}) \\ \frac{\partial \boldsymbol{f}(\mathsf{X})}{\partial x_1} \\ \vdots \\ \frac{\partial \boldsymbol{f}(\mathsf{X})}{\partial x_{n_d}} \end{bmatrix}, \tag{8}$$

where $\boldsymbol{f}(\mathsf{X})$ is the evaluation of the function of interest evaluated at each of the rows of X. A challenge of using gradient-enhanced Gaussian processes is that Σ_∇ can become extremely ill-conditioned. This is addressed in Section IV.

The hyperparameters β, $\hat{\sigma}_{\mathsf{K}}^2$, and $\boldsymbol{\gamma}$ are set by maximizing the marginal log-likelihood [27, 36–38]:

$$L(\boldsymbol{\gamma}, \beta, \hat{\sigma}_{\mathsf{K}}^2; \mathsf{X}, \boldsymbol{f}_\nabla, \eta_{\mathsf{K}_\nabla}) = \frac{e^{-\frac{(\boldsymbol{f}_\nabla - \beta\hat{\mathbf{1}})^\top \Sigma_\nabla^{-1} (\boldsymbol{f}_\nabla - \beta\hat{\mathbf{1}})}{2}}}{(2\pi)^{\frac{n_x(n_d+1)}{2}} \sqrt{\det\left(\Sigma_\nabla\right)}}. \tag{9}$$



(a) Gradient-free GP

(b) Gradient-enhanced GP

Fig. 1 GPs with and without gradients that are approximating the function from Eq. (13) with $\beta = -0.62$, $\hat{\sigma}_{\mathsf{K}}^2 = 1.07$, and $\gamma = 1.7$.

Closed form solutions that maximize L can be found for β and, for the noise-free case considered in this paper, $\hat{\sigma}_{\mathsf{K}}^2$:

$$\beta(\boldsymbol{\gamma}; \mathsf{X}, \boldsymbol{f}_\nabla, \eta_{\mathsf{K}_\nabla}) = \frac{\check{\mathbf{1}}^\top \Sigma_\nabla^{-1} \boldsymbol{f}_\nabla}{\check{\mathbf{1}}^\top \Sigma_\nabla^{-1} \check{\mathbf{1}}} \tag{10}$$

$$\hat{\sigma}_{\mathsf{K}}^2(\boldsymbol{\gamma}; \mathsf{X}, \boldsymbol{f}_\nabla, \eta_{\mathsf{K}_\nabla}, \beta) = \frac{\left(\boldsymbol{f}_\nabla - \beta\check{\mathbf{1}}\right)^\top \left(\mathsf{K}_\nabla + \eta_{\mathsf{K}_\nabla}\mathsf{W}\right)^{-1} \left(\boldsymbol{f}_\nabla - \beta\check{\mathbf{1}}\right)}{n_x(n_d+1)}. \tag{11}$$

Substituting the solution for $\hat{\sigma}_{\mathsf{K}}^2$ from Eq. (11) into Eq. (9), applying a natural logarithm, and dropping constant terms gives

$$\ln(L(\boldsymbol{\gamma}; \mathsf{X}, \eta_{\mathsf{K}_\nabla}, \hat{\sigma}_{\mathsf{K}})) = -\frac{n_x(n_d+1)\ln(\hat{\sigma}_{\mathsf{K}}^2) + \ln(\det(\mathsf{K}_\nabla + \eta_{\mathsf{K}_\nabla}\mathsf{W}))}{2}. \tag{12}$$

The function $\ln(L)$ can be efficiently maximized with a gradient-based optimizer since it is smooth and continuous [27, 37].

To highlight the advantage of using gradients to construct a GP, the following one-dimensional function is approximated

$$f(x) = \sin(x) + \sin\left(\frac{10x}{3}\right). \tag{13}$$

This function was evaluated at four points and its hyperparameters were selected by maximizing the marginal log-likelihood from Eqs. (10), (11), and (12). The gradient-free and gradient-enhanced GP can be seen in Figs. 1a and 1b, respectively. It is clear from these figures that the use of gradients to construct the GP provides a significantly more accurate surrogate with lower uncertainty. The benefit of using gradients grows as the dimensionality of the problem increases since they provide more information relative to a single function evaluation. For a gradient-enhanced GP to be effective, it is crucial that the ill-conditioning of its covariance matrix be addressed, which is considered in the following section.

IV. Addressing the ill-conditioning of K_∇

The gradient-free kernel matrix K and especially the gradient-enhanced kernel matrix K_∇ are known to quickly become severely ill-conditioned as the number of evaluation points increases and as they get closer together [21, 39]. The preconditioning method from Marchildon and Zingg [30] to address this ill-conditioning problem is summarized in this section.

A common approach to mitigate this is to regularize the matrix, i.e. to add a nugget $\eta > 0$ to its diagonal [40]. The minimum and maximum eigenvalues of K are $\lambda_{\min}$ and $\lambda_{\max}$, respectively, which are real and nonnegative since the

Algorithm 1 Stable Cholesky decomposition for gradient-enhanced GP

```
Select evaluation points $\mathsf{X}$ and hyperparameters $\boldsymbol{\gamma}$
Calculate $\dot{\mathsf{K}}_\nabla$ with Eq. (16)
Calculate $\eta_{\dot{\mathsf{K}}_\nabla}$ with Eq. (19), or Eq. (18) for the Gaussian kernel
$\dot{\mathsf{L}}\dot{\mathsf{L}}^\top = \dot{\mathsf{K}}_\nabla + \eta_{\dot{\mathsf{K}}_\nabla}\mathsf{I}$
$\mathsf{L} = \mathsf{P}\dot{\mathsf{L}}$, where $\hat{\sigma}_\mathsf{K}^2\mathsf{L}\mathsf{L}^\top = \hat{\sigma}_\mathsf{K}^2\left(\mathsf{K}_\nabla + \eta_{\dot{\mathsf{K}}_\nabla}\mathsf{P}\mathsf{P}\right) = \Sigma_\nabla$
```

matrix is symmetric positive semidefinite [17]. To ensure that $\kappa(\mathsf{K} + \eta_\mathsf{K}\mathsf{I}) \leq \kappa_{\max}$, we select η_K to satisfy the following relation:

$$\kappa(\mathsf{K} + \eta_\mathsf{K}\mathsf{I}) - \frac{\lambda_{\max} + \eta_\mathsf{K}}{\lambda_{\min} + \eta_\mathsf{K}} \leq \frac{\lambda_{\max} + \eta_\mathsf{K}}{\eta_\mathsf{K}} \leq \kappa_{\max}$$

$$\eta_\mathsf{K} \geq \frac{\lambda_{\max}}{\kappa_{\max} - 1}, \tag{14}$$

where the condition number is based on the ℓ_2 norm and $\lambda_{\max}$ is the largest eigenvalue of K. For a kernel that gives $k(\boldsymbol{x}, \boldsymbol{x}; \boldsymbol{\gamma}) = 1$, such as the Gaussian and Matérn $\frac{5}{2}$ kernels from Eqs. (1) and (2), respectively, the diagonal of K is unity and thus its trace is n_x. Since $\sum \lambda_i = \operatorname{tr}(\mathsf{K}) = n_x$ and $\lambda_i \geq 0 \,\forall\, i \in \{1, \ldots, n_x\}$, we have $\lambda_{\max} \leq \operatorname{tr}(\mathsf{K}) = n_x$. From Eq. (14), a sufficient nugget η_K to ensure that $\kappa(\mathsf{K}(\boldsymbol{\gamma}) + \eta_\mathsf{K}\mathsf{I}) \leq \kappa_{\max} \,\forall\, \boldsymbol{\gamma} > 0$ is

$$\eta_{\mathsf{K}_\nabla} = \frac{n_x}{\kappa_{\max} - 1}. \tag{15}$$

For the gradient-enhanced case, this approach to select a nugget value cannot be used on its own to ensure that $\kappa(\mathsf{K}_\nabla(\boldsymbol{\gamma}) + \eta_{\mathsf{K}_\nabla}\mathsf{I}) \leq \kappa_{\max} \,\forall\, \boldsymbol{\gamma} > 0$ since $\operatorname{tr}(\mathsf{K}_\nabla) = n_x(1 + \boldsymbol{\gamma}^T\mathbf{1})$. Consequently, using Eq. (14) to select η_{K_∇} with $\lambda_{\max} \leq \operatorname{tr}(\mathsf{K}_\nabla)$ results in a value of η_{K_∇} that can become arbitrary large if any entry in $\boldsymbol{\gamma}$ is large. To avoid this, the gradient-enhanced kernel matrix can be preconditioned with [24, 30]

$$\dot{\mathsf{K}}_\nabla = \mathsf{P}^{-1}\mathsf{K}_\nabla\mathsf{P}^{-1} \tag{16}$$

$$\mathsf{P} = \operatorname{diag}\left(\sqrt{\operatorname{diag}(\mathsf{K}_\nabla)}\right). \tag{17}$$

Eq. (16) ensures that the diagonal entries of $\dot{\mathsf{K}}_\nabla$ are all unity and thus $\operatorname{tr}\left(\dot{\mathsf{K}}_\nabla\right) = n_x(n_d + 1)$, which is independent of $\boldsymbol{\gamma}$. Eq. (14) could be used to provide a sufficient $\eta_{\dot{\mathsf{K}}_\nabla}$ to ensure that $\kappa(\dot{\mathsf{K}}_\nabla(\boldsymbol{\gamma}) + \eta_{\dot{\mathsf{K}}_\nabla}\mathsf{I}) \leq \kappa_{\max} \,\forall\, \boldsymbol{\gamma} > 0$, but this would give a nugget value that scales as $\eta_{\dot{\mathsf{K}}_\nabla} = \mathcal{O}(n_x n_d)$. Instead, the Gershgorin circle theorem can be used to prove that $\kappa(\dot{\mathsf{K}}_\nabla(\boldsymbol{\gamma}) + \eta_{\dot{\mathsf{K}}_\nabla}\mathsf{I}) \leq \kappa_{\max} \,\forall\, \boldsymbol{\gamma} > 0$ can be ensured for the Gaussian kernel with the following nugget:

$$\eta_{\dot{\mathsf{K}}_\nabla}(n_x, n_d; \kappa_{\max}) = \frac{1 + (n_x - 1)\frac{1+\sqrt{1+4n_d}}{2} e^{-\frac{1+2n_d-\sqrt{1+4n_d}}{4n_d}}}{\kappa_{\max} - 1}, \tag{18}$$

which scales as $\eta_{\dot{\mathsf{K}}_\nabla} = \mathcal{O}(n_x\sqrt{n_d})$ [30]. Eq. (18) is specific to the Gaussian kernel and does not guarantee that $\kappa(\dot{\mathsf{K}}_\nabla(\boldsymbol{\gamma}) + \eta_{\dot{\mathsf{K}}_\nabla}\mathsf{I}) \leq \kappa_{\max} \,\forall\, \boldsymbol{\gamma} > 0$ for all kernels. To ensure the last inequality holds for any kernel, the following nugget can be used [30]:

$$\eta_{\dot{\mathsf{K}}_\nabla}(\hat{\sigma}_\mathsf{K}, \boldsymbol{\gamma}, \eta_{\mathsf{K}_\nabla}, \hat{\sigma}_f, \hat{\sigma}_{\nabla f}; \mathsf{X}, \kappa_{\max}) = \frac{\max_i \sum_{j=1}^{n_x(n_d+1)} \left|\dot{\mathsf{K}}_\nabla\right|_{ij}}{\kappa_{\max} - 1}, \tag{19}$$

which depends on $\boldsymbol{\gamma}$, but for the Gaussian kernel it is bounded from above by Eq. (18). Eq. (19) could also be applied to K in order to provide a smaller nugget value than Eq. (15) while still guaranteeing $\kappa(\mathsf{K}(\boldsymbol{\gamma}) + \eta_\mathsf{K}\mathsf{I}) \leq \kappa_{\max} \,\forall\, \boldsymbol{\gamma} > 0$.

Algorithm 1 provides the five required steps to get a stable Cholesky decomposition of the gradient-enhanced covariance matrix Σ_∇. This algorithm can be used when evaluating the mean and variance of the surrogate with Eqs. (6) and (7), respectively, and when optimizing the hyperparameters with Eqs. (10), (11), and (12). Algorithm 1 considers the case when there is no noise on the function and gradient evaluations. For the case when there are noisy evaluations, see Marchildon and Zingg [30].

V. Bayesian optimization framework for local optimization

A. Optimization problem

Consider the following general constrained optimization problem

$$\begin{aligned} \min_{\underline{\boldsymbol{x}} \le \boldsymbol{x} \le \overline{\boldsymbol{x}}} f(\boldsymbol{x}) \quad \text{subject to} \qquad & \mathsf{A}_g \boldsymbol{x} \le \boldsymbol{b}_g \\ & \mathsf{A}_h \boldsymbol{x} = \boldsymbol{b}_h \\ & g_i(\boldsymbol{x}) \le 0 \;\; \forall\, i \in \{1, \dots, n_{g,\text{nlin}}\} \\ & h_i(\boldsymbol{x}) = 0 \;\; \forall\, i \in \{1, \dots, n_{h,\text{nlin}}\}, \end{aligned} \tag{20}$$

where $\underline{\boldsymbol{x}}$ and $\overline{\boldsymbol{x}}$ are the lower and upper bounds on the design variables $\boldsymbol{x}$, respectively, for the linear constraints we have $\mathsf{A}_g \in \mathbb{R}^{n_{g,\text{lin}} \times n_d}$, $\boldsymbol{b}_g \in \mathbb{R}^{n_{g,\text{lin}}}$, $\mathsf{A}_h \in \mathbb{R}^{n_{h,\text{lin}} \times n_d}$, $\boldsymbol{b}_h \in \mathbb{R}^{n_{h,\text{lin}}}$, and $\boldsymbol{g}(\boldsymbol{x}) \in \mathbb{R}^{n_{g,\text{nlin}}}$ and $\boldsymbol{h}(\boldsymbol{x}) \in \mathbb{R}^{n_{h,\text{nlin}}}$ are the nonlinear inequality and equality constraints, respectively.

To track the progress of constrained optimizations we use a merit function and the optimality. First, we consider the Lagrangian, which is given by

$$\mathcal{L} = f(\boldsymbol{x}) + \boldsymbol{\psi}_g^\top \boldsymbol{g}_{\text{all}} + \boldsymbol{\psi}_h^\top \boldsymbol{h}_{\text{all}}, \tag{21}$$

where $\boldsymbol{g}_{\text{all}}$ and $\boldsymbol{h}_{\text{all}}$ are vectors holding all of the inequality and equality constraints, respectively, i.e. the bound, linear and nonlinear constraints from Eq. (20), while $\boldsymbol{\psi}_g$ and $\boldsymbol{\psi}_h$ are the Lagrange multipliers for the inequality and equality constraints, respectively [14]. The necessary first-order optimality conditions for a solution $\boldsymbol{x}^*$ to be a local solution to Eq. (20) are $\frac{\partial \mathcal{L}}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}^*} = \boldsymbol{0}$, $\boldsymbol{g}(\boldsymbol{x}^*) \le \boldsymbol{0}$, $\boldsymbol{h}(\boldsymbol{x}^*) = \boldsymbol{0}$, $\boldsymbol{\psi}_g \ge \boldsymbol{0}$, and $\psi_{g_i} g_i = 0 \, \forall\, i \in \{1, \dots, n_{g,\text{nlin}}\}$ [14].

Our merit function is given by

$$m(\boldsymbol{x}) = f(\boldsymbol{x}) + \boldsymbol{\psi}_g^\top \boldsymbol{g}_{\text{all}} + \boldsymbol{\psi}_h^\top \boldsymbol{h}_{\text{all}} + \rho \left(\|\boldsymbol{g}^+\|_2^2 + \|\boldsymbol{h}\|_2^2 \right), \tag{22}$$

which is the Lagrangian along with a quadratic penalty term with $\rho = 10$ and $g_i^+ = \max(g_i, 0) \, \forall\, i \in \{1, \dots, n_{g,\text{nlin}}\}$. The optimality and feasibility are calculated with

$$\phi_{\text{opt}}(\boldsymbol{x}) = \left\| \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} + \boldsymbol{\psi}_h^\top \frac{\partial \boldsymbol{h}_{\text{all}}(\boldsymbol{x})}{\partial \boldsymbol{x}} \boldsymbol{\psi}_g^\top \frac{\partial \boldsymbol{g}_{\text{all}}(\boldsymbol{x})}{\partial \boldsymbol{x}} \right\|_2 \tag{23}$$

$$\phi_{\text{fsb}}(\boldsymbol{x}) = \sum_{i=1}^{n_h} |h_i(\boldsymbol{x})| + \sum_{i=1}^{n_g} |\max(g_i(\boldsymbol{x}), 0)|, \tag{24}$$

where the optimality is the ℓ_2 norm of the gradient of the Lagrangian with respect to the design variables. The Lagrange multipliers are selected by solving a constrained least-squares problem that seeks to minimize the optimality with the constraint $\boldsymbol{\psi}_g \ge \boldsymbol{0}$, which comes from the first-order optimality conditions.

B. Data region

For global optimization, which is the class of problems that Bayesian optimizers are most commonly applied to, all of the evaluation points are typically used to construct a surrogate that can be evaluated across the entire design space [15, 41]. However, for local optimization the surrogate only needs to be accurate in the region around one local minimum. The maximization of the marginal log-likelihood from Eq. (12) using all of the evaluation points results in hyperparameters that provide a surrogate that is reasonably accurate around all evaluation points. However, the surrogate can be more accurate near the local minimum when only evaluation points near the local minimum are used. A consequence of this is that the surrogate is not as accurate near evaluation points far from the local minimum, which is not problematic for local optimization.

A data region is used to select the evaluation points that are used to maximize the marginal log-likelihood to select the hyperparameters and to evaluate the posterior of the GP. The objective and gradient evaluations from all of the evaluation points in the data region are used. The data region includes the three most recent evaluation points along with the 20 evaluation points with the shortest Euclidean distance to $\boldsymbol{x}_{\text{best}}$, which is the evaluation point with the lowest merit function evaluation from Eq. (22). If too few evaluation points are used, the resulting surrogate will only be accurate in a very small region around $\boldsymbol{x}_{\text{best}}$. On the other hand, if too many evaluation points are kept in the data region, then the resulting surrogate is less accurate around $\boldsymbol{x}_{\text{best}}$. The inclusion of the 20 closest points was found to be a good compromise between these two factors.

It is important to contrast the differing goals of the data region and of the method that reduces the number of evaluation points to mitigate the ill-conditioning of the covariance matrix [28]. The data region selects evaluation points to get an accurate local surrogate, while the only consideration of the latter method is to have the condition number of the covariance matrix below a set threshold. This results in dropping evaluation points that are clustered together, which occurs near a local minimum, since points that are close to each other make the ill-conditioning problem worse [39].

Another benefit of using the data region is that it reduces the computational cost of selecting the hyperparameters and of evaluating the posterior of the GP. Each time the hyperparameters are changed, a new Cholesky decomposition is required for the gradient-enhanced covariance matrix Σ_∇, which has a cost that scales as $O\left(n_{\text{data}}^3 (n_d + 1)^3\right)$, where n_{data} is the number of evaluation points in the data region. Once the Cholesky decomposition has been performed, the marginal cost of evaluating the posterior of the GP is $O\left(n_{\text{data}}^2 (n_d + 1)^2\right)$. Therefore, using a data region with $n_{\text{data}} \ll n_x$ can significantly reduce the computational cost of selecting the hyperparameters and of evaluating the posterior of the GP.

C. Trust region

The use of trust regions is common for local optimizers since it helps ensure that the merit function is reduced after each function evaluation. A trust region can also be used with a Bayesian optimizer to avoid evaluating the posterior of the GP in regions where it has a large uncertainty.

For quasi-Newton optimizers the objective is approximated with a quadratic model, the constraints are typically linearized [14], and the trust region is usually a hypersphere around $\boldsymbol{x}_{\text{best}}$:

$$\begin{aligned} g_{\text{trc}}(\boldsymbol{x}; \boldsymbol{x}_{\text{best}}) &= \|\boldsymbol{x} - \boldsymbol{x}_{\text{best}}\|_2^2 \qquad (25) \\ &\leq \overline{g}_{\text{trc}}^j, \end{aligned}$$

where $\overline{g}_{\text{trc}}^j$ is the maximum squared Euclidean distance around $\boldsymbol{x}_{\text{best}}$ that satisfies the trust region at the j-th optimization iteration. The uncertainty estimate of the GP can also be used to construct a trust region:

$$\begin{aligned} g_{\text{tr}\sigma}(\boldsymbol{x}; \sigma_{\text{GP},f}, \hat{\sigma}_{\mathsf{K},f}) &= \frac{\sigma_{\text{GP},f}^2(\boldsymbol{x})}{\hat{\sigma}_{\mathsf{K},f}^2} \qquad (26) \\ &= \left(1 - \boldsymbol{k}_\nabla^\top \left(\mathsf{K}_\nabla + \eta_{\mathsf{K}_\nabla} \mathsf{W}\right)^{-1} \boldsymbol{k}_\nabla\right) \qquad (27) \\ &\leq \overline{g}_{\text{tr}\sigma}^j, \end{aligned}$$

where $\overline{g}_{\text{tr}\sigma}^j$ is the upper bound for this trust region, and the second equality holds for all kernels that satisfy $k(\boldsymbol{x}, \boldsymbol{x}) = 1$ and when there are no noisy function or gradient evaluations. Since $\mathsf{K}_\nabla + \eta_{\mathsf{K}_\nabla} \mathsf{W}$ is positive definite, we have $0 \leq g_{\text{tr}\sigma}(\boldsymbol{x}) \leq 1$ and thus the upper bound should be set to $0 < \overline{g}_{\text{tr}\sigma}^j < 1 \, \forall \, j \in \mathbb{Z}_+$. The circular trust region from Eq. (25) is used at all iterations. On the other hand, the probabilistic surrogate from Eq. (26) is only used once there have been at least 10 function evaluations since the uncertainty estimate from the posterior of the GP is not accurate if there are too few function evaluations.

The upper bounds $\overline{g}_{\text{trc}}^j$ and $\overline{g}_{\text{tr}\sigma}^j$ for the two trust regions change depending on whether the optimizer is making progress or not, i.e. whether the merit function is being reduced. The upper bounds for the trust regions are increased by a factor of two if progress was made at the last iteration, kept constant if progress was made at the second last iteration but not the last one, and they are reduced if no progress was made during the last two iterations.

D. Hidden constraints

For aerodynamic shape optimization, the numerical optimizer returns new values of the design variables that control the shape of the geometry. However, certain geometries can cause mesh movement failures or can result in the flow solver not converging. In these cases, there are no function or gradient evaluations that are returned to the optimizer for the objective function or for the nonlinear constraints. It is generally not possible to explicitly define a constraint that determines when these mesh movement or flow solve failures will occur. Therefore, these are commonly referred to as hidden constraints [42, 43]. These are binary constraints since the only information known about them is whether or not they are satisfied at an evaluation point. This makes them challenging to handle since the numerical optimizer does not know if a small change in the variables will result in these hidden constraints being satisfied or not. Furthermore, the lack of function and gradient evaluations for the objective and nonlinear constraints prevents the surrogates that are approximating these functions from being updated when these hidden constraints are not satisfied.

Support vector machines have previously been used to enable a Bayesian optimizer to handle binary constraints [44]. Another approach that has been used to handle hidden constraints is to model them using a radial basis function [42], or a GP [45]. The only data available to select the hyperparameters of the surrogate approximating them is the sign of the hidden constraints at the evaluation points, i.e. -1 if they are not satisfied and $+1$ otherwise. A recent method that was used to handle hidden constraints for a Bayesian optimizer applied to the aerodynamic shape optimization of an aircraft was a k-nearest neighbour classifier [43].

Here the hidden constraints are handled by adding a hypersphere around each of the points where the hidden constraints are not satisfied. These points are denoted as $\check{\boldsymbol{x}}$ and the constraint is given by

$$g_{\check{\boldsymbol{x}}_i}(\boldsymbol{x};\check{\boldsymbol{x}}_i) = \|\check{\boldsymbol{x}}_i - \boldsymbol{x}\|_2^2 \geq \underline{g}_{\check{\boldsymbol{x}}_i} \quad \forall\, i \in \{1, \ldots, n_{\check{\boldsymbol{x}}}\}, \tag{28}$$

where $n_{\check{\boldsymbol{x}}}$ is the number of evaluation points where the hidden constraints are not satisfied and $\underline{g}_{\check{\boldsymbol{x}}_i}$ is the lower bound for the constraint. The squared Euclidean distance is used, just like the circle trust region from Eq. (25), since its gradient is always well defined. However, the constraint around $\check{\boldsymbol{x}}$ uses a lower bound on the hypersphere, unlike the circle trust region $g_{\text{trc}}(\boldsymbol{x})$, which uses an upper bound. The lower bound $\underline{g}_{\check{\boldsymbol{x}}_i}$ is set to 60% of the Euclidean distance between $\check{\boldsymbol{x}}_i$ and the closest evaluation point $\boldsymbol{x}$ that satisfies the hidden constraints. This method was used to handle the hidden constraints since it is simple to implement, inexpensive to evaluate, and it was found to be effective with the local optimization framework that was developed.

E. Acquisition function minimization

The acquisition function is minimized to determine the next point in the design space where the objective and the nonlinear constraints will be evaluated along with their gradients. This function is intended to balance exploration and exploitation. The upper confidence and expected improvement are popular choices for acquisition functions and are given by

$$q_{\text{UC}}(\boldsymbol{x};\omega) = \mu_{\text{GP,f}}(\boldsymbol{x}) - \omega\sigma_{\text{GP},f}(\boldsymbol{x}) \tag{29}$$

$$\begin{aligned} q_{\text{EI}}(\boldsymbol{x}; f_{\text{best}}) &= \int_{-\infty}^{f_{\text{best}}} (f_{\text{best}} - f)\theta_{\text{pdf}}\left(\frac{f - \mu_f(\boldsymbol{x})}{\sigma_f(\boldsymbol{x})}\right) df \\ &= \left(f_{\text{best}} - \mu_f(\boldsymbol{x})\right)\theta_{\text{cdf}}\left(\frac{f_{\text{best}} - \mu_f(\boldsymbol{x})}{\sigma_f(\boldsymbol{x})}\right) + \sigma_f(\boldsymbol{x})\theta_{\text{pdf}}\left(\frac{f_{\text{best}} - \mu_f(\boldsymbol{x})}{\sigma_f(\boldsymbol{x})}\right), \end{aligned} \tag{30}$$

where $\omega \geq 0$ promotes exploration when it is larger, $f_{\text{best}} = f(\boldsymbol{x}_{\text{best}})$, while $\theta_{\text{pdf}}(\cdot)$ and $\theta_{\text{cdf}}(\cdot)$ are the Gaussian probability and cumulative distribution functions, respectively.

It is common for nonlinear constrained Bayesian optimization to use a separate GP to approximate each of the individual nonlinear constraints in addition to the objective function. A popular approach to handle the nonlinear constraints is to use the expected improvement acquisition function with the probability of feasibility [31, 34, 46].

Two drawbacks of this method are that it requires the initial solution to be feasible and it cannot handle nonlinear equality constraints since in both cases the probability of feasibility will be zero. For aerodynamic shape optimization nonlinear equality constraints are common, e.g. a lift target. To be able to solve Eq. (20), which includes nonlinear inequality and equality constraints, the following constrained optimization formulation is used for the minimization of the acquisition function:

$$
\begin{aligned}
\boldsymbol{x}_{\text{next}} = \underset{\underline{\boldsymbol{x}} \leq \boldsymbol{x} \leq \overline{\boldsymbol{x}}}{\operatorname{argmin}} \; q(\boldsymbol{x}) \quad \text{subject to} \qquad\qquad & \mathsf{A}_g \boldsymbol{x} \leq \boldsymbol{b}_g \\
& \mathsf{A}_h \boldsymbol{x} = \boldsymbol{b}_h \\
& g_{\text{trc}}(\boldsymbol{x}) \leq \overline{g}_{\text{trc}}^{j} \\
& g_{\text{tr}\sigma}(\boldsymbol{x}) \leq \overline{g}_{\text{tr}\sigma}^{j} \\
& g_{\check{\boldsymbol{x}}_i}(\boldsymbol{x}; \check{\boldsymbol{x}}_i) \geq \underline{g}_{\check{\boldsymbol{x}}_i} \quad \forall\, i \in \{1, \ldots, n_{\check{\boldsymbol{x}}}\} \\
& \mu_{\text{GP},g_i}(\boldsymbol{x}) \leq 0 \quad \forall\, i \in \{1, \ldots, n_{g,\text{nlin}}\} \\
& \mu_{\text{GP},h_i}(\boldsymbol{x}) = 0 \quad \forall\, i \in \{1, \ldots, n_{h,\text{nlin}}\},
\end{aligned}
\tag{31}
$$

where $\boldsymbol{x}_{\text{next}}$ is the next point in the design space where the objective and nonlinear constraints will be evaluated, the trust regions $g_{\text{trc}}(\boldsymbol{x})$ and $g_{\text{tr}\sigma}(\boldsymbol{x})$ are obtained from Eqs. (25) and (26), respectively, and Eq. (28) provides $g_{\check{\boldsymbol{x}}_i}(\boldsymbol{x})$. The nonlinear constraints are enforced by providing the mean of the posterior of the GPs approximating the nonlinear constraints and setting the bounds to be the same as the nonlinear constraints from the original optimization problem from Eq. (20). The means of the posteriors of the GPs approximating the nonlinear constraints are used since they are inexpensive to evaluate and should be a good approximation to the nonlinear constraints in the trust regions.

After Eq. (31) is solved, the objective and nonlinear constraints are evaluated at $\boldsymbol{x}_{\text{next}}$ along with their gradients. A new data region is selected as detailed in Section V.B, the hyperparameters are updated by maximizing the marginal log-likelihood from Eq. (12), the upper bounds for the circular and trust regions are updated depending on whether progress was made during the last two iterations or not, and then Eq. (31) is solved once again. This process is repeated until the desired convergence criteria are achieved.

VI. Aerodynamic shape optimization

A. Methodology and problem definition

For our aerodynamic shape optimization study we use Jetstream, which has been developed at the University of Toronto [47, 48]. The computational fluid dynamics simulations come from Diablo, which solves the Reynolds-averaged Navier–Stokes (RANS) equations with the Spalart-Allmaras one-equation turbulence model [49, 50]. Second-order summation-by-parts operators are used for the spatial discretization on a multiblock structured mesh, and a parallel Newton-Krylov-Schur algorithm is used to converge to a steady state. Geometry control is achieved with the use of free-form deformations and B-splines, and the mesh deformation is handled with an efficient linear elasticity model [47]. Gradients for the objective and nonlinear constrains are calculated in Jetstream using the discrete adjoint method [47]. Jetstream has historically used the gradient-based optimizer SNOPT, which uses a sequential quadratic programming method with the Hessian approximated using the BFGS update formula [51]. A cross validation of Jetstream and the in-house aerodynamic shape optimization framework at Bombardier demonstrates Jetstream's effectiveness for complex aerodynamic shape optimization problems [48].

Lift-constrained drag minimization of an airfoil at transonic speed was performed using Jetstream along with the Bayesian optimizer presented here and its performance is compared to the performance of SNOPT. The initial geometry is the RAE 2822 airfoil. Due to a limitation in the mesh movement implementation for two-dimensional cases, the airfoil needed to be extruded to three dimensions for this case. A single block was used in the extruded direction with 11 nodes, and symmetry boundary conditions were applied. The three-dimensional mesh had a total of 292 248 nodes, 26 568 nodes for each two-dimensional slice around the airfoil. The three-dimensional RANS equations were solved with the Spalart-Allmaras one-equation turbulence model at a (two-dimensional) Mach number of 0.74, a Reynolds number of 9×10^6, and the flow was assumed to be fully turbulent.

Two cross sections with design variables were used, which were linked together with linear equality constraints. Each cross section has six design variables on the top, and six on the bottom that control the thickness of the airfoil. This makes for a total of 25 design variables with the angle of attack. Bound constraints were used but none of them were active at the solution. Linear inequality constraints were used to ensure that the thickness of the airfoil at each design point was no smaller than 0.25 times and no greater than 4.0 times the initial thickness of the airfoil. Similar to the bound constraints, none of these linear inequality constraints were active at the solution. A nonlinear equality constraint was included to set the coefficient of lift C_{L} to 0.78, and a nonlinear inequality constraint was used to ensure that the optimized geometry has a cross-sectional area g_{area} that is no smaller than that of the initial geometry. Both nonlinear constraints were active at the solution. The objective to be minimized is the coefficient of drag C_{D}, and the optimization problem is given by

$$\begin{aligned} \min_{\underline{\boldsymbol{x}} \le \boldsymbol{x} \le \overline{\boldsymbol{x}}} C_{\text{D}} \quad \text{subject to} \quad & \mathsf{A}_g \boldsymbol{x} \le \boldsymbol{b}_g \\ & \mathsf{A}_h \boldsymbol{x} = \boldsymbol{b}_h \\ & C_{\text{L}} = 0.78 \\ & g_{\text{area}} \ge 0.0778. \end{aligned} \tag{32}$$

B. Dimension reduction with linear equality constraints

By extruding the airfoil to a third dimension the number of design variables is nearly doubled, from 13 to 25, where the angle of attack is the only design variable that is not impacted. Linear equality constraints are used such that the design variables at the same location on the two cross sections are equal. This does not impact the final optimized shape of the airfoil, but it does significantly impact the performance of the Bayesian optimizer. It is shown in Appendix A.A that increasing the number of design variables results in weaker correlations between evaluation points and thus greater uncertainty in the surrogate.

The linear equality constraints can be used to reduce the number of free variables. We begin by assuming that the matrix A_h for the set of linear equality constraints $\mathsf{A}_h \boldsymbol{x} = \boldsymbol{b}_h$ has full row rank and that $n_{h,\text{lin}} < n_d$, i.e. there are fewer linear equality constraints than design variables. If the rank of A_h is less than $n_{h,\text{lin}}$, then there are redundant constraints that can be removed. A general solution that satisfies the linear equality constraints is given by

$$\boldsymbol{x} = \boldsymbol{x}_p + \mathsf{Z}_h \tilde{\boldsymbol{x}}, \tag{33}$$

where $\boldsymbol{x}_p$ is a non-unique solution (if $n_{h,\text{lin}} < n_d$) that satisfies the linear equality constraints, $\tilde{\boldsymbol{x}} \in \mathbb{R}^{n_d - n_{h,\text{lin}}}$ is the solution vector in the reduced dimension space, and $\mathsf{Z}_h \in \mathbb{R}^{n_d \times (n_d - n_{h,\text{lin}})}$ spans the null space of A_h, i.e. $\mathsf{A}_h \mathsf{Z}_h = \mathbb{O}$. The null space matrix Z_h is non-unique and its selection is described in Appendix A.B. One method of calculating $\boldsymbol{x}_p$ is with

$$\boldsymbol{x}_p = \mathsf{A}_h^{+} \boldsymbol{b}_h, \tag{34}$$

where A_h^{+} is the Moore-Penrose pseudoinverse of A_h. The optimization of the nonlinearly constrained optimization problem from Eq. (20) in the lower-dimensional problem is given by

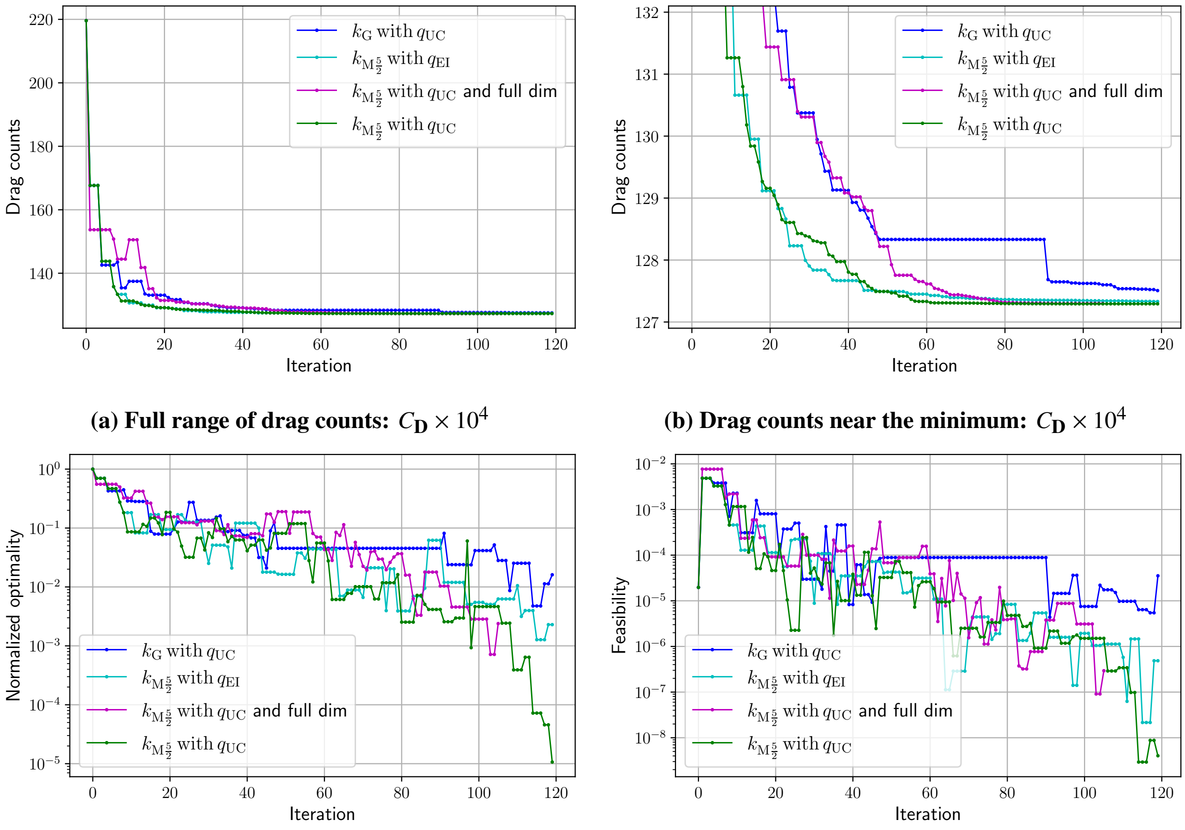
$$\begin{aligned} \min_{\tilde{\boldsymbol{x}}} f(\tilde{\boldsymbol{x}}) \quad \text{subject to} \quad & \tilde{\mathsf{A}}_g \tilde{\boldsymbol{x}} \le \tilde{\boldsymbol{b}}_g \\ & g_{\text{trc}}(\boldsymbol{x}) \le \overline{g}_{\text{trc}}^{j} \\ & g_{\text{tr}\sigma}(\boldsymbol{x}) \le \overline{g}_{\text{tr}\sigma}^{j} \\ & g_{\check{\boldsymbol{x}}_i}(\tilde{\boldsymbol{x}}; \check{\boldsymbol{x}}_i) \ge \underline{g}_{\check{\boldsymbol{x}}_i} \quad \forall i \in \{1, \ldots, n_{\check{\boldsymbol{x}}}\} \\ & \mu_{\text{GP},g_i}(\tilde{\boldsymbol{x}}) \le 0 \quad \forall i \in \{1, \ldots, n_{g,\text{nlin}}\} \\ & \mu_{\text{GP},h_i}(\tilde{\boldsymbol{x}}) = 0 \quad \forall i \in \{1, \ldots, n_{h,\text{nlin}}\}, \end{aligned} \tag{35}$$

where

$$\tilde{\mathsf{A}}_g = \begin{bmatrix} \mathsf{Z}_h \\ -\mathsf{Z}_h \\ \mathsf{A}_g \mathsf{Z}_h \end{bmatrix}, \quad \tilde{\boldsymbol{b}}_g = \begin{bmatrix} \overline{\boldsymbol{x}} - \boldsymbol{x}_p \\ \boldsymbol{x}_p - \underline{\boldsymbol{x}} \\ \boldsymbol{b}_g - \mathsf{A}_g \boldsymbol{x}_p \end{bmatrix}.$$

Appendix A.C derives $\tilde{\mathsf{A}}_g$ and $\tilde{\boldsymbol{b}}_g$ by applying a linear transformation to the linear inequality constraints and the bound constraints. The gradient of the objective function and nonlinear constraints with respect to $\tilde{\boldsymbol{x}}$ is also provided in Appendix A.C.

For the aerodynamic shape optimization problem from Eq. (32) there are 14 linear equality constraints, 12 that relate the two cross sections and the last two relate the first and last design variables on the top and bottom of the cross sections. The optimization problem from Eq. (35) therefore has $n_d - n_{h,\text{lin}} = 11$ design variables, while Eq. (20) has $n_d = 25$ design variables.



(a) Full range of drag counts: $C_\mathbf{D} \times 10^4$

(b) Drag counts near the minimum: $C_\mathbf{D} \times 10^4$

(c) Normalized optimality from Eq. (23)

(d) Feasibility from Eq. (24)

Fig. 2 Convergence history for the Bayesian optimizer with different settings for the constrained aerodynamic shape optimization of a transonic airfoil. The Bayesian optimizer uses the Gaussian $k_\mathbf{G}$ and Matérn $\frac{5}{2}$ $k_{\mathbf{M}\frac{5}{2}}$ kernels from Eqs. (1) and (2), respectively. The acquisition functions $q_\mathbf{UC}$ and $q_\mathbf{EI}$ come from Eqs. (29) and (30), respectively, and the dimension reduction from Section VI.B is used for all cases except the one in magenta.

C. Optimization results

Fig. 2 shows the optimization results for the Bayesian optimizer with different settings to solve Eq. (32). Figs. 2a and 2b plot the drag count ($C_\mathrm{D} \times 10^4$) for the entire optimization process and the final five drag count reduction, respectively. The optimality and feasibility are shown in Figs. 2c and 2d, respectively. For all of these plots, the value shown at each iteration is from the iteration with the lowest merit function evaluated thus far. The initial geometry has a drag count of 220, i.e. $C_\mathrm{D} \times 10^4$, which is reduced below 160 in the first five iterations for all settings of the Bayesian optimizer. For all of the cases, except for the data in magenta, the linear equality constraints are used to reduce the dimensionality of the optimization, as described in Section VI.B. The Bayesian optimizer using the Matérn $\frac{5}{2}$ kernel and the upper confidence acquisition function with dimension reduction achieves the deepest convergence, as shown by the green data in Fig. 2c.

The blue line in Fig. 2b shows that the use of the Gaussian kernel significantly slows down the progress of the Bayesian optimizer compared to when the Matérn $\frac{5}{2}$ kernel is used. The twice continuously differentiable Matérn $\frac{5}{2}$ kernel is thought to provide a better approximation to the highly nonlinear objective function than with the infinitely differentiable Gaussian kernel.

In Fig. 2b the magenta case is the only one for which the linear equality constraints were not used to reduce the number of design variables. Comparing the magenta case to the one in green, which uses the dimension reduction technique along with the same kernel and acquisition function as the magenta case, we can see that the Bayesian optimizer makes faster progress at reducing the drag when the number of design variables is reduced.

The expected improvement acquisition function, which is shown in cyan in Fig. 2, promotes more exploration than the upper confidence acquisition function, which is shown in green. This is beneficial for the first 60 iterations when

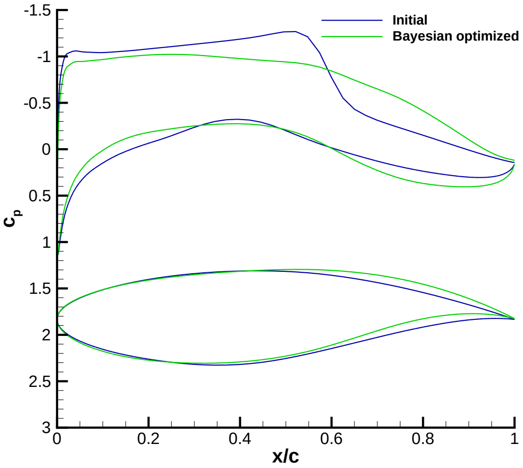


Fig. 3 The airfoil and coefficient of pressure for the initial RAE 2822 airfoil and the optimized airfoil using the Bayesian optimizer with the Matérn $\frac{5}{2}$ kernel with dimension reduction and the upper confidence acquisition function.

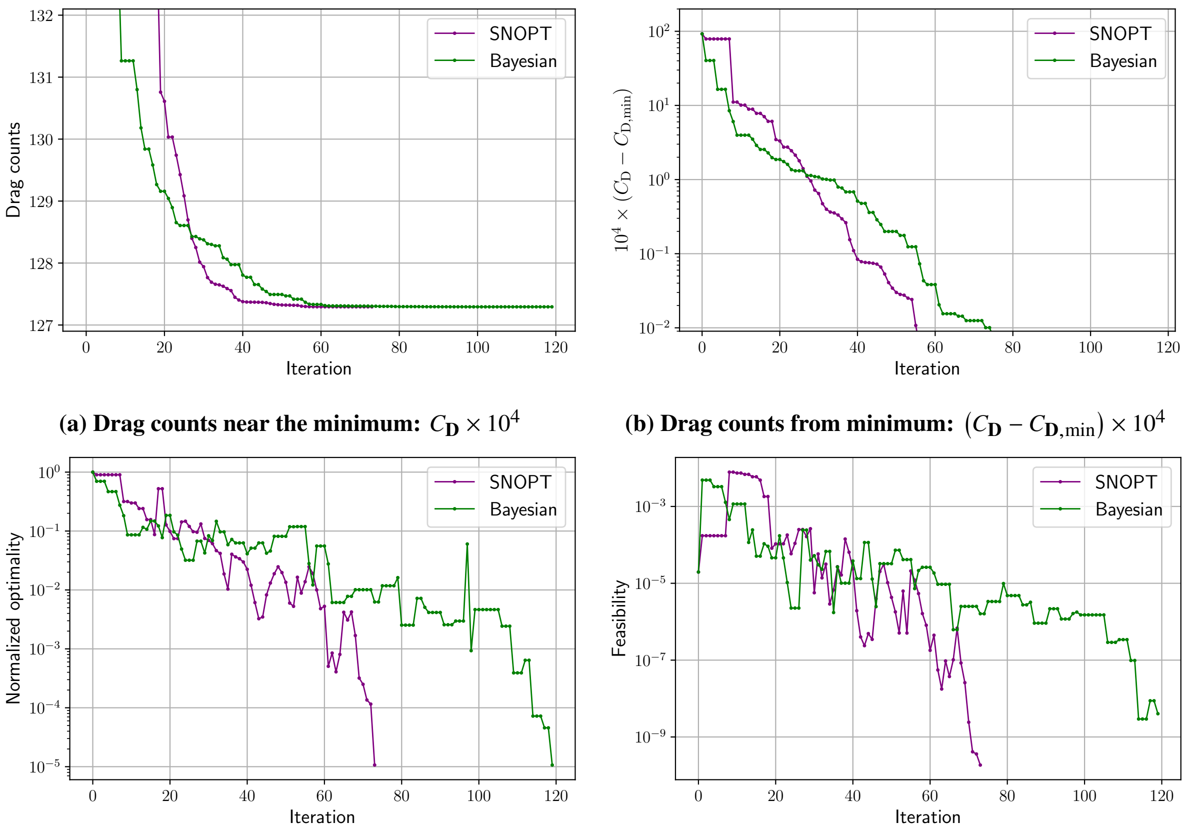
there are significant reductions in the drag. However, this additional exploration impedes the Bayesian optimizer in reducing the final few drag counts and in converging the optimality when it is near the minimum, unlike the upper confidence acquisition function.

Fig. 3 shows the shape and pressure coefficient of the initial and optimized airfoil using the Bayesian optimizer with the Matérn $\frac{5}{2}$ kernel, the upper confidence acquisition function, and the dimension reduction from Section VI.B. The converged optimization runs for the Bayesian optimizer with its different settings all provided final solutions at the same local minimum, i.e. the same optimized airfoil geometry. The primary source of the drag reduction is seen to be the elimination of the shock wave on the upper surface of the airfoil.

Fig. 4 compares the Bayesian optimizer with the upper confidence acquisition function, the dimension reduction from Section VI.B, and the Matérn $\frac{5}{2}$ kernel to the quasi-Newton optimizer SNOPT [51]. Fig. 4a shows the reduction of the final five drag counts while Fig. 4b plots the remaining reduction of the drag on a log plot to achieve $C_{\text{D,min}} = 127.29 \times 10^{-4}$, which is the value that both optimizers converged to. The results in these figures show that the Bayesian optimizer initially makes faster progress at reducing the drag, and gets passed by SNOPT for the reduction of the final 1.4 drag counts. The Bayesian optimizer remains competitive with SNOPT for the final drag count reduction and takes only 7 additional iterations to get within 0.02 of the final drag count. Quasi-Newton optimizers are known to converge quickly once they are near a minimum [14]. With additional development, the convergence of the Bayesian optimizer near a minimum could be accelerated. Figs. 4c and 4d show that both the Bayesian and SNOPT optimizers achieve a final optimality reduction of five orders of magnitude and a feasibility below 10^{-8}. The final airfoil geometry for SNOPT and the Bayesian optimizer are indistinguishable from each other.

VII. Conclusions

In order to complement the capabilities of Bayesian optimization with respect to multimodal problems and problems with inaccurate gradients, an efficient gradient-enhanced Bayesian optimizer has been developed for local nonlinearly constrained optimization. The ill-conditioning of the gradient-enhanced covariance matrix was addressed with a recently developed simple but effective preconditioning method. To enable efficient local optimization, a data region was used to select a subset of evaluation points that provides a more accurate surrogate near the local minimum. Furthermore, the Bayesian optimizer uses two trust regions, one of which leverages the probabilistic surrogate to limit the exploration to regions where the surrogates for the objective and nonlinear constraints are accurate. The means of the posteriors of the GPs approximating the nonlinear constraints are provided to the acquisition function minimizer as nonlinear constraints, enabling nonlinear inequality and equality constraints to be handled.



(a) Drag counts near the minimum: $C_\mathbf{D} \times 10^4$

(b) Drag counts from minimum: $(C_\mathbf{D} - C_\mathbf{D,min}) \times 10^4$

(c) Normalized optimality from Eq. (23)

(d) Feasibility from Eq. (24)

Fig. 4 Comparison of the Bayesian and SNOPT optimizers for the nonlinearly constrained optimization of Eq. (32) for an airfoil at transonic speeds. The Bayesian optimizer uses the dimension reduction from Section VI.B, the upper confidence acquisition function, and the Matérn $\frac{5}{2}$ kernel. Both optimizers converge to a solution with a final drag coefficient of $C_\mathbf{D,min} = 127.29 \times 10^{-4}$.

The Bayesian optimizer developed was applied to nonlinearly constrained aerodynamic shape optimization of an airfoil at transonic speed with the flow governed by the Reynolds-averaged Navier-Stokes equations and gradients computed via the discrete adjoint method. Using the linear equality constraints to reduce the number of design variables was found to improve the performance of the Bayesian optimizer. Evaluation points are farther from each other in a higher-dimensional design space, resulting in lower correlations between evaluation points, and thus greater uncertainty in the posterior of the GPs. The Bayesian optimizer was found to be more effective when it used the Matérn $\frac{5}{2}$ kernel instead of the Gaussian kernel. Finally, it was found that using the expected improvement acquisition function was competitive initially with the use of the upper confidence acquisition function. However, the expected improvement acquisition function promotes more exploration, which was found to impede the deep convergence of the Bayesian optimizer relative to when it used the upper confidence acquisition function.

The Bayesian optimizer with the Matérn $\frac{5}{2}$ kernel and upper confidence acquisition function was compared to the quasi-Newton optimizer SNOPT. The Bayesian optimizer is initially faster than SNOPT at reducing the drag while SNOPT is faster at reducing the final 1.4 drag counts. Both optimizers achieve an optimality reduction of five orders of magnitude and a feasibility below 10^{-8}. These results demonstrate that gradient-enhanced Bayesian optimization can be competitive with a popular quasi-Newton optimizer that has been extensively used for this class of problems. Further enhancements of gradient-enhanced Bayesian optimizers for nonlinearly constrained local optimization should enable faster convergence once the optimizer is near a minimum. With the developments presented here, Bayesian optimization provides a versatile option applicable to a range of aerodynamic shape optimization problems, including problems that are unimodal or multimodal, and problems where accurate gradients are not available, such as chaotic flows.

A. Dimension reduction with linear equality constraints

A. Impact of a higher-dimensional design space on correlations

To satisfy the linear equality constraints, the design variables at the same location on the two cross sections must be equal. The derivative of the objective and nonlinear constraints with respect to each paired design variable will be equal since there is no spanwise variation in the shape of the extruded airfoil. The use of design variables on two cross sections linked with linear equality constraints will not impact the final solution but it does have a significant impact on the performance of the Bayesian optimizer. To demonstrate this, consider a problem with $2m$ parameters, where there are linear equality constraints equating parameter i and $i+m$ and $\frac{\partial f}{\partial x_i} = \frac{\partial f}{\partial x_{i+m}} \,\forall\, i \in \{1, \ldots, m\}$. The symmetry of this problem ensures that the hyperparameters that maximize the marginal log-likelihood will satisfy $\gamma_i = \gamma_{i+m} \,\forall\, i \in \{1, \ldots, m\}$. For the Gaussian kernel from Eq. (1) we have

$$\begin{aligned} k(\boldsymbol{x}, \boldsymbol{y}; n_d = 2m) &= e^{-\frac{1}{2}\sum_{i=1}^{2m} \gamma_i^2 (x_i - y_i)^2} \\ &= e^{-2\left(\frac{1}{2}\sum_{i=1}^{m} \gamma_i^2 (x_i - y_i)^2\right)} \\ &= \left(k(\boldsymbol{x}', \boldsymbol{y}'; n_d = m)\right)^2, \end{aligned}$$

where $\boldsymbol{x}'$ and $\boldsymbol{y}'$ hold the first m entries of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. The Gaussian and Matérn $\frac{5}{2}$ kernels from Eqs. (1) and (2), respectively, satisfy $0 \leq k(\boldsymbol{x}, \boldsymbol{y}) < 1 \,\forall\, \boldsymbol{x} \neq \boldsymbol{y}$. Therefore, for $\boldsymbol{x}' \neq \boldsymbol{y}'$ we have

$$\begin{aligned} k(\boldsymbol{x}, \boldsymbol{y}; n_d = 2m) &= \left(k(\boldsymbol{x}', \boldsymbol{y}'; n_d = m)\right)^2 \\ &< k(\boldsymbol{x}', \boldsymbol{y}'; n_d = m). \end{aligned}$$

This demonstrates that having $2m$ parameters with m linear equality constraints results in weaker correlations than a problem with only m parameters and no linear equality constraints. From a geometric perspective, evaluation points in a $2m$-dimensional space satisfying the m linear equality constraints are $\sqrt{2}$ times farther from each other than in an equivalent m-dimensional parameter space without the linear equality constraints. The weaker correlations create additional uncertainty in the posterior of the GPs approximating the objective function and nonlinear constraints. This results in the Bayesian optimizer taking shorter steps in the parameter space, thus slowing down the optimization.

B. Selecting Z_h

The null space of A_h that satisfies $\mathsf{A}_h \mathsf{Z}_h = \mathbb{0}$ is not unique. We seek a matrix Z_h that is as sparse as possible such that the entries in the vector of design variables $\tilde{\boldsymbol{x}}$ in the reduced dimensional space depend on the fewest number of entries from the vector $\boldsymbol{x}$ from the full-dimensional design space. We thus select Z_h to satisfy the following optimization problem:

$$\begin{aligned} \mathsf{Z}_h = \underset{\mathsf{Z}_h}{\operatorname{argmin}} \|\mathsf{Z}_h\|_0 \quad \text{subject to} \qquad & \mathsf{A}_h \left(\mathsf{Z}_h\right)_{:i} = \mathbf{0} \,\forall\, i\{1, \ldots, n_d - n_{h,\text{lin}}\} \\ & \operatorname{rank}(\mathsf{Z}_h) = n_d - n_{h,\text{lin}} \\ & \sum_{j=1}^{n_d} \left|(\mathsf{Z}_h)_{ji}\right| = 1 \,\forall\, i\{1, \ldots, n_d - n_{h,\text{lin}}\}. \end{aligned} \tag{36}$$

The solution of Eq. (36) is also not unique since swapping the order of columns of Z_h does not impact the objective value or the constraints. For simple linear equality constraints, the null space Z_h can be formed to satisfy Eq. (36) without resorting to a numerical optimizer. Consider the following linear equality constraints for a five-dimensional design space:

$$\mathsf{A}_h = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}, \ \boldsymbol{b}_h = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{37}$$

which requires that $x_1 = x_2$ and $x_3 = x_4$ for the linear constraints to be satisfied. A null space for A_h that satisfies Eq. (36) is

$$\mathsf{Z}_h = \frac{1}{2}\begin{bmatrix} 1 & & \\ 1 & & \\ & 1 & \\ & 1 & \\ & & 2 \end{bmatrix}. \tag{38}$$

The null space for the matrix of linear equality constraints from Eq. (32) takes a similar form to the one in Eq. (38).

C. Linear transformation for constraints and gradients

With the change of variable from $\boldsymbol{x}$ to $\tilde{\boldsymbol{x}}$ from Eq. (33) the bound constraints become

$$\begin{aligned} \underline{\boldsymbol{x}} \le \boldsymbol{x} \quad & \le \overline{\boldsymbol{x}} \\ \underline{\boldsymbol{x}} \le \boldsymbol{x}_p + \mathsf{Z}_h \boldsymbol{y} & \le \overline{\boldsymbol{x}} \\ \underline{\boldsymbol{x}} - \boldsymbol{x}_p \le \mathsf{Z}_h \boldsymbol{y} \quad & \le \overline{\boldsymbol{x}} - \boldsymbol{x}_p, \end{aligned} \tag{39}$$

which are now linear inequality constraints. The linear inequality constraints for $\boldsymbol{x}$ remain linear inequality constraints for $\tilde{\boldsymbol{x}}$ and they given by

$$\begin{aligned} \mathsf{A}_g \boldsymbol{x} &\le \boldsymbol{b}_g \\ \mathsf{A}_g \left(\boldsymbol{x}_p + \mathsf{Z}_h \tilde{\boldsymbol{x}}\right) &\le \boldsymbol{b}_g \\ \mathsf{A}_g \mathsf{Z}_h \tilde{\boldsymbol{x}} &\le \boldsymbol{b}_g - \mathsf{A}_g \boldsymbol{x}_p. \end{aligned} \tag{40}$$

The gradients for the objective function and nonlinear constraints with respect to $\tilde{\boldsymbol{x}}$ are given by

$$\frac{\partial f}{\partial \tilde{\boldsymbol{x}}} = \frac{\partial f}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial \tilde{\boldsymbol{x}}} = \frac{\partial f}{\partial \boldsymbol{x}} \mathsf{Z}_h \tag{41}$$

$$\frac{\partial \boldsymbol{g}}{\partial \tilde{\boldsymbol{x}}} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial \tilde{\boldsymbol{x}}} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} \mathsf{Z}_h \tag{42}$$

$$\frac{\partial \boldsymbol{h}}{\partial \tilde{\boldsymbol{x}}} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial \tilde{\boldsymbol{x}}} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}} \mathsf{Z}_h. \tag{43}$$

Acknowledgments

The authors would like to acknowledge the financial support provided by the Natural Sciences and Engineering Research Council of Canada and the Ontario Graduate Scholarship Program.

References

[1] Reist, T. A., Zingg, D. W., Rakowitz, M., Potter, G., and Banerjee, S., "Multi-fidelity Optimization of Hybrid Wing–Body Aircraft with Stability and Control Requirements," *Journal of Aircraft*, Vol. 56, No. 2, 2019, pp. 442–456. https://doi.org/10.2514/1.C034703.

[2] Chernukhin, O., and Zingg, D. W., "Multimodality and Global Optimization in Aerodynamic Design," *AIAA Journal*, Vol. 51, No. 6, 2013, pp. 1342–1354. https://doi.org/10.2514/1.J051835, URL https://arc.aiaa.org/doi/10.2514/1.J051835.

[3] Streuber, G. M., and Zingg, D. W., "Evaluating the Risk of Local Optima in Aerodynamic Shape Optimization," *AIAA Journal*, Vol. 59, No. 1, 2021, pp. 75–87. https://doi.org/10.2514/1.J059826.

[4] Blonigan, P. J., Wang, Q., Nielsen, E. J., and Diskin, B., "Least-Squares Shadowing Sensitivity Analysis of Chaotic Flow Around a Two-Dimensional Airfoil," *AIAA Journal*, Vol. 56, No. 2, 2018, pp. 658–672. https://doi.org/10.2514/1.J055389.

[5] Ashley, A., Crean, J., and Hicken, J., "Towards Aerodynamic Shape Optimization of Unsteady Turbulent Flows," *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics, San Diego, California, 2019. https://doi.org/10.2514/6.2019-0168.

[6] Lea, D. J., Allen, M. R., and Haine, T. W. N., “Sensitivity analysis of the climate of a chaotic system,” *Tellus A: Dynamic Meteorology and Oceanography*, Vol. 52, No. 5, 2000, pp. 523–532. https://doi.org/10.1034/j.1600-0870.2000.01137.x.

[7] Chater, M., Ni, A., Blonigan, P. J., and Wang, Q., “Least Squares Shadowing Method for Sensitivity Analysis of Differential Equations,” *SIAM Journal on Numerical Analysis*, Vol. 55, No. 6, 2017, pp. 3030–3046. https://doi.org/10.1137/15M1039067.

[8] Garai, A., and Murman, S. M., “Stabilization of the Adjoint for Turbulent Flows,” *AIAA Journal*, 2021, pp. 1–13. https://doi.org/10.2514/1.J059998.

[9] Zingg, D. W., Nemec, M., and Pulliam, T. H., “A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization,” *European Journal of Computational Mechanics*, Vol. 17, No. 1-2, 2008, pp. 103–126. https://doi.org/10.3166/remn.17.103-126, URL https://www.tandfonline.com/doi/full/10.3166/remn.17.103-126.

[10] Jameson, A., Martinelli, L., and Pierce, N., “Optimum Aerodynamic Design Using the Navier-Stokes Equations,” *Theoretical and Computational Fluid Dynamics*, Vol. 10, No. 1-4, 1998, pp. 213–237. https://doi.org/10.1007/s001620050060.

[11] Giles, M. B., and Pierce, N. A., “An Introduction to the Adjoint Approach to Design,” *Flow, Turbulence and Combustion*, Vol. 65, 2000, pp. 393–415. https://doi.org/10.1023/A:1011430410075.

[12] Gagnon, H., and Zingg, D. W., “Euler-Equation-Based Drag Minimization of Unconventional Aircraft Configurations,” *Journal of Aircraft*, Vol. 53, No. 5, 2016, pp. 1361–1371. https://doi.org/10.2514/1.C033591.

[13] Reist, T. A., and Zingg, D. W., “High-Fidelity Aerodynamic Shape Optimization of a Lifting-Fuselage Concept for Regional Aircraft,” *Journal of Aircraft*, Vol. 54, No. 3, 2017, pp. 1085–1097. https://doi.org/10.2514/1.C033798.

[14] Nocedal, J., and Wright, S. J., *Numerical Optimization*, second edition ed., Springer series in operation research and financial engineering, Springer, New York, NY, 2006.

[15] Jones, D. R., Schonlau, M., and Welch, W. J., “Efficient Global Optimization of Expensive Black-Box Functions,” *Journal of Global Optimization*, Vol. 13, 1998, pp. 455–492. https://doi.org/https://doi.org/10.1023/A:1008306431147.

[16] Ameli, S., and Shadden, S. C., “Noise Estimation in Gaussian Process Regression,” *arXiv*, 2022. URL https://arxiv.org/abs/2206.09976.

[17] Rasmussen, C. E., and Williams, C. K. I., *Gaussian Processes for Machine Learning*, Adaptive computation and machine learning, MIT Press, Cambridge, Mass, 2006.

[18] Priem, R., Gagnon, H., Chittick, I., Dufresne, S., Diouane, Y., and Bartoli, N., “An efficient application of Bayesian optimization to an industrial MDO framework for aircraft design.” *AIAA AVIATION 2020 FORUM*, American Institute of Aeronautics and Astronautics, Virtual Event, 2020. https://doi.org/10.2514/6.2020-3152.

[19] Mortished, C., Ollar, J., Toropov, V., and Sienz, J., “Aircraft Wing Optimization based on Computationally Efficient Gradient-Enhanced Ordinary Kriging Metamodel Building,” *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, San Diego, California, USA, 2016. https://doi.org/10.2514/6.2016-0420.

[20] Bagheri, A. K., and Da Ronch, A., “Adjoint-Based Surrogate Modelling of Spalart-Allmaras Turbulence Model Using Gradient Enhanced Kriging,” *AIAA AVIATION 2020 FORUM*, American Institute of Aeronautics and Astronautics, Virtual Event, 2020. https://doi.org/10.2514/6.2020-2991.

[21] Wu, A., Aoi, M. C., and Pillow, J. W., “Exploiting gradients and Hessians in Bayesian optimization and Bayesian quadrature,” *arXiv:1704.00060 [stat]*, 2018. URL http://arxiv.org/abs/1704.00060.

[22] Laurent, L., Le Riche, R., Soulier, B., and Boucard, P.-A., “An Overview of Gradient-Enhanced Metamodels with Applications,” *Archives of Computational Methods in Engineering*, Vol. 26, No. 1, 2019, pp. 61–106. https://doi.org/10.1007/s11831-017-9226-3.

[23] Marchildon, A. L., and Zingg, D. W., “A Non-intrusive Solution to the Ill-Conditioning Problem of the Gradient-Enhanced Gaussian Covariance Matrix for Gaussian Processes,” *Journal of Scientific Computing*, Vol. 95, No. 3, 2023. https://doi.org/10.1007/s10915-023-02190-w.

[24] Dalbey, K., “Efficient and robust gradient enhanced Kriging emulators.” Tech. Rep. SAND2013-7022, 1096451, Sandia National Laboratories, Aug. 2013. https://doi.org/10.2172/1096451.

[25] Osborne, M. A., Garnett, R., and Roberts, S. J., “Gaussian Processes for Global Optimization,” *3rd International Conference on Learning and Intelligent Optimization*, Learning and Intelligent Optimization (LION), Trento, Italy, 2009.

[26] Won, J. H., and Kim, S.-J., "Maximum Likelihood Covariance Estimation with a Condition Number Constraint," *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*, IEEE, Grove, CA, USA, 2006, pp. 1445–1449. https://doi.org/10.1109/ACSSC.2006.354997.

[27] Ollar, J., Mortished, C., Jones, R., Sienz, J., and Toropov, V., "Gradient based hyper-parameter optimisation for well conditioned kriging metamodels," *Structural and Multidisciplinary Optimization*, Vol. 55, No. 6, 2017, pp. 2029–2044. https://doi.org/10.1007/s00158-016-1626-8.

[28] March, A., Willcox, K., and Wang, Q., "Gradient-based multifidelity optimisation for aircraft design using Bayesian model calibration," *The Aeronautical Journal*, Vol. 115, No. 1174, 2011, pp. 729–738. https://doi.org/10.1017/S0001924000006473.

[29] Shende, S., Gillman, A., Buskohl, P., and Vemaganti, K., "Systematic cost analysis of gradient- and anisotropy-enhanced Bayesian design optimization," *Structural and Multidisciplinary Optimization*, Vol. 65, No. 8, 2022, p. 235. https://doi.org/10.1007/s00158-022-03324-8.

[30] Marchildon, A. L., and Zingg, D. W., "A solution to the ill-conditioning of gradient-enhanced covariance matrices for Gaussian processes," *International Journal for Numerical Methods in Engineering*, 2024, p. e7498. https://doi.org/10.1002/nme.7498.

[31] Gardner, J. R., Kusner, M. J., Xu, Z., Weinberger, K. Q., and Cunningham, J. P., "Bayesian Optimization with Inequality Constraints," *ICML*, 2014, pp. 937–945.

[32] Gramacy, R. B., Gray, G. A., Le Digabel, S., Lee, H. K. H., Ranjan, P., Wells, G., and Wild, S. M., "Modeling an Augmented Lagrangian for Blackbox Constrained Optimization," *Technometrics*, Vol. 58, No. 1, 2016, pp. 1–11. https://doi.org/10.1080/00401706.2015.1014065, URL https://www.tandfonline.com/doi/full/10.1080/00401706.2015.1014065.

[33] Picheny, V., Gramacy, R. B., Wild, S., and Le Digabel, S., "Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian," *Advances in Neural Information Processing Systems*, Vol. 29, 2016, p. 9.

[34] Durantin, C., Marzat, J., and Balesdent, M., "Analysis of multi-objective Kriging-based methods for constrained global optimization," *Computational Optimization and Applications*, Vol. 63, No. 3, 2016, pp. 903–926. https://doi.org/10.1007/s10589-015-9789-6.

[35] Wu, J., Poloczek, M., Wilson, A. G., and Frazier, P., "Bayesian Optimization with Gradients," *31st Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 5273–5284.

[36] Toal, D. J. J., Bressloff, N. W., and Keane, A. J., "Kriging Hyperparameter Tuning Strategies," *AIAA Journal*, Vol. 46, No. 5, 2008, pp. 1240–1252. https://doi.org/10.2514/1.34822.

[37] Toal, D. J. J., Forrester, A. I. J., Bressloff, N. W., Keane, A. J., and Holden, C., "An adjoint for likelihood maximization," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Vol. 465, No. 2111, 2009, pp. 3267–3287. https://doi.org/10.1098/rspa.2009.0096.

[38] Toal, D. J., Bressloff, N. W., Keane, A. J., and Holden, C. M., "The development of a hybridized particle swarm for kriging hyperparameter tuning," *Engineering Optimization*, Vol. 43, No. 6, 2011, pp. 675–699. https://doi.org/10.1080/0305215X.2010.508524.

[39] Davis, G. J., and Morris, M. D., "Six Factors Which Affect the Condition Number of Matrices Associated with Kriging," *Mathematical Geology*, Vol. 29, No. 5, 1997, pp. 669–683. https://doi.org/10.1007/BF02769650.

[40] Pepelyshev, A., "The Role of the Nugget Term in the Gaussian Process Method," *mODa 9 – Advances in Model-Oriented Design and Analysis*, Physica-Verlag HD, Heidelberg, 2010, pp. 149–156. https://doi.org/10.1007/978-3-7908-2410-0_20, series Title: Contributions to Statistics.

[41] Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N., "Taking the Human Out of the Loop: A Review of Bayesian Optimization," *Proceedings of the IEEE*, Vol. 104, No. 1, 2016, pp. 148–175. https://doi.org/10.1109/JPROC.2015.2494218.

[42] Müller, J., and Day, M., "Surrogate Optimization of Computationally Expensive Black-Box Problems with Hidden Constraints," *INFORMS Journal on Computing*, Vol. 31, No. 4, 2019, pp. 689–702. https://doi.org/10.1287/ijoc.2018.0864.

[43] Tfaily, A., Diouane, Y., Bartoli, N., and Kokkolaras, M., "Bayesian optimization with hidden constraints for aircraft design," *Cahiers du GERAD*, 2024. URL https://www.gerad.ca/fr/papers/G-2024-10.

[44] Basudhar, A., Dribusch, C., Lacaze, S., and Missoum, S., "Constrained efficient global optimization with support vector machines," *Structural and Multidisciplinary Optimization*, Vol. 46, No. 2, 2012, pp. 201–221. https://doi.org/10.1007/s00158-011-0745-5.

[45] Bachoc, F., Helbert, C., and Picheny, V., "Gaussian process optimization with failures: classification and convergence proof," *Journal of Global Optimization*, Vol. 78, No. 3, 2020, pp. 483–506. https://doi.org/10.1007/s10898-020-00920-0.

[46] Schonlau, M., Welch, W. J., and Jones, D. R., "Global versus local search in constrained optimization of computer models," *Institute of Mathematical Statistics Lecture Notes - Monograph Series*, Institute of Mathematical Statistics, Hayward, CA, 1998, pp. 11–25. https://doi.org/10.1214/lnms/1215456182, URL http://projecteuclid.org/euclid.lnms/1215456182.

[47] Hicken, J. E., and Zingg, D. W., "Aerodynamic Optimization Algorithm with Integrated Geometry Parameterization and Mesh Movement," *AIAA Journal*, Vol. 48, No. 2, 2010, pp. 400–413. https://doi.org/10.2514/1.44033.

[48] Reist, T. A., Koo, D., Zingg, D. W., Bochud, P., Castonguay, P., and Leblond, D., "Cross Validation of Aerodynamic Shape Optimization Methodologies for Aircraft Wing-Body Optimization," *AIAA Journal*, Vol. 58, No. 6, 2020, pp. 2581–2595. https://doi.org/10.2514/1.J059091.

[49] Spalart, P., and Allmaras, S., "A one-equation turbulence model for aerodynamic flows," *30th Aerospace Sciences Meeting and Exhibit*, American Institute of Aeronautics and Astronautics, Reno, NV, U.S.A., 1992. https://doi.org/10.2514/6.1992-439.

[50] Osusky, M., and Zingg, D. W., "Parallel Newton–Krylov–Schur Flow Solver for the Navier–Stokes Equations," *AIAA Journal*, Vol. 51, No. 12, 2013, pp. 2833–2851. https://doi.org/10.2514/1.J052487.

[51] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, Vol. 47, No. 1, 2005, pp. 99–131. https://doi.org/10.1137/S0036144504446096.