Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 305 (2016) 562–578
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

A density-matching approach for optimization under uncertainty

Pranay Seshadri[a,*], Paul Constantine[b], Gianluca Iaccarino[c], Geoffrey Parks[a]

[a] *Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK*
[b] *Department of Applied Mathematics and Statistics, Colorado School of Mines, Golden, CO 80401, United States*
[c] *Mechanical Engineering Department, Stanford University, Stanford, CA 94305, United States*

Received 24 September 2014; received in revised form 2 March 2016; accepted 3 March 2016
Available online 24 March 2016

Abstract

Modern computers enable methods for design optimization that account for uncertainty in the system—so-called *optimization under uncertainty* (OUU). We propose a metric for OUU that measures the distance between a designer-specified probability density function of the system response (the *target*) and the system response's density function at a given design. We study an OUU formulation that minimizes this distance metric over all designs. We discretize the objective function with numerical quadrature, and we approximate the response density function with a Gaussian kernel density estimate. We offer heuristics for addressing issues that arise in this formulation, and we apply the approach to a CFD-based airfoil shape optimization problem. We qualitatively compare the density-matching approach to a multi-objective robust design optimization to gain insight into the method.

Keywords: Optimization under uncertainty; Design under uncertainty; Density-matching

1. Introduction

Modern computing power enables industrial-scale design optimization with high-fidelity numerical simulations of physical systems. Simulation-based design is found in aircraft [1], engine [2], automotive [3] and shipping [4] industries, among many others. To optimize, designers must precisely specify operating scenarios and manufactured production. Off-design operation and manufacturing tolerances are typically incorporated afterward. A more complete perspective on design optimization accounts for these uncertainties, e.g., by employing statistical performance metrics within the design optimization. This perspective leads to *optimization under uncertainty* (OUU).

The computational engineering literature is chock full of formulations and approaches for OUU. Allen and Maute [5] give an excellent overview that broadly categorizes these formulations as either *robust design optimization* (RBO) or *reliability-based design optimization* (RBDO). The essential idea behind RBO formulations is to

* Corresponding author.
E-mail addresses: ps583@cam.ac.uk (P. Seshadri), paul.constantine@mines.edu (P. Constantine), jops@stanford.edu (G. Iaccarino), gtp10@cam.ac.uk (G. Parks).
URL: http://inside.mines.edu/~pconstan (P. Constantine).

http://dx.doi.org/10.1016/j.cma.2016.03.006

simultaneously maximize a statistical measure of the system performance (e.g., the mean) while minimizing a statistical measure of system variability (e.g., the variance), thus improving robustness to variability in operating conditions. The optimization is often formulated with multiple objective functions (e.g., maximize mean and minimize variance), which leads to a Pareto front of solutions representing a trade-off between robustness and performance. Alternative formulations treat performance as the objective function and robustness as a constraint or vice versa. Some applications of RBO include the design of Formula One brake ducts [6], compressor blades [7], compression systems [8], airfoils [9], and structures [10]. The RBDO formulations seek designs that satisfy reliability criteria, such as maintaining a sufficiently small probability of failure, while minimizing a cost function of the design [11]. Estimating the failure probabilities within the optimization with randomized methods (e.g., Monte Carlo) can be prohibitively expensive for large-scale models; several methods exist for approximating regions of low failure probability [12]. Engineering examples of RBDO include transonic compressors [13], aeroelasticity [14], structures [5], and vehicle crash worthiness [15].

The statistical measures in the RDO and RBDO objective functions and constraints are typically low-order moments – e.g., mean and variance – or probabilities associated with the system response. The chosen statistical measures affect the optimal design, so they must be chosen carefully for each specific application.

In this paper, we propose an alternative OUU formulation. We assume the designer has described the desired system performance as a probability density function (pdf), which we call the *target pdf*, and we seek to minimize the distance between the design-dependent system response pdf and the target pdf. In other words, all criteria on the stochastic system's moments or failure probabilities are encoded in the target pdf. Mathematically, we treat the target pdf as given; it is not a tunable parameter. In any real-world scenario, this pass-the-buck attitude places tremendous responsibility on the designer to devise the perfect target pdf. We expect that a practical methodology including the proposed statistical measure will involve some back and forth between designer and optimizer to devise the most appropriate target pdf. Using a designer-specified response pdf has some precedent in the OUU literature. Rangavajhala and Mahadevan [16] assume a designer-specified pdf in their *optimum threshold design*, which finds thresholds that satisfy the given joint probability while allowing for preferences among multiple objectives.

Compared to other OUU formulations, density-matching is appropriate when the designer is able to specify her desiderata for the uncertain response as a pdf. The density-matching approach finds the design that best matches the designer's specified pdf, and there is no need to estimate the Pareto front of a multi-objective optimization (as in RDO) or minimize a failure probability (as in RBDO). Tolerated variability and failures are encoded in the target pdf.

We present a single-objective OUU formulation where the distance between target and response pdfs is the objective function. We explore some interesting properties of this optimization problem, namely how the objective's gradient behaves when the two pdfs are not sufficiently large on the same support (Section 2). We propose a consistent discretization of the objective function – based on numerical quadrature and kernel density estimation – that produces a continuous approximation well-suited for gradient-based optimization (Section 3). Our prior work uses histograms to approximate the response pdf, which leads to a less scalable optimization problem with integer variables [17]. There are some drawbacks to the density-matching formulation; we offer heuristics for addressing these drawbacks in Section 4. In Section 5, we test the formulation on an algebraic test problem and a shape optimization problem with the NACA0012 airfoil. In the latter case, we qualitatively compare the optimal designs to those generated by a multi-objective RDO strategy.

2. Mathematical formulation

Consider a function $f = f(s, \omega)$ that represents the response of a physical model with design variables $s \in \mathcal{S} \subseteq \mathbb{R}^n$ and random variables $\omega \in \Omega \subseteq \mathbb{R}^m$; the random variables represent the uncertainty in the physical system. The space $\mathcal{S}$ encodes the application-specific constraints on the design variables, e.g., bounds or linear inequality constraints. We assume that ω are defined on a probability space with sample space Ω and probability density function $p = p(\omega)$, which encode all available knowledge about the system's uncertainties.[1] We assume that f is scalar-valued, $f \in \mathcal{F} \subseteq \mathbb{R}$, and continuous in both s and ω. For a fixed $s \in \mathcal{S}$, let $q_s : \mathcal{F} \to \mathbb{R}_+$ be a probability density function of

[1] The final results depend on Ω and $p(\omega)$. If multiple probability density functions are consistent with the available information, then one should check the sensitivity of the results to perturbations in these quantities.

$f(s, \omega)$. We assume that $f(s, \omega)$ admits a square-integrable pdf for all values s in the design space $\mathcal{S}$. The shape of q_s will be different for different values of s.

The given target pdf expresses the designer's desired system performance accounting for uncertainty in operating conditions. Denote the target pdf by $t : \mathbb{R} \to \mathbb{R}_+$, which we assume is square-integrable. To find the values of the design variables s that bring the system's response as close as possible to the designer's target, we pose the following optimization problem:

$$s^* = \underset{s \in \mathcal{S}}{\operatorname{argmin}} \; d(t, q_s), \tag{1}$$

where $d(\cdot, \cdot)$ is a distance metric between two comparable probability density functions. The values s^* correspond to the optimal design under uncertainty.

A few comments on this optimization problem are in order. Since d is a distance metric, $d \geq 0$. However, $d(t, q_s)$ is not generally a convex function of s. Therefore, s^* may not be unique, and the optimization problem may need a regularization term to make it well-posed (e.g., Tikhonov regularization).

The minimum value of the objective function $d(t, q_{s^*})$ measures how well the optimal design meets the designer's specifications. A non-zero value at the minimum means that the model cannot completely satisfy the designer's specification. If the minimizing design's pdf is deemed too far from the target, then the model may need to be improved, e.g., by incorporating more controls or otherwise modifying the relationship between the design variables and the system behavior.

The formulation in (1) uses d as the sole objective function, and (1) has no constraints that depend on the uncertainties ω; constraints on the design variables are incorporated by the space $\mathcal{S}$. We study this formulation because of its simplicity. Given a target pdf, one could use the metric d as a one objective in a multi-objective RDO formulation or as a measure of reliability in an RBDO formulation. We do not pursue these ideas in this paper.

There are many possible choices for the distance metric d; Gibbs and Su [18] review several metrics and the relationships between them. To enable efficient, scalable gradient-based methods for the optimization (1), we choose the differentiable squared L_2-norm,

$$d(t, q_s) = \int_{-\infty}^{\infty} (t(f) - q_s(f))^2 \, df. \tag{2}$$

This integral is finite by the square-integrability assumption on t and q_s.

2.1. The trouble with non-overlapping response and target pdfs

Something peculiar happens to d from (2) when the supports of t and q_s do not overlap—i.e., t is zero if q_s is positive and vice versa. Expanding the integrand in (2),

$$d(t, q_s) = \int_{-\infty}^{\infty} t(f)^2 \, df - 2 \int_{-\infty}^{\infty} t(f) \, q_s(f) \, df + \int_{-\infty}^{\infty} q_s(f)^2 \, df. \tag{3}$$

Since the target t is independent of the design variables s, the minimizer of d is the same as the minimizer of d' defined as

$$d'(t, q_s) := \int_{-\infty}^{\infty} q_s(f)^2 \, df - 2 \int_{-\infty}^{\infty} t(f) \, q_s(f) \, df. \tag{4}$$

If the supports of t and q_s do not overlap, then the second term in (4) vanishes, and

$$d'(t, q_s) = \int_{-\infty}^{\infty} q_s(f)^2 \, df. \tag{5}$$

In words, when t and q_s do not overlap, the objective function has no information from the target t. The gradient of d' with respect to the design variables s may point in a direction that decreases d', but there is no guarantee that a step along that direction in the design space moves q_s closer (by the distance metric) to the target t. The following example illustrates this issue.

Example 1. Let $f(s, \omega) = s + \omega$, where ω is a random variable distributed uniformly on $[0, 1]$ and $s \in [0, 2]$. The response pdf $q_s(f)$ is a uniform density function on the interval $[s, s + 1]$. Let the target pdf $t(f)$ be a uniform density on the interval $[2, 3]$. The minimizer of (1) is $s^* = 2$, and $d(t, q_{s^*}) = 0$. However, for $s \in [0, 1)$, the first and second derivatives of $d(t, q_s)$ with respect to s are zero. Thus, a gradient-based optimization method would stop at any candidate minimizer in the interval $[0, 1)$.

The overlap issue forces us to make the following assumption to ensure that the minimizer from a gradient-based method applied to (1) with distance metric (2) produces a response pdf with some relationship to the given target pdf.

Assumption 1. Assume that for all $s \in \mathcal{S}$, the intersection of the support of the target pdf $t(f)$ and the support of the design-dependent response pdf $q_s(f)$ is non-empty.

Assumption 1 is sufficient but not necessary; it can be relaxed to (i) the initial design point produces a response pdf whose support overlaps the target's support and (ii) all iterates of the optimization method produce response pdfs whose supports overlap the target's support. For computation, we exploit choices in the kernel density estimates to ensure that Assumption 1 is satisfied. This approach also suggests a heuristic to accelerate the numerical optimization; see Section 4.1.

In principle, our construction can be extended to f's that return a vector of responses from the system. However, this case requires (i) a joint probability density for the target, (ii) a multivariate density estimation method for the response pdf, and (iii) multivariate integration to compute the distance metric. Thus, the approach suffers the dreaded curse of dimensionality as the number of components in f increases. The density-matching approach is not appropriate when the number of responses is more than two or three.

3. Discretization and computation

We turn to the computational aspects of solving the optimization problem (1) using the distance metric (2). There are two main issues to address: (i) discretizing the integral in the distance metric and (ii) estimating the response density q_s.

3.1. Discretizing the distance metric

To avoid issues with numerical integration on unbounded domains, we assume that $f(s, \omega)$ is bounded for all s and ω,

$$f_\ell \leq f(s, \omega) \leq f_u, \quad s \in \mathcal{S},\ \omega \in \Omega. \tag{6}$$

This implies that the support of $q_s(f)$ is always finite. Such an assumption is not terribly restrictive. For a particular design point, q_s may have a long tail, but any computer representation of this long tail necessarily imposes finite bounds. The bounds f_ℓ and f_u need not be tight. But finite bounds help us devise a practical discretization. The bounds imply

$$d(t, q_s) = \int_{f_\ell}^{f_u} (t(f) - q_s(f))^2 \, df + \int_{-\infty}^{f_\ell} t(f)^2 \, df + \int_{f_u}^{\infty} t(f)^2 \, df. \tag{7}$$

Since the target pdf t is independent of the design variables s, the optimization can ignore the last two terms in (7).

We choose an N-point numerical quadrature rule on the interval $[f_\ell, f_u]$ with points $\gamma_i \in [f_\ell, f_u]$ and associated weights w_i with $i = 1, \ldots, N$. The number N of points in the integration rule can be very large without a large computational burden. We need to evaluate the given target pdf t and an estimate of the response pdf q_s at each quadrature node, but this is very cheap. The discretized objective function is

$$\hat{d}(t, q_s) = \sum_{i=1}^{N} (t(\gamma_i) - q_s(\gamma_i))^2 \, w_i = (\mathbf{t} - \mathbf{q}_s)^T \boldsymbol{W} (\mathbf{t} - \mathbf{q}_s), \tag{8}$$

where

$$\boldsymbol{W} = \begin{bmatrix} w_1 & & \\ & \ddots & \\ & & w_N \end{bmatrix}, \qquad \mathbf{t} = \begin{bmatrix} t(\gamma_1) \\ \vdots \\ t(\gamma_N) \end{bmatrix}, \qquad \mathbf{q}_s = \begin{bmatrix} q_s(\gamma_1) \\ \vdots \\ q_s(\gamma_N) \end{bmatrix}. \tag{9}$$

The resolution of the points $\{\gamma_i\}$ should be fine enough to resolve both the target t and the response pdf q_s for all $s \in \mathcal{S}$. In an extreme example of insufficient resolution, the support of t may be entirely inside an interval defined by two neighboring quadrature nodes. In such a case, t does not affect the discretized objective $\hat{d}$ in (8). If the support of t or a particular q_s is very small relative to the interval $[f_\ell, f_u]$, then one might consider a non-uniform distribution of quadrature nodes to properly resolve the pdfs. However, as noted, an extremely fine grid does not greatly increase the cost of evaluating the discretized objective function. So resolution – and, consequently, discretization error in the integral from (2) – is not a primary concern.

The choice of quadrature rule depends on the smoothness of the target t and the response q_s (or, its estimate) [19]. If these pdfs are very smooth on the interval $[f_\ell, f_u]$, then one could use high-order, interpolatory quadrature rules like Gaussian quadrature or Clenshaw–Curtis quadrature [20]. However, say the target is a uniform density on a small interval. Then a low-order method like the trapezoidal rule is more appropriate. We prefer a highly resolved trapezoidal rule in general; recent analysis shows that it compares well to high-order methods for smooth functions [21].

3.2. Estimating the response density

For a fixed design point $s \in \mathcal{S}$, the density q_s is, in general, not a known function of f and must be estimated. We propose to use a kernel density estimate for q_s [22]. We draw a set of M points $\{\omega_j\}$ independently according to the given density $p(\omega)$ on the random variables representing uncertainty. Define the functions

$$f_j(s) = f(s, \omega_j), \quad j = 1, \ldots, M. \tag{10}$$

For a bandwidth parameter h and a radial kernel $K(\cdot)$ that depends on h, we approximate q_s by

$$q_s(f) \approx \hat{q}_s(f) = \frac{1}{M} \sum_{j=1}^{M} K(f - f_j(s)). \tag{11}$$

We approximate the vector $\mathbf{q}_s$ from (8) as

$$\mathbf{q}_s \approx \hat{\mathbf{q}}_s = \boldsymbol{K}\mathbf{e}, \quad \boldsymbol{K} \in \mathbb{R}^{N \times M}, \tag{12}$$

where

$$\boldsymbol{K}_{ij} = \frac{1}{M} K(\gamma_i - f_j(s)), \quad i = 1, \ldots, N, \ j = 1, \ldots, M, \tag{13}$$

and $\mathbf{e}$ is an M-vector of ones. For computation, we replace $\mathbf{q}_s$ by $\hat{\mathbf{q}}_s$ in the approximate objective function $\hat{d}$ in (8).

For a sufficiently small h, the asymptotic mean-squared error in the kernel density estimate decreases as the number M of samples increases [22, Chapter 6]. In practice, one can increase the bandwidth parameter h to create smooth estimates that compensate for too few samples. Such heuristics are appropriate, since our goal is not perfect representation of the response density q_s. Our goal is to find a design point s^* whose corresponding response pdf is sufficiently close to the given target pdf. Nevertheless, if the number n of components in ω is large, then one might be concerned that M is not large enough to represent the system response over the high-dimensional space Ω, resulting in a poor approximation of q_s—potentially poor enough to adversely affect the optimization. This concern is valid when evaluating f is computationally expensive, e.g., if the system involves complex computational fluid dynamics model, thus limiting M. In this case, we might construct a response surface of $f(s, \omega)$ as a function of ω to sample in place of the true system response. We discuss the benefits and drawbacks of response surfaces in Section 4.2.

We propose a Gaussian kernel for the density estimate,

$$K(r) = \frac{1}{\sqrt{2\pi}} \exp\left(-(r/h)^2/2\right). \tag{14}$$

There are two main advantages for using a Gaussian kernel. First, it is differentiable at all points in its infinite domain, so we can take the derivative of the density estimate without worrying about non-differentiability at kernel support boundaries; many compactly supported kernels do not enjoy such an advantage. Second, the infinite support of the kernel implies that, at least mathematically, we can address the non-overlapping issue discussed in Section 2.1 and satisfy Assumption 1. It might seem inconsistent to use a kernel with infinite support when we assume that the support of q_s is bounded according to (6). But approximating compactly supported densities with Gaussian kernel density estimates is common; we can control the error in the tails to keep it from heavily influencing the objective function in (8). In Section 4.1, we propose a heuristic that exploits the freedom in the bandwidth parameter to help ensure that the q_s's kernel estimate is sufficiently large on the target pdf's support.

3.3. Computing the gradient

We can compute the gradient of the approximate objective $\hat{d}$ with respect to the design variables s. For the kth component of s, denoted s_k,

$$\frac{\partial \hat{d}}{\partial s_k} = 2\,(\mathbf{t} - \hat{\mathbf{q}}_s)^T\, \boldsymbol{W} \left(\frac{\partial \hat{\mathbf{q}}_s}{\partial s_k} \right). \tag{15}$$

Note that

$$\frac{\partial \hat{q}_s}{\partial s_k} = \frac{1}{M} \sum_{j=1}^{M} K'(f - f_j(s))\, \frac{\partial f_j}{\partial s_k}, \tag{16}$$

where K' is the derivative of the kernel with respect to its argument, which is easily computed from (14). To reiterate, the partial derivative $\partial f_j / \partial s_k$ is the derivative of the response $f(s, \omega)$, with $\omega = \omega_j$, with respect to the kth design variable s_k. This partial derivative is a function of the design variables. Define

$$\mathbf{f}'_k = \begin{bmatrix} \dfrac{\partial f_1}{\partial s_k} \\ \vdots \\ \dfrac{\partial f_M}{\partial s_k} \end{bmatrix}, \qquad \boldsymbol{K}'_{ij} = \frac{1}{M} K'(\gamma_i - f_j(s)). \tag{17}$$

Then we can concisely write the derivative of $\hat{d}$ from (8) with respect to the kth component of s as

$$\frac{\partial \hat{d}}{\partial s_k} = 2\ (\mathbf{t} - \boldsymbol{K}\mathbf{e})^T\ \boldsymbol{W}\boldsymbol{K}'\mathbf{f}'_k. \tag{18}$$

Define the $M \times n$ Jacobian matrix $\boldsymbol{F}'$ by

$$\boldsymbol{F}' = \begin{bmatrix} \dfrac{\partial f_1}{\partial s_1} & \cdots & \dfrac{\partial f_1}{\partial s_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_M}{\partial s_1} & \cdots & \dfrac{\partial f_M}{\partial s_n} \end{bmatrix}. \tag{19}$$

We can write the gradient of the objective function – oriented as a row vector – as

$$\nabla_s \hat{d} = 2\ (\mathbf{t} - \boldsymbol{K}\mathbf{e})^T\ \boldsymbol{W}\boldsymbol{K}'\boldsymbol{F}'. \tag{20}$$

The elements of $\boldsymbol{K}$, $\boldsymbol{K}'$, and $\boldsymbol{F}'$ all depend on s. Recall the dimensions of the terms in (20). The gradient vector $\nabla_s \hat{d}$ has n components, which is the number of random variables describing the system's uncertainty. The vector $\mathbf{t}$ has N components, which is the number of quadrature nodes from (8); we expect this to be a very large number. The matrix $\boldsymbol{K}$ has size $N \times M$, where M is the number of randomly chosen points in Ω used to estimate the pdf q_s. If evaluating the response is cheap, or if the response is approximated by a response surface, then M may also be very large. The

vector $\mathbf{e}$ of ones has length M. The diagonal matrix $\boldsymbol{W}$ has N nonzero elements on the diagonal. The matrix $\boldsymbol{K}'$ has size $N \times M$, and the matrix $\boldsymbol{F}'$ has size $M \times n$. Our numerical studies have not needed special methods to evaluate the matrix–vector products in (20). However, with a Gaussian kernel, we could perform extremely large computations (i.e., large M and N) with a fast Gaussian transform [23].

3.4. Interfaces and cost

In terms of interfaces to the simulation code, we need to evaluate (i) f given s and ω, and (ii) the gradient $\nabla_s f$ given s and ω—similar to a deterministic optimization. In this sense, the approach is *non-intrusive*. If we use a gradient-based optimization algorithm, such as a sequential quadratic program [24, Chapter 16], then each iteration uses M evaluations of f and its gradient with respect to s.

4. Computational heuristics

In this section, we discuss two heuristics for the optimization in (1). The first is an approach to the kernel bandwidth selection that alleviates the non-overlapping issue discussed in Section 2.1. The second is the use of response surfaces in place of the true response for expensive simulations. We end this section with a short discussion of some implementation details.

4.1. Bandwidth parameter and the overlap problem

There is a great deal of work on the proper bandwidth choice in kernel density estimation [22, Chapter 6]. In most statistical inference, the data determines the bandwidth parameter [25]. Our goal is somewhat different. Indeed, we want a reasonable estimate of the response pdf q_s at a design point s, where we can treat the set of scalars $\{f_j(s)\}$ as data. However, we can also use the bandwidth parameter to help ensure sufficient overlap between the estimate $\hat{q}_s$ and the target pdf t, thus aiding the optimization. Mathematically, by using the Gaussian kernel in (14) with infinite support, the kernel estimate $\hat{q}_s$ is strictly positive over the entire support of t. Numerically, the value of $\hat{q}_s$ may be too small on t's support to produce a useful gradient.

If t's support resides in $\hat{q}_s$'s tail, then increasing the bandwidth h increases $\hat{q}_s$ over t's support. Loosely speaking, we can use a large h to help the response pdf find the target pdf. The large h produces estimates of q_s that are too smooth with large half-widths. But the large width produces useful gradients for the optimizer to help bring the response pdf closer (in terms of the L_2 distance) to the target. For unimodal response pdfs, this leads to more overlap between the target and the response. Once the optimizer has found a region of the design space $\mathcal{S}$ where there is sufficient overlap, we reduce h to data-driven values to better estimate q_s. For a kernel estimate of a Gaussian pdf using a Gaussian kernel, the optimal bandwidth is

$$h_{\text{opt}} = \left(\frac{4}{3M}\right)^{1/5} \sigma, \tag{21}$$

where σ is the Gaussian's standard deviation. The formula (21) is known as Scott's rule [26,22]. In the first few optimization iterations, we use an initial bandwidth $h = (f_u - f_\ell)/5$. Once we are satisfied that the design point s yields an estimate of q_s whose support sufficiently overlaps t, we reduce the bandwidth to $h = h_{\text{opt}}$. The results in Section 5.2 use this heuristic.

4.2. Response surfaces

When the simulation is expensive, the number m of random variables is sufficiently small, and the response $f(s, \omega)$ is a sufficiently smooth function of ω, it may be more efficient to use a response surface when approximating the pdf q_s. Response surfaces for approximating pdfs are common in uncertainty quantification [27, Chapter 13]. Popular response surface constructions include polynomial approximations [28, Chapter 7] [29, Chapter 3] and radial basis approximations [30]. The essential idea is, for a fixed design point s, evaluate $f(s, \omega)$ at a few points $\omega_k \in \Omega$

with $k = 1, \ldots, P$, where P is smaller than the number of points needed to accurately estimate the pdf q_s; let $f_k(s) = f(s, \omega_k)$. Most response surface constructions are linear models of the data,

$$f(s, \omega) \approx \tilde{f}(s, \omega) = \sum_{k=1}^{P} a_k(\omega)\, f_k(s), \tag{22}$$

where the coefficients $a_k(\omega)$ depend on type of response surface. The approximation can be cheaply sampled by computing the coefficients $a_k(\omega)$ at the M points in Ω needed to estimate q_s. This approximation introduces additional error in the estimate of q_s, and one should validate that $\tilde{f}$ is sufficiently accurate—e.g., that P is large enough to produce a good approximation. Asymptotically, an L_2-convergent response surface implies convergence in distribution, i.e., the pdf of $\tilde{f}$ converges to the pdf of f [31, Chapter 2]. However, this well-known result does not account for the finite sampling used to estimate the pdfs, and asymptotic results do not always give confidence when P is small.

The airfoil example in Section 5.2 has a smooth response that is a function of one parameter representing uncertainty, and the system uses a relatively expensive CFD solver in two spatial dimensions. We use a response surface – validated several points in the design space – for both the response and its partial derivatives with respect to the design variables.

4.3. Implementation details

In this section, we collect the pieces needed to implement a numerical solver for the optimization (1). This summarizes the method and provides some details about the choices we make in implementation.

Optimization package. From MATLAB's Optimization Toolbox, we use the `fmincon` function with `Algorithm` option set to `sqp` (sequential quadratic program). An open source alternative is SciPy's `minimize` function from its Optimization package with the `method` set to `SLSQP` (sequential least-squares quadratic program). We provide subroutines for computing the objective function, implemented as (8) with the approximation (12), and the objective's gradient, implemented as (20).

Numerical integration. We hand code a trapezoidal rule to evaluate the objective function (8). This includes forming the diagonal matrix $\boldsymbol{W}$ in (8) and (20).

Kernel density estimation. From MATLAB's Statistics and Machine Learning Toolbox, we use the `ksdensity` function for kernel density estimation. The default kernel is the Gaussian as in (14), and the interface takes an optional bandwidth parameter argument, which we use to implement the heuristic described in Section 4.1. We hand code the gradient of the Gaussian kernel to compute $\boldsymbol{K}'$ in (20). An open source alternative is SciPy's `gaussian_kde` function, which also accepts a user-specified bandwidth parameter.

Response surfaces. The variety of response surface types and applications make it difficult to create a general purpose toolbox for response surfaces. For the specific experiment in this paper, we use a hand-coded least-squares-fit 5th degree global polynomial approximation for the univariate response. Since the airfoil application has only a single parameter representing uncertainty, we are able to visually validate the response surface quality with plots.

5. Numerical examples

We use two numerical examples to study the characteristics of the proposed density-matching OUU approach. The first is a simple response function that produces surprisingly complex behavior in the optimization. The second is an airfoil shape optimization problem with uncertainty in the freestream Mach number.

5.1. Simple response function

Consider the model

$$f(s, \omega) = s\omega + 3.5, \tag{23}$$

where $s \in \mathbb{R}$ and ω is a standard normal random variable, so f is a normal random variable with mean 3.5 and standard deviation s. The goal is to find s that minimizes the L_2-norm distance between the pdf of $f(s, w)$ and a

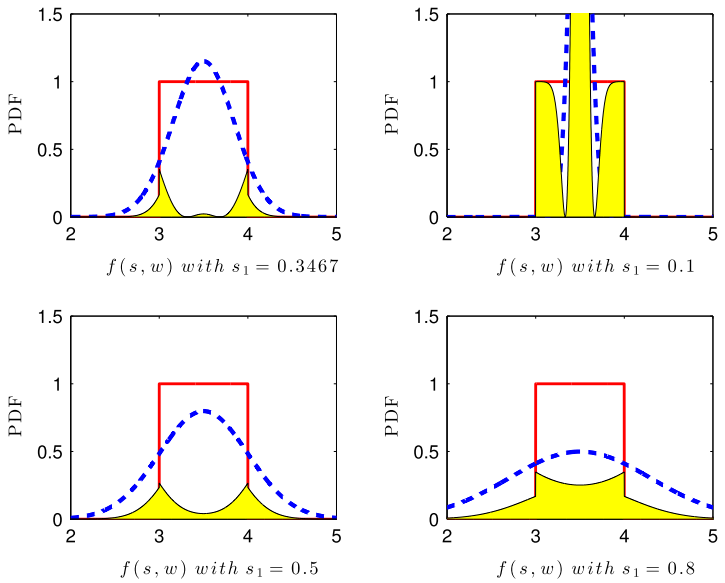

Fig. 1. Response (23) with different values of the design parameter s. The red rectangle is the uniform target pdf and the blue dashed line is the response pdf. The goal is to minimize the yellow area in the figures. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

uniform target pdf

$$t(f) = \begin{cases} 1, & \text{for } f \in [3,4], \\ 0, & \text{elsewhere.} \end{cases} \tag{24}$$

The pdf of f is

$$q_s(f) = \frac{1}{\sqrt{2\pi}s} \exp\left(\frac{-(f-3.5)^2}{2s^2}\right). \tag{25}$$

The objective function is

$$\begin{aligned} d(t, q_s) &= \int_{-\infty}^{\infty} (t(f) - q_s(f))^2 \, df \\ &= \int_{-\infty}^{3} q_s(f)^2 \, df + \int_{4}^{\infty} q_s(f)^2 \, df + \int_{3}^{4} (1 - q_s(f))^2 \, df. \end{aligned} \tag{26}$$

Fig. 1 plots the two densities and their squared difference. The red density is the target pdf and the blue dashed line is the response pdf. The yellow-shaded region is the squared difference in (26). Fig. 1 shows the effect of varying s on the yellow region, with $s = 0.3467$ yielding the smallest area under the curve, minimizing the distance (26). Note the complexity of the integrand. Table 1 shows the effect of varying the number of quadrature points. Above 1000 points the error in the optimal s is in the fourth decimal place. Table 2 repeats this study with kernel density estimates for the response pdf.

5.2. Airfoil design

Next we apply the density matching scheme to the design of an airfoil under uncertainty. MATLAB and Python codes used for this numerical study can be found at https://github.com/psesh/density-matching. The airfoil used in this example is a NACA0012 at an angle of attack of 5°. The uncertainty is in the inlet Mach number, which is characterized by a $\beta(2, 2)$ distribution between Mach numbers of 0.66 and 0.69. Flow computations for this airfoil are carried out by solving the compressible Euler equations using Stanford University's SU2 flow solver [32]. The airfoil

Table 1
Effect of the number of quadrature points for the simple problem (23).

Number of quadrature points	Optimal s	Minimum distance
10	0.3531	0.00670
100	0.3556	0.017385
1000	0.3469	0.016048
10,000	0.3467	0.016022
100,000	0.3467	0.016023

Table 2
Effect of the number of quadrature points and kernel density estimate (KDE) samples for the simple problem (23).

Number of quadrature points	Optimal s	Minimum distance	Number of samples
10	0.3488	0.006759	10^5
10	0.3505	0.006729	10^6
100	0.3513	0.01743	10^5
100	0.3537	0.01741	10^6
1000	0.3464	0.01610	10^5
1000	0.3465	0.01610	10^6
10,000	0.3466	0.01608	10^5
10,000	0.3467	0.01605	10^6

Table 3
Hicks–Henne bump function heights and locations as a proportion of chord. The heights and locations are shown for the upper surface; the lower surface has the same parameterization.

Location	Bump amplitude
0.05	±0.0007
0.15	±0.0030
0.30	±0.0090
0.40	±0.0090
0.55	±0.0090
0.65	±0.0060
0.75	±0.0030
0.90	±0.0007

is parameterized with 16 Hicks–Henne bump functions: 8 on the upper surface and 8 on the lower surface. The design space is the height of each bump; a point in the design space produces a perturbation from the NACA0012 shape. The height ranges and locations for the bumps are shown in Table 3.

For a point in the design space, the airfoil mesh is deformed using a torsional spring analogy. The flow solver runs on the new mesh producing the lift-to-drag ratio L/D, which is the response of interest. To connect to the notation in Sections 2 and 3, the response f is L/D, the design variables s are the 16 Hicks–Henne bump heights, and the random variable ω is the Mach number with a $\beta(2, 2)$ density on the interval [0.66, 0.69].

5.3. Robust design optimization

For qualitative comparison, we perform a multi-objective RDO on the same physical model, though we emphasize that this is a different optimization problem than the density-matching objective. A design that is optimal with respect to the following RDO formulation need not be optimal with respect to a density-matching formulation with a given target and vice versa. The RDO problem is

$$\underset{s\in\mathcal{S}}{\text{minimize}} \quad (\mathbb{E}\left[L/D\right])^{-1} \quad \text{and} \quad \text{Var}\left[L/D\right], \tag{27}$$

Table 4
NSGA-II algorithm parameters used for RDO.

Parameters	Value
Population size	100
Number of generations	35
Crossover probability	0.9
Mutation probability	0.0625
Crossover distribution index	20
Mutation distribution index	20

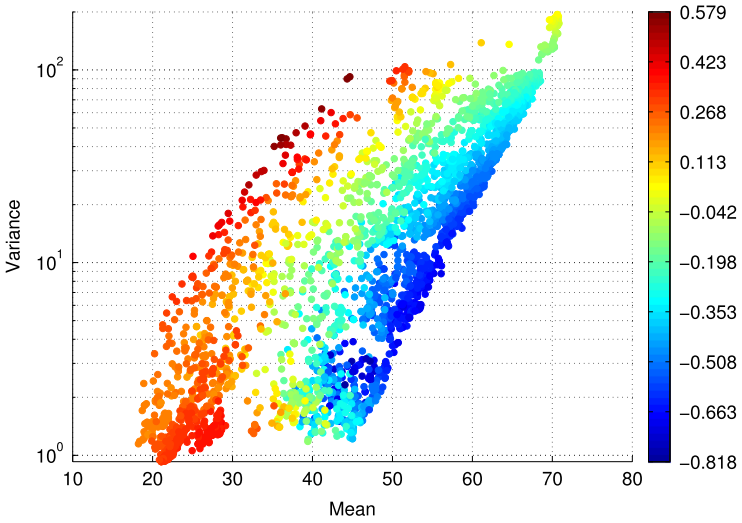

Fig. 2. Results of the RDO airfoil design problem. Mean and log-variance of L/D are plotted on the horizontal and vertical axes, respectively. Individual designs are colored by their skewness values as shown on the color bar legend. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where $\mathbb{E}[\cdot]$ is the mean and $\mathrm{Var}[\cdot]$ is the variance. These two moments both depend on the design variables s that parameterize the airfoil shape. We use the genetic algorithm NSGA-II [33] to estimate the Pareto front for (27). The default parameters for NSGA-II are shown in Table 4.

We use a population size of 100 with 35 generations, yielding a total of 3500 function calls from the optimizer. Each function call uses 21 CFD computations to fit a least-squares 5th degree polynomial response surface, which is used to estimate the objectives in (27). Thus, the RDO study used a total of $3500 \times 21 = 73{,}500$ CFD runs.

Fig. 2 plots the moments from each computation. The mean is plotted on the horizontal and the variance on the vertical on a logarithmic scale. The nominal NACA0012 design has a mean L/D ratio of 27.2356 and a variance of 7.4615. Fig. 2 indicates the skewness values of individual designs by the marker color. Negatively skewed designs are blue while positively skewed designs are red. The RDO took approximately two days to run on an 8-core workstation. It should be noted that the L/D values are very high here because drag predictions from an Euler code do not include viscous and induced contributions.

5.4. Density-matching with a designer-specified target

We begin with some details of the density-matching optimization applied to the airfoil design. The initial design for all cases is the NACA0012 airfoil. To compute the response pdf's kernel density estimates, we draw 10^5 independent samples from a least-squares-fit 5th degree polynomial response surface of L/D as a function of Mach number. We fit the response surface with $P = 21$ flow computations at uniformly spaced Mach numbers between 0.66 and 0.69; the 21 independent runs were executed in parallel. The nominal design (the NACA0012 airfoil) produces a response pdf that is far from each of synthetic targets described below; see Fig. 3 for a comparison of the nominal design pdf with

Table 5
Density-matching parameters.

Quantity	Definition	Value
N	quadrature points	2500
M	random samples	10^5
n	design parameters	16
$h_{\text{stage 1}}$	stage 1 bandwidth parameter	50.0
$h_{\text{stage 2}}$	stage 2 bandwidth parameter	1.0
f_ℓ	lower bound for L/D	-100
f_u	upper bound for L/D	150
K_h	kernel function	Gaussian

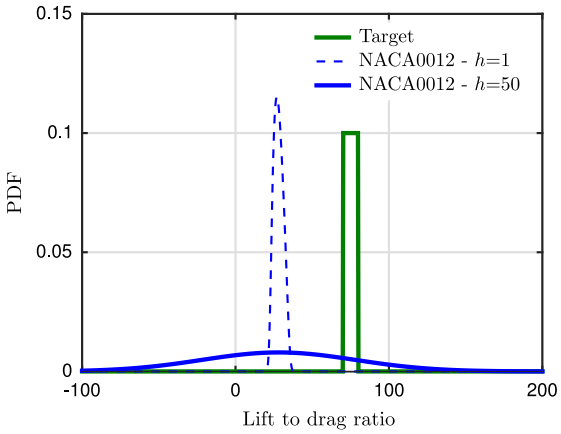

Fig. 3. Initial (NACA0012) design pdfs and the uniform target. The initial design is shown with bandwidth parameter values of 1 and 50.

two different bandwidths and one of the targets. We use the two-stage heuristic discussed in Section 4.1. We run 3 iterations of the SQP solver with a large bandwidth of $h = 50$. The remaining iterations used a bandwidth $h = 1$—slightly larger than h_{opt} in (21). We use a trapezoidal rule with $N = 2500$ quadrature points on the interval $[-100, 150]$ in the objective function (8). The negative lower bound on L/D accommodates kernel density estimates with large bandwidths. We tested larger values of N, but they did not lead to substantial improvement in the optimized design.

The gradient in (20) includes the matrix of kernel evaluations $\boldsymbol{K}$, their derivatives $\boldsymbol{K}'$, and design parameter sensitivities in the matrix $\boldsymbol{F}'$. The size of these matrices is determined by the number of quadrature points and the number of random samples used to estimate the pdfs. Both $\boldsymbol{K}$ and $\boldsymbol{K}'$ have dimensions 2500×10^5. We used SU2's adjoint capabilities at the $P = 21$ Mach numbers to compute partial derivatives of L/D with respect to 16 bump heights. We used a least-squares-fit 5th degree polynomial response surface to approximate the partial derivatives at all the sample points needed to compute the elements of $\boldsymbol{F}'$ in (19). The response surfaces for all 16 partial derivatives are shown in Fig. 4 for a perturbed NACA0012. From these and other similar plots, we determined that the polynomial response was sufficiently accurate. We used least-squares to avoid interpolating noisy partials, such as parameter 12 in Fig. 4. Table 5 summarizes the parameters used in the optimization.

We assume that the designer has provided us with three target pdfs for the response L/D:

1. a uniform density in the interval [75, 80],
2. a Gaussian density with mean 50 and variance 10,
3. a $\beta(1.5, 3.5)$ density on the interval [50, 80].

In what follows, we describe the results for each of these three cases.

5.4.1. Uniform target

For the uniform target, we repeat the density-matching optimization four times to see the effects of the random sampling used to estimate the response pdf at each optimization iteration. The four independent trials produced nearly

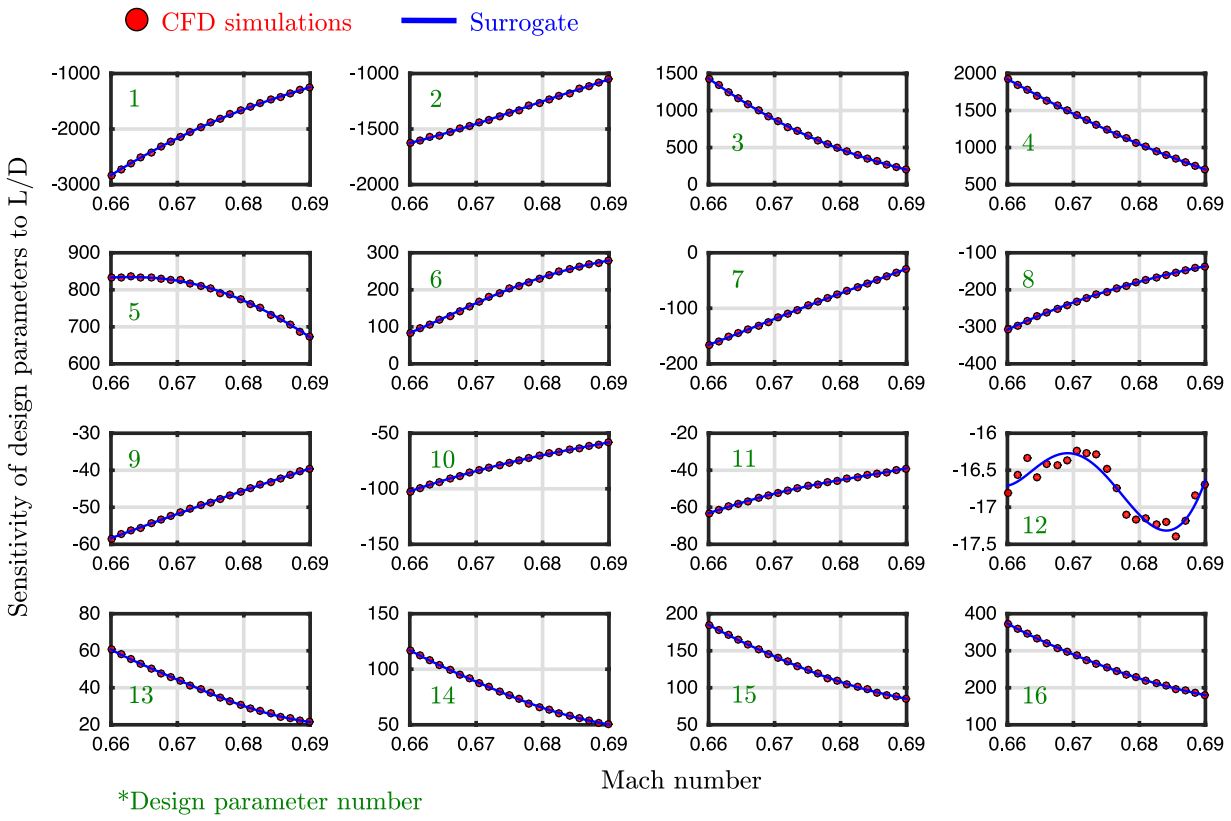

Fig. 4. Polynomial response surfaces and adjoint-based partial derivatives of L/D with respect to Mach number for a random design point.

identical results. The results with the uniform target are shown in Fig. 5. Fig. 5(a) compares the kernel density estimate for the initial design with bandwidth $h = 50$ (blue), the uniform target (green), and the final designs from stage 1 of the optimization using bandwidth $h = 50$ across all four independent trials (red). Using the final stage 1 designs with bandwidth $h = 1.0$ produces the red density estimates shown in Fig. 5(b); these are the initial designs for stage 2 using bandwidth $h = 1$. The final optimized designs from stage 2 produce the black density estimates in Fig. 5(b). We plot the final designs from both stages in Fig. 5(c) to compare them with the RDO designs obtained in Section 5.3. In this case, the results from both stage 1 and stage 2 lie well beyond the Pareto front obtained from NSGA-2.

Optimization convergence histories for all four tests are shown in Fig. 5(d). All objective function values are normalized by the initial objective value. Stage 1 for this target took an average of 1 h and 34 min while stage 2 took an average of 3 h and 36 min. While there are some differences in the convergence histories, the differences in the final designs are very minor across the four independent trials. The final designs from stage 2 are plotted in Fig. 5(e) with a close-up in Fig. 5(f).

5.4.2. *Gaussian target*

Fig. 6 shows results with the Gaussian target similar to Fig. 5. In this case, we ran a single test instead of four independent trials. Fig. 6(a) shows the initial and final designs from stage 1, and Fig. 6(b) shows the initial and final designs for stage 2. Note that the final design for stage 1 is the initial design for stage 2. The final design is extremely close to matching the target. Fig. 6(d) shows that changes in the objective function value are negligible after 12 optimization calls in the second stage. Here stage 1 took 46 min while stage 2 took 5 h and 12 min.

5.4.3. *Beta target*

With beta density, we examine a case where the target is positively skewed. For this case, we use only one trial as in the Gaussian target case. The results are shown in Fig. 7. The target has mean 59.0 and variance 31.5. The initial

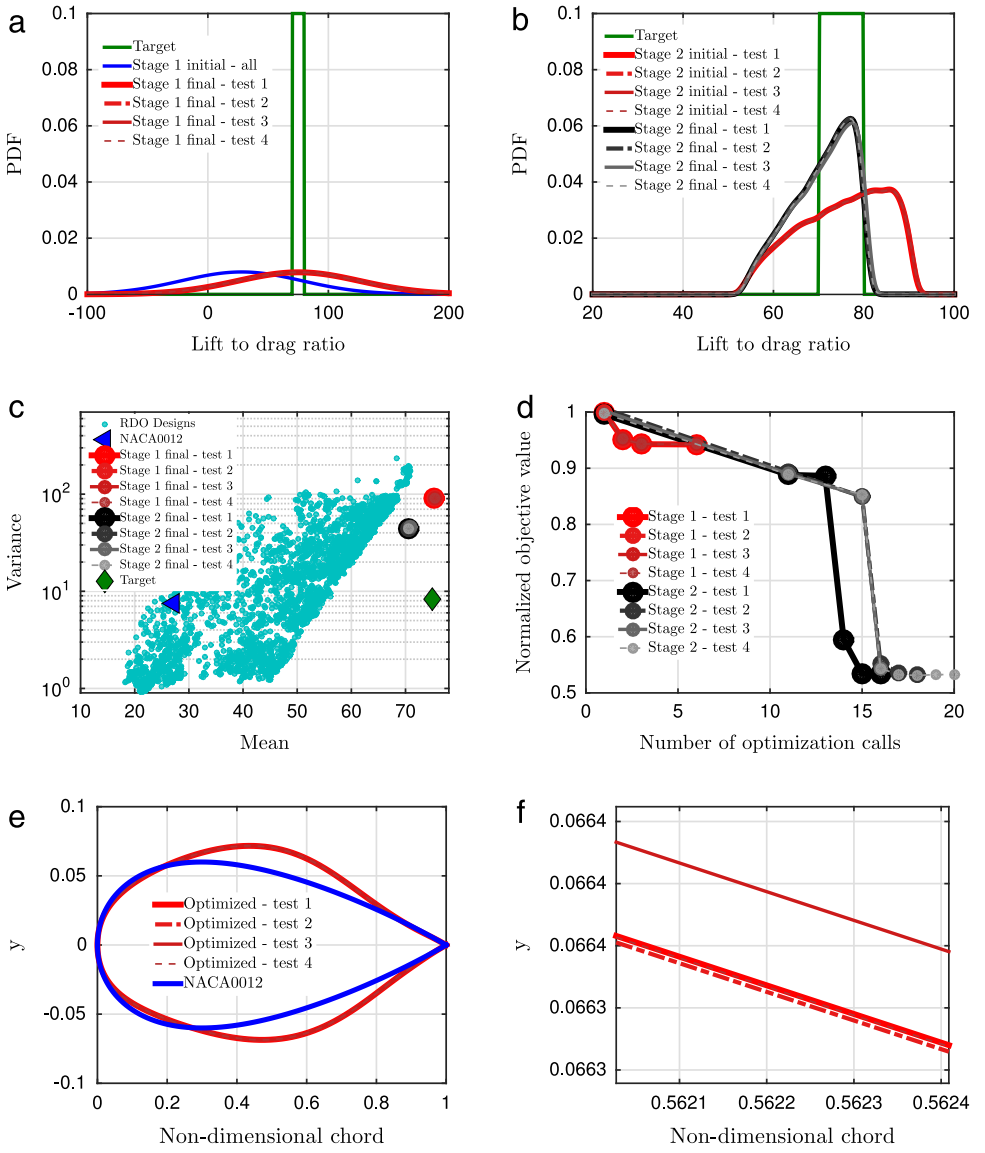

Fig. 5. Uniform target results: (a) stage 1 ($h = 50$), (b) stage 2 ($h = 1$), (c) comparison with RDO designs, (d) stage 1 and stage 2 convergence plots, (e) stage 2 optimal designs with a close-up in (f). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

designs are shown in Fig. 7(a), the final designs in Fig. 7(b), and the convergence history in Fig. 7(e). Stage 1 used 7 function calls with 3 major iterations, while stage 2 used 26 with 7 major iterations. For the beta target, stage 1 took 1 h and 24 min while stage 2 took 5 h and 8 min.

In the final result, we find that the optimizer tried to get as close as possible to the positively skewed target and produced a design whose density estimate has skewness −0.08. Neighboring solutions in the Pareto front – with similar means and variances – exhibited large negative skewness; see Fig. 7(c) and a closeup in Fig. 7(d). An RDO formulation that uses only the first two moments has no control over the design pdf's skewness. By trying to match the response pdf to a positively skewed target (i.e., a synthetic example where a designer favors positive skewness in the stochastic response), the density-matching approach attempts to satisfy the skewness criteria while simultaneous finding a nearly Pareto optimal mean and variance. The final design is shown in Fig. 7(f).

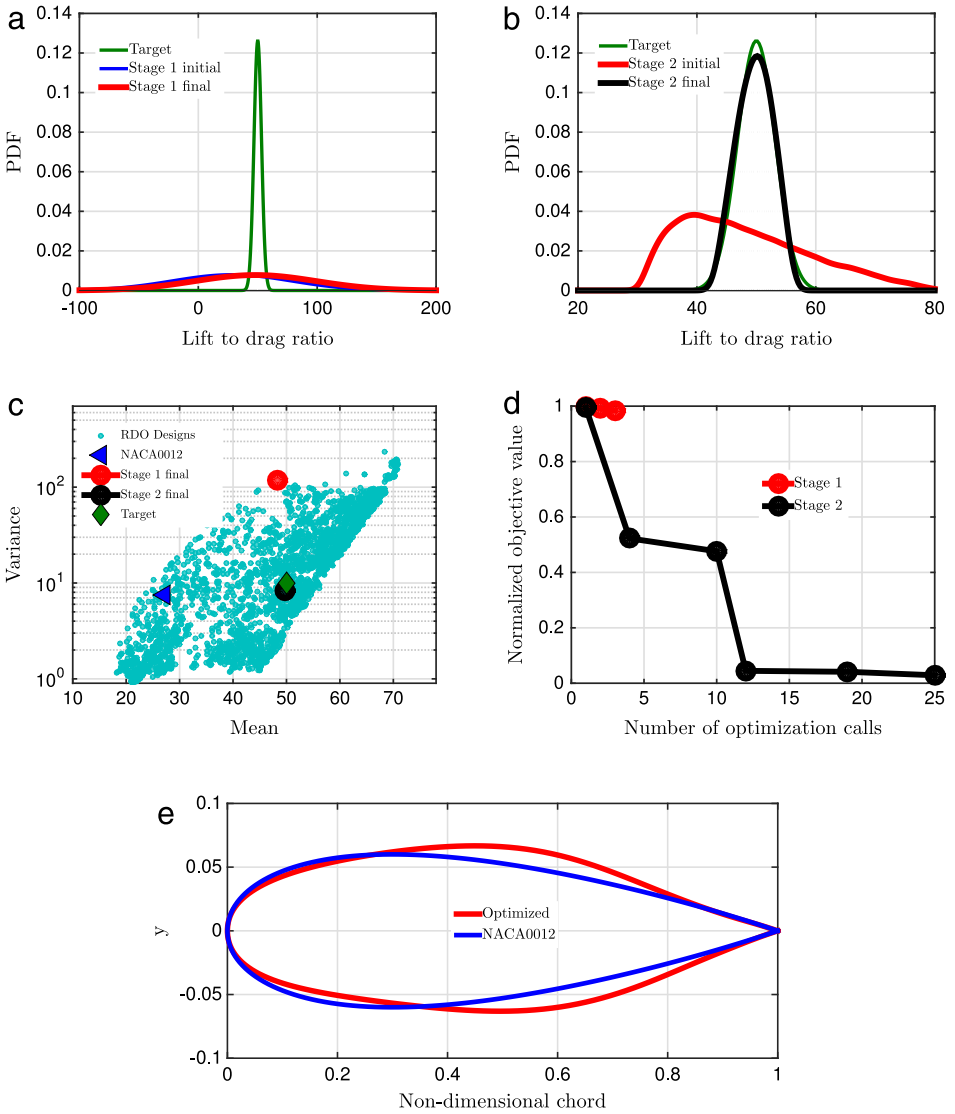

Fig. 6. Gaussian target results: (a) stage 1 ($h = 50$), (b) stage 2 ($h = 1$), (c) comparison with RDO designs, (d) stage 1 and stage 2 convergence plots, (e) stage 2 optimal design.

6. Conclusions and future directions

We present a metric for optimization under uncertainty formulations. We assume that a designer has provided a target pdf of the system response, and we minimize the distance between the design-dependent response pdf and the given target over possible designs. We study the differentiable L_2-norm between the response and target pdfs, though other distance metrics may be employed. One drawback of the L_2-norm is that if the target and response pdfs are not sufficiently large on the same support, then the objective function's gradient may not be useful for the OUU. We present a particular discretization of the L_2-norm objective function that uses a numerical integration rule and a kernel density estimate for the response pdf. The kernel density estimate with the Gaussian kernel has a simple form for the discretized objective's gradient. We offer two computational heuristics: (i) a two-stage strategy in the bandwidth choice for the kernel estimate that alleviates the support issue and (ii) a response surface approach for computationally expensive system responses. We apply this approach in two examples: (i) a simple function that produces a Gaussian response pdf whose variance is the design parameter and (ii) a CFD-based airfoil shape optimization.

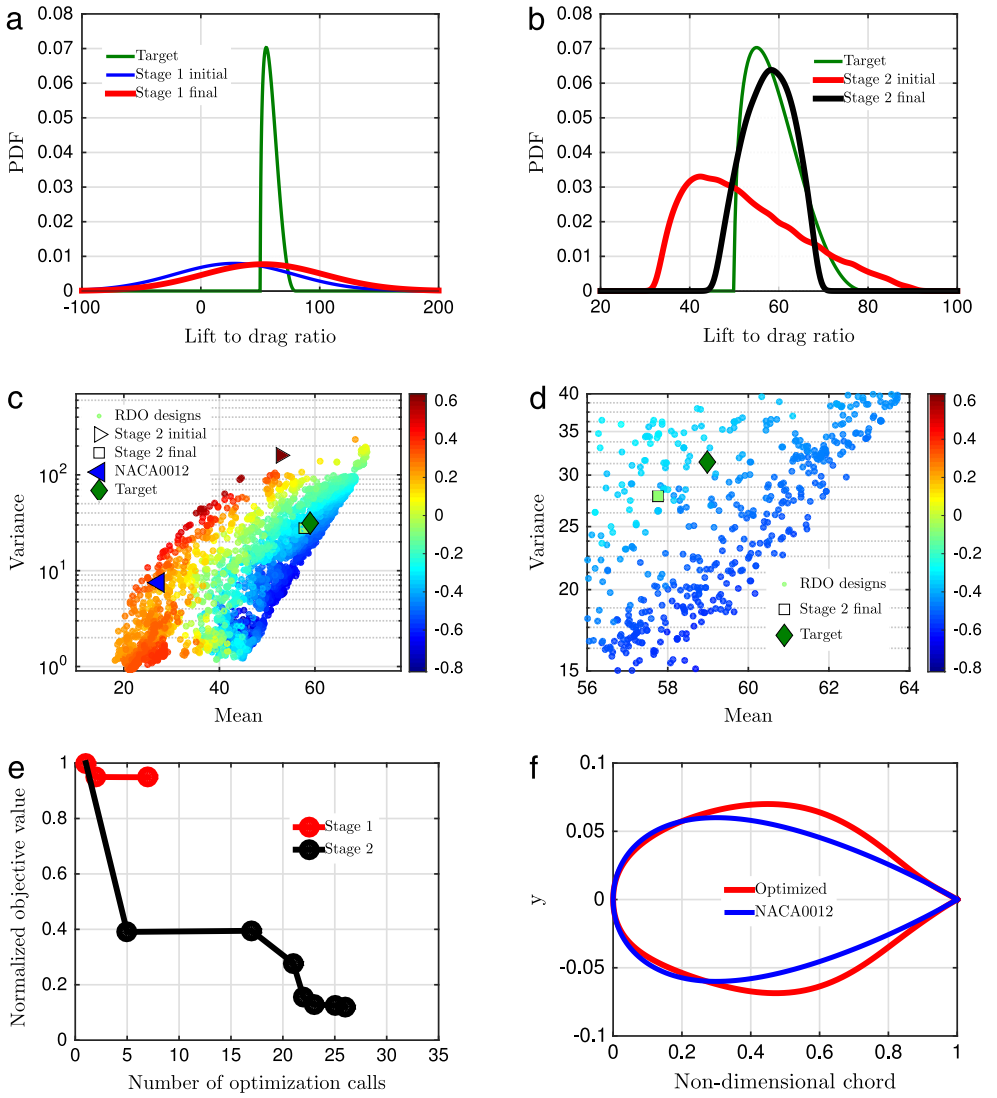

Fig. 7. Beta target results: (a) stage 1 ($h = 50$), (b) stage 2 ($h = 1$), (c) comparison with RDO designs with a close-up in (d), (e) stage 1 and stage 2 convergence plots, (f) stage 2 optimal design.

In principle, one could use the density-matching objective as one criterion in a hybrid OUU formulation if the designer (i) has specified some desired data as a pdf and (ii) includes other measures of reliability or robustness, e.g., failure probabilities or moments. In such a scenario, a numerical optimizer would need to balance the potentially competing objectives. Such ideas may be explored in future work.

Acknowledgments

This research was funded through a Dorothy Hodgkin Postgraduate Award, which is jointly sponsored by the Engineering and Physical Sciences Research Council (EPSRC) (UK) and Rolls-Royce plc. The first author would like to acknowledge the financial assistance provided by the Center for Turbulence Research at Stanford University and St. Edmund's College, Cambridge. The authors would like to thank Shahrokh Shahpar of Rolls-Royce plc for his advice on various aspects of this work. The authors also thank the reviewers for their suggestions and comments, which improved the overall quality of this manuscript. The second author's work is supported by the U.S. Department

of Energy Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under Award Number DE-SC-0011077.

References

[1] A. Abbas-Bayoumi, K. Becker, An industrial view on numerical simulation for aircraft aerodynamic design, J. Math. Ind. 1 (1) (2011) 1–14. URL http://dx.doi.org/10.1186/2190-5983-1-10.
[2] S. Shahpar, S. Caloni, L. dePrieelle, Automatic design optimisation of profiled endwalls including real geometrical effects to minimize turbine secondary flows, in: ASME Turbo Expo 2014: Turbine Technical Conference and Exposition Paper: GT2014-26628, Dusseldorf, Germany, June 16–20, 2014. URL http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1907592.
[3] S. Chandra, A. Lee, S. Gorrell, C. Jensen, CFD analysis of pace formula-1 car, Comput. Aided Design Appl. -PACE 1 (2011) 1–14.
[4] A. Papanikolaou, Holistic ship design optimization, Comput. Aided Design 42 (11) (2010) 1028–1044. URL http://www.sciencedirect.com/science/article/pii/S0010448509001973.
[5] M. Allen, K. Maute, Reliability-based design optimization of aeroelastic structures, Struct. Multidiscip. Optim. 27 (4) (2004) 228–242. URL http://dx.doi.org/10.1007/s00158-004-0384-1.
[6] J. Axerio-Cilles, Predicting formula 1 tire aerodynamics: Sensitivites, uncertainties and optimization (Ph.D. thesis), Stanford University, 2012.
[7] P. Seshadri, S. Shahpar, G. Parks, Robust compressor blades desensitizing operational tip clearance variations, in: ASME Turbo Expo 2014: Turbine Technical Conference and Exposition Paper: GT2014-26624, Dusseldorf, Germany, June 16–20, 2014. URL http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1907441.
[8] T. Ghisu, G. Parks, J. Jarrett, P. Clarkson, Robust design optimization of gas turbine compression systems, J. Propul. Power 27 (2) (2011) 282–295.
[9] H. Tachikawa, D. Schiavazzi, T. Arima, G. Iaccarino, Robust optimization for windmill airfoil design under variable wind conditions, in: Center for Turbulence Research, Proceedings of the Summer Program, Stanford, CA, 2012. URL https://web.stanford.edu/group/ctr/Summer/SP12/03.06_tachikawa.pdf.
[10] I. Doltsinis, Z. Kang, Robust design of structures using optimization methods, Comput. Methods Appl. Mech. Engrg. 193 (2004) 2221–2237.
[11] D. Frangopol, K. Maute, Life-cycle reliability-based optimization of civil and aerospace structures, Comput. Struct. 81 (7) (2003) 397–410. URL http://www.sciencedirect.com/science/article/pii/S0045794903000208.
[12] B. Bichon, M. Eldred, L. Swiler, S. Mahadevan, J. McFarland, Efficient global reliability analysis for nonlinear implicit functions, AIAA J. 46 (10) (2008) 2459–2468.
[13] Y. Lian, N. Kim, Reliability-based design optimization of a transonic compressor, AIAA J. 44 (2) (2006) 368–375.
[14] S. Missoum, C. Dribusch, P. Beran, Reliability-based design optimization of nonlinear aeroelasticity problems, AIAA J. Aircr. 47 (3) (2010) 992–998. URL http://arc.aiaa.org/doi/pdf/10.2514/1.46665.
[15] B. Youn, K. Choi, R.-J. Yang, L. Gu, Reliability-based design optimization for crashworthiness of vehicle side impact, Struct. Multidiscip. Optim. 26 (3–4) (2004) 272–283. URL http://dx.doi.org/10.1007/s00158-003-0345-0.
[16] S. Rangavajhala, S. Mahadevan, Joint probability formulation for multiobjective optimization under uncertainty, J. Mech. Des. 133 (5) (2011) 051007. URL http://mechanicaldesign.asmedigitalcollection.asme.org/data/Journals/JMDEDB/27946/051007_1.pdf.
[17] P. Seshadri, P. Constantine, G. Iaccarino, Aggressive design under uncertainty, Amer. Inst. Aeronaut. Astronaut. (2014) URL http://dx.doi.org/10.2514/6.2014-1007.
[18] A. Gibbs, F. Su, On choosing and bounding probability metrics, Internat. Statist. Rev. 70 (3) (2002) 419–435. URL http://dx.doi.org/10.1111/j.1751-5823.2002.tb00178.x.
[19] P. Davis, P. Rabinowitz, Methods of Numerical Integration, second ed., Dover, Mineola, 2007.
[20] L. Trefethen, Is gauss quadrature better than clenshawcurtis? SIAM Rev. 50 (1) (2008) 67–87. URL http://dx.doi.org/10.1137/060659831.
[21] L. Trefethen, J. Wideman, The exponentially convergent trapezoidal rule, SIAM Rev. 56 (3) (2014) 385–458. URL http://dx.doi.org/10.1137/130932132.
[22] D. Scott, Multivariate Density Estimation: Theory, Practice, and Visualization, second ed., John Wiley & Sons, Hoboken, 2014, URL http://www.worldcat.org/isbn/0471547700.
[23] L. Greengard, J. Strain, The fast gauss transform, SIAM J. Sci. Stat. Comput. 12 (1) (1991) 79–94. URL http://dx.doi.org/10.1137/0912004.
[24] J. Nocedal, S. Wright, Numerical Optimization, second ed., Springer, New York, 2006.
[25] S.J. Sheather, M.C. Jones, A reliable data-based bandwidth selection method for kernel density estimation, J. R. Stat. Soc. Ser. B Stat. Methodol. 53 (3) (1991) 683–690. URL http://www.jstor.org/stable/2345597.
[26] A.W. Bowman, A. Azzalini, Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations (Oxford Statistical Science Series), Oxford University Press, USA, 1997, URL http://www.worldcat.org/isbn/0198523963.
[27] R. Smith, Uncertainty Quantification: Theory, Implementation, and Applications, SIAM, Philadelphia, 2013.
[28] D. Xiu, Numerical Methods for Stochastic Computations: A Spectral Method Approach, Princeton University Press, Princeton, 2010.
[29] O. Le Maître, O. Knio, Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics, Springer, New York, 2010.
[30] H. Wendland, Scattered Data Approximation, Cambridge University Press, Cambridge, 2005.
[31] R. Durrett, Probability: Theory and Examples, Brooks/Cole—Thomson Learning, Belmont, 2005.
[32] Aerospace Design Lab, Stanford University SU2 unstructured solver, 2011 http://su2.stanford.edu/.
[33] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.