AIAA SciTech Forum
9 - 13 January 2017, Grapevine, Texas
58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference

Delaunay-based optimization in CFD leveraging multivariate adaptive polyharmonic splines (MAPS)

Shahrouz Alimohammadi*, Pooriya Beyhaghi†, Gianluca Meneghello‡, Thomas R. Bewley§

Delaunay-based derivative-free optimization leveraging global surrogates (Δ-DOGS) is a recently-developed optimization algorithm designed for nonsmooth functions in a handful of adjustable parameters. The first implementation of the original Δ-DOGS algorithm used polyharmonic splines to develop an inexpensive interpolating "surrogate" of the (expensive) function of interest. The behavior of this surrogate was found to be irregular in cases for which the function of interest turned out to be much more strongly dependent on some of the adjustable parameters than others. This irregularity of the surrogate led to the optimization algorithm requiring many more function evaluations than might have otherwise been necessary. In the present work, a modified interpolation strategy, dubbed multivariate adaptive polyharmonic splines (MAPS), is proposed to mitigate this irregular behavior, thereby accelerating the convergence of Δ-DOGS. The MAPS approach modifies the natural polyharmonic spline (NPS) approach by rescaling the parameters according to their significance in the optimization problem based on the data available at each iteration. This regularization of the NPS approach ultimately reduces the number of function evaluations required by Δ-DOGS to achieve a specified level of convergence in optimization problems characterized by parameters of varying degrees of significance. The importance of this rescaling of the parameters during the interpolation step is problem specific. To quantify its beneficial impact on a practical problem, we compare Δ-DOGS with MAPS to Δ-DOGS with NPS on an application related to hydrofoil shape optimization in seven parameters; results indicate a notable acceleration of convergence leveraging the MAPS approach.

I. Introduction

Factors contributing to the choice of an algorithm for optimizing a function $f(x)$ include the cost of computing $f(x)$, the local smoothness of $f(x)$, the availability of derivative information, the number of design parameters, and the shape of the feasible domain considered in parameter space. This paper considers simulation-based optimization problems for which the cost of computing $f(x)$ is high, $f(x)$ may be locally nonsmooth, and derivative information may be unavailable, but the number of design parameters is relatively low (say, $n \lesssim 10$), and the feasible domain in parameter space is a simple convex region bounded by linear inequality constraints.

Over the last thirty years, as computational power has increased, derivative-free optimization algorithms have become increasingly valuable for shape optimization leveraging commercial off-the-shelf (COTS) computer-aided design (CAD) tools. Response surface methods [28] are perhaps the most computationally efficient derivative-free approaches available for shape optimization problems today, with recent applications including the design of helicopter blades and airfoils [14, 16, 30].

Response surface methods use a computationally inexpensive model, $p(x)$, of the (computationally expensive) function of interest, $f(x)$, to approximate the trends evident in the data available at each iteration. Correlation-based interpolation models [6, 7, 8, 31] have been widely used in such methods as surrogates to model the underlying function $f(x)$, and simultaneously to model the uncertainty associated with this surrogate. A method of this class considers the objective function as a "realization of a random process", and the parameters of the statistical model inherent to the method are tuned, using a maximum likelihood approach, to maximize the probability of the observed data at each iteration. Unlike polyharmonic spline interpolation, no metric of smoothness of the interpolant is minimized in correlation-based interpolation strategies. Thus, surrogates developed using such strategies are sometimes nonsmooth (see, e.g., the appendix of [13]), which can significantly decrease the convergence rate of the associated optimization algorithm [17, 27, 34].

The recently-developed Delaunay-based derivative-free optimization via global surrogates (Δ-DOGS) family of methods [2, 3, 4, 9, 10, 11, 12, 13] are response surface methods which are built upon the framework of a Delaunay triangulation of the available datapoints at each iteration. These methods are provably globally convergent under the

*Graduate Research Assistant, Department of Mechanical & Aerospace Eng., UC San Diego, salimoha@eng.ucsd.edu
†Graduate Research Assistant, Department of Mechanical & Aerospace Eng., UC San Diego, p.beyhaghi@gmail.com
‡Post Doctoral Researcher, Department of Mechanical & Aerospace Eng., UC San Diego, gianluca.meneghello@gmail.com
§Professor, Department of Mechanical & Aerospace Eng., UC San Diego, bewley@eng.ucsd.edu

appropriate assumptions, and are found to be remarkably computationally efficient on a range of benchmark as well as application-based problems.

As with other response surface methods, algorithms in the Δ-DOGS family iteratively minimize a search function $s(x)$ based on both an interpolation of the existing datapoints as well as a model of the uncertainty of this interpolant. Significantly, methods in the Δ-DOGS family decouple the tasks of interpolation and uncertainty modeling. A simple synthetic uncertainty model is used which is zero at each datapoint and piecewise quadratic within each simplex; this approach proves to be both effective and easy to generalize (see [2, 10]). The present paper introduces and demonstrates a new interpolation approach that is well suited for optimization algorithms in this family.

In previous implementations of optimization algorithms in the Δ-DOGS family [2, 3, 4, 5, 10, 11, 12, 13], the natural polyharmonic spline (NPS) interpolation approach was used. Applications included hydrofoil design optimization [29]; in this particular application, though satisfactory results were ultimately achieved, a non-uniform dependence of $f(x)$ on the design parameters x over the parameter space considered was observed, as well an associated (yet, unanticipated) irregularity of the associated NPS interpolants used at each iteration.

The present paper specifically addresses these shortcomings by introducing a new interpolation strategy, dubbed *multivariate adaptive polyharmonic splines (MAPS)* which, prior to performing the interpolation at each iteration, rescales the coordinate directions of the domain based on the observed variation of the available data in each direction. The hydrofoil optimization problem described in [29] represents a typical challenge problem for this effort, as its objective function $f(x)$, which characterizes the lift/drag ratio of the foil, is much more strongly dependent on some of the design parameters than others. This behavior is common in shape optimization.

The present paper specifically employs MAPS in the Delaunay-based optimization strategy developed in [3]. For comparison, Δ-DOGS with MAPS and Δ-DOGS with NPS are both applied to the hydrofoil optimization problem developed in [29].

The structure of the paper is as follows: Section II briefly reviews the essential ideas of Delaunay-based optimization (Δ-DOGS) with acceleration based on Cartesian grids. Section III introduces our new interpolation strategy, MAPS, which automatically scales the parameter space prior to performing each interpolation, thereby regularizing the interpolant and, ultimately, accelerating convergence. Section IV applies the new interpolation strategy within Δ-DOGS to optimize a high-performance sailboat hydrofoil design. Some conclusions are presented in Section V.

II. A brief review of Δ-DOGS

Algorithms in the Δ-DOGS family are already well suited for many low-dimensional shape optimization problems. In this paper, we consider specifically the optimization of a nonconvex objective function $f(x)$ inside a convex feasible domain bounded by linear constraints:

$$\text{minimize } f(x) \text{ with } x \in \Omega = \{x \in \mathbb{R}^n | A\,x \leq b\}, \tag{1}$$

where $x \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$, and the feasible domain Ω is assumed to be compact (closed and bounded). The compactness assumption guarantees that there is at least one solution of (1).

Algorithms of the Δ-DOGS family attempt to solve (1) using successive function evaluations at feasible points $x_k \in \Omega$ in search of the global minimum of $f(x)$ for $x \in \Omega$. To accomplish this efficiently, a search function $s(x)$ is minimized at each iteration; this search function is built using an interpolation of the existing datapoints, $p(x)$, a synthetic model of the uncertainty of this interpolant, $e(x)$, and a target value for the function itself, y_0.

In this work, we assume that a target value y_0 is known which is achievable, and the goal is to find a point x^* such that $f(x^*) \leq y_0$. The interpolation and the uncertainty function at each iteration k are denoted $p^k(x)$ and $e^k(x)$, respectively. The uncertainty function and the search function are defined as follows.

Definition II.1. Take S as a set of points that includes the vertices of domain Ω, and Δ as a Delaunay triangulation of S. The *local uncertainty function* $e_i(x)$ for each simplex $\Delta_i \in \Delta$ is defined

$$e_i(x) = r_i^2 - \|x - Z_i\|^2, \tag{2}$$

where r_i and Z_i are the circumradius and circumcenter of Δ_i. The *global uncertainty function* $e(x)$ is defined

$$e(x) = e_i(x), \quad \text{for all } x \in \Delta_i. \tag{3}$$

The uncertainty function $e(x)$ is illustrated in Figure 1 in a parameter space of dimension $n = 2$. The uncertainty function $e(x)$ is characterized by the following useful properties:

1. the uncertainty function $e(x)$ is non-negative $e(x) \geq 0$ for all points $x \in \Omega$, and $e(x) = 0$ for all $x \in S$,

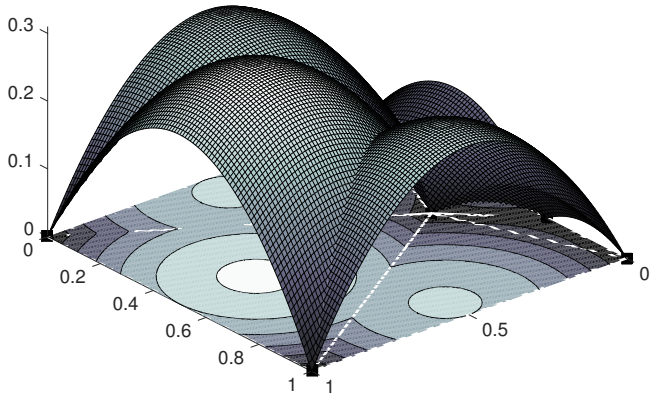


Figure 1: Illustration of the uncertainty function $e(x)$ in $n = 2$ dimensions.

2. the uncertainty function $e(x)$ is continuous, Lipschitz, and piecewise quadratic.

3. the uncertainty function $e(x)$ is everywhere equal to the maximum of the local uncertainty functions $e_i(x)$; i.e.,

$$e(x) = \max_{1 \le i \le |\Delta|} e_i(x) \quad \text{for all } x \in \Delta. \tag{4}$$

A number of additional useful properties of $e(x)$ are established in Lemmas [2:5] of [13].

Using the uncertainty function $e(x)$ and a suitable interpolation $p(x)$ of the available data, the search function $s(x)$ is defined as follows.

Definition II.2. Take S as a set of datapoints that includes the vertices of the domain Ω, Δ as a Delaunay triangulation of S, $p(x)$ as an interpolation of the function $f(x)$ over S, and $e(x)$ as the global uncertainty function defined in (3) and built on the framework of Δ. The global search function $s(x)$ is defined as follows:

$$s(x) = \begin{cases} \dfrac{p(x) - y_0}{e(x)}, & \text{if } p(x) \le y_0, \\ p(x) - y_0, & \text{otherwise,} \end{cases} \tag{5}$$

where y_0 is the target value of $f(x)$ (that is, an estimate of its lower bound).

Based on the constructions given above, the essential steps of Δ-DOGS are given in Algorithm 1. Figure 2 illustrates one iteration of Δ-DOGS on an representative problem.

One of the challenges of the basic Δ-DOGS algorithm [13] is its overexploration of the boundaries of feasibility. This issue may be addressed by using a Cartesian grid [11] or, more generally, a dense lattice [3] to help coordinate the search, and successively refining this grid or lattice as convergence is approached. These coordination steps are useful for minimizing the accumulation of function evaluations along the boundary of the feasible domain, though they must be implemented with care. The particular variant of the Δ-DOGS optimization algorithm that is used in the present work is that described in [3], which modifies the basic Δ-DOGS algorithm to coordinate the search with lattices over Ω and $\partial\Omega$ that are successively refined as convergence is approached. The technical details of the modifications necessary to implement this idea correctly are somewhat involved, and discussed in detail in [3, 11], together with formal proofs of convergence and illustration on model problems.

One of the parameters that is essential for accelerating the convergence of the Δ-DOGS algorithm is the estimate of the lower bound of the objective function, y_0, over the feasible domain Ω. It is shown in [13] that, if $y_0 \le f(x^*)$, convergence to the global minimum is guaranteed for any twice differentiable function $f(x)$; however, values of y_0 for which $y_0 \ll f(x^*)$ tend to reduce the convergence rate. If $y_0 > f(x^*)$, the algorithm will stop at some feasible point $\tilde{x} \in \Omega$ such that $f(\tilde{x}) \le y_0$; in this case, convergence to the global minimum is not guaranteed.

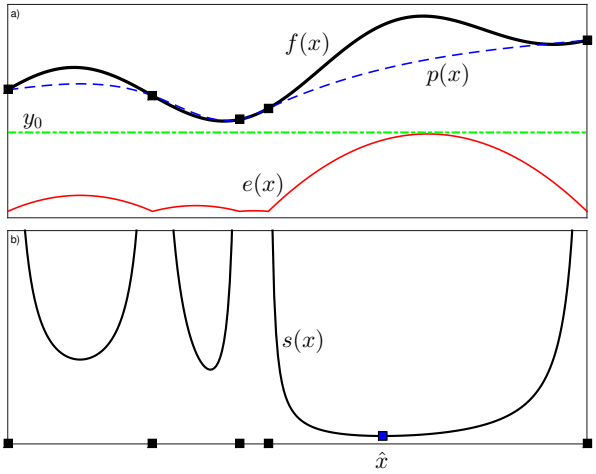


Figure 2: The essential elements of Δ-DOGS. Subfigure $a)$ indicates (black solid) the truth function $f(x)$, (green dot-dashed) the target value y_0, (blue dashed) the interpolating surrogate function $p(x)$, (red solid) the synthetic model of the uncertainty $e(x)$, and (black squares) the datapoints. Subfigure $b)$ indicates the search function $s(x)$, defined in (5), and (blue squares) the minimizer $\hat{x}$ of the search function.

Algorithm 1 The essential steps of Δ-DOGS.

1: Set $k = 0$. Take the set of initialization points S_0 as all M of the vertices of the feasible domain L.
2: Calculate (or, for $k > 0$, update) an appropriate interpolating function $p_k(x)$ through all points in S_k.
3: Calculate (or, for $k > 0$, update) a Delaunay triangulation Δ^k over all of the points in S_k.
4: Find x_k as a global minimizer of $s_k(x)$ in Ω to obtain x_k.

$$\hat{x}_k = \text{argmin}_x \; s_k(x) \quad \text{subject to } x \in \Omega.$$

5: Calculate $f(x)$ at $\hat{x}_k$, and take $S^{k+1} = S^k \cup x_k$. Repeat from 2 until convergence.

Another important factor affecting the performance of Δ-DOGS algorithms is the choice of the interpolation strategy used. As mentioned previously, algorithms in the Δ-DOGS family can leverage any well-behaved interpolation strategy. In the previous implementations of Δ-DOGS, natural polyharmonic spline (NPS) interpolation has been used. The ultimate performance of Δ-DOGS algorithms depends strongly on the smoothness of the interpolations used. In some situations (specifically, when the variation of the function $f(x)$ with respect to the parameters is nonuniform, with much stronger variation in some coordinate directions than others), it is has been found that NPS interpolants are not sufficiently smooth. In the section that follows, we thus introduce a new interpolation method that rescales the parameter domain appropriately while performing the interpolation, thus developing a significantly smoother interpolant (and thereby, ultimately, accelerating convergence of the associated optimization algorithm).

III. A new polyharmonic spline interpolation algorithm for Δ-DOGS

As discussed above, the choice of the strategy to be used to construct the interpolant $p(x)$ is subtle, and strongly affects the rate of convergence of the associated optimization algorithm. Natural polyharmonic spline (NPS) interpolation is, in general, one of the most popular interpolation strategies available today, as its formulation specifically minimizes a metric measuring the curvature of the resulting interpolant. Numerical experiments in [13] showed that NPS interpolation is fairly well behaved, as compared with Kriging-based approaches, even when the available datapoints are clustered in various distinct regions of parameter space. Note also that NPS interpolation has also been used in various response surface methods developed by other groups, including [27, 34].

Appropriate rescaling of parameter space is a valuable step in numerical optimization, and various recent papers have attempted to address the rescaling issue in the optimization setting. In the numerical optimization literature, there

are two main approaches taken for the automatic rescaling of parameter space[a]. The first approach (see, e.g., [18]) is to use a correlation-based model to develop the interpolant, as discussed in the third paragraph of the introduction; such approaches naturally solve for correlation length scales during the computation of the interpolant via a maximum likelihood formulation, but do not guarantee smoothness of the resulting interpolant. The second approach (see, e.g., [1]) uses a statistical method to identify the variation of the function with respect to each parameter in the available dataset, and uses this statistical information to rescale the parameters prior to performing the interpolation at each iteration. This approach works well if the data that is used for this sensitivity analysis is well-distributed over the feasible domain; however, during the optimization processes, the dataset is expected to become clustered in some regions of the feasible domain while remaining sparse in others, which renders this rescaling approach unreliable.

In the following two sections, we first review NPS, then develop and analyze our new interpolation strategy, dubbed MAPS, which includes, during the interpolation process, an automatic rescaling of the parameters based on the available data.

A. Natural polyharmonic spline (NPS)

Polyharmonic spline interpolation is a widely-used strategy to interpolate scattered data in multiple dimensions [23, 33]. An interpolant $p(x)$ is defined as a smooth function, which is typically inexpensive to compute, which models a "truth" function $f(x)$, which might be expensive to compute, such that

$$p(x_i) = f(x_i) \quad \text{for} \quad i = 1, \ldots, N. \tag{6}$$

To maximize the smoothness of $p(x)$, it is suggested in [33] that the following term should be minimized

$$\int_\Omega \|\nabla^m p(x)\|_2^2 \, dx, \tag{7}$$

subject to $p(x_i) = f(x_i), \forall i = 1, \ldots, N$, where m is an integer such that $m \leq N$. Under these conditions, the minimizer of (7) gives a polyharmonic spline interpolant [33]. By choosing $m = 2$ in (7), the resulting interpolant will be contained in the Beppo-Levi space of distributions on $\mathbb{R}^n$ with square integrable second derivatives [15]; this choice is termed natural polyharmonic spline (NPS) interpolation, and is by far the most common choice in this class.

NPS may be defined as a combination of a weighted sum of a set of radial basis functions $\varphi(r)$ built around the location of each evaluation point, $\{x_i\}_{i=1}^N$, and a linear function of x:

$$p(x) = \sum_{i=1}^{N} w_i \, \varphi(||x - x_i||) + v^T \begin{bmatrix} 1 \\ x \end{bmatrix}. \tag{8}$$

The coefficients w_i and v_i are real numbers in which v_i represent the coefficients of a linear polynomial.

The following orthogonality conditions are applied:

$$\sum_{i=1}^{N} w_i = 0, \qquad \sum_{i=1}^{N} w_i \, x_{i\ell} = 0 \quad \forall \, \ell = 1, \, 2, \, \ldots, \, n; \tag{9}$$

this imposes $n + 1$ constraints, coupled with the interpolation condition (6), which imposes N constraints, gives the linear system (10) below, from which one can solve[b] for the parameters w and v in the NPS interpolation formula (8):

$$\begin{bmatrix} F & V^T \\ V & 0 \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} = \begin{bmatrix} f(x_i) \\ 0 \end{bmatrix} \tag{10a}$$

where

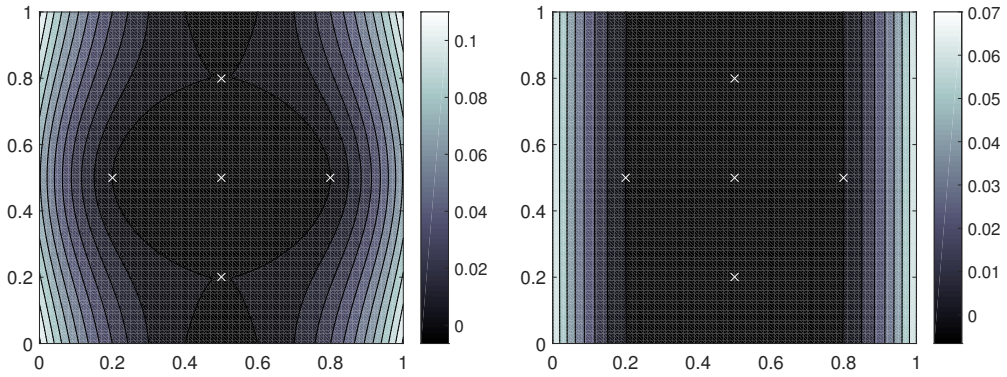
$$F_{i,j} = \varphi(||x_i - x_j||), \quad i, j = 1, \ldots, N, \tag{10b}$$

$$\text{and} \quad \varphi(r) = r^3, \quad r = \|x - x_i\|, \quad V = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_N \end{bmatrix}. \tag{10c}$$

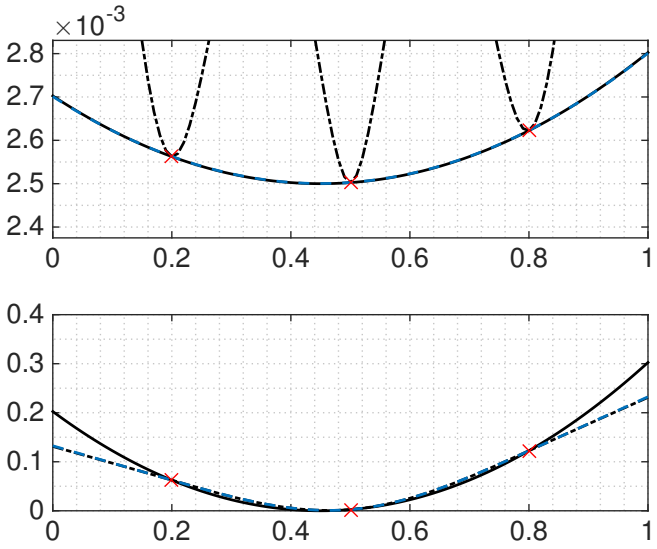
The solution of the above linear system is unique [33]. Thus, by solving the linear system (10), the coefficients w and v are found, and $p(x)$ is determined.

[a] The two approaches described here may be considered specifically for the problem of dimension reduction; that is, for the exploration of $f(x)$ over a reduced number of parameters during certain steps of the optimization algorithm. This possibility will be explored in the present setting in a future paper.

[b] The reader is encouraged to read about the details associated with efficiently finding these weights as described in [33].



a) Error contour for the NPS interpolant (8). **b) Error contour for the MAPS interpolant (11).**



c) Blue dashed: MAPS interpolant. Black dashed: NPS interpolant. Black solid: truth function $f(x)$. Red $\times$: available datapoints. Top figure: $f(x_1, 0.45)$ v.s. x_1. Bottom figure: $f(0.45, x_2)$ v.s. x_2.

Figure 3: Behavior of NPS and MAPS interpolants, based on the datapoints marked, with respect to the truth function $f(x) = 0.01\,(x_1 - 0.45)^2 + (x_2 - 0.45)^2$, shown solid. Across the domain $[0, 1]$ in each coordinate direction, the variation of $f(x)$ in the x_2 direction is 1000 times stronger than the variation of $f(x)$ in the x_1 direction; i.e., $f(x)$ is much more sensitive to x_2 than it is to x_1. In (a) and (b), the deviation of the interpolants $p(x)$ from the truth $f(x)$ is quantified; using NPS, as seen in (a), creates deviations in the x_1 (vertical) direction that are absent using the MAPS approach, as seen in (b). These deviations reduce the convergence rate of the associated optimization. In (c), the behavior of MAPS, NPS, and the truth $f(x)$ are plotted with respect to x_1 and x_2 across the center of the domain.

The natural polyharmonic spline interpolation formula in (8) has various shortcomings, the most significant of which is the ill conditioning of the linear system (10) that is solved to fit the polyharmonic spline to the datapoints. This stiffness is a direct result of the function $f(x)$ having a nonuniform variation in the design parameters. The result of this stiffness is spurious oscillations of the resulting interpolant in coordinate directions with less pronounced variation of $f(x)$. Figure 3 illustrates this behaviour for a representative problem, and compares with the outcome of the new interpolation strategy proposed below.

B. Multivariate adaptive polyharmonic splines (MAPS)

Consider now an interpolant of the form:

$$p_s(x) = \sum_{i=1}^{N} w_i \, \varphi(a \otimes \|x - x_i\|) + v^T \begin{bmatrix} 1 \\ x \end{bmatrix}, \tag{11}$$

$$\text{where} \quad \varphi(a \otimes r) := (a \otimes r)^3 \quad \text{and} \quad a \otimes r := \sum_{\ell=1}^{n} a_\ell \, r_\ell,$$

where the a_ℓ are scaling parameters, and inherent functions of w and v. The following condition is imposed:

$$\sum_{\ell=1}^{n} a_\ell^2 = n. \tag{12}$$

Unfortunately, the quadratic constraint (12) is more challenging than a linear constraint from the perspective of optimizing the weights. Thus, (12) is restated as a linear constraint using the change of variables $\theta_\ell = a_\ell^2$:

$$\sum_{\ell=1}^{n} \theta_\ell = n, \qquad \text{where} \qquad \theta_\ell \geq 0. \tag{13}$$

The formulation developed below thus works with θ instead of a. The problem of computing a MAPS interpolant thus reduces to the problem of solving for the variables

$$X = (w_1 \ldots, w_N, v_1, \ldots, v_{n+1}, \theta_1, \ldots, \theta_n)^T.$$

In contrast with the NPS interpolation formula (8), which has $N + n + 1$ unknowns, the MAPS interpolation formula (11) has $N + 2n + 1$ unknowns. With the additional n degrees of freedom (i.e., the scaling parameters θ_ℓ) in MAPS, a wider range of parameters is used in order to obtain a smoother interpolant.

To improve the convergence of Δ-DOGS, we desire to use smooth interpolants at each iteration. We achieve this in the present context by performing minimum Frobenius norm (MFN) interpolations. In an MFN formulation, the Hessian of the interpolant, $\nabla^2 p_s(x)$, is minimized by minimizing the L_2-norm of the vector w [21, 17]. The scaling θ could thus, in theory, be optimally tuned by solving

$$\begin{aligned} \min_{w,v,\theta} \quad & \sum_{i=1}^{N} w_i^2, \\ \text{subject to} \quad & \sum_{\ell=1}^{n} \theta_\ell = n, \quad \theta_\ell \geq 0, \quad \ell = 1, 2, \ldots, n, \\ & \sum_{i=1}^{N} w_i = 0, \quad \sum_{i=1}^{N} w_i \, x_{i\ell} = 0, \quad \ell = 1, \, 2, \, \ldots, \, n, \\ & p_s(x_i) = f(x_i), \quad i = 1, 2, \ldots, N. \end{aligned} \tag{14}$$

The above optimization problem is nonconvex, however, as the last constraint above is a nonlinear function of θ. For a fixed value of θ, this constraint is satisfied by solving a linear system, and the resulting objective function may be rewritten as:

$$Q(\theta) := b^T \, A(\theta)^{-T} \, L \, A(\theta)^{-1} \, b = \sum_{i=1}^{N} w_i^2, \tag{15a}$$

where

$$L = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, \quad A(\theta) = \begin{bmatrix} F(\theta) & V^T \\ V & 0 \end{bmatrix}, \qquad b = \begin{bmatrix} f(x_i) \\ 0 \end{bmatrix}, \tag{15b}$$

noting that

$$F(\theta)_{i,j} = \varphi\Big(\|(x_i - x_j)\|_\theta\Big), \quad i, j = 1, \ldots, N, \tag{15c}$$

$$\text{where} \quad \varphi(r) := r^3 \quad \text{and} \quad \|r\|_\theta := \Big(\sum_{\ell=1}^{n} \theta_\ell \, r_\ell^2\Big)^{1/2}, \quad V = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_N \end{bmatrix}. \tag{15d}$$

Algorithm 2 Scheme to find the scaling parameters of MAPS (11)

1: Set $j = 0$. Take the set of initialization points S_0 and consider λ_0, $\theta_\ell = 1$ for all $\ell = 1, 2, \ldots, n$.
2: For $j > 0$, initialize the following system by $\theta_0 \leftarrow \theta_{\lambda_{j-1}}$ with fixed λ_j
3: Find θ_{λ_j} by solving the quadratic programming

$$\theta_{\lambda_j} = \min_{\theta \in \mathbb{R}^n} \quad Q_{\lambda_j}(\theta) \qquad \text{subject to} \quad 1^T\,\theta = n, \quad 0 \le \theta \le n,$$

4: Update $\lambda_{j+1} \leftarrow \lambda_j/2$, increment j by one, and repeat from step 2 until (19) is satisfied.

The optimum scaling parameter θ^* can thus be found using an appropriate sequential quadratic programming (SQP) solver (see, e.g., [26]), with the Hessian approximated using BFGS:

$$\theta^* = \min_{\theta \in \mathbb{R}^n} \quad Q_j(\theta) \qquad \text{subject to} \quad 1^T\,\theta = n, \quad \theta \ge 0. \tag{16}$$

The main challenge in minimizing $Q(\theta)$ is the singularity of $A(\theta)$ as one of the elements of the vector θ approaches zero. This issue is illustrated for a simple quadratic problem in Figure 12 (see Appendix), which illustrates that there is a significant deviation in the value of $Q(\theta)$ as determined via SQP applied to (15), when compared with the optimum value of $Q(\theta)$, as one of the scaling parameters $\theta_\ell \to 0$. It is seen that the SQP approach tends to get caught in a local minimum for small θ_ℓ. This is a common situation, especially when the actual variation of the objective function with respect to some of the parameters is small. To address this issue, a series of new problems $Q_{\lambda_j}(\theta)$ is defined and iteratively minimized as the relaxation parameter λ_j is gradually decreased towards zero:

$$Q_\lambda(\theta) = b^T\,(A_\lambda(\theta))^{-T}\,L\,(A_\lambda(\theta))^{-1}\,b, \quad \text{where} \quad A_\lambda(\theta) = A(\theta) + \lambda\,L. \tag{17}$$

Note that, as $\lambda \to 0$, (15) is recovered from (17). To minimize $Q_\lambda(\theta)$ efficiently, for any given λ, we need the following derivative information:

$$\frac{d\,Q_\lambda(\theta)}{d\,\theta_\ell} = 2\,B_\ell^T(\theta)\,L\,W(\theta), \tag{18a}$$

$$A_\lambda(\theta)\,B_\ell(\theta) = -\frac{\partial\,A_\lambda(\theta)}{\partial\,\theta_\ell}\,W(\theta), \tag{18b}$$

$$A_\lambda(\theta)\,W(\theta) = b \quad \text{where} \quad W(\theta) = \begin{bmatrix} w \\ v \end{bmatrix}, \tag{18c}$$

where $A_\lambda(\theta)$ is defined in (17) and L, b, $A(\theta)$ are defined in (15b). Having this information on the derivative of $Q_\lambda(\theta)$, we may use the BFGS method to minimize it. It is known (see, e.g., [22, 32]) that such a procedure will converge to a local minimum of (14). As codified in Algorithm 2, by performing a set of relaxations towards the solution[c], for successively smaller values of λ, it is found that reliable convergence to the desired (global) minimum of (14) is obtained; this optimized value of θ may then be used in the MAPS interpolation strategy (11).

Step 3 of Algorithm 2 can be solved using, e.g., the SNOPT optimization package [26], given the function of interest, $Q_{\lambda_j}(\theta)$, as in (17), and the gradient information, $d\,Q_{\lambda_j}(\theta)/d\,\theta_\ell$, as in (18), at each iteration k. The initial value of the relaxation parameter in Algorithm 2, λ_0, must be sufficiently large to give a well-conditioned linear system. To determine an appropriate stopping condition in this relaxation, define first the interpolation error δy_i over all points x_i for a given λ_j:

$$\delta y_i = f(x_i) - p_s(x_i; \theta_{\lambda_j}).$$

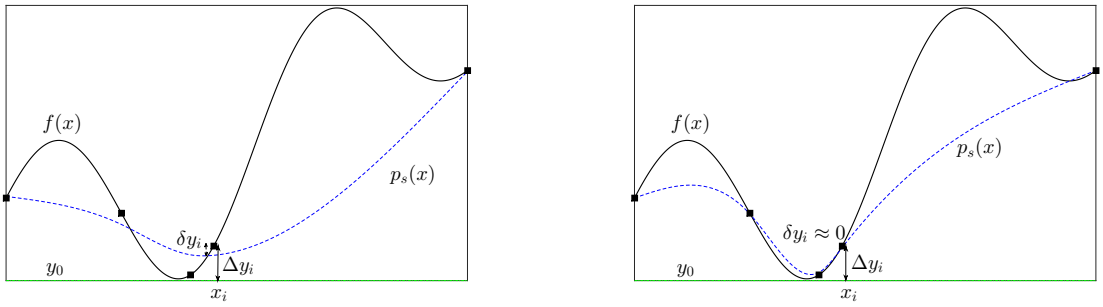
In addition, define the distance between the value of the objective function at x_i and the target value y_0 as

$$\Delta y_i = f(x_i) - y_0.$$

An effective stopping criteria is reached when the error of interpolation, δy_i, reduces to, say, less than 10% of Δy_i at each datapoint x_i; that is,

$$\frac{\delta y_i}{\Delta y_i} < 0.1 \qquad \text{for all} \qquad i = 1, 2, \ldots, N. \tag{19}$$

[c]Note that the initial point for minimizing $Q_{\lambda_{j+1}}(\theta)$ is the minimizer of $Q_{\lambda_j}(\theta)$ (see step 2 of Algorithm 2).



a) MAPS interpolant failing to satisfy (19) for some points x_i (before λ is made sufficiently small).

b) MAPS interpolant satisfying (19) for all points x_i (once λ is made sufficiently small).

Figure 4: Illustration of the termination condition (19) for the refinement of λ in Algorithm 2.

After a finite number of refinements j of Algorithm 2, the stopping criterion (19) will be satisfied. An important observation is that, for the initial iterations of Δ-DOGS, since Δy_i is large, (19) is satisfied even for relatively large values of λ_j. As Δ-DOGS proceeds, Δy_i becomes smaller, which necessitates an increased number of refinements of λ_j to make δy_i sufficiently small to satisfy (19) at all points x_i, thus driving the MAPS surrogate towards a true interpolant as convergence is approached.

IV. Implementation of Δ-DOGS with both MAPS and NPS on hydrofoil design

For validation purposes, Algorithm 1 was applied to the hydrofoil optimization problem considered in detail in [29], using both NPS (8) and MAPS (11).

In [29], the shape of a racing catamaran's hydrofoil was characterized by 7 design parameters, and Δ-DOGS(C) with NPS was used to maximize the hydrofoil efficiency, defined as its lift/drag ratio, at a fixed working condition. During the optimization, two specific challenges were encountered: (a) Δ-DOGS apparently overexplored the function $f(x)$ near the boundary of feasibility, Ω, due in part to the fact that the objective function $f(x)$ itself had somewhat irregular behavior close to the boundary of Ω, and (b) the non-uniform dependence of the objective function $f(x)$ on the various design parameters x apparently resulted in overexploration of $f(x)$ in the vicinity of the optimizated solution. These challenges resulted in many apparently unnecessary function evaluations during the optimization. The issue described in (a) above was resolved well in [3, 11] by leveraging a grid or lattice in Δ-DOGS to help coordinate the search. In the following, we incorporate MAPS interpolation (11) in Δ-DOGS to resolve issue (b).

In our previous work on the hydrofoil optimization problem [29], we observed that the objective function $f(x)$ depended much more strongly on some design parameters than others. This nonuniform dependence on the 7 design parameters[d] in the vicinity of the optimum solution is shown in Figure 5, where the stars indicate the optimum values of each parameter, and the red line indicates the variation in the efficiency of the foil as one of the parameters at a time is varied over its full range.

Simulation results (see Figure 8) indicate that, after about 30 function evaluations, the optimization algorithm approaches the optimal solution in the present problem, with a lift/drag ratio of about 36.

As mentioned previously (in particular, see Figure 3), spurious oscillations caused by nonuniform interpolants can be problematical in such optimization problems, significantly slowing convergence. We illustrate this issue by comparing NPS (Figure 6, bottom) and MAPS (Figure 6, top) interpolations of the data given by the optimum solution x^* together with the first 30 datapoints generated by Algorithm 1. The NPS interpolant is characterized by spurious peaks away from the optimal point, in both the x_1 and x_2 coordinate directions. These spurious peaks in the x_1 and x_2 coordinate directions are absent in the MAPS interpolant, which much more accurately captures the trends evident in the truth function itself (see Figure 5); this is remarkable, given that the interpolation is based on only 31 datapoints in a practical 7-dimensional problem.

We now summarize the design parameters used in this work, as suggested by [29], which represent a rectangular hydrofoil with an aspect ratio (AR) of 10 and a cross section of a NACA64_1-412 foil. The optimization is performed to minimize the drag of the foil subject to a design vertical and horizontal lift of $SC_z = 0.120$ and $SC_y = 0.066$.

[d]Note that, to ease the visualization, all of the design parameters are normalized to lie between 0 and 1.

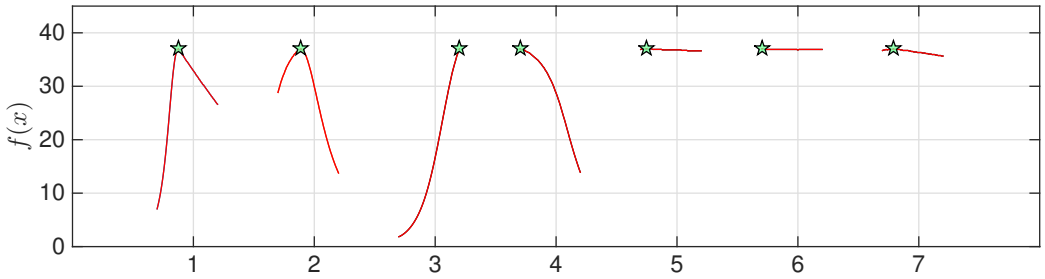


Figure 5: Nonuniform dependence, over the parameter space considered, of the objective function (i.e., the lift/drag ratio of the hydrofoil) on the 7 adjustable parameters defining the hydrofoil shape (see Table 1 and Figure 7). The stars indicate the optimal values, x_ℓ^*, of each of the 7 parameters of the foil. The red line indicates the variation in the efficiency of the foil when 6 of the parameters are held at their optimal values, and 1 of the parameters at a time is varied over its full range.

Table 1: Summary of the adjustable parameters used in the hydrofoil shape design problem.

Number (j)	Variable x_j	Description	lower bound	upper bound
1	S	planform surface	0.2	0.5
2	z_1	rational Bezier curve	0.5	1.5
3	dy	rational Bezier curve	0.5	1.5
4	dz	rational Bezier curve	-0.3	0.3
5	w_1	vertical plane weight	4.3	11
6	c_{tip}	tip chord length	0.05	0.5
7	w_c	weight spanwise	1.5	11

Figure 7 shows the geometry of the foil, where z is the vertical coordinate, y is the horizontal cross-flow coordinate, x is the horizontal stream-wise coordinate, s is the curvilinear coordinate, and S is the planform area. Other parameters of the optimization govern the $y-z$ plan and the chord distribution along the curvilinear coordinate s. Both the shape of the foils' quarter-chord line and the chord distribution are represented using Bezier curves, where the w_i are the weights of the corresponding control points (for details, see [29]). In this manner, a realizable foil is characterized efficiently with only seven parameters subject to simple bound constraints, as listed in Table 1.

In this work, as suggested by and validated in [29], AVL [20] has been used to calculate the lift/drag of a foil. AVL is an easy-to-use vortex-lattice-based software package for inviscid aerodynamic analysis problems of this sort. The estimate of the optimal objective function value, $y_0 = \max(C_L/C_D)$, can be obtained by classical aerodynamic analysis. The drag coefficient can be obtained as the sum of the viscous and inviscid (3D) drag components. For a foil with a specified aspect ratio of AR and an elliptic spanwise load, it can be estimated as follows:

$$C_D = C_{D_\nu}(C_L) + \frac{C_L^2}{\pi\, AR}, \tag{20}$$

where $C_{D_\nu}(C_L)$ is the viscous drag coefficient, which can be determined for a given 2D foil section either via an experiment or a 2D computational model, such as XFoil [19]. Following the detailed analysis in [29], a value of $y_0 = 37$ is used in the present work.

A. Optimization results and comparison

The optimization process balances the contribution of the viscous drag, proportional to the foil surface, and the inviscid drag, proportional to the square of the lift surface. The objective function $f(x)$ used is the lift/drag ratio. We actually minimize $\log[1 + 1/f(x)]$, instead of maximizing the efficiency of the foil $f(x)$ itself, since Algorithm 1 is designed for minimization problems. Note that the plots provided in this paper show $f(x)$, for ease of interpretation.

By switching from Δ-DOGS(C) with NPS to lattice-based Δ-DOGS with MAPS interpolation, the numerical results in Figure 8 indicate a significant improvement in the objective function value, $f(x)$, after a fixed number of function evaluations. Implementing lattice-based Δ-DOGS with NPS improves the coverage speed as compared with Δ-DOGS(C) with NPS. After only 33 function evaluations, lattice-based Δ-DOGS with NPS finds a solution

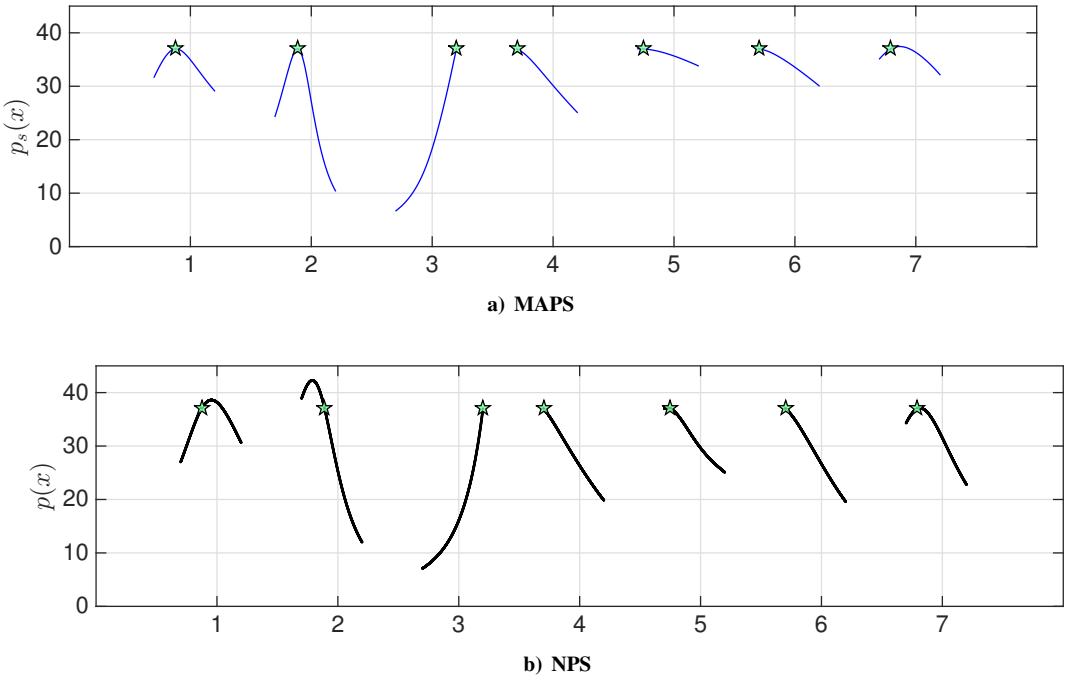


Figure 6: Performance of MAPS v.s. NPS give the first 30 datapoints generated using Algorithm 1 with NPS as well as the optimum solution x^*. Vertical axis: interpolant value for the efficiency of the foil with respect to the seven adjustable parameters. Horizontal axis: normalized design parameters listed in Table 1. a): multivariate adaptive polyharmonic spline (MAPS) surrogate (11). b): natural polyharmonic spline (NPS) surrogate (8).

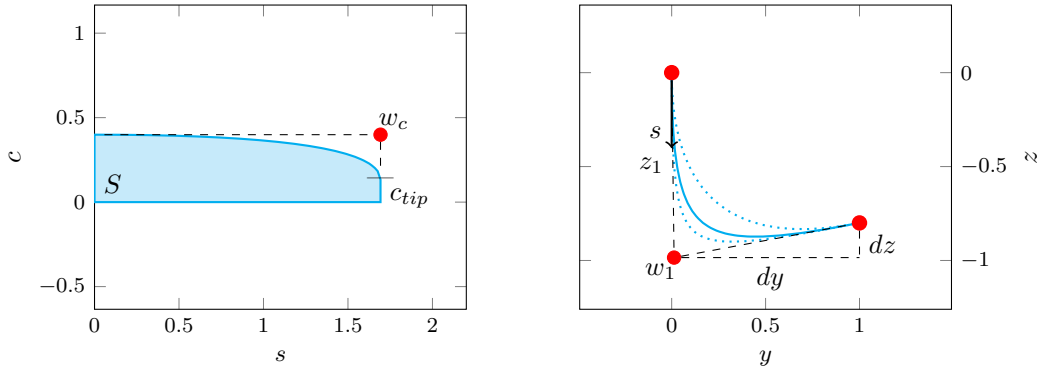


Figure 7: The optimization design parameters listed in Table 1 (see [29] for details of parametrization).

with a lift/drag ratio of 36; convergence to this level required 100 function evaluations using Δ-DOGS(C) with NPS. Moreover, using lattice-based Δ-DOGS with MAPS, a lift/drag ratio of 36 is achieved in only 26 function evaluations. That is, lattice-based Δ-DOGS with MAPS has 74% improvement compared to the Δ-DOGS(C) with NPS, and a 21% enhancement compared to lattice-based Δ-DOGS with NPS. Furthermore, after 55 function evaluations, lattice-based Δ-DOGS with MAPS found a solution with a lift/drag ratio of 36.89; on the other hand, [29] reported 160 function evaluations to reach a maximum lift/drag ratio of 36.81. Table 2 shows the optimized parameters of the hydrofoil design as computed in this paper, after various numbers of iterations, and in [29].

An important observations is that dy, the total length of the foil, reaches its maximum possible value in the optimized result. Intuition also suggests that, by increasing the foil aspect ratio, the foil efficiency will increase.

It can be seen in Figure 10 (top plot) that variables y_2 and z_2 do not vary after 30 function evaluations. This indicates it would be beneficial to project the optimization problem to a lower-dimensional parameter space, and to solve the optimization problem in that reduced parameter space. This idea, directly leveraging the MAPS formulation,

Table 2: Comparison between the results reported in [29] using Δ-DOGS(C) with NPS and lattice-based Δ-DOGS with MAPS for $dy < 1.50$.

variable	parameter	[29] (200 iterations)	40 iterations	80 iterations	200 iterations	269 iterations
x_1	S	0.305	0.305	0.305	0.297	0.303
x_2	z_1	0.89	0.900	0.875	0.862	0.881
x_3	dy	1.50	1.500	1.500	1.500	1.500
x_4	dz	-0.29	-0.300	-0.300	-0.300	-0.300
x_5	w_1	7.25	7.250	10.09	7.060	9.377
x_6	c_{tip}	0.21	0.163	0.129	0.100	0.050
x_7	w_c	2.58	2.896	2.896	2.814	3.220
$f(x)$	C_L/C_D	36.81	36.807	36.890	36.897	36.993

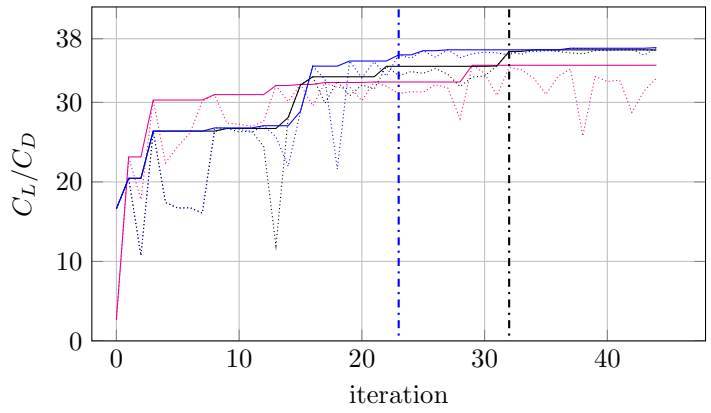


Figure 8: Solid lines show the best lift-drag ratio C_L/C_D at constant lift during the optimization: Blue curves illustrate the convergence of lattice-based Δ-DOGS using MAPS, Black curves illustrate the convergence of lattice-based Δ-DOGS using NPS, and Magenta curves illustrate the convergence of Δ-DOGS(C) using NPS [10], as originally reported in [29]. In all three cases, the point generated by Δ-DOGS at each iteration is shown by dotted curves.

will be investigated in future work.

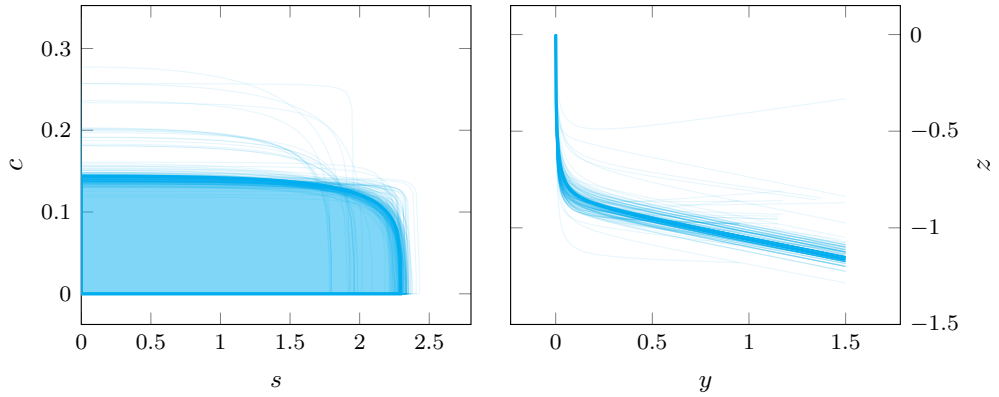
Convergence histories for the values of the optimized design parameters, using Δ-DOGS(C) with NPS [29] and lattice-based Δ-DOGS with MAPS, are shown in Figure 10. The designs achieved by both algorithms are illustrated in Figure 9. The final solutions obtained using both methods, illustrated in Figure 11, are quite similar. However, Figure 9a shows that the optimized design is found faster than in Figure 9b. These trends are also evident in Figure 10 and Figure 8.

V. Conclusions

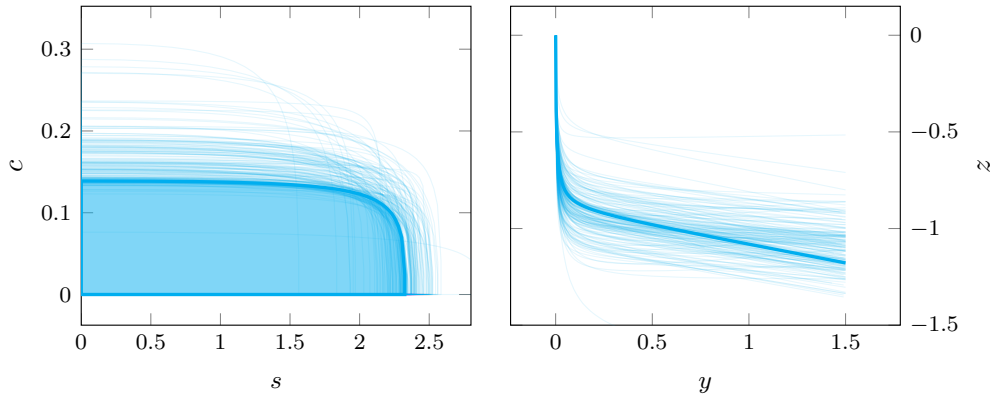
This paper presents a new interpolation strategy, multivariate adaptive polyharmonic spline (MAPS), for regularizing the interpolant used in response surface methods for derivative-free optimization. We have demonstrated that MAPS significantly accelerates the convergence rate of our own family of response surface methods, dubbed Δ-DOGS, when applied to practical design optimization problems of engineering interest.

MAPS is an interpolation strategy in the family of radial basis functions that rescales parameter space, based on the available data generated by an optimization algorithm, in order to reduce the spurious oscillations in the resulting interpolant. In the present work, MAPS is implemented in the Δ-DOGS family of optimization algorithms, and its performance is tested on a simulation-based shape design optimization problem to maximize the lift/drag ratio of a hydrofoil design. Results indicate fewer function evaluations are required following the new approach to achieve a given level of convergence. An improved method of coordinating the search with lattices is also introduced and shown to be effective.

A limitation of the interpolation strategy developed in this work is that its computational expense increases as the



a) lattice-based Δ-DOGS with MAPS



b) Δ-DOGS(C) with NPS

Figure 9: The first 200 evaluated foil geometries in the optimization algorithms: $a)$ lattice-based Δ-DOGS [3] with MAPS. $b)$ Δ-DOGS(C) with NPS [29]. The optimized geometry is shown by a thick curve in both figures.

number of data points increases, since at each iteration it needs to solve an optimization problem to find the scaling parameters based on the available datapoints. However, in practice, the determination of these scaling parameters is found to be relatively inexpensive as compared with the function evaluations themselves, which are typically determined from expensive CFD simulations.

In future work, we will implement the present algorithm on additional test problems, and more computationally efficient implementations of MAPS will be developed. Also, the use of MAPS for the problem of dimension reduction in Δ-DOGS will be explored; that is, extending MAPS to aid in the exploration of $f(x)$ over a reduced number of parameters during intermediate steps of the optimization procedure.

Appendix: Comparison of direct & regularized approaches to find the θ parameters of MAPS

We now provide a brief discussion about the inadequacy of the direct approach to find θ^* when one of the scaling parameters is much smaller that the other. Consider a model problem $f(x_1, x_2) = \rho\,(x_1 - 0.45)^2 + (x_2 - 0.45)^2$ the scaling parameter is found both using solution of (16) with BFGS (black curve) and solution of Algorithm 2 (blue curve). It is observed that even in a simple 2D problem using the same datapoints shown in Fig 3, as one of the scaling parameters approaches zero, the direct SQP with BFGS solution is not able to provide a correct scaling parameter and converges to a local minimum. In contrast, the approach used in Algorithm 2, even for large ρ, does not deviate from the optimum path of the solutions, and converges to a better solution. Algorithm 2 constantly initialize the solution of $Q_{\lambda_j}(\theta)$ for finding the $\theta_{\lambda_{k+1}}$ and enables Algorithm 2 to converge to the appropriate scaling solution.

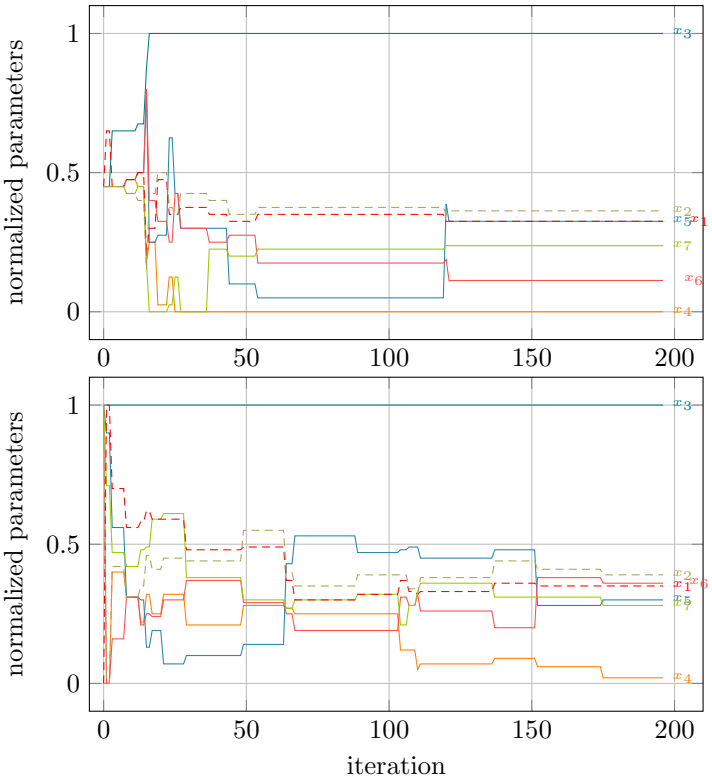


Figure 10: Convergence history of the optimal design parameter's values during the optimization. Top: lattice-based Δ-DOGS [3] with MAPS. Bottom: Δ-DOGS(C) [10] with NPS. (as it is reported in [29]). Dashed curves are the most dominant parameters, x_1 and x_2 in both plots.

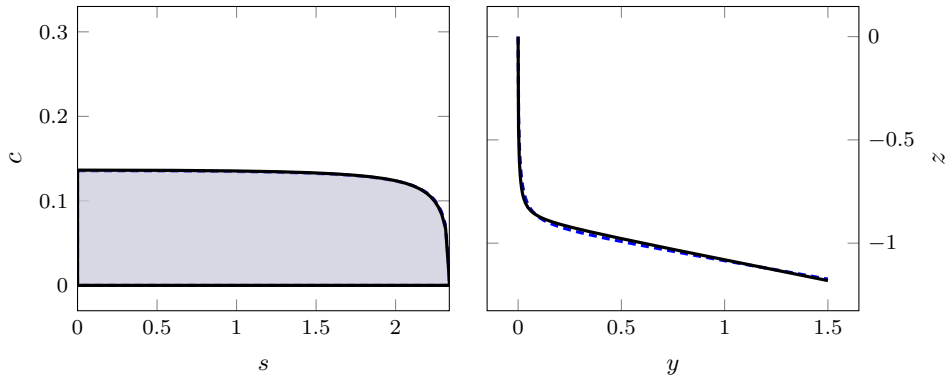


Figure 11: The optimum hydrofiol design using lattice-based Δ-DOGS with MAPS compared with the reported optimum foil in [29]. The dashed-blue is [29] and the solid line the optimum design reported in Table 2.

Acknowledgements

The authors gratefully acknowledge the funding from AFOSR FA 9550-12-1-0046, Cymer Center for Control Systems & Dynamics, and Leidos corporation in support of this work.

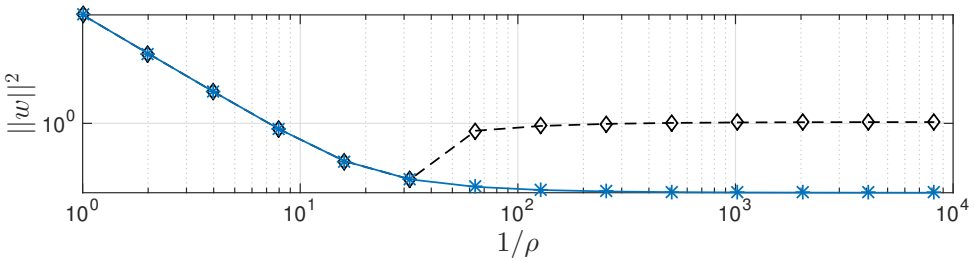


Figure 12: The black dashed line is the solution of (15) using direct approach and blue dashed-line is the solution of (15) using the iterative approach presented in Algorithm 2. Vertical axis is the smoothness criteria $\|w\|^2$, and the horizontal axis is the variation in x_1 direction respect to x_2 for a sample problem $f(x_1, x_2) = \rho\,(x_1 - 0.45)^2 + (x_2 - 0.45)^2$ both in loglog scale. In the plot $\frac{1}{\rho}$ is plotted. It is observed that for direct SQP with BFGS (black dashed line) converges to a local minimum of $\|w\|^2$ as $\rho \to 0$.

References

[1] Adjengue, L., Audet, C., and Yahia, I. B. (2014). *A variance-based method to rank input variables of the mesh adaptive direct search algorithm.* Optimization Letters, 8(5), 1599-1610.

[2] Alimohammadi, S., Beyhaghi, P., and Bewley. T. : *Delaunay-based Derivative-free Optimization via Global Surrogates, Part III: nonconvex constraints.* Journal of Global Optimization. Under review.

[3] Alimohammadi, S., Beyhaghi, P., and Bewley, T. : *Implementation of dense lattices to accelerate Delaunay-based optimization.* Under preparation.

[4] Alimohammadi. S., Cavaglieri, D., Beyhaghi, B., and Bewley, T. R. : *Discovery of an IMEXRK time integration scheme via Delaunay-based derivative-free global optimization, submitted to J. Opt. & Eng.*.

[5] Alimohammadi, S., Cavaglieri, D., Beyhaghi, P., and Bewley, T. R. (2016, November). *Application of a derivative-free global optimization algorithm to the derivation of a new time integration scheme for the simulation of incompressible turbulence.* In APS Division of Fluid Dynamics Meeting.

[6] Alimohammadi, S., and He, D. (2016, July). *Multi-stage algorithm for uncertainty analysis of solar power forecasting.* In Power and Energy Society General Meeting (PESGM), 2016 (pp. 1-5). IEEE.

[7] Alimohammadi, S., and Kleissl, J. P. (2015, November). *A new framework to increase the efficiency of large-scale solar power plants.* In APS Division of Fluid Dynamics Meeting.

[8] Belitz, P., and Bewley, T. (2013). *New horizons in sphere-packing theory, part II: lattice-based derivative-free optimization via global surrogates.* Journal of Global Optimization, 1-31.

[9] Beyhaghi, P., Alimohammadi, S., and Bewley, T. *A multiscale, asymptotically unbiased approach to uncertainty quantification in the numerical approximation of infinite time-averaged statistics.* Under preparation.

[10] Beyhaghi, P., and Bewley, T. (2016). *Delaunay-based derivative-free optimization via global surrogates, part II:convex constraints.* Journal of Global Optimization.

[11] Beyhaghi. P., and Bewley. T.: *Implementation of Cartesian grids to accelerate Delaunay-based Optimization.* Journal of Global Optimization. Under review.

[12] Beyhaghi. P., and Bewley. T.: *A derivative-free optimization algorithm for the efficient minimization of time-averaged statistics.* Journal of Computational Optimization and Applications. Under review.

[13] Beyhaghi, P., Cavaglieri, D., and Bewley, T. (2015). *Delaunay-based derivative-free optimization via global surrogates, part I: linear constraints.* Journal of Global Optimization.

[14] Booker, A. J., Dennis Jr, J. E., Frank, P. D., Serafini, D. B., Torczon, V., and Trosset, M. W. (1999). *A rigorous framework for optimization of expensive functions by surrogates.* Structural optimization, 17(1), 1-13.

[15] Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., and Evans, T. R. *Reconstruction and representation of 3D objects with radial basis functions.* In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (pp. 67-76). ACM. (2001, August).

[16] Choi, S., Alonso, J. J., Kroo, I. M., and Wintzer, M. (1997). *Multi-fidelity design optimization of low-boom supersonic business multi-fidelity design optimization of low-boom supersonic business jets.* In Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, No. AIAA 2004-4371 in AIAA Paper, 2004. 14 of 15 American Institute of Aeronautics and Astronautics 8 Alexandrov.

[17] Conn, A. R., and Le Digabel, S. (2013). *Use of quadratic models with mesh-adaptive direct search for constrained black box optimization.* Optimization Methods and Software, 28(1), 139-158.

[18] Constantine, P. G., Dow, E., and Wang, Q. (2014). *Active subspace methods in theory and practice: applications to kriging surfaces.* SIAM Journal on Scientific Computing, 36(4), A1500-A1524.

[19] Drela, M., and Giles, M. B. (1987). *Viscous-inviscid analysis of transonic and low Reynolds number airfoils.* AIAA journal, 25(10), 1347-1355.

[20] Drela, M., and Youngren, H. (2008). *Athena vortex lattice (AVL).* Computer software. AVL, 4. http://web.mit.edu/drela/Public/web/avl/.

[21] Duchon, J. (1977). *Splines minimizing rotation-invariant semi-norms in Sobolev spaces.* In Constructive theory of functions of several variables (pp. 85-100). Springer Berlin Heidelberg.

[22] Fuhry, M., and Reichel, L. (2012). *A new Tikhonov regularization method.* Numerical Algorithms, 59(3), 433-445.

[23] Hastie, T. , Tibshirani, R., Friedman, J., and Franklin J., *The elements of statistical learning: data mining, inference and prediction.* The Mathematical Intelligencer 27.2 (2005): 83-85.

[24] Hanke, M., and Hansen, P. C. (1993). *Regularization methods for large-scale problems.* Surv. Math. Ind, 3(4), 253-315.

[25] Golub, G., and Kahan, W. (1965). *Calculating the singular values and pseudo-inverse of a matrix.* Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis, 2(2), 205-224.

[26] Gill, P. E., Murray, W., and Saunders, M. A. (2005). *SNOPT: An SQP algorithm for large-scale constrained optimization.* SIAM review, 47(1), 99-131.

[27] Gutmann, H. M. (2001). *A radial basis function method for global optimization.* Journal of Global Optimization, 19(3), 201-227.

[28] Jones, D. R. (2001). *A taxonomy of global optimization methods based on response surfaces.* Journal of global optimization, 21(4), 345-383.

[29] Meneghello, G., Beyhaghi, P., and Bewley, T. : *Simulation-based optimization of the hydrofoil of a flying catamaran.* Journal of Ocean Engineering and Marine Energy. Under review.

[30] Marsden, A. L., Wang, M., Dennis Jr, J. E., and Moin, P. (2004). *Optimal aeroacoustic shape design using the surrogate management framework.* Optimization and Engineering, 5(2), 235-262.

[31] Rasmussen, C. E. (2006). *Gaussian processes for machine learning.*

[32] Riley, J. (1955). *Solving Systems of Linear Equations With a Positive Definite, Symmetric, but Possibly Ill-Conditioned Matrix.* Mathematical Tables and Other Aids to Computation, 9(51), 96-101.

[33] Wahba. G. (1990). *Spline Models for Observational Data*, Vol. 59. SIAM.

[34] Wild, S. M., Regis, R. G., and Shoemaker, C. A. (2008). *ORBIT: Optimization by radial basis function interpolation in trust-regions.* SIAM Journal on Scientific Computing, 30(6), 3197-3219.