Global Stochastic Optimization of Stellarator Coil Configurations

Silke Glas¹, Misha Padidar², Ariel Kellison¹ and David Bindel¹

¹Department of Computer Science, Cornell University, Ithaca, NY 14853, USA
²Center for Applied Mathematics, Cornell University, Ithaca, NY 14853, USA.

In the construction of a stellarator, the manufacturing and assembling of the coil system is a dominant cost. These coils need to satisfy strict engineering tolerances, and if those are not met the project could be canceled as in the case of the National Compact Stellarator Experiment (NCSX) project [25]. Therefore, our goal is to find coil configurations that increase construction tolerances without compromising the performance of the magnetic field.

In this paper, we develop a gradient-based stochastic optimization model which seeks robust stellarator coil configurations in high dimensions. In particular, we design a two-step method: first, we perform an approximate global search by a sample efficient trust-region Bayesian optimization; second, we refine the minima found in step one with a stochastic local optimizer. To this end, we introduce two stochastic local optimizers: BFGS applied to the Sample Average Approximation and Adam, equipped with a control variate for variance reduction. Numerical experiments performed on a W7-X-like coil configuration demonstrate that our global optimization approach finds a variety of promising local solutions at less than 0.1% of the cost of previous work, which considered solely local stochastic optimization.

1 Introduction

The design process of finding promising stellarator coil configurations is traditionally split into two steps. First, one aims to find the optimal plasma shape with respect to performance criteria such as, e.g., the magnetohydrodynamic (MHD) stability or alpha particle confinement. For the second step - the coil design - one tries to reproduce the target magnetic field confining the plasma. Construction and placement of these coils are difficult tasks since minor errors in the fabrication or alignment can lead to major modifications of the magnetic field and thus poor particle confinement [2] [22].

To be able to guarantee good confinement when the stellarator is in operation, stringent engineering tolerances on the coils are required during the stellarator fabrication and assembly process. Unfortunately, such stringent tolerances can increase the cost and production timeline of a stellarator, which led to cancellation of the National Compact Stellarator Experiment (NCSX) project
During the Columbia Nonneutral Torus (CNT) design, a stochastic perturbation analysis was used to identify the robustness of coil-induced magnetic fields to coil alignment errors [17]. This allowed the reduction of engineering coil tolerances significantly [16]. Furthermore, recent work has shown that stochastic optimization is a promising method to improve the robustness of the generated magnetic field to fabrication and alignment errors in the associated coils [21, 34]. However, all aforementioned approaches do not globally explore the coil design space. Indeed, design engineers are usually interested in having not only one, but multiple designs to choose from. In addition to the magnetic field, engineers care about other physical properties such as specific aspect ratio and rotational transform; see Figure 1 for an example of different coils. Thus, there is a need for efficient stochastic methods for global exploration in order to provide multiple coil configurations in the design process.

In this work, we develop an algorithm for the efficient global stochastic optimization of stellarator coil configurations. We seek a coil set that is robust to random errors in the coils in expectation:

\[
\min_{x \in \mathbb{R}^N} \mathbb{E}[f_{\text{stoc}}(x + U)] + f_{\text{reg}}(x)
\]

\[
c_i(x) \geq \epsilon_i \quad i = 1, \ldots, N_C - 1
\]

where \(x \in \mathbb{R}^N\) defines the geometry of the coil set, \(U\) is a random error to the coils, \(\mathbb{E}\) is the expectation over \(U\), \(N_C\) is the number of coils in a single field period, \(\epsilon \in \mathbb{R}\) is a constraint value, and the functions \(f_{\text{stoc}}, f_{\text{reg}}, c_i : \mathbb{R}^N \to \mathbb{R}\) measure the quality of the coils and induced magnetic field. We define the components of the optimization model precisely in Section 3. Similarly to [21], we consider fabrication errors to be spatially correlated Gaussian perturbations to the coils. In order to optimize (1), we present a two-step global-to-local algorithm for stochastic optimization. In the first step, we perform a global exploration of the stochastic optimization model given in (1) using a trust-region Bayesian optimization method based off [6]. This global stage finds multiple approximate minima which, in the second step, are resolved by local stochastic optimizers. To perform local stochastic optimization we apply the BFGS optimizer [24] to the Sample Average Approximation (SAA) [28] of (1) as well as the Adam optimizer [13] enhanced by a control variate for variance reduction. Finally, we arrive at multiple local stochastic minima from which one can choose the most promising configuration.
From a stellarator optimization point of view, the main new ingredients of our optimization routine are (to the best of our knowledge):

1. Method for efficient stochastic global exploration of the design space.

2. Local stochastic optimization with Adam enhanced with novel control variate.

Moreover, this is the first work to perform stochastic optimization using the coil optimization code FOCUS [36].

This paper is organized as follows. We briefly describe FOCUS and the internal representation of the coils in Section 2. In Section 3 we introduce the formulation of the stochastic optimization model. Subsequently in Section 4 we describe our global-to-local stochastic optimization algorithm. Numerical experiments on a W7-X configuration are presented in Section 5. We conclude with a summary and ideas for future work in Section 6.

2 FOCUS

For the second design stage of a stellarator, multiple codes are available to optimize for coil configurations which replicate a target magnetic field. Well-known examples are NESCOIL [23], REGCOIL [18], ONSET [5], COILOPT [30], COILOPT++ [4] and FOCUS [36]. In order to perform stochastic optimization, we decided to use FOCUS for two reasons: first, it allows coils to move freely in space whereas the other aforementioned coil optimization codes restrict to a so-called winding surface; and second, FOCUS provides analytic first-order derivatives which improve efficiency in the optimization process.

In FOCUS, the coils are described by a 3-dimensional Fourier representation. FOCUS takes as input a vector of Fourier coefficients \( \mathbf{x} = (x^1, \ldots, x^{N_C}) \in \mathbb{R}^N \) describing the geometry of \( N_C \) filamentous coils. From the Fourier coefficients \( x^i = (x_{c,0}^i, \ldots, x_{s,N_F}^i, y_{c,0}^i, \ldots, y_{s,N_F}^i, z_{c,0}^i, \ldots, z_{s,N_F}^i)^T \) for the \( i \)-th coil, FOCUS constructs coils as a parametrized curve with \( x \)-coordinate

\[
x^i(x, t) = x_{c,0}^i + \sum_{n=1}^{N_F} \left[ x_{c,n}^i \cos(nt) + x_{s,n}^i \sin(nt) \right], \quad t \in [0, 2\pi),
\]

with \( N_F \) being the number of Fourier modes and analogous forms for the \( y-, z \)-coordinates \( y^i(x, t), z^i(x, t) \). By \( \mathbf{X}^i(x, t) = (x^i(x, t), y^i(x, t), z^i(x, t))^T \) we denote the parametric representation of the \( i \)-th coil filament with \( t \in [0, 2\pi) \). The number of parameters per coil is given by \( 3(2N_F + 1) \) such that the total number of parameters results in \( N := 3N_C(2N_F + 1) \). Note, that FOCUS only requires the description of a single field period of coils, constructing the other coils through reflection and rotation around the stellarator.

With the Fourier representation FOCUS can, among other things, efficiently compute coil metrics related to coil curvature, coil length, normal magnetic field, quasisymmetry, magnetic island width and their respective gradients with respect to the Fourier coefficients [35]. In this study, we only make use of the normal component of the magnetic field \( f_B \) and length metric \( f_L \), which we describe in the following.

\( ^1 \)Note that the coil currents are held constant in this work.
The metric $f_B$ addresses deviations in the magnetic field produced by a coil set $x$ compared to a target magnetic field $B$. Given a target plasma boundary $S$ and a target magnetic field $B$ the normal field error $f_B$ is given by

$$f_B(x) = \int_S \frac{1}{2} (B \cdot n)^2 ds,$$

where $n$ is the unit normal on the target plasma boundary. Typically the target plasma boundary is generated from a MHD equilibrium solver such as e.g., VMEC [9], SPEC [10].

The metric $f_L$ is introduced to influence the length of the coils such that they are not prohibitively long. For the $i$-th coil with length $L_i(x) \in \mathbb{R}$ and user-specified target length $L_i^{\text{target}}$, FOCUS computes the length metric

$$f_L(x) = \frac{1}{N_C} \sum_{i=1}^{N_C} \frac{1}{2} \frac{(L_i(x) - L_i^{\text{target}})^2}{(L_i^{\text{target}})^2}.$$ 

In the next section, we detail how we design our stochastic optimization model using the introduced FOCUS metrics.

3 Optimization Model Formulation

In this section we describe our stochastic optimization model for finding a set of coils which generate a target magnetic field and simultaneously hedge against errors in the coil fabrication. The decision variables $x \in \mathbb{R}^N$ for our problem are the $N = 3N_C(2N_F + 1)$ Fourier coefficients defining the geometry of the $N_C$ coils, see Section 2. The stochasticity is motivated by fabrication errors in the coils and modeled as spatially correlated Gaussian perturbations as described in Section 3.1. In Section 3.2 we detail the stochastic and non-stochastic parts of the objective function. We design a coil-to-coil separation constraint and formulate the final optimization problem used in the numerical experiments in Section 3.3.

3.1 Coil Fabrication Uncertainty

Errors during coil fabrication can alter the shape of the coil and in turn the relative positioning of the coil to the plasma. Mathematically, we model coil fabrication errors as spatially correlated Gaussian perturbations. Numerically, the perturbations are independent of the coil discretization by considering additive perturbations modeled by Gaussian Processes (GPs). Similar approaches to model coil perturbations for stellarators have been presented in [21, 34]. Compared to [34], instead of a squared exponential kernel with added periodicity, we take the period kernel as the basis for our perturbations. For the ease of notation, we omit the dependency on $i$ as the $i$-th coil in this subsection as without loss of generality we consider the derivation for one coil.

Let $X(x, t) = (x(x, t), y(x, t), z(x, t))^T$ be a parametric representation of a coil filament with $t \in [0, 2\pi)$. We consider the distribution of the fabrications errors to be smooth along the entirety of the coil. As such we model them by zero-mean Gaussian random variables that effect every point on the coil. Equivalently, the coil filament $X(x, t)$ is perturbed by a zero-mean Gaussian process (GP)

$$(g_x(t), g_y(t), g_z(t))^T =: g(t) \sim \mathcal{GP}(0, \kappa_{\text{per}}(t, t')),$$ 

$t, t' \in [0, 2\pi)$,
with a $2\pi$-periodic kernel $\kappa_{\text{per}}$. We assume that $g(t)$ is an isotropic GP, i.e., the kernel $\kappa_{\text{per}}$ is only a function of distance of $t, t'$, and that $g_x(t), g_y(t), g_z(t)$ are independent of each other. Then the perturbed coil filament $X_P(x, t)$ is again a GP with mean $X(x, t)$

$$X_P(x, t) := X(x, t) + g(t) \sim \mathcal{GP}(X(x, t), \kappa_{\text{per}}(t, t')),$$  \hspace{1cm} t, t' \in [0, 2\pi). \hspace{1cm} (4)$$

To compute relevant properties of the perturbed coil $X_P(x, t)$ in FOCUS, we need a Fourier representation $x_P(t)$ of $X_P(x, t)$. As we already know the Fourier representation of $X(x, t)$, we only need to calculate the Fourier representation of $g(t)$. The $k$-th cosine and sine Fourier coefficients of the $x$-coordinate of the GP $g_x(t)$, are denoted $\hat{x}_{ck}, \hat{x}_{sk}$, and can be calculated with

$$\hat{x}_{ck} = \frac{1}{\pi} \int_0^{2\pi} g_x(t) \cos(kt) dt, \quad \hat{x}_{sk} = \frac{1}{\pi} \int_0^{2\pi} g_x(t) \sin(kt) dt,$$

with analogous forms for the $y$-, $z$-coordinates using $g_y(t), g_z(t)$. We collect these Fourier coefficients into the vector $U$. As $g(t)$ is a GP and integration is a linear operation, it follows that $U$ is a normally distributed random vector, see [27]. The mean of $U$ is zero, i.e. $\mathbb{E}[U] = 0$ and the covariance can be written in terms of the Fourier coefficients of the GP kernel $\kappa_{\text{per}}$, [27], i.e.,

$$\text{Cov}[\hat{x}_{ck}, \hat{x}_{sj}] = \frac{1}{\pi^2} \int_0^{2\pi} \int_0^{2\pi} \kappa_{\text{per}}(s, t) \cos(kt) \sin(js) ds \, dt. \hspace{1cm} (5)$$

From the assumption that the $g_x(t), g_y(t), g_z(t)$ are independent of each other, it follows that their Fourier coefficients are independent as well. Thus, the covariance matrix $C$ of the perturbation Fourier coefficients $U$ is block diagonal, where each diagonal block has entries from (5). Due to (4), a randomly perturbed coil can then be described in FOCUS' Fourier representation by $x_P = x + U$, where $U \sim \mathcal{N}(0, C)$. 

---

Figure 2: Random perturbations of a circular coil generated from a GP with lengthscale $\ell = 0.1$ (left), $\ell = 0.5$ (center) and $\ell = 1.0$ (right). Note that perturbations generated from a GP with a larger lengthscale have lower frequency oscillations.
3.1.1 The Kernel and Hyperparameters

To ensure the GP perturbations are periodic, we use the 2π-periodic kernel

$$\kappa_{\text{per}}(t, t') = h \exp \left( -\frac{2 \sin^2 \left( \frac{|t-t'|}{\ell^2} \right)}{\ell^2} \right), \quad t, t' \in [0, 2\pi),$$

where $h > 0$ denotes the size of the perturbations and $\ell > 0$ is the lengthscale of the GP. We set the hyperparameters $h, \ell$ such that at any point the mean norm squared of the multi-output GP equals the desired mean perturbation size $p$ squared, i.e., $\mathbb{E}[\|g(t)\|_2^2] = p^2$. As $g_x(t), g_y(t), g_z(t)$ all have identical moments and a first moment of zero we simplify the equality to $3\text{Var}[g_x(t)] = p^2$ which yields $h = p^2/3$.

The lengthscale $\ell$ determines the frequency of the perturbations, as depicted in Figure 2. While we did not have any specific value given for $\ell$, we found $\ell = 0.5$ to model realistic manufacturing errors such that we chose this value for our numerical experiments.

3.2 Objectives

The primary goal of the second stage of stellarator optimization is to find a set of coils which reproduce a target magnetic field while satisfying engineering targets, such as reasonable coil length and separation between coils. To this end, we chose our objective function $f$ to penalize the normal component of the magnetic field produced by the coils $f_B$ and regularize the coil length $f_L$ to discourage excessively long coils. We find that adding a simple penalty $f_L$ to the objective, rather than explicitly constraining the length, is sufficient to regularize coil length. While additional coil regularization functions such as curvature are available in FOCUS, using a length regularization proved to be sufficient.

Not only do we seek a set of coils which generates the target magnetic field $B$ - we simultaneously want to hedge against errors in the coil fabrication. To this end, we formulate the following stochastic objective

$$f(x) = \mathbb{E}[\omega_B f_B(x + U)] + \omega_L f_L(x), \quad x \in \mathbb{R}^N,$$

where $\omega_B, \omega_L \in \mathbb{R}$ are weights of the respective objective, $U \sim \mathcal{N}(0, C)$ is a multivariate normal random variable described in Section 3.1 and $\mathbb{E}$ denotes the expectation over $U$. Referring to (1), we set the stochastic and the regularizing part of the objective to

$$f_{\text{stoc}}(x) := \omega_B f_B(x), \quad f_{\text{reg}}(x) := \omega_L f_L(x).$$

Note that we only penalize perturbations in the magnetic field penalty and not perturbations in the length penalty as fabrication errors that minorly increase coil length are not necessarily problematic from an engineering perspective.

3.3 Coil-to-Coil Separation Constraint

Engineering requirements state that a reasonable configuration must have sufficient spacing between all pairs of coils, see e.g., [3]. In order to ensure that this condition is satisfied we include
it as a constraint in our optimization model. The implication is that the coil-to-coil distance for adjacent coils is now bounded from below, but any larger distance is allowed and will not add unnecessary penalties to the optimization model. We find that without enforcing this constraint coils come too close during numerical experiments. We model the minimum distance between coils $i$ and $i+1$ by

$$c_i(x) = \min_{s,t \in [0,2\pi]^2} \|X^i(x,s) - X^{i+1}(x,t)\|_2^2.$$  

(7)

Thus, for our optimization model in (1), we enforce the constraint

$$\min_{s,t \in [0,2\pi]^2} \|X^i(x,s) - X^{i+1}(x,t)\|_2^2 \geq \epsilon^2_{ci} \quad i \in \{1, ..., N_C - 1\}.$$  

(8)

To handle the constraints (8) we discretize our coils into $N_{seg}$ segments and compute the minimum across a total of $N_{seg}^2$ discrete points for any adjacent coil pairs. Note, that it is important to also include the width of the coil into $\epsilon_c$ as FOCUS models coils as infinitely thin filaments.

In order to continue to use derivative-based optimization techniques we calculate the minimum distance and the derivative between a pair of coils with a smooth approximation to the minimum function, the so-called $\alpha$-quasimax function, see [20] (also known as LogSumExp function). The $\alpha$-quasimax function approximates the minimum

$$\min(x_1, ..., x_n), \quad x_i \in \mathbb{R}, i = 1, ..., n$$

by

$$Q_\alpha(x_1, ..., x_n) := \frac{1}{-\alpha} \log \left( \sum_{i=1}^{n} \exp \left( -\alpha x_i \right) \right),$$  

(9)

with $\alpha > 0$. Therefore, in the limit $Q_\alpha(x_1, ..., x_n) \rightarrow \min(x_1, ..., x_n)$ for $\alpha \rightarrow \infty$ and the following bound holds

$$\min(x_1, ..., x_n) - \frac{\log(n)}{\alpha} \leq Q_\alpha(x_1, ..., x_n) < \min(x_1, ..., x_n).$$

Thus, we arrive at the final optimization model considered in this paper: we seek $x \in \mathbb{R}^N$ such that

$$\min_{x \in \mathbb{R}^N} \mathbb{E}[\omega_B f_B(x + U)] + \omega_L f_L(x)$$

$$\min_{s,t \in [0,2\pi]^2} \|X^i(x,s) - X^{i+1}(x,t)\|_2^2 \geq \epsilon^2_{ci}, \quad i = \{1, ..., N_c - 1\}.$$  

(10)

In the next section, we present our efficient global optimization algorithm for the stochastic optimization of (10).

## 4 Two-stage Global Stochastic Optimization

In this section, we introduce our global-to-local stochastic optimization algorithm for solving (10). To this end, we modify two methods, the global trust-region Bayesian optimization TuRBO [6] method and the Adam [13] algorithm. The Bayesian derivative-free optimization routine TuRBO is designed to efficiently globally optimize a nonlinear function in a high-dimensional bound
constrained space. As FOCUS provides derivative information we use DTuRBO [26], which is a further development of TuRBO incorporating derivative information. To prioritize efficient exploration DTuRBO does not resolve minima to high orders of accuracy. Therefore, we apply a stochastic local optimization starting from the points we get from the final stage of DTuRBO. Thus our approach can be summarized in two stages:

1. perform an approximate global optimization with DTuRBO and select a set of promising solution points,

2. resolve these solution points locally utilizing a stochastic optimizer.

In order to provide choices for prospective future users of this two-stage approach, we introduce two options for a local stochastic optimizer, the Sample Average Approximation (SAA) method and the Adam algorithm enhanced with a novel control variate (AdamCV) for variance reduction. This two-stage approach makes efficient use of the computational budget as the local optimizer will resolve the minima much more efficiently than DTuRBO. We start by describing the global optimization.

### 4.1 Stage 1: Efficient Global Exploration of Design Space

In this section, we detail the global optimization by first describing TuRBO and subsequently commenting on the modifications made to arrive at DTuRBO. The TuRBO algorithm is a derivative-free method for global optimization of a nonlinear function across a high-dimensional hypercube $\Omega = [lb, ub]^N$ with $lb, ub \in \mathbb{R}^N$ being a vector of lower and upper bounds for the design space variables. The TuRBO algorithm starts by performing a Latin hypercube sampling on $\Omega$ receiving a sample $X_{\text{init}} = [x_1, \ldots, x_{n_{\text{sample}}}]$ of size $n_{\text{sample}} \in \mathbb{N}$ and then evaluates the to be optimized function $f$ for each $x \in X_{\text{init}}$. Subsequently, $M$ local Bayesian optimization (BO) runs are started at the $M$ best points from $X_{\text{init}}$ by building local GPs within $M$ distinct rectangular trust-regions. At each iteration Thompson sampling, see [31], is performed within each trust-region to generate a set of candidate points. A batch of the most promising of these candidate points is evaluated by the function $f$ and these function values are used to update the GP surrogates. Note, that the batch of evaluations is chosen from the union of candidate points across the trust-regions. Therefore, the evaluations are distributed to each trust-region by the region’s predicted success through Thompson Sampling. The trust-regions are centered around the best point found in the evaluation history, and are expanded or contracted by a factor of 2 depending on consecutive successes/failures of decreasing the function value. A local BO will terminate after the trust-region reaches a minimum size. This indicates that the local BO is no longer making improvement and is near an optima. Precisely these approximate minima are gathered and passed on to the second stage of our approach for further refinement with the local optimizer. Then, a new Latin hypercube sampling of $\Omega$ is performed and a local BO is again started at the best point. The algorithm terminates when the computational budget is reached.

The success of TuRBO in high dimensions is found in its ability to leverage multiple local surrogates and simultaneously efficiently distribute the computational budget across the local BO runs. While the common BO can only be used up to approximately 20 dimensions, see e.g. [7], TuRBO
is designed for significantly higher-dimensional problems. DTuRBO further improves on TuRBO by equipping it with a scalable method of incorporating derivatives into the GP models, which yields several advantages. For example, gradients encode the local descent direction ensuring that the surrogate is locally accurate and a decreasing direction can be found more easily. Internally DTuRBO uses a stochastic variational Gaussian process [8, 11] to scalably incorporate the high-dimensional gradient data by approximating it with a low dimensional representation, see [26]. For practical use, DTuRBO takes in a pair of noisy function and gradient evaluations rather than only noisy function evaluations of $f$. The approximate minima gathered from DTuRBO are passed as starting points for the local stochastic optimizers to be resolved further.

4.2 Stage 2: Local Stochastic Optimization

Approximate minima found in stage one by the DTuRBO algorithm can further be refined by local stochastic optimizers. The benefit of using local stochastic optimization techniques is that they will converge to the minima faster than global optimizers as their main focus is on exploitation rather than exploration. Additionally, some of those methods provide convergence guarantees. Common choices for first-order techniques are variants of the stochastic gradient method (SGD) [29] or the application of non-stochastic optimizers within the SAA method, which has also been used in [21, 34]. The SAA approximation determines a fixed-accuracy approximation to the true stochastic objective, where the error in the approximation decays with the square root of the number of samples. However, this approximation can be optimized in a relatively small number of steps of an algorithm like BFGS. In contrast, stochastic gradient descent methods, including our AdamCV approach, converge to the optimum of the true stochastic objective using a large number of relatively inexpensive steps. Which method is most appropriate depends strongly on how accurately the stochastic optimization problem should be solved.

In the following, we give an introduction to the Adam algorithm used in this work for the stochastic local optimization. Subsequently, we enhance Adam with a novel control variate, which is a variance reduction technique to improve convergence. Additionally, we give a brief overview over the SAA method.

4.2.1 Adam Algorithm

In order to efficiently arrive at a well-refined solution, we use a variant of the Adam algorithm [13]. Adam is a popular stochastic optimization method in machine learning due to its improved performance over traditional stochastic gradient methods. Adam’s success is largely due to the inclusion of a raw second moment estimator $v_k$ of the stochastic gradient. At each iteration Adam evaluates a small number $N_A$ (also called batchsize) of gradients and averages to form a gradient estimator

$$g^k = \frac{1}{N_A} \sum_{i=1}^{N_A} \nabla f(x_k + u_{k,i}), \quad k = 1, \ldots, N_{\text{max}}, \quad (11)$$

where $N_{\text{max}}$ is the maximum number of iterations and $u_{k,i}$ are $N_A$ realizations of the random variable $U$. To stabilize gradient estimates, Adam employs exponential moving averages of the
first and second raw moments of the stochastic gradient $m_k, v_k \in \mathbb{R}^N$ in each step

$$
\begin{align*}
m_k &= \beta_1 m_k + (1 - \beta_1) g_k, \quad k = 1, \ldots, N_{\text{max}}, \\
v_k &= \beta_2 v_k + (1 - \beta_2) g_k^2, \quad k = 1, \ldots, N_{\text{max}},
\end{align*}
$$

where the parameters $\beta_1, \beta_2 \in [0, 1]$ are usually set close to 1. The first and second moments estimators $m_k, v_k$ are biased as they are initialized as the vector of all zeros. To correct the bias, the estimators $m_k, v_k$ are divided by $(1 - \beta_1^k)$ and $(1 - \beta_2^k)$ to create the bias corrected estimators $\hat{m}_k, \hat{v}_k \in \mathbb{R}^N$

$$
\begin{align*}
\hat{m}_k &= \frac{m_k}{(1 - \beta_1^k)} \quad \hat{v}_k = \frac{v_k}{(1 - \beta_2^k)} \quad k = 1, \ldots, N_{\text{max}}.
\end{align*}
$$

Utilizing these estimators Adam converges to a local minima of (10) with the following step sequence

$$
x_{k+1} = x_k - \eta_k \hat{m}_k / (\sqrt{\hat{v}_k} + \epsilon_A),
$$

where $\epsilon_A \in \mathbb{R}^+$ is a small parameter to improve conditioning. To improve the convergence rate we use the decreasing step size sequence $\eta_k = \eta / (1 + \sqrt{k\gamma})$ where $\eta, \gamma$ are tunable parameters. It is essential that the learning rates for Adam are well-tuned for good convergence.

### 4.2.2 Adam with Control Variates (AdamCV)

When the variance of the perturbations $U$ become large it is beneficial to use variance reduction techniques to reduce the variance of the gradient estimate $g_k$. We follow [33] in developing such a variance estimator for $g_k$ with control variates. By using a Taylor expansion of the FOCUS objective function we can derive an approximate stochastic gradient to $f$ with moments that are easy to calculate. This approximate gradient can be combined with the true gradient to create an unbiased estimator for $g_k$ with a reduced variance. The first-order Taylor expansion of $\nabla f$ is $\nabla f(z) \approx \nabla f(x) + H(z - x)$ where $H$ denotes the Hessian matrix of $f$. Evaluating the Taylor expansion at $z = x + U$ yields the control variate $\tilde{g}(x)$

$$
\tilde{g}(x) = \nabla f(x) + HU,
$$

which is an estimator for $\mathbb{E}[\tilde{g}(x)] = \nabla f(x)$. Combining the estimator $\tilde{g}$ [12] with our original gradient estimator $g_k$ [11] leads us to an unbiased gradient estimator with lower variance

$$
g(x) = \nabla f(x + U) + A(\tilde{g}(x) - \mathbb{E}[\tilde{g}(x)]),
$$

where the choice of the diagonal matrix $A \in \mathbb{R}^{N\times N}$ is detailed in the following. This unbiased gradient estimator with lower variance can be applied to $g_k$ [11] as a replacement for $\nabla f(x + U)$ to accelerate the convergence of the Adam routine. By following this approach, we ensure that the control variate is highly correlated with $\nabla f(x + U)$, such that we obtain the variance reduction with respect to the gradient estimates.

As shown in [33] the optimal diagonal matrix $A \in \mathbb{R}^{N\times N}$ is chosen in order to minimize the trace of the variance of $g(x)$, i.e.,

$$
A = \frac{\text{diag} (\text{Cov}(HU, \nabla f(x + U)) + \text{Cov}(\nabla f(x + U), HU))}{2 \text{ diag } (HUU^TH^T)},
$$
where we use $\text{diag}$ to indicate the diagonal entries of a matrix. By using the optimal $A$ in (13) we arrive at a reduced variance for $\tilde{g}(x_k)$ given by

$$\text{Var}[\tilde{g}(x_k)] = (1 - \rho^2)\text{Var}[g_k],$$

where $\rho \in \mathbb{R}^N$ has entries

$$\rho_i = \text{Corr}(\partial_x f(x + U), \tilde{g}(x_k)_i), \quad i = 1, \ldots, N.$$ As the analytic Hessian $H$ of $f$ is not available in FOCUS we use an approximate Hessian $H_k$ in optimization step $k$ according to the BFGS Hessian approximation

$$y_{k-1} := \nabla f(x_k) - \nabla f(x_{k-1}),$$

$$s_{k-1} := x_k - x_{k-1},$$

$$H_k = H_{k-1} + \frac{y_{k-1}y_{k-1}^T}{y_{k-1}^T H_{k-1} s_{k-1}} - \frac{H_{k-1} s_{k-1}(H_{k-1} s_{k-1})^T}{s_{k-1}^T H_{k-1} s_{k-1}},$$

with $H_0$ initialized as the $N \times N$ identity matrix. We find that equipping Adam with the control variate approach leads to rapid convergence rates in practice, particularly when warm-starting our optimization from a solution from stage one. We call the combination of the Adam algorithm enhanced with the control variate $\text{AdamCV}$.

### 4.2.3 Sample Average Approximation

The Sample Average Approximation (SAA) method, see [28, 12, 14], is a method of forming a non-stochastic approximation to the stochastic problem (10) by using the Monte-Carlo method. For instance, in order to approximate the stochastic component $\mathbb{E}[f_{\text{stoc}}(x + U)]$, we pick $\{u_i\}_{i=1}^{N_{\text{SAA}}}$, which are $N_{\text{SAA}} \in \mathbb{N}$ independent realizations of the random variable $U$. These realizations $\{u_i\}_{i=1}^{N_{\text{SAA}}}$ are taken to form the approximation

$$f_{\text{SAA}}(x) = \frac{1}{N_{\text{SAA}}} \sum_{i=1}^{N_{\text{SAA}}} f_{\text{stoc}}(x + u_i),$$

with analogous form for the gradient approximation. This objective is straightforward to implement and can be minimized efficiently with standard non-stochastic optimizers such as BFGS [24] since the samples $u_i$ are kept fixed. Solving the SAA comes with large sample size guarantees. In the limit as the sample size $N_{\text{SAA}}$ approaches infinity, the order of convergence is $O(1/\sqrt{N_{\text{SAA}}})$ as for standard Monte-Carlo methods. For a finite batchsize $N_{\text{SAA}}$, the minima of the SAA may not converge to minima of the stochastic problem. So it is recommended that any minima to the SAA is re-evaluated under the stochastic objective to estimate the “out-of-sample” performance.

### 5 Stellarator Experiments

In this section, we present numerical results for a W7-X configuration. In the following, we detail our model data and perform the global-to-local stochastic optimization described in Section 4 to find multiple stochastic minima. To better differentiate between the global and the local
| Parameter | $\omega_B$ | $\omega_L$ | $\epsilon_c$ | $L_{i\text{target}}$ | $\ell$ |
|-----------|------------|------------|------------|----------------|--------|
| Value     | 100.0      | 0.5        | 0.23       | 8.0             | 0.5    |

Table 1: Optimization parameter used in numerical experiments: weights in FOCUS objective function $\omega_B, \omega_L$, target length $L_{i\text{target}}$, coil-to-coil separation $\epsilon_c$ and lengthscale $\ell$.

optimizer, we perform a comparison of the local optimizers SAA and AdamCV initialized at equispaced circular coils in Section 5.2. Then, in Section 5.3 we use our two-step global optimization method to find multiple promising stochastic minima.

### 5.1 Model Data

We perform experiments on a W7-X configuration, see e.g. [15] for a description of W7-X. The stellarator consists of five distinct modular coils, such that after applying the stellarator symmetry as well as 5-field period symmetry we arrive at a total of 50 coils. For this study, we find setting the number of Fourier modes to $N_F = 6$ to be sufficient, which results in a state dimension of $N = 195$. The weights used in our experiments as well as other optimization parameters for FOCUS can be found in Table 5. Heuristically, we found that these weights struck a nice balance between minimizing the field error and finding smooth coils. Moreover, we used $N_{\text{seg}} = 64$ segments per coil as well as $N_{\text{theta}} = N_{\text{zeta}} = 64$ nodes in either discretization direction of the plasma boundary. In order to enforce a reasonable coil-to-coil constraint for all pairs of adjacent coils, we have to take a minimum distance and the width of the coils into account. The latter is crucial as FOCUS models coils as infinitely thin filaments. Therefore, we motivate the value of $\epsilon_c$ by coil separation distances for the W7-X candidate configurations HS-5-7 and HS-5-8 given in [3]. The minimum distance between the coils for the candidate configurations were 0.06m and 0.04m, respectively. The average lateral coil width of the coils was 0.18m, such that we chose the coil-to-coil separation to be at least $\epsilon_c = 0.23 = \frac{1}{2} (0.04 + 0.06) + 0.18m$ in all of our experiments. The coil-to-coil separation constraints were included in the model via a quadratic penalty method, see [24]. Thus all optimizations were performed on the penalty objective with $\lambda \in \mathbb{R}^+$

$$
\tilde{f(x)} := f(x) + \lambda \sum_{i=1}^{N_C-1} \min[c_i(x) - \epsilon_c^2, 0]^2,
$$

with $f(x)$ given in (6) and $c_i(x)$ (7). A $\lambda$ value of 100 was found to be sufficient in consistently achieving constraint satisfaction. Moreover, we set $\alpha = 10000$ for the $\alpha$-quasimax function in (9).

### 5.2 Numerical Results for Local Optimization

We compare the AdamCV algorithm described in Section 4.2.2 to the BFGS algorithm applied to SAA 4.2.3. We initialized the local optimization of (10) from a configuration of equispaced circular coils using the perturbation size $p = 10$mm for all experiments in this section. We ran this experiment for the different perturbation sizes $p = 2$mm, $p = 5$mm, $p = 20$mm and made similar observations.
In order to compare the two algorithms as fair as possible, we set the number of gradient evaluations per step to 10 and the maximum number of gradient evaluations to 50,000. The remaining parameters of AdamCV are set to \( \eta = 0.04 \), \( \gamma = 0.1 \), \( \beta_1 = \beta_2 = 0.95 \), \( \epsilon_A = 10^{-10} \). The SAA approximation was optimized with the deterministic SciPy optimizer BFGS \[32\] and was restarted with new sample values \( u_i \) once a minimum for a fixed sample set had been reached. Due to the restarting, we find that a sample size of 10 is indeed enough for the SAA algorithm, as we find very similar values for e.g., batchsize 100. We plot the final coil sets found by SAA and AdamCV in Figure 3 and find the coils to be similar. We compute the mean squared curvature values

\[
\frac{1}{N_C} \sum_{i=1}^{N_C} \int_0^{2\pi} \kappa_i^2(t) \, dt,
\]

with \( \kappa_i \) being the curvature for coil \( i \) for the respective coil sets: the SAA coils obtain a squared curvature value of 5.313, while the coils optimized with AdamCV are a bit smoother with a mean squared curvature value of 3.134. Additionally, the BFGS needs to evaluate the objective function as many times as the gradient such that we have an additional 50,000 function evaluations adding to the cost.

We measure the quality of the coil sets by looking at 3 measures: the stochastic objective function \( f(x) \) \[10a\], the normal field error \( f_B(x) \) \[2\] and the stochastic normal field error \( \mathbb{E}[f_B(x + U)] \). We provide an overview of our findings in Table 5.2. We find that when optimizing with the AdamCV algorithm we arrive at a similar stochastic function value, with the AdamCV algorithm providing a 0.4% smaller value than the SAA. When we evaluate the stochastic component of the objective function, i.e., the stochastic normal field error, AdamCV finds a 2.1% smaller stochastic field error value than SAA.

We find that AdamCV arrives at slightly lower stochastic function values/field error, at a improved computational expense as it does not need to evaluate the function values additionally to the gradient evaluations. This success is in part due to proper selection of the learning rate parameters \( \eta, \gamma \). The SAA procedure performs similarly well, is easy to implement, and works well with only having to choose the sample size. We recommend restarting the SAA optimization with a new
Table 2: Values of stochastic objective function (10a), normal field error (2) and stochastic normal field error $E[f_B(x + U)]$ for W7-X. The stochastic values are computed using perturbation size $p = 10\text{mm}$ and we are averaging over 1000 realizations of $U$. For the stochastic values we include the 95% confidence interval after the function value.

| Coil Configuration | Stochastic Obj. Function | Field Error | Stochastic Field Error |
|--------------------|--------------------------|-------------|------------------------|
| SAA 10mm           | $4.728 \times 10^{-1} \pm [1.10 \times 10^{-2}]$ | $1.472 \times 10^{-3}$ | $1.972 \times 10^{-3} \pm [1.10 \times 10^{-4}]$ |
| AdamCV 10mm        | $4.707 \times 10^{-1} \pm [1.05 \times 10^{-2}]$ | $1.424 \times 10^{-3}$ | $1.930 \times 10^{-3} \pm [1.05 \times 10^{-4}]$ |

In this section, we describe the global stochastic optimization of a W7-X configuration with model data given in Section 5.1. For our two-stage approach detailed in Section 4, we use the pair DTuRBO and AdamCV for the efficient global search and local refinement steps, respectively. In our experiments, we set the average perturbations amplitudes to $p = 5\text{mm}$ and $p = 10\text{mm}$. The global exploration algorithm DTuRBO was given a maximum number of 100 000 evaluations, a batchsize of 100 and 200 initial evaluations. The bounding boxes for DTuRBO should be set large enough such that there is enough flexibility in the design space, while not so large as to capture poor regions of the design space. As the design variables are Fourier coefficients the box constraints should get narrower for higher-order Fourier modes. To this end, the lower and upper bounds were computed using the variance of the perturbations as an approximate lengthscale. We set the bounding boxes for DTuRBO to be centered around $x_0$, which denotes circular coils of radius 1.5m, resulting in $lb, ub = x_0 \pm \delta \text{Var}[U]$. The scalar $\delta = \frac{1.5}{2 \text{Var}[U_0]}$ resizes the box width such that the translational modes have perturbations bounded by $1.5/2 \text{m}$, where $\text{Var}[U_0]$ is the perturbation variance to the translational mode.

Using this optimization setting, DTuRBO finds around 15 approximate stochastic minima, which were subsequently resolved with the minimizer AdamCV with a maximum number of 2000 iterations with a batchsize of 10 and parameters $\eta = 0.001$, $\gamma = 0.01$, $\beta_1 = \beta_2 = 0.95$, $\epsilon_A = 10^{-10}$. Using this optimization setting, on average, the combined optimization routine DTuRBO & AdamCV found 8 approximate minima within 116 000 evaluations, which had low enough stochastic objective value/field error to use them for further study of physical properties. In Figure 4, we show three final coil sets found by the global optimization for the different perturbation sizes. Although it seems like the coils are close in the plots, the minimum distance is satisfied in all configurations. All of these coil sets achieved a low normal field error $f_B(x)$ and stochastic normal field error $E[f_B(x + U)]$, as seen in Table 5.3. We also compute the 95% confidence interval of each coil configuration and find the size to be at least a magnitude smaller than the respective value, varying slightly with respect to the coil configuration.
Figure 4: Final coil sets for the optimization with DTuRBO and AdamcV. Left: \( p = 5 \)mm; right: \( p = 10 \)mm. Corresponding field error can be found in Table 5.3. For each plot, we sorted and colored the coils according to their stochastic objective value with respect to Table 5.3: low (red), medium (blue), high (black). In plot (a), the two rightmost blue coils appear to be very close. This is deceptive as the rightmost coil extends outward while the other passes behind it.

Comparing to the available literature we find that our total costs for arriving at an approximate stochastic global minimum is less than 0.1\% compared to the evaluation budget in [21], where solely local stochastic optimization has been used. Here, we assume that function evaluations in the respective codes take a similar amount of time/resources and we use a 5-to-1 conversion factor to convert the time for gradient evaluations to the time for function evaluations.\footnote{We run our experiments on an Ubuntu 20.04 cluster with MPI using 14 cores. Across \( 10^4 \) calls to FOCUS, the mean time for the function evaluations is around 0.1s and for the gradient evaluations is around 0.5s.} We attribute this improvement in efficiency to a judicious choice of algorithms and the availability of gradients in FOCUS.

\section{Conclusion and Future Work}

In this paper we develop a stochastic optimization model for stellarator coil configurations in order to hedge against fabrication errors in the construction. Our model considers the effects of normally distributed coil fabrication uncertainties on the normal field error, a length regularization and a coil-to-coil separation distance constraint. Our novel global-to-local approach leverages the efficient high-dimensional derivative-based Bayesian optimizer DTuRBO, as well as SAA and the Adam algorithm equipped with control variates to perform an efficient global exploration. In our numerical experiments for a W7-X-like configuration, we found many satisfactory minima at a low
Table 3: Values of stochastic objective function (10a), normal field error (2) and stochastic normal field error \( \mathbb{E}[f_B(x + U)] \) for 6 different coil configurations for W7-X shown in Figure 4. The stochastic values are computed using the respective perturbation size, and averaged over 1000 realizations of \( U \). For the stochastic values we include the 95% confidence interval after the function value. The color adjacent to the coil configuration denotes the corresponding colored coil set in Figure 4. All coils have been optimized with DTuRBO and AdamCV (D-ACV) and the number in the coil configuration indicates the perturbation size.

| Coil Configuration | Stochastic Obj. Function | Field Error | Stochastic Field Error |
|--------------------|--------------------------|-------------|------------------------|
| D-ACV-5 (red)      | \( 4.360 \times 10^{-1} \pm 2.86 \times 10^{-3} \) | \( 1.472 \times 10^{-3} \pm 2.86 \times 10^{-5} \) |                       |
| D-ACV-5 (blue)     | \( 4.795 \times 10^{-1} \pm 2.78 \times 10^{-3} \) | \( 1.635 \times 10^{-3} \pm 2.78 \times 10^{-5} \) |                       |
| D-ACV-5 (black)    | \( 4.811 \times 10^{-1} \pm 3.19 \times 10^{-3} \) | \( 1.584 \times 10^{-3} \pm 3.19 \times 10^{-5} \) |                       |
| D-ACV-10 (red)     | \( 4.701 \times 10^{-1} \pm 1.06 \times 10^{-2} \) | \( 1.419 \times 10^{-3} \pm 1.06 \times 10^{-4} \) |                       |
| D-ACV-10 (blue)    | \( 5.102 \times 10^{-1} \pm 1.17 \times 10^{-2} \) | \( 1.638 \times 10^{-3} \pm 1.17 \times 10^{-4} \) |                       |
| D-ACV-10 (black)   | \( 5.157 \times 10^{-1} \pm 9.78 \times 10^{-3} \) | \( 1.549 \times 10^{-3} \pm 9.78 \times 10^{-5} \) |                       |

computational expense, approximately less than 0.1% of previous work. Note, that previous work only addresses local stochastic optimization, whereas we perform global stochastic optimization. Possible further directions of work include the investigation of other objective functions which show high sensitivity to coil errors. For instance, [1] [17] showed the magnetic island width to be a quantity highly affected by errors in the coils. Another direction might be to perform this global-to-local approach with e.g., different physical properties using different codes such as SIMSOPT [19] or PyPlasmaOpt [34]. Both of these codes include derivative information, which has been indispensable in this work. Furthermore, the model posed here only considers fabrication uncertainties whereas coil placement and alignment uncertainties, considered in [17], are yet another source of error with a distinct distribution. Future work could investigate the distributional assumptions of the models.

Acknowledgements
Silke Glas and David Bindel acknowledge support from the Simons Foundation in the Collaboration on Hidden Symmetries and Fusion Energy. In addition David Bindel has been supported by the NSF under (1934985) and Ariel Kellison has been supported by the DOE CSGF under (DE-SC0021110). All authors thank M. Landreman for fruitful discussions and support, J.-F. Lobsien for sharing his FOCUS W7-X input file, and C. Zhu for his help with the FOCUS code.

References
[1] T. Andreeva, T. Bräuer, V. Bykov, K. Egorov, M. Endler, J. Fellinger, J. Kißlinger, M. Köppen, and F. Schauer. Tracking of the magnet system geometry during Wendelstein 7-x construction to achieve the designed magnetic field. Nuclear Fusion, 55(6):063025, 2015.
[2] T. Andreeva, T. Bräuer, M. Endler, J. Kisslinger, and U. Toussaint. Influence of construction
errors on wendelstein 7-x magnetic configurations. *Fusion engineering and design*, 84(2-6):408–412, 2009.

[3] C. Beidler, G. Grieger, F. Herrnegger, E. Harmeyer, J. Kisslinger, W. Lotz, H. Maassberg, P. Merkel, J. Nührenberg, F. Rau, et al. Physics and engineering design for wendelstein vii-x. *Fusion Technology*, 17(1):148–168, 1990.

[4] T. Brown, J. Breslau, D. Gates, N. Pumphrey, and A. Zolfaghari. Engineering optimization of stellarator coils lead to improvements in device maintenance. In *2015 IEEE 26th Symposium on Fusion Engineering (SOFE)*, pages 1–6, 2015.

[5] M. Drevlak. Coil designs for a quasi-axially symmetric stellarator. In *In 20th Symposium on Fusion Technology Marseille France, Plasma Physics and Fusion Technology*, p. 883. *Euratom*, 1999.

[6] D. Eriksson, M. Pearce, J. Gardner, R. D. Turner, and M. Poloczek. Scalable global optimization via local bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 32, pages 5496–5507. *Curran Associates, Inc.*, 2019.

[7] P. I. Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[8] J. Hensman, A. Matthews, and Z. Ghahramani. Scalable variational gaussian process classification. In *Artificial Intelligence and Statistics*, pages 351–360. *PMLR*, 2015.

[9] S. P. Hirshman and J. Whitson. Steepest-descent moment method for three-dimensional magnetohydrodynamic equilibria. *The Physics of fluids*, 26(12):3553–3568, 1983.

[10] S. Hudson, R. Dewar, G. Dennis, M. Hole, M. McGann, G. Nessi, and S. Lazerson. Computation of multi-region relaxed magnetohydrodynamic equilibria. *Physics of Plasmas*, 19, 11 2012.

[11] M. Jankowiak, G. Pleiss, and J. Gardner. Parametric gaussian process regressors. In *International Conference on Machine Learning*, pages 4702–4712. *PMLR*, 2020.

[12] S. Kim, R. Pasupathy, and S. G. Henderson. A guide to sample average approximation. *Handbook of simulation optimization*, pages 207–243, 2015.

[13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

[15] T. Klinger, T. Andreeva, S. Bozhenkov, C. Brandt, R. Burhenn, B. Buttenschön, G. Fuchert, B. Geiger, O. Grulke, H. Laqua, et al. Overview of first wendelstein 7-x high-performance operation. *Nuclear Fusion*, 59(11):112004, 2019.
[16] J. Kremer, T. S. Pedersen, N. Pumphrey, W. Reiersen, and F. Dahlgren. The status of the design and construction of the columbia non-neutral torus. In *AIP Conference Proceedings*, volume 692, pages 320–325. American Institute of Physics, 2003.

[17] J. P. Kremer. The creation and first studies of electron plasmas in the columbia non-neutral torus. *Ph. D. Thesis*, 2007.

[18] M. Landreman. An improved current potential method for fast computation of stellarator coil shapes. *Nuclear Fusion*, 57(4):046003, feb 2017.

[19] M. Landreman, B. Medasani, F. Wechsung, A. Giuliani, R. Jorge, and C. Zhu. Simsopt: A flexible framework for stellarator optimization. *Journal of Open Source Software*, 6(65):3525, 2021.

[20] M. Lange, D. Zühlke, O. Holz, and T. Villmann. Applications of lp-norms and their smooth approximations for gradient based learning vector quantization. In *ESANN*, pages 271–276, 2014.

[21] J.-F. Lobsien, M. Drevlak, T. Kruger, S. Lazerson, C. Zhu, and T. S. Pedersen. Improved performance of stellarator coil design optimization. *Journal of Plasma Physics*, 86(2):815860202, 2020.

[22] J.-F. Lobsien, M. Drevlak, T. S. Pedersen, et al. Stellarator coil optimization towards higher engineering tolerances. *Nuclear Fusion*, 58(10):106013, 2018.

[23] P. Merkel. Solution of stellarator boundary value problems with external currents. *Nuclear Fusion*, 27(5):867–871, may 1987.

[24] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[25] R. Orbach. Statement about the future of the princeton plasma physics laboratory. *Under Secretary for Science and Director, Office of Science, US Department of Energy*, 2008.

[26] M. Padidar, X. Zhu, L. Huang, J. R. Gardner, and D. Bindel. Scaling gaussian processes with derivative information using variational inference, 2021.

[27] E. Parzen. *Stochastic processes*. SIAM, 1999.

[28] A. Shapiro. Monte carlo simulation approach to stochastic programming. In *proceeding of the 2001 winter simulation conference (cat. no. 01CH37304)*, volume 1, pages 428–431. IEEE, 2001.

[29] J. C. Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control*, volume 65. John Wiley & Sons, 2005.

[30] D. J. Strickler, L. A. Berry, and S. P. Hirshman. Designing coils for compact stellarators. *Fusion Science and Technology*, 41(2):107–115, 2002.

[31] W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 12 1933.
[32] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

[33] C. Wang, X. Chen, A. J. Smola, and E. P. Xing. Variance reduction for stochastic gradient optimization. In *Advances in Neural Information Processing Systems*, pages 181–189, 2013.

[34] F. Wechsung, A. Giuliani, M. Landreman, A. Cerfon, and G. Stadler. Single-stage gradient-based stellarator coil design: stochastic optimization, 2021.

[35] C. Zhu, D. A. Gates, S. R. Hudson, H. Liu, Y. Xu, A. Shimizu, and S. Okamura. Identification of important error fields in stellarators using the hessian matrix method. *Nuclear Fusion*, 59(12):126007, 2019.

[36] C. Zhu, S. R. Hudson, Y. Song, and Y. Wan. New method to design stellarator coils without the winding surface. *Nuclear Fusion*, 58(1):016008, nov 2017.