Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

SF-SFD: STOCHASTIC OPTIMIZATION OF FOURIER COEFFICIENTS TO GENERATE SPACE-FILLING DESIGNS

Manisha Garg

Department of Mathematics
University of Illinois Urbana-Champaign
Urbana, IL, USA 61801

Tyler H. Chang

Mathematics and Computer Science Division
Argonne National Laboratory
Lemont, IL, USA 60439

Krishnan Raghavan

Mathematics and Computer Science Division
Argonne National Laboratory
Lemont, IL, USA 60439

ABSTRACT

Due to the curse of dimensionality, it is often prohibitively expensive to generate deterministic space-filling designs. On the other hand, when using naïve uniform random sampling to generate designs cheaply, design points tend to concentrate in a small region of the design space. Although, it is preferable in many cases to utilize quasi-random techniques such as Sobol sequences and Latin hypercube designs over uniform random sampling, these methods have their own caveats especially in high-dimensional spaces. In this paper, we propose a technique that addresses the fundamental issue of measure concentration by updating high-dimensional distribution functions to produce better space-filling designs. Then, we show that our technique can outperform Latin hypercube sampling and Sobol sequences by the discrepancy metric while generating moderately-sized space-filling samples for high-dimensional problems.

1 INTRODUCTION

Design of experiments is a critical first step in numerous application areas including statistical response surface methodology (RSM), surrogate-based optimization, and modeling of complex systems (Myers et al. 2016). To give a few examples, deterministic, random, and quasi-random experimental designs are generally applied in engineering applications ranging from particle accelerator designs (Neveu et al. 2022), to high-performance computing (HPC) performance analysis (Wang et al. 2023), and generic blackbox optimization solvers (Custódio and Madeira 2018; Chang and Wild 2023).

In such applications, a set of design points is generated and evaluated (through simulation or experimentation) to produce an initial data set. This data set is used to fit a surrogate model of the underlying blackbox process, which is then used for the purpose of approximation or optimization in downstream applications. The accuracy of the resulting approximation or global convergence of the optimization is greatly affected by the quality of the initial design of experiments.

To facilitate the initial design, a significant amount of research (Garud et al. 2017; Joe and Kuo 2008; Johnson et al. 1990) and software (Attia and Ahmed 2023; Lee, Abraham et al. 2015; Virtanen et al. 2020; Wang and Dowling 2022) are dedicated to generating space-filling experimental designs and intelligent sampling techniques. While approaches may differ based on specific applications, in the context of surrogate modeling and design optimization for deterministic processes, a space-filling design will constitute data

979-8-3503-6966-3/23/$31.00 ©2023 IEEE

sample locations. These locations are typically obtained from a simply-bounded subset of $\mathbb{R}^d$, which we will refer to as the *design space*. Then the problem is that of generating data samples that guarantee accuracy and representation across a large percentage of the design space. One challenge for generating design of experiments is that the deterministic techniques such as those in Myers et al. (2016), Ch. 3 & 4 require exponentially many samples with increasing dimension.

It is known that the need for computational resources increases exponentially with the number of samples and the dimensionality of data. Although, with the advent of high performance computers, it is sometimes possible to fulfill the computational necessity for moderate dimensions, the concentration of measure still prevalent in high dimensions is rather tricky to handle (Gorban and Tyukin 2017). Therefore, favorable properties in high-dimensions is a prime requirement in generation of space-filling designs. These designs are often based upon quasi-random and low-discrepancy sequences that attempt to produce random samples . While these techniques are cheap and effective, they only address the fundamental challenge of measure collapse in high-dimensions heuristically.

In this paper, we propose a novel technique (SF-SFD) for generating high-dimensional distribution functions. Our technique is designed to generate probability distribution functions (pdfs) that are as robust as possible against measure collapse. One can sample from these pdfs to produce high-dimensional space-filling designs with high probability. To achieve this, we will optimize the Fourier coefficients of the pdf in order to minimize the expected statistical discrepancy of each sample. This approach directly addresses the collapse of the underlying distribution and produces better space-filling designs in high dimensions than other randomized methods on a limited budget.

The remainder of this paper is organized as follows. In Section 2 we will provide additional information on techniques for design of experiments and metrics for assessing their quality. We conclude that for the class of problems that we are interested in, randomized and quasi-randomized methods are most appropriate. In Section 3 we will explain how a concentration of measure makes randomized techniques ineffective for high-dimensional problems. In Section 4 we will introduce a novel method for tuning pdfs in order to slow the concentration of measure. In Section 5 we will provide some initial results showing that our method can succeed beyond random sampling, and even becomes more effective than other state-of-the-art techniques in very high-dimensional design spaces. In Section 6 we will summarize these results and summarize the next steps for this work.

2 BACKGROUND

Existing experimental design methods can be broadly categorized as adaptive methods, which utilize response values when selecting sample points, and non-adaptive methods, which do not. It is well-known that *adaptive* search methods are often more efficient in practice, when only considering the accuracy in solving a given task as a function of the number of samples taken. Well-known adaptive search techniques include, DIRECT (Jones et al. 1993), Bayesian optimization (Garnett 2023), and various forms of active learning (Sapsis and Blanchard 2022). However, *non-adaptive* methods are useful in situations where large batches are needed (e.g., pre-planning batched chemistry experiments); can be used to multi-start local modeling and optimization techniques; and are used to initialize many adaptive techniques (e.g., Latin hypercubes to start Bayesian optimization). Therefore, our focus in this paper is limited to non-adaptive methods, which should be considered separately.

The goal of non-adaptive methods in the space-filling design setting is to obtain a design $\mathcal{X}$ consisting of n points from a simply-bounded $d-$dimensional region of $\mathbb{R}$. Without loss of generality, we assume that we are sampling from the unit cube $[0,1]^d$. Since the response values of the samples are not available at this time, the utility of a sample is measured purely by how well it fills the space. Therefore, it should (approximately) solve

$$\max_{\{\mathcal{X}:\mathcal{X}\subset[0,1]^d,|\mathcal{X}|=n\}} T(\mathcal{X}) \quad \text{or} \quad \min_{\{\mathcal{X}:\mathcal{X}\subset[0,1]^d,|\mathcal{X}|=n\}} T(\mathcal{X}), \tag{1}$$

where $T(\cdot)$ is a design optimality criteria, and n is the sample size. The nature of $T(\cdot)$ determines whether the problem is to maximize or minimize. Common examples of design optimality criteria $T(\cdot)$ include

1. the discrepancy of the sample (should be minimized) (Joe and Kuo 2008),
2. the A, E, or D-optimality score of the information matrix (should be maximized) (Attia and Ahmed 2023; Wang and Dowling 2022), and
3. geometry-based criteria such as maximin (maximized) and minimax (minimized) distance criteria (Johnson et al. 1990; Pronzato 2017).

In this work, the sample size n will typically be in the hundreds. There are numerous applications where n may be larger or smaller, but this is an appropriate range for many real-world design optimization problems such as those listed in Section 1 (Chang and Wild 2023; Custódio and Madeira 2018; Neveu et al. 2022; Wang et al. 2023).

The A, E, and D optimality criteria are based on information theory, and generally require access to the Fisher information matrix. Therefore, many existing techniques for generating samples based on these criteria are model-based and not applicable for our blackbox setting (Wang et al. 2023). Geometric distance-based criteria, such as maximin distance, can be related to properties of the Fisher information matrix, but are easier to compute for a generic surrogate modeling method. However, such designs can still be combinatorially expensive to generate in high dimensions due to their connection to Delaunay triangulations (Pronzato 2017). All of these optimality criteria are commonly used in the field of optimal experimental design, where the location of each design point is posed as a variable in an optimization problem (Attia and Ahmed 2023). Although these techniques are appropriate when n is small in applications such as sensor placement, these techniques are difficult to scale for our target application.

In this paper we will mainly focus on capturing the effect of measure concentration. While additional details about measure concentration can be found in section 3, in brief, we are concerned with the phenomenon where uniformly distributed random samples congregate to a small region of the sample space. As this behavior reflects through the imbalance in the density function, discrepancy of a sample is an appropriate metric to capture it (Kuipers and Niederreiter 1974). Precisely, let $\mathcal{Y}$ be an infinite sequence of points in $[0,1]^d$, and let $\mathcal{Y}_N$ denote the first N points in $\mathcal{Y}$. Then

$$D_N(\mathcal{Y}) = \sup_{B \in J} \left| \frac{|\{\mathbf{y_i} : \mathbf{y_i} \in B \text{ and } y_i \in \mathcal{Y}_N\}|}{N} - \mu(B) \right|, \tag{2}$$

where $\mu(\cdot)$ is the Lebesgue measure in $\mathbb{R}^d$ and J is the set of all Lebesgue measurable subsets of $[0,1]^d$.

The sequence $\mathcal{Y}$ is said to be low-discrepancy if $\lim_{N\to\infty} D_N(\mathcal{Y}) = 0$, and the discrepancy $D_N(\cdot)$ is often used as a measure of a finite sample's uniformity. When used as a measure of uniformity, a discrepancy $D_n(\mathcal{X})$ that is close to 1 corresponds to an imbalance in the distribution of $\mathcal{X}$ in $[0,1]^d$, while a discrepancy close to 0 corresponds to a general uniformity of $\mathcal{X}$ in $[0,1]^d$. Since it is desirable to fill all areas of the design region when sampling, low-discrepancy samples are considered better, and many optimization libraries use low-discrepancy sequences, such as the Sobol sequence (Balandat et al. 2020; Custódio and Madeira 2018) as a substitute for uniform-random sampling.

Moreover, as it is impossible to compute the exact discrepancy as it is defined in (2), various approximations of the discrepancy are utilized in the literature. One may refer to Kuipers and Niederreiter (1974) for further details. We use $\mathcal{L}_2$ discrepancy for our approach, which is a special case of the $\mathcal{L}_p$ discrepancy defined as

$$\mathcal{L}_p(\mathcal{X}) := \left(\int_{[0,1]^d} \left| \frac{|\mathcal{X} \cap [0,\mathbf{x})|}{N} - \mu\left([0,\mathbf{x})\right) \right|^p d\mathbf{x} \right)^{(1/p)}$$

where $1 \leq p < \infty$.

In this work, we use the centered $\mathcal{L}_2$ discrepancy of Hickernell (1998)

$$\mathcal{L}_2^C(\mathcal{X}) = \left(\frac{13}{12}\right)^d - \frac{2}{n}\sum_{i=1}^{n}\prod_{k=1}^{d}\left(1+\frac{1}{2}\left|x_{i,k}-0.5\right| - \frac{1}{2}\left|x_{i,k}-0.5\right|^2\right)$$
$$+\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\prod_{k=1}^{d}\left(1+\frac{1}{2}\left|x_{i,k}-0.5\right|+\frac{1}{2}\left|x_{j,k}-0.5\right|-\frac{1}{2}\left|x_{i,k}-x_{j,k}\right|^2\right)$$

where x_i and x_j are points in $\mathcal{X}$, and $x_{i,k}$ indicates the kth component of x_i. Note that although the discrepancy should take a value between 0 and 1, $\mathcal{L}_2^C$ is an approximation based on numerical quadrature, which can take on much larger values for ill-spaced samples. In particular, note that when all $x_i \in \mathcal{X}$ are clustered near the center of the design region $[0,1]^d$ (as is often the case during measure collapse), the value of $\mathcal{L}_2^C(\mathcal{X})$ may approach $\left(\frac{13}{12}\right)^d - 1$. This is an important observation since we will observe large values of $\mathcal{L}_2^C(\mathcal{X})$ in Section 5.

Many techniques have been implemented to produce samples of low discrepancy (Sobol 1967; Wong et al. 1997). In practice, the randomized Sobol sequence (Bratley and Fox 1988; Joe and Kuo 2003; Joe and Kuo 2008) is most commonly used in RSM applications (Custódio and Madeira 2018). However, for the Sobol sequence and other low-discrepancy sequences, performance can still degrade drastically for sample sizes that are overly small with respect to the dimension, or for sample sizes that are not multiples of some preferred size (e.g., powers of 2).

Therefore, many RSM applications use the heuristic of Latin hypercube sampling (Neveu et al. 2022; Chang and Wild 2023; Müller 2017). This is even typically recommended as a means to start adaptive sampling techniques, such as Bayesian optimization (Garnett 2023, Ch. 9.3). While Latin hypercube sampling is effective in practice, it is essentially generates samples that are stratified over a single dimension, which is useful for single-variable analysis. However, there are no guarantees of uniformity over multiple dimensions, and analyses based on Latin hypercubes may miss multivariate interactions. Therefore, Latin hypercubes are only considered optimal when they have been optimized with respect to another design optimality criteria, such as A, E, or D optimality, which is a combinatorially hard problem (Viana 2016).

2.1 Summary and Key Challenges

It is well-known that as the dimension of the space grows, the number of samples needed to construct an accurate statistical, numerical, or machine learning model also grows exponentially. This challenge is known as the *curse of dimensionality*. Therefore, for a fixed sample size, the quality of the surrogate model will reduce with increasing dimension, regardless of the sampling technique used due to a reduction in sample density.

In regimes where n is too large for an optimal experimental design and d is too large for a deterministic design of experiments, applications typically resort to Sobol sequences (Custódio and Madeira 2018) or Latin hypercube designs (Chang and Wild 2023; Müller 2017). Both of these techniques are generally seen as approximations to uniform random sampling with better high-dimensional properties. However, none of these directly address the fundamental issue, which is the collapse of the measure in high dimensions.

In our case, when a sample point x_i is drawn uniformly from $[0,1]^d$, its squared distance to the center of the design space is given by $\|x_i - \frac{1}{2}\|_2^2 = \sum_{k=1}^{d}(x_{i,k}-\frac{1}{2})^2$. So for each component $k = 1, \ldots, d$,

$$\mathbb{E}\left[\left(x_{i,k}-\frac{1}{2}\right)^2\right] = \int_0^1\left(x-\frac{1}{2}\right)^2 dx = \frac{1}{3}-\frac{1}{2}+\frac{1}{4} = \frac{1}{12}$$

and the variance of the expected value is a finite constant v.

Thus, by the central limit theorem (CLT), for all $x_i \in \mathcal{X}$, $\mathbb{E}[\|x_i - \frac{1}{2}\|_2^2] = \frac{d}{12}$ with variance $\frac{v}{d}$. So, x_i will concentrate on a sphere of radius $\sqrt{\frac{d}{12}}$ centered at $(0.5,\ldots,0.5)^\top$ with vanishing standard deviation as d increases. For large values of d, this will leave both the center and corners of the design space empty.

This motivates the need for a technique that directly addresses this issue of measure collapse and scales to larger samples in moderate to high dimensional design spaces. In the following section, we will explain how the measure collapses for high-dimensional samples and what can be done to address this issue in the context of sampling for surrogate modeling.

3 CONCENTRATION OF MEASURE IN HIGH DIMENSIONS

Consider a random variable X in $\mathbb{R}^d$ and let the probability density function (pdf) be given as $p(x)$. Say, we seek to write the exact expression for the pdf. Traditionally, one would begin by writing the characteristic function $C(t,X)$ for the random variable X which is achieved by applying a Fourier transform on the random variable X and writing a series expansion on the Fourier transform (Adeniran et al. 2020). Particularly, in the case when X corresponds to continuous Lebesgue measure, the characteristic function is written as

$$C(t,X) = \mathbb{E}\left[e^{itX}\right] = \int \left[e^{itX}\right] d\mu, \tag{3}$$

where $\mu \in \mathbb{R}^d$ is a Lebesgue measure on a d dimensional vector space. The precise derivation of the density from this point requires evaluating the inversion of the characteristic function through the Levy's inversion formulae (Hewitt 1953). This precise expression is given as

$$p(x) = \int e^{-it\mathbb{E}\left[e^{itX}\right]} dt, \tag{4}$$

where $p(x)$ describes the probability density function (pdf). Solving the integral on the right hand side would provide the pdf of Gaussian. To solve the integral (4), we must expand $e^{-it\mathbb{E}\left[e^{itX}\right]}$ expansion applied on the expected value of e^{itX}. Therefore, the quality of the pdf estimation depends on the availability of samples from X. In the scenario when $d \to \infty$, the volume will concentrate (Gorban and Tyukin 2017), which means that the samples from X will describe a small region from the high dimensional space. In other words, the volume of a cube that was well spread across different areas becomes concentrated towards as a sphere around the origin point as detailed in the previous section.

This phenomenon leads to two scenarios, first, the estimation of $\mathbb{E}\left[e^{itX}\right]$ does not change for different samples of X as the concentrated volume will provide the same samples over and over again. Therefore, the inversion will provide the same density values for different samples assuming that the inverse is well defined. Second, the concentration phenomenon leads to a situation where the presence of many zeros or close to zero values in the data samples will introduces zero modes in $\mathbb{E}\left[e^{itX}\right]$. In the matrix sense, many of the eigenvalues of $\mathbb{E}\left[e^{itX}\right]$ will end up being zero and will lead to singular modes in the density. That is, the *empirical* density function will be ill-defined in many regions of the design space. These two issues prevalent in high dimensions will prohibit the use to uniform sampling because uniform sampling from the original design space will end up with an ill-defined empirical distribution function.

An alternative avenue is to generate samples more intelligently by considering the density rather than blind uniform sampling. Towards this end, we will not solve the problem by sampling-based likelihood estimation like a Monte Carlo approach, but, rather construct a distribution function that is optimized through an iterative procedure. This distribution function is an approximation of inverse map $M^{-1}\mathbb{E}\left[e^{itX},\right]$, on the design space. In particular, we will write the pdf as a linear combination of individual terms of the Fourier series with coefficients. Then, we will solve for the coefficients through an iterative optimization approach. By successively deriving samples and the corresponding pdf, we will attempt to find the approximated pdf that best explains the distribution of the data. It is our hypothesis that by optimizing the distribution function instead of the likelihood we will sidestep the impact of measure concentration on sample and the density will be better defined. In what follows next, we will detail our approach and later describe the advantages of our method in this domain through a simulation study.

4 OUR APPROACH

Now that we have established the issue with existing sampling techniques and the issue of measure collapse, we are ready to propose our solution, which we refer to as the Stochastic Fourier space-filling design (SF-SFD). In this approach, we create a sample, $\mathcal{X}$, of any size n from a simply bounded $d-$dimensional space $[0,1]^d \subset \mathbb{R}^d$. We achieve this by optimizing the Fourier coefficients of the pdf, in order to compensate for measure collapse at the specified values of n and d.

To measure the collapse of the pdf, we use the centered $\mathcal{L}_2$ discrepancy, $\mathcal{L}_2^C(\mathcal{X})$ from Section 2. We argue that the discrepancy is an appropriate choice of performance metrics since large discrepancy values are correlated to measure collapse. Additionally, the centered $\mathcal{L}_2$ approximation will fail and produce unrealistically large numbers when points are overly clustered near the origin. Since the radius of our sphere of concentration (see Section 2) is a small percentage of the total design space, excessively large discrepancies will be indicative of measure collapse.

At first glance, the fact that we are optimizing our design toward well-known optimality criteria could be seen as similar to optimal experimental design (Attia and Ahmed 2023; Wang et al. 2023) or Latin hypercube optimization (Viana 2016). However, we focus on optimizing properties of the underlying pdf to prevent a collapse of measure, rather than focusing on the placement of individual data points. This is a key difference, and we believe that the proposed technique will be more scalable while also directly addressing the underlying issue suffered by naïve randomized techniques. This strategy is motivated by the connection between measure collapse and singularity of the Fourier transform of the pdf, as discussed in Section 3.

Our approach can be summarized by the following 3-step process:

1. We create an initial probability distribution function for SF-SFD based on a uniform distribution.
2. We take a discrete Fourier transform (DFT) of the square-roots of the probabilities to obtain tunable coefficients. See Section 4.1 for further details.
3. Since the DFT is a unitary operator, we can take perturbations on the surface of the unit sphere in order to generate new (square-root) probability density functions. We will use a constrained optimization procedure to iteratively generate perturbations to our Fourier coefficients with the objective of minimizing the expected *empirically observed* discrepancy of the resulting pdf. See Section 4.2 for more details.

The process described above is outlined in Algorithm 1, with further details in Sections 4.1 and 4.2.

Algorithm 1 SF-SFD

```
Let $P^{(1,m)}$ denote the current 1D pdf and $Q^{(1,m)}$ denote $\sqrt{P^{(1,m)}}$, as described in Section 4.1;
Let $C^{(1,m)}$ denote the complex-valued FFT of $Q^{(1,m)}$ as described in Section 4.1;
Let $\theta$ denote the optimization variables, as described in Section 4.2;
$a_i$ is the current number of draws to estimate expected-value, as described in Section 4.2;
**Initialize** $P^{(1,m)} =$ 1D uniform distribution; $Q^{(1,m)}$, $C^{(1,m)}$, and $\theta$ are set accordingly;
**while** optimization stop conditions not met **do**
    Optimizer iterates to generate perturbation $C'^{(1,m)}$ to $C^{(1,m)}$;
    Reverse the process from Section 4.1 to recover the perturbed 1D pdf $P'^{(1,m)}$;
    Estimate expected discrepancy by drawing $a_i$ iid $d$-dimensional samples of size $n$ from $P'^{(1,m)}$;
    For the next iteration, update: $P^{(1,m)} \leftarrow P'^{(1,m)}$;
    Update $Q^{(1,m)}$, $C^{(1,m)}$, and $\theta$ accordingly;
    Increment $a_i$ if needed;
    Return the estimated expected discrepancy to the optimizer for the next iteration;
**end while**
```

4.1 Obtaining the Fourier coefficients

In Step 1 of our three-step process, we start from a uniform distribution on $[0,1]^d$. Ideally, we would like to optimize the Fourier transform of this d-dimensional distribution function $P^{(d)}$, but this would be computationally intractable for large values of d. Instead, we make the simplifying assumption that our distribution function will always be symmetrical in each dimension (in other words, the projection onto each coordinate axis is identical). This allows us to instead train a one-dimensional distribution function $P^{(1)}$ and draw d i.i.d. samples from it to obtain a single d-dimensional design point. In order to make the optimization problem finite, we discretize $P^{(1)}$ into m discrete cells of equal mass $p_1, \ldots, p_m$. This defines our one-dimensional probability mass function $P^{(1,m)}$.

Next, let $Q^{(1,m)} = \{q_i\}_{i=1}^m$, where $q_i = \sqrt{p_i}$ for $i = 1, \ldots, m$. This is the square-root probability mass function, upon which we now apply a DFT. In this work, we use the unitary form of the DFT such that the 2-norms are preserved in the Fourier space and through the inverse Fourier transform (IFT) by Parseval's theorem (Lax 2002, Theorem 21). Since $\sum_{i=0}^m |q_i|^2 = 1$, it implies $\sum_{i=0}^m |c_i|^2 = 1$ where $C^{(1,m)} = \begin{bmatrix} c_0 & c_1 & \cdots & c_m \end{bmatrix}$ are the complex-valued Fourier coefficients. We can then perturb the coefficients c_i for $i \in \{0,\ldots,m\}$ to obtain c_i' such that $\sum_{i=0}^m |c_i'|^2 = 1$ is still satisfied. Once we have the new coefficients for $C'^{(1,m)}$, we can invert the above process to obtain the updated square-root mass function $Q'^{(1,m)}$, and eventually our updated one-dimensional mass function $P'^{(1,m)}$. Since we are assuming that the distribution function is symmetric in all dimensions, we can draw a d-dimensional sample of size n via $d \times n$ i.i.d. draws from $P'^{(1,m)}$ to sample the perturbed d-dimensional pdf.

4.2 Optimization of Fourier Coefficients

In order to tune our pdf using optimization (Step 3 above), we can introduce any perturbation to the Fourier coefficients, so long as the coefficients have a unit 2-norm. These perturbations on c_i can be chosen to specifically decrease $\mathbb{E}\left[\mathcal{L}_2^C(\mathcal{X})\right]$. In order to tune this criteria while maintaining the requirement that $\sum_{i=0}^m |c_i'|^2 = 1$, we must solve an optimization problem on a $(m-1)$-dimensional complex unit sphere (Bloch sphere).

To put this concisely, we are solving the complex-valued optimization problem

$$\min_{C^{(1,m)} \in \mathbb{C}^m} \mathbb{E}\left[L_2^C(X)|X \sim P^{(d,m)}; \sum_i |c_i|^2 = 1\right]$$

where $P^{(d,m)}$ is a d-dimensional pdf with implicit dependence upon $C^{(1,m)}$, via the process from Section 4.1.

In practice, it is not easy to solve a complex-valued optimization problem with nonlinear (spherical) constraints. Thus, we represent c_i in polar form eliminating the need for the spherical constraint and complex variables. Specifically, we generate the real-valued Euler angles θ_i, $i = 1, \ldots, 2m-1$, which point to coordinates on the m-dimensional complex sphere. These angles can be optimized by a blackbox solver subject only to the linear constraints $0 \leq \theta_i \leq 2\pi$, $i = 1, \ldots, 2m-1$. Again, note that because we assume symmetry in all dimensions, the dimension of the optimization problem depends linearly on m, but is independent of both d and n.

Recall that we are attempting to generate a distribution function that can be used to produce good samples of size n in $[0,1]^d$. Thus, our objectives are based on the expected performance of a realized sample of a pre-specified size n from the resulting distribution. Note that the above is a stochastic problem since there will be variation in each individual design drawn from our pdf. In order to address this problem, in this work, we estimate the expected value of each performance measure based on an average over a_i designs of size n drawn from the resulting distribution. In order to guarantee convergence to the true expected value, we gradually increase our sample size a_i with the iteration index i. In this work, we start with a sample size of $a_1 = 50$ and increase a_i by 1 every 10 iterations. This guarantees that the error term in the stochastic approximation (SA) vanishes in the limit, which is a necessary condition to the convergence of SA methods (Lai 2003).

Now that we have posed a bound-constrained optimization problem of moderate dimension, we calculate $\mathcal{L}_2^C\mathcal{X}$ using the implementation in `scipy.stats.qmc` (Hickernell 1998; Virtanen et al. 2020). Then we use the COBYLA implementation in `scipy.optimize.minimize` for solving this bound-constrained blackbox optimization problem (Powell 1994; Virtanen et al. 2020). Although COBYLA was originally proposed as a deterministic solver, due to its similarities to stochastic gradient descent, COBYLA is known to perform very well for stochastic problems such as this (Shi et al. 2021).

5 RESULTS

To test our method, we compare the expected discrepancy of our final pdf against the expected discrepancy of a Latin hypercube sample (LHS), Sobol sequence (Sobol), and random sample of size n. In the case of SF-SFD, we have optimized our pdf using the exact methods described in Section 4. For the discretization of our mass function, we used a value of $m = 10$, and for COBYLA we use the default setting in `scipy` (Virtanen et al. 2020). For the comparisons, we have used the Latin hypercube sampling and Sobol sequence implementations from `scipy.stats.qmc` (Roy et al. 2023). In every case, we average results over 10 distinct random seeds.

We have performed experiments with sample sizes of $n = 100$, 200, 300, 400, and 500, and $d = 5$, 10, 15, 20, 25, and 30. The averaged discrepancies for all four methods at all problem sizes are shown in Table 1 and the increase in discrepancy with d (averaging over all values of n) is plotted in Figure 1.

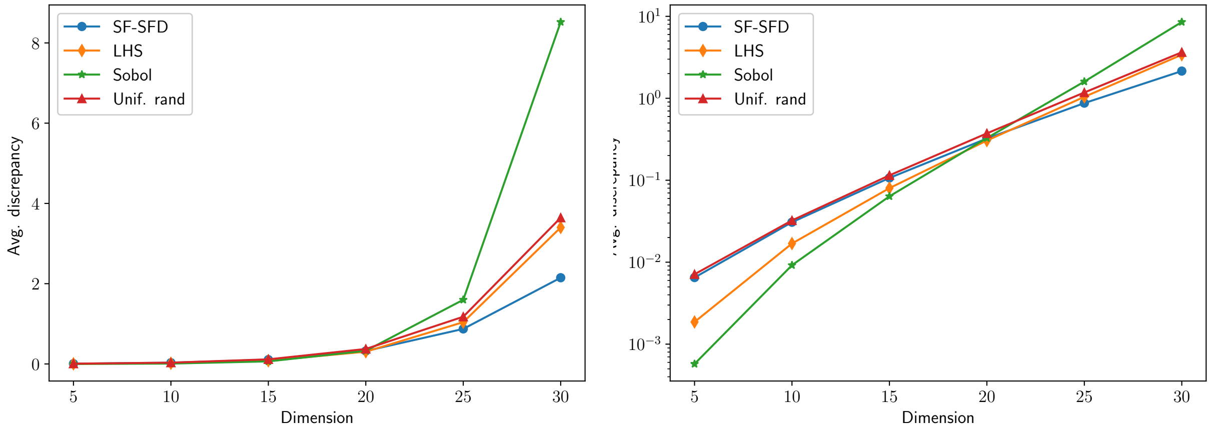

Figure 1: Average discrepancy of generated samples with increasing dimension at linear (left) and logarithmic (right) scales.

As seen in Figure 1, in low dimensions our method performs similarly to random sampling for the tested values of n. On the other hand, the Sobol sequence and Latin hypercube samples (LHS) are significantly better. As the dimension increases, LHS becomes increasingly similar to uniform random sampling, while the Sobol sequence eventually becomes worse than random sampling as the value of n becomes too sparse with respect to d. On the other hand, our method performs significantly better than uniform random sampling in high dimensions, and greatly slows the rate of measure collapse. Based on these results, for values of n in the low hundreds, our method is preferable when $d \geq\sim 20$. In Table 1 we see that SF-SFD overtakes LHS and Sobol sooner for small values of n.

It is worth noting that at these problem sizes, the $\mathcal{L}_2^C$ discrepancy approximation becomes extremely inaccurate due to the bunching of samples near the origin, leaving the corners of the $[0,1]^d$ completely empty. This is reflected in discrepancy approximations that well-exceed one. Even our method suffers from this issue, but we are able to mitigate the issue in comparison to other techniques. In general, although the

Dimension	**Method**	**Sample sizes**				
		100	**200**	**300**	**400**	**500**
5	SF-SFD	0.0142	0.0071	0.0047	0.0036	0.0029
	LHS	0.0042	0.0020	0.0014	0.0010	0.0008
	Sobol	**0.0017**	**0.0006**	**0.0003**	**0.0002**	**0.0001**
	Unif. Rand.	0.0157	0.0078	0.0052	0.0039	0.0031
10	SF-SFD	0.0670	0.0335	0.0226	0.0170	0.0136
	LHS	0.0376	0.0179	0.0125	0.0091	0.0073
	Sobol	**0.0240**	**0.0100**	**0.0058**	**0.0037**	**0.0025**
	Unif. Rand.	0.0710	0.0353	0.0236	0.0177	0.0141
15	SF-SFD	0.2259	0.1183	0.0801	0.0607	0.0487
	LHS	0.1765	0.0870	0.0578	0.0444	0.0348
	Sobol	**0.1645**	**0.0659**	**0.0395**	**0.0276**	**0.0207**
	Unif. Rand.	0.2520	0.1255	0.0834	0.0627	0.0501
20	SF-SFD	**0.6557**	0.3628	0.2500	0.1907	0.1549
	LHS	0.6688	0.3357	0.2212	0.1665	0.1330
	Sobol	0.8618	**0.3331**	**0.1979**	**0.1386**	**0.1052**
	Unif. Rand.	0.8171	0.4089	0.2723	0.2044	0.1636
25	SF-SFD	**1.6263**	**0.9964**	**0.7221**	**0.5621**	0.4591
	LHS	2.2569	1.1541	0.7625	0.5745	0.4564
	Sobol	4.4776	1.5757	0.8858	0.6033	**0.4514**
	Unif. Rand.	2.5771	1.2875	0.8578	0.6439	0.5154
30	SF-SFD	**3.7137**	**2.4248**	**1.8572**	**1.4976**	**1.2504**
	LHS	7.3923	3.7545	2.4905	1.8693	1.4979
	Sobol	25.5500	8.0504	4.2211	2.7673	2.0118
	Unif. Rand.	7.9939	3.9847	2.6560	1.9928	1.5952

Table 1: Empirical expected value of $\mathcal{L}_2^C(\mathcal{X})$ averaged over 10 random seeds for SF-SFD, Latin hypercube sampling (LHS), Sobol sequences, and uniform random sampling at various dimensions (d) and sample sizes (n). All values are rounded to 4 decimal places. The best performing method at each problem size is emphasized in bold.

measure concentration may be impossible to avoid, we are able to slow the rate of concentration significantly through SF-SFD's optimization procedure.

6 CONCLUSION AND FUTURE WORK

In this paper we have proposed a novel method, which we call SF-SFD, for tuning distribution functions in high-dimensional spaces in order to prevent concentration of measure for a finite sample. We argue that this technique directly addresses the issue of measure concentration and scales better to large dimensions than existing heuristic techniques such as Latin hypercube samples and low-discrepancy sequences such as the randomized Sobol sequence. We use SF-SFD to generate space-filling design at several common problem dimensions, and show that the average $\mathcal{L}_2^C$ discrepancy for our designs grows more slowly than other techniques as the dimension of the problem becomes extremely large. The experimental results and analysis presented in this paper can be reproduced by accessing the corresponding GitHub repository at https://github.com/sfdsampling/sfsfd. The repository contains the necessary code, datasets, and instructions to replicate the experiments and generate the reported results.

Although initially it is difficult to compete with quasi-random methods such as Sobol sequences or even Latin hypercubes, for dimensions exceeding twenty, sample sizes of $100-500$ are not large enough to guarantee reasonable discrepancies with these methods. In fact, for the $\mathcal{L}_2^C$ discrepancy approximation used in this paper, samples become so concentrated that the discrepancy estimates begin to blow up beyond the reasonable range. While all methods are affected, ours is able to slow this rate of concentration as much as possible, achieving better performance for sparse data samples in high dimensions. It is worth noting that although we have only shown samples of size $100-500$ in dimensions $20-30$ to be sparse enough to warrant our method, every finite sample size will become relatively sparse in sufficiently high dimensions due to the curse of dimensionality.

The next step for this work is to also consider other design optimality criteria, such as A, E, and D optimality and maximin or minimax distances in the formulation of our objective. While these criteria are not directly related to measure collapse, they are common design optimality criteria in the literature with a direct connection to surrogate model accuracy. We would also like to prove our method more rigorously by showing that we can obtain a slower rate of measure concentration for our method. Finally, we will need to show empirically that "good" designs generated through our method translate to better approximation performance downstream in real-world applications.

ACKNOWLEDGEMENT

We would like to thank Stefan Wild and Kesav Krishnan for their feedback regarding an earlier version of this work.

This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research's SciDAC program under Contract No. DE-AC02-06CH11357, the RAPIDS institute and by the National Science Foundation (NSF) Mathematical Sciences Graduate Internship (MSGI) Program sponsored by the NSF Division of Mathematical Sciences. This program is administered by the Oak Ridge Institute for Science and Education (ORISE) through an inter agency agreement between the U.S. Department of Energy (DOE) and NSF. ORISE is managed for DOE by ORAU. All opinions expressed in this paper are the authors' and do not necessarily reflect the policies and views of NSF, ORAU/ORISE, or DOE.

REFERENCES

Adeniran, A., O. Faweya, T. Ogunlade, K. Balogun et al. 2020. "Derivation of Gaussian Probability Distribution: A New Approach". *Applied Mathematics* 11(06):436.

Attia, A., and S. E. Ahmed. 2023. "PyOED: An Extensible Suite for Data Assimilation and Model-Constrained Optimal Design of Experiments". Technical report, arXiv preprint.

Balandat, M., B. Karrer, D. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy. 2020. "BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization". In *Advances in Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Volume 33, 21524–21538. Curran Associates, Inc.

Bratley, P., and B. L. Fox. 1988. "Algorithm 659: Implementing Sobol's Quasirandom Sequence Generator". *ACM Transactions on Mathematical Software* 14(1):88–100.

Chang, T. H., and S. M. Wild. 2023. "ParMOO: A Python Library for Parallel Multiobjective Simulation Optimization". *Journal of Open Source Software* 8(82):4468.

Custódio, A. L., and J. F. A. Madeira. 2018. "MultiGLODS: Global and Local Multiobjective Optimization Using Direct Search". *Journal of Global Optimization* 72(2):323–345.

Garnett, R. 2023. *Bayesian Optimization*. Cambridge University Press. to appear.

Garud, S. S., I. A. Karimi, and M. Kraft. 2017. "Smart Sampling Algorithm for Surrogate Model Development". *Computers & Chemical Engineering* 96:103–114.

Gorban, A. N., and I. Y. Tyukin. 2017. "Stochastic Separation Theorems". *Neural Networks* 94:255–259.

Hewitt, E. 1953. "Remarks on the Inversion of Fourier-Stieltjes Transforms". *Annals of Mathematics* 57(3):458–474.

Hickernell, F. J. 1998. "A Generalized Discrepancy and Quadrature Error Bound". *Mathematics of Computation* 67(221):299–322.

Joe, S., and F. Y. Kuo. 2003. "Remark on Algorithm 659: Implementing Sobol's Quasirandom Sequence Generator". *ACM Transactions on Mathematical Software* 29(1):49–57.

Joe, S., and F. Y. Kuo. 2008. "Constructing Sobol Sequences with Better Two-dimensional Projections". *SIAM Journal on Scientific Computing* 30(5):2635–2654.

Johnson, M., L. Moore, and D. Ylvisaker. 1990. “Minimax and Maximin Distance Designs”. *Journal of Statistical Planning and Inference* 26(2):131–148.

Jones, D. R., C. D. Perttunen, and B. E. Stuckman. 1993. “Lipschitzian Optimization without the Lipschitz Constant”. *Journal of Optimization Theory and Applications* 79:157–181.

Kuipers, L., and H. Niederreiter. 1974. *Uniform Distribution of Sequences*. Pure and Applied Mathematics. New York-London-Sydney: Wiley-Interscience [John Wiley & Sons].

Lai, T. L. 2003. “Stochastic Approximation”. *The Annals of Statistics* 31(2):391–406.

Lax, P. D. 2002. *Functional analysis*. Pure and Applied Mathematics (New York). New York, USA: Wiley-Interscience [John Wiley & Sons].

Lee, Abraham et al. 2015. “pyDOE: The Experimental Design Package for Python”.

Müller, J. 2017. “SOCEMO: Surrogate Optimization of Computationally Expensive Multiobjective Problems”. *INFORMS Journal on Computing* 29(4):581–596.

Myers, R. H., D. C. Montgomery, and C. M. Anderson-Cook. 2016. *Response Surface Methodology: Process and Design Optimization Using Designed Experiments*. 4 ed. Hoboken, NJ, USA: John Wiley & Sons, Inc.

Neveu, N., T. H. Chang, P. Franz, S. Hudson, and J. Larson. 2022. “Comparison of Multiobjective Optimization Methods for the LCLS-II Photoinjector”. Technical report, arXiv preprint.

Powell, M. J. D. 1994. “A Direct Search Optimization Method that Models the Objective and Constraint Functions by Linear Interpolation”. In *Advances in Optimization and Numerical Analysis*, edited by S. Gomez and J. P. Hennart, 51–67. Dordrecht: Springer Netherlands.

Pronzato, L. 2017. “Minimax and Maximin Space-filling Designs: Some Properties and Methods for Construction”. *Journal de la Société Française de Statistique* 158(1):7–36.

Roy, P. T., A. B. Owen, M. Balandat, and M. Haberland. 2023. “Quasi-Monte Carlo Methods in Python”. *Journal of Open Source Software* 8(84):5309.

Sapsis, T. P., and A. Blanchard. 2022. “Optimal Criteria and their Asymptotic Form for Data Selection in Data-driven Reduced-order Modelling with Gaussian Process Regression”. *Philosophical Transactions of the Royal Society A* 380(2229):20210197.

Shi, H.-J. M., M. Q. Xuan, F. Oztoprak, and J. Nocedal. 2021. “On the Numerical Performance of Derivative-Free Optimization Methods Based on Finite-Difference Approximations”. Technical report, arXiv preprint.

Sobol, I. M. 1967. “Distribution of Points in a Cube and Approximate Evaluation of Integrals”. *Ž. Vyčisl. Mat i Mat. Fiz.* 7:784–802.

Viana, F. A. 2016. “A Tutorial on Latin Hypercube Design of Experiments”. *Quality and Reliability Engineering International* 32(5):1975–1985.

Virtanen, P. et al. 2020. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. *Nature Methods* 17(3):261–272.

Wang, J., and A. W. Dowling. 2022. “Pyomo.DOE: An Open-source Package for Model-based Design of Experiments in Python”. *AIChE Journal* 68(12):e17813.

Wang, Y., L. Xu, Y. Hong, R. Pan, T. H. Chang, T. C. H. Lux, J. Bernard, L. T. Watson, and K. W. Cameron. 2023. “Design Strategies and Approximation Methods for High-Performance Computing Variability Management”. *Journal of Quality Technology* 55(1):88–103.

Wong, T.-T., W.-S. Luk, and P.-A. Heng. 1997. “Sampling with Hammersley and Halton points”. *Journal of Graphics Tools* 2(2):9–24.

AUTHOR BIOGRAPHIES

MANISHA GARG is a Ph.D. student in the Dept. of Mathematics at University of Illinois Urbana Champaign and a former NSF MSGI recipient at Argonne National Laboratory. Her interests include combinatorial group theory and multiobjective design optimization. Her email address is manisha8@illinois.edu.

TYLER H. CHANG (Ph.D., 2020, Virginia Tech) is a postdoctoral appointee in the Mathematics and Computer Science Division at Argonne, where he served as Manisha Garg’s host for her NSF MSGI appointment. His research interests include multiobjective design optimization, surrogate modeling, approximation theory, and scalable algorithms. His email address is tchang@anl.gov, and his website is https://thchang.github.io.

KRISHNAN RAGHAVAN (Ph.D., 2019, Missouri S.&T.) is an assistant computational mathematician in the Mathematics and Computer Science Division at Argonne. His research interests include statistics, machine learning, and dynamical systems. His email address is kraghavan@anl.gov and his website is https://krm9c.github.io/.